Universidad
Rey Juan Carlos

GRADO EN INGENIERÍA DE ROBÓTICA SOFTWARE

Curso Académico 2023/2024

Trabajo Fin de Grado

MINERÍA Y VISUALIZACIÓN DE DATOS DE CONTRATACIÓN PÚBLICA DEL ESTADO

Autor: Miguel Azores Picón
Tutor: Jesús María González Barahona

A mis amigas, mi madre y mi padre
por quererme en este proceso.
A mí, por el esfuerzo.

Agradecimientos

Realmente es complicado eso de poner los agradecimientos en un trabajo que pone fin a una etapa tan larga e intensa como es la universidad. Han pasado tantas cosas en estos años que es difícil concretar los agradecimientos, pero vamos a ello.

A mis amigas, a todas ellas, porque sin ellas estos años no habrían sido así de felices. A las que han estado siempre al otro lado del teléfono, a las que me han hecho echar de menos Jerez, a las que han hecho que los reencuentros en Jerez fuesen tan chulos y a todas las que han aparecido por el camino en fiestas, asambleas, cenitas y viajes. Mención especial a todas las que este último año de TFG me han acompañado en el día a día de este trabajo, María, Marta, Rebeca, Claudia, Paula, Amaro...gracias por motivarme para terminarlo.

A mi familia, sobre todo a mi madre y a mi padre. Porque siempre me han apoyado y me han dado libertad. Porque me han enseñado a esforzarme y a disfrutar. Por enseñarme a tomar decisiones y obligarme a tomarlas. Por darme la oportunidad de estudiar lo que quería y de vivir como quiero. También a mis abuelas y abuelos, porque recordarlos sigue enseñándome y acompañándome en cada cosa que vivo.

A mi tutor, que sin conocerme previamente aceptó una idea de proyecto sin demasiado contexto y me ha acompañado durante todo el proceso, a pesar de mis idas y venidas. También le agradezco esa charla que en su momento dio en la universidad: "La tecnología no es neutra", que me permitió ver que existía gente que pensaba la tecnología desde otro punto de vista. También a Vanessa, otra de las personas que me acercó durante la carrera a la ética de la tecnología y me acompañó en ese descubrimiento.

A los Scouts, las asambleas estudiantiles y Cultura de Barrio, proyectos a los que probablemente he dedicado más tiempo que a la universidad y que me han hecho y me hacen muy feliz.

Y a mí, por el esfuerzo realizado y por seguir trabajando cada día en ser mejor persona, pensando en lo común y poniendo mi granito de arena para construir un mundo mejor y una tecnología al servicio de las personas.

Resumen

Este proyecto ha consistido en desarrollar un conjunto de herramientas que mejoren el acceso a la información y la explotación de los **datos de contratación pública.** Para ello, se han desarrollado varias herramientas que permiten **extraer, almacenar y visualizar datos** de la Plataforma de Contratación del Sector Público (PLACSP). Estas tres herramientas son independientes y pueden funcionar por separado o de manera conjunta según el propósito deseado.

Con el fin de mostrar la utilidad de las mismas, se han desarrollado **dos prototipos** de aplicaciones que las integran. El Portal de Visualización de Datos está orientado a cualquier ciudadano y permite visualizar los datos de contratación de forma interactiva y accesible. El segundo prototipo desarrollado tiene como usuario objetivo a investigadores y permite la extracción, almacenamiento y visualización de los datos de cualquier órgano de contratación de forma automática.

El proyecto ha sido diseñado y desarrollado con el fin de ser presentado como Trabajo de Fin de Grado en Ingeniería de Robótica Software. Parte de la idea de implementar una herramienta que **mejore la trasparencia y el acceso a los datos** de la administración pública mediante el uso de tecnologías avanzadas, poniendo en práctica y ampliando todo lo aprendido durante el grado en lo que se refiere a desarrollo software, automatización de procesos y sistemas robótico, en este caso digitales.

Summary

This project involved developing a set of tools to improve access to and exploitation of public procurement data. Several tools were created to extract, store, and visualize data from the Spanish public procurement platform: Plataforma de Contratación del Sector Público (PLACSP). These three tools are independent and can operate separately or together, depending on the desired purpose.

To demonstrate the utility of these tools, two prototype applications were developed. One targets the general public and provides a data visualization portal for public procurement. The second prototype is designed for researchers and enables the automatic extraction, storage, and visualization of data from any contracting authority.

The project was designed and developed as a Final Degree Project in Software Robotics Engineering. It aims to implement a tool that improves transparency and access to public administration data through advanced technologies, applying and expanding all that has been learned during the university degree in software development, process automation, and digital robotic systems.

Índice general

Índice de figuras

Capítulo 1

Introducción

La idea de realizar este proyecto nace de la necesidad de almacenar, consultar y explotar los datos de contratación pública. En varios proyectos anteriores y debido a la curiosidad por la temática, he trabajado con la **Plataforma de Contratación del Sector Público**(PLACSP) [13]. Durante ese tiempo, observé una falta de herramientas que permitieran no solo visualizar los datos de manera accesible, sino también de extraerlos, estructurarlos y almacenarlos de forma eficiente. Esta deficiencia impide que los ciudadanos podamos acceder, comprender y analizar la información de los procesos de contratación pública de manera efectiva.

Entre las necesidades, se encontraba poder almacenar los datos, llevar un registro histórico de los mismos y tener la flexibilidad de **trabajar con los datos de forma local**. Este proyecto surge para cubrir estas necesidades, proporcionando un conjunto de herramientas que faciliten no solo la visualización de los datos, sino también su extracción, estructuración y almacenamiento.

1.1 Objetivos del proyecto

1.1.1 Objetivo general

Después de estudiar en profundidad el Portal de Contratación del Sector Público, el objetivo principal de este proyecto es **desarrollar un conjunto de herramientas modulares** que permitan la extracción, estructuración, almacenamiento y visualización de los datos de contratación pública, mejorando así la transparencia y el acceso a los datos de la administración.

1.1.2 Objetivos específicos

Con el fin de alcanzar el objetivo general, se han definido los siguientes objetivos específicos:

- Desarrollar un **sistema automatizado** para la extracción de datos de la Plataforma de Contratación del Sector Público.
- Implementar un **sistema eficiente de actualización de datos** que permita conocer el estado de las versiones anteriores de los expedientes.
- Desarrollar un **sistema de procesamiento de datos** que transforme datos crudos en información estructurada y estandarizada.
- Establecer una **base de datos robusta** que albergue datos limpios y estructurados, capaz de soportar consultas complejas.
- Proveer una **interfaz de visualización web interactiva** para que los usuarios puedan explorar los datos de contratación pública.
- Garantizar la **robustez, estabilidad y rendimiento del sistema** mediante pruebas exhaustivas.
- Disponer de una **documentación detallada** que incluya manuales de usuario y guías de uso para diversos niveles de usuario.
- Realizar una **evaluación externa y anónima** del funcionamiento y utilidad de la herramienta.

1.2 Estructura de la memoria

En esta sección del capítulo se muestra una visión general de la estructura del resto de la memoria con el fin de facilitar su lectura.

- **Capítulo 2:** En este capítulo se ofrece contexto del proyecto desde varias perspectivas. Primero se pone el foco en el contexto social y legal, luego se definen las técnicas y tecnologías usadas y por último se pone en contexto al lector sobre la plataforma desde la que se extraen los datos y las diferentes formas de acceder a ellos.
- **Capítulo 3:** En este capítulo se muestra el desarrollo del proyecto. Al haber basado la planificación del mismo en las metodologías *'Agile'*, este capítulo está separado por secciones que detallan cada sprint.

- **Capítulo 4:** En este capítulo se muestran los resultados del proyecto. El resultado final de este son tres herramientas modulares y dos prototipos, uno para un usuario general y otro más orientado a desarrolladores o investigadores. Este capítulo describe las herramientas, su implementación y su uso dentro de los prototipos.
- **Capítulo 5:** En este capítulo se exponen los métodos de evaluación y experimentación que se han utilizado durante el desarrollo del TFG. En el se pueden encontrar pruebas implementadas durante el desarrollo del código y un caso de uso real para probar la usabilidad de la herramienta y la experiencia de uso de los usuarios.
- **Capítulo 6:** En este último capítulo se presentan las conclusiones del proyecto, información sobre el desarrollo temporal del mismo, así como los trabajos futuros que podrían derivarse del proyecto.

1.3 Información adicional sobre el proyecto

A continuación se enlazan una serie de recursos adicionales que pueden ser de interés:

- **Página web del TFG:** En la página web del Trabajo fin de Grado podrá encontrar todos los recursos del proyecto, videos demostrativos e información relevante sobre el mismo.

  https://sites.google.com/view/tfg-miguelazores/inicio

- **Repositorio del TFG:** En la página web del Trabajo fin de Grado podrá encontrar todos los recursos del proyecto, videos demostrativos e información relevante sobre el mismo.

  https://github.com/MiguelAzores/TFG_MiguelAzores

- **Portal de Visualización de Datos de Contratación:** En este enlace encontrará la versión publicada del prototipo implementado, descrito en la sección 4.1.

  https://miguelazores.pythonanywhere.com/

Capítulo 2

Contexto del proyecto

En la primera parte de este capítulo se presentan los conceptos teóricos y legales en torno a los datos, el buen gobierno y la Plataforma de Contratación del Sector Público (Sección 2.1). En un segundo punto se expone una revisión detallada de las técnicas, tecnologías y metodologías relevantes utilizadas en el desarrollo del proyecto. Se presentan en orden de aparición dentro de los distintos módulos del proyecto, proporcionando el contexto necesario para entender la elección de cada tecnología y su papel dentro del proyecto (Sección 2.2). Por último, se muestra un análisis de la PLACSP, los datos que almacena y las herramientas que se ponen a disposición de ciudadanos, investigadores y empresas para el uso de los mismos (Sección 2.3).

2.1 Contexto social y legal

A lo largo de esta sección se profundiza en el contexto teórico que envuelve la idea del proyecto. Se expone la filosofía de datos abiertos enmarcada en el contexto gubernamental, fundamental para entender la motivación de este proyecto y conocer cómo deberían presentarse los datos públicos. También se trabaja sobre la Ley de Contratos del Sector Público para conocer las obligaciones legales de la plataforma de la que se extrae la información.

2.1.1 Datos abiertos

Los Datos de Gobierno Abiertos u *Open Government Data* son aquellos datos relacionados con el Estado que cualquier persona puede utilizar, reutilizar y redistribuir libremente, con la única condición, en su caso, de atribuir su fuente o reconocer la autoría. Esta forma de entender los datos públicos promueve la transparencia, la participación ciudadana y la innovación.

Un elemento principal de este modelo de gobernanza de datos es la posibilidad de ser reutilizados por cualquier entidad o ciudadano. La reutilización de la información del sec-

tor público implica que personas físicas o jurídicas puedan usar la información generada por organismos públicos con fines comerciales o no comerciales. Esta reutilización puede incluir actividades como la copia, difusión, modificación, adaptación, extracción, reordenación y combinación de la información. Gracias a esta práctica, se facilita el desarrollo de nuevos productos, servicios y soluciones que pueden aportar un alto valor social o económico.

La disponibilidad de estos datos permite a los ciudadanos ver cómo se gestionan los recursos públicos y tomar un papel activo en la vigilancia de las actividades gubernamentales. Esto refuerza la transparencia y la rendición de cuentas, promoviendo una administración pública más abierta y confiable. Además, contribuye a diseñar servicios más eficientes y cercanos a las necesidades de la ciudadanía. La publicación de datos mediante esta filosofía ayuda a mejorar la fiabilidad y seguridad de la información gestionada por las administraciones públicas. La transparencia en la gestión de datos públicos permite una mayor supervisión y mejora continua.

Existen diversas definiciones de datos abiertos y de las características que los datos publicados tienen que tener para serlo. En este proyecto se ha utilizado la aproximación que se recoge en OpenGovData.org [5], creada por treinta defensores del Gobierno Abierto en 2007. Los ocho principios de datos de gobierno abierto son:

- **Completos:** Los datos deben ser públicos y no estar sujetos a limitaciones de privacidad, seguridad o privilegios.
- **Primarios:** Deben ser recopilados en la fuente con el mayor nivel posible de granularidad.
- **Oportunos:** Deben estar disponibles rápidamente para preservar su valor.
- **Accesibles:** Deben estar disponibles para la más amplia gama de usuarios y propósitos.
- **Procesables por máquina:** Deben estar estructurados para permitir el procesamiento automatizado.
- **No discriminatorios:** Deben estar disponibles para cualquier persona sin necesidad de registro.
- **No propietarios:** Deben estar en un formato sobre el cual ninguna entidad tenga control exclusivo.
- **Sin licencia:** No deben estar sujetos a regulaciones de derechos de autor, patentes, marcas comerciales o secretos comerciales, salvo restricciones razonables de privacidad y seguridad.

En España existe Datos.gob.es [15], este portal gestionado por el Ministerio para la Transformación Digital y de la Función Pública se define como *"un punto de encuentro entre*

los diferentes actores que forman parte del ecosistema de los datos abiertos: administraciones públicas, infomediarios y usuarios." Esta plataforma busca ser un punto de localización y entrada a los datos públicos disponibles, así como, servir de escaparate a las aplicaciones y modelos de negocio que trabajan con estos datos.

2.1.2 Ley de Contratos del Sector Público (*Ley 9/2017*)

La Ley de Contratos del Sector Público (Ley 9/2017) [10] aprobada en 2017 y en vigor desde el 9 de marzo de 2018, tiene como objetivo modernizar y mejorar el sistema de contratación pública en España. Esta normativa se alinea con las directivas europeas sobre contratación pública y contratos de concesión y promueve una mayor transparencia, eficiencia e integridad en la gestión de los recursos públicos. Según el Artículo 1, la ley busca regular la contratación del sector público para garantizar que se ajuste a los principios de libertad de acceso a las licitaciones, publicidad, transparencia de los procedimientos, no discriminación e igualdad de trato entre los licitadores.

Transparencia y datos públicos

La ley refuerza la transparencia en los procedimientos de contratación, obligando a una mayor publicidad de los contratos y facilitando el acceso a la información tanto para los ciudadanos como para las empresas. Esto incluye la publicación de todos los contratos en la Plataforma de Contratación del Sector Público. No solo obliga a todos los órganos contratantes a publicar sus procedimientos de contratación, sino que además recoge la obligación de publicar información sobre la ejecución y cumplimiento de los contratos, permitiendo a los ciudadanos y a las empresas conocer el estado de los proyectos adjudicados en cada momento del proceso.

La Plataforma de Contratación del Sector Público (PLACSP)

Esta ley recoge la existencia y el funcionamiento de la Plataforma de Contratación del Sector Público, actúa como el punto central de información para todos los contratos del sector público. Esto centraliza y facilita el acceso a la información contractual. Según el Artículo 347, la Plataforma centralizará toda la información relevante de los contratos del sector público, garantizando la accesibilidad y transparencia.

El Artículo 63 establece que los órganos de contratación deben mantener un perfil del contratante dentro de esta plataforma en el que se publique toda la información relevante sobre los procedimientos de contratación. Esto incluye anuncios, pliegos y cualquier otro documento que forme parte del expediente de contratación.

Esta normativa recoge que la información publicada debe presentarse en estándares abiertos y reutilizables, promoviendo así su acceso y reutilización por parte de cualquier interesado. Para garantizar la trazabilidad de la información, la plataforma debe contar

con un sistema de sellado de tiempo que permite acreditar el inicio de la difusión pública de la información incluida en la misma.

2.2 Contexto tecnológico

A lo largo de esta sección se mencionan y comentan las tecnologías utilizadas en el proyecto, tanto en su versión final como en el desarrollo del mismo. Estas tecnologías se presentan agrupadas en varias subsecciones ordenadas según orden de aparición a lo largo del desarrollo del proyecto. En la sección 2.2.1 se presenta el principal lenguaje de programación utilizado y los etornos utilizados para desarrar el proyecto. Tras esto se presentan las tecnologías RPA (Sección 2.2.2). En la sección 2.2.3 se presentan las tecnologías utilizadas para la extracción de los datos, tras esto, las utilizadas para el procesamiento y la limpieza de los datos (Sección 2.2.4) y cosecutivamente (Sección 2.2.5) se presentan las herramientas usadas para el almacenamiento de los mismos. Por último en la sección 2.2.6 las tecnologías utilizadas en la visualización de los datos y el desplique de la aplicación web.

2.2.1 Entornos de trabajo y lenguaje de programación

Python

Python [18] ha sido el lenguaje de programación utilizado para el desarrollo del proyecto. Este es un lenguaje de alto nivel, orientado a objetos y con una sintaxis sencilla que permite tener código legible y limpio. Una de las grandes ventajas de Python es que dispone de un gran número de bibliotecas desarrolladas específicamente para la extracción, procesamiento y visualización de datos, además de ofrecer una comunidad de desarrolladores que han documentado y documentan todas las funcionalidades de las mismas. Dentro de campos como la extracción de datos o la visualización de los mismos, Python es una de las herramientas más utilizadas. Dentro de su amplia gama de bibliotecas se mencionarán posteriormente varias que permiten crear visualizaciones y gráficos de datos interactivos, crear interacciones automáticas con los navegadores y gestionar bases de datos, aportando eficiencia y versatilidad.

Visual Studio Code

El entorno de desarrollo utilizado durante todo el proceso ha sido Visual Studio Code (VSCode) [29]. Este es un editor de código desarrollado por Microsoft. Es software libre, multiplataforma y dispone de un gran número de extensiones que permiten adaptar el entorno de programación a tareas concretas y facilitar el proceso de desarrollo. Según la encuesta de Stack Overflow [6] sobre preferencia de los usuarios en cuanto a entornos de programación, Visual Studio Code es el IDE con mayor aceptación entre todos los usuarios con un 73.71%.

La decisión de utilizarlo frente a otros entornos centrados en Python está motivada por el conocimiento de la herramienta antes de empezar el proyecto y las dudas sobre la necesidad de utilizar otros lenguajes de programación en el proceso de desarrollo.

LaTeX

Para la redacción y maquetación de la memoria se ha utilizado LaTeX [28] en el editor de textos en línea Overleaf [7]. Este es un sistema de composición tipográfica de alta calidad que incluye características especialmente diseñadas para la producción de documentación técnica y científica. Cuenta con la posibilidad de incluir expresiones matemáticas, fragmentos de código, tablas y referencias. Estas características, junto al hecho de que es software libre, han hecho que LaTeX se convierta en el estándar para la redacción y publicación de artículos académicos, tesis y todo tipo de documentos científico-técnicos.

Para la escritura de este documento se ha utilizado una plantilla [8] de LaTeX creada por Gregorio Robles y Felipe Ortega, miembros del grupo de Sistemas y Comunicaciones (GSyC) de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad Rey Juan Carlos.

2.2.2 Robotic Process Automation (RPA)

La Automatización Robótica de Procesos, mayormente conocida por su nombre en Inglés Robotic Process Automation (RPA) [1], es una tecnología que utiliza robots de software para automatizar tareas repetitivas y basadas en reglas que normalmente realizan los seres humanos en las interfaces digitales. Estos sistemas, generalmente conocidos como *bots* pueden interactuar con aplicaciones y sistemas digitales de la misma forma que lo haría un ser humano, utilizando interfaces gráficas para navegar, extraer información o rellenar formularios.

En el contexto de este proyecto, las RPA se utilizan para automatizar la extracción de datos de la PLACSP de la mano de técnicas de web scraping. Se ha decidido implementar este tipo de tecnologías porque permiten simular acciones humanas para navegar por la plataforma, localizar y extraer la información relevante. Una de las características interesantes de estos sistemas es que son sencillamente escalables, puesto que pueden manejar fácilmente un incremento en la carga de trabajo, algo que permite que el ritmo de descarga y actualización de los datos no sea un problema.

2.2.3 Web scraping

El web scraping [12] es un conjunto de prácticas para extraer información de sitios web de manera automatizada. En el contexto de este proyecto, implica la simulación de la navegación web para acceder a los contenidos deseados, tratar el código de la página y

extraer la información deseada. Los métodos de extracción de datos han evolucionado significativamente desde sus inicios, cuando se empleaban técnicas manuales o elementales para la obtención de datos de sitios web estáticos. Con la creciente complejidad de las aplicaciones web y la necesidad de automatizar la recolección de datos, se han desarrollado herramientas más sofisticadas y eficientes.

Selenium

Selenium [22] es un software de código abierto que integra una serie de herramientas y bibliotecas con el fin de facilitar actividades de automatización de navegadores web. Esta herramienta permite a los desarrolladores realizar pruebas automatizadas de aplicaciones web, interactuar automáticamente con elementos de la interfaz de usuario y extraer datos de las páginas web. Esta capacidad de interactuar de manera automática con navegadores lo convierte en una herramienta poderosa para una variedad de tareas, desde pruebas de calidad hasta extracción de datos.

Selenium está compuesto de de varios elementos principales, siendo WebDriver uno de los más importantes. WebDriver permite una interacción directa con los navegadores web, simulando la acción de un usuario real. Esto incluye la capacidad de llenar formularios, hacer clic en botones, navegar entre páginas y extraer contenido dinámico generado por JavaScript. Su diseño facilita la creación de scripts de automatización que son robustos y reutilizables.

