



UNIVERSIDAD DE MÁLAGA



E.T.S. INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

Realizado por Antonio Santiago Díaz

Tutorizado por Daniel Garrido Márquez y Enrique Soler
Castillo

Departamento de Lenguajes y Ciencias de la Computación

MÁLAGA, Junio de 2022



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA DE SOFTWARE

**Aplicación web para la Gestión y Monitorización de
Valores Energéticos Recogidos por Contadores
Inteligentes**

Control en tiempo real del consumo de energía en la página web.

**Web Application for the Management and Monitoring of
Energy Values Collected by Smart Meters.**

**Real-time monitoring of energy consumption in the web management
energy system.**

Realizado por
Antonio Santiago Díaz

Tutorizado por
Daniel Garrido Márquez
Enrique Soler Castillo

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JUNIO 2022



UNIVERSIDAD
DE MÁLAGA

Resumen

El objetivo principal de este Trabajo de Fin de Grado es el desarrollo de una aplicación web, la cual, está incluida dentro de un proyecto más grande llamado "Contadores inteligentes y accesibles para el hogar", el cual forma parte de la iniciativa K-Project de la Universidad de Málaga. La idea fundamental es que cualquier usuario que tenga un contador inteligente diseñado en el proyecto pueda usar esta aplicación para ver el registro histórico de medida que ha recogido el contador inteligente.

La principal funcionalidad consiste en que un usuario que ha iniciado sesión en el sistema puede ver todos los hogares que tiene asociado así mismo (o que alguien le haya compartido). Dentro de esos hogares habrá una serie de contadores inteligentes que irán tomando medidas durante un cierto tiempo. El usuario en un momento dado puede acceder a uno de esos contadores inteligentes para poder ver las estadísticas históricas que ha recogido, ya sea kilovatios consumidos en un tramo de tiempo, una gráfica de consumo de kilovatios hora en tiempo acotado, etc.

La motivación de este trabajo y por ende de este proyecto es que durante el último año hemos notado todos los que vivimos en este país cómo nuestra factura de la luz aumentaba considerablemente y continuamente. Aunque este proyecto no consigue que ese precio desaparezca o a minore, lo que hace es ofrecer un control al usuario sobre la energía que gasta en casa para que pueda conforme a esos gasto actuar para que esa factura no sea tan descomunal.

Cabe destacar que este trabajo forma parte de un TFG grupal y por tanto tienes unas partes conjuntas y algunas partes específicas de este trabajo que se especifican en la introducción.

Palabras clave: Aplicación web, contador inteligente, energía, electricidad, consumo

Abstract

The main objective of this Final Degree Project is the development of a web application, which is included in a larger project called "Smart and accessible meters for the home", which is part of the K-Project initiative of the University of Malaga. The basic idea is that any user who has a smart meter designed in the project can use this application to view the historical metering record collected by the smart meter.

The main functionality is that a user who is logged into the system can see all the households that he/she has associated with him/herself (or that someone else has shared with him/her). Within these households there will be a series of smart meters that will take measurements over a certain period of time. The user at any given time can access one of these smart meters to see the historical statistics that it has collected, whether it be kilowatts consumed over a period of time, a graph of kilowatt hour consumption over a given time, etc.

The motivation for this work and therefore for this project is that during the last year all of us who live in this country have noticed how our electricity bill has risen considerably and continuously. Although this project does not make this price disappear or decrease, what it does do is to offer users control over the energy they spend at home so that they can act in accordance with these expenses so that the bill is not so enormous.

It should be noted that this work is part of a group TFG and therefore has some joint parts and some specific parts of this work that are specified in the introduction.

Keywords: Web application, smart meter, power, electricity, consumption

Índice

Resumen	1
Abstract	1
Índice	1
Índice de Figuras	1
Introducción.....	1
1.1 Motivación	1
1.2 Objetivos	4
1.3 Estructura de la memoria	5
Metodología.....	7
Tecnologías usadas.....	9
3.1. Python.....	9
3.2. Django	10
3.3. MongoDB	10
3.4 MongoDB Atlas	11
3.4. JavaScript	11
3.5. React	11
3.6. Bootstrap.....	12
3.7. Visual Studio Code	12
3.8. Git	13
3.9. MagicDraw	14
3.10. JWT	14
3.11. Swagger	14
3.12. Heroku.....	15
3.13. IFML	15
3.14. HTTP y HTTPS.....	16
3.15. JSON	17
Análisis del sistema	19
4.1. Descripción general del sistema.....	19
4.2. Requisitos funcionales.....	20
4.3. Requisitos no funcionales.....	22
Especificación.....	23
5.1. Modelo entidad-relación	23
5.2. Dependencias	24
5.3. Diagrama de Casos de Uso.....	26
5.4. Especificación de Casos de uso	30
5.5. Diagramas de secuencia	62
Diseño.....	71
6.1. Arquitectura general de la aplicación	71

6.2. Diseño de la base de datos	72
6.3. Módulo backend	73
6.4. Modulo frontend	75
Implementación	79
7.1. Modulo backend	79
7.2. Modulo frontend	85
7.3. Ejemplo de pasos al completo	88
Pruebas	91
Conclusiones y líneas futuras	111
Referencias	115
Manual de Instalación	117
A.1.Requerimientos	117
A.2.Instalación Backend	117
A.3.Instalación Frontend	119
Manual de usuario	121

Índice de Figuras

Figura 1: Imagen de todos los trabajos del proyecto	1
Figura 2: Logo de Python	9
Figura 3: Logo de django	10
Figura 4: Logo de MongoDB	10
Figura 5: Logo de MongoDB Atlas	11
Figura 6: Logo de JavaScript	11
Figura 7: Logo de React	12
Figura 8: Logo de Bootstrap	12
Figura 9: Logo de Visual Studio Code	13
Figura 10: Logo de Git.....	13
Figura 11: Logo de GitHub	13
Figura 12: Logo de MagicDraw	14
Figura 13: Logo JWT.....	14
Figura 14: Logo Swagger.....	15
Figura 15: Logo de Heroku.....	15
Figura 16: Logo IFML.....	15
Figura 17: Formato objeto JSON.....	17
Figura 18: Formato Array JSON	17
Figura 19: Modelo entidad-relación	23
Figura 20: Diagrama de secuencia creación de un contador inteligente	25
Figura 21: Diagrama de casos de uso de la gestión de usuarios	26
Figura 22: Diagrama de casos de uso de la gestión de hogares	26
Figura 23: Diagrama de casos de uso gestión de invitaciones al hogar	27
Figura 24: Diagrama de casos de uso del registro de contadores inteligentes.....	27
Figura 25: Diagrama de casos de uso de la gestión de contadores inteligentes	27
Figura 26: Diagrama de casos de uso de la visualización de información de contadores inteligentes	28
Figura 27: Diagrama de casos de uso del filtro de información de contadores inteligentes	28
Figura 28: Diagrama de casos de uso de la visualización del coste de la energía.....	28
Figura 29: Diagrama de casos de uso de gestión de notificaciones	29
Figura 30: Diagrama de secuencia RF1.....	62
Figura 31: Diagrama de secuencia RF4.....	63
Figura 32: Diagrama de secuencia RF6.....	64
Figura 33: Diagrama de secuencia RF8.....	65
Figura 34: Diagrama de secuencia RF12.....	66
Figura 35: Diagrama de secuencia RF16.....	66
Figura 36: Diagrama de secuencia RF22.....	67
Figura 37: Diagrama de secuencia RF27.....	68
Figura 38: Diagrama de secuencia RF28.....	69

Figura 39: Diagrama de secuencia RF32	70
Figura 40: Arquitectura general de la aplicación.....	71
Figura 41: Descripción de las colecciones en Base de datos	72
Figura 42: Primera parte de la API.....	73
Figura 43: Segunda parte de la API.....	74
Figura 44: Tercera parte de la API	74
Figura 45: Diagrama de interacción 1ª parte	75
Figura 46: Diagrama de interacción parte 2	76
Figura 47: Diagrama de interacción parte 3	77
Figura 48: Diagrama de interacción parte 4	78
Figura 49: Ejemplo de clases en models.py.....	79
Figura 50: Ejemplo DTO para obtener hogares sin dispositivos.....	80
Figura 51: Ejemplo DTO para obtener un hogar solo con toda su información	80
Figura 52: Serializer para devolver un usuario	81
Figura 53: Serializer para modificar los datos de un usuario	81
Figura 54: Ejemplo de función Get	82
Figura 55: Ejemplo de función Post.....	83
Figura 56: Fichero urls.py	83
Figura 57: Líneas para la configuración del correo	84
Figura 58: Foto función ejemplo que envía correo	84
Figura 59: Código de cómo se crea y envía el token del dispositivo	85
Figura 60: Código para autorizar un dispositivo.....	85
Figura 61: App.js	86
Figura 62: Carpeta components	86
Figura 63: Carpeta web.....	87
Figura 64: Ejemplo paso 1	87
Figura 65: Ejemplo paso 2	87
Figura 66: Ejemplo paso 3	88
Figura 67: Ejemplo de useEffect ejemplo completo	88
Figura 68: Foto paso 1 instalación Backend	117
Figura 69: Foto paso 2 instalación Backend	118
Figura 70: Foto paso 4 instalación Backend	118
Figura 71: Foto del paso 5 instalación Backend	118
Figura 72: Foto paso 5 instalación Backend	119
Figura 73: Foto paso 6 opcional de instalación Backend	119
Figura 74: Foto paso 4 instalación Frontend	119
Figura 75: Foto paso 5 instalación Frontend	120
Figura 76: Pantalla de login	121
Figura 77: Opciones de la parte superior derecha	121
Figura 78: Ventana emergente de mis invitaciones	122
Figura 79: Desplegable Hola,	122
Figura 80: Ventana de mi perfil	122
Figura 81: Desplegable "Hogares"	123
Figura 82: Ventana registrar hogar.....	123
Figura 83: Hogar del que eres dueño	124
Figura 84: Editar un hogar	124
Figura 85: Ventana de usuarios compartidos.....	124

Figura 86: Ventana para compartir	125
Figura 87: Hogar cuando soy un invitado	125
Figura 88: Imagen pestaña comparar días	125
Figura 89: Imagen ventana emergente comparación de días.....	126
Figura 90: Lista de opciones valores de energía.....	126
Figura 91: Pantalla tarifa de las empresas.....	126
Figura 92: Pantalla para añadir una tarifa	127
Figura 93: Ventana ejemplo en cuanto a la media de €/KWh	127

Introducción

1.1 Motivación

K-Project [1] es una iniciativa que surge del I Plan Propio Integral de Docencia de la Universidad de Málaga para mejorar la empleabilidad, innovación y emprendimiento, gracias a equipos de trabajos tutorizados para el desarrollo de soluciones en el mundo real.

Cabe destacar que este trabajo se está haciendo dentro de K-Project denominado "Contadores inteligentes y accesibles para el hogar" donde existen otros trabajos con los cuales se va a interaccionar como puede ser el contador inteligente, una aplicación móvil, una API REST para obtener los valores de energía, etc. A continuación, se muestra un diagrama de los diferentes trabajos que hay dentro del proyecto:

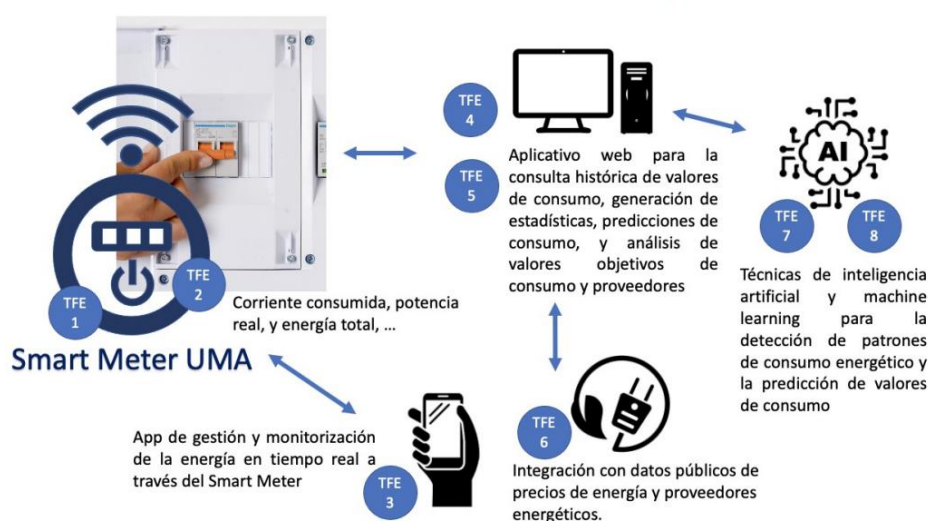


Figura 1: Imagen de todos los trabajos del proyecto

En la figura 1 se explican los distintos trabajos que existen:

- TFE 1 y 2: estos trabajos están relacionados con la realización de un prototipo hardware de los contadores inteligentes.
- TFE 3: este trabajo está relacionado con la realización de una aplicación móvil que interactúa tanto con el contador como con la aplicación web.
- TFE 4 y 5: estos trabajos están relacionados con la realización de una página web. Los dos trabajos forman el trabajo grupal y uno de ellos comprende este trabajo específico.
- TFE 6: este trabajo consiste en realizar una integración con los datos de los precios públicos.
- TFE 7 y TFE 8: estos trabajos consisten en realizar una IA capaz de predecir tanto el consumo (uno de los trabajos) como el precio (otro de los trabajos).

En esta parte del K-Project nos centramos en la creación de un sistema para la monitorización de los contadores inteligentes. Durante los últimos meses vivimos en una situación donde el precio de la luz ha crecido de manera muy considerable, con un aumento en la luz del hogar de un 52%, algo inasumible por muchas de las familias españolas [2]. Para poder reducir la factura de la luz se puede hacer de 2 formas:

- Primera opción: reducir el precio de la luz. Algo que por ahora no se ha conseguido y que además no está al alcance de nuestra mano.
- Segunda opción: reducir el consumo que tenemos. Con esta opción la factura notaría una bajada, para ello es necesario tener un control de lo que gastamos día a día.

Una aplicación web es una herramienta que es accesible a través de un navegador, mediante la red, y que esta alojada en un servidor. El planteamiento de usar este acercamiento como solución se debe a lo fácil que es usar un navegador web, no se necesita tener un ordenador potente para usar este tipo de aplicaciones, independencia del sistema operativo que tengas instalado desde la máquina que uses debido a que es posible acceder desde cualquier dispositivo (móvil u ordenador), etc. Este y otros

motivos hacen que esta solución sea la que más se utilice cada vez más en el desarrollo de aplicaciones.

Este TFG, como se ha comentado en la descripción de la figura 1, forma parte de TFG grupal en el cual habrá una serie de aspectos grupales y una serie de características únicas del TFG:

- Objetivos grupales: la aplicación web en común tendrá los siguientes objetivos comunes:
 - Gestión de usuarios
 - Gestión de hogares
 - Gestión de contadores inteligentes
 - Visualización y filtrado de los valores de los contadores inteligentes
- Objetivos específicos: centrándonos más en este trabajo sus objetivos más específicos son:
 - Gestión de las notificaciones tanto que procedan de los contadores como de las invitaciones
 - Visualización de los valores de energía para ello se hará uso del trabajo realizado en el proyecto de la integración de datos públicos TFE 6
 - Filtrado de los valores de energía para ello se hará uso del trabajo realizado en el proyecto de la integración de datos públicos TFE 6
 - Costes de energía según empresas para ello se hará uso del trabajo realizado en el proyecto de la integración de datos públicos TFE 6

A modo de resumen, el flujo normal de uso del sistema sería que primero el usuario iniciaría sesión en el sistema. Luego el usuario tendría que crear un hogar y un dispositivo para ese hogar. Después de su creación podría acceder a ellos y ver todas las medidas que se recojan. Una vez creados también cabe la posibilidad de que se pueda cambiar su información.

A parte de este flujo tiene otras alternativas que sería la de ver los costes de energía, ver las tarifas de las empresas o añadir la tarifa que tienes contratada.

1.2 Objetivos

La meta de este trabajo es el desarrollo de una aplicación web que permita:

- A los contadores inteligentes desarrollados en el proyecto guardar las mediciones.
- Al usuario acceder a esa información de manera amigable y que pueda usarla para tomar acción en cuanto al consumo que tiene.

Para que se pueda cumplir con esto se ha definido los siguiente objetivos que cumpla la aplicación:

1. La aplicación debe proporcionar un medio para que el usuario pueda registrar los contadores inteligentes que será acordado con los integrantes del K-Project
2. La aplicación debe proporcionar un medio para que los contadores inteligentes puedan subir las medidas en un formato preestablecido por los componentes del K-Project
3. La aplicación debe proporcionar una gestión de usuarios para que estos puedan acceder a la misma y poder hacer las gestiones pertinentes
4. La aplicación debe proporcionar al usuario la posibilidad de ver los datos que han recogido sus contadores inteligentes.
5. La aplicación debe proporcionar al usuario una serie de parámetros por los cuales el usuario pueda filtrar o generar estadísticas conforme a sus intereses
6. La aplicación debe proporcionar una opción al usuario de que cuando se pase unos límites modificables por el mismo, reciba una notificación

1.3 Estructura de la memoria

Este Trabajo de Fin de Grado tiene una organización de 9 capítulos, en los que entra la *Introducción*, que es la que contiene este punto y los demás puntos son los que siguen:

- *Capítulo 2. Metodología*

En este capítulo se va a describir la metodología de trabajo que se ha seguido en la realización del proyecto.

- *Capítulo 3. Tecnologías usadas*

En este capítulo se comentan las tecnologías que se ha usado y que componen el trabajo.

- *Capítulo 4. Análisis del sistema*

En este capítulo se describen la idea general del sistema y se expone tanto los requisitos funcionales como los no funcionales.

- *Capítulo 5. Especificación*

En este capítulo se expone una especificación del sistema cogiendo los requisitos extraídos del capítulo anterior.

- *Capítulo 6. Diseño*

En este capítulo se expone el diseño del sistema en general, para la base de datos y para los 2 módulos diferenciados

- *Capítulo 7. Implementación*

En este capítulo se describen los pasos y métodos seguidos para implementar el Backend y el Frontend

- *Capítulo 8. Pruebas*

En este capítulo se describen las pruebas realizadas al sistema.

- *Capítulo 9. Conclusiones y líneas futuras*

En este capítulo hay una conclusión sobre el trabajo en general y sobre cuales podrían ser los caminos que podría tomar.

2

Metodología

En este capítulo se va definir algunas de las diferentes metodologías que existen y la que se ha elegido para el desarrollo de este Trabajo Fin de Grado.

Una metodología se podría definir como la ejecución de una serie de métodos, pasos y técnicas, las cuales tienen un rigor científico aplicables al proceso de ejecución del proyecto que se realiza. Dentro de la ingeniería del software esos pasos podrían ser la recogida de requisitos, el diseño del modelo, la implementación o la ejecución de las pruebas.

La metodología se usa para poder estructurar, planear y dominar todo el proceso que conlleva el desarrollo del proyecto. Dentro de la ingeniería de software existen diversas metodologías debido a la abundancia de proyectos distintos que existen y se escoge la metodología que más se amolda a lo que el proyecto necesita y por tanto es una de las decisiones más importantes y críticas a la hora de hacer un proyecto.

Algunas de las metodologías más usadas y/o conocidas se agrupan en 3 grandes grupos:

- **Metodologías clásicas:**
 - *En cascada*: esta tiene una serie de fases y como su nombre indica no se pasa a la siguiente fase hasta que no se ha acabado con la anterior

- *Basado en prototipos*: este generalmente sus fases se dividen en los prototipos que se generan, como su nombre indica
- *Incremental*: su idea se basa en descomponer el proyecto en una serie de "incrementos" los cuales se van haciendo por fases de manera que la aplicación cree de manera "incremental".

- **Metodologías ágiles:**

- Iterativo: este se basa en que tiene una serie de fase las cuales se llaman "iteraciones" y cada iteración tienen las mismas subfases, que son aplicadas a distintos objetivos del proyecto.
- Programación extrema (Extreme Programming): en esta metodología se basa en saber que a lo largo del proyecto puede haber muchos cambios y se suele ejecutar en equipos de desarrollo pequeños debido a que se necesita de mucha comunicación, de proyectos simples y mucha retroalimentación.
- Scrum: es un proceso que se basa en definir objetivos (sprints, demos, reuniones, etc) y una serie de roles (scrum master, product owner, etc).

Para la realización del proyecto se decidió seguir una metodología ágil con un desarrollo iterativo que nos permite acondicionar la forma de trabajar a los requisitos del proyecto, aportándonos flexibilidad e inmediatez de respuesta para adecuar el proyecto a diversas circunstancias.

Tecnologías usadas

A continuación se exponen todas las tecnologías utilizadas para la realización del proyecto.

3.1. Python

Python [3] es un lenguaje de programación de alto nivel interpretado, cuya principal virtud se basa en la legibilidad de su código. Este es multiparadigma, debido a que soporta el paradigma orientado a objetos, el paradigma imperativo y en menor medida el paradigma funcional y se trata de un lenguaje dinámico, multiplataforma y posee una licencia de código abierto.



Figura 2: Logo de Python

Este lenguaje es administrado por la Python Software y se trata de un lenguaje simple, de sintaxis clara y de propósito general. Es un lenguaje interpretado, lo cual significa que no tiene necesidad de compilarlo y por tanto proporciona un desarrollo ágil.

3.2. Django

Django [4] es un framework de alto nivel que sirve para hacer desarrollo web. Este framework está escrito en Python, es de código abierto y usa el patrón de arquitectónico Modelo-Vista-Controlador.



Figura 3: Logo de django

Además se usa la librería Django Rest Framework la cual permite hacer que se pueda usar Django como una API Rest, para poder crear microservicios que serán usados, como veremos más adelante, en el Frontend de la aplicación

3.3. MongoDB

MongoDB [5] es un gestor de bases de datos NoSQL, de código abierto, orientado a documentos y que nos da una alta escalabilidad y flexibilidad de datos.



Figura 4: Logo de MongoDB

Este, en vez de guardar las típicas relaciones y tablas que se hacen en los modelos de datos relacionales, guarda estructuras de datos BSON (Binary JSON) con un esquema dinámico para la integración sencilla de ciertas aplicaciones.

3.4 MongoDB Atlas

MongoDB Atlas [6] es un servicio que permite la configuración, implementación y escalado de bases de datos sin tener en cuenta dónde se encuentra alojada.



Figura 5: Logo de MongoDB Atlas

Este servicio maneja toda la complejidad que lleva la administración y reparación de las implementación en el proveedor de servicios en la nube.

3.4. JavaScript

JavaScript [7] es un lenguaje de programación interpretado, dialecto o implementación del estándar ECMAScript. Es un lenguaje débilmente tipado y que soporta los paradigmas de orientado a objetos e imperativo.



Figura 6: Logo de JavaScript

Este lenguaje se usa normalmente del lado del cliente en los navegadores web, permitiendo que las interfaces mejoren considerablemente, ya que dejan de ser estáticas y pasan a ser web dinámicas. Pero también se usan en otros ámbitos como es en los documentos PDF, aplicaciones de escritorio, etc.

