UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Aplicación de gestión y análisis de documentos de contratación pública en las fases de diseño y construcción mediante el uso de técnicas de IA

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Ramalho, Marta Filipa

Tutor/a: Heras Barberá, Stella María

Cotutor/a: Montalbán Domingo, María Laura

CURSO ACADÉMICO: 2023/2024

Dedicatoria

A mi abuela, por la ilusión que le hacía verme terminar la carrera. Aunque no pudiste ver el final de este proyecto, te lo dedico allá donde estés.

Agradecimientos

En primer lugar, me gustaría agradecer a mis padres y a mi hermano, por el apoyo moral y económico que me han dado a lo largo de estos 4 años. Sin su ayuda financiera no habría podido mudarme a una ciudad tan lejos de casa a estudiar lo que me apasiona en una buena universidad.

En segundo lugar, quiero agradecer a Gabriel Rodríguez Díaz por aguantarme y ayudarme siempre incluso en los peores momentos. También por ser mi fiel compañero de madrugadas sin dormir estudiando en época de exámenes.

Por último, agradezco profundamente a mis dos tutoras por su ayuda y apoyo a lo largo del desarrollo de este proyecto. Siempre han estado disponibles para resolver cualquier duda y ofrecer la orientación necesaria.

Resumen

Este trabajo presenta SmarTenderAI, una herramienta diseñada para enfrentar los desafíos de la gestión y análisis de licitaciones públicas, caracterizados frecuentemente por el manejo de grandes volúmenes de información. SmarTenderAI automatiza la extracción de datos de la Página de Contratación del Sector Público, tanto desde el HTML de cada licitación como de documentos PDF, utilizando Python. Además, incorpora un modelo de lenguaje avanzado, GPT-4o-mini, que mejora significativamente la precisión en la interpretación de datos complejos y contextuales, como las valoraciones de empresas y los criterios de evaluación.

El desarrollo de SmarTenderAI integró diversas tecnologías. Los datos extraídos se almacenan de manera segura en una base de datos SQL Server, mientras que su visualización y análisis se facilitan mediante una aplicación web interactiva desarrollada con React y Django. Esta aplicación permite a los usuarios no solo visualizar la información de forma clara y accesible, sino también acceder a estadísticas detalladas y exportar los datos en formato Excel.

Las pruebas realizadas y las estadísticas calculadas aseguran el correcto funcionamiento de la herramienta, aunque también existen algunas áreas donde la precisión en la extracción de ciertas variables podría mejorarse. Estas observaciones subrayan el potencial continuo de SmarTenderAI como una herramienta eficaz para organizaciones que buscan optimizar sus participaciones en la contratación pública, con margen para futuras mejoras.

Palabras clave: LLM, Licitaciones, Contratación Pública, NLP, Análisis de datos, Desarrollo Web, HTML Scraping, PDF Parsing

Abstract

This project presents SmarTenderAI, a tool designed to address the challenges of managing and analyzing public tenders, often characterized by handling large volumes of information. SmarTenderAI automates the extraction of data from the Public Sector Procurement Website, both from the HTML of each tender and from PDF documents, using Python. Additionally, it incorporates an advanced large language model, GPT-4o-mini, which significantly improves accuracy in interpreting complex and contextual data, such as company evaluations and evaluation criteria.

The development of SmarTenderAI integrated various technologies. The extracted data is securely stored in a SQL Server database, while its visualization and analysis is accessible through an interactive web application developed with React and Django. This application allows users not only to visualize the information clearly but also to access detailed statistics and export the data in Excel format.

The tests conducted and the calculated statistics ensure the correct functioning of the tool, although there are also some areas where the accuracy in extracting certain variables could be improved. These observations highlight the ongoing potential of SmarTenderAI as an effective tool for organizations seeking to optimize their participation in public procurement processes, with room for future improvements.

Key words: LLM, Tenders, Public Procurement, NLP, Data Analysis, Web Development, HTML Scraping, PDF Parsing

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1
Introducción

En el contexto actual, la gestión eficiente de la información es un componente crucial para el éxito en diversas áreas, incluyendo la administración pública y la adjudicación de contratos. La necesidad de manejar grandes volúmenes de datos de manera eficaz es más relevante que nunca, especialmente en un entorno donde la precisión y la rapidez en la toma de decisiones pueden marcar la diferencia.

El problema global que aborda este trabajo se centra en la necesidad de un sistema integrado que permita la extracción y análisis de información relevante de licitaciones. Tradicionalmente, este proceso puede ser manual, tedioso y propenso a errores, especialmente cuando se manejan documentos en formatos no estructurados o poco homogéneos. La solución propuesta busca superar estos desafíos mediante la implementación de un sistema automatizado que combine técnicas avanzadas de procesamiento de datos con una base de datos robusta y una aplicación web interactiva. El nombre otorgado a esta herramienta es **SmarTenderAI**.

Para su desarrollo se han ido estableciendo objetivos progresivos a cumplir durante el desarrollo. Estos objetivos se han ido revisando con reuniones semanales con las tutoras a modo de seguimiento y supervisión. Además, todo el código del proyecto se encuentra en un repositorio de Github[1] en el que se han ido subiendo las diferentes versiones de los componentes del proyecto para el control de versiones.

1.1 Motivación

Las agencias públicas encargadas del desarrollo de infraestructuras de transporte están demandando el desarrollo de modelos digitales de gestión que optimicen la contratación en las fases de diseño y construcción. El objetivo de dichos modelos es reducir los sobrecostes y retrasos en el diseño y construcción de dichas infraestructuras.

El proyecto **OPTI-CON**, propuesta del Departamento de Ingeniería de la Construcción y de Proyectos de Ingeniería Civil y desarrollado conjuntamente con el Instituto Valenciano para la Inteligencia Artificial (VRAIN), surge como una respuesta a esta necesidad, proponiendo el desarrollo de un modelo digital basado en técnicas de optimización de modo que sea capaz de buscar las mejores condiciones de contratación en las fases de diseño y construcción con el objetivo de minimizar la ocurrencia de riesgos y mejorar el resultado final dichas fases.

En este contexto, mi Trabajo de Fin de Grado (TFG) se presenta como un paso previo crucial para la implementación del proyecto OPTI-CON. Este trabajo se centra en la

[1] https://github.com

extracción y visualización de datos de licitaciones, información esencial que servirá de base para el desarrollo de los modelos digitales de gestión propuestos en OPTI-CON. La capacidad de extraer datos precisos y organizados de las licitaciones permitirá alimentar los modelos de predicción, asegurando que sean rigurosos y precisos.

La motivación para llevar a cabo este TFG radica en la posibilidad de contribuir significativamente a la mejora de los procesos de contratación de las entidades públicas. Al proporcionar una base sólida de datos estructurados y visualizados adecuadamente, este trabajo apoya el desarrollo de modelos de gestión más eficientes y precisos.

1.2 Objetivos

El propósito de este Trabajo de Fin de Grado (TFG) es desarrollar una solución integral para la extracción, análisis y gestión de información de licitaciones a partir de documentos en formatos HTML y PDF. Este sistema busca mejorar la eficiencia en el procesamiento de datos y facilitar la toma de decisiones en el contexto de adjudicación de contratos. Los objetivos específicos del trabajo son los siguientes:

1. **Analizar los sistemas actuales de gestión de datos de contratación pública**: Hacer un estudio de las diferentes herramientas ya existentes y disponibles de gestión de datos de contratación pública y cómo se diferencia la herramienta propuesta.

2. **Desarrollar un módulo de extracción de datos**: Crear un sistema automatizado para extraer información clave de documentos HTML y PDF relacionados con licitaciones, incluyendo la capacidad de manejar diferentes formatos y estructuras documentales.

3. **Integrar un modelo de lenguaje avanzado**: Implementar un modelo de lenguaje (LLM) para extraer y analizar información compleja y contextualizada, como valoraciones y criterios de evaluación, mejorando así la precisión en la interpretación de los datos.

4. **Diseñar y construir una base de datos SQL**: Crear una base de datos eficiente y segura para almacenar la información extraída. La base de datos debe permitir una gestión efectiva de los datos y garantizar su integridad y consistencia.

5. **Desarrollar una aplicación web interactiva**: Construir una interfaz web intuitiva y accesible para la visualización y gestión de los datos almacenados. La aplicación debe incluir funcionalidades para búsqueda, filtrado y visualización de detalles y estadísticas relacionadas con las licitaciones.

6. **Hacer uso de tecnologías avanzadas**: Emplear tecnologías modernas y actualizadas durante el desarrollo, asegurando no solo su correcta aplicación, sino también un aprendizaje profundo de las mismas. Esto garantiza que la solución desarrollada utilice herramientas y metodologías actuales y eficaces, manteniendo su relevancia y potencia en un entorno tecnológico en constante evolución.

7. **Asegurar la calidad y consistencia de los datos**: Establecer mecanismos para la validación y normalización de los datos extraídos, asegurando que la información almacenada sea completa, consistente y confiable.

8. **Realizar pruebas exhaustivas del sistema**: Llevar a cabo pruebas unitarias, de integración y de usuario para garantizar que todos los componentes del sistema funcionen correctamente y cumplan con los requisitos establecidos.

9. **Documentar el proceso de desarrollo y resultados**: Elaborar una documentación detallada del proceso de desarrollo, incluyendo el diseño técnico, las pruebas realizadas y los resultados obtenidos, para servir como referencia para futuras mejoras y desarrollos.

10. **Diseñar un sistema escalable y mantenible**: Asegurar que la solución desarrollada sea escalable y fácil de mantener, permitiendo adaptaciones y actualizaciones futuras sin comprometer la funcionalidad del sistema.

11. **Ofrecer una forma de exportación de los datos para su uso en otras herramientas**: Proporcionar una funcionalidad que permita la exportación de los datos procesados en formatos compatibles con herramientas de cálculo.

Estos objetivos proporcionan una guía clara para el desarrollo del proyecto, asegurando que la solución propuesta cumpla con las expectativas y necesidades identificadas.

1.3 Metodología de Trabajo

El desarrollo del proyecto siguió un enfoque secuencial y estructurado. Se establecieron objetivos progresivos para cada fase, los cuales fueron revisados en reuniones semanales con las tutoras para asegurar un seguimiento y supervisión adecuados. La gestión de tareas se realizó mediante Trello[2], lo que permitió un seguimiento claro del progreso. Además, todo el código del proyecto se almacenó en un repositorio de GitHub[3], facilitando el control de versiones y la actualización de los diferentes componentes del proyecto. La comunicación continua con las tutoras, a través de reuniones y la plataforma Teams[4], permitió una retroalimentación constante y la resolución rápida de problemas, asegurando el avance del proyecto.

1.4 Estructura de la memoria

El documento está organizado en nueve capítulos, que describen detalladamente el proceso llevado a cabo durante este proyecto para acabar teniendo la herramienta esperada, cumpliendo con los objetivos propuestos. La estructura se presenta de la siguiente manera:

1. **Introducción**: En esta sección se hace una pequeña introducción al proyecto y se comenta la motivación que ha llevado al desarrollo del proyecto, así como los objetivos específicos que el proyecto busca alcanzar.

2. **Estado del Arte**: En este apartado se presentan y comentan varios conceptos clave como las plataformas de contratación y los modelos de lenguaje. También se hace un análisis extensivo de herramientas existentes similares a SmarTenderAI. Se desarrolla una crítica a estas herramientas destacando las mejoras que introduce la aplicación SmarTenderAI en comparación con las demás plataformas. Por último, se hace una presentación general de la propuesta, explicando su funcionamiento e innovación.

[2] https://trello.com
[3] https://github.com
[4] https://www.microsoft.com/es-co/microsoft-teams

3. **Análisis del problema**: En esta sección se lleva a cabo un análisis detallado del problema que se aborda con el proyecto. Primero, se presenta la especificación de requisitos, distinguiendo entre requisitos funcionales, que definen las funciones que la herramienta debe cumplir, y requisitos no funcionales, que establecen las cualidades y restricciones de rendimiento, usabilidad y seguridad. A continuación, se realizan varios análisis importantes: análisis energético, análisis del marco legal y ético y análisis de riesgos.

   - **Análisis Energético**: Se evalúa el costo energético asociado con el entrenamiento de un modelo de lenguaje desde cero y se justifica la elección de un modelo preentrenado para minimizar el impacto energético.
   - **Análisis del Marco Legal y Ético**: Se revisan las normativas legales y éticas relevantes que afectan el desarrollo y uso de la herramienta, incluyendo la protección de datos y la privacidad y el uso de datos abiertos del Estado.
   - **Análisis de Riesgos**: Se identifican y analizan los posibles riesgos asociados con el desarrollo y la implementación de la herramienta, así como las estrategias para mitigar dichos riesgos.

   Se exploran y justifican las posibles soluciones al problema, destacando las razones detrás de las decisiones tomadas. Posteriormente, se describe en detalle la solución propuesta, SmarTenderAI, explicando el enfoque adoptado y lo que se llevará a cabo en cada fase del proyecto.

4. **Tecnologías**: Se describen las tecnologías empleadas en el desarrollo de la herramienta. Se incluyen las bibliotecas de Python más relevantes para el módulo de extracción de datos, las tecnologías utilizadas para desarrollar la aplicación web y la base de datos seleccionada para almacenar la información.

5. **Diseño de la solución**: En esta sección se aborda el diseño detallado de la herramienta, que incluye varios aspectos importantes:

   - **Análisis de Variables a Extraer**: Se realiza un análisis exhaustivo de las variables que deben ser extraídas de los documentos de licitación. Se identifica la ubicación de cada sección en el HTML de una licitación y se examina cómo se presentan y estructuran los datos en el documento.
   - **Módulo de Extracción**: Se describe el diseño del módulo de extracción de datos, destacando:
     - **Precondiciones**: Se enumeran las condiciones necesarias que deben cumplirse para que el módulo pueda procesar una licitación de manera efectiva.
     - **Entradas y Salidas**: Se especifican los tipos de datos que el módulo recibe como entrada y los resultados que genera como salida. Esto incluye detalles sobre el formato y la estructura de los datos procesados.
   - **Diseño de la Base de Datos**: Se presenta el diseño conceptual de la base de datos que almacena la información extraída. Se detalla la estructura de la base de datos, incluyendo las tablas, relaciones entre ellas, y las claves primarias y foráneas necesarias para asegurar la integridad y accesibilidad de los datos.
   - **Arquitectura de la Aplicación Web**: Se describe la arquitectura general de la aplicación web, incluyendo tanto el frontend como el backend. Se explican los componentes principales de cada parte y cómo interactúan entre sí para proporcionar la funcionalidad completa de la herramienta.

6. **Desarrollo e Implementación de la solución**: En esta sección se describe el proceso de desarrollo e implementación de la herramienta. Se detalla la construcción de cada componente de la solución, incluyendo el módulo de extracción de datos, la base de datos y la aplicación web. Se explican las técnicas y tecnologías utilizadas, los desafíos encontrados durante el desarrollo y las soluciones aplicadas. Además, se abordan los pasos seguidos para integrar todos los componentes y poner en funcionamiento la herramienta de manera efectiva.

7. **Pruebas**: Esta sección cubre las pruebas realizadas para garantizar el correcto funcionamiento de la herramienta. Se describen los diferentes tipos de pruebas llevadas a cabo, como pruebas unitarias, pruebas de integración y las estadísticas calculadas para una serie de variables que nos permite evaluar el sistema desarrollado.

8. **Conclusiones**: En esta sección se presentan las conclusiones finales tras finalizar el desarrollo del proyecto. Se resumen los principales logros y beneficios de la solución implementada, destacando cómo se han cumplido los objetivos del proyecto. También se reflexiona sobre las lecciones aprendidas y la efectividad de la herramienta en la resolución del problema planteado.

9. **Trabajos futuros**: Esta sección propone posibles direcciones para futuros desarrollos y mejoras de la herramienta. Se sugieren nuevas funcionalidades, optimizaciones y expansiones que podrían realizarse para incrementar la utilidad y el rendimiento de la herramienta. También se discuten áreas de investigación adicionales y posibles colaboraciones para avanzar en el proyecto.

CAPÍTULO 2
Estado del arte

Actualmente las agencias públicas ofrecen datos de las distintas licitaciones en formato digital, almacenando un conjunto de datos abiertos al público para facilitar la accesibilidad y transparencia. Sin embargo, estos datos se encuentran estructurados en diferentes documentos PDF que necesitan ser procesados y estructurados.

La enorme cantidad de datos disponibles y la heterogeneidad de los formatos hace que esta información, aún siendo pública, sea difícilmente manejable por los usuarios de la plataforma, reduciendo enormemente su utilidad y posibles aplicaciones subyacentes.

Una posible mejora en este aspecto podría ser la incorporación de la inteligencia artificial (IA) y los modelos de lenguaje de gran tamaño (LLMs) para automatizar y optimizar la extracción y estructuración de estos datos complejos, para su posterior procesado y visualización. Este estado del arte revisa las herramientas y metodologías actuales en este área, así como las innovaciones y desafíos presentes.

2.1 Plataformas de contratación

Existen plataformas cuyo fin es gestionar y mostrar de forma transparente la información relacionada con diferentes licitaciones y contratos públicos. Estas plataformas están gestionadas por entidades gubernamentales que se encargan de añadir los diferentes expedientes con sus documentos e información relacionados. En la actualidad, la disponibilidad de las plataformas y su tipología varía según los diferentes países. Aunque trabajaremos únicamente con la plataforma de contratación del sector público español, también cabe mencionar la existencia de la plataforma referente a las licitaciones de la Unión Europea, que podría beneficiarse de la tecnología desarrollada en el presente proyecto.

2.1.1. TED (Tenders Electronic Daily)

Es la página dedicada a la contratación pública europea. Esta plataforma contiene información sobre licitaciones de contratación pública de acuerdo con las normas de la UE. TED [1] proporciona una base de datos que incluye detalles sobre licitaciones, adjudicaciones y contratos públicos de países pertenecientes al Espacio Económico Europeo (EEE) y otros países europeos como Suiza. Esto permite a las empresas y a los ciudadanos acceder a una gran cantidad de información relevante para oportunidades comerciales y de inversión. Los avisos publicados en TED abarcan gran variedad de sectores y regiones, lo que facilita la búsqueda de oportunidades específicas para diferentes tipos de negocios y localizaciones geográficas.

2.1.2. Plataforma de Contratación del Sector Público

La Plataforma de Contratación del Sector Público, PLACSP [2] , anteriormente llamada Plataforma de Contratación del Estado, PLACE, es un buscador de licitaciones públicas en el ámbito nacional y será la plataforma utilizada para el desarrollo de este trabajo. Es un buscador gratuito y muy completo en diversos aspectos que permite a los usuarios visualizar y buscar toda la información relacionada con licitaciones, aunque ciertos datos pueden ser difíciles de extraer debido a la estructura en la que se presenta la información.

Funcionalidades y ventajas

Es el principal punto de acceso a la información sobre la actividad contractual del Sector Público, facilitando el acceso a la información sobre las convocatorias de licitaciones y los resultados de todos los organismos que lo componen. La plataforma permite a los usuarios buscar y visualizar detalles de licitaciones, ofreciendo un motor de búsqueda que permite filtrar en base a una amplia gama de criterios, como el tipo de contrato, la entidad contratante, y el estado del procedimiento. Estos conjuntos de datos abiertos están relacionados con los órganos de contratación cuyo perfil de contratante se aloja en la plataforma.

Integración con Comunidades Autónomas

Además de centralizar la información de los organismos estatales, PLACSP también hace público un conjunto de datos referente a las licitaciones recibidas a través del mecanismo de agregación de las Comunidades Autónomas que tienen sus propias plataformas. Esto asegura que los usuarios puedan acceder a una visión integral y unificada de las oportunidades de contratación pública a nivel nacional y regional, sin tener que consultar múltiples fuentes.

Desafíos y Mejoras

Aunque PLACSP es una herramienta muy completa, también presenta ciertas carencias y desafíos, principalmente debido a la forma en la que se estructura la información. La inconsistencia de los documentos a la hora de redactar la información puede ocasionar problemas a la hora de extraerlos de forma 'tradicional', es decir, separando por secciones el texto y extrayendo el contenido de cada sección de forma automática. Esto puede presentar problemas para la extracción de alguna información ya que no siempre se encuentra dentro de la misma sección o no se presenta de la misma forma. Un ejemplo claro es el caso del apartado 'Determinación del precio', en el que el sistema de precios puede aparecer de las siguientes fromas:

- Escrito directamente el que corresponde.
- Lista con todos los sistemas y se marca con una X el elegido.
- Lista con todos los sistemas y cada uno contiene una casilla de 'Sí' y 'No' en la que se marca una de las dos para cada uno.

Esto dificulta su extracción automática ya que no se sabe el formato que tomará esa información si no se revisa manualmente de antemano.

Además, hay también una inconsistencia en cuanto al formato de los diferentes documentos PDF, ya que en algunos casos se presentan como documentos escritos y en otros

casos se presentan como documentos escaneados de los cuales no se puede extraer el texto sin la utilización de un procedimiento adicional de reconocimiento óptico de caracteres (OCR). Para abordar este problema, se está explorando el uso de tecnologías avanzadas como la inteligencia artificial (IA) y los modelos de lenguaje de gran tamaño (LLMs) para mejorar la extracción y estructuración de la información de manera más eficiente y precisa.

2.2 Modelos de Lenguaje de Gran Tamaño (LLM)

Los Modelos de Lenguaje de Gran Tamaño (Large Language Models, LLMs) han revolucionado el campo de la inteligencia artificial y el procesamiento del lenguaje natural (NLP) en los últimos años. Basados en arquitecturas de redes neuronales profundas, estos modelos son capaces de comprender, generar y manipular texto con un alto grado de coherencia y precisión. A medida que han evolucionado, han encontrado aplicaciones en una amplia gama de dominios, incluyendo la generación de contenido, la traducción automática, la asistencia virtual y el procesamiento de texto.

Hoy en día existen diversos LLMs con diferentes características, unos más potentes que otros, unos de código abierto y otros no. Los principales LLMs de la actualidad son los siguientes:

2.2.1. Llama 2

Llama 2 [3] es una familia de modelos de lenguaje preentrenados y afinados desarrollada por Meta [4], que abarca desde los siete mil millones hasta los 70 mil millones de parámetros. Estos modelos se han destacado por su competitividad frente a los modelos de chat de código abierto existentes y han demostrado una competencia comparable a algunos modelos propietarios en diversas evaluaciones. Sin embargo, aún no alcanzan el nivel de otros modelos más avanzados como GPT-4 .

Uno de los aspectos más importantes de Llama 2 es su diseño orientado a la utilidad y la seguridad. Los métodos y técnicas utilizados en su desarrollo han sido elaborados con meticulosidad para garantizar que los modelos no solo sean efectivos, sino también seguros y útiles para los usuarios .

La familia de modelos de Llama 2 se han hecho accesibles al público. Esto permite a la comunidad científica y a los desarrolladores explorar y utilizar estos modelos para una variedad de aplicaciones.

2.2.2. Llama 3

El 18 de abril de 2024, Meta hizo público Llama 3 [5], un nuevo modelo que representa una mejora significativa respecto a su predecesor, Llama 2. Al igual que sus versiones anteriores, Llama 3 es de código abierto, lo que permite a los usuarios utilizarlo en una variedad de contextos y fomenta la innovación en la comunidad de IA.

Llama 3 presenta modelos con ocho mil millones y 70 mil millones de parámetros, los cuales han demostrado un rendimiento sólido en diversos benchmarks industriales. Además, ofrece capacidades mejoradas, como un razonamiento más sólido.

El objetivo de Llama 3 es el de ser el mejor modelo de código abierto hasta la fecha, centrándose en ser multilingüe y multimodal, con ventanas de contexto más amplias y un rendimiento mejorado en áreas clave como el razonamiento y la generación de código.

2.2.3. GPT 3

GPT-3 [6], abreviatura de 'Generative Pre-trained Transformer 3', es un modelo de lenguaje desarrollado por OpenAI [7] que ha revolucionado el campo del NLP. Con un tamaño de 175 mil millones de parámetros, GPT-3 ha establecido nuevos estándares en la capacidad de generación de texto y comprensión del lenguaje.

Este modelo ha destacado por su capacidad para producir texto coherente y relevante en una amplia variedad de contextos y dominios. Desde la redacción de artículos hasta la generación de código, GPT-3 ha demostrado ser versátil y capaz de adaptarse a diversas tareas con un mínimo de ajustes.

Sin embargo, GPT-3 también enfrenta desafíos significativos. Se ha señalado que el modelo puede generar respuestas incoherentes o sesgadas en ciertos contextos, lo que plantea preocupaciones sobre su fiabilidad y ética. Además, su tamaño masivo y los recursos computacionales necesarios para su entrenamiento y ejecución pueden limitar su accesibilidad y escalabilidad en ciertos entornos.

GPT 3.5

GPT-3.5 es una versión mejorada de GPT-3, lanzada por OpenAI en 2022. Esta actualización ha incorporado varias mejoras en términos de eficiencia y precisión. GPT-3.5 ha refinado su capacidad para generar respuestas más coherentes y menos sesgadas, abordando algunas de las críticas dirigidas a su predecesor. Además, este modelo ha mejorado su rendimiento en tareas de generación de código y procesamiento de lenguaje natural en múltiples idiomas, lo que lo hace aún más versátil y robusto para aplicaciones diversas.

2.2.4. GPT 4

GPT-4 [8] es la iteración más avanzada de la serie GPT, lanzada por OpenAI en 2023. Con un impresionante tamaño de 500 mil millones de parámetros, GPT-4 ha superado significativamente a sus predecesores en términos de capacidad y rendimiento. Este modelo ha mejorado considerablemente en áreas como el razonamiento lógico, la generación de texto creativo y la comprensión contextual profunda. Gracias a estas mejoras, GPT-4 puede abordar tareas más complejas y ofrecer respuestas más precisas y coherentes, lo que amplía su aplicabilidad en diversos dominios.

Además de las mejoras en la generación de texto y la comprensión, GPT-4 ha incorporado técnicas avanzadas para minimizar sesgos y mejorar la seguridad. Esto incluye la implementación de métodos más sofisticados de filtrado de contenido y algoritmos que reducen la probabilidad de respuestas inapropiadas o dañinas, cosa que sí pasaba en GPT-3 o GPT-3.5.

GPT-4 también destaca por su capacidad para manejar tareas multimodales, lo que le permite interpretar y generar contenido en múltiples formatos, como texto, imágenes y audio. Esta capacidad lo hace ideal para aplicaciones avanzadas de inteligencia artificial, como asistentes virtuales más intuitivos, sistemas de generación de contenido multimedia y plataformas de análisis de datos más integrales. Además, a pesar de su mayor tamaño y complejidad, ha sido optimizado para funcionar de manera más eficiente, facilitando su implementación en entornos con limitaciones de hardware sin comprometer el rendimiento.

2.3 Herramientas similares

La herramienta a desarrollar en este trabajo tiene como objetivo facilitar extracción, procesamiento y visualización de los datos de las licitaciones a partir de la extracción y estructuración de los mismos. Actualmente existen diversas herramientas que cumplen una función similar a la que se pretende desarrollar, pero con diferencias significativas. Estas herramientas pueden variar en diversos aspectos, pero principalmente en la cantidad de datos que se extraen de las licitaciones y la forma de llevarlo a cabo.

En este contexto, el desarrollo de esta nueva herramienta para la visualización de datos busca llenar un espacio específico en el mercado, mostrando datos que son difíciles de extraer. De esta forma se facilita el acceso a los usuarios, sin necesidad de consultar diversos documentos para encontrar una información específica. Además, se busca crear una interfaz que facilite su interpretación y observación. A través de un enfoque centrado en el usuario y la incorporación de funcionalidades innovadoras, se espera que esta herramienta pueda proporcionar una experiencia mejorada para aquellos que necesitan acceder y analizar datos de licitaciones de manera efectiva y eficiente.

2.3.1. Gobierto

Gobierto es una empresa especialista en diseño de productos digitales aplicados al ámbito de la administración pública, el gobierno abierto, la visualización y los datos. Definen su objetivo principal como el de mejorar los servicios de la administración pública. Para ello, desarrollan diversas herramientas con funcionalidades dirigidas a las empresas para simplificar el análisis de las licitaciones y facilitar así la presentación de ofertas por parte de estas empresas.

Gobierto Contratación

Gobierto Contratación [9] son una serie de herramientas desarrolladas para mejorar los procesos de contratación pública para empresas y administración pública.

Por un lado la empresa ofrece una herramienta de **Control y Planificación** [10] de la contratación de entidades públicas, una herramienta cloud que permite a las entidades públicas realizar un control en los contratos en curso y la planificación y preparación de licitaciones. Esta herramienta permite realizar la gestión y control de los contratos en curso incorporando la planificación como una herramienta de gestión. Entre sus funcionalidades destacan:

- Obtener una visión completa del estado de todos los contratos en curso.
- Permite editar notificaciones que se desean recibir sobre fechas de finalización y otros aspectos.
- Contiene un modelo de predicción que estima el número de ofertas que recibirá una licitación en base a sus características y al histórico de su base de datos.

Por otro lado se ofrece un **Explorador de datos de contratación pública** [11] que incluye un buscador global de licitaciones y adjudicaciones de todo el estado. En este buscador se permite filtrar las licitaciones por diferentes conceptos tales como por ejemplo ”Inteligencia Artificial” en lugar de necesitar buscar un número de expediente concreto como en PLACSP. Así, permite obtener una lista de varias licitaciones relacionadas con un concepto en el que estamos interesados para facilitar el proceso de participación de las empresas.

Este explorador reúne la información de la Plataforma de Contratación del Estado y de las principales plataformas autonómicas y permite visualizar y analizar los datos con diferentes fines. Además, la búsqueda avanzada permite filtrar por múltiples criterios tales como palabra clave, adjudicador, adjudicatario, tipo de entidad, fecha, procedimiento, lugar de ejecución, etc.

El explorador también permite consultar los pliegos de una licitación directamente desde la herramienta, sin necesidad de consultar documentos externos. Además, permite formar equipos de trabajo dentro de la herramienta y escribir anotaciones en los diferentes documentos que podrán ser consultadas por todo el equipo.

Este buscador tiene como objetivo también el de permitir a las diferentes empresas hacer un seguimiento de las licitaciones en las que han participado y recibir notificaciones sobre los resultados finales, para poder así analizar también a sus competidores. Permite monitorizar la competencia y analizar el tipo de licitaciones en las que participan y las que ganan, pudiendo así mejorar la estrategia interna de la empresa. También disponen de la funcionalidad de predicción para poder estimar las licitaciones a las que se presentarán los competidores.

Por último, esta herramienta permite también la descarga de los datos y resultados en diferentes formatos para facilitar su análisis en otras herramientas, proporcionando también la opción de acceder a ellos mediante una API que se podría integrar en aplicaciones externas.

En resumen, Gobierto Contratación recopila la misma información que PLACSP pero facilitando su visualización e incorporando modelos de IA que permiten hacer estimaciones sobre el número de ofertas presentadas y la calidad de las mismas. Es una herramienta muy completa y útil tanto para las empresas para poder hacer seguimiento de licitaciones y analizar a la competencia, como para la contratación pública para poder gestionar las licitaciones y todo el proceso de adjudicación con una interfaz intuitiva y orientada al usuario. Sin embargo, esta herramienta no extrae todos los datos de las licitaciones, sino que extrae únicamente la información básica de las licitaciones como el importe, el adjudicatario, las diferentes fechas y el número de licitadores, y reúne los documentos PDF en el que se encuentra el resto de información para que el usuario los pueda consultar, pero no extrae información de estos documentos PDF, sino que el usuario debe procesarlos de forma manual.