La elección de Selenium en este proyecto frente a otras herramientas similares como Puppeteer o Scrapy se debe a varios motivos. Primero, su compatibilidad con una amplia gama de navegadores y sistemas operativos proporciona una gran flexibilidad ya que Selenium es compatible no solo con Chrome, sino también con Firefox, Safari y Edge. También puede ejecutarse en diversos sistemas operativos como Windows, macOS y Linux. Esto es esencial para asegurar que las pruebas y automatizaciones sean aplicables en diferentes entornos. Otro factor importante ha sido su madurez, es una herramienta con mucho recorrido y una gran comunidad que garantiza estabilidad y soporte al proyecto. Esta larga trayectoria y su comunidad facilitan una gran cantidad de recursos para aprender a utilizar la herramienta, foros de resolución de dudas y numerosos ejemplos abiertos de sus usos.

2.2.4 Procesamiento y limpieza de datos

El procesamiento y limpieza de datos es una fase crítica en cualquier proyecto de que trabaje con datos, ya que garantiza que la información utilizada sea precisa y esté en un formato adecuado para el análisis. Estas técnicas implican transformar los datos brutos en un formato adecuado para el análisis, lo que incluye la limpieza de datos, la eliminación de duplicados, comprobación de los tipos, la imputación de valores faltantes y la normalización de datos. Python se ha convertido en un lenguaje perfecto para estas tareas, ya que en

torno a este se han desarrollado bibliotecas que permiten tratar los datos de forma rápida y eficaz. A continuación se mencionan algunas de las bibliotecas utilizadas en el proyecto.

Unidecode

Unicode es un sistema de codificación de caracteres utilizado por los equipos informáticos para el almacenamiento y el intercambio de datos en formato de texto. Asigna un número único a cada carácter de los principales sistemas de escritura del mundo. La biblioteca Unidecode de Python es una herramienta esencial para la normalización de cadenas de texto. Se ha utilizado para eliminar acentos y caracteres especiales, asegurando la consistencia en nombres y otros campos de texto.

Datetime

Este es un módulo estándar en Python que proporciona clases para manipular fechas y horas. Este módulo es utilizado para el manejo de datos temporales, permitiendo realizar operaciones como la conversión entre diferentes formatos de fecha, el cálculo de diferencias entre fechas y la manipulación de fechas y horas. Datetime ha sido utilizado para manejar las fechas de los expedientes de contratación, como las fechas de publicación, adjudicación y finalización de ofertas.

2.2.5 Bases de datos

Una base de datos relacional es un sistema que organiza los datos en estructuras tabulares, donde los datos se disponen en filas y columnas dentro de tablas. Estas tablas permiten la representación de conjuntos de datos de manera estructurada y accesible, facilitando la relación entre diferentes conjuntos de datos mediante el uso de claves primarias y externas.

Las claves primarias son identificadores únicos que distinguen cada fila dentro de una tabla, mientras que las claves externas son campos que vinculan una tabla con otra, estableciendo relaciones entre los datos en diferentes tablas. Esta estructura permite que las bases de datos relacionales manejen grandes volúmenes de información de manera eficiente, asegurando la integridad referencial y facilitando operaciones complejas de consulta y manipulación de datos.

Las bases de datos relacionales se basan en el modelo relacional, que utiliza el álgebra relacional para definir y manipular los datos. Este modelo permite realizar operaciones como selección, proyección y unión, que extraen y combinan datos de varias tablas basándose en sus relaciones. La implementación de estas operaciones se realiza a través del lenguaje de consulta estructurado (SQL), que se ha convertido en el estándar para interactuar con bases de datos relacionales.

SQLite

Para el almacenamiento y manejo de los datos se ha utilizado SQLite [24] como motor de bases de datos SQL. SQLite es una base de datos ligera basada en disco que no requiere de ningún servidor separado, lo que la hace rápida, autónoma y de alta confiabilidad. Esta característica es particularmente útil para aplicaciones de tamaño medio y bajo, ya que ofrece una solución eficiente y portátil para la gestión de datos sin la necesidad de una configuración de servidor compleja.

SQLite fue desarrollada en el año 2000 con el objetivo de proporcionar un sistema de bases de datos ligero y portátil. Su diseño embebido permite que sea integrada directamente en aplicaciones, destacándose por su ligereza y eficiencia en operaciones de lectura/escritura.

En este proyecto, se ha optado por SQLite en lugar de MySQL debido a la eficiencia y simplicidad que ofrece para el volumen de datos manejado. Los datos recopilados y con los que se trabaja no son pesados ni excesivos en cantidad, lo que hace que el almacenamiento en disco sea adecuado y eficiente. SQLite permite un acceso rápido y eficiente a la información durante las fases de análisis y visualización.

Para interactuar con SQLite desde Python, se ha utilizado el módulo Sqlite3 [25], que proporciona una interfaz SQL completa. Este módulo facilita la conexión y ejecución de consultas SQL, permitiendo la gestión de la base de datos de manera sencilla. La elección de SQLite y Sqlite3 se ha basado en la necesidad de una solución de base de datos que combine eficiencia, facilidad de uso y la capacidad de manejar adecuadamente el volumen de datos del proyecto.

SQLModel

La librería SQLModel [26] ha sido utilizada para interactuar con bases de datos SQL desde los códigos escritos en Python en este proyecto. SQLModel se basa en dos potentes librerías de Python: SQLAlchemy y Pydantic, combinando funcionalidades complementarias de ambas. Esta librería facilita la definición de modelos de datos utilizando clases de Python, y aprovecha las capacidades de Pydantic para la validación de datos. SQLModel permite a los desarrolladores definir modelos de datos de manera declarativa y asegurarse de que los datos sean válidos antes de ser almacenados en la base de datos. Esto reduce significativamente los errores y garantiza la consistencia de los datos.

Implementa una integración fluida con SQLAlchemy, lo que permite aprovechar todas las ventajas de un ORM completo y robusto. La facilidad para definir relaciones entre tablas, realizar consultas complejas y gestionar transacciones convierte a SQLModel en una herramienta idónea para la gestión de bases de datos en aplicaciones que necesiten trabajar con datos.

SQAlchemy

SQLAlchemy [23] es un potente ORM (Object-Relational Mapping) utilizado para mapear objetos relacionales y proporcionar una capa de abstracción sobre la interacción con la base de datos. Esta herramienta permite a los desarrolladores definir modelos de datos como clases de Python y proporciona una sintaxis intuitiva para realizar operaciones de bases de datos como inserciones, consultas, actualizaciones y eliminaciones. SQLAlchemy es conocido por su capacidad para gestionar eficientemente la interacción con bases de datos relacionales, ofreciendo un acceso de alto rendimiento a los datos.

Esta herramienta ofrece funcionalidades avanzadas como la gestión de transacciones, soporte para múltiples motores de bases de datos y la posibilidad de realizar consultas SQL personalizadas. Estas características hacen que sea una herramienta flexible y adaptable a una amplia variedad de necesidades y entornos de desarrollo. Su arquitectura modular permite a los desarrolladores utilizar solo las partes de la herramienta que necesitan, optimizando el rendimiento y la eficiencia del proyecto. SQLAlchemy proporciona una capa de abstracción que simplifica la interacción con la base de datos, permitiendo realizar operaciones complejas de manera eficiente.

Pydantic

Pydantic [17] es una herramienta de validación de datos y serialización que utiliza anotaciones de tipo (*type annotations*) de Python para definir y validar los datos de manera eficiente. Pydantic facilita la validación de datos a través de la definición de modelos de datos que garantizan que solo los datos correctos y válidos sean aceptados y procesados. Esta capacidad de validación es fundamental para mantener la integridad y consistencia de los datos en la base de datos.

Esta herramienta es capaz de trabajar con una gran variedad de tipos de datos, incluyendo tipos complejos como *dataclass* y *TypedDict*. Esta versatilidad permite integrar Pydantic a una gran variedad de proyectos. La herramienta también ofrece funcionalidades de serialización, lo que facilita la conversión de datos entre diferentes formatos, asegurando que los datos sean consistentes y correctos antes de ser almacenados. Esta robusta capacidad de validación y serialización es fundamental para la fiabilidad y estabilidad de las aplicaciones.

2.2.6 Visualización de datos

La visualización de datos es un conjunto de técnicas que trabajan para ofrecer una representación gráfica de la información y los datos. Mediante el uso de elementos visuales, como gráficos y mapas, la visualización de datos ofrece una manera accesible para detectar y comprender los datos, las tendencias y los patrones que hay en estos. La visualización de datos permite contar historias basadas en datos con un propósito, historias que son más

difíciles de contar desde una hoja de cálculos.

La importancia de la visualización de datos radica en su capacidad para transformar datos complejos en representaciones visuales comprensibles y significativas. Esto facilita la toma de decisiones basada en datos, ya que los patrones y tendencias pueden ser identificados rápidamente. Las visualizaciones efectivas no solo comunican información, sino que también pueden influir en la interpretación y comprensión de los datos, ayudando a resaltar datos u observaciones que de otra manera podrían pasar desapercibidas.

A medida que la era del Big Data avanza, las visualizaciones se convierten en herramientas claves para comprender la gran cantidad de datos que se generan en todos los sectores.

En el módulo de visualización de este proyecto se han utilizado herramientas basadas en Python que han facilitado una interacción directa con los datos, permitiendo crear una aplicación web que aloja gráficos interactivos. A continuación se detallan las herramientas utilizadas.

Plotly

Plotly [14] ha sido la biblioteca de generación de gráficos en Python elegida para la parte de visualización de este proyecto. Esta herramienta permite generar visualizaciones de datos interactivas de gran calidad con el fin de ser publicadas. Su curva de aprendizaje no es muy elevada, ya que dispone de una gran galería de ejemplos de uso bien documentados. Plotly permite crear una amplia variedad de gráficos, desde simples gráficos de barras y líneas hasta complejas visualizaciones tridimensionales y mapas.

Una de las principales ventajas de Plotly es su interactividad. Los usuarios pueden acercarse, alejarse, y desplazarse por los gráficos, lo que facilita una exploración más accesible de los datos. Además, Plotly permite la integración con otros lenguajes de programación y plataformas, como R, MATLAB y JavaScript, lo que lo hace muy versátil y útil a la hora de elegirlo como herramienta. La capacidad de exportar gráficos en diversos formatos, como imágenes estáticas, HTML interactivo y documentos PDF, es otro punto a favor para utilizarlo como herramienta de visualización en este proyecto.

Dash

Dash [4] ha sido la biblioteca utilizada para la visualización web de los datos. Este es un framework de Python pensado para construir aplicaciones web y que permite crear visualizaciones de forma sencilla. Las aplicaciones se renderizan en el navegador y son multiplataforma con soporte para móviles. Dash combina la potencia de las visualizaciones interactivas de Plotly con la simplicidad y versatilidad de un framework web, permitiendo desarrollar aplicaciones analíticas interactivas sin necesidad de conocimientos profundos en desarrollo web.

Esta herramienta facilita la creación de interfaces de usuario interactivas que pueden incluir gráficos, tablas y otros elementos de visualización. Utiliza componentes de React.js de forma interna, lo que proporciona una experiencia de usuario fluida y reactiva. Además, Dash permite la integración con diversas fuentes de datos y servicios en la nube, lo que facilita la conexión y el análisis de datos en tiempo real.

PythonAnyware

PythonAnyware [19] es una plataforma de alojamiento en la nube diseñada para alojar aplicaciones web basadas en Python. Permite desplegar aplicaciones desde entornos virtuales, facilitando la conexión y gestión con las bases de datos. Una de sus ventajas es el acceso remoto, que permite a los desarrolladores acceder a las aplicaciones y gestionarlas desde cualquier lugar, sin necesidad de infraestructura local. Otra de las ventajas de trabajar con esta plataforma es el soporte y la comunidad con la que cuenta. La documentación oficial es bátante clara y resolver problemas durante el desarrollo ha resultado sencillo. Otro de los puntos fuertes de la herramienta es la escalabilidad, estudiando los planes de pago se observa que no habría problema en alojar la aplicación en la plataforma si el proyecto creciese.

2.3 Análisis de la Plataforma de Contratación del Sector Público

En este capítulo se hace un análisis en profundidad de los datos que se encuentran publicados en la Plataforma de Contratación del Sector Público [13]. Esta plataforma es la que aloja los datos que nos interesan extraer y a lo largo de la sección se presentará dónde se encuentran, cómo se organizan y qué interés tienen.

2.3.1 Diccionario de términos

Para entender la forma en la que se almacenan los datos en la plataforma debemos conocer el lenguaje que esta usa y el significado de los elementos principales sobre los que se trabaja en este proyecto.

1. **Perfil del Contratante:** Es una entidad que permite identificar y gestionar las organizaciones autorizadas para abrir procedimientos de contratación pública. Cada perfil de contratante incluye información detallada sobre la organización, como su nombre, dirección, datos de contacto y otros datos relevantes. Además, este agrupa y organiza todos los expedientes de contratación asociados a esa entidad. Estas entidades tienen la capacidad de gestionar y publicar licitaciones y contratos menores.
2. **Licitaciones:** Son los procedimientos mediante los cuales una entidad del sector público invita a los proveedores a presentar ofertas para la provisión de bienes, ser-

vicios o la ejecución de obras. Dentro del perfil de contratante, las licitaciones son publicaciones específicas que describen los detalles del contrato propuesto, incluidos los requisitos técnicos, el presupuesto estimado, los plazos y las condiciones de participación.

3. **Expedientes:** Los expedientes de contratación son los registros que contienen toda la información y documentación relacionada con un proceso de contratación desde su inicio hasta su conclusión. Cada expediente incluye todas las etapas del proceso de licitación, desde la preparación y publicación de la licitación hasta la adjudicación y formalización del contrato.

2.3.2 Ubicación de los datos en la plataforma

En esta sección se muestra dónde se alojan los datos dentro de la plataforma y cuál es la interacción con la web que un usuario tiene que hacer para llegar a ellos. Esta ruta que el usuario debe hacer, es la ruta que realiza en el navegador el código de extracción de datos desarrollado.

1. **Perfil del contratante:** Cada perfil del contratante tiene una URL asignada que dirige a su página principal, estas son las URLs con las que trabajan los módulos de este proyecto. En este se encuentra toda la información relacionada con la institución pública seleccionada que crea y gestiona procesos de contratación. Como se observa en la siguiente figura (Figura 2.1), en la parte superior de la página se encuentra una pestaña que alberga las licitaciones, se debe hacer clic en la misma para llegar al listado de expedientes de licitación.

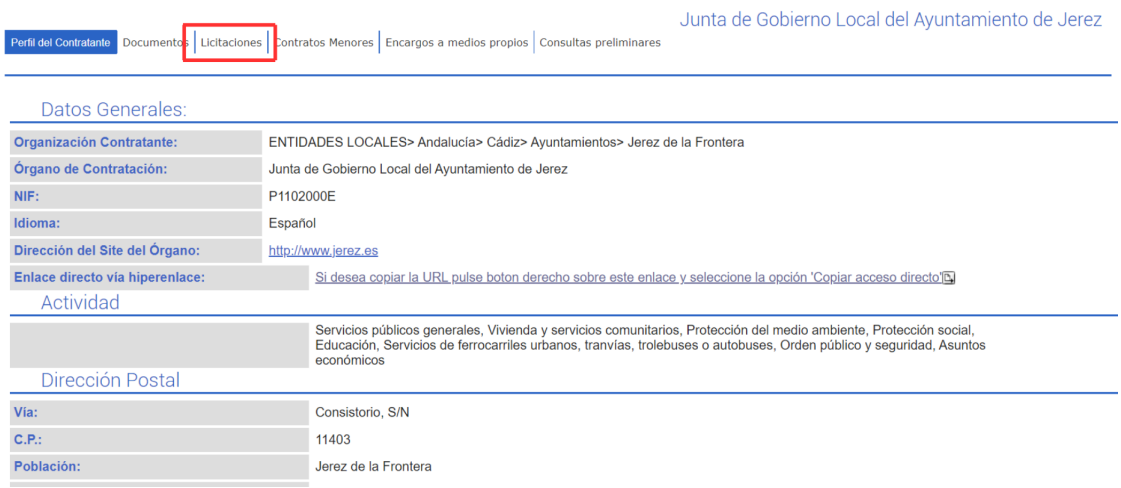


Figura 2.1: Captura de la PLACSP en la ventana que muestra el perfil del contratante.

2. **Pestaña de licitaciones:** Dentro de la pestaña de licitaciones se encuentra un listado con todos los expedientes de licitaciones que tiene el órgano de contratación seleccionado. Estos se encuentran paginados por orden de modificación y de 15 en 15. De

cada expediente se muestra información que permite identificar el estado actual del mismo, a esta información se le ha llamado **cabecera del expediente**. Esta cabecera recoge el nombre de expediente, el tipo, el objeto del contrato, su estado, el importe y las fechas relevantes en ese momento. Todos los nombres de expedientes tienen un enlace directo a la página que contiene la información detallada del mismo.

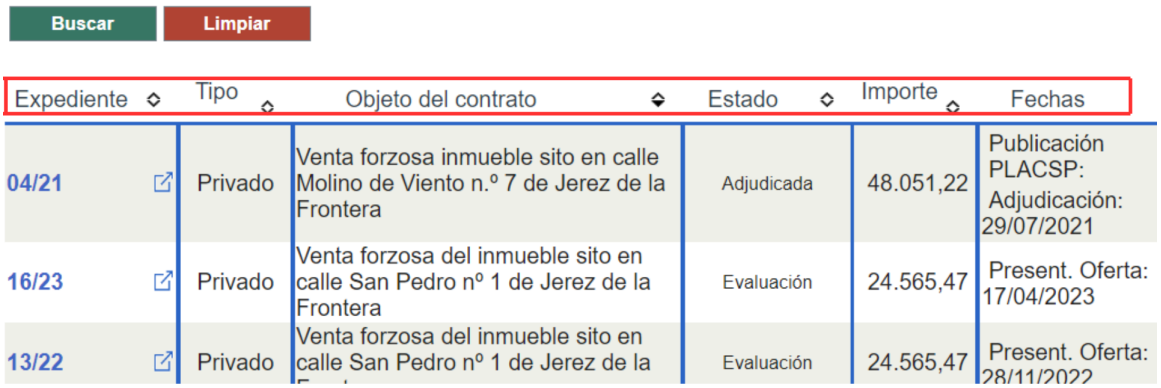
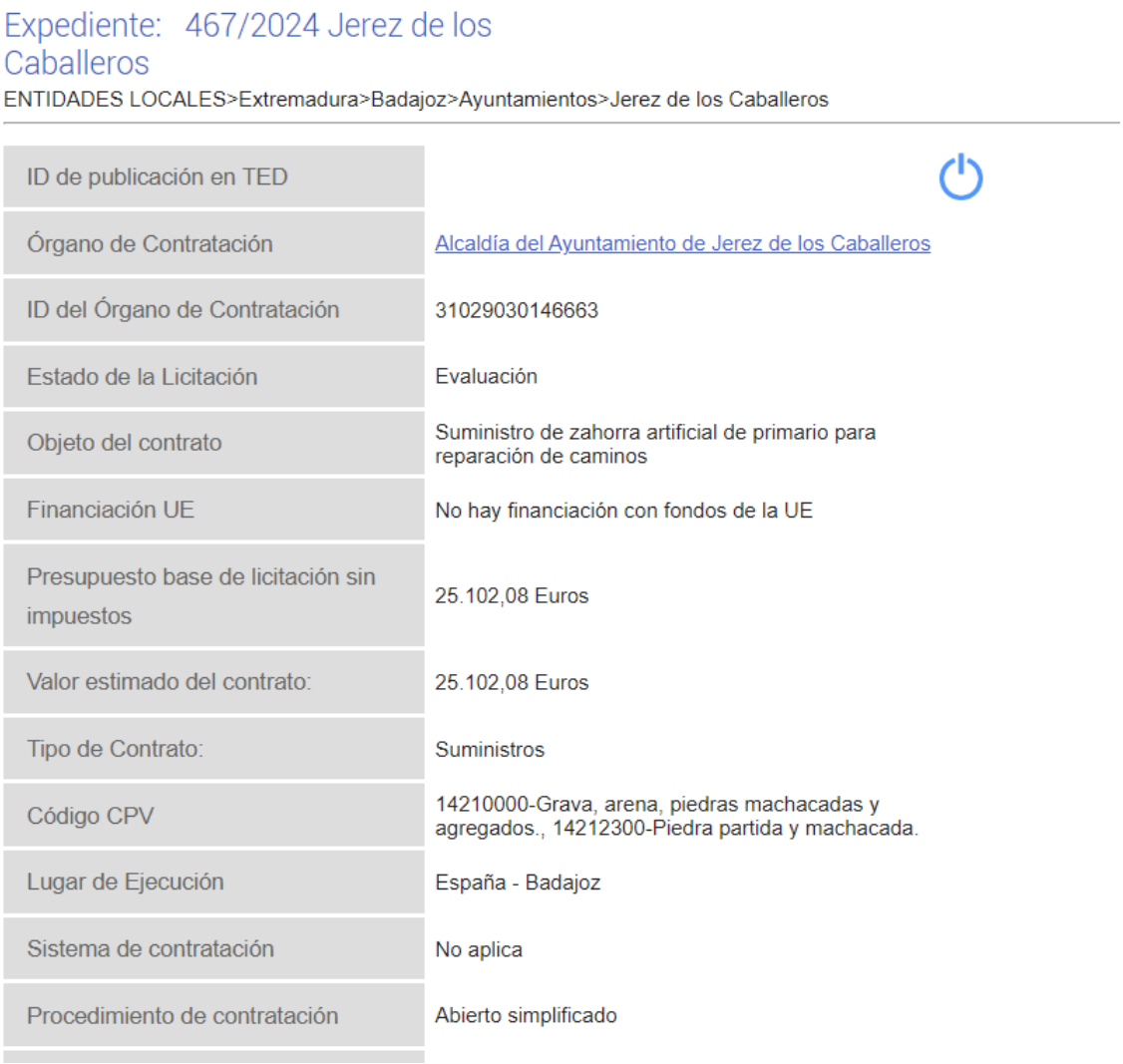


Figura 2.2: Captura del listado de expedientes de licitación. Resaltados los datos de la cabecera

3. **Detalles del expediente:** En esta ventana es donde se encuentra toda la información del expediente de contratación. Se pueden observar dos tablas, la primera con información resumen de la convocatoria de la oferta y una segunda tabla con detalles sobre el estado actual del expediente. Desde esta página también se puede acceder a los documentos relacionados con dicho expediente como son los pliegos o las actas.