3.5. React

React [8] es una biblioteca de código abierto de JavaScript que se usa para crear interfaces de usuario amigables y adaptables. Esta biblioteca es mantenida tanto por Facebook como por la comunidad de software libre.



Figura 7: Logo de React

Su principal función es ayudar a los desarrolladores que crean aplicaciones web en las cuales hay datos que varían durante todo el tiempo. Está siendo utilizada en múltiples paginas como Imgur, Airbnb, etc.

3.6. Bootstrap

Bootstrap [9] se puede definir como una biblioteca multiplataforma de código abierto que sirve para el diseño de las pantallas de las aplicaciones web. Esta contiene múltiples componentes como pueden ser botones, menús modales, etc.



Figura 8: Logo de Bootstrap

Para integrarla con React usaremos React-Bootstrap, que básicamente lo que tiene es construido cada componente de Bootstrap en una componente de React sin necesidad de dependencia de otras librerías.

3.7. Visual Studio Code

Visual Studio Code [10] (VS Code) es un editor de código fuente que fue hecho por Microsoft. Este incluye múltiples ayudas como puede ser soporte de depuración, integración con Git, finalización inteligente de código, etc.



Figura 9: Logo de Visual Studio Code

VS Code se basa en Electron, un framework que se utiliza para implementar Chromium y Node.js como aplicaciones de escritorio.

3.8. Git

Git [11] es un software que se usa para el control de versiones. Este fue diseñado pensando en la compatibilidad, confiabilidad y eficiencia del mantenimiento de versiones cuando se tiene un gran número de archivos.



Figura 10: Logo de Git

Git se usará a través de GitHub [12] el cual es una plataforma donde se puede alojar proyectos y que tiene como sistema de control de versiones Git.



Figura 11: Logo de GitHub

3.9. MagicDraw

MagicDraw [13] es una herramienta CASE (Computer Aided Software Engineering) que fue desarrollada por No Magic, desarrollado en Java y se distribuye bajo licencia de propietario.



Figura 12: Logo de MagicDraw

Esta herramienta es compatible con el estándar UML 2.3 (Lenguaje de Modelado Unificado) cuyo propósito es el modelado de aplicaciones.

3.10. JWT

JWT [14] (JSON Web Token) es un estándar que sirve para la creación de tokens los cuales pueden ser usados para dar acceso y para verificación de identidad. Su diseño es compactos para que puedan ser enviados en las URLs.



Figura 13: Logo JWT

En nuestro caso lo usaremos en el backend y por ello usaremos PyJwt que es una librería de Python, que sigue este estándar y nos da facilidades para la creación y comprobación de tokens.

3.11. Swagger

Swagger [15] es un conjunto de herramientas que sirven para diseñar y documentar servicios web RESTful.



Figura 14: Logo Swagger

En este caso se usará para la documentación del módulo Backend.

3.12. Heroku

Heroku [16] es una plataforma PaaS (Platform as a Service) de computación en la nube. Esta es capaz de soportar múltiples lenguajes de programación.



Figura 15: Logo de Heroku

Al ser PaaS nos ofrece una gestión de recursos básico y suficiente para el sistema que se va a desarrollar. Esta herramienta la usaremos para poder acceder tanto al Frontend como al Backend de manera online.

3.13. IFML

IFML (standard Interaction Flow Modeling Language) [17] es un lenguaje estándar que sirve para expresar el contenido, interacción del usuario y el comportamiento del software.



Figura 16: Logo IFML

Para hacer uso, se hará a través de la página web [18]

3.14. HTTP y HTTPS

HTTP [19] (Protocolo de transferencia de hipertexto) es un protocolo de comunicación que sirve para la transferencia de información. HTTPS [20] no es más que HTTP pero con una capa segura que protege las comunicaciones con integridad y confidencialidad de los datos. Este protocolo tiene un intercambio de mensajes en texto plano, legible y fácil de depurar. Los mensajes tienen la siguiente estructura:

- Línea inicial: esta primera parte varía si es una petición o es una respuesta:
 - Peticiones: en este caso será el método de la petición los cuales pueden ser muchos pero los que vamos a usar nosotros van a ser:
 - GET: Sirve para solicitar una representación de un recurso
 - POST: Sirve para el envío de datos para que sean almacenados
 - PUT: Parecido al POST pero este se suele usar para la actualización de datos
 - DELETE: Sirve para borrar el recurso especificado
 - Respuestas: en este caso contendrán el código de respuesta y la frase asociada. Estas pueden estar en 5 grupos:
 - 1xx: Son respuestas informativas
 - 2xx: Respuestas correctas
 - 3xx: Respuestas de redirección (se necesita de más acciones del cliente)
 - 4xx: Respuestas del error que en este caso significa que es culpa del cliente
 - 5xx: Respuestas del error que en este caso significa que es culpa del servidor

En nuestro caso los códigos más usados van a ser:

- 200: OK
- 201: Created
- 204: No content (se usa en las respuestas correctas de los PUT y DELETE)
- 400: Bad Request
- 401: Unauthorized
- 404: Not Found

- Cabeceras: son metadatos que le dan gran flexibilidad al protocolo. En nuestro caso usaremos sobre todo las cabeceras:
 - Authorization: es donde irá el token para poder saber quién hace la petición
 - X-total-count: esta es utilizada las respuestas de las peticiones GET que recibe un Array JSON.
- Cuerpo del mensaje: este es opcional y normalmente lo usaremos cuando hagamos una petición PUT o POST para indicarle la información.

HTTPS es el protocolo que se usará para la comunicación entre módulo como se explicará más adelante.

3.15. JSON

JSON [21] (JavaScript Object Notation) es un formato de texto cuya notación es subconjunto de la notación literal de objetos de JavaScript. Una de las ventajas de JSON es que muchos más fácil escribir un analizador sintáctico de las otras alternativas que existen. El formato de un JSON es pueden ser de dos tipos:

- *Objetos JSON*

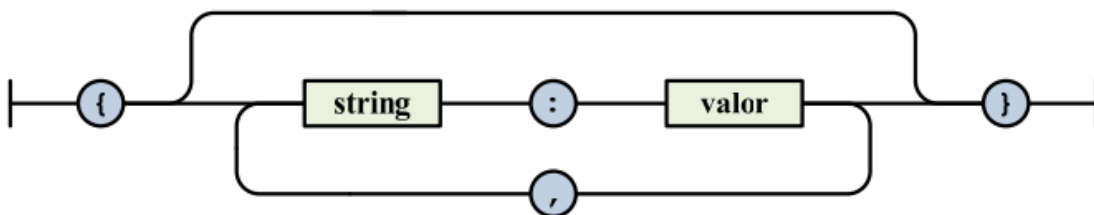


Figura 17: Formato objeto JSON

- *Array JSON*

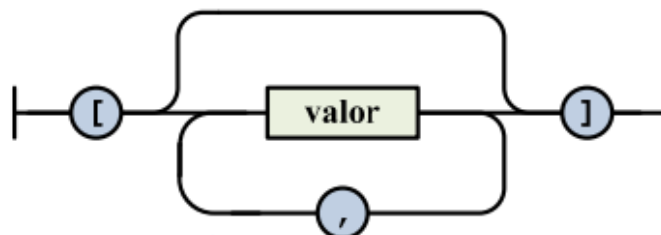


Figura 18: Formato Array JSON

Donde los valores pueden ser lo siguiente:

- String: una cadena de caracteres
- Número: un valor numérico de cualquier tipo (entero o real)
- Objeto JSON
- Array JSON
- Valores booleanos: True o False
- Null

El formato de texto que usaremos para que los módulos se entiendan serán JSON como se comenta más adelante.

4

Análisis del sistema

En este punto se va a describir todo el análisis del sistema. El análisis del sistema es una etapa en la cual, se va a describir el comportamiento que va a tener el sistema e irá acompañado por un documento, en este caso el TFG, donde irán todos los requisitos que debe cumplir la aplicación tanto funcionales como no funcionales.

4.1. Descripción general del sistema

La aplicación web está formada por 2 componentes:

- Backend: esta componente se corresponde con la lógica de negocio. Los objetivos de esta componente son:
 - Dar soporte a la componente Frontend.
 - Dar soporte a los contadores inteligentes para que puedan guardar las medidas en base de datos.
 - Dar soporte a la aplicación móvil que se está desarrollando dentro del K-Project
- Frontend: esta componente se corresponde con la vista de la aplicación. Su principal objetivo es dar soporte al usuario para que pueda acceder a los datos que se guardan de sus contadores inteligentes.

Como resumen, si pensamos en una arquitectura Modelo-Vista-Controlador, el Backend sería la parte de modelo y controlador y el Frontend sería la vista.

4.2. Requisitos funcionales

Los requisitos funcionales es una lista donde se describe todo el comportamiento que debe tener la aplicación.

La lista de requisitos funcionales del sistema es:

- *Gestión de usuarios.*
 - RF1. Un usuario puede iniciar sesión en la plataforma a través de Google
 - RF2. Un usuario puede cerrar sesión
 - RF3. Un usuario puede visualizar sus datos personales
 - RF4. Un usuario puede modificar sus datos personales
 - RF5. Un usuario puede darse de baja del sistema
- *Gestión de los hogares.*
 - RF6. Un usuario puede registrar hogares
 - RF7. Un usuario puede visualizar los datos del hogar
 - RF8. Un usuario puede modificar los datos del hogar
 - RF9. Un usuario puede eliminar hogares
- *Gestión de invitaciones al hogar.*
 - RF10. Un usuario puede visualizar los usuarios con los que tiene compartido un hogar
 - RF11. Un usuario puede visualizar las invitaciones que ha recibido
 - RF12. Un usuario puede invitar a otro usuario a un hogar para que vea los datos que recogen sus dispositivos
 - RF13. Un usuario puede responder a la invitación de otro usuario
 - RF14. Un usuario dueño de una casa puede para de invitar, a otro usuario, para que deje de ver la información de ese hogar.
 - RF15. Un usuario invitado puede salirse de un hogar

- *Registro de contadores inteligentes*
 - RF16. Un usuario puede registrar contadores inteligentes.
- *Gestión de contadores inteligentes*
 - RF17. Un usuario puede visualizar la información de sus contadores inteligentes
 - RF18. Un usuario puede modificar los datos de sus contadores inteligentes
 - RF19. Un usuario puede eliminar sus contadores inteligentes.
- *Visualización de la información recogida por los contadores inteligentes*
 - RF20. Un usuario puede visualizar una estadística de los KWh de los contadores inteligentes. Dicha estadística mostrará:
 - Lo consumido durante el día de hoy
 - Lo consumido durante el mes en el que esta
 - La media consumida por día
 - La media consumida por mes
 - Los mínimo y máximos tanto de un día como de un mes en concreto
 - RF21. Un usuario puede visualizar una estadística del consumo de los contadores inteligentes. Dicha estadística mostrará:
 - El dinero gastado durante el día de hoy
 - El dinero gastado durante el mes en el que esta
 - El dinero gastado de media por día
 - El dinero gastado de media por mes
- *Filtro de la información recogida por los contadores inteligentes*
 - RF22. Un usuario puede filtrar la información entre dos fechas
 - RF23. Un usuario puede filtrar la información entre dos valores
 - RF24. Un usuario puede comparar valores de 2 días

- *Visualización del coste de la energía*
 - RF25. Un usuario puede visualizar los costes de un día en concreto
 - RF26. Un usuario puede ver las tarifas de otras compañías eléctricas
 - RF27. Un usuario puede añadir su tarifa eléctrica al sistema
 - RF28. Un usuario puede ver la media de precios de la última semana
 - RF29. Un usuario puede ver la media de precios del último mes
 - RF30. Un usuario puede ver la media de precios de los últimos 3 meses

- *Gestión de notificaciones*
 - RF31. Un usuario puede configurar la recepción de notificaciones sobre sus contadores inteligentes
 - RF32. Un usuario puede fijar el valor máximo diario sobre el contador inteligente a partir de lo que quiere que se le notifique
 - RF33. Un usuario puede configurar la recepción de notificaciones sobre las invitaciones. Dichas notificaciones serán sobre:
 - Un usuario te ha invitado
 - Un usuario ha aceptado o denegado tu invitación
 - Un usuario te ha dejado de compartir el hogar
 - Un usuario se ha salido de tu hogar

4.3. Requisitos no funcionales

Los requisitos no funcionales es una lista donde se describe los criterios que necesita la aplicación, es decir, restricciones o condiciones que se imponen.

La lista de requisitos no funcionales del sistema es:

- RNF1. Todas las comunicaciones tanto del Backend como del Frontend serán seguras
- RNF2. Las notificaciones de los requisitos funcionales de notificación serán vía correo electrónico.
- RNF3. El registro de contadores inteligentes serán vía la aplicación móvil y el Backend dará soporte para guardar esos registros en la base de datos.

5

Especificación

5.1. Modelo entidad-relación

En este apartado se muestra el modelo descrito a partir de los requisitos expuestos en el capítulo anterior.

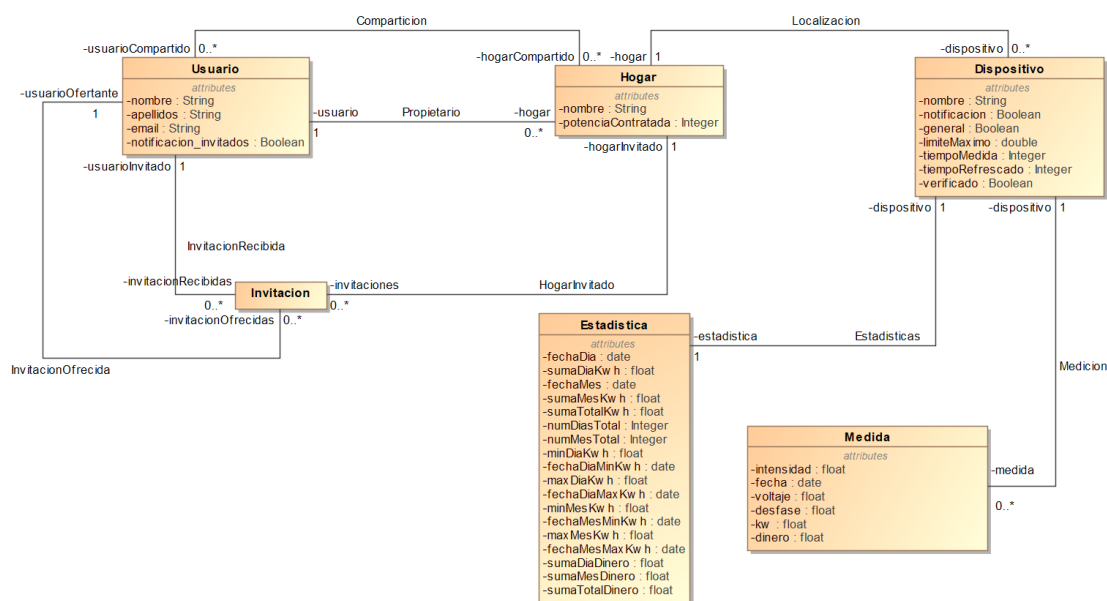


Figura 19: Modelo entidad-relación

Del modelo anterior se entiende que en el sistema hay:

- Usuarios (tabla Usuario), los cuales tiene una serie de hogares (tabla Hogar y relación llamada Propietario).
- Un usuario, además puede estar invitado a varios hogares para ver sus estadísticas (relación Compartición).

- Un usuario puede invitar a otro usuario a un hogar del que es propietario (tabla Invitación).
- Un hogar puede tener varios contadores inteligentes (tabla Dispositivo y relación Localización)
- Un contador puede recoger varias medidas a lo largo del tiempo (tabla Medida y relación Medición)
- Cada contador inteligente tiene una única estadística asociada (tabla Estadística y relación Estadísticas)

En este trabajo se usará una base de datos no sql debido a las ventajas que nos ofrecen:

- Podemos tener un almacenamiento distribuido en vez de tenerlo todo centralizado y esto puede ser interesante si el sistema en algún futuro se hiciera distribuido.
- Con el punto anterior también podemos sacar que en vez de estar alojado en una gran máquina pueden estar en diferentes máquinas por lo que permite un buen escalado.

5.2. Dependencias

Al ser estar este trabajo sumergido dentro de un proyecto mucho más grande algunos de los requisitos expuesto anteriormente van a interactuar con otros trabajos. En concreto:

- *Registro de contadores inteligentes*

Este requisito necesitara de la interacción tanto del trabajo de la aplicación móvil y del trabajo de los contadores inteligentes. Mas concretamente el Backend le dará soporte a la aplicación móvil para que esta pueda registrar un contador inteligente. El flujo de creación de un contador inteligente es:

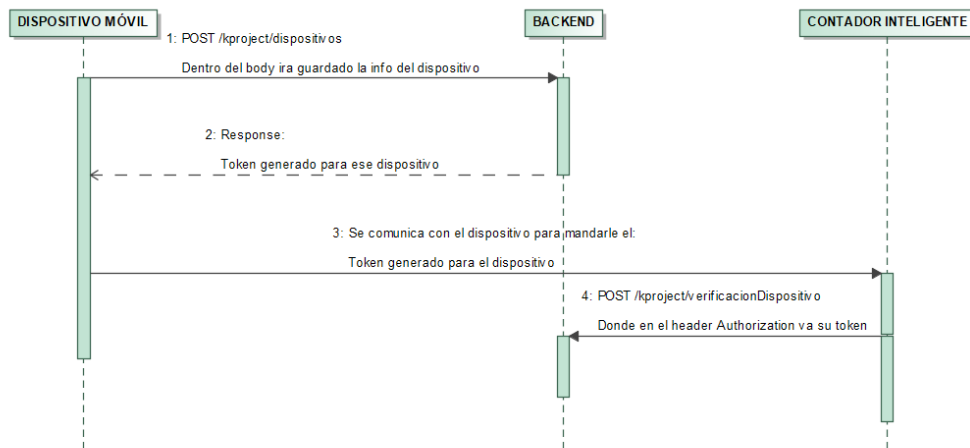


Figura 20: Diagrama de secuencia creación de un contador inteligente

Como se puede ver en la figura el flujo para la creación de un dispositivo seguirá 4 pasos:

1. El usuario quiere crear un contador inteligente que ya tiene instalado y para ello lo que hace es a través de la aplicación móvil crea un dispositivo. Al crearlo el dispositivo se comunica a con el Backend del sistema y le envía la información facilitada por el usuario
2. El Backend recibe la información y ve que todo está en regla (hogar correcto, el usuario es dueño del hogar y el nombre asignado al contador inteligente no está vacío). Una vez que todo está en regla le envía en la respuesta el token que necesita el contador inteligente para poder enviar mediciones al sistema.
3. El dispositivo móvil recibe el token del dispositivo y este lo que hace es enviárselo al contador inteligente.
4. El contador inteligente recibe el token y le envía una petición para certificarle al sistema que ha recibido correctamente el token.

- *Coste de la energía*

Los requisitos relacionados con coste de energía como el RF21 y el paquete de requisitos *Visualización del coste de la energía* usará el trabajo relacionado con los coste de energía del proyecto a través de una API Rest creada

5.3. Diagrama de Casos de Uso

En este apartado se exponen todos los diagramas de casos de uso:

- *Gestión de usuarios*

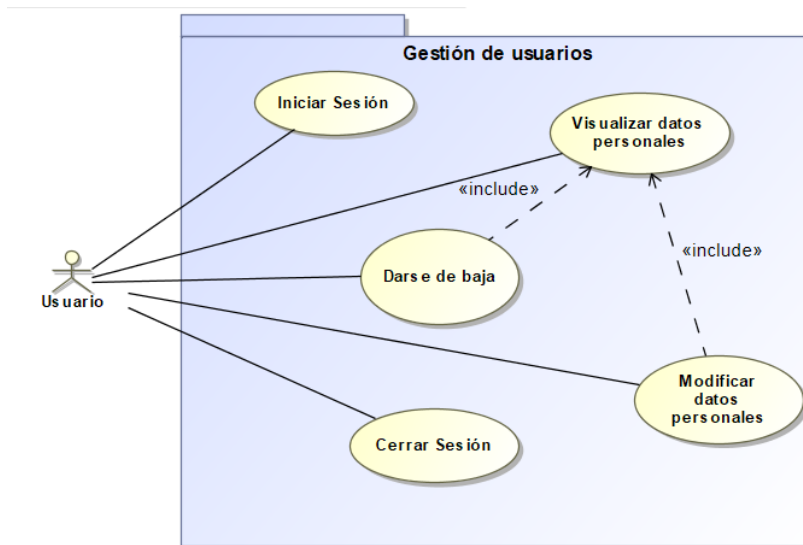


Figura 21: Diagrama de casos de uso de la gestión de usuarios

- *Gestión de hogares*

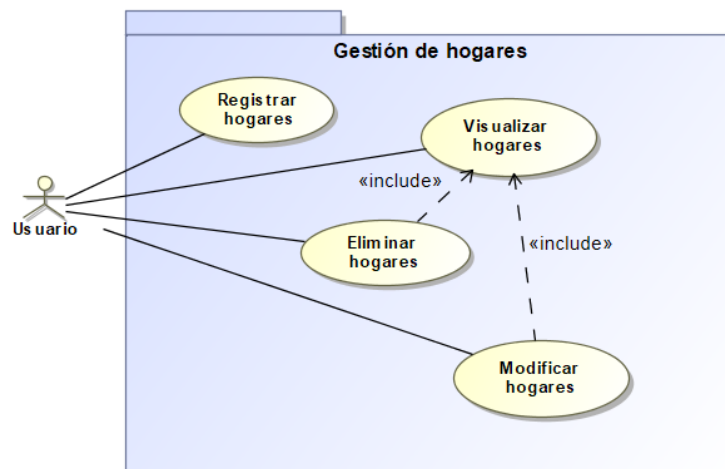


Figura 22: Diagrama de casos de uso de la gestión de hogares

- *Gestión de invitaciones al hogar*



Figura 23: Diagrama de casos de uso gestión de invitaciones al hogar

- *Registro de contadores inteligentes*



Figura 24: Diagrama de casos de uso del registro de contadores inteligentes

- *Gestión de contadores inteligentes*

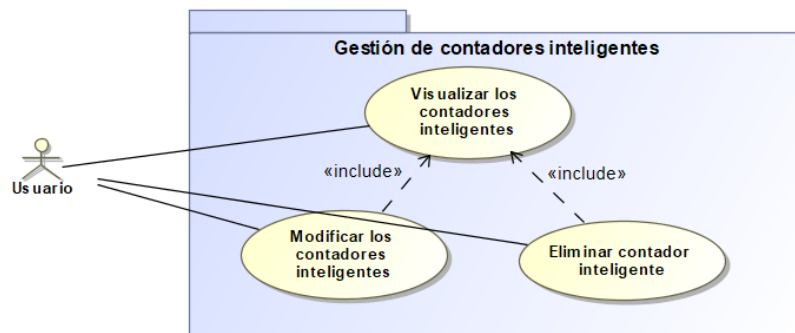


Figura 25: Diagrama de casos de uso de la gestión de contadores inteligentes

- *Visualización de información de contadores inteligentes*

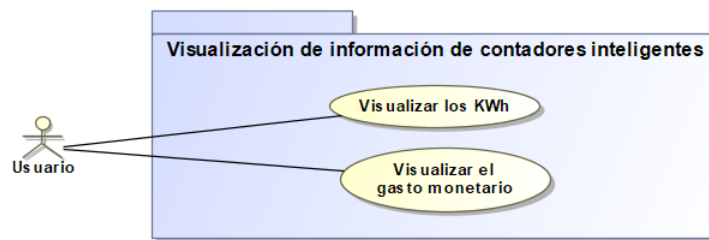


Figura 26: Diagrama de casos de uso de la visualización de información de contadores inteligentes

- *Filtro de información de contadores inteligentes*

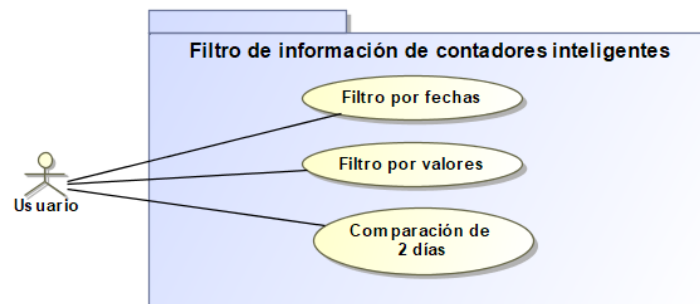


Figura 27: Diagrama de casos de uso del filtro de información de contadores inteligentes

- *Visualización del coste de la energía*

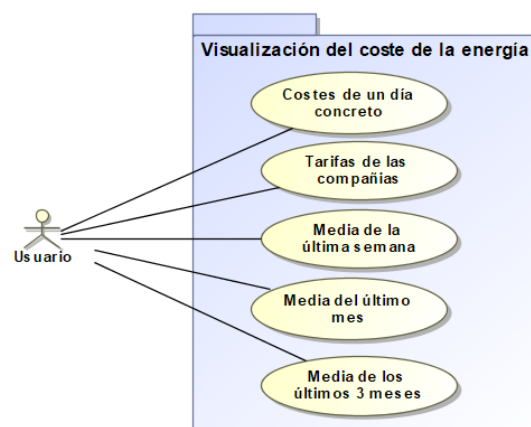


Figura 28: Diagrama de casos de uso de la visualización del coste de la energía

- *Gestión de notificaciones*

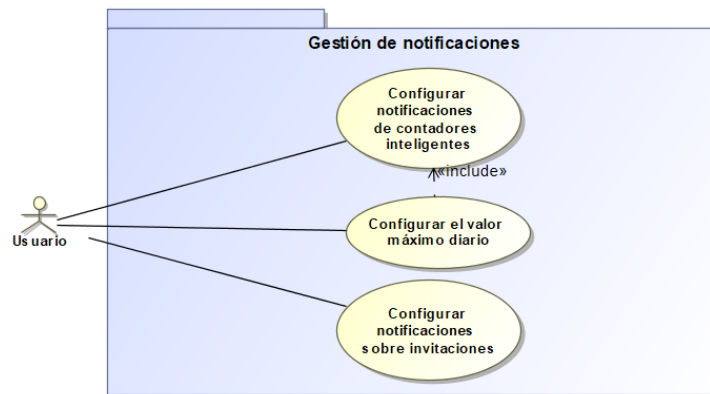


Figura 29: Diagrama de casos de uso de gestión de notificaciones

5.4. Especificación de Casos de uso

En este apartado se exponen todos los casos de uso que se extraen de todos los requisitos del capítulo anterior.