2.3.2. OpenPLACSP

OpenPLACSP [12] es una herramienta desarrollada por la Subdirección General de Coordinación de la Contratación Electrónica [13] cuyo objetivo es facilitar la transformación de los ficheros de datos abiertos en un documento de hoja de cálculo con los principales datos de las licitaciones seleccionadas.

Con esta herramienta se pretende que cualquier interesado pueda trabajar de una forma rápida y sencilla con los datos abiertos que se ponen a disposición. OpenPLACPS es una herramienta que puede ser ejecutada en entornos Windows o Linux que dispongan de una máquina virtual Java.

Para su utilización es necesaria la adquisición de datos abiertos que pueden ser descargados manualmente de la propia página web de PLACSP. Estos datos aparecen comprimidos en formato zip y en diferentes agrupaciones. Cada archivo zip contiene un conjunto de archivos Atom/XML que van almacenando entradas en orden cronológico, las cuales contienen actualizaciones con datos de las licitaciones.

Entre sus funcionalidades principales se encuentran las siguientes:

Transformación de Datos

OpenPLACSP convierte los datos abiertos disponibles en formato Atom/XML a hojas de cálculo en formato XLSX. Este proceso permite que los datos puedan ser fácilmente manipulados por diversos programas de procesamiento de hojas de cálculo, lo que facilita su análisis y reutilización.

Modos de Generación de Datos

- **Generación de datos en dos tablas**: Este modo recibe ese nombre porque la información se agrupa en dos pestañas separadas. La primera pestaña, "Licitaciones" alude a la convocatoria y toda la información referida a cada una de las licitaciones. La segunda, "Resultados", muestra la información relativa a las adjudicaciones y resoluciones.
- **Generación de datos en una tabla**: Alternativamente, OpenPLACSP puede generar todos los datos en una única tabla, lo que puede ser útil para ciertos tipos de análisis que requieren la información en un formato consolidado.

Visualización de datos

La herramienta permite la visualización de diferentes datos de las licitaciones y adjudicaciones. Es una herramienta bastante completa con más de 40 variables que se muestran en la hoja de cálculo obtenida. Sin embargo, es únicamente una herramienta de visualización de datos y no de generación de estadísticas, por lo que no se incluye información alguna sobre las valoraciones de cada empresa a los diferentes criterios ni se ofrecen estadísticas sobre estas.

Facilitación del Acceso a la Información

La herramienta está diseñada para simplificar el acceso y el procesamiento de los datos abiertos de PLACSP. Esto es particularmente importante porque los formatos de datos abiertos (Atom/XML) pueden requerir conocimientos técnicos avanzados para su manejo. OpenPLACSP elimina estas barreras técnicas, permitiendo que cualquier usuario, independientemente de su nivel técnico, pueda trabajar con los datos de contratación pública.

Reutilización de la Información del Sector Público (RISP)

Al convertir los datos abiertos en hojas de cálculo, OpenPLACSP promueve la reutilización de la información del sector público. Los datos estandarizados y fácilmente manipulables permiten a los ciudadanos, investigadores y empresas utilizar esta información para diversos fines, desde análisis y estudios hasta la mejora de la transparencia y la toma de decisiones informadas.

2.3.3. Tendios

Tendios [14] es una plataforma diseñada para facilitar la participación en licitaciones públicas. Contiene una plataforma digital que ayuda a encontrar, analizar y gestionar licitaciones públicas. Es una herramienta todo en uno que facilita la centralización del trabajo y la gestión en equipo.

Funcionalidades clave:

1. **Búsqueda Avanzada**: Tendios permite buscar entre una vasta cantidad de licitaciones y adjudicaciones utilizando múltiples criterios de búsqueda, como palabras clave, sectores, localizaciones, importes o fechas. Además, ofrece alertas en tiempo real para mantener a los usuarios informados sobre las licitaciones de su interés.

2. **Gestión de Equipo**: La plataforma facilita la asignación de responsabilidades dentro del equipo, permitiendo a los usuarios gestionar estados, favoritos y analizar la competencia mediante analíticas avanzadas.

3. **Accesibilidad y Multi-dispositivo**: Tendios es accesible desde cualquier navegador web y dispositivo en cualquier momento, lo que garantiza que los usuarios puedan acceder a la plataforma en cualquier momento y lugar.

4. **Integraciones y APIs**: La herramienta se integra con una amplia gama de aplicaciones populares como Asana, Salesforce, Microsoft Teams, y más. Además, ofrece APIs que permiten leer y exportar información en diversos formatos (HTML, XML, JSON y CSV).

2.3.4. Tender

Tender [15] es una aplicación web desarrollada por la empresa Artabro Tech. Es un software de inteligencia artificial que ofrece soluciones para optimizar la participación en concursos públicos, permitiendo a las empresas encontrar y competir por licitaciones de manera más eficiente. Tender ayuda a maximizar las posibilidades de éxito en la obtención de contratos públicos.

Es una aplicación con suscripción mensual en la que hay diferentes planes donde elegir, pero ofrece la posibilidad de probarse gratuitamente durante siete días para evaluar su efectividad.

Funcionalidades clave:

1. **Recomendaciones personalizadas**: Tender tiene un motor de inteligencia artificial que recomienda a cada empresa las licitaciones con mayor probabilidad de ganar según su perfil y características, permitiendo a las empresas centrarse en las oportunidades más relevantes para su perfil y evitar perder tiempo con licitaciones que no se adaptan a ellos.

2. **Análisis de competidores**: Proporciona información detallada sobre las estrategias y resultados de los competidores. Esto permite comparar fácilmente la actividad de la empresa con la de sus competidores, ayudando a tomar decisiones más informadas sobre dónde y cómo competir.

3. **Predicciones de precios de adjudicación**: Utilizando inteligencia artificial, Tender predice el precio probable de adjudicación de las licitaciones, facilitando una mejor planificación financiera y estratégica.

4. **Enfoque del trabajo de los equipos**: Permite seleccionar concursos relevantes y proporciona herramientas para enfocar el trabajo de los equipos en la elaboración de ofertas con mayores posibilidades de éxito.

5. **Panel de control interactivo**: Tender ofrece un panel de control interactivo que brinda acceso a toda la información relevante sobre las licitaciones públicas en plazo y

adjudicadas. Esto incluye informes detallados por región, sector de actividad, organismos, entre otros. Con esta información, las empresas pueden tomar decisiones más fundamentadas y optimizar sus recursos para aumentar sus posibilidades de éxito en la contratación pública.

2.4 Crítica al estado del arte

Hemos podido observar que existen diversas herramientas que difieren en términos de su facilidad de uso, flexibilidad y capacidad para adaptarse a las necesidades específicas del usuario. Algunas son más intuitivas y fáciles de usar, mientras que otras requieren de un mayor nivel de conocimiento técnico para su configuración y uso eficaz.

Además, unas son más completas que otras, y algunas incluso utilizan inteligencia artificial para hacer predicciones sobre el precio de las licitaciones o para hacer recomendaciones personalizadas a los usuarios.

Sin embargo, el objetivo del presente trabajo es el desarrollo de una herramienta que pueda extraer los datos de forma automática de todos los documentos de una licitación y no solo aquella información presente en formato web en la propia página de cada expediente. Con esta información, se podría llevar a cabo un entrenamiento de modelos de inteligencia artificial que hagan predicciones parecidas a otras herramientas, pero ese no será el punto de comparación de esta sección.

Vamos a valorar la cantidad y dificultad de datos que presenta cada herramienta mencionada anteriormente, así como de la que se pretende desarrollar con el presente trabajo, SmarTenderAI. Además, se hará una comparación del coste monetario y facilidad de utilización de cada herramienta. Se ha recopilado esta información en la Tabla 2.1

Acceso a la plataforma

Gobierto, Tendios, Tender y la herramienta que se pretende desarrollar ofrecen acceso a través de una página web, lo cual facilita el acceso inmediato y la usabilidad para cualquier usuario con conexión a internet. En contraste, OpenPLACSP requiere la descarga de una aplicación, lo que puede presentar una barrera adicional para algunos usuarios. Además, la aplicación requiere la instalación de Java y puede dar problema en ciertos sistemas operativos.

Forma de visualización de los datos

La mayoría de las herramientas (Gobierto, Tendios, Tender y la herramienta en desarrollo) permiten la visualización de datos directamente en la web, lo que mejora la accesibilidad y la experiencia del usuario. OpenPLACSP, sin embargo, presenta los datos en hojas de cálculo que se deben generar en base a los archivos .ATOM descargados, lo cual, aunque útil para análisis detallados, puede ser menos intuitivo para usuarios no familiarizados con esta forma de presentación.

Variables que se extraen

En cuanto a la cantidad de variables, se puede observar que nuestra herramienta sobresale con aproximadamente 60 variables en comparación a las demás. Además, todas estas variables se extraen de forma automática de diferentes fuentes (tanto archivos PDFs como de HTML). OpenPLACSP también ofrece una gran cantidad de variables (más de

	Gobierto	OpenPLACSP	Tendios	Tender	SmarTenderAI
Acceso a la plataforma	Página Web	Descargar Aplicación	Página Web	Página Web	Página Web
Forma de visualización de los datos	En la web o descargando informes	Hoja de cálculo	En la web	En la web	En la web
Variables que se extraen	Unas 15 variables	Más de 40 variables, lista completa en [16]	Menos de 20 variables	Unas 15 variables	Unas 60 variables
Información presente en PDFs	No se extrae, pero se incluyen los links	Sí, pero se extrae manualmente	No se extrae, pero se incluyen los links	No se extrae, pero se incluyen los links	Se extrae de forma automática
Información de criterios	No se incluye	Sí se incluye	No se incluye	Sí se incluye	Sí se incluye
Valoraciones de empresas	No	No	No	No	Sí
Precio	Pago mensual con diferentes opciones de planes	Gratis	Tiene un plan gratis limitado y planes de pago	Pago mensual con diferentes opciones de planes	Gratis
Usuarios	Empresas	Cualquier persona	Cualquier persona	Cualquier persona	Cualquier persona

Tabla 2.1: Tabla comparativa de herramientas

40) pero todas ellas se extraen a mano, mientras que Gobierto, Tendios y Tender se limitan a extraer entre 15 y 20 variables, lo cual puede restringir la profundidad del análisis posible.

Información presente en PDFs

Una de las ventajas principales de la herramienta en desarrollo es su capacidad para extraer información automáticamente de PDFs, lo cual mejora la eficiencia y precisión del manejo de datos. OpenPLACSP también extrae información de PDFs, pero de manera manual, lo que puede ser un proceso más laborioso y propenso a errores. Gobierto, Tendios y Tender no extraen información de PDFs, aunque sí incluyen los enlaces a estos documentos para que se puedan revisar de forma manual.

Información de criterios

La inclusión de información de criterios es fundamental para un análisis completo y preciso. Esta información incluye el nombre de los criterios y su valoración máxima entre otros detalles. En este aspecto, Tender y la herramienta en desarrollo incluyen esta información, proporcionando una visión más detallada y contextualizada de los datos. Gobierto y Tendios no incluyen esta información, lo cual puede limitar la comprensión del contexto de los datos.

Valoraciones de empresas

Un aspecto único de la herramienta en desarrollo es la inclusión de valoraciones de empresas, lo que proporciona una capa adicional de información útil para los usuarios. Esta información incluye la lista de empresas participantes en cada licitación y sus respectivas puntuaciones a cada criterio, información que podría ser útil para desarrollar herramientas futuras de predicción o recomendación. Ninguna de las otras herramientas comparadas incluye valoraciones de empresas, lo cual puede ser una limitación para ciertos tipos de análisis.

Precio

En términos de coste, OpenPLACSP y la herramienta en desarrollo son gratuitas, lo cual las hace accesibles para una amplia gama de usuarios sin restricciones económicas. Gobierto y Tender, por otro lado, requieren un pago mensual con diferentes opciones de planes, lo cual puede ser un impedimento para algunos usuarios. Tendios ofrece un plan gratuito limitado, además de planes de pago, lo que proporciona cierta flexibilidad a la hora de satisfacer las necesidades de varios tipos de usuarios.

Usuarios

Todas las herramientas, excepto Gobierto que está orientada principalmente a empresas, están diseñadas para ser utilizadas por cualquier persona. Esto incluye a investigadores, estudiantes, analistas de datos y cualquier usuario interesado en acceder a datos abiertos de manera fácil y eficiente.

En el caso de Gobierto es necesario registrarse con el dominio de una empresa para tener acceso a la aplicación, por lo que no cualquier persona puede acceder a ella.

2.5 Propuesta

La herramienta desarrollada en este proyecto destaca por su capacidad para extraer una mayor cantidad de variables, su automatización en la extracción de información de PDFs, la inclusión de valoraciones de empresas y su gratuidad. Estas características la convierten en una opción robusta y accesible para el análisis y manejo de datos abiertos, ofreciendo ventajas significativas en comparación con las herramientas existentes.

La capacidad de extraer una mayor cantidad de variables es un aspecto crucial que permite a los usuarios acceder a una riqueza de datos sin precedentes. Esto no solo amplía el alcance de los análisis posibles, sino que también proporciona una base de datos más detallada, lo cual es esencial para la toma de decisiones informadas. Al tener acceso a unas 60 variables, los usuarios pueden explorar relaciones y patrones que serían difíciles de identificar con un conjunto de datos más limitado, o incluso en la propia plataforma de PLACSP por la dispersión de los datos. Esta funcionalidad es especialmente valiosa para investigadores y analistas que buscan comprender fenómenos complejos y multidimensionales, aunque también para empresas que quieren informarse sobre licitaciones pasadas para evaluar si les conviene participar en alguna licitación actual.

La automatización en la extracción de información de PDFs representa un avance significativo en la eficiencia y precisión del manejo de datos. En muchas aplicaciones, los datos importantes se encuentran dispersos en documentos PDF, lo que hace que su extracción manual sea un proceso laborioso y propenso a errores. La herramienta propuesta

elimina esta barrera al automatizar la extracción, lo que no solo ahorra tiempo, sino que también reduce la posibilidad de errores humanos.

La inclusión de valoraciones de empresas es una característica única que añade una capa adicional de información valiosa. Las valoraciones de empresas pueden proporcionar detalles importantes sobre la reputación y desempeño de las mismas, lo que puede ser crucial para decisiones comerciales y análisis de mercado. Al incorporar esta información, la herramienta no solo proporciona datos estáticos, sino que también ofrece contexto y evaluación cualitativa, lo que enriquece el análisis y la interpretación de los datos.

Por otro lado, la inclusión de información de criterios también es un aspecto importante que diferencia a esta herramienta de otras. Al proporcionar contexto y detalles sobre los criterios de evaluación de las empresas, los usuarios pueden tener una comprensión más profunda y precisa de los resultados de adjudicación de las licitaciones .

La gratuidad de la herramienta es otro factor distintivo que la hace accesible a una amplia gama de usuarios. A diferencia de otras soluciones que requieren suscripciones mensuales, esta herramienta está diseñada para ser utilizada sin coste alguno. La accesibilidad económica es fundamental para fomentar una cultura de datos abierta y colaborativa, donde el conocimiento y las herramientas están disponibles para todos, independientemente de sus recursos financieros.

Además de estas ventajas principales, la herramienta propuesta también se ha diseñado con la usabilidad en mente. La interfaz intuitiva y fácil de usar garantiza que incluso aquellos con habilidades técnicas limitadas puedan aprovechar al máximo sus capacidades. La visualización de datos directamente en la web facilita la interpretación y el análisis de la información, permitiendo a los usuarios analizar la información rápidamente y sin necesidad de herramientas adicionales.

En conclusión, la herramienta que se propone desarrollar se perfila como una solución integral y avanzada para el manejo y análisis de datos abiertos. Su capacidad para manejar una amplia variedad de variables, automatizar procesos complejos, proporcionar valoraciones cualitativas y ser accesible de forma gratuita la posicionan como un proyecto útil y con previsión a futuro.

CAPÍTULO 3
Análisis del problema

Una vez finalizado el estudio sobre el estado del arte, es crucial realizar un análisis detallado del problema para identificar oportunidades de innovación o de negocio. El propósito de esta sección es proporcionar una comprensión profunda del problema a abordar, estableciendo un marco claro que guíe el desarrollo del proyecto. Se analizarán los requisitos necesarios y se modelará conceptualmente la solución propuesta. Esto incluye la especificación de requisitos detallados y concretos, los cuales deben ser satisfechos por la solución propuesta y determinarán la implementación final.

En este proyecto es especialmente importante definir cómo se extraerá y manejará la información de los documentos PDF utilizando un modelo de lenguaje (LLM) y de los documentos HTML, analizando su estructura previamente, y cómo se almacenará y presentará dicha información de manera eficiente y accesible. Este análisis sentará las bases para una implementación exitosa y permitirá identificar áreas donde se puedan introducir mejoras e innovaciones significativas.

3.1 Especificación de Requisitos

La especificación de requisitos es un paso fundamental para asegurar que la solución propuesta cumpla con las expectativas de los usuarios y se ajuste a las necesidades del proyecto. A continuación, se detallan los requisitos funcionales y no funcionales de la herramienta propuesta.

3.1.1. Requisitos Funcionales

Los requisitos funcionales describen las funcionalidades y servicios específicos que el sistema debe ofrecer. Estos requisitos definen las acciones que el sistema debe ser capaz de realizar y cómo interactúa con los usuarios, datos y otros sistemas.

1. **Extracción de Información**
    - **RF1.1**: El sistema debe ser capaz de extraer automáticamente información relevante del HTML de los diferentes apartados de cada licitación (anuncio, pliego, adjudicación y formalización).
    - **RF1.2**: El sistema debe ser capaz de extraer la información relevante de los diferentes documentos PDF de cada licitación mediante el uso del LLM y otras técnicas de separación de texto.
2. **Visualización de Información**

- **RF2.1**: La aplicación debe permitir a los usuarios visualizar la información extraída de manera clara y organizada meiante tablas y gráficos interactivos.

3. **Búsqueda y Filtrado**
   - **RF3.1**: Los usuarios deben poder buscar y filtrar licitaciones según diferentes criterios (por ejemplo, fecha, entidad contratante, importe del contrato, lugar de ejecución e incluso palabras clave).
4. **Actualización y Mantenimiento de Datos**
   - **RF4.1**: El sistema debe ser capaz de actualizar la información regularmente y manejar nuevas licitaciones.
5. **Almacenamiento de los Datos**
   - **RF5.1**: El sistema debe incluir una base de datos para almacenar de manera estructurada la información extraída.
   - **RF5.2**: La base de datos debe soportar consultas eficientes y estar optimizada para el manejo de grandes volúmenes de datos.
6. **Valoraciones de Empresas**
   - **RF6.1**: El sistema debe incluir valoraciones de empresas en la información extraída de forma automática.
   - **RF6.2**: Presentar las valoraciones de manera que sean fácilmente accesibles y comprensibles para los usuarios.
   - **RF6.3**: Incluir estadísticas relevantes sobre las valoraciones tales como media, mediana, desviación típica y diferencia entre primera y segunda posición.
7. **Información de Criterios**
   - **RF7.1**: El sistema debe proporcionar detalles sobre los criterios de evaluación de las empresas para cada licitación.
   - **RF7.2**: Incluir esta información en la visualización de datos para proporcionar contexto adicional.
8. **Exportación de los datos**
   - **RF8.1**: El sistema debe proporcionar un mecanismo que permita exportar los datos a un formato compatible con hojas de cálculo.
   - **RF8.2**: La información debe estar organizada de manera que sea posible trabajar con los datos con facilidad.

3.1.2. Requisitos no funcionales

Los requisitos no funcionales describen los criterios de calidad y las restricciones que afectan el funcionamiento del sistema. Estos requisitos especifican cómo debe ser el comportamiento del sistema en términos de rendimiento, seguridad, usabilidad y otros factores que no están relacionados directamente con las funciones específicas del sistema.

1. **Rendimiento**
   - **RNF1.1**: El sistema debe procesar y presentar la información en un tiempo razonable, optimizando la carga de la aplicación.

- **RNF1.2**: El sistema debe optimizar el tiempo de extracción de los datos y su carga en la base de datos.

2. **Escalabilidad**
   - **RNF2.1**: La arquitectura del sistema debe permitir la escalabilidad para manejar un aumento en el número de datos de la base de datos, ya que se debe mantener actualizada.
3. **Usabilidad**
   - **RNF3.1**: La interfaz de usuario debe ser intuitiva y fácil de usar, facilitando la navegación y el acceso a la información.
   - **RNF3.2**: La visualización de datos debe ser clara y accesible, permitiendo a los usuarios interpretar la información sin necesidad de herramientas adicionales.
4. **Accesibilidad**
   - **RNF4.1**: La herramienta debe ser accesible de forma gratuita para los usuarios.
   - **RNF4.2**: La interfaz debe ser compatible con diferentes dispositivos y navegadores, garantizando el acceso a una amplia gama de usuarios.

3.2 Análisis energético

En este trabajo se empleará un Modelo de Lenguaje de Gran Tamaño (LLM) para el procesamiento y extracción de información de documentos PDF. Los LLM, a pesar de ser herramientas revolucionarias y con utilidad en diversas áreas, también son herramientas costosas en términos computacionales y de consumo energético.

Como referencia, el entrenamiento de GPT-3 estima haber consumido aproximadamente 1287 MWh de energía [17]. Para poner esto en perspectiva, un hogar español consume alrededor de 3,487 MWh de energía en un año, lo que significa que el entrenamiento de GPT-3 equivale al consumo energético de aproximadamente 370 hogares españoles en un año.

Estos datos son asombrosos y subrayan la consideración crítica que debe tenerse al abordar el entrenamiento de un LLM desde cero. En nuestro caso, utilizaremos un LLM preentrenado y nos centraremos en su aplicación y optimización para nuestros propósitos específicos.

Al hacerlo, minimizamos el impacto ambiental asociado con el entrenamiento inicial y nos enfocamos en maximizar el beneficio del modelo ya disponible para nuestras tareas de procesamiento y extracción de información de documentos PDF.

Esta estrategia no solo es más eficiente desde el punto de vista energético, sino que también nos permite dedicar nuestros recursos a optimizar el uso y la aplicación práctica del LLM en nuestro proyecto.

3.3 Análisis del marco legal y ético

Como ya se ha mencionado, en este trabajo trataremos con una colección de datos abiertos que proporciona el propio Estado Español a través de su Plataforma de Contratación del Sector Público. Para ello, es necesario analizar los diversos aspectos legales y éticos relacionados con el desarrollo de una aplicación de código abierto para el manejo y análisis de datos abiertos proporcionados por el Estado Español.

3.3.1. Datos Abiertos

Análisis de la Protección de Datos

A pesar de que los datos utilizados son datos abiertos proporcionados por el Estado, es importante garantizar que la manipulación y presentación de estos datos cumplan con las normativas de protección de datos vigentes, como la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales [18] (LOPDGDD), y el Reglamento General de Protección de Datos [19] (RGPD) de la Unión Europea.

Reutilización de la Información del Sector Público

Se debe hacer un análisis exhaustivo de la Ley 37/2007, de 16 de noviembre, sobre reutilización de la información del sector público [20]. Esta ley tiene como objetivo regular el régimen jurídico aplicable a la reutilización de documentos elaborados por las Administraciones y organismos del sector público. A continuación se resaltan los aspectos clave a tener en cuenta en el desarrollo del proyecto:

- **Objeto y Ámbito de Aplicación (Título I):**
  - La ley tiene como objetivo regular el régimen jurídico aplicable a la reutilización de los documentos elaborados por el sector público. En el contexto del proyecto, esto implica que todos los datos utilizados deben cumplir con los requisitos establecidos por la ley para su reutilización.
  - La ley se aplica a todas las Administraciones y organismos del sector público. Esto significa que los datos extraídos y utilizados en el proyecto deben provenir de fuentes públicas autorizadas y cumplir con los criterios establecidos por la ley.
- **Régimen Jurídico de la Reutilización (Título II):**
  - La ley establece que las Administraciones pueden permitir la reutilización de sus documentos sin condiciones o bajo licencias claras, justas y no discriminatorias. En el caso de PLACSP, se permite el uso de los datos abiertos sin condiciones.
  - Las condiciones incluyen el uso correcto de los documentos y la indicación de la fuente. La aplicación debe cumplir con estas condiciones, asegurando que siempre se indique la fuente de los datos proporcionando enlaces a cada licitación y sus documentos.

3.3.2. Propiedad Intelectual

La propiedad intelectual es otro aspecto crucial a considerar en este proyecto. Dado que se está desarrollando una aplicación de código abierto, es esencial determinar el tipo de licencia que se utilizará. En este caso, se puede optar por una licencia permisiva como la MIT o la GPL, que permiten la libre distribución y modificación del software, siempre y cuando se mantengan los derechos de autor y se incluyan las licencias originales en las copias o derivaciones del software.

Además, se debe prestar atención a otros aspectos convencionales de la propiedad intelectual, como los derechos de las imágenes, gráficos o cualquier otro contenido multimedia que se utilice en el producto final. Asegurarse de que todos los materiales utilizados están correctamente licenciados es fundamental.

3.3.3. Ética

Además de las consideraciones legales, es crucial abordar los posibles dilemas morales que pueden surgir durante el desarrollo de la aplicación. El uso de datos abiertos implica una responsabilidad ética en cuanto a la forma en que se presentan y utilizan los datos. Se debe garantizar que la información se maneja de manera justa y transparente, evitando cualquier uso indebido que pueda afectar negativamente a los usuarios o al público en general. Así, evitando también la desinformación a los usuarios mediante la divulgación de datos erróneos o alterados.

3.4 Análisis de riesgos

En esta sección se abordarán los riesgos potenciales que podrían surgir durante el desarrollo del proyecto SmarTenderAI. Es crucial identificar estos riesgos para comprender cómo podrían afectar el producto final y qué medidas pueden implementarse para mitigar su impacto.

Riesgos Identificados

1. **Problemas de Acceso a los Datos**: Puede surgir dificultad para acceder de manera consistente y fiable a los datos proporcionados por la plataforma de contratación del sector público. Esto podría retrasar el desarrollo de la herramienta y afectar su funcionalidad a la hora de mantenerla actualizada.

2. **Complejidad en la Extracción de Variables**: La automatización en la extracción de información de PDFs puede enfrentar desafíos técnicos debido a la variabilidad en la estructura y formato de los documentos, lo que podría afectar la precisión y fiabilidad de la herramienta.

3. **Dificultad para Seleccionar los Documentos Relevantes**: a la hora de filtrar los documentos PDFs y considerar aquellos con información relevante, se podrían introducir errores que terminasen dando prioridad a documentos poco o nada relevantes para la extracción de los datos, lo que podría llevar a información errónea o inexistente.

4. **Interfaz poco Intuitiva**: el desarrollo de la interfaz podría ocasionar problemas para los usuarios si no se desarrolla correctamente debido a la gran cantidad de datos a mostrar.

Impacto en el Producto Final

1. **Problemas de Acceso a los Datos**: Si hay dificultades para acceder consistentemente a los datos de la plataforma de contratación del sector público, podría resultar en retrasos en el desarrollo de la herramienta. Esto afectaría la capacidad de mantener actualizada la información, reduciendo la utilidad y precisión de la herramienta para los usuarios.

2. **Complejidad en la Extracción de Variables**: La variabilidad en la estructura y formato de los documentos PDF puede comprometer la precisión y fiabilidad de la herramienta en la extracción automatizada de datos. Esto podría llevar a información incompleta o incorrecta para el análisis posterior.

3. **Dificultad para Seleccionar los Documentos Relevantes**: La selección errónea de documentos puede resultar en la inclusión de datos irrelevantes o la omisión de información crucial, afectando la calidad y relevancia de los resultados finales.

4. **Interfaz poco Intuitiva**: Una interfaz mal diseñada podría dificultar la navegación y comprensión de los datos para los usuarios, reduciendo la eficiencia y usabilidad de la herramienta.

Medidas Preventivas y Correctivas

Para mitigar los riesgos antes mencionados se deben implementar una serie medidas:

1. **Almacenamiento de Datos**: Establecer una base de datos en la que se almacenarán los datos a medida que se extraen, para asegurar que la aplicación esté siempre en funcionamiento y mostrando los datos disponibles aunque se produzcan fallos a la hora de extraer datos posteriores.

2. **Optimización de Algoritmos de Extracción**: Desarrollar algoritmos robustos de extracción que puedan manejar la variabilidad en la estructura de documentos así como optimizar el uso del LLM para que procese únicamente información relevante a la variable que queremos extraer y evitar así respuestas erróneas.

3. **Definición de Criterios para la Selección de Documentos**: Para evitar procesar documentos irrelevantes, se debe implementar una serie de criterios que debe seguir el sistema automático para evitar la selección errónea de estos. Criterios tales como el nombre del documento, el número de páginas e incluso palabras clave en su contenido pueden ayudar a evitar errores.

4. **Diseño Claro y Organizado**: Para evitar una mala experiencia por parte de los usuarios, será necesaria una buena organización de la interfaz, utilizando pestañas, filtros y tablas para facilitar la visualización y manipulación de los datos. Además, se pueden realizar pruebas con usuarios beta para corregir errores antes de la versión final de la aplicación.

3.5 Identificación y Análisis de Soluciones Posibles

En el proceso de desarrollo de esta herramienta, es fundamental considerar diversas soluciones posibles que puedan satisfacer los requisitos establecidos. Esto incluye tomar decisiones sobre la forma de extracción de los datos, la forma en la que se almacenan y el tipo de sistema que se desarrollará para su visualización. A continuación, se presentan varias alternativas, analizando sus pros y contras, y estableciendo un criterio de selección para determinar la solución más adecuada para este TFG.

3.5.1. Extracción de Información

La extracción de información de documentos PDF y HTML es un aspecto crucial en el desarrollo de la herramienta. Los PDFs, al tener formatos muy diversos y heterogéneos, hacen que su procesamiento sea un desafío, mientras que los HTML presentan estructuras más homogéneas y fáciles de generalizar. Para abordar esta tarea, se han considerado dos técnicas diferentes que se describen a continuación, así como la justificación de la solución elegida.

1. **Procesamiento de Texto Tradicional**

   Esta técnica se basa en el uso de expresiones regulares y palabras claves para separar el texto en diferentes apartados que facilita su extracción posteriormente o encontrar datos concretos.

   Pros:

   - **Eficiencia:** Las técnicas tradicionales de procesamiento de texto, como la separación de apartados y la extracción de información con expresiones regulares, son rápidas y eficientes para datos bien estructurados.
   - **Control:** Proporcionan un alto grado de control sobre el proceso de extracción, permitiendo ajustes específicos para diferentes tipos de documentos.
   - **Simplicidad:** Relativamente simples de implementar y mantener, especialmente para documentos con estructuras predecibles.

   Contras:

   - **Limitaciones en Complejidad:** Pueden tener dificultades para manejar información compleja y no estructurada que requiere comprensión contextual.
   - **Escalabilidad:** Pueden no escalar bien con documentos de formato variado y contenido inconsistente.