Expediente: 467/2024 Jerez de los Caballeros

ENTIDADES LOCALES>Extremadura>Badajoz>Ayuntamientos>Jerez de los Caballeros

ID de publicación en TED	
Órgano de Contratación	Alcaldía del Ayuntamiento de Jerez de los Caballeros
ID del Órgano de Contratación	31029030146663
Estado de la Licitación	Evaluación
Objeto del contrato	Suministro de zahorra artificial de primario para reparación de caminos
Financiación UE	No hay financiación con fondos de la UE
Presupuesto base de licitación sin impuestos	25.102,08 Euros
Valor estimado del contrato:	25.102,08 Euros
Tipo de Contrato:	Suministros
Código CPV	14210000-Grava, arena, piedras machacadas y agregados., 14212300-Piedra partida y machacada.
Lugar de Ejecución	España - Badajoz
Sistema de contratación	No aplica
Procedimiento de contratación	Abierto simplificado

Figura 2.3: Captura de la página de detalles del expediente.

2.3.3 Canales de extracción de información de la PLACSP

Toda la información publicada en PLACSP es pública y accesible sin necesidad de registro para cualquier persona. Los datos se pueden consultar directamente desde el espacio de publicaciones en la propia plataforma o acceder a ellos de diferentes formas mediante los canales que esta pone a disposición. Tras estudiarlos, estos métodos presentan diversas limitaciones que dificultan el acceso y la gestión de los datos. A lo largo de este apartado se detallan las distintas formas de acceder a la información más allá de la interfaz web de la plataforma.

Portal de Datos Abiertos del Ministerio de Hacienda

El Portal de Datos Abiertos del Ministerio de Hacienda [9]es el espacio utilizado por este ministerio para poner a disposición de los ciudadanos los conjuntos de datos abiertos de los que dispone. Desde este portal, pueden descargarse distintos conjuntos de datos

en formato XML relacionados con la contratación pública. Estos datos están agrupados en distintas categorías y pueden ser reutilizados.

En el portal actualmente se ponen a disposición seis conjuntos de datos abiertos que agrupan la información de las licitaciones y los órganos de contratación. Estos seis conjuntos son: Licitaciones publicadas en los perfiles del contratante, licitaciones publicadas mediante mecanismos de agregación, contratos menores publicados, encargos a medios propios, consultas preliminares de mercado y perfiles de contratante de los órganos de contratación.

Dependiendo del conjunto de datos que se consulten, estos datos están agrupados por categoría y año, es por ello que es necesario descargar múltiples ficheros para obtener un conjunto de datos completo. Se descargan en formato ZIP y cada archivo contiene un conjunto de archivos Atom/XML que van almacenando entradas en orden cronológico, las cuales contienen los datos de contratación. Una vez descargados, estos datos pueden ser convertidos a formatos más accesibles utilizando la herramienta específica OpenPLACSP, paso prácticamente obligatorio para poder trabajar con los datos debido a su complejo formato.

Mediante este sistema, el ministerio cumple con su obligación de publicar los datos de forma procesable, pero realmente este sistema es muy complejo y nada accesible a la ciudadanía, puesto que requiere muchos conocimientos técnicos. Otra de las limitaciones que tiene este sistema es la organización de la información, esta está dividida por fechas y categorías, pero no por órganos, fondos o tipo de contrato, lo que dificulta el análisis de ciertos campos si no es mediante una descarga masiva.

OpenPLACSP

OpenPLACSP es la herramienta desarrollada por la Subdirección General de Coordinación de la Contratación Electrónica que permite visualizar los datos de contratación pública en forma de tablas adaptadas para ser legibles por humanos. Esta herramienta permite transformar los ficheros de datos abiertos publicados por el ministerio en hojas de cálculo con formato XLSX. Esta herramienta procesa los datos y permite filtrarlos por categorías predefinidas.

Esta herramienta requiere la descarga manual de los ficheros específicos del portal del Ministerio de Hacienda, ficheros que, como se ha comentado en el apartado anterior, están organizados de una forma poco intuitiva. Las categorías por las que permite filtrar la herramienta están predefinidas, limitación que hace que el análisis de estos datos sea limitado. Otra de sus limitaciones es que no permite la actualización automática de los datos ya transformados.

Servicios de Interacción Sistemática

La PLACSP ofrece un conjunto de servicios web que facilitan la gestión de la información sobre las licitaciones publicadas en el perfil del contratante directamente desde el sistema de gestión del órgano de contratación. Estos servicios solo son accesibles para aquellos órganos de contratación que tengan acceso al entorno de producción de PLACSP, por tanto, no es un servicio accesible al ciudadano.

Conclusión sobre los diferentes canales

El análisis de los diferentes canales de extracción de información de la PLACSP muestra varias limitaciones de las herramientas que dificultan un acceso sencillo y eficiente a los datos de contratación pública. La herramienta OpenPLACSP en combinación con los datos proporcionados por el Ministerio de Hacienda puede ser útil para usuarios con conocimientos sobre la materia y la herramienta, pero no es accesible a la ciudadanía y tiene limitaciones que hacen que el análisis sea limitado, como la necesidad descarga y actualización manual o no poder hacer un filtrado antes de la descarga.

Capítulo 3

Desarrollo del proyecto

A lo largo de este capítulo se explican los pasos que se han llevado a cabo en el desarrollo del proyecto. A efectos de documentar el proceso se ha utilizado como referencia el modelo de metodología *'Agile'* [20]. Durante este capítulo se mencionarán diferentes fases (sprints) con su respectiva definición, especificación, tareas y objetivos.

3.1 Sprint 1: Investigación sobre las tecnologías a utilizar

Definición

En este primer sprint, se busca conocer las herramientas de extracción de datos con el fin de elegir la que más se adapta al proyecto. Debe realizarse una investigación sobre las herramientas para automatizar la interacción con el navegador y poder seleccionar los datos claves. También se debe profundizar en la estructura de la plataforma de contratación para definir qué datos se desean extraer.

Objetivos

1. Conocer la estructura de la Plataforma de Contratación del Sector Público y **definir los datos** que se desean extraer de la misma.

2. Investigar sobre las **herramientas de automatización** de los navegadores para definir la que será utilizada en el desarrollo del proyecto.

3. Crear un **código de prueba** que interaccione con la web y extraiga datos de la misma.

Estructura de la PLACSP y datos a extraer

En esta primera fase el proyecto era relevante tener un conocimiento en profundidad de la plataforma de la que se desea extraer la información y definir cuáles eran los datos que se querían extraer para el proyecto. Para ello se realizó un mapeo intensivo con el fin de entender qué información se publicaba en las pestañas de perfil de contratante y de licitaciones. También se pusieron a prueba los distintos formularios de búsqueda de información. El formulario de búsqueda de licitaciones tiene bastantes limitaciones, puesto que interpreta los caracteres de forma literal y la información publicada no está estandarizada, lo que hace que algunos expedientes concretos sean difícilmente identificables.

En lo referido a los expedientes de licitación, están listados dentro de la página de licitaciones con la información principal que lo identifica y mostrando su estado actual, a esta información se le denomina cabecera. Cada expediente cuenta con una página propia en la que se encuentra la información detallada del mismo. Una de las dificultades encontradas es que los nombres de expedientes en la mayoría de los órganos de contratación no siguen una estructura clara ni un patrón, por tanto, la búsqueda por nombre se hace prácticamente imposible.

Dentro de la página del expediente se encuentran una serie de campos dedicados a proporcionar información detallada sobre la oferta publicada, su estado y el proceso de adjudicación. Esta información está medianamente estructurada y en todos los expedientes aparece de la misma forma. También se puede observar un apartado con los documentos relacionados con la oferta y que varían según el expediente. Estos documentos no siempre están presentes y no tienen un formato común, muchos son documentos escaneados, otros PDF digitalizados y otros documentos HTML con información no estructurada y más enlaces en su interior.

Análisis de canales oficiales de publicación de datos PLACSP

En este trabajo previo de conocer la plataforma y los datos de contratación pública, se han encontrado varias herramientas y conjuntos de datos publicados por las fuentes oficiales. Entre ellas se encuentra el Portal de Datos Abiertos del Ministerio de Hacienda, en el que se publican conjuntos de datos en formato Atom y la aplicación OpenPLACSP, que permite transformar estos datos a un formado de tablas. También se encontraron varios servicios de interacción sistemática restringidos solo para los órganos de contratación, lo que limita su acceso al público general.

Todas estas herramientas presentaron limitaciones en cuanto a accesibilidad para los ciudadanos, puesto que los formatos son complejos, los datos están fragmentados y agrupados de forma poco intuitiva. La explotación de los mismos no es sencilla.

Herramientas para interaccionar con los navegadores

Durante el desarrollo de este sprint se realizaron varias pruebas centradas en investigar y examinar las diferentes herramientas que permiten automatizar la interacción con los navegadores. El objetivo principal fue identificar la más adecuada para interactuar con el navegador y los formularios con el fin de extraer los datos necesarios. A continuación se detallan las herramientas evaluadas y el proceso seguido para tomar la decisión.

- BeautifulSoup: [3] Se probó esta herramienta por su capacidad y popularidad en tareas de análisis y extracción de datos, sobre todo con archivos HTML y XML. Es una gran herramienta al combinarla con solicitudes HTTP utilizando la biblioteca requests, pero no ofrece la robustez necesaria para interactuar dinámicamente con sitios web.

- Scrapy:[21] Scrapy es un framework para la extracción de datos web a través de spiders. Es una herramienta muy utilizada en grandes proyectos de scraping, pero su curva de aprendizaje es elevada y no dispone de soporte nativo para la interacción dinámica con JavaScript, es por ellos que se descartó en este proceso de selección.

-Pupeteer:[16] Esta es una herramienta desarrollada por Google para automatizar su navegador. Está basada en Node.js y proporciona una API de alto nivel para controlar Chrome. Cuenta con una gran capacidad de interacción con sitios web dinámicos, pero requiere algo de aprendizaje en torno a Node.js y resultó complejo integrarla con otros componentes del proyecto. Además, la limitación de poder trabajar solo en Chrome hizo que no se considerase como primera opción entre las herramientas probadas.

- Selenium:[22] Selenium destacó como una opción muy completa para la automatización de navegadores, puesto que cuenta con el soporte para múltiples lenguajes de programación y una gran capacidad para interactuar con elementos web, gestionar ventanas emergentes y lanzar pruebas automáticas. En las pruebas mostró simplicidad y consistencia a la hora de ejecutar scripts en navegadores reales y acceder a datos concretos de los mismos. Su compatibilidad con diferentes navegadores y la gran cantidad de información y comunidad de la que dispone hicieron Selenium una opción interesante dentro de estas pruebas.

Códigos de prueba

Para poner a prueba las distintas herramientas encontradas y empezar a conocer el sitio web del que se extraen los datos en el proyecto, en esta fase se realizaron algunos scripts de prueba con los que se interacciona de forma sencilla con la plataforma de contratación. La misión que estos programas tenían era abrir un perfil del contratante mediante un enlace y acceder a su pestaña de licitaciones.

BeautifulSoup y Scrapy se descartaron inicialmente debido a su limitación para trabajar con interacciones complejas en páginas web dinámicas, a pesar de que BeautifulSoup era

una herramienta con la que se había trabajado previamente.

Puppeteer es una herramienta capaz de trabajar con páginas web dinámicas y utilizada en proyectos similares de extracción de datos encontrados en GitHub. El desarrollo de este código de prueba fue complejo, puesto que según las recomendaciones encontradas en su documentación, debía ir acompañado del uso de la biblioteca Async, desconocida hasta el momento de las pruebas. Estas dificultades, problemas con la configuración del navegador y la limitación de que solo se podía usar la herramienta con Chrome, hicieron que tras tener el código funcionando se descartase seguir trabajando con esta herramienta.

A la hora de desarrollar el código de prueba con Selenium, fue sencillo acceder a un navegador e interaccionar con el mismo puesto que el componente WebDriver de la herramienta está bien documentado dentro de la página de la misma con varios ejemplos. Fue rápido lograr un código que interactuase con la plataforma de contratación, puesto que, al ser una herramienta consolidada, tiene mucha documentación y ejemplos al respecto. Localizar los elementos concretos y ejecutar acciones como clics o introducir campos en los formularios no fue complejo.

Conclusiones

A lo largo de este primer sprint se ha trabajado en torno a los cimientos del módulo de extracción de datos, que permite obtener la información de la plataforma de contratación con la que se trabaja en los siguientes módulos del proyecto. Tras dedicar tiempo a descubrir la PLACSP, se concluye que no es una web cómoda para navegar. A pesar de estar diseñada para tener interacciones dinámicas, los formularios de búsqueda de información, las paginaciones con un número muy pequeño y no modificable de entradas y la limitación en cuanto a volver a páginas navegadas anteriormente dentro de la misma, hacen que consultar grandes volúmenes de datos de forma manual sea verdaderamente complejo y pesado.

En cuanto a la información publicada sobre cada expediente, los datos publicados en la página de detalles del expediente es suficiente para el proyecto. En esta se recogen los datos de la oferta, su estado actual, el procedimiento de adjudicación y la información sobre los adjudicatarios, datos suficientes para el análisis que posteriormente se desea hacer. Se decide **no recoger los documentos adjuntos** de cada expediente, puesto que no tienen la información estructurada ni estandarizada y puede suponer un gran coste de recursos trabajar sobre ellos.

Tras realizar un análisis en profundidad sobre las herramientas existentes para realizar trabajos de extracción de datos e interacción automática con navegadores web, se ha seleccionado **Selenium** como la más apropiada para este proyecto. La implementación del código de prueba fue sencilla, su curva de aprendizaje no es muy elevada gracias a la cantidad de documentación a disposición en su página oficial. Su robustez y compatibilidad con diferentes navegadores ha sido un elemento clave a la hora de tomar la decisión, puesto que era interesante que el producto final pudiese adaptarse a distintos navegadores con

facilidad. Selenium cuenta con diversas formas de localizar elementos dentro de las páginas, es consistente trabajando con webs dinámicas y eficaz a la hora de rellenar formularios e interaccionar con los elementos.

Concluyendo este primer sprint quedan definidos los datos que se desean extraer de la PLACSP y la herramienta que se usará para ello.

3.2 Sprint 2: Recogida y almacenamiento de los datos

En este segundo sprint se busca desarrollar un código funcional que permita extraer todos los datos de todos los expedientes de un órgano de contratación especifico y almacenarlos en un CSV. Además, se debe investigar sobre la limpieza y estructuración de datos en la disciplina del análisis de datos.

Objetivos

1. Desarrollar un **script utilizando Selenium** que extraiga todos los datos de los expedientes de un órgano de contratación específico.
2. Almacenar los datos en un **fichero CSV**.
3. Investigar sobre los **tipos de datos** en los campos de información de los expedientes y cómo **limpiar y estructurar** estos datos para trabajar con ellos.

Desarrollo del Script

Continuando con el código del sprint anterior, se desarrolló un script que permite acceder a la PLACSP, navegar hasta la página de licitaciones de un órgano de contratación pasado como parámetro e iterar sobre la tabla de expedientes obteniendo una lista con todos los nombres de expedientes y su URL asociada. Posteriormente, se amplió este código accediendo a cada una de las URLs asociadas y descargando todos los datos organizados en tablas de cada expediente. Estos datos conforme se extraían se copiaban en un diccionario para cada expediente y se pegaban en un fichero CSV.

El script fue diseñado para interactuar con elementos dinámicos de la página, enfrentándose a la pestaña de cookies y los largos tiempos de carga de la página. También fue un reto la paginación de los expedientes, puesto que no todas las páginas tenían el mismo número de expedientes y algunas de ellas estaban vacías.

Se realizaron pruebas para asegurar que el script funciona correctamente y extrae todos los datos necesarios, comprobando el número de datos extraídos y su coincidencia con los originales. También se probó con distintos órganos de contratación y diferentes casuísticas de error. A lo largo de estas pruebas se fue creando un código de testing para futuras modificaciones del script.

Investigación sobre estructuración y limpieza de datos

Para comenzar con esta tarea se estudiaron los campos de información que se deseaban descargar de la plataforma de contratación con el fin de conocer qué tipos de datos sería interesante manejar. Entre los campos recogidos, se decide trabajar con fechas, texto, y valores numéricos enteros y flotantes.

La fase de tratamiento de los datos era algo completamente desconocido en este momento, por ello fue fundamental incluir un objetivo en esta fase del proyecto para entender qué era necesario y cómo otros proyectos relacionados con los datos trabajan esta fase. Para ello se han estudiado técnicas de limpieza de datos, incluyendo la eliminación de duplicados, el manejo de valores faltantes y la normalización de las cadenas de textos.

En este proceso de búsqueda se encontró Pydantic, una biblioteca de Python diseñada para la validación de datos. Esta permite definir modelos de datos con tipos estrictos y realizar validaciones automáticas de los mismos al cargarlos. Se estudió la forma de introducir esta herramienta en el proyecto con el fin de tener los datos estructurados y un sistema de validación de tipos robusto. Se marcó como objetivo para el próximo sprint.

Conclusiones

Al finalizar este sprint se superaron con éxito los objetivos marcados. Se desarrolló un script funcional que extrae todos los datos deseados de los expedientes de un órgano de contratación específico y los almacena en un fichero CSV. Además, se adquirió un conocimiento sólido sobre técnicas de limpieza y estructuración de datos crudos y se descubrieron herramientas para la validación de tipos de datos en Python que en próximos pasos deberán usarse para implementar las estructuras de datos.

3.3 Sprint 3: Almacenamiento, limpieza y actualización de los datos

En este punto del proyecto es necesario empezar a trabajar en la organización de los datos para explotarlos. Se deben estudiar las distintas vías para almacenar los datos. La información publicada de los expedientes en la plataforma no es estática, cambia a lo largo del tiempo para actualizar el estado del mismo, además, cada cierto tiempo los órganos publican nuevas ofertas de contratación que deben ser recogidas.

Objetivos

1. Crear una **estructura de tablas** útil para la explotación de los datos.

2. **Integrar Pydantic** en el proceso de normalización de los datos, definiendo clases y

realizando validaciones de datos en la entrada de la base de datos limpia.

3. Desarrollar un programa en Python para **procesar los datos** crudos, limpiarlos y almacenarlos de forma estructurada en una **base de datos SQLite.**

4. Implementar un primer código para probar la **explotación de los datos**.

¿Por qué una "Base de datos Limpia"?

La necesidad de tener un almacenamiento de datos estructurado y coherente para la posterior explotación y análisis de los datos, ha conducido a la decisión de implementar una base de datos limpia para almacenar los datos. Utilizar archivos CSV para el almacenamiento de datos crudos es una buena opción, pero carece de la robustez necesaria para su explotación. Al implementar una base de datos limpia y estructurada en SQLite se consigue:

- Integridad de datos: asegurando que los datos almacenados cumplan con ciertos criterios.
- Facilidad de actualización: con una base de datos bien estructurada es más sencillo implementar mecanismos de actualización y un historial de cambios.
- Eficiencia en la explotación: Una base de datos relacional permite realizar consultas complejas de manera eficiente, algo complicado de lograr con archivos CSV.
- Escalabilidad: A medida que el proyecto crece, las bases de datos escalan mejor que los archivos CSV y tienen mejor rendimiento a mayor volumen de datos.

Implementación de la base de datos

Se estudiaron distintas opciones a la hora de organizar la base de datos construida sobre SQLite, finalmente se decidió crear tres tablas conectadas entre ellas, **expediente**, **órgano de contratación** y **adjudicatario**.