Título	RF1
Descripción	Un usuario puede iniciar sesión en la plataforma a través de una plataforma segura
Pre-condición	El usuario está en la pantalla de iniciar sesión
Post-condición	El usuario iniciar sesión sin problemas
Escenario principal	
1. El usuario selecciona iniciar sesión. 2. El sistema carga el inicio de sesión. 3. El usuario selecciona la cuenta con que quiere iniciar sesión 4. El Frontend lo manda al Backend para comprobar su token 5. El sistema lo redirige a la pantalla de inicio	
Escenario alternativo	
4b. El usuario no puede iniciar sesión con Google 4b1. El sistema le muestra un modal explicándole que no ha conseguido iniciar sesión 4b2. Vuelta al paso 1	

Título	RF2
Descripción	Un usuario puede cerrar sesión
Pre-condición	El usuario ya ha iniciado sesión
Post-condición	El usuario cierra sesión sin problemas
Escenario principal	
1. El usuario selecciona "Hola," 2. El sistema le muestra una lista desplegada 3. El usuario selecciona la opción "Cerrar sesión" 4. El sistema cierra sesión y lo redirige a la página de iniciar sesión	

Título	RF3
Descripción	Un usuario puede visualizar sus datos personales
Pre-condición	El usuario ha iniciado sesión.
Post-condición	El usuario visualiza sus datos personales
Escenario principal	
1. El usuario selecciona "Hola," 2. El sistema le muestra una lista desplegada 3. El usuario selecciona la opción "Mi perfil" 4. El sistema le muestra la pantalla de su perfil con su información	

Título	RF4
Descripción	Un usuario puede los modificar sus datos personales
Pre-condición	El usuario ha iniciado sesión.
Post-condición	El usuario modifica sus datos perfectamente
Escenario principal	
1. El usuario selecciona "Hola," 2. El sistema le muestra una lista desplegada 3. El usuario selecciona la opción "Mi perfil" 4. El sistema le muestra la pantalla de su perfil con su información 5. El usuario modifica los campos que quiera 6. El usuario selecciona la opción "Actualizar mis datos" 7. El sistema comprueba los datos y guarda la información en Base de datos 8. El sistema recarga la página de "Mi perfil"	
Escenario alternativo	
7b. El usuario deja el campo nombre vacío 7b2. El sistema le dice que no puede dejar el campo nombre a vacío 7b3. Vuelta al paso 4	

Título	RF5
Descripción	Un usuario puede darse de baja del sistema
Pre-condición	El usuario ha iniciado sesión.
Post-condición	El usuario se da de baja del sistema
Escenario principal	
1. El usuario selecciona "Hola," 2. El sistema le muestra una lista desplegada 3. El usuario selecciona la opción "Mi perfil" 4. El sistema le muestra la pantalla de su perfil con su información 5. El usuario selecciona "Darse de baja" 6. El sistema le informa de que esto será irreversible 7. El usuario selección "Aceptar" 8. El sistema lo borra de la Base de datos y lo redirige a la pantalla de iniciar sesión	
Escenario alternativo	
7b. El usuario selecciona "Cancelar" 7b2. El sistema le muestra la pantalla de perfil 7b3. Vuelta al paso 4	

Título	RF6
Descripción	Un usuario puede registrar hogares
Pre-condición	El usuario ha iniciado sesión.
Post-condición	El usuario registra el hogar
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable 3. El usuario selecciona "Nuevo hogar" 4. El sistema muestra el formulario para añadir un hogar 5. El usuario rellena los campos existentes 6. El usuario selecciona "Añadir" 7. El sistema guarda la información en base de datos 8. El sistema notifica que se ha registrado un hogar con éxito 9. El sistema le redirige hacia la página del hogar creado	
Escenario alternativo	
6b. El usuario deja vacío el campo nombre 6b2. El sistema le informa que no puede dejar ese campo vacío 6b3. Vuelta al paso 4 6c. El usuario pone una palabra o frase en el campo potencia contratada 6c2. El sistema le informa que ese campo tiene que ser numérico 6c3. Vuelta al paso 4	

Título	RF7
Descripción	Un usuario puede visualizar los datos del hogar
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar
Post-condición	El usuario visualiza la información del hogar
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar que quiere visualizar su información 4. El sistema le muestra la página del hogar 5. El usuario selecciona "Editar hogar" 6. El sistema muestra el formulario donde puede visualizar la información del hogar	

Título	RF8
Descripción	Un usuario puede modificar los datos de un hogar
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar
Post-condición	El usuario modifica los datos del hogar
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar que quiere modificar su información 4. El sistema le muestra la página del hogar 5. El usuario selecciona "Editar hogar" 6. El sistema muestra el formulario donde visualiza la información del hogar 7. El usuario modifica los datos del hogar 8. El usuario selecciona "Actualizar hogar" 9. El sistema guarda la información en base de datos 10. El sistema le notifica que se ha podido actualizar correctamente 11. El sistema le redirige a la página del hogar	
Escenario alternativo	
8b. El usuario deja vacío el campo nombre 8b2. El sistema le informa que no puede dejar ese campo vacío 8b3. Vuelta al paso 6 8c. El usuario pone una palabra o frase en el campo potencia contratada 8c2. El sistema le informa que ese campo tiene que ser numérico 8c3. Vuelta al paso 6	

Título	RF9
Descripción	Un usuario puede eliminar hogares
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar
Post-condición	El usuario elimina el hogar correctamente
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar que quiere eliminar 4. El sistema le muestra la página del hogar 5. El usuario selecciona "Editar hogar" 6. El sistema muestra el formulario donde puede visualizar la información del hogar 7. El usuario selecciona "Eliminar hogar" 8. El sistema le informa que esta acción es irreversible 9. El usuario selecciona "Aceptar" 10. El sistema lo borra de la base datos 11. El sistema lo redirige a la página principal	
Escenario alternativo	
9b. El usuario selecciona "Cancelar" 9b2. Vuelta al paso 6	

Título	RF10
Descripción	Un usuario puede visualizar los usuarios con los que tiene compartido un hogar
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar
Post-condición	El usuario visualiza los invitados del hogar
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar que quiere visualizar los invitados 4. El sistema le muestra la página del hogar 5. El usuario selecciona "Compartir" 6. El sistema muestra un modal con todos los usuarios con los que ha compartido el hogar	

Título	RF11
Descripción	Un usuario puede visualizar las invitaciones que ha recibido
Pre-condición	El usuario ha iniciado sesión
Post-condición	El usuario visualiza sus invitaciones
Escenario principal	
1. El usuario selecciona "Ver invitaciones" 2. El sistema muestra un modal con todas las invitaciones recibidas	

Título	RF12
Descripción	Un usuario puede invitar a otro usuario a un hogar para que vea los datos que recogen sus dispositivos
Pre-condición	Los usuarios deben estar registrados y el ofertante debe ser dueño del hogar
Post-condición	El usuario ofertante invita a otro usuario
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar que quiere invitar a otro usuario 4. El sistema le muestra la página del hogar 5. El usuario selecciona "Compartir" 6. El sistema muestra todos los usuarios con los que ha compartido el hogar 7. El usuario selecciona "Invitar" 8. El sistema muestra un modal donde puede introducir el correo del usuario 9. El usuario introduce el correo del usuario 10. El usuario selecciona "Compartir" 11. El sistema guarda la invitación en base de datos	
Escenario alternativo	
10b. El usuario introduce un correo no valido 10b1. El sistema le informa que ese correo no existe o no es valido 10c. El usuario introduce un usuario al que ya ha invitado pero todavía no ha contestado 10c1. El sistema le informa que a ese usuario ya se le ha invitado	

Título	RF13
Descripción	Un usuario puede responder a la invitación de otro usuario
Pre-condición	El usuario ha iniciado sesión y ha recibido la invitación
Post-condición	El usuario invitado responde a la invitación
Escenario principal	
1. El usuario selecciona "Ver invitaciones" 2. El sistema muestra un modal con todas las invitaciones recibidas 3. El usuario selecciona "Aceptar" para aceptar la invitación o "Denegar" para denegar la invitación 4. El sistema le notifica que ha sido aceptada o denegada la invitación	

Título	RF14
Descripción	Un usuario puede dejar de invitar a otro usuario a su hogar
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar
Post-condición	El usuario deja de invitar a otro usuario
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar que quiere dejar de compartir con un usuario 4. El sistema le muestra la página del hogar 5. El usuario selecciona "Compartir" 6. El sistema muestra un modal con todos los usuarios con los que ha compartido el hogar 7. El usuario selecciona "Dejar de compartir" al lado del usuario que se quiere dejar de compartir 8. El sistema borra la compartición de la base de datos	

Título	RF15
Descripción	Un usuario invitado puede salirse de un hogar
Pre-condición	El usuario ha iniciado sesión y está invitado en el hogar
Post-condición	El usuario se sale del hogar invitado
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar compartido del que quiere salirse 4. El sistema le muestra la página del hogar 5. El usuario selecciona "Abandonar hogar" 6. El sistema muestra un modal comentándole que esto será permanente 7. El usuario selecciona "Aceptar" 8. El sistema lo borra de base de datos	
Escenario alternativo	
7b. El usuario selecciona "Cancelar" 7b1. Vuelta al paso 1	

Título	RF16
Descripción	Un usuario puede registrar contadores inteligentes y debe hacerlo a través de la aplicación móvil la cual enviará la información al sistema
Pre-condición	El usuario debe estar registrado y tener al menos un hogar creado
Post-condición	El usuario se sale del hogar invitado
Escenario principal	
1. El sistema recibe la información del móvil 2. El sistema comprueba que toda la información es correcta 3. El sistema le devuelve el token de acceso del contador inteligente al móvil para que se lo envíe al contador inteligente	
Escenario alternativo	
2b. El nombre está vacío 2b1. El sistema le envía un error donde le comenta que el nombre no puede estar vacío	

Título	RF17
Descripción	Un usuario puede visualizar la información de sus contadores inteligentes
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar
Post-condición	El usuario visualiza los datos de los contadores inteligentes
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar del que quiere ver el contador inteligente 4. El sistema le muestra la página del hogar 5. El usuario selecciona "Editar hogar" 6. El sistema muestra el formulario donde puede visualizar la información del hogar 7. El usuario selecciona el contador inteligente que quiere visualizar 8. El sistema la muestra la información del contador inteligente seleccionado	

Título	RF18
Descripción	Un usuario puede modificar la información de sus contadores
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar
Post-condición	El usuario modifica la información de los contadores inteligentes
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar del que quiere modificar el contador inteligente 4. El sistema le muestra la página del hogar 5. El usuario selecciona "Editar hogar" 6. El sistema muestra la información del hogar 7. El usuario selecciona el contador inteligente que quiere modificar 8. El sistema le muestra la información del contador inteligente seleccionado 9. El usuario modifica la información del contador inteligente 10. El usuario selecciona "Actualizar contador inteligente" 11. El sistema guarda la información y lo redirige a la página del hogar	
Escenario alternativo	
10b. El usuario deja vacío el campo nombre 10b2. El sistema le informa que no puede dejar ese campo vacío 10b3. Vuelta al paso 8 10c. El usuario introduce una cadena de texto en el campo límite máximo 10c2. El sistema le informa que ese campo debe ser numérico 10c3. Vuelta al paso 8	

Título	RF19
Descripción	Un usuario puede eliminar sus contadores inteligentes
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar
Post-condición	El usuario elimina el contador inteligente
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar del que quiere eliminar el contador inteligente 4. El sistema le muestra la página del hogar 5. El usuario selecciona "Editar hogar" 6. El sistema muestra el formulario donde puede visualizar la información del hogar 7. El usuario selecciona el contador inteligente que quiere eliminar 8. El sistema le muestra la información del contador inteligente seleccionado 9. El usuario selecciona el botón con icono de basura 10. El sistema le informa que esta acción es irreversible 11. El usuario selecciona "Aceptar" 12. El sistema lo borra de la base de datos 13. El sistema lo redirige a la página del hogar	
Escenario alternativo	
11b. El usuario selecciona "Cancelar" 11b2. Vuelta al paso 8	

Título	RF20
Descripción	Un usuario puede visualizar las estadísticas de los KWh de los contadores inteligentes
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar o le han invitado
Post-condición	El usuario visualiza las estadísticas del contador inteligente
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar del que quiere ver los kwh del contador inteligente 4. El sistema le muestra la página del hogar 5. El usuario selecciona el contador inteligente 6. El sistema muestra las estadísticas del contador inteligente seleccionado	

Título	RF21
Descripción	Un usuario puede visualizar las estadísticas del dinero gastado por los contadores inteligentes
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar o le han invitado
Post-condición	El usuario visualiza las estadísticas del dinero gastado del contador inteligente
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar del que quiere ver el dinero gastado del contador inteligente 4. El sistema le muestra la página del hogar 5. El usuario selecciona el contador inteligente 6. El sistema muestra las estadísticas del contador inteligente seleccionado	

Título	RF22
Descripción	Un usuario puede filtrar la información entre dos fechas
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar o le han invitado
Post-condición	El usuario filtra la información
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar del que quiere filtrar el contador inteligente 4. El sistema le muestra la página del hogar 5. El usuario selecciona el contador inteligente 6. El sistema muestra las estadísticas del contador inteligente seleccionado 7. El usuario selecciona "Tramos de tiempo" 8. El usuario inserta las fechas por las que quiere filtrar 9. El usuario selecciona "Filtrar" 10. El sistema muestra la gráfica con los filtros especificados	
Escenario alternativo	
9b. El usuario introduce una fecha mínima más grande que una fecha máxima 9b2. El sistema le informa que esto no puede ocurrir 9b3. Vuelta al paso 8	

Título	RF23
Descripción	Un usuario puede filtrar la información entre dos valores
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar o le han invitado
Post-condición	El usuario filtra la información
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar del que quiere filtrar el contador inteligente 4. El sistema le muestra la página del hogar 5. El usuario selecciona el contador inteligente 6. El sistema muestra las estadísticas del contador inteligente seleccionado 7. El usuario selecciona "Valores máximo y mínimos de consumo" 8. El usuario inserta los valores máximo y mínimo por los que quiere filtrar 9. El usuario selecciona "Filtrar" 10. El sistema muestra la gráfica con los filtros especificados	
Escenario alternativo	
9b. El usuario introduce un valor que no es numérico 9b2. El sistema le informa que debe ser numérico 9b3. Vuelta al paso 8	

Título	RF24
Descripción	Un usuario puede comparar valores de 2 días
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar o le han invitado
Post-condición	El usuario puede comparar los valores
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar del que quiere filtrar el contador inteligente 4. El sistema le muestra la página del hogar 5. El usuario selecciona el contador inteligente 6. El sistema muestra las estadísticas del contador inteligente seleccionado 7. El usuario selecciona la pestaña "Comparar días" 8. El usuario inserta los 2 días que quiere comparar 9. El usuario selecciona "Comparar" 10. El sistema muestra las gráficas de los días seleccionados	

Título	RF25
Descripción	Un usuario puede visualizar los costes de un día en concreto
Pre-condición	El usuario ha iniciado sesión
Post-condición	El usuario visualizar los costes de un día concreto
Escenario principal	
1. El usuario selección un día 2. El usuario selecciona "Buscar" 3. El sistema le muestra en una gráfica los costes de ese día	

Título	RF26
Descripción	Un usuario puede ver las tarifas de otras compañías eléctricas
Pre-condición	El usuario ha iniciado sesión
Post-condición	El usuario visualiza la tarifa de una compañía concreta
Escenario principal	
1. El usuario selecciona "Valores de energía" 2. El sistema despliega una lista de opciones 3. El usuario selecciona "Precios de las empresas" 4. El sistema le muestra la página donde sale los precios de las empresas 5. El usuario selecciona la empresa que quiere ver 6. El sistema le muestra la empresa con sus precios	

Título	RF27
Descripción	Un usuario puede añadir su tarifa eléctrica al sistema
Pre-condición	El usuario ha iniciado sesión
Post-condición	El usuario añade su tarifa correctamente
Escenario principal	
1. El usuario selecciona "Valores de energía" 2. El sistema despliega una lista de opciones 3. El usuario selecciona "Precios de las empresas" 4. El sistema le muestra la página donde sale los precios de las empresas 5. El usuario selecciona "Añadir tarifa" 6. El sistema le muestra un formulario para añadir los datos 7. El usuario rellena el formulario 8. El usuario selecciona "Nueva tarifa" 9. El sistema guarda la tarifa y lo redirige a la página anterior	
Escenario alternativo	
8b. El usuario introduce valores incorrectos 8b2. El sistema informa que no se ha podido añadir dicha tarifa	

Título	RF28
Descripción	Un usuario puede ver la media de precios de la última semana
Pre-condición	El usuario ha iniciado sesión
Post-condición	El usuario visualiza la media de precios
Escenario principal	
1. El usuario selecciona "Valores de energía" 2. El sistema despliega una lista de opciones 3. El usuario selecciona "Media de precios de la última semana" 4. El sistema le muestra una pantalla emergente donde le muestra la estadística	

Título	RF29
Descripción	Un usuario puede ver la media de precios del último mes
Pre-condición	El usuario ha iniciado sesión
Post-condición	El usuario visualiza la media de precios
Escenario principal	
1. El usuario selecciona "Valores de energía" 2. El sistema despliega una lista de opciones 3. El usuario selecciona "Media de precios del último mes" 4. El sistema le muestra una pantalla emergente donde le muestra la estadística	

Título	RF30
Descripción	Un usuario puede ver la media de precios de los últimos 3 meses
Pre-condición	El usuario ha iniciado sesión
Post-condición	El usuario visualiza la media de precios
Escenario principal	
1. El usuario selecciona "Valores de energía" 2. El sistema despliega una lista de opciones 3. El usuario selecciona "Media de precios de los últimos 3 meses" 4. El sistema le muestra una pantalla emergente donde le muestra la estadística	

Título	RF31
Descripción	Un usuario puede configurar la recepción de notificaciones sobre sus contadores inteligentes
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar
Post-condición	El usuario configura la recepción de notificaciones
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar del que quiere configurar el contador inteligente 4. El sistema le muestra la página del hogar 5. El usuario selecciona "Editar hogar" 6. El sistema muestra el formulario donde puede visualizar la información del hogar 7. El usuario selecciona el contador inteligente que quiere configurar 8. El sistema la muestra la información del contador inteligente seleccionado 9. El usuario selecciona si quiere recibir o no notificación 10. El usuario selecciona "Actualizar contador inteligente" 11. El sistema guarda la información en la base de datos 12. El sistema lo redirige a la página del hogar	

Título	RF32
Descripción	Un usuario puede fijar el valor máximo diario sobre el contador inteligente a partir que quiere que se le notifique
Pre-condición	El usuario ha iniciado sesión y es dueño del hogar
Post-condición	El usuario fija el valor máximo para recibir notificaciones
Escenario principal	
1. El usuario selecciona "Hogares" 2. El sistema le muestra un desplegable con sus hogares 3. El usuario selecciona el hogar del contador inteligente del que quiere fijar el valor máximo 4. El sistema le muestra la página del hogar 5. El usuario selecciona "Editar hogar" 6. El sistema muestra la información del hogar 7. El usuario selecciona el contador inteligente que modificar el valor máximo 8. El sistema la muestra la información del contador inteligente seleccionado 9. El usuario modifica el valor máximo 10. El usuario selecciona "Actualizar contador inteligente" 11. El sistema guarda la información en la base de datos 12. El sistema lo redirige a la página del hogar	
Escenario alternativo	
10b. El usuario introduce texto 10b2. El sistema le informa que ese campo solo permite números 10b3. Vuelta al paso 8	

Título	RF33
Descripción	Un usuario puede configurar la recepción de notificaciones sobre las invitaciones
Pre-condición	El usuario ha iniciado sesión
Post-condición	El usuario configura la recepción de notificaciones
Escenario principal	
1. El usuario selecciona "Hola," 2. El sistema le muestra una lista desplegada 3. El usuario selecciona la opción "Mi perfil" 4. El sistema le muestra la pantalla de su perfil con su información El usuario selecciona el contador inteligente 5. El usuario en el apartado "Notificaciones" selecciona si quiere recibir información 6. El usuario selecciona la opción "Actualizar mis datos" 7. El sistema comprueba los datos y guarda la información en Base de datos 8. El sistema recarga la página de "Mi perfil"	