2. **Modelos de Lenguaje de Gran Escala (LLM)**

   Esta técnica se basa en el uso de un LLM al que se proporciona un contexto y una pregunta a la que contestará con el contexto proporcionado. Este contexto puede ser un extracto de texto o incluso una tabla extraída del documento.

   Pros:

   - **Comprensión Contextual:** Los LLM tienen una capacidad avanzada para comprender y extraer información compleja, capturando contexto y relaciones entre datos.
   - **Flexibilidad:** Pueden adaptarse a una amplia variedad de documentos y formatos sin necesidad de reglas específicas de extracción.
   - **Precisión:** Mejoran la precisión en la extracción de información compleja que requiere comprensión semántica.

   Contras:

   - **Tiempo de Procesamiento:** Pueden ser más lentos en comparación con técnicas tradicionales, especialmente cuando el contexto proporcionado es de gran tamaño o la tarea demasiado compleja.
   - **Dependencia de Datos:** La calidad de la extracción depende de la calidad de los datos que se proporcionan al modelo. Si el texto que se proporciona es de mala calidad el LLM puede no extraer las variables con la mayor precisión y caer en errores.

Solución Elegida: Combinación de Procesamiento de Texto Tradicional y LLM

Para maximizar la eficiencia y la precisión en la extracción de información de PDFs y HTML, se ha optado por una solución híbrida que combina técnicas de procesamiento de

texto tradicional y el uso de un LLM. Esta estrategia aprovecha las fortalezas de ambas técnicas y mitiga sus debilidades. De esta manera, se utiliza el procesamiento tradicional para extraer aquellas variables que siempre se presentan de la misma forma y en los mismos apartados dentro del PDF y para la información del HTML que siempre presentan los mismom formatos y etiquetas en todas las licitaciones, mientras que se usa el LLM para extraer aquella información más compleja que necesita comprensión contextual del texto, tales como los diferentes criterios de adjudicación y las propias valoraciones de las empresas a esos criterios.

Ventajas de la Solución Híbrida

- **Eficiencia y Precisión: (RNF1.2)** La combinación de ambas técnicas permite una extracción rápida para aquellas variables estructuradas que lo permitan, y una extracción precisa para aquellas variables que necesiten el uso del LLM. Así, se alcanza un balance entre optimización del tiempo de extracción y calidad de los datos.
- **Flexibilidad:** La solución es adaptable para extraer la información tanto de texto plano como de tablas presentes en el documento.
- **Control y Comprensión:** Mantiene el control específico sobre la extracción de datos bien estructurados mientras se beneficia de la comprensión contextual avanzada del LLM para información más compleja.

3.5.2. Almacenamiento de Datos

El almacenamiento de datos es un componente crítico en el desarrollo de la herramienta. Una gestión eficiente y segura de los datos garantiza no solo la integridad y disponibilidad de la información, sino también un rendimiento óptimo de la aplicación. A continuación, se presentan las diferentes alternativas evaluadas y la justificación de la elección.

1. **Bases de Datos SQL**

   Pros:

   - **Estructura y Normalización:** Las bases de datos SQL son ideales para datos estructurados y permiten la normalización, lo que reduce la redundancia y asegura la integridad de los datos.
   - **Consistencia y Confiabilidad:** Garantizan transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad), lo cual es crucial para la precisión y confiabilidad de los datos.
   - **Consultas Complejas:** SQL (Structured Query Language) facilita la realización de consultas complejas y la manipulación de grandes volúmenes de datos de manera eficiente.
   - **Soporte y Comunidad:** Amplio soporte y una comunidad de usuarios extensa, lo cual facilita la resolución de problemas y el acceso a recursos y documentación.

   Contras:

   - **Rigidez en el Esquema:** La necesidad de definir un esquema rígido puede ser una limitación si los requisitos de datos cambian con frecuencia.

2. **Bases de Datos NoSQL**

   Pros:

   - **Flexibilidad:** Permiten almacenar datos no estructurados y semiestructurados, ofreciendo flexibilidad en la gestión de diferentes tipos de datos.
   - **Escalabilidad Horizontal:** Diseñadas para escalar horizontalmente, lo que facilita el manejo de grandes volúmenes de datos distribuidos.
   - **Rendimiento:** Pueden ofrecer un rendimiento superior para ciertas operaciones de lectura/escritura de datos no estructurados.

   Contras:

   - **Consistencia Eventual:** Muchas bases de datos NoSQL operan bajo el modelo de consistencia eventual, lo que puede ser un problema para aplicaciones que requieren transacciones ACID.
   - **Consultas Complejas:** No son tan eficientes como SQL para realizar consultas complejas, lo que puede limitar su uso en aplicaciones que requieren análisis de datos detallados.

Solución Elegida: Base de Datos SQL

Basado en los criterios de consistencia, integridad, capacidad para realizar consultas complejas y la necesidad de manejar datos estructurados de manera eficiente, se ha decidido utilizar una **Base de Datos SQL** como la solución de almacenamiento de datos para esta herramienta. Esta elección se justifica por las siguientes razones:

- **Datos Estructurados**: La herramienta manejará datos que requieren una estructura rígida y bien definida, como los criterios y las valoraciones de empresas. Las bases de datos SQL son ideales para organizar y gestionar este tipo de datos estructurados. La normalización en SQL asegura que cada aspecto del dato se alinee correctamente dentro del esquema, facilitando la integridad y la eficiencia en el almacenamiento y recuperación de datos.

- **Consistencia y Confiabilidad:** La capacidad de las bases de datos SQL para manejar transacciones ACID asegura que todas las operaciones se realicen de manera precisa y confiable, lo cual es fundamental para la integridad de los datos.

- **Eficiencia en Consultas:** SQL permite la realización de consultas complejas que son esenciales para el análisis detallado de los datos.

- **Soporte y Recursos:** La amplia comunidad y soporte para las bases de datos SQL facilitan el desarrollo, mantenimiento y resolución de problemas, asegurando un ecosistema robusto y confiable.

- **Escalabilidad: (RNF2.1)** Las bases de datos SQL modernas ofrecen capacidades de escalabilidad que permiten el manejo eficiente de grandes volúmenes de datos. Utilizando técnicas de particionamiento, replicación y balanceo de carga, estas bases de datos pueden escalar horizontal y verticalmente para manejar aumentos en la cantidad de datos sin comprometer el rendimiento.

3.5.3. Aplicación para la visualización de los datos

Debemos considerar también las diversas soluciones que hay ante el desarrollo de la aplicación de visualización. A continuación, se presentan varias alternativas, analizando sus pros y contras, y estableciendo un criterio de selección para determinar la solución más adecuada para este TFG.

1. **Desarrollo de una aplicación de escritorio**

   Pros:

   - **Rendimiento:** Las aplicaciones de escritorio suelen tener un mejor rendimiento en comparación con las aplicaciones web, ya que no dependen de la velocidad de conexión a internet.
   - **Acceso Offline:** Permite el uso de la herramienta sin necesidad de estar conectado a internet, lo cual es beneficioso en entornos con conectividad limitada.
   - **Integración:** Facilita la integración con otros programas y sistemas locales, aprovechando la capacidad del hardware del usuario.

   Contras:

   - **Compatibilidad:** Es necesario desarrollar versiones para diferentes sistemas operativos (Windows, macOS, Linux), lo que puede incrementar el tiempo y los costes de desarrollo.
   - **Actualización:** Las actualizaciones pueden ser más complicadas de gestionar y distribuir, requiriendo que los usuarios descarguen e instalen nuevas versiones manualmente.

2. **Desarrollo de una aplicación móvil**

   Pros:

   - **Accesibilidad:** Permite a los usuarios acceder a la herramienta desde cualquier lugar y en cualquier momento, siempre que tengan su dispositivo móvil.
   - **Usabilidad:** Interfaz optimizada para dispositivos móviles, lo cual puede resultar más intuitivo para algunos usuarios.

   Contras:

   - **Limitaciones Técnicas:** Las aplicaciones móviles pueden tener limitaciones en cuanto a procesamiento y almacenamiento en comparación con aplicaciones de escritorio o web.
   - **Desarrollo Duplicado:** Requiere el desarrollo de versiones separadas para iOS y Android, lo que puede incrementar los recursos necesarios.

3. **Desarrollo de una aplicación web**

   Pros:

   - **Accesibilidad Universal:** Los usuarios pueden acceder a la herramienta desde cualquier dispositivo con conexión a internet, independientemente del sistema operativo.

- **Actualización Simplificada:** Las actualizaciones se implementan en el servidor, lo que asegura que todos los usuarios tengan acceso a la última versión sin necesidad de descargas adicionales.
- **Interfaz Intuitiva:** Una aplicación web puede diseñarse con una interfaz amigable y accesible, facilitando su uso incluso para aquellos con habilidades técnicas limitadas.
- **Escalabilidad:** Las aplicaciones web pueden escalar fácilmente para manejar grandes volúmenes de datos.

Contras:

- **Dependencia de Internet:** La aplicación requiere una conexión constante a internet, lo que puede ser una limitación en entornos con conectividad inestable.

4. **Desarrollo de una API Pública**

Pros:

- **Flexibilidad:** Permite a los desarrolladores externos crear sus propias aplicaciones y servicios utilizando los datos proporcionados por la herramienta.
- **Escalabilidad:** Facilita la integración con otros sistemas y plataformas, promoviendo la interoperabilidad.
- **Actualización:** Facilita la actualización y mantenimiento de la herramienta, ya que los cambios en la API no requieren que los usuarios actualicen sus aplicaciones.

Contras:

- **Dependencia:** Los usuarios dependen de sus propias habilidades de desarrollo o de terceros para crear aplicaciones que utilicen la API.

Solución Elegida: Desarrollo de una Aplicación Web

La aplicación a desarrollar debe seguir una serie de Requisitos No Funcionales que se listan en la Sección 3.1.2. Si nos basamos en esos criterios para elegir la solución óptima, podemos ver cómo el desarrollo de una Aplicación Web es el más adecuado:

- RNF1.1: Las aplicaciones web permiten el uso de técnicas avanzadas de optimización del rendimiento, como la carga asíncrona y el uso de caché. Estas técnicas reducen el tiempo de carga percibido y permiten una experiencia de usuario fluida y rápida.
- RNF3.1: Utilizando tecnologías web modernas se pueden desarrollar interfaces de usuario altamente intuitivas y fáciles de usar. Así, es posible desarrollar interfaces dinámicas y atractivas que mejoran significativamente la experiencia del usuario.
- RNF3.2: Las aplicaciones web pueden integrar bibliotecas y frameworks de visualización de datos que permiten presentar la información de manera clara y comprensible. Estas herramientas proporcionan gráficos interactivos y otras representaciones visuales que facilitan la interpretación de datos complejos.
- RNF4.1: Una página web puede ser de uso gratuito para cualquier usuario y además puede ser utilizada por personas con diferentes niveles de conocimiento informático, ya que no necesitará de instalación de ninguna aplicación ni de conocimiento de informática para el uso de una API.

- RNF4.2: Una interfaz web puede hacer uso de diferentes frameworks que permitan el desarrollo de una interfaz responsiva y compatible con diversos navegadores. Así, se podrá acceder desde cualquier dispositivo independientemente de su sistema operativo y dimensiones.

3.6 Solución Propuesta

Una vez analizadas las alternativas y razonadas las decisiones tomadas, procedemos a describir en detalle la solución elegida para este proyecto, incluyendo las fases de desarrollo, el proceso de implementación y las pruebas de validación. El desarrollo del proyecto se centrará en tres componentes principales: el módulo de análisis y extracción de información de licitaciones, la base de datos y la aplicación web. A continuación, se detallan las diferentes fases para cada uno de estos componentes.

3.6.1. Diseño

Módulo de Análisis y Extracción de Información de Licitaciones

Antes de proceder al diseño del módulo, es importante analizar las variables que se desean extraer y donde encontrarlas. Es decir, algunas de ellas será posible extraerlas directamente del HTML de los diferentes apartados de la licitación, lo que facilitará mucho el proceso, y otras deberemos extraerlas de los diferentes documentos PDF.

En ambos casos, es importante realizar un análisis exhaustivo del formato de origen. Para el HTML es necesario saber en qué secciones y con qué etiquetas se representan las variables que queremos extraer, mientras que para el PDF es necesario analizar la estructura de varios ejemplos para detectar aquella información que siempre se presenta de la misma manera y es fácil de extraer, y aquella que no tiene un formato homogéneo y se puede presentar de diversas formas o que necesita una comprensión semántica del texto para poder ser extraída.

Posteriormente, se pasa a diseñar el flujo de procesamiento que seguirá la información desde su origen (PDF y HTML) hasta su almacenamiento en la base de datos. Esto incluye la extracción inicial, el procesamiento y la integración de los datos.

Para ello, es importante definir las bibliotecas y herramientas que se van a utilizar, así como la implementación del LLM en la herramienta.

Base de Datos SQL

El diseño detallado de la base de datos es fundamental para garantizar que la información extraída de los documentos HTML y PDF se almacene de manera eficiente y segura. Primero, es necesario identificar todas las variables a almacenar, así como sus tipos de datos, y definir las entidades y sus relaciones.

A continuación, se llevará a cabo el diseño de un diagrama entidad-relación (ER) que se puede consultar en la Figura 5.3 para representar visualmente las tablas correspondientes a cada entidad, sus campos y las relaciones entre ellas. Este diagrama servirá como una guía para la implementación de la base de datos, facilitando la comprensión de la estructura y las interacciones entre los datos.

Finalmente, se determinarán las claves primarias y foráneas para asegurar la integridad referencial y mantener la consistencia de los datos. La correcta definición de estas

claves es esencial para garantizar que las relaciones entre las tablas se mantengan intactas y que la base de datos funcione de manera coherente.

Aplicación Web

El diseño de la aplicación web se centra en dos aspectos principales: la interfaz de usuario y la arquitectura técnica.

Primero, se determinarán las tecnologías y frameworks a utilizar para el frontend y el backend. Se evaluarán opciones para el frontend que faciliten la creación de una interfaz interactiva y responsiva, como React, Angular o Vue.js, así como el uso de otros framworks como Bootstrap. Para el backend, se considerarán tecnologías que ofrezcan buen rendimiento y fácil integración con la base de datos.

La arquitectura técnica se diseñará para optimizar el rendimiento y permitir una fácil escalabilidad. Esto incluirá la definición de la estructura del backend, la configuración del servidor y la integración con la base de datos, así como el desarrollo de las API necesarias para la comunicación entre el frontend y el backend.

3.6.2. Desarrollo

Módulo de Análisis y Extracción de Información de Licitaciones

El desarrollo de este módulo se llevará a cabo en varias fases, asegurando una implementación progresiva y validación continua.

1. Implementación del Procesamiento de HTML: Se desarrollarán scripts para analizar y extraer datos de documentos HTML utilizando bibliotecas como BeautifulSoup. Estos scripts identificarán secciones específicas del documento y extraerán la información relevante utilizando selectores CSS y etiquetas HTML.

2. Extracción de Información de PDF: Se utilizarán herramientas como PyMuPDF para transformar los documentos PDF a texto procesable. Se implementarán algoritmos para manejar diversas estructuras y formatos, capturando tanto información estructurada como texto libre que requiera análisis semántico.

3. Integración del Modelo de Lenguaje (LLM): Se integrará un LLM para extraer información compleja y contextual, como valoraciones de empresas y criterios de evaluación.

4. Procesamiento y Normalización de Datos: Se procesarán y normalizarán los datos extraídos para asegurar su consistencia y calidad. Se implementarán scripts para limpiar, transformar y validar la información antes de su almacenamiento en la base de datos.

Base de Datos SQL

Se llevará a cabo el desarrollo de la base de datos siguiendo el diseño establecido en el diagrama entidad-relación de la Figura 5.3.

Para ello, es necesario la instalación del software de gestión de bases de datos (DBMS), donde se configurará el entorno para garantizar un rendimiento óptimo y seguridad.

Basado en el diagrama entidad-relación, se crearán las tablas necesarias en la base de datos. Esto incluye la definición de campos, tipos de datos, restricciones y la implementación de claves primarias y foráneas para mantener la integridad referencial entre las tablas.

Aplicación Web

En la fase de desarrollo de la aplicación web, se llevarán a cabo los siguientes pasos:

1. Implementación del Frontend: Utilizando los frameworks seleccionados, se desarrollará la interfaz de usuario. Se integrarán los componentes de la interfaz, como las vistas de licitaciones y empresas, los formularios de búsqueda y filtros, y las páginas de estadísticas y detalles de licitaciones. Además, se aplicarán estilos y diseño responsivo utilizando herramientas como Bootstrap para asegurar una experiencia de usuario intuitiva y accesible en diferentes dispositivos.

2. Desarrollo del Backend: Este proceso incluye la configuración del servidor, la creación de las APIs necesarias para la comunicación entre el frontend y el backend, y la integración con la base de datos SQL. Se desarrollarán las funcionalidades requeridas para manejar la lógica de negocio y el acceso a datos.

3. Integración con la Base de Datos: Se conectará la aplicación web con la base de datos mediante el uso de las APIs y servicios desarrollados en el backend. Esto implicará la implementación de operaciones para la inserción, actualización, eliminación y consulta de datos almacenados en la base de datos. También se configurarán las consultas necesarias para la visualización de la información en la interfaz de usuario.

3.6.3. Validación

La validación en este contexto implicará varios pasos para asegurar que el sistema cumpla con los requisitos y funcione correctamente:

- **Pruebas Unitarias:** Se llevarán a cabo pruebas unitarias para el módulo de análisis y extracción de información. Las pruebas unitarias son cruciales para verificar que cada función y método opere según lo esperado en un entorno aislado. Se utilizarán marcos de pruebas unitarias como pytest que permitirá automatizar y repetir estas pruebas con facilidad.

- **Verificación de Precisión:** Se verificará que los datos extraídos de los documentos HTML y PDF sean precisos y completos. Esto implica comparar los datos extraídos con los datos originales para asegurar que no haya pérdida o distorsión de la información durante el proceso de extracción.

- **Pruebas de Integración:** Una vez realizadas las pruebas unitarias, se procederá a verificar la interacción entre los componentes. Las pruebas de integración aseguran que los módulos trabajen juntos de manera coherente. Se probará que el frontend pueda comunicarse eficazmente con el backend a través de las APIs. Estas pruebas ayudan a identificar problemas en los puntos de interacción entre componentes, que podrían no ser evidentes durante las pruebas unitarias.

3.7 Plan de Trabajo

Este trabajo se ha desarrollado de manera secuencial, siguiendo un enfoque estructurado que permitió completar cada fase del proyecto de manera ordenada y eficiente.

En la primera fase del proyecto, se centró en el desarrollo del módulo de extracción de información. Este módulo se encargó de analizar y procesar los expedientes, extrayendo datos relevantes que luego se almacenaron en formato JSON. Este paso inicial fue

fundamental, ya que sentó las bases para el manejo y la manipulación de los datos que se utilizarían en las etapas posteriores del proyecto.

Una vez completado el módulo de extracción de información, se procedió al diseño y desarrollo de la base de datos SQL. Esta fase implicó la creación de un esquema de base de datos robusto que pudiera almacenar eficientemente los datos extraídos. Posteriormente, se utilizó un script para introducir los datos de los archivos JSON en la base de datos. Este proceso aseguró que toda la información estuviera organizada y accesible para su uso en la aplicación web.

En la fase final, se diseñó y desarrolló la aplicación web. Esta aplicación se construyó para permitir la visualización de los datos almacenados en la base de datos, ofreciendo una interfaz de usuario amigable e intuitiva. El desarrollo de la aplicación web incluyó tanto el frontend como el backend, asegurando que los usuarios pudieran interactuar con los datos de manera eficiente y efectiva.

Durante todo el desarrollo se ha mantenido una comunicación constante y efectiva con las tutoras. Las tutoras han jugado un papel fundamental en la evaluación continua de los resultados semanales, proporcionando retroalimentación constructiva y orientación valiosa en cada etapa del proyecto. Estas reuniones han sido esenciales no solo para presentar los avances alcanzados, sino también para discutir desafíos y obstáculos que surgieron durante el desarrollo.

Además de las reuniones programadas, se mantuvo un canal abierto de comunicación a través de la plataforma Teams[1]. Este flujo continuo de comunicación permitió abordar cuestiones urgentes de manera inmediata y obtener aclaraciones rápidas sobre aspectos específicos del proyecto. Las tutoras estuvieron siempre disponibles para proporcionar apoyo y resolver dudas, lo que fue crucial para mantener el impulso y la eficiencia del trabajo.

Para la gestión del proyecto, se hizo uso de la herramienta Trello [2]. Se creó un tablero específico para el proyecto, con listas dedicadas a cada parte del desarrollo: el módulo de extracción, la base de datos, la aplicación web y la memoria. En cada lista se añadieron tareas detalladas con fechas límite, lo que facilitó un seguimiento claro y estructurado del progreso. Este enfoque permitió identificar y priorizar las tareas, asegurando que se cumplieran los plazos establecidos y que cada fase del proyecto se completara de manera ordenada.

[1] https://www.microsoft.com/es-co/microsoft-teams
[2] https://trello.com

CAPÍTULO 4
Tecnologías

4.1 Extracción de la Información

4.1.1. Python

Python [1] es el lenguaje de programación principal utilizado para desarrollar los scripts y módulos de procesamiento de la información. Es conocido por su sintaxis clara y legible, así como por su amplia gama de bibliotecas y frameworks. Estas características lo hacen ideal para tareas de extracción, manipulación de datos, automatización de procesos y desarrollo de aplicaciones complejas. Junto a este lenguaje, se ha hecho uso de las siguientes bibliotecas:

- **PyMuPDF** [2] es una biblioteca de Python para trabajar con documentos PDF, proporcionando herramientas para la extracción de texto, imágenes y metadatos. En esta aplicación, PyMuPDF se utiliza para extraer el contenido del Anexo I en formato PDF, permitiendo acceder y manipular la información contenida en estos documentos de manera precisa. Esto es esencial para obtener datos relevantes que luego son procesados y estructurados por otros módulos de la aplicación.
- **BeautifulSoup** [3] es una biblioteca de Python para analizar documentos HTML y XML. Se utiliza para extraer datos estructurados de las páginas HTML de las licitaciones. BeautifulSoup facilita la navegación y búsqueda de elementos específicos dentro del HTML, lo cual es crucial para recopilar la información necesaria de manera automatizada y eficiente.
- **LangChain** [4] es una biblioteca que facilita la integración y el uso de LLMs en aplicaciones complejas. En nuestra aplicación, LangChain se utiliza para orquestar los flujos de trabajo que involucran la extracción de información y su procesamiento mediante LLMs. Esto incluye tareas como la extracción de datos específicos de documentos, el análisis de textos y la estructuración de la información de manera coherente.
- **Camelot** [5] es una biblioteca de Python para la extracción de tablas de documentos PDF. Camelot permite extraer datos tabulares de PDFs de manera precisa y eficiente, facilitando la conversión de información en formatos estructurados que pueden ser fácilmente analizados y procesados.

[1] https://www.python.org
[2] https://pymupdf.readthedocs.io/en/latest/
[3] https://pypi.org/project/beautifulsoup4/
[4] https://www.langchain.com
[5] https://camelot-py.readthedocs.io/en/master/

- **Playwright** [6] es una biblioteca de automatización de pruebas para aplicaciones web. Playwright soporta múltiples navegadores y proporciona una API robusta para interactuar con la interfaz de usuario, simular eventos y recopilar información de las páginas web. En nuestra aplicación, Playwright se utiliza para automatizar la navegación y extracción de datos de sitios web, asegurando que la información sea recopilada de manera precisa y eficiente.

4.2 Base de datos

4.2.1. Microsoft SQL Server

Microsoft SQL Server [7] es un sistema de gestión de bases de datos relacional (RDBMS) que ofrece robustez, escalabilidad y seguridad. Es utilizado para almacenar toda la información extraída y procesada por los scripts de nuestra aplicación. SQL Server permite manejar grandes volúmenes de datos y realizar consultas complejas de manera eficiente, garantizando la integridad y disponibilidad de los datos.

4.2.2. SQL Server Management Studio 20

SQL Server Management Studio 20 (SSMS) [8] es una herramienta de administración para trabajar con Microsoft SQL Server. SSMS proporciona una interfaz gráfica que facilita la gestión de bases de datos, la escritura y ejecución de consultas SQL, y la realización de tareas administrativas como copias de seguridad, restauraciones y optimización de rendimiento. Es una herramienta esencial para la gestión y mantenimiento de la base de datos de nuestra aplicación.

4.3 Frontend

4.3.1. React

React [9] es una biblioteca de JavaScript para construir interfaces de usuario interactivas y dinámicas. Es desarrollada y mantenida por Facebook y una comunidad de desarrolladores. En nuestra aplicación, React se utiliza para desarrollar el frontend, proporcionando una experiencia de usuario responsiva y eficiente. Su enfoque basado en componentes permite crear interfaces reutilizables y mantenibles, facilitando el desarrollo y la actualización de la aplicación.

Se ha decidido emplear React debido a su enfoque en componentes, lo que facilita la creación de interfaces de usuario reutilizables y mantenibles. Además, la gran comunidad de soporte y su amplia adopción en la industria aseguran el acceso a numerosos recursos, bibliotecas y herramientas adicionales, lo que ha permitido aprender y aplicar la tecnología en poco tiempo. Comparado con otras opciones como Angular o Vue.js, React ofrece una curva de aprendizaje más suave y mayor flexibilidad para integrarse con otras tecnologías.

[6] https://playwright.dev/python/
[7] https://www.microsoft.com/en-us/sql-server/sql-server-downloads
[8] https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms
[9] https://react.dev

4.3.2. Vite

Vite [10] es un build tool moderno y rápido que se utiliza para el desarrollo de aplicaciones web. Vite proporciona una experiencia de desarrollo ágil con tiempos de arranque casi instantáneos y actualizaciones en caliente rápidas, lo que mejora significativamente la productividad. En nuestra aplicación, Vite se utiliza para construir y servir el frontend, optimizando el rendimiento y la eficiencia durante el desarrollo.

4.3.3. Bootstrap

Bootstrap [11] es un framework de CSS que facilita el diseño y la creación de sitios web responsivos y modernos. En combinación con React, Bootstrap se utiliza para estilizar la interfaz de usuario, asegurando un diseño consistente y atractivo en diferentes dispositivos y tamaños de pantalla. Ofrece una amplia gama de componentes predefinidos y estilos que aceleran el desarrollo del frontend.

4.3.4. CSS

CSS (Cascading Style Sheets) [12] se utiliza para definir y aplicar estilos personalizados a los elementos HTML en la interfaz de usuario. En nuestra aplicación, CSS complementa a Bootstrap, permitiendo ajustes y personalizaciones específicas que mejoran la apariencia y usabilidad del frontend. CSS permite un control preciso sobre el diseño y la disposición de los elementos en la página, asegurando una experiencia de usuario agradable y coherente.

4.4 Backend

4.4.1. Django

Django [13] es un framework de desarrollo web en Python que sigue el principio de 'no te repitas' (DRY) y promueve el desarrollo rápido y limpio. En nuestra aplicación, Django se encarga del backend, gestionando la lógica del servidor, la comunicación con la base de datos y la administración de usuarios. Proporciona una estructura sólida y segura para desarrollar y desplegar aplicaciones web de manera eficiente. Django incluye una serie de herramientas y características integradas, como un ORM (Object-Relational Mapping), un sistema de autenticación, y una interfaz administrativa, que facilitan el desarrollo y la gestión de la aplicación.

4.5 Pruebas

4.5.1. Postman

Postman[14] es una herramienta de desarrollo de APIs que permite realizar pruebas y enviar solicitudes HTTP a servicios web. Su principal propósito es ayudar a los desarro-

[10] https://vitejs.dev
[11] https://react-bootstrap.netlify.app
[12] https://developer.mozilla.org/en-US/docs/Web/CSS
[13] https://www.djangoproject.com
[14] https://www.postman.com

lladores a diseñar, probar y depurar APIs de manera eficiente. Ofrece una interfaz gráfica para interactuar con APIs, validar sus respuestas y documentar las pruebas realizadas. De esta manera, ayuda a identificar errores en la API y a verificar que los endpoints cumplan con los requisitos especificados, mejorando así la calidad y funcionalidad de las interfaces de programación.

4.5.2. unittest

unittest[15] es un marco de pruebas integrado en el lenguaje de programación Python. Está diseñado para facilitar la creación y ejecución de pruebas unitarias, que verifican el funcionamiento de unidades individuales de código, como funciones y métodos. Proporciona herramientas para escribir, organizar y ejecutar pruebas unitarias en Python, incluyendo características para verificar que cada componente del software opere de manera correcta de forma aislada. Esto permite a los desarrolladores detectar y corregir errores en las unidades de código de manera temprana en el ciclo de desarrollo, asegurando que cada pieza del software funcione según lo previsto.

[15] https://docs.python.org/3/library/unittest.HTML

CAPÍTULO 5
Diseño de la solución

En esta sección, llevaremos a cabo un diseño detallado de las diferentes partes de la herramienta, asegurando que cada componente se planifique meticulosamente para garantizar su funcionalidad y eficiencia. El proceso de diseño se abordará en varias etapas, comenzando con la identificación y análisis de los datos necesarios y finalizando con la planificación de la arquitectura de la aplicación web.

Finalmente, destacaremos las herramientas y tecnologías a utilizar para cada componente del proyecto, determinando su utilidad y justificando las elecciones basadas en criterios de rendimiento, facilidad de uso y capacidad de integración.

5.1 Análisis de las variables y documentos

En esta sección se aborda el análisis de los diferentes documentos HTML y PDF para identificar dónde está disponible cada información y optimizar su extracción.

El número total de variables a extraer es aproximadamente de 60. Estas variables corresponden con las que se muestran en el diseño de la base de datos de la Figura 5.3. Algunos ejemplos son: los importes del contrato, el tipo de tramitación, tipo de procedimiento, información de criterios de adjudicación, penalidades a tener en cuenta, valoraciones de las empresas, entre otras.

Existen cuatro apartados principales de la licitación a los cuales se puede acceder en formato HTML. Estos apartados son: Anuncio de Licitación, Pliego, Adjudicación y Formalización.

Resumen Licitación

Publicación en plataforma	Documento	Ver documentos
⊞ 11/06/2021 09:41:03	Rectificación de Anuncio de Licitación	Html Xml Pdf Sello de Tiempo
⊞ 11/06/2021 09:51:51	Rectificación de Pliego	Html Xml Pdf Sello de Tiempo
04/02/2022 11:47:39	Adjudicación	Html Xml Pdf Sello de Tiempo
09/03/2022 12:39:04	Formalización	Html Xml Pdf Sello de Tiempo

Figura 5.1: Apartados de un expediente

Por otro lado, los documentos PDF que es importante analizar son:

- **Anexo I**: se encuentra dentro del apartado del Pliego y se encuentra información relacionada con los criterios de adjudicación, diferentes componentes del valor es-

timado del contrato, penalidades, la unidad encargada, las solvencias técnica y económica, la clasificación exigida y todo lo relacionado con el pago.

- **Acta del órgano de asistencia**: Estos documentos se encuentran en el apartado 'Otros documentos' (Ver Figura 5.2) y contienen las valoraciones de las diferentes empresas respecto a los diferentes criterios. También pueden llamarse: Propuesta de Adjudicación, Acta de Adjudicación, Aprobación de Expediente, Apertura de Ofertas e Informe de Criterios.

- **Informes de valoración de los criterios de adjudicación cuantificables mediante juicio de valor**: Estos documentos se encuentran también en 'Otros Documentos'. En ellos se encuentran las valoraciones de las empresas a los subcriterios de los criterios de adjudicación cuantificables mediante juicio de valor.

Otros Documentos

Publicación en plataforma	Documento		
21/05/2021 10:37:51	Acuerdo de iniciación del expediente	Ver	Sello de Tiempo
21/05/2021 10:38:35	instrucciones y recomendaciones	Ver	Sello de Tiempo
21/05/2021 10:41:32	Memoria justificativa	Ver	Sello de Tiempo
21/05/2021 10:45:58	nota informativa ANEXO I BIS y huella electronica	Ver	Sello de Tiempo
21/05/2021 10:46:57	Composición de la mesa de contratación	Ver	Sello de Tiempo
21/05/2021 10:49:05	Documento de aprobación del expediente	Ver	Sello de Tiempo
25/05/2021 08:49:36	presupuesto bc3	Ver	Sello de Tiempo
06/07/2021 09:44:14	Acta del órgano de asistencia	Ver	Sello de Tiempo
12/07/2021 10:01:41	Acta del órgano de asistencia	Ver	Sello de Tiempo
27/07/2021 08:08:39	Acta del órgano de asistencia	Ver	Sello de Tiempo
27/07/2021 08:09:34	Informe de valoración de los criterios de adjudicación cuantificables mediante juicio de valor	Ver	Sello de Tiempo
28/07/2021 12:59:47	nota de aplazamiento de apertura	Ver	Sello de Tiempo
29/07/2021 09:25:12	PRESENTACIÓN RECURSO ESPECIAL	Ver	Sello de Tiempo
06/08/2021 12:01:35	Resolución TARC medidas cautelares	Ver	Sello de Tiempo
29/10/2021 14:00:30	Resolución TACRC levantamiento Suspensión	Ver	Sello de Tiempo
04/11/2021 15:18:03	Acta del órgano de asistencia	Ver	Sello de Tiempo
16/11/2021 14:50:44	aplazamiento mesa apertura plicas	Ver	Sello de Tiempo
22/11/2021 12:28:37	Acta del órgano de asistencia	Ver	Sello de Tiempo
22/11/2021 12:35:27	Informe de valoración de los criterios de adjudicación cuantificables mediante juicio de valor	Ver	Sello de Tiempo
01/12/2021 12:18:40	Acta del órgano de asistencia	Ver	Sello de Tiempo
14/12/2021 07:58:14	Acta del órgano de asistencia	Ver	Sello de Tiempo
14/12/2021 07:59:16	Informe sobre las ofertas incursas en presunción de anormalidad	Ver	Sello de Tiempo

Figura 5.2: Otros Documentos del expediente

Una vez establecidas las diferentes fuentes de las que disponemos para extraer la información, procedemos a analizar una por una todas las variables que queremos extraer y la mejor forma de proceder, es decir, de dónde podemos extraerlas con facilidad y precisión.

- Es necesario utilizar el LLM para el procesamiento y extracción de los criterios de adjudicación con su peso, ya que estos se encuentran en una sección que requiere comprensión gramatical para poder ser extraídos con precisión. Debido a la complejidad y variabilidad del lenguaje natural utilizado en estos documentos, no es posible proceder con la extracción utilizando expresiones regulares ni otras técnicas de separación de texto. En su lugar, se requiere el uso de una herramienta capaz de entender el texto tanto gramatical como contextualmente.

- Por otro lado, variables como el sistema de precios, el abono a cuentas y la unidad encargada son variables que, aunque a primera vista pueden parecer sencillas de extraer de forma directa con expresiones regulares, la realidad es que la presentación de esta información varía significativamente entre los expedientes. A veces, estas variables aparecen simplemente mencionadas en el texto; en otras ocasiones, se presentan seleccionadas con una cruz entre varias opciones, o bien se muestran con casillas de 'Sí' y 'No' donde se marca la respuesta correcta. Debido a esta variabilidad en la presentación, es necesario el uso del LLM, que puede comprender el texto en cualquiera de sus formatos y extraer la información de manera precisa.

5.2 Diseño del Módulo de Extracción de Información

Antes de empezar con el diseño del módulo, es importante establecer las precondiciones que deben cumplir los expedientes para ser procesados, así como la entrada y salida del sistema.

Precondiciones

Los expedientes deben cumplir una serie de requisitos para ser procesados y extraídos por la aplicación:

- El **lugar de ejecución** del proyecto debe ser en la Comunidad Valenciana, ya que este es el ámbito específico de desarrollo del proyecto. Esta delimitación geográfica se realiza para acotar y focalizar el desarrollo, permitiéndonos trabajar de manera más eficiente en la extracción de datos de los documentos específicos de esta comunidad. Cada comunidad autónoma en España puede tener documentos con diferentes nombres y estructuras, lo que requeriría desarrollar un sistema adaptado a cada caso. Al centrarnos exclusivamente en la Comunidad Valenciana, podemos estandarizar y optimizar el proceso de análisis y extracción de datos, asegurando una mayor precisión y relevancia en los resultados obtenidos.

- Solo se considerarán las licitaciones **publicadas** a partir de 2018, ya que antes de ese año solo se publicaban los anuncios de licitación sin los documentos asociados, lo que impedía consultar los detalles y fuentes necesarias para un análisis completo. En 2018 se implementó una estandarización que incorporó la publicación de todos los documentos relevantes junto con las licitaciones, manteniendo una estructura uniforme. Al limitar el análisis a las licitaciones desde 2018, podemos desarrollar un sistema de extracción de datos más eficiente y preciso, enfocado en una estructura definida y consistente, y evitando la falta de información de los años anteriores.

- Solo consideraremos las licitaciones cuya **forma de presentación** sea electrónica, ya que, en el caso de la presentación manual, aumenta el riesgo de encontrarnos con documentos escaneados y faltantes. Esta elección asegura que los documentos sean más uniformes y legibles, reduciendo significativamente la posibilidad de errores y omisiones en el proceso de extracción de datos. Al enfocarnos en las presentaciones electrónicas, garantizamos una mayor calidad y coherencia en la información extraída.

- Se tendrán en cuenta únicamente las licitaciones cuyo **estado** sea 'Resuelta', ya que el objetivo de esta herramienta es la visualización completa de la información sobre una licitación, incluyendo los resultados de la adjudicación y todo el proceso de

evaluación de las empresas. En el caso de las licitaciones no resueltas, esta información aún no estaría disponible, lo que limitaría la capacidad de la herramienta para ofrecer una visión integral y detallada del proceso de contratación. Al enfocarnos en licitaciones resueltas, aseguramos que todos los datos necesarios estén presentes para su análisis y visualización.

- Solo se tendrán en cuenta los expedientes **sin lotes**. Los lotes se refieren a la división de un proyecto o contrato en partes más pequeñas y específicas que se pueden adjudicar por separado. Esto permite a diferentes proveedores o contratistas presentar ofertas para uno o varios lotes, en lugar de para el contrato completo. Esto complicaría mucho la extracción dado que se deberían de extraer las valoraciones y el adjudicatario de cada lote en vez del contrato completo, se ha decidido considerar únicamente aquellos que no tengan dicha división.

Entrada

La entrada del módulo será el HTML de los detalles de la licitación de donde se podrá acceder a los diferentes archivos PDF y apartados. De esta manera, necesitamos un paso previo que nos permita acceder a la plataforma y establecer los criterios de búsqueda especificados en las precondiciones. Para ello usaremos librerías y scripts Python que nos permitan acceder a una URL y manipularla para obtener las licitaciones con su HTML correspondiente. Este HTML será procesado por el módulo y se accederá a todos sus apartados y documentos relevantes.

Salida

La salida del módulo será un JSON en el que se especifica toda la información extraída de esa licitación. A medida que se procesen los distintos archivos, se irán añadiendo secciones al JSON hasta finalizar el análisis de todos los documentos relevantes. Este JSON será utilizado posteriormente para introducir los datos de ese expediente en la base de datos.

Partes del módulo

El módulo estará formado por diferentes partes encargadas de procesar diferente información.

- **Main**: Script principal que se encarga de extraer el HTML de cada licitación y llamar a las demás funciones para su procesamiento.
- **Extraer Anuncio y Formalización**: Esta sección se encarga de extraer las fechas del anuncio de la licitación y de la formalización desde el HTML correspondiente.
- **Extraer Adjudicación**: Script encargado de extraer la información necesaria de la Adjudicación tal como la fecha y la información del adjudicatario.
- **Extraer Pliego**: parte encargada de extraer la información del HTML del pliego y llamar al script que procesa el PDF Anexo I.
- **Procesar Anexo I**: Script que se encarga de leer y separar por secciones el contenido del Anexo I. Utiliza expresiones regulares para extraer la información estructurada relevante.

- **LLM Anexo I**: Script Utiliza el LLM para extraer información específica de algunas secciones del texto del Anexo I que se han definido como relevantes.
- **LLM Valoraciones**: Script que se encarga de leer los diferentes documentos donde se reflejan las valoraciones de las empresas según diferentes criterios y extrae las tablas en formato de texto con dichas puntuaciones. Estas tablas se proporcionan al LLM junto al listado de criterios que se desean extraer para cada empresa. El LLM devuelve un diccionario donde cada empresa está asociada a un sub-diccionario con cada criterio y su correspondiente puntuación.

5.3 Diseño de la Base de Datos SQL

Para el diseño de la base de datos es necesario definir un modelo de datos que refleje adecuadamente la estructura de la información que se va a almacenar. Esto incluye la identificación de entidades, atributos y relaciones entre ellas. El diseño conceptual de la base de datos se puede ver en la Figura 5.3

Este diseño conceptual se interpreta de la siguiente forma:

- *Licitaciones*: contiene todos los detalles de una licitación, incluyendo las diferentes fechas, importe, formas de pago, penalidades, garantías, entre otros.
- *Empresas*: contiene el nombre de la empresa, su NIF y si es PYME o no.
- *Criterios*: criterio que deben cumplir las empresas que participan en una licitación, se almacena su nombre, siglas, puntuación máxima que pueden alcanzar y puntuación mínima para no ser excluidas.
- *Participaciones*: cada empresa debe participar en una licitación, por lo que es necesaria una tabla que almacene las participaciones junto al importe económico que ha presentado esa empresa en la licitación, así como si ha presentado una oferta anormal y si ha resultado excluída.
- *Valoraciones*: cada participación tendrá una serie de valoraciones a diferentes criterios.
- *Códigos CPV*: se almacenan los códigos CPV junto a su descripción.
- *Tipo de tramitación*: se almacenan los diferentes tipos de tramitaciones que puede tener una licitación.
- *Procedimiento*: se almacenan los diferentes procedimientos que debe tener una licitación.
- *Tipo de contrato*: se almacenan los tipos de contrato que puede tener una licitación.
- *Links*: se almacenan los diferentes enlaces con su URL que llevan a una sección específica de la licitación indicada por su *Tipo Link*.
- *Tipo Link*: Almacena los diferentes tipos que puede tener un enlace, es decir, la sección a la que apunta dicho enlace.

En cuanto a las relaciones entre las diferentes tablas:

- Cada *Licitación* debe tener un *Adjudicatario*, es decir, una *Empresa* a la que se adjudica dicha licitación.

- Cada *Licitación* contiene una serie de *Participaciones* de *Empresas*. Estas *Participaciones* deben tener una *Valoración* asociada a cada uno de los *Criterios* de adjudicación de dicha *Licitación*.
- Un *Criterio* puede ser un subcriterio de otro. De la misma manera, un *Criterio* puede tener varios subcriterios asociados.
- Una *Licitación* debe contener un *Tipo de contrato*, *Tipo de tramitación* y *Procedimiento* si la información se encuentra disponible.
- Una *Licitación* puede tener varios *Códigos CPV* asociados, por lo que se deberá crear una tabla intermedia debido a la relación *many-to-many* que existe entre *Licitación* y *Códigos CPV*.
- Una *Licitación* tendrá varios *Links* que apuntarán a diferentes secciones de la *Licitación*. El *Tipo Link* indica la sección a la que apunta dicho enlace.

5.4 Arquitectura de la Aplicación Web

La arquitectura de la aplicación web está diseñada para ser modular y escalable, aprovechando tecnologías modernas tanto en el frontend como en el backend. A continuación se describe cada componente y su interacción dentro del sistema:

5.4.1. Frontend

El frontend de la aplicación está desarrollado utilizando React, una biblioteca de JavaScript para la construcción de interfaces de usuario interactivas. A continuación se describen los principales componentes y tecnologías utilizados en el frontend:

- **React**: Utilizamos React para crear una interfaz de usuario dinámica y receptiva. Los componentes de React permiten una actualización eficiente y la gestión del estado de la interfaz.
- **Fetch API**: Para la comunicación con el backend, utilizamos la Fetch API de JavaScript. Esta API permite realizar solicitudes HTTP asíncronas al servidor Django, obteniendo datos y enviando información según sea necesario.
- **CSS y Bootstrap**: La apariencia y el diseño de la interfaz se manejan utilizando CSS y Bootstrap. CSS proporciona un estilo personalizado y detallado, mientras que Bootstrap facilita el diseño responsivo y componentes de interfaz de usuario preconstruidos.

5.4.2. Backend

El backend de la aplicación está construido con Django y Django REST Framework, proporcionando una API robusta y segura. Los principales componentes del backend incluyen:

- **Django**: Django es el framework principal utilizado para el desarrollo del backend. Proporciona una estructura robusta para el desarrollo web, incluyendo administración de usuarios, seguridad y manejo de datos.

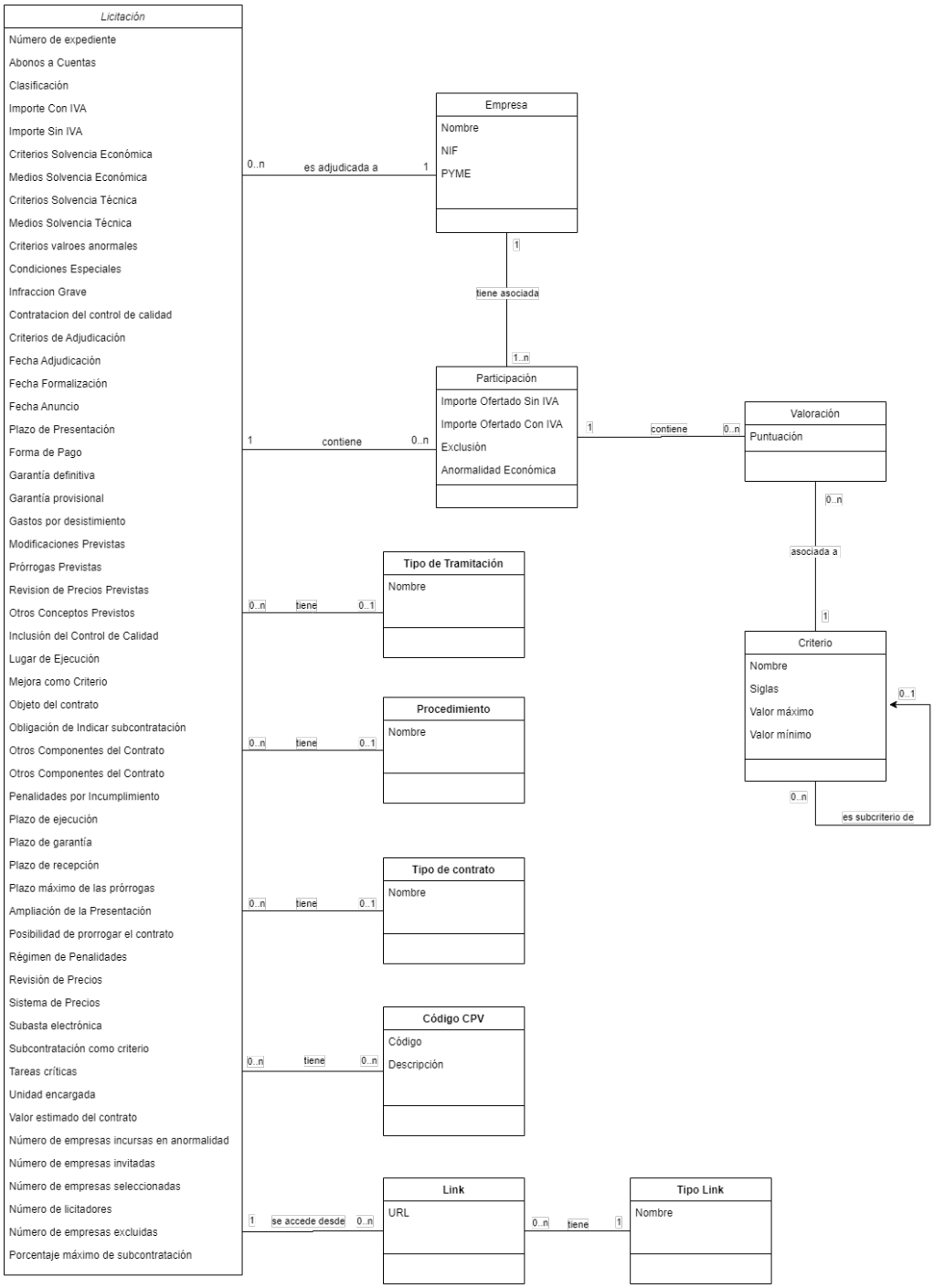


Figura 5.3: Diseño conceptual de la base de datos

- **Django REST Framework (DRF)**: Utilizamos DRF para construir la API RESTful que permite la comunicación con el frontend. DRF facilita la serialización de datos y proporciona herramientas para manejar autenticación, permisos y vistas de API.
- **Microsoft SQL Server**: La base de datos SQL se gestiona a través de Microsoft SQL Server. Django se conecta a esta base de datos para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) utilizando su ORM (Object-Relational Mapping) integrado.
- **Módulo de Extracción de Datos**: En el backend también tenemos scripts que se conectan a la URL de la plataforma de contratación del sector público. Estos scripts extraen información relevante y la insertan en la base de datos SQL. Estos procesos pueden ser automatizados y programados para ejecutarse periódicamente.

5.4.3. Interacción entre Componentes

La interacción entre los componentes del sistema se realiza de la siguiente manera:

- **Comunicación Frontend-Backend**: El frontend en React utiliza la Fetch API para enviar solicitudes HTTP al backend. Estas solicitudes pueden ser GET para obtener datos, POST para enviar nuevos datos, PUT para actualizar datos existentes y DELETE para eliminar datos.
- **Manejo de Datos en el Backend**: Django REST Framework recibe las solicitudes del frontend, procesa los datos y realiza las operaciones necesarias en la base de datos SQL Server.
- **Actualización de Datos**: Los scripts de extracción de datos en el backend se conectan a la plataforma de contratación del sector público, extraen la información y la almacenan en la base de datos. Esto garantiza que los datos de la aplicación estén siempre actualizados.
- **Presentación de Datos**: Una vez que los datos están disponibles en el backend, pueden ser solicitados por el frontend a través de la API RESTful, permitiendo a los usuarios interactuar con la información actualizada en tiempo real.

5.4.4. Diagrama de Arquitectura

Se muestra un diagrama simplificado de la arquitectura de la aplicación web en la Figura 5.4. donde se muestran los diferentes componentes mencionados anteriormente.

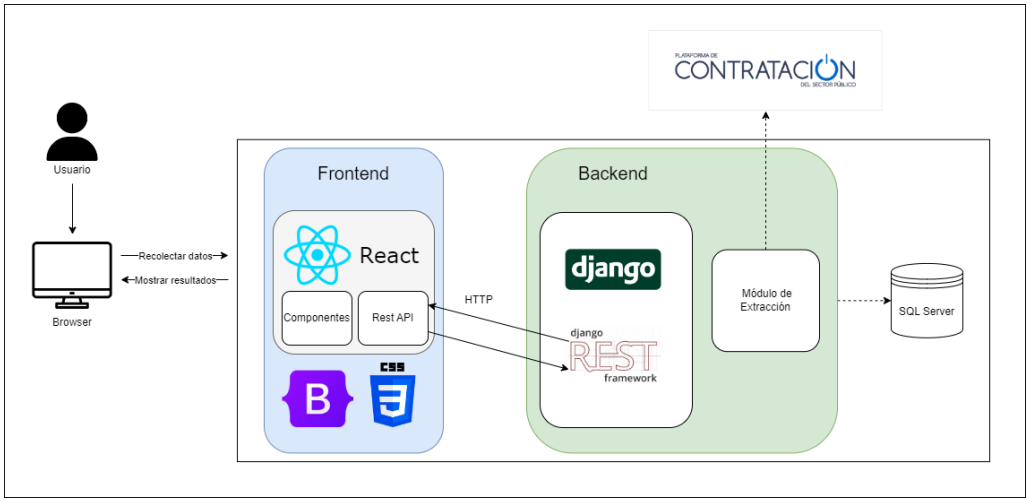


Figura 5.4: Arquitectura Web

CAPÍTULO 6
Desarrollo e Implementación de la solución

En esta sección, abordaremos el proceso de desarrollo e implementación de la solución propuesta, detallando cada etapa y los componentes involucrados. A lo largo del capítulo, se describirán la integración de los diferentes módulos que conforman la aplicación.

Cada sección de este capítulo se centrará en un aspecto específico del desarrollo, proporcionando una visión clara y detallada del proceso completo. Desde la configuración inicial hasta las pruebas finales y la optimización, se explicará cómo se ha construido la solución para garantizar su funcionalidad, eficiencia y escalabilidad.

6.1 Módulo de Extracción de Información

En primer lugar se debe de establecer la conexión con PLACSP para obtener el HTML asociado a la URL de los diferentes expedientes.

Para ello, es crucial emplear Playwright debido a la complejidad de la interacción con la plataforma. La página web de PLACSP está diseñada de tal manera que gran parte de su funcionalidad está basada en JavaScript. Específicamente, los botones y enlaces en la interfaz de usuario llaman a funciones JavaScript que no están directamente accesibles a través de métodos tradicionales de extracción de HTML.

Cuando se realiza una acción en la interfaz, como hacer clic en un botón de búsqueda o seleccionar filtros, estas acciones disparan funciones JavaScript que generan dinámicamente el contenido de la página o redirigen a otras secciones. Si intentamos copiar la URL directamente del navegador y acceder a ella en una sesión privada o en una herramienta de solicitud HTTP, el resultado es una página que indica que el recurso no existe. Esto ocurre porque la URL generada no es directamente accesible sin que las funciones JavaScript adecuadas se ejecuten primero.

Playwright resuelve este problema proporcionando un entorno en el que podemos simular interacciones de usuario de forma precisa. Utilizando Playwright, podemos navegar por la plataforma, aplicar filtros y realizar búsquedas exactamente como lo haría un usuario real. El script desarrollado con Playwright se encarga de interactuar con los elementos dinámicos de la página y de manejar la redirección y la carga de contenido generado por JavaScript. De este modo, garantizamos que el HTML que obtenemos refleja el estado completo y actualizado de la página, incluyendo toda la información que de otro modo no estaría disponible si se accede directamente a través de una URL estática.

AccessPagePlaywright.py

Una vez que hemos comprendido la necesidad de utilizar Playwright para interactuar con la plataforma PLACSP, el siguiente paso es diseñar y desarrollar un script que establezca la conexión con la plataforma PLACSP y almacene el HTML de cada uno de los expedientes que se desean extraer. Este script debe ser capaz de interactuar con la página web de manera dinámica, simulando las acciones de un usuario real para aplicar filtros de búsqueda y navegar por los resultados. Una vez identificados los expedientes de interés, el script debe acceder a las páginas específicas de cada expediente y extraer su contenido HTML para almacenarlo de forma segura, asegurando que la información se encuentra completa y actualizada.

El diseño del script se centró en dividir la tarea en módulos o funciones clave, cada uno de los cuales se encargaba de un aspecto específico del proceso:

1. **Conexión con la base de datos**: Una función inicial que estableciera una conexión con la base de datos SQL Server utilizando `pyodbc`. Esta conexión es fundamental para realizar consultas para verificar si un expediente ya ha sido procesado anteriormente.

2. **Navegación y búsqueda en PLACSP**: La función principal que se encargaría de abrir la página web de PLACSP, aplicar los filtros de búsqueda (como fechas, tipo de presentación, estado del expediente, etc.), y navegar a través de los diferentes resultados. Esta parte del script debía manejar interacciones como hacer clic en botones, seleccionar opciones de menú desplegable, y manejar ventanas emergentes o diálogos que pudieran aparecer.

3. **Extracción y verificación de expedientes**: Una vez obtenidos los resultados de la búsqueda, se diseñó una función para recolectar los enlaces de los expedientes. Para cada expediente, se verificaba si ya existía en la base de datos. Si no, el script procedía a extraer el HTML de la página específica de ese expediente.

4. **Gestión de múltiples páginas de resultados**: Dado que las búsquedas en PLACSP pueden arrojar una gran cantidad de resultados, fue necesario implementar una lógica que permitiera al script navegar por múltiples páginas de resultados. Esto incluía hacer clic en el botón 'Siguiente' hasta que se llegara al final de los resultados disponibles.

5. **Extracción de HTML detallado**: Para cada expediente que no estuviera ya en la base de datos, se desarrolló una función que accediera a la página individual del expediente y extrajera todo el contenido HTML. Esta información se almacenaría para su posterior procesamiento.

extraerAdjudicacion.py

Este script tiene como objetivo extraer información relevante sobre la adjudicación de la licitación a partir de su HTML. Para la extracción de la información utiliza BeautifulSoup para analizar el contenido HTML y obtener datos específicos. Los datos que se busca son:

- Fecha de Adjudicación
- Nombre/Razón Social del Adjudicatario
- Plazo de Ejecución

- NIF del Adjudicatario
- Si el Adjudicatario es una PYME
- Importe total ofertado (sin IVA) por el Adjudicatario
- Importe total ofertado (con IVA) por el Adjudicatario

Para que este script funcione correctamente, es necesario analizar bien el contenido HTML de esta sección ya que es la misma estructura para todas las licitaciones. Por ello, debemos saber qué tipo de etiqueta tiene cada uno de los elementos que debemos extraer.

Se utiliza una función `extraer_fecha` que busca la fecha de adjudicación en el primer encabezado de nivel cinco en el HTML, utilizando una expresión regular para encontrar el formato de fecha `dd-mm-yyyy`. Si encuentra una fecha, la devuelve; si no, retorna `None`.

Inicialmente, crea un diccionario con los campos necesarios, todos inicializados como `None`. Luego, llama a `extraer_fecha` para obtener y almacenar la fecha de adjudicación. A continuación, busca el nombre del adjudicatario en el primer encabezado de nivel 4. Finalmente, revisa todos los elementos de tipo `<span>` en el HTML para extraer la información adicional, comparando el texto con los datos que se buscan. Toda esta información se guarda en el diccionario y se devuelve al final.

extraerPliego.py

Similar a *extraerAdjudicacion.py*, este script utiliza BeautifulSoup para leer el contenido HTML del Pliego y almacenar en un diccionario una serie de variables. Estas variables son:

- Número de expediente
- Objeto del contrato
- Lugar de ejecución del contrato
- Tipo de contrato
- Tipo de tramitación
- Importe (sin IVA)
- Importe (con IVA)
- Valor estimado del contrato
- Tipo de procedimeinto
- Plazo de ejecución
- Condiciones especiales de ejecución
- Plazo de presentación
- Clasificación CPV

Antes de proceder a la extracción de estas variables, el script analiza el HTML para identificar si el expediente está dividido en lotes. Solo se consideran expedientes sin lotes, por lo que, si se detecta la existencia de uno o más lotes, el expediente se descarta automáticamente y se continúa con el procesamiento del siguiente.

Además, el script se encarga de localizar y extraer el documento PDF correspondiente al Anexo I, un documento que puede contener información adicional relevante. Para ello, se buscan enlaces en el HTML que incluyan palabras clave como 'ANEXO I'. Si se encuentra un enlace adecuado, se descarga el PDF mediante la biblioteca requests, y luego se convierte en un objeto manipulable utilizando *fitz*.

El contenido del PDF se procesa en dos etapas: primero, se pasa al script *procesarTextoPDF.py*, que extrae las variables asociadas a secciones completas del documento que no requieren el uso de un modelo de lenguaje para su interpretación. Luego, se invoca el script *extraerSeccionesLLM.py*, que se encarga de extraer aquellas variables que necesitan ser analizadas mediante un LLM para su correcta comprensión.

Finalmente, si se extrae información adicional del Anexo I, esta se integra con la información obtenida del pliego, combinando ambas fuentes en un único diccionario. Este diccionario se devuelve junto con el enlace al Anexo I.

procesarTextoPDF.py

Este código está diseñado para procesar el documento PDF del Anexo I y organizar su contenido en secciones, facilitando la extracción de variables clave de manera precisa y eficiente.

El proceso comienza leyendo cada página del documento PDF y extrayendo el texto correspondiente. Durante la extracción, se eliminan elementos no deseados como encabezados, pies de página y números de página, asegurando que el texto resultante esté limpio y enfocado en la información relevante.

Una vez extraído, el texto se divide en fragmentos o 'tokens', que corresponden a frases individuales del documento. Estas frases se agrupan en un diccionario de secciones basado en el formato del texto: si una frase está en mayúsculas, se interpreta como un título de sección y se utiliza como clave en el diccionario. Las frases que siguen a este título se almacenan como el valor correspondiente a esa clave, hasta que se encuentra un nuevo título de sección.

A continuación, se revisa el diccionario y se compara el contenido de las claves (títulos de secciones) con una lista predefinida de variables de interés. Si hay una coincidencia, el texto asociado a esa sección se extrae y se asigna a la variable correspondiente.

El resultado final es un diccionario que contiene todas las variables de interés con su texto asociado.

extraerSeccionesLLM.py

Este script está diseñado para la extracción estructurada de la información relevante con estructuras variables que dificultan la extracción mediante métodos tradicionales como expresiones regulares.

El proceso comienza con la lectura completa del documento PDF. Tras extraer todo el texto, este se divide en secciones lógicas usando expresiones regulares para identificar la palabra clave 'APARTADO', seguida de una letra en mayúscula que denota una sección específica. Esta división en secciones facilita la organización del contenido del documento en partes manejables, tratadas como unidades independientes. Cada sección se almacena en un diccionario, donde las claves son los nombres de las secciones (por ejemplo, 'APARTADO LL', 'APARTADO T') y los valores corresponden al texto asociado.

El siguiente paso es la generación de prompts específicos que se envían al LLM. En este caso, usamos GPT-4o-mini por su bajo coste y buena eficacia. Estos prompts están

diseñados para extraer información detallada de cada sección del documento. Ejemplos de información a extraer incluyen criterios de adjudicación, subcriterios, penalidades por incumplimiento y condiciones de subcontratación. Cada prompt está cuidadosamente formulado para asegurar que el modelo de lenguaje interprete correctamente la sección del documento y devuelva la información relevante en el formato deseado.

