TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN ENXEÑARÍA DE COMPUTADORES

Aplicación de Modelos Generativos de Inteligencia Artificial en la Automatización de Procesos Contables: Desarrollo y Evaluación de una Herramienta Web Avanzada para Empresas

Estudante: Adrián Villegas Duque

Dirección: Pablo Alejandro Calviño Padín

Santiago Carrera Morodo

A Coruña, febrero de 2024.

A Biti

Agradecimientos

Me gustaría dedicar este proyecto a todas las personas que lo han hecho posible. Lo primero de todo, gracias a Pablo y Santi por confiar en mí, guiándome en este camino como tutores y amigos, potenciando mi creatividad y encauzando mis ideas. Muchas gracias también a Vento, por darnos esta gran oportunidad y por confiar tanto en mí.
También quiero agradecer a mis padres por darme la posibilidad de estudiar lo que me gusta y me apasiona y que me han apoyado incondicionalmente en esta carrera de fondo.
Por último, gracias a mis compañeros de carrera, que me han acompañado en los momentos más arduos. En especial a Mario quien ha sido un gran compañero de trabajo y, sobre todo, gran amigo.
Gracias a todos.

Resumen

Con el crecimiento de los nuevos Modelos de Lenguaje de Grandes Dimensiones (LLM, por sus siglas en inglés), nos encontramos ante una increíble oportunidad para estudiar su potencial a nivel empresarial.

Esta memoria detalla el desarrollo de una aplicación, en colaboración con Vento Abogados & Asesores. Dicha aplicación tiene como objetivo incorporar inteligencia artificial en un entorno empresarial real para automatizar la generación de asientos contables a partir de facturas. Este trabajo profundizará en los aspectos técnicos de su desarrollo, analizando al detalle la implementación de las funcionalidades y discutiendo el uso y definición de los LLM stacks.

Para terminar, se expondrán las conclusiones finales, se establecerán líneas de trabajo para el futuro y se revisarán en profundidad cuestiones de seguridad y ética relacionadas con el uso de estas tecnologías.

Abstract

With the growth of new Large Language Models (LLMs), we are presented with an incredible opportunity to study the potential of these models at a business level.

This report details the development of an application in collaboration with Vento Abogados & Asesores, focused on integrating artificial intelligence into a real business environment to automate accounting processes. This document will delve into the technical aspects of its development, thoroughly analyzing the implementation of functionalities and discussing the use of LLM stacks and their formal definition.

Finally, the report presents the conclusions, establishes future lines of work, and conducts an in-depth analysis of security and ethical issues related to the use of these technologies.

Palabras clave:

- Procesamiento de Facturas
- LLM Stack
- Contabilidad
- GPT
- Qdrant
- Embeddings
- Python

Keywords:

- Invoice Processing
- LLM Stack
- Accounting
- GPT
- Qdrant
- Embeddings
- Python

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1 Contexto

Nos encontramos en una era definida por un avance tecnológico acelerado, un período en el cual la Inteligencia Artificial (IA) y los modelos de lenguaje avanzados de gran escala, como ChatGPT, se han posicionado como potentes propuestas tecnológicas, abriendo un amplio y hasta ahora poco explorado abanico de posibilidades.

Con la aparición de los Modelos de Lenguaje de Grandes Dimensiones (LLM) de las últimas generaciones, parece necesario reconsiderar y actualizar los actuales *framework* utilizados para el análisis de grandes volúmenes de datos. Esta nueva tecnología, con su potente capacidad para procesar y analizar texto, tiene el potencial de mejorar significativamente el área empresarial, optimizando procesos, facilitando la toma de decisiones y abriendo caminos hacia innovaciones disruptivas en el sector. En particular, la aplicación de los LLM en el ámbito contable se presenta como un campo de estudio muy prometedor.

Tradicionalmente, la gestión de facturas ha presentado desafíos significativos debido a la necesidad de extraer y analizar información de manera precisa y eficiente. Las facturas, con su variedad de formatos y terminologías específicas, requieren un sistema capaz de comprender y procesar grandes volúmenes de texto y datos numéricos de manera coherente. Aquí es donde los LLM muestran unas posibilidades notables, al ofrecer soluciones avanzadas para estos desafíos tradicionales en la contabilidad.

Estos modelos pueden ser entrenados para reconocer y extraer información relevante de las facturas, independientemente de las variaciones en la presentación de los datos. Esta aplicación no solo mejora la precisión en la extracción de datos, sino que también acelera significativamente el proceso de contabilización de facturas, crucial en un entorno empresarial donde el tiempo y la precisión son esenciales.

En definitiva, en el actual contexto de revolución tecnológica, se presenta un desafío fundamental: el análisis exhaustivo de la amplia gama de capacidades y modelos disponibles en

este campo debido a su extensa variedad y complejidad. Por ello, resulta una tarea ardua abarcar todas las posibles aplicaciones de estos modelos. En consecuencia, este estudio ha seleccionado de manera deliberada el procesamiento de facturas en el ámbito contable como un área focal específica. Esta elección obedece a la necesidad imperante de investigar un caso práctico y realista, donde los impactos y beneficios de los LLM puedan ser medidos y evaluados de manera objetiva y concreta.

1.2 Objetivos de la investigación

Este trabajo busca examinar cómo los Modelos de Lenguaje de Grandes Dimensiones (LLM), representados en este caso por ChatGPT, pueden mejorar el procesamiento de facturas en la contabilidad. Se enfoca en la integración de estos modelos para aumentar la eficiencia y precisión, reducir la dependencia de la gestión manual y proponer soluciones prácticas a los retos existentes en el procesamiento de facturas.

Por tanto, el propósito de esta investigación es llevar a cabo un análisis riguroso y orientado a la aplicación práctica de cómo los LLM pueden ser implementados de forma efectiva en la contabilidad de una empresa. El estudio se enfocará en comprender la manera en que estos modelos avanzados pueden contribuir a la mejora de la eficiencia, la reducción de errores y la optimización de recursos en los procesos contables. Este enfoque práctico y orientado a un caso de uso real, además de aportar claridad sobre el potencial de los LLM en este sector específico, también servirá como un estudio de caso valioso para futuras investigaciones en el campo de la inteligencia artificial aplicada.

1.3 Estructura del trabajo

Esta sección tiene como objetivo presentar de forma resumida y clara la estructura seguida en esta memoria:

- Capítulo 1. Introducción: capítulo introductorio, en el cual nos encontramos ahora y que establece el proyecto en su contexto y sus objetivos finales.
- Capítulo 2. Descripción del Problema: este capítulo pretende definir el contexto técnico del proyecto e introducir conceptos esenciales para un correcto entendimiento de esta memoria.
- Capítulo 3. Planificación: este capítulo introduce los aspectos logísticos del proyecto. Se hablará de la metodología utilizada, se hará una breve descripción de los LLM stack y la planificación empleada. También se desglosarán los recursos y costes del proyecto.

- Capítulo 4. Análisis: en este capitulo se analizarán en profundidad los requisitos y casos de uso del sistema.
- Capítulo 5. Aplicación: este capítulo se adentra en una explicación más exhaustiva de la aplicación y de su funcionamiento. Además, se expondrán aspectos técnicos de la implementación.
- Capítulo 6. Seguridad y Ética: en este capítulo se profundizará en los campos de la seguridad y la ética en el uso de los LLM. Se analizará la seguridad de los modelos GPT y se reflexionará sobre su ética.
- Capítulo 7. Conclusiones: en este capítulo se comentarán los resultados del proyecto y se presentará una conclusión final acompañada de una reflexión sobre las perspectivas de futuro de este trabajo.

Capítulo 2

Descripción del problema

2.1 Fundamentos teóricos

En este capítulo se introducen los conceptos teóricos sobre los cuales se trabajará a lo largo de la memoria, de manera que se facilite una base fundamental para la comprensión del marco en el que se desarrolla el proyecto:

2.1.1 Principios básicos de la contabilidad

La contabilidad, una disciplina vital en el mundo empresarial, sirve como el lenguaje universal de los negocios, pues proporciona un marco para registrar, analizar y comunicar la información financiera. Como cualquier campo formal, la contabilidad se rige por unos estándares muy marcados, descritos por los Principios Contables Generalmente Aceptados, las Normas de Información Financiera y el Plan General Contable.

Los Principios Contables Generalmente Aceptados (PCGA) constituyen el pilar normativo de la contabilidad. Gracias a estos principios, que establecen los fundamentos de uniformidad y fiabilidad en el registro contable, existe una estandarización que permite la medición y comparación entre empresas y procesos contables dentro de la misma empresa. Además de los PCGA, en el Plan General de Contabilidad, aprobado por el Real Decreto 1514/2007, se establecen las directrices y normas en la clasificación y registro de las operaciones financieras, lo que asegura una representación precisa y homogénea. Bajo estas premisas, las Normas de Información Financiera desempeñan un papel crucial, no solo asegurando que la información financiera cumpla con los estándares de los PCGA y del Plan General de Contabilidad, sino también complementándolos al aportar transparencia, comprensibilidad y utilidad para los usuarios, inversores, reguladores y otros profesionales del ámbito.

2.1.2 Asientos contables

El asiento contable es una anotación clave en el libro de contabilidad de una empresa, que registra tanto las entradas como las salidas de recursos, así como la información relevante asociada a estos procesos.

En el marco del sistema de contabilidad de partida doble, cada asiento contable se representa mediante dos anotaciones: el debe y el haber. Esta estructura, basada en la oposición de los dos tipos de movimientos, afecta tanto al activo como al pasivo de la empresa. Se construye a partir de la premisa de que todo asiento o movimiento contable impacta en al menos dos cuentas contables, y que cada movimiento contable tiene una contrapartida.

La complejidad técnica inherente al proceso de asientos contables representa un desafío significativo. La correcta interpretación de las transacciones financieras y su adecuado registro requieren una comprensión profunda de los principios contables y la capacidad de aplicarlos de manera coherente. En el contexto de una empresa en crecimiento o con operaciones financieras complejas, este proceso puede volverse aún más desafiante, ya que aumenta el riesgo de errores y la carga de trabajo del departamento contable.

2.1.3 Cuentas contables

En los asientos contables, cada transacción se asocia con dos códigos numéricos que representan y registran las dos operaciones contrarias que típicamente aparecen en los movimientos contables. Un ejemplo práctico de este sistema sería el generado al realizar un pago a un proveedor. En este caso, por un lado, se registra una salida de dinero de la cuenta bancaria de la empresa y, por otro lado, se documenta la disminución de la deuda con el proveedor.

Estas cuentas forman parte del Plan General de Contabilidad, donde se define más a fondo los códigos numéricos de las cuentas y su relación con los elementos a los que hacen referencia. El conjunto de todas las cuentas contables utilizadas por una empresa es denominado Plan de Cuentas. La mayoría de las empresas adoptan el Plan de Cuentas estipulado por el Registro Mercantil y el Plan General de Contabilidad, aunque a menudo se personaliza para ajustarse a necesidades específicas y lograr un mayor control contable. El Plan contable proporciona una estructura y un lenguaje comunes para la categorización y el registro de transacciones financieras, de tal modo que se asegure así una representación homogénea y precisa de la realidad económica de la empresa.

2.1.4 Plan contable

Un plan de cuentas contables representa una estructura integral y ordenada de las cuentas usadas en la contabilidad de una empresa. Este listado es esencial para ilustrar la posición financiera de la organización, destacando tanto sus activos como sus pasivos. Las cuentas

reflejadas en este plan son aquellas seleccionadas por la empresa para documentar sus actividades contables de manera precisa,con una estructura basada en el Plan de Contabilidad General.

El objetivo principal de este plan es proporcionar un esquema sistemático de todas las cuentas implicadas en los procesos contables, alineándose con las directrices establecidas por el Servicio de Impuestos Internos. Este alineamiento se logra a través del seguimiento del Manual de Cuentas, que detalla las normas para el manejo adecuado de estas cuentas.

La relevancia de un plan de cuentas radica en su capacidad para estructurar y simplificar el sistema contable, brindando acceso claro a información financiera crucial. Este acceso facilita la comprensión de los activos de la empresa, que son recursos generadores de beneficio, y sus pasivos, que representan las obligaciones financieras con otras entidades.

Para ser eficaz, un plan de cuentas debe incorporar tres atributos esenciales: amplitud, para cubrir todas las operaciones del negocio; flexibilidad, para adaptarse a los cambios y evoluciones dentro de la empresa, lo cual incluye la incorporación de nuevas cuentas cuando sea necesario; y un sistema de codificación numérica que facilite la agrupación y clasificación ordenada de las cuentas.

Un plan de cuentas contable eficiente debe cumplir con ciertos criterios fundamentales. Debe ser completo, abarcando todas las cuentas necesarias para registrar cada transacción del negocio. Debe ser sistemático, al seguir un orden lógico que clasifique las cuentas según su naturaleza y propósito. Además, debe ser flexible, para permitir la adición o eliminación de cuentas según lo requieran las operaciones de la empresa. Por último, debe emplear una terminología clara y específica, donde cada nombre de cuenta refleje de manera precisa su función y contenido.

2.1.5 Tipos y estructura de asientos contables

Tipos de asientos contables

existen dos clasificaciones diferenciadas de los registros o asientos contables. Según su fondo:

- **Asiento Operativo:** refiere a las operaciones que la empresa ha realizado día a día a lo largo de un año.
- **Asiento de Apertura:** es aquel que da inicio al ejercicio contable.
- **Asiento de Ajuste:** se realiza al momento de la presentación del balance.
- **Asiento de Regularización:** a través de este asiento se clasifican adecuadamente las pérdidas y ganancias.

- **Asiento de Cierre:** con este asiento se finalizan las cuentas contables del ejercicio.

Según su forma:

- **Asientos Simples:** involucran una cuenta en el Debe y una en el Haber.
- **Asientos Compuestos:** implican más de una cuenta en el Debe y/o en el Haber.

En este proyecto nos centraremos en los asientos contables simples, en particular aquellos relacionados con los asientos de regulación generados por las facturas recibidas.

Estructura de los asientos contables

Los asientos contables constan de varios componentes clave:

- **Fecha:** indica el día en que se realiza la transacción.
- **Emisor:** especifica el nombre de la persona jurídica o empresa que emite la factura.
- **Código de Cuenta:** cada cuenta contable tiene un código único que la identifica. En un asiento contable, el número de cuentas puede variar según las características del movimiento contable. Generalmente, se encuentran al menos dos cuentas para el Debe y el Haber, además de una adicional para el IVA.
- **Descripción:** proporciona una breve explicación de la naturaleza de la transacción.
- **Debe y Haber:** dos columnas que registran los montos de la transacción, reflejando el principio de partida doble. El 'Debe' indica el destino de los recursos, y el 'Haber' su origen.
- **IVA:** al registrar una transacción, es crucial diferenciar el monto base del IVA, ya que cada uno se asienta en una cuenta distinta. Por ejemplo, en una compra, el valor del IVA se registra en una cuenta separada de IVA soportado, mientras que en una venta, se registra en una cuenta de IVA repercutido.
- **Retenciones:** se registran en cuentas específicas. Por ejemplo, en el pago de una factura con retención, el monto retenido se registra en una cuenta separada de retenciones a pagar, indicando la obligación de entregar ese importe a la autoridad fiscal.

2.1.6 Proceso de conversión de facturas a asientos contables

Cuando una factura es recibida, el contable inicia el proceso verificando la integridad del documento: revisa que todos los elementos esenciales estén presentes, como la fecha de emisión, la identificación del proveedor, el concepto de la factura, el importe neto, el IVA

aplicado y cualquier retención fiscal pertinente. Este paso es vital para asegurarse de que la factura cumple con los requisitos legales y contables.

El proceso se profundiza en la asignación de cuentas, una de las facetas más complejas de la contabilidad. Cada elemento de la factura debe asociarse a una cuenta contable específica, lo cual requiere del conocimiento especializado del contable. Este conocimiento no se limita solo al ámbito general de la contabilidad, sino que también abarca el contexto contable específico de la empresa. Esta asignación es crítica, ya que determina cómo se reflejará la transacción en los estados financieros y cómo afectará el análisis de costos y la planificación fiscal.

Por otro lado, el IVA y las retenciones fiscales requieren una atención especial. El contable debe registrar el o los IVAs soportados en una cuenta diferenciada, asegurando que el gasto refleje únicamente el costo neto del bien o servicio. Este paso es esencial para la recuperación o compensación del IVA en la declaración de impuestos. Del mismo modo, las retenciones fiscales deben ser cuidadosamente contabilizadas en cuentas de pasivo dedicadas, dado que representan fondos que la empresa retiene y posteriormente debe remitir a la autoridad tributaria.

2.2 Fundamentos tecnológicos

A continuación, se discutirán en profundidad los diferentes conceptos necesarios que sirvieron de base para la construcción del aplicativo web.

2.2.1 Large Language Models (LLM)

Los modelos de lenguaje de gran tamaño (LLM) [5], o LLMs por sus siglas en inglés 'Large Language Models', son modelos de aprendizaje profundo con grandes capacidades que se preentrenan con cantidades masivas de datos, llegando a centenas de *gigabytes*. El *transformer* subyacente, un conjunto de redes neuronales, consta de un codificador y un decodificador con capacidades de autoatención. Este mecanismo de autoatención permite al modelo enfocarse en diferentes partes de la entrada para generar una salida relevante. La atención se calcula utilizando puntuaciones que determinan la importancia que se debe dar a cada palabra durante la traducción o generación de texto. Por otro lado, el codificador y el decodificador interpretan el significado de una secuencia de texto y comprenden las relaciones entre las palabras y frases que contiene.

A diferencia de las redes neuronales recurrentes (RNN) anteriores, que procesaban las entradas de manera secuencial, los transformadores procesan secuencias enteras en paralelo. Esto permite a los científicos de datos utilizar GPU para entrenar LLM basados en transformadores, lo que reduce significativamente el tiempo de entrenamiento.

Los transformadores presentan una arquitectura basada en codificadores y decodificadores, como se puede apreciar mejor en la siguiente figura.

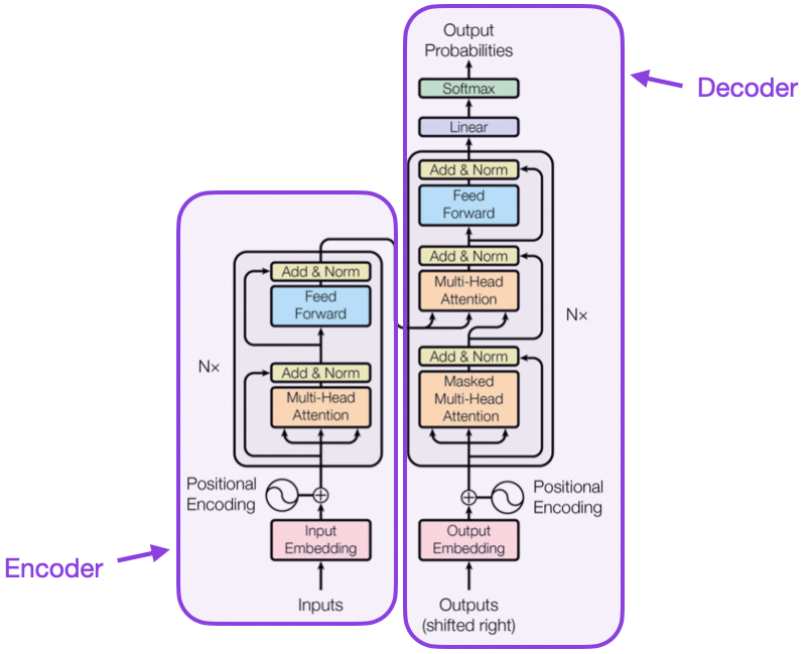


Figura 2.1: Arquitectura típica de los LLM [1]

Tokenización

Los LLM, entrenados en texto para predecir texto, utilizan la *tokenización* como una etapa esencial de preprocesamiento. Su objetivo es descomponer el texto en unidades básicas llamadas *tokens*, que pueden ser caracteres, subpalabras, símbolos o palabras, dependiendo del modelo. La *tokenización* transforma el texto en listas de números enteros de manera biyectiva, utilizando esquemas como la codificación de pares de *bytes*. Además, juega un rol importante en la compresión de texto, reduciendo la cantidad de cómputo necesario.

2.2.2 *Link Prediction*

La predicción de enlace o *link prediction* es una tarea en el análisis de redes que busca predecir conexiones faltantes o futuras entre nodos. Considerando una red

$$\mathbf{G} = (\mathbf{V}, \mathbf{E})$$

donde V son los nodos y E los enlaces verdaderos, el objetivo es identificar enlaces verdaderos no observados basándose en las conexiones existentes y la estructura de la red. En este

proyecto, se utiliza una búsqueda semántica basada en la similitud del coseno para la implementación de la predicción de enlaces.

2.2.3 Búsqueda Semántica

La búsqueda semántica se centra en comprender el significado y el contexto de las palabras en las consultas y los documentos, a diferencia de la búsqueda textual tradicional basada en coincidencias de palabras clave. Este enfoque permite una interpretación más profunda de las intenciones del usuario y el contenido de los documentos.

Embeddings

Los *embeddings* son representaciones vectoriales de palabras, frases o documentos que capturan aspectos semánticos y relacionales. Estos vectores, generados a través de modelos de Procesamiento del Lenguaje Natural, por sus siglas en inglés (NLP) y aprendizaje automático, permiten calcular la similitud semántica entre la consulta de búsqueda y los documentos, con el fin de facilitar la identificación de contenidos relevantes más allá de las coincidencias literales de términos.