5.5. Diagramas de secuencia

En este apartado se exponen algunos de los diagramas de secuencia de los casos de uso:

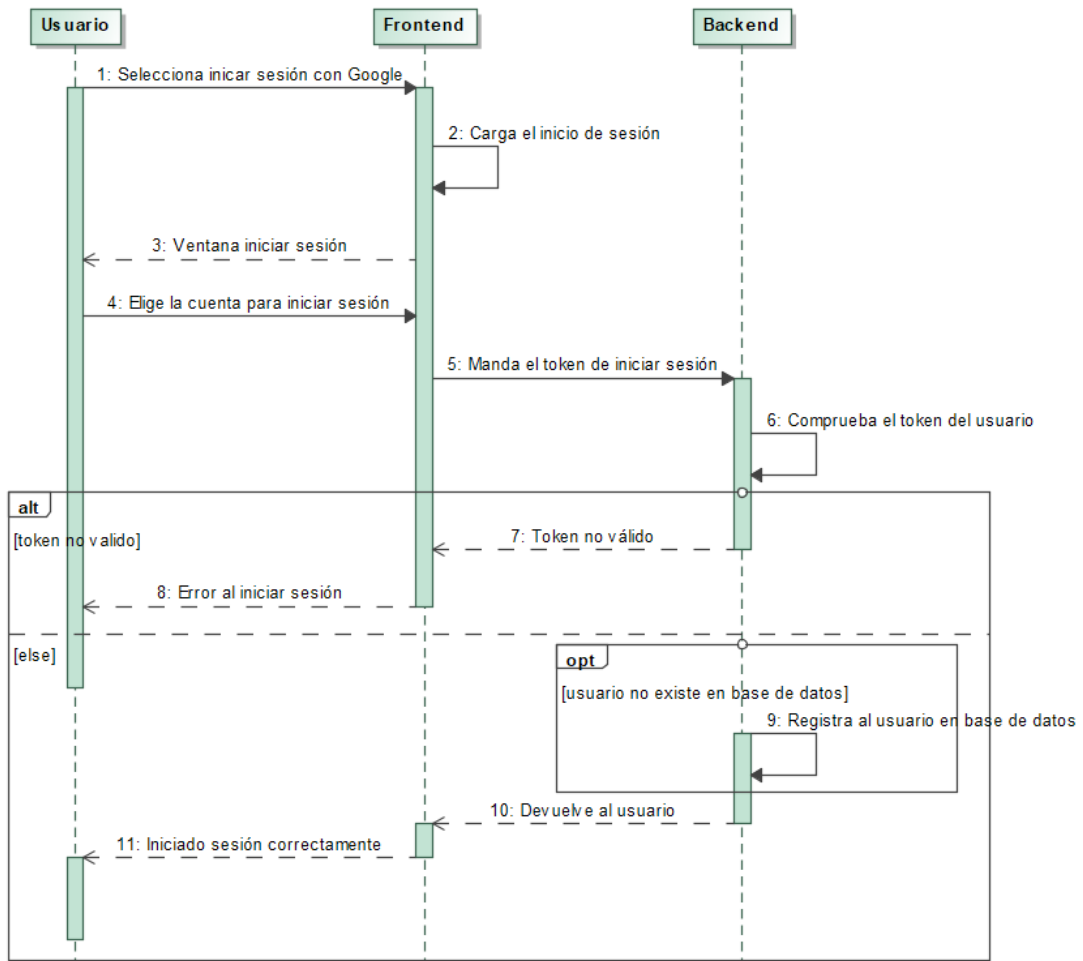


Figura 30: Diagrama de secuencia RF1

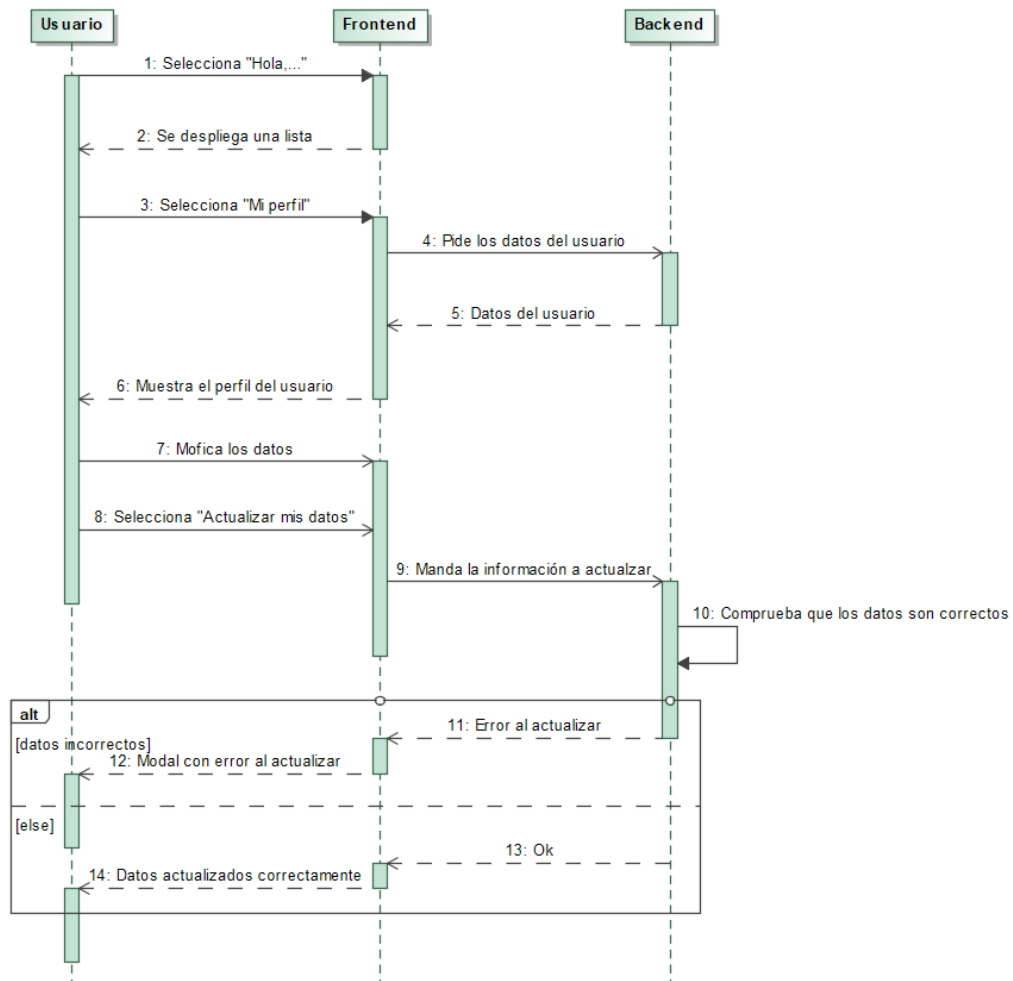


Figura 31: Diagrama de secuencia RF4

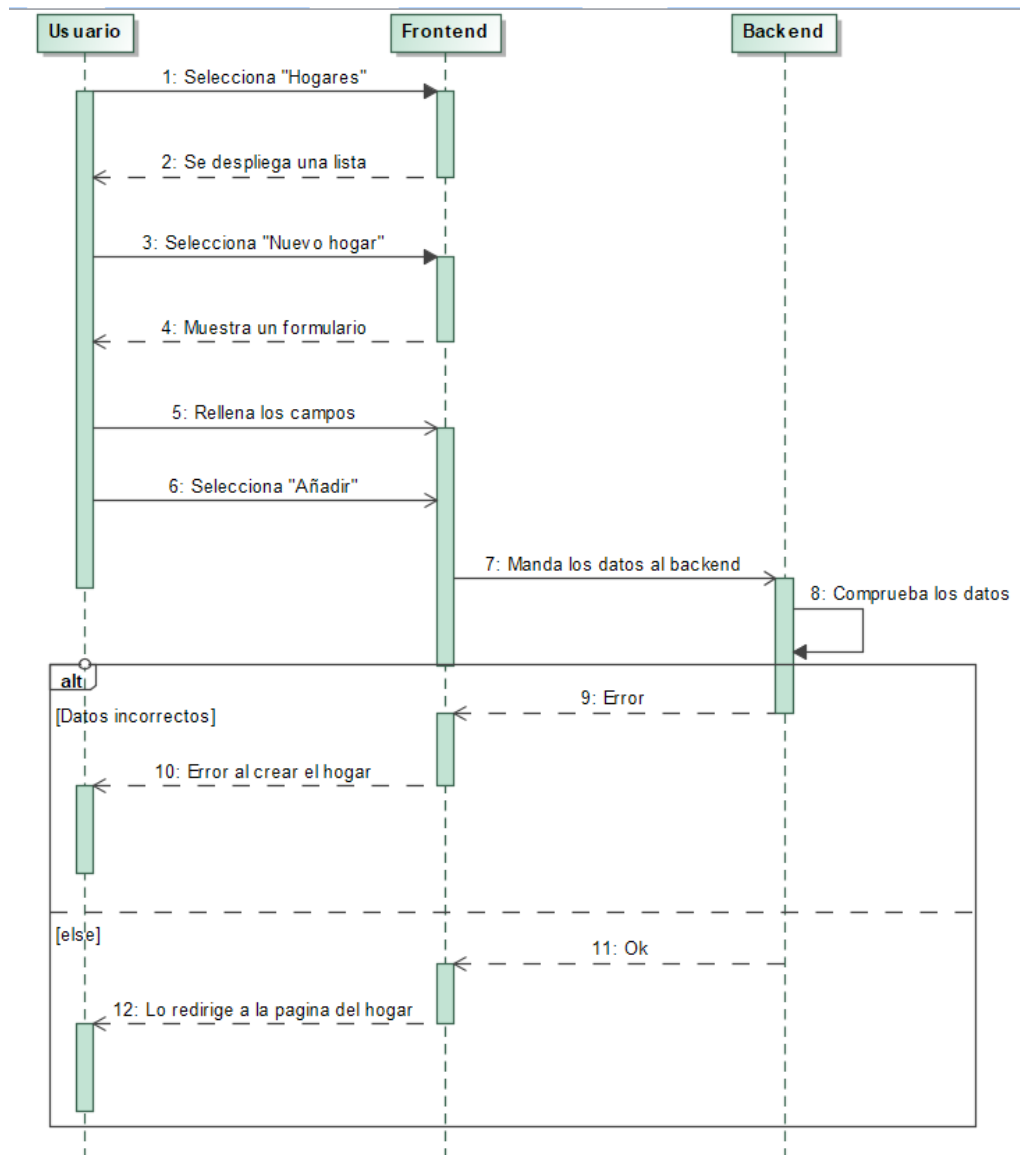


Figura 32: Diagrama de secuencia RF6

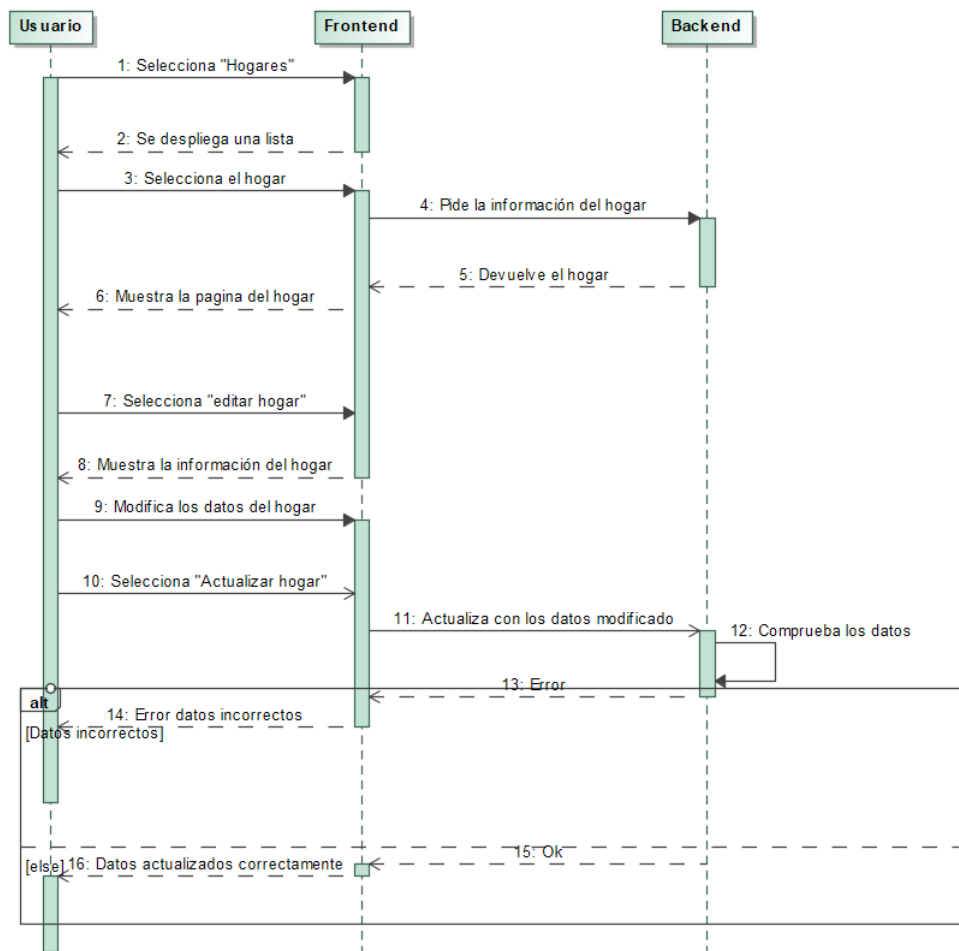


Figura 33: Diagrama de secuencia RF8

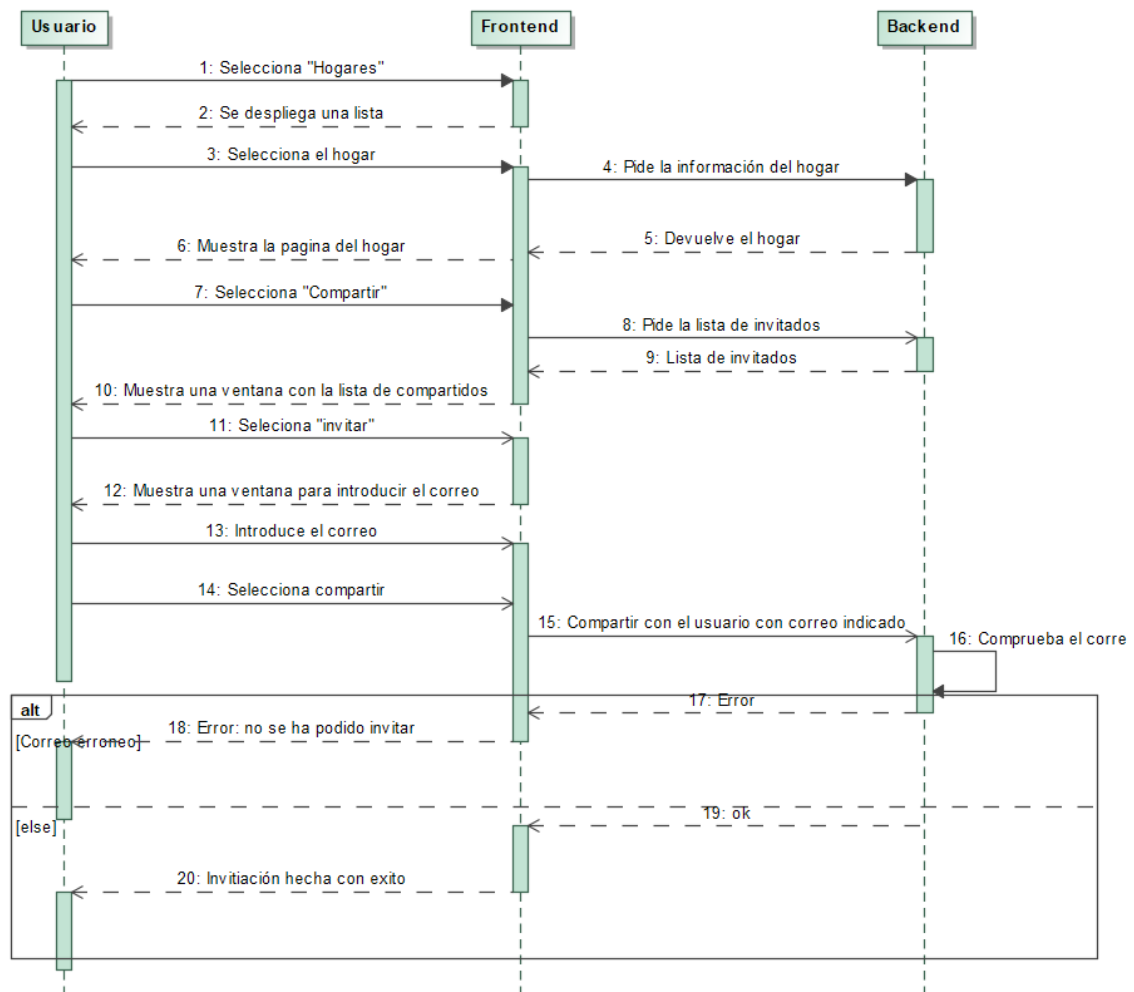


Figura 34: Diagrama de secuencia RF12

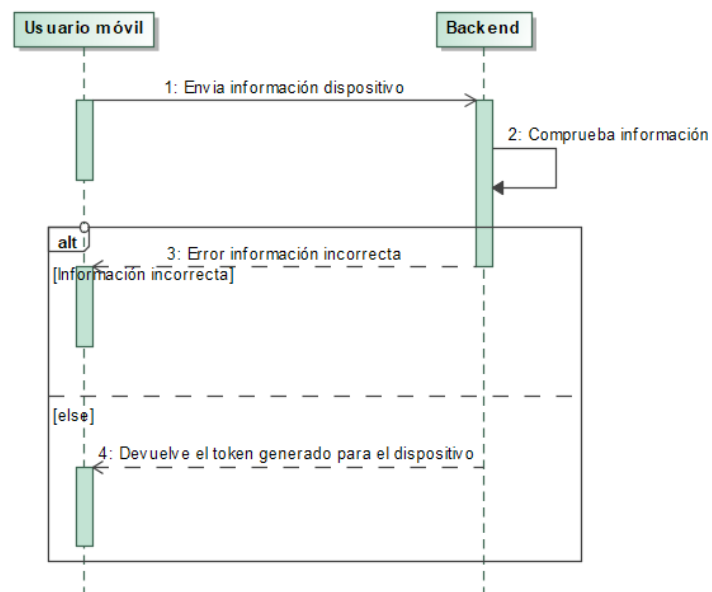


Figura 35: Diagrama de secuencia RF16

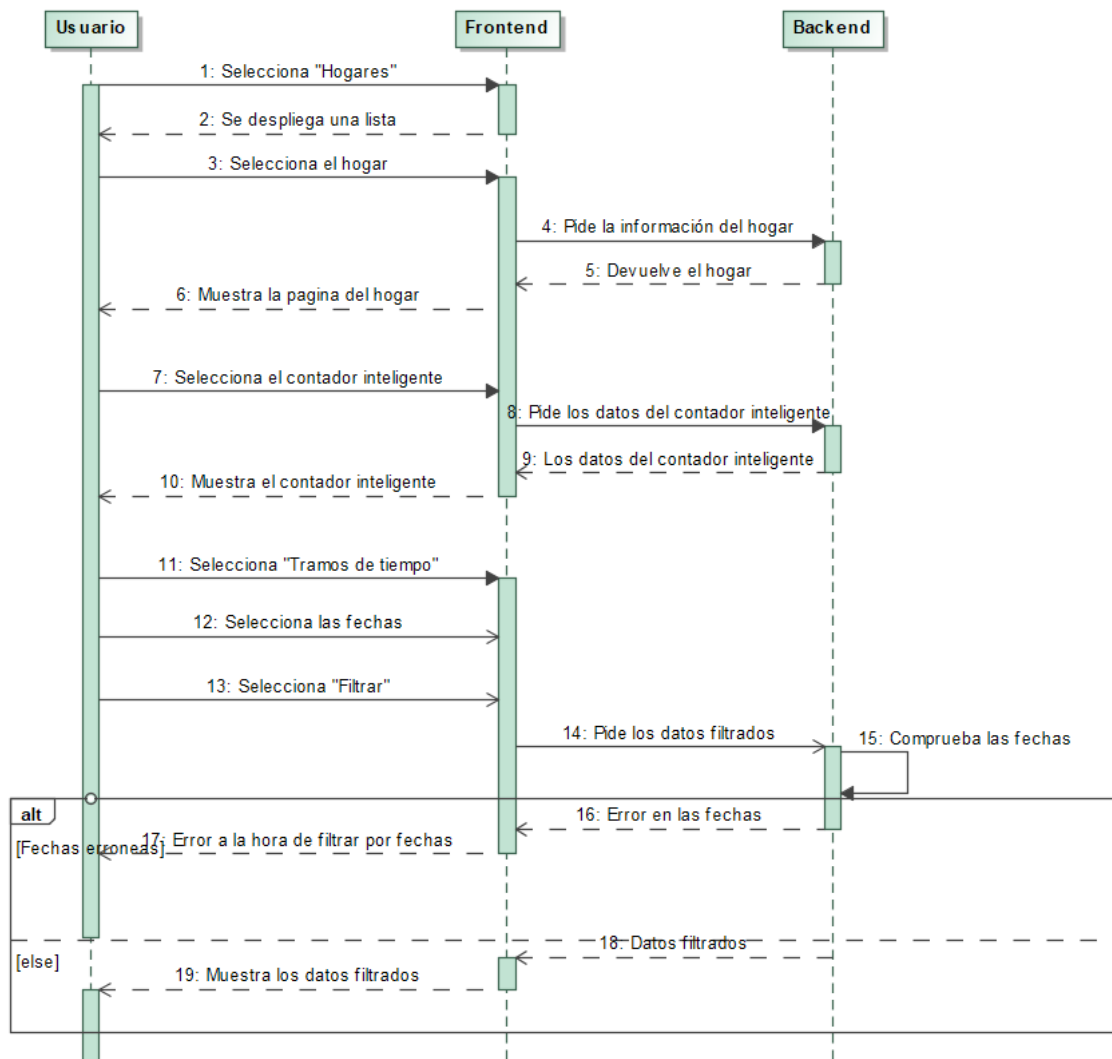


Figura 36: Diagrama de secuencia RF22

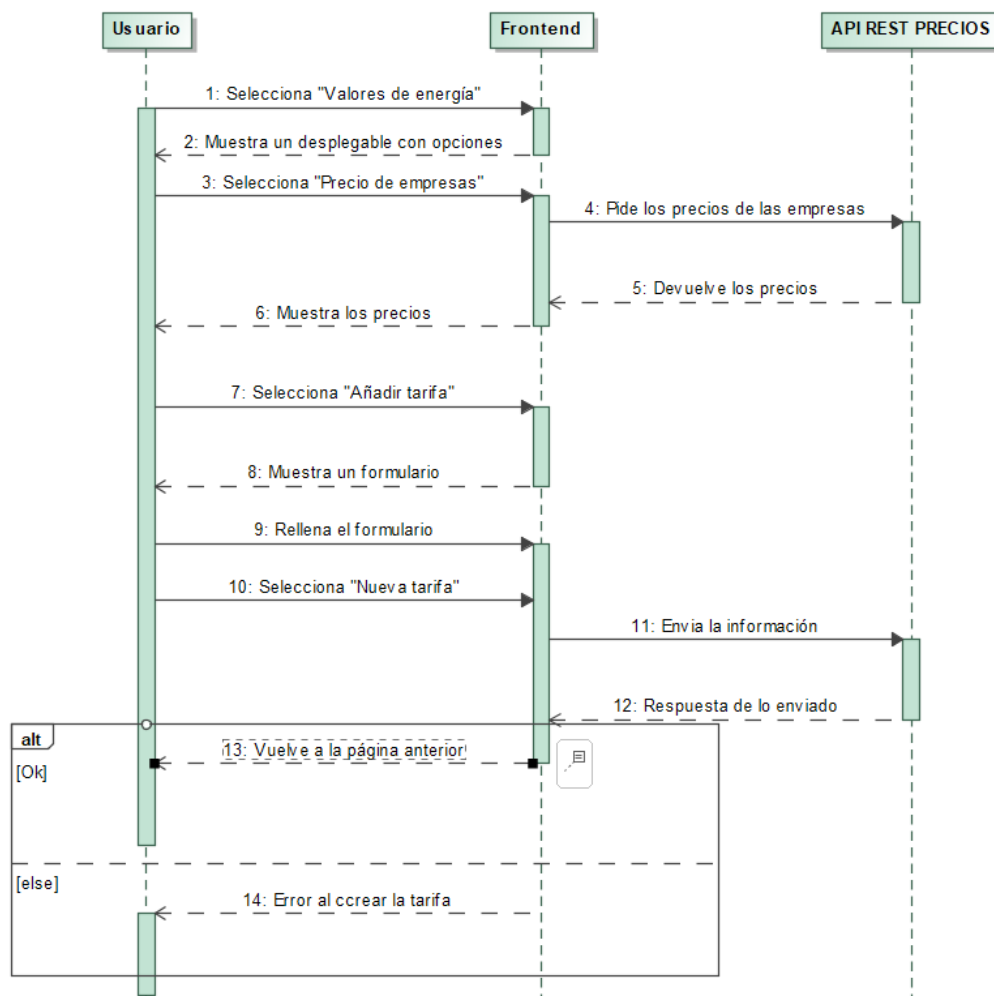


Figura 37: Diagrama de secuencia RF27

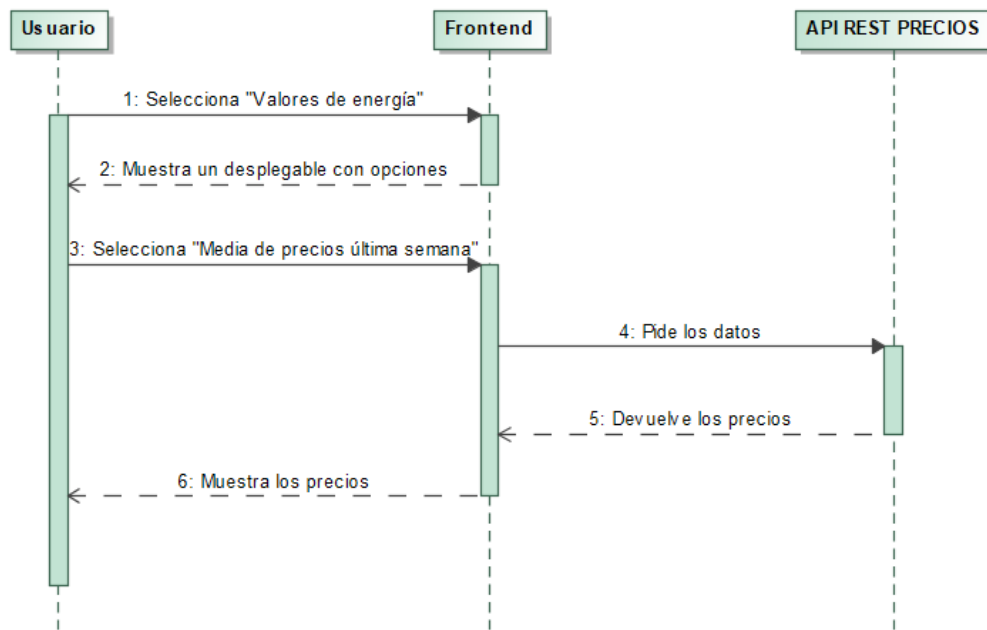


Figura 38: Diagrama de secuencia RF28

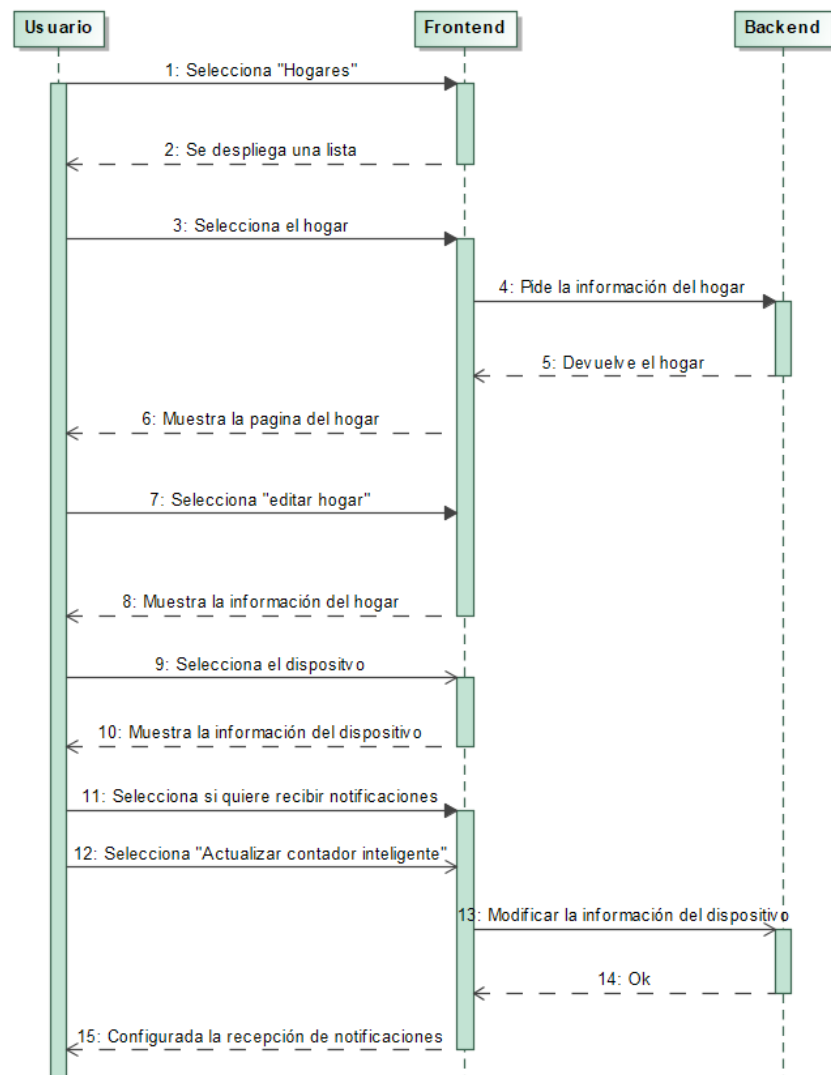


Figura 39: Diagrama de secuencia RF32

6

Diseño

6.1. Arquitectura general de la aplicación

La arquitectura general de la aplicación es la siguiente:

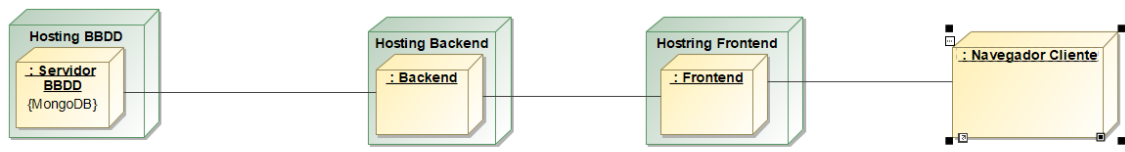


Figura 40: Arquitectura general de la aplicación

Donde se pueden visualizar 4 componentes bien diferenciadas:

- Servidor BBDD: donde irá la base de datos del sistema
- Backend: que será el módulo encargado de la lógica y comunicación de servidor y vista
- Frontend: es la vista de la aplicación web

Y como se puede ver estos módulos pueden estar en una única máquina o pueden estar en máquinas por separado debido al uso de la arquitectura de microservicios que lo hace totalmente modular.

La comunicación entre módulos usará el protocolo HTTPS y como medio de comunicación se usará JSON como se menciona en el capítulo de Tecnologías usadas.

Esta arquitectura se asemeja al arquitectura Modelo-Vista-Controlador, donde tenemos:

- Vista: que es el módulo Frontend que será el que se descargue en el navegador del usuario y pueda interactuar con él para sus propósitos.
- Modelo y Controlador: es el módulo Backend donde podremos distinguir:
 - Controlador: que serán los endpoints que desarrollaremos que controlaran a que funciones llamar
 - Modelo: es aquí donde se definirán los modelos para interactuar con la base de datos.

6.2. Diseño de la base de datos

La base de datos del sistema será no relacional en la cual guardaremos colecciones con documentos. Teniendo en cuenta el modelo entidad-relación, se ha hecho una aproximación de qué contendrá cada colección.

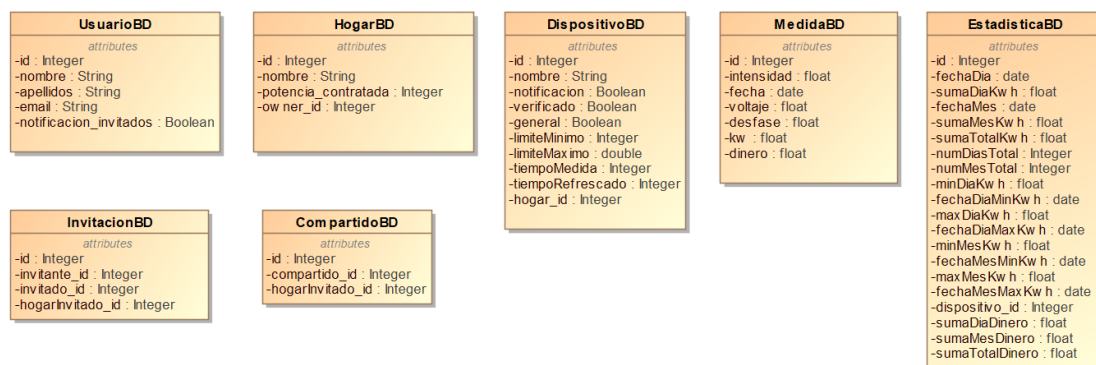


Figura 41: Descripción de las colecciones en Base de datos

Como se muestra para materializar las relaciones se han seguido las siguientes pautas:

- Relación 1 a *: en este caso lo que se hace es guardar en el documento de la colección que tiene la multiplicidad 1 el id del documento de la colección que tiene multiplicidad *
- Relación 1 a 1: en este caso lo que se hace es guardar en una de las 2 colecciones el id del otro
- Relación * a *: en este caso se necesita de una colección intermedia donde guardar los id de las dos colecciones que se relacionan

6.3. Módulo Backend

Como se ha comentado varias veces este módulo se va a encargar de la lógica del sistema y de ser el enlace entre la base de datos y el módulo Frontend. Para ello este módulo se compondrá de una API REST.

Una API RESTful [22] o simplemente API REST es una interfaz de programación de aplicaciones que permite los servicios web. Esta permite todas las peticiones HTTP o HTTPS que serán las que usaremos como comunicación como antes se ha mencionado. Para la creación de este módulo crearemos varios endpoints. Un endpoint no es más que una URL a la que llamar para poder hacer uso de los recursos que hay. Para la documentación de este módulo se ha usado Swagger y se puede ver en la siguiente figura:

Login Sirve para verificar el login de usuarios		^
POST	/login	Login en la aplicación web
Usuarios Obtención de la información de los usuarios		^
GET	/usuarios	Obtención de todos los usuarios existentes
PUT	/usuarios/{id}	Actualización de la información (solo nombre y apellidos) del usuario que esta loggeado
DELETE	/usuarios/{id}	Eliminación de la cuenta del usuario loggeado
Hogar Acceso a la información de hogar		^
GET	/hogars	Obtención de todos los hogares existentes del usuario loggeado
POST	/hogars	Creación de un hogar asignado al usuario loggeado
GET	/hogars/{id}	Obtención del hogar con id
PUT	/hogars/{id}	Actualización de hogar
DELETE	/hogars/{id}	Eliminación de un hogar
GET	/hogars/{id}/compartidos	Obtención de los usuarios que tiene acceso por compartir