Inicialmente, se decidió utilizar SQLAlchemy como ORM para definir las clases que construyen la base de datos, pero tras indagar en otras opciones se encontró SQLModel, una biblioteca que ofrece un ORM implementado sobre SQLAlchemy integrando la validación de tipos de Pydantic, una combinación idónea para la implementación de la base de datos limpia. Al definir las clases con SQLModel, gracias a Pydantic, estas clases obligan a definir los tipos de datos y aseguran que se cumplen los esquemas esperados antes de ser insertados en la base de datos. Además, ofrece la opción de desarrollar *validator* que son funciones que se ejecutan antes de comprobar el tipo de dato y permite hacer modificaciones previas en estos. Se ha usado esta funcionalidad para eliminar los campos vacíos. Finalmente, se trabajó en un código que recorre los archivos CSV para introducir los datos en la base de datos. La ejecución del mismo comienza extrayendo del archivo de datos

crudos una lista de los órganos de contratación sin repeticiones y crea un registro para cada uno de ellos en la tabla de los órganos. De la misma forma rellena la tabla de adjudicatarios. Por último, itera por todos los expedientes y los añade a la tabla de expedientes controlando el formato de los datos.

Explotación de los datos

En este punto, el proyecto ya cuenta con una base de datos con información suficiente como para empezar a probar la explotación de la misma. Utilizando las peticiones que ofrece SQAlchemy, se han diseñado varias funciones a modo de API para extraer información.

- **Top 10 expedientes:** Función que devuelve los 10 expedientes más caros de un órgano de contratación.
- **Lista de órganos de contratación:** Devuelve una lista de todos los órganos de contratación que hay en la base de datos.
- **Lista de adjudicatarios:** Devuelve una lista de todos los adjudicatarios que hay en la base de datos.
- **Lista de expedientes por órgano:** Lista con todos los expedientes de un órgano de contratación.
- **Lista de expedientes por adjudicatario:** Lista de todos los expedientes adjudicados a una empresa en concreto.
- **Expedientes desde X fecha:** Devuelve una lista con la información de todos los expedientes de un órgano de contratación desde la fecha indicada.

Conclusiones

Este tercer sprint logró los objetivos marcados, mejorando significativamente el almacenamiento, limpieza y estructuración de los datos con el fin de poder ser explotados. Se ha conseguido un programa que transforma los datos crudos de un CSV a una base de datos con tipos controlados y estructurada en tablas. Además, se ha desarrollado un código que permite hacer una primera explotación de los datos.

3.4 Sprint 4: Robustez de los módulos y visualización de datos

En este punto, el proyecto cuenta con dos módulos en funcionamiento, uno que interacciona con la web para extraer datos y almacenarlos en un archivo CSV y otro que procesa

los datos del CSV y los almacena de forma estructurada en una base de datos. Tras ponerlos a prueba, estos módulos funcionan, pero no son robustos frente a cambios y peticiones compleja, es por ello que en este sprint deben ser revisados y mejorados para que gestione bien los errores, sea capaz de enfrentarse a situaciones excepcionales y sean más rápidos. Por otro lado, es un buen momento para investigar sobre las diferentes herramientas de visualización existentes con el fin de definir las que se usarán en el proyecto y hacer un primer esquema de la misma.

Objetivos

1. Mejorar la **consistencia** del módulo de extracción de datos.
2. Implementar el **manejo de errores y excepciones**, así como crear test que pongan a prueba el código.
3. Investigar sobre las distintas herramientas de **visualización y publicación de datos** y definir las que se van a usar en el proyecto.

Creación de la base de datos crudos

En este punto del proyecto surge la necesidad de empezar a trabajar con un mayor volumen de datos, para ello la extracción debe ser más eficiente y el almacenamiento en ficheros CSV se vuelve una limitación. Es por ello que se decide cambiar el CSV de datos crudos por una base de datos SQLite en la que almacenar estos datos crudos. Esta tendrá la misma función que el archivo anteriormente implementado, solo que mejora la eficiencia y permite realizar consultas y otras acciones.

Para crear esta base de datos se ha utilizado SQLite y SQLModel como ORM. En ella solo se ha definido una tabla llamada *ExpedienteRaw* que contiene todos los campos de información que puede tener un expediente, todos ellos con el tipo de dato opcional str, tal como se extrae del navegador.

Optimización del módulo de extracción de datos

Por el momento, el código que extrae los datos y los almacena, funciona para un solo órgano de contratación y finaliza con error frecuentemente debido a que no es consistente. Se decide crear un módulo con funciones para la extracción de los datos con el fin de mejorar la escalabilidad y limpieza del proyecto, creando en este proceso una nueva versión del programa.

En este módulo se implementan varias funciones para la apertura del navegador y la lectura de parámetros de entrada, esta última con la biblioteca ***argparse*** de Python. Se implementa una función llamada collect_licitaciones que se encarga de recopilar el nombre y los enlaces de todas las licitaciones de un órgano, iterando por la tabla de expedientes y

sus páginas. La función RecopilaExpediente recibe como argumento, entre otras, la lista de nombres de expedientes y su enlace e itera sobre ella, abriendo una ventana del navegador para cada expediente, accediendo así a la página que contiene la información detallada del mismo. Por último, esta llama a ExtractTableInfo que localiza y extrae la información que se desea recoger de cada expediente.

En todas estas funciones se implementa un correcto tratamiento de excepciones y se añade la biblioteca logging para la gestión de mensajes de error y advertencia. Además, se añade una latencia aleatoria entre 1 y 5 segundos para cada apertura de pestaña en el navegador con el fin de simular una interacción humana y evitar baneos. Este módulo se utiliza desde un código cliente que realiza la petición de expedientes de un órgano en concreto y almacena la información proporcionada en la base de datos crudos.

Para probar y demostrar la consistencia del código se implementaron una serie de test utilizando la biblioteca *unittest* para probar distintos casos de extracción de datos y diferentes errores a la hora de cargar los mismos en la base de datos limpia.

Herramientas de visualización

Las herramientas de visualización eran un campo prácticamente desconocido al inicio del proyecto, es por ello que se dedicó bastante tiempo a conocer las diferentes opciones y realizar pruebas. Se investigaron herramientas como Tableau, Power BI, Google Data Studio y bibliotecas de Python como Matplotlib y Plotly.

Tras ver varios proyectos de ejemplo y realizar algunas pruebas, se decidió utilizar Plotly combinado con Dash, ya que conjuntamente permiten la creación de interfaces interactivas.

- **Plotly** es una herramienta que permite crear gráficos interactivos, cuenta con un conjunto de funciones sencillas y bien documentadas con una gran cantidad de ejemplos en su página oficial que hacen que su uso sea bastante modular.

- **Dash** es una herramienta desarrollada por los desarrolladores de Plotly, es un framework de Python para la creación de aplicaciones web analíticas. Permite construir dashboards interactivos y visualizaciones de datos con facilidad, utilizando componentes reactivos que se actualizan dinámicamente en respuesta a la interacción del usuario.

Una vez definida la herramienta y conociendo las posibilidades de las mismas, se empezó a diseñar un esquema provisional de la herramienta de visualización. En esta debía aparecer una pestaña donde poder consultar los expedientes de un órgano ordenados por precio y por fecha. También se implementaría un sistema de búsqueda de expedientes por adjudicatario y un ranking de los adjudicatarios que más dinero han ganado.

Conclusiones

Al finalizar esta fase se consiguió una mejora significativa de la robustez y eficiencia de los módulos de extracción y procesamiento de los datos y se empezó a plantear el módulo de visualización.

3.5 Sprint 5: Implementación de un sistema de actualización de datos y una primera visualización

En este punto del proyecto, es necesario implementar un sistema de actualización de los datos, puesto que hasta ahora cada vez que se quieren actualizar se sobreescriben y esto supone una gran carga para la web que aloja los datos y para los tiempos del programa. Por otro lado, una vez definida las herramientas para visualización de los datos, era el momento de implementar una primera visualización de los mismos.

Objetivos

1. Implementar un sistema de **actualización de datos**.
2. **Reducir el número de interacciones** con la página web a la hora de actualizar los datos.
3. Crear una primera **visualización web**.

Sistema de actualización de datos

Para implementar un sistema que permita actualizar la información de los expedientes sin tener que sobreescribir todos los datos, se ha desarrollado una mejora del código del módulo de extracción de datos. A continuación se describe el flujo de funcionamiento que sigue el módulo, pero primero es importante recordar el concepto cabecera del expediente.

La cabecera del expediente es la información que se muestra sobre un expediente en el listado de licitaciones. Esta información sirve de resumen e identificación del expediente. Para esta implementación, se ha añadido un campo de fecha y hora. La cabecera que se usa en este módulo tiene la siguiente estructura:

[URL,nombre,timetrack,estado]

- **URL:** La dirección de la página web donde se encuentran todos los detalles del expediente.

- **Nombre:** El nombre del expediente.
- **Timetrack:** La marcha de fecha y hora en la que se ha consultado la información.
- **Estado:** El estado actual del expediente

Esta nueva implementación del módulo de extracción de datos se basa en el concepto de cabeceras para la actualización de los datos. También se ha añadido un campo llamado **reciente** a la tabla *ExpedienteRaw*. Este campo toma valores booleanos y sirve para indicar si ese registro es el más reciente o no. El flujo de funcionamiento es el siguiente:

1. Se extraen de la web **todas las cabeceras** de los órganos de contratación.
2. Se **comprueba si la información de las cabeceras está actualizada** en la base de datos crudos. Esto se hace consultando si el nombre del expediente que aparece en la cabecera existe, si es así, se comprueba si el estado más reciente del expediente es el mismo. Si los estados no coinciden o el expediente no se encuentra en la base de datos, se guarda en una **lista de expedientes que hay que actualizar**.
3. Una vez que se han comprobado todas las cabeceras, se itera sobre la lista de expedientes que hay que actualizar y se accede a la **información completa** de los mismos utilizando la URL que se encuentra en el primer campo de la cabecera.
4. Una vez en la página de detalles del expediente, se recoge toda la información relevante y **se carga en la base de datos** marcando el antiguo registro, si existe, como "**no reciente**".

Con este nuevo sistema de actualización de datos, se mantiene una base de datos crudos con la última información de todos los expedientes y un histórico de los estados anteriores. Este sistema de cabeceras mejora significativamente la eficiencia y los tiempos del módulo, puesto que evita tener que entrar en la página de detalles de todos los expedientes, consultando solo los que necesitan ser actualizados.

Visualización de los datos

Para crear la visualización web, se han utilizado las herramientas seleccionadas en el sprint anterior, Dash y Plotly. De estas se han aprovechado las capacidades que ofrece en torno a la representación de datos y la interactividad. En este código primero se estableció una conexión con la base de datos limpia para poder hacer consultas. Se definieron funciones para cada consulta, basadas en las que se crearon en el script de explotación de los datos. Posteriormente, se definió la estructura HTML de la página diseñando dos pestañas, una para el top 10 y otra para buscar por nombre de adjudicatario. Por último, se diseñaron los callbacks relativos a estas pestañas para poder cargar la información y dar soporte a la interactividad.

Conclusiones

En este sprint se han logrado avances que han permitido escalar la funcionalidad del proyecto y mejorar su eficiencia. Se implementó el sistema de actualización de datos que reduce las interacciones con la página web y se introducen las cabeceras con el mismo fin. En la parte de la visualización, se ha logrado una primera visualización web local en la que se puede trabajar con dos pestañas, una con un ranking de expedientes y otra con un listado de expedientes vinculados a un cierto adjudicatario.

3.6 Sprint 6: Visualización final, publicación del portal y automatización del proceso

Este sprint se plantea como el último en esta fase de desarrollo. En este punto se trabaja por tener una versión definitiva de la visualización y se pretende publicar el portal para que esté accesible a cualquier usuario. Por otro lado, se deben ultimar los detalles de la automatización que permite que los tres módulos funcionen de forma sincronizada y periódica.

Objetivos

1. **Finalizar la implementación** de la visualización de los datos en el portal.
2. **Publicar el portal** con una base de datos para que pueda acceder cualquier usuario desde el navegador.
3. Crear un **sistema automatizado** que permita que los módulos funcionen de forma sincronizada con el fin de mantener un sistema de datos actualizado y visualizable.

Portal de Visualización de Datos

Para implementar la visualización se ha trabajado sobre el código desarrollado en el sprint anterior. El código del portal se divide en tres partes:

- **Funciones de consulta:** Son las funciones que, utilizando SQLModel como ORM, hacen consultas SQL a la base de datos limpia para extraer la información que se visualiza en el portal. Entre las funciones se encuentran: top de expedientes con importe de adjudicación más elevado, detalles de un expediente concreto, lista de adjudicatarios, lista de órganos, lista de expedientes por adjudicatario, importe total recibido por adjudicatario. Se ha intentado diseñar un conjunto de consultas lo más sólido y concreto posible para no sobrecargar la base de datos y evitar consultas innecesarias que ralenticen el funcionamiento de la página.

- **Estructura de la página:** En esta parte del código se diseña la estructura (layout) de la aplicación utilizando HTML y una plantilla de Bootstrap para la interfaz. Se han creado 5 pestañas, una que sirve de introducción en la que se muestra una guía de usuario y el resto, una para cada tipo de visualización de datos. En este punto, los nombres de las pestañas son: introducción, top expedientes más caros, últimos contratos, buscar por adjudicatario y total recibido por adjudicatario.

- **Callbacks:** En la tercera y última parte del código se encuentran los callbacks. Los callbacks son las funciones que permiten la interactividad de la página y los encargados de crear y actualizar los gráficos, realizar la llamada a las consultas y gestionar la interactividad de la página. Se han implementado 7, uno para cada pestaña en la que se visualizan datos y tres para la visualización de los detalles del expediente.

-

Durante el proceso de desarrollo final de este módulo, se ha consultado la galería de gráficos publicada en la página oficial de Plotly. Esta galería contiene una gran cantidad de ejemplos de gráficos y visualizaciones con las que experimentar con el fin de encontrar la visualización más interesante. También permite utilizar de forma modular el código de estas visualizaciones, algo que ha facilitado el desarrollo de esta herramienta. Tras varias pruebas, con el fin de utilizar un formato básico y familiar para todos los usuarios, se han usado principalmente tablas y gráficos de barras.

Publicación del portal en la web

Debido a que la base de datos no está construida y publicada en un servidor accesible desde la red, se ha realizado un trabajo de búsqueda de información con el fin de encontrar una alternativa para poder publicar el portal. Una de las opciones fue migrar la base de datos SQLlite a sistemas como MariaDB, pero debido al poco tiempo y falta de conocimientos al respecto, se decidió seguir buscando alternativas. Tras conversaciones con el tutor del TFG se decide utilizar *PythonAnyware*, esta es una plataforma de alojamiento en la nube diseñada para alojar aplicaciones web basadas en Python. Permite desplegar aplicaciones desde entornos virtuales, facilitando la conexión y gestión con las bases de datos.

A continuación se enumeran los pasos que se siguieron para publicar la web desde esta plataforma:

1. Tras crear la cuenta en PythonAnyware, se configura un nuevo entorno virtual dentro de una de las máquinas que se ponen a disposición en la plataforma.

2. Se sube el código de la aplicación junto a los archivos necesarios al sistema de ficheros que tiene la plataforma. Entre estos archivos se encuentra la base de datos limpios.

3. Se instalaron las dependencias del proyecto utilizando el archivo *requirements.txt*.

4. Siguiendo el tutorial de la página, se configuró el archivo WSGI que se encuentra en la zona de aplicaciones web para que este apunte al código del portal.

5. También tuvo que modificarse la ruta de la base de datos del código de la aplicación y hacer unos cuantos ajustes.

6. Tras solucionar varios errores que se exponen a continuación y reiniciar la aplicación web desde el panel de control, esta se publicó correctamente.

A lo largo de este proceso se han encontrado varias complicaciones que se han ido solventando sobre la marcha gracias a la comunidad de PythonAnyware y sus manuales. Entre otros, se encontraron problemas con la instalación de módulos que se solventaron con instalaciones manuales y cambios de versiones. otro de los problemas fue el error de conexión con el puerto 8050, que estaba configurado para la aplicación local y hubo que cambiar los ajustes.

Creación de un sistema automático para el funcionamiento de los módulos

Con el fin de automatizar la ejecución de los módulos y ofrecer distintas configuraciones para poner en valor la modularidad de los mismos, se ha trabajado en crear scripts que automaticen la ejecución e interacción de los distintos módulos. Para ello se comenzó implementando scripts de bash, que era el lenguaje conocido para este tipo de tareas. Con el fin de utilizar un solo lenguaje de programación para dar una apariencia unificada, se dejaron a un lado los scripts de bash y finalmente se implementaron en Python. Para ello se ha usado la biblioteca *subprocess* [27] que permite ejecutar nuevos procesos desde un script de Python y argparse [2], una biblioteca que facilita la creación y lectura de líneas de comando amigables. Finalmente, se han implementado 2 scripts que ejecutan de forma combinada varios módulos, uno que ejecuta de forma combinada los módulos de extracción y almacenamiento y otro que ejecuta los tres módulos.

Conclusiones

A lo largo de este último sprint, se ha logrado tener una versión estable y con diversas formas de visualizar los datos de contratación de forma interactiva y con una interfaz sencilla. También se ha logrado publicar la aplicación web mediante la plataforma PythonAnyware de forma que cualquier usuario puede acceder a ella desde el navegador. Por último, se ha conseguido automatizar la ejecución de los módulos y la interacción entre ellos.

Capítulo 4

Resultados del proyecto

A lo largo de este capítulo se presentan los resultados de este Trabajo de Fin de Grado. El proyecto se ha centrado en el desarrollo de un **conjunto de herramientas modulares** que permiten la extracción, estructuración y visualización de datos de contratación pública extraídos de la Plataforma de Contratación del Sector Público. Estas herramientas pueden ser utilizadas de forma independiente o en conjunto, proporcionando flexibilidad según las necesidades e intereses del usuario. Se han desarrollado **dos prototipos** utilizando estos módulos con el fin de demostrar la utilidad y versatilidad del sistema desarrollado y mostrar ejemplos de usos del mismo.

Con el fin de presentar los resultados de forma ordenada, al inicio de este capítulo se presenta una visión general de las herramientas y prototipos implementados. En las siguientes secciones (Secciones 4.1, 4.2) se describen estos prototipos con el fin de dar a conocer los usos finales de las herramientas desarrolladas. Tras esto, en la sección 4.3 se detalla la implementación del conjunto de herramientas y posteriormente, la del prototipo Portal de Visualización de Datos (4.4).

En el siguiente diagrama (Figura 4.1) se pueden observar los tres módulos, extracción, almacenamiento y visualización de datos interaccionando con el resto de elementos del sistema, la PLACSP, las bases de datos y la visualización web. Además, se puede ver un esquema de la herramienta completa y de cada uno de los prototipos implementados.

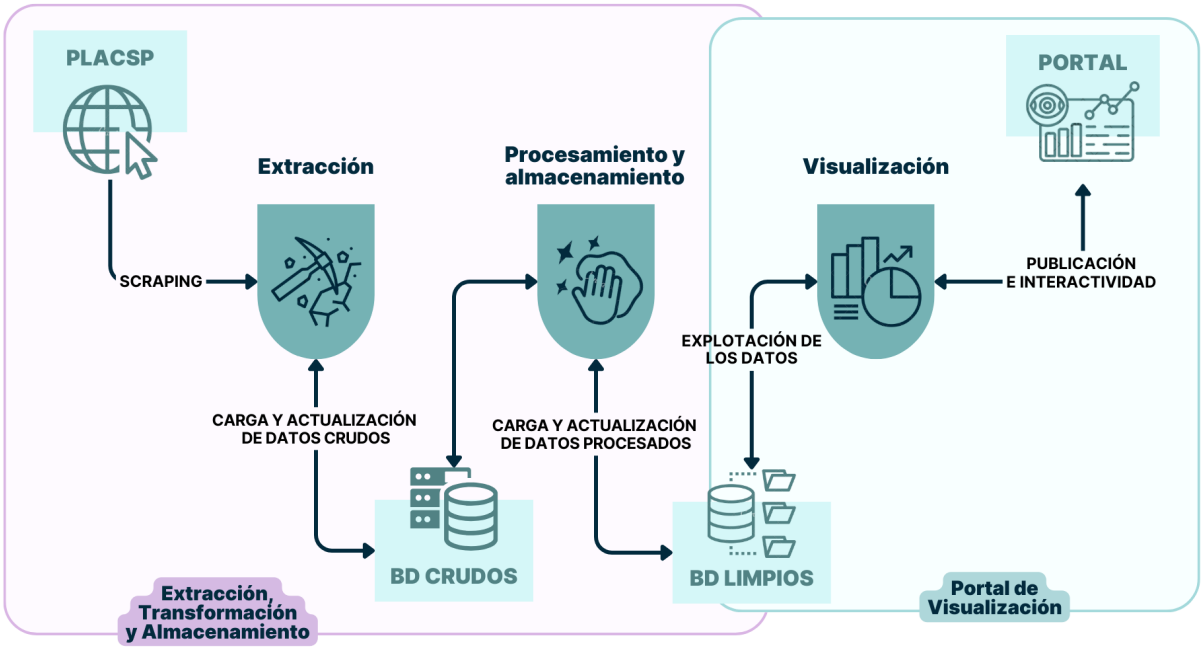


Figura 4.1: Diagrama descriptivo del flujo del programa. En verde se observan los tres módulos y en azul los sistemas con los que interacciona. En rojo se observa el prototipo Extracción y Almacenamiento y en amarillo el Portal de Visualización

Como se menciona anteriormente, se han implementado dos prototipos para mostrar las capacidades del conjunto de herramientas desarrollado, cada uno destinado a un tipo de usuario y utilizando herramientas diferentes. A continuación se hace una breve presentación de los mismos.