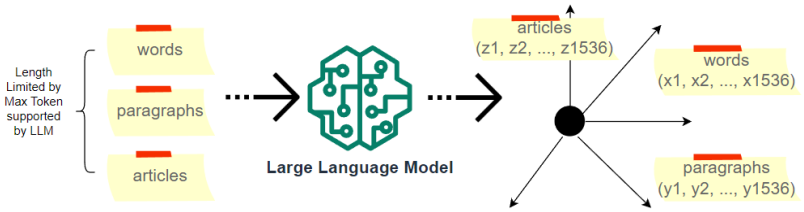


Figura 2.2: Proceso genérico de generación de *embeddings* usando LLM [2]

Cosine Similarity

La métrica de similitud coseno evalúa la similitud entre dos vectores dentro de un espacio de producto interno. Su cálculo implica determinar el coseno del ángulo existente entre ambos vectores, es decir, realizar el producto punto de los vectores y dividirlo por el producto de sus magnitudes. Es importante destacar que la similitud coseno no se ve influida por las magnitudes de los vectores, sino que depende exclusivamente de la orientación relativa de los mismos. Sus valores siempre se encuentran en el intervalo de $[-1, 1]$.Por ejemplo, dos vectores proporcionales tienen una similitud coseno de 1, mientras que dos vectores ortogonales obtienen una similitud de 0, y dos vectores opuestos resultan en una similitud de -1. En algunos casos, los componentes de los vectores no pueden ser negativos, lo que acota la similitud coseno en el rango de $[0, 1]$.

La *cosine similarity* de dos vectores no nulos se puede derivar utilizando la fórmula del producto punto euclidiano:

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\|\|\mathbf{B}\| \cos\theta$$

Dado dos vectores n-dimensionales de atributos, A y B, la similitud coseno, $\cos(\theta)$, se representa utilizando un producto punto y magnitud como:

$$\text{similitud coseno} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}},$$

donde A_i y B_i son las componentes i-ésimas de los vectores **A** y **B**, respectivamente.

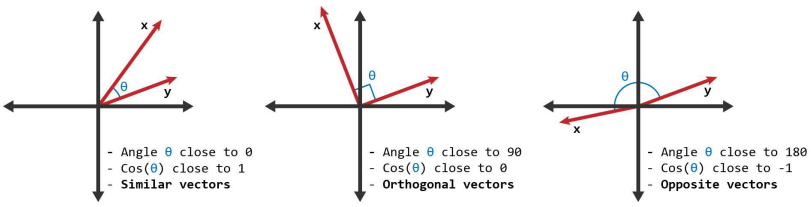


Figura 2.3: De izquierda a derecha: Vectores similares, Vectores ortogonales y Vectores contrarios [3]

Prompt Engineering

El *prompt engineering* es un campo interdisciplinar emergente en el dominio de la ***inteligencia artificial*** (IA), particularmente en la interacción con modelos de *procesamiento de lenguaje natural*(PLN). Este campo se centra en la metodología para formular y estructurar *prompts* o indicaciones, que son consultas o comandos diseñados para obtener respuestas específicas y precisas de sistemas de IA avanzados, como los modelos de lenguaje GPT-3 y GPT-4 de OpenAI. La eficacia de un *prompt* es determinante en la calidad de la información generada por estos modelos de IA, haciendo del *prompt engineering* una habilidad esencial para maximizar la utilidad y la precisión de las respuestas de la IA. El campo combina elementos de lingüística, psicología cognitiva y ciencias de la computación, y requiere un entendimiento profundo tanto de las capacidades como de las limitaciones de los modelos de lenguaje.

2.2.4 LLM Stacks

La evolución en el ámbito de los Modelos de Lenguaje de Gran Escala (LLM, por sus siglas en inglés) ha experimentado una notable expansión, impulsada por la introducción de avanzados LLM como GPT-3.5(2.3.3), GPT-4(2.3.3) y BERT(2.3.3). Esta expansión, acelerada

no solo en el nivel de desarrollo sino también en la producción y el sector empresarial, ha desencadenado una rápida transformación de los *stacks* tecnológicos previamente establecidos. Los nuevos modelos de procesamiento de lenguaje natural preentrenados han facilitado a los desarrolladores la creación de metodologías innovadoras y *stacks* más eficientes para su trabajo.

A pesar de que el término *LLM Stacks* aún no ha sido establecido formalmente, el amplio uso de estas tecnologías ha permitido el formar un conjunto de prácticas ya muy extendidas entre los desarrolladores. No obstante, estos modelos de lenguaje cuentan apenas con varios meses en el mercado, lo que desemboca en una falta de consenso que provoca que la información relacionada con los *LLM stacks* sea aún escasa.

Por lo tanto, en esta sección se pretende proporcionar un contexto y una descripción formal al término *LLM stacks*.

Contexto de los NLP Stacks

Anteriormente, la implementación de los *stacks* tecnológicos en el campo del Procesamiento del Lenguaje Natural (NLP) presentaba una serie de desafíos y limitaciones significativas. Estos *stacks*, cuya estructura se ilustra en la Figura 2.4, estaban especialmente diseñados y optimizados para tareas específicas de NLP, tales como la clasificación de textos y el reconocimiento y desambiguación de entidades nombradas. Generalmente, se componían de una secuencia de procesamiento de datos, un sistema de aprendizaje automático y múltiples bases de datos destinadas al almacenamiento de *embeddings* y datos estructurados.

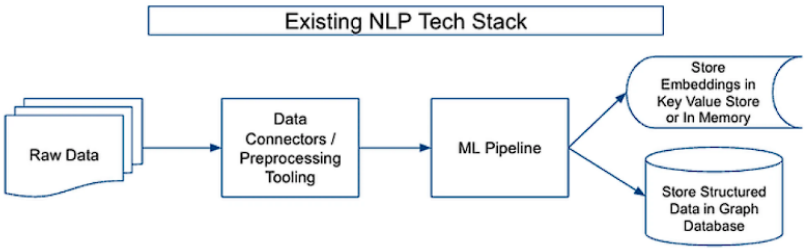


Figura 2.4: NLP stacks existentes

Aunque esta configuración era efectiva para generar una gran cantidad de información estructurada, como tríadas, *embeddings* de palabras y frases, y probabilidades de modelos de lenguaje, tenía importantes inconvenientes. Los desarrolladores solían almacenar estos datos en bases de datos como ElasticSearch, Postgres o Neo4j, formando un grafo de conocimiento accesible para los usuarios o servicios. A pesar de ser adecuados para producir datos estructurados fiables y su integración en sistemas empresariales para automatizar procesos clave, estos *stacks* presentaban problemas para lograr una adopción más amplia.

Uno de los principales retos era la lentitud en su implementación, debido a la necesidad de grandes volúmenes de datos etiquetados y un ajuste exhaustivo de los modelos. Además, resultaban costosos de mantener en funcionamiento, a menudo con más de tres docenas de modelos diferentes en un solo sistema. Otro aspecto problemático era su fragilidad frente a nuevos formatos de documentos y tipos de datos, lo que limitaba su capacidad de generalización y adaptación a contextos variados. Estas dificultades representaban un obstáculo significativo para su adopción generalizada y planteaban la necesidad de buscar enfoques más flexibles y robustos en el ámbito del NLP.

Diseño de los LLM Stacks

Desde la aparición de los nuevos modelos de lenguaje, se ha desarrollado una nueva estructura tecnológica diseñada para maximizar las capacidades de los Modelos de Lenguaje de Gran Escala (LLM). La principal diferencia con la estructura anterior radica en que la nueva no se basa tanto en grafos de conocimiento para almacenar datos estructurados, como lo hacían modelos anteriores. Esto se debe a que LLM como ChatGPT, Claude y BERT poseen una mayor cantidad de información codificada. Esta nueva estructura, cuyos detalles se ilustran en la Figura 2.5, se compone de cuatro componentes clave: un sistema de preprocesamiento de datos, un punto de integración y almacenamiento de vectores, interfaces para LLM y un marco de programación específico para LLM.

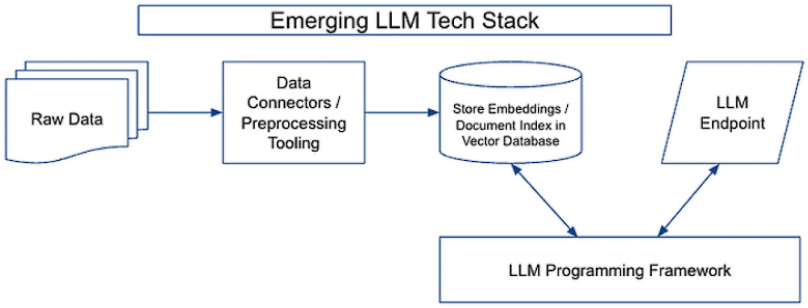


Figura 2.5: LLM Stacks

Estructura

- ***Canal de Preprocesamiento de Datos***: Este primer componente esencial de la nueva infraestructura tecnológica mantiene muchas similitudes con su versión anterior. Incluye elementos como conectores para la introducción de datos desde diversas fuentes y una capa para la transformación de datos, junto con conectores para bases de datos vectoriales.

En el pasado, este proceso se personalizaba manualmente para cada aplicación por científicos de datos, mediante la utilización de herramientas como OCR y numerosas expresiones regulares para preparar los datos para el aprendizaje automático. Actualmente, esta tarea se resuelve fácilmente mediante un preprocesado gracias a los LLM.

- ***Almacén de Vectores y Puntos Finales de Embedings***: La nueva arquitectura evoluciona significativamente en cómo se almacenan y acceden a los datos. Los *embeddings*, anteriormente utilizados para tareas específicas en casos muy concretos, ahora se almacenan como representaciones de los documentos en bases de datos vectoriales, de forma que se facilite la interacción con los LLM. La ventaja principal es almacenar *embeddings* en su forma original, permitiendo un procesamiento y recuperación de datos más rápidos. Esta estrategia mejora el manejo de grandes volúmenes de datos, simplifica el procesamiento en tiempo real (como en *chatbots*) y apoya técnicas avanzadas como el aprendizaje por transferencia. Además, los nuevos modelos LLM permiten generar estos *embeddings* por lo que la tarea se resuelve aún más fácilmente.

- ***LLM Endpoint***: Como tercer elemento clave, el punto final de LLM gestiona la entrada de datos y produce resultados. Es responsable de administrar recursos como memoria y procesamiento, y ofrece una interfaz escalable y robusta. Aunque hay varios tipos de *endpoints* de LLM, nos centramos en los dedicados a la generación de texto, simplificando la interacción a través de campos de texto para entrada y salida.

- ***Framework de Programación LLM***: El último pilar es un marco para desarrollar aplicaciones con modelos de lenguaje. Este *framework* servirá para aislar la lógica de negocio a arquitecturas más cercanas a las ya conocidas. Este marco facilitará al desarrollador aislar la comunicación con los LLM de la lógica de modelo tradicional, impulsando el desarrollo de nuevos sistemas más potentes gracias a embeber código con estructuras tradicionales en este nuevo ecosistema de los LLM.

Futuro de los LLM Stacks

Desde la entrada de la Web 3.0, hemos vivido un avance acelerado hacia el diseño de la comunicación como servicios distribuidos, un concepto que concuerda con los nuevos diseños de sistemas que estamos empezando a ver. Compañías como Zapier o Microsoft están apostando por un diseño orientado a los datos y a la integración de los LLM. A pesar del poco tiempo transcurrido y de los escasos ejemplos que tenemos a nuestra disposición, ya es posible observar que este tipo de diseños permite explotar al máximo nuestros esquemas antiguos.

En definitiva, estos *stacks* se establecen como interesantes respuestas para aprovechar al máximo las nuevas tecnologías emergentes.

2.2.5 Programación de Lenguaje Natural

La programación ha evolucionado de manera significativa a lo largo de los años, y en la actualidad, estamos siendo testigos de un nuevo paradigma en este campo gracias a los LLM . Estos modelos están revolucionando la forma en que interactuamos con el *software*, al permitir el uso del lenguaje natural para su control, en lugar de depender únicamente del código tradicional.

LangChain es un ejemplo de la génesis de estos avances. Se trata de un *framework* que está catalizando este cambio de paradigma al priorizar el uso del lenguaje y la gestión semántica de la información. Como se aprecia mejor en la Figura 2.6, este enfoque se centra en almacenar la información de manera semántica y utilizar el lenguaje natural par gestionarlo, gracias a los LLM ya entrenados

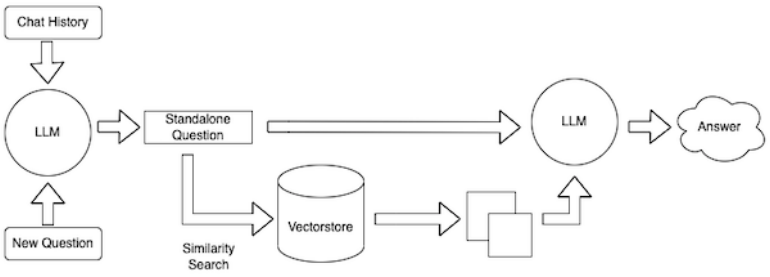


Figura 2.6: Diagrama típico de un paso de recuperación[4]

Esta aproximación refleja un cambio significativo en la interacción humano-computadora, ya que se acerca más a cómo los humanos procesan y utilizan el lenguaje. En lugar de utilizar sintaxis de programación rígida y específica, los usuarios ahora pueden interactuar con el *software* utilizando instrucciones basadas en el lenguaje natural.

Pero LangChain no es el único ejemplo de esta nueva forma de programación. El siguiente trabajo [6] lleva a cabo un estudio exhaustivo de la planificación de procesos con un LLM utilizando el lenguaje natural(*prompts*). Como se comenta en el propio artículo, con los modelos LLM actuales: ”para completar tareas complejas, todavía necesitamos un plan para la tarea que guíe a los LLM para generar las soluciones específicas paso a paso. Los LLM pueden generar directamente planes de tareas, pero estos planes aún pueden contener errores fácticos o estar incompletos.”

Para resolver esta planificación, el trabajo propone una solución de dos pasos. En la primera fase se actualiza de forma iterativa el plan generado gracias a la evaluación de los resultados de los *prompts* a partir del cálculo de error de la respuesta. Por otro lado, en la segunda fase se traslada el plan generado en la primera fase a un conjunto de pruebas. Gracias a este mé-

todo, el estudio no solo genera un plan óptimo para el problema, sino que se demuestra que el mismo plan es capaz orquestar otros LLM para mejorar su actuación.

En conclusión, el desarrollo de los LLM están marcando el comienzo de una era donde el lenguaje natural y la comprensión semántica se encuentran en el centro de la programación. A pesar de que las herramientas de desarrollo de software actuales nos dan un marco de programación, nos encontramos aún en un contexto sin metodologías claras para el buen uso de estos *frameworks*, los cuales llegarán con el tiempo.

2.3 Estado del arte de herramientas de facturación

El procesamiento automático de facturas o de cualquier elemento de la contabilidad, a pesar de consistir en una tarea cuanto menos necesaria y útil, sufre de una dificultad técnica muy alta difícilmente paliable con las tecnologías existentes hasta la fecha. A pesar de que, gracias a esta *primavera de la IA* que estamos viviendo, actualmente muchas herramientas nuevas surgen proponiendo soluciones novedosas y diferentes, el mercado de las herramientas de automatización contable peca de ser escaso y con soluciones poco eficientes. Aun así, a continuación, se repasarán a algunas de las herramientas más importantes y más utilizadas en este ámbito.

2.3.1 Ubyquo

Ubyquo, como herramienta de automatización, integra módulos variados para optimizar distintos procesos empresariales. Sus funciones abarcan desde el procesamiento automático de archivos internos hasta la centralización de un portal para interacción con clientes y la gestión automática de pedidos. No obstante, su módulo de gestión de facturas, aunque se presenta como semiautomático, evidencia limitaciones significativas. Aunque la herramienta puede manejar facturas estándar automáticamente, en situaciones de mayor complejidad depende de la intervención manual de un equipo humano, lo que incrementa los costos operativos y reduce la eficiencia del proceso. Además, su dependencia de la intervención humana en ciertas situaciones pone en duda la eficacia de su automatización y puede limitar su utilidad en entornos empresariales que requieren una gestión de facturas más ágil y completamente automatizada.

2.3.2 Sage

Sage es uno de los *softwares* más importantes de gestión contable y empresarial en general. Esta herramienta abarca todos los procesos contables que una empresa pueda tener y trabaja con ellos en un entorno controlado y centralizado con el objetivo de facilitar la labor. A pesar de que el programa como tal no ofrece una solución implícita a este problema, el *software*

se comporta como un entorno versátil para el cual existen una gran variedad de *plugins* y librerías de automatización de procesos. Si bien el sistema, sobre todo la parte de base de datos, se presenta como una solución fácil y rápida de utilizar, las APIs y dll son muy antiguas y están obsoletas, por lo que construir soluciones con ellos es casi inviable.

2.3.3 Modelos LLM

GPT-3.5

Este modelo es una versión refinada de GPT-3, desarrollado por OpenAI. Se lanzó en enero de 2022 y tiene variantes con 1.3B, 6B y 175B parámetros. GPT-3.5 se enfoca en reducir la generación de respuestas tóxicas y utiliza una técnica conocida como Aprendizaje por Refuerzo con Retroalimentación Humana, por sus siglas en inglés (RLHF) para mejorar la alineación del modelo con los valores humanos. Se caracteriza por su capacidad para generar texto coherente y versátil, aplicable en una variedad de tareas de procesamiento del lenguaje natural.

GPT-4

Lanzado por OpenAI, GPT-4 es una evolución significativa de GPT-3.5, ofreciendo mayor fiabilidad, creatividad y una mejor capacidad para manejar instrucciones más matizadas. GPT-4 puede procesar tanto texto como imágenes, y es conocido por su desempeño excepcional en pruebas académicas y por su capacidad para generar contenido en diversos formatos y estilos. Aunque OpenAI no ha revelado el tamaño exacto del modelo, se calcula que tiene alrededor de 1.76 billones de parámetros. GPT-4 también mejora en la ”alineación” del modelo, es decir, su habilidad para seguir instrucciones más precisas y generar respuestas más relevantes.

BERT

Desarrollado por Google, BERT (Bidirectional Encoder Representations from Transformers) es un modelo *open source* pionero en el ámbito del procesamiento de lenguaje natural. La principal innovación de BERT radica en su habilidad para analizar el contexto de palabras en oraciones de manera bidireccional, lo cual representó un avance significativo en el entendimiento del lenguaje natural. BERT se ha implementado ampliamente en aplicaciones que requieren comprensión del lenguaje, incluyendo motores de búsqueda y sistemas de respuesta a preguntas, debido a su capacidad para procesar el lenguaje de manera más efectiva y con un entendimiento contextual profundo.

LLaMA-2

Desarrollado por Meta, anteriormente conocido como Facebook, LLaMA-2 representa una avanzada serie de modelos *open source* de lenguaje autoregresivos, fundamentados en la ar-

quitectura de transformadores. La gama de estos modelos varía en complejidad, puede abarcar desde 7 mil millones hasta 70 mil millones de parámetros. Este modelo fue meticulosamente entrenado utilizando una vasta colección de 2 billones de *tokens*, derivados de diversas fuentes de dominio público. La ingeniería detrás de LLaMA-2 se enfocó en optimizar tanto la seguridad como la utilidad en contextos de diálogo y generación de lenguaje natural. Uno de los logros más notables de LLaMA-2 es su rendimiento superior en una gran variedad de tareas de procesamiento de lenguaje natural, con un énfasis particular en mejorar la coherencia y consistencia en diálogos de múltiples sesiones, hasta mantener un seguimiento riguroso de las instrucciones proporcionadas desde el inicio de las interacciones.

Capítulo 3

Planificación

3.1 Aproximación metodológica

Para este proyecto, se adoptó una metodología ágil, específicamente Scrum, con el fin de garantizar una ejecución flexible y adaptable a los diferentes retos y circunstancias. Se mantuvo una comunicación fluida y constante con los tutores y se organizaron reuniones periódicas, denominadas *sprints*, para evaluar el progreso y determinar las próximas acciones. Este enfoque iterativo y objetivo se refleja en el esquema de la Figura 3.1.

Cada *sprint* concluía con una evaluación exhaustiva y una retrospectiva minuciosa, lo que permitía identificar tanto logros como desafíos. Este análisis era fundamental para el aprendizaje continuo y la mejora progresiva del proyecto. Además, servía para establecer los objetivos del siguiente *sprint*, que se ajustaban a partir de los resultados obtenidos y las lecciones aprendidas previamente.

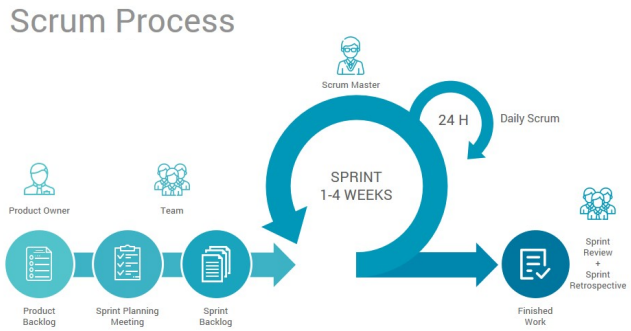


Figura 3.1: Diagrama de enfoque Scrum

3.2 Fases del proyecto

Debido al carácter de análisis del proyecto, no se realizó una planificación previa. Por el contrario, el uso de metodologías hábiles permitió establecer objetivos y tareas en cada *sprint*. A continuación se describirá en profundidad cada una de las fases realizadas para la conclusión del sistema.