Figura 42: Primera parte de la API

Dispositivo Acceso a la información de dispositivo			^
POST	/dispositivos	Creación de un dispositivo asociado a un hogar del usuario loggeado	▼
POST	/verificacionDispositivo	Verificación del dispositivo	▼
GET	/dispositivos/{id}	Obtención del dispositivo con id	▼
PUT	/dispositivos/{id}	Actualización de dispositivo con id	▼
DELETE	/dispositivos/{id}	Eliminación de un dispositivo	▼
GET	/dispositivos/{id}/medidas	Lista de medidas de un dispositivo	▼
Medida Acceso a la información de las medidas			^
GET	/medidas	Obtención de todas las medidas existentes	▼
POST	/medidas	Creación de una medida nueva	▼ ↩

Figura 43: Segunda parte de la API

Invitacion Acceso a la información de las invitaciones			^
DELETE	/invitaciones/{id}	Borra la invitación con id	▼
GET	/invitacionesOfrecidas	Obtención de todas las invitaciones ofrecidas	▼ ↩
POST	/ofrecerInvitacion	Creación de una nueva invitación	▼
GET	/invitacionesRecibidas	Obtención de todas las invitaciones recibidas	▼
Compartidos Acceso a la información de las comparticiones			^
POST	/compartidos	Creación de la compartición cuando acepta la invitación	▼
DELETE	/compartidos/{id}	Eliminación de una compartición	▼

Figura 44: Tercera parte de la API

Para esta parte se ha dividido en 3 partes:

- La primera parte del usuario y el hogar: es aquí donde están los endpoints para la gestión de usuarios y hogares del sistema
- La segunda parte del dispositivo y las medidas: es aquí donde están los endpoints para la gestión de dispositivos y medidas.
- La tercera parte de las invitaciones y las comparticiones: aquí donde están los endpoints para la gestión de invitaciones y las comparticiones.

Para que todas las componentes del proyecto y el módulo Frontend tenga acceso a ella, este módulo se desplegará en Heroku.

6.4. Módulo Frontend

Para el diseño del módulo Frontend se hará tanto un diagrama de interacción entre las pantallas que tendrá este módulo con IFML. Se ha dividido en 4 partes donde todas las partes tendrá una parte común, el home.

- *Diagrama de interacción parte 1*

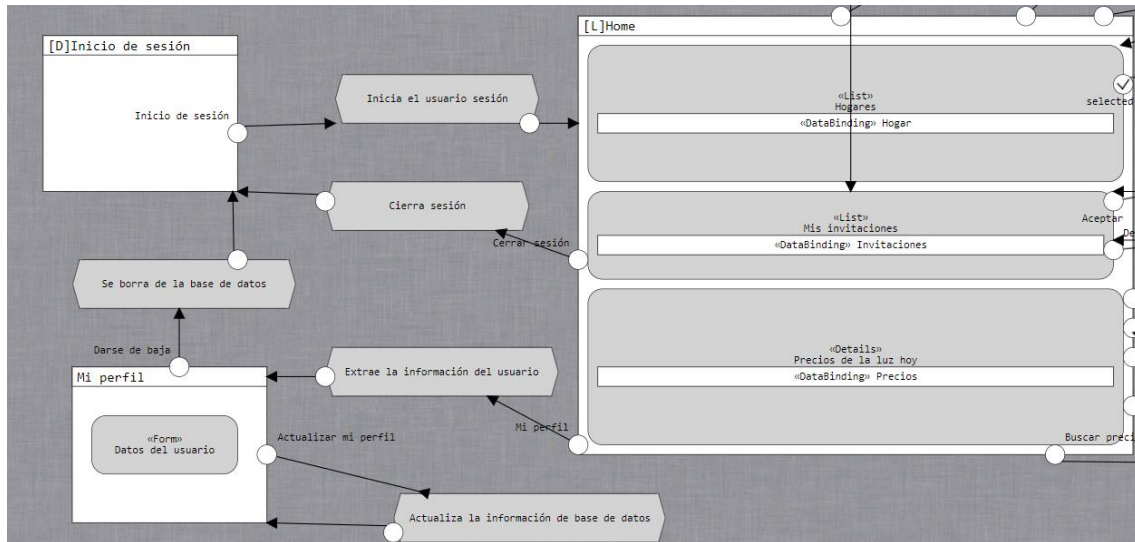


Figura 45: Diagrama de interacción 1º parte

En esta primera parte se refleja la parte de gestión de usuarios donde podemos destacar 2 pantallas aparte del home:

- Inicio de sesión: esta será la primera pantalla que verá el usuario y es donde el usuario iniciara sesión. Una vez inicie sesión el sistema lo redirigirá al home.
- Mi perfil: a esta pantalla se lleva a través del home y es donde se podrá ver la información relativa al usuario logado.

- Diagrama de interacción parte 2

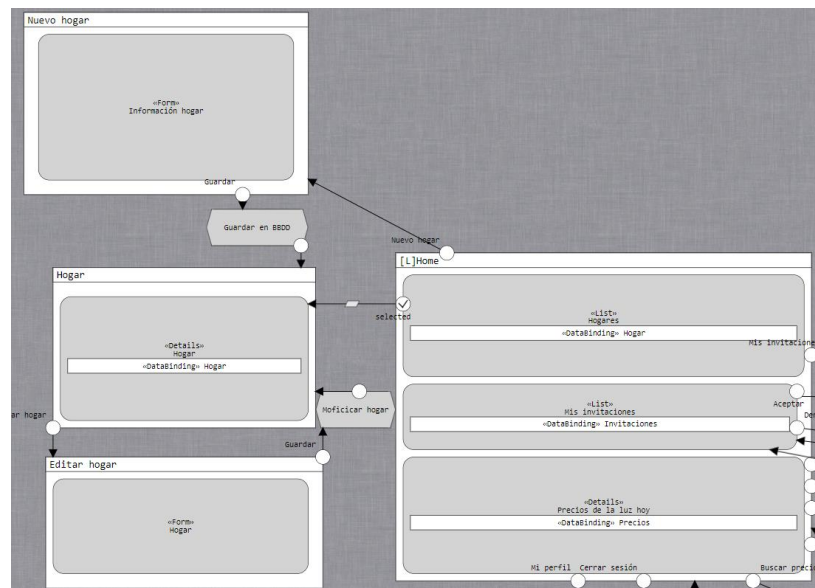


Figura 46: Diagrama de interacción parte 2

En esta segunda parte se refleja la parte de gestión de hogares y dispositivos. En ella se puede destacar 3 pantallas a parte del home:

- Nuevo hogar: en esta pantalla, que se llega a través del home, es donde el usuario encontrara un formulario y pueda registrar los datos necesarios para crear un hogar.
- Hogar: en esta pantalla, que se puede llegar a través del home pero también se llega la primera vez cuando se crea el hogar, es donde se podrá encontrar los dispositivos y se pueda ver las medidas recogidas.
- Editar hogar: en esta pantalla, que se puede llegar a través del hogar, es donde se encuentra la información del hogar y sus dispositivos y donde el usuario puede cambiar dicha información.

- Diagrama de interacción parte 3

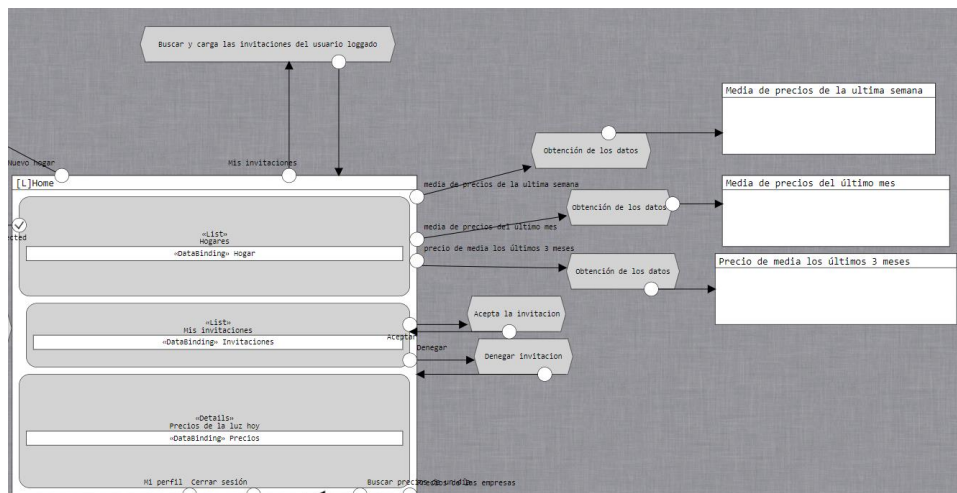


Figura 47: Diagrama de interacción parte 3

En esta segunda parte se refleja la parte de invitaciones y valores de energía:

- Invitaciones: las invitaciones serán accesible a través del NavBar y dará una lista de invitaciones que te han hecho. En ella podrás aceptarlas o denegarlas.
- Precios de la luz: esta gráfica estará en el home y se podrá ver al principio el precio del día y además se podrá elegir el día que se quiere ver los precios de la luz.
- Media de los valores de la luz: en el NavBar habrá 3 opciones para generar 3 estadísticas en cuanto a los valores de la luz:
 - Media en la última semana
 - Media del último mes
 - Media de los últimos 3 meses

- Diagrama de interacción parte 4

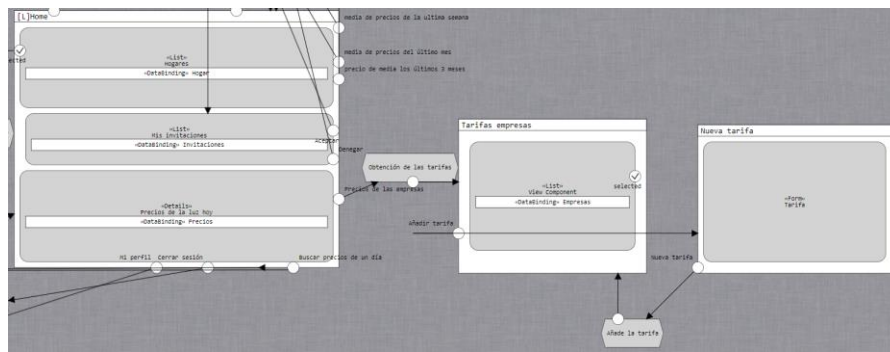


Figura 48: Diagrama de interacción parte 4

En esta parte destacaremos la parte de las tarifas de las empresas. Para ello tendremos 2 ventanas:

- Tarifas de empresas: donde tendremos la información de todas las tarifas de las empresas
- Nueva tarifa: donde tendremos un formulario para poder añadir una tarifa.