- **Portal de Visualización de Datos de Contratación:** Esta herramienta permite a los usuarios visualizar los datos de licitaciones públicas de forma interactiva e intuitiva. Está destinada a ciudadanos sin conocimientos técnicos con interés por la contratación pública y el análisis de estos datos.

- **Extracción, Transformación y Almacenamiento de Datos de Contratación:** Este prototipo se centra en la extracción de datos mediante técnicas de web scraping y su estructuración y almacenamiento en una base de datos limpia y actualizada. Su propósito es ofrecer una base de datos con capacidad de actualización preparada para ser explotada, bien usando el módulo de extracción desarrollado o con otras técnicas de explotación de datos.

4.1 Descripción del Portal de Visualización de Datos de Contratación

En esta sección se presenta el **Portal de Visualización de Datos de Contratación**. Este portal es una aplicación web organizada por pestañas que cuenta con diversas formas de mostrar y organizar datos de contratación pública. El prototipo se ha implementado utilizando el **módulo de visualización de datos** del proyecto y que es capaz de trabajar con cualquier base de datos de contratación pública que siga la misma estructura. Para mostrar la funcionalidad de esta herramienta, se han utilizado los datos de contratación pública relacionados con Jerez de la Frontera, pero podría implementarse con cualquier otra base de datos que siga el mismo formato.

El usuario final de esta herramienta es cualquier ciudadano o ciudadana que tenga interés por explorar los datos de contratación pública sin necesidad de tener conocimientos previos de tecnologías específicas. Además, todos los conceptos necesarios sobre contratación pública están descritos en la guía con el fin de acercar a las personas que desconocen estos procesos de contratación a la información visualizada. En la siguiente figura (Figura 4.2) se puede observar la página de introducción del Portal de Visualización de Datos de Contratación del municipio Jerez de la Frontera.

Figura 4.2: Página de introducción del Portal de Visualización. Esta página contiene la guía de uso y el diccionario de términos. En la parte superior se observan las pestañas de navegación.

4.1.1 Acceso y estructura de la aplicación

El Portal de Visualización es una aplicación web accesible desde cualquier navegador mediante su URL. Al acceder a la aplicación, se presentará una página de inicio (Figura 4.2) con una pestaña de introducción que proporciona una visión general de los conceptos claves para entender el contenido de la misma y una guía de uso.

Para navegar por la aplicación web se utilizan las **cinco pestañas** que aparecen en la parte superior. En cada una de ellas se encuentra una forma diferente de visualizar los datos, con el fin de que los gráficos no sean demasiado complejos y ofrecer diferentes formas de explotación de los mismos. A continuación se describen cada una de las pestañas:

Introducción

Esta es la pestaña de presentación del Portal de Visualización (Figura 4.2). En ella se presenta la herramienta y se aporta un diccionario de términos con el fin de acercar al usuario a los conceptos de contratación pública que aparecen a lo largo de la aplicación. También se muestra una guía de uso del Portal, en la que se explica la navegación dentro del mismo y el contenido de cada pestaña.

Ranking de Expedientes por Importe

En esta pestaña se muestra un gráfico de barras horizontales para mostrar un ranking por importe de las licitaciones de un órgano de contratación en concreto. En las pestañas de selección se puede seleccionar el órgano del que se desea visualizar la información y el número de expedientes que aparecerán en la tabla. Tras rellenar estos campos, el gráfico aparece automáticamente.

En el eje vertical aparecen los nombres con los que están registrados los expedientes y en el horizontal el importe total de la adjudicación. El gráfico es **interactivo**, al situar el ratón sobre una de las barras muestra una etiqueta con el nombre y el importe exacto, al hacer clic, se abre en la parte inferior de la página una tabla con todos los detalles del expediente. En las figuras que se muestran a continuación se puede observar el gráfico del ranking (Figura 4.3) y un ejemplo de la tabla de detalles del expediente (Figura 4.4).

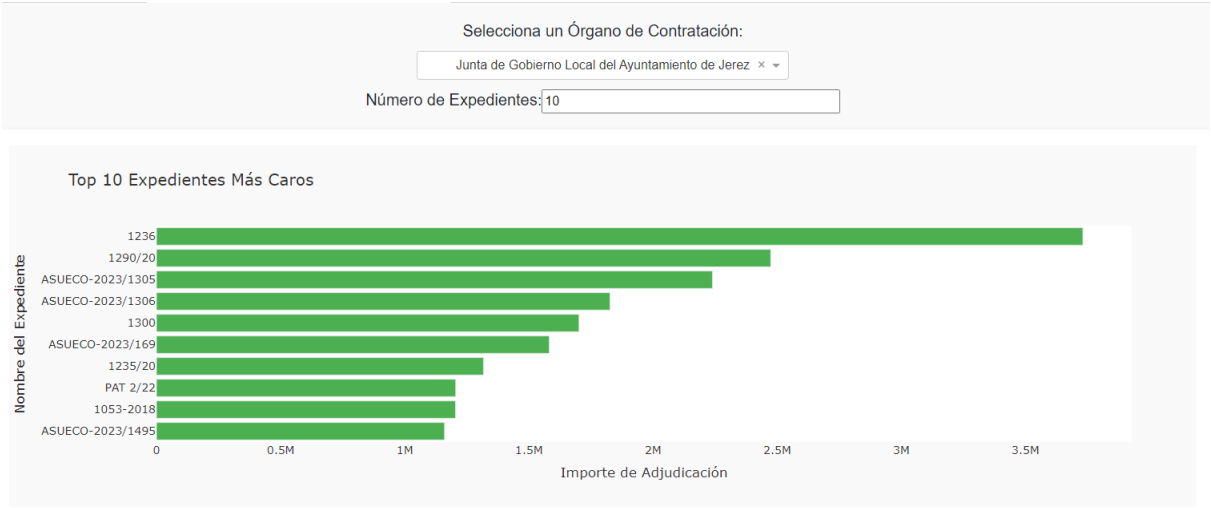


Figura 4.3: Ranking de expedientes ordenados por importe.

Detalles del Expediente: ASUECO-2023/1305

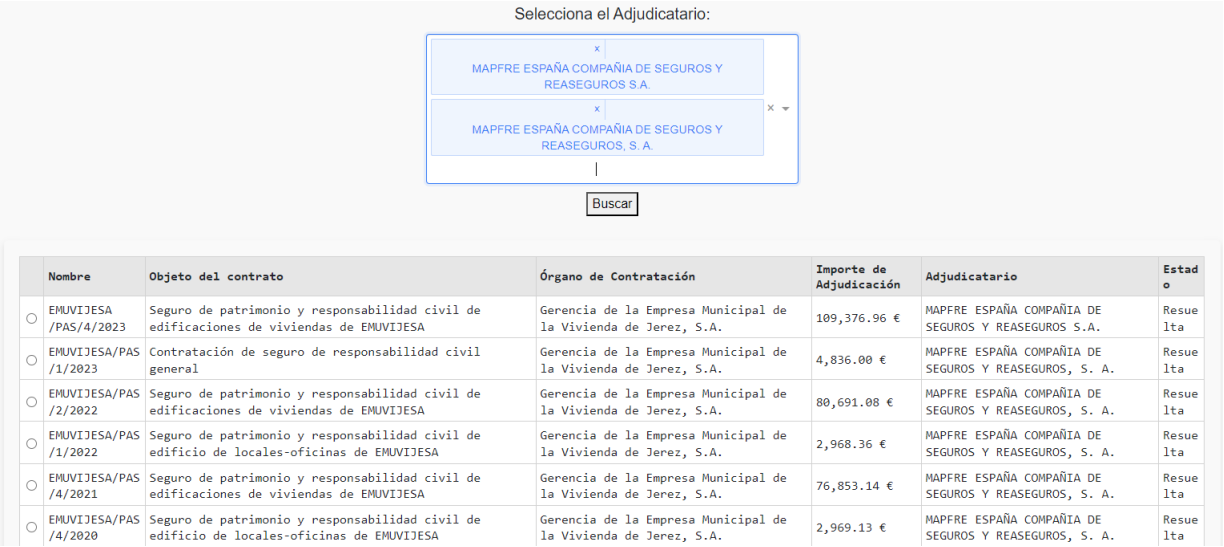
Campo	Valor
Nombre del Expediente	ASUECO-2023/1305
Estado	Resuelta
Importe de Adjudicación	2,239,190.40 €
Adjudicatario	Vivendio Sostenibilidad Energética, S.L.- Añil Servicios, Ingeniería y Obras SAU
Fecha de Fin de Solicitud	
Órgano de Contratación	Junta de Gobierno Local del Ayuntamiento de Jerez
Objeto del Contrato	Redacción de proyecto constructivo y ejecución de las obras de rehabilitación del C.E.I.P. Elio Antonio de Nebrija,
Financiación UE	Asociado al Plan de Recuperación, Transformación y Resiliencia
Presupuesto sin Impuestos	2,239,629.48 €
Valor Estimado	2,239,629.48 €
Tipo de Contrato	Obras
Código CPV	45000000-Trabajos de construcción., 71242000-Elaboración de proyectos y diseños, presupuestos.
Lugar de Ejecución	España - Cádiz
Sistema de Contratación	No aplica
Procedimiento de Contratación	Abierto simplificado
Tipo de Tramitación	Ordinaria
Método de Presentación	Electrónica
Fecha Fin de Oferta	
Número de Licitadores	2
Resultado	Formalizado
Enlace:	Ver expediente en la Plataforma

Figura 4.4: Tabla de detalles del expediente seleccionado.

Expedientes por Adjudicatario

En esta pestaña se muestran todos los expedientes de uno o varios adjudicatarios. En el cuadro de selección que aparece en la parte superior, aparece un desplegable con todos nombres de adjudicatarios que aparecen en la base de datos. En este cuadro se pueden seleccionar **uno o varios nombres a la vez** con el fin de poder agrupar los expedientes de aquellos adjudicatarios que tienen varios nombres para una misma empresa u organización.

Una vez seleccionado el o los adjudicatarios que se desean visualizar, el portal muestra automáticamente una tabla con todas las licitaciones adjudicadas (Figura 4.5). En esta tabla aparece algo de información sobre cada expediente y estos están paginados de diez en diez. Al pinchar en el selector que aparece a la izquierda de cada nombre de expediente, se despliega en la parte inferior de la página una tabla con todos los detalles del expediente, con el mismo formato que la de la figura (Figura 4.4).



	Nombre	Objeto del contrato	Órgano de Contratación	Importe de Adjudicación	Adjudicatario	Estado
○	EMUVIJESA /PAS/4/2023	Seguro de patrimonio y responsabilidad civil de edificaciones de viviendas de EMUVIJESA	Gerencia de la Empresa Municipal de la Vivienda de Jerez, S.A.	109,376.96 €	MAPFRE ESPAÑA COMPAÑIA DE SEGUROS Y REASEGUROS S.A.	Resuelta
○	EMUVIJESA/PAS /1/2023	Contratación de seguro de responsabilidad civil general	Gerencia de la Empresa Municipal de la Vivienda de Jerez, S.A.	4,836.00 €	MAPFRE ESPAÑA COMPAÑIA DE SEGUROS Y REASEGUROS, S. A.	Resuelta
○	EMUVIJESA/PAS /2/2022	Seguro de patrimonio y responsabilidad civil de edificaciones de viviendas de EMUVIJESA	Gerencia de la Empresa Municipal de la Vivienda de Jerez, S.A.	80,691.08 €	MAPFRE ESPAÑA COMPAÑIA DE SEGUROS Y REASEGUROS, S. A.	Resuelta
○	EMUVIJESA/PAS /1/2022	Seguro de patrimonio y responsabilidad civil de edificio de locales-oficinas de EMUVIJESA	Gerencia de la Empresa Municipal de la Vivienda de Jerez, S.A.	2,968.36 €	MAPFRE ESPAÑA COMPAÑIA DE SEGUROS Y REASEGUROS, S. A.	Resuelta
○	EMUVIJESA/PAS /4/2021	Seguro de patrimonio y responsabilidad civil de edificaciones de viviendas de EMUVIJESA	Gerencia de la Empresa Municipal de la Vivienda de Jerez, S.A.	76,853.14 €	MAPFRE ESPAÑA COMPAÑIA DE SEGUROS Y REASEGUROS, S. A.	Resuelta
○	EMUVIJESA/PAS /4/2020	Seguro de patrimonio y responsabilidad civil de edificio de locales-oficinas de EMUVIJESA	Gerencia de la Empresa Municipal de la Vivienda de Jerez, S.A.	2,969.13 €	MAPFRE ESPAÑA COMPAÑIA DE SEGUROS Y REASEGUROS, S. A.	Resuelta

Figura 4.5: Listado con el resumen de los expedientes encontrados en la búsqueda por adjudicatario del portal

Total Recibido por Adjudicatario

En esta pestaña se muestran los adjudicatarios que más dinero reciben mediante licitaciones de un órgano de contratación en concreto. En los cuadros de selección que aparecen en la parte superior se puede seleccionar el órgano de contratación del que se desea visualizar la información y el número de adjudicatarios que se desea ver. Una vez seleccionados, la aplicación muestra automáticamente un gráfico de barras horizontales con el nombre de los adjudicatarios ordenados de mayor a menor según el importe total que reciben de dicho órgano. En la siguiente figura (Figura 4.6) se muestra un ranking de los 7 adjudicatarios que más dinero reciben en licitaciones del Ayuntamiento de Jerez. Como se observa en la figura, el eje vertical del gráfico de barras se muestran los nombres de los adjudicatarios ordenados por importe de mayor a menos y en el horizontal se muestra el importe total.

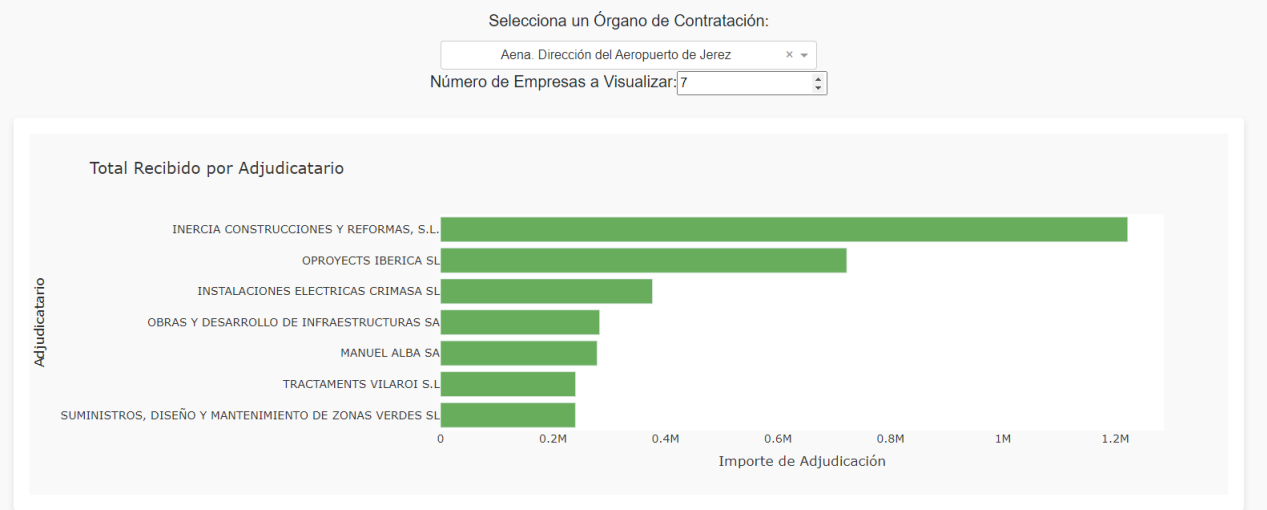


Figura 4.6: Ranking de adjudicatarios según el importe total recibido.

4.2 Descripción de la Herramienta de Extracción, Transformación y Almacenamiento de Datos de Contratación

En esta sección se presenta el segundo de los prototipos desarrollados, un sistema automático que extrae, transforma y almacena datos de contratación. Esta herramienta genera una base de datos estructurada y validada que se actualiza periódicamente extrayendo los datos publicados en la PLACSP. Este prototipo integra, como elementos principales de su código, los módulos de extracción y almacenamiento de datos implementados en el Trabajo Fin de Grado.

El usuario final de esta herramienta es un ciudadano con interés en la contratación pública y ciertos conocimientos de desarrollo y bases de datos. El producto generado es una **base de datos lista para ser explotada**, cargada con toda la información sobre licitaciones de los órganos de contratación que el usuario elija. Para mostrar el funcionamiento de la herramienta en esta memoria, se han seleccionado todos los órganos de contratación del municipio de Jerez de la Frontera con el fin de tener una base de datos de ejemplo con un gran número de expedientes y varios órganos de contratación diferentes.

Acceso y funcionamiento de la herramienta

Esta herramienta está diseñada para ser fácil de utilizar y personalizar por aquellos usuarios con un cierto conocimiento en desarrollo y bases de datos. Está disponible en GitHub y se puede desplegar y ejecutar en cualquier entorno que soporte Python y bases de datos QSL. Para ello, se debe clonar el repositorio e instalar las dependencias del mismo, así como configurar el entorno.

La herramienta está compuesta por **dos módulos**, que trabajan de forma conjunta para extraer, procesar y almacenar los datos con un sistema de actualización continua. Además, utiliza **dos bases de datos** para almacenar la información, una de ellas con datos en crudo y otra organizada en tablas y con control de tipo de datos.

La ejecución de la herramienta está completamente automatizada mediante un script de Python. A la hora de ejecutar el script, se pueden introducir diferentes parámetros para configurar su modo de ejecución. En la siguiente figura (Figura 4.7) se observa la configuración de estos parámetros.

```
PS C:\Users\migue\Desktop\TFG\Código para entregar 06_07> python .\extraccion_automatica.py -h
usage: extraccion_automatica.py [-h] [-l LINKS [LINKS ...]] [-p PATIENCE] [-hd]

options:
  -h, --help            show this help message and exit
  -l LINKS [LINKS ...], --links LINKS [LINKS ...]
                        Enlaces de los perfiles de contratación, por defecto carga los perfiles del municipio Jerez de la Frontera
  -p PATIENCE, --patience PATIENCE
                        Paciencia del scrapper en segundos para las esperas implicitas (default=5)
  -hd, --headless       Activa el flag para que desaparezca la ventana de navegador durante el scrapping
```

Figura 4.7: Configuración de los argumentos de entrada del Script

Como se observa en la configuración mostrada al ejecutar el script con el flag *–help*, este tiene tres argumentos opcionales.

- *- l | - - links*: Permite al usuario introducir uno o varios enlaces a perfiles de contratación de la PLACSP. El programa realizará la extracción de datos de cada uno de estos perfiles. Si no se activa este flag, se descargarán por defecto los datos de contratación de los perfiles de contratación relacionados con el municipio de Jerez de la Frontera.

- *- p | - - patience*: Este flag permite configurar el tiempo de espera que tiene el programa mientras busca un elemento en la web. Por defecto son 5 segundos.

- *- hd | - - headless*: Cuando se activa este flag, la búsqueda y extracción de los datos se ejecuta sin mostrar la ventana del navegador.

Una vez ejecutado, el módulo de extracción de datos se encarga de extraer información de la Plataforma de Contratación del Sector Público y almacenarla en una base de datos cruda. Primero, el módulo accede a las URLs de los perfiles de contratación y extrae datos básicos, conocidos como cabeceras, que contienen la información mínima necesaria para identificar cada expediente. Luego, verifica si estas cabeceras ya existen en la base de datos cruda. Si encuentra nuevas cabeceras o cambios en el estado de los expedientes, las almacena en una lista para ser actualizadas. Posteriormente, extrae los datos completos de los expedientes que necesitan ser actualizados y los almacena en un diccionario. Finalmente, los datos detallados se actualizan en la base de datos cruda, marcando los registros como recientes para indicar que la información está actualizada.

Tras esto, entra en funcionamiento el segundo de los módulos. El módulo de procesamiento de datos transforma y estandariza la información extraída para almacenarla en una base de datos limpia, lista para ser explotada de la forma que el usuario decida. Este módulo comienza conectándose tanto a la base de datos cruda, que contiene datos sin procesar, como a la base de datos limpia, donde se almacenan los datos estructurados. Luego, compara los identificadores de los órganos de contratación en la base de datos cruda con los de la base de datos limpia y añade los nuevos órganos de contratación y adjudicatarios. A continuación, selecciona los expedientes recientes de la base de datos cruda y los actualiza o inserta en la base de datos limpia.