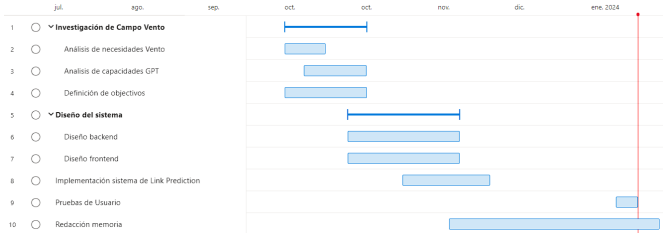


Figura 3.2: Diagrama de Gantt del proyecto

3.2.1 Fase 1: estudio de Campo

Durante la primera fase del proyecto el objetivo final de la herramienta aun no estaba definido, el único objetivo era el de analizar las capacidades y oportunidades de un modelo como GPT en un entorno profesional. Por ello, en esta primera fase se realizó un estudio de campo donde se intentó analizar en profundidad el flujo de trabajo de Vento Abogados & Asesores y a su vez estudiar las capacidades de GPT para poder establecer los objetivos finales.

Estudio del trabajo realizado en Vento

Durante esta primera fase, el trabajo estuvo orientado a un estudio de las herramientas de trabajo y procesos llevados a cabo en el despacho. Esta fase tenía como objetivo final analizar los campos o procesos de la oficina donde la implementación de la inteligencia artificial supusiese un avance para el bufete.

Estudio capacidades GPT

En paralelo con la tarea anteriormente descrita, se llevó a cabo un estudio del funcionamiento y capacidades de los modelos de GPT que había disponibles en ese momento. Este fase tuvo como resultado un conjunto de varias herramientas que plantearon la base técnica y la estructura inicial de las funcionalidades utilizadas en la herramienta final.

3.2.2 Fase 2: desarrollo

El desarrollo de la aplicación estuvo separado en dos etapas principales, las cuales coinciden con los dos requisitos funcionales del aplicativo.

Implementación del extractor de conocimiento

El primer requisito funcional al que se atendió y que actúo como criterio de evaluación de la capacidad a futuro de la herramienta. Durante esta fase se buscó un *prompt* con el que se obtuviesen resultados constantes y consistentes, así como la generación de todas las APIs necesarias para la extracción de texto de los PDF.

Diseño e implementación del sistema de *Link Prediction*

Otro de los objetivos principales fue el diseño e implementación del sistema de *Link Prediction*. Gracias al estudio previo del problema y sus alternativas, así como de la generación de varios prototipos iniciales, se pudo asentar previamente las bases de esta fase, lo que permitió un ágil desarrollo de la funcionalidad.

3.2.3 Fase 3: implementación en el despacho

Durante esta fase se llevó a cabo un proceso de prueba y validación del uso real de la herramienta en un entorno empresarial real. Para ello se escogió un pequeño grupo de pruebas dentro de la oficina de Vento Abogados & Asesores. Una vez realizada las pruebas se facilitó una encuesta para el análisis de resultados.

3.2.4 Fase 4: redacción de la memoria

En esta última fase, se documentarán todos los resultados, metodologías y conclusiones en una memoria escrita. Esta memoria detallará los objetivos, las fases del proyecto, los modelos implementados y las evaluaciones realizadas, así como las posibles aplicaciones y extensiones futuras del trabajo.

3.3 Recursos

Durante el desarrollo del sistema, ha sido usado una gran variedad de recursos esenciales que han permitido llevar a cabo las tareas de investigación, desarrollo y evaluación. Estos recursos pueden ser divididos en las siguientes categorías:

3.3.1 Recursos de *software*

En cuanto a los requisitos de *software*, el proceso de desarrollo ha necesitado del uso de los siguientes recursos de *software*.

Python

El *backend* en su totalidad ha sido construido en python 3.11 [7]. Para los *endpoints* de la aplicación se ha decidido usar la librería Flask la cual ha sido implementada con tecnología de *multi-threading* gracias a la librería Waitress.

Se ha utilizado las librerías PDFPlumber para la extracción de texto de los *PDF*, así como la librería PDF2Image y Poppler para la transformación de *PDF* en imágenes y la librería Pandas para la gestión de archivos.

Otras librerías usadas por el *backend* han sido la API de OpenAI para el diseño de la comunicación con chatGPT, las bibliotecas de OCR de Azure para la extracción de texto de imágenes y las librerías de comunicación con Qdrant y Numpy para el control de *embeddings* y vectores.

React

En el proceso de desarrollo del frontend del proyecto, se ha optado por utilizar React [8], una poderosa biblioteca JavaScript para la creación de interfaces de usuario dinámicas y modernas, conocida por su eficiencia y facilidad de uso.

Para maximizar la productividad y acelerar el desarrollo, se ha aprovechado las capacidades ofrecidas por Vite, una herramienta de construcción rápida que ha permitido optimizar el flujo de trabajo. Gracias a Vite, se ha podido beneficiar de una rápida recarga en tiempo real, lo que ha agilizado significativamente la iteración y la depuración del código.

En cuanto al diseño, se ha confiado en Chakra UI para dotar a la aplicación de una estética atractiva, coherente y responsive. Chakra UI es una librería de componentes de interfaz de usuario que se integra perfectamente con React, lo que ha permitido construir interfaces de usuario elegantes y altamente personalizables de manera eficiente. Además, la documentación detallada y la facilidad de uso de Chakra UI han permitido ahorrar tiempo y esfuerzo en la implementación de diseños atractivos y accesibles.

Jupyter Notebook

Jupyter Notebook[9] es una aplicación web interactiva diseñada para la creación y compartición de documentos que incorporan código en vivo, ecuaciones, visualizaciones y texto explicativo. Conocidos como *notebooks*, estos documentos son ampliamente utilizados en campos como la ciencia de datos, el análisis de datos, la educación en programación y otras áreas de investigación y enseñanza.

Esta herramienta forma parte del proyecto Jupyter, una iniciativa de código abierto dedicada al desarrollo de *software* libre, estándares abiertos y servicios para la computación interactiva en una amplia gama de lenguajes de programación. Aunque inicialmente se creó

para Python, Jupyter ha evolucionado para admitir múltiples lenguajes mediante sus *kernels*. Estos *kernels* son extensiones que habilitan la ejecución de código en distintos lenguajes de programación.

API REST

Una API REST (Representational State Transfer API) es un estilo de arquitectura de *software* utilizado principalmente en sistemas distribuidos y aplicaciones web. Esta arquitectura se fundamenta en los principios REST propuestos por Roy Fielding en 2000 y opera sobre el protocolo HTTP.

En una API REST, cada interacción es independiente, y cada petición HTTP contiene toda la información necesaria para su procesamiento, lo que la convierte en una petición sin estado. Esto significa que el servidor no guarda información sobre el estado del cliente entre distintas peticiones. La comunicación se realiza a través de una interfaz uniforme, que incluye el uso de URIs estándar para identificar recursos y el uso de métodos HTTP estándar (GET, POST, PUT, DELETE) para realizar operaciones.

La arquitectura REST permite un sistema de capas, donde cada una puede proporcionar funcionalidades adicionales, como balanceo de carga o almacenamiento en caché, sin alterar la comunicación directa entre cliente y servidor. Los recursos, ya sean datos o funcionalidades, se acceden a través de URIs y generalmente se representan en formatos como JSON o XML. Adicionalmente, la API REST puede ofrecer código bajo demanda, como JavaScript, para extender la funcionalidad del cliente.

La popularidad de la API REST se debe a su simplicidad, escalabilidad y su capacidad para integrarse fácilmente con tecnologías *frontend*.

OpenAI API

Para el correcto desarrollo de la aplicación, así como para su funcionamiento, ha sido necesaria una cuenta con acceso a la API de chatGPT[10] así como saldo en esta misma para el uso de la API. De esta API se han usado varios de los servicios que ofrece:

1. **GPT-4**: ultimo LLM diseñado por OpenAI con grandes capacidades que se diferencia de otros modelos por su capacidad para realizar tareas más complejas.

2. **GPT-3.5 Turbo**: LLM diseñado por OpenAI que se caracteriza por permitir contextos de gran tamaño y ofrece una baja latencia en la comunicación con el modelo.

3. **Embeddings**: modelo específico diseñado por OpenAI que permite la generación eficiente de *embeddings*.

Azure

Para el correcto funcionamiento de la extracción de texto de la herramienta precisamos la utilización del OCR de Microsoft Azure. Para ello fue necesario el acceso a una subscripción del mismo.

Git

Git [11] es un *software* de control de versiones de gestión distribuido, lo que quiere decir que genera copias locales del repositorio sobre las que se trabaja directamente. Una de sus características más interesantes es que no es necesario tener conectividad a la red, excepto para sincronizar el repositorio, y que es más robusto frente a los errores.

Github Desktop

GitHub Desktop [12] es la aplicación de escritorio de GitHub. Es una Interfaz Gráfica de Usuario (GUI, por sus siglas en inglés) diseñada para facilitar el uso de Git, ofrece las mismas características que este, pero es más visual y permite reducir la curva de aprendizaje.

Visual Studio Code

VSCode [13] es una IDE (Integrated Development Enviroment) diseñado para facilitar el trabajo de los programadores al desarrollar proyectos *software*. Es un entorno de desarrollo integrado, de código abierto y multiplataforma. Contiene un compilador e intérprete, ofrece auto-completado inteligente de código, un *debugger*. Además es uno de los IDEs de código abierto más populares y contiene una gran cantidad de *plugins* disponibles.

Docker

Docker[14] es una plataforma de contenedores de código abierto que permite a los desarrolladores y administradores de sistemas automatizar el despliegue de aplicaciones en entornos ligeros y portátiles, conocidos como contenedores. Estos contenedores son entornos de ejecución independientes y aislados que comparten el núcleo del sistema operativo subyacente, pero operan como instancias discretas. Docker utiliza tecnologías de aislamiento de recursos del sistema operativo, como *cgroups* y espacios de nombres del *kernel* de Linux, para permitir que múltiples contenedores se ejecuten en un solo sistema sin interferencia. Cada contenedor tiene su propia pila de *software*, bibliotecas y archivos, lo que garantiza la consistencia del entorno a través de diferentes infraestructuras de desarrollo y producción.

En este proyecto se ha utilizado una imagen del *docker* presentado por Qdrant para la gestion de indices de vectores.

Qdrant

Qdrant[15] es un sistema de gestión de bases de datos avanzado, enfocado en la indexación, búsqueda y almacenamiento eficiente de vectores, características que lo hacen especialmente adecuado para aplicaciones de inteligencia artificial y aprendizaje automático. Este *software* de código abierto se destaca en el ámbito de la búsqueda semántica y en la implementación de sistemas de recomendación, gracias a su capacidad para manejar grandes volúmenes de datos vectoriales en espacios de alta dimensión.

Una de las principales fortalezas de Qdrant radica en su arquitectura, optimizada para vectores. Esto permite a los usuarios realizar búsquedas basadas en la similitud de los vectores, un enfoque fundamental en aplicaciones que dependen de la comparación y análisis de datos complejos, como el procesamiento de lenguaje natural o la identificación de patrones en grandes conjuntos de datos.

Qdrant ofrece una interfaz flexible y eficiente para la gestión de vectores. Su diseño está centrado en garantizar una alta velocidad en las consultas y una eficiente utilización de los recursos de almacenamiento, lo cual es crítico para aplicaciones que requieren un procesamiento rápido y eficiente de grandes cantidades de datos. Además, la capacidad de Qdrant para realizar búsquedas aproximadas de vecinos más cercanos, Red de Neuronas Artificial, por sus siglas en inglés (ANN), lo hace particularmente valioso para tareas de recuperación de información y clasificación en contextos donde las relaciones entre los datos no son inmediatamente evidentes.

Otra característica notable de Qdrant es su escalabilidad. Está diseñado para crecer con las necesidades del usuario, permitiendo la expansión y el manejo eficiente de bases de datos vectoriales a medida que aumentan en tamaño y complejidad. Esta prestación lo convierte en una solución ideal para empresas y organizaciones que manejan grandes volúmenes de datos y requieren una solución escalable que se adapte a sus necesidades cambiantes.

Zotero

Zotero [16] es un *software* de gestión de referencias bibliográficas y datos de investigación que sobresale por su naturaleza de código abierto y su versatilidad de uso en distintas plataformas. Diseñado para asistir tanto a investigadores individuales como a grupos académicos, Zotero ofrece una solución integral para la organización de material bibliográfico y datos de investigación.

Una de las características más destacadas de Zotero es su capacidad para detectar automáticamente contenido en navegadores web, facilitando la recopilación de referencias de una variedad de fuentes en línea como catálogos de bibliotecas, bases de datos académicas y sitios web generales. Esta funcionalidad se extiende a través de extensiones de navegador para Google Chrome, Mozilla Firefox y Safari.

En términos de gestión de referencias, Zotero permite a los usuarios organizar su bibliografía en colecciones personalizables y etiquetarlas para una fácil navegación. Además, soporta una amplia gama de estilos bibliográficos y se integra con software de procesamiento de texto como Microsoft Word y LibreOffice, permitiendo insertar citas y generar bibliografías automáticamente en documentos.

Microsoft Visio

Microsoft Visio es una aplicación de *software* que proporciona herramientas para la creación de diagramas y representaciones gráficas. Forma parte del conjunto de programas de Microsoft Office, aunque no está incluido en todos los paquetes de esta *suite*. Visio es ampliamente utilizado en entornos empresariales y profesionales por su capacidad para facilitar la visualización de procesos complejos, sistemas, y redes de información.

Con Visio, los usuarios pueden crear diagramas de flujo, organigramas, planos de planta, diseños de red, modelos de procesos de negocios, y muchas otras formas de representaciones visuales. Ofrece una amplia gama de plantillas y formas predefinidas que ayudan a los usuarios a empezar rápidamente y a personalizar sus diagramas según las necesidades específicas. Además, Visio permite la colaboración en tiempo real y la integración con otras aplicaciones de Microsoft, como Word y Excel, lo que mejora su funcionalidad y utilidad en proyectos colaborativos y la gestión de la información.

3.3.2 Recursos Materiales

Entre los recursos materiales utilizados se encuentra el ordenador portátil proporcionado por Vento Abogados y Asesores. Este portátil está equipado con las siguientes características:

Componente	Modelo
Procesador	Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz
RAM	8,00 GB
Sistema	Windows 10

Tabla 3.1: Especificaciones portátil

3.3.3 Recursos Humanos

Debido al carácter multidisciplinar del proyecto, en su desarrollo se ha contado con variedad de profesionales con diferentes tareas. Entre los profesionales implicados en el proyecto:

1. **Jefes de proyecto**: un profesor y un tutor en Vento Abogados & Asesores para establecer los objetivos a corto y largo plazo y guiar la investigación.
2. **Desarrollador**: alumno encargado de la programación, de la implementación del proyecto y de la toma de decisiones necesarias sobre el diseño y arquitectura del mismo, así como de la realización de los experimentos pertinentes.
3. **Contable**: profesional contable de Vento, que participó en el proyecto como experto en contabilidad, supervisando el desarrollo en el apartado de la contabilidad.

3.4 Costes

A continuación se presenta un desglose de los gastos de los diferentes recursos utilizados. Debido al carácter colaborativo con la empresa de Vento Abogados y Asesores para la realización del proyecto, los cálculos de los costes se presentan como los costes reales del proyecto.

3.4.1 Costes materiales

El coste total del uso de la Interfaz de Programación de Aplicaciones (API) de GPT ha sido de 23.10 dólares americanos. A continuación, se mostrará de forma desglosada el coste total de cada mes de la API, así como un desglose mas exhaustivo de los servicios utilizados.

Costes por meses

Mes	Coste
Octubre	1.28 $
Noviembre	13.14 $
Diciembre	8.67 $

Tabla 3.2: Coste de la API de OpenAI por meses

Costes por servicio

A continuación se presentan las figuras de los costes de cada servicio.

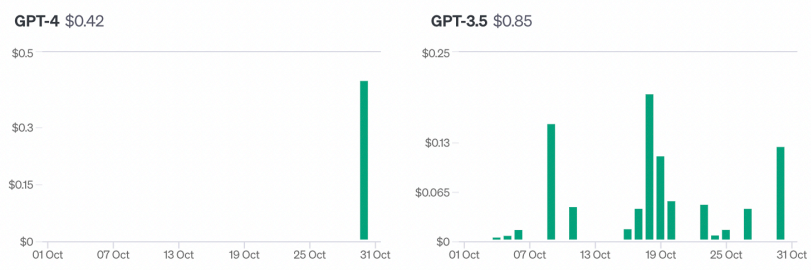


Figura 3.3: Gastos GPTs mes de Octubre

Figura 3.4: Gastos embeddings mes de Octubre

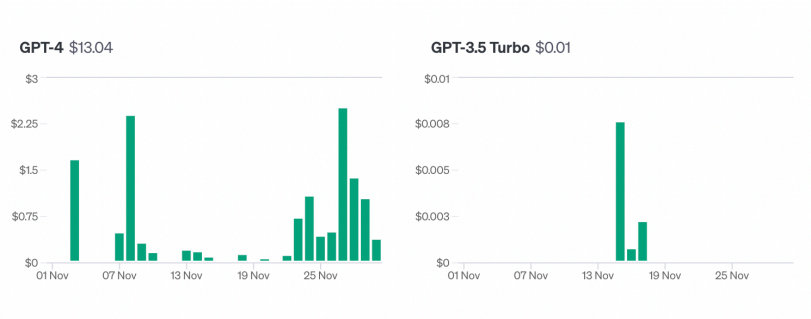


Figura 3.5: Gastos GPTs mes de Noviembre

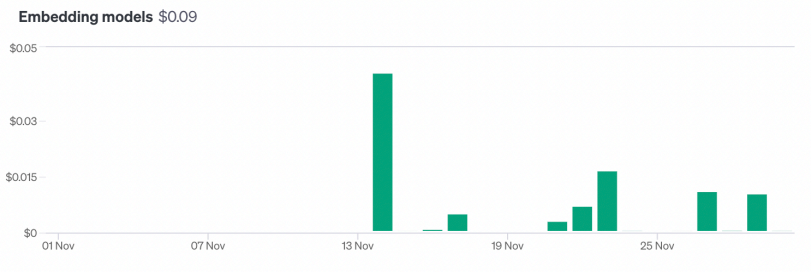


Figura 3.6: Gastos embeddings mes de Noviembre

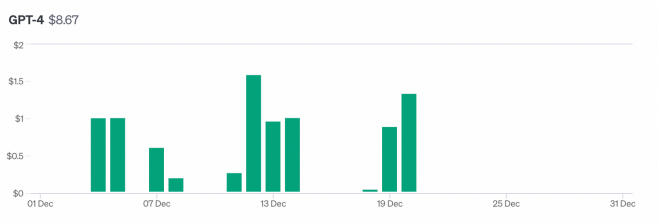


Figura 3.7: Gastos GPTs mes de Diciembre

3.4.2 Costes humanos

Los siguientes datos de la Tabla 3.3 han sido calculados a partir de los datos reales del desarrollo del proyecto. El cálculo de los costes de los desarrolladores se ha realizado a partir del contrato del convenio de investigación realizado entre la Universidad de Coruña y la empresa Vento Abogados & Asesores. Tomando en cuenta el período de trabajo, de octubre a febrero, y considerando las 17 horas de trabajo semanales, se puede calcular el coste del desarrollador a 8.7 euros la hora para un total de 340 horas netas. En cuanto a los costes de los jefes de proyecto, para el cálculo de Pablo Alejandro Padín, la información fue recuperada de la Tabla de Retribuciones para el personal docente de la UDC, USC y UDV. Por otro lado, para el cálculo de Santiago Carrera, se utilizaron los datos de costes de Vento Abogados & Asesores. Teniendo en cuenta las reuniones previas, la revisión de *sprints* cada dos semanas con una duración aproximada de media hora y las revisiones finales, se estima un total de 20 horas (entre ambos).

	Número	**Coste por hora**	**Horas**	**Total**
Jefe proyecto Universidad	1	19 €	10	190 €
Jefe proyecto Vento	1	150 €	10	1500 €
Experto contable	1	19	30	570 €
Desarrollador	1	8.7 €	340	2958 €
				5218 €

Tabla 3.3: Costes de los recursos humanos involucrados.

Capítulo 4

Análisis

4.1 Análisis de requisitos

En esta sección se analizarán en profundidad la los requisitos funcionales y no funcionales diseñados para este proyecto.

Antes de nada cabe destacar que, debido al desarrollo atípico de este proyecto, el análisis de requisitos fue variando a lo largo de los *sprints* hasta dar con una análisis completo y final.

4.1.1 Requisitos funcionales

Para el diseño de los requisitos funcionales, se realizó un estudio previo durante la Fase 1 del proyecto. En esta fase se celebraron diversas reuniones con diferentes departamentos dentro de Vento Abogados & Asesores con el objetivo de estudiar a fondo las posibles necesidades y oportunidades de una herramienta basada en modelos LLM a nivel empresarial. Definidas las primeras funcionalidades y realizadas las pruebas para medir las capacidades de los modelos, se establecieron, finalmente, los requisitos funcionales.

Requisitos	Descripción
Cargar facturas en PDF	El usuario debe ser capaz de subir una lista de facturas en formato PDF al sistema.
Procesar Facturas	El usuario puede procesar el conjunto de facturas previamente subido y procesar estas, generando un asiento contable.
Generar predicción de cuentas contables	El sistema generará una predicción de los códigos de cuentas correspondiente a cada factura.
Validar datos	El usuario podrá validar y corregir los datos incorrectos.
Exportar asientos contables	El usuario podrá exportar los asientos contables una vez validados en formato tabla de excel

Tabla 4.1: Requisitos funcionales.