Implementación

7.1. Módulo Backend

Para la implementación de este módulo se ha usado el framework Django con rest_framework. Para su implementación se expondrá los principales ficheros que se han usado:

- *Models.py*

En esta fichero es donde serán alojados todos los clases del modelo expuesto anteriormente. Para ello, usaremos la clase models que proporciona Django, de la cual todas las clases del modelo heredarán de esta. Un ejemplo de alguna de las clases hechas:

```
class Usuario(models.Model):
    email = models.EmailField()
    nombre = models.CharField(max_length=30)
    apellidos = models.CharField(max_length=30)
    notificacion_invitados = models.BooleanField()

class Hogar(models.Model):
    nombre = models.CharField(max_length=20)
    potencia_contratada = models.IntegerField()
    owner = models.ForeignKey(Usuario, related_name="owner", on_delete=models.CASCADE)

class Compartido(models.Model):
    compartido = models.ForeignKey(Usuario, related_name='compartido', on_delete=models.CASCADE)
    hogarCompartido = models.ForeignKey(Hogar, related_name='hogarCompartido', on_delete=models.CASCADE)

class Invitacion(models.Model):
    invitado = models.ForeignKey(Usuario, related_name='invitado', on_delete=models.CASCADE)
    hogarInvitado = models.ForeignKey(Hogar, related_name='hogarInvitado', on_delete=models.CASCADE)
    invitante = models.ForeignKey(Usuario, related_name='invitante', on_delete=models.CASCADE)
```

Figura 49: Ejemplo de clases en models.py

Como se puede ver tendremos tantas clases como colecciones en base de datos tenemos y sus respectivas propiedades. Para indicarle de que tipo son las propiedades se usa `models` y como se ha mencionado anteriormente. Un ejemplo podría ser cuando queremos una variable de tipo `String` en base de datos pues como se ve en la imagen sería del tipo `models.CharField()`.

- Dto.py

En este fichero se declaran todas las clases que usaremos para extraer la información, de los objetos, que sea necesaria y para obviar información que el usuario no necesita en un determinado momento. Un ejemplo claro es cuando se quiere obtener todos los hogares de un usuario, en ese caso no se necesita de los dispositivos. Sin embargo, cuando se quiere mostrar toda la información del hogar con sus dispositivos entonces si se necesita de la información del dispositivo. Ejemplos de dto:

```
class HogarObtenerDTO():
    def __init__(self, hogar, compartido):
        self.id = hogar.id
        self.potencia_contratada = hogar.potencia_contratada
        self.nombre = hogar.nombre
        self.compartido = compartido

    @staticmethod
    def toHogarObtenerDTO(hogars, compartidos):
        lista = []
        for h in hogars:
            lista.append(HogarObtenerDTO(h, False))
        for c in compartidos:
            lista.append(HogarObtenerDTO(c.hogarCompartido, True))
        return lista
```

Figura 50: Ejemplo DTO para obtener hogares sin dispositivos

```
class HogarObtenerByIdDTO():
    def __init__(self, hogar, dispositivos, idCompartido):
        self.id = hogar.id
        self.potencia_contratada = hogar.potencia_contratada
        self.nombre = hogar.nombre
        self.dispositivos = DispositivoDTO.toDispositivoDTO(dispositivos)
        self.idCompartido = idCompartido
```

Figura 51: Ejemplo DTO para obtener un hogar solo con toda su información

En estas clases tendremos:

- `__init__`: que será el constructor que usaremos para crear el DTO y poder asignar las propiedades que se pasan por parámetros.
- `ToDtos`: esta función es opcional y son estáticas para poder pasarle una lista de objetos y que los pase todos a dtos.
- *Serializers.py*

En este fichero se definirán clases, las cuales van a heredar de `Serializer`. La clase `Serializer` la proporciona Django y sirve para detallar tanto como serán los body que recibimos y como serán las respuestas que devolverá la API. Por tanto, para cada respuesta distinta en la API habrá un serializador distinto. Dos ejemplos de un serializador de entrada y otro de salida:

```
class UsuarioObtenerSerializer(serializers.Serializer):
    id = serializers.IntegerField()
    email = serializers.EmailField()
    nombre = serializers.CharField(max_length=30)
    apellidos = serializers.CharField(max_length=30, allow_blank=True)
    notificacion_invitados = serializers.BooleanField()
```

Figura 52: Serializer para devolver un usuario

```
class UsuarioModificarSerializer(serializers.Serializer):
    nombre = serializers.CharField(max_length=30)
    apellidos = serializers.CharField(max_length=30, allow_blank=True)
    notificacion_invitados = serializers.BooleanField()
```

Figura 53: Serializer para modificar los datos de un usuario

En cada clase lo que tendremos serán las propiedades que esperamos recibir o enviar y para indicarle de que tipo son las propiedades. Es parecido a como se hace en los modelos, pero en vez de usar `models` se usa `serializers`.

- *Views.py*

En este fichero se definirán las funciones que deberán usarse cuando se llame a un endpoint. Para ello crearemos clases que heredaran de la clase que proporciona `rest_framework`. Estas nos permitirá definir clases y que cada clase tenga sus funciones `get`, `post`, `delete`, etc. A continuación vienen ejemplo de funciones obtener información (`get`) y creación, actualización o eliminado de información (`post` o `put`):

```

347 class UsuarioView(APIView):
348     def get(self, request, format=None):
349         usuario = autorizar_usuario(request, False)
350         if usuario == None:
351             return Response({"message": "Usuario no autorizado"}, status=status.HTTP_401_UNAUTHORIZED)
352
353         usuarios = Usuario.objects.all()
354
355         if request.query_params.get("email") is not None:
356             usuarios = usuarios.filter(email__icontains=request.query_params.get("email"))
357
358         if request.query_params.get("nombre") is not None:
359             usuarios = usuarios.filter(nombre__icontains=request.query_params.get("nombre"))
360
361         if request.query_params.get("apellidos") is not None:
362             usuarios = usuarios.filter(apellidos__icontains=request.query_params.get("apellidos"))
363
364         usuariosDTO = UsuarioObtenerDTO.toUsuarioObtenerDTO(usuarios)
365
366         serializer = UsuarioObtenerSerializer(usuariosDTO, many=True)
367
368         return Response(serializer.data, status=status.HTTP_200_OK, headers={"X-TOTAL-COUNT": usuarios.count()})

```

Figura 54: Ejemplo de función Get

Este sería un ejemplo estándar de un endpoint que lo que hace es coger información de base de datos, hace algún tipo de filtrado y devuelve la información. Por tanto se podría dividir el endpoint en 4 parte:

- 1º parte: sería hasta la línea 352, donde siempre certifica que el que pide el recurso es un usuario autorizado.
- 2º parte: sería la línea 353, donde se recoge la información de la base de datos
- 3º parte: sería de la línea 355 hasta la 362, está sería una parte opcional dependiendo de si se le ofrece el servicio de filtrar la información
- 4º parte: sería de la línea 364 hasta la 368, donde se le pasa la información al dto para que coja solo la información que queremos que se envíe, después se lo pasamos al serializer para que posteriormente lo podamos enviar en formato JSON. Para enviarlo hacemos uso de la clase Response la cual recibe:
 - JSON
 - STATUS
 - HEADERS

```

435 def post(self,request,format=None):
436     usuarioToken = autorizar_usuario(request,False)
437     if usuarioToken == None:
438         return Response({"mensaje":"Usuario no autorizado"},status=status.HTTP_401_UNAUTHORIZED)
439
440     serializer = HogarCrearSerializer(data=request.data)
441     if serializer.is_valid():
442         hogar = Hogar(
443             nombre = serializer.validated_data.get("nombre"),
444             potencia_contratada = serializer.validated_data.get("potencia_contratada"),
445             owner = usuarioToken
446         )
447         try:
448             hogar.save()
449             return Response(data={"id":hogar.id},status=status.HTTP_201_CREATED)
450         except:
451             return Response({"mensaje":"Error: El hogar no ha podido ser creado."},status=status.HTTP_400_BA
452     else:
453         return Response({"mensaje":"Error: El formato del hogar es incorrecto."},status=status.HTTP_400_BAD_
454

```

Figura 55: Ejemplo de función Post

Esta función sería un ejemplo de cómo se hace una función post en el sistema:

- 1º parte: sería hasta la línea 439, donde siempre certifica que el usuario que hace esta petición está autorizado
 - 2º parte: sería las línea 440 y 441 donde se certifica que lo que se ha enviado en el body es correcto.
 - 3º parte: sería lo que hay dentro del if, que lo se hace es coger la información y almacenarla en base de datos
- Urls.py

En este fichero serán donde se defina los endpoints y que función deben llamar cuando sean llamados. Para ello tendremos un array donde se guardaran estas referencias.

```

from django.urls import path

from api.views import CompartidoView, CompartidosIDView, DispositivoIDMedidaView, DispositivoIDView,

urlpatterns = [
    path("login/", LoginView.as_view()),
    path("usuarios/", UsuarioView.as_view()),
    path("usuarios/<int:id>", UsuarioIDView.as_view()),
    path("hogars/", HogarView.as_view()),
    path("hogars/<int:id>", HogarIDView.as_view()),
    path("hogars/<int:id>/compartidos", HogarsIDCompartidosView.as_view()),
    path("dispositivos/", DispositivosView.as_view()),
    path("dispositivos/<int:id>", DispositivoIDView.as_view()),
    path("dispositivos/<int:id>/medidas", DispositivoIDMedidaView.as_view()),
    path("medidas/", MedidaView.as_view()),
    path("compartidos/", CompartidoView.as_view()),
    path("compartidos/<int:id>", CompartidosIDView.as_view()),
    path("ofrecerInvitacion/", OfrecerInvitacionView.as_view()),
    path("invitacionesRecibidas/", InvitacionesRecibidasView.as_view()),
    path("invitaciones/<int:id>", InvitacionesIDView.as_view()),
    path("verificacionDispositivo/", DispositivosVerificacionView.as_view())
]

```

Figura 56: Fichero urls.py

Estos pasos son los seguidos para implementar la mayoría de requisitos que se han descrito. Pero para la realización de otros se necesita de más implementaciones que son las que siguen.

- Notificaciones

Para las notificaciones no se ha creado endpoints específicos sino que cada vez que ocurra alguna de las situaciones descritas en los requisitos de notificación y el usuario haya accedido a recibir notificaciones a su correo, el sistema mandará un correo al usuario. Para ello se ha usado la clase *EmailMultiAlternatives* que ofrece Django. Para su uso primero se ha tenido que añadir las siguientes línea en el fichero settings.py:

```
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_HOST_USER = "kproject2122@gmail.com"
EMAIL_HOST_PASSWORD = "dbgusgawimevcppu"
EMAIL_USE_TLS = True
```

Figura 57: Líneas para la configuración del correo

Una vez que se ha configurado para cada situación se ha creado un función que lo que hace es enviar un correo con el cuerpo correspondiente. Un ejemplo de la función es la siguiente:

```
def enviarCorreoAceptadoInvitacion(usuarioToken, hogarCompartido, invitante):
    html_content = """
    <h1>Hola, "" + invitante.nombre + ""</h1><br/>
    Este correo sirve para notificarle de que el usuario ""+usuarioToken.nombre+""
    (""+usuarioToken.email+"" ) ha aceptado la invitación para que pueda
    ver las estadísticas de ""+hogarCompartido.nombre+""<br/><br/>

    Recuerda que puede ver la lista de usuarios compartidos en:<br/><br/>
    <URL_KPROJECT><br/><br/>

    KProject.
    """

    message = EmailMultiAlternatives(
        subject="Invitación aceptada",
        body='',
        from_email=settings.EMAIL_HOST_USER,
        to=[invitante.email],
        cc=[]
    )

    message.attach_alternative(html_content, "text/html")
    message.send()
```

Figura 58: Foto función ejemplo que envía correo

- Seguridad

Como se asegura en los requisitos no funcionales las comunicaciones serán seguras, para ello se aparte del uso de HTTPS se crean tokens para los dispositivos para que así no cualquiera pueda insertar datos en la base de datos. Es por ello que se ha usado librería PyJwt que es capaz de crear tokens JWT. Por tanto cada vez que se crea un dispositivo en el paso 2 como se especifica anteriormente se envía el token del dispositivo. Su creación y envío es el siguiente:

```
objeto = {
    'id':dispositivo.id,
    'id_hogar': dispositivo.hogar.id,
    'id_owner':dispositivo.hogar.owner.id
}
token = jwt.encode(objeto,key=KEY_SECRET,algorithm=ALGORITMO_JWT[0])
return Response({"token" : token},status=status.HTTP_201_CREATED)
```

Figura 59: Código de cómo se crea y envía el token del dispositivo

Después para cuando se crea una medida hay otra función que lo hace es leer ese token y buscar el dispositivo por el id y si no está. La función:

```
def autorizar_dispositivo(request):
    if request.headers.get('Authorization') is not None:
        token = request.headers.get('Authorization')
        dispositivo = jwt.decode(token, key=KEY_SECRET, algorithms=ALGORITMO_JWT)
        try:
            return Dispositivo.objects.get(id=dispositivo.get('id'))
        except:
            return None
    else:
        return None
```

Figura 60: Código para autorizar un dispositivo

7.2. Módulo Frontend

Para la implementación de este módulo se ha usado la biblioteca de JavaScript React. Para su implementación se expondrá la estructura que tiene y cómo se ha seguido su creación. Su implementación tiene 3 partes bien diferenciadas:

- Fichero App.js

En esta parte es lo que la aplicación mostrará. Para poder cambiar entre pantallas se hará uso de la biblioteca Routes que da soporte para poder indicar que fichero debe llamar dependiendo del path en el que nos encontremos.

```
function App() {
  return (
    <div className="App">
      <BarraSup />
      <Routes>
        <Route path="/" element={<Home />}/>
        <Route path="/login" element={<Login />}/>
        <Route path="/valoresEnergia" element={<ValoresEnergia />}/>
        <Route path="/perfil" element={<Perfil />}/>
        <Route path="/registro/hogar" element={<Registro_Hogar />}/>
        <Route path="/hogar/:id" element={<Hogar />}/>
        <Route path="/edicionHogar/:id" element={<EdicionHogar />}/>
        <Route path="/addTarifa" element={<AddTarifa />} />
      </Routes>
    </div>
  );
}
```

Figura 61: App.js

- Carpeta components

En esta carpeta se encontrarán aquellas componentes que se puedan usar en todas las ventanas de la aplicación. En este caso tendremos 3 componentes:

- BarraSup.js: En este fichero definiremos la barra superior que será visible a través de todas las pantallas
- Loading.js: En este fichero se definirá un spinner que se usará cuando la aplicación cambie entre pantallas o necesite de tiempo para obtener datos del Backend.
- LoadingVentanaEmergente.js: en este fichero está definido un spinner que se usará en las pantallas emergentes para esperar los datos del Backend.

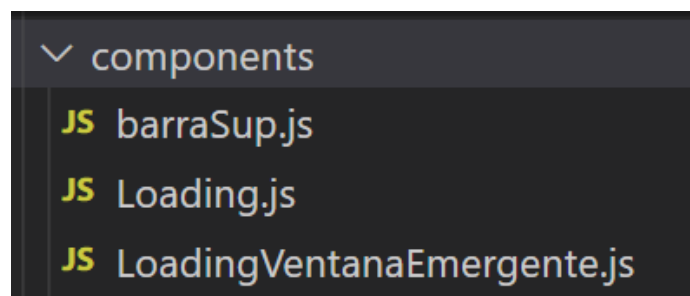


Figura 62: Carpeta components

- Carpeta web

En esta carpeta se encontraran los ficheros necesarios para todas las páginas definidas. En total tendremos 9 ficheros.

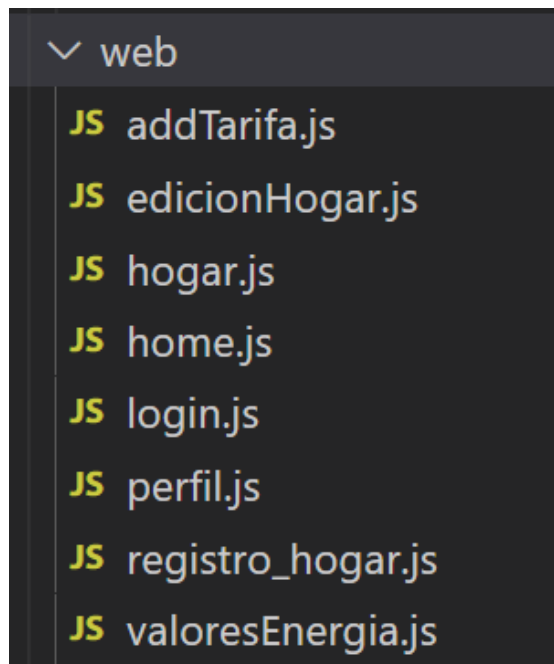


Figura 63: Carpeta web

El flujo que se ha seguido en cada vez que se accede a una pantalla es el siguiente, y con él se mostrará una serie de imágenes del login.js como ejemplo:

1. Se muestra que el sistema está cargando a través de la componente Loading.js

```
if(!loaded){  
  return <Loading />  
}
```

Figura 64: Ejemplo paso 1

2. El sistema, a través de la función useEffect, hace todas las operaciones previas antes de cargar la página.

```
useEffect(() => {  
  if(sessionStorage.getItem('token') != null){  
    window.location.replace("/")  
  }else{  
    isLoading(true)  
  }  
}, [])
```

Figura 65: Ejemplo paso 2

3. Una vez que el sistema ha hecho las comprobaciones previas se muestra la pantalla completa

```
<Container>
<Row>
  <Col><h3>Iniciar sesión con Smart meter</h3></Col>
</Row>
<Row>
  <GoogleLogin
    onSuccess={responseGoogleOnSuccess}
    onError={() => {
      console.log('Login Failed');
    }}
  />
</Row>
  {error !== "" && <p>{error}</p>}
</Container>
```

Figura 66: Ejemplo paso 3

7.3. Ejemplo de pasos al completo

En este subcapítulo se explican como son los pasos que sigue toda la aplicación desde que el usuario pide algo hasta que se le muestra dicha información. Para explicarlo se apoyara en el requisito de mostrar un hogar concreto. Suponemos que estamos ya en el home de la aplicación y ya ha iniciado sesión.

1. El usuario selecciona el desplegable hogares y selecciona un hogar. Esto lo que hace es llevarnos al path /hogar/<idHogar>
2. Una vez que llegamos a la página lo primero que se ejecuta como se ha comentado ya es el useEffect. En este caso:

```
132   useEffect(() =>{
133     var requestOptions = {
134       method: 'GET',
135       headers: { 'Authorization' : sessionStorage.getItem("token") }
136     };
137     fetch(process.env.REACT_APP_BASE_URL + "hogars/" + id, requestOptions).then
138     (response => response.json()).then
139     ((data) => {
140       setHogar(data)
141       setHayDispositivos(data.dispositivos.length != 0)
142       setDispositivos(data.dispositivos)
```

Figura 67: Ejemplo de useEffect ejemplo completo

3. Esto lo que hace es hace es una petición al Backend que como se muestra en la imagen es Get /hogars/id.

4. Una vez que hace la petición esta llega al Backend, está la redistribuye a la función correspondiente.
5. Cuando llega a la función correspondiente ejecuta toda la función que esta vez al ser un Get que es parecido al ejemplo de Get de usuario anteriormente mostrado.
6. Cuando devuelve los resultados el Frontend coge el JSON y lo guarda en unas variables para posteriormente mostrarlo como se ve en la figura 67 en la línea 140.
7. Una vez que se ha cogido la información se muestra la información en el Frontend

8

Pruebas

Las pruebas son una parte muy importante del desarrollo software. Estas sirven para obtener información objetiva sobre la calidad del producto. Debido a que el proyecto es extenso se han hecho pruebas manuales de sistema. Estas se tratan de evaluar el sistema al completo para ver si satisface todos los requisitos descritos.

Las pruebas que se han hecho son las descritas en los casos de pruebas que salen de todos los casos de uso. Dichos casos de pruebas han sido descritos en gherkin y son los siguientes.

Título	RF1
Caso de prueba	
Feature: Inicio de sesión de un usuario Una persona inicia sesión en la plataforma Background: Given el sistema carga la página de acceso And el usuario selecciona la opción de iniciar sesión And el sistema carga las distintas cuentas de Google que posee el usuario Scenario: Iniciar sesión con éxito. Given el usuario estaba registrado en el sistema. When el usuario selecciona una cuenta de Google Then el sistema carga la página de inicio. Scenario: El usuario inicia sesión sin estar registrado Given el usuario no estaba registrado en el sistema.	

When	el usuario selecciona una cuenta de Google
Then	el sistema notifica que se ha creado la cuenta con éxito
And	el sistema carga la página de inicio.

Título	RF2
Caso de prueba	
Feature: Cierre de sesión de un usuario Una persona cierra sesión en la plataforma Background: Given el usuario ha iniciado sesión Scenario: Cierra sesión con éxito When el usuario selecciona "Hola, ..." And el usuario pulsa "Cerrar sesión" Then el sistema carga la página de iniciar sesión	

Título	RF3
Caso de prueba	
Feature: Visualización de los datos personales de un usuario Una persona visualiza sus datos personales. Background: Given el usuario ha iniciado sesión Scenario: Visualizar los datos con éxito. When el usuario selecciona la opción "Hola, ..." And el usuario selecciona "Mi perfil" Then el sistema carga la página de visualización de datos del usuario.	

Título	RF4
Caso de prueba	
Feature: Modificación de los datos personales de un usuario Una persona modifica sus datos personales. Background: Given el usuario ha iniciado sesión Scenario: Modificar los datos con éxito. When el usuario selecciona la opción “Hola, ...” And el usuario selecciona la opción “Mi perfil” And el usuario cumplimenta los campos que desea modificar con un formato válido And el usuario selecciona la opción “Actualizar mis Datos” Then el sistema notifica que se han actualizado los datos con éxito And el sistema carga la página de visualización de datos actualizados del usuario. Scenario: Modificar los datos con éxito sin éxito. When el usuario selecciona la opción “Hola, ...” And el usuario selecciona la opción “Mi perfil” And el usuario cumplimenta los campos que desea modificar con un formato inválido And el usuario selecciona la opción “Actualizar mis Datos” Then el sistema notifica que se revisen los datos introducidos.	

Título	RF5
Caso de prueba	
Feature: Baja del usuario en la plataforma Un usuario se da de baja del sistema Background: Given el usuario ha iniciado sesión Scenario: Darse de baja con éxito. When el usuario selecciona la opción "Hola, ..." And el usuario selecciona la opción "Mi perfil" And el usuario selecciona "Darse de baja" And el sistema pregunta al usuario si está seguro And el usuario selecciona la opción "Sí". Then el sistema elimina al usuario And el sistema muestra la página de iniciar sesión.	

