

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

Trabajo Fin de Grado

Aplicación Web para el análisis de datos sobre el consumo eléctrico español



Realizado por

Carlos Nuchera Bolaños

Para la obtención del título de

Grado en Ingeniería Informática - Ingeniería del Software

Dirigido por

María del Mar Martínez Ballesteros

Manuel Jesús Jiménez Navarro

Realizado en el departamento de

Lenguaje y Sistemas Informáticos

Agradecimientos

Quiero expresar mi más sincero agradecimiento a mis tutores, María del Mar Martínez Ballesteros y Manuel Jesús Jiménez Navarro. Su atención constante y su seguimiento detallado han sido fundamentales para la realización de este trabajo. Sus valiosos consejos, así como su comprensión y cercanía, me han ayudado enormemente a alcanzar mi objetivo.

Este proyecto ha sido una experiencia compleja, un reto que ocupará un espacio en mi memoria para siempre. No solo he puesto en práctica todos mis conocimientos técnicos, sino que también me he dado cuenta, tras muchos años en la carrera, de que el verdadero aprendizaje no radica en adquirir conocimientos específicos, sino en desarrollar la capacidad de estar cómodo en situaciones adversas y desconocidas. Este reto ha sido una prueba de esta capacidad, y estoy profundamente agradecido a mis tutores por guiarme en este proceso.

Resumen

La página oficial de ESIOS (Entidad de Servicios de los Mercados Eléctricos) ofrece acceso a datos en tiempo real e históricos sobre el mercado eléctrico, la generación de energía, la demanda eléctrica y otros aspectos relevantes para el sector energético en España.

El proyecto se enfoca en el desarrollo de una aplicación web para el análisis de datos del consumo eléctrico español, aprovechando las APIs tanto de Energía en Sistema de Información del Operador del Sistema, Red Eléctrica de España y Mercado Ibérico del Gas para la obtención de datos clave.

El propósito fundamental de esta iniciativa es facilitar el análisis de datos relacionados con el consumo eléctrico en España, permitiendo a los usuarios explorar y comprender patrones, tendencias y anomalías en los datos a través de una interfaz web intuitiva y accesible. La aplicación estará diseñada para usuarios sin experiencia en análisis de datos avanzados, asegurando que puedan extraer información valiosa de manera sencilla.

Para lograr este objetivo, la aplicación deberá ser capaz de acceder a los datos proporcionados por las APIs, seleccionando la información importante. Una vez recopilados los datos, se realizan análisis para identificar patrones de consumo y detectar posibles anomalías o tendencias.

La implementación de este proyecto requerirá la integración de diversas tecnologías, incluyendo Django como framework principal para el desarrollo web. Además, se emplearán técnicas de visualización de datos para presentar los resultados de manera clara y comprensible.

El desarrollo de la aplicación incluirá varias etapas, desde la definición de requisitos y el diseño de la arquitectura hasta la implementación y la prueba de la aplicación. Se seguirá un enfoque iterativo para garantizar la calidad y la eficacia del producto final.

Al concluir el proyecto, se espera obtener una aplicación web totalmente funcional, denominada "Electricity Consumption Analyzer", que permita a los usuarios analizar y comprender fácilmente los datos del consumo eléctrico en España. Se proporcionará documentación detallada sobre el desarrollo del proyecto, así como conclusiones y lecciones aprendidas durante el proceso.

Summary

The official ESIOS (Entidad de Servicios de los Mercados Eléctricos) website offers access to real-time and historical data on the electric market, energy generation, electric demand, and other relevant aspects of the energy sector in Spain.

The project focuses on developing a web application for analyzing Spanish electricity consumption data, leveraging APIs from the Energy Information System of the System Operator, Red Eléctrica de España, and the Iberian Gas Market to obtain key data.

The fundamental purpose of this initiative is to facilitate the analysis of electricity consumption data in Spain, allowing users to explore and understand patterns, trends, and anomalies in the data through an intuitive and accessible web interface. The application is designed for users without advanced data analysis experience, ensuring they can easily extract valuable information.

To achieve this goal, the application must be capable of accessing the data provided by the APIs, selecting important information. Once the data is collected, analyses will be conducted to identify consumption patterns and detect possible anomalies or trends.

The implementation of this project will require the integration of various technologies, including Django as the main framework for web development. Additionally, data visualization techniques will be employed to present the results clearly and comprehensively.

The development of the application will include several stages, from defining requirements and designing the architecture to implementing and testing the application. An iterative approach will be followed to ensure the quality and effectiveness of the final product.

Upon completing the project, the goal is to have a fully functional web application called "Electricity Consumption Analyzer," which will allow users to easily analyze and understand electricity consumption data in Spain. Detailed documentation on the project's development, as well as conclusions and lessons learned during the process, will be provided.

Índice general

I-INTRODUCCIÓN	11
1. Introducción	12
1.1 Contexto	12
1.2 Motivación	12
1.3 Definición de objetivos	13
II-ESTUDIO PREVIO	14
2 Estudio previo	15
2.1 Fuentes de datos	15
2.1.1 REE (Red Eléctrica de España):	15
2.1.2 MIBGAS	17
2.1.3 ESIOS (Entidad de Servicios de los Mercados Eléctricos):	19
2.2 Atributos de datos obtenidos	21
2.3 Series temporales	24
2.3.1 Definición y características	24
2.3.2 Importancia del análisis de series temporales	24
2.3.3 Aplicación de series temporales	25
III-SISTEMA DESARROLLADO	26
3 Objetivos	27
4. Análisis	28
4.1 Requisitos	28
4.1.1 Roles	28
4.1.2 Requisitos de información	29
4.1.3 Requisitos funcionales	34
4.1.4 Requisitos no funcionales	36
4.1.5 Reglas de negocio	38
4.1.6 Matriz de trazabilidad	39
4.2 Casos de uso	41
5 Riesgos	42
5.1 Análisis cualitativo y cuantitativo de los riesgos	42
5.2 Tabla de riesgos	44
5.3 Plan de respuesta a riesgos	45
6 Metodología	46
6.1 Fases del modelo en cascada	46
6.2 Gestión de código con GitHub	47
7. Diseño	48
7.1 Diseño de tipos de análisis	48
7.1.1 Gráficas de autocorrelación	48
7.1.2 Descomposición de series temporales	50
7.1.3 Detección de outliers	52

7.2 Diagrama de clases	54
7.3 Arquitectura del sistema	59
7.3.1 API	60
7.3.2 Problema de procesamiento de datos y solución	60
7.3.3 Herramientas de Documentación de la API	63
7.4 Tecnologías	63
7.4.1 Herramientas de documentación	63
7.4.2 Herramientas de desarrollo	64
7.5 Diseño API	66
IV-PLANIFICACIÓN	68
8 Planificación	69
8.1 Listado de hitos	69
8.2 Estructura de descomposición del trabajo	69
8.3 Diccionario de la EDT	71
8.4 Cronograma	86
8.5 Costes	90
8.5.1 Coste de personal	90
8.5.2 Coste de material	90
8.5.3 Coste por consumo	91
8.5.4 Resumen de Costes	91
V-SEGUIMIENTO	92
9 Introducción del seguimiento	93
9.1 Primera fase	94
9.1.1 Incidencias	94
9.1.2 Desviaciones	95
9.1.3 Riesgos que han aparecido	96
9.1.4 Replanificación	97
9.1.5 Retrospectiva	97
9.2 Segunda fase	98
9.2.1 Incidencias	98
9.2.2 Desviaciones	98
9.2.3 Riesgos que han aparecido	99
9.2.4 Replanificación	100
9.2.5 Retrospectiva	100
9.3 Tercera fase	101
9.3.1 Incidencias	101
9.3.2 Desviaciones	101
9.3.3 Riesgos que han aparecido	102
9.3.4 Replanificación	103
9.3.5 Retrospectiva	103
9.4 Cuarta fase	104
9.4.1 Incidencias	104
9.4.2 Desviaciones	104
9.4.3 Riesgos que han aparecido	105

9.4.4 Replanificación	105
9.4.5 Retrospectiva	105
10. Esfuerzo empleado	106
11. Coste final	110
VI-CIERRE	112
12 Producto final	113
12.1 Manual de instalación	113
12.2 Manual de usuario	120
12.2.1 Usuario no registrado	120
11.2.2 Usuario registrado	123
12.3 Pruebas	133
13. Cumplimiento de objetivos	134
14. Conclusiones	135
14.1. Lecciones aprendidas	135
15. Trabajo futuro	136
VII-APÉNDICE	138
16. Glosario	139
17. Bibliografía	140

Índice de figuras

2.1.1	Página principal de Ree.es	16
2.1.2	Página principal de mibgas.es	18
2.1.3	Página principal de esios.ree.es	20
2.2.1	Ejemplo de datos de demanda	21
2.2.2	Ejemplo de datos de generaciones	22
2.2.3	Ejemplo de datos de gas	22
2.2.4	Ejemplo de datos de precio	23
4.2	Casos de uso	41
7.1.1	Ejemplo de autocorrelación parcial	48
7.1.2	Ejemplo de descomposición aditiva	50
7.1.3	Ejemplo de Outliers	52
7.2	Diagrama UML	54
7.3	Diagrama de arquitectura	59
7.3.2.1	Diagrama de arquitectura Celery	60
7.3.2.2	Diagrama de flujo	62
7.5	Vista de Analistas	66
8.2	Diagrama EDT general	70
8.3.1	Diagrama EDT FASE 1	71
8.3.2	Diagrama EDT FASE 2	72
8.3.3	Diagrama EDT FASE 3	72
8.3.4	Diagrama EDT FASE 4	72
8.4.1	Diagrama de Gantt-Planificación	86
8.4.2	Diagrama de Gantt-Fase 1	87
8.4.3	Diagrama de Gantt-Fase 2	87
8.4.4	Diagrama de Gantt-Fase 3	88
8.4.5	Diagrama de Gantt-Fase 4	89
8.4.6	Diagrama de Gantt-Cierre	89
11.1	Desviación de costes	110
11.2	Costes acumulados	110
12.1.1	Instalación de PgAdmin 4	114
12.1.2	Contraseña Postgres	114
12.1.3	Creación de base de datos	115
12.1.4	Instrucción load_data.py	116
12.1.5	RabbitMQ instalado como servicio	117
12.1.6	Celery Beat	118
12.1.7	Celery Worker	119
12.2.1	Vista de Inicio	120
12.2.2	Vista de Analistas	120
12.2.3	Vista de API	121
12.2.4	Vista de Swagger	121
12.2.5	Vista de Contacto	122
12.2.6	Vista de Análisis	123
12.2.7	Vista de Mis Análisis En proceso	123
12.2.8	Vista de Mis Análisis Terminado	124
12.2.9	Vista Resultados	124

12.2.10	CSV datos preprocesados	125
12.2.11	Vista formulario Autocorrelación	125
12.2.12	Vista gráficas Autocorrelación	126
12.2.13	Guardar estudio Autocorrelación	127
12.2.14	Vista Resultados Autocorrelación estudio	127
12.2.15	Vista Resultados Autocorrelación gráficas adjuntas	128
12.2.16	CSV resultados estudios Autocorrelación	128
12.2.17	Formulario y visualizar Detección de Outliers	129
12.2.18	Formulario Descomposición de Series Temporales	130
12.2.19	Visualizar Descomposición de Series Temporales-1	130
12.2.20	Visualizar Descomposición de Series Temporales-2	131
12.2.21	Vista previsualizar PDF	132
12.2.22	Resultados de análisis PDF	132

Índice de cuadros

4.1.1	Roles	28
4.1.2	Requisitos de información	29
4.1.3	Requisitos funcionales	34
4.1.4	Requisitos no funcionales	37
4.1.5	Reglas de negocio	38
4.1.6	Matriz de trazabilidad	39
5.1	Riesgos por Impacto-Probabilidad	43
5.2	Riesgos	44
5.3	Respuesta a riesgos	45
8.1	Hitos	69
8.3	Diccionario EDT-Planificación	73
8.3.1	Diccionario EDT-Fase 1	75
8.3.2	Diccionario EDT-Fase 2	77
8.3.3	Diccionario EDT-Fase 3	79
8.3.4	Diccionario EDT-Fase 4	83
8.3.5	Diccionario EDT-Cierre	85
8.5.1	Coste de personal	90
8.5.2	Coste de material	90
8.5.3	Coste por consumo	91
8.5.4	Resumen de costes	91
9.1.1	Incidencias Fase 1	94
9.1.2.1	Variación de coste Fase 1	95
9.1.2.2	Variación del trabajo Fase 1	95
9.1.2.3	Completitud Fase 1	95
9.2.1	Incidencias Fase 2	98
9.2.2.1	Variación de coste Fase 2	98
9.2.2.2	Variación del trabajo Fase 2	98
9.2.2.3	Completitud Fase 2	99
9.3.1	Incidencias Fase 3	101
9.3.2.1	Variación de coste Fase 3	101
9.3.2.2	Variación del trabajo Fase 3	102
9.3.2.3	Completitud Fase 3	102
9.4.2.1	Variación de coste Fase 4	104
9.4.2.2	Variación del trabajo Fase 4	104
9.4.2.3	Completitud Fase 4	104
10	Esfuerzo real empleado	106
11.1	Coste estimado	111
11.2	Coste real	111

Primera parte

I-INTRODUCCIÓN

1. Introducción

Este capítulo introduce el tema del trabajo, ofreciendo contexto, explicando la motivación detrás del proyecto y delineando sus objetivos principales. Además, se describe la estructura del documento.

1.1 Contexto

El proyecto ESIOS (Entidad de Servicios de los Mercados Eléctricos) [1] es una plataforma central para acceder a datos clave relacionados con el sistema eléctrico en España. Su objetivo principal es proporcionar información en tiempo real y datos históricos sobre el mercado eléctrico español, incluyendo aspectos como la generación de energía, la demanda eléctrica, la producción de energía renovable y otros aspectos relevantes para el sector energético.

A través de ESIOS, se recopila y se pone a disposición del público una amplia gama de datos relacionados con el sistema eléctrico español. Esto incluye información detallada sobre la producción y el consumo de energía en diferentes regiones del país, así como datos sobre precios de la electricidad, interconexiones internacionales y otros aspectos clave del mercado eléctrico.

La plataforma ofrece acceso abierto a esta información a través de su portal en línea, permitiendo a investigadores, profesionales del sector energético y otras partes interesadas acceder a datos actualizados y realizar análisis en profundidad sobre el sistema eléctrico español. Esta disponibilidad de datos en tiempo real y la transparencia en la información son fundamentales para comprender el funcionamiento del mercado eléctrico y para informar la toma de decisiones en el ámbito energético.

1.2 Motivación

La principal motivación detrás de este Trabajo de Fin de Grado radica en el profundo interés por el aprendizaje y la aplicación de técnicas de machine learning en el sector eléctrico. En la actualidad, el sector energético se encuentra en constante evolución, enfrentándose a desafíos cada vez más complejos en términos de eficiencia, sostenibilidad y gestión de recursos.

“Tras siglos de dependencia absoluta de los combustibles fósiles, nos encontramos en un punto de inflexión donde las energías renovables lideran la transición hacia un modelo más sostenible y resiliente. Con sus luces y sus sombras, la última cumbre del clima celebrada en Dubai, la COP 28, acordó triplicar la producción de energías renovables y duplicar la eficiencia energética de aquí a 2030, destacando también por escrito la necesidad de “dejar atrás” los combustibles fósiles.” [2]

Mediante el análisis y la interpretación de los datos proporcionados por empresas ya existentes, es posible identificar tendencias, prever patrones de consumo, optimizar la distribución de energía y mejorar la eficiencia operativa en toda la red eléctrica.

Por lo que aprovechando las técnicas que veremos en profundidad más adelante comprobaremos que servirán como base para impulsar la innovación y la eficiencia en el mercado. La aplicación de estas técnicas puede contribuir significativamente a la transición hacia un sistema eléctrico más inteligente, adaptable y sostenible en el futuro. Además, el dominio de estas habilidades técnicas es fundamental para la formación de ingenieros informáticos y por lo tanto para mi caso particular, abriéndome camino para el conocimiento de nuevas tecnologías.

1.3 Definición de objetivos

El propósito central de este proyecto es crear una plataforma web centrada en el desarrollo de software para investigadores del sector eléctrico, que permita el análisis de datos de manera eficiente acompañado de una interfaz de usuario intuitiva. Se busca proporcionar una herramienta robusta que les permita realizar estudios de forma interactiva, así como también descargar los datos resultantes de sus análisis.

Durante la fase de análisis, el enfoque estará puesto en la implementación de técnicas y algoritmos proporcionados por los tutores del Trabajo de Fin de Grado para garantizar el éxito del proyecto. Esto incluirá la utilización de gráficas de autocorrelación y autocorrelación parcial, detección de outliers y descomposición de series temporales, así como la investigación sobre requisitos interactivos y parametrizables.

Es fundamental que la aplicación no solo ejecute estos análisis de manera eficiente, sino que también presente los resultados de forma visual y comprensible a través de diversos gráficos y representaciones gráficas. La accesibilidad y usabilidad serán prioridades, permitiendo que incluso usuarios sin un profundo conocimiento técnico puedan interpretar los hallazgos.

Además, se considera crucial implementar una funcionalidad que permita a los usuarios descargar los resultados de los análisis realizados. Esto no solo fomentará la transparencia y la reproducibilidad de los resultados, sino que también empodera a los usuarios para realizar análisis adicionales fuera de la aplicación.

Dentro del marco del departamento de Lenguajes y Sistemas Informáticos, se prestará especial atención a la planificación y seguimiento del desarrollo del proyecto. Se aplicarán metodologías y mejores prácticas aprendidas a lo largo de la carrera para garantizar una ejecución eficaz y una entrega exitosa del producto final.

Es importante destacar que estos objetivos podrían evolucionar durante la fase de planificación del proyecto, permitiendo ajustes en su alcance o enfoque según sea necesario para alcanzar los objetivos finales de manera óptima.

Segunda Parte

II-ESTUDIO PREVIO

2 Estudio previo

Para llevar a cabo este proyecto, es esencial adquirir un conocimiento básico sobre el dominio del problema sobre el análisis de los datos obtenidos de ESIOS y las técnicas utilizadas para su estudio. Esta comprensión inicial resulta fundamental para la identificación y obtención de los requisitos necesarios en el desarrollo de la aplicación. Para este proyecto, nos centraremos en técnicas específicas como las gráficas de autocorrelación y autocorrelación parcial, la detección de outliers, la descomposición de series temporales, así como en la investigación sobre requisitos interactivos y parametrizables.

2.1 Fuentes de datos

En este apartado se detallarán los datos que se utilizarán para llevar a cabo los diversos estudios en el proyecto. Se han recopilado datos de diferentes fuentes para garantizar una visión integral y precisa de los aspectos relevantes del sector eléctrico.

2.1.1 REE (Red Eléctrica de España):

¿Qué es REE?

REE [3] es una empresa española que se encarga de la gestión y operación del sistema eléctrico en España. Su responsabilidad abarca desde la transmisión de la energía eléctrica hasta la coordinación de la operación del sistema en tiempo real. Además, REE desempeña un papel crucial en la integración de las energías renovables en el sistema eléctrico y en la planificación y desarrollo de infraestructuras energéticas.

Datos disponibles en REE

- *Demanda Eléctrica:* Proporciona datos sobre la demanda eléctrica en tiempo real y datos históricos. Estos datos son fundamentales para comprender el consumo de electricidad en diferentes regiones y momentos del día.
- *Generación de Energía:* La empresa ofrece información detallada sobre la generación de energía eléctrica, incluyendo datos sobre la producción de energía renovable (eólica, solar, hidroeléctrica) y no renovable (nuclear, carbón, gas).
- *Balance Energético:* REE proporciona datos sobre el balance entre la demanda y la oferta de energía eléctrica en el sistema. Esto incluye información sobre la capacidad de generación disponible, las interconexiones internacionales y el intercambio de energía con otros países.
- *Operación del Sistema:* La empresa ofrece información en tiempo real sobre la operación del sistema eléctrico, incluyendo datos sobre la frecuencia del sistema, la capacidad de transporte de energía y la gestión de la demanda.

Página web de REE

Se ofrece una amplia variedad de secciones y servicios destinados a diferentes usuarios, desde consumidores domésticos hasta profesionales del sector energético. A continuación, se describen las principales secciones que se pueden encontrar en su página web(Figura 2.1.1).

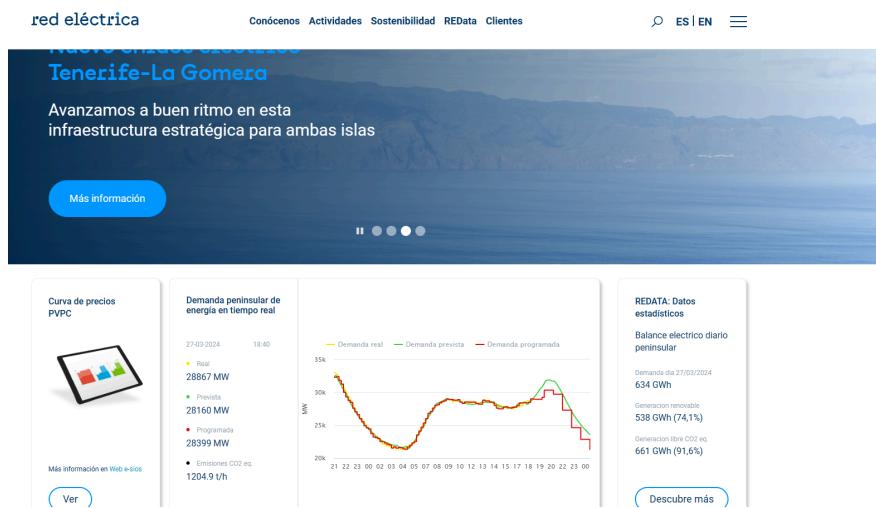


Figura 2.1.1: Página principal de Ree.es.

1. *Inicio:* Esta sección proporciona información general sobre REE, incluyendo su misión, visión y valores, así como noticias y eventos relacionados con la empresa.
2. *Operación del Sistema:* En esta sección, se pueden encontrar datos en tiempo real sobre la operación del sistema eléctrico, incluyendo la demanda, la generación de energía y el balance energético. También se proporciona información sobre los procedimientos de operación y los servicios ofrecidos por REE.
3. *Planificación Energética:* Aquí se puede acceder a información sobre la planificación y el desarrollo de infraestructuras energéticas en España, incluyendo proyectos de ampliación de redes eléctricas y conexiones internacionales.
4. *Transparencia:* REE promueve la transparencia en el sector energético proporcionando acceso a información relevante sobre su actividad y sus operaciones. En esta sección, se pueden encontrar informes anuales, datos financieros y otros documentos de interés público.
5. *Noticias y Publicaciones:* Esta sección incluye noticias y artículos relacionados con el sector energético, así como publicaciones técnicas y científicas producidas por REE y otros organismos relacionados.
6. *Atención al Cliente:* REE ofrece servicios de atención al cliente para resolver consultas y proporcionar asistencia técnica a usuarios y consumidores. En esta

sección, se pueden encontrar información de contacto y formularios para realizar consultas o reclamaciones.

7. *Empleo y Carreras Profesionales*: Aquí se pueden encontrar oportunidades de empleo y formación en REE, así como información sobre programas de prácticas y becas para estudiantes y jóvenes profesionales.

2.1.2 MIBGAS

¿Qué es MIBGAS?

MIBGAS [4] es una entidad española que opera como mercado organizado de gas natural en España. Su principal función es promover la transparencia, la competencia y la eficiencia en el mercado del gas, facilitando la negociación y la contratación de gas entre los diferentes agentes del mercado. MIBGAS desempeña un papel crucial en la integración y el desarrollo del mercado del gas en España, contribuyendo a la seguridad y la estabilidad del suministro de gas natural en el país.

Datos disponibles en MIBGAS

- *Precios del Gas*: Proporciona datos sobre los precios del gas natural en diferentes momentos y puntos de entrega en España. Estos datos incluyen precios diarios, precios intradiarios y precios futuros, que son fundamentales para comprender la dinámica del mercado del gas y tomar decisiones informadas sobre la compra y venta de gas natural.
- *Volumen de Intercambio*: La plataforma ofrece información sobre el volumen de gas natural negociado en el mercado, tanto en términos absolutos como por punto de entrega. Esto permite evaluar la liquidez y la actividad del mercado, así como identificar tendencias y patrones en la demanda y la oferta de gas natural.
- *Contratos y Servicios*: Contiene la información sobre los diferentes tipos de contratos de gas natural disponibles en el mercado, así como los servicios y productos ofrecidos por los participantes del mercado, incluyendo servicios de almacenamiento, transporte y distribución de gas.

Página web de MIBGAS

La página web de MIBGAS ofrece una amplia variedad de servicios y herramientas destinadas a diferentes usuarios, desde traders y analistas financieros hasta empresas de gas y reguladores del sector energético. A continuación, se describen las principales secciones que se pueden encontrar en su página web (Figura 2.1.2).

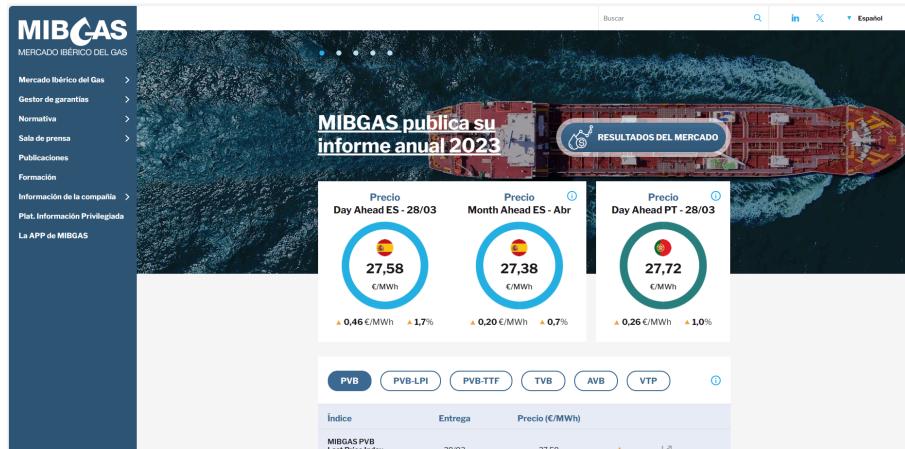


Figura 2.1.2: Página principal de mibgas.es.

- Inicio:* Esta sección proporciona información general sobre MIBGAS, incluyendo su misión, visión y valores, así como noticias y eventos relacionados con el mercado del gas en España.
- Precios del Gas:* En esta sección, se pueden encontrar datos sobre los precios del gas natural en diferentes puntos de entrega y momentos del día. Se proporciona acceso a gráficos y tablas de precios, así como herramientas de análisis y comparación de precios.
- Volumen de Intercambio:* Aquí se puede acceder a información sobre el volumen de gas natural negociado en el mercado, incluyendo datos sobre el número de transacciones, el volumen total intercambiado y la distribución geográfica de las operaciones.
- Contratos y Servicios:* Contiene la información sobre los diferentes tipos de contratos de gas natural disponibles en el mercado, así como los servicios y productos ofrecidos por los participantes del mercado. Se proporcionan detalles sobre los requisitos de contratación, los plazos de entrega y las condiciones de servicio.
- Regulación y Normativa:* Esta sección incluye información sobre la regulación y la normativa aplicables al mercado del gas en España. Se proporcionan enlaces a documentos legales y regulatorios relevantes, así como información sobre los procedimientos de supervisión y cumplimiento.
- Noticias y Publicaciones:* Publica noticias y artículos relacionados con el mercado del gas en España, así como informes técnicos y estudios de mercado. Esta sección ofrece acceso a información actualizada sobre tendencias, eventos y desarrollos en el sector del gas natural.
- Atención al Cliente:* Ofrece servicios de atención al cliente para resolver consultas y proporcionar asistencia técnica a usuarios y participantes del mercado. Se

proporciona información de contacto y formularios para realizar consultas o solicitudes de información.

2.1.3 ESIOS (Entidad de Servicios de los Mercados Eléctricos):

¿Qué es ESIOS?

ESIOS [1] es una entidad española encargada de la gestión y operación de los mercados eléctricos en España. Su objetivo principal es garantizar la transparencia, la eficiencia y la seguridad en el suministro eléctrico, facilitando el acceso a información relevante sobre la operación del sistema eléctrico y los precios de la electricidad. ESIOS desempeña un papel crucial en la integración de las energías renovables y la promoción de la eficiencia energética en España.

Datos disponibles en ESIOS

- *Precios de la Electricidad:* ESIOS proporciona datos sobre los precios de la electricidad en diferentes mercados y períodos de tiempo. Esto incluye precios horarios, diarios y mensuales, así como precios de mercado spot y precios de mercado de futuros. Estos datos son esenciales para comprender la dinámica del mercado eléctrico y tomar decisiones informadas sobre la compra y venta de electricidad.
- *Indicadores del Sistema:* La plataforma ofrece información en tiempo real sobre la operación del sistema eléctrico, incluyendo indicadores de frecuencia, tensión y capacidad de generación. Estos indicadores son fundamentales para monitorear la estabilidad y la seguridad del sistema eléctrico y prevenir posibles incidentes o interrupciones en el suministro eléctrico.
- *Datos Meteorológicos:* ESIOS proporciona datos meteorológicos relevantes para el sector eléctrico, incluyendo información sobre la radiación solar, la velocidad del viento y la temperatura ambiente. Estos datos son cruciales para prever la generación de energía renovable y planificar la operación del sistema eléctrico de manera eficiente.

Página web de ESIOS

La página web de ESIOS ofrece una amplia gama de servicios y herramientas destinadas a diferentes usuarios, desde operadores del sistema eléctrico hasta consumidores finales. A continuación, se describen las principales secciones que se pueden encontrar en su página web (Figura 2.1.3).



Figura 2.1.3: Página principal de esios.ree.es.

- Inicio:* Esta sección proporciona información general sobre ESIOS, incluyendo su misión, visión y valores, así como noticias y eventos relacionados con el mercado eléctrico en España.
- Precios de la electricidad:* En esta sección, se pueden encontrar datos sobre los precios de la electricidad en diferentes mercados y períodos de tiempo. Se proporciona acceso a gráficos y tablas de precios, así como herramientas de análisis y comparación de precios.
- Indicadores del Sistema:* Aquí se puede acceder a información en tiempo real sobre la operación del sistema eléctrico, incluyendo indicadores de frecuencia, tensión y capacidad de generación. Se proporciona acceso a gráficos y tablas de indicadores, así como herramientas de análisis de la estabilidad del sistema.
- Datos Meteorológicos:* ESIOS ofrece datos meteorológicos relevantes para el sector eléctrico, incluyendo información sobre la radiación solar, la velocidad del viento y la temperatura ambiente. Se proporciona acceso a gráficos y tablas de datos meteorológicos, así como herramientas de análisis de la generación de energía renovable.
- Regulación y Normativa:* Esta sección incluye información sobre la regulación y la normativa aplicables al mercado eléctrico en España. Se proporcionan enlaces a documentos legales y regulatorios relevantes, así como información sobre los procedimientos de supervisión y cumplimiento.
- Noticias y Publicaciones:* ESIOS publica noticias y artículos relacionados con el mercado eléctrico en España, así como informes técnicos y estudios de mercado. Esta sección ofrece acceso a información actualizada sobre tendencias, eventos y desarrollos en el sector eléctrico.
- Atención al Cliente:* ESIOS ofrece servicios de atención al cliente para resolver consultas y proporcionar asistencia técnica a usuarios y participantes del mercado eléctrico. Se proporciona información de contacto y formularios para realizar consultas o solicitudes de información.

2.2 Atributos de datos obtenidos

Dada la amplia variedad de datos disponibles en el sector energético español, en este proyecto nos centraremos en la demanda y las generaciones de energía solar, eólica, nuclear y solar fotovoltaica proporcionadas por REE, el precio de la electricidad proporcionado por ESIOS (Entidad de Servicios de los Mercados Eléctricos) y el precio del gas proporcionado por MIBGAS.