Esta sería la plantilla que se utiliza para cada prompt que se desarrolla:

```
prompt = PromptTemplate.from_template(
    """ Contesta a la pregunta basandote en el contexto.
Contexto: {section}
Pregunta: {query}
Respuesta: """
)
```

Para cada llamada al LLM tenemos que formatear el prompt con la pregunta asociada a la variable que queremos extraer y el contexto sería el apartado donde se encuentra dicha variable. Un ejemplo sería:

```
promptCriterios = prompt.format(
    section=sections["APARTADO LL"],
    query="Imprime una lista Python en formato [{'Nombre': None, 'Siglas': None
        , 'Puntuación máxima': None, 'Puntuación mínima': None}] con los
        criterios principales separados por comas.",
)
```

Para la creación de prompts y llamadas al LLM se utiliza la biblioteca Langchain que proporciona diversos métodos que facilitan la comunicación con el LLM y el formateo de las respuestas obtenidas.

Una vez que el modelo de lenguaje procesa los prompts, devuelve la información solicitada. Esta información se organiza en un diccionario centralizado, que actúa como repositorio de toda la información extraída del documento y es retornado al final del proceso.

extraerTablas.py

Este script está diseñado para extraer las valoraciones y las ofertas de cada empresa participante en un proceso de licitación pública. Esta información generalmente se encuentra en tablas dentro de los documentos de actas, lo que hace esencial la capacidad de extraer y procesar la información contenida en estas tablas de archivos PDF. Para lograr esto, el script utiliza `fitz` (PyMuPDF) para extraer el texto plano y `camelot` para las tablas.

Al igual que el script anterior, este también integra el mismo LLM mediante Langchain para interpretar la información contenida en las tablas y asignar correctamente las valoraciones a los criterios correspondientes para cada empresa.

La función principal de este script es `extraer_info_acta`, que recibe dos argumentos clave:

1. **Lista de Documentos PDF**: Estos documentos son extraídos previamente del HTML de la licitación.

2. **Lista de Nombres de Criterios**: Estos nombres de criterios son extraídos del Anexo I del pliego y son las valoraciones asociadas a estos criterios las que se desea extraer de las actas.

El script se encarga de procesar cada uno de los documentos que recibe como argumento hasta encontrar la informaión que se buscaba.

También se extrae de las actas otra información como:

- **Número de empresas invitadas**: Identifica cuántas empresas fueron invitadas a participar en la licitación.
- **Número de licitadores**: Establece cuántas empresas presentaron ofertas.
- **Número de empresas seleccionadas**: Identifica cuántas y cuáles empresas fueron seleccionadas en el proceso.
- **Número de empresas incursas en anormalidad y excluidas, y cuáles**: Informa sobre las empresas que fueron descartadas por incurrir en anomalías, detallando cuáles son.
- **Si la oferta de la empresa adjudicataria es anormal o no**: Evalúa si la oferta final adjudicada presenta características anómalas según los criterios establecidos.

El script también incluye una función secundaria llamada `extraer_ofertas`. Esta función recibe una lista de documentos PDF de las actas y se encarga de llamar a las otras funciones para obtener la oferta económica de cada empresa participante. Esta extracción puede provenir tanto de las tablas como del texto contenido en las actas.

De forma más específica, el flujo de trabajo general del script puede desglosarse en las siguientes etapas:

1. **Extracción de Texto y Datos Tabulares**:
    - La función `read_pdf` se encarga de leer el contenido de los archivos PDF. Utiliza la biblioteca `fitz` para abrir el documento y extraer el texto completo página por página. Este texto es el que luego se procesa para buscar la información relevante.
    - Para la extracción de tablas, el script utiliza `camelot`, una herramienta que permite identificar y extraer tablas de los PDFs, especialmente cuando estas son complejas o contienen datos distribuidos en varias páginas. Dependiendo del número de páginas del documento, el script decide si debe analizar solo las tablas o el texto completo para encontrar las valoraciones de las empresas.
2. **Análisis de Datos**:
    - La función `extract_table_info` se encarga de procesar las tablas extraídas para obtener información sobre las valoraciones de las empresas en diferentes criterios. Este proceso incluye el uso de un modelo de lenguaje que responde a preguntas específicas sobre los datos, estructurando las respuestas en un formato de diccionario Python. Esto es especialmente útil para tratar datos complejos, como valoraciones cualitativas o cantidades numéricas expresadas en diferentes formatos. En caso de que lo que se desea extraer de las tablas sea el importe de la oferta económica de cada empresa, el prompt sería más específico para ese caso para evitar confusiones con cualquier otro valor de puntuaciones.
    - La función `extract_text` complementa este proceso al buscar y extraer información específica, como el número de empresas invitadas, licitadoras y aquellas incursas en anormalidad. Este análisis se realiza utilizando prompts que guían al modelo de lenguaje a generar respuestas precisas a partir del texto extraído del PDF.

3. **Procesamiento y Limpieza de Datos**: A lo largo del script, se implementan diversas técnicas para limpiar y formatear los datos extraídos, como la corrección de nombres de empresas, normalización de números y filtrado de información irrelevante. Por ejemplo, se aplican expresiones regulares para identificar y corregir números en formatos no estándar (como cifras con puntos y comas) y se descartan tablas o textos que no contienen la información deseada.

4. **Construcción de Diccionarios de Resultados**: Finalmente, la información extraída y procesada se organiza en diccionarios Python que contienen claves descriptivas y sus correspondientes valores, como las valoraciones de las empresas, el importe de las ofertas económicas y detalles sobre empresas excluidas. Estos diccionarios se juntan al final para ser retornados con toda la información conjunta.

introducirDatosBD.py

Este script está diseñado para gestionar e insertar datos relacionados con licitaciones en una base de datos SQL Server, donde se recopila, procesa y almacena información sobre empresas participantes, criterios de evaluación, adjudicatarios y detalles específicos de cada licitación.

El proceso comienza con la conexión a la base de datos SQL Server, utilizando el módulo `pyodbc`. Este módulo permite que Python se conecte a bases de datos compatibles con ODBC, en este caso, a una base de datos SQL Server. La conexión se establece a través de un string de conexión que especifica el controlador ODBC, el servidor, la base de datos y las credenciales de acceso necesarias.

La función principal del script, `insertar_expediente`, recibe un diccionario Python que contiene toda la información de una licitación. Esta función actúa como un controlador, llamando a otras funciones especializadas que gestionan la inserción de diferentes tipos de datos en sus respectivas tablas en la base de datos.

A continuación, se definen varias funciones para procesar y transformar los datos extraídos del diccionario Python. Por ejemplo, 'convertNumber' toma una cadena de texto que representa un número en diferentes formatos (con símbolos de moneda, comas, puntos, etc.) y la convierte en un número de punto flotante. Esta función es útil para estandarizar los valores numéricos antes de almacenarlos en la base de datos.

Una de las tareas más complejas del script es la gestión de los nombres de empresas, criterios y subcriterios. Dado que las mismas entidades pueden aparecer con nombres ligeramente diferentes en los datos, se utilizan técnicas de coincidencia difusa ('fuzzy matching') para encontrar las mejores coincidencias entre nombres similares. Las funciones `find_matching_company` y `match_list` implementan estas técnicas, ayudando a asegurar que los datos se unan correctamente a pesar de las variaciones en la nomenclatura. Estas funciones son importantes para detectar si ya existe una empresa en la base de datos con un nombre similar para así obtener su identificador y evitar insertar duplicados, y también porque en el diccionario que contiene las valoraciones a los criterios y a los subcriterios, como estos se obtienen de diferentes documentos PDF puede ocurrir que la misma empresa aparezca con una abreviatura o con diferentes símbolos y es importante detectarlo para sí poder otorgarle a la empresa correspondiente todas sus valoraciones.

Para insertar los datos en la base de datos, se han desarrollado funciones específicas para manejar los diferentes aspectos de la estructura de la base de datos. La función `insertar_criterios` se encarga de insertar los criterios y subcriterios de evaluación en la tabla correspondiente, verificando primero si ya existen para evitar duplicados. De manera similar, `insertar_empresa` gestiona la inserción de las empresas participan-

tes, buscando posibles coincidencias en la base de datos existente. La primera llamada a `insertar_empresa` se realiza con los datos de la empresa adjudicataria para obtener su identificador único, generado automáticamente por la base de datos. Este identificador es necesario para asociar correctamente la empresa adjudicataria con la licitación en cuestión.

La función para la inserción de licitaciones es `insertar_licitacion`, que compila la información sobre una licitación específica y la inserta en múltiples tablas relacionadas. Esta función maneja una gran variedad de campos y tipos de datos, desde información financiera hasta fechas y enlaces web, asegurándose de que todos los detalles de la licitación se registren correctamente en la base de datos.

Por otro lado, la función `insertar_participacion_adjudicatario` se encarga de registrar la participación de las empresas adjudicatarias en cada licitación, incluyendo detalles como el importe ofertado, si la empresa fue excluida, o si su oferta fue considerada anormal. Además, esta función inserta las valoraciones del adjudicatario para cada criterio y subcriterio en la tabla `Valoraciones`, utilizando la función `insertar_valoraciones` junto con el identificador del adjudicatario.

Finalmente, la función `insertar_participacion` se utiliza para registrar la participación de cada una de las empresas no adjudicatarias, llamando a su vez a `insertar_valoraciones` de manera similar a como se hace con la empresa adjudicataria. Esto asegura que todas las valoraciones, tanto de las empresas adjudicatarias como de las participantes, se registren de manera completa y correcta en la base de datos.

main.py

Su función es coordinar la ejecución de múltiples subprocesos y funciones específicas que se encargan de interactuar con la web, extraer la información necesaria, procesarla, y almacenarla en una base de datos. Además, el `main` maneja la integración y estructuración de los datos extraídos, garantizando que toda la información relevante se capture de manera completa y se almacene correctamente para su uso posterior.

El script comienza con la importación de diversas bibliotecas y módulos necesarios para su funcionamiento. Entre estas bibliotecas se incluyen `asyncio` para la gestión de tareas asíncronas necesario para ejecutar el script AccessPagePlaywright, `BeautifulSoup` de `bs4` para el procesamiento de HTML, y `requests` para la realización de solicitudes HTTP. Además, se importan módulos específicos de los scripts mencionados anteriormente.

Se define un diccionario global denominado `links`, que sirve para almacenar los enlaces a los diferentes documentos asociados a cada expediente, tales como el pliego de condiciones, el anuncio de licitación, las actas de adjudicación, y otros documentos relevantes.

El script define varias funciones auxiliares para realizar tareas específicas:

- `get_names`: Esta función extrae y formatea los nombres de los criterios de adjudicación, opcionalmente incluyendo siglas, y los retorna en una lista. Esto es útil para utilizar como argumento para la función de extracción de valoraciones ya que se necesitan los nombres de todos los criterios.

- `open_link`: Esta función recibe un enlace HTML, realiza una petición para obtener el contenido del documento y extrae la URL definitiva del archivo PDF, que luego es descargado y retornado junto con la URL.

- `docs_valoraciones` y `docs_juicio_valor`: Estas funciones buscan documentos específicos relacionados con las valoraciones y los criterios de juicio de valor dentro del HTML de la página, los descargan y los agregan al diccionario `links`.
- `fechas_anuncio_form`: Extrae las fechas de anuncio de licitación y formalización del expediente, si están disponibles, y las almacena en un diccionario.
- `acceder_seccion`: Permite acceder a secciones específicas de la página web que contienen los documentos de interés, como el pliego o la adjudicación. Si encuentra la sección deseada, descarga el documento y guarda su enlace en el diccionario `links`.
- `get_link_licitacion`: Extrae y retorna el enlace directo a la licitación desde el HTML de la página web.

La función `main` es el núcleo del script. Coordina la recopilación de expedientes llamando a `AccessPagePlaywright.py` y la extracción de datos relevantes de cada uno. La función itera sobre los expedientes obtenidos, extrae la información utilizando las funciones auxiliares y los scripts mencionados y compila todos los datos en un diccionario estructurado. Luego, intenta insertar este diccionario en la base de datos llamando al script `introducirDatosBD.py`.

6.2 Base de Datos

A continuación, se describen las etapas seguidas para la instalación del sistema de gestión de bases de datos, la creación del esquema relacional y la configuración de las relaciones entre las tablas.

1. Descarga e Instalación de SQL Server y SQL Server Management Studio

Para el desarrollo de la base de datos utilizada en este proyecto, se procedió a la descarga e instalación de Microsoft SQL Server y SQL Server Management Studio (SSMS). SQL Server se descargó desde la página oficial de Microsoft, eligiendo la versión SQL Server Express, adecuada para el propósito de desarrollo. La instalación de SQL Server se realizó siguiendo el asistente proporcionado, seleccionando las características esenciales para el funcionamiento del sistema de gestión de bases de datos.

Posteriormente, se descargó e instaló SQL Server Management Studio (SSMS) desde la misma página, facilitando la gestión y administración de SQL Server a través de una interfaz gráfica intuitiva.

2. Creación de la Base de Datos

Una vez completada la instalación, se procedió a la creación de la base de datos en SQL Server. Utilizando SSMS, se estableció una nueva base de datos denominada `TFG`. Esta base de datos sirve como contenedor para todas las tablas y objetos necesarios para almacenar y gestionar la información relacionada con las licitaciones.

3. Diseño e Implementación del Esquema de la Base de Datos

El diseño conceptual de la base de datos se implementó siguiendo el esquema definido en el apartado de diseño, el cual incluye las tablas principales y sus relaciones.

En todos los scripts antes de proceder a la creación de la tabla, se debe comenzar con el comando USE TFG, que cambia el contexto a la base de datos llamada TFG. Esto significa que todas las operaciones siguientes se realizarán dentro de esta base de datos.

Todas las tablas se crean utilizando el comando `CREATE TABLE`, seguido del nombre deseado para la tabla. Cada columna se define dentro de la tabla especificando su nombre, tipo de dato y cualquier restricción asociada. Además, todas las tablas incluyen una columna `id`, que es un entero autoincremental y actúa como clave primaria. Esto asegura que cada registro en la tabla tenga un identificador único en esta columna.

A continuación se enseñan algunos de los scripts de creación de algunas tablas importantes, pero los scripts completos de cada tabla se encuentran en el Apéndice.

1. **Licitaciones**

```
USE TFG;
GO

-- Crear la tabla Licitaciones con columnas esenciales
CREATE TABLE Licitaciones (
    id_licitacion INT IDENTITY(1,1) PRIMARY KEY,
    num_expediente VARCHAR(50) NOT NULL,
    importe_con_impuestos MONEY NULL,
    importe_sin_impuestos MONEY NULL,
    ...
    -- Se omiten algunas columnas por brevedad
    ...
    fecha_adjudicacion DATE NULL,
    fecha_formalizacion DATE NULL,
    fecha_anuncio DATE NULL,
    forma_pago NVARCHAR(MAX) NULL,
    tipo_contrato INT NULL,
    tramitacion INT NULL,
    valor_estimado MONEY NULL,
    adjudicatario INT NULL,
    porcentaje_max_subcontratacion FLOAT NULL
);
GO

-- Agregar valores predeterminados
ALTER TABLE Licitaciones
ADD CONSTRAINT DF_Licitaciones_porcentaje_max_subcontratacion
DEFAULT ((60)) FOR porcentaje_max_subcontratacion;
GO

-- Agregar restricciones de clave externa
ALTER TABLE Licitaciones
ADD CONSTRAINT FK_Licitaciones_Empresas
FOREIGN KEY (adjudicatario) REFERENCES Empresas(id_empresa);
GO

ALTER TABLE Licitaciones
ADD CONSTRAINT FK_Licitaciones_TipoContrato
FOREIGN KEY (tipo_contrato) REFERENCES TipoContrato(id_tipo_contrato);
GO

ALTER TABLE Licitaciones
ADD CONSTRAINT FK_Licitaciones_TipoProcedimiento
FOREIGN KEY (procedimiento) REFERENCES TipoProcedimiento(id_procedimiento
    );
GO

ALTER TABLE Licitaciones
```

```
ADD CONSTRAINT FK_Licitaciones_TipoTramitacion
FOREIGN KEY (tramitacion) REFERENCES TipoTramitacion(id_tramitacion);
GO
```

Este script se encraga de crear la tabla de licitaciones con todas sus columnas. Debido a que es una tabla con una gran cantidad de columnas, se han omitido algunas para simplificar el script.

El script continúa añadiendo varias restricciones de clave externa. Estas restricciones aseguran la integridad referencial entre tablas relacionadas. Por ejemplo, la columna `adjudicatario` en la tabla Licitaciones debe coincidir con un valor en la columna `id_empresa` de la tabla Empresas. De manera similar, las columnas `tipo_contrato`, `procedimiento`, y `tramitacion` están relacionadas con las tablas TipoContrato, TipoProcedimiento, y TipoTramitacion, respectivamente. Estas claves externas garantizan que los valores en estas columnas correspondan a valores válidos en las tablas relacionadas, evitando datos huérfanos y asegurando la coherencia de los datos entre las tablas.

2. **Participaciones**

```
USE TFG;

-- Crear la tabla Participaciones
CREATE TABLE Participaciones (
    id_participacion INT IDENTITY(1,1) PRIMARY KEY,
    id_licitacion INT NOT NULL,
    id_empresa INT NOT NULL,
    importe_ofertado_sin_iva MONEY NULL,
    importe_ofertado_con_iva MONEY NULL,
    excluida BIT NULL,
    anormalidad_economica BIT NULL
);

-- Agregar clave foranea hacia la tabla Empresas
ALTER TABLE Participaciones
ADD CONSTRAINT FK_Participaciones_Empresas
FOREIGN KEY (id_empresa) REFERENCES Empresas(id_empresa)
ON DELETE CASCADE;

-- Agregar clave foranea hacia la tabla Licitaciones
ALTER TABLE Participaciones
ADD CONSTRAINT FK_Participaciones_Licitaciones
FOREIGN KEY (id_licitacion) REFERENCES Licitaciones(id_licitacion)
ON DELETE CASCADE
ON UPDATE CASCADE;
```

El script agrega restricciones de clave externa para asegurar la integridad referencial. La columna `id_empresa` debe coincidir con un valor en `id_empresa` de la tabla `Empresas`. De manera similar, `id_licitacion` debe coincidir con un valor en `id_licitacion` de la tabla `Licitaciones`. La restricción `ON DELETE CASCADE` asegura que si una empresa o licitación se elimina, también se eliminarán las participaciones asociadas.

3. **Criterios**

```
USE TFG;

-- Crear la tabla Criterios
CREATE TABLE Criterios (
    id_criterio INT IDENTITY(1,1) PRIMARY KEY,
    nombre VARCHAR(200) NOT NULL,
```

```
    siglas VARCHAR(50) NULL,
    valor_max FLOAT NULL,
    valor_min FLOAT NULL,
    id_padre INT NULL
);

-- Agregar la restriccion de clave foranea para la relacion jerarquica
ALTER TABLE Criterios
ADD CONSTRAINT FK_Criterios_Criterios
FOREIGN KEY (id_padre) REFERENCES Criterios(id_criterio);
```

Este script crea la tabla de Criterios con sus columnas y añade una restricción consigo misma. Esta restricción asegura que si se proporciona un valor para `id_padre`, este valor debe coincidir con un `id_criterio` existente en la misma tabla. Esto permite la creación de una relación jerárquica entre criterios. Si se elimina un criterio que tiene hijos, esos hijos también se eliminarán, manteniendo la integridad referencial.

Esto significa que, si un criterio tiene un `id_padre` diferente de `NULL`, entonces es un subcriterio del criterio al que apunta ese identificador.

4. **Valoraciones**

```
USE TFG;

-- Crear la tabla Valoraciones
CREATE TABLE Valoraciones (
    id_valoracion INT IDENTITY(1,1) PRIMARY KEY,
    id_participacion INT NOT NULL,
    id_criterio INT NOT NULL,
    puntuacion FLOAT NULL
);

-- Agregar las restricciones de clave foranea
ALTER TABLE Valoraciones
ADD CONSTRAINT FK_Valoraciones_Criterios
FOREIGN KEY (id_criterio) REFERENCES Criterios(id_criterio)
ON DELETE CASCADE;

ALTER TABLE Valoraciones
ADD CONSTRAINT FK_Valoraciones_Participaciones
FOREIGN KEY (id_participacion) REFERENCES Participaciones(
    id_participacion)
ON DELETE CASCADE;
```

En este script se crea la tabla `Valoraciones`. La tabla incluye las columnas `id_participacion` e `id_criterio`, que son enteros no nulos. Estas columnas establecen vínculos con las tablas `Participaciones` y `Criterios`, respectivamente. La columna `puntuacion` es de tipo `FLOAT` y almacena los puntos dados a la participación según el criterio.

La columna `id_criterio` debe coincidir con un valor en `id_criterio` de la tabla `Criterios`, y `id_participacion` debe coincidir con un valor en `id_participacion` de la tabla `Participaciones`. Ambas restricciones utilizan `ON DELETE CASCADE`, lo que significa que si un criterio o una participación se elimina, las valoraciones asociadas también se eliminarán.

6.3 Aplicación Web

En este apartado se detalla el proceso de desarrollo de la interfaz web de la aplicación SmarTenderAI, que integra un frontend desarrollado con React y un backend basado

en Django. El objetivo es ofrecer una plataforma eficiente y accesible para la consulta y visualización de las licitaciones almacenadas en la base de datos. A continuación, se describe la configuración de las tecnologías utilizadas y la implementación de los modelos en Django, que reflejan la estructura de la base de datos previamente definida, así como los componentes principales de la interfaz y su funcionamiento.

6.3.1. Instalación y Configuración Inicial

Antes de iniciar la configuración e instalación, se ha creado un directorio principal que contendrá tanto el frontend como el backend, centralizando así todos los componentes de la aplicación en una única ubicación. Este enfoque facilita la organización y gestión del proyecto, asegurando que todos los elementos necesarios se encuentren en un solo lugar.

1. Frontend: React con Vite

Para el desarrollo del frontend, se optó por utilizar React debido a su flexibilidad y amplia adopción en el desarrollo de aplicaciones web modernas. Para optimizar la velocidad de desarrollo y la eficiencia del proyecto, se empleó Vite como herramienta de construcción, en lugar de la tradicional Create React App, debido a su capacidad de ofrecer un entorno de desarrollo más rápido y ligero.

El proceso de configuración se realizó de la siguiente manera:

1. **Inicialización del Proyecto con Vite**:

   Se comenzó creando un nuevo proyecto React utilizando Vite. Esto se logró mediante el siguiente comando en la terminal:

```
npm create vite@latest react-frontent-app --template react
```

   Este comando genera una estructura de proyecto preconfigurada para React, optimizada para un arranque rápido y un tiempo de desarrollo reducido.

2. **Instalación de Dependencias Necesarias**:

   Una vez inicializado el proyecto, se procedió a instalar las dependencias adicionales necesarias para el desarrollo de la aplicación, tales como react-router-dom para la gestión de rutas, axios para la comunicación con el backend y Bootstrap, un popular framework CSS que facilita la creación de interfaces atractivas y responsivas.

2. Backend: Django

El backend de la aplicación se desarrolló utilizando Django, un framework de desarrollo web que facilita la construcción de aplicaciones web robustas y escalables. Django fue elegido por su capacidad para manejar aplicaciones complejas y su excelente integración con bases de datos relacionales, además de la facilidad para integrar los scripts de python que se han desarrollado para la extracción de la información.

El proceso de configuración y desarrollo del backend incluyó los siguientes pasos:

1. **Instalación de Django**:

   Para comenzar con el desarrollo del backend, primero se deben instalar Django y Django REST Framework, que son esenciales para el desarrollo del backend y la creación de APIs:

```
pip install django djangorestframework
```

2. **Creación del Proyecto Django**:

   Con Django instalado, se puede crear un nuevo proyecto y una aplicación dentro de él. El proyecto Django actúa como el contenedor principal de la configuración, mientras que la aplicación es un módulo donde se desarrollan las funcionalidades específicas.

```
django-admin startproject backend
cd backend
python manage.py startapp api
```

3. **Configuración de la Base de Datos**:

   En el archivo `settings.py` del proyecto, es necesario configurar la base de datos para que Django pueda interactuar correctamente con SQL Server.

```
DATABASES = {
    'default': {
        'ENGINE': 'mssql', # Microsoft Server SQL
        'NAME': 'TFG',
        'HOST': 'ASUSMARTA\TFG',
        'PORT': '',  #default port
        'OPTIONS': {
            'driver': 'ODBC Driver 17 for SQL Server',  # driver
        },
    },
}
```

   Este fragmento configura la conexión a una base de datos SQL Server llamada 'TFG' en el servidor 'ASUSMARTA\TFG', donde `ASUSMARTA` es el nombre del equipo (host), y `TFG` es el nombre de la instancia de SQL Server, y usando el controlador ODBC 17. Si no se especifica el puerto, Django utilizará el puerto predeterminado.

4. **Creación de los Modelos en Django para la Base de Datos**

   Cuando se trabaja con una base de datos existente que ya tiene tablas definidas, Django ofrece una herramienta llamada `inspectdb`, que facilita la integración de esta base de datos en un nuevo proyecto Django. Esta herramienta genera automáticamente los modelos de Django basados en las tablas de la base de datos existente.

   El comando `inspectdb` se utiliza para inspeccionar la estructura de la base de datos y generar un archivo `models.py` que contiene los modelos correspondientes. Este archivo se genera a partir de las tablas presentes en la base de datos.

```
    python manage.py inspectdb > api/models.py
```

5. **Creación de Vistas (Views) y Serializadores (Serializers)**

   Creación de Vistas:

   Con los modelos definidos, las vistas se crean utilizando Django REST Framework (DRF). Las vistas son responsables de manejar la lógica de las solicitudes HTTP y proporcionar respuestas adecuadas en formato JSON, que pueden ser consumidas por el frontend de React. DRF proporciona varias clases de vistas, como `ModelViewSet`, que facilita la implementación de las operaciones CRUD (Crear, Leer, Actualizar, Eliminar).

   Ejemplo de una vista basada en un modelo:

```
from rest_framework import viewsets
from .models import Licitaciones
from .serializers import LicitacionesSerializer

class LicitacionesViewSet(viewsets.ModelViewSet):
    queryset = Licitaciones.objects.all()
    serializer_class = LicitacionesSerializer
```

En este ejemplo, `LicitacionesViewSet` es una clase de vista que maneja todas las operaciones CRUD para el modelo `Licitaciones`. Utiliza un `viewset`, que es una clase que agrupa toda la funcionalidad CRUD para un modelo en una sola clase.

Creación de Serializadores:

Los serializadores (serializers) son componentes clave en DRF, ya que se encargan de convertir los objetos de los modelos de Django a formatos como JSON, que pueden ser fácilmente enviados al frontend. También permiten convertir datos JSON recibidos en instancias de los modelos, facilitando la creación y actualización de registros en la base de datos.

Ejemplo de un serializador para `Licitaciones`:

```
class LicitacionesSerializer(serializers.ModelSerializer):
    procedimiento = TipoprocedimientoSerializer(read_only=True)
    tramitacion = TipotramitacionSerializer(read_only=True)
    tipo_contrato = TipocontratoSerializer(read_only=True)
    adjudicatario= EmpresasSerializer(read_only=True)
    class Meta:
        model = Licitaciones
        fields = '__all__'
```

Dentro del `LicitacionesSerializer`, hay varios campos que no son simples atributos del modelo `Licitaciones`, sino relaciones con otros modelos. Para manejar estas relaciones, se utilizan serializadores anidados:

- `procedimiento = TipoprocedimientoSerializer(read_only=True)`: Este campo representa una relación entre `Licitaciones` y el modelo `Tipoprocedimiento`. Al usar `TipoprocedimientoSerializer`, se asegura que la información relacionada con `procedimiento` se incluya en la serialización de `Licitaciones`. La opción `read_only=True` indica que este campo es de solo lectura y no se puede modificar directamente a través de este serializador.
- `tramitacion = TipotramitacionSerializer(read_only=True)`: Similar al anterior, este campo gestiona la relación con el modelo `Tipotramitacion`.
- `tipo_contrato = TipocontratoSerializer(read_only=True)`: Este campo gestiona la relación con el modelo `Tipocontrato`.
- `adjudicatario = EmpresasSerializer(read_only=True)`: Este campo gestiona la relación con el modelo `Empresas`, que representa la empresa adjudicataria de la licitación.

Configuración de URLs:

Para que las vistas sean accesibles como endpoints de la API, es necesario configurar las URLs en el archivo `urls.py` de la aplicación. Esto se hace utilizando un enrutador (`router`) que registra las vistas y las asocia con rutas específicas.

```
from django.urls import path, include
from rest_framework.routers import DefaultRouter
from . import views
```

```
4
5 router = DefaultRouter()
6 router.register(r'licitaciones', views.LicitacionesViewSet)
7 # se registraria un router por cada vista
8
9 urlpatterns = [
10     path('', include(router.urls)),
11 ]
```

En el archivo `urls.py` del proyecto debemos incluir las rutas definidas en el archivo `urls.py` de la aplicación. Esto se hace utilizando la función `include` de Django.

```
1 from django.contrib import admin
2 from django.urls import path, include
3
4 urlpatterns = [
5     path("admin/", admin.site.urls),
6     path('api/', include('api.urls')),
7 ]
```

3. Integración del Frontend y Backend

El frontend de React se comunica con el backend de Django a través de las APIs REST creadas con Django REST Framework. La biblioteca axios se utiliza en React para realizar peticiones HTTP a estos endpoints.

6.3.2. Estructura y Fucionalidad de la Aplicación

Una vez configurados el frontend y el backend, y asegurada la correcta comunicación entre ambos, se puede proceder al desarrollo de la interfaz de usuario y a la implementación de las funcionalidades clave de la aplicación.

Cabe destacar que para garantizar una consistencia visual y una experiencia de usuario coherente, se ha utilizado el framework Bootstrap en el desarrollo del frontend de la aplicación.

La aplicación cuenta con dos pestañas principales, una dedicada a **Licitaciones** y otra a **Empresas**, cada una diseñada para facilitar la visualización y análisis de la información relevante. A continuación, se describen en detalle las características y funcionalidades clave de cada sección, así como algunas consideraciones importantes sobre el rendimiento y posibles problemas que pueden surgir.

Pestañas Principales: Licitaciones y Empresas

Ambas pestañas permiten al usuario acceder a una lista completa de licitaciones (ver Figura 6.1) o empresas (ver Figura 6.2), así como a una serie de estadísticas relacionadas con ellas. Estas estadísticas proporcionan un resumen visual y numérico de los datos, facilitando el análisis y la toma de decisiones.

- **Buscador:** Las listas de licitaciones y empresas incluyen un buscador que permite localizar rápidamente nombres específicos o palabras clave. Por ejemplo, sería útil buscar en las licitaciones la palabra ‘Construcción‘ para visualizar aquellas licitaciones que pretenden contratar con el objetivo de llevar a cabo diferentes tipos de construcciones.

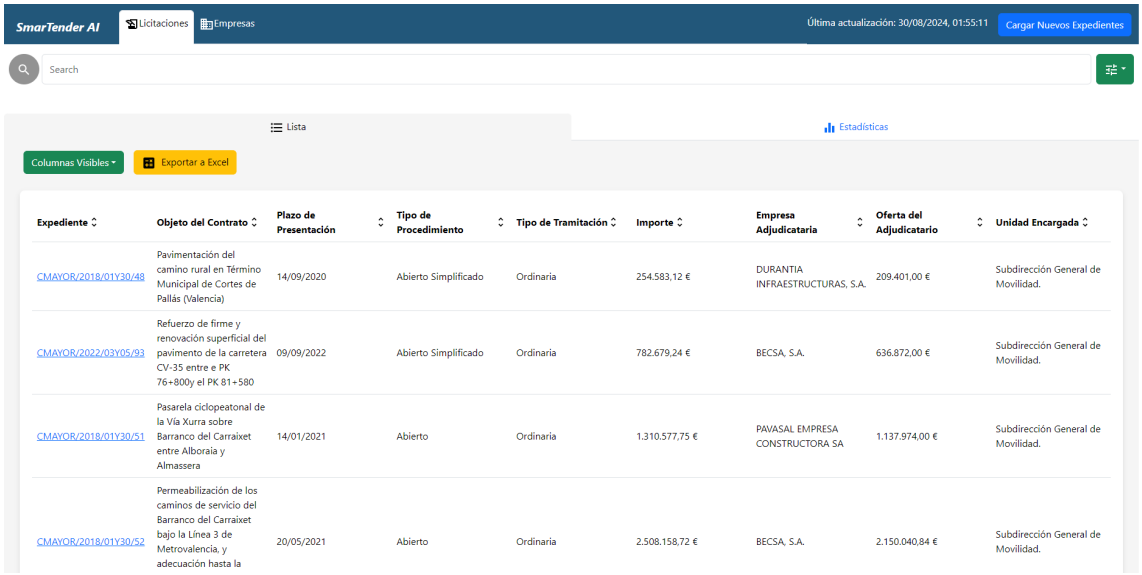


Figura 6.1: Tabla de Licitaciones

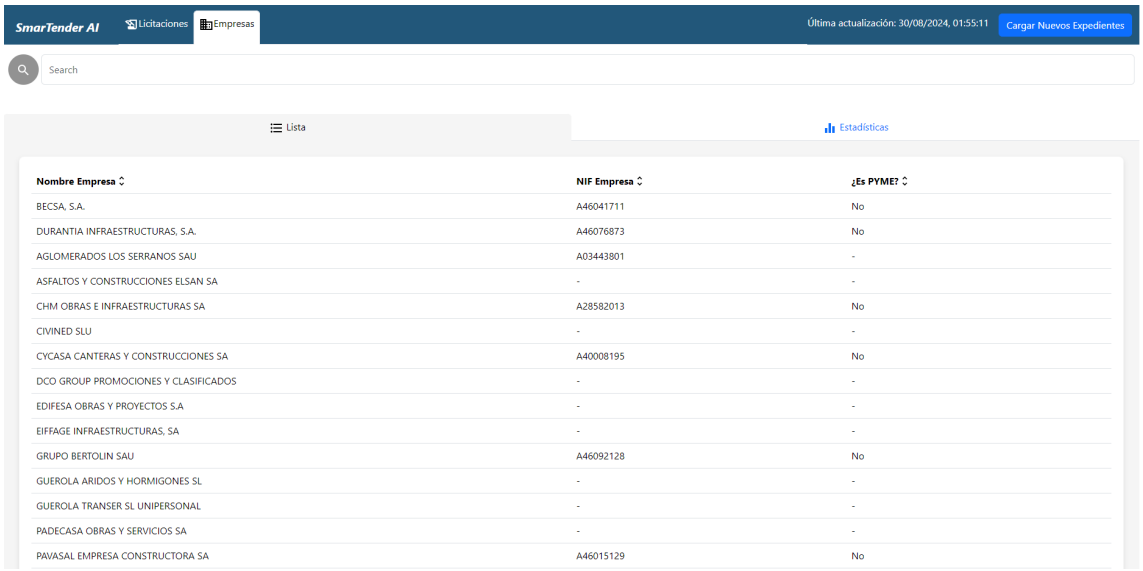


Figura 6.2: Tabla de Empresas

- **Menú de Filtrado (solo en Licitaciones):** En la pestaña de licitaciones, los usuarios pueden aplicar diversos filtros para refinar la lista según criterios específicos, como el tipo de procedimiento o la fecha. Estos filtros también afectan las estadísticas, que se actualizan dinámicamente para reflejar solo las licitaciones que cumplen con los criterios seleccionados. Se puede ver este menú de filtrado en la Figura 6.3.

- **Ordenación por Columnas:** Tanto en la pestaña de licitaciones como en la de empresas, las tablas se pueden ordenar por cualquier columna. Esto permite a los usuarios reorganizar la información según sus necesidades, por ejemplo, ordenando las licitaciones por fecha, importe, o cualquier otro atributo relevante.

- **Selección de Columnas (Licitaciones):** En la lista de licitaciones, se ofrece la posibilidad de seleccionar u omitir columnas, permitiendo a los usuarios personalizar la vista según sus necesidades. Esto facilita la concentración en la información más relevante para cada análisis. Esta función se puede ver en la Figura 6.4.

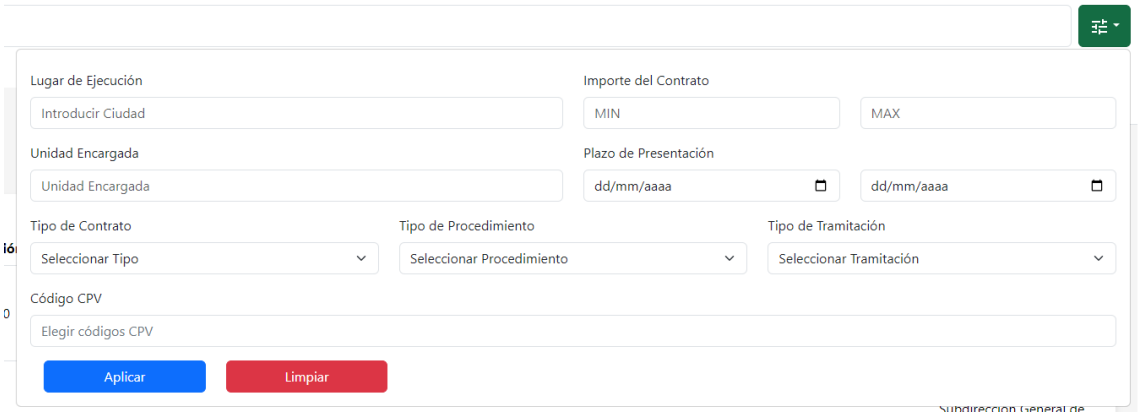


Figura 6.3: Menú de Filtros

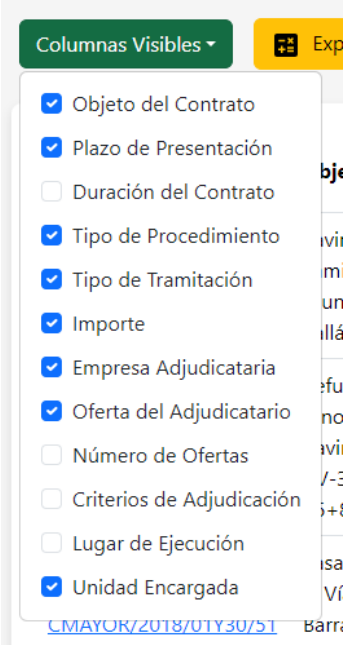


Figura 6.4: Selección de Columnas Visibles

Estadísticas de Licitaciones

- **Tipos de Contrato, Procedimiento y Tramitación:** Las estadísticas de licitaciones incluyen datos agrupados por tipo de contrato, tipo de procedimiento, y tipo de tramitación. Esto permite a los usuarios identificar patrones y tendencias en las diferentes categorías de licitaciones (ver Figura 6.5).

- **Análisis de Criterios de Adjudicación:** Se proporciona un desglose estadístico que indica si las licitaciones fueron adjudicadas basándose en uno o varios criterios, facilitando así el análisis de la complejidad y competencia de cada licitación (ver Figura 6.5).

- **Gráfico de Barras de Empresas:** Se incluye un gráfico de barras que muestra el número de adjudicaciones, participaciones y expulsiones por empresa. Este gráfico es interactivo, permitiendo ordenar los datos de mayor a menor según cualquiera de estos indicadores, lo que facilita la identificación de las empresas más activas y exitosas en las licitaciones. Debido al gran número de empresas que hay, solo se incluyen las 20 primeras (ver Figura 6.6).

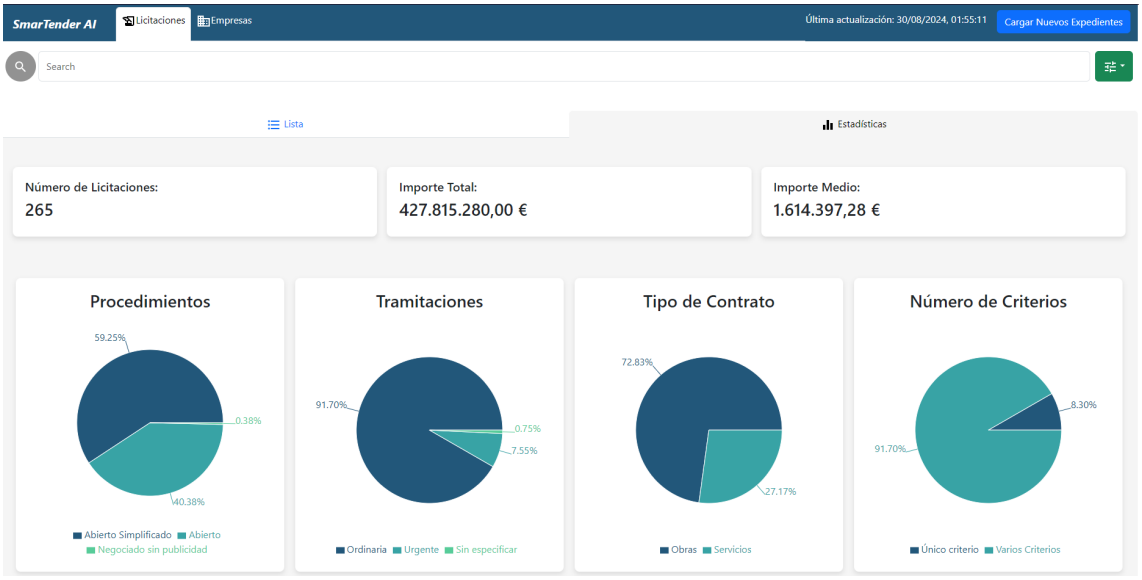


Figura 6.5: Estadísticas de Licitaciones

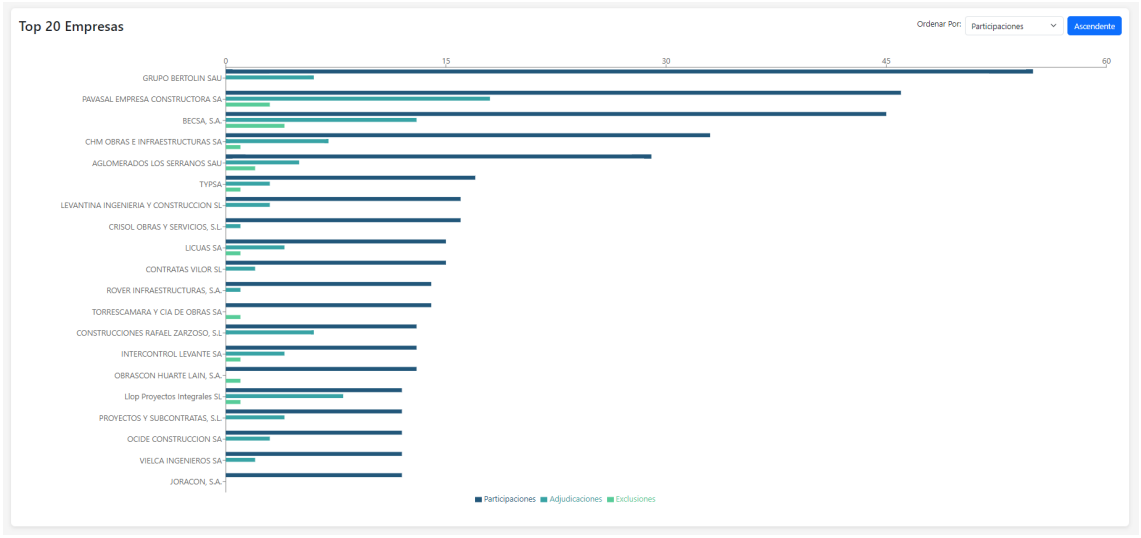


Figura 6.6: Tabla con Top 20 Empresas

- **Porcentajes Medios de Baja y Número Medio de Licitadores:** Las estadísticas también incluyen el porcentaje medio de baja de los adjudicatarios y el número medio de licitadores agrupados por tamaño de contrato (definido por rangos de importe). La **baja** de cada adjudicatario se calcula como:

$$\text{Baja} = \frac{\text{Presupuesto base de licitación (sin IVA)} - \text{Oferta del adjudicatario (sin IVA)}}{\text{Presupuesto base de licitación (sin IVA)}}$$

  Este cálculo proporciona una medida del descuento ofrecido por los adjudicatarios, lo que es crucial para analizar la competitividad en las ofertas. Ambas gráficas se pueden ver en la Figura 6.7

Estadísticas de Empresas

- **Gráfico de Tarta sobre el Tamaño de las Empresas:** En la pestaña de empresas, se incluye un gráfico de tarta que muestra la proporción de empresas que son PYME,

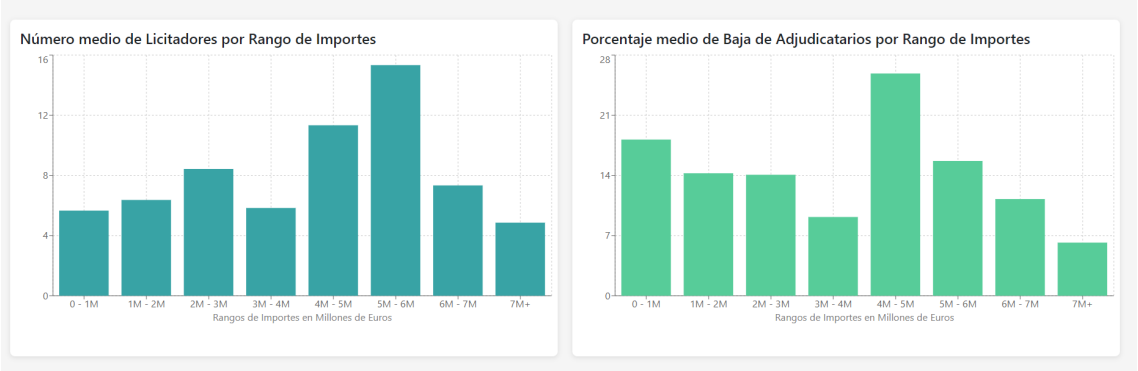


Figura 6.7: Gráficas de Estadísticas por Rangos de Importe

no PYME, o aquellas de las que no se tiene información suficiente. Este gráfico ofrece una visión rápida de la distribución del tamaño de las empresas participantes en las licitaciones (ver Figura 6.8).

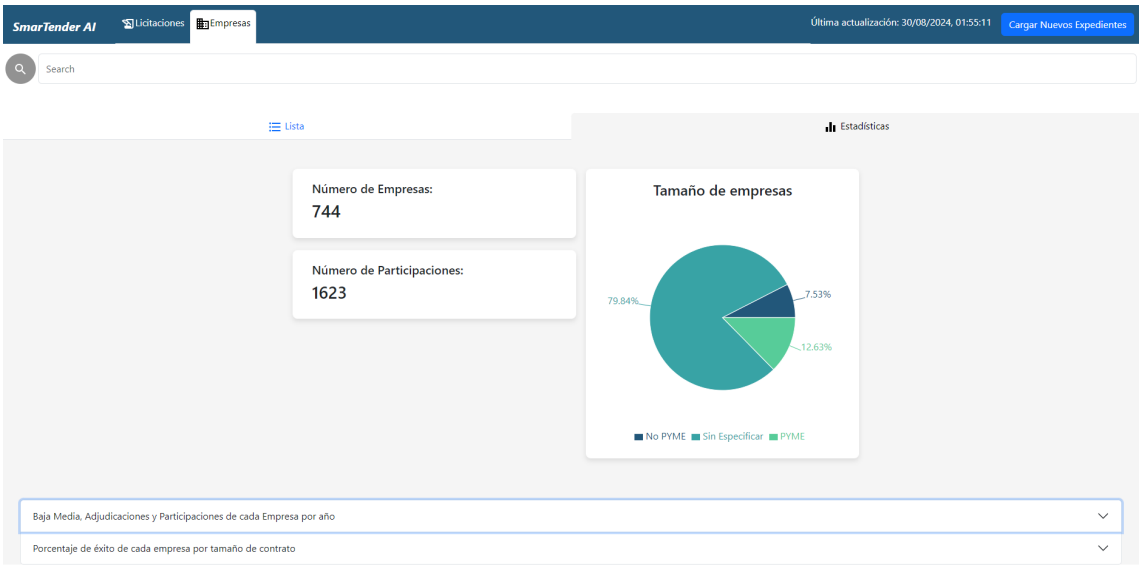
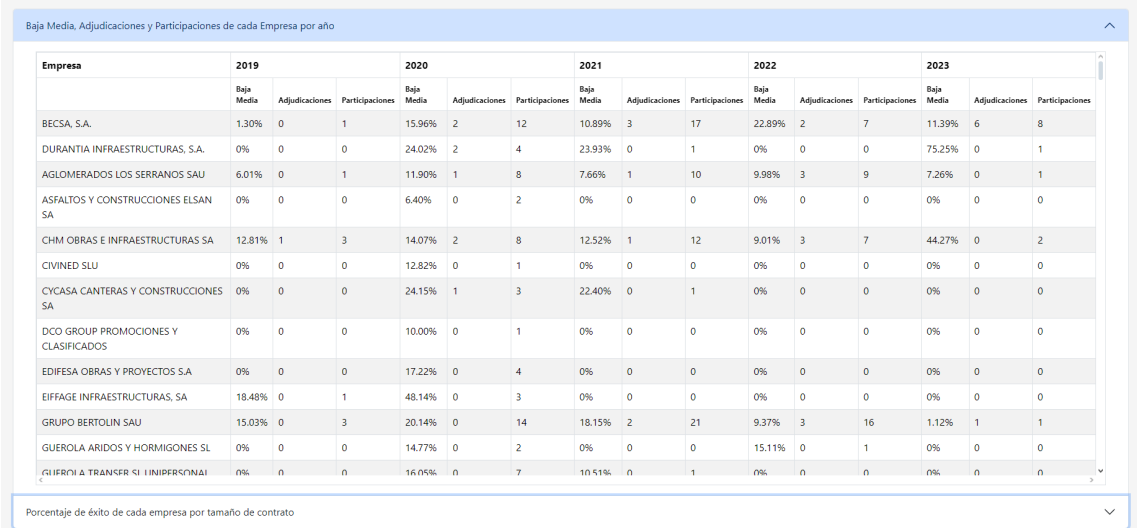


Figura 6.8: Estadísticas de empresas

- **Tablas Detalladas de Rendimiento de las Empresas:**
  - **Baja Media por Año:** Esta tabla (ver Figura 6.9) muestra la baja media de cada empresa, desglosada por año, así como la baja total acumulada. Esto permite analizar cómo han variado las ofertas de las empresas a lo largo del tiempo.
  - **Porcentaje de Éxito por Tamaño de Contrato:** Esta tabla detalla el porcentaje de éxito de cada empresa (ver Figura 6.10), calculado como el número de adjudicaciones dividido entre el número de participaciones, agrupado por tamaño de contrato. Este análisis es crucial para entender el rendimiento de las empresas en función del tamaño de las licitaciones en las que participan.

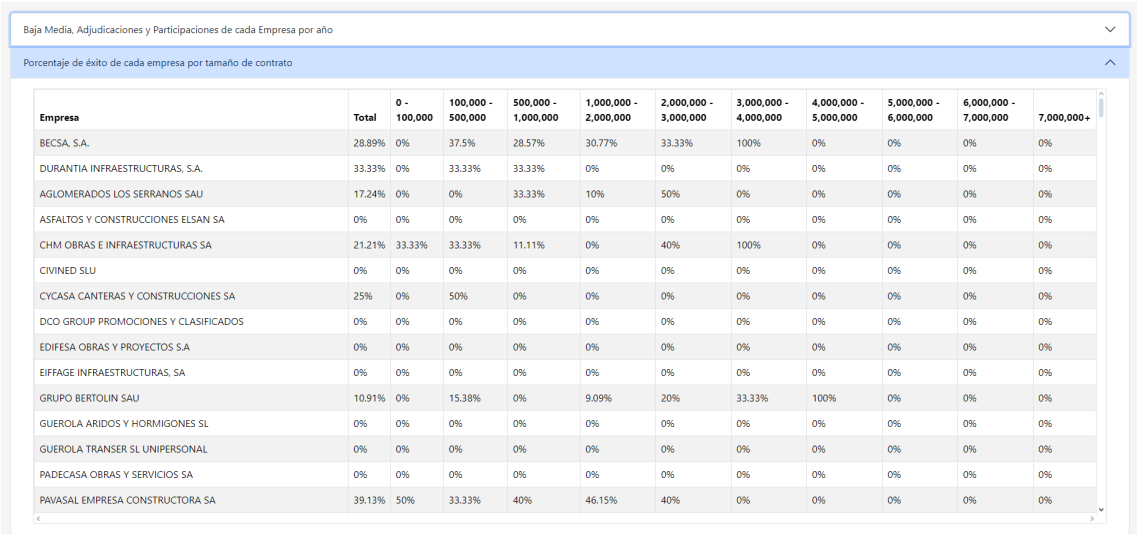
Visualización Detallada de Licitaciones

- **Detalles de Licitaciones:** Al seleccionar una licitación específica desde la lista, se accede a una vista detallada que muestra diferentes apartados organizados para



Empresa	2019			2020			2021			2022			2023		
	Baja Media	Adjudicaciones	Participaciones	Baja Media	Adjudicaciones	Participaciones	Baja Media	Adjudicaciones	Participaciones	Baja Media	Adjudicaciones	Participaciones	Baja Media	Adjudicaciones	Participaciones
BECSA, S.A.	1.30%	0	1	15.96%	2	12	10.89%	3	17	22.89%	2	7	11.39%	6	8
DURANTIA INFRAESTRUCTURAS, S.A.	0%	0	0	24.02%	2	4	23.93%	0	1	0%	0	0	75.25%	0	1
AGLOMERADOS LOS SERRANOS SAU	6.01%	0	1	11.90%	1	8	7.66%	1	10	9.98%	3	9	7.26%	0	1
ASFALTOS Y CONSTRUCCIONES ELSAN SA	0%	0	0	6.40%	0	2	0%	0	0	0%	0	0	0%	0	0
CHM OBRAS E INFRAESTRUCTURAS SA	12.81%	1	3	14.07%	2	8	12.52%	1	12	9.01%	3	7	44.27%	0	2
CIVINED SLU	0%	0	0	12.82%	0	1	0%	0	0	0%	0	0	0%	0	0
CYCASA CANTERAS Y CONSTRUCCIONES SA	0%	0	0	24.15%	1	3	22.40%	0	1	0%	0	0	0%	0	0
DCO GROUP PROMOCIONES Y CLASIFICADOS	0%	0	0	10.00%	0	1	0%	0	0	0%	0	0	0%	0	0
EDIFESA OBRAS Y PROYECTOS S.A	0%	0	0	17.22%	0	4	0%	0	0	0%	0	0	0%	0	0
EIFFAGE INFRAESTRUCTURAS, SA	18.48%	0	1	48.14%	0	3	0%	0	0	0%	0	0	0%	0	0
GRUPO BERTOLIN SAU	15.03%	0	3	20.14%	0	14	18.15%	2	21	9.37%	3	16	1.12%	1	1
GUEROLA ARIDOS Y HORMIGONES SL	0%	0	0	14.77%	0	2	0%	0	0	15.11%	0	1	0%	0	0
GUEROLA TRANSER SL UNIPERSONAL	0%	0	0	16.05%	0	7	10.51%	0	1	0%	0	0	0%	0	0

Figura 6.9: Tabla de Baja Media por Año de cada Empresa



Empresa	Total	0 - 100,000	100,000 - 500,000	500,000 - 1,000,000	1,000,000 - 2,000,000	2,000,000 - 3,000,000	3,000,000 - 4,000,000	4,000,000 - 5,000,000	5,000,000 - 6,000,000	6,000,000 - 7,000,000	7,000,000+
BECSA, S.A.	28.89%	0%	37.5%	28.57%	30.77%	33.33%	100%	0%	0%	0%	0%
DURANTIA INFRAESTRUCTURAS, S.A.	33.33%	0%	33.33%	33.33%	0%	0%	0%	0%	0%	0%	0%
AGLOMERADOS LOS SERRANOS SAU	17.24%	0%	0%	33.33%	10%	50%	0%	0%	0%	0%	0%
ASFALTOS Y CONSTRUCCIONES ELSAN SA	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
CHM OBRAS E INFRAESTRUCTURAS SA	21.21%	33.33%	33.33%	11.11%	0%	40%	100%	0%	0%	0%	0%
CIVINED SLU	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
CYCASA CANTERAS Y CONSTRUCCIONES SA	25%	0%	50%	0%	0%	0%	0%	0%	0%	0%	0%
DCO GROUP PROMOCIONES Y CLASIFICADOS	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
EDIFESA OBRAS Y PROYECTOS S.A	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
EIFFAGE INFRAESTRUCTURAS, SA	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
GRUPO BERTOLIN SAU	10.91%	0%	15.38%	0%	9.09%	20%	33.33%	100%	0%	0%	0%
GUEROLA ARIDOS Y HORMIGONES SL	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
GUEROLA TRANSER SL UNIPERSONAL	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
PADECASA OBRAS Y SERVICIOS SA	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
PAVASAL EMPRESA CONSTRUCTORA SA	39.13%	50%	33.33%	40%	46.15%	40%	0%	0%	0%	0%	0%

Figura 6.10: Tabla del Porcentaje de Éxito de cada empresa por Tamaño de Contrato

facilitar la navegación y búsqueda de información. Esto incluye detalles sobre los procedimientos, las empresas participantes, y otros atributos clave de la licitación. Se puede ver alguna de esta información en la Figura 6.11

- **Criterios de Adjudicación y Estadísticas:** Esta sección muestra una lista de criterios de adjudicación asociados a la licitación, acompañada de estadísticas como media, mediana, y desviación típica. Se puede ver en la Figura 6.12.

  Estas estadísticas se calculan en el backend y se accede a ellas mediante una petición a la API desde el frontend con el identificador de la licitación actual.

  Además, se presenta una tabla comparativa que muestra la evaluación de cada empresa en relación con cada criterio y la oferta económica, permitiendo un análisis profundo de cómo se ha valorado a cada participante.

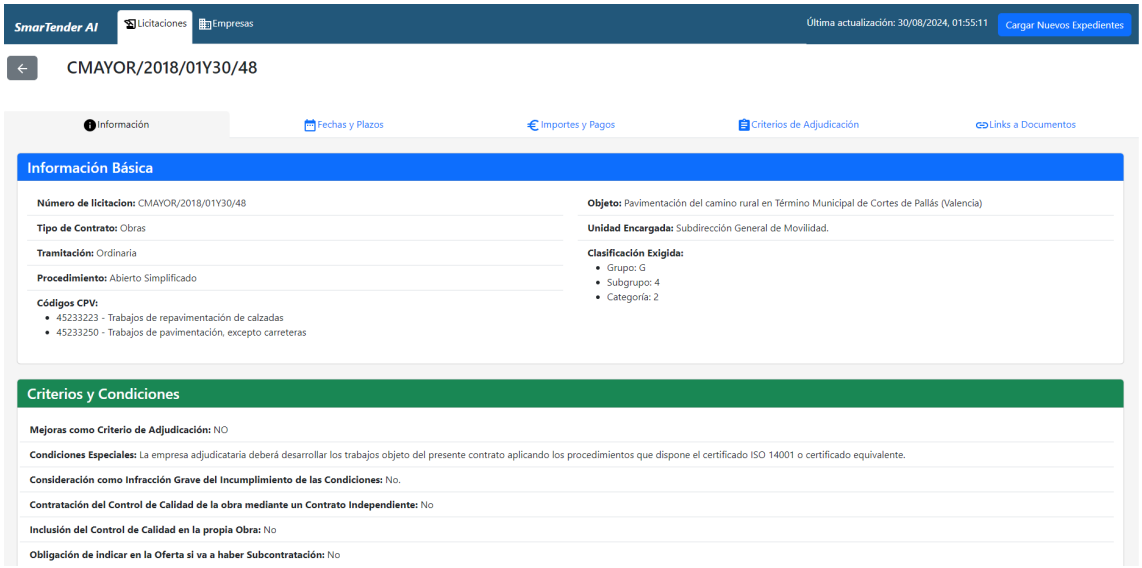


Figura 6.11: Información Detallada de una Licitación

3. Funcionalidades Clave Adicionales

- **Exportación a Excel:** La aplicación permite exportar a Excel las licitaciones filtradas, incluyendo todos los criterios, valoraciones y atributos relevantes. Este Excel contiene una pestaña inicial que muestra la fecha de la última actualización de la base de datos, así como los filtros aplicados en la exportación, lo que asegura que la información exportada sea precisa y relevante para el momento en que fue generada.

  El archivo generado consta de varias hojas, cada una con información específica:

  - Una hoja inicial con los filtros aplicados y la última fecha de actualización de la base de datos.
  - Una hoja con el listado de licitaciones y sus atributos, incluyendo la información del adjudicatario. Esta hoja proporciona un detalle exhaustivo de cada licitación.
  - Una hoja con el listado de las empresas junto al número de adjudicaciones, participaciones y porcentaje de éxito de cada una. Esto permite analizar el desempeño de las empresas en las licitaciones.
  - Una hoja con el listado de criterios que se incluyen en las licitaciones filtradas.
  - Una hoja con la oferta económica de cada empresa para cada licitación si participan en ella. Esta hoja detalla las ofertas presentadas, permitiendo comparar las propuestas económicas entre las empresas.
  - Una hoja para cada criterio asociado a las licitaciones, donde se incluye la valoración de cada empresa en cada licitación únicamente a ese criterio. Esto proporciona un análisis detallado de cómo cada empresa fue evaluada en función de los distintos criterios.

  La exportación a Excel se gestiona a través de una vista en el backend de la aplicación. Esta vista se activa al presionar el botón correspondiente en la interfaz de usuario. La vista maneja la lógica necesaria para recopilar los datos filtrados y generar el archivo Excel.