4.1.2 Requisitos no funcionales

El diseño de los requisitos no funcionales se ejecutó tomando en cuenta las necesidades y peticiones de la oficina. El diseño de estos se ultimó en paralelo con el diseño de los requisitos funcionales. A continuación se expondrá una lista más detallada de cada uno:

Requisitos	Descripción
Seguridad de los datos	Debido al uso de modelos de lenguaje como GPT-4 es esencial que el sistema garantice la seguridad y privacidad de los datos del usuario.
Rapidez del sistema	Es importante que el sistema procese rápidamente las facturas así como las predicciones.
Bajo coste de procesamiento por factura	El sistema deberán garantizar una reducción del coste de procesado por cada factura, incluyendo el coste humano.
Sistema accesible para cualquier nivel de contabilidad	El sistema y uso deberá garantizar un fácil acceso de cualquier perfil contable, incluso a personas sin muchos conocimientos.

Tabla 4.2: Requisitos no funcionales.

4.2 Casos de uso

En esta sección se profundizará en el diseño de los casos de uso del sistema y se explicarán más a fondo las decisiones tomadas.

Debido al diseño del LLM stack(2.5) el sistema se construye en torno a un único caso de uso: *Generar asientos contables*. Este caso de uso es el encargado de disparar las funciones que hacen posible el procesado de las facturas.

Por consecuencia, se optó por diseñar los casos de uso como funciones de alto nivel que interactúan entre si. Estas funciones se decidieron agrupar, por diseño, en diferentes módulos dando servicio al caso de uso *generar asientos contables*, como se puede ver en la Figura 4.1. El apartado técnico de cada módulo se analizará más en profundidad en el capítulo de *Aplicación*.

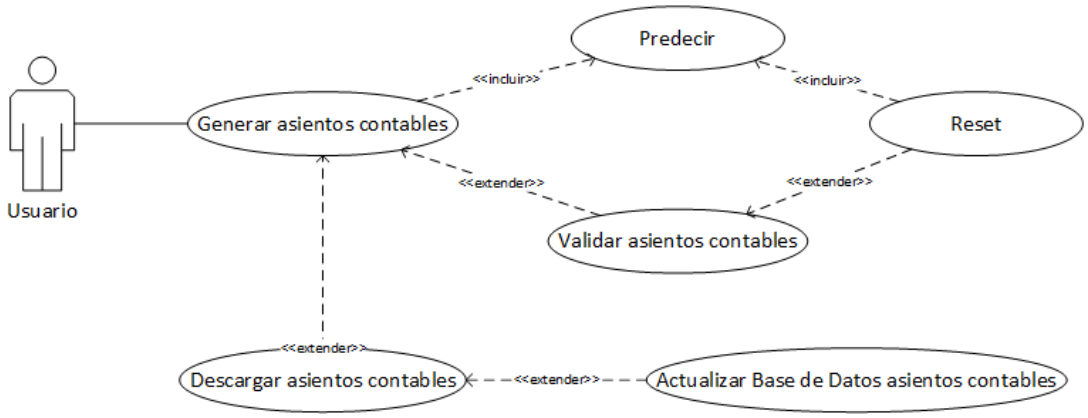


Figura 4.1: Casos de Uso

Como se puede apreciar en la figura, el usuario interactúa con el sistema a través de un único caso de uso: *Generar asientos contables*. Este caso de uso, a su vez, implementa el caso de uso *Predecir*; este caso de uso permite generar las predicciones de los códigos de cuentas. Además, también es implementado por el caso de uso *Reset*.

Por otro lado, *Validar asientos contables* permite al usuario comprobar los datos extraidos una vez generados los asientos. Este caso de uso extiende de *Reset*, permitiendo al usuario regenerar las predicciones si ha corregido algún dato del asiento.

Por último, *Generar asientos contables* extiende de *Descargar asientos contables*, este caso de uso se da una vez todas las facturas están validadas por el usuario. Por ultimo, se permite al usuario *Actualizar la base de datos* con las nuevas cuentas generadas en los asientos contables.

Capítulo 5

Desarrollo

En este capítulo se procederá a explicar la arquitectura de la aplicación, así como una presentación detallada de los diferentes módulos y elementos con los que cuenta.

5.1 Arquitectura general del proyecto

El proyecto se ha construido bajo la arquitectura de tipo *arquitectura de cliente servidor*, el cual consta de dos subsistemas:

- **Servidor o *Backend***, encargado del procesamiento de datos y la lógica de negocio. Opera en el servidor y no es visible directamente para el usuario final.
- **Cliente o Frontend**, sitio web que interactúa directamente con el usuario, abarcando la interfaz de usuario y la experiencia de este, ejecutándose generalmente en su navegador o dispositivo.

Puede observarse que la aplicación se divide en *frontend* y *backend* y cómo estas a su vez se subdividen en subsistemas de los que hablaremos a lo largo de este capítulo. Para el intercambio de información entre los dos componentes se ha utilizado el protocolo REST, el cual consigue que ambos subsistemas sean a su vez programas independientes. Esto supone una ventaja de cara al desarrollo, ya que podemos estar haciendo pruebas independientemente en cada uno de los subproyectos. Además, desde el punto de vista de la implantación de este sistema en un entorno real, por ejemplo, cabe la posibilidad de poder tener instalados ambos subsistemas en distintas máquinas.

Otras ventajas con las que cuenta la *arquitectura cliente-servidor* es la posibilidad de tener repartida la *capacidad del proceso*, la *separación de responsabilidades* que facilita y clarifica el diseño final del sistema y un *acoplamiento débil* entre ambas capas, característica deseable en un buen diseño *software*.

En conclusión, una arquitectura de este estilo proporciona diversas ventajas que la convierten en una solución atractiva a la hora de desarrollar nuestra herramienta. De todas formas, la mayor dificultad que presenta este tipo de arquitecturas está relacionado con la complejidad anexa al desarrollo de la propia estructura.

Cliente ligero

La aplicación además presenta una estructura de *cliente ligero/servidor pesado*, ya que la mayor parte de la aplicación corre en el lado servidor, es decir, que la carga computacional reside principalmente en el servidor. Entre las dos opciones (cliente o servidor ligero) la solución escogida es la más popular, pues ofrece mayor flexibilidad, al ser más común la existencia de muchos clientes y pocos servidores.

En la figura 5.1 podemos apreciar la estructura general de comunicación entre el *backend* y el *frontend* utilizando la tecnología API REST 3.3.1

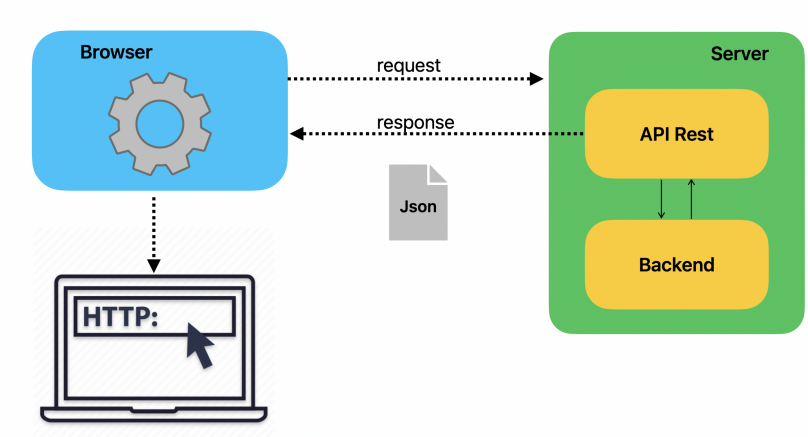


Figura 5.1: Comunicaión API REST entre backend y frontend

5.1.1 *Frontend* o cliente

El *frontend* es el encargado de mostrar los datos al usuario, permitiéndole navegar por una interfaz que abstraiga la implementación del sistema, así como interactuar con el sistema de manera cómoda.

La implementación que se ha decidido llevar a cabo es una implementación web que utiliza el lenguaje de programación *React* (3.3.1). Este presenta una arquitectura modular basada en componentes, descrita más adelante(5.1.1). Debido a la aproximación escogida de cliente ligero, este sistema es más ligera que el *backend*. Para comunicarse con el *backend* se dispone de cuatro *endpoints*, cada uno con un servicio diferente:

- **upload**: operación POST de HTTP que permite subir una lista de pdf para que sean

procesados en el *backend*. Este *endpoint* recibe como respuesta un json con todos los datos necesarios para ser mostrados en la UI.

- **reloadEntries**: operación GET de HTTP que permite recalcular las contrapartidas.
- **saveAccounts**: operación POST de HTTP que permite actualizar, guardando las nuevas cuentas contables, el índice de vectores del *backend*.
- **generate_xls**: operación POST de HTTP que permite la creación de un archivo Excel. Este archivo se utiliza como medio para exportar los datos contenidos en la aplicación.

Estructura

El *frontend* se estructura en diferentes componentes con relación jerárquica que permite acceder a los diferentes servicios disponibles en el *backend* de manera modular e independiente. La estructura de estos componentes se puede apreciar más en detalle en la figura 5.2

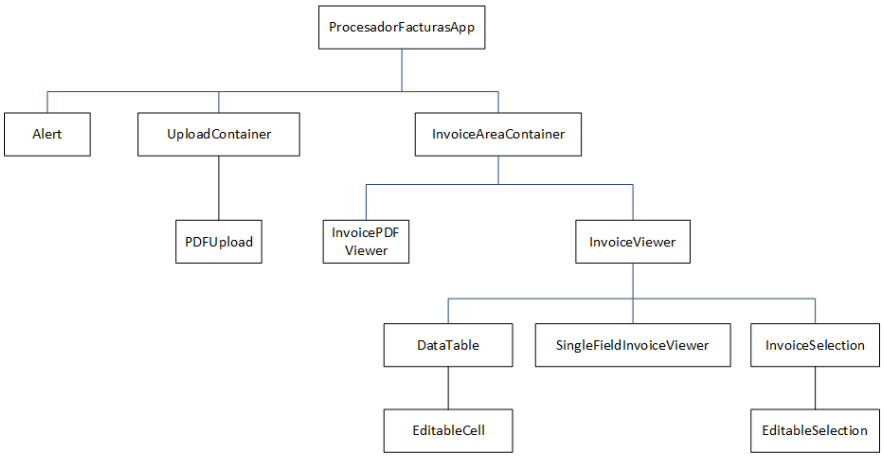


Figura 5.2: Estructura de los componentes

Estos componentes se comportan de manera independiente gracias a su naturaleza modular y se relacionan entre sí mediante la composición para aportar las funcionalidades necesarias al *frontend*. A continuación se describirá más en detalle cada uno de ellos.

- **Alert:** utiliza el componente propio de React, Alert para implementar un componente de alertas personalizado. Este componente se activa cuando el sistema detecta una excepción, mostrando un mensaje con el error ocurrido.
- **UploadContainer:** componente que actúa como contenedor de los pdf subidos, los muestra y gestiona el envío al *backend*.
- **PDFUpload:** componente que gestiona la subida de los documentos al *frontend*.

- **InvoiceAreaContainer:** Componente que actúa de contenedor de todos los componenetes de UI de gestión de los datos ya procesados.
- **InvoicePDFViewer:** componente que permite mostrar el pdf de cada factura.
- **InvoiceViewer:** componente contenedor de los datos de las facturas.
- **DataTable:** tabla que muestra los tipos de IVA y las retenciones y gestiona los cambios para que sean permanentes.
- **EditableCell:** permite editar las entradas de la tabla.
- **SingleFieldInvoiceViewer:** este componente permite mostrar y editar un campo dentro de la estructura de datos de las facturas.
- **InvoiceSelection:** componente de tipo selección que permite seleccionar entre una lista de contrapartidas.
- **EditableSelection:** añade un campo editable a la selección de InvoiceSelection permitiendo así añadir una selección que no exista en la lista.

5.1.2 *Backend* o servidor

El *backend* es la estructura encargada de implementar toda la lógica de negocio, además de la comunicación con las estructuras de datos y sistemas de terceros utilizados en el proceso. Debido a la aproximación de cliente ligero, es en el *backend* donde reside la mayor carga de trabajo. Este presenta varios *endpoints* con tecnología API REST(3.3.1) para dar los diferentes servicios disponibles al *frontend*.

Estructura

El *backend* presenta una estructura compleja debido a que ofrece cuatro servicios. Cada uno de ellos tiene su propia aproximación. A continuación se describirá la estructura en profundidad de los mismos.

UploadFiles

Este servicio implementa la funcionalidad principal de la herramienta. Para ello, sigue la estructura típica de un *pipeline*. El *pipeline* consta de un conjunto de pasos o filtros independientes que se ejecutan de forma secuencial para producir la salida deseada, como se puede observar en la figura 5.3.

Figura 5.3: Pipeline de UploadFiles

El *pipeline* utilizado consta de los siguientes filtros:

- **Filtro de Extracción de Tetxo:** recibe una lista de documentos pdf, extrae el contenido de cada documento y lo añade a una lista.
- **Filtro de extracción de Datos:** recibe la lista de contenidos. Con cada contenido de cada factura realiza una extracción de datos mediante el módulo de *comunicación GPT* devuelve una lista con los datos extraídos.
- **Filtro de Formateo:** formatea las estructuras de datos para devolverlos de manera correcta.

Reset

Este servicio aporta la funcionalidad de recargar las contrapartidas en función de los campos de datos modificados por el usuario. Permite al usuario, una vez extraídos los datos de las facturas, actualizar las contrapartidas si este ha realizado algún cambio en el emisor o en el concepto de la factura debido a un error en la extracción. Para ello, este servicio recurre a estructuras de datos y al módulo de *link prediction* para realizar su trabajo. Su estructura se puede ver más claramente en la figura 5.4

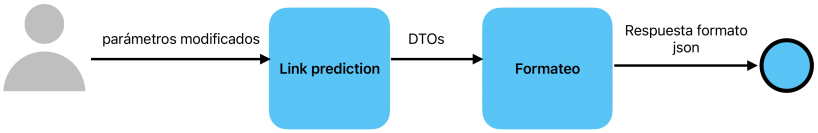


Figura 5.4: Pipeline de Reset

SaveAccounts

Este servicio aporta la funcionalidad de aprendizaje al sistema. Gracias a él, el usuario tendrá la opción de añadir nuevas cuentas para que el sistema tenga acceso a ellas en la siguiente

predicción. De esta manera, aprenderá ante la aparición de nuevas cuentas contables que antes no existían. Este servicio se sirve del módulo de *link prediction* para cargar las nuevas entradas en el indice de vectores.

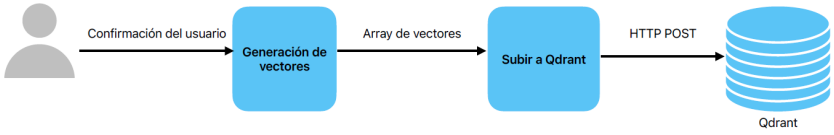


Figura 5.5: Pipeline de saveAccounts

Generate xlm

Permite al usuario exportar los datos resultantes. El usuario será capaz de generar un archivo xls con el formato correcto de asiento contable, lo que garantiza una fácil importación en otras herramientas de contabilidad. Este servicio le aporta al sistema una vía de comunicación adaptable a otros sistemas contables externos.

Figura 5.6: *Pipeline* de generate xml

5.2 Módulos

Un *módulo* es una división del *software* clara y manejable con interfaces modulares perfectamente definidas. En el diseño de alto nivel hay que ver un módulo como una caja negra, donde se contemplan exclusivamente sus entradas y sus salidas y no los detalles de la lógica interna del módulo[1].

Estructura

La aplicación está dividida en cinco módulos que se trataran a continuación en detalle. Presentaremos el *contexto y objetivo, el análisis, diseño estructura de datos y procesos* de cada uno.

- ***Módulos de Comunicación***: encargados de la comunicación con sistemas de externos.
  - ***Módulo de comunicación con GPT***: encargado de enviar y recibir peticiones a GPT.

[1] https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-estructura/

  - ***Módulos de comunicación con Azure***: encargado de enviar y recibir las peticiones a Azure.
  - ***Módulo de comunicación con Qdrant***: encargado de gestionar las peticiones a Qdrant.

- ***Módulos Funcionales***: encargados de gestionar y coordinar las funcionalidades del sistema.
  - ***Módulo textExtraction***: encargados de orquestar la extracción de texto de las facturas.
  - ***Módulos de Link Prediction***: orquestan la generacion de las cuentas contables.

- ***Módulo de Pipeline***: orquestador de todos los procesos del sistema.

- ***Módulo de Excepciones***: implementa y gestiona las excepciones personalizadas.

En la Figura 5.7 se puede apreciar la comunicación directa entre módulos, aunque estén totalmente desacoplados. Es importante tener este esquema en mente para comprender cómo funciona el sistema en todo su conjunto.

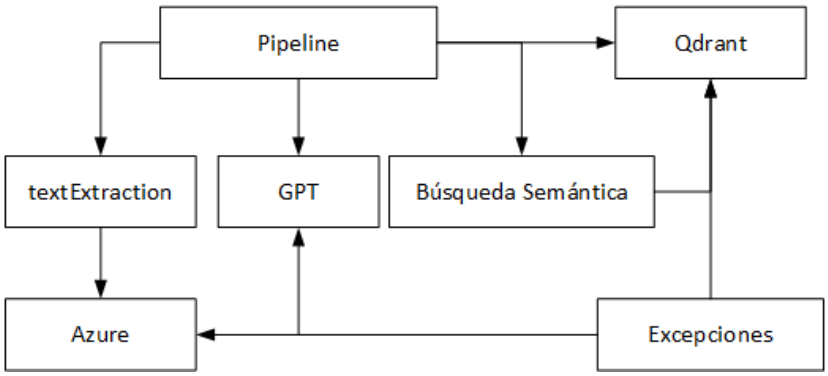


Figura 5.7: Diagrama de estructura de los módulos

5.2.1 GPT

La función de este módulo es la de abstraer la comunicación con GPT, aportando funcionalidades ajustadas a cada uno de los casos de uso de esta comunicación. Por ello, este módulo permite, para un conjunto de contenidos de facturas, extraer los campos necesarios para la creación de asientos contables(2.1.5).

Contexto y objetivos

En el proceso de manejo de facturas, los contables se enfrentan al desafío inicial de extraer los datos necesarios de estos documentos. Aunque pueda parecer una tarea sencilla, en

realidad requiere un conocimiento experto, debido a la diversidad en los formatos de las facturas. Estos documentos no siguen un estándar uniforme, lo que motiva que la extracción de datos de manera automatizada sea compleja, si se utilizan los métodos convencionales de procesamiento de texto.

Ante esta dificultad, se identificó la potencialidad de utilizar la tecnología de GPT de OpenAI como una herramienta de conocimiento experto. Gracias a su amplio entrenamiento y su capacidad para manejar tareas de complejidad moderada, GPT proporciona una solución más accesible y manejable. La clave de su eficacia reside en la identificación y optimización del *prompt* adecuado que le permita realizar la tarea de forma correcta y controlada. La selección del *prompt* correcto es crucial para garantizar que GPT comprenda y ejecute la tarea de extracción de datos de la manera más eficiente y precisa posible.

El objetivo principal de este módulo, por tanto, es proporcionar una extracción de datos precisa y abstracta a través de la API de GPT. Este enfoque no solo simplifica el proceso de extracción de datos, sino que también lo hace más flexible y adaptable a una variedad de formatos de factura. Mediante la utilización de GPT, se busca superar las limitaciones de los métodos tradicionales de procesamiento de texto, mediante una solución más dinámica y eficiente para el procesamiento contable de facturas.

Análisis de la solución

En el contexto de procesamiento de facturas en el ámbito contable, donde la precisión y eficiencia en la extracción de datos son críticas, se desarrolló un módulo de programación dedicado a mejorar esta tarea mediante la comunicación con GPT. La naturaleza variada y no estandarizada de los formatos de facturas representa un desafío considerable en este sector, lo que ha motivado la creación de una solución tecnológica específica.

La solución inicial consistió en la implementación de GPT como una herramienta con conocimiento experto, lo que facilitaba la extracción de datos relevantes. Sin embargo, se identificó que la falta de un formato de salida estándar en las respuestas de GPT, especialmente en situaciones de errores, presentaba un problema considerable.

Para abordar esta dificultad, se realizó una serie de iteraciones centradas en la mejora del *prompt* utilizado en la comunicación con GPT. Se adoptó una solución descriptiva, donde se incluyeron ejemplos detallados que especificaban el contenido esperado de cada campo. Esta estrategia mejoró notablemente la precisión en la extracción de datos y añadió información adicional para manejar posibles errores en el formato de salida.

Además, se integraron instrucciones de validación específicas, contribuyendo a la estandarización de las respuestas de GPT. Esta metodología no solo resolvió el problema de la inconsistencia en los formatos de salida, sino que también aseguró que los datos extraídos se alinearan correctamente con los requisitos de formato necesarios para su posterior uso en

procesos contables.

Aunque esta aproximación ha permitido una extracción de datos más robusta y confiable, persiste la oportunidad de optimizar aún más los *prompts* y expandir la capacidad de validación para incluir una variedad más amplia de formatos de factura. Este enfoque iterativo y adaptativo es crucial para mejorar continuamente la eficacia del módulo en el procesamiento de datos no estandarizados.