Primero los datos son descargados en formato CSV gracias a una serie de scripts que se han diseñado gracias a un notebook de referencia [5]. Una vez guardados en local se exportan a la base de datos, no obstante, montones de datos proporcionados por las apis suelen ser indiferentes, por lo que se ha filtrado la información seleccionando la que se considera más significativa para la realización de los diversos estudios.

Demanda

La Figura 2.2.1 representa los atributos obtenidos más interesantes de la demanda energética del uno de enero de 2021 procedentes de REE. Estos incluyen información sobre la demanda real, programada y prevista de electricidad en diferentes intervalos de tiempo. La estructura de los datos incluye los siguientes campos.

1	Nombre	Fecha	Porcentaje	Consumo
2	Demanda real	2021-01-01 00:00:00+01:00	0,335742242	25134
3	Demanda real	2021-01-01 00:10:00+01:00	0,334560157	24907
4	Demanda real	2021-01-01 00:20:00+01:00	0,334290878	24787

Figura 2.2.1: Ejemplo de datos de demanda.

- *Tipo*: Este campo indica el tipo de dato, lo que nos permite distinguir entre diferentes tipos de mediciones o registros en el conjunto de datos.
- *Fecha*: La fecha y hora del registro son esenciales para comprender cuándo ocurrieron las mediciones de demanda eléctrica. Esto nos permite analizar patrones diarios, semanales o estacionales en la demanda de electricidad.
- *Porcentaje*: El porcentaje de la demanda en relación con un valor base puede proporcionar información sobre la carga relativa en el sistema eléctrico en un momento dado. Esto puede ser útil para evaluar la variabilidad de la demanda a lo largo del tiempo.
- *Valor*: El valor absoluto de la demanda en megavatios (MW) es un indicador directo de la cantidad de electricidad requerida en un determinado momento. Este dato es fundamental para comprender la carga en el sistema eléctrico y planificar la generación y distribución de energía de manera eficiente.

Generaciones

Los datos de generación de energía proporcionados por REE muestran la cantidad de energía generada por diferentes fuentes (por ejemplo, eólica, solar) en diferentes días. Podemos ver un fragmento de los datos obtenidos en la Figura [2.2.2](#).

1	Fecha	Nombre	Consumo
2	2021-01-01	Hidráulica	111077,579
3	2021-01-02	Hidráulica	112211,332
4	2021-01-03	Hidráulica	116969,983

Figura 2.2.2: Ejemplo de datos de generaciones.

La estructura de los datos incluye los siguientes campos:

- *Fecha*: Al igual que en el caso de la demanda, la fecha nos permite situar las mediciones de generación de energía en el tiempo y analizar patrones temporales en la producción de energía.
- *Tipo*: Este campo indica el tipo de generación de energía, lo que nos permite distinguir entre diferentes fuentes de energía (por ejemplo, eólica, solar, nuclear). Conocer la contribución de cada fuente de energía es crucial para comprender la composición del mix energético y su impacto en el sistema eléctrico.
- *Generación*: La cantidad de energía generada en megavatios-hora (MWh) proporciona información sobre la producción real de energía por cada fuente. Este dato es fundamental para evaluar la capacidad de generación y la disponibilidad de recursos energéticos renovables y no renovables.

Gas

Los datos de precios del gas proporcionados por MIBGAS muestran los precios del gas natural en diferentes momentos. Los datos de interés se representan en la Figura [2.2.3](#).

1	Fecha	Nombre	Precio
2	2021-01-01	GDAES_D+1	20,58
3	2021-01-02	GDAES_D+1	21,56
4	2021-01-03	GDAES_D+1	23,56

Figura 2.2.3: Ejemplo de datos de gas.

La estructura de los datos incluye los siguientes campos:

- *Fecha*: La fecha y hora del registro nos permiten situar los precios del gas en el tiempo y analizar patrones temporales en los precios del mercado del gas.
- *Producto*: Este campo indica el tipo de contrato de gas, lo que nos permite distinguir entre diferentes productos o servicios ofrecidos en el mercado del gas. Esto puede ser útil para evaluar la disponibilidad de diferentes tipos de contratos y sus precios asociados.
- *Precio*: El precio del gas en euros por megavatio-hora (€/MWh) es un indicador clave de los costos de producción y suministro de gas natural. Este dato es esencial para evaluar la rentabilidad de la generación de energía a partir de gas natural y su impacto en los precios de la electricidad.

Precio

ESIOS proporciona una amplia variedad de datos sobre el precio de la electricidad, incluidos los precios del mercado SPOT en diferentes intervalos de tiempo. Dado el volumen de datos, nos centraremos en los datos del mercado SPOT diario con datos intradiarios de la sesión del 1 al 7, limitados a un conjunto diario de datos. Podemos ver los datos en la Figura [2.2.4](#):

1	Fecha,Nombre,Precio
2	2021-01-01,Precio mercado SPOT Diario,1020.24
3	2021-01-02,Precio mercado SPOT Diario,1169.29
4	2021-01-03,Precio mercado SPOT Diario,1126.26

Figura 2.2.4: Ejemplo de datos de precio.

- *Precio*: El valor del precio de la electricidad en euros por megavatio-hora (€/MWh) es un indicador crucial de la dinámica del mercado eléctrico. Este dato nos permite entender los cambios en los precios de la electricidad a lo largo del tiempo y su impacto en los costos de producción y consumo de energía.
- *Fecha*: La fecha del registro es esencial para situar los precios de la electricidad en el tiempo y analizar patrones temporales en los precios del mercado eléctrico.
- *Nombre*: Estos campos proporcionan información sobre el indicador asociado con los precios de la electricidad, lo que nos permite entender el contexto y la naturaleza de los datos. En este caso, nos centramos en el precio del mercado SPOT diario.

2.3 Series temporales

Una serie temporal [32] es una secuencia de datos de puntos, típicamente consistentes en mediciones sucesivas hechas durante intervalos de tiempo. En el ámbito de la estadística y el análisis de datos, las series temporales se utilizan para observar cómo cambian las variables a lo largo del tiempo, permitiendo identificar patrones, tendencias y posibles estacionalidades.

2.3.1 Definición y características

Las series temporales se caracterizan por tener un orden específico en el tiempo, lo cual diferencia a estos datos de otros tipos de datos estadísticos. Esta dependencia temporal implica que el análisis de series temporales debe considerar el orden y el intervalo de tiempo entre las observaciones. Las principales características de una serie temporal incluyen:

- *Tendencia*: Componente de largo plazo que muestra la dirección general del movimiento de los datos a lo largo del tiempo.
- *Estacionalidad*: Patrón recurrente dentro de un intervalo de tiempo específico, como las fluctuaciones diarias, semanales, mensuales o anuales.
- *Ciclicidad*: Fluctuaciones que ocurren en períodos más largos y que no necesariamente son fijas en el tiempo, a diferencia de la estacionalidad.
- *Aleatoriedad*: Componente impredecible o ruido, que no sigue ningún patrón específico y puede ser causado por factores externos no medidos.

2.3.2 Importancia del análisis de series temporales

El análisis de series temporales es fundamental para tomar decisiones informadas basadas en datos históricos. Permite no solo la comprensión de los datos pasados sino también la predicción de comportamientos futuros, lo que es esencial para la planificación estratégica y la toma de decisiones en tiempo real. Además, el análisis de series temporales puede ayudar a identificar anomalías y cambios abruptos en los datos, lo que es crucial para el monitoreo y control en tiempo real de diversos sistemas.

2.3.3 Aplicación de series temporales

- *Economía y Finanzas*: Predicción de valores de mercado, análisis de tendencias económicas, y gestión de riesgos financieros.
- *Meteorología*: Pronósticos del tiempo y análisis de patrones climáticos.
- *Medicina*: Análisis de datos de pacientes, como ritmos cardíacos o niveles de glucosa en sangre.
- *Ingeniería*: Monitoreo y predicción de fallos en sistemas mecánicos o eléctricos.
- *Energía*: Predicción de la demanda y generación de energía, análisis de consumo eléctrico, etc.

En este último es en el que nos centraremos a lo largo de todo el desarrollo de este proyecto.

Tercera parte

III-SISTEMA DESARROLLADO

3 Objetivos

El propósito de este proyecto es desarrollar una solución integral que permita el análisis eficiente y comprensible del consumo eléctrico a través de una aplicación web robusta y escalable. A continuación, se detallan los objetivos principales del proyecto:

Objetivo 1: Desarrollo de una aplicación web robusta y escalable

Se desea crear una aplicación web accesible a través de navegadores estándar, que sea robusta, segura y capaz de escalar según la demanda de los usuarios.

Será necesaria la implementación de una infraestructura web que soporte el acceso concurrente y proporcione una experiencia de usuario fluida y fiable.

Objetivo 2: Integración y obtención de datos desde múltiples APIs

Será necesario integrar diversas APIs para la obtención de datos sobre el consumo eléctrico, asegurando la actualización constante y la precisión de la información.

Además, será necesario investigar en profundidad y aprender sobre el procesamiento de los mismos.

Objetivo 3: Realización de análisis

Es necesario desarrollar una comprensión profunda de los métodos y técnicas de análisis de datos aplicados al consumo eléctrico. Esto incluye el manejo de grandes volúmenes de datos, la aplicación de algoritmos de análisis avanzados y la interpretación de resultados.

Los conocimientos adquiridos servirán para la implementación de algoritmos de análisis con un enfoque en la precisión y la eficiencia.

Objetivo 4: Resultados de análisis

Los usuarios deben tener la posibilidad de consultar los resultados obtenidos a través de los análisis realizados. La aplicación debe proporcionar una interfaz clara y fácil de usar para acceder a estos datos.

Objetivo 5: Visualización de resultados

La aplicación debe ser capaz de representar visualmente los resultados de los análisis mediante gráficas, lo que facilitará la interpretación y comprensión de los datos por parte de los usuarios.

Estos objetivos marcan el punto de partida para iniciar el proceso de análisis de requisitos, donde se definirán con mayor detalle las funcionalidades y características específicas que la aplicación deberá cumplir para satisfacer las necesidades de los usuarios y alcanzar los objetivos del proyecto.

4. Análisis

En esta sección se realiza un planteamiento del sistema a desarrollar, es fundamental realizar correctamente esta sección para tener claro producto a desarrollar a futuro.

4.1 Requisitos

En esta sección se recaban las especificaciones y funcionalidades que deben cumplirse para que la aplicación satisfaga las necesidades del usuario y cumpla con sus objetivos. Estos requisitos son la base sobre la cual se desarrolla, diseña y prueba la aplicación web.

4.1.1 Roles

A lo largo de este proyecto, se han definido los siguientes roles y responsabilidades para asegurar el éxito en su desarrollo y entrega ([Cuadro 4.1.1](#)):

Cuadro 4.1.1: Roles.

Rol	Funciones
Jefe de proyectos	<ul style="list-style-type: none">- Responsable de la planificación, ejecución y cierre del proyecto.- Supervisar y coordinar al equipo de trabajo.- Asegurar que el proyecto se desarrolle dentro del alcance, tiempo y presupuesto definidos.- Gestionar los riesgos y resolver problemas que puedan surgir durante el proyecto.- Mantener comunicación constante con los stakeholders y reportar el progreso del proyecto.
Analista	<ul style="list-style-type: none">- Recoger y documentar los requisitos del cliente.- Analizar y modelar los procesos de negocio para entender las necesidades del sistema.- Colaborar con el equipo de desarrollo para asegurar que los requisitos se entienden y se implementan correctamente.- Participar en la validación y verificación del software para asegurar que cumple con los requisitos especificados.- Asistir en la resolución de problemas y en la mejora continua del sistema.
Programador	<ul style="list-style-type: none">- Diseñar y desarrollar el software conforme a los requisitos especificados.- Escribir y mantener el código fuente, asegurando su calidad y eficiencia.- Realizar pruebas unitarias y de integración para garantizar el correcto funcionamiento del software.- Colaborar con los analistas para resolver dudas y clarificar requisitos.- Participar en la revisión del código y en la implementación de mejoras y optimizaciones.

4.1.2 Requisitos de información

Los requisitos definidos en el Cuadro [4.1.2](#) indican qué información debe ser recopilada, cómo debe ser organizada y cómo debe ser utilizada dentro de la aplicación para satisfacer las necesidades del usuario. Los requisitos de información abarcan aspectos como la estructura de los datos, la integridad de la información, los formatos de entrada y salida, y los procedimientos para acceder y manipular los datos dentro de la aplicación.

En la tabla de más abajo se recaban los requisitos en el contexto de la aplicación desarrollada:

Cuadro 4.1.2: Requisitos de información.

ID	Rol	Título	Descripción
RI-001	Analista	Usuarios	<p>Como analista quiero tener la información de los usuarios para tener un registro con sus datos para futuras funcionalidades.. Los campos que se recaban de los mismos son:</p> <ul style="list-style-type: none"> - <i>nombre</i> - <i>apellidos</i> - <i>nombre de usuario</i> - <i>email</i> - <i>contraseña</i>
RI-002	Analista	Datos	<p>Como analista quiero disponer de todos los datos disponibles de la demanda eléctrica, los precios de gas y electricidad, así como las generaciones solar, eólica, fotovoltaica y nuclear para realizar análisis completos. Los componentes más interesantes almacenados son:</p> <ul style="list-style-type: none"> - <i>precio</i> - <i>consumo</i> - <i>porcentaje</i> - <i>fecha</i> - <i>tipo de dato</i> - <i>nombre</i> <p>El <i>tipo de dato</i> es un enumerado que nos identifica un conjunto completo de datos. Pueden tomar los siguientes valores:</p> <ul style="list-style-type: none"> - <i>gas</i> - <i>demand</i> - <i>precio</i> - <i>generaciones</i> <p>El <i>nombre</i> es un enumerado que nos indica la categoría de información de los datos. Pueden tomar los siguientes valores:</p>

			<ul style="list-style-type: none"> - <i>solar</i> - <i>eólica</i> - <i>nuclear</i> - <i>hidráulica</i> - <i>demandas real</i> - <i>demandas programadas</i> - <i>demandas previstas</i> - <i>GADES D+1</i> - <i>precio mercado SPOT intradiario sesión 1</i> - <i>precio mercado SPOT intradiario sesión 2</i> - <i>precio mercado SPOT intradiario sesión 3</i> - <i>precio mercado SPOT intradiario sesión 4</i> - <i>precio mercado SPOT intradiario sesión 5</i> - <i>precio mercado SPOT intradiario sesión 6</i> - <i>precio mercado SPOT intradiario sesión 7</i> - <i>precio mercado SPOT diario</i>
RI-003	Analista	Análisis	<p>Como analista quiero guardar los parámetros necesarios para generar un análisis con los datos obtenidos.</p> <ul style="list-style-type: none"> - <i>uuid</i> - <i>nombre</i> - <i>descripción</i> - <i>fecha de creación</i> - <i>id usuario</i> - <i>frecuencia</i> <p>El <i>uuid</i> es un identificador integrado con motivos de seguridad. La <i>frecuencia</i> es un enumerado indica la frecuencia del análisis. Pueden tomar los siguientes valores:</p> <ul style="list-style-type: none"> - <i>10 minutos</i> - <i>horas</i> - <i>días</i> - <i>meses</i> - <i>años</i>.
RI-004	Analista	Datos Preprocesados	<p>Como analista quiero almacenar todos los datos preprocesados resultantes del análisis con una determinada frecuencia para poder realizar los tipos de análisis correctamente. Estos son los datos almacenados:</p> <ul style="list-style-type: none"> - <i>tipo de dato</i>

			<ul style="list-style-type: none"> - <i>nombre</i> - <i>precio</i> - <i>consumo</i> - <i>porcentaje</i> - <i>fecha</i> <p>El <i>precio</i>, <i>consumo</i>, <i>porcentaje</i> y <i>fecha</i> deben variar en función de la frecuencia seleccionada durante el análisis.</p>
RI-005	Analista	Autocorrelación	<p>Como analista quiero disponer de los datos parametrizables para generar una gráfica de autocorrelación para visualizar el resultado de un caso de estudio.</p> <p>Estos son los parámetros necesarios::</p> <ul style="list-style-type: none"> - <i>uuid</i> - <i>lag</i> - <i>fecha</i> - <i>título</i> - <i>nombre</i> - <i>tipo</i> - <i>método</i> - <i>estilos</i> <p>El <i>tipo</i> es un enumerado que indica el tipo de autocorrelación que queremos realizar.</p> <p>Puede tomar los valores:</p> <ul style="list-style-type: none"> - <i>parcial</i> - <i>simple</i> <p>El <i>método</i> es un enumerado que representa los métodos que se pueden emplear para un estudio de autocorrelación.</p> <p>Puede tomar los valores:</p> <ul style="list-style-type: none"> - <i>spearman</i> - <i>pearson</i>. <p>El <i>estilo</i> es un enumerado que se emplea para seleccionar el modo de visualización de los resultados del estudio que deseemos.</p> <p>Puede tomar los valores:</p> <ul style="list-style-type: none"> - <i>simple</i> - <i>puntos</i>
RI-006	Analista	Outliers	<p>Como analista quiero disponer de los datos parametrizables para generar una gráfica sobre detección de outliers para identificar anomalías en los datos.</p> <p>Estos son los parámetros necesarios:</p>

			<ul style="list-style-type: none"> - <i>uuid</i> - <i>umbral</i> - <i>métodos</i> - <i>título</i> - <i>nombre</i> - <i>fecha</i> - <i>estilos</i> <p>El <i>método</i> es un enumerado que representa los métodos que se pueden emplear para un estudio de detección de outliers. Puede tomar los valores:</p> <ul style="list-style-type: none"> - <i>desviación estándar</i> - <i>rango intercuartílico</i> <p>El <i>estilo</i> es un enumerado que se emplea para seleccionar el modo de visualización de los resultados del estudio que deseemos. Puede tomar los valores:</p> <ul style="list-style-type: none"> - <i>simple</i> - <i>histograma</i> - <i>boxplot</i>
RI-007	Analista	Descomposición	<p>Como analista quiero disponer de los datos parametrizables para generar una gráfica sobre descomposición de series temporales para analizar componentes individuales de la serie. Estos son los parámetros necesarios:</p> <ul style="list-style-type: none"> - <i>uuid</i> - <i>título</i> - <i>nombre</i> - <i>fecha</i> - <i>ventana estacionalidad</i> - <i>ventana tendencia</i> - <i>métodos</i> - <i>componentes</i> - <i>suavizado</i> <p>El <i>método</i> es un enumerado que representa los métodos que se pueden emplear para un estudio de descomposición de series temporales. Puede tomar los valores:</p> <ul style="list-style-type: none"> - <i>aditiva</i> - <i>multiplicativa</i> <p>El <i>estilo</i> es un enumerado que se emplea para seleccionar el modo de visualización de</p>

			<p>los resultados del estudio que deseemos. Puede tomar los valores:</p> <ul style="list-style-type: none"> - <i>simple</i> - <i>gráfico de área</i> - <i>gráfico de barras</i>
RI-008	Analista	Resultados Autocorrelación	<p>Como analista quiero almacenar todos los estudios seleccionados del tipo de análisis de autocorrelación para poder revisarlos y compararlos posteriormente. Son necesarios los atributos:</p> <ul style="list-style-type: none"> - <i>lag</i> - <i>valor</i> - <i>nombre dato</i>
RI-009	Analista	Resultados Outliers	<p>Como analista quiero almacenar todos los estudios seleccionados del tipo de análisis de detección de outliers para poder revisarlos y compararlos posteriormente. Son necesarios los atributos:</p> <ul style="list-style-type: none"> - <i>valor</i> - <i>nombre dato</i> - <i>fecha dato</i>
RI-010	Analista	Resultados Descomposición	<p>Como analista quiero almacenar todos los estudios seleccionados del tipo de análisis de descomposición de series temporales para poder revisarlos y compararlos posteriormente. Son necesarios los atributos:</p> <ul style="list-style-type: none"> - <i>valor</i> - <i>nombre dato</i> - <i>fecha dato</i> - <i>tendencia</i> - <i>estacionalidad</i> - <i>residuo</i>
RI-011	Analista	Gráfica	<p>Como analista quiero almacenar las gráficas generadas para documentar visualmente los análisis realizados. Son necesarios los atributos:</p> <ul style="list-style-type: none"> - <i>fecha creación</i> - <i>título</i> - <i>tipo de dato</i> - <i>imagen html</i>

4.1.3 Requisitos funcionales

Los requisitos funcionales (Cuadro 4.1.3) describen las acciones específicas que el sistema debe ser capaz de realizar para cumplir con los objetivos del proyecto. Estos requisitos se centran en las funciones y características que deben estar presentes en el sistema para satisfacer las necesidades de los usuarios finales y por tanto son la base sobre la cual se desarrolla la lógica y la funcionalidad del sistema.

Cuadro 4.1.3: Requisitos funcionales.

ID	Rol	Título	Descripción
RF-001	Programador	Descargar datos	Como programador quiero que la aplicación pueda obtener datos de MIBGAS, ESIOS o REE para proporcionar un producto adecuado.
RF-002	Programador	Registro, login y logout	Como programador quiero que los usuarios puedan registrarse, iniciar sesión y desloguearse en la aplicación para gestionar su acceso y autenticación.
RF-003	Programador	Autocorrelación/parcial	Como programador quiero que la aplicación realice análisis de autocorrelación y autocorrelación parcial por los métodos de Pearson o Spearman para evaluar relaciones temporales.
RF-004	Programador	Detección de outliers	Como programador quiero que la aplicación realice análisis y genere gráficas de detección de outliers usando desviación estándar o rango intercuartílico para identificar anomalías en los datos.
RF-005	Programador	Descomposición de series temporales	Como programador quiero que la aplicación realice análisis de descomposición de series temporales y obtenga resultados con representaciones aditivas o multiplicativas para analizar componentes individuales.
RF-006	Programador	Resultados	Como programador quiero que los usuarios consulten los resultados de cada tipo de análisis en una tabla para facilitar su revisión y evaluación.
RF-007	Programador	Descargar datos	Como programador quiero que los usuarios puedan descargar los datos de las tablas en formato CSV para su uso y análisis fuera de la

			aplicación.
RF-008	Programador	Estilos	Como programador quiero que las gráficas generadas se representen con diferentes estilos para mejorar la visualización y comprensión de los datos.
RF-009	Programador	Descargar resultados	Como programador quiero que los usuarios puedan descargar los resultados de los estudios en formato CSV para su uso y análisis fuera de la aplicación.
RF-010	Programador	Descargar gráficas	Como programador quiero que los usuarios puedan descargar las gráficas como imágenes para tener una copia en local.
RF-011	Programador	PDF	Como programador quiero que el analista pueda descargar los resultados de sus análisis en formato PDF para su fácil distribución y revisión.
RF-012	Programador	API	Como programador quiero que cualquier persona con permiso pueda acceder a los datos a través de una API para obtener los resultados de los análisis realizados en la aplicación.

4.1.4 Requisitos no funcionales

Los requisitos no funcionales (Cuadro 4.1.4) definen atributos del sistema que no están directamente relacionados con las funciones específicas del sistema, pero que son igualmente importantes para su éxito y aceptación por parte de los usuarios. Estos requisitos abordan aspectos como el rendimiento, la seguridad, la usabilidad y la fiabilidad del sistema. Nos centraremos en la calidad global del sistema y en cómo se comporta en diferentes situaciones y contextos de uso.

Categorías:

- Usabilidad: Describe los requisitos relacionados con la facilidad de uso, la accesibilidad y la experiencia del usuario para garantizar que el sistema sea intuitivo, fácil de aprender y utilizar por una amplia variedad de usuarios.
- Portabilidad: Define los requisitos relacionados con la capacidad del sistema para funcionar de manera efectiva en diferentes entornos de hardware, sistemas operativos o plataformas tecnológicas sin necesidad de modificaciones significativas.
- Rendimiento: Establece los estándares de rendimiento que el sistema debe cumplir en términos de velocidad, tiempo de respuesta y capacidad de manejo de carga para garantizar una experiencia de usuario óptima.
- Fiabilidad: Especifica los niveles de disponibilidad, confiabilidad y tolerancia a fallos que el sistema debe mantener para garantizar la continuidad del servicio y la recuperación de datos en caso de fallos o interrupciones.

Cuadro 4.1.4: Requisitos no funcionales.

ID	Rol	Título	Descripción	Categoría
RNF-001	Programador	Error 404	Como programador quiero que la aplicación redireccione a los usuarios a una vista 404 cuando no encuentre la URL solicitada para mejorar la experiencia del usuario.	Usabilidad
RNF-002	Programador	Error 500	Como programador quiero que la aplicación redireccione a los usuarios a una vista de error 500 cuando ocurra un error del sistema para informar adecuadamente al usuario.	Usabilidad
RNF-003	Programador	Web Responsive	Como programador quiero que la interfaz se visualice correctamente desde cualquier dispositivo para asegurar la accesibilidad.	Portabilidad
RNF-004	Programador	Intuitiva	Como programador quiero que la página aporte claridad y facilidad para comprender las funciones disponibles para mejorar la usabilidad.	Usabilidad
RNF-005	Programador	Actualización Automática	Como programador quiero que la base de datos se actualice automáticamente para asegurar la disponibilidad de datos recientes.	Fiabilidad
RNF-006	Programador	Seguridad	Como programador quiero que los análisis sean seguros, garantizando que otros usuarios no puedan acceder a la información del análisis para proteger la privacidad de los datos.	Seguridad

4.1.5 Reglas de negocio

Las reglas de negocio (Cuadro 4.1.5) son declaraciones que definen o restringen algún aspecto del negocio y que tienen un impacto directo en su operación. Estas reglas establecen políticas, procedimientos o restricciones que guían el comportamiento, las decisiones y las actividades dentro de una organización.

Cuadro 4.1.5: Reglas de negocio.

ID	Rol	Título	Descripción
RN-001	Programador	Usuario registrado	Como programador quiero que el usuario deba estar registrado para poder realizar cualquier análisis para asegurar que solo los usuarios autorizados tengan acceso a las funcionalidades de análisis.
RN-002	Programador	Limitación de Análisis por Usuario	Como programador quiero que el analista solo pueda hacer un análisis al mismo tiempo por motivos de rendimiento del sistema para evitar sobrecargas y asegurar un funcionamiento óptimo.
RN-003	Programador	Cumplimiento de Normativas de Protección de Datos	Como programador quiero que la aplicación cumpla con todas las normativas y regulaciones de protección de datos para garantizar la privacidad y seguridad de la información del usuario. Esto incluye el almacenamiento seguro de datos, la obtención del consentimiento del usuario para el procesamiento de datos y la implementación de medidas de seguridad adecuadas para proteger los datos contra accesos no autorizados.

4.1.6 Matriz de trazabilidad

En este apartado indicaremos las dependencias entre los requisitos de información, requisitos funcionales, no funcionales y reglas de negocio especificadas anteriormente en relación con los objetivos generales del proyecto. El objetivo es entender claramente cómo alcanzar los objetivos.

El Cuadro [4.1.6](#) representa la matriz que relaciona estos puntos:

Cuadro 4.1.6 Matriz de trazabilidad

Requisitos	Objetivo-1	Objetivo-2	Objetivo-3	Objetivo-4	Objetivo-5
RI-001			X		
RI-002		X	X		
RI-003			X	X	
RI-004			X	X	
RI-005			X		
RI-006			X		
RI-007			X		
RI-008				X	X
RI-009				X	X
RI-010				X	X
RI-011				X	X
RF-001		X			
RF-002			X		
RF-003			X		
RF-004			X		
RF-005			X		
RF-006				X	
RF-007			X	X	
RF-008			X		X
RF-009				X	X
RF-010				X	X
RF-011				X	X

RF-012					X
RNF-001	X				
RNF-002	X				
RNF-003	X				
RNF-004	X				
RNF-005	X				
RNF-006	X				
RN-001			X		
RN-002			X		
RN-003			X		

4.2 Casos de uso

En esta sección describiremos cómo interactúan los usuarios que utilizan la aplicación con cada una de sus funcionalidades y el sistema (Figura 4.2).

- Realización de análisis: El usuario podrá realizar un análisis, el cual debe incluir los siguientes elementos:
 - Obtener datos: El usuario debe seleccionar los datos de los cuales desea realizar el estudio.
 - Autocorrelación: El análisis debe proporcionar un estudio de la autocorrelación con los valores de los datos seleccionados.
 - Detección de outliers: El análisis debe proporcionar un estudio de la detección de outliers con los valores de los datos seleccionados.
 - Descomposición de series temporales: El análisis debe proporcionar un estudio sobre la descomposición de series temporales con los valores de los datos seleccionados.
- Consultar resultados: El analista debe poder consultar los resultados de sus análisis.
- Consultar gráficos: El analista debe poder consultar los resultados de sus análisis representados viéndolos representados en una gráfica.
- Descargar resultados: El usuario analista debe poder descargar los resultados de cualquiera de sus análisis.

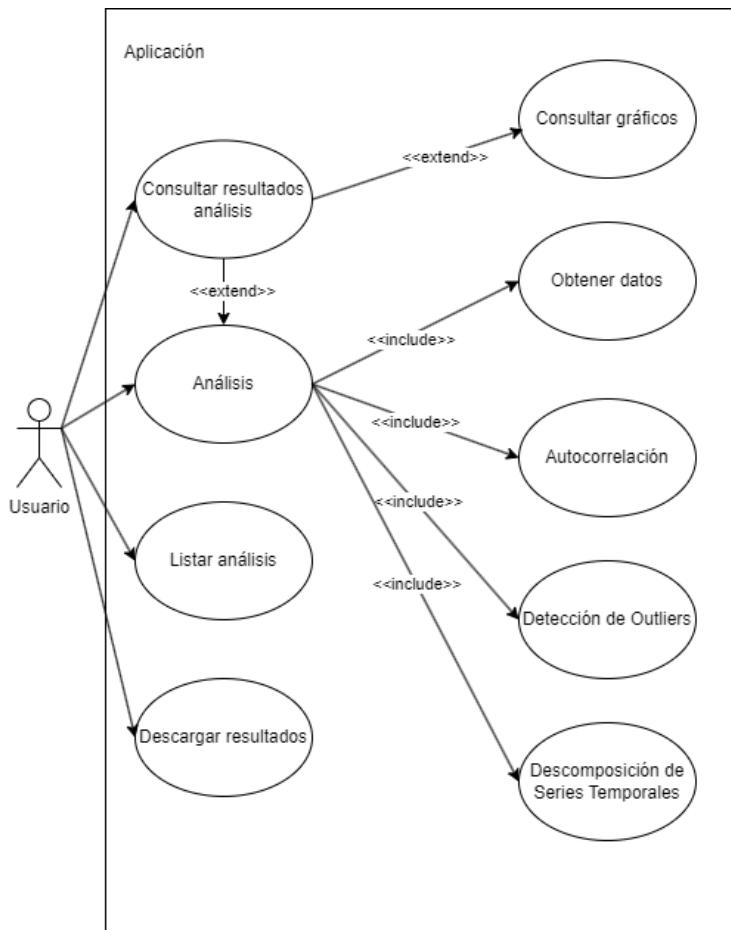


Figura 4.2: Casos de uso.

5 Riesgos

En el desarrollo de aplicaciones web, la identificación y gestión de riesgos es un componente crítico para asegurar el éxito del proyecto. Los riesgos en un proyecto pueden surgir de diversas fuentes, incluyendo problemas técnicos, cambios en los requisitos, limitaciones de recursos y amenazas a la seguridad. Es fundamental conocer las adversidades a las que nos podremos enfrentar con motivo de implementar estrategias efectivas de mitigación de riesgos, garantizando que el desarrollo cumpla con los estándares de calidad, plazos y presupuestos establecidos.

En esta sección, se analizarán los principales riesgos asociados con el proyecto y se propondrán medidas preventivas y correctivas para abordarlos de manera efectiva.