  Para crear el archivo Excel y gestionar sus hojas y datos, se utiliza la librería de Python `openpyxl`. Esta librería permite crear un nuevo cuaderno Excel y añadir las

Listado Criterios

Criterio	Peso	Calidad inaceptable	Valoración Máxima	Valoración Mínima	Media	Mediana	Desviación Típica	Diferencia Primera y Segunda Posición
Criterio Precio	100		100	83.93	92.86	92.69	4.42	1.34

Valoraciones de las empresas

Empresa	Oferta Económica	Criterio Precio
DURANTIA INFRAESTRUCTURAS, S.A.	209.401,00 €	100
AGLOMERADOS LOS SERRANOS SAU	238.646,22 €	87.75
ASFALTOS Y CONSTRUCCIONES ELSAN SA	249.491,46 €	83.93
BECSA, S.A.	206.716,00 €	
CHM OBRAS E INFRAESTRUCTURAS SA	242.694,09 €	86.28
CIVINED SLU	221.945,56 €	94.35
CYCASA CANTERAS Y CONSTRUCCIONES SA	214.919,07 €	97.43
DCO GROUP PROMOCIONES Y CLASIFICADOS	229.124,81 €	91.39
EDIFESA OBRAS Y PROYECTOS S.A	220.698,00 €	94.88
EIFFAGE INFRAESTRUCTURAS, SA	225.917,06 €	92.69
GRUPO BERTOLIN SAU	220.672,65 €	94.89
GUEROLA ARIDOS Y HORMIGONES SL	226.655,35 €	92.39
GUEROLA TRANSER SL UNIPERSONAL	229.710,35 €	91.16
PADECASA OBRAS Y SERVICIOS SA	212.245,95 €	98.66
PAVASAL EMPRESA CONSTRUCTORA SA	216.091,00 €	96.9
SERRANO AZNAR OBRAS PUBLICAS SLU	231.976,14 €	90.27

Figura 6.12: Listado de Criterios y Valoraciones de una Licitación

hojas y datos correspondientes. Se crean diferentes hojas dentro del cuaderno, cada una con sus correspondientes columnas y datos extraídos de la base de datos. Esto asegura que la información esté organizada de manera clara y accesible.

Es importante tener en cuenta que, al exportar un gran número de licitaciones, el proceso puede tardar un tiempo considerable debido al volumen de datos. La generación del archivo Excel puede verse afectada por la cantidad de información incluida, por lo que se recomienda ser consciente del tamaño de los datos seleccionados para evitar demoras excesivas en la descarga.

- **Actualización de la Base de Datos:** Un botón en la interfaz permite ejecutar los scripts de extracción que actualizan la base de datos con nueva información. Esta funcionalidad está vinculada a una vista que se encarga de ejecutar los scripts cuando el botón es presionado. Además, la interfaz muestra la última fecha de actualización, para que los usuarios sepan cuándo se realizó la última sincronización de datos.

  Se recomienda realizar actualizaciones periódicas para mantener la base de datos al día con los expedientes faltantes. Dado el extenso número de licitaciones, es importante planificar estas actualizaciones con anticipación para asegurar que la base de datos se mantenga completa y actualizada. Además, dependiendo del volumen de datos a extraer, el proceso de actualización puede llevar un tiempo considerable. Por otro lado, destacar que cada actualización de la base de datos con nuevos expedientes tiene un coste asociado al uso de la API de OpenAI, por lo que se debe de comprobar que hay saldo disponible antes de llevar a cabo una actualización.

CAPÍTULO 7
Pruebas

Una vez finalizado el desarrollo de la aplicación, se deben realizar una serie de pruebas de validación para garantizar su correcto funcionamiento. Para ello, se han desarrollado pruebas unitarias para los scripts del módulo de extracción, con el objetivo de asegurar su desempeño adecuado, incluso en casos de prueba incompletos, y verificar que manejan los errores sin causar la interrupción de su ejecución. Además de las pruebas unitarias, se llevaron a cabo pruebas de integración utilizando Postman, una herramienta ampliamente utilizada para probar y documentar API.

Además, se han calculado la precisión, el recall y el F1-score para diversas variables extraídas con el LLM, lo que proporciona una visión clara de la eficacia del modelo, con especial énfasis en la extracción de criterios y valoraciones de empresas.

7.1 Pruebas Unitarias

Las pruebas unitarias son un tipo de prueba de software que se centra en verificar el comportamiento de una única unidad de código, como una función o un método. Estas pruebas se ejecutan de manera independiente y aislada del resto del sistema para asegurar que el componente probado funcione correctamente en diferentes escenarios y con distintos conjuntos de datos. Las pruebas unitarias ayudan a garantizar que cada parte del código cumpla con las expectativas y permiten detectar errores y problemas antes de que el software se implemente en un entorno de producción.

Para llevar a cabo las pruebas unitarias en Python, se utilizó el módulo `unittest`, que proporciona una estructura para crear y ejecutar pruebas unitarias, organizar las pruebas en grupos, y generar informes detallados sobre los resultados de las pruebas.

Se han desarrollado una serie de pruebas unitarias para cada uno de los scripts del módulo de extracción, con el fin de verificar su funcionamiento en diversos escenarios. Estas pruebas se han diseñado para simular diferentes condiciones utilizando *mockups*, lo que permite revisar el comportamiento de las distintas partes del sistema de manera aislada. Los *mockups* se emplean para imitar objetos y comportamientos específicos que el código bajo prueba necesita para funcionar, facilitando así la validación de la lógica y el manejo de errores sin depender de componentes externos o de un entorno completo.

AccessPagePlaywright.py

1. **Interacción con la Base de Datos**: Se validó que las funciones encargadas de verificar la existencia de expedientes en la base de datos y manejar errores de conexión

operen correctamente. Esto incluye comprobar que los expedientes se detectan adecuadamente y que se gestionan adecuadamente los errores de base de datos.

2. **Extracción de Datos Web**: Se probó la funcionalidad de recolección de expedientes desde la página web, evaluando el correcto funcionamiento de la función que recoge los expedientes y maneja casos de elementos múltiples o ausencia de elementos, así como el caso en el que se tiene que visitar varias páginas de la tabla para obtener la totalidad de expedientes.

3. **Procesos Completos de Scraping**: Se verificó que el proceso completo de scraping, que incluye la recolección de expedientes y la obtención de contenido HTML, se ejecute sin errores y devuelva los resultados esperados.

4. **Manejo de Errores**: Se evaluó cómo el sistema maneja situaciones de error, como fallos al obtener el HTML de una página o errores en la conexión a la base de datos, asegurando que el sistema responda de manera adecuada a estas situaciones.

extraerAdjudicacion.py

1. **Extracción de Fecha**: Verifica que la función `extraer_fecha` identifique correctamente una fecha en el formato esperado dentro del HTML y la devuelva. Además, asegura que la función devuelva `None` cuando no se encuentra una fecha válida en el HTML.

2. **Extracción Completa de Información de Adjudicación**: Comprueba que la función `extraer_info_adjudicacion` pueda extraer toda la información relevante del HTML cuando todos los campos están presentes y son correctos.

3. **Extracción de Información Incompleta de Adjudicación**: Valida que la función `extraer_info_adjudicacion` devuelva `None` para campos que no están presentes en el HTML y que extraiga correctamente la información disponible.

4. **Extracción sin Fecha**: Verifica que en caso de que la fecha no se encuentre en el HTML, la función `extraer_info_adjudicacion` devuelva `None` para la fecha y extraiga correctamente el resto de la información.

5. **Formato de Fecha Inválido**: Asegura que la función `extraer_fecha` devuelva `None` si la fecha en el HTML está en un formato no esperado o inválido.

6. **Manejo de Información Extra**: Comprueba que la función `extraer_info_adjudicacion` ignore información no esperada o irrelevante y que el resultado extraído siga siendo el esperado para los campos relevantes.

extraerPliego.py

1. **Extracción de Información Completa**: Verifica que la función pueda extraer toda la información relevante de un pliego cuando el HTML está completo y bien estructurado. Esto incluye el expediente, objeto, plazo de ejecución, importe, fecha de presentación y otros datos esperados. También se prueba que el enlace al anexo se extrae correctamente y que los detalles del anexo se procesan adecuadamente.

2. **Manejo de HTML sin Anexo**: Comprueba que la función maneja correctamente los casos en los que no hay un anexo asociado. En esta prueba, el contenido del PDF no está disponible, y se verifica que la función devuelve correctamente la información extraída del HTML y que el enlace al anexo es `None`.

3. **Tratamiento de Lotes**: Verifica que la función maneja adecuadamente los casos en los que se especifican lotes en el HTML. En esta situación, la función debe devolver `None` para el resultado de la extracción, indicando que la función no maneja pliegos con lotes.

4. **HTML Incompleto**: Evalúa cómo la función responde cuando el HTML proporcionado está incompleto o falta información importante. Se verifica que la función aún puede extraer la información disponible y devolver `None` para los campos que no están presentes en el HTML.

5. **Información Inesperada**: Verifica que la función puede manejar HTML que contiene información inesperada o no estándar. La prueba asegura que la función ignora los datos no relevantes y solo extrae la información que se espera.

6. **Datos Duplicados**: Comprueba cómo la función maneja HTML con datos duplicados. Se verifica que la función extrae correctamente los datos sin ser afectada por duplicados, manteniendo la integridad de los datos extraídos.

procesarTextoPDF.py

1. **Extracción de Texto de Páginas :** Verificó que el texto se extrae y limpia correctamente, eliminando datos irrelevantes como encabezados y números de página.

2. **Procesamiento de Secciones:** Comprobó que la función segmenta correctamente el texto en secciones definidas y organiza el contenido de cada sección de manera adecuada.

3. **Formato de Texto Modificado:** Validó que las modificaciones de formato, como la sustitución de caracteres y normalizaciones, se aplican correctamente según las reglas establecidas.

4. **Datos Completos:** Se verificó que la función extrae y organiza todos los datos relevantes según las claves especificadas en la lista de datos.

5. **Datos Incompletos o Faltantes:** La función también se probó con PDFs que contenían secciones o datos faltantes. En estos casos, la función manejó adecuadamente la ausencia de datos, devolviendo `None` o valores predeterminados para los datos que no estaban presentes.

6. **Valores Previstos:** Finalmente, se validó la función `sacar_valores_previstos` para confirmar que extrae correctamente los valores previstos para conceptos como 'Modificaciones' y 'Prórrogas'.

extraerSeccionesLLM.py

1. **Extracción Correcta de Secciones**: Verifica que la función `extract_sections` extraiga correctamente las secciones del texto dado.

2. **Identificación de Secciones en PDF**: Comprueba que la función `read_pdf` pueda identificar correctamente las secciones del PDF simulado y que se maneje la página correcta para cada sección.

3. **Generación de Prompts para Criterios y Subcriterios**: Verifica que se genere correctamente el prompt para los criterios y subcriterios utilizando el formato esperado y que el modelo LLM responda adecuadamente.

4. **Manejo de Respuestas del Modelo LLM**:
   - **Respuestas Válidas**: Asegura que el modelo LLM pueda devolver respuestas válidas que coincidan con el formato esperado para criterios y subcriterios.
   - **Respuestas Vacías**: Verifica que las respuestas vacías del modelo LLM se manejen adecuadamente.
   - **Respuestas Malformadas**: Comprueba que las respuestas malformadas del modelo LLM sean manejadas de forma adecuada.
5. **Secciones Vacías**: Verifica que se manejen adecuadamente las secciones que están vacías o no contienen información relevante.
6. **PDF con Secciones Faltantes**: Comprueba que el sistema maneje correctamente un PDF en el que faltan algunas secciones esperadas.
7. **Conteo de Llamadas al LLM**: Asegura que el número de llamadas realizadas al modelo LLM coincida con el esperado para confirmar que cada sección relevante se procesa correctamente.

extraerTablas.py

1. **Lectura de PDF**: Verifica que la función `read_pdf` extraiga correctamente el texto y el número de páginas del PDF simulado.
2. **Extracción Correcta de Información de Tablas**: Verifica que la función `extract_table_info` extraiga la información de las tablas del PDF simulado y la transforme en un formato de diccionario adecuado.
3. **Manejo de Criterios Existentes**: Asegura que la función `extract_table_info` maneje correctamente los criterios que están presentes en el DataFrame simulado.
4. **Manejo de Criterios Faltantes**: Verifica que la función `extract_table_info` maneje la ausencia de criterios en el DataFrame simulado devolviendo `None` para criterios no presentes.
5. **Extracción de Ofertas Correctas**: Verifica que la función `extraer_ofertas` procese correctamente la información de las ofertas extraídas de un PDF simulado, devolviendo un diccionario con los resultados esperados.
6. **Extracción Correcta de Información del Acta**: Verifica que la función `extraer_info_acta` combine la información de las tablas y el texto del PDF simulado para devolver un diccionario con las claves esperadas, asegurando que la información sobre valoraciones y números de empresas sea correcta.
7. **Manejo de Criterios Extra**: Verifica que la función `extract_table_info` maneje correctamente los casos en los que hay criterios adicionales en el DataFrame que no están en la lista de criterios que se desean extraer, consiguiendo devolver únicamente la información relevante de los criterios que se solicitan.

main.py

1. **Obtención de Nombres**: Verifica que la función `get_names` devuelva una lista de nombres de criterios con las siglas formateadas correctamente cuando las siglas están presentes y cuando no.

2. **Apertura de Enlaces**: Verifica que la función `open_link` extraiga correctamente el enlace del contenido HTML simulado y devuelva el enlace junto al contenido de dicho enlace como bytes.

3. **Extracción Correcta de Documentos**: Verifica que la función `docs_valoraciones` identifique y extraiga los documentos de valoraciones del HTML simulado.

4. **Extracción Correcta de Documentos**: Verifica que la función `docs_juicio_valor` identifique y extraiga los documentos de juicio de valor del HTML simulado.

5. **Extracción de Fechas**: Verifica que la función `fechas_anuncio_form` extraiga correctamente las fechas del HTML simulado y devuelva un diccionario con las fechas esperadas.

6. **Acceso Correcto a Secciones**: Verifica que la función `acceder_seccion` acceda correctamente a una sección específica del HTML simulado y que actualice el diccionario `links` con el enlace correcto.

7. **Funcionamiento del Main**: Verifica el correcto funcionamiento de la función `main` asegurando que hace las llamadas necesarias a las funciones de los otros scripts.

7.2 Pruebas de Integración

Las pruebas de integración se centraron en asegurar que los diferentes componentes de la aplicación web interactúan correctamente a través de las APIs. El objetivo principal fue verificar que los endpoints de la API funcionan como se espera y que la comunicación entre el frontend y el backend es fluida y sin errores.

Para ello se ha definido una colección en Postman como podemos ver en la Figura 7.1 con cada una de las solicitudes a los diferentes endpoints del backend. Las pruebas incluyeron solicitudes de tipo GET para verificar la recuperación de los datos y POST para la ejecución del módulo de extracción puesto que carga nuevos registros en la base de datos. Una vez definidas las solicitudes con sus tipos correspondientes, se ejecutaron para

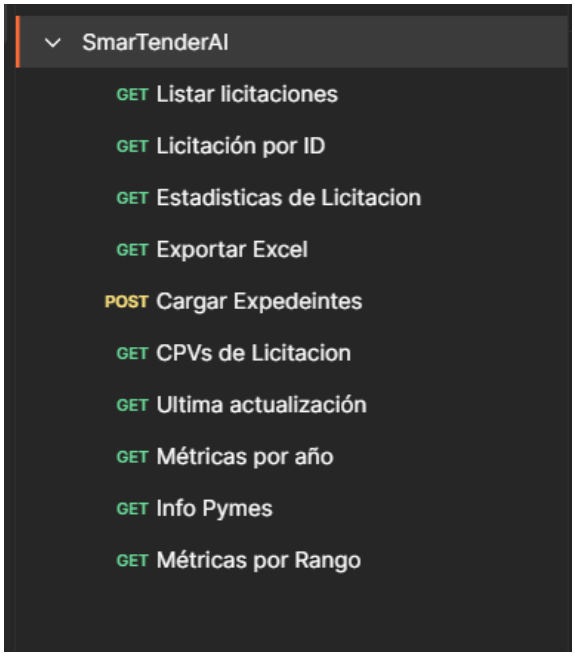


Figura 7.1: Colección de Postman

comprobar que los códigos de estado HTTP de las respuestas de las APIs, los formatos de datos y los mensajes de respuesta fueran correctos y acordes a lo esperado.

Se comprobó que las respuestas del servidor fueron consistentes con los datos enviados y que la lógica de negocio implementada en los endpoints funciona correctamente.

7.3 Estadísticas de variables

Para evaluar el desempeño del modelo en la extracción de información, se utilizan tres métricas clave: **Precisión**, **Recall** y **F1-Score**.

1. **Precisión**: Esta métrica mide la proporción de instancias identificadas por el modelo que son realmente correctas. En otras palabras, indica qué tan exacto es el modelo cuando predice una instancia positiva. Se calcula con la siguiente fórmula:

$$\text{Precisión} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos}}$$

   Donde:

   - **Verdaderos Positivos (True Positives, TP)**: El número de instancias correctamente identificadas como positivas.
   - **Falsos Positivos (False Positives, FP)**: El número de instancias incorrectamente identificadas como positivas.

2. **Recall (Sensibilidad o Exhaustividad)**: Esta métrica mide la proporción de instancias positivas que el modelo ha identificado correctamente. En otras palabras, evalúa qué tan bien el modelo encuentra todas las instancias positivas. Se calcula con la siguiente fórmula: Donde:

$$\text{Recall} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$$

   - **Falsos Negativos (False Negatives, FN)**: El número de instancias positivas que el modelo no identificó.

3. **F1-Score**: Esta métrica es la media armónica entre la precisión y el recall. Es útil cuando se necesita un balance entre ambas métricas y se desea un solo valor que represente el desempeño del modelo en general. Se calcula con la siguiente fórmula:

$$\text{F1-Score} = 2 \times \frac{\text{Precisión} \times \text{Recall}}{\text{Precisión} + \text{Recall}}$$

Las métricas de rendimiento del modelo para la extracción de variables se han evaluado utilizando una muestra de 50 expedientes revisados a mano. Este análisis permite evaluar la eficacia del modelo en la identificación de diferentes variables, reflejando su capacidad para manejar datos de manera precisa y completa. Los resultados se presentan en la Tabla 7.1

1. **Rendimiento Perfecto**: El modelo ha alcanzado una precisión, recall y F1-Score de uno en variables como:

   - Unidad encargada del seguimiento y ejecución del contrato
   - Gastos por desistimiento o renuncia

Variable	Precisión	Recall	F1-Score
Unidad encargada del seguimiento y ejecución del contrato	1	1	1
Consideración como infracción grave del incumplimiento de las condiciones	0,94	0,84	0,89
Contratación del control de calidad	1	0,97	0,98
Gastos por desistimiento o renuncia	1	1	1
Inclusión del control de calidad en la propia obra	0,94	0,84	0,89
Criterios de adjudicación del procedimiento abierto	0,84	0,84	0,84
Mejoras como criterio de adjudicación	0,9	0,9	0,9
Obligación de indicar en la oferta si va a haber subcontratación	0,94	0,94	0,94
Penalidades en caso de incumplimiento de las condiciones	0,94	0,89	0,91
Plazo de recepción	1	1	1
Revisión de precios	0,89	0,94	0,91
Subasta electrónica	0,88	0,88	0,88
Tareas críticas que no podrán ser objeto de subcontratación	0,86	0,86	0,86
Clasificación	0,9	0,82	0,86
Subcontratación como criterio	1	1	1
Abonos a cuenta	0,96	0,96	0,96
Sistema de precios	0,78	0,9	0,84

Tabla 7.1: Estadísticas para cada variable extraida por el LLM

- Plazo de recepción
- Subcontratación como criterio

Esto indica que el modelo ha identificado correctamente todas las instancias de estas variables sin errores, mostrando una capacidad excelente en la extracción de estos datos. Esto se puede dar porque son variables que se encuentran mencionadas siempre una vez en el texto y son fáciles e identificar.

2. **Rendimiento Muy Alto**: Algunas variables muestran un rendimiento casi perfecto:
   - **Contratación del control de calidad** (Precisión: 1, Recall: 0,97, F1-Score: 0,98)
   - **Abonos a cuenta** (Precisión, Recall, F1-Score: 0,96)

   Variables como 'Contratación del control de calidad' y 'Abonos a cuenta' muestran métricas casi perfectas, con una precisión, recall y F1-Score muy elevados. Esto indica que el modelo no solo identifica correctamente la información relevante, sino que también cubre de manera exhaustiva todos los casos necesarios.

3. **Rendimiento Sólido con Variaciones**:

   - **Consideración como infracción grave del incumplimiento de las condiciones** (Precisión: 0,94, Recall: 0,84, F1-Score: 0,89)
   - **Inclusión del control de calidad en la propia obra** (Precisión: 0,94, Recall: 0,84, F1-Score: 0,89)

   La precisión es alta, pero el recall es algo menor, lo que sugiere que aunque la mayoría de las extracciones son correctas, el modelo no está capturando todas las instancias posibles de estas variables.

4. **Rendimiento Moderado**:

   - **Criterios de adjudicación del procedimiento abierto** (Precisión, Recall, F1-Score: 0,84)
   - **Clasificación** (Precisión: 0,9, Recall: 0,82, F1-Score: 0,86)
   - **Tareas críticas que no podrán ser objeto de subcontratación** (Precisión, Recall, F1-Score: 0,86)

   Estas variables muestran un rendimiento equilibrado pero no excepcional. Los valores de precisión y recall para estas variables indican que el modelo tiene un desempeño razonable en la identificación de la información relevante, pero hay margen para mejorar. En particular, para la variable Clasificación, aunque el modelo tiene una buena precisión, no ha logrado identificar todas las clasificaciones requeridas en algunos documentos de licitación.

5. **Desempeño Menos Óptimo**: Para variables como:

   - **Sistema de precios** (Precisión: 0,78, Recall: 0,9, F1-Score: 0,84)

   La precisión en la identificación del 'Sistema de precios' es relativamente baja en comparación con el recall. Esto sugiere que, aunque el modelo logra identificar muchas instancias correctas de esta variable, también comete un número significativo de errores. La causa principal de esta discrepancia es la variabilidad en el formato del apartado relacionado con el sistema de precios en los documentos de licitación.

   En los expedientes, este apartado puede presentarse de diferentes maneras: en algunos casos se menciona únicamente los sistemas de precios aplicables; en otros, se enumeran todos los sistemas y se marca con una cruz los que se aplican; y en otros documentos, se utilizan casillas de verificación con opciones 'Sí' y 'No' para indicar los sistemas aplicables. Esta diversidad en los formatos ha demostrado ser un desafío para el modelo, que tiene dificultades para interpretar estos textos variados. Como resultado, aunque el modelo generalmente identifica los sistemas correctos, también tiende a incluir sistemas que no se aplican debido a la falta de consistencia en la presentación del texto.

7.3.1. Criterios y Valoraciones

Dado que los criterios de adjudicación y las valoraciones de las empresas son información crucial y, a la vez, la más difícil de extraer, hablaremos de sus estadísticas más en profundidad. Para ello, se revisaron manualmente 20 expedientes y se anotaron los criterios de adjudicación y las valoraciones asignadas por las empresas a cada criterio. Posteriormente, se compararon estos datos con los criterios y valoraciones extraídos por el modelo, disponibles en la aplicación web desarrollada SmarTenderAI.

Se calcularon las métricas de precisión, recall y F1-score para cada licitación, tanto para los criterios de adjudicación como para las valoraciones. A partir de estos cálculos, se determinó la media de estas métricas a través de todas las licitaciones analizadas.

En este contexto:

- **Un Verdadero Positivo (TP)** se refiere a los criterios relevantes que el modelo ha identificado correctamente.
- **Un Falso Positivo (FP)** corresponde a criterios irrelevantes o información adicional que el modelo ha extraído erróneamente.
- **Un Falso Negativo (FN)** son los criterios relevantes que el modelo no ha logrado extraer.

Criterios

Para el caso específico de los criterios de adjudicación, las métricas calculadas son las siguientes:

- **Precisión**: 0,89
- **Recall**: 0,78
- **F1-Score**: 0,83

La precisión de 0,89 indica que el modelo tiene una alta capacidad para identificar correctamente los criterios de adjudicación relevantes. Es decir, de todas las veces que el modelo ha identificado un criterio como relevante, el 89 % de estas identificaciones eran realmente correctas. Esto refleja que el modelo tiene una baja tasa de errores en los casos que considera positivos.

Un recall de 0,78 muestra que el modelo ha logrado identificar el 78 % de todos los criterios de adjudicación que realmente estaban presentes en el texto. Aunque el modelo es bastante efectivo para encontrar los criterios relevantes, existe un 22 % de criterios relevantes que no fueron detectados. Esto indica que el modelo podría mejorar en la cobertura de todos los criterios que deberían ser extraídos.

El F1-score de 0,83 proporciona una medida equilibrada entre la precisión y el recall. Un F1-score de 0,83 sugiere que el modelo tiene un buen equilibrio entre identificar correctamente los criterios relevantes y no incluir información irrelevante, aunque todavía hay margen para mejorar. Algunos posibles motivos que pueden explicar las métricas obtenidas y los desafíos asociados son los siguientes:

1. **Texto Largo**: En textos extensos, como los encontrados en los expedientes de licitación, el modelo puede tener dificultades para extraer toda la información relevante. La longitud del texto puede hacer que el modelo pase por alto algunos criterios o confunda información relacionada.
2. **Criterios Mencionados Varias Veces con Nombres Diferentes**: En algunos documentos, los criterios de adjudicación pueden ser mencionados varias veces con nombres diferentes o variaciones en la forma en que se presentan. Este problema puede complicar la tarea del modelo al intentar identificar y consolidar los diferentes nombres o formatos para el mismo criterio.

3. **Confusión en la Extracción**: Los modelos de lenguaje pueden confundir criterios similares o relacionados, especialmente si están descritos de manera que se superponen o son ambiguos. Si los criterios están descritos de manera imprecisa, el modelo puede tener dificultades para distinguir entre criterios distintos o para identificar la relevancia de ciertos elementos del texto. La falta de claridad en la redacción de los documentos puede llevar a errores en la extracción y en la categorización de los criterios.

Valoraciones

En la evaluación del modelo para la extracción de valoraciones de las empresas, las métricas obtenidas son:

- **Precisión**: 0,85
- **Recall**: 0,66
- **F1-Score**: 0,74

La precisión de 0,85 indica que cuando el modelo identifica una valoración, en el 85 % de los casos es correcta. Esto sugiere que el modelo es bastante preciso en términos de evitar errores de inclusión de información no relevante. Sin embargo, un valor de 0,85 también muestra que hay un 15 % de casos en los que el modelo extrae información incorrecta o irrelevante.

El recall de 0,66 indica que el modelo identifica correctamente el 66 % de las valoraciones que deberían ser extraídas. En otras palabras, el modelo está perdiendo el 34 % de las valoraciones relevantes que deberían haber sido extraídas. Este valor bajo sugiere que el modelo tiene dificultades para captar todas las valoraciones presentes en los documentos.

El F1-Score de 0,74 proporciona una medida combinada de la precisión y el recall, equilibrando ambos aspectos. Un F1-Score de 0,74 indica un desempeño razonable en términos de equilibrio entre precisión y recall, pero no excepcional. Este valor muestra que todavía hay margen para mejorar tanto en precisión como en recall.

Algunos posibles motivos que pueden explicar las métricas obtenidas y los desafíos asociados son los siguientes:

- **Tablas que No Se Logran Extraer Correctamente**: Las valoraciones a menudo están en formato de tabla, que puede ser difícil de extraer y procesar. Si el sistema no puede extraer algunas tablas correctamente, el modelo no recibirá toda la información que puede ser crucial, afectando el recall.
- **Confusión en las Columnas Asociadas a los Criterios**: Si las valoraciones están asociadas a diferentes columnas en las tablas y el modelo confunde estas asociaciones, puede extraer información incorrecta o incompleta, lo que afecta la precisión.
- **Documentos Extensos**: En documentos largos y complejos, el modelo puede no revisar todas las secciones relevantes con suficiente detalle, lo que puede resultar en la omisión de valoraciones importantes y, por lo tanto, un bajo recall.
- **Diversidad en el Formato de Datos**: Diferentes formatos para presentar valoraciones a diferentes criterios pueden causar dificultades en la interpretación y extracción precisa por parte del modelo, contribuyendo a los errores observados.

- **Confusión entre valoraciones de criterios relacionados con el precio y ofertas económicas**: En algunos casos, el modelo confunde las puntuaciones asignadas a criterios relacionados con el precio y las ofertas económicas de las empresas. Esta confusión ocurre porque ambos tipos de datos pueden aparecer en contextos similares o utilizar nombres de columnas que se asemejan entre sí, ya que ambos están relacionados con el ámbito económico. Como resultado, el modelo puede extraer incorrectamente la oferta económica en lugar de la puntuación correcta asignada.

CAPÍTULO 8
Conclusiones

El desarrollo de SmarTenderAI ha permitido cumplir satisfactoriamente con los objetivos establecidos, demostrando la capacidad de integrar conocimientos tecnológicos avanzados y aplicarlos para resolver la problemática planteada en la introducción. A continuación, se indica dónde se cubre cada objetivo:

1. **Analizar los sistemas actuales de gestión de datos de contratación pública**: El análisis y comparación de los diferentes sistemas conjuntamente con SmarTender AI se realiza en la Sección 2 de Crítica al Estado del Arte, donde se ha realizado un análisis de cada herramienta disponible y se ha creado una tabla comparativa entre todas que reúne las principales diferencias entre ellas.

2. **Desarrollo del módulo de extracción de datos e integración del LLM:** Se logró crear un sistema automatizado capaz de extraer información clave de documentos HTML y PDF utilizando Python, tal como se describe en la Sección 6.1. Para ello, se ha implementado el LLM que permitió mejorar la precisión en la interpretación de datos complejos y contextuales, como valoraciones y criterios de evaluación, elevando la capacidad analítica del sistema. Esta integración se desarrolló utilizando la biblioteca de Python Langchain y el modelo GPT-4o-mini de OpenAI.

3. **Diseño y construcción de una base de datos SQL:** Se diseñó y construyó una base de datos en SQL Server que asegura la integridad y consistencia de la información almacenada, facilitando una gestión eficiente de los datos extraídos. El diseño de la base de datos se puede ver en la Sección 5.3 y el desarrollo de la misma, indicando los scripts utilizados para su creación, se puede consultar en la Sección 6.2.

4. **Desarrollo de una aplicación web interactiva:** Se creó una interfaz web intuitiva que facilita la visualización y gestión de los datos, ofreciendo un diseño minimalista pero fácil de entender. Este objetivo se cumple en la Sección 6.3 donde se detallan las partes principales de la aplicación.

5. **Uso de tecnologías avanzadas:** Uno de lo objetivos más sencillos de explicar y a la vez de los más complejos de poner en marcha, ya que supuso un aprendizaje constante durante todo el desarrollo del proyecto. Tecnologías como Python y SQL se pudieron emplear con el conocimiento base adquirido en la carrera, pero los conocimientos de tecnologías como React y Django han sido adquiridos a lo largo del desarrollo del proyecto mediante la investigación y autoaprendizaje. Las tecnologías que se han utilizado se detallan en la Sección 4.

6. **Aseguramiento de la calidad y consistencia de los datos:** Se establecieron mecanismos sólidos para la validación y normalización de datos antes de ser introducidos

en la base de datos. Algunos de estos mecanismos son la comprobación de que cada variable tiene el tipo que le corresponde y transformarlas en caso de que sea necesario para evitar excepciones a la hora de cargar los objetos en la base de datos. Este objetivo se alcanza en la Sección 6.1 donde se explica el desarrollo del script encargado de estas tareas.

7. **Pruebas exhaustivas del sistema:** Se realizaron pruebas unitarias y de integración que demostraron la correcta funcionalidad de todos los componentes del sistema, garantizando su fiabilidad. También se calcularon estadísticas sobre las variables para ofrecer una visión general de la eficacia de la extraccion. Este objetivo se cubre en la Sección 7 donde se detallan las pruebas.

8. **Documentación del proceso de desarrollo y resultados:** Esta memoria sirve como documentación detallada que cubre todo el proceso de desarrollo, las pruebas realizadas y los resultados obtenidos, sirviendo como referencia para futuras mejoras.

9. **Diseño de un sistema escalable y mantenible:** El sistema fue diseñado para ser escalable y fácil de mantener, asegurando su adaptabilidad a futuras actualizaciones sin comprometer la funcionalidad. Es por ello que se han utilizado tecnologías modernas y se ha asegurado que el sistema interactúa con cada componente de forma correcta.

Una vez establecido el cumplimiento de los objetivos, cabe destacar también los problemas que han surgido a lo largo del desarrollo. Uno de los primeros errores que se cometió fue el uso del LLM sin una estructura clara de cómo debía usarse. Dado que se trataba de un modelo de OpenAI de pago, esto supuso un gasto económico un poco mayor debido al uso poco cuidado al principio. Este gasto adicional se debió a la falta de un diseño claro del módulo y los scripts antes de su implementación. Esto se solucionó desarrollando un diseño más claro del módulo aclarando dónde debía usarse el modelo.

Por otro lado, la elección del modelo al principio también supuso un problema, ya que se utilizó GPT-3, que tenía una precisión inferior y muchos resultados no eran los esperados. Aunque GPT-4 ofrecía una mejora significativa en precisión, su alto coste era un impedimento. Afortunadamente, la disponibilidad de GPT-4o-mini, que tenía un coste similar al de GPT-3 pero una precisión considerablemente mejor, permitió solucionar este problema.

A nivel profesional, este proyecto ha permitido adquirir y consolidar conocimientos en tecnologías avanzadas de desarrollo web como React y Django que no se habían explorado en profundidad durante la carrera. Además, la capacidad para integrar y aplicar estos conocimientos en un proyecto real ha demostrado ser de gran valor.

Finalmente, a nivel personal este proyecto me ha ayudado a sobrellevar la frustración y a desarrollar la capacidad de centrarme en mis obligaciones, incluso cuando el entorno personal presenta problemas. La experiencia me ha enseñado a manejar mejor el estrés y a mantener la concentración en los objetivos del proyecto, independientemente de las dificultades externas. Esta habilidad no solo ha sido valiosa para el éxito de SmarTenderAI, sino que también ha contribuido a mi crecimiento como persona.

Por último, es importante destacar las asignaturas del grado cuyos contenidos han resultado de especial utilidad para el desarrollo de este proyecto:

1. **Bases de Datos y Sistemas de Información**: Esta asignatura me proporcionó el conocimiento necesario sobre bases de datos SQL, lo cual fue esencial para desarrollar los scripts de creación de tablas y las consultas necesarias para comprobar y manipular los datos.

2. **Interfaces Persona Computador**: La creación de la interfaz web intuitiva emplea principios de diseño de interfaces persona-computador, basados en los conocimientos adquiridos en el curso sobre experiencia de usuario y usabilidad.

3. **Algorítmica**:En esta asignatura, se estudiaron diversos algoritmos avanzados y, especialmente, se adquirió un profundo conocimiento del lenguaje de programación Python.

4. **Sistemas de Almacenamiento y Recuperación de Información**: Este curso incluyó proyectos de web scraping, que me proporcionaron las habilidades necesarias para extraer información de sitios web. Este conocimiento se aplicó directamente para extraer datos de licitaciones a partir del HTML.

5. **Aprendizaje Automático**: En esta asignatura se han estudiado más a fondo diferentes técnicas de inteligencia artificial, lo que me ha permitido comprender el funcionamiento de los modelos de lenguaje.

CAPÍTULO 9
Trabajos Futuros

A pesar de que SmarTenderAI ha alcanzado los objetivos planteados, hay varios aspectos y posibles mejoras que podrían ser explorados en futuros desarrollos:

1. **Optimización de Procesos de Extracción:** Aunque el sistema actual es funcional, una optimización adicional en el proceso de extracción de datos podría mejorar la velocidad y la precisión, especialmente con documentos de gran tamaño o complejidad.

2. **Ampliación del Alcance de Documentos:** El sistema actualmente se centra en documentos HTML y PDF. Ampliar la compatibilidad para incluir otros formatos, como DOCX o formatos específicos de licitación, podría mejorar la versatilidad de la herramienta.

3. **Ampliación de Expedientes de otras Administraciones**: Extender la capacidad del sistema para manejar expedientes de otras administraciones públicas que presentan diferentes formas de expresar los contenidos y estructurar los documentos.

4. **Uso de Modelos de Lenguaje Más Avanzados:** Con el avance continuo en la tecnología de modelos de lenguaje, la integración de modelos más avanzados o especializados en el futuro podría mejorar aún más la precisión y el análisis contextual de los datos extraídos.

5. **Modelo de Predicción de Resultados de Licitaciones:** Una mejora significativa sería la incorporación de un modelo de predicción que utilice técnicas de aprendizaje automático para estimar los resultados de las licitaciones. Este modelo podría analizar patrones en las licitaciones de SmarTenderAI para predecir las probabilidades de éxito y ayudar a los usuarios a tomar decisiones más informadas.

6. **Integración con Sistemas de Gestión de Proyectos:** Integrar el sistema con herramientas de gestión de proyectos o plataformas de colaboración podría facilitar la incorporación de datos extraídos directamente en entornos de trabajo existentes, mejorando la gestión de licitaciones y proyectos asociados.

Bibliografía

[1] "TED - Portal de anuncios de licitación pública de la Unión Europea." [Online]. Available: https://ted.europa.eu/es/

[2] S. G. de Coordinación de la Contratación Electrónica, "PLACSP - Plataforma de Contratación del Sector Público." [Online]. Available: https://contrataciondelestado.es/wps/portal/plataforma

[3] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[4] Meta, "Página web de meta." [Online]. Available: https://about.meta.com

[5] ——, "Introducing meta llama 3: The most capable openly available llm to date," 2024. [Online]. Available: https://ai.meta.com/blog/meta-llama-3/

[6] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[7] OpenAI, "Openai," https://openai.com, 2024, accessed: 2024-06-31.

[8] ——, "Gpt-4," https://openai.com/index/gpt-4/, 2023, accessed: 2024-06-31.

[9] Gobierto, "Contratación pública," 2024. [Online]. Available: https://www.gobierto.es/contratacion

[10] ——, "Control y planificación," 2024. [Online]. Available: https://www.gobierto.es/contratacion-control

[11] ——, "Explorador de datos," 2024. [Online]. Available: https://www.gobierto.es/contratacion-explorador

[12] D. G. del Patrimonio del Estado, *Manual de uso de OpenPLACSP*, Subdirección General de Coordinación de la Contratación Electrónica. [Online]. Available: https://contrataciondelestado.es/datosabiertos/DGPE_PLACSP_OpenPLACSP_v.1.3.pdf

[13] "Subdirección general de coordinación de la contratación electrónica," Ministerio de Hacienda y Función Pública, Gobierno de España. [Online]. Available: https://www.hacienda.gob.es/es-ES/El%20Ministerio/Paginas/Organigrama/CVs/DireccionGeneraldelPatrimoniodelEstado.aspx

[14] Tendios, "Página web de tendios," 2024. [Online]. Available: https://tendios.com

[15] Tender-Licitaciones, "Tender-licitaciones," https://tender-licitaciones.com, 2024, accessed: 2024-06-22.

[16] D. G. de Patrimonio del Estado, "Resumen de contenido en conjuntos de datos abiertos," 2023. [Online]. Available: https://contrataciondelsectorpublico.gob.es/datosabiertos/DGPE_PLACSP_ResumenDatosAbiertos.pdf

[17] M. Aibin, "Energy consumption of chatgpt responses," 2024, reviewed by Milos Simic. [Online]. Available: https://www.baeldung.com/cs/chatgpt-large-language-models-power-consumption

[18] Gobierno de España, "Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales," Dec. 2018, BOE número 294, de 6 de diciembre de 2018. [Online]. Available: https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673

[19] Parlamento Europeo y Consejo de la Unión Europea, "Reglamento (ue) 2016/679 del parlamento europeo y del consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos," Apr. 2016, diario Oficial de la Unión Europea, L 119, 4 de mayo de 2016. [Online]. Available: https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX%3A32016R0679

[20] Gobierno de España, "Ley 37/2007, de 16 de noviembre, sobre reutilización de la información del sector público," Nov. 2007, BOE número 276, de 17 de noviembre de 2007. [Online]. Available: https://www.boe.es/buscar/act.php?id=BOE-A-2007-19814

APÉNDICE A
Glosario de términos y abreviaturas

- **Frontend**: Parte de una aplicación que el usuario ve e interactúa, desarrollado con HTML, CSS y JavaScript.
- **Backend**: Parte del servidor en una aplicación que maneja la lógica, el acceso a la base de datos y la comunicación con el frontend.
- **API (Interfaz de Programación de Aplicaciones)**: Conjunto de reglas para que diferentes programas interactúen entre sí.
- **REST (Representational State Transfer)**: Arquitectura para diseñar APIs que utiliza HTTP para la comunicación entre cliente y servidor.
- **REST API**: API que sigue los principios REST para permitir la comunicación entre clientes y servidores mediante HTTP.
- **Endpoint**: URL específica en una API para acceder a recursos o realizar operaciones.
- **LLM (Large Language Model)**: Modelo de inteligencia artificial entrenado para generar y entender texto en lenguaje natural.
- **Prompt**: Entrada dada a un modelo de lenguaje para generar una respuesta.
- **SQL (Structured Query Language)**: Lenguaje para gestionar y manipular bases de datos relacionales.
- **CRUD (Create, Read, Update, Delete)**: Operaciones básicas para gestionar datos en aplicaciones.
- **JSON (JavaScript Object Notation)**: Formato de intercambio de datos, fácil de leer y escribir para humanos y máquinas.
- **Token**: Cadena de caracteres utilizada para la autenticación y autorización en sistemas seguros.
- **Test Unitario**: Pruebas que verifican el funcionamiento de componentes individuales del software de forma aislada.
- **Tests de Integración**: Pruebas que verifican la interacción entre diferentes componentes o servicios del software.

APÉNDICE B

Scripts SQL de Creación de Tablas

```sql
    -- Tabla CodigosCPV
CREATE TABLE [dbo].[CodigosCPV] (
    [id_cpv] INT IDENTITY(1,1) PRIMARY KEY,
    [num_cpv] VARCHAR(500) NOT NULL,
    [descripcion] VARCHAR(500)
);

-- Tabla CPVLicitacion
CREATE TABLE [dbo].[CPVLicitacion] (
    [id_combinacion] INT IDENTITY(1,1) PRIMARY KEY,
    [id_licitacion] INT NOT NULL,
    [id_cpv] INT NOT NULL,
    FOREIGN KEY ([id_cpv]) REFERENCES [dbo].[CodigosCPV]([id_cpv]),
    FOREIGN KEY ([id_licitacion]) REFERENCES [dbo].[Licitaciones]([
        id_licitacion])
);

-- Tabla Criterios
CREATE TABLE [dbo].[Criterios] (
    [id_criterio] INT IDENTITY(1,1) PRIMARY KEY,
    [nombre] VARCHAR(200) NOT NULL,
    [siglas] VARCHAR(50),
    [valor_max] FLOAT,
    [valor_min] FLOAT,
    [id_padre] INT,
    FOREIGN KEY ([id_padre]) REFERENCES [dbo].[Criterios]([id_criterio])
);

-- Tabla django_admin_log
CREATE TABLE [dbo].[django_admin_log] (
    [id] INT IDENTITY(1,1) PRIMARY KEY,
    [action_time] DATETIMEOFFSET(7) NOT NULL,
    [object_id] NVARCHAR(MAX),
    [object_repr] NVARCHAR(200) NOT NULL,
    [action_flag] SMALLINT NOT NULL,
    [change_message] NVARCHAR(MAX) NOT NULL,
    [content_type_id] INT,
    [user_id] INT NOT NULL,
    FOREIGN KEY ([content_type_id]) REFERENCES [dbo].[django_content_type]([id
        ]),
    FOREIGN KEY ([user_id]) REFERENCES [dbo].[auth_user]([id]) -- Asumiendo
        que existe una tabla auth_user para usuarios
);

-- Tabla django_content_type
CREATE TABLE [dbo].[django_content_type] (
    [id] INT IDENTITY(1,1) PRIMARY KEY,
    [app_label] NVARCHAR(100) NOT NULL,
```

```
    [model] NVARCHAR(100) NOT NULL
);

-- Tabla django_migrations
CREATE TABLE [dbo].[django_migrations] (
    [id] BIGINT IDENTITY(1,1) PRIMARY KEY,
    [app] NVARCHAR(255) NOT NULL,
    [name] NVARCHAR(255) NOT NULL,
    [applied] DATETIMEOFFSET(7) NOT NULL
);

-- Tabla django_session
CREATE TABLE [dbo].[django_session] (
    [session_key] NVARCHAR(40) PRIMARY KEY,
    [session_data] NVARCHAR(MAX) NOT NULL,
    [expire_date] DATETIMEOFFSET(7) NOT NULL
);

-- Tabla Empresas
CREATE TABLE [dbo].[Empresas] (
    [id_empresa] INT IDENTITY(1,1) PRIMARY KEY,
    [nombre_empresa] VARCHAR(200) NOT NULL,
    [nif] VARCHAR(50),
    [pyme] BIT
);

-- Tabla Licitaciones
CREATE TABLE [dbo].[Licitaciones] (
    [id_licitacion] INT IDENTITY(1,1) PRIMARY KEY,
    [num_expediente] VARCHAR(50) NOT NULL,
    [abonos_cuenta] NVARCHAR(MAX),
    [clasificacion_subgrupo] VARCHAR(50),
    [clasificacion_grupo] VARCHAR(50),
    [clasificacion_cat] VARCHAR(50),
    [importe_con_impuestos] MONEY,
    [importe_sin_impuestos] MONEY,
    [criterios_economica] NVARCHAR(MAX),
    [criterios_tecnica] NVARCHAR(MAX),
    [criterios_valores_anormales] NVARCHAR(MAX),
    [condiciones_especiales] NVARCHAR(MAX),
    [infraccion_grave] NVARCHAR(MAX),
    [contratacion_control] BIT,
    [crit_adjudicacion] NVARCHAR(MAX),
    [fecha_adjudicacion] DATE,
    [fecha_formalizacion] DATE,
    [fecha_anuncio] DATE,
    [forma_pago] NVARCHAR(MAX),
    [garantia_def] NVARCHAR(MAX),
    [garantia_prov] NVARCHAR(MAX),
    [gastos_desistimiento] NVARCHAR(MAX),
    [modificaciones_prev] FLOAT,
    [prorrogas_prev] FLOAT,
    [revision_precios_prev] FLOAT,
    [otros_conceptos_prev] FLOAT,
    [inclusion_control_calidad] BIT,
    [lugar_ejecucion] VARCHAR(1000),
    [medios_economica] NVARCHAR(MAX),
    [medios_tecnica] NVARCHAR(MAX),
    [mejora_criterio] VARCHAR(50),
    [objeto] NVARCHAR(MAX),
    [obligacion_subcontratacion] BIT,
    [otros_componentes] VARCHAR(200),
    [penalidades_incumplimiento] NVARCHAR(MAX),
    [plazo_ejecucion] VARCHAR(50),
```

```
110     [plazo_garantia] VARCHAR(2000),
111     [plazo_presentacion] DATE,
112     [plazo_recepcion] NVARCHAR(MAX),
113     [plazo_maximo_prorrogas] NVARCHAR(MAX),
114     [ampliacion_presentacion] NVARCHAR(MAX),
115     [posibilidad_prorroga] NVARCHAR(MAX),
116     [procedimiento] INT,
117     [regimen_penalidades] NVARCHAR(MAX),
118     [revision_precios] NVARCHAR(MAX),
119     [sistema_precios] VARCHAR(1000),
120     [subasta_electronica] BIT,
121     [subcontratacion_criterio] BIT,
122     [tareas_criticas] NVARCHAR(MAX),
123     [tipo_contrato] INT,
124     [tramitacion] INT,
125     [unidad_encargada] NVARCHAR(MAX),
126     [valor_estimado] MONEY,
127     [adjudicatario] INT,
128     [num_incursas_anormalidad] INT,
129     [num_invitadas] INT,
130     [num_seleccionadas] INT,
131     [num_licitadores] INT,
132     [num_excluidas] INT,
133     [pag_info_criterios] INT,
134     [porcentaje_max_subcontratacion] FLOAT
135 );
136
137 -- Tabla Links
138 CREATE TABLE [dbo].[Links] (
139     [id_link] INT IDENTITY(1,1) PRIMARY KEY,
140     [link] NVARCHAR(200) NOT NULL,
141     [type_link] INT NOT NULL,
142     [id_licitacion] INT NOT NULL,
143     FOREIGN KEY ([id_licitacion]) REFERENCES [dbo].[Licitaciones]([
            id_licitacion]),
144     FOREIGN KEY ([type_link]) REFERENCES [dbo].[TipoLink]([id_tipo_link])
145 );
146
147 -- Tabla Participaciones
148 CREATE TABLE [dbo].[Participaciones] (
149     [id_participacion] INT IDENTITY(1,1) PRIMARY KEY,
150     [id_licitacion] INT NOT NULL,
151     [id_empresa] INT NOT NULL,
152     [importe_ofertado_sin_iva] MONEY,
153     [importe_ofertado_con_iva] MONEY,
154     [excluida] BIT,
155     [anormalidad_economica] BIT,
156     FOREIGN KEY ([id_licitacion]) REFERENCES [dbo].[Licitaciones]([
            id_licitacion]),
157     FOREIGN KEY ([id_empresa]) REFERENCES [dbo].[Empresas]([id_empresa])
158 );
159
160 -- Tabla TipoContrato
161 CREATE TABLE [dbo].[TipoContrato] (
162     [id_tipo_contrato] INT IDENTITY(1,1) PRIMARY KEY,
163     [nombre_tipo_contrato] VARCHAR(50) NOT NULL
164 );
165
166 -- Tabla TipoLink
167 CREATE TABLE [dbo].[TipoLink] (
168     [id_tipo_link] INT IDENTITY(1,1) PRIMARY KEY,
169     [texto_tipo_link] VARCHAR(100) NOT NULL
170 );
171
```

```
172 -- Tabla TipoProcedimiento
173 CREATE TABLE [dbo].[TipoProcedimiento] (
174     [id_procedimiento] INT IDENTITY(1,1) PRIMARY KEY,
175     [nombre_procedimiento] VARCHAR(50) NOT NULL
176 );
177
178 -- Tabla TipoTramitacion
179 CREATE TABLE [dbo].[TipoTramitacion] (
180     [id_tramitacion] INT IDENTITY(1,1) PRIMARY KEY,
181     [nombre_tramitacion] VARCHAR(50) NOT NULL
182 );
183
184 -- Tabla Valoraciones
185 CREATE TABLE [dbo].[Valoraciones] (
186     [id_valoracion] INT IDENTITY(1,1) PRIMARY KEY,
187     [id_participacion] INT NOT NULL,
188     [id_criterio] INT NOT NULL,
189     [puntuacion] FLOAT,
190     FOREIGN KEY ([id_participacion]) REFERENCES [dbo].[Participaciones]([
        id_participacion]),
191     FOREIGN KEY ([id_criterio]) REFERENCES [dbo].[Criterios]([id_criterio])
192 );
```

APÉNDICE C
Manual de Usuario

En esta sección se explica como desplegar la aplicación para su uso tanto en local como en la nube.

En caso de cualquier problema o duda contactar con: martaramalho.work@gmail.com

C.1 Despliegue local

Clonación del Repositorio de la Aplicación

1. Asegurarse de tener Git instalado en el ordenador. Descargar e instalar desde Git.
2. Abrir una terminal o línea de comandos.
3. Clonar el repositorio de la aplicación ejecutando el siguiente comando:

```
git clone https://github.com/MartaRamalho/TFG-IngInformatica-UPV.git
```

Instalación de Herramientas Necesarias

1. Asegurarse de tener Node.js y Python instalados en el ordenador. Descargar e instalar desde los siguientes enlaces:
   - Node.js
   - Python
2. Instalar SQL Server, el sistema de base de datos utilizado por la aplicación, disponible en SQL Server. También es necesario el SQL Server Management Studio (SSMS) para gestionar SQL Server. Descargar desde SSMS

Instalación de Dependencias del Proyecto

Dentro de la carpeta del proyecto clonado, accedemos a la carpeta `\project` y en la carpeta del frontend ejecutamos:

```
npm install
```

Y en la del backend ejecutamos:

```
pip install -r requirements.txt
```

Con esto instalamos las dependencias necesarias para el proyecto.

Configuración de la Base de Datos

1. Abrir SQL Server Management Studio (SSMS).
2. Conectarse a la instancia de SQL Server.
3. Cargar el archivo `.sql` proporcionado en el repositorio, que contiene las instrucciones para crear la base de datos, las tablas y poblar la base de datos:
   - En SSMS, hacer clic en "Nueva consulta".
   - Copiar y pegar el contenido del archivo `.sql` en la ventana de consulta.
   - Ejecutar la consulta para crear la base de datos, las tablas y poblarlas con los datos iniciales.

Configuración del Proyecto para Conectar con la Base de Datos

1. Editar el archivo `settings.py` en el proyecto Django en la carpeta `\backend\backend`.
2. Localizar la sección `DATABASES` y configurar los parámetros de conexión con la base de datos SQL Server:

```
DATABASES = {
    'default': {
        'ENGINE': 'mssql',
        'NAME': 'TFG',
        'HOST': 'HOSTNAME\TFG',
        'PORT': '',  # Empty string for default port
        'OPTIONS': {
            'driver': 'ODBC Driver 17 for SQL Server',
        },
    },
}
```

   Donde HOSTNAME es el configurado en SQL Server por el usuario.

3. Acceder a la carpeta `\backend\api\scripts` y modificar la conexión de la base de datos en los archivos `AccessPagePlaywright.py` y `introducirDatosBD.py`

Configurar Token de OpenAI

Si se va a desear usar la función de cargar nuevos expedientes, se debe configurar en los archivos de los scripts en `\backend\api\scripts` un nuevo token personal de OpenAI

Iniciar la Aplicación

Se deben usar dos terminales separadas, una en la carpeta del frontend y otra en la carpeta del backend:

1. Iniciar el servidor del frontend ejecutando:

```
    npm run dev
```

2. Iniciar el servidor del backend de Django ejecutando:

```
python manage.py runserver
```

Acceder a la Aplicación

Abrir un navegador web e ingresar la dirección `http://localhost:puerto` para acceder a la aplicación desplegada localmente donde 'puerto' es el que se indica al iniciar el servidor del frontend.

C.2 Despliegue en la nube

Preparativos Iniciales

Crear una cuenta en un proveedor de servicios en la nube. Algunos proveedores comunes son:

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)

Configuración del Entorno de la Nube

1. Crear una instancia de servidor (máquina virtual) en el proveedor de nube seleccionado. Elegir una imagen con una distribución de Linux, como Ubuntu.
2. Conectarse a la instancia del servidor utilizando SSH. En una terminal, ejecutar:

```
ssh usuario@direccion-ip
```

   Sustituir `usuario` por el nombre de usuario de la instancia y `direccion-ip` por la dirección IP pública proporcionada por el proveedor de la nube.

Instalación de Herramientas en el Servidor en la Nube

1. Actualizar los paquetes del sistema:

```
sudo apt update && sudo apt upgrade -y
```

2. Instalar Git, Node.js, Python y SQL Server:
   - Git:

```
sudo apt install git -y
```

   - Node.js:

```
curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -
sudo apt install -y nodejs
```

   - Python:

```
sudo apt install python3-pip -y
```

   - SQL Server: Seguir las instrucciones oficiales de instalación de SQL Server en Linux proporcionadas por Microsoft, disponibles aquí.

Clonación del Repositorio en el Servidor

1. Clonar el repositorio del proyecto desde GitHub en la instancia del servidor:

```
git clone https://github.com/MartaRamalho/TFG-IngInformatica-UPV.git
cd TFG-IngInformatica-UPV/project
```

Instalación de Dependencias en el Servidor

1. Instalar las dependencias del frontend:

```
npm install
```

2. Instalar las dependencias del backend utilizando el archivo `requirements.txt` (si existe):

```
pip install -r requirements.txt
```

Configuración de la Base de Datos en la Nube

1. Crear una base de datos en SQL Server siguiendo el mismo procedimiento descrito en la sección de despliegue local. Conectarse a SQL Server instalado en la instancia y ejecutar el script `.sql` para crear las tablas y poblar la base de datos.

2. Configurar el archivo `settings.py` en el proyecto Django para conectarse a la base de datos en la instancia. La cadena de conexión en `settings.py` puede ser configurada como sigue:

```
    DATABASES = {
        'default': {
            'ENGINE': 'mssql',
            'NAME': 'TFG',
            'USER': 'usuario',
            'PASSWORD': 'password',
            'HOST': 'direccion-ip-del-servidor',
            'PORT': '1433',
            'OPTIONS': {
                'driver': 'ODBC Driver 17 for SQL Server',
            },
        }
    }
```

3. Acceder a la carpeta `\backend\api\scripts` y modificar la conexión de la base de datos en los archivos `AccessPagePlaywright.py` y `introducirDatosBD.py`

Configurar Token de OpenAI

Si se va a desear usar la función de cargar nuevos expedientes, se debe configurar en los archivos de los scripts en `\backend\api\scripts` un nuevo token personal de OpenAI

Configuración del Servidor Web

1. Instalar un servidor web como **Nginx**:

```
sudo apt install nginx -y
```

2. Configurar Nginx para servir la aplicación frontend y redirigir las solicitudes API al backend de Django.

Despliegue de la Aplicación

1. Iniciar el backend de Django como un servicio utilizando un servidor WSGI como **Gunicorn**:

```
gunicorn --workers 3 nombre_proyecto.wsgi:application
```

2. Iniciar el servidor del frontend:

```
npm run dev
```

   Configurar Nginx para servir los archivos estáticos generados.

Configuración del Dominio y HTTPS

1. Configurar un nombre de dominio en el proveedor de nube para apuntar a la dirección IP de la instancia.

2. Instalar **Certbot** y configurar un certificado SSL para habilitar HTTPS:

```
sudo apt install certbot python3-certbot-nginx -y
sudo certbot --nginx
```

Acceso a la Aplicación

Acceder a la aplicación desde cualquier navegador web utilizando el nombre de dominio configurado (o la dirección IP si no se dispone de un dominio).

APÉNDICE D
Objetivos de Desarrollo Sostenible

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	**Alto**	**Medio**	**Bajo**	**No procede**
ODS 1. **Fin de la pobreza.**				X
ODS 2. **Hambre cero.**				X
ODS 3. **Salud y bienestar.**				X
ODS 4. **Educación de calidad.**				X
ODS 5. **Igualdad de género.**				X
ODS 6. **Agua limpia y saneamiento.**				X
ODS 7. **Energía asequible y no contaminante.**				X
ODS 8. **Trabajo decente y crecimiento económico.**	X			
ODS 9. **Industria, innovación e infraestructuras.**	X			
ODS 10. **Reducción de las desigualdades.**	X			
ODS 11. **Ciudades y comunidades sostenibles.**				X
ODS 12. **Producción y consumo responsables.**	X			
ODS 13. **Acción por el clima.**				X
ODS 14. **Vida submarina.**				X
ODS 15. **Vida de ecosistemas terrestres.**				X
ODS 16. **Paz, justicia e instituciones sólidas.**	X			
ODS 17. **Alianzas para lograr objetivos.**				X

Tabla D.1: Tabla de Objetivos de Desarrollo Sostenible

Este Trabajo de Fin de Grado se alinea de manera significativa con varios Objetivos de Desarrollo Sostenible (ODS), destacando su contribución en áreas clave como el crecimiento económico, la innovación y la transparencia institucional. A continuación, se detalla cómo este proyecto se relaciona con los ODS:

ODS 8: Trabajo decente y crecimiento económico: La herramienta desarrollada facilita un análisis detallado de licitaciones y valoraciones de empresas, contribuyendo a un entorno más transparente y competitivo. La herramienta permite a las empresas tomar decisiones informadas sobre su participación en procesos de licitación. Esto promueve un crecimiento económico inclusivo y sostenible al asegurar que todas las empresas, independientemente de su tamaño, tengan acceso equitativo a la información necesaria para competir en el mercado. De esta manera, la herramienta ayuda a crear un entorno de negocios más justo y competitivo, fomentando así el desarrollo económico.

ODS 9: Industria, innovación e infraestructuras: La herramienta ofrece una capacidad avanzada para extraer una amplia gama de variables de datos gubernamentales

y automatizar la extracción de información desde documentos PDF de licitaciones. Esta innovación facilita la creación de soluciones tecnológicas más efectivas en la gestión de información pública, lo cual puede impulsar el desarrollo de infraestructuras digitales más robustas y avanzadas, esenciales para una industria tecnológica en crecimiento.

ODS 10: Reducción de las desigualdades: Un aspecto fundamental de la herramienta es su gratuidad, lo cual juega un papel crucial en la reducción de las desigualdades entre empresas de diferentes tamaños y capacidades económicas. El acceso libre a la herramienta asegura que todas las empresas, independientemente de su tamaño o situación económica, puedan beneficiarse de la información y los análisis que proporciona. Las pequeñas y medianas empresas (PYMES) a menudo enfrentan más barreras para acceder a herramientas avanzadas de análisis de datos de licitaciones debido a los costes asociados. Esta igualdad en el acceso a la información permite a las empresas más pequeñas competir de manera más efectiva en el mercado y tomar decisiones más informadas. Al reducir las barreras económicas para acceder a información crítica, la herramienta promueve un entorno más competitivo y justo, en el que todas las empresas tienen la oportunidad de crecer y prosperar.

ODS 12: Producción y Consumo Responsables : Al optar por usar modelos de lenguaje preentrenados en lugar de desarrollar nuevos modelos desde cero, se contribuye significativamente a la reducción del desperdicio de recursos y al uso más eficiente de los mismos. Entrenar un modelo de lenguaje desde el principio requiere un consumo intensivo de energía y recursos computacionales, lo cual tiene un impacto ambiental considerable. Al utilizar un modelo preentrenado como es GPT-4o, se minimiza esta necesidad y se evita el consumo excesivo de recursos. Esto no solo reduce la huella de carbono asociada con el desarrollo de nuevas tecnologías, sino que también apoya un enfoque más responsable en la producción tecnológica.

ODS 16: Paz, justicia e instituciones sólidas: La transparencia y la accesibilidad proporcionadas por la herramienta contribuyen a la construcción de instituciones más sólidas y responsables. La capacidad de evaluar y analizar datos sobre licitaciones y valoraciones de empresas ayuda a mejorar la rendición de cuentas y la justicia en los procesos de licitación. Al ofrecer un análisis claro y detallado de la información relevante, la herramienta fortalece la confianza en las instituciones involucradas en el proceso de licitación y mejora la transparencia en la toma de decisiones. Esto contribuye a evitar prácticas corruptas y asegurar la integridad en los procesos de licitación y evaluación.