Diseño de procesos

El módulo diseñado para la extracción de datos de facturas opera de manera individualizada para cada conjunto de contenido, adaptándose a las variaciones en la estructura y el formato de cada factura. Inicia su proceso construyendo un contexto específico para cada documento, identificando los elementos clave como fechas, importes, descripciones de productos o servicios, y datos del emisor y receptor. Posteriormente, envía una solicitud personalizada a la API de OpenAI, mediante la utilización de algoritmos avanzados de procesamiento del lenguaje natural para interpretar y extraer la información relevante.

Una vez recibida la respuesta de la API, el módulo procesa los datos gracias a técnicas de análisis semántico y validación para asegurar la precisión y la integridad de la información extraída. Los datos son luego organizados en una lista estructurada, lo que facilita su revisión y permite una rápida integración en sistemas de contabilidad o plataformas de análisis financiero. Esta lista representa los datos relevantes extraídos, ahora en un formato optimizado para su uso en diversas aplicaciones comerciales y financieras.

Además, el módulo cuenta con capacidades de aprendizaje automático, de modo que se adapta y mejora su rendimiento con cada factura procesada. Esta prestación asegura una mayor precisión en la extracción de datos a lo largo del tiempo y una adaptación a los cambios en los formatos de facturas y a las nuevas tendencias en la presentación de datos financieros.

5.2.2 *Text Extraction*

El módulo en cuestión se especializa en procesar una lista de documentos en formato PDF para extraer su contenido. Un aspecto crucial de su funcionamiento es la capacidad de mantener un formato de salida correcto y válido. Esta característica es esencial para garantizar que la extracción subsiguiente de datos de los PDF sea precisa y fiable.

En su operación, el módulo no solo se centra en la extracción de datos brutos, sino que también pone un énfasis particular en la estructuración y el formateo de estos datos. Este proceso implica una conversión cuidadosa de la información contenida en los PDF a un formato que sea tanto legible como compatible con las fases posteriores. Este enfoque asegura que los datos no solo se extraigan de manera eficiente, sino que también se preparen adecuadamente para cualquier análisis o procesamiento posterior que se requiera.

También cabe destacar su funcionalidad como anonimizador. Debido a la falta de seguridad a la hora de pasar datos en bruto por un LLM, es importante la capacidad de asegurar la anonimidad con datos sensibles como DNI de personas. Por ello, este módulo cuenta con un submódulo que detecta contenido sensible y lo anonimiza.

Contexto y objetivos

A pesar de que la extracción de texto puede parecer una tarea relativamente sencilla, especialmente con el uso de un lenguaje de programación como Python, que ofrece una amplia gama de librerías útiles (3.3.1), la extracción de contenido de archivos PDF es notablemente más compleja. Dicha dificultad se debe principalmente a la naturaleza única de la estructura y metadatos de los archivos PDF, que presenta desafíos específicos no encontrados en formatos de texto más sencillos.

Resulta crucial no solo ser capaces de extraer texto, sino también de preservar la coherencia espacial de elementos clave dentro de los documentos, como las tablas y los pies de página. La posición y el diseño de estos elementos son a menudo tan informativos como el texto en sí, y su correcta interpretación es crucial para entender el contenido completo y la intención del documento.

Por lo tanto, el objetivo del módulo va más allá de la mera extracción de texto. Se centra también en mantener la integridad de la información codificada en la estructura del documento, lo que implica identificar, extraer texto y también interpretar el diseño y la disposición de los elementos en la página, de forma que se asegure que la información estructural no se pierda en el proceso de extracción.

El enfoque integral resulta clave para garantizar que la extracción de datos de los PDF sea, no solo precisa en términos de contenido textual, sino también fiel a la presentación original del documento. Al lograr este propósito, el módulo se convierte en una herramienta valiosa para el procesamiento de documentos PDF, capaz de extraer información de manera coherente con su representación original, solución que resulta crucial para análisis posteriores y aplicaciones de procesamiento de datos avanzados.

Análisis de la solución

Como se mencionó en la sección anterior, el sistema afrontó desafíos específicos al principio del proceso de desarrollo. La tarea de extraer datos de forma precisa y eficiente de las facturas no era trivial, dada la diversidad y la complejidad de los formatos de facturación. Para abordar esta problemática, se optó por una solución que implicaba el uso de una variedad de librerías especializadas. Estas librerías fueron seleccionadas cuidadosamente para permitir una extracción de datos controlada y precisa, lo que facilitaba la gestión de los elementos de manera aislada y específica. Esta metodología resultó ser especialmente efectiva en mantener

la integridad estructural de las tablas en las facturas, una cuestión crucial para la correcta interpretación y procesamiento de los datos.

Sin embargo, el proyecto evolucionó y se presentaron nuevos retos cuando se decidió cambiar a una base de datos de facturas más amplia y detallada. Esta nueva base de datos incluía una gama más diversa de facturas, algunas de las cuales eran escaneos de documentos físicos. Los escaneos planteaban un desafío único: la extracción de texto se volvía problemática debido a la variabilidad en la calidad de las copias y a la presencia de artefactos visuales que podían alterar su legibilidad.

Para superar este obstáculo, se tomó la decisión estratégica de integrar un módulo de comunicación con Azure 5.2.3. La elección de Azure como herramienta complementaria no fue casual; se basó en su robusta capacidad para manejar documentos escaneados y su avanzada tecnología de reconocimiento óptico de caracteres (OCR). La incorporación de este módulo mejoró significativamente la capacidad del sistema para procesar y extraer texto de manera eficiente, incluso de facturas escaneadas que anteriormente presentaban problemas.

Gracias a esta integración, el módulo de extracción de texto del sistema mantuvo su eficacia en el procesamiento de facturas estándar y amplió su alcance para incluir una gama más amplia de facturas, incluyendo aquellos escaneados de documentos físicos. Esta solución no solo resolvió el problema inmediato de la extracción de texto en documentos de baja calidad, sino que también preparó al sistema para enfrentar desafíos similares en el futuro, asegurando así su escalabilidad y adaptabilidad a medida que las necesidades y las tecnologías continúan evolucionando.

Diseño de procesos

Para garantizar una extracción de texto eficaz y precisa, el módulo desarrollado incorpora la capacidad de leer y procesar facturas de manera individual desde un directorio específico. Este proceso comienza con una comprobación inicial para determinar la naturaleza del documento. En el caso de que la factura no sea un escaneo, el módulo procede a analizarla en busca de elementos estructurales, como tablas, que son comunes en los formatos de facturas. Una vez identificadas, utiliza un procedimiento especializado para extraer estas tablas. Este proceso implica la recolección de datos, además de la interpretación correcta de su estructura y contenido para asegurar que la información relevante se capture de manera integral. Tras esta extracción, el módulo compila y forma un *string* que representa el contenido completo de la factura, manteniendo la fidelidad del formato original y la precisión de los datos.

En los casos donde la factura es un escaneo de un documento físico, el módulo emplea una estrategia diferente. Reconociendo la naturaleza gráfica del documento, transforma primero la factura en una imagen. Esta imagen es luego procesada mediante el módulo de comunicación con Azure. La capacidad de Azure para manejar el OCR capacita al módulo extraer texto

plano a partir de la imagen, conservando información espacial crucial que puede ser vital para entender la disposición y el contexto de los elementos en la factura. Una vez completado este proceso, el módulo genera un *string* que refleja fielmente el contenido y formato de la factura original, adaptándose a la variabilidad y complejidad inherentes a los documentos escaneados.

Como último paso del proceso, el módulo reliza un análisis sobre el contenido en busca de información sensible y la anonimiza en el caso de ser necesario.

5.2.3 Azure

Este módulo abstrae la comunicación con los servicios de OCR que ofrece Azure y aporta funcionalidades especificas para cada caso de uso. El módulo, en conclusión, posibilita extraer el texto contenido en cada factura, a la vez que mantiene la información espacial.

Contexto y objetivos

Este módulo aporta una capa de abstracción innovadora en la comunicación con los servicios de Reconocimiento Óptico de Caracteres (OCR) que ofrece Microsoft Azure. Su principal valor añadido es la capacidad de manejar eficazmente facturas en formato PDF escaneado, una tarea frecuentemente desafiante en la gestión contable.

Para cada factura presentada en este formato, el módulo utiliza avanzadas técnicas de OCR con el fin de extraer el texto contenido en el documento. Una de las características distintivas de este sistema es su habilidad para preservar la información espacial del texto extraído. Además de reconocer el texto, el módulo es capaz de mantener la disposición y estructura original de los datos dentro del documento.

Esta capacidad de conservar la organización espacial es crucial, ya que en muchas facturas, la posición del texto puede ser tan importante como el contenido mismo para entender la información. Por ejemplo, la disposición de los elementos en una factura puede indicar qué números corresponden a fechas, cantidades, nombres de productos o servicios, y otros datos relevantes.

Además, la integración del módulo con Azure OCR permite aprovechar la robustez y precisión de una de las herramientas de OCR más avanzadas disponibles. De esta manera aseguramos que la extracción de texto sea no solo precisa, sino también eficiente y confiable, incluso en facturas con variados formatos y calidades de impresión o escaneo.

Análisis de las solución

Para proporcionar las funcionalidades anteriormente descritas, este módulo se integra con el servicio de Inteligencia Artificial Azure de Microsoft, específicamente mediante la API de

Computer Vision. Esta integración se realiza a través de un cliente desarrollado en Python, que garantiza una gestión eficiente y precisa de las peticiones y respuestas.

La clave de esta implementación reside en el uso del objeto *ImageAnalyzer* de Azure. *ImageAnalyzer* brinda un control detallado sobre las peticiones al servicio de OCR y sobre el manejo de las respuestas. La gran ventaja de utilizar ImageAnalyzer es que proporciona una respuesta rica en metadatos, lo cual facilita un procesamiento posterior más controlado y preciso de los datos extraídos. Todo esto significa que, además de extraer el texto de las facturas, el módulo es capaz de capturar información contextual y estructural relevante, esencial para la correcta interpretación y uso de los datos en aplicaciones contables.

La solución implementada, aprovechando las capacidades avanzadas del modelo de Azure y un cliente Python eficiente para la comunicación, supone una manera sencilla y efectiva de abordar el problema de la extracción de texto de PDF escaneados. Con esta integración, se logra una extracción de datos robusta y fiable, crucial para aplicaciones en el ámbito contable y financiero, donde la precisión y fiabilidad de los datos son de suma importancia.

Diseño de procesos

El funcionamiento del módulo comienza con la recepción de un *imageStream*, que es una representación en forma de flujo de datos de un PDF que contiene la factura en formato de imagen.

Una vez que se ha recibido el *imageStream*, este se envía al objeto *ImageAnalyzer* de Azure. El *ImageAnalyzer* es una herramienta clave en este proceso, ya que es responsable de transformar la imagen de la factura en datos procesables. Con la configuración adecuada, el *ImageAnalyzer* analiza el *imageStream* y lo convierte en un archivo JSON. Este archivo JSON contiene no solo el texto extraído de la imagen, sino también metadatos valiosos que pueden incluir información sobre la estructura del documento, la disposición de los elementos en la página y otros detalles relevantes para el procesamiento de la factura, por lo que este sufre un proceso de formateo. El formateo puede incluir la identificación de campos específicos como fechas, cantidades, descripciones de artículos, y detalles del proveedor o cliente, así como la organización de estos datos en un formato coherente y estandarizado.

Por último, el resultado final es devuelto al módulo de *textExtraction* (5.2.2) para que continúe su procesado.

5.2.4 Excepciones

El módulo define el conjunto de excepciones propias y personalizadas del sistema para una gestión más concreta de los errores.

Contexto y objetivos

Este módulo facilita al sistema una batería de excepciones personalizadas. Este conjunto de excepciones ofrece a la aplicación una gestión más específica de los errores. También permite una mayor facilidad en cuanto a la comunicación de los errores al usuario gracias a la implementación de mensajes propios para cada excepción.

Análisis de la solución

En el contexto del sistema, se decidió definir un conjunto de cinco excepiones, una para cada caso de error. Las excepciones se agruparon conceptualmente en dos grupos: *Errores de comunicación* y *Errores de gestión.*

Llamaremos *Errores de comunicación* a las excepciones que engloban todos los errores relacionados con la comunicación de los módulos que entablan acciones con sistemas externos. La implementación de estas excepciones se adaptó a los mensajes de error devueltos por los diferentes sistemas externos y a sus características propias. Entre estos están:

- **GPTComunicationException**: esta excepción captura los errores al interactuar con la API de GPT. La excepción es lanzada cuando el sistema detecta un error con el cliente de la API.
- **QdrantComunicationException**: esta excepción captura los errores de comunicación con la base de vectores de Qdrant. La excepción lanzada cuando el cliente devuelve un *timeout* o ha ocurrido algún error en la búsqueda sobre los vectores.
- **AzureOCRException**: esta excepción captura los errores de comunicación con el modelo OCR de Azure. La excepción es lanzada cuando el sistema detecta cualquier fallo en la comunicación con el servicio de Azure.

Por otro lado, tenemos los *Errores de gestión*, excepciones que controlan cualquier problema ocurrido en los servicios ofrecidos por el sistema:

- **InternalServerException**: *excepción padre*, lanzada a nivel de comunicación con el *frontend*. Esta excepción se activa cuando se recoge alguna de las anteriores y representa un fallo crítico en el sistema que no permite el correcto funcionamiento de algunos de los servicios. El *frontend* captura y gestiona posteriormente esta excepción.

Diseño de procesos

Aunque no está concebido para la gestión de procesos específicos, este módulo cuenta con la capacidad de intervenir eficazmente en situaciones de error, gracias a su diseño estructurado y jerárquico en el manejo de excepciones.

Cuando ocurre una excepción de comunicación, el sistema la identifica y responde de manera rápida y eficiente, iniciando un proceso de comunicación con el *frontend*. En este intercambio, se transmite el mensaje de error pertinente, seguido por la activación del protocolo de recuperación de errores. Este mecanismo garantiza que el módulo reaccione de manera ágil y adecuada frente a cualquier anomalía, preservando así la estabilidad y operatividad del sistema.

5.2.5 Qdrant

Este componente constituye uno de los elementos fundamentales del sistema, gracias a las ventajas inherentes a los *embeddings* o representaciones vectoriales densas. Esta capacidad le permite administrar un ambiente orientado a la búsqueda semántica, y simultáneamente, proveer una interfaz a través de la cual otros módulos pueden interactuar eficientemente con este entorno.

Contexto y objetivos

En los sistemas tradicionales, era inusual el empleo generalizado de los *embeddings* o representaciones vectoriales. Se utilizaban en circunstancias limitadas y para tareas altamente específicas. Sin embargo, las capacidades avanzadas de los nuevos Modelos de Lenguaje de Gran Tamaño (LLM) para sintetizar y *tokenizar* información han facilitado el trabajo con estos vectores de manera más eficiente y simplificada.

Gracias al servicio de generación de *embeddings* proporcionado por OpenAI, este proyecto ha logrado desarrollar un marco de trabajo en el cual la información se codifica con un coste espacial mínimo y una escalabilidad manejable.

Consecuentemente, el objetivo principal del módulo ha sido el desarrollo de una interfaz que habilite al sistema para vincularse al índice de vectores alojado en Docker (véase Sección 3.3.1). De este modo el sistema no solo crea nuevas colecciones de datos y añade índices a las mismas, sino también realiza búsquedas efectivas dentro de estas colecciones.

Análisis de la solución

Para el correcto funcionamiento de nuestro sistema se han implementado dos colecciones de Qdrant con diferente información para permitir las funcionalidades necesarias. Las colecciones son las siguientes:

- ***Coleccion diario***: representación vectorial de parte del diario contable de Vento Abogados & Asesores. Los vectores están compuestos por los siguientes campos: un *id* generado mediante un *hash* del *string* que almacena la información del código de cuenta y el nombre de la cuenta; un campo *embedding* con el embedding representante del

nombre de la cuenta y un *string text* con el código de cuenta y el nombre de la cuenta. Esta colección almacena el histórico de los movimientos contables del despacho. Esta colección cuenta actualmente con un número total de 3558 vectores.

- ***Coleccion plan de cuentas***: representación vectorial del plan de cuentas de Vento Abogados & Asesores. Los vectores están compuestos por los siguientes campos: un *id* generado mediante un *hash* del *string* que almacena la información del código de cuenta y el nombre de la cuenta; un campo *embedding* con el vector representante del nombre de la cuenta y un *string* text con el código de cuenta y el nombre de la cuenta. Esta colección contiene el plan de cuentas de la empresa, permitiendo búsquedas sobre todas las cuentas contables existentes. Esta colección cuenta actualmente con un número total de 9883 vectores.

En la etapa inicial, se procedió a una meticulosa operación de parseo sobre los datos contenidos en el plan contable de Vento y en el diario contable de la misma. Específicamente en el diario contable se aplicó un filtro riguroso para restringir el análisis exclusivamente a las entradas vinculadas con facturas recibidas. Este procedimiento selectivo fue esencial para garantizar la extracción y el posterior análisis solo de los datos pertinentes, enfocándose en su relevancia para el proceso de vectorización.

Seguidamente, se implementó el modelo GPT con el propósito de generar vectores representativos de estos datos seleccionados. Una vez generados los vectores, estos fueron cargados y almacenados en las dos colecciones de Qdrant previamente configuradas. La configuración anticipada de dichas colecciones fue crucial para asegurar una integración efectiva y funcional de los vectores en el sistema, así como permitir las búsquedas.

La implementación de estos índices vectoriales ha desembocado en una mejora sustancial de la eficiencia del sistema en términos de acceso y manejo de datos. Esta optimización no solo ha facilitado la interacción con los modelos GPT a través de la incorporación de conocimiento experto adicional, sino que también supone una recuperación más rápida y eficiente de grandes volúmenes de información. La adopción de estos índices ha redefinido el paradigma de acceso y gestión de datos dentro del sistema, elevando su eficiencia general.

Además, la integración de vectores en el sistema ha simplificado considerablemente tareas de alta complejidad como la predicción. Estas operaciones, que anteriormente requerían de métodos más elaborados, se han simplificado significativamente mediante el empleo de un sistema de búsqueda semántica basado en la similaridad de coseno. Este enfoque avanzado facilita la identificación y recuperación de información relevante de manera más intuitiva y precisa, apoyándose en la proximidad semántica de los vectores. Dicha característica es fundamental para tareas de análisis y toma de decisiones en el ámbito contable y financiero.

Diseño de procesos

A continuación se analizarán en profundidad los procesos que permiten al módulo ofrecer las diferentes funcionalidades.

- ***Proceso de Administración de Colecciones y Vectores.*** Este procedimiento se desglosa en dos componentes críticos:
  - ***Administración de Colecciones:*** este segmento constituye un elemento crucial, facilitando el mantenimiento y actualización eficiente de las colecciones de datos. Un aspecto distintivo de este proceso es la capacidad de reiniciar una colección, retornando a un estado base o "cero", donde se mantiene la configuración, pero se eliminan todos los vectores almacenados. Esta funcionalidad resulta esencial en contextos que requieren una limpieza o reinicio de operaciones, lo que asegura un punto de partida claro y estructurado.
  - ***Incorporación de Vectores:*** adicionalmente, la administración de colecciones incorpora la habilidad de añadir vectores a conjuntos de datos existentes, con lo que se brinda una actualización constante y enriquecimiento de estos, y garantiza su pertinencia y actualización con información reciente. La adición de nuevos vectores a las colecciones existentes representa un paso fundamental en la expansión y enriquecimiento de las bases de vectores. Este proceso se lleva a cabo con meticulosidad y precisión, así se asegura la cohesión y la integridad de los datos en las colecciones.
- ***Proceso de Ejecución de Búsquedas.*** El segundo proceso central es la realización de búsquedas. Esta operación se efectúa a través de un cliente que, al seleccionar una colección específica, emplea la API de Qdrant para conectarse y realizar los cálculos correspondientes. La API de Qdrant es una herramienta avanzada que facilita la interacción con las colecciones de datos y posibilita la ejecución de búsquedas especializadas y detalladas.

5.2.6 Búsqueda Semántica

Este módulo está diseñado para resolver la tarea de predecir contrapartidas contables y crear un conjunto de códigos de cuenta que coincidan con los detalles de la factura. Además, este sistema se caracteriza por su habilidad para mejorar y ampliar su conocimiento a través de la interacción y el uso continuado por parte del usuario. Por lo tanto, se clasifica como un sistema de predicción con facultades de aprendizaje. A continuación se detallará más exhaustivamente su funcionamiento.

Contexto y objetivos

Las cuentas contables, destacadas en la sección 2.1.3, son fundamentales para la creación de asientos contables. La tarea de extraer códigos de cuentas de las facturas implica una complejidad considerable y requiere un conocimiento especializado profundo. No solo es imprescindible tener una comprensión exhaustiva de la contabilidad, sino también un conocimiento detallado del contexto contable específico en el que opera la empresa. Una extracción efectiva de los códigos de cuentas exige conocer el Plan General de Contabilidad, mencionado en la subsección 2.1.1, y comprender el método de trabajo y el Plan de Cuentas de la empresa, descrito en 2.1.4.

Durante el desarrollo del módulo, se consideraron diversas opciones que influyeron en el enfoque y los objetivos del mismo. En las primeras etapas de análisis del sistema, durante la fase de estudio de campo y mientras se evaluaban las posibles aplicaciones del sistema, el módulo se concibió inicialmente como una especialización del módulo de extracción de texto, utilizando el conocimiento experto de GPT-4 para extraer los códigos. Sin embargo, debido a las limitaciones y errores (alucinaciones) de estos modelos, esta solución fue rápidamente descartada. Estos análisis preliminares supusieron definir con mayor precisión el problema, hasta categorizarlo como un problema de predicción. Este cambio en la percepción del problema facilitó la reevaluación de los objetivos del módulo y ahí surgió una versión más alineada con los objetivos del sistema.