5.1 Análisis cualitativo y cuantitativo de los riesgos

Abordando el cálculo del valor de riesgo (nivel), es esencial recordar que este proceso implica evaluar múltiples dimensiones críticas, incluyendo el alcance, el tiempo y el coste. Estos factores desempeñan un papel fundamental en la determinación del nivel de riesgo y se ponderan de la siguiente manera: alcance con un 20%, tiempo con un 50% y coste con un 30%. El valor de riesgo se obtiene a través de una fórmula que toma en cuenta la probabilidad, el impacto y el peso del impacto, distribuyendo equitativamente la importancia de estos aspectos en la evaluación.

- Alcance 0.2
- Tiempo 0.5
- Coste 0.3

$$\text{valor riesgo} = \sum (\text{probabilidad} \times \text{impacto} \times \text{peso_impacto}) / 3$$

Se obtiene el nivel para cada riesgo, el cual categorizamos numéricamente, tomando los valores que podemos ver en el Cuadro [5.1](#).

Cuadro 5.1 Riesgos por Impacto-Probabilidad

		Probabilidad				
		10%	30%	50%	70%	90%
Impacto	1	0.1	0.3	0.5	0.7	0.9
	3	0.3	0.9	1.5	2.1	2.7
	5	0.5	1.5	2.5	3.5	4.5
	7	0.7	2.1	3.5	4.9	6.3
	9	0.9	2.7	4.5	6.3	8.1

5.2 Tabla de riesgos

En el Cuadro [5.2](#) se mostrarán todos los posibles riesgos a los que podremos enfrentarnos a lo largo del desempeño del proyecto.

Cuadro 5.2 Riesgo

ID	Categoría	Riesgo	Tipo	Alcance	Tiempo	Costes	Probabilidad	Valor riesgo (NIVEL)	Estrategia
R-001	SEGURIDAD	Seguridad de datos y privacidad	Negativo	9	9	9	50%	4,5	MITIGAR
R-002	MERCADO	Competencia en línea	Negativo	1	3	5	90%	2,88	EVITAR
R-003	CALIDAD	Calidad del producto insatisfactoria	Negativo	7	3	7	30%	1,5	MITIGAR
R-004	ORGANIZACIÓN	Falta de implicación durante el desempeño.	Negativo	9	9	9	10%	0,9	MITIGAR
R-005	TIEMPO	Retrasos en la entrega.	Negativo	3	1	5	70%	1,82	MITIGAR
R-006	COSTE	Exceder el presupuesto.	Negativo	1	1	9	30%	1,02	EVITAR
R-007	TÉCNICOS	Falta de conocimientos técnicos.	Negativo	9	9	3	50%	3,6	MITIGAR
R-008	ESTIMACIÓN	Holgura durante la fase de desarrollo	Negativo	9	9	9	70%	2,1	MITIGAR
R-009	REQUISITOS	Requisito identificado	Negativo	9	1	1	30%	0,26	MITIGAR

5.3 Plan de respuesta a riesgos

En la siguiente tabla (Cuadro 5.3) mostraremos las posibles soluciones para resolver los problemas previamente mencionados.

Cuadro 5.3 Respuesta a riesgos

ID	DESCRIPCIÓN	Respuesta Preventiva	Respuesta Correctiva
R-001	Seguridad de datos y privacidad	Implementar cifrado de datos y políticas de seguridad robustas.	Realizar auditorías de seguridad.
R-002	Competencia en línea	Realizar análisis de mercado y estrategias de diferenciación.	Mejora de las funcionalidades del producto.
R-003	Calidad del producto insatisfactoria	Establecer estándares de calidad y pruebas exhaustivas.	Implementar revisiones de calidad y planes de mejora continua.
R-004	Falta de implicación	Realizar reuniones regulares de seguimiento y motivación.	Redistribuir tareas y reforzar la comunicación entre el cliente y el desarrollador.
R-005	Retrasos en la entrega	Planificar con márgenes de tiempo y seguimiento del cronograma.	Reasignar recursos y ajustar el cronograma para recuperar el tiempo perdido.
R-006	Exceder el presupuesto	Monitorear el gasto y establecer un presupuesto detallado.	Revisar y ajustar el presupuesto, recortar gastos innecesarios.
R-007	Falta de conocimientos técnicos	Proporcionar formación y documentación.	Consultar con expertos externos y realizar capacitación adicional.
R-008	Holgura en fases de desarrollo	Establecer horario de trabajo para distribuir correctamente el tiempo.	Ajustar las fases de desarrollo y reasignar recursos según sea necesario.
R-009	Requisito no identificado	Realizar revisiones exhaustivas de requisitos y análisis detallado con los interesados.	Realizar reuniones de seguimiento para identificar y gestionar los nuevos requisitos.

6 Metodología

En esta sección, hablaremos del conjunto de principios, prácticas y procesos estructurados que se siguen durante el ciclo de vida de desarrollo del proyecto.

En la fase de inicio del proyecto, queda claro que abordamos un producto del que se desconoce en un porcentaje muy alto la tecnología necesaria para su implementación, lo que conlleva a montones de horas investigando cómo cumplir con cada uno de los requisitos necesarios solicitados por el cliente. Además, podemos darnos cuenta de que para el desarrollo se sigue un ritmo secuencial, causado por las muchas dependencias entre tareas tal y como vemos en el cronograma del apartado [8.4](#).

Aunque esta metodología tradicionalmente implica una progresión lineal a través de las fases de requisitos, diseño, implementación, verificación y mantenimiento, en este proyecto no se ha seguido de forma estricta. En lugar de ello, se ha decidido dejar la fase de pruebas (testing) para el final del ciclo de desarrollo. Esto se debe a que la mayoría de las fases de desarrollo no requieren pruebas continuas, permitiendo un ajuste más eficiente del tiempo de trabajo.

Esta adaptación permite abordar de manera efectiva las complejidades y dependencias del proyecto, al mismo tiempo que se asegura que el producto final sea probado exhaustivamente antes de su entrega.

A continuación hablaremos en detalle sobre la distribución del trabajo desde que dió comienzo el diseño del producto hasta su finalización en cuatro fases.

6.1 Fases del modelo en cascada

1. *Primera fase Familiarización y exploración (Primeros dos meses):*

Esta fase inicial, consistió en la etapa de planificación del proyecto, se dedicó mucho tiempo a familiarizarse con Django y los conceptos teóricos relacionados con los análisis de machine learning y de una correcta comprensión de los requisitos principales del proyecto. Además, fue necesario definir y obtener correctamente todos los datos de fuentes externas, un elemento clave para el desempeño del producto. Esta etapa fue crucial para establecer una base sólida de conocimientos técnicos y teóricos, asegurando una comprensión profunda de las herramientas y técnicas que se utilizarían en el proyecto.

2. *Segunda fase prototipo inicial (Tercer mes):*

Durante el tercer mes, se realizó el preprocesado y análisis de datos. Esta fase involucró la creación de un prototipo inicial que permitiera visualizar y entender los datos con los que se trabajaría. Se implementaron técnicas de limpieza y transformación de datos para preparar el conjunto de datos para el análisis posterior, así como la implementación tareas asíncronas.

3. *Tercera fase Implementación de algoritmos (Siguientes cuatro semanas):*

- *Primeras dos semanas*: Se implementó el algoritmo de Autocorrelación. Esta funcionalidad fue desarrollada y probada de manera iterativa, asegurando que el algoritmo funciona correctamente con los datos preprocesados.
 - *Tercera semana*: Se añadió el algoritmo de Detección de Outliers.
 - *Cuarta semana*: Se implementó el tipo de análisis de la Descomposición de Series temporales.
4. *Cuarta fase Desarrollo de funcionalidades adicionales (Última semana)*:
- En la última fase del proyecto, se desarrolló una API y otras funcionalidades adicionales necesarias para completar la aplicación web. Estas funcionalidades fueron implementadas de manera rápida y eficiente, aprovechando el conocimiento y la experiencia adquirida en las fases anteriores del proyecto.

6.2 Gestión de código con GitHub

Para gestionar el código del proyecto, se utilizó un repositorio en GitHub. Aunque no se siguió una estructura de commits formal, se realizaron commits una vez que cada requisito fue completamente implementado. Esta estrategia ayudó a mantener un historial de cambios claro y organizado, facilitando la revisión y el seguimiento del progreso del proyecto.

7. Diseño

En este apartado, exploramos los elementos fundamentales que componen la estructura y funcionalidad de nuestro proyecto, destacando su enfoque en el diseño integral. Analizamos la concepción del sistema desde una perspectiva arquitectónica, resaltando las técnicas y tecnologías empleadas para su desarrollo.

Además, examinaremos el diseño de la API, elemento central para la interacción y comunicación dentro del sistema.

7.1 Diseño de tipos de análisis

El análisis de datos es un componente fundamental en el proceso de extracción de conocimiento a partir de conjuntos de datos. En el contexto del desarrollo web, lo principal es que el usuario seleccione los datos de los cuales quiere obtener el análisis haciendo un preprocesado [6] previo especificando la frecuencia de los mismos y así poder estudiar los resultados obtenidos tras aplicar las técnicas de las que hablaremos en profundidad a continuación:

7.1.1 Gráficas de autocorrelación

Las gráficas de autocorrelación [7] desempeñan un papel crucial en la exploración de la dependencia temporal presente en los datos de consumo eléctrico español. Al aplicar estas técnicas a los datos obtenidos de las APIs de ESIOS, REE y MIBGAS, podemos identificar patrones de correlación entre los valores del conjunto de datos seleccionado en diferentes intervalos de tiempo. Esto nos permite comprender mejor la estructura temporal de los datos seleccionados, incluyendo posibles tendencias, estacionalidades y ciclos recurrentes.

Para que vean la estructura, la Figura 7.1.1 representa un ejemplo de gráfica de autocorrelación parcial que podréis generar en la aplicación:

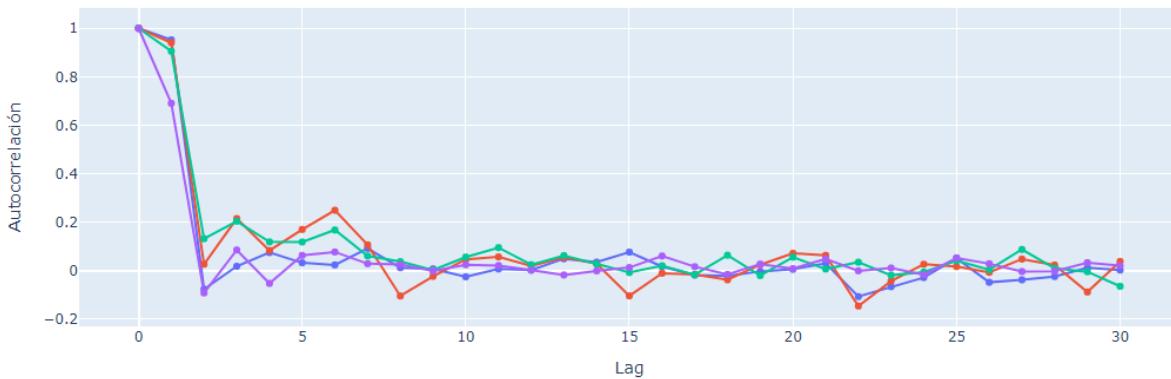


Figura 7.1.1: Ejemplo de autocorrelación parcial.

Dentro del contexto de esta aplicación, la capacidad de ajustar y personalizar el análisis es fundamental para proporcionar a los usuarios una experiencia significativa y adaptable a sus necesidades específicas. En este sentido, los requisitos parametrizables e interactivos de las gráficas de autocorrelación y autocorrelación parcial adquieren una importancia destacada.

- El **Lag o retraso**, que permite especificar el número de períodos de retraso para calcular la autocorrelación. Esto es crucial, ya que el usuario puede estar interesado en analizar la correlación entre los datos en diferentes momentos del tiempo, y ajustar el lag le permite explorar diferentes intervalos temporales con mayor detalle.
- Además, el **Método de cálculo** es otra variable ajustable que permite al usuario elegir entre diferentes métodos de cálculo de autocorrelación, como el método de Pearson o el método de Spearman. Esta opción brinda al usuario la capacidad de adaptar el análisis según las características específicas de los datos o las preferencias metodológicas.

Método de Pearson:

El coeficiente de correlación de Pearson [8] mide la fuerza y la dirección de la relación lineal entre dos variables. Para calcular la autocorrelación utilizando el método de Pearson, se sigue la siguiente fórmula:

$$r_k = \frac{\sum_{i=1}^{n-k} (x_i - \bar{x})(x_{i+k} - \bar{x})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (x_{i+k} - \bar{x})^2}}$$

donde:

- r es el coeficiente de correlación de Pearson.
- x_i son los valores de una variable.
- \bar{x} es la media del conjunto de datos.
- k es el lag o retraso.

Método de Spearman:

El coeficiente de correlación de Spearman [9] mide la fuerza y la dirección de la relación monótona entre dos variables. Este método se basa en los rangos de los datos en lugar de los valores originales. La fórmula para calcular la correlación de Spearman es similar a la de Pearson, pero aplicada a los rangos de los datos:

$$r_s = \frac{\sum_{i=1}^{n-k} (R(x_i) - \bar{R})(R(x_{i+k}) - \bar{R})}{\sqrt{\sum_{i=1}^n (R(x_i) - \bar{R})^2 \sum_{i=1}^n (R(x_{i+k}) - \bar{R})^2}}$$

- Por último, los **estilos de la visualización de resultados** es una variable clave que permite al usuario elegir entre gráficos de líneas o representación por puntos. Esta flexibilidad en la presentación de los resultados facilita la interpretación y comprensión de los patrones de correlación identificados, permitiendo al usuario explorar visualmente la estructura temporal de los datos de consumo eléctrico.

7.1.2 Descomposición de series temporales

Nos permite analizar la tendencia, la estacionalidad y los componentes residuales presentes en los datos de consumo eléctrico español. Al aplicar esta técnica a los datos obtenidos [10], podemos desglosar la demanda eléctrica en sus factores subyacentes y entender cómo estos contribuyen a la variabilidad observada en los datos.

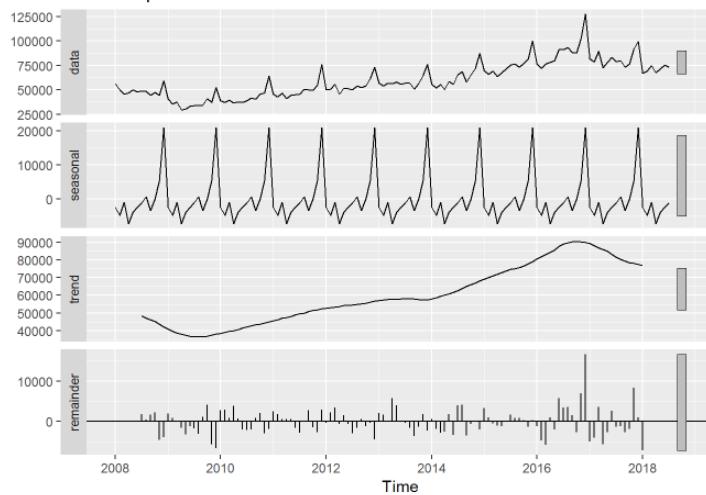


Figura 7.1.2: Ejemplo de descomposición aditiva.

Las variables ajustables en el contexto de esta aplicación interactivas en este caso son:

- Los **Métodos de descomposición** [11], que permite al usuario elegir entre diferentes estilos, como el aditivo o multiplicativo. Esta opción brinda al usuario la flexibilidad de seleccionar el método que mejor se ajuste a la naturaleza de los datos y a los objetivos del análisis.

Descomposición Aditiva:

En el método aditivo, se asume que los componentes de la serie temporal (tendencia, estacionalidad y residuos) se combinan de manera aditiva. Esto es adecuado cuando la variación estacional es aproximadamente constante a lo largo del tiempo.

La fórmula para la descomposición aditiva es:

$$Y_t = T_t + S_t + R_t$$

donde:

- Y_t = es el valor observado de la serie temporal en el tiempo t .
- T_t = es el componente de tendencia en el tiempo t .
- S_t = es el componente de estacionalidad en un tiempo t .
- R_t = es el componente de error o residual en ese tiempo t .

Descomposición Multiplicativa:

En el método multiplicativo, se asume que los componentes de la serie temporal se combinan de manera multiplicativa. Esto es adecuado cuando la variación estacional cambia proporcionalmente con el nivel de la serie temporal.

La fórmula para la descomposición multiplicativa es:

$$Y_t = T_t \times S_t \times R_t$$

- La **Visualización de componentes** es otro aspecto crucial que se puede ajustar en la descomposición. Esto permite al usuario elegir qué componentes de la serie temporal desea visualizar, como la tendencia, la estacionalidad y el componente residual. Esta capacidad de personalización facilita la interpretación de los resultados y permite al usuario enfocarse en los aspectos específicos de la serie temporal que son de mayor interés para su análisis.
Entre los distintos métodos de visualización encontraremos el simple o de líneas, parecido al de la Figura [7.1.2](#) de arriba, así como mediante un Gráfico de barras o un gráfico de área según la preferencia del analista.
- Por último, para establecer un rango de datos más reducido siempre y cuando el analista lo considere oportuno, se podrá introducir un rango de fechas **Fecha Inicio** y **Fecha Fin** para filtrar los datos según dicho rango, pudiendo hacer un estudio más riguroso.
- También se podrán introducir dos **Ventanas**, una para la **tendencia** y otra para la **estacionalidad**, lo que permite al usuario ajustar el nivel de detalle con el que se suavizan estos componentes. La ventana para la tendencia determina el período sobre el cual se promedia la serie temporal para identificar la tendencia a largo plazo, mientras que la ventana para la estacionalidad controla la longitud del ciclo estacional considerado en el análisis. Ajustando estas ventanas, el usuario puede influir en la sensibilidad del análisis a cambios rápidos (tendencia) o patrones repetitivos (estacionalidad).
- El **Factor de suavizado** se trata de un ajuste adicional, que permite al usuario controlar el nivel de suavizado aplicado a cada componente de la serie temporal.

Estos parámetros brindan al usuario la capacidad de adaptar la descomposición a las características específicas de los datos y a sus objetivos de análisis.

Al descomponer la serie temporal de consumo eléctrico, identificamos una tendencia creciente en el consumo a lo largo de los años, así como una estacionalidad marcada por picos de demanda durante ciertas horas del día o ciertos días de la semana. Esta información es esencial para modelar con precisión el comportamiento futuro del consumo eléctrico y tomar decisiones informadas sobre la gestión y planificación del suministro eléctrico.

7.1.3 Detección de outliers

La detección de outliers [12], es una técnica fundamental en el análisis de datos que se utiliza para identificar observaciones inusuales o atípicas en un conjunto de datos. Estas observaciones, conocidas como outliers, pueden representar puntos de datos que se desvían significativamente del comportamiento esperado o típico del resto de los datos. Es muy importante porque puede ayudar a identificar anomalías, errores en los datos o eventos inesperados que pueden afectar la interpretación y la validez de los resultados del análisis.

La Figura 7.1.3 representa mediante un diagrama de puntos los outliers de un conjunto de datos:

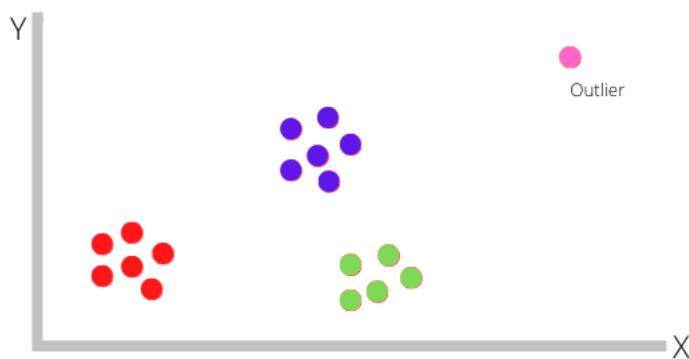


Figura 7.1.3: Ejemplo de Outliers.

El analista podrá modificar los siguientes parámetros como deseé:

- **Umbral de detección**, que permite al usuario especificar el nivel de sensibilidad para identificar outliers. Al ajustar este umbral, el usuario puede controlar la cantidad de puntos de datos identificados como outliers, lo que le permite adaptar la detección a sus requisitos de precisión y tolerancia a falsos positivos.
- Otro parámetro importante es el **método de detección**, que permite al usuario elegir entre diferentes enfoques para detectar outliers. Por ejemplo, el usuario puede optar por métodos basados en la desviación estándar o en el rango intercuartílico. Esta opción brinda al usuario flexibilidad para seleccionar el método que mejor se ajuste a la distribución de los datos y a las características específicas del conjunto de datos de consumo eléctrico.

Desviación estándar:

El método basado en la desviación estándar [13] identifica los outliers calculando la puntuación z (z-score) de cada punto de datos. La puntuación z mide cuántas desviaciones estándar se aleja un dato de la media de los datos. Si la puntuación z de un dato supera un umbral especificado, este dato se considera un outlier.

La fórmula para la puntuación z es:

$$z = \frac{X - \mu}{\sigma}$$

donde:

- X es el valor del dato.
- μ es la media de los datos.
- σ es la desviación estándar de los mismos.

Si $|z| > \text{umbral}$, el dato se considera un outlier.

Rango intercuartílico:

El método basado en el rango intercuartílico (IQR) [14] utiliza los percentiles para identificar outliers. Se calcula el primer cuartil (Q_1) y el tercer cuartil (Q_3) de los datos, y el rango intercuartílico se define como la diferencia entre Q_3 y Q_1 . Los outliers se identifican como aquellos datos que caen por debajo de $Q_1 - \text{umbral} * IQR$ o por encima de $Q_3 + \text{umbral} * IQR$.

Las fórmulas para detectar outliers por rango intercuartílico son:

$$\text{IQR} = Q_3 - Q_1$$

$$\text{Límite inferior} = Q_1 - \text{umbral} \times \text{IQR}$$

$$\text{Límite superior} = Q_3 + \text{umbral} \times \text{IQR}$$

Si un dato está fuera del rango definido por los límites superior e inferior, se considera un outlier.

- Los **estilos de visualización** es otro aspecto crucial ajustable. Permite al usuario elegir entre diferentes opciones para resaltar los outliers en los datos, en la gráfica se verán representados por puntos. Esta capacidad de personalización facilita la interpretación de los resultados y permite al usuario identificar de manera efectiva los puntos de datos anómalos en el conjunto de datos.
Además, de este modo de visualización simple se podrán representar los datos mediante un Histograma y un BoxPlot.

7.2 Diagrama de clases

En el diagrama de clases se mostrará cómo se relacionan las principales entidades de la aplicación y que van a ser guardadas en la base de datos. Las clases vienen definidas en los requisitos de información previamente mencionados.

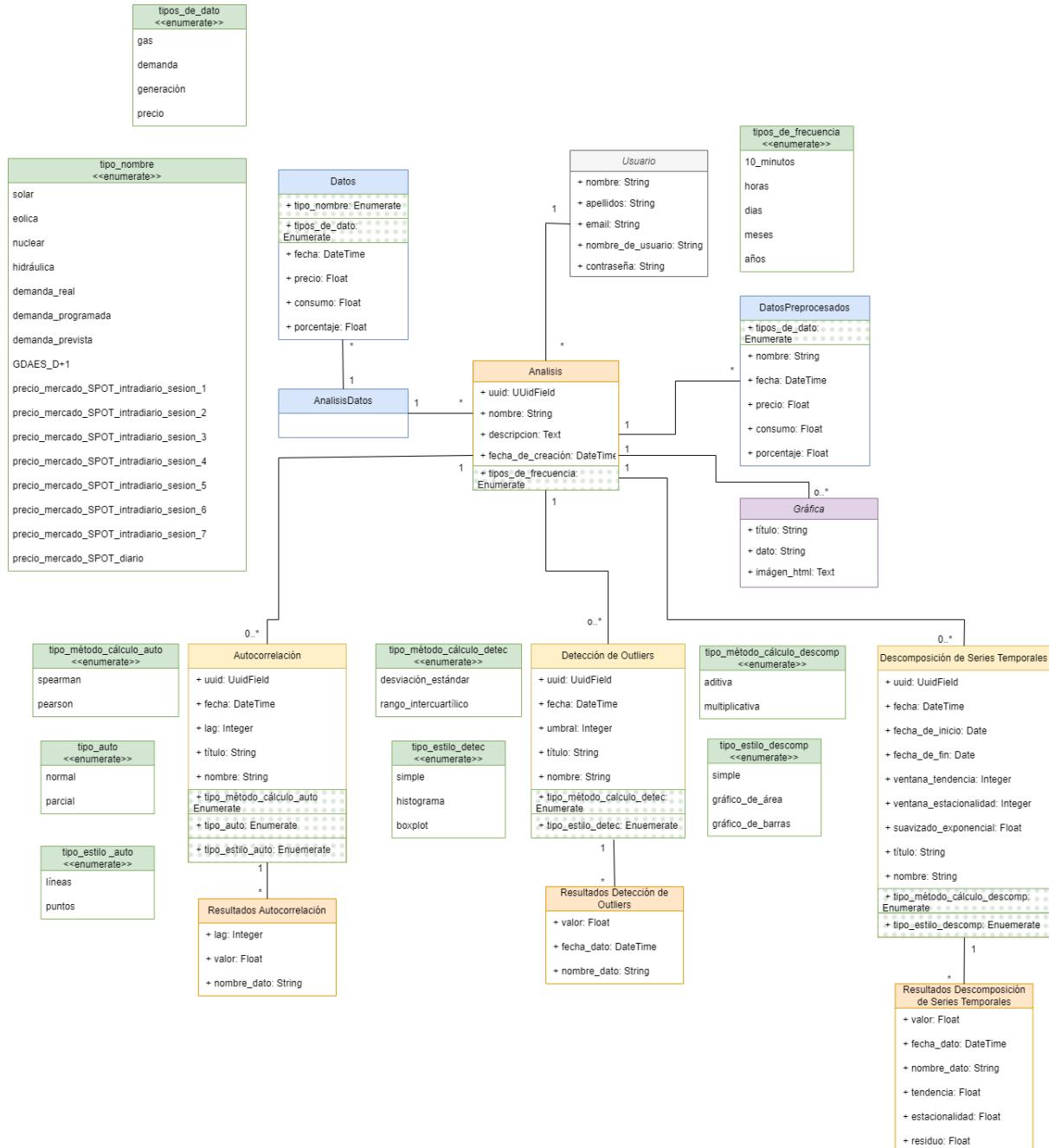


Figura 7.2: Diagrama UML.

La Figura 7.2 representa la estructura de modelos de nuestro producto. A continuación procederé a explicar en qué consiste cada uno de los elementos en detalle:

- **Usuario:** Se ha aprovechado el modelo de usuario proporcionado por Django puesto que otorga un sistema de cifrado de la contraseña predefinido en formato SHA256. Los atributos heredados son el “*nombre*”, “*apellidos*”, “*email*”, “*nombre de usuario*” y “*contraseña*”.
- **Datos:** Se trata del modelo en el que se almacenan los datos procedentes de ESIOS, MIBGAS y REE. Pese a que los datos originales poseen muchos más atributos, gracias a los scripts de descarga mediante el token de acceso hemos podido seleccionar los más interesantes de cara a la generación de análisis. Todos los datos recogidos son a nivel provincial, se han descartado los datos restantes para la realización de análisis más rigurosos y localizados:
 - “*tipo_nombre*”: Enumerado que guarda los nombres de todos los tipos de datos según su funcionalidad original. Estos son los posibles valores: *solar*, *eólica*, *nuclear*, *hidráulica*, *demandas_reales*, *demandas_programadas*, *demandas_previstas*, *GADES_D+1*, *precio_mercado_SPOT_intradiario_sesión_1*, *precio_mercado_SPOT_intradiario_sesión_2*, *precio_mercado_SPOT_intradiario_sesión_3*, *precio_mercado_SPOT_intradiario_sesión_4*, *precio_mercado_SPOT_intradiario_sesión_5*, *precio_mercado_SPOT_intradiario_sesión_6*, *precio_mercado_SPOT_intradiario_sesión_7* y *precio_mercado_SPOT_diario*.
 - “*tipo_de_dato*”: Enumerado que identifica el dato por su conjunto, entre los que actualmente se encuentran los datos de *gas*, *precio*, *generación* y *demandas*.
 - “*fecha*”: Simboliza el momento en el que se registró el dato en tiempo real.
 - “*precio*”: Son los valores del precio en euros para los tipos de datos que tengan este campo, no todos los datos tienen por qué tener precio.
 - “*consumo*”: Simbolizan los valores del consumo energético en megavatios por hora (Mw/H). Al igual que el precio, no todos tienen por qué contener este dato.
 - “*porcentaje*”: Representa el porcentaje en comparación con una referencia o máximo histórico según el tipo de dato. Para este proyecto sólo los datos de la demanda eléctrica contienen este atributo.
- **Análisis:** Es el modelo principal en torno al que gira todo el proyecto:
 - “*uivid*”: Identificador proporcionado para incrementar la seguridad en nuestra aplicación, lo utilizaremos para realizar las consultas a través de la Api, así como para ocultar el id del análisis en la url del navegador.
 - “*nombre*”: Identificador muy útil para diferenciar los análisis realizados por los distintos usuarios registrados en nuestra aplicación.
 - “*descripción*”: Es la descripción del análisis.

- “*frecuencia*”: Este atributo define el intervalo de tiempo entre cada punto de datos en el análisis. Es un elemento fundamental para el preprocesamiento de los datos, ya que determina la granularidad con la que se analizarán y procesarán los datos de la demanda eléctrica. Se puede seleccionar entre cinco frecuencias diferentes. Por ejemplo, si se selecciona una frecuencia de “10_minutos”, significa que se tomará un punto de datos cada 10 minutos.
 - “*fecha_de_creación*”: Fecha en la que se crea el análisis.
- **Análisis datos:** Modelo generado por la relación ManyToMany entre Análisis y Datos.
- **Datos preprocesados:** Es el modelo en el que se guardan los datos de nuestro modelo de Datos una vez preprocesados según la frecuencia que haya seleccionado el usuario para su análisis. Los atributos “*tipo_de_dato*” y “*nombre*” se mantienen constantes, no obstante hablaré de las diferencias fundamentales con respecto a los atributos de “*precio*”, “*consumo*”, “*porcentaje*” y las “*fechas*” en la sección de Preprocesado.
- **Autocorrelación:** Modelo del primer tipo de Análisis. Se guardará la “fecha de creación” del análisis de autocorrelación así como un identificador del tipo “*uuid*” para que los usuarios sin registrar puedan consultar los resultados de dicho análisis. Los atributos para crear un objeto de Autocorrelación son los siguientes:
 - “*lag*”: Número de períodos de tiempo que se retroceden en los datos para calcular la autocorrelación. Este valor determina la distancia temporal en la serie de datos entre los puntos que se están correlacionando. Por ejemplo, un lag de 1 significa que se está midiendo la correlación entre valores consecutivos, un lag de 2 mide la correlación entre valores con un intervalo de un periodo entre ellos, y así sucesivamente.
 - “*título*”: Título del objeto Autocorrelación, único para todos los objetos de Autocorrelación de un cierto análisis.
 - “*nombre*”: Simboliza los tipos de datos seleccionados en el formulario de análisis.
 - “*tipo_método_cálculo_auto*”: Enumerado por el cual podremos calcular el tipo de autocorrelación mediante diferentes métodos. Los elementos del enumerado son: *pearson* o *spearman*.
 - “*tipo_estilo_auto*”: Enumerado que permite visualizar los gráficos de Ploty con diferentes estilos. Los elementos del enumerado son: *líneas* o *puntos*.
 - “*tipo_auto*”: Enumerado que nos permite seleccionar entre el cálculo de autocorrelación parcial o simple. Los elementos del enumerado son: *normal* o *parcial*.
- **Resultados de autocorrelación:** Modelo para almacenar los resultados que el usuario desee de un objeto de autocorrelación. Cabe mencionar que se creará un objeto Resultados por cada “*lag*” del objeto Autocorrelación y cada “*nombre*” diferente de Datos Preprocesados. Se incluyen los siguientes atributos:
 - “*lag*”: Representa el número de lag.
 - “*valor*”: Es el valor resultado del cálculo de la autocorrelación según el lag.
 - “*nombre_dato*”: Es el nombre del dato para un valor y el lag asociado.

- **Detección de outliers:** Modelo del segundo tipo de análisis. Se guardará la “fecha_de_creación” del análisis de detección de outliers así como un identificador del tipo “uuid” para que los usuarios sin registrar puedan consultar los resultados de dicho análisis. Los atributos para crear un objeto de Detección de Outliers son los siguientes:
 - “*uuid*”: Identificador único proporcionado para incrementar la seguridad en nuestra aplicación. Se utilizará para realizar las consultas a través de la API y para ocultar el id del análisis en la URL del navegador.
 - “*fecha*”: Fecha en la que se crea el análisis de detección de outliers.
 - “*título*”: Título del objeto Detección de Outliers, único para todos los objetos de Detección de Outliers de un cierto análisis.
 - “*nombre*”: Simboliza los tipos de datos seleccionados en el formulario de análisis.
 - “*umbral*”: Valor límite utilizado para identificar outliers. Cualquier dato que supere este umbral se considerará un outlier.
 - “*tipo método cálculo detección*”: Enumerado por el cual podremos calcular la detección de outliers mediante diferentes métodos basados en machine learning. Los elementos del enumerado son: *desviación_estándar* o *rango_intercuartílico*.
 - “*tipo estilo detec*”: Enumerado que permite visualizar los gráficos de Plotly con diferentes estilos, facilitando la interpretación y análisis visual de los outliers detectados. Los elementos del enumerado son: *simple*, *histograma* o *boxplot*.
- **Resultados de detección de outliers:** Modelo para almacenar los resultados que el usuario desee de un objeto de Detección de Outliers. Cabe mencionar que se creará un objeto Resultados por cada dato detectado como outlier. Se incluyen los siguientes atributos:
 - “*valor*”: Es el valor del dato que ha sido identificado como outlier.
 - “*fecha_dato*”: Fecha y hora en la que se registró el dato identificado como outlier.
 - “*nombre_dato*”: Es el nombre del tipo de dato que ha sido identificado como outlier.
- **Descomposición de series temporales:** Modelo para realizar el análisis de descomposición de series temporales. Se guardará la “fecha de creación” del análisis de descomposición así como un identificador del tipo “uuid” para que los usuarios sin registrar puedan consultar los resultados de dicho análisis. Los atributos para crear un objeto de Descomposición de Series Temporales son los siguientes:
 - “*uuid*”: Identificador único proporcionado para incrementar la seguridad en nuestra aplicación. Se utilizará para realizar las consultas a través de la API y para ocultar el id del análisis en la URL del navegador
 - “*fecha*”: Fecha en la que se crea el análisis de descomposición de series temporales.
 - “*fecha_de_inicio*”: Fecha de inicio del periodo de tiempo que se analizará en la descomposición de la serie temporal.

- “*fecha_de_fin*”: Fecha de fin del periodo de tiempo que se analizará en la descomposición de la serie temporal.
 - “*ventana_tendencia*”: Tamaño de la ventana utilizada para calcular la tendencia de la serie temporal. Define cuántos datos se considerarán para suavizar la serie y extraer la tendencia.
 - “*ventana_estacionalidad*”: Tamaño de la ventana utilizada para calcular la estacionalidad de la serie temporal. Define el periodo de repetición para identificar patrones estacionales.
 - “*suavizado_exponencial*”: Parámetro utilizado para aplicar suavizado exponencial a los datos. Ayuda a reducir el ruido y resaltar los componentes subyacentes de la serie temporal.
 - “*título*”: Título del objeto Descomposición de Series Temporales, único para todos los objetos de Descomposición de Series Temporales de un cierto análisis.
 - “*nombre*”: Simboliza los tipos de datos seleccionados en el formulario de análisis.
 - “*tipo de estilo de descomposición*”: Enumerado que permite visualizar los gráficos de Plotly con diferentes estilos, facilitando la interpretación visual de la descomposición. Los elementos del enumerado son: *simple*, *gráfico_de_área* o *gráfico_de_barras*.
 - “*tipo de método de cálculo de descomposición*”: Enumerado que permite seleccionar entre diferentes métodos de descomposición de series temporales. Los elementos del enumerado son: *aditiva* o *multiplicativa*.
- **Resultados de descomposición de series temporales:** Modelo para almacenar los resultados que el usuario desee de un objeto de Descomposición de Series Temporales. Cabe mencionar que se creará un objeto Resultados por cada serie temporal descompuesta. Se incluyen los siguientes atributos:
 - “*valor*”: En este caso el valor es el dato como tal según el atributo “*nombre*” del modelo de Descomposición.
 - “*fecha_dato*”: Fecha y hora en la que se registró el dato de la serie temporal.
 - “*nombre dato*”: Es el nombre del tipo de dato que se ha descompuesto.
 - “*tendencia*”: Valor numérico que representa el componente de tendencia de la serie temporal descompuesta.
 - “*estacionalidad*”: Valor numérico que representa el componente de estacionalidad de la serie temporal descompuesta.
 - “*residuo*”: Valor numérico que representa el componente de residuo de la serie temporal descompuesta.

7.3 Arquitectura del sistema

El sistema sigue el patrón arquitectónico Model-View-Template (MVT) de Django [15]. Este patrón es fundamental para organizar y estructurar el desarrollo de aplicaciones web, separando claramente la lógica del negocio, la interfaz de usuario y la presentación. Esta separación facilita el mantenimiento y escalabilidad del sistema (Figura 7.3).

A continuación, se detalla cada una de las componentes de la arquitectura:

Modelo (Model)

El modelo en Django representa la estructura de la base de datos, definiendo las entidades y las relaciones entre ellas.

Vista (View)

La vista en Django gestiona la lógica de la aplicación, interactuando con los modelos y generando la respuesta que se envía al usuario.

Plantilla (Template)

La plantilla en Django se encarga de presentar los datos al usuario, separando la lógica de presentación del código Python.

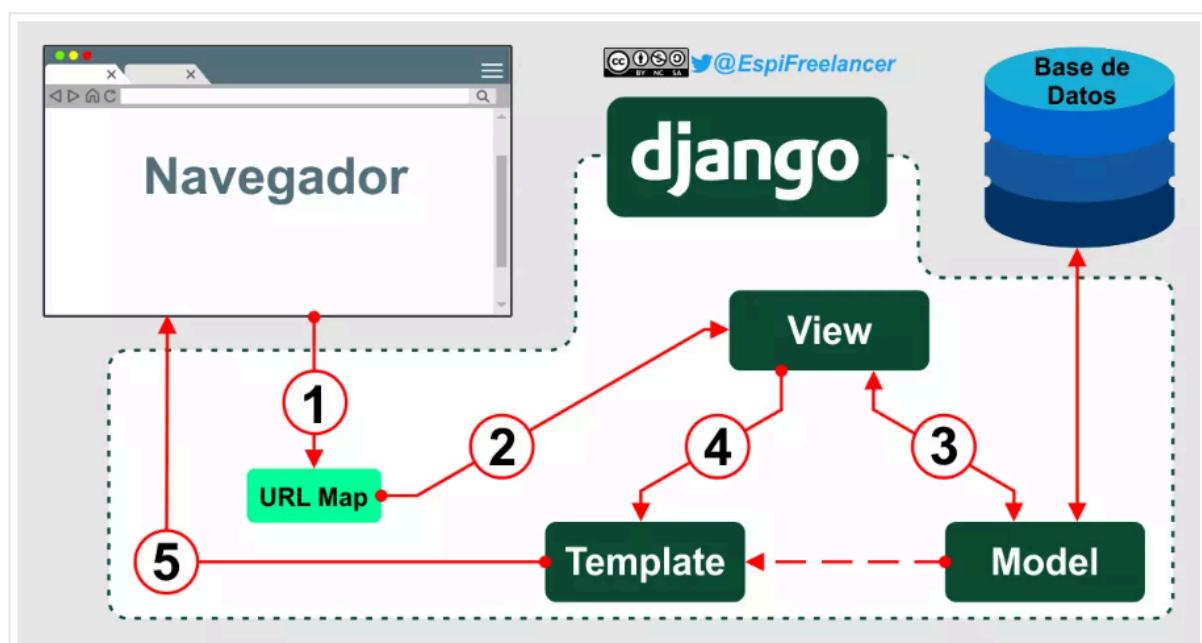


Figura 7.3: Diagrama de arquitectura

7.3.1 API

La API del sistema está implementada utilizando Django REST Framework (DRF) [16]. Este marco de trabajo es robusto y flexible, permitiendo la creación de APIs RESTful de manera eficiente. A través de esta API, los usuarios pueden acceder a los distintos tipos de análisis realizados y obtener los resultados de cada análisis de forma pública, siempre que dispongan de las credenciales adecuadas.

El diseño está pensado principalmente para que cualquier usuario que tenga el UUID del análisis, proporcionado previamente por un analista registrado, pueda consultar los resultados de dicho análisis. Las peticiones se realizan mediante el método GET y son facilitadas con herramientas como Swagger y ReDoc [17], de las cuales se hablará en detalle en la sección del diseño de la API.

7.3.2 Problema de procesamiento de datos y solución

Uno de los principales retos con Django es la velocidad en el procesamiento de datos. Para abordar este problema, se ha integrado Celery [18] y RabbitMQ [19] en el sistema junto con Django y la base de datos.

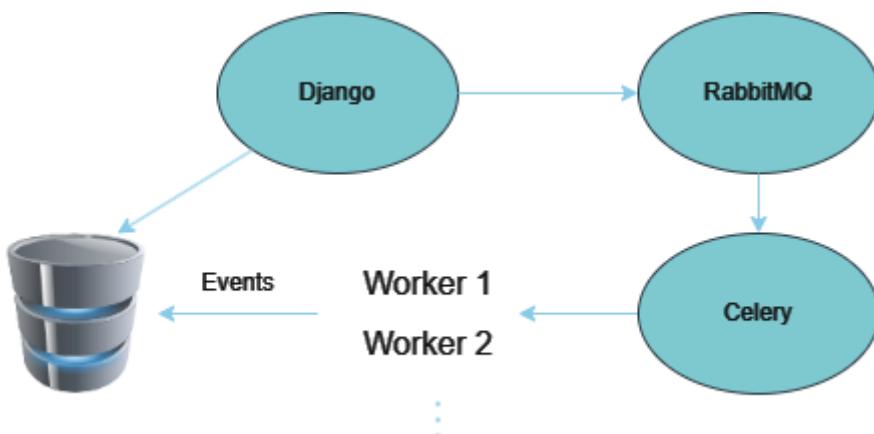


Figura 7.3.2.1: Diagrama de arquitectura Celery.

Celery es una herramienta de administración de colas de tareas distribuidas, que permite la ejecución de tareas de manera asíncrona y en paralelo (Figura 7.3.2.1). En este sistema, Celery se utiliza no solo para preprocesar los datos en cola de cada uno de los análisis realizados, sino también para ejecutar tareas periódicas. Esto se realiza mediante dos componentes principales:

1. **Worker de Celery:** Este componente se encarga de procesar los datos en cola en función de la frecuencia que el analista considere oportuna. Esto permite que los usuarios puedan ser redirigidos directamente a la vista de "mis_analisis" sin tener que esperar a que los datos sean preprocesados.

2. **Celery Beat:** Este componente se utiliza para mantener la base de datos actualizada en todo momento, ejecutando un script de actualización diario a las doce de la noche. Este script descarga los datos de manera local utilizando un token de ESIOS, asegurando que la información esté siempre al día.

RabbitMQ actúa como intermediario de mensajes escrito en el lenguaje de programación Erlang [20] para sistemas distribuidos y tolerantes a fallos, manejando la cola de tareas y asegurando que las tareas sean procesadas de manera eficiente y fiable. La integración de RabbitMQ con Celery garantiza que las tareas se distribuyan correctamente y se ejecuten sin sobrecargar el sistema.

Para que les sirva de guía, en la Figura 7.3.2.2 de abajo les mostraré el diagrama de flujo para realizar un análisis dentro de la aplicación haciendo uso de la herramienta de celery:

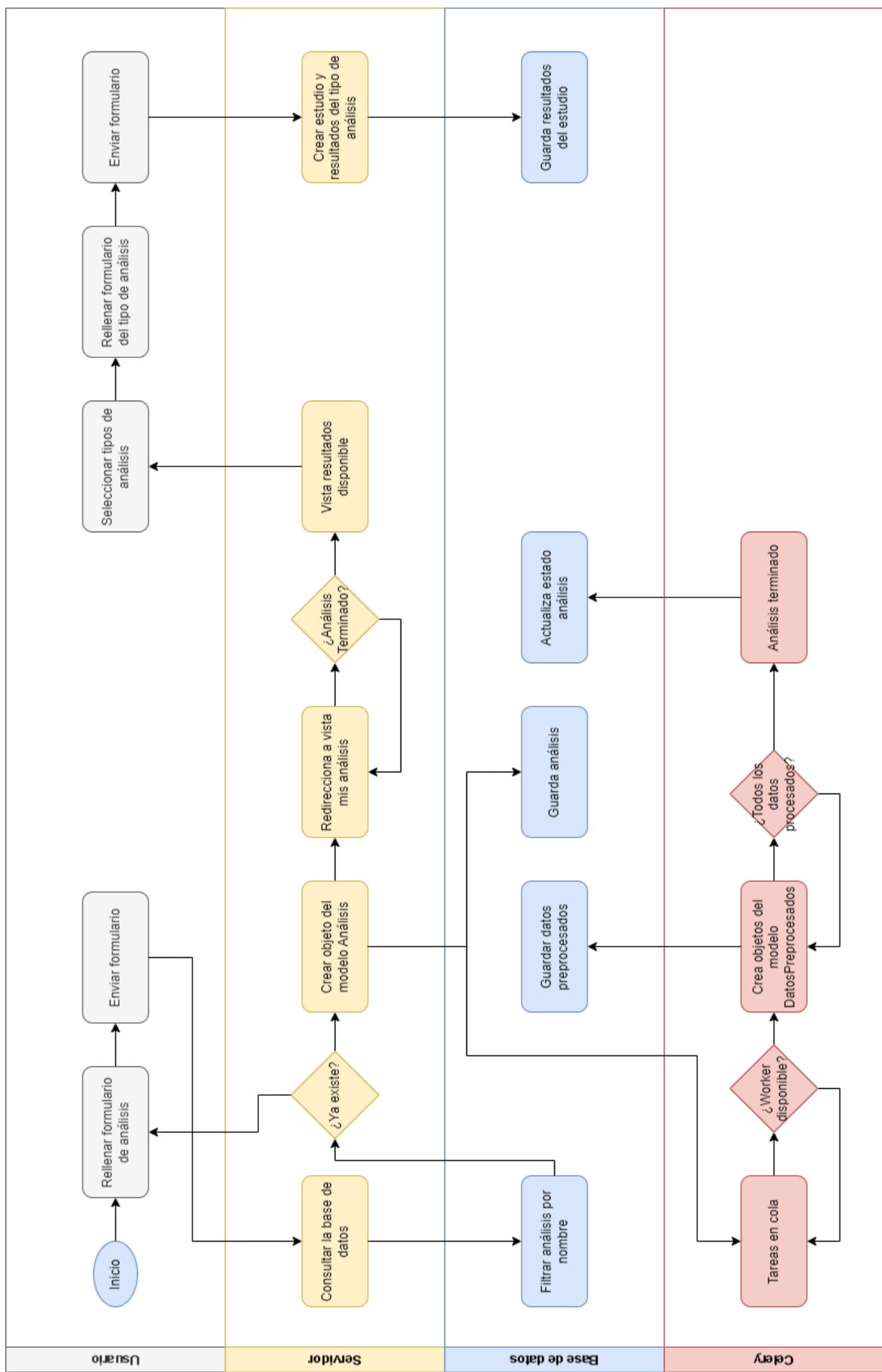


Figura 7.3.2.2: Diagrama de flujo.

¿Cómo se integra Celery con el MTV?

Interacción con el modelo:

Celery puede interactuar con los modelos de Django para ejecutar tareas asíncronas que impliquen operaciones con la base de datos. Por ejemplo, una tarea de Celery puede crear, leer, actualizar o eliminar registros en la base de datos. Un uso común es la ejecución de tareas periódicas para actualizar datos o realizar limpiezas.

Interacción con la vista:

Las vistas de Django pueden delegar tareas a Celery para que se ejecuten en segundo plano, mejorando el rendimiento y la capacidad de respuesta de la aplicación. Por ejemplo, una vista que procesa un formulario de usuario puede enviar una tarea a Celery para enviar un correo electrónico de confirmación sin bloquear la respuesta al usuario.

Interacción con la plantilla:

Aunque las plantillas no interactúan directamente con Celery, pueden beneficiarse de las tareas realizadas en segundo plano. Por ejemplo, una plantilla que muestra datos complejos o resultados de cálculos intensivos puede depender de tareas asíncronas que actualicen esos datos sin afectar la experiencia del usuario.

7.3.3 Herramientas de Documentación de la API

La documentación de la API es fundamental para la transparencia y facilidad de uso del sistema. En este caso, se utilizan Swagger y ReDoc [17] para generar y presentar la documentación de la API de manera interactiva y accesible. Estas herramientas permiten a los desarrolladores y usuarios finales explorar y entender las capacidades de la API, facilitando su uso y adopción.

1. **Swagger:** Ofrece una interfaz gráfica interactiva para probar y visualizar las endpoints de la API, proporcionando una descripción detallada de los métodos disponibles, parámetros y respuestas.
2. **ReDoc:** Proporciona una documentación estructurada y amigable para el usuario, que es fácil de navegar y comprender, ayudando a los desarrolladores a integrar y utilizar la API de manera más eficiente.

7.4 Tecnologías

Aquí hablaremos sobre todas las herramientas utilizadas durante el desempeño del proyecto así como de la documentación asociada al mismo.

7.4.1 Herramientas de documentación

Las aplicaciones de las que hablaremos a continuación han sido imprescindibles para la elaboración de una documentación correcta y clara.

- **Google drive:** Este servicio de google ha sido muy útil principalmente para elaborar la memoria del TFG, así como para anotar los enlaces de los cuáles he tomado

algunas referencias, para los problemas encontrados durante el desarrollo del producto o para anotar apuntes en el seguimiento tras las reuniones.

- **Word-2019:** Servicio corporativo para redactar ciertos fragmentos del documento.
- **Adobe Premiere Pro 2020:** Aplicación para editar el vídeo explicativo del proyecto.
- **Draw.io:** Es una herramienta en línea utilizada para realizar diagramas de todo tipo.
- **PowerPoint:** Herramienta para desarrollar la presentación.
- **GanttProject:** Herramienta de gestión de proyectos que se utiliza para definir y planificar el cronograma de un proyecto, así como para realizar un seguimiento del progreso durante su desarrollo.

7.4.2 Herramientas de desarrollo

- **Visual Studio Code:** Es un editor de código fuente desarrollado por Microsoft, lo he utilizado para el desarrollo del producto debido a su versatilidad y amplia gama de extensiones que facilitan el proceso de codificación.
- **GitHub:** Es una plataforma de alojamiento y colaboración para proyectos de desarrollo de software. Lo he utilizado para subir el repositorio de mi proyecto.
- **Python 3.12.0:** Es la última versión del popular lenguaje de programación Python, que ofrece nuevas características y mejoras en rendimiento y seguridad. Lo he utilizado para desarrollar el núcleo del proyecto debido a su eficiencia y amplia comunidad de soporte.
- **Django [21]:** Es un framework web de alto nivel y de código abierto escrito en Python que permite el desarrollo rápido y seguro de aplicaciones web. He utilizado Django como el framework principal para la construcción del backend del proyecto.
- **JavaScript:** Es un lenguaje de programación ampliamente utilizado en el desarrollo web para crear interactividad en las páginas. Se utiliza para agregar funcionalidades dinámicas y mejorar la experiencia del usuario.
- **Django-rest-framework [16]:** Es una poderosa biblioteca que facilita la creación de API web utilizando Django. Sirve para desarrollar y exponer APIs RESTful en la aplicación.
- **Crispy-forms [22]:** Biblioteca que facilita la creación de formularios HTML elegantes y concisos en Django. Se ha utilizado para mejorar el diseño y la usabilidad de los formularios.
- **Crispy-bootstrap 5 [23]:** Es una extensión de crispy-forms que proporciona integración con Bootstrap 5, un popular framework de diseño frontend. Se utiliza para aplicar estilos de Bootstrap a los formularios.
- **Django-celery-results [18]:** Es una biblioteca que permite almacenar los resultados de las tareas Celery en la base de datos de Django. He utilizado django-celery-results para gestionar y almacenar los resultados de las tareas asíncronas en mi aplicación.
- **Django-celery-beat [18]:** Es una extensión de Celery que proporciona un programador de tareas basado en Django. Se ha utilizado para programar tareas periódicas y asíncronas.
- **Drf-yasg [17]:** Es una biblioteca que genera automáticamente documentación Swagger para las API creadas con Django-rest-framework. Genera documentación clara y detallada de las API.

- **Postgresql** [26]: Es un sistema de gestión de bases de datos relacional y de código abierto. Se ha utilizado para almacenar y gestionar los datos de la aplicación debido a su robustez, escalabilidad y soporte para operaciones complejas.
- **Plotly** [25]: Es una biblioteca de gráficos interactivos y análisis de datos en Python. Ha sido fundamental para la creación de gráficos interactivos y visualizaciones avanzadas en la aplicación web.
- **Matplotlib** [26]: Biblioteca de Python para la creación de gráficos estáticos, animados e interactivos.
- **Numpy** [27]: Es una biblioteca de Python que proporciona soporte para arrays y matrices de gran tamaño, junto con una colección de funciones matemáticas de alto nivel para operar con estos arrays.
- **Pandas** [28]: Es una biblioteca de Python especializada en la manipulación y análisis de datos.
- **Request** [29]: Es una biblioteca de Python que facilita el envío de solicitudes HTTP de manera simple. Ha sido muy útil para interactuar con APIs externas y gestionar las solicitudes HTTP en la aplicación.

7.5 Diseño API

Esta sección detalla el diseño de la API, la cual está concebida como una interfaz de acceso público. La API ha sido desarrollada exclusivamente para el método "GET", lo que permite a los usuarios no logueados obtener información sin necesidad de autenticación. Su objetivo principal es facilitar a cualquier usuario pueda realizar consultas siempre y cuando se hayan puesto en contacto con alguno de los analistas, para ello previamente pueden acceder a la vista de 'Analistas' de la aplicación tal y como se muestra en la Figura 7.5, en la que aparece la información de contacto para solicitar el UUID al analista que considere oportuno. Este si lo desea podrá proporcionar el uuid, el nombre y la frecuencia de algún análisis para que el solicitante pueda ver los datos.

Nombre del análisis	Descripción	Nombre del analista	Apellidos del analista	Email de contacto
prueba	Esto es una descripción de prueba	carlos	nuchera	carmucbol@alum.us.es

Figura 7.5: Vista de Analistas.

A continuación mostraremos todas las consultas que se pueden hacer mediante nuestra api desde swagger (URL `/api/swagger/`). A continuación hablaremos de cada una de las consultas indicando la descripción de las mismas así como los parámetros necesarios para realizarlas:

- ***GET /analisis/***
 - Descripción: Con esta consulta podremos obtener los objetos del modelo de Análisis de nuestra base de datos. Si un usuario ajeno al creador de dicho análisis desea tener acceso a los datos debe consultar previamente
 - Parámetros:
 - *uuid*: Identificador único por cada análisis.
 - *nombre*: Título del análisis.
 - *frecuencia*: Frecuencia seleccionada por el analista.
- ***GET /analisis/autocorrelacion/***
 - Descripción: Podremos consultar los objetos de Autocorrelación asociados a un análisis en concreto.
 - Parámetros:
 - *uuid*: Identificador único por cada análisis.
- ***GET /analisis/autocorrelacion/resultados_autocorrelacion/***
 - Descripción: Con esta consulta accedemos a los resultados para una determinada configuración de Autocorrelación.
 - Parámetros:
 - *uuid*: Identificador único para un objeto de Autocorrelación.
- ***GET /analisis/deteccion_de_outliers/***
 - Descripción: Podremos consultar los objetos de Detección de Outliers de un determinado análisis.

- Parámetros:
 - *uuid*: Identificador único por cada análisis.
- *GET /analisis/deteccion_de_outliers/resultados_deteccion_de_outliers/*
 - Descripción: Podemos consultar los resultados para un determinado objeto de Detección de Outliers.
 - Parámetros:
 - *uuid*: Identificador único para un objeto de Detección de Outliers.
- *GET /analisis/descomposicion_de_series_temporales/*
 - Descripción: Con esta consulta accederemos a todos los objetos de descomposición asociados a un determinado análisis.
 - Parámetros:
 - *uuid*: Identificador único por cada análisis.
- *GET /analisis/descomposicion_de_series_temporales/resutlados_descomposicion_de_series_temporales/*
 - Descripción: Se utiliza para acceder a todos los objetos ‘resultados’ de una determinada configuración de Descomposición de serie temporal.
 - Parámetros:
 - *uuid*: Identificador único para un objeto de Descomposición de Serie Temporal.

Cuarta parte

IV-PLANIFICACIÓN

8 Planificación

La planificación es una etapa fundamental en cualquier proyecto de software, ya que define el camino a seguir para alcanzar los objetivos establecidos. En esta sección, describiremos el enfoque metodológico adoptado, las fases del desarrollo, la gestión del código y los hitos clave que marcan el progreso del proyecto. La estructura de descomposición del trabajo también será detallada para proporcionar una visión clara de la jerarquía y las tareas involucradas.

8.1 Listado de hitos

Para establecer de manera precisa los plazos de las tareas, se han identificado y definido una serie de hitos (Cuadro 8.1) que corresponden a las fases clave del proyecto. Estos hitos actúan como puntos de referencia significativos a lo largo del ciclo de vida del proyecto, marcando el inicio o la finalización de etapas importantes.

La identificación y definición de estos hitos permiten una planificación más efectiva y un seguimiento más preciso del progreso del proyecto, ya que brindan una visión clara de los momentos críticos en los que se deben alcanzar objetivos específicos. Además, los hitos facilitan la comunicación y la coordinación entre los miembros del equipo, o en este caso el desarrollador y las partes interesadas, proporcionando puntos de control claros para evaluar el avance del proyecto y asegurarse de que se cumplan los plazos establecidos.

Cuadro 8.1: Hitos.

Inicio del proyecto	31/01/2024
Planificación	08/02/2024
Fase 1	02/04/2024
Fase 2	07/05/2024
Fase 3	30/05/2024
Fase 4	7/06/2024
Cierre	24/06/2024

8.2 Estructura de descomposición del trabajo

La Estructura de Descomposición del Trabajo (EDT) es una técnica fundamental en la gestión de proyectos, facilitando la comprensión y organización de las distintas partes que componen al producto completo. A continuación, explicaré la estructura de trabajo de manera visual para comprender de una manera más sencilla la jerarquía de paquetes de nuestro proyecto:

- Proyecto: Es el nivel más alto de la jerarquía, y abarca todo el proyecto.
 - Color: GRIS

- Fases del proyecto: Representan los elementos de nivel superior acordes a cada una de las fases del desempeño del proyecto, así como distintos componentes fundamentales o responsabilidades importantes.
 - Color: MORADO
- Paquetes de seguimiento: Los paquetes de seguimiento representan la planificación de las tareas, están por encima de los paquetes de tareas, deberían estar al mismo nivel, pero es más oportuno diferenciarlos como paquetes de mayor importancia de cara al desglose de las Fases como veremos a continuación.
 - Color: NARANJA
- Paquetes de tareas: Se trata del nivel más bajo, engloba a todas las tareas realizadas a lo largo del proyecto. Comprenden desde análisis de requisitos, la configuración del entorno y las pruebas realizadas hasta la documentación asociada.
 - Color: AMARILLO

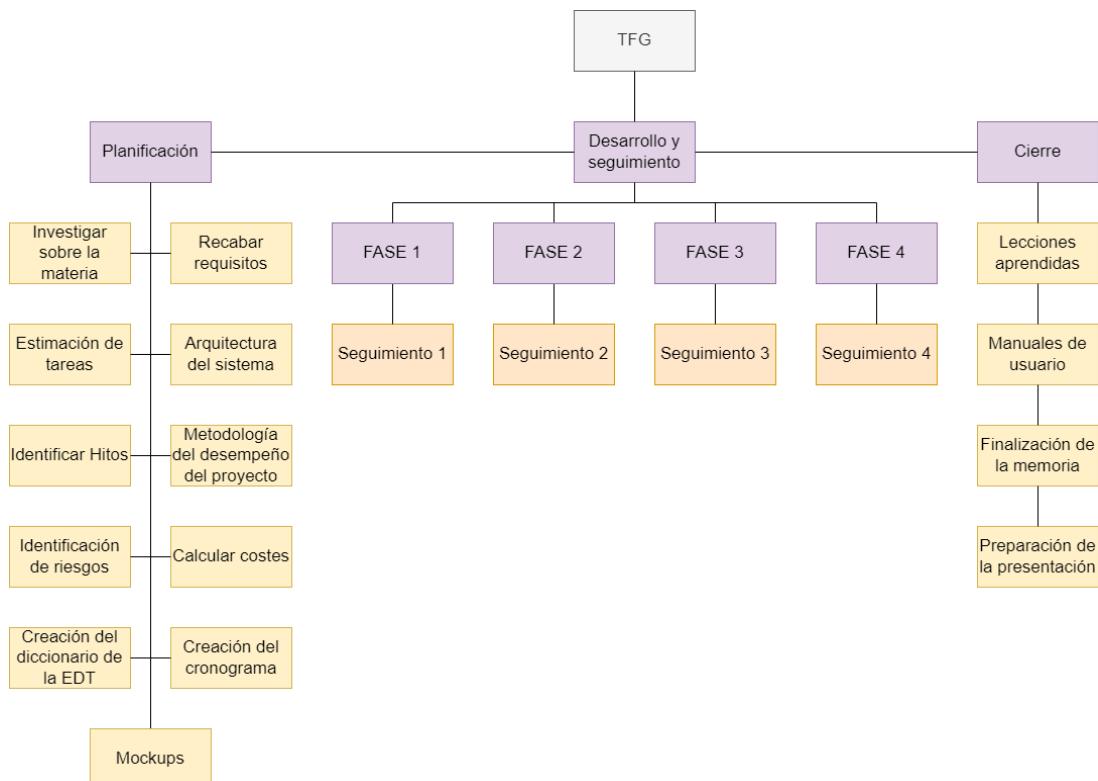


Figura 8.2: Diagrama EDT general.

8.3 Diccionario de la EDT

En esta sección se detallan los elementos de la EDT de la Figura 8.2, además, mostraremos en detalle los diferentes bloques de tareas durante el desarrollo y seguimiento del producto representados en las Figuras 8.3.1, 8.3.2, 8.3.3 y 8.3.4.

Para cada elemento del EDT proporcionaremos la siguiente información:

- Identificador: Código único que se asigna a cada elemento de la EDT para su fácil referencia y seguimiento.
- Título: Nombre del elemento de la EDT que describe de manera breve y clara la tarea o fase específica.
- Descripción: Una breve explicación que proporciona más detalles sobre el trabajo que implica el elemento de la EDT. Esto ayudará a entender el propósito y alcance del elemento.
- Fecha de inicio: Fecha en la que se espera comenzar el trabajo del elemento específico.
- Fecha de fin: Fecha de finalización de dicho elemento.
- Estimación en horas de trabajo: Cantidad de horas que se estima que tomará completar el elemento de la EDT.
- Responsable: Encargado de la realización de la tarea, como sólo hay una persona implicada, se representará la responsabilidad acorde a la tarea realizada.
- Coste: Estimación del coste asociado a la tarea, incluyendo recursos humanos, materiales y otros gastos.

Podrán ver más en detalle cada tarea de la planificación en el Cuadro 8.3, para cada una de las fases de desarrollo en los Cuadros 8.3.1, 8.3.2, 8.3.3 y 8.3.4. Por otro lado, el cierre, lo podremos ver en el Cuadro 8.3.5.

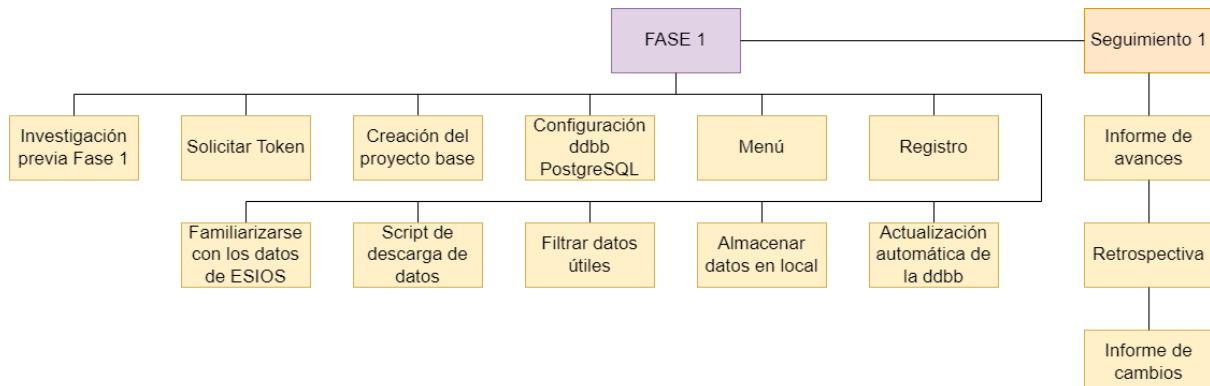


Figura 8.3.1: Diagrama EDT Fase 1.

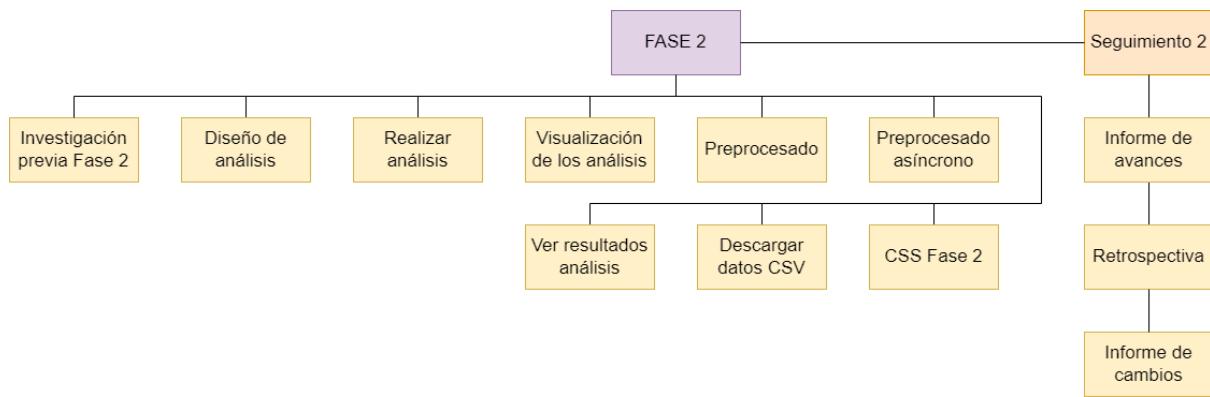


Figura 8.3.2: Diagrama EDT Fase 2.

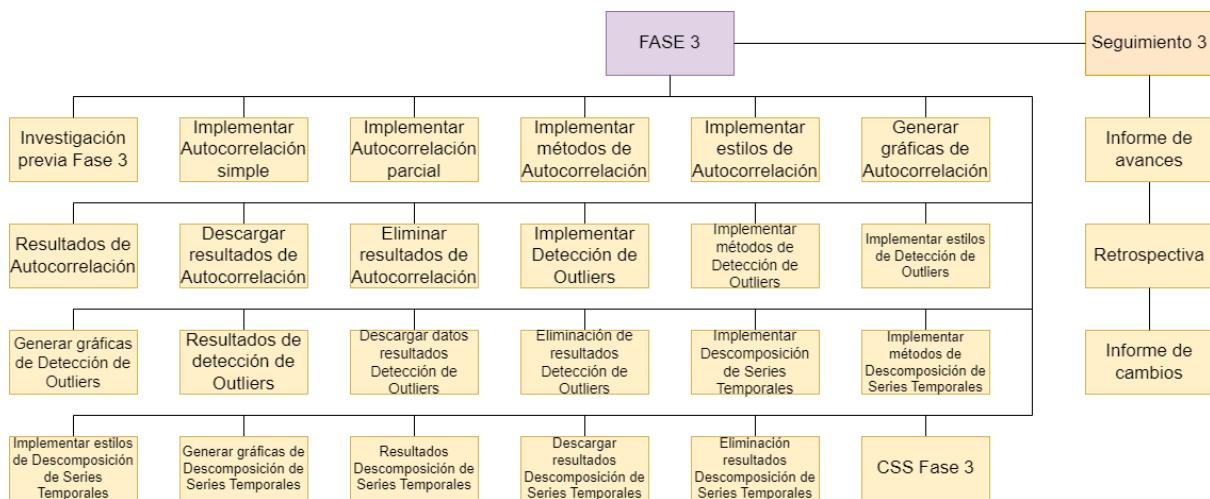


Figura 8.3.3: Diagrama EDT Fase 3.

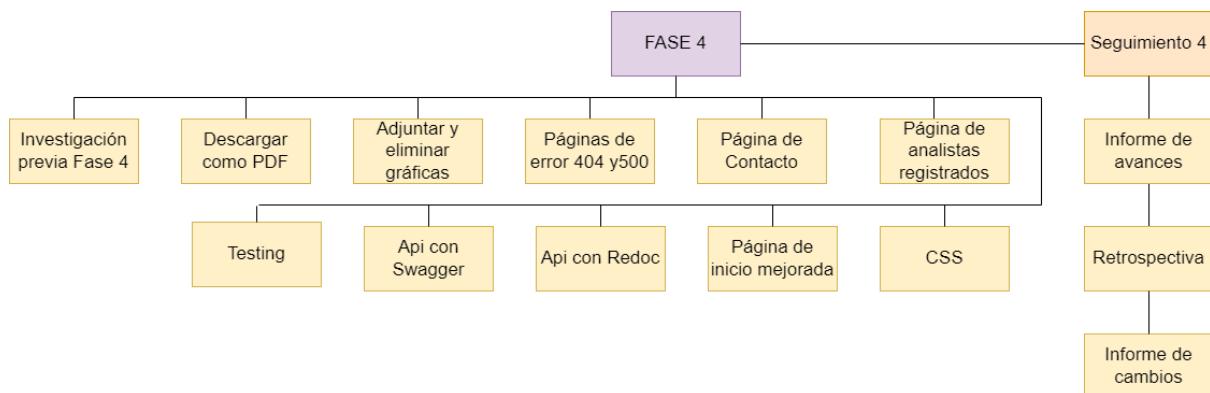


Figura 8.3.4: Diagrama EDT Fase 4.

Cuadro 8.3 Diccionario EDT-Planificación.

EDT	Título	Descripción	Fecha de inicio	Fecha de fin	Horas estimadas	Horas reales	Responsable	Coste (euros)
1.1	Planificación	Actividades necesarias para organizar y planificar el proyecto.	31/01/2024	14/02/2024	40	40	—	734.07
1.1.1	Investigación previa	Recopilación de información relevante para comprender mejor el contexto.	31/01/2024	31/01/2024	4	4	Analista	57.56
1.1.2	Definición de requisitos	Identificación y documentación de los requisitos del proyecto.	01/02/2024	02/02/2024	9	10	Analista	129.51
1.1.3	Estimación de tareas	Estimación del tiempo y recursos necesarios para cada tarea.	05/02/2024	05/02/2024	2	1	Jefe de Proyectos	42.56
1.1.4	Arquitectura del sistema	Diseño de la arquitectura técnica del sistema.	06/02/2024	06/02/2024	2	2	Analista	28.78
1.1.5	Identificación de riesgos	Evaluación y documentación de posibles riesgos y su mitigación.	07/02/2024	07/02/2024	2	1	Analista	28.78
1.1.6	Metodología durante el desempeño	Definición de la metodología a seguir durante el proyecto.	08/02/2024	08/02/2024	2	1	Jefe de Proyectos	42.56
1.1.7	Creación del cronograma	Desarrollo del cronograma del proyecto.	09/02/2024	09/02/2024	4	5	Jefe de Proyectos	85.12
1.1.8	Creación del diccionario de la EDT	Desarrollo y documentación del diccionario de la EDT.	12/02/2024	12/02/2024	5	8	Jefe de Proyectos	106.40
1.1.9	Calcular costes	Cálculo y documentación de los costes del proyecto.	13/02/2024	13/02/2024	4	2	Jefe de Proyectos	85.12

<i>EDT</i>	<i>Título</i>	<i>Descripción</i>	<i>Fecha de inicio</i>	<i>Fecha de Fin</i>	<i>Horas estimadas</i>	<i>Horas reales</i>	<i>Responsable</i>	<i>Coste (euros)</i>
1.1.10	<i>Identificar hitos</i>	<i>Definición y documentación de los hitos clave del proyecto.</i>	13/02/2024	13/02/2024	1	1	<i>Jefe de Proyectos</i>	21.28
1.1.11	<i>Mockups</i>	<i>Creación de mockups o prototipos visuales de la aplicación.</i>	14/02/2024	14/02/2024	5	5	<i>Jefe de Proyectos</i>	106.40

Cuadro 8.3.1 Diccionario EDT-Fase 1

EDT	Título	Descripción	Fecha de inicio	Fecha de Fin	Horas estimadas	Horas reales	Responsable	Coste (euros)
1.2	FASE 1	Desarrollo de la infraestructura básica y scripts de datos iniciales.	15/02/2024	15/03/2024	57	68	—	986.72
1.2.1	Investigación previa Fase 1	Investigación previa para administrar correctamente las tareas a desarrollar en esta fase, así como la correcta comprensión del trabajo a realizar para la Fase 1.	15/02/2024	15/02/2024	4	4	Analista	57.56
1.2.2	Solicitar token	Solicitud del token de acceso para API de datos ES/OS.	16/02/2024	16/02/2024	1	1	Jefe de Proyectos	21.28
1.2.3	Creación proyecto base	Configuración inicial del proyecto en el entorno de desarrollo en Django.	16/02/2024	16/02/2024	2	1	Programador	34.88
1.2.4	Configuración ddbb PostgreSQL	Instalación y configuración de la base de datos PostgreSQL.	19/02/2024	19/02/2024	2	2	Programador	34.88
1.2.5	Menú	Desarrollo del menú de navegación de la aplicación.	19/02/2024	19/02/2024	1	1	Programador	17.44
1.2.6	Registro	Implementación del sistema de registro de usuarios.	19/02/2024	19/02/2024	1	1	Programador	17.44
1.2.7	Familiarizarse con datos ES/OS	Estudio y comprensión de los datos proporcionados por ES/OS.	20/02/2024	20/02/2024	3	6	Programador	52.32
1.2.8	Script de descarga de datos	Desarrollo del script para la descarga de datos desde ES/OS, REE y MIBGAS.	21/02/2024	26/02/2024	13	19	Programador	226.72

EDT	Título	Descripción	Fecha de inicio	Fecha de fin	Horas estimadas	Horas reales	Responsable	Coste (euros)
1.2.9	Filtrar datos útiles	Estudiar los datos de entrada y seleccionar únicamente los datos útiles para el estudio.	27/02/2024	04/03/2024	10	7	Programador	174.40
1.2.10	Almacenar datos en local	Diseño de almacenamiento de datos en local.	05/03/2024	05/03/2024	4	10	Programador	69.76
1.2.11	Actualización automática de la dbb	Manejo de Celery para mantener la base de datos actualizada constantemente.	06/03/2024	14/03/2024	10	17	Programador	174.40
1.2.12	Seguimiento Fase 1	Son los documentos del seguimiento del proyecto que incluyen el informe de avances, la retrospectiva y el informe de cambios para la Fase 1.	15/03/2024	15/03/2024	6	6	Programador	104.64

Cuadro 8.3.2 Diccionario EDT-Fase 2

EDT	Título	Descripción	Fecha de inicio	Fecha de fin	Horas estimadas	Horas reales	Responsable	Coste (euros)
1.3	FASE 2	Fase enfocada en el análisis de los datos.	18/03/2024	08/04/2024	53	66	—	912.12
1.3.1	Investigación previa Fase 2	Investigación previa para administrar correctamente las tareas a desarrollar en esta fase, así como la correcta comprensión del trabajo a realizar para la Fase 2.	18/03/2024	18/03/2024	4	4	Analista	57.56
1.3.2	Diseño de análisis	Diseño e implementación del modelo de análisis de datos.	19/03/2024	19/03/2024	2	3	Programador	34.88
1.3.3	Realizar análisis	Realización del análisis sin procesar de datos según el modelo diseñado.	20/03/2024	25/03/2024	14	5	Programador	244.16
1.3.4	Visualización de los análisis	Se muestran todos los análisis realizados por un usuario.	26/03/2024	26/03/2024	2	3	Programador	34.88
1.3.5	Preprocesado	Preprocesamiento de los datos según la frecuencia seleccionada por el usuario en el formulario de análisis.	27/03/2024	03/04/2024	15	19	Programador	261.60
1.3.6	Preprocesado asíncrono	Redirección automática al listado de análisis; el preprocesamiento se realiza en segundo plano.	04/04/2024	04/04/2024	4	12	Programador	69.76
1.3.7	Ver resultados de análisis	Visualización previa del correcto guardado de los análisis para los datos preprocesados.	04/04/2024	04/04/2024	1	1	Programador	17.44

<i>EDT</i>	<i>Título</i>	<i>Descripción</i>	<i>Fecha de inicio</i>	<i>Fecha de Fin</i>	<i>Horas estimadas</i>	<i>Horas reales</i>	<i>Responsable</i>	<i>Coste (euros)</i>
1.3.8	Descargar datos CSV	<i>Implementación de la funcionalidad para descargar los datos del estudio en formato CSV.</i>	04/04/2024	04/04/2024	1	2	Programador	17.44
1.3.9	CSS Fase 2	<i>Aplicar estilos para que la página se vea más comercial durante la Fase 2.</i>	05/04/2024	08/04/2024	4	4	Programador	69.76
1.3.10	Seguimiento Fase 2	<i>Son los documentos del seguimiento del proyecto que incluyen el informe de avances, la retrospectiva y el informe de cambios para la Fase 2.</i>	09/04/2024	09/04/2024	6	6	Programador	104.64

Cuadro 8.3.3 Diccionario EDT-Fase 3

EDT	Título	Descripción	Fecha de inicio	Fecha de fin	Horas estimadas	Horas reales	Responsable	Coste (euros)
1.4	FASE 3	Desarrollo de funcionalidades avanzadas para el análisis de datos, centrándose en el desarrollo de los tipos de análisis disponibles.	10/04/2024	14/05/2024	94	102	—	1671.98
1.4.1	Investigación previa Fase 3	Investigación previa para administrar correctamente las tareas a desarrollar en esta fase, así como la correcta comprensión del trabajo a realizar para la Fase 3.	10/04/2024	12/04/2024	9	9	Analista	129.51
1.4.2	Implementar Autocorrelación simple	Diseño e implementación del modelo de Autocorrelación.	15/04/2024	17/04/2024	7	15	Programador	122.08
1.4.3	Implementar Autocorrelación parcial	Diseño e implementación del modelo de Autocorrelación parcial.	18/04/2024	18/04/2024	2	3	Programador	34.88
1.4.4	Implementar métodos Autocorrelación	Implementación de múltiples métodos de cálculo para la Autocorrelación.	18/04/2024	18/04/2024	3	3	Programador	52.32
1.4.5	Implementar estilos Autocorrelación	Implementación de múltiples estilos para la visualización de Autocorrelación.	19/04/2024	19/04/2024	2	6	Programador	34.88
1.4.6	Generar gráficas Autocorrelación	Creación de gráficos dinámicos para la autocorrelación con Plotly.	22/04/2024	22/04/2024	4	7	Programador	69.76

EDT	Título	Descripción	Fecha de inicio	Fecha de Fin	Horas estimadas	Horas reales	Responsable	Coste (euros)
1.4.7	Resultados de Autocorrelación	Visualización y guardado de resultados para un objeto de Autocorrelación.	23/04/2024	23/04/2024	3	5	Programador	52.32
1.4.8	Descargar resultados de Autocorrelación	Descarga de los resultados de la Autocorrelación en formato CSV.	24/04/2024	24/04/2024	1	1	Programador	17.44
1.4.9	Eliminar resultados de Autocorrelación	Se deben eliminar los resultados deseados de Autocorrelación.	25/04/2024	25/04/2024	2	3	Programador	34.88
1.4.10	Implementar Detección de Outliers	Diseño e implementación del modelo de Detección de Outliers.	26/04/2024	29/04/2024	8	5	Programador	139.52
1.4.11	Implementar métodos Detección de Outliers	Implementación de múltiples métodos de cálculo para la Detección de Outliers.	30/04/2024	30/04/2024	4	2	Programador	69.76
1.4.12	Implementar estilos Detección de Outliers	Implementación de múltiples estilos para la visualización de Detección de Outliers.	01/05/2024	01/05/2024	3	3	Programador	52.32
1.4.13	Generar gráfica de Detección de Outliers	Creación de gráficos dinámicos para la Detección de Outliers con Plotly.	02/05/2024	02/05/2024	4	2	Programador	69.76
1.4.14	Resultados de Detección de Outliers	Visualización y guardado de resultados para un objeto de Detección de Outliers.	03/05/2024	03/05/2024	3	2	Programador	52.32

EDT	Título	Descripción	Fecha de inicio	Fecha de Fin	Horas estimadas	Horas reales	Responsable	Coste (euros)
1.4.15	Descargar datos resultados de Detección de Outliers	Descarga de los resultados de la Detección de Outliers en formato CSV.	03/05/2024	03/05/2024	1	1	Programador	17.44
1.4.16	Eliminación de resultados de Detección de Outliers	Se deben eliminar los resultados deseados para la Detección de Outliers.	03/05/2024	03/05/2024	2	1	Programador	34.88
1.4.17	Implementar Descomposición de Series Temporales	Diseño e implementación del modelo de Descomposición de Series Temporales	06/05/2024	07/05/2024	9	9	Programador	156.96
1.4.18	Implementar métodos Descomposición de Series Temporales	Implementación de múltiples métodos de cálculo para la Descomposición de Series Temporales.	08/05/2024	08/05/2024	4	6	Programador	69.76
1.4.19	Implementar estilos Descomposición de Series Temporales	Implementación de múltiples estilos para la visualización de Descomposición de Series Temporales.	08/05/2024	08/05/2024	3	3	Programador	52.32
1.4.20	Generar gráficas de Descomposición de Series Temporales	Creación de gráficos dinámicos para la Descomposición de Series Temporales con Plotly.	09/05/2024	09/05/2024	4	2	Programador	69.76

EDT	Título	Descripción	Fecha de inicio	Fecha de Fin	Horas estimadas	Horas reales	Responsable	Coste (euros)
1.4.21	Resultados de Descomposición de Series Temporales	Visualización y guardado de resultados para un objeto de Descomposición de Series Temporales.	10/05/2024	10/05/2024	3	2	Programador	52.32
1.4.22	Descargar datos resultados de Descomposición de Series Temporales	Descarga de los resultados de la Descomposición de Series Temporales en formato CSV.	10/05/2024	10/05/2024	1	1	Programador	17.44
1.4.23	Eliminación de resultados de Descomposición de Series Temporales	Se deben eliminar los resultados deseados para la Descomposición de Series Temporales.	10/05/2024	10/05/2024	2	1	Programador	34.88
1.4.24	CSS Fase 3	Aplicar estilos para que la página se vea más comercial durante la Fase 3.	13/05/2024	13/05/2024	4	4	Programador	69.76
1.4.25	Seguimiento Fase 3	Son los documentos del seguimiento del proyecto que incluyen el informe de avances, la retrospectiva y el informe de cambios para la Fase 3.	14/05/2024	14/05/2024	6	6	Programador	104.64

Cuadro 8.3.4 Diccionario EDT-Fase 4

EDT	Título	Descripción	Fecha de inicio	Fecha de Fin	Horas estimadas	Horas reales	Responsable	Coste (euros)
1.5	FASE 4	Desarrollo de funcionalidades adicionales y mejoras en la aplicación.	15/05/2024	30/05/2024	42	42	Programador	720.28
1.5.1	Investigación previa Fase 4	<i>Investigación previa para administrar correctamente las tareas a desarrollar en esta fase, así como la correcta comprensión del trabajo a realizar para la Fase 4.</i>	15/05/2024	15/05/2024	4	3	Analista	57.56
1.5.2	Descargar PDF	Desarrollo de la funcionalidad para descargar todos los resultados de un análisis en PDF.	16/05/2024	17/05/2024	7	7	Programador	122.08
1.5.3	Adjuntar y eliminar gráficas en vistas	Método para adjuntar una gráfica a la vista de resultados correspondiente.	20/05/2024	21/05/2024	7	10	Programador	122.08
1.5.4	Páginas de error 404 y 500s	Creación de páginas de error personalizadas para códigos 404 y 500.	22/05/2024	22/05/2024	2	2	Programador	34.88
1.5.5	Página de contacto	Desarrollo de la página de contacto del desarrollador.	23/05/2024	23/05/2024	2	2	Programador	34.88
1.5.6	Página de analistas registrados	Implementación de un listado con todos los analistas registrados en la aplicación con información relevante de cara a la api.	23/05/2024	23/05/2024	2	1	Programador	34.88

EDT	Título	Descripción	Fecha de inicio	Fecha de Fin	Horas estimadas	Horas reales	Responsable	Coste (euros)
1.5.7	Api con Swagger	Implementación de la documentación de la API utilizando Swagger.	24/05/2024	24/05/2024	3	3	Programador	52.32
1.5.8	Api con Redoc	Implementación de la documentación de la API utilizando Redoc.	27/05/2024	27/05/2024	1	1	Programador	17.44
1.5.9	Página de inicio mejorada	Mejoras en el diseño y funcionalidad de la página de inicio.	28/05/2024	28/05/2024	2	1	Programador	34.88
1.5.10	CSS	Desarrollo de estilos CSS para la fase 4 del proyecto.	29/05/2024	29/05/2024	1	1	Programador	17.44
1.5.11	Testing	Desarrollo de tests unitarios y de selenium.	29/05/2024	29/05/2024	5	5	Programador	87.2
1.5.12	Seguimiento Fase 4	Son los documentos del seguimiento del proyecto que incluyen el informe de avances, la retrospectiva y el informe de cambios para la Fase 4.	30/05/2024	30/05/2024	6	6	Programador	104.64

Cuadro 8.3.5 Diccionario EDT-Cierre

<i>EDT</i>	<i>Título</i>	<i>Descripción</i>	<i>Fecha de inicio</i>	<i>Fecha de Fin</i>	<i>Horas estimadas</i>	<i>Horas reales</i>	<i>Responsable</i>	<i>Coste (euros)</i>
1.6	Cierre	Actividades finales para la fase de cierre del proyecto.	31/05/2024	24/06/2024	44	44	—	936.32
1.6.1	Lecciones aprendidas	Documentación de las lecciones aprendidas.	31/05/2024	31/05/2024	1	1	Jefe de Proyectos	21.28
1.6.2	Manuales de usuario	Creación de manuales de usuario.	03/06/2024	03/06/2024	3	3	Jefe de Proyectos	63.84
1.6.3	Finalización de la memoria	Finalización y revisión de la memoria del proyecto.	04/06/2024	19/06/2024	30	30	Jefe de Proyectos	638.40
1.6.4	Presentación	Preparación y realización de la presentación final.	20/06/2024	24/06/2024	10	10	Jefe de Proyectos	212.80

8.4 Cronograma

En esta sección detallaremos las actividades y tareas específicas, sus duraciones, el elemento del diccionario de la EDT con el que se relaciona y las horas invertidas desde el inicio hasta el final de la aplicación. Este cronograma se estructura según el Diccionario de la Estructura de Desglose del Trabajo, tal y como lo definimos en la sección anterior. Cada entrada del cronograma corresponde a un elemento del diccionario, y proporciona una visión integral y organizada del progreso y la gestión del proyecto durante la fase de planificación.

En la Figura [8.4.1](#) se observa el diagrama de Gantt para la planificación. Cada una de las fases del proyecto se ven en las Figuras [8.4.2](#), [8.4.3](#), [8.4.4](#) y [8.4.5](#). Por último, la etapa de cierre podrán verla en la Figura [8.4.6](#).

Figura 8.4.1: Diagrama de Gantt-Planificación.

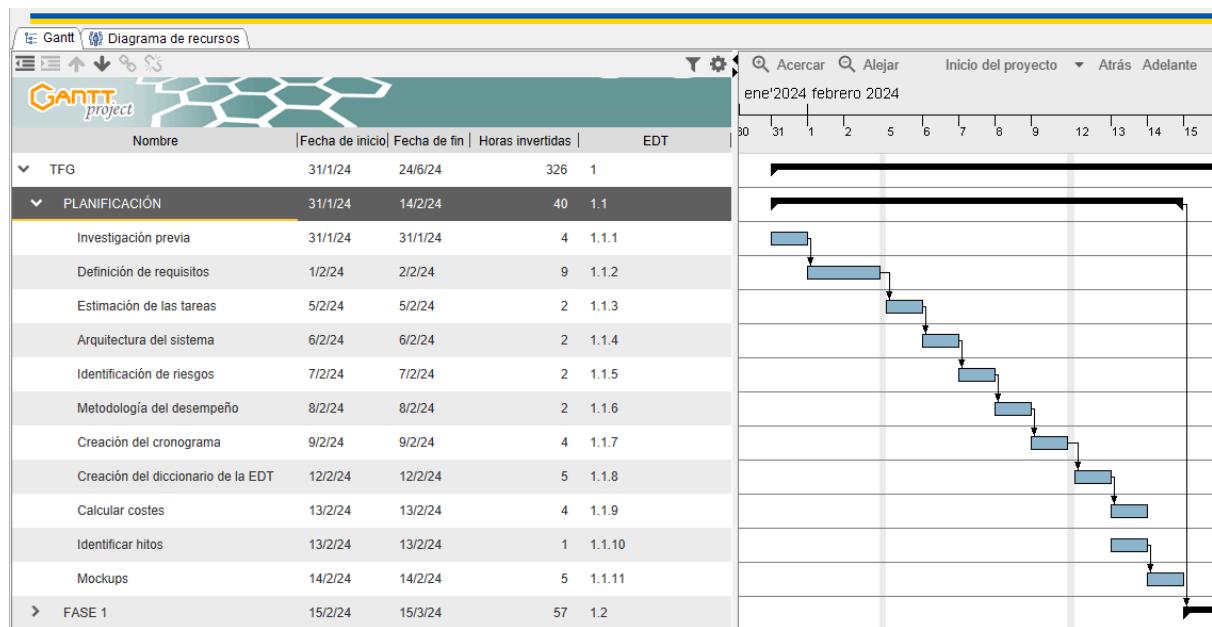


Figura 8.4.2: Diagrama de Gantt-Fase 1.

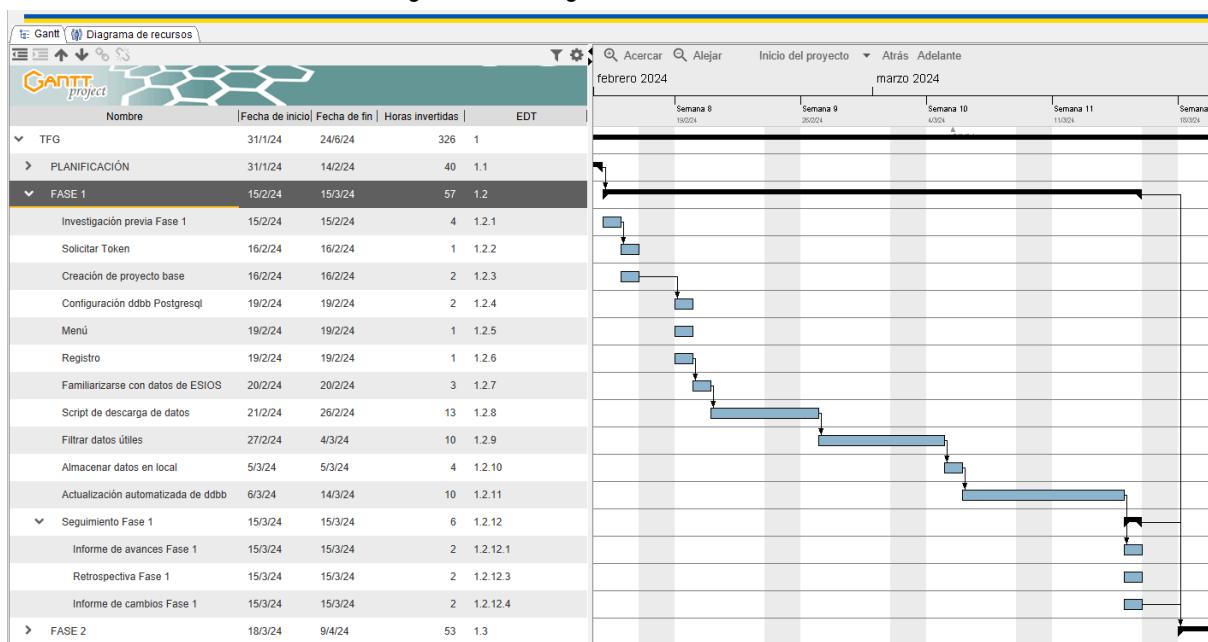


Figura 8.4.3: Diagrama de Gantt-Fase 2.

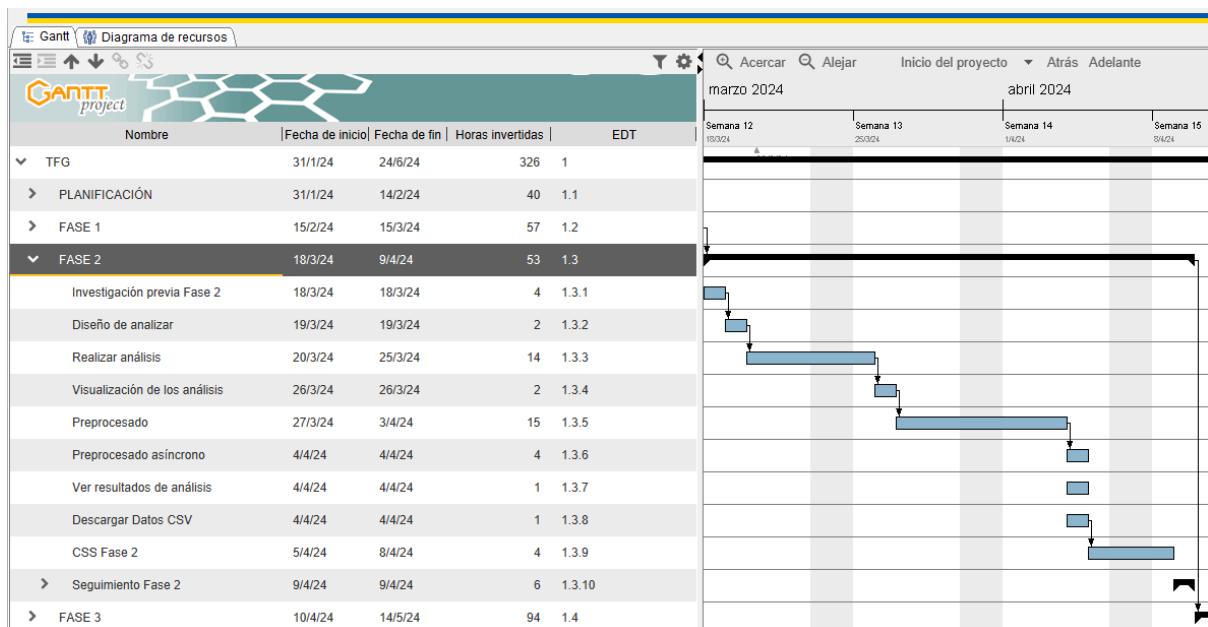


Figura 8.4.4: Diagrama de Gantt-Fase 3.

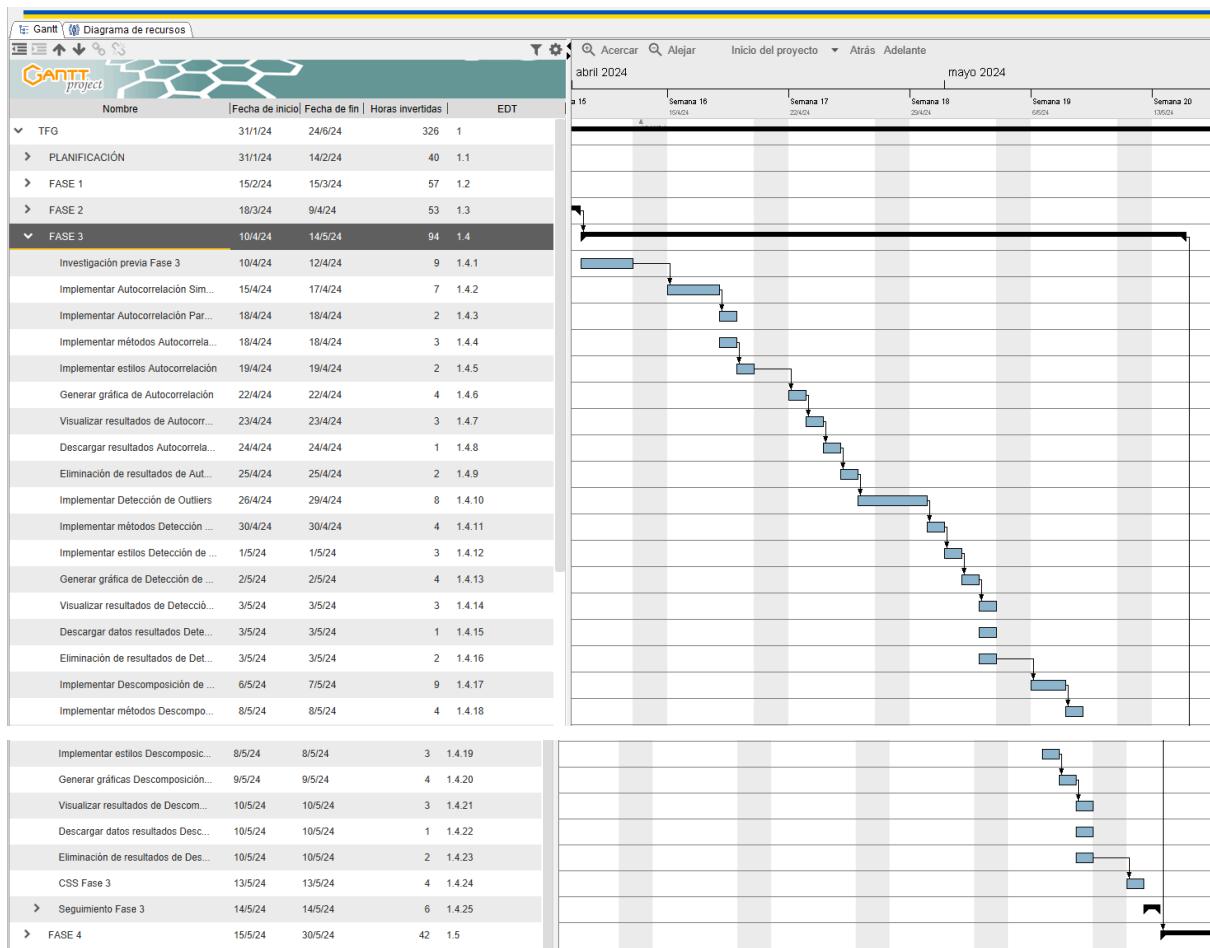


Figura 8.4.5: Diagrama de Gantt-Fase 4.

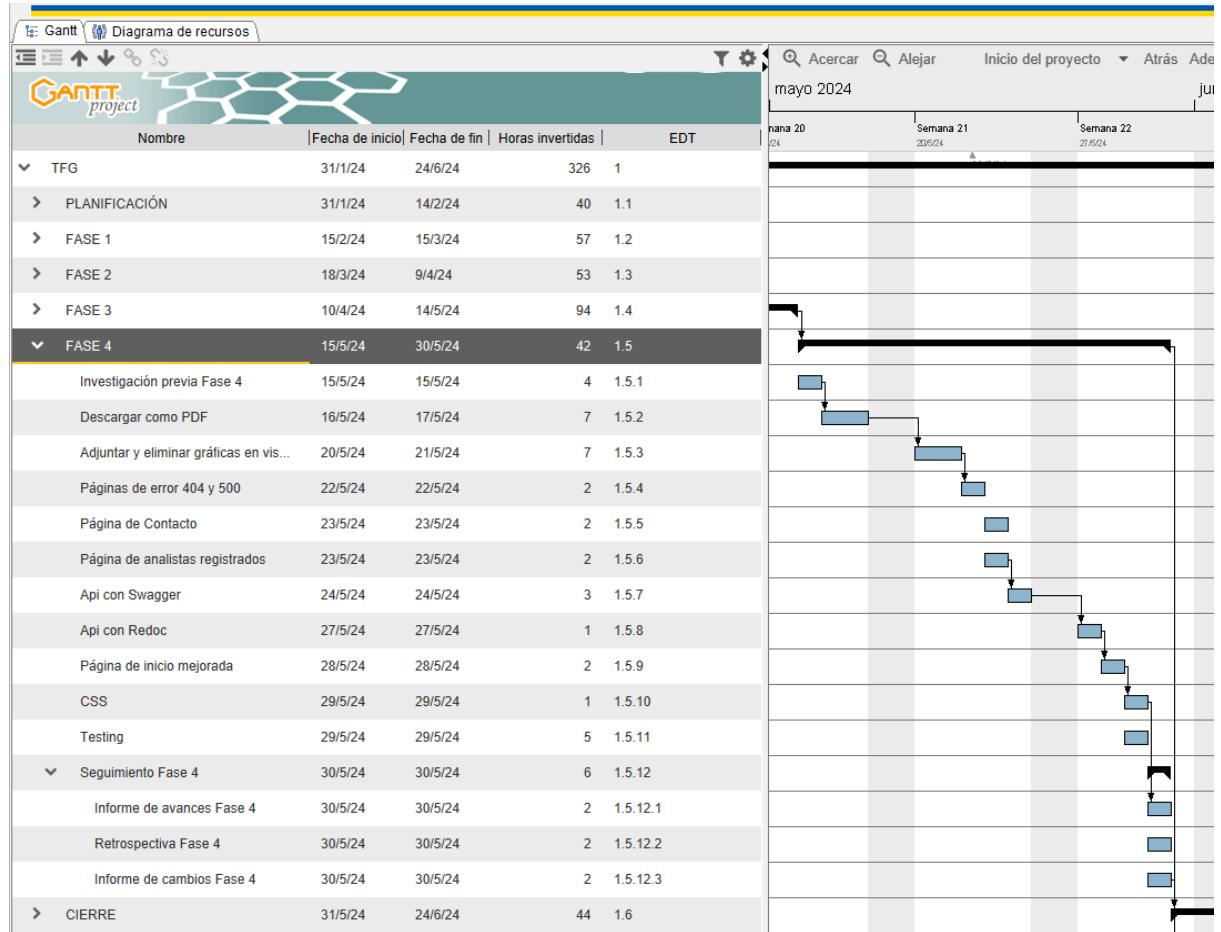
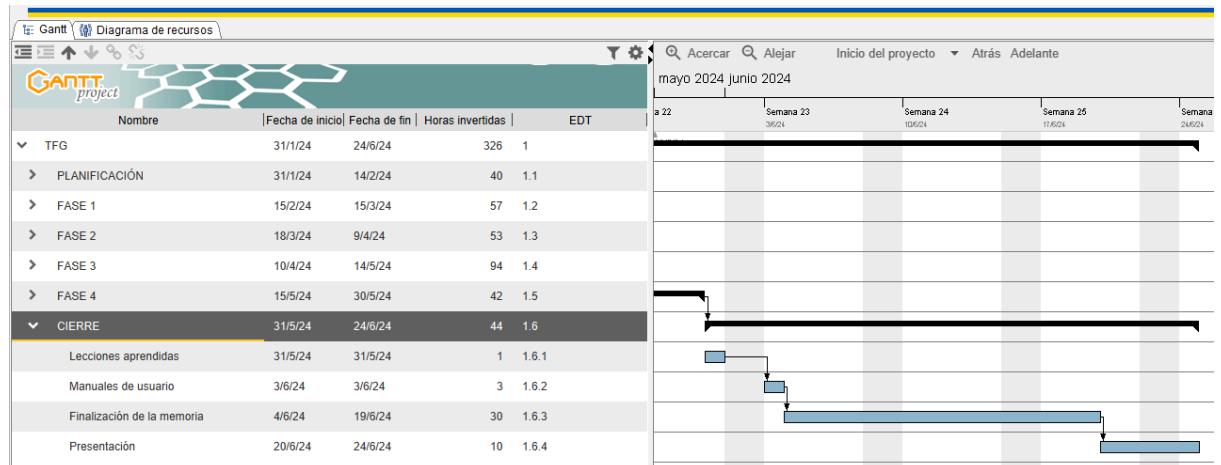


Figura 8.4.6: Diagrama de Gantt-Cierre.



8.5 Costes

En esta sección, se presenta una visión detallada de los costes asociados al desarrollo del producto final. La evaluación precisa de los costes es fundamental para asegurar la viabilidad y el control del presupuesto del proyecto. Los componentes de los costes del proyecto incluyen:

8.5.1 Coste de personal

Este apartado abarca los salarios y beneficios de todos los involucrados en el proyecto, como analistas, programadores y jefes de proyectos. Se calcula en función de las horas invertidas por cada responsable en las distintas actividades y tareas.

- Jefe de Proyectos:
- Analista:
- Programador:

En el Cuadro [8.5.1](#) se ven representados los ingresos promedios para cada uno de estos roles en España [\[30\]](#).

Cuadro 8.5.1: Coste de personal.

Rol	Coste/Hora	Horas/Rol	Coste total
Jefe de Proyectos	21.28 €	21.5	457.5 €
Analista	14.39 €	15	215.85 €
Programador	17.44 €	295.5	5153.52 €
		332	5844.87 €

8.5.2 Coste de material

Componen los gastos asociados a la compra de hardware, software y otros suministros necesarios para el desarrollo y ejecución del proyecto (Cuadro [8.5.2](#)).

Cuadro 8.5.2: Coste de material.

Nombre del material	Precio
PC de trabajo	1300 €
	Total: 1300 €

8.5.3 Coste por consumo

Este coste (Cuadro 8.5.3) se refiere al gasto en electricidad y otros recursos energéticos consumidos durante el desarrollo del proyecto, incluyendo el uso de servidores y estaciones de trabajo.

El coste del Kw hora aparece según el coste medio en España [31].

Cuadro 8.5.3: Coste por consumo.

Nombre	Unidad	Coste/Unidad	Horas totales	Coste total
Electricidad	Kw/H	0.2 €	332	66.4 €

8.5.4 Resumen de Costes

El presupuesto final (Cuadro 8.5.4) es el resultado de la suma de todos los costes mencionados. Este total ofrece una perspectiva completa del gasto necesario para la realización del proyecto, asegurando una gestión eficiente de los recursos financieros y permitiendo ajustes necesarios para mantener la viabilidad económica del proyecto. Se establecerá un margen de contingencia del 10% en caso de retrasos o problemas durante el desarrollo.

Cuadro 8.5.4: Resumen de costes.

Tipo de coste	Coste
Personal	5.926,61 €
Material	1.300 €
Consumo	66,4 €
Suma de Costes	7.327,89 €
Contingencia del 10%	732.79 €
Presupuesto total	8.060,68 €

Quinta parte

V-SEGUIMIENTO

9 Introducción del seguimiento

El seguimiento del proyecto es una fase crucial en el desarrollo de una aplicación web, ya que garantiza que todas las actividades y tareas planificadas se ejecuten conforme a los objetivos establecidos. Este proceso permite identificar desviaciones, gestionar riesgos y asegurar la calidad del producto final.

En cada fase del desarrollo se documentarán las incidencias que surjan, las desviaciones y los riesgos, así como la replanificación necesaria para abordar estas situaciones. También se incluirá una retrospectiva final que resuma los puntos positivos y negativos de cada fase del desarrollo.

Para las **incidencias** (Cuadros [9.1.1](#), [9.2.1](#), [9.3.1](#)) se abordarán los siguientes puntos:

- *EDT*: Identificador de la tarea del diccionario de la EDT.
- *Nombre*: Nombre de la tarea afectada por la incidencia.
- *Descripción del problema*: Descripción detallada de la incidencia.
- *Solución del problema*: Estrategia para la resolución o reestructuración del problema.

En cuanto a las **desviaciones**, se considerarán los siguientes aspectos:

- *Variación de coste* (Cuadros [9.1.2.1](#), [9.2.2.1](#), [9.3.2.1](#), [9.4.2.1](#)): Descripción de las diferencias en el costo planificado. Los costes representan lo que se lleva gastado hasta ese determinado punto del proyecto, por lo que también se tendrán en cuenta los gastos de materiales y por consumo (este se suma al principio, no es proporcional).
- *Variación del trabajo* (Cuadros [9.1.2.2](#), [9.2.2.2](#), [9.3.2.2](#), [9.4.2.2](#)): Diferencias en el esfuerzo requerido.
- *Completitud* (Cuadros [9.1.2.3](#), [9.2.2.3](#), [9.3.2.3](#), [9.4.2.3](#)): Estado de avance de la tarea respecto al plan original.

Sobre los **riesgos** surgidos:

- *Descripción*: Explicación del riesgo identificado.
- *Análisis*: Evaluación del impacto y la probabilidad del riesgo.
- *Estrategia*: Plan de acción para mitigar el riesgo.

La **replanificación** detalla la nueva estrategia a seguir para asegurar la correcta finalización del producto, las desviaciones podrán ser afectadas en este caso.

Finalmente, la **retrospectiva** permitirá evaluar la metodología utilizada, identificando áreas de mejora para las fases subsecuentes del proyecto.

9.1 Primera fase

En esta sección hablaremos sobre cómo ha sido el desarrollo del proyecto durante la primera etapa en su fase inicial.

9.1.1 Incidencias

A continuación detallaremos las incidencias sufridas a lo largo de esta etapa.

Cuadro 9.1.1: Incidencias Fase 1.

EDT	Nombre	Problema	Solución
1.2.8	<i>Script de descarga de datos</i>	Demasiada generalidad de datos.	Se decidió seleccionar un conjunto de datos más específico y descargar únicamente los necesarios para no saturar la base de datos local.
1.2.8	<i>Script de descarga de datos</i>	Complejidad a la hora de obtener los datos.	Se obtuvo un script de referencia para la obtención de datos, que sirvió para el diseño de muchos scripts diferentes para obtener datos más específicos.
1.2.10	<i>Almacenar datos en local</i>	No se podían cargar correctamente los datos con la instrucción del manage.py loaddata.	Se diseñó una instrucción personalizada llamada load_data que clasifica los datos y los guarda.
1.2.11	<i>Actualización automática de la base de datos</i>	No se ha completado esta tarea en el plazo establecido.	Reestructuración de la fase 2 y priorizar esta tarea.

9.1.2 Desviaciones

En esta sección hablaremos de las desviaciones, en cuanto a costes se refiere, sufridas al final de esta primera fase.

Variación del coste

Cuadro 9.1.2.1: Variación de coste Fase 1.

Variación	Coste estimado	Coste real	Variación total	Coste acumulado previsto	Coste acumulado real
19.44%	986,72	1.178,56	6.21 %	3.086,83	3.278,67

Justificación: Un incremento de horas trabajadas afecta directamente a un incremento de los costos.

Variación del trabajo

Cuadro 9.1.2.2: Variación del trabajo Fase 1.

Variación	Trabajo estimado	Trabajo real
19.29%	57	68

Justificación: La dificultad de las tareas, la dificultad para comprender los datos y el desconocimiento general de la tecnología han conducido a una mala estimación de las horas para esta primera fase.

Completitud

Cuadro 9.1.2.3: Completitud Fase 1.

Porcentaje	Tareas pendientes
90.90%	1

Justificación: El incremento de horas en realizar otras tareas que aparentemente iban a tener una duración menor y la necesidad de aprendizaje de nuevas tecnologías no permitieron completar la tarea del guardado de datos automatizado en la fecha establecida en el cronograma.

9.1.3 Riesgos que han aparecido

Aquí identificaremos los riesgos con los que se ha lidiado de los mencionados en la sección de riesgos al comienzo de la planificación, así como otros imprevistos.

- R 005:
 - Descripción: Retrasos en la entrega.
 - Análisis: Para el guardado de datos CSV a la base de datos en local .
 - Estrategia: Se dejará para la siguiente fase dando preferencia a esta tarea.
- R 006:
 - Descripción: Exceder el presupuesto.
 - Análisis: El incremento de horas trabajadas ha influido proporcionalmente en un incremento de los costos.
 - Estrategia: Mejora en la estimación de las tareas para la siguiente Fase.
- R 009:
 - Descripción: Requisito no identificado.
 - Análisis: Para almacenar los datos de los CSVs en la base de datos ha sido necesaria la implementación de la instrucción *load_data*, ejecutable con *python manage.py load_data*.
 - Estrategia: Dedicar horas de la tarea de automatización en terminar el almacenamiento de los datos.
- R 007:
 - Descripción: Falta de conocimientos técnicos.
 - Análisis Se ha notado una falta de conocimiento de nuevas tecnologías tanto para la obtención de los datos, como para la implementación de la tarea de automatización que aún sigue pendiente.
 - Estrategia: Solicitar información a tutores, así como dedicar más horas a la investigación para aclarar ciertas nociones básicas.
- R 008:
 - Descripción: Holgura durante la fase de desarrollo.
 - Análisis: Como consecuencia de los riesgos anteriores, la Fase 1 ha durado más de lo previsto.
 - Estrategia: Replanificación de la Fase 2 para una mejor administración del tiempo.
- R 003:
 - Descripción: Calidad del producto insatisfactoria.
 - Análisis: No se han cumplido todas las tareas para esta Fase.
 - Estrategia:Replanificación de la Fase 2 para una mejor administración del tiempo.

9.1.4 Replanificación

Al comienzo de la Fase 2 se debe realizar la tarea de guardado antes de continuar con el resto de tareas, dedicando más tiempo al estudio previo.

- Se aumentarán las horas de la Fase 2 de 53 a 60.

9.1.5 Retrospectiva

- Aspectos positivos:
 - Buena comunicación entre el cliente y el desarrollador.
 - Constancia a la hora de trabajar.
 - Implicación y ganas de seguir aprendiendo.
- Aspectos negativos:
 - Mala estimación de tiempo en cada tarea.
 - Desconocimiento de la materia sobre todo en cuanto al uso de Celery.
 - Tareas pendientes para esta fase.
 - No se ha cumplido con los costos esperados

Comentarios: De cara a la siguiente Fase se hará un estudio más rugoso antes de tratar de implementar una determinada tarea, además, mejoraremos aún más la comunicación con el cliente para tener más claros los conceptos teóricos y mejorar el producto final.

9.2 Segunda fase

En esta sección hablaremos sobre cómo ha sido el desarrollo del proyecto durante la segunda etapa teniendo en cuenta la retrospectiva anterior.

9.2.1 Incidencias

A continuación detallaremos las incidencias sufridas a lo largo de esta etapa.

Cuadro 9.2.1: Incidencias Fase 2.

EDT	Nombre	Problema	Solución
1.3.6	Preprocesado asíncrono	Bloqueo en tarea del preprocesamiento con el servidor de Redis	Cambiar de Redis a RabbitMQ.

9.2.2 Desviaciones

En esta sección hablaremos de las desviaciones, en cuanto a costes se refiere, sufridas al final de esta primera fase.

Variación del coste

Cuadro 9.2.2.1: Variación de coste Fase 2.

Variación	Coste estimado	Coste real	Variación total	Coste acumulado previsto	Coste acumulado real
24.85 %	912,12	1.138,84	10,46 %	3.998,95	4.417,51

Justificación: La tarea de automatización que no se terminó en la Fase 1 ha supuesto un incremento en horas de trabajo, que ha provocado un incremento del costo para la Fase 2, así como un problema con el servidor de Redis unido al desconocimiento de la tecnología, provocó un incremento sustancioso en cuanto a horas de trabajo se refiere.

Comentarios: El nuevo coste estimado se debe al incremento de horas de trabajo por la tarea que no se realizó durante la Fase anterior.

Variación del trabajo

Cuadro 9.2.2.2: Variación del trabajo Fase 2.

Variación	Trabajo estimado	Trabajo real
24.52 %	53	66

Justificación: La tarea de automatización que no se terminó en la Fase 1 ha supuesto un incremento de las horas trabajadas durante la Fase 2.

Completitud

Cuadro 9.2.2.3: Completitud Fase 2.

Porcentaje	Tareas pendientes
100%	0

Justificación: Gracias a la mejora en cuanto a la distribución del trabajo, un incremento sustancioso de implicación y un conocimiento mayor para la mayoría de tareas por la experiencia al haber trabajado en proyectos similares, se ha podido terminar con todas las tareas satisfactoriamente en el plazo acordado.

9.2.3 Riesgos que han aparecido

Aquí identificaremos los riesgos con los que se ha lidiado de los mencionados en la sección de riesgos al comienzo de la planificación, así como otros imprevistos.

- R 006:
 - Descripción: Exceder el presupuesto.
 - Análisis: El incremento de horas trabajadas ha influido proporcionalmente en un incremento de los costos.
 - Estrategia: No procede.

- R 007:
 - Descripción: Falta de conocimientos técnicos.
 - Análisis: Pese a que se hayan cumplido con los plazos establecidos, se ha incrementado de manera muy significativa las horas necesarias para realizar las tareas de automatización.
 - Estrategia: No procede.

9.2.4 Replanificación

Se ha terminado satisfactoriamente con todas las tareas por lo que no es necesario replanificar para la siguiente fase.

9.2.5 Retrospectiva

- Aspectos positivos:

- Buena comunicación entre el cliente y el desarrollador.
- Mejora sustancial en la constancia a la hora de trabajar.
- Mucha más implicación.
- Desenvoltura a la hora de afrontar problemas.
- Aumento sustancial de conocimientos y comprensión de la tecnología.

- Aspectos negativos:

- La estimación de horas sigue sin estar bien definida.

Comentarios: Pese a los riesgos que siguen presentes, brindados por etapas de bloqueo, se ha adquirido gran desenvoltura a la hora de realizar las tareas, aumentando el conocimiento generalizado en la tecnología. No obstante se debe mejorar aún más en cuanto a la organización, estableceremos un horario específico para el desarrollo del producto.

9.3 Tercera fase

En esta sección hablaremos sobre cómo ha sido el desarrollo del proyecto durante la tercera etapa teniendo en cuenta la retrospectiva anterior.

9.3.1 Incidencias

A continuación detallaremos las incidencias sufridas a lo largo de esta etapa.

Cuadro 9.3.1: Incidencias Fase 3.

EDT	Nombre	Problema	Solución
1.4.6	<i>Generar gráficas de autocorrelación</i>	Después de una reunión con el cliente, se especificó que las gráficas debían ser necesariamente dinámicas.	Implementación con la librería de Ploty.
1.4.9	<i>Eliminar resultados de autocorrelación</i>	Se rediseñó la manera de eliminar los resultados.	Se sustituyó la eliminación de resultados por checkbox a la eliminación por el título de la Autocorrelación.

9.3.2 Desviaciones

En esta sección hablaremos de las desviaciones, en cuanto a costes se refiere, sufridas al final de esta primera fase.

Variación del coste

Cuadro 9.3.2.1: Variación del coste Fase 3.

Variación	Coste estimado	Coste real	Variación total	Coste acumulado previsto	Coste acumulado real
8.34 %	1.671,98	1.811,50	9,66 %	5.670,93	6.229,01

Justificación: Los cambios en las tareas han supuesto un leve incremento del coste

Variación del trabajo

Cuadro 9.3.2.2: Variación del trabajo Fase 3.

Variación	Trabajo estimado	Trabajo real
8.51 %	94	102

Justificación: Los cambios han supuesto un incremento en horas de trabajo.

Completitud

Cuadro 9.3.2.3: Completitud Fase 3.

Porcentaje	Tareas pendientes
100%	0

Justificación: La desenvoltura en la tecnología y la inclusión en la materia en cuanto al apartado teórico sobre lo que se debe hacer, ha hecho más fácil conocer el cómo hacerlo.

9.3.3 Riesgos que han aparecido

Aquí identificaremos los riesgos con los que se ha lidiado de los mencionados en la sección de riesgos al comienzo de la planificación, así como otros imprevistos.

- R 006:
 - Descripción: Exceder el presupuesto.
 - Análisis: El incremento de horas trabajadas ha influido proporcionalmente en un incremento de los costos.
 - Estrategia: No procede.

9.3.4 Replanificación

Se ha terminado satisfactoriamente con todas las tareas por lo que no es necesario replanificar para la siguiente fase.

9.3.5 Retrospectiva

- Aspectos positivos:
 - Se han mantenido y mejorado los aspectos positivos de la fase anterior.
 - Se ha visto reflejado el aumento progresivo de conocimientos y herramientas para afrontar situaciones complejas
- Aspectos negativos:
 - La reestructuración de requisitos ha afectado a los costes.

Comentarios: Esta fase refleja el aumento de conocimiento y desenvoltura con la tecnología y el lenguaje de Django.

9.4 Cuarta fase

En esta sección hablaremos sobre cómo ha sido el desarrollo del proyecto durante la última etapa de desarrollo.

9.4.1 Incidencias

Aunque no hubo ninguna incidencia durante esta etapa, quisiera mencionar que en la tarea 1.5.2, que consiste en generar el PDF con los resultados, me habría gustado incluir las imágenes adjuntas en las vistas de resultados. A pesar de que este requisito no estaba definido, me habría gustado implementarlo. No obstante, debido a la falta de tiempo, no fue posible. Sin embargo, el PDF con los resultados se genera correctamente.

9.4.2 Desviaciones

En esta sección hablaremos de las desviaciones, en cuanto a costes se refiere, sufridas al final de esta primera fase.

Variación del coste

Cuadro 9.4.2.1: Variación del coste Fase 4.

Variación	Coste estimado	Coste real	Variación total	Coste acumulado previsto	Coste acumulado real
0 %	720,40	720,40	8,62 %	6.391,21	6.949,3

Justificación: Menos horas invertidas justifican unos costos menores.

Comentario: Gracias a la reserva de contingencia del 10% definida en la planificación, se ha logrado cumplir con los costes totales.

Variación del trabajo

Cuadro 9.4.2.2: Variación del trabajo Fase 4.

Variación	Trabajo estimado	Trabajo real
0 %	42	42

Justificación: Los conocimientos adquiridos han facilitado terminar el producto antes de lo imaginado .

Compleitud

Cuadro 9.4.2.3: Completitud Fase 4.

Porcentaje	Tareas pendientes
100%	0

Justificación: Se ha logrado obtener el producto completo.

9.4.3 Riesgos que han aparecido

No procede.

9.4.4 Replanificación

No procede.

9.4.5 Retrospectiva

- Aspectos positivos:
 - Se han mantenido y mejorado los aspectos positivos de la fase anterior.
 - Se ha visto reflejado el aumento progresivo de conocimientos y herramientas para afrontar situaciones complejas.
- Aspectos negativos:
 - Pese a haber realizado todas las tareas como se especificó en los requisitos, no ha sido posible agregar las gráficas al PDF.

Comentarios: Esta fase refleja las mejoras en el desempeño y conocimientos, logrando obtener un producto correcto en el plazo acordado.

10. Esfuerzo empleado

En esta sección se detallan las horas de trabajo reales que se han empleado para el desempeño de este proyecto. Podrán ver el resumen en el Cuadro 10.

Las horas de trabajo acumulan un total de 362 horas.

Cuadro 10 Esfuerzo real empleado.

EDT	Tarea	Horas de trabajo reales
1	PROYECTO FINAL	362
1.1	<i>Planificación</i>	40
1.1.1	<i>Investigación previa</i>	4
1.1.2	<i>Definición de requisitos</i>	10
1.1.3	<i>Estimación de tareas</i>	1
1.1.4	<i>Arquitectura del sistema</i>	2
1.1.5	<i>Identificación de riesgos</i>	1
1.1.6	<i>Metodología durante el desempeño</i>	1
1.1.7	<i>Creación del cronograma</i>	5
1.1.8	<i>Creación del diccionario de la EDT</i>	8
1.1.9	<i>Calcular costes</i>	2
1.1.10	<i>Identificar hitos</i>	1
1.1.11	<i>Mockups</i>	5
1.2	FASE 1	68
1.2.1	<i>Investigación previa Fase 1</i>	4
1.2.2	<i>Solicitar token</i>	1
1.2.3	<i>Creación proyecto base</i>	1
1.2.4	<i>Configuración de la base de datos PostgreSQL</i>	2
1.2.5	<i>Menú</i>	1
1.2.6	<i>Registro</i>	1
1.2.7	<i>Familiarizarse con datos ESIOS</i>	6
1.2.8	<i>Script de descarga de datos</i>	19

1.2.9	<i>Filtrar datos útiles</i>	7
1.2.10	<i>Almacenar datos en local</i>	10
1.2.11	<i>Actualización automática de la base de datos</i>	17
1.2.12	<i>Seguimiento Fase 1</i>	6
1.3	FASE 2	66
1.3.1	<i>Investigación previa Fase 2</i>	4
1.3.2	<i>Diseño de análisis</i>	3
1.3.3	<i>Realizar análisis</i>	5
1.3.4	<i>Visualización de los análisis</i>	3
1.3.5	<i>Preprocesado</i>	19
1.3.6	<i>Preprocesado asíncrono</i>	12
1.3.7	<i>Ver resultados de análisis</i>	1
1.3.8	<i>Descargar datos CSV</i>	2
1.3.9	<i>CSS Fase 2</i>	4
1.3.10	<i>Seguimiento Fase 2</i>	6
1.4	FASE 3	102
1.4.1	<i>Investigación previa Fase 3</i>	9
1.4.2	<i>Implementar Autocorrelación simple</i>	15
1.4.3	<i>Implementar Autocorrelación parcial</i>	3
1.4.4	<i>Implementar métodos Autocorrelación</i>	3
1.4.5	<i>Implementar estilos Autocorrelación</i>	6
1.4.6	<i>Generar gráficas Autocorrelación</i>	7
1.4.7	<i>Resultados de Autocorrelación</i>	5
1.4.8	<i>Descargar resultados de Autocorrelación</i>	1
1.4.9	<i>Eliminar resultados de Autocorrelación</i>	3

1.4.10	<i>Implementar Detección de Outliers</i>	5
1.4.11	<i>Implementar métodos Detección de Outliers</i>	2
1.4.12	<i>Implementar estilos Detección de Outliers</i>	3
1.4.13	<i>Generar gráfica de Detección de Outliers</i>	2
1.4.14	<i>Resultados de Detección de Outliers</i>	2
1.4.15	<i>Descargar datos resultados de Detección de Outliers</i>	1
1.4.16	<i>Eliminación de resultados de Detección de Outliers</i>	1
1.4.17	<i>Implementar Descomposición de Series Temporales</i>	9
1.4.18	<i>Implementar métodos Descomposición de Series Temporales</i>	6
1.4.19	<i>Implementar estilos Descomposición de Series Temporales</i>	3
1.4.20	<i>Generar gráficas de Descomposición de Series Temporales</i>	2
1.4.21	<i>Resultados de Descomposición de Series Temporales</i>	2
1.4.22	<i>Descargar datos resultados de Descomposición de Series Temporales</i>	1
1.4.23	<i>Eliminación de resultados de Descomposición de Series Temporales</i>	1
1.4.24	CSS Fase 3	4
1.4.25	Seguimiento Fase 3	6
1.5	FASE 4	42
1.5.1	<i>Investigación previa Fase 4</i>	3
1.5.2	Descargar PDF	7
1.5.3	Adjuntar y eliminar gráficas en	10

	vistas	
1.5.4	Páginas de error 404 y 500s	2
1.5.5	Página de contacto	2
1.5.6	Página de analistas registrados	1
1.5.7	Api con Swagger	3
1.5.8	Api con Redoc	1
1.5.9	Página de inicio mejorada	1
1.5.10	CSS	1
1.5.11	Testing	5
1.5.12	<i>Seguimiento Fase 4</i>	6
1.6	<i>Cierre</i>	44
1.6.1	Lecciones aprendidas	1
1.6.2	Manuales de usuario	3
1.6.3	Finalización de la memoria	30
1.6.4	Presentación	10

11.Coste final

En esta sección veremos las desviaciones sufridas a lo largo de este proyecto.

Si observamos la Figura 11.1, podremos observar cómo ha sido cada una de las etapas. En la fase de planificación vemos que no han habido desvíos, puesto que la experiencia adquirida gracias a otros proyectos, han facilitado una estimación correcta del tiempo de trabajo. Durante las fases de desarrollo podemos ver que lamentablemente la estimación de las horas de trabajo realizadas en la fase inicial no ha sido la más acertada, no obstante, se ha logrado realizar la aplicación en su totalidad sin pasarnos necesariamente del presupuesto gracias al 10% de contingencia que se tuvo en cuenta por si ocurrían problemas durante el desarrollo. Podrán encontrar más información sobre los costes del proyecto en el apartado [8.5](#).

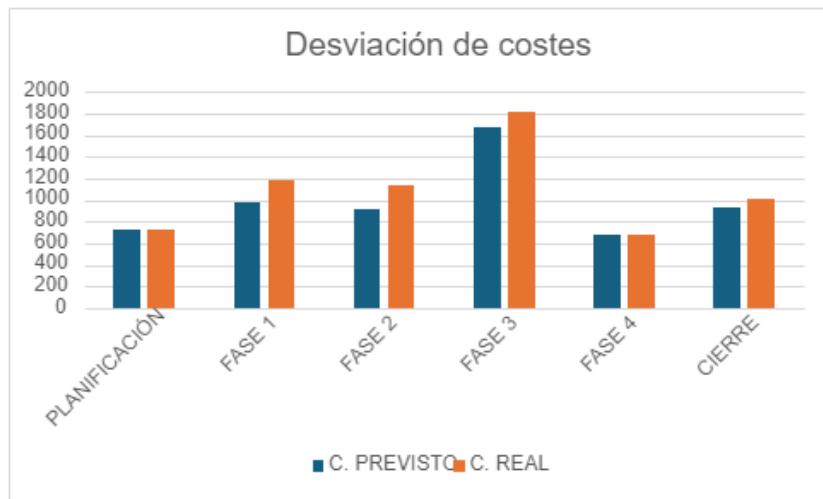


Figura 11.1: Desviación de costes.

Por otro lado, en la Figura 11.2 se ilustra la evolución de los costes acumulados previstos y reales a lo largo de las diferentes fases. Se puede apreciar que la variación no es muy significativa, aunque los costes acumulados reales son mayores que los previstos en prácticamente todas las etapas del proyecto excepto durante la etapa planificación o la Fase 4 de desarrollo:

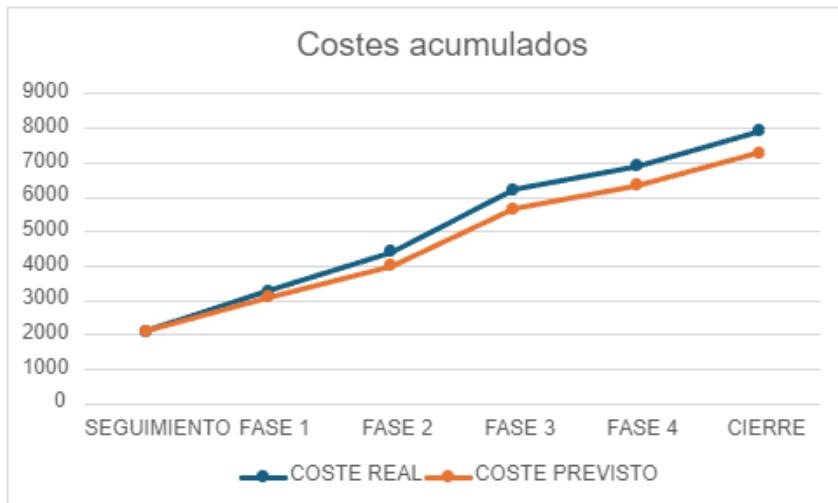


Figura 11.2: Costes acumulados.

En estas dos tablas podremos ver mejor la fluctuación de costes a lo largo de todas las etapas del proyecto (Cuadros [11.1](#) y [11.2](#)).

Coste estimado:

Cuadro 11.1: Coste estimado.

Equipo y consumo	Planificación	Desarrollo	Cierre	Coste estimado
1.366,4 €	734,07 €	4.291,1 €	936,32 €	7.327,89€

Coste real:

Cuadro 11.2: Coste real.

Equipo y consumo	Planificación Real	Desarrollo Real	Cierre Real	Coste real
1.366,4 €	734,07 €	4.839,18 €	1.021,44 €	7.961,09 €

El coste final sería de **7.961,09 €** contando con los costes de materiales y consumo, lo cual es inferior al presupuesto mencionado de **8.060,68 €** en la sección [8.5.4](#), cumpliendo así con el presupuesto acordado.

Para cualquier consulta sobre la evolución de los costes durante las fases de desarrollo vaya a la sección de [seguimiento](#).o bien consulte cualquiera de los costes estimados en el diccionario de la EDT en el apartado [8.3](#).

Sexta Parte

VI-CIERRE

12 Producto final

En esta sección ahondaremos en el producto como resultado de la ejecución de todas las fases del proyecto, explicaré cómo descargar el proyecto, así como instalar las dependencias necesarias para ejecutarlo, además, se mostrarán las vistas con cada una de las funcionalidades ya implementadas.

12.1 Manual de instalación

El manual de instalación proporciona la información necesaria para desplegar el proyecto en local. Los pasos necesarios para el despliegue son los siguientes:

1. **Descargar aplicación:** Será necesario descargar el repositorio de GitHub del proyecto. Acceda a este enlace para la descarga:

URL: <https://github.com/CarlosNuchera/TFG>

2. **Descargar python:** Descarga de la versión 3.12 desde la página oficial. Acceda a este enlace para la descarga:

URL: <https://www.python.org/downloads/>

3. **Creación de un entorno virtual:** Creación dentro de nuestro proyecto para evitar conflictos con las dependencias. Estos son los comandos necesarios para inicializarlo:

```
pip install virtualenv      #Instalar virtualenv
```

```
virtualenv venv            #Crear entorno virtual venv
```

```
venv\Scripts\activate        #Activación del entorno virtual (Windows)
```

4. **Instalación de dependencias de pip:** Dentro de la carpeta principal de nuestro proyecto, se encuentra un archivo llamado requirements.txt en el que vienen todas las dependencias que deben ser instaladas para un correcto funcionamiento del proyecto. Para la instalación será necesario abrir una ventana en la terminal con el entorno virtual activado, dirigirnos hasta la carpeta que contiene dicho archivo y escribir el siguiente comando:

```
pip install -r requirements.txt      #Comando de instalación de dependencias
```

5. **Instalación de Postgresql:** Será necesario acceder a la página oficial de Postgresql para descargar e instalar la base de datos. Deben prestar atención a dos puntos importantes durante la instalación.

Aparecerán una serie de casillas para la instalación de diversos programas de apoyo y gestión con postgresql, entre los cuales, descargamos pgAdmin 4 (Figura 12.1.1) para crear la base de datos.

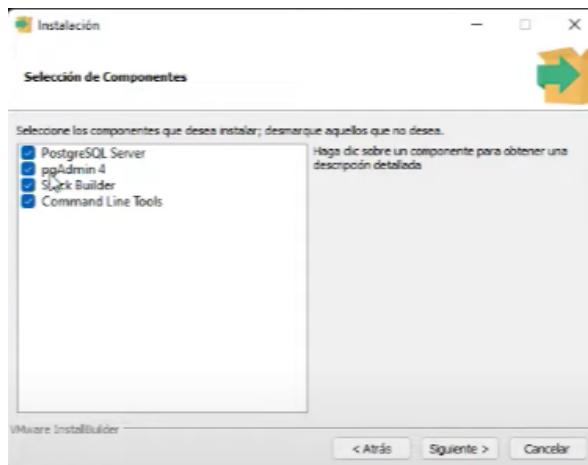


Figura 12.1.1: Instalación de PgAdmin 4.

En algún momento, se solicitará una contraseña para el superusuario deben escribir *admin*, esta será la contraseña para el usuario por defecto *postgres*, es muy importante recordarla (Figura 12.1.2).

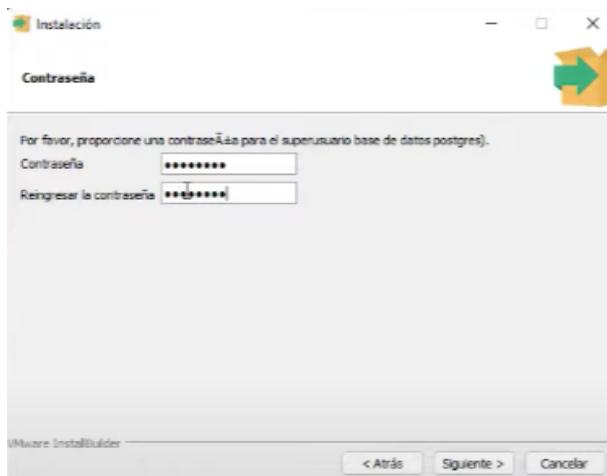


Figura 12.1.2: Contraseña Postgres.

Acceda a este enlace para la descarga:

URL: <https://www.postgresql.org/download/>

6. **Configuración de Postgresql:** En primer lugar debemos abrir PgAdmin descargado durante el paso previo. Una vez ejecutado, nos solicitarán la contraseña del superusuario, la introducimos y accederemos al servidor Postgres. Debemos hacer clic derecho sobre la sección *Databases*, le damos a *Create>Database...*. Nos aparecerá un formulario en el que únicamente debemos llenar estos campos (Figura 12.1.3):

*Database -> tfg
Owner -> postgres*

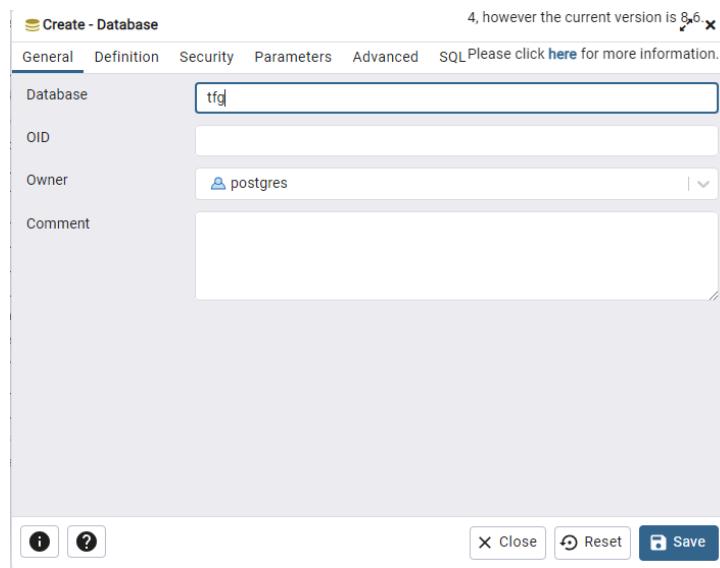


Figura 12.1.3: Creación de base de datos.

7. **Migraciones:** Con la base de datos correctamente configurada y todas las dependencias instaladas procederemos a realizar las migraciones para cargar nuestros modelos en la base de datos. Para ello debemos escribir en una terminal, apuntando al archivo manage.py dentro de nuestro proyecto estas instrucciones:

```
python manage.py makemigrations #Cargar modelos en local
```

```
python manage.py migrate #Enviar migraciones a la base de datos
```

8. **Carga de datos iniciales en la base de datos:** Para realizar los análisis deben guardar los datos de nuestro proyecto en la base de datos, para ello los deben guardar localmente mediante una instrucción ejecutable con manage.py llamada load_data (Figura 12.1.4).

```
import os
import csv
from django.core.management.base import BaseCommand
from esios.models import Datos

class Command(BaseCommand):
    help = 'Guarda datos desde archivos CSV en la base de datos'

    def add_arguments(self, parser):
        parser.add_argument('csv_folder', type=str, help='Ruta a la carpeta que contiene los archivos CSV')

    def handle(self, *args, **kwargs):
        csv_folder = kwargs['csv_folder']

        for root, dirs, files in os.walk(csv_folder):
            for file_name in files:
                if file_name.endswith('.csv'):
                    file_path = os.path.join(root, file_name)
                    if 'demanda' in root:
                        tipo_dato = 'demanda'
```

```

        elif 'gas' in root:
            tipo_dato = 'gas'
        elif 'generaciones' in root:
            tipo_dato = 'generacion'
        elif 'indicadores' in root:
            tipo_dato = 'precio'
        else:
            tipo_dato = 'otro'

        with open(file_path, 'r', encoding='utf-8') as file:
            reader = csv.DictReader(file)
            for row in reader:
                fecha = row['Fecha']
                nombre = row['Nombre']
                precio = row.get('Precio')
                consumo=row.get('Consumo')
                porcentaje = row.get('Porcentaje')
                porcentaje = float(porcentaje) if porcentaje else None
                if not Datos.objects.filter(fecha=fecha, nombre=nombre,
                precio=precio, consumo=consumo, porcentaje=porcentaje).exists():
                    my_model_instance = Datos(
                        tipo_dato=tipo_dato,
                        nombre=nombre,
                        fecha=fecha,
                        precio=precio,
                        consumo=consumo,
                        porcentaje=porcentaje
                    )
                    my_model_instance.save()

        self.stdout.write(self.style.SUCCESS('Datos guardados exitosamente desde los
archivos CSV'))
    
```

Figura 12.1.4: Instrucción load_data.py.

Cabe mencionar que para mayor velocidad de guardado de datos, debemos eliminar el bucle *if* previo a la creación del objeto ‘*datos*’ (*if not Datos...*), así como indentar hacia atrás tanto *my_model_instance= Datos(...)* como *my_model_instance.save()*. Dentro de su entorno virtual deben seguir esta ruta y pegar esta clase llamandola *load_data.py*:

entorno virtual>Lib\site-packages>django>management>commands>load_data.py

Para cargar los datos debe escribir estas instrucciones, una para cada tipo de dato:

```

python manage.py load_data demanda          #Guardar datos de demanda

python manage.py load_data gas              #Guardar datos de gas

python manage.py load_data generaciones    #Guardar datos de generaciones

python manage.py load_data indicadores     #Guardar datos de precio
    
```

Es muy importante que después de guardar los datos, la instrucción *load_data* se deje como estaba.

9. Instalar RabbitMQ: El servidor de rabbit será necesario para realizar las tareas asíncronas del proyecto con Celery. Para el correcto funcionamiento debemos seguir los siguientes pasos:

a. Instalación de Erlang: RabbitMQ requiere Erlang para funcionar, así que primero necesitas instalar Erlang. Acceda a esta url para instalar:

URL: <https://www.erlang.org/downloads>

b. Instalación de RabbitMQ: Una vez que Erlang esté instalado, puedes proceder a instalar RabbitMQ. Acceda a esta url para instalar:

URL: <https://www.rabbitmq.com/docs/download>

c. Despues de instalar RabbitMQ, necesitas configurarlo para que se ejecute como un servicio en Windows.

Abre una terminal con privilegios de administrador y navega al directorio de instalación de RabbitMQ. Por defecto, suele estar en `C:\Program Files\RabbitMQ Server\rabbitmq_server-{version}\sbin`.

Comandos en la terminal:

```
cd "C:\Program Files\RabbitMQ Server\rabbitmq_server-{version}\sbin"
```

```
rabbitmq-service install      #Instalar rabbit como servicio
```

```
rabbitmq-service start       #Inicializar rabbit
```

Verificamos que funcione correctamente accediendo a la administración de tareas y haciendo clic sobre Servicios, verificamos que Rabbit esté correctamente instalado y en ejecución como vemos en la Figura [12.1.5](#).

Servicios					
	Nombre	PID	Descripción	Estado	Grupo
PNRPSvc			Protocolo d...	Detenido	LocalService...
PolicyAgent			Agente de ...	Detenido	NetworkService
postgresql-x64-16	5920	postgresql-...	En ejecución		
Power	1400	Energía	En ejecución	DcomLaunch	
PrintNotify			Extensiones...	Detenido	print
PrintWorkflowUserSvc			PrintWorkfl...	Detenido	PrintWorkflow
PrintWorkflowUserSvc_4...			PrintWorkfl...	Detenido	PrintWorkflow
ProfSvc	2172	Servicio de ...	En ejecución	UserProfile...	
PushToInstall			Servicio Pus...	Detenido	netsvcs
QWAVE			Experiencia ...	Detenido	LocalService...
RabbitMQ	6412	RabbitMQ	En ejecución		
RasAuto			Administrad...	Detenido	netsvcs

Figura 12.1.5: RabbitMQ instalado como servicio.

10. Despliegue: Para el despliegue y el funcionamiento total del proyecto debemos tener un total de tres terminales inicializadas y apuntando hacia el proyecto base con el entorno virtual inicializado.

- a. Primera terminal: Crearemos un superusuario para acceder a menú de administración de django, aunque es un paso opcional:

```
python manage.py createsuperuser
```

Acto seguido inicializamos el servidor para desplegar la aplicación en local:

```
python manage.py runserver
```

- b. Segunda terminal: Con esta terminal inicializamos el proceso de actualización de los datos automatizada con la ayuda de Celery Beat:

```
celery -A tfg beat -l info
```

Debe salirle algo como en la Figura [12.1.6](#).

```
(venv) C:\Users\carlo\Desktop\djangoproject>celery -A tfg beat -l info
celery beat v5.4.0 (opalent) is starting.

[2024-06-16 20:47:14,064: INFO/MainProcess] beat: Starting...
[2024-06-16 20:47:14,272: INFO/MainProcess] DatabaseScheduler: Schedule changed.
```

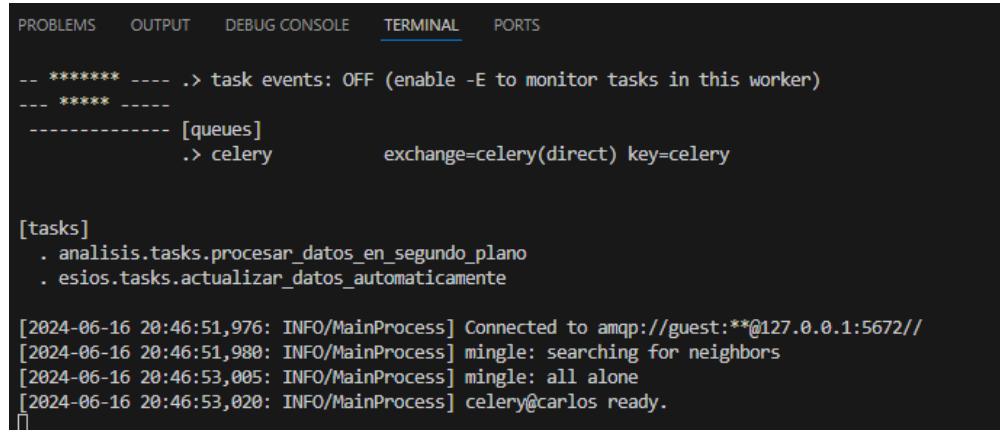
Figura 12.1.6: Celery Beat.

Para acabar con el proceso bastará únicamente con cerrar la terminal.

- c. Tercera terminal: Esta última terminal está enfocada en el worker de Celery para realizar las tareas en segundo plano:

```
celery -A tfg worker -l info --pool=solo
```

Le debe salir algo así por la terminal, además, no debe iniciar ningún análisis antes de que aparezca el mensaje de la Figura [12.1.7](#).



The screenshot shows a terminal window with several tabs at the top: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The terminal content displays the configuration and startup logs for a Celery worker. It includes settings for task events, queues (specifically 'celery' with exchange='celery(direct)' and key='celery'), and tasks from modules 'analisis.tasks' and 'esios.tasks'. It also shows the connection to an AMQP broker and the worker's initial state.

```
-- *----- .> task events: OFF (enable -E to monitor tasks in this worker)
-- *----- [queues]
      .> celery          exchange=celery(direct) key=celery

[tasks]
  . analisis.tasks.procesar_datos_en_segundo_plano
  . esios.tasks.actualizar_datosAutomaticamente

[2024-06-16 20:46:51,976: INFO/MainProcess] Connected to amqp://guest:**@127.0.0.1:5672// 
[2024-06-16 20:46:51,980: INFO/MainProcess] mingle: searching for neighbors
[2024-06-16 20:46:53,005: INFO/MainProcess] mingle: all alone
[2024-06-16 20:46:53,020: INFO/MainProcess] celery@carlos ready.
```

Figura 12.1.7: Celery Worker.

Como con Celery beat, para acabar con el proceso bastará únicamente con cerrar la terminal.

Con estos tres procesos activos podremos ponernos en marcha y realizar tantos análisis como queramos. Accedan a la url que les aparece en la primera terminal para acceder a la aplicación en local.

12.2 Manual de usuario

En esta sección les muestro una visita completa por la aplicación mediante capturas del producto final, realizando un tour indicando todas las funcionalidades que otorga.

12.2.1 Usuario no registrado

Estas vistas serán accesibles para cualquier usuario que acceda a nuestra aplicación.

Inicio:

La primera vista (Figura 12.2.1) tras el despliegue nos muestra la página de inicio con un breve apartado informativo sobre los tipos de análisis que se pueden llevar a cabo mediante nuestra aplicación:



Figura 12.2.1: Vista de Inicio.

Analistas:

En esta vista aparecen los nombres y descripciones de los análisis terminados junto con los analistas encargados de sus respectivos análisis (Figura 12.2.2). Esta vista está planteada como una forma en la que cualquier usuario pueda ponerse en contacto con un analista y así poder consultar los resultados mediante la API. El correo utiliza la funcionalidad mailto, que abre el sistema de mensajería que utilice el ordenador para agilizar el contacto con el analista:

Nombre del análisis	Descripción	Nombre del analista	Apellidos del analista	Email de contacto
prueba	Esto es una descripción de prueba	carlos	nuchera	carmuchol@alum.us.es

Figura 12.2.2: Vista de Analistas.

API:

Pulsando en API (Figura 12.2.3) del menú de navegación, accederemos a la vista de información de la API, en la que se resume la información necesaria para que cualquier usuario acceda a la información de cualquier análisis realizado:

The screenshot shows a web page for 'ESIOS API'. At the top, there's a navigation bar with links for 'Inicio', 'Contactar', 'Analistas', 'API', 'Ingresar', and 'Registrarse'. The main title is 'API de Consulta de Resultados de Análisis'. Below the title, there's a message to the client: 'Estimado cliente,' followed by a detailed description of the API's purpose and security features. A section titled '¿Cómo funciona?' lists various analysis types like 'Autocorrelaciones', 'Detección de Outliers', and 'Descomposiciones de Series Temporales'. It also mentions using a UUID to access results. Below this, a heading 'Documentación de la API' leads to a 'Swagger' interface.

Para consultar los resultados de sus análisis, simplemente introduzca el UUID correspondiente al análisis que desea revisar. Una vez proporcionado el UUID, tendrá acceso a los siguientes tipos de análisis:

- Autocorrelaciones
- Detección de Outliers
- Descomposiciones de Series Temporales

Realice la consulta utilizando el UUID de su análisis y acceda a los resultados de manera rápida y sencilla.

Documentación de la API

Para obtener más información sobre cómo utilizar nuestra API, consulte nuestra documentación:

Swagger
Redoc

Figura 12.2.3: Vista de API.

Los botones de Swagger y Redoc ofrecen la documentación y el sistema de consultas para poder acceder a la información de los análisis. En la imagen de abajo veremos la vista de Swagger (Figura 12.2.4) las consultas que se pueden realizar:

The screenshot shows the 'Swagger' interface for the 'ESIOS API'. The top navigation bar includes links for 'Requirements', 'Responses', 'Webhooks', and 'Todos los marcadores'. The main header is 'ESIOS API' with a 'Base URL: 127.0.0.1:8000/api/' link. Below the header, there's a brief description: 'Aquí podemos realizar todo tipo de consultas, pudiendo visualizar todos los datos de la base de datos para este proyecto'. It includes links for 'Terms of service', 'Contact the developer', and 'BSD License'. On the left, there's a dropdown for 'Schemes' set to 'HTTP'. On the right, there are 'Django Login' and 'Authorize' buttons. The main content area shows a tree view of API endpoints under the 'analysis' category, each with a 'GET' method and a corresponding URL path. For example, 'analisis_list' is associated with '/analisis/'. There are also sections for 'analisis_autocorrelacion_list', 'analisis_descomposicion_de_series_temporales_list', and 'analisis_deteccion_de_outliers_list'.

Figura 12.2.4: Vista de Swagger.

Contacto:

En esta sección aparece la información de contacto del desarrollador (Figura 12.2.5), diseñada para un sistema de mantenimiento de la página:

The screenshot shows a contact page with the following details:

- Nombre:** Carlos Nuchera Bolaños
- Correo:** carmucbol@alum.us.es
- Teléfono:** 669081623
- Repositorio de GitHub:** <https://github.com/CarlosNucheraTFG.git>

On the right side of the page, there is a circular profile picture of a man with dark hair and a beard, wearing a dark t-shirt.

Figura 12.2.5: Vista de Contacto.

11.2.2 Usuario registrado

En este apartado se visualizarán las funcionalidades implementadas para un analista. Omitimos las vistas anteriores además de la vista de Registro y Login que proporciona Django para evitar hacer demasiado extenso este apartado.

Análisis:

En el menú de navegación aparece una pestaña nueva llamada análisis, al hacer clic accedemos al formulario de análisis de la aplicación (Figura 12.2.6).

Rellene el formulario de análisis

Nombre: prueba

Descripción: Esto es una descripción de prueba

Tipos de dato:

demanda
 gas
 generacion
 precio

Frecuencia:

Días

Acepto los términos y condiciones:

Consultar

Analizar

Figura 12.2.6: Vista de Análisis.

Mis Análisis:

En la vista anterior, si el formulario es válido, se nos reenvía automáticamente a la vista de Mis Análisis (Figura 12.2.7) para que el analista pueda navegar libremente por la aplicación mientras que el análisis se procesa en segundo plano:

Nombre	Fecha de Creación	Tipo de Datos Utilizados	Estado
prueba	June 16, 2024, 8:15 p.m.		En proceso

Figura 12.2.7: Vista de Mis Análisis En proceso.

Cuando el análisis termina (Figura 12.2.8), su estado cambia a Terminado, dando acceso a sus resultados:

The screenshot shows a user interface for managing analyses. At the top, there are navigation links: ESIOS, Inicio, Contactar, Analistas, API, Analizar, Mis Análisis, Perfil, and Salir. Below this, a heading says "Análisis de CARNUCBOL:". A red button labeled "Eliminar seleccionados" is visible. A table lists one analysis entry:

Nombre	Fecha de Creación	Tipo de Datos Utilizados	Estado
<input type="checkbox"/> prueba	June 16, 2024, 8:15 p.m.	generacion, gas	Terminado

Figura 12.2.8: Vista de Mis Análisis Terminado.

Resultados:

Al hacer clic sobre el nombre del análisis, accedemos a la vista resultados (Figura 12.2.9), en la que aparece toda la información del análisis, así como los cuadros para realizar los distintos tipos de análisis con los datos preprocesados. Además, los dos botones de abajo sirven para descargar los datos preprocesados de este análisis, así como la previsualización de los resultados de todos los tipos de análisis realizados para su posterior descarga como PDF:

The screenshot shows the results page for the 'prueba' analysis. At the top, there are navigation links: ESIOS, Inicio, Contactar, Analistas, API, Analizar, Mis Análisis, Perfil, and Salir. A green button labeled "Volver ←" is visible. The page title is "Detalles del Análisis". It displays the following information:

- Nombre:** prueba
- Descripción:** Esto es una descripción de prueba
- Frecuencia:** Días
- Fecha de Creación:** June 16, 2024, 8:15 p.m.
- Usuario:** carnucbol
- Tipos de Datos Utilizados:**
 - gas
 - generacion

Below this, there are three plots:

- Autocorrelación:** A line plot showing autocorrelation values from -0.4 to 1.0 across 40 time points.
- Detección de outliers:** A scatter plot showing data points colored by group (red, purple, green) with one point labeled "Outlier".
- Descomposición de series temporales:** A multi-panel plot for CO2 showing seasonal, trend, and residual components from 1960 to 2000.

At the bottom, there are download and preview options:

- Descargue los datos:**
 - CSV**
 - Previsualizar PDF:**
 - PDF**

Figura 12.2.9: Vista Resultados.

Al hacer clic sobre CSV se descarga un fichero CSV con el resultado de los datos preprocesados, como vemos en la Figura 12.2.10.

1	Tipo	Dato	Nombre	Fecha	Precio	Consumo	Porcentaje	Nombre	Analisis
2	gas	GDAES_D+1		2021-01-01 00:00:00+00:00	20.58	,	,	prueba	
3	gas	GDAES_D+1		2021-01-02 00:00:00+00:00	21.56	,	,	prueba	
4	gas	GDAES_D+1		2021-01-03 00:00:00+00:00	23.56	,	,	prueba	
5	gas	GDAES_D+1		2021-01-04 00:00:00+00:00	26.69	,	,	prueba	

Figura 12.2.10: CSV datos preprocesados.

Autocorrelación:

Al seleccionar 'Autocorrelación' o su imagen, aparece el formulario de Autocorrelación, en el que debemos introducir los parámetros que queramos para realizar el estudio (Figura 12.2.11).

The screenshot shows a web-based application interface for 'Autocorrelación'. At the top, there's a navigation bar with links: 'ESIOS' (highlighted), 'Inicio', 'Contactar', 'Analistas', 'API', 'Analizar', 'Mis Análisis', 'Perfil', and 'Salir'. Below the navigation, there's a green 'Volver ←' button. The main area contains several input fields and dropdown menus:

- Titulo:**
- Lag:**
- Tipo:**
- Método:**
- Visualización:**
- Mostrar datos:**

At the bottom left are two buttons: 'Visualizar' and 'Guardar'.

Figura 12.2.11: Vista formulario Autocorrelación.

Una vez completado, al darle al botón de Visualizar, se muestran dos gráficas, ambas dinámicas, debajo del formulario. La gráfica de arriba son los datos preprocesados según el atributo de los datos que deseamos visualizar, en este caso los del consumo, la de abajo son los resultados de autocorrelación para el lag introducido.

En función de los datos introducidos se visualizarán unas gráficas u otras (Figura 12.2.12).



Figura 12.2.12: Vista gráficas Autocorrelación.

Resultados Autocorrelación:

La vista de Autocorrelación, además de ‘Visualizar’ tiene un botón de ‘Guardar’, si lo pulsamos se guardarán todos los resultados del estudio según los parámetros introducidos en el formulario (Figura 12.2.13). En la vista de Resultados se podrían ver los diversos resultados de multitud de estudios diferentes:

The screenshot shows the 'Autocorrelation' study creation form. The fields filled in are:

- Título: prueba autocorrelación
- Lag: 3
- Tipo: Parcial
- Método: Pearson
- Visualización: Lineas
- Mostrar datos: Consumo

At the bottom, there are three buttons: 'Visualizar' (disabled), 'Guardar' (disabled), and 'Consultar resultados' (highlighted in blue). Below these buttons, a message says 'Autocorrelación añadida correctamente'.

Figura 12.2.13: Guardar estudio Autocorrelación.

Tras pulsar el botón de ‘Consultar Resultados’ accedemos a este listado con los resultados del estudio (Figura 12.2.14) junto con la gráfica adjunta (Figura 12.2.15). Se ha adjuntado una de las gráficas de cara a generar el PDF :

The screenshot shows the 'Resultados para el análisis prueba' page. At the top, there is a search bar labeled 'Título' and two buttons: 'Borrar Autocorrelación por Título' (red background) and 'Descargar CSV' (blue background).

Below is a table listing the results:

Título	Dato analizado	Nombre del dato	Estilo	Método de cálculo empleado	Tipo de Autocorrelación	Lag	Valor
prueba autocorrelación	consumo	Hidráulica	lineas	Pearson	parcial	1	0.9395819340889819
prueba autocorrelación	consumo	Hidráulica	lineas	Pearson	parcial	2	-0.029213748625288694
prueba autocorrelación	consumo	Hidráulica	lineas	Pearson	parcial	3	0.20628050780076784
prueba autocorrelación	consumo	Solar fotovoltaica	lineas	Pearson	parcial	1	0.917278758758879
prueba autocorrelación	consumo	Solar fotovoltaica	lineas	Pearson	parcial	2	0.11299823673851139
prueba autocorrelación	consumo	Solar fotovoltaica	lineas	Pearson	parcial	3	0.19801903384167951
prueba autocorrelación	consumo	Nuclear	lineas	Pearson	parcial	1	0.9598962194658471
prueba autocorrelación	consumo	Nuclear	lineas	Pearson	parcial	2	-0.19131921382711692
prueba autocorrelación	consumo	Nuclear	lineas	Pearson	parcial	3	0.09398317075285345
prueba autocorrelación	consumo	Eólica	lineas	Pearson	parcial	1	0.7215801978350423
prueba autocorrelación	consumo	Eólica	lineas	Pearson	parcial	2	-0.1013238649777757
prueba autocorrelación	consumo	Eólica	lineas	Pearson	parcial	3	0.06155624916907637

Figura 12.2.14: Vista Resultados Autocorrelación estudio.

IMÁGENES:

Figura 1 de Autocorrelación

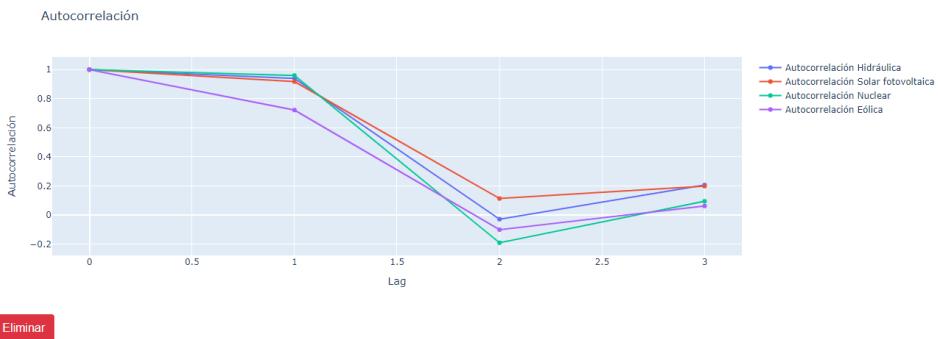


Figura 12.2.15: Vista Resultados Autocorrelación gráficas adjuntas.

Como vemos en las imágenes de arriba, se pueden eliminar los distintos estudios realizados introduciendo el título del estudio o la imagen pulsando sobre el botón de 'Eliminar' a su izquierda. También podremos descargar los resultados que aparecen en la tabla en formato CSV por si el analista los requiere con el formato de la Figura [12.2.16](#).

```
1 |título,Dato analizado,Nombre del dato,Estilo,Método de cálculo empleado,Tipo de Autocorrelación,Lag,Valor
2 prueba autocorrelación,consumo,Hidráulica,lineas,Pearson,parcial,1,0.9395819340889819
3 prueba autocorrelación,consumo,Hidráulica,lineas,Pearson,parcial,2,-0.029213748625288694
4 prueba autocorrelación,consumo,Hidráulica,lineas,Pearson,parcial,3,0.20628050780076784
```

Figura 12.2.16: CSV resultados estudios Autocorrelación.

Detección de Outliers:

Si regresamos a la vista Resultados y accedemos al siguiente tipo de análisis, podremos ver el formulario para la Detección de Outliers (Figura 12.2.17). Para no hacer tedioso este recorrido, en la figura de abajo mostramos el formulario junto con la gráfica generada si pulsamos ‘Visualizar’:

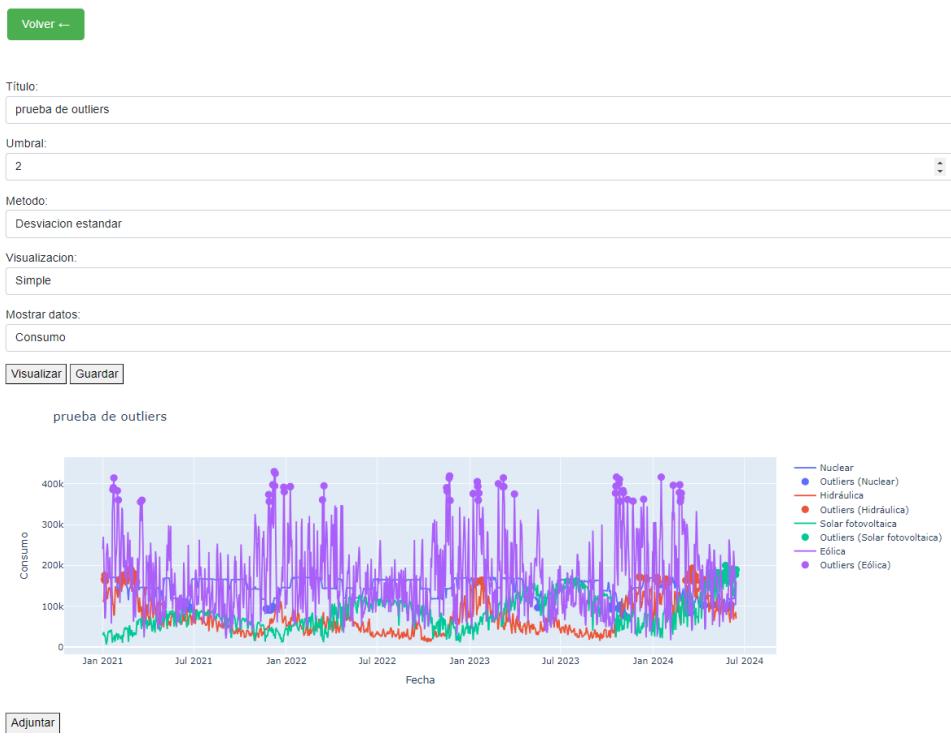


Figura 12.2.17: Formulario y visualizar Detección de Outliers.

Descomposición de Series Temporales:

Al regresar de nuevo a Resultados, podremos acceder al último tipo de análisis, la descomposición de series temporales, en la que podremos llenar el siguiente formulario (Figura 12.2.18) y visualizar sus resultados (Figuras 12.2.19 y 12.2.20).

ESIOS Inicio Contactar Analistas API Analizar Mis Análisis Perfil Salir

Volver ←

Título: prueba descomposición

Método: Multiplicativa

Visualización: Simple

Datos a Mostrar: Consumo

Ventana de Suavizado de Tendencia: 12

Ventana de Suavizado de Estacionalidad: 10

Factor de Suavizado Exponencial: ,7

Fecha de Inicio: 01/01/2024

Fecha de Fin: 14/06/2024

Visualizar Guardar

Figura 12.2.18: Vista formulario Descomposición de Series Temporales.

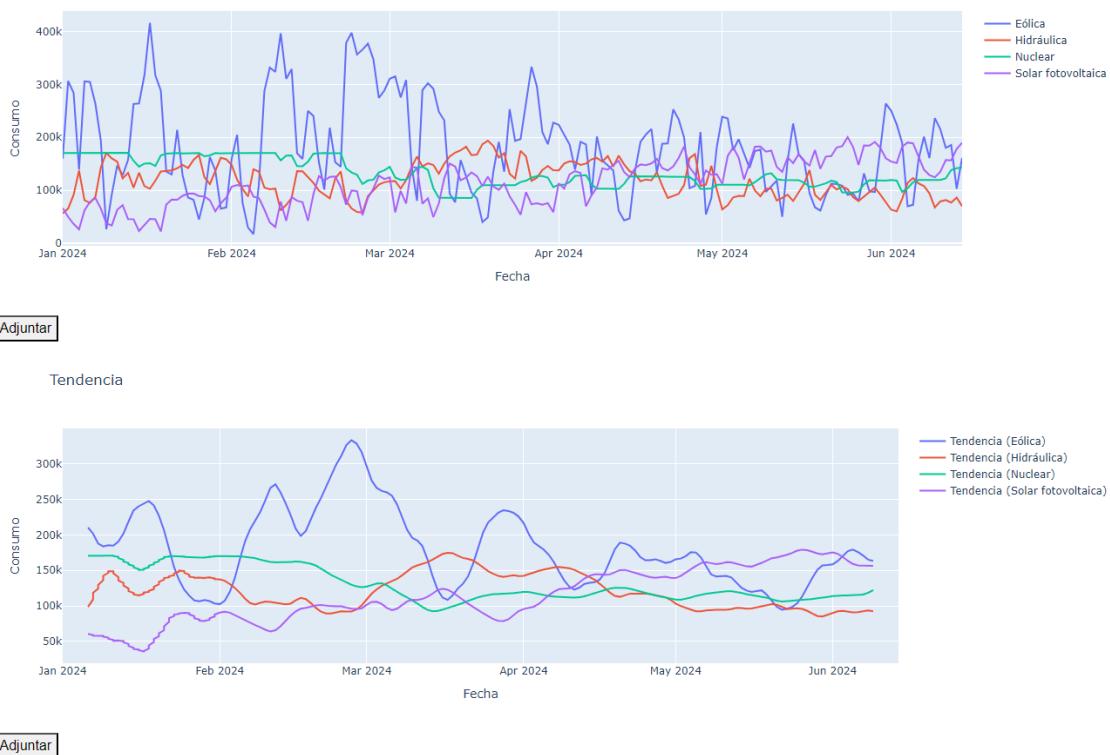
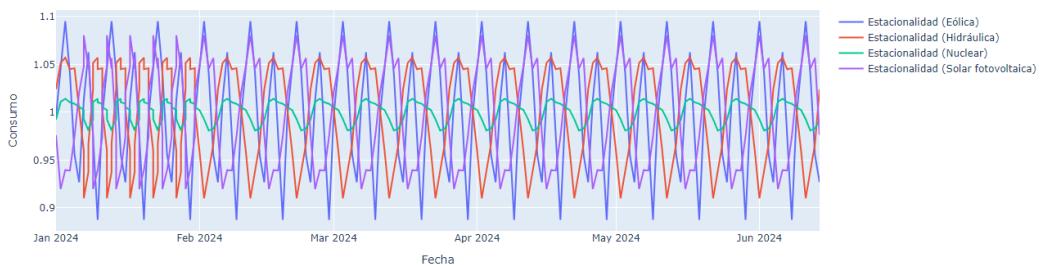


Figura 12.2.19: Visualizar Descomposición de Series Temporales-1.



Residuo

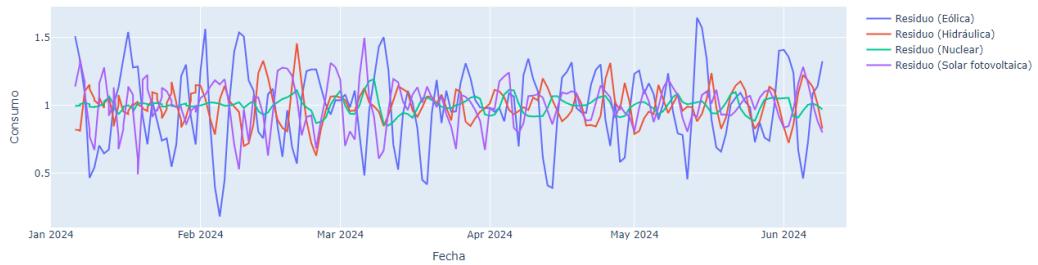


Figura 12.2.20: Visualizar Descomposición de Series Temporales-2.

Resultados Detección de Outliers y Resultados de Descomposición de Series Temporales:

El funcionamiento de estas vistas son idénticos al de Resultados de Autocorrelación por lo que se mostrarán en el apartado de pre visualización del PDF.

Previsualizar PDF:

Para descargar los resultados de los múltiples estudios realizados a lo largo de nuestro recorrido, podremos pulsar en el botón ‘PDF’ de la vista de Resultados, que nos dará acceso a la pre visualización sobre la información que se mostrará en el PDF final.

No olvidemos que podremos regresar en cualquier momento a las vistas de resultados para eliminar o agregar las gráficas o estudios que queramos en nuestro documento (Figura 12.2.21).

The screenshot shows the ESIOS software interface with the following details:

- Header:** ESIOS, Inicio, Contactar, Analistas, API, Analizar, Mis Análisis, Perfil, Salir.
- Buttons:** Descargar como PDF (highlighted in blue).
- Title:** Resultados para el análisis prueba:
- Section:** Resultados de la Autocorrelación:
- Table:** A table listing autocorrelation results for various data types (consumo, hidráulica, solar fotovoltaica, nuclear, eólica) across different lags (1, 2, 3). The table includes columns for Title, Dato analizado, Nombre del dato, Estilo, Método de cálculo empleado, Tipo de Autocorrelación, Lag, and Valor.

GRÁFICAS DE AUTOCORRELACIÓN:

Figura 1 de Autocorrelación

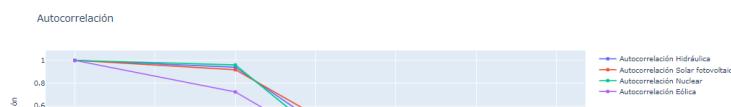


Figura 12.2.21: Vista previsualizar PDF.

Por desgracia no ha sido posible implementar el requisito extra sobre adjuntar las gráficas durante la Fase 4 del desarrollo del proyecto, por lo que el PDF queda representado tal y como se muestra en la Figura 12.2.22.

The screenshot shows the ESIOS software interface with the following details:

- Section:** Resumen del análisis prueba:
- Section:** Resultados de la Autocorrelación:
- Table:** A table listing autocorrelation results for various data types (consumo, hidráulica, solar fotovoltaica, nuclear, eólica) across different lags (1, 2, 3). The table includes columns for Title, Dato analizado, Nombre del dato, Estilo, Método de cálculo empleado, Tipo de Autocorrelación, Lag, and Valor.
- Section:** GRÁFICAS DE AUTOCORRELACIÓN:
- Section:** Resultados de la Detección de Outliers:
- Table:** A table listing outlier detection results for consumption data using the Nuclear method. The table includes columns for Title, Dato analizado, Nombre del dato, Método de cálculo empleado, Umbral, Fecha, and Valor.

Figura 12.2.22: Resultados de análisis PDF.

12.3 Pruebas

Para el proyecto se han diseñado un conjunto de pruebas, que podremos localizar dentro de la app de análisis. Encontraremos dos clases de tests diferentes, las clases test.py y test_selenium.py. El test de selenium corre gracias a un driver de Chrome, que podremos encontrar dentro de la carpeta driver en el directorio raíz del proyecto. Es muy importante que esté instalado este navegador para su correcta ejecución. Procederé a explicar en qué consisten los tests que se han diseñado para esta aplicación:

- Test.py: Aquí encontraremos un total de diez tests unitarios muy sencillos para probar la creación de objetos de nuestros modelos relacionados con el análisis de datos.
- Test_selenium.py: Se trata del test más importante, realiza un recorrido total del sistema. Primero registra a un nuevo usuario que procede a realizar un análisis con los datos del gas y las generaciones (eólica, solar, ...), que se consideran las más interesantes por la variedad de datos y la rapidez del preprocesamiento para optimizar el tiempo de ejecución del test.

Una vez preprocesados los datos, se realizan los tres tipos de análisis y se comprueba que funcione correctamente el guardado del estudio así como el proceso de adjuntar una gráfica asociada al mismo. Además, se comprueba que funcione correctamente la eliminación de resultados por título.

Una vez terminado todo este proceso se comprueba la correcta eliminación del análisis y la eliminación total del usuario registrado accediendo a su perfil.

Para ejecutar las pruebas debemos acceder a una terminal con la ubicación del proyecto raíz donde se encuentra el archivo *manage.py*. Acto seguido deben escribir el siguiente comando:

```
python manage.py test #Comando para ejecutar todas las pruebas
```

13. Cumplimiento de objetivos

En este capítulo se detalla cómo ha sido el cumplimiento de objetivos que se definieron en el apartado [3](#) del presente documento.

Objetivo 1: Desarrollo de una Aplicación Web

Gracias al framework de Django se ha logrado satisfacer todos los requisitos del proyecto y proporcionar un producto satisfactorio.

Objetivo 2: Obtención de Datos

Se han obtenido los datos de fuentes fiables tales como ESIOS, MIBGAS y REE mediante una serie de scripts de descarga que hacen uso de un token de acceso solicitado por el Jefe de Proyectos al inicio de la Fase 1.

Objetivo 3: Realización de Análisis

Mediante nuestra aplicación, podremos seleccionar los datos sobre los cuales queramos hacer un análisis previamente guardados y posteriormente preprocesados, ambas funcionalidades desarrolladas durante la Fase 2 del proyecto. De esta forma podremos realizar tres tipos de análisis diferentes: Autocorrelación, Detección de Outliers y Descomposición de Series Temporales desarrollados durante la Fase 3.

Objetivo 4: Resultados de Análisis

Para cada tipo de análisis se podrá acceder a una vista con los resultados de todos los estudios realizados, además podremos previsualizar y descargar un PDF con los resultados de todos los análisis realizados.

Objetivo 5: Visualización de Resultados

Los resultados de los análisis se podrán visualizar gracias a gráficas interactivas diseñadas con Ploty, en las que el usuario podrá seleccionar los datos o resultados que desee, así como descargarlas en formato PNG.

14. Conclusiones

En este Trabajo de Fin de Grado se ha logrado alcanzar el mínimo producto viable con éxito, desarrollando una aplicación web funcional para el análisis de datos sobre la red eléctrica española a nivel provincial. Esta aplicación utiliza datos obtenidos de ESIOS, MIBGAS y REE.

La aplicación permite realizar varios tipos de análisis, incluyendo autocorrelación, detección de outliers y descomposición de series temporales. Los usuarios registrados podrán consultar los resultados, seleccionar los que les interesen y descargarlos en formato CSV para su uso en otros estudios, así como en formato PDF.

Además, la aplicación busca facilitar la interacción entre los usuarios registrados (analistas) y los usuarios no registrados a través de la API. Cualquier usuario podría ver los resultados de los análisis realizados por los analistas si estos deciden compartirlos, promoviendo así la colaboración y el intercambio de información con fines de investigación.

A lo largo de este proyecto, he tenido la oportunidad de aplicar de manera práctica todos los conocimientos adquiridos durante la carrera de Ingeniería Informática del Software. Este proceso me ha permitido profundizar en diversas áreas, desde el desarrollo web hasta el análisis de datos y la implementación de algoritmos avanzados. La experiencia ha sido enormemente enriquecedora y me ha proporcionado una comprensión más completa y aplicada de los conceptos aprendidos, demostrando la capacidad de integrar y utilizar de manera efectiva estas habilidades en un proyecto real.

14.1. Lecciones aprendidas

Durante el desarrollo de este Trabajo de Fin de Grado, he tenido la oportunidad de aprender y crecer en varios aspectos técnicos y de gestión de proyectos. A continuación, detallaré las principales conclusiones que he sacado de este proyecto:

Conocimiento en Django: A lo largo del proyecto y sobre todo en las últimas etapas de desarrollo me han demostrado lo mucho que he aprendido, ganando mucha fluidez a la hora de desarrollar y aprender nuevos conceptos. Estoy seguro que podré abordar futuros proyectos con mayor confianza y eficiencia.

Problemas con Celery: Uno de los desafíos más significativos que enfrenté fue la implementación de tareas asíncronas utilizando Celery en un entorno Windows, perdí mucho tiempo solucionando los problemas, de hecho, al principio de la tarea traté de utilizar Celery con Redis, pero ocurrió un fallo que no logré solucionar, obligandome en cierta forma a cambiar a RabbitMQ. Se presentaron varias complicaciones, especialmente en la configuración y gestión de los workers de Celery. A pesar de estos problemas, logré superar las dificultades y completar la integración de Celery, lo que me proporcionó una valiosa experiencia en la resolución de problemas y en la adaptación de soluciones con el uso de tecnologías diferentes.

Resolución de tareas: Este proyecto me exigió aprender y dominar muchas tareas nuevas. Muchas de ellas probablemente podría haberlas realizado con más soltura de haber tenido más claros los conceptos, por lo que resulta inevitable darse cuenta de que la planificación es muy importante y hay que prestarle mucha atención para que después no surjan complicaciones.

Holguras durante la planificación: Una lección crucial aprendida ha sido la importancia de una planificación más precisa y con mayores holguras para futuros proyectos. En varias ocasiones, los plazos planificados inicialmente resultaron ser insuficientes debido a imprevistos y la complejidad de ciertas tareas. Esta experiencia me ha enseñado a incluir márgenes de tiempo más amplios y a ser más flexible y realista en la gestión del tiempo y los recursos, lo cual será vital para la planificación de futuros proyectos.

Centrarse en el objetivo: Durante el desarrollo, también aprendí la importancia de centrarse directamente en los objetivos y requisitos del proyecto sin intentar hacerlos más complejos de lo necesario. En varias ocasiones, me encontré tentado a agregar funcionalidades adicionales o a complicar los requisitos más allá de lo especificado.

15. Trabajo futuro

El proyecto desarrollado presenta una base sólida y funcional, pero existen múltiples áreas donde se puede expandir y mejorar la aplicación para proporcionar un mayor valor a los usuarios. A continuación, se detallan algunas de las posibles mejoras y extensiones futuras:

Agregar las gráficas de los estudios al PDF: Actualmente, los resultados de los análisis pueden descargarse en formato PDF y CSV. Una mejora importante sería incluir las gráficas generadas a partir de los estudios directamente en los archivos PDF, proporcionando así una presentación visual más completa y útil de los datos analizados.

Integrar algoritmos de Machine Learning: La idea es que a futuro se puedan implementar algoritmos predictivos avanzados con machine learning.

Incluir más tipos de análisis: La adición de nuevos tipos de análisis avanzados enriquecería la funcionalidad de la aplicación. Esto podría incluir análisis predictivos, clustering, y análisis de correlación cruzada, entre otros. Estos nuevos tipos de análisis permitirían a los usuarios obtener una comprensión más profunda y diversa de los datos de la red eléctrica.

Trabajar con más tipos de datos: Ampliar la aplicación para trabajar con diferentes tipos de datos, no solo los obtenidos de ESIOS, MIBGAS y REE, sino también incluir otros conjuntos de datos relevantes, como datos meteorológicos o datos de consumo energético, podría proporcionar análisis más completos y contextuales.

Informes por correo: Implementar una funcionalidad que notifique a los analistas por correo electrónico cuando sus análisis hayan concluido. Esto mejoraría la usabilidad y

permitiría a los analistas estar al tanto del progreso de sus tareas sin necesidad de verificar constantemente la aplicación.

Desplegar en Apache: Migrar la aplicación para su despliegue en un servidor Apache podría mejorar la escalabilidad y la capacidad de manejo de tráfico, asegurando un rendimiento óptimo incluso bajo alta demanda.

Sonar Cloud: Integrar SonarCloud en el proceso de desarrollo para realizar análisis continuos de la calidad del código. Esto ayudaría a identificar y corregir problemas de manera proactiva, manteniendo altos estándares de calidad y evitando la acumulación de deuda técnica.

Soporte técnico: Establecer un sistema de soporte técnico para ayudar a los usuarios con problemas o preguntas sobre la aplicación. Esto podría incluir una base de conocimientos, un sistema de tickets y asistencia en tiempo real, mejorando la experiencia del usuario y la satisfacción general.

Mejorar el frontend: Realizar mejoras en el diseño y la funcionalidad del frontend para hacerlo más intuitivo y atractivo. Esto podría incluir la actualización del framework de frontend, la mejora de la experiencia del usuario (UX) y la implementación de un diseño responsive que funcione bien en dispositivos móviles.

Mejorar las validaciones: Migrar las validaciones actuales para que se realicen en el cliente utilizando JavaScript, en lugar de depender de mensajes del servidor. Esto proporciona una retroalimentación más rápida y mejoraría la experiencia del usuario al reducir el tiempo de espera y la carga en el servidor.

Incluir más testing: Incrementar la cobertura de testing tanto para el backend como para el frontend. Esto incluiría pruebas unitarias, pruebas de integración y pruebas de extremo a extremo para asegurar que todas las funcionalidades de la aplicación funcionan correctamente y para identificar cualquier problema antes de que afecte a los usuarios.

Séptima parte

VII-APÉNDICE

16. Glosario

Análisis de Autocorrelación: Consultar el apartado [7.1.1](#)

Análisis de Descomposición de Series Temporales: Consultar el apartado [7.1.2](#)

Análisis de Detección de Outliers: Consultar el apartado [7.1.3](#)

API: Una API (Interfaz de Programación de Aplicaciones) es un conjunto de reglas y definiciones que permite que diferentes programas de software se comuniquen entre sí.

Preprocesamiento: El preprocesamiento es una fase crucial en el ciclo de vida del desarrollo de software, especialmente en proyectos que involucran el manejo de datos. Por norma general, consiste en preparar y limpiar los datos antes de su uso en el modelo o aplicación. Esto incluye tareas como la eliminación de valores atípicos, la normalización de datos, la conversión de formatos y el manejo de valores faltantes. Para nuestro caso, nos encontramos con unos datos de entrada ya filtrados, por lo que únicamente debemos generar unos datos de salida según el conjunto específico de datos y la frecuencia que el usuario haya seleccionado en la fase inicial del análisis.

Analista: Cuando hacemos referencia a un analista, nos referimos a un usuario registrado en el sistema.

Estudio: Al realizar cualquier tipo de análisis, el estudio hace referencia al conjunto de parámetros y datos seleccionados en el formulario sobre un tipo de análisis ‘guardado’, en otras palabras, del que se almacenan los resultados que el analista desea guardar..

Fase: Consultar el apartado [6](#).

17. Bibliografía

La bibliografía se mostrará según se estipula en formato APA 2009.

[1] ESIOS. (s.f). Red Eléctrica de España. Recuperado de
<https://www.esios.ree.es/es>

[2] Jose Manuel Moreno. (2024). ¿Qué tendencias en el sector energético veremos este 2024? Recuperado de
<https://cuervaenergia.com/es/comunidad/innovacion/tendencias-energeticas-para-este-2024/>

[3] REE. (s.f). Red Eléctrica de España. Recuperado de <https://www.ree.es/es>

[4] MIBGAS. (s.f.). MIBGAS S.A. Recuperado de <https://www.mibgas.es/>

[5] Notebook de uso de apis. (s.f). Recuperado de
<https://colab.research.google.com/drive/1w8peG-9E9VP2oky1ij0Q4pSqbeB7Uz>

[6] InteractiveChaos. (s.f). Preprocesamiento de datos: Recuperado de
<https://interactivechaos.com/es/manual/tutorial-de-machine-learning/preprocesamiento-de-datos>

[7] J.M. Arranz , M.M. Zamora. (s.f). Análisis de Autocorrelación. Recuperado de
<http://ciberconta.unizar.es/Leccion/autocorrelacion/analisis%20de%20autocorrelacion.PDF>

[8] DATatab. (s.f). Correlación de Pearson. Recuperado de
<https://datatab.es/tutorial/pearson-correlation>

[9] DATatab. (s.f). Correlación de Spearman. Recuperado de
<https://datatab.es/tutorial/spearman-correlation>

[10] Paola Rosmery Sanchez Reyes. (2021). Descomposición de Series Temporales (Modelo Aditivo y Multiplicativo Clásicos). Recuperado de
https://rpubs.com/Paolareyes/series_temporales

[11] Juan Francisco Islas. Miguel Heras. (2012). Descomposición de Series de Tiempo. Recuperado de
<https://seriesdetiempo.wordpress.com/wp-content/uploads/2012/08/descomposicic3b3n-de-series-de-tiempo-base-cap-3-makridakis-et-al-fe-ago-2012.pdf>

[12] Victor Yepes Piqueras. (s.f). ¿Qué hacemos con los valores atípicos (outliers)? Recuperado de
[https://victoryepes.blogs.upv.es/2022/02/21/que-hacemos-con-los-valores-atipicos-outliers/#:_text=Un%20valor%20at%C3%ADpico%20\(outlier%2C%20en.proceso%20o%20en%20una%20poblaci%C3%B3n.](https://victoryepes.blogs.upv.es/2022/02/21/que-hacemos-con-los-valores-atipicos-outliers/#:_text=Un%20valor%20at%C3%ADpico%20(outlier%2C%20en.proceso%20o%20en%20una%20poblaci%C3%B3n.)

[13] Khan Academy. (s.f). What are z-scores?. Recuperado de <https://www.khanacademy.org/math/statistics-probability/modeling-distributions-of-data/z-scores/a/z-scores-review>

[14] Khan Academy. (s.f). Interquartile Range (IQR). Recuperado de <https://www.khanacademy.org/math/statistics-probability/summarizing-quantitative-data/interquartile-range-igr/a/interquartile-range-review>

[15] EspiFreelancer. (s.f). Que es el patrón MTV (Model Template View). Recuperado de <https://espifreelancer.com/mtv-django.html>

[16] Tom Christie and contributors. (2021). Django Rest Framework Documentation. Django REST framework. Recuperado de <https://www.django-rest-framework.org/>.

[17] drf-yasg. (s.f). Recuperado de <https://drf-yasg.readthedocs.io/en/stable/>

[18] Celery 5.4.0 documentation. (s.f). First Steps with Celery. <https://docs.celeryq.dev/en/stable/getting-started/first-steps-with-celery.html#installing-celery>

[19] RabbitMq. (s.f). Installing on Windows. Recuperado de <https://www.rabbitmq.com/docs/install-windows>

[20] Erlang. (s.f). Download Erlang/OTP. Recuperado de <https://www.erlang.org/downloads>

[21] Django Software Foundation. (2023). Django web framework (Version 5.0.4) [Software]. Recuperado de <https://www.djangoproject.com/>

[22] django-crispy-forms. (s.f). Forms have never been this crispy. Recuperado de <https://django-crispy-forms.readthedocs.io/en/latest/>

[23] Jacob Thornton Mark Otto. (s.f). Bootstrap. Recuperado de <https://getbootstrap.com/>

[24] PostgreSQL Global Development. (s.f). Windows installers. Recuperado de <https://www.postgresql.org/download/windows/>

[25] Plotly Technologies Inc. (2015). Open-source graphing library for Python (Version 5.22.0) [Software]. Recuperado de <https://plotly.com/python/>

[26] Hunter, J. D. (2007). matplotlib: A 2D graphics environment (Version 3.8.4) [Software]. Recuperado de <https://matplotlib.org/>

[27] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2023). numpy. Recuperado de <https://numpy.org/>

[28] Pandas: McKinney, W., & others. (2010). Powerful data structures for data analysis, time series, and statistics (Version 2.2.2) [Software]. Recuperado de <https://pandas.pydata.org/>

[29] Requests Development Team. (2022). requests: HTTP for humans (Version 2.31.0) [Software]. Recuperado de <https://docs.python-requests.org/>

[30] Talent.com. (s.f). Salaries for Django programmers. Recuperado de <https://es.talent.com/salary?job=programador+django>

[31] Jorge Morales Fernández. (2024). ¿Cuánto cuesta el kilovatio hora de luz (kWh) en España?. Recuperado de <https://tarifaluzhora.es/info/precio-kwh>

[32] Rob J Hyndman. George Athanasopoulos. (2024). Forecasting: Principles and Practice. Recuperado de <https://otexts.com/fpp3/index.html>