Ambas bases de datos se crean automáticamente si no existen. La base de datos crudos solo acepta información en cadena de texto, que es el formato en el que se extraen los datos de la PLACSP. La base de datos estructurada está organizada en tres tablas, Organo-Contratación, Expediente, Adjudicatario, relacionadas entre ellas (Para más información se puede consultar la sección 2.2.4). Esta base de datos tiene un sistema de **control de tipos** y estandarización que garantiza que los datos que entren en la base de datos tengan el formato deseado para poder ser explotados adecuadamente.

4.3 Implementación del conjunto de herramientas

En esta sección se detalla la arquitectura y los componentes técnicos del proyecto, destacando el desarrollo de tres herramientas independientes: el **módulo de extracción, el módulo de procesamiento y almacenamiento y el módulo de visualización**. Aunque estas herramientas pueden funcionar de manera independiente, también pueden integrarse para trabajar conjuntamente en pares o como un sistema completo. A continuación, una visión detallada de cada módulo y su interacción con los demás.

4.3.1 Módulo de descarga de datos

Este primer módulo es el encargado de **extraer los datos** de la plataforma de Contratación del Sector Público, almacenar los datos crudos en una base de datos y llevar un control de las últimas versiones de los datos almacenados en la misma. Este módulo está diseñado para ejecutarse de manera cíclica y periódica, asegurando que la base de datos esté **siempre actualizada** con las últimas versiones de los datos. Funciona de forma autónoma y ofrece distintas opciones en cuanto a la descarga de datos. En la siguiente figura (Figura 4.8) se muestra el funcionamiento de este módulo.

Figura 4.8: Flujo del módulo de extracción de datos.

1. **Extracción de cabeceras:**

   En la primera etapa del proceso, denominada "Extracción de Cabeceras", el programa accede a una serie de URLs asociadas a los perfiles contratantes de los que se desea extraer la información. A través de un navegador, itera sobre estas URLs para acceder a la sección de licitaciones de cada perfil. Posteriormente, recorre el paginado

de cada sección para extraer la información de lo que en este proyecto se ha denominado la "cabecera" de cada expediente. Esta cabecera incluye los datos mínimos necesarios para identificar el expediente y su estado actual, los cuales se encuentran disponibles en la página de licitaciones sin necesidad de profundizar en los detalles de cada expediente. Como resultado, el programa tiene una lista con todas las cabeceras por cada órgano de contratación.

2. **Consulta a base sobre las cabeceras:** En esta fase, se comprueba si los nombres de expediente presentes en la lista de cabeceras se encuentran en la base de datos cruda. Si alguno de los expedientes no coincide con los registros de la base de datos, se añade a una lista de cabeceras que requieren información completa. Por otro lado, si los expedientes coinciden, se verifica el estado de la licitación. Si el estado es el mismo, no se realizan modificaciones. Sin embargo, si ha habido algún cambio en el estado de la licitación, se añade la cabecera correspondiente a una lista, indicando que es necesario buscar la información completa del expediente. Además, se utiliza un campo en la tabla de expedientes como flag para marcar si un expediente es reciente o no. Cuando el estado de un expediente no coincide con el de la cabecera, este flag se marca como False para indicar que se trata de información antigua. Como resultado de esta fase, el programa tiene una lista de cabeceras que hacen referencia a los expedientes que necesitan la información completa para ser actualizados en la base de datos.

3. **Información de los expedientes:** En esta fase el programa itera sobre cada registro de la lista de cabeceras que necesitan ser actualizadas, accediendo a la página en la que se encuentra la información detallada de cada expediente. Una vez en esta página, extrae los datos y los almacena en un diccionario. Este proceso culmina cuando se recopila la información de todos los expedientes referenciados en la lista de cabeceras.

4. **Actualización en la base de datos:** Una vez que el programa cuenta con la información completa de los expedientes que necesitan ser actualizados, accede a la base de datos y actualiza los registros correspondientes, marcando el flag de reciente como activo para indicar que se trata de información actualizada.

Ejecución

Para poner en marcha la funcionalidad de este módulo debe ejecutarse el script ***raw_db.py*** que se puede encontrar en el repositorio GitHub del proyecto. La ejecución de este módulo es configurable, admitiendo argumentos de entrada opcionales. Al ejecutar el script con el flag - - *help*, se muestra una guía de ejecución del script. Todos los flags que se muestran a continuación son opcionales.

- *- l | - - links*: Permite al usuario introducir uno o varios enlaces a perfiles de contratación de la PLACSP. El programa realizará la extracción de datos de cada uno de estos

perfiles. Si no se activa este flag, se descargarán por defecto los datos de contratación de los perfiles de contratación relacionados con el municipio de Jerez de la Frontera.

- *-p | - - patience*: Este flag permite configurar el tiempo de espera que tiene el programa mientras busca un elemento en la web. Por defecto son 5 segundos.
- *-hd | - - headless*: Cuando se activa este flag, la búsqueda y extracción de los datos se ejecuta sin mostrar la ventana del navegador.

4.3.2 Módulo de procesamiento de datos

El módulo de procesamiento de datos toma la información extraída y almacenada en la base de datos cruda, la transforma y estandariza, y luego la almacena en una base de datos limpia. Este módulo **asegura que los datos sean consistentes y listos para su análisis y visualización.** Al igual que el módulo de extracción, puede operar de forma independiente o en conjunto con los otros módulos, especialmente útil cuando se integra con los módulos de extracción y visualización para ofrecer datos actualizados y estructurados. El flujo de funcionamiento es el siguiente:

1. **Conexión con las bases de datos:** Comienza conectándose a las dos bases de datos con las que trabaja, la de datos crudos, en la que se encuentra la información extraída de la plataforma sin tratar, y la base de datos limpia, en la que se cargan los datos de manera estructurada y procesados con el fin de poder trabajar sobre ellos con garantías. Si esta segunda todavía no existe, se crea en este paso.
2. **Añadir órganos de contratación:** En este punto se seleccionan todos los identificadores de los órganos de contratación que hay en la base de datos cruda y se comparan con los existentes en la base de datos limpia, añadiendo a esta los que no están presentes. De la misma forma, añade los adjudicatarios que no están presentes en la base de datos limpia.
3. **Transformación y guardado de expedientes:** En esta fase se seleccionan los expedientes recientes de la base de datos cruda y se verifica si existen en la limpia. i un expediente existe y su versión en la base de datos cruda es más reciente, se actualiza en la base de datos limpia, si no existe, se inserta como un nuevo registro. Al terminar este proceso con todos los expedientes, cierra las sesiones de las bases de datos.

La base de datos limpia es el espacio final donde se almacenan los datos ya procesados y estandarizados. Consta de varias tablas relacionadas, entre ellas que almacenan los datos de manera organizada y estructurada. Las principales tablas son: **OrganoContratacion, Adjudicatario y Expediente.** Estas tablas están diseñadas para garantizar la integridad y consistencia de los datos y hacer más eficientes las consultas. En las siguientes tablas (Tablas 4.1, 4.2, 4.3) se muestran todos los campos que contienen cada una de las tablas de la base de datos implementada con una descripción del campo y su tipo de datos.

OrganoContratacion		
Campo	**Tipo de dato**	**Descripción**
nombre	str	Nombre del órgano de contratación
id	int	Id único del órgano de contratación (clave primaria)
nif	str	Número de Identificación Fiscal del órgano de contratación
url	str	URL del perfil del órgano de contratación

Tabla 4.1: Tabla OrganoContratacion

Adjudicatario		
Campo	**Tipo de dato**	**Descripción**
internal_id	int	Id único del adjudicatario (clave primaria)
nombre	str	Nombre del adjudicatario
nif	str	Número de Identificación Fiscal del adjudicatario
url	str	URL del perfil del adjudicatario
nombres_similares	str	Nombres similares asociados al adjudicatario
num_nombres_similares	int	Número de nombres similares

Tabla 4.2: Tabla Adjudicatario

Expediente		
Campo	**Tipo de dato**	**Descripción**
internal_id	int	Identificador único del expediente (clave primaria)
timetrack	datetime	Marca temporal de registro del expediente
nombre_exp	str	Nombre del expediente
route	str	Ruta o URL del expediente
id_organo	int	Id de su órgano de contratación (clave foránea)
organo_contratacion	str	Nombre del órgano de contratación
estado_lic	str	Estado de la licitación
objeto_contrato	str	Objeto del contrato
financiacion_UE	str	Indicación de financiación por la Unión Europea
presupuesto_sin_impuestos	float	Presupuesto sin incluir impuestos
valor_estimado	float	Valor estimado del contrato
tipo_contrato	str	Tipo de contrato
codigo_CPV	str	Código CPV
lugar_ejecucion	str	Lugar de ejecución del contrato
sistema_contratacion	str	Sistema de contratación utilizado
procedimiento	str	Procedimiento de adjudicación
tipo_tramitacion	str	Tipo de tramitación
metodo_presentacion	str	Método de presentación
fecha_fin_oferta	datetime	Fecha de fin de oferta
resultado	str	Resultado de la licitación
adjudicatario	str	Nombre del adjudicatario asociado (clave foránea)
n_licitadores	int	Número de licitadores
importe_adjudicacion	float	Importe de adjudicación
fecha_fin_solicitud	datetime	Fecha de fin de solicitud
url	str	URL del expediente

Tabla 4.3: Tabla Expediente

Ejecución

Para poner en marcha la funcionalidad de este módulo debe ejecutarse el script ***clean_db.py*** que se puede encontrar en el repositorio GitHub del proyecto. Para ejecutar este módulo es necesario contar con una base de datos crudos que siga el formato descrito anteriormente. Este script no es configurable mediante argumentos de entrada, puesto que la transformación y carga de los datos sigue una estructura fija.

4.3.3 Módulo de visualización

El módulo de visualización utiliza las bibliotecas Dash y Plotly para crear una **aplicación web interactiva**, permitiendo a los usuarios explorar y analizar los datos almacenados en la base de datos limpia. Este módulo está diseñado con el objetivo de fomentar la inter-

acción del usuario con los datos a través de gráficos y tablas dinámicas. Puede funcionar de manera autónoma, consultando directamente una base de datos ya existente, o en combinación con los módulos de extracción y procesamiento para ofrecer una solución completa de extracción, procesamiento y visualización de datos.

La aplicación está diseñada con un enfoque en la usabilidad y la interactividad. Está pensada para que cualquier usuario con conocimientos básicos sobre contratación pública pueda explorar y visualizar los datos de contratación de una forma sencilla. Se ha utilizado un tema de Bootstrap como plantilla y está estructurada en varias pestañas para organizar las diferentes funcionalidades.

Inicialización

Se importan y configuran las herramientas utilizadas para la aplicación web y para la visualización. Se inicia la conexión con la base de datos limpia y se configura el layout de la aplicación, incluyendo pestañas para las diferentes visualizaciones y secciones.

Consulta de datos

Se definen funciones para consultar datos específicos de la base de datos en función a las visualizaciones creadas en la aplicación web. Las funcionalidades principales de esta herramienta son:

- **Top expedientes más caros:** Permite seleccionar un órgano de contratación y recuperar los expedientes ordenados por el importe de adjudicación de mayor a menor. Esta función se implementa mediante una consulta SQL que filtra los expedientes del órgano de contratación seleccionado, selecciona el número de expedientes indicado y ordena los resultados por importe de adjudicación en orden descendente.

- **Últimos contratos:** Permite seleccionar un órgano de contratación y un estado del expediente para ver los contratos más recientes del mismo. Esta función se implementa mediante una consulta SQL, que filtra los expedientes por el órgano de contratación y el estado del expediente, y ordena los resultados por la fecha de fin de oferta en orden descendente. Al seleccionar un contrato de la tabla, una consulta adicional obtiene los detalles completos del contrato seleccionado.

- **Búsqueda por adjudicatario:** Permite buscar todos los expedientes de uno o varios adjudicatarios al mismo tiempo. Esta función se implementa mediante una consulta SQL, que filtra los expedientes por los nombres de los adjudicatarios seleccionados. La tabla resultante muestra el nombre del expediente, objeto del contrato, órgano de contratación, importe de adjudicación, adjudicatario y estado del expediente. Al seleccionar un contrato de la tabla, una consulta adicional recupera los detalles completos del expediente.

- **Total recibido por adjudicatario:** Permite seleccionar un órgano de contratación y visualizar el total recibido por cada adjudicatario. Esta función se implementa mediante una consulta SQL, que agrupa los expedientes por adjudicatario, calcula el importe total recibido por cada adjudicatario y ordena los resultados de mayor a menor. La tabla muestra el nombre de los adjudicatarios y el importe total recibido por cada uno.

Interactividad mediante callbacks

Se implementan callbacks para actualizar los gráficos y tablas en función de la interacción del usuario con la aplicación. Estos callbacks permiten la actualización en tiempo real de los datos presentados, haciendo que la interacción con los datos sea rápida y dinámica. Están diseñados para soportar diversas interacciones, como la selección de elementos dentro de una tabla o un gráfico para mostrar los detalles del expediente o para realizar las consultas necesarias cuando los campos seleccionables se modifican.

Ejecución

Para poner en marcha la funcionalidad de este módulo debe ejecutarse el script ***visualization.py*** que se puede encontrar en el repositorio GitHub del proyecto. La ejecución de este módulo permite introducir como argumento opcional la base de datos que se desea explotar. Si no se introduce este argumento en la ejecución, se consulta por defecto la base de datos *clean_database.db.*

4.4 Implementación Portal de Visualización de Datos de Contratación

Como se ha comentado en la introducción de este capítulo, se han desarrollado tres módulos independientes que pueden funcionar conjuntamente. El Portal de Visualización de Datos de Contratación es un prototipo desarrollado para mostrar la funcionalidad del módulo de visualización de datos de forma independiente.

Esta aplicación web tiene **tres componentes principales**: una base de datos estructurada, el módulo de visualización de datos y una herramienta para el despliegue de la aplicación web. Nos vamos a centrar en los dos primeros, puesto que la forma de desplegar la aplicación web queda completamente a elección la persona que la implemente, en este prototipo se ha utilizado PythonAnywhere.

4.4.1 Uso del módulo de visualización

El módulo de visualización de datos es una herramienta desarrollada íntegramente para este TFG, que implementa la **interfaz web** y los **gráficos interactivos** para visualizar los datos. Además, se encarga de realizar las consultas necesarias a la base de datos para mostrar los datos en la aplicación. Utiliza las bibliotecas de Python Dash y Plotly para la creación de la interfaz y los gráficos, dando soporte a la interactividad de la web haciendo uso del sistema de callbacks de estas librerías. El diseño de la aplicación está basado en una plantilla de Bootstrap y utiliza HTML para su estructura. Esta herramienta está diseñada para funcionar de forma independiente, solo necesita una base de datos que siga la estructura descrita en el punto siguiente.

En la sección 4.3.3 se explica el funcionamiento en detalle el módulo de visualización al completo.

4.4.2 La base de datos

Este módulo realiza una serie de consultas a la base de datos que contiene la información sobre contratación pública. Esta base de datos puede ser generada manualmente, con herramientas de elaboración propia o con los otros módulos desarrollados en este proyecto. La base de datos está compuesta de tres tablas relacionadas entre ellas: **OrganoContratación**, que recoge la información propia del perfil de contratante; **Expediente**, que almacena todos los campos de información que puede tener un expediente según la PLACSP; **Adjudicatario**, que recoge la información de los adjudicatarios. En la siguiente figura (Figura 5.1) se muestra el diagrama de relación entre las tablas de la base de datos.

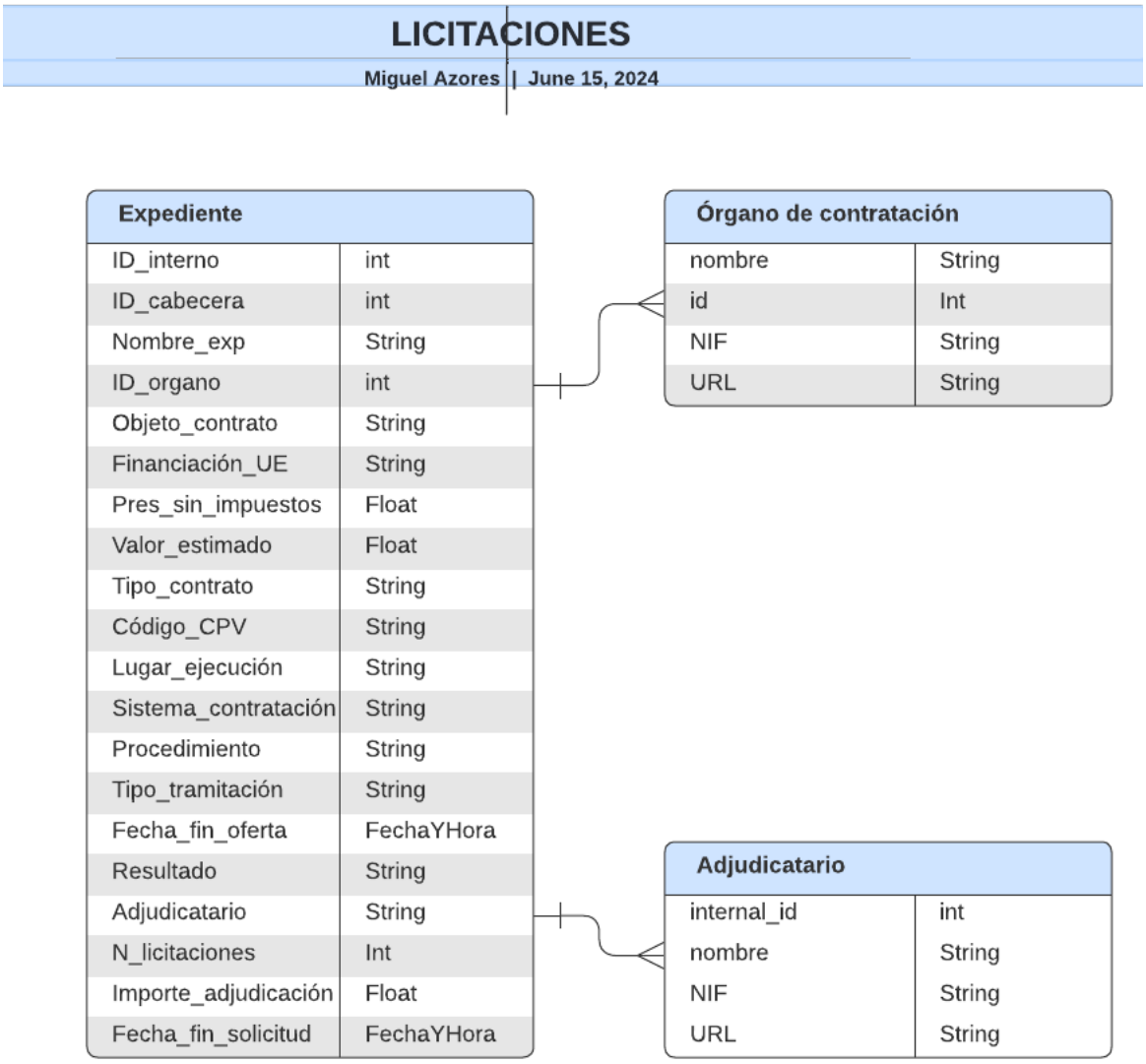


Figura 4.9: Diagrama de relación entre las tablas de la base de datos limpios.

Como se puede observar en el diagrama (Figura 5.1), un órgano de contratación y un adjudicatario pueden tener múltiples expedientes, y un expediente puede estar asociado a un solo órgano de contratación y, en esta fase del proyecto, solo a un adjudicatario.

Capítulo 5

Evaluación y experimentación

A lo largo de este capítulo se describen las distintas pruebas y evaluaciones por las que ha pasado la herramienta en las diferentes fases de desarrollo y desde distintos puntos de vista.

5.1 Testing durante el desarrollo del código

A lo largo del proceso de desarrollo de la herramienta, se han ido implementando diferentes pruebas con el fin de probar la funcionalidad y estabilidad del código. Para ello se comenzó realizando pruebas manuales y posteriormente se implementaron una serie de tests unitarios que han permitido la detección de errores durante el proceso de desarrollo.

Tras crear la base de datos estructurados por primera vez, se creó un conjunto de tests con el fin de poner a prueba las validaciones y funciones de transformación para que ningún dato con formato inadecuado entrase en la base de datos. Trabajando con la base de datos, también se creó un conjunto de test con pruebas unitarias para validar la creación y actualización de las entidades y sus relaciones, validando también el manejo de errores y las operaciones de escritura y lectura. También se crearon un conjunto de test con el fin de probar la herramienta de extracción, accediendo a PLACSP y verificando los datos extraídos.

5.2 Caso de uso para los usuarios finales del Portal de Visualización de Datos

La herramienta de visualización de datos tiene como usuario objetivo a ciudadanos y ciudadanas sin conocimientos de programación con interés en los datos de contratación pública. Al ser una aplicación desarrollada para uso público, se ha decidido diseñar un caso de uso para evaluar la experiencia del usuario interaccionando con el portal, buscando

información y comprendiendo los datos.

Este estudio sobre el funcionamiento de la herramienta se ha realizado mediante una encuesta anónima recogida en un formulario de Google que no recoge datos que permitan identificar al usuario encuestado, con el fin de cumplir con los requisitos de protección de datos, se ha implementado una casilla de consentimiento informado del manejo de datos.

El formulario comienza con cuatro preguntas que tienen como objetivo realizar una breve recogida de datos demográficos sobre el encuestado a fin de poder hacer un análisis de los resultados según diferentes perfiles. Tras esto, se presentan tres ejercicios prácticos que los usuarios deben realizar utilizando el portal de visualización. Tras estos ejercicios hay unas preguntas que recogen el resultado y evalúan su dificultad. Además, se ofrece posibilidad de cronometrar los ejercicios con el fin de profundizar en la experiencia de usuario. La sección de ejercicios prácticos finaliza con un ejercicio final opcional algo más complejo que los anteriores, con el fin de poner a prueba la herramienta con usuarios más exigentes. Por último, tras la sección de ejercicios prácticos, se realizan cuatro cuestiones con el fin de conocer la visión general del usuario respecto al portal de visualización.

En total han completado los ejercicios prácticos y han respondido al formulario 12 personas. A continuación se muestra una relación de las preguntas presentadas en el formulario y un resumen de las respuestas ofrecidas por los usuarios.

5.2.1 Información demográfica de los usuarios

En la siguiente tabla 5.1 se muestra un listado de las cuatro peguntas realizadas en esta sección y un resumen de las respuestas dadas por los usuarios.

Resumen de los Participantes	
Categoría	**Descripción**
Edad	Edades comprendidas entre 21 y 62 años, con una media de 35 años.
Sector en el que trabaja	Pertenecen a una amplia variedad de sectores, incluyendo estudiantes, asesoría, mantenimiento de aeronaves y periodistas.
¿Cuánto sabe sobre contratación pública? (1 nada, 5 mucho)	La mayoría de los usuarios indicaron un nivel de conocimiento entre 1 y 3.
¿Ha usado alguna vez la Plataforma de Contratación del Sector Público?	Solo un 25% de los participantes ha usado alguna vez.

Tabla 5.1: Relación de preguntas y respuestas sobre la demografía de los participantes.

Analizando estos datos demográficos observamos que el perfil de los participantes es variado y, a pesar de ser una muestra pequeña debido a la limitación en cuanto a tiempo y recursos, es lo suficientemente dispersa como para dar una visión general.

5.2.2 Tareas de Búsqueda

En el formulario se plantean diversas tareas de búsqueda dentro del portal con el fin de evaluar la accesibilidad, la comprensión y la facilidad de uso del mismo. Esta sección de preguntas se presenta en el formulario con el siguiente texto:

"Enlace: https://miguelazores.pythonanywhere.com/ Se recomienda abrir el portal de transparencia desde un ordenador para mejor experiencia del usuario.

Ahora va a tener que abrir el enlace que se adjunta con el formulario. En ese enlace encontrará el portal de transparencia desarrollado por Miguel Azores Picón para el Trabajo Fin de Grado. En este portal de transparencia se recogen todos los datos relacionados con las licitaciones de Jerez de la Frontera. Lee la introducción detenidamente, en ella se explican los conceptos básicos sobre contratación pública y el funcionamiento de la página. A continuación, realiza las tareas que se describen en el formulario y responde a las preguntas.

De manera opcional, tras leer el enunciado de cada tarea, cronometre cuanto tarda en realizar la búsqueda y párelo justo cuando termine, antes de rellenar las respuestas del formulario."

Consulte el nombre de los tres expedientes más caros del Ayuntamiento de Jerez.

Enunciado: *Sitúese en la pestaña de introducción y realice las búsquedas necesarias para visualizar el nombre de los tres expedientes más caros del Ayuntamiento de Jerez. Si va a cronometrar el tiempo, recuerde activar el cronómetro cuando esté en la pestaña de introducción y desactivarlo al finalizar la búsqueda.*

1. **Indique los nombres de los expedientes encontrados:** El 75% de los participantes anotó correctamente el nombre de los tres expedientes de mayor importe del Ayuntamiento de Jerez.

2. **¿Cómo de complejo le ha resultado encontrar los datos?:** En una escala del 1 al 5, donde 1 es muy fácil y 5 es muy difícil, el 91,7% de los participantes puntuó como muy fácil la tarea descrita.

3. **Si se ha cronometrado, anote el tiempo exacto que ha tardado en realizar la búsqueda:** Los tiempos de búsqueda de los usuarios variaron, pero fueron generalmente cortos, con una media de 31 segundos y una media de 41 segundos.

Consulte el nombre y el "presupuesto sin impuestos" del expediente con estado "publicada" del órgano de contratación Aena. Dirección del Aeropuerto de Jerez.

Enunciado: *Sitúese en la pestaña de introducción y realice las búsquedas necesarias para visualizar el nombre del último expediente publicado por la Dirección del Aeropuerto de Jerez. Pincha en este expediente para visualizar todos sus detalles y encontrar el presupuesto sin impuestos. Si va a cronometrar el tiempo, recuerde activar el cronómetro cuando esté en la pestaña de introducción y desactivarlo al finalizar la búsqueda.*

1. **Indique el nombre del expediente y el importe del presupuesto sin impuestos.** En esta segunda tarea, el 50% de los participantes respondieron correctamente al nombre y al presupuesto. Entre los que no, algunos anotaron bien el nombre del expediente, pero no el importe correcto.
2. **¿Cómo de complejo le ha resultado encontrar los datos?:** Observando las respuestas de la siguiente tabla (Tabla 5.2), se observa que la mayoría de los participantes consideran que fue sencillo encontrar los resultados.

Nivel de Complejidad	Número de Respuestas	Porcentaje
Muy fácil	2	16,7%
Fácil	7	58,3%
Medio	0	0%
Difícil	2	16,7%
Muy difícil	1	8,3%

Tabla 5.2: Dificultad percibida por los usuarios a la hora de completar la tarea 2.

3. **Si se ha cronometrado, anote el tiempo exacto que ha tardado en realizar la búsqueda:** Los tiempos de búsqueda de los usuarios variaron, pero fueron generalmente cortos, con una media de 32 segundos minutos y una media de 57 segundos.

Consulte el nombre de los tres adjudicatarios que más dinero reciben de la Presidencia del Circuito de Jerez.

Enunciado: *Sitúese en la pestaña de introducción y realice las búsquedas necesarias para visualizar el nombre de las tres empresas que más dinero han recibido de la Presidencia del Circuito de Jerez. Si va a cronometrar el tiempo, recuerde activar el cronómetro cuando esté en la pestaña de introducción y desactivarlo al finalizar la búsqueda.*

1. **Indique el nombre de los tres adjudicatarios encontrados:** En esta tercera tarea, el 83% de los participantes respondió correctamente.
2. **¿Cómo de complejo le ha resultado encontrar los datos?:** Observando las respuestas de la siguiente tabla (Tabla 5.3), se observa que la mayoría de los participantes consideran que fue sencillo encontrar los resultados.

Nivel de Complejidad	Número de Respuestas	Porcentaje
Muy fácil	8	66,7%
Fácil	2	16,7%
Medio	0	0%
Difícil	1	8,3%
Muy difícil	1	8,3%

Tabla 5.3: Dificultad percibida por los usuarios a la hora de completar la tarea 3.

3. **Si se ha cronometrado, anote el tiempo exacto que ha tardado en realizar la búsqueda:** Los tiempos de búsqueda de los usuarios variaron, pero fueron generalmente cortos, con una media de 78 segundos minutos y una media de 88 segundos.

Tarea opcional

Enunciado: *Mapfre es una empresa de seguros que suele aparecer en los listados de adjudicatarios con distintos formatos de nombre. Observa el ranking de dinero recibido por adjudicatario de la Gerencia de la Empresa municipal de la Vivienda de Jerez. Usando el resto de pestañas, si lo considera necesario, calcula el total de dinero que recibe MAPFRE (con todos sus nombres) del órgano de contratación "Gerencia de la Empresa Municipal de la Vivienda de Jerez" y cuenta el número de licitaciones adjudicadas. Navegue hasta la pestaña de introducción del portal. Anota los resultados en un papel y cuando termines de realizar la búsqueda, proceda a rellenar las respuestas del formulario.*

1. **Total de dinero que recibe MAPFRE por parte de la Gerencia de la Empresa Municipal de la Vivienda de Jerez.:** En esta tarea opcional respondieron 10 participantes, el 80% respondió correctamente.

2. **Número total de licitaciones (expedientes) adjudicados a MAPFRE por parte de la Gerencia de la Empresa Municipal de la Vivienda de Jerez:** En cuanto al número total de licitaciones, 7 de los 10 participantes respondieron correctamente.

5.2.3 Preguntas sobre la experiencia de uso

En esta parte del cuestionario se realizaron tres preguntas para conocer la experiencia de los usuarios con el Portal y su impresión. Además, se dejó un último apartado para comentarios y aportaciones.

1. **Le ha parecido útil la herramienta:** De las 12 respuestas obtenidas, 10 puntuaron como "muy útil" la herramienta y 2 como "bastante útil". Estas respuestas reflejan que la impresión de todos los usuarios respecto a la utilidad de la herramienta fue buena.

2. **¿Le ha parecido útil y suficiente la guía de usuario que aparece en la pestaña de introducción?:** En la siguiente tabla (Tabla 5.4) se reflejan las evaluaciones de los usuarios. Como se puede observar, la mayoría consideran suficiente la guía, pero algunos apuntan que podría mejorar.

Nivel de Utilidad	Número de Respuestas	Porcentaje
Nada útil	0	0%
Poco útil	0	0%
Medio	2	16,7%
Útil	2	16,7%
Muy útil	8	66,7%

Tabla 5.4: Resultados a la pregunta sobre utilidad de la guía de usuario.

3. **En cuanto al acceso a información, ¿Qué le parece la herramienta en comparación con la Plataforma de Contratación del Sector Público?** Para responder a esta pregunta se planteaban varias opciones seleccionables y un campo en el que se permitía añadir otras opciones. En la siguiente tabla se muestran las respuestas de los participantes:

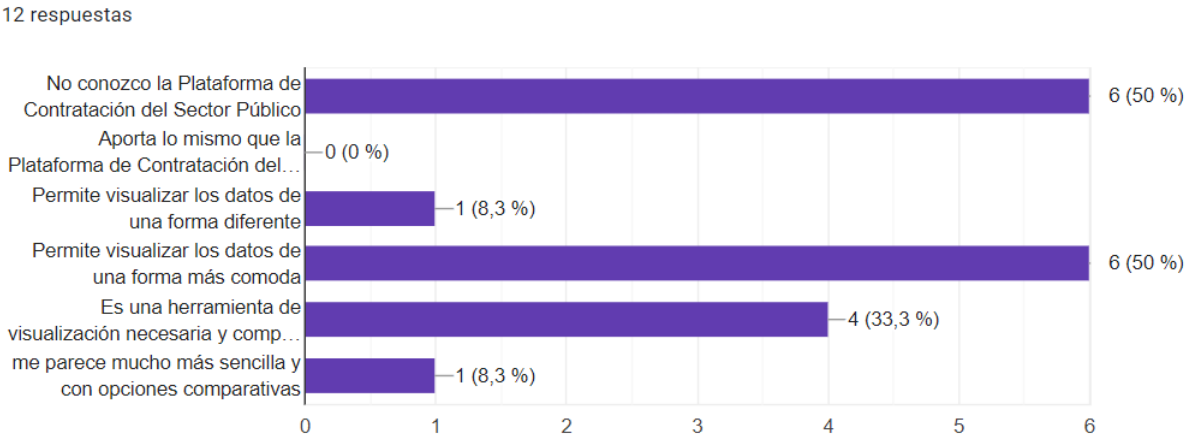


Figura 5.1: Impresiones sobre la herramienta frente a la PLACSP.

Comentarios de los usuarios

Al final del cuestionario se dio la opción a los participantes de aportar algún comentario aportación sobre la experiencia de usuario o el Portal. En general, los comentarios recibidos han sido positivos y algunos aportan ideas de mejora para continuar desarrollado el proyecto.

Entre los aspectos positivos mencionados, se destaca la facilidad de uso y la claridad de la aplicación con comentarios como *"Fácil de utilizar y muy intuitiva"* o *"Es interesante poder conocer estos datos de una forma fácil y clara"*. Los participantes también mencionaron que

es una herramienta que le da acceso a esta información a ciudadanos sin conocimientos técnicos *"Simplifica la búsqueda de información a la ciudadanía y profesionales que desconocen estos procesos"*.

En los comentarios también se mencionan áreas en las que mejorar la herramienta. Una de las limitaciones más comentadas es la mala adaptación de la aplicación en dispositivos móviles, puesto que los textos son demasiado largos y se solapan. Otro usuario mencionó la falta de claridad en los títulos de las pestañas y propuso mejorar la estética de la cabecera del portal: *"hasta que no te lees la Guía de Uso no sabes que es"*. Se propone también ampliar la Guía de Uso con el fin de ayudar a los usuarios a comprender mejor el funcionamiento del portal y los datos, así como añadir un cuadro explicativo en cada pestaña.

5.2.4 Evaluación de las respuestas del formulario

Analizando las respuestas de los usuarios que han experimentado con el Portal, parece que la impresión general ha sido positiva. El 100% de los usuarios preguntados considera útil el portal desarrollado. En cuanto a su uso, la mayoría de los usuarios considera que el portal es sencillo de usar, solo uno o dos usuarios han evaluado las tareas como difícil o muy difícil. La mayoría fueron capaces de responder correctamente a las tareas. Entre las respuestas incorrectas se observa que varios errores parecen ser por falta de conocimiento sobre los términos de contratación o por mala elección en los desplegables. Los tiempos de búsqueda son bastante razonables.

Este caso de uso ha permitido identificar algún error menor del portal que ha sido solventado sobre la marcha y varias mejoras para futuras versiones de la aplicación. Una de las mejoras que se ha implementado tras esta evaluación es la modificación de la guía de uso para la próxima versión de la aplicación, con el fin de explicar de forma más concreta el funcionamiento del portal y ampliar la información ofrecida sobre contratación pública. Otras de las modificaciones que se van a estudiar para la siguiente versión es el cambio de nombre de las pestañas y el formato de la cabecera. Para mejorar la retención de los usuarios y su experiencia con la interfaz, se plantea modificar la cabecera con una imagen más descriptiva y visual y modificar el nombre de las pestañas para que representen mejor su contenido.

Ha sido una buena decisión realizar este caso de uso con usuarios reales, me ha permitido tener una visión realista de cómo interaccionan los ciudadanos con el Portal de Visualización. La evaluación general ha sido buena y los comentarios sobre la herramienta ponen en valor este tipo de aplicaciones que acercan los datos de contratación a los ciudadanos.

Capítulo 6

Conclusiones y trabajos futuros

A lo largo de este capítulo se exponen las conclusiones del proyecto, lo que ha implicado en cuanto a aprendizaje y los posibles próximos pasos. En él se puede encontrar una sección en la que se analiza la consecución de los objetivos planteados al inicio del proyecto (1.1). Además, se ha realizado una reflexión sobre los conocimientos aprendidos en la carrera y cómo se han aplicado en el proyecto. También se exponen los nuevos conocimientos técnicos adquiridos durante el desarrollo del TFG. Por último, se ha incluido una sección en la que se recogen posibles mejoras y trabajos relacionados que pueden hacer que la herramienta evolucione y que este Trabajo Fin de Grado tenga una continuidad como proyecto.

6.1 Consecución de objetivos

En este punto, tras concluir la parte de desarrollo del proyecto, se puede evaluar como cumplido el objetivo general del TFG: Diseñar, implementar y desplegar un sistema robótico que automatice la extracción de datos de la Plataforma de Contratación del Sector Público, procese estos datos de manera estructurada y los muestre a través de una interfaz web interactiva y sencilla. Partiendo de esto, a continuación se analizan los objetivos específicos con el fin de recopilar y analizar el trabajo realizado y si este cumple con lo propuesto inicialmente:

- Módulo de extracción de datos: Se ha desarrollado un módulo de extracción de datos robusto utilizando técnicas de web scraping basado en Selenium, capaz de extraer los datos deseados de la PLACSP. Se ha conseguido que la interacción con la plataforma sea consistente, manejando los errores y tolerando excepciones.

- Sistema de actualización de datos: Se ha implementado un sistema que permite una actualización constante de los datos, pudiendo elegir el periodo de consulta de la plataforma, intentando evitar consultas innecesarias. Este sistema utiliza las cabeceras de los expedientes para consultar las nuevas incorporaciones o los estados modificados y actua-

liza la base de datos. Además, almacena los registros antiguos con su información y el HTML de la página para tener trazabilidad de los datos.

Módulo de procesamiento de datos: Se ha implementado un módulo que transforma los datos crudos extraídos y los almacena en una base de datos limpia, estructurada y estandarizada. Se ha diseñado una base de datos SQLite que almacena estos datos y es la que soporta las consultas de los mismos para su explotación. En este módulo se utilizan herramientas como Pydantic y SQLModel para la validación de datos y su estructuración.

- Interfaz de visualización web interactiva: Se ha desarrollado un módulo de visualización con diversas consultas a la base de datos para explotarlos. Los resultados de estas consultas se visualizan en una aplicación web interactiva desplegada con PythonAnyware y desarrollada con Dash y Plotly. Esta cuenta con una interfaz de usuario sencilla e intuitiva y tiene distintas formas de visualizar los datos.

- Pruebas exhaustivas y evaluación externa: Durante todo el proceso de desarrollo se han realizado pruebas desde distintos enfoques con el fin de crear una herramienta robusta y de utilidad para los usuarios finales. Se han implementado test para probar el código, mostrar su robustez y depurar errores. Por otro lado, se ha diseñado y llevado a cabo un caso de uso con el objetivo de conocer la sensación del usuario final del portal de visualización.

- Documentación completa: Puesto que uno de los propósitos de este proyecto era hacer más accesible los datos de contratación pública a la ciudadanía, crear una documentación clara y sencilla era fundamental. También existe necesidad de documentar el proceso de desarrollo para el TFG. Por estos motivos, se ha desarrollado una guía de usuario en la que se explica el funcionamiento del portal de visualización desplegado. También se ha diseñado una guía de instalación y uso para que usuarios con conocimientos de desarrollo pueda desplegar su propio sistema de extracción y visualización de datos. Por último, se ha documentado todo el proceso en esta memoria.

6.2 Planificación temporal

En esta sección se comenta el desarrollo del proyecto en cuanto a tiempo y planificación. El desarrollo de este TFG ha durado un curso completo contando todas sus fases. Uno de los obstáculos principales durante este tiempo ha sido la organización, mientras he desarrollado el proyecto he estado trabajando a tiempo completo, esto ha hecho que el tiempo disponible para este proyecto fuese menor del que era necesario. Finalmente, el proyecto ”Minería y visualización de datos de contratación pública del Estado” se define a finales de septiembre de 2023, siendo esta la fecha en la que empieza el trabajo de investigación y desarrollo en torno al mismo.

La planificación se ha llevado a cabo siguiendo una metodología *Agile* adaptándola a las necesidades del proyecto y avanzando de forma dinámica. Se han utilizado sprints para la

organización de las fases, creando así fases con objetivos claros y prototipos entregables. En el capítulo Desarrollo del proyecto (3) se puede ver en detalle la evolución que ha seguido el proyecto y los objetivos concretos de cada sprint. A continuación se muestra un diagrama de GANTT con la planificación temporal de cada sprint y bloques como la planificación y la memoria, teniendo en cuenta que la dedicación ha sido variable en cada fase del proyecto y con una intensidad mayor en los dos últimos meses del mismo.

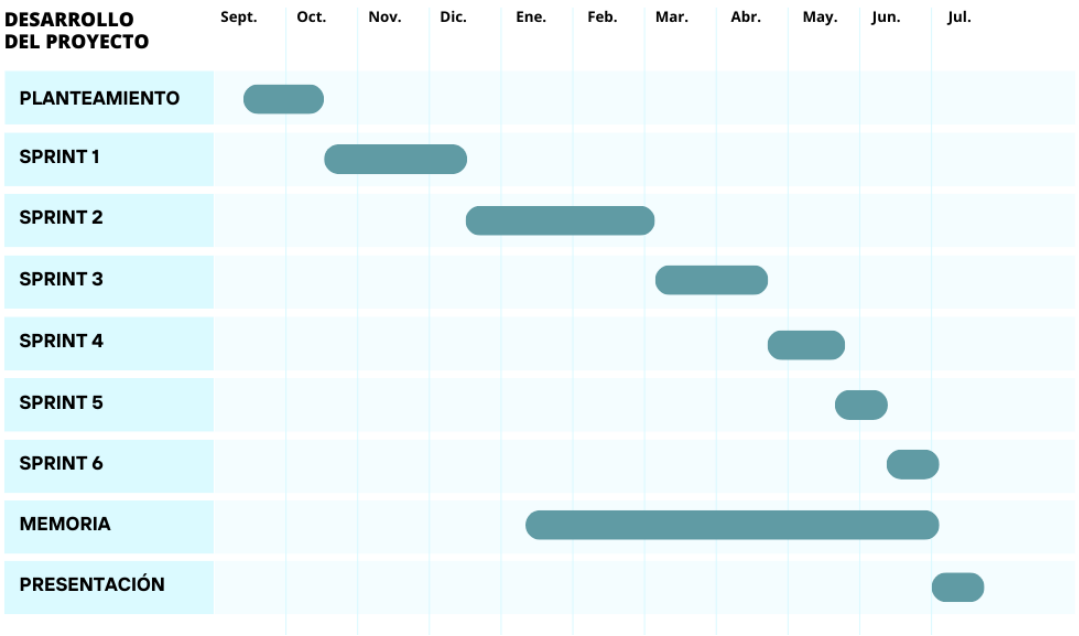


Figura 6.1: Diagrama de Gantt, planificación temporal del proyecto

Como se ha mencionado anteriormente, en el capítulo **??** se describen los detalles de cada sprint del proyecto. En la siguiente tabla (tabla 6.1) se muestra la dedicación temporal aproximada que ha supuesto cada sprint.

Horas dedicadas a cada fase		
Sprint	**Horas totales**	**Fechas**
Planteamiento	20	Sept - Oct
Sprint 1	60	Oct - Dic
Sprint 2	60	Dic - Mar
Sprint 3	60	Mar - Abr
Sprint 4	80	Abr - May
Sprint 5	110	May - Jun
Sprint 6	80	Jun - Jul
Memoria	130	Ene - Jul
Total	**600**	**Sep - Jul**

Tabla 6.1: Horas dedicadas a cada fase.

6.3 Aplicación de lo aprendido

El desarrollo de este Trabajo Fin de Grado se fundamenta en los conocimientos adquiridos a lo largo del grado en Ingeniería de Robótica Software. Estos conocimientos adquiridos durante mi periodo universitario, junto a conocimientos nuevos que he tenido que aprender para llevar a cabo el proyecto, han hecho que este TFG sea posible. A continuación se mencionan algunas asignaturas del plan de estudios del grado en las que se han trabajado las herramientas, conocimientos y habilidades necesarios para llevar a cabo el proyecto:

- **Fundamentos de programación, diseño software y algoritmos y estructuras de datos:** Estas asignaturas fueron las principales a la hora de aprender sobre lógica de programación, diferentes lenguajes, programación orientada a objetos y estructuras de datos. También aportaron un conocimiento profundo sobre algoritmos y su eficiencia. Estos conocimientos me han aportado las bases para mejorar mis conocimientos de Python y ser capaz de entender las diferentes herramientas y bibliotecas disponibles para la implementación del proyecto. También me han aportado conocimientos en cuanto a buenas prácticas de programación y estructuración del código que he intentado aplicar a este trabajo.

- **Laboratorio de sistemas y sistemas operativos:** Estas asignaturas me aportaron un conocimiento profundo sobre cómo funcionan los sistemas, la gestión eficiente de sus recursos y a entender las necesidades de una herramienta en relación con el sistema. En este proyecto he tenido poner en práctica estos conocimientos para entender la interacción con la web y mi sistema y para la publicación del prototipo Portal de Visualización en PythonAnyware.

- **Sistemas distribuidos y concurrentes:** En esta asignatura se trabajó con la concurrencia de los sistemas y la comunicación entre ellos. Aunque mi sistema no implementa concurrencia como tal, los conocimientos de esta me han ayudado a mejorar la interacción con la web a la hora de extraer los datos. También fueron útiles los conocimientos para el trabajo de carga y consulta en las bases de datos para mantener la coherencia de los mismos.

- **Arquitectura software y modelado y simulación de robots:** En estas asignaturas me han dado conocimientos sólidos sobre las distintas arquitecturas software y cómo diseñar un sistema compuesto por distintas herramientas que se comunican entre ellas.

- **Bases de datos relacionales:** Durante mi Erasmus en la Universidad de Angers en el curso 2021/2022 cursé la asignatura Bases de datos relacionales. En esta asignatura tuve la oportunidad de trabajar en la creación, gestión y explotación de bases de datos relacionales, utilizando principalmente SQL y Sqlite. Estos conocimientos me han servido para diseñar las bases de datos de mi proyecto y trabajar con ellas.

6.4 Lecciones aprendidas

Para poder llevar a cabo este proyecto he necesitado ampliar mis conocimientos en ciertas materias estudiadas durante el grado universitario, además, he necesitado aprender a utilizar nuevas herramientas y técnicas necesarias. A continuación se hace una breve descripción de las lecciones aprendidas durante el proyecto y su contexto dentro del mismo. Con el fin de facilitar su lectura, las he ordenado por ámbitos.

- **Tecnologías Web y Scraping de datos:** Una de las áreas en las que más he tenido que trabajar ha sido en las tecnologías web y web scraping. Antes de comenzar este proyecto no había trabajado nunca en torno a la extracción de datos. Durante el proyecto he tenido que aprender a utilizar todas las herramientas relacionadas con la interacción, extracción y tratamiento de los datos. Ya que todas ellas trabajaban con Python, he tenido que ampliar mis conocimientos sobre este lenguaje para poder cubrir las necesidades del proyecto.

- **Estructura de páginas webs:** Otro de los retos ha sido trabajar con el DOM de la PLACSP para extraer los datos deseados. Para ello he tenido que investigar sobre HTML, sus etiquetas y cómo se estructura la información dentro de una página web.

- **Testing:** Con el fin de asegurar la robustez y la fiabilidad de las herramientas implementadas, he aprendido a desarrollar pruebas unitarias utilizando la biblioteca unittest de Python. Esto me ha permitido identificar y corregir errores durante el desarrollo del proyecto.

- **Diseño de bases de datos relacionales:** Aunque ya tenía algunos conocimientos sobre bases de datos relacionales, he tenido que trabajan en refrescar y ampliar estos conocimientos para poder diseñar las bases de datos del proyecto para que estas sean eficientes y útiles.

- **SQLModel, SQLAlchemy y Pydantic:** aAntes de comenzar el proyecto solo había trabajado con bases de datos desde una interfaz gráfica, nunca había trabajado desde código python con un ORM ni con herramientas de control de tipos como Pydantic. SQLModel combina las capacidades de Pydantic y SQLAlchemy, lo que me permitió diseñar y gestionar las bases de datos desde mis códigos con control de tipo. Tuve que aprender sobre definición de modelos de datos, realización de consultas con el ORM y asegurar la consistencia de los datos.

- **Desarrollo de aplicaciones Web:** El desarrollo de la interfaz web para la visualización de datos ha sido otra área en la que no tenía conocimientos previos. Leyendo en varios artículos descubrí Dash y Plotly que combinados permitían crear visualizaciones interactivas en una aplicación web. He tenido que aprender a estructurar y diseñar una interfaz, gestionar los gráficos y su interactividad mediante callbacks y conectar esta web con una base de datos.

- **Diseño de visualizaciones de datos:** Dar con las visualizaciones adecuadas para mostrar todos los datos ha sido otro de los retos que me han llevado a aprender sobre visualización de datos y explicabilidad de los mismos. Más allá de los gráficos y tablas que se muestran en el portal de visualización, he realizado muchos experimentos de visualización con Plotly que por falta de tiempo no se muestran en esta versión del proyecto.

- **Despliegue en la nube:** Otro reto ha sido el despliegue de la aplicación en la nube, campo que desconocía. He aprendido a configurar un entorno en PythonAnyware, gestionar las dependencias e implementar actualizaciones.

- **Datos abiertos:** Durante el proyecto he necesitado formarme sobre la idea de datos abiertos y accesibilidad a los datos. He aprendido sobre las diferentes propuestas en torno a esto, normativa y mejores prácticas para la publicación, reutilización y visualización de los mismos.

- **Gestión de proyectos y metodología Agile:** Gracias a la propuesta de mi tutor del proyecto, he utilizados principios de metodología Agile para planificar el trabajo y el desarrollo. He aprendido a planificar y dividir el trabajo en sprints, marcar objetivos y crear una evolución dinámica del proyecto.

- **Documentación y escritura técnica:** Ha sido un verdadero reto escribir esta memoria. A pesar de haber escrito diversos informes en otros trabajos, el carácter técnico y descriptivo de esta memoria ha supuesto un reto y me ha aportado conocimientos sobre documentación de proyectos.

6.5 Próximos pasos

Durante el desarrollo del proyecto, la definición del mismo ha cambiado para ir adaptándolo a los nuevos descubrimientos y nuevas ideas que surgían manteniendo el objetivo principal. Muchas de estas ideas se han podido implementar y muchas otras se han quedado fuera de esta entrega por falta de recursos y tiempo. Ha sido un trabajo interesante crear de cero un conjunto de herramientas como este porque me ha permitido descubrir muchos campos que no conocía. En esta sección se describen un conjunto de mejoras que pueden implementarse sobre el trabajo presentado y dos proyectos relacionados.

6.5.1 Posibles mejoras

A continuación se describen un conjunto de mejoras que idealmente, si los recursos y el tiempo me lo permiten, me gustarían implementar en un futuro con el fin de ofrecer a los usuarios finales una herramienta que explote al máximo sus capacidades. Con el fin de agrupar los contenidos, primero se presentan las mejoras para el conjunto de herramientas y posteriormente las mejoras para el Portal de Visualización de Datos de Contratación.

Posibles mejoras del conjunto de herramientas

Son varias las mejoras y ampliaciones que se han ideado a lo largo del desarrollo de este conjunto de herramientas. A continuación se describen estas ideas y los pasos que se han empezado a dar para su implementación futura.

1. **Descarga de Contratos Menores:** Los datos de contratación publicados en la PLACSP se dividen en dos categorías principales, licitaciones y contratos menores. En esta versión de la herramienta solo se han extraído las licitaciones debido a la limitación temporal. Se plantea una nueva versión de la herramienta en la que se extraiga la información de los contratos menores y se almacenen en una nueva tabla de la base de datos. Con esta mejora, se tendrá en una sola base de datos toda la información disponible sobre los expedientes de contratación de los órganos y los adjudicatarios.

2. **Etiquetado de los expedientes mediante técnicas de procesamiento del lenguaje natural:** En uno de los sprints finales del desarrollo se investigó la posibilidad de incluir bibliotecas Python basadas en técnicas de procesamiento del lenguaje natural (NLP) con el fin de procesar el campo objeto del contrato de cada expediente y crear un conjunto de etiquetas de forma automática para facilitar la búsqueda por categorías. De esta forma, se podría implementar una nueva forma de explotación de los datos en la que, mediante un buscador, se pudiesen identificar contratos según su temática.

Posibles mejoras del Portal de Visualización de Datos de Contratación

Durante el desarrollo de este portal surgieron muchas ideas de mejora para futuras versiones del mismo. Entre las mejoras se encuentran formas diferentes de explotar los datos, mejoras en la estética y cambios en la forma de publicarlo. A continuación se muestran varias de las ideas estudiadas.

1. **Mejora de la estética del portal:** La interfaz del portal se ha diseñado con el fin de ser sencilla, intuitiva y sin elementos que puedan desorientar al usuario. Siguiendo con la estructura actualmente implementada, sería interesante trabajar en que la estética y la interacción del usuario con el portal fuese más dinámica y actual, pudiendo añadir imágenes, un diseño de cabecera más atractivo y unos bloques para los datos más visuales.

2. **Incluir los contratos menores:** Como se comenta en las mejoras propuestas para el conjunto de herramientas (Subsección 1), los contratos menores son un apartado relevante dentro de la contratación pública y sería interesante incluirlos dentro de este portal. Estos podría tener varias pestañas propias para su análisis y también permitirían cruzar los datos con los de licitación para tener una perspectiva completa de la contratación pública del órgano seleccionado.

3. **Migrar la aplicación a un servidor dedicado:** Actualmente, el portal de visualización de datos está alojado en PythonAnywhere. Para mejorar el rendimiento, la escalabilidad y la capacidad de personalización, sería interesante migrar en un futuro el proyecto a un servidor dedicado. Con esta migración se conseguiría un mayor control sobre el servidor, la capacidad de manejar un mayor tráfico de usuarios y la posibilidad de implementar configuraciones personalizadas que no son posibles en la plataforma actual.

4. **Interfaz para teléfonos móviles:** Uno de los comentarios más recibidos en el caso de uso (Sección 5.2) fue sobre los problemas a la hora de visualizar el portal desde el teléfono móvil. Desde un inicio se pensó la aplicación para ser utilizada desde un ordenador, pero es cierto que actualmente los ciudadanos utilizan principalmente el teléfono móvil para consultar información en internet. Para hacer más atractivo el portal, se plantea una adaptación del mismo con el fin de adaptarse a las tendencias de uso.

6.5.2 Proyectos futuros

Aparte de las mejoras planteadas, durante el desarrollo de este TFG han surgido dos proyectos interesantes que tienen como base el trabajo desarrollado. Estos proyectos son ideas futuras que pueden llegar a desarrollarse en algún momento con algo de esfuerzo, dedicación y recursos sin necesidad de tener que realizar un desarrollo desde cero.

Portal de Visualización de Datos de Contratación del Sector Público

Un futuro proyecto interesante es desplegar el prototipo del Portal de Visualización implementado en este proyecto con una base de datos que contenga todos los datos de contratación pública publicados en la PLACSP. Este proyecto es un gran paso en cuanto a escalabilidad de las herramientas desarrolladas, puesto que el volumen de datos es mucho mayor que el utilizado en este proyecto. Sería interesante que el portal no solo incluya los datos de un municipio específico, sino que abarque todas las licitaciones a nivel nacional, permitiendo un acceso completo y a la información de contratación pública.

Este proyecto implicaría escalar el módulo de almacenamiento de datos, así como optimizar las consultas a la base de datos y las visualizaciones para asegurar un buen rendimiento. También se podrían añadir funciones adicionales, como filtros avanzados, análisis de tendencias entre comunidades autónomas o a nivel general y análisis basados en todo el conjunto de datos, no solo por municipio.

¿Eres cotilla? Cotillea al Estado.

Durante la fase final de este proyecto tuve la oportunidad de participar en una formación de **Maldita.es** [11] llamada **"Juventud Antibulera"**. En esta formación se habló sobre

verificación de datos, noticias falsas, fuentes de información y explicabilidad de los datos a la ciudadanía. En uno de los talleres de esta jornada formativa tuvimos que desarrollar individualmente un proyecto contra la desinformación y utilicé el prototipo del Portal de Visualización de Datos como elemento central de este proyecto.

La idea presentada consiste en una campaña para animar a la ciudadanía, sobre todo a los jóvenes, a acceder a fuentes de información fiables en cuanto a presupuestos y contratación. Con esta campaña se pretendía dar a la ciudadanía herramientas para combatir la desinformación entrono a los presupuestos, pero también herramientas y conocimientos para ser auditores de las instituciones. Para ello, se planteaba una campaña llamada **"¿Eres cotilla? Cotillea al estado."** en la que se explicaba a los jóvenes, de manera sencilla y con un lenguaje cercano, la forma de consultar los datos de contratación pública dentro del Portal de Visualización de Datos implementado en este proyecto. A continuación se muestra la identidad visual desarrollada para la presentación de esta idea.

Figura 6.2: Identidad visual de la idea "Eres cotilla? Cotillea al Estado". Desarrollada durante el evento *"Juventud Antibulera"* organizado por Maldita.es

A pesar de ser solo una idea experimental, tuvo una gran acogida entre las asistentes y la organización. Entre las opiniones, se destacó la novedad del tema, puesto que la mayoría de los ciudadanos desconocen la PLACSP y es una buena herramienta para tener un seguimiento sobre el dinero público.

Contando con los recursos y el tiempo necesario, sería una gran experiencia continuar desarrollando el Portal de Visualización de Datos y las herramientas desarrolladas en este TFG, acompañando la publicación de este con campañas como la mencionada, que acerquen el portal los ciudadanos.

Referencias

[1] *¿Qué es RPA?* IBM. URL: https://www.ibm.com/es-es/topics/rpa.

[2] *argparse — Parser for command-line options, arguments and sub-commands*. Python.org. URL: https://docs.python.org/es/3/library/argparse.html.

[3] *Beautiful Soup Documentation*. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[4] *Dash - A Python framework for building analytical web applications*. URL: https://dash.plotly.com/.

[5] *Datos de gobierno abierto*. Lawrence Livermore National Laboratory. URL: https://opengovdata.org.

[6] *Developer Survey StackOverflow 2023*. URL: https://survey.stackoverflow.co/2023/#section-most-popular-technologies-integrated-development-environment.

[7] *Editor de textos en línea Overleaf*. URL: https://es.overleaf.com/.

[8] Gregorio Robles y Felipe Ortega. *Plantilla TFG Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad Rey Juan Carlos*. URL: https://github.com/gregoriorobles/plantilla-memoria/blob/master/memoria.tex.

[9] *Gobierno Abierto - Ministerio de Hacienda*. Ministerio de HAcienda - Gobierno de España. URL: https://www.hacienda.gob.es/es-ES/GobiernoAbierto/Datos%20Abiertos/Paginas/licitaciones_plataforma_contratacion.aspx.

[10] *Ley 9/2017, de 8 de noviembre, de Contratos del Sector Público*. Boletín Oficial del Estado - Gobierno de España. URL: https://www.boe.es/buscar/act.php?id=BOE-A-2017-12902.

[11] *Maldita.es*. URL: https://maldita.es/nosotros-maldita/.

[12] Ryan Mitchell. *Web Scraping with Python: Collecting More Data from the Modern Web*. Sebastopol, CA: O'Reilly Media, Inc., 2015.

[13] *Plataforma de Contratación del Sector Público*. Ministerio de Hacienda - Gobierno de España. URL: https://contrataciondelestado.es/wps/portal/plataforma.

[14] *Plotly - The front end for ML and data science models*. URL: https://plotly.com/.

[15] *Portal oficial de datos abiertos del Gobierno de España*. Gobierno de España. URL: https://datos.gob.es/es/.

[16] *Puppeteer Documentation*. URL: https://pptr.dev/.

[17] *Pydantic - Data validation and settings management using Python type annotations*. URL: https://pydantic-docs.helpmanual.io/.

[18] *Python España*. URL: https://es.python.org/.

[19] *PythonAnywhere - Host, run, and code Python in the cloud*. URL: https://www.pythonanywhere.com/.

[20] Bhaskar S. *¿Qué es la metodología ágil?* NimbleWork. URL: https://www.nimblework.com/es/agile/metodologia-agil/.

[21] *Scrapy Documentation*. URL: https://docs.scrapy.org/en/latest/.

[22] *Selenium*. URL: https://www.selenium.dev/.

[23] *SQLAlchemy Documentation*. URL: https://www.sqlalchemy.org/.

[24] *SQLite Home Page*. URL: https://www.sqlite.org/index.html.

[25] *sqlite3 — DB-API 2.0 interface for SQLite databases*. URL: https://docs.python.org/3/library/sqlite3.html.

[26] *SQLModel Documentation*. URL: https://sqlmodel.tiangolo.com/.

[27] *subprocess — Subprocess management*. Python.org. URL: https://docs.python.org/3/library/subprocess.html.

[28] *The LaTeX Project*. URL: https://www.latex-project.org/.

[29] *Visual Estudio Code*. URL: https://code.visualstudio.com/.

Bibliografía

[1] Limpieza de datos con Python - Al mal tiempo, buena data. https://lauralpezb.medium.com/limpieza-de-datos-con-python-48d436ca9ace. Medium.

[2] SQLAlchemy 2.0 ORM. - Jod35 https://github.com/jod35/SQLAlchemy-2.0-ORM. GitHub.

[3] Cómo hacer y publicar un dashboard con Plotly Dash y Heroku. - Gonzalezhomar https://medium.com/tacosdedatos/c%C3%B3mo-hacer-y-publicar-un-dashboard-con-plotly-dash-y-heroku-d6aa0bb70c6c. Medium.

[4] Deploying Dash Apps https://dash.plotly.com/deployment. Plotly.

[5] Filtered Download Example. - Dash Example Index https://dash-example-index.herokuapp.com/filtered-download. Heroku.

[6] What is Open Data? https://opendatahandbook.org/guide/es/what-is-open-data/. Open Data Handbook.

[7] El mapa de la contratación pública en Cantabria - Jaime Obregón https://contratosdecantabria.es/buscar/radio. Contratos de Cantabria.

[8] Guía Básica de la Ley de Contratos del Sector Público para Principiantes. https://civio.es/quien-cobra-la-obra/guia-basica-de-la-ley-de-contratos-del-sector-publico-para-principiantes/. Civio.

[9] Jaime Obregón. https://x.com/JaimeObregon. Twitter.

[10] Tu Derecho a Saber. https://civio.es/tu-derecho-a-saber/. Civio.

[11] Ryan Mitchell, *Web Scraping with Python: Collecting More Data from the Modern Web*, https://edu.anarcho-copy.org/Programming%20Languages/Python/Web%20Scraping%20with%20Python,%202nd%20Edition.pdf, O'Reilly Media, 2015.

[12] Mark Lutz, *Learning Python*, https://cfm.ehu.es/ricardo/docs/python/Learning_Python.pdf, O'Reilly Media, 5th Edition, 2013.

[13] Edward R. Tufte, *The Visual Display of Quantitative Information*, https://faculty.salisbury.edu/~jtanderson/teaching/cosc311/fa21/files/tufte.pdf, Graphics Press, 2nd Edition, 2001.

[14] Jay A. Kreibich, *Using SQLite*, https://it.dru.ac.th/o-bookcs/pdfs/17.pdf, O'Reilly Media, 2010.