La predicción de cuentas en el procesamiento de facturas es una de las tareas más desafiantes y complejas en el ámbito contable. Esta tarea, lejos de ser trivial, a menudo requiere el uso de herramientas de apoyo que faciliten la labor del profesional. La complejidad de este problema aumenta con el tamaño del despacho o la envergadura de su contabilidad. En grandes despachos con voluminosos procesos contables, los planes de cuentas (2.1.4) pueden ser extraordinariamente extensos y detallados.

Dado este contexto, se definió el objetivo principal del módulo como el desarrollo de una herramienta capaz de adaptarse a las necesidades particulares de cada plan contable de diferentes empresas. Entre los objetivos específicos del módulo destacamos la habilidad del sistema para cargar planes contables ya existentes y para registrar nuevas entradas. Esta destreza permite al sistema aprender y evolucionar con el tiempo, ajustándose de manera óptima a las necesidades concretas de cada usuario.

Análisis de la solución

Como se mencionó anteriormente, en las etapas iniciales del desarrollo del módulo se consideró adaptar el proceso de predicción de cuenta al de extracción de datos de las facturas. El plan era realizar solicitudes a GPT-4 junto con la información de las facturas para generar predicciones. Sin embargo, esta idea se desechó rápidamente a consecuencia de los problemas

de *alucinaciones* del modelo, es decir, su tendencia a generar información inexacta o ficticia cuando no puede encontrar datos relevantes.

Tras abandonar este enfoque inicial, se reevaluó el problema y se analizaron herramientas existentes que realizaban tareas similares, entre las que sobresalía Ubyquo (2.3.1) por ser la usada en el despacho. Esta herramienta emplea un enfoque basado en datos históricos para intentar predecir el código de cuenta que debe utilizarse. Esta metodología resultó ser interesante y se convirtió en un componente esencial en el desarrollo del módulo. El uso de registros históricos como conocimiento experto para generar predicciones personalizadas para cada plan contable ayudaría a alcanzar uno de los objetivos clave del proyecto, además de permitir la incorporación de nuevos asientos contables a estos registros históricos como conocimiento adicional.

Por lo tanto, el siguiente paso lógico fue diseñar un sistema de almacenamiento que, primero, fuera fácilmente accesible y, segundo, permitiera almacenar el conocimiento en el formato adecuado. Fue en este punto cuando se decidió utilizar la tecnología de *embeddings*, tras haber evaluado previamente sus capacidades.

Se optó por diseñar un sistema de índices vectoriales para almacenar el conocimiento en forma de *embeddings*, de manera que se facilitara así una búsqueda semántica. Esta base de conocimiento está encapsulada en un *container* de Docker y contiene un índice creado en Qdrant, accesible a través del módulo de Qdrant (5.2.5).

La solución final incluye una API para realizar búsquedas en el índice de vectores y un *Jupyter Notebook* (3.3.1) para gestionar todos los índices. Es importante señalar que este módulo interactúa con el módulo de GPT para generar todos los *embeddings* necesarios, tanto los almacenados en el índice de vectores como los requeridos para realizar las búsquedas.

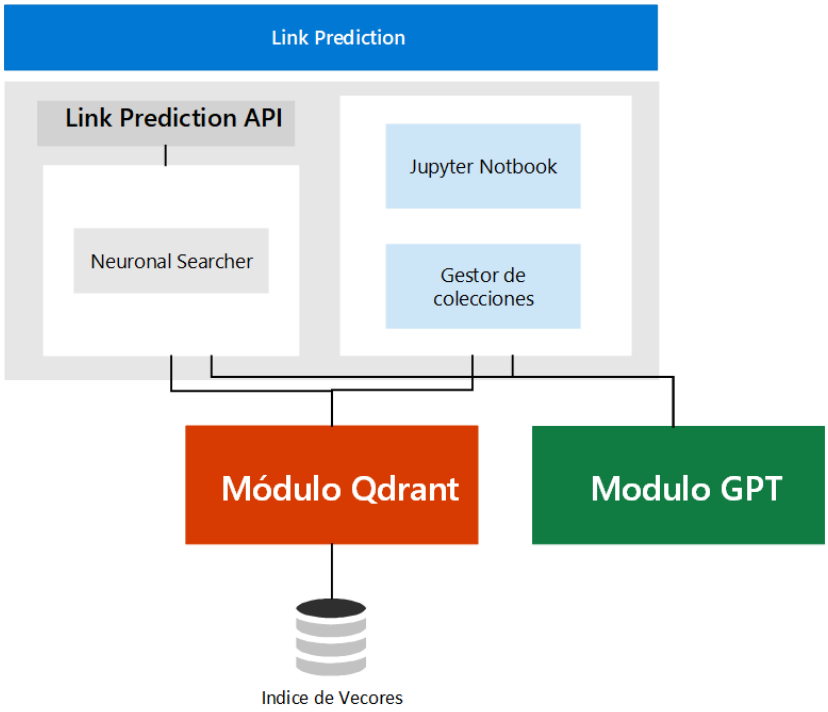


Figura 5.8: Diagrama de los componentes del módulo *Link Prediction*

Esta solución cumple eficazmente con los requisitos funcionales establecidos, con el fin de atender de manera inherente al problema de aprendizaje. Gracias a la capacidad de codificar la información en forma de *embeddings*, simplemente es necesario generar nuevos *embeddings* y añadirlos a la base de datos para incorporar nuevo conocimiento de manera eficiente y estructurada.

Diseño de procesos

El módulo se estructura en torno a dos procesos principales para ejecutar sus funcionalidades. El primero de ellos se inicia a través de la API del módulo, la cual desencadena el proceso de búsqueda. Esta API interactúa con la clase *Neuronal Searcher* para configurar la consulta correspondiente. Dicho componente crea un *embedding* de la consulta utilizando el módulo de GPT. Luego, el componente *Neuronal Search* ejecuta la consulta sobre el índice mediante el uso del módulo de Qdrant. Los resultados obtenidos en este proceso se almacenan en una estructura de datos, lo cual facilita su posterior análisis y envío de vuelta a la interfaz de usuario.

El segundo proceso se centra en la gestión de la base de datos. Este permite configurar (crear, eliminar o reiniciar) colecciones en Qdrant, así como la creación y adición de nuevos vectores. En caso de ser necesario, este proceso también puede recurrir al módulo GPT para

generar los *embeddings* necesarios.

5.2.7 *Pipeline*

En la siguiente sección, profundizaremos en el análisis del módulo de *Pipeline*. Desempeña un papel esencial como orquestador de todos los demás, pues establece la estructura fundamental del sistema. Actúa como el eje central que coordina y gestiona la interacción entre los distintos componentes, asegurando que el flujo de trabajo se mantenga organizado y eficiente.

El módulo de *Pipeline* no solo facilita la comunicación y el traspaso de datos entre los módulos, sino que también supervisa y controla el proceso general, desde la entrada inicial hasta la salida final. Es responsable de definir y mantener la lógica de procesamiento a la vez que determina el orden en que se deben ejecutar las tareas y cómo deben interactuar los distintos componentes para alcanzar los objetivos del sistema.

Además, el módulo de *Pipeline* juega un papel crucial en la escalabilidad y mantenimiento del sistema. Garantiza la incorporación o modificación de componentes individuales sin alterar significativamente la arquitectura general. Esta flexibilidad es fundamental para adaptarse a los cambios y mejoras en el sistema, así como para responder a las nuevas necesidades y desafíos que puedan surgir.

Contexto y objetivos

Al diseñar herramientas que se aprovechan de las capacidades avanzadas de los Modelos de Lenguaje a Gran Escala (LLM), es esencial considerar que estos modelos requieren un enfoque de *software* orientado específicamente a procesos. Los LLM, por su complejidad y versatilidad, demandan una estructura de *software* que pueda manejar múltiples tareas de manera eficiente y coherente.

En este marco, el rol del orquestador se vuelve crucial. Más que un componente adicional, se erige en el eje central que coordina y gestiona los diferentes procesos y acciones que se llevan a cabo. Su función consiste en garantizar que todas las operaciones, desde el procesamiento de datos hasta la generación de resultados, se realicen de manera armónica y sin errores, de modo que se mantenga la coherencia y robustez del proceso en su conjunto.

El objetivo del orquestador va más allá de la mera coordinación de tareas; busca integrar las diversas funcionalidades específicas de la herramienta para ofrecer una solución cohesiva. Tal función implica un control detallado de los flujos de trabajo, una gestión eficiente de los recursos y la capacidad de adaptarse a diferentes escenarios y requerimientos. Además, el orquestador debe ser capaz de interactuar con otros sistemas y módulos, en favor de una integración fluida y una expansión de las capacidades del *software*.

Análisis de la solución

Los módulos hasta ahora vistos representan funcionalidades concretas o trabajan como adaptadores que permiten a otros sistemas acoplarse al *framework*. Por ello, es necesario una estructura centralizada que sea capaz de coordinar las acciones y controle el flujo de procesos. Como consecuencia, este módulo se presenta como un orquestador de los otros módulos ya descritos anteriormente. Presenta una estructura de *pipeline*, clave para gestionar de forma ordenada y secuencial las diferentes interacciones con los subsistemas y asegurar así un flujo válido de los datos.

El módulo trabaja como una API de los procesos principales del sistema, y asegura un fácil acople a la aplicación de Flask montada como *backend*. En este módulo se presentan tres funciones principales:

- ***process_directory_invoices***: permite al *backend* acceder a las facturas e iniciar el procesado de estas. Representa la funcionalidad principal del sistema. esta función ejecuta uno a uno los filtros descritos en el apartado de uploadFiles(5.1.2) generando la estructura de datos final con toda la información necesaria para enviarla al *frontend*.
- ***format_responses***: esta función auxiliar es la encargada de parsear las estructuras de datos extraídos de las facturas al formato correcto para ser enviadas al *frontend*.
- ***reload_accounting_entries***: esta función genera el hilo de procesos de la funcionalidad de *reset*(5.1.2) del *backend*. La función orquesta las acciones necesarias para recalcular los códigos de cuenta de la factura seleccionada.

Gracias al diseño de estas tres funciones, la API expone de forma sencilla los procesos necesarios para los servicios del *backend*. La solución utilizada, mediante el uso de la arquitectura *pipeline*, asegura una independencia entre los diferentes filtros de cada proceso. Así se impulsa el desarrollo de nuevos filtros y la modificación de los ya establecidos. Con esta solución, además de aportar versatilidad al sistema para futuras modificaciones, favorecemos una rápida y sencilla implementación del mismo.

Diseño de procesos

El módulo de *Pipeline* presenta dos procesos, descritos en el apartado del *backend* como *uploadFiles*(5.1.2) y *reset*(5.1.2).

En el proceso de *uploadFiles*, el módulo realiza la siguiente secuencia:

1. ***Carga los archivos***
2. ***Extrae el texto de los archivos***

3. ***Extrae la información de los contenidos de las facturas***
4. ***Realiza las predicciones de las contrapartidas***
5. ***Formatea las respuestas***

Por otro lado, el proceso de *reset* consta de los siguientes pasos:

1. ***Realiza las predicciones de las contrapartidas***
2. ***Formatea las respuestas***

5.3 Pruebas

5.3.1 Verificación y validación

En esta sección, se examinarán detalladamente los procesos de validación y verificación de la herramienta empleados en este proyecto.

Para la verificación del sistema, se ejecutaron un conjunto de pruebas unitarias y de integración para cada *sprint*. Estas pruebas fueron diseñadas en función de las necesidades e ítems establecidos en cada *sprint*.

Por otro lado, la validación del sistema se realizó de manera iterativa, formando parte integral de la validación de cada *sprint*. El propósito principal de este proceso fue evaluar la calidad de las respuestas proporcionadas por el modelo GPT-4. Más específicamente, se buscó verificar la eficacia de los *prompts* en la extracción de imágenes y la precisión en las predicciones de las cuentas contables.

Para realizar estas pruebas, se contó con la colaboración de una contable experta, quien ayudó a crear un conjunto de facturas específicamente seleccionadas para este proceso. En cada *sprint*, las facturas se escogieron basándose en las dificultades identificadas en la iteración previa. Este enfoque controlado permitió abordar y corregir los problemas de manera sistemática, desafiando al sistema con los casos más complejos.

Una vez seleccionado el conjunto de facturas, se procedió a simular el uso del sistema en colaboración con la experta contable. Tras obtener los resultados, la contable experta procedió a analizar las respuestas y a emitir un juicio sobre la exactitud y corrección de la información representada. Este método de validación permitió comprobar la fiabilidad de la herramienta y aportó información valiosa para su mejora continua y su ajuste a las necesidades reales del entorno contable.

5.3.2 Pruebas de usabilidad

En esta sección se detallará el diseño y la implementación de la prueba de usabilidad diseñada para este proyecto.

El objetivo principal de esta prueba fue emular el ambiente empresarial real donde se espera implementar la herramienta. Para lograrlo, se seleccionó un grupo de cinco contables de Vento Abogados & Asesores, cada uno perteneciente a un departamento diferente de contabilidad, encargados de variadas tareas dentro de la oficina. Esta selección buscaba asegurar que las pruebas supusieran una representación significativa de los diversos perfiles contables presentes en una empresa.

En la fase de validación, al igual que en los *sprints* anteriores, se creó un conjunto de pruebas utilizando facturas, pero, en este caso, que reflejaran el uso diario típico de la herramienta. Una vez elegidos el conjunto de facturas y los participantes para la prueba, se les proporcionó una introducción general al uso de la aplicación, sin entrar en detalles exhaustivos sobre cada botón o elemento interactivo. El propósito de esta aproximación fue evaluar el carácter intuitivo de la interfaz y la facilidad de uso de la herramienta.

Después de esta breve introducción, los contables procedieron a utilizar la herramienta de manera normal durante cinco minutos cada uno. Al finalizar, respondieron a una encuesta (disponible en el anexo A), en la que se les formularon varias preguntas para evaluar su opinión respecto a los requisitos funcionales establecidos. Esta encuesta fue diseñada para recoger comentarios sobra la funcionalidad y la usabilidad de la herramienta, al mismo tiempo que obteníamos un *feedback* directo de los usuarios finales que contribuyera a futuras mejoras y ajustes en el sistema.

A continuación se puede ver la lista de *ítems* y requisitos no funcionales que se pretende analizar con esta encuesta:

Ítems y requisitos	Descripción
Legibilidad de la interfaz	Se pretende valorar por parte del usuario la legibilidad de la interfaz.
Comodidad de la interfaz	Se pretende valorar por parte del usuario la comodidad de la interfaz, comparándola con otras herramientas disponibles.
Rapidez del sistema	Es importante que el sistema procese rápidamente las facturas, así como las predicciones.
Sistema accesible para cualquier nivel de contabilidad	El sistema y uso deberá garantizar un fácil acceso de cualquier perfil contable, incluso a personas sin muchos conocimientos en este ámbito.
Sensación de error	Se analizó también la sensación por parte del usuario frente a los errores, para valorar si sentía que la herramienta entorpecía el proceso por culpa de los posibles errores.

Tabla 5.1: Ítems y requisitos de las pruebas de usabilidad.

5.3.3 Resultados de las pruebas

En esta sección se analizarán las respuestas generadas por las pruebas de usabilidad. A continuación se expondrán los resultados por cada ítem analizado:

- ***Legibilidad de la interfaz***: la evaluación de la legibilidad se realizó mediante una escala de calificación, donde 1 indica que la interfaz es completamente incomprensible, y 5, que es totalmente clara. La encuesta arrojó una calificación promedio de 4 puntos, lo cual sugiere que la interfaz es legible e intuitiva para los usuarios.
- ***Comodidad de la interfaz***: de forma similar, la comodidad de uso se evaluó mediante una escala de calificación, donde 1 representa una interfaz incómoda y difícil de usar, y 5, una interfaz muy cómoda de manejar. El resultado fue una calificación unánime de 4 puntos, indicando que la interfaz cumple satisfactoriamente con el requisito de

comodidad. Es importante destacar que se incluyó una pregunta adicional para comparar nuestra herramienta con la que se usa actualmente en la oficina, y el 75% de los participantes encontraron nuestro sistema más cómodo de usar.

- ***Rapidez del sistema***: la rapidez del sistema se evaluó a través de las respuestas a la pregunta: "¿Qué es lo que más te ha llamado la atención de la herramienta?". Todas las respuestas destacaron la rapidez y eficiencia con la que el sistema procesa las facturas y genera los archivos para exportación.
- ***Sistema accesible para cualquier nivel de contabilidad***: Para evaluar este ítem se preguntó al usuario directamente si consideraban esta herramienta apta para un usuario no contable. Los resultados fueron que el 75% de los usuarios están de acuerdo con que es una herramienta accesible para un perfil no contable.
- ***Sensación de error***: Para medir este ítem, se les preguntó a los usuarios si emplearían la herramienta en sus día a día para completar sus tareas contables. Los resultados mostraron que todos los usuarios confiarían en el sistema para trabajar.

En conclusión, nuestra herramienta se encuentra actualmente en una fase inicial respecto a su implantación en entornos empresariales. A pesar de este carácter prematuro, los resultados obtenidos en las diversas pruebas realizadas indican que la herramienta satisface plenamente los criterios establecidos inicialmente. Específicamente, el sistema destaca por ofrecer una interfaz legible y amigable para el usuario, cumpliendo así con el requisito no funcional de rapidez en el procesamiento. Además, se ha diseñado para ser accesible a usuarios con limitada formación en contabilidad, lo que amplía significativamente su aplicabilidad y utilidad potencial. Este conjunto de características subraya el compromiso de desarrollar una solución que no solo atienden las necesidades técnicas, sino que también se centra en la experiencia y la inclusividad del usuario final.

Capítulo 6

Seguridad, Legislación y Ética

El avance en los modelos de lenguaje introduce retos técnicos, aunque también impone, especialmente a los desarrolladores, la obligación de garantizar un uso seguro y ético de estas herramientas. Por esta razón, en este capítulo profundizaremos en los aspectos relacionados con la seguridad y la ética en la utilización de los Modelos de Lenguaje a Gran Escala.

6.1 Seguridad y legislación

La seguridad y privacidad de los datos en internet son esenciales en la salvaguarda de información personal y corporativa. Estas medidas y prácticas están diseñadas para proteger contra accesos no autorizados, abusos o robos. La amplitud de este ámbito se extiende desde la protección de la identidad digital de individuos hasta la seguridad de datos corporativos y gubernamentales. En un mundo cada vez más interconectado, donde la información se transmite y almacena digitalmente, la importancia de estas medidas no puede ser subestimada. La seguridad de los datos se ocupa de evitar el acceso indebido o la alteración de la información, mientras que la privacidad se centra en el manejo adecuado de datos personales, respetando la confidencialidad y las preferencias de los usuarios. Con el aumento de ciberamenazas y la evolución constante de la tecnología, la seguridad y privacidad de los datos se han convertido en pilares fundamentales en la gestión de la información digital.

La Unión Europea (UE) se ha posicionado a la vanguardia en la regulación de la seguridad y privacidad de datos con la implementación del Reglamento General de Protección de Datos (GDPR, por sus siglas en inglés) [17]. Este reglamento es un hito significativo en la protección de datos, estableciendo un marco legal para la gestión y el procesamiento de datos personales dentro de la UE y el Espacio Económico Europeo.

El GDPR se centra en otorgar a los individuos un mayor control sobre sus datos personales. Bajo este reglamento, se exige a las organizaciones ser transparentes en cómo y por qué

recopilan datos personales, además de garantizar el consentimiento explícito de los usuarios. Por otra parte, el GDPR refuerza el derecho al olvido, de forma que los individuos puedan solicitar la eliminación de sus datos de los sistemas de las empresas.

Estos estándares no solo han mejorado la seguridad de los datos en Europa, sino que también han planteado desafíos significativos para las empresas. La implementación del GDPR requiere una revisión exhaustiva de las políticas de datos y sistemas de TI, situación que a menudo repercute en la necesidad de nuevas inversiones en seguridad y capacitación del personal.

La convergencia de estos estándares representa un enfoque integral para la protección de datos: incluye desde la seguridad cibernética hasta la privacidad del usuario. Las empresas que operan en la UE deben adaptarse a estos estándares, lo que a su vez promueve una cultura de seguridad y privacidad más sólida a nivel global. Esta normativa ha sentado un precedente, que ha influido en legislaciones similares en otras regiones del mundo.

6.1.1 Seguridad y LLM

Los Modelos de Lenguaje a Gran Escala (LLM), entrenados en vastos conjuntos de datos de texto, pueden generar texto, responder preguntas y realizar una variedad de tareas lingüísticas con cada vez más eficacia. Sin embargo, su ascenso también ha planteado dilemas de seguridad significativos.

Exposición de Datos Sensibles: uno de los riesgos más notables de los LLM es la posibilidad de que expongan datos sensibles. Durante el entrenamiento, los modelos pueden incorporar inadvertidamente información confidencial presente en los datos de entrenamiento, situación que podría derivar en a divulgación no intencionada de dicha información.

Manipulación y Generación de Desinformación: los LLM son capaces de generar texto convincente, lo que los hace susceptibles de ser utilizados para crear desinformación o contenido engañoso a gran escala. Esta capacidad plantea serios problemas en términos de seguridad y veracidad de la información.