Título	RF6
Caso de prueba	
Feature: Registro de nuevos hogares. Un usuario desea registrar un nuevo hogar. Background: Given el usuario inicia sesión And el usuario selecciona "Hogares" And el usuario selecciona "Nuevo hogar" Scenario: Registro de hogar con éxito When el usuario cumplimenta los campos del registro correctamente And el usuario selecciona la opción "Añadir". Then el sistema guarda el registro del nuevo hogar y notifica que se ha registrado con éxito. Scenario: Deja el campo nombre vacío When el usuario cumplimenta los campos del registro pero deja el campo nombre vacío And el usuario selecciona la opción "Añadir". Then el sistema le informa que no puede dejar el nombre vacío.	

Título	RF7
Caso de prueba	
Feature: Visualizar datos del hogar. Un usuario desea visualizar datos del hogar. Background: Given el usuario inicia sesión And el usuario tiene registrado al menos un hogar. Scenario: Visualización de datos del hogar con éxito When el usuario selecciona “Hogares” And el usuario selecciona un hogar de la lista And el usuario selecciona “Editar” Then el sistema carga los datos del hogar seleccionado	

Título	RF8
Caso de prueba	
Feature: Modificación datos del hogar. Un usuario desea modificar datos de un hogar. Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar a modificar And el usuario es el dueño del hogar Scenario: Modificar datos hogar con éxito When el usuario selecciona la opción “Editar” And el sistema carga la página de editar hogar And el usuario cumplimenta los campos del registro And el usuario selecciona la opción “Actualizar hogar”. Then el sistema guarda la modificación del hogar y notifica que se ha actualizado con éxito. Scenario: Deja el nombre del hogar vacío When el usuario selecciona la opción “Editar” And el sistema carga la página de editar hogar And el usuario modifica los campos y deja el nombre libre And el usuario selecciona la opción “Actualizar hogar”.	

Then el sistema informa que el campo nombre no se puede dejar vacío

Scenario: Poner una palabra o frase en la potencia contratada

When el usuario selecciona la opción "Editar"

And el sistema carga la página de editar hogar

And el usuario modifica los campos e introduce una palabra o frase en el campo potencia contratada

And el usuario selecciona la opción "Actualizar hogar".

Then el sistema informa que el campo potencia contratada debe ser numérico

Título	RF9
--------	-----

Caso de prueba

Feature: Eliminación datos del hogar.

Un usuario desea eliminar datos de un hogar.

Background:

Given el usuario inicia sesión

And el usuario se encuentra en la sección del hogar a eliminar

And el usuario es el dueño del hogar

Scenario: Eliminar datos hogar con éxito

When el usuario selecciona la opción "Editar"

And el sistema carga la página de editar hogar

And el usuario selecciona la opción "Eliminar hogar"

And el sistema alerta al usuario si está seguro

And el usuario selecciona la opción "Sí"

Then el sistema elimina los datos del hogar y notifica que se ha eliminado con éxito.

Título	RF10
Caso de prueba	
Feature: Visualización de usuarios compartidos. Un usuario desea visualizar los usuarios compartidos en un hogar. Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar And el usuario es el dueño del hogar Scenario: Visualizar usuarios compartidos When el usuario selecciona la opción "Compartidos" Then el sistema muestra los usuarios con los que se ha compartido el hogar.	

Título	RF11
Caso de prueba	
Feature: Visualizar las invitaciones recibidas. Un usuario desea visualizar las invitaciones recibidas. Background: Given el usuario inicia sesión Scenario: Visualizar las invitaciones recibidas When el usuario selecciona la opción "Mis invitaciones" Then el sistema muestra las invitaciones recibidas	

Título	RF12
Caso de prueba	
Feature: Compartición de un contador inteligente con otro usuario. Un usuario comparte el contador con otro usuario. Background: Given el usuario inicia sesión And el usuario se encuentra en la página del hogar And el usuario es el dueño del hogar Scenario: Invitar al hogar con éxito When el usuario selecciona la opción “Compartidos” And el usuario selecciona la opción “Añadir” And el usuario introduce un correo de un usuario existente Then el sistema le informa que se ha enviado la invitación Scenario: Invitar a otro usuario que ya se ha invitado When el usuario selecciona la opción “Compartidos” And el usuario selecciona la opción “Añadir” And el usuario introduce un correo de un usuario existente pero que ya ha sido invitado anteriormente y no ha contestado Then el sistema le informa que a ese usuario ya se le ha invitado Scenario: Invitar a otro usuario que no existe When el usuario selecciona la opción “Compartidos” And el usuario selecciona la opción “Añadir” And el usuario introduce un correo de un usuario no existente Then el sistema le informa que ese usuario no existe	

Título	RF13
Caso de prueba	
Feature: Responde la invitación de un hogar compartido Un usuario responde la invitación de otro usuario Background: Given el usuario invitado inicia sesión And el usuario selecciona la opción “Mis invitaciones” Scenario: Acepta la invitación When el usuario selecciona la opción “Aceptar” Then el sistema guarda la compartición y notifica al usuario compartido que se ha unido con éxito. Scenario: Deniega la invitación When el usuario selecciona el icono “Rechazar” correspondiente. Then El sistema guarda el rechazo a la compartición del hogar y notifica al usuario invitado que se ha rechazado con éxito.	

Título	RF14
Caso de prueba	
Feature: Deja de compartir un hogar con otro usuario Un usuario deja de compartir un hogar con otro usuario Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar And el usuario es el dueño del hogar Scenario: Deja de compartir con éxito When el usuario selecciona la opción “Dejar de compartir” que hay al lado del usuario Then el sistema borra la compartición e informa que se ha dejado de compartir correctamente	

Título	RF15
Caso de prueba	
Feature: Deja de compartir un hogar con otro usuario Un usuario deja de compartir un hogar con otro usuario Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar And el usuario es un invitado del hogar Scenario: Abandona hogar con éxito When el usuario selecciona la opción “Abandonar hogar” que hay al lado del usuario And el sistema le informa que no hay marcha a tras And el usuario selecciona “Aceptar” Then el sistema borra la compartición e informa que se ha dejado de compartir correctamente	

Título	RF17
Caso de prueba	
Feature: Visualizar la información de un contador inteligente Un usuario visualiza la información del contador inteligente Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar And el usuario es dueño del hogar Scenario: Visualiza los datos correctamente When el usuario selecciona la opción “Editar” And el usuario selecciona el contador inteligente que quiere visualizar Then el sistema muestra los datos del contador inteligente	

Título	RF18
Caso de prueba	
Feature: Modificar los datos de un contador inteligente Un usuario modifica los datos de un contador inteligente Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar And el usuario es dueño del hogar Scenario: Modifica los datos correctamente When el usuario selecciona la opción "Editar" And el usuario selecciona el contador inteligente And el usuario introduce los datos correctamente Then el sistema le informa que se ha actualizado correctamente Scenario: Deja vacío el campo nombre When el usuario selecciona la opción "Editar" And el usuario selecciona el contador inteligente And el usuario introduce los datos pero deja vacío el nombre Then el sistema le informa que no se puede dejar vacío Scenario: Introduce una palabra o frase en el campo límite máximo When el usuario selecciona la opción "Editar" And el usuario selecciona el contador inteligente And el usuario introduce los datos pero introduce una palabra o frase en el campo límite máximo Then el sistema le informa que no puede insertar eso en ese campo	

Título	RF19
Caso de prueba	
Feature: Eliminar un contador inteligente Un usuario quiere eliminar un contador inteligente Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar And el usuario es dueño del hogar Scenario: Eliminar el contador inteligente correctamente When el usuario selecciona la opción "Editar" And el usuario selecciona el contador inteligente And el usuario selecciona el icono de basura And el sistema le pregunta si está seguro And el usuario selecciona "Aceptar" Then el sistema lo borra y se lo comunica al usuario	

Título	RF20
Caso de prueba	
Feature: Visualizar la estadística de los KWh de un contador inteligente Un usuario quiere visualizar la estadística de un contador inteligente Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar And el usuario es dueño o se lo han compartido Scenario: Visualiza la estadística When el usuario selecciona el contador inteligente Then el sistema le muestra la estadística	

Título	RF21
Caso de prueba	
Feature: Visualizar la estadística del dinero gastado de un contador inteligente Un usuario quiere visualizar la estadística de un contador inteligente Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar And el usuario es dueño o se lo han compartido Scenario: Visualiza la estadística When el usuario selecciona el contador inteligente Then el sistema le muestra la estadística	

Título	RF22
Caso de prueba	
Feature: Filtrar los datos por fecha de un contador inteligente Un usuario quiere filtrar por las fechas Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar And el usuario es dueño o se lo han compartido Scenario: Filtra correctamente When el usuario selecciona el contador inteligente And el usuario selecciona "Filtrar por fechas" And el usuario introduce las fechas And el usuario selecciona "Filtrar" Then el sistema le muestra los datos filtrados Scenario: Introduce la fecha máxima inferior a la mínima When el usuario selecciona el contador inteligente And el usuario selecciona "Filtrar por fechas" And el usuario introduce la fecha máxima menor que mínima And el usuario selecciona "Filtrar" Then el sistema le informa que la fecha máxima no puede ser menor que la mínima	

Título	RF23
Caso de prueba	
Feature: Filtrar los datos por valores de un contador inteligente Un usuario quiere filtrar por los valores Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar And el usuario es dueño o se lo han compartido Scenario: Filtra correctamente When el usuario selecciona el contador inteligente And el usuario selecciona "Filtrar por Valores máximo y mínimo" And el usuario introduce los valores And el usuario selecciona "Filtrar" Then el sistema le muestra los datos filtrados Scenario: Introduce el valor máximo menor que el mínimo When el usuario selecciona el contador inteligente And el usuario selecciona "Filtrar por Valores máximo y mínimo" And el usuario introduce el valor máximo menor que el mínimo And el usuario selecciona "Filtrar" Then el sistema le informa que el valor máximo no puede ser menor que el mínimo Scenario: Introduce texto en los campos When el usuario selecciona el contador inteligente And el usuario selecciona "Filtrar por Valores máximo y mínimo" And el usuario introduce palabra o frases And el usuario selecciona "Filtrar" Then el sistema le informa que esos valores no son validos	

Título	RF24
Caso de prueba	
Feature: Comparar datos de 2 días cualesquiera Un usuario quiere comparar los datos de 2 días Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar And el usuario es dueño o se lo han compartido Scenario: Compara los 2 días correctamente When el usuario selecciona el contador inteligente And el usuario selecciona "Comparar días" And el usuario introduce las fechas And el usuario selecciona "Comparar" Then el sistema le muestra las gráficas de los 2 días	

Título	RF25
Caso de prueba	
Feature: Visualizar los costes de un día en concreto Un usuario quiere visualizar los costes de un día Background: Given el usuario inicia sesión And el usuario está en el home Scenario: Visualiza los costes de un día elegido When el usuario selecciona la fecha And el usuario selecciona "Buscar" Then el sistema le muestra la figura con los costes de ese día	

Título	RF26
Caso de prueba	
Feature: Visualizar las tarifas de otras compañías Un usuario puede visualizar las tarifas de otras compañías Background: Given el usuario inicia sesión And el usuario está en el home Scenario: Visualiza las tarifas de una compañía When el usuario selecciona "Valores de energía" And el usuario selecciona "Tarifa empresas" And el usuario selecciona la empresa que quiere visualizar Then el sistema le muestra las tarifas de esa empresa	

Título	RF27
Caso de prueba	
Feature: Añadir una tarifa al sistema Un usuario puede añadir una tarifa al sistema Background: Given el usuario inicia sesión And el usuario está en el home Scenario: Introduce correctamente la tarifa When el usuario selecciona "Valores de energía" And el usuario selecciona "Tarifa empresas" And el usuario selecciona "Nueva tarifa" And el usuario introduce los datos correctamente And el usuario selecciona "Nueva tarifa" Then el sistema la guarda y le informa que se ha guardado Scenario: No introduce la tarifa por introducir datos erróneos When el usuario selecciona "Valores de energía" And el usuario selecciona "Tarifa empresas" And el usuario selecciona "Nueva tarifa" And el usuario introduce datos incorrectamente And el usuario selecciona "Nueva tarifa" Then el sistema le informa que los datos son incorrecto	

Título	RF28
Caso de prueba	
Feature: Ver la media de precios por hora de la última semana Un usuario puede ver la media de la última semana Background: Given el usuario inicia sesión And el usuario está en el home Scenario: Ver la media correctamente When el usuario selecciona “Valores de energía” And el usuario selecciona “Media de la última semana” Then el sistema le muestra en una figura los datos	

Título	RF29
Caso de prueba	
Feature: Ver la media de precios por hora del último mes Un usuario puede ver la media del último mes Background: Given el usuario inicia sesión And el usuario está en el home Scenario: Ver la media correctamente When el usuario selecciona “Valores de energía” And el usuario selecciona “Media del último mes” Then el sistema le muestra en una figura los datos	

Título	RF30
Caso de prueba	
Feature: Ver la media de precios por hora de los últimos 3 meses Un usuario puede ver la media de los últimos 3 meses Background: Given el usuario inicia sesión And el usuario está en el home Scenario: Ver la media correctamente When el usuario selecciona "Valores de energía" And el usuario selecciona "Media de los últimos 3 meses" Then el sistema le muestra en una figura los datos	

Título	RF31
Caso de prueba	
Feature: Configurar las notificaciones sobre un contador inteligente Un usuario configura las notificaciones sobre un contador inteligente Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar And el usuario es dueño del hogar Scenario: Acepta la recepción de notificaciones When el usuario selecciona la opción "Editar" And el usuario selecciona el contador inteligente And el usuario selecciona "Si" en contadores inteligentes And el usuario selecciona "Actualizar contador inteligente" Then el sistema le informa que ha sido configurado correctamente Scenario: Deniega la recepción de notificaciones When el usuario selecciona la opción "Editar" And el usuario selecciona el contador inteligente And el usuario selecciona "No" en contadores inteligentes And el usuario selecciona "Actualizar contador inteligente" Then el sistema le informa que ha sido configurado correctamente	

Título	RF32
Caso de prueba	
Feature: Configurar el valor máximo diario a notificar en el contador inteligente Un usuario configura las notificaciones sobre un contador inteligente Background: Given el usuario inicia sesión And el usuario se encuentra en la sección del hogar And el usuario es dueño del hogar Scenario: Modifica el valor máximo correctamente When el usuario selecciona la opción "Editar" And el usuario selecciona el contador inteligente And el usuario introduce el valor máximo And el usuario selecciona "Actualizar contador inteligente" Then el sistema le informa que ha sido configurado correctamente Scenario: Introduce un valor incorrecto When el usuario selecciona la opción "Editar" And el usuario selecciona el contador inteligente And el usuario introduce un valor incorrecto And el usuario selecciona "Actualizar contador inteligente" Then el sistema le informa que ese valor no es correcto	

Título	RF33
Caso de prueba	
Feature: Configurar la recepción de notificaciones sobre las invitaciones Un usuario desea configurar la recepción de notificaciones sobre las invitaciones Background: Given el usuario inicia sesión And el usuario se encuentra en su perfil Scenario: Acepta la recepción de notificaciones When el usuario selecciona "Sí" And el usuario selecciona "Actualizar" Then el sistema informa que se configurado correctamente Scenario: Deniega la recepción de notificaciones When el usuario selecciona "No" And el usuario selecciona "Actualizar" Then el sistema informa que se configurado correctamente	

Conclusiones y líneas futuras

La realización de este Trabajo de Fin de Grado ha dado como resultado la realización de un sistema, cuyos propósitos son:

- Dar soporte a los usuarios del sistema para poder ver las estadísticas históricas que se recogen de los contadores inteligentes.
- Dar soporte a los diferentes trabajos que necesiten de esta información en sus respectivos trabajos del proyecto K-Project.
- Dar soporte a los contadores inteligentes para poder guardar la información que recogen.

Después de la realización de este trabajo, se puede concluir que el desarrollo seguido sobre la metodología expuesta contra las demás ha sido un acierto debido a la naturaleza que tiene dicho trabajo. Además, tanto la arquitectura, como la especificación, como la implementación se ajustan a los propósitos generales que requieren el sistema y todo ello siguiendo un proceso documentado y dinámico que se ajusta a los objetivos generales del proyecto más grande en el cual está envuelto este trabajo.

Por tanto se puede decir que se han cumplido los objetivos del proyecto descritos en el anteproyecto y aplicar todos los conocimientos adquiridos durante los años en el grado. Algunas de las posibles mejoras y ampliaciones que tiene el trabajo y que se podrían hacer en los sucesivos meses ya que el proyecto no termina aquí serán:

- La base de datos está hecha en MongoDB pero esta tecnología tiene el problema que tiene buena escalabilidad hasta 100GB y como está pensado el proyecto, seguramente sobrepase esa cantidad de datos, por tanto una buena opción sería cambiar de base de datos.
- Mejorar la seguridad: es verdad que hay una aproximación para que los dispositivos tengan tokens y así nadie intente meter datos falsos en la aplicación. El problema es que estos token ahora mismo son perpetuos y por tanto, una posible mejora sería el uso de tokens que expiren para una mejor seguridad.
- Añadir estadísticas relacionadas con el precio: en estos momento la aplicación te permite ver estadísticas de tu contador inteligente de los kwh y poder filtrarlos pero no permite el filtrado y ver gráficas de lo gastado por tus contadores inteligentes. Esto sería algo a tener en cuenta.
- Si los usuarios lo permiten hacer uso de sus datos para poder mostrar y generar estadísticas relacionados a sus contadores inteligentes. Ejemplos:
 - ¿Cuál es el consumo de una persona que vive en Málaga en agosto a las 3 de la tarde?
 - ¿Cuáles son las comunidades/ciudades autónomas que menos consumo tiene?
- Si concreta lo anterior y usa datos sensible seguir con lo dispuesto en el Reglamento General de Protección de Datos.
- Notificaciones sobre el dinero gastado: en estos momentos se permite una notificación sobre los kwh gastado pero sería interesante que esto se trasladara también al dinero gastado por el contador inteligente.
- Permitir que las invitaciones se compartan en algunas redes sociales
- Las pruebas realizadas son manuales por tanto en el futuro estaría bien hacer pruebas unitarias y de integración

En cuanto a lo vivido, creo que ha sido y está siendo una experiencia muy beneficiosa, porque nunca había hecho un proyecto de estas características, donde tenga que hacer un proyecto entero con mi compañero y además tengamos que comunicarnos con otros para hacer que dicho proyecto funcione y tenga conexión entre los diversos trabajos.

El cambio de trabajar con las personas que tú conoces durante mi periodo en el grado, a trabajar con personas que no conocía, te enseña cómo se trabaja en realidad y amoldarte a todo tipo de situaciones que antes no me habían ocurrido. Además de que cada 2 semanas tengamos una reunión donde tengamos que tener algo hecho y poder enseñarle a los profesores creo que es algo que ha ayudado al flujo del trabajo y que en muchos trabajos también se reflejan cuando usan metodologías ágiles.

Además, las tecnologías usadas creo que están al orden del día y por tanto me sirven para crecer como ingeniero. Aunque es verdad que muchas de las tecnologías ya la había usado en la carrera, este trabajo me ha servido para profundizar en muchas de ellas y conocer a alguna que otra como puede ser JWT o Python que no se enseñan en el grado pero gracias a este trabajo he adquirido conocimientos.

Referencias

- [1] "KProject," [Online]. Available: <https://www.link.uma.es/k-project/>. [Accessed 11 06 2022].
- [2] A. Pérez, "La factura eléctrica de los españoles subirá 475 euros entre 2021 y 2022," 05 01 2022. [Online]. [Accessed 11 06 2022].
- [3] "Python," Wikipedia, la enciclopedia libre, 09 06 2022. [Online]. Available: <https://es.wikipedia.org/w/index.php?title=Python&oldid=135459242>. [Accessed 11 06 2022].
- [4] "Django (framework)," Wikipedia, la enciclopedia libre, 10 06 2022. [Online]. Available: [https://es.wikipedia.org/wiki/Django_\(framework\)](https://es.wikipedia.org/wiki/Django_(framework)). [Accessed 11 06 2022].
- [5] "MongoDB," Wikipedia, la enciclopedia libre, 01 05 2022. [Online]. Available: <https://es.wikipedia.org/wiki/MongoDB>. [Accessed 11 06 2022].
- [6] "MongoDB Atlas Tutorial," MongoDB, [Online]. Available: <https://www.mongodb.com/basics/mongodb-atlas-tutorial>. [Accessed 11 06 2022].
- [7] "JavaScript," Wikipedia, la enciclopedia libre, 09 06 2022. [Online]. Available: <https://es.wikipedia.org/wiki/JavaScript>. [Accessed 11 06 2022].
- [8] "React," Wikipedia, la enciclopedia libre, 22 05 2022. [Online]. Available: <https://es.wikipedia.org/wiki/React>. [Accessed 11 06 2022].
- [9] "Bootstrap (framework)," Wikipedia, la enciclopedia libre, 14 05 2022. [Online]. Available: [https://es.wikipedia.org/wiki/Bootstrap_\(framework\)](https://es.wikipedia.org/wiki/Bootstrap_(framework)). [Accessed 11 06 2022].
- [10] "Visual Studio Code," Wikipedia, la enciclopedia libre, 09 06 2022. [Online]. Available: https://es.wikipedia.org/wiki/Visual_Studio_Code. [Accessed 11 06 2022].
- [11] "Git," Wikipedia, la enciclopedia libre, 01 05 2022. [Online]. Available: <https://es.wikipedia.org/wiki/Git>. [Accessed 11 06 2022].
- [12] "GitHub," Wikipedia, la enciclopedia libre, 23 05 2022. [Online]. Available: <https://es.wikipedia.org/wiki/GitHub>. [Accessed 11 06 2022].
- [13] "MagicDraw UML," Wikipedia, la enciclopedia libre, 4 09 2019. [Online]. Available: https://es.wikipedia.org/wiki/MagicDraw_UML. [Accessed 11 06 2022].
- [14] "JSON Web Token," Wikipedia, la enciclopedia libre, 29 05 2022. [Online]. Available: https://es.wikipedia.org/wiki/JSON_Web_Token. [Accessed 11 06 2022].

- [15] "Swagger (software)," Wikipedia, la enciclopedia libre, 09 10 2021. [Online]. Available: [https://es.wikipedia.org/wiki/Swagger_\(software\)](https://es.wikipedia.org/wiki/Swagger_(software)). [Accessed 11 06 2022].
- [16] "Heroku," Wikipedia, la enciclopedia libre, 07 10 2021. [Online]. Available: <https://es.wikipedia.org/wiki/Heroku>. [Accessed 11 06 2022].
- [17] "IFML: The Interaction Flow Modeling Language," Object Management Group, Inc, [Online]. Available: <https://www.ifml.org/>. [Accessed 11 06 2022].
- [18] "IFMLEdit," IFML, [Online]. Available: <https://www.ifmledit.org/>. [Accessed 11 06 2022].
- [19] "Protocolo de transferencia de hipertexto," Wikipedia, la enciclopedia libre, 31 05 2022. [Online]. Available: https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto. [Accessed 11 06 2022].
- [20] "Proteger sitios con el protocolo HTTPS," Developers Google, [Online]. Available: <https://developers.google.com/search/docs/advanced/security/https?hl=es>. [Accessed 11 06 2022].
- [21] "JSON," Wikipedia, la enciclopedia libre, 27 05 2022. [Online]. Available: <https://es.wikipedia.org/wiki/JSON>. [Accessed 11 06 2022].
- [22] "¿Qué es una API de REST?," Red Hat, 08 05 2020. [Online]. Available: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>. [Accessed 11 06 2022].