Sesgo en la Inteligencia Artificial: los LLM pueden perpetuar o amplificar sesgos presentes en los datos de entrenamiento. Tal comportamiento puede originar resultados sesgados o discriminatorios. Especialmente en aplicaciones que afectan a decisiones críticas en la vida real, dichos sesgos generan dilemas éticos y de seguridad.

Para mitigar estos riesgos, se están adoptando diversas estrategias. La limpieza de datos es fundamental para eliminar información sensible y reducir sesgos en los conjuntos de entrenamiento. La supervisión humana en las respuestas generadas y la auditoría continua de los modelos ayudan a identificar y corregir errores y sesgos. Además, la implementación de políticas de uso responsable, que definen límites claros sobre cómo se pueden utilizar estos modelos, es crucial para prevenir abusos.

La evolución de los LLM hacia modelos más seguros y éticos es un proceso continuo. Los desarrolladores y usuarios deben colaborar para fomentar que estos poderosos sistemas se utilicen de manera que protejan la privacidad y la seguridad, al tiempo que proporcionan sus significativos beneficios.

El uso de GPT-4 en este proyecto subraya la necesidad de discutir las medidas de seguridad y privacidad implementadas por OpenAI. GPT-4, como tecnología avanzada de procesamiento de lenguaje natural, plantea desafíos y responsabilidades en términos de manejo de datos.

6.1.2 Seguridad y OpenAI

Con su notable expansión y el lanzamiento reciente en GPT Store [18], OpenAI ha integrado una variedad de medidas de seguridad en sus modelos y servicios. Como se menciona en el Portal de Seguridad de OpenAI [19], la empresa actualmente posee varios certificados de seguridad, demostrando así su compromiso con la protección y el manejo responsable de datos. Los certificados son los siguientes:

- ***CCPA***: *California Consumer Privacy Act*, o Ley de Privacidad del Consumidor de California en español. Es una ley de privacidad con alcance estatal que regula cómo las organizaciones pueden gestionar la información personal de los residentes de California. Es la primera ley integral de protección de datos moderna en los Estados Unidos.
- ***GDPR***, Reglamento General de Protección de Datos.
- ***SOC 2***, los Controles de Servicio y Organización 2 son una auditoría de los procedimientos de control en las organizaciones informáticas que ofrecen servicios.
- ***SOC3***, es un informe público sobre controles internos en materia de seguridad, disponibilidad, integridad del tratamiento de datos y confidencialidad.

A pesar de los certificados y las medidas de seguridad, los LLM presentan dificultades añadidas de seguridad a causa de su procesamiento mediante razonamiento. Diversos estudios apuntan a que existen brechas de seguridad en los modelos de ChatGPT.

Un estudio reciente [20] plantea que las estrategias defensivas actuales en los Modelos de Lenguaje a Gran Escala (LLM), incluyendo el ajuste fino y la restricción de respuestas, no aseguran resultados totalmente eficaces. A pesar de estos métodos, los LLM todavía pueden producir contenidos inapropiados. Las técnicas de censura tradicionales, basadas en el aprendizaje automático, requieren otro LLM para filtrar respuestas indeseadas. Este estudio señala las limitaciones teóricas de dichos métodos de censura semántica, que pueden considerarse como problemas sin solución definitiva. Dicha situación subraya los retos a los que se enfrentan estos sistemas de censura, que se ven agravados por las capacidades de programación y

la adaptabilidad a las instrucciones que caracterizan a los LLM. Además, el estudio plantea la problemática inherente al funcionamiento de los modelos GPT, ya que se ciñen ciegamente al deber de seguir instrucciones.

Otro estudio [21] explora la intersección de los Modelos de Lenguaje de Gran Escala (LLMs) con la seguridad y privacidad. Específicamente, investiga cómo los LLMs impactan positivamente en la seguridad y privacidad, en los riesgos potenciales asociados con su uso, y en las vulnerabilidades inherentes dentro de los LLMs. A través de una revisión exhaustiva de la literatura, el artículo categoriza los hallazgos en "Lo Bueno" (aplicaciones beneficiosas de los LLMs), "Lo Malo" (aplicaciones ofensivas) y "Lo Feo" (vulnerabilidades y sus defensas). Por ejemplo, demostraron que los LLMs mejoran la seguridad del código y de los datos, hasta superar los métodos tradicionales. Sin embargo, también pueden ser aprovechados para varios ataques (particularmente ataques a nivel de usuario) por sus capacidades de razonamiento similares a las humanas. Las áreas que requieren esfuerzo de investigación son las siguientes:

- **Ataques a nivel de usuario**. El estudio subraya las siguientes problemáticas a nivel de usuario:
  - ***Misinformation***: el contenido falso producido por los Modelos de Lenguaje de Gran Escala (LLMs) está generando serias preocupaciones en cuanto a la seguridad del contenido en línea.
  - ***Ingeniería Social***: los Modelos de Lenguaje de Gran Escala (LLMs) no solo tienen el potencial de crear contenido a partir de datos de entrenamiento, sino que también ofrecen a los atacantes una nueva perspectiva para la ingeniería social.
  - ***Conducta Científica Inapropiada***; el uso irresponsable de los Modelos de Lenguaje de Gran Escala (LLMs) puede originar problemas relacionados con la mala conducta científica, derivados de su capacidad para generar textos originales y coherentes.
  - ***Fraude***: los ciberdelincuentes han desarrollado una nueva herramienta llamada FraudGPT, que funciona como ChatGPT, pero facilita los ciberataques.
- **Vulnerabilidades y amenazas en los LLM**. El estudio analiza una lista de vulnerabilidades y amenzas de las cuales se destacan las siguientes:
  - ***Data Poisoning***: el envenenamiento de datos ocurre cuando atacantes manipulan el proceso de entrenamiento mediante la inserción de información dañina en el conjunto de datos utilizado para el entrenamiento.
  - ***Backdoor Attacks***: los ataques de puerta trasera implican la manipulación maliciosa de los datos de entrenamiento y el procesamiento del modelo, que crea una

vulnerabilidad a través de la cual los atacantes pueden incrustar una puerta trasera oculta en el modelo.

- ***Attribute Inference Attacks***: el Ataque de Inferencia de Atributos es un tipo de amenaza en la que un atacante intenta deducir información sensible o personal de individuos o entidades mediante el análisis del comportamiento o las respuestas de modelos de aprendizaje automático.
- ***Membership Inferences***: el Ataque de Inferencia de Membresía es un tipo específico de ataque de inferencia en el campo de la seguridad y privacidad de datos, que consiste en determinar si un registro de datos fue parte del conjunto de datos de entrenamiento de un modelo para acceder así al registro de datos específico.
- ***Jailbreaking***: el *jailbreaking* en los Modelos de Lenguaje de Gran Escala (LLMs) implica eludir características de seguridad para permitir respuestas a preguntas que de otro modo estarían restringidas o serían inseguras. De este modo, se desbloquean capacidades que normalmente están limitadas por protocolos de seguridad. Numerosos estudios han demostrado varios métodos para realizar el *jailbreaking* de LLMs con éxito.
- ***Prompt Injection***: el método de ataque conocido como *inyección de solicitudes* se centra en alterar el comportamiento de los Modelos de Lenguaje a Gran Escala (LLMs) para generar respuestas imprevistas y posiblemente perjudiciales. Esta estrategia consiste en formular entradas de tal manera que se sorteen las medidas de protección del modelo o se generen respuestas no deseadas. Existe una amplia investigación sobre la automatización de la identificación de inyecciones de solicitudes que mantienen el significado original, a partir distintas técnicas. Además, la adaptabilidad de los LLMs a través del ajuste fino permite la inserción de vulnerabilidades ocultas mediante estos ataques de inyección de solicitudes.
- ***Insure Plugins***; un *plugin* para un LLM es un módulo de extensión o adición que mejora las capacidades de un LLM. Se han desarrollado complementos de terceros para expandir su funcionalidad, para que los usuarios realicen diversas tareas, en las que se incluyen búsquedas web y análisis de texto. Sin embargo, algunas de las preocupaciones planteadas por expertos en seguridad consideran la posibilidad de que los complementos sean utilizados para robar historiales de chat, acceder a información personal o ejecutar código en las máquinas de los usuarios.

Dada la convergencia de los desafíos de seguridad en los Modelos de Lenguaje a Gran Escala (LLM) y la insuficiencia de conocimiento en el ámbito, emergen importantes dilemas éticos. Esta brecha en el conocimiento y la seguridad subraya la necesidad urgente de un

enfoque ético más riguroso y una evaluación constante de los impactos de los LLM en la sociedad.

6.2 Ética

Los desarrollos recientes en el campo del procesamiento de lenguaje natural (NLP) han hecho posible la síntesis y la comprensión de textos coherentes de manera más abierta, hasta convertir así los conceptos teóricos en aplicaciones prácticas tangibles. Los Modelos de Lenguaje de Gran Tamaño (LLMs) han ejercido una influencia notable en sectores como el desarrollo de *software* para resúmenes de informes y la redacción automatizada. No obstante, diversas observaciones sugieren que los LLMs pueden manifestar sesgos sociales y elementos tóxicos, con riesgos éticos y sociales significativos asociados a su uso irresponsable. A pesar de que investigaciones empíricas han identificado desafíos éticos en LLMs avanzados, existe una limitada exploración sistemática y evaluación por parte de los usuarios sobre los riesgos y comportamientos perjudiciales inherentes al uso actual de estos modelos.

Un estudio relevante en este contexto es "Red teaming ChatGPT via Jailbreaking: Bias, Robustness, Reliability, and Toxicity" de Zhuo et al. (2023) [22]. Esta investigación realiza un análisis exhaustivo de ChatGPT, y lo evalúa desde cuatro dimensiones críticas: *Sesgo*, *Confiabilidad*, *Robustez*, *Toxicidad*. Es importante resaltar que el estudio identificó una cantidad significativa de riesgos éticos que no fueron adecuadamente abordados por los *benchmarks* convencionales. A través de los *benchmarks* estableciudos llegaron a las siguientes conclusiones:

- ***Sesgo***:
  - ***Limitaciones en el Entendimiento Multilingüe***: se observa que ChatGPT no domina a la perfección varios idiomas. Este inconveniente se notó también en la versión prototipo de GPT-3. Sin embargo, usuarios regulares señalan que ChatGPT actúa más eficientemente como herramienta multilingüe. La insuficiente habilidad en comprensión de múltiples idiomas podría conducir a un sesgo en la toma de decisiones y en la generación de contenido creativo.
  - ***Sesgo de género***: el sesgo de género en los modelos de lenguaje, como GPT, emerge como una preocupante limitación en el desarrollo de tecnologías de inteligencia artificial éticas y equitativas. A pesar de los avances significativos en la precisión y capacidad de estos modelos, persiste la tendencia a reproducir y, en ocasiones, amplificar prejuicios y estereotipos de género preexistentes en los datos de entrenamiento.
  - ***Multimodalidad***: como consecuencia del procesamiento de lenguaje natural, ChatGPT

podría presentar un sesgo en la generación de código debido a un enfoque excesivamente simplista. Este sesgo en la multimodalidad puede representar un riesgo ético en la programación cotidiana, situación que puede derivar en errores significativos en aplicaciones del mundo real donde los programas tienden a ser más complejos.

- ***Robustez y Toxicidad***: la técnica de *prompt injections* ha demostrado ser un método efectivo para eludir las restricciones del modelo. A pesar de que ChatGPT podría estar entrenado para ser seguro, puede ser susceptible a riesgos emergentes relacionados con estas inyecciones de comandos. Con las capacidades avanzadas en Modelos de Lenguaje a Gran Escala (LLMs), los modelos pueden ser fácilmente manipulados para inducir comportamientos perjudiciales.

- ***Confiabilidad***: ChatGPT no incorpora adecuadamente conocimiento factual, lo cual disminuye significativamente su fiabilidad. La mayoría de las aplicaciones diarias dependen de información precisa y justificada. A causa de la generación de contenido erróneo o ilusorio, el modelo podría propagar desinformación y tomar decisiones inapropiadas en ámbitos críticos como la medicina y el derecho.

Tras un exhaustivo análisis de los problemas de seguridad, resulta evidente que nos enfrentamos a dilemas éticos relevantes. Los Modelos de Lenguaje a Gran Escala (LLMs) se perfilan como tecnologías con gran potencial, capaces de procesar vastas cantidades de información y realizar tareas de creciente complejidad. Sin embargo, como señala el estudio, aún estamos lejos de alcanzar un uso generalizado que nos permita consolidar un conjunto de métodos y prácticas recomendadas para garantizar una aplicación ética de estas tecnologías.

Pese a estas limitaciones, es crucial mantener una vigilancia continua sobre los grandes modelos de lenguaje, asegurando su uso apropiado. Como ocurre con cualquier tecnología emergente, su carácter *positivo* o *negativo* depende del uso que se le dé y no de la tecnología en sí misma. En resumen, en este nuevo horizonte que se abre en el ámbito de la inteligencia artificial, recae en los desarrolladores la responsabilidad de asegurar tanto el avance técnico como el progreso ético.

Capítulo 7

Conclusiones y Trabajo a Futuro

En este capítulo se presentará la situación final del trabajo, así como las lecciones aprendidas y las posibles líneas de trabajo futuras.

7.1 Conclusión final

El objetivo principal de este proyecto era explorar y evaluar las capacidades de los LLM cuando se integran en sistemas de *software*, con un enfoque particular en el ámbito empresarial. Para lograrlo, se llevó a cabo un doble enfoque: un estudio detallado de las capacidades inherentes a los LLM y un análisis de mercado para identificar áreas específicas dentro del mundo empresarial, en concreto, en el despacho de Vento Abogados & Asesores, donde estos modelos podrían tener una aplicación significativa.

Como parte de esta investigación, se desarrolló un conjunto de herramientas prototipo destinadas a ejecutar diversas tareas dentro del despacho de abogados. Estas herramientas además de servir como prueba de concepto de la viabilidad de los LLM en entornos profesionales específicos, facilitaron un aprendizaje práctico sobre el desarrollo de aplicaciones basadas en LLM.

Uno de los logros más notables de este proyecto fue el desarrollo de una herramienta automatizada para el procesamiento de facturas. Esta herramienta, refinada a través de iteraciones basadas en los comentarios de los usuarios, demostró cumplir con los objetivos propuestos. En particular, nuestra solución logró aprovechar las capacidades avanzadas de los LLM para extraer datos clave de las facturas, implementar un sistema de almacenamiento basado en *embeddings* eficiente y generar predicciones de enlaces con precisión.

En resumen, el sistema que hemos desarrollado cumple con los requisitos técnicos de extracción de datos y generación de predicciones, también representa un avance significativo en términos de eficiencia operativa y reducción de costos para el despacho de Vento Abogados. Según nuestros cálculos, la implantación de esta herramienta podría resultar en una reducción

de tiempo y costes de casi el 900% (porcentaje calculado a partir del coste por factura actual de Vento Abogados & Asesores) en comparación con las herramientas de procesamiento de facturas actualmente disponibles en el mercado (2.3.1). Este proyecto, por una parte valida, la utilidad de los LLM en aplicaciones empresariales específicas; por otra, subraya el potencial de estas tecnologías para revolucionar las operaciones empresariales, al ofrecer soluciones más rápidas, precisas y coste-eficientes.

7.2 Trabajo futuro

En esta sección se analizarán elementos de la herramienta actual que son mejorables y funcionalidades que podrían ser añadidas en futuras iteraciones de la herramienta.

Esta herramienta se está desarrollando dentro de un contexto de producción real. Por lo tanto, la planificación futura de su desarrollo se dividirá en dos áreas principales. La primera, denominada *Sistema como producto empresarial*, se orientará hacia la aplicación práctica a nivel de usuario y la utilidad de la herramienta en el contexto comercial. La segunda área, el *Trabajo técnico a futuro*, se enfocará en los aspectos técnicos, mediante el tratamiento de las mejoras continuas y la evolución tecnológica de la herramienta.

7.2.1 Trabajo técnico futuro

Como se ha estudiado en apartados anteriores, los LLM Stacks se establecen como *frameworks* de trabajo muy novedosos y con pocos análisis técnicos, por lo que aún no se han definido y establecido buenas prácticas o metodologías específicas para este tipo de arquitecturas.

Con todo, durante el desarrollo del proyecto se ha podido establecer un conjunto de mejoras técnicas sustanciales que permitirían enriquecer el sistema:

- Una de estas posibles mejoras sería la de desarrollar un módulo específico que gestione la comunicación con el modelo LLM de forma más exhaustiva. A pesar de que en el proyecto existe un módulo de comunicación con GPT, este solo funciona como una interfaz que abstrae la gestión del cliente de la API, con el fin de facilitar la interacción entre el sistema y GPT. Durante el desarrollo se ha comprobado que una mejora sustancial consistiría en ampliar la funcionalidad de este módulo hasta conseguir, además de la funcionalidad que implementa actualmente, que actúe de forma más versátil, es decir, que lograra comportarse como un cliente dentro del sistema con el que poder interactuar independientemente del modelo implementado por detrás. Esta opción permitiría gestionar de forma mucho más eficiente y con más capacidades la comunicación con los LLM utilizados. Gracias a este cambio, no solo se podría hacer uso de varios LLM

o instancias del mismo para la implementación de varias funcionalidades específicas, sino que también permitiría abstraer de forma total el módelo utilizado, lo que supondrá incluso el intercambio de modelos en una misma tarea. De este modo se facilitaría la implementación de varios modelos intercambiables bajo el patrón de diseño estrategia.

- Otra de las mejoras del sistema sería el desarrollo de un proceso de extracción de texto más específico y propio de la herramienta. Podría alcanzarse este propósito gracias al uso de OCR de código libre como Tesseract, que garantiza un gran control y adaptabilidad del proceso de extracción por parte del programador. Con este cambio se avanzaría en las capacidades de extracción, puesto que facilitaría implementar un proceso mucho más acorde con las necesidades específicas del caso de uso y permitiría anexarse de los servicios de Microsoft.

- Por último, una de las principales ampliaciones de la herramienta sería la utilización de un modelo LLM propio de código libre. Estos últimos meses han surgido varios modelos LLM de código libre con resultados asombrosos, muy parecidos a los resultados de los grandes modelos(GPT-4, Gemini) e incluso superando a algunos (GPT-3.5). Además, estos modelos de código libre, como podría ser Mixtral o LLaMa 2, cada vez están más optimizados y requieren de menos potencia de ejecución, por lo que sería muy interesante recurrir a una instancia de estos modelos propia. A pesar de que esta tarea se presenta como ardua y compleja, estae avance prepararía al sistema para garantizar la privacidad total de los datos, ya que se gestionarían al completo localmente, además de reducir costes, al no depender de un servicio de pago externo.Otra consecuencia positiva sería la reducción de tiempos de procesado, pues las API de OpenAi pueden tener fluctuaciones en la latencia.

7.2.2 Sistema como producto empresarial

A pesar de que la herramienta actualmente funciona de forma correcta y consigue los objetivos impuestos en esta iteración, cabe destacar que, como producto empresarial, se pueden definir varias mejoras que lanzarían al sistema a competir con las aplicaciones de gestión de facturas que existen en el mercado. A continuación se presentan las posibles mejoras definidas gracias a las pruebas de usabilidad realizadas (5.3.2):

- **Comunicación con sistemas contables**: actualmente la herramienta funciona de manera paralela a los sistemas de contabilidad utilizados en la oficina. A pesar de que, gracias a la exportación en formato *excel*, se permite una fácil importación de los asientos, el acoplamiento de estos sistemas a la herramienta permitiría cargar de forma automática las cuentas contables en los indices de vectores, así como una gestión conjunta de

los datos contables. Todo ello conduciría a una mayor flexibilidad y adaptación a las prácticas contables de cada empresa.

- **Creación de usuarios y roles**: otra de las actualizaciones de la herramienta que se plantean es la capacidad del sistema para aislar funcionalidades o procesos mediante usuarios y roles. Esta modificación supondría gestionar de forma aislada diferentes facturaciones y entornos de trabajo en función del perfil de las personas que lo usan, muy necesario en despachos con varios contables. Esta abstracción de roles también resolvería de forma muchos más sencilla la administración de las bases de datos y otros componentes, separando, por ejemplo, el rol de contable del de administrador del sistema.

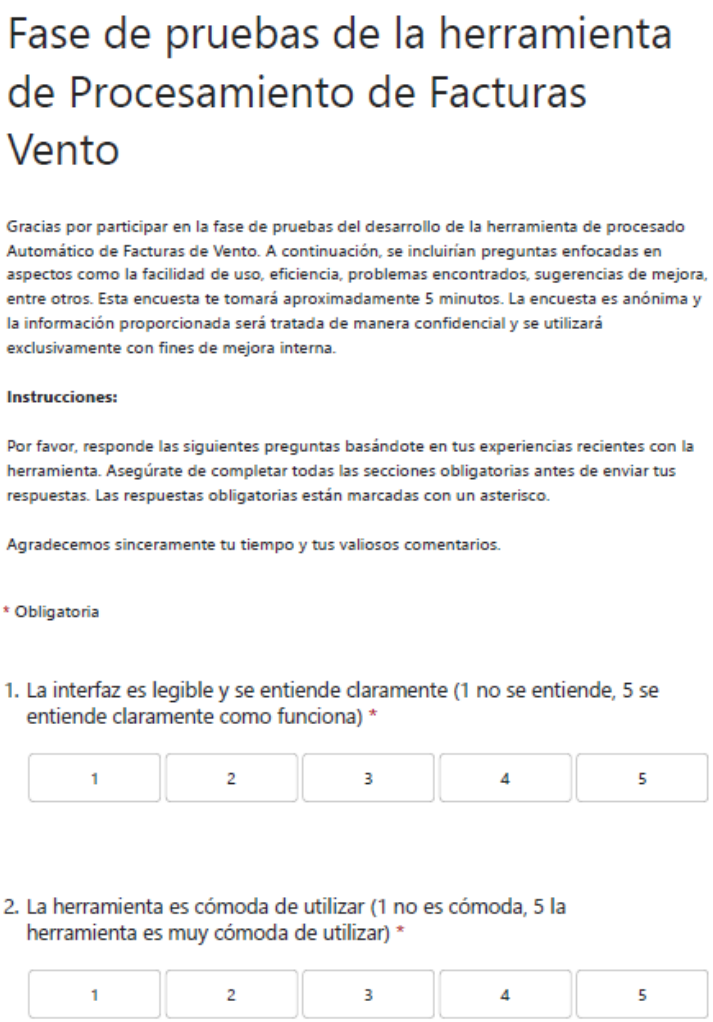
- **Mejoras menores del sistema**: por último, se ha definido una batería de modificaciones y correcciones menores para que la herramienta fuese más competente y completa:

  - ***Capacidad para procesar archivos con varias facturas***: se propone como mejora añadir la capacidad del sistema de procesar archivos PDF con más de una factura en su interior.
  - ***Visor paralelo de factura***: en el caso de uso de validación de los asientos, se plantea como mejora modificar el visor de factura para que esté siempre activo: de esta manera, la factura estaría siempre visible para validar los datos, y no sería el usuario quien elige activarlo o no.
  - ***Calculo automático***: se plantea como mejora añadir la capacidad al *frontend* de recalcular automáticamente los montos del IVA o la retención si los porcentajes se vieran modificados.

Apéndices

Apéndice A

Encuesta Fase de pruebas



Fase de pruebas de la herramienta de Procesamiento de Facturas Vento

Gracias por participar en la fase de pruebas del desarrollo de la herramienta de procesado Automático de Facturas de Vento. A continuación, se incluirían preguntas enfocadas en aspectos como la facilidad de uso, eficiencia, problemas encontrados, sugerencias de mejora, entre otros. Esta encuesta te tomará aproximadamente 5 minutos. La encuesta es anónima y la información proporcionada será tratada de manera confidencial y se utilizará exclusivamente con fines de mejora interna.

Instrucciones:

Por favor, responde las siguientes preguntas basándote en tus experiencias recientes con la herramienta. Asegúrate de completar todas las secciones obligatorias antes de enviar tus respuestas. Las respuestas obligatorias están marcadas con un asterisco.

Agradecemos sinceramente tu tiempo y tus valiosos comentarios.

* Obligatoria

1. La interfaz es legible y se entiende claramente (1 no se entiende, 5 se entiende claramente como funciona) *

1	2	3	4	5

2. La herramienta es cómoda de utilizar (1 no es cómoda, 5 la herramienta es muy cómoda de utilizar) *

1	2	3	4	5

Figura A.1: Encuesta fase de pruebas

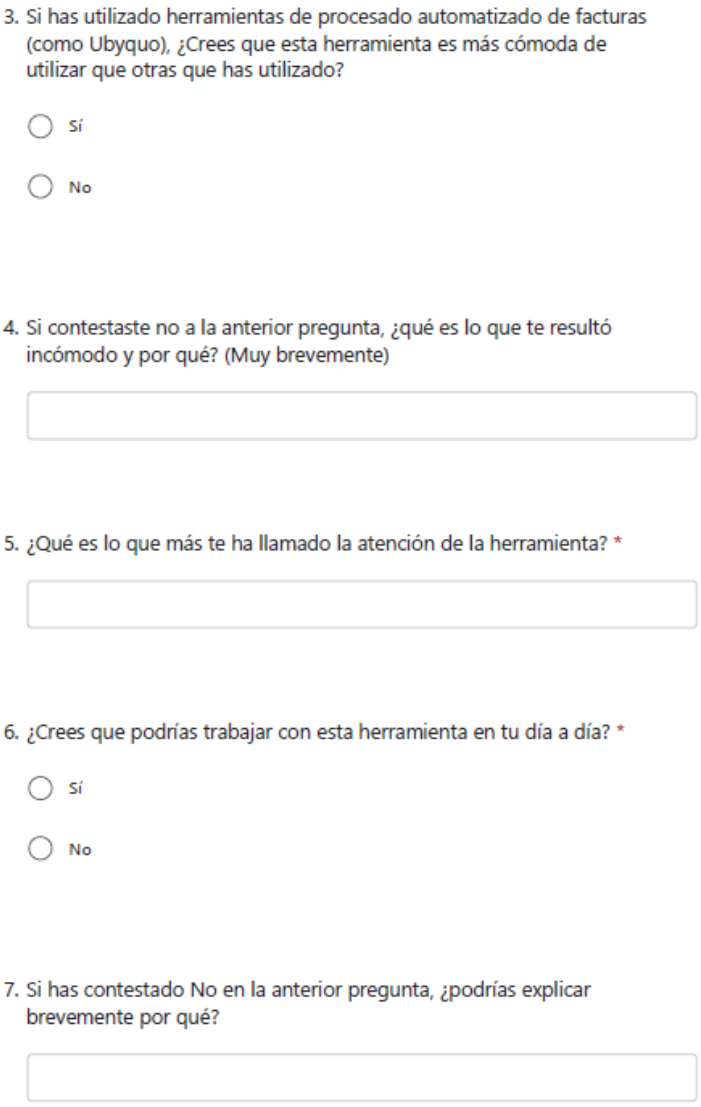


Figura A.2: Encuesta fase de pruebas

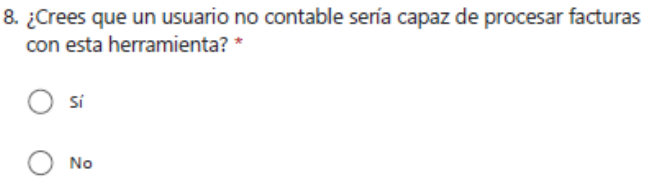


Figura A.3: Encuesta fase de pruebas

Apéndice B

Manual de usuario

B.1 Pantalla inicial

Una vez se han iniciado el *frontend* y el *backend*, se accederá a la url en la que se aloja el *frontend*.Una vez haya accedido a esta *url*, se le presentará una pantalla de Inicio que le invita a seleccionar las facturas que desea procesar. Haga clic en ”Elegir archivos” y seleccione los archivos de facturas desde su dispositivo.

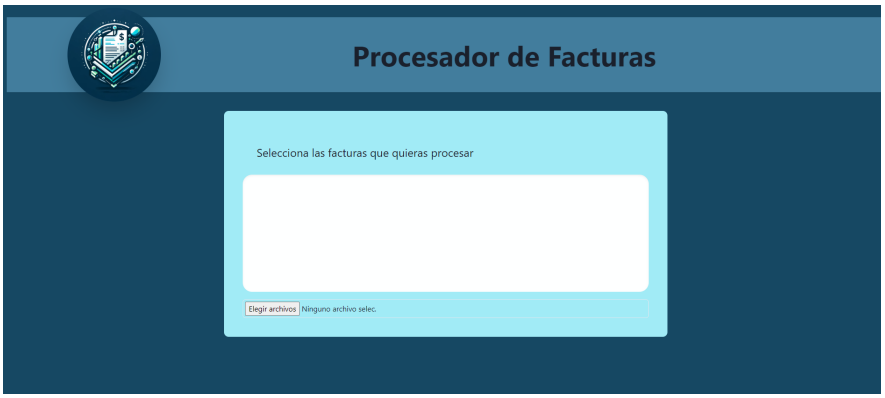


Figura B.1: Pantalla de Inicio

Finalmente, una vez subidos los archivos que se desean procesar, aparecerá un botón que nos permitirá iniciar el procesado de las facturas.

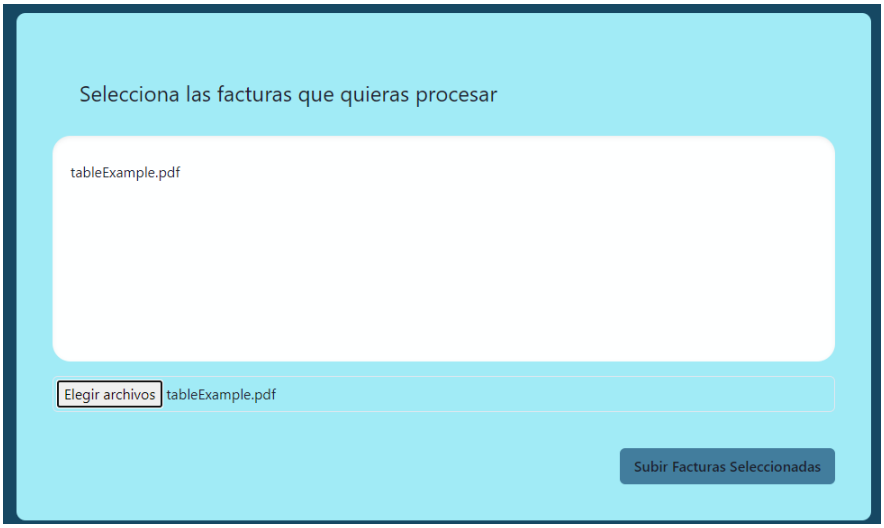


Figura B.2: Pantalla de Inicio con las facturas subidas

B.2 Pantalla de carga

Una vez iniciado el procesado, aparecerá una pantalla de carga con una barra de progreso que indica el estado del proceso de carga de las facturas al sistema.

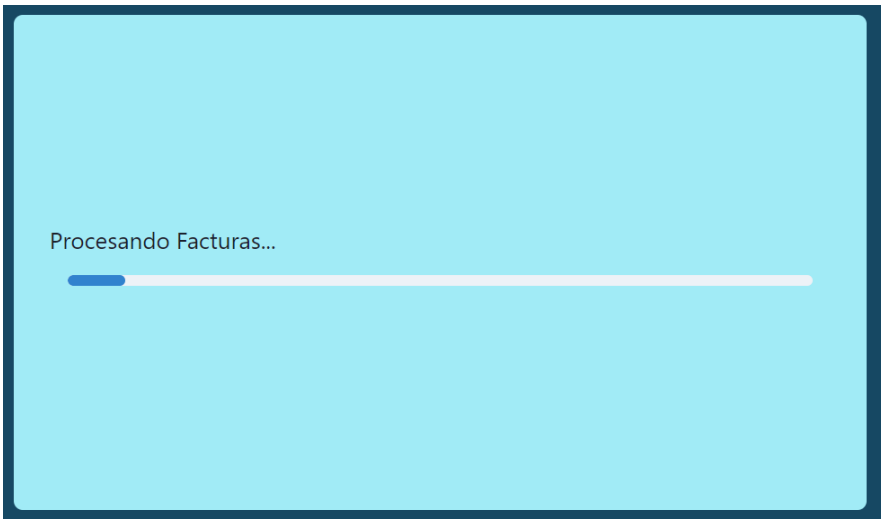


Figura B.3: Pantalla de carga

B.3 Pantalla de validación

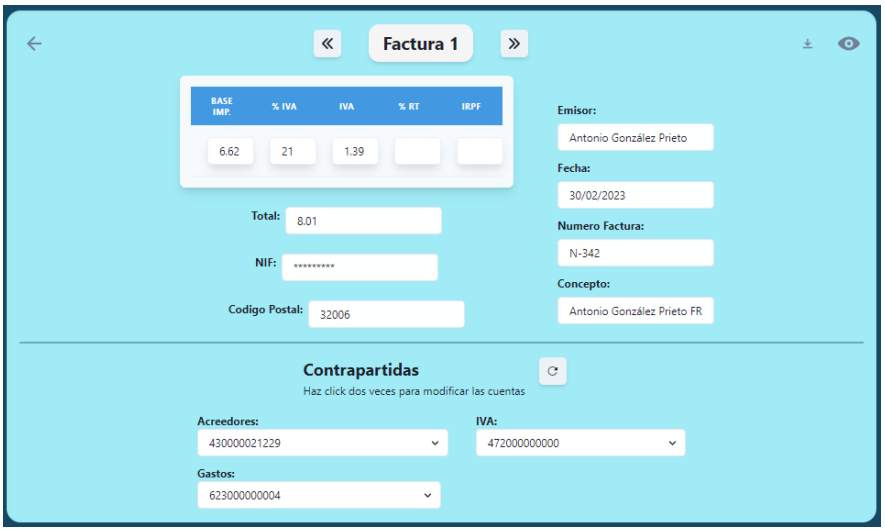


Figura B.4: Pantalla de validación

B.3.1 Visualización de la factura

Una vez que las facturas se han cargado completamente, se mostrará la primera factura, denominada *Factura 1*. En la parte superior de la pantalla, encontrará los botones para navegar entre las facturas cargadas. También encontrará a la derecha de la pantalla un botón que mostrará la factura y a la izquierda una flecha para volver a la pantalla de inicio.

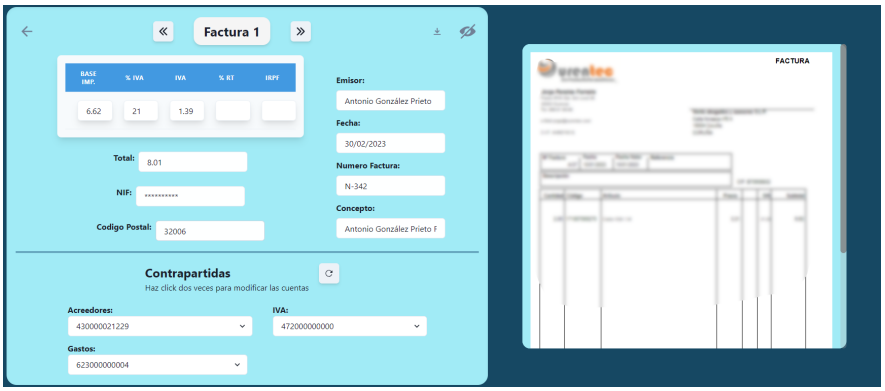


Figura B.5: Pantalla de validación con el visor de facturas activado

B.3.2 Detalles de facturación

Revise la información procesada que aparece en pantalla para asegurarse de que todos los datos de la factura son correctos. Todos los campos mostrados, incluyendo la Base Imponible,

% IVA, IVA, % RT, IRPF, y la información del emisor son completamente editables. Puede hacer clic en cualquier campo para hacer cambios si encuentra discrepancias.

Asegúrese de que el total de la factura refleje correctamente la suma de la base imponible y los impuestos calculados. Confirme que el NIF (Número de Identificación Fiscal) y el código postal son correctos. Estos campos también son editables.

B.3.3 Contrapartidas

Si es necesario ajustar las selecciones de las cuentas, haga clic en las listas desplegables para seleccionar la nueva cuenta. Si la cuenta correcta no se encontrase entre la selección, haga doble *click* sobre el campo para editar manualmente la información.

Además, esta sección cuenta con un botón de recarga de las contrapartidas. Este botón es útil si se han encontrado errores en la extracción de los emisores, ya que permitirá realizar una nueva predicción con los datos una vez validados.

Figura B.6: Editor de contrapartidas

B.4 Guardar y Descargar

Antes de descargar las facturas procesadas, el sistema le preguntará si desea guardar las cuentas de las contrapartidas. Este proceso es necesario si ha creado una nueva cuenta durante el proceso de validación. Seleccione *Sí* para guardar las cuentas o *No* si no desea conservar las modificaciones.

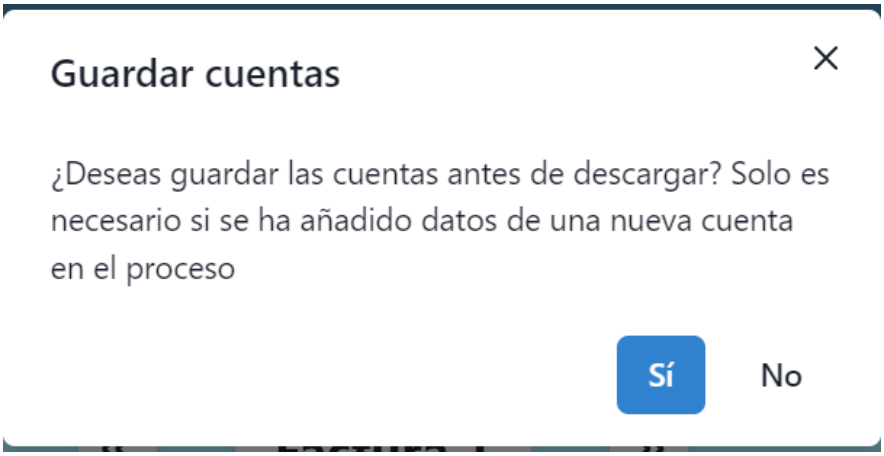


Figura B.7: *Mensaje de guardar cuentas*

Una vez seleccionada la opción, se descargarán automáticamente los asientos contables en su dispositivo en formato *excel*.

Lista de acrónimos

ANN Red de Neuronas Artificial, por sus siglas en inglés. 25

API Interfaz de Programación de Aplicaciones. 27, 55

IA Inteligencia Artificial. 1

LLM Modelo de Lenguaje de Grandes Dimensiones. 1, 2

NLP Procesamiento del Lenguaje Natural, por sus siglas en inglés. 10

RLHF Aprendizaje por Refuerzo con Retroalimentación Humana, por sus siglas en inglés. 17

Glosario

embedding En inteligencia artificial, se refiere a la representación de datos, generalmente texto o imágenes, en vectores de características en un espacio de menor dimensión. 48, 49, 53

framework Estructura conceptual y tecnológica de soporte definida, generalmente con software específico, que sirve para construir y organizar aplicaciones y sistemas. 1, 14

inteligencia artificial Software o sistemas que exhiben capacidades de percepción, razonamiento, aprendizaje y adaptación, imitando en cierta medida la inteligencia humana. Ejemplos incluyen GPT-4, sistemas de reconocimiento de voz, y robots autónomos. Estos sistemas pueden aprender de datos, identificar patrones, y tomar decisiones con cierta independencia. 2

multi-threading Técnica de programación que permite que un proceso se divida en dos o más subprocesos que se ejecutan concurrentemente, aprovechando al máximo los recursos de procesamiento de una computadora. 22

OCR Tecnología que convierte diferentes tipos de documentos, como imágenes escaneadas de texto impreso o manuscritos, en texto editable. 14, 22, 24, 44–47

OpenAI Organización de investigación en inteligencia artificial que desarrolla tecnologías avanzadas de IA, conocida por crear modelos como GPT-3 y GPT-4. 11, 17, 22, 23

redes neuronales Modelos computacionales inspirados en el funcionamiento del cerebro humano, diseñados para reconocer patrones y resolver problemas de aprendizaje automático y IA. 8

retención En contextos financieros, se refiere a la cantidad de dinero retenida antes de completar una transacción. 7, 8

stack Conjunto de tecnologías usadas para construir una aplicación. 2

Bibliografía

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need." [En línea]. Disponible en: http://arxiv.org/abs/1706.03762

[2] D. A. Wang. OpenAI text embeddings. [En línea]. Disponible en: https://www.wwwinsights.com/ai/openai-text-embedding/

[3] Cosine similarity. [En línea]. Disponible en: https://www.learndatasci.com/glossary/cosine-similarity/

[4] Retrieval. [En línea]. Disponible en: https://blog.langchain.dev/retrieval/

[5] A comprehensive overview of large language models. [En línea]. Disponible en: https://ar5iv.labs.arxiv.org/html/2307.06435

[6] Y. Guo, Y. Liang, C. Wu, W. Wu, D. Zhao, and N. Duan, "Learning to plan with natural language." [En línea]. Disponible en: http://arxiv.org/abs/2304.10464

[7] 3.11.7 documentation. [En línea]. Disponible en: https://docs.python.org/3.11/

[8] React. [En línea]. Disponible en: https://es.react.dev/

[9] Project jupyter. [En línea]. Disponible en: https://jupyter.org

[10] OpenAI API. [En línea]. Disponible en: https://openai.com/blog/openai-api

[11] GitHub: Let's build from here. [En línea]. Disponible en: https://github.com/

[12] GitHub desktop. [En línea]. Disponible en: https://desktop.github.com/

[13] Visual studio code - code editing. redefined. [En línea]. Disponible en: https://code.visualstudio.com/

[14] Docker: Accelerated container application development. [En línea]. Disponible en: https://www.docker.com/

[15] Qdrant - vector database. [En línea]. Disponible en: https://qdrant.tech/

[16] start [zotero documentation]. [En línea]. Disponible en: https://www.zotero.org/support/

[17] Reglamento general de protección de datos. [En línea]. Disponible en: https://www.consilium.europa.eu/es/policies/data-protection/data-protection-regulation/

[18] Introducing the GPT store. [En línea]. Disponible en: https://openai.com/blog/introducing-the-gpt-store

[19] OpenAI | security portal. [En línea]. Disponible en: https://trust.openai.com/

[20] D. Glukhov, I. Shumailov, Y. Gal, N. Papernot, and V. Papyan, "LLM censorship: A machine learning challenge or a computer security problem?" [En línea]. Disponible en: http://arxiv.org/abs/2307.10719

[21] Y. Yao, J. Duan, K. Xu, Y. Cai, E. Sun, and Y. Zhang, *A Survey on Large Language Model (LLM) Security and Privacy: The Good, the Bad, and the Ugly.*

[22] T. Y. Zhuo, Y. Huang, C. Chen, and Z. Xing, "Red teaming ChatGPT via jailbreaking: Bias, robustness, reliability and toxicity." [En línea]. Disponible en: http://arxiv.org/abs/2301.12867