Apéndice A

Manual de Instalación

A.1.Requerimientos

Para poder Instalar el sistema entero en local se necesita:

- Para ambas partes:
 - Tener instalado Git
- Para Backend
 - Tener instalado Python 3.9.0
 - Tener instalado pip
- Para Frontend
 - Tener instalado Node.js

A.2.Instalación Backend

- 1) Abrir la consola con git en la carpeta donde se vaya a descargar

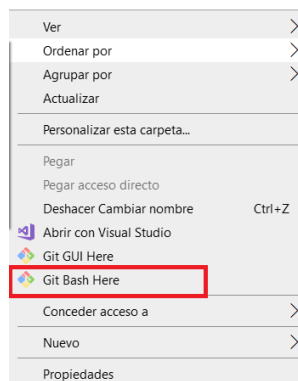
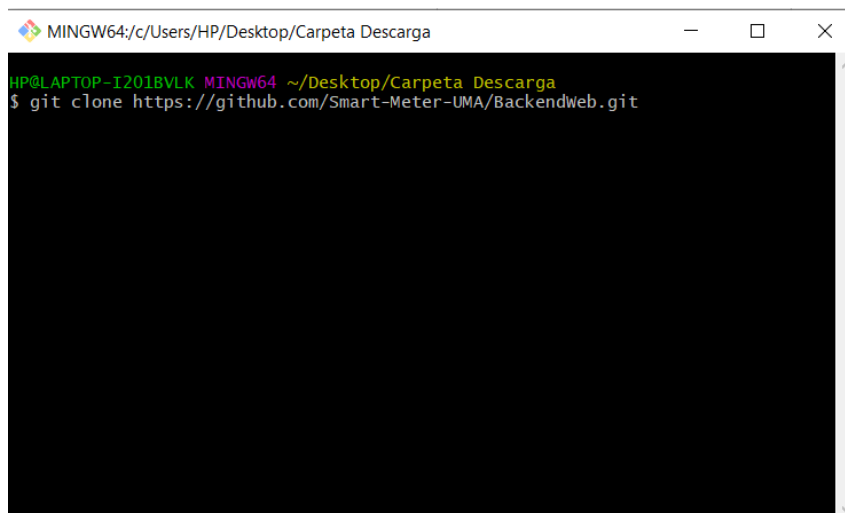


Figura 68: Foto paso 1 instalación Backend

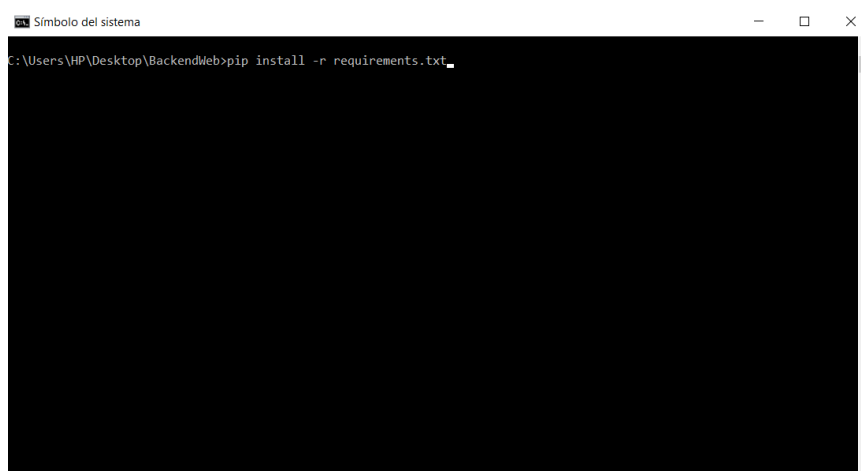
- 2) Usar el comando `git clone https://github.com/Smart-Meter-UMA/BackendWeb.git`



```
MINGW64:/c:/Users/HP/Desktop/Carpeta Descarga
HP@LAPTOP-I201BVLK MINGW64 ~/Desktop/Carpeta Descarga
$ git clone https://github.com/Smart-Meter-UMA/Backendweb.git
```

Figura 69: Foto paso 2 instalación Backend

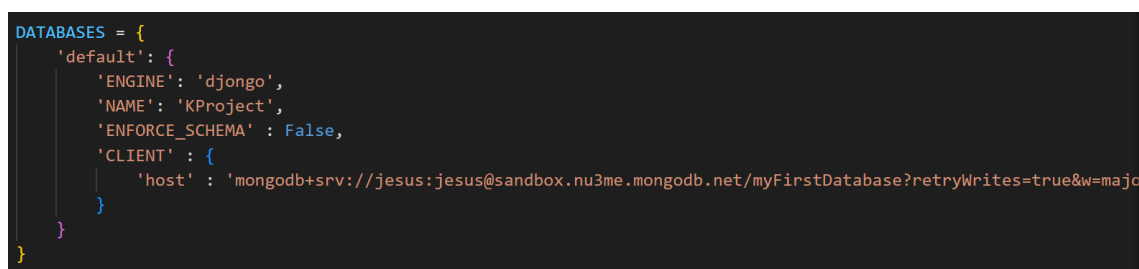
- 3) Abrir una consola en la carpeta donde hayamos descargado el código
- 4) Usar el comando *pip install -r requirements.txt*



```
Símbolo del sistema
C:\Users\HP\Desktop\Backendweb>pip install -r requirements.txt
```

Figura 70:Foto paso 4 instalación Backend

- 5) Si se desea cambiar de base de datos remota, habría que ir al fichero settings.py y en la parte de "DATABASES" habría que cambiar el host del cliente.



```
DATABASES = {
    'default': {
        'ENGINE': 'django',
        'NAME': 'KProject',
        'ENFORCE_SCHEMA' : False,
        'CLIENT' : {
            'host' : 'mongodb+srv://jesus:jesus@sandbox.nu3me.mongodb.net/myFirstDatabase?retryWrites=true&w=majo
        }
    }
}
```

Figura 71: Foto del paso 5 instalación Backend

- 6) Sí se ha cambiado el host de la base de datos o el nombre se debe volver a migrar el modelo a la base de datos. Los comandos serían

- a. *python manage.py makemigrations*
- b. *python manage.py migrate*

7) Para iniciar el servidor usar el comando *python manage.py runserver*, este se desplegará en <http://localhost:8000>

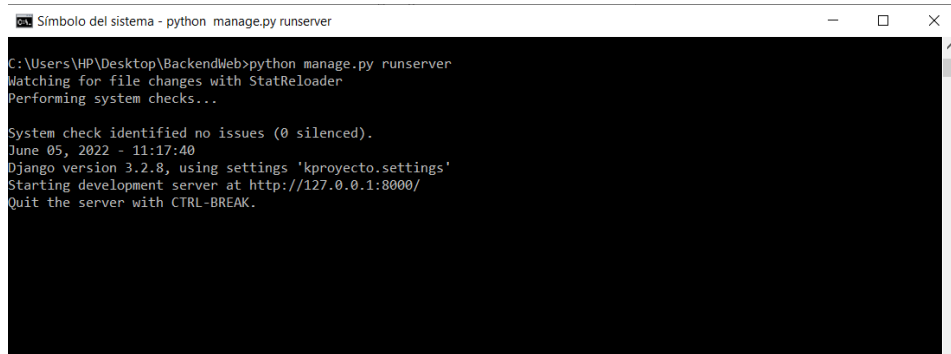


Figura 72: Foto paso 5 instalación Backend

8) Opcional, si se desea desplegar en otro puerto usar el comando *python manage.py runserver 127.0.0.1:puerto*



Figura 73: Foto paso 6 opcional de instalación Backend

A.3.Instalación Frontend

- 1) Abrir la consola con git en la carpeta donde se vaya a descargar
- 2) Usar el comando *git clone https://github.com/Smart-Meter-UMA/frontend-web.git*
- 3) Abrir una consola en la carpeta donde hayamos descargado el código
- 4) Usar el comando *npm install*

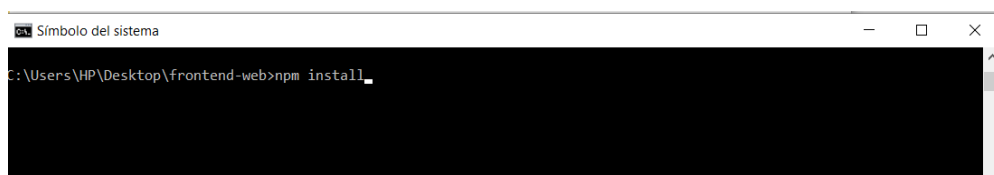


Figura 74: Foto paso 4 instalación Frontend

- 5) Para iniciar el servidor usar el comando *npm start*, este se desplegará en <http://localhost:3000>



Figura 75: Foto paso 5 instalación Frontend

Apéndice B

Manual de usuario

En este apartado se va a exponer cual sería flujo normal de uso de la aplicación web

- 1) Una vez que entras en la aplicación el sistema te redirige a /login

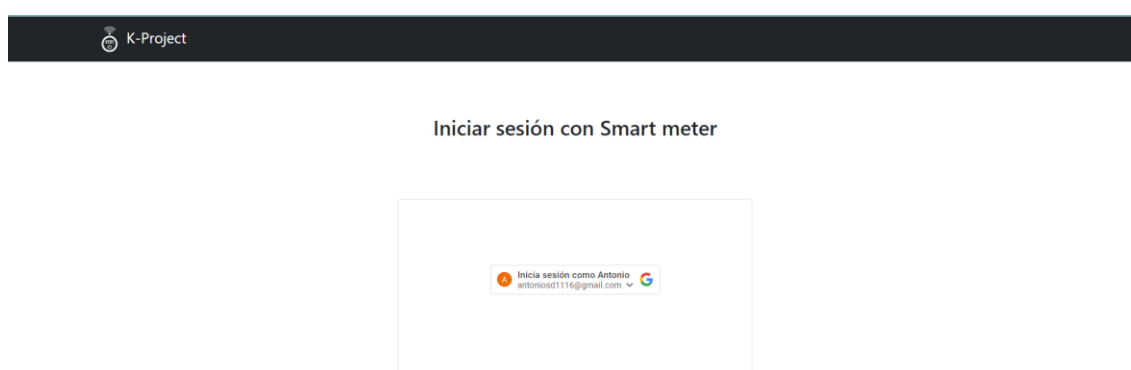


Figura 76: Pantalla de login

- 2) Una vez que inicias sesión te lleva a la pantalla de home donde te muestra los precios de la luz hoy
- 3) En la parte superior de la derecha se tiene 2 opciones:

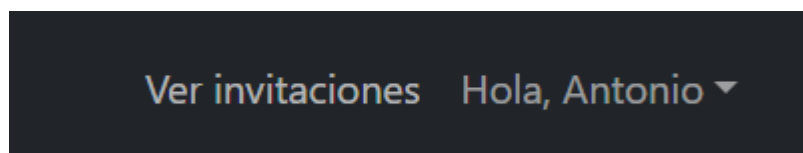


Figura 77: Opciones de la parte superior derecha

- a. "Mis invitaciones" que sirve para la gestión de las invitaciones que recibes. Este te abre una ventana emergente con las invitaciones que te han mandado para aceptarlas o rechazarlas

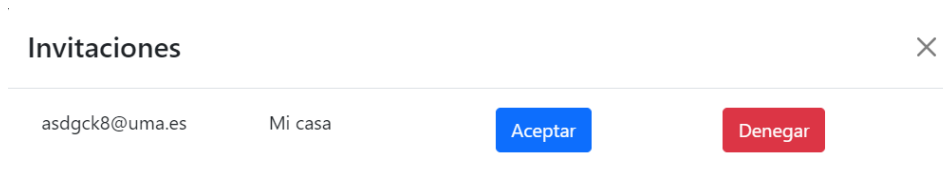


Figura 78: Ventana emergente de mis invitaciones

b. "Hola, ..." que es un desplegable donde tenemos dos opciones:

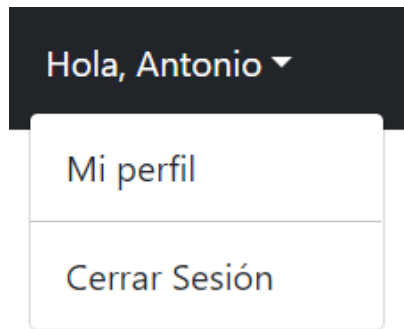


Figura 79: Desplegable Hola, ...

i. "Mi perfil" donde puedes modificar tu perfil y darte de baja

Figura 80: Ventana de mi perfil

ii. "Cerrar sesión" que sirve para cerrar sesión de la aplicación

4) En la parte superior izquierda tenemos "Hogares" el cual es un desplegable que tiene 2 partes diferenciadas por un separador.

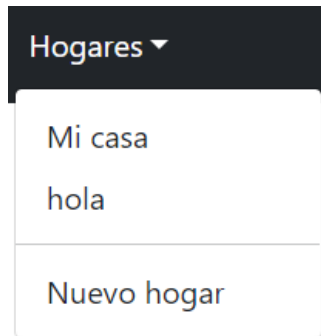


Figura 81: Desplegable "Hogares"

- a. La primera parte donde tenemos todos tus hogares tanto creados como invitados
- b. La segunda parte donde se puede crear un nuevo hogar en la opción "Nuevo hogar" que te redirige a un formulario para poder crear un hogar.

A dark grey header bar at the top contains the 'K-Project' logo and navigation links: 'Hogares', 'Valores de energía', 'Ver invitaciones', and 'Hola, Antonio'. Below the header, the main content area has the title 'Registre su nuevo hogar'. Underneath the title are two form fields: 'Nombre:' with an empty text input box, and 'Potencia contratada (KW):' with a text input box containing the number '0'. At the bottom of the form is a blue button with the text 'Añadir hogar'.

Figura 82: Ventana registrar hogar

- 5) Una vez que seleccionas un hogar, el hogar puede ser:
- a. Seas dueño del hogar: en este caso tendrá 2 opciones que serán

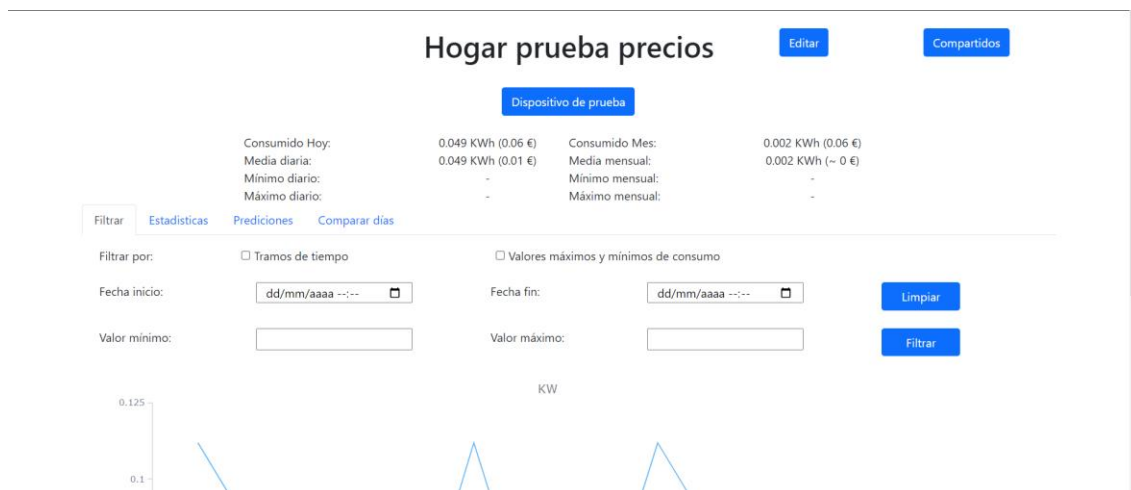


Figura 83: Hogar del que eres dueño

- i. "Editar" donde te redirige a un formulario para poder modificar la información tanto del hogar como de los dispositivos asociados

The screenshot shows the 'Hogar: Hogar prueba precios' edit form. At the top, there is a header with 'K-Project', 'Hogares', 'Valores de energía', 'Ver invitaciones', and 'Hola, Antonio'. The form has a title 'Hogar: Hogar prueba precios' and fields for 'Nombre:' (Hogar prueba precios) and 'Potencia contratada (KW):' (1000). There is an 'Actualizar hogar' button. Below this is a 'Smart meters' section with a 'Dispositivo de prueba' button. The 'Smart meters' section includes fields for 'Nombre:' (Dispositivo de prueba), 'General:' (radio buttons for 'Sí' and 'No', with 'No' selected), 'Notificaciones' (radio buttons for 'Sí' and 'No', with 'No' selected), 'Límite mínimo:' (0), and 'Límite máximo:' (0). There is also a 'Configuración dispositivo' section.

Figura 84: Editar un hogar

- ii. "Compartir" el cual te abre una ventana emergente con todos los usuarios que has invitado y han aceptado dicha invitación. Aquí podrás dejar de compartir el hogar en "Dejar de compartir" y podrás invitar a gente en "Invitar" que te muestra otra ventana donde indicar el correo del usuario que quieres invitar.

The screenshot shows the 'Compartidos' window. It has a title 'Compartidos' and a close button 'X'. There is a list of users, with 'yisusr3kt@gmail.com' visible. There are buttons for 'Invitar' and 'Dejar de compartir'.

Figura 85: Ventana de usuarios compartidos

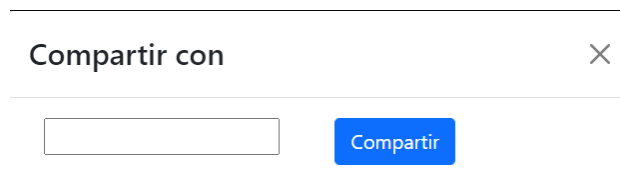


Figura 86: Ventana para compartir

- b. No eres dueño del hogar: en este caso solo tenemos una opción "Abandonar hogar" con el cual puedes dejar de estar invitado a ese hogar cuando se selecciona

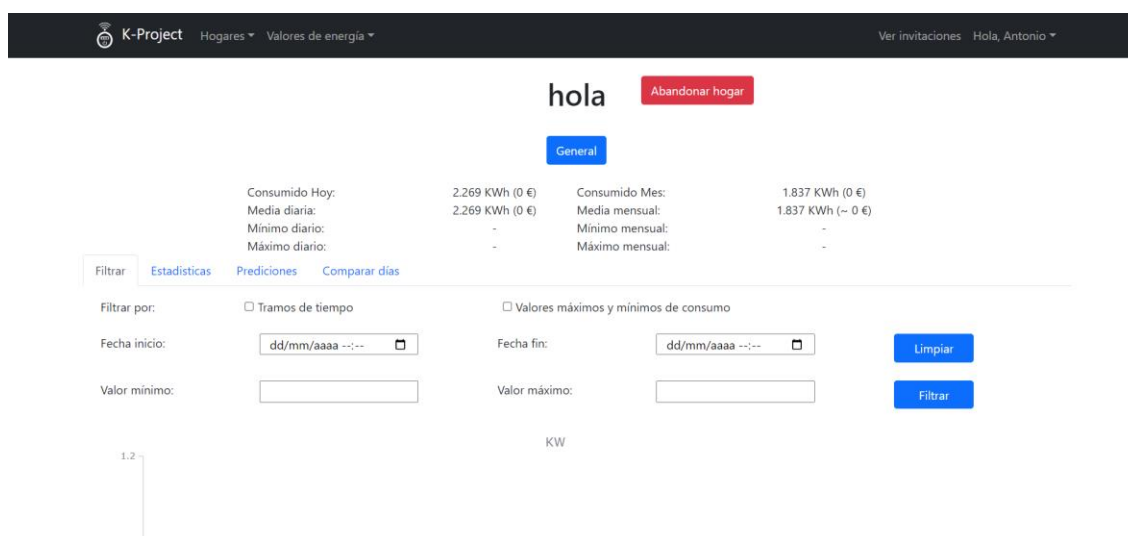


Figura 87: Hogar cuando soy un invitado

En ambos casos tendrás 2 pestaña:

- a. Filtrar: en esta pestaña podrás filtrar por fecha (con fecha inicio y fecha fin) y por kw (con valor mínimo y valor máximo).
- b. Comparar Días: en esta pestaña tendrás la opción de elegir los 2 días que quieres comparar:



Figura 88: Imagen pestaña comparar días

Una vez que se seleccione los días se mostrar una ventana emergente con las gráficas pertinentes

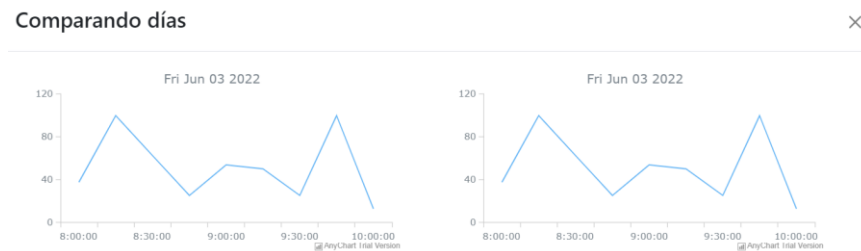


Figura 89: Imagen ventana emergente comparación de días

6) Al seleccionar "Valores de energía" se nos despliega una lista de opciones:

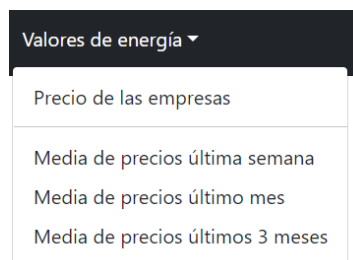


Figura 90: Lista de opciones valores de energía

- a. Precio de las empresas: esta opción nos llevará a otra página donde podremos ver las tarifas que hay registradas de cada empresa

Tipo factura	Precio llano	Precio alto	Precio día	Fecha de actualización
tarifacompromiso(p3)	-	-	0.15€/KWh	2022-06-07
tarifaporuso(p3)	-	-	0.23€/KWh	2022-06-07
tarifanoche(p3)	0.3€/KWh	0.24€/KWh	-	2022-06-07

Figura 91: Pantalla tarifa de las empresas

A parte de elegir la empresa para ver sus tarifas, aquí también podrás añadir tu tarifa al sistema. Esto se hace seleccionando "Añadir tarifa" que te lleva a otra página con un formulario:

Nueva tarifa¹

Comercializadora:	
Nombre de la tarifa:	
Potencia periodo punta:	
Potencia periodo valle:	
Precio llano/día:	
Precio alto:	

Nueva tarifa

Figura 92: Pantalla para añadir una tarifa

- Media de los precios de la última semana: esta opción te mostrará una pestaña emergente con la media de los precios de la última semana
- Media de los precios del último mes: esta opción te mostrará una pestaña emergente con la media de los precios del último mes
- Media de los precios de los últimos 3 meses: esta opción te mostrará una pestaña emergente con la media de los precios de los últimos 3 meses



Figura 93: Ventana ejemplo en cuanto a la media de €/KWh



UNIVERSIDAD
DE MÁLAGA | **uma.es**

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga