



Desarrollo de un asistente personal para la gestión del consumo eléctrico doméstico

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Antonio Martínez Álvarez

Tutor: Javier García Guzmán

Campus de Colmenarejo, 26 de Septiembre de 2017

Contenido

Índice de ilustraciones	6
Resumen	8
Abstract	9
Introduction	10
Motivation.....	10
Problem statement	11
Objectives.....	11
Work method	12
Resources used.....	14
Description of the application and results	15
Architecture design	15
Description of the functionalities.....	16
Test results. System Test Plan	16
Conclusions and future work	17
1.-Introducción	20
1.1.- Motivación	20
1.2.- Planteamiento del problema	21
1.3.- Objetivos	22
1.4.- Método de trabajo y recursos utilizados	23
1.4.1- Fase de Planificación	23
1.4.2.- Fase de Investigación	24
1.4.3.- Fase de Desarrollo.....	24
1.4.4.- Fase de pruebas	25
1.4.5.- Fase de documentación	25
1.4.6.- Recursos utilizados.....	26
1.5.- Introducción al resto de contenidos del documento.....	27
2.-Estado del arte	29
2.1.-Tarifas eléctricas en España	29
2.2.- Android.....	30

2.2.1.-Características y especificaciones de Android	30
2.2.2.-Android Studio	31
2.2.3.- Eclipse con plugin de Herramientas de Desarrollo Android	33
2.3.- Breve Análisis de Sistemas Gestores de Bases de Datos	33
2.3.1.- SGBD elegido para este proyecto	34
2.4.- Entorno socioeconómico del proyecto	34
2.4.1.- Análisis del entorno socioeconómico.....	34
2.4.2.- Evolución de las tarifas eléctricas en los últimos años y tendencia futura.....	41
2.4.3- Conclusión al análisis del entorno socioeconómico	42
2.5.- Marco regulador	43
2.5.1.- Aspectos legales generales	43
2.5.2.- Legislación concreta.....	44
2.5.3.- Herramientas utilizadas	44
2.5.4.- Servicios de la web de Esios.....	45
2.6.- Búsqueda de aplicaciones similares.....	46
2.6.1.- Análisis comparativo de las aplicaciones encontradas	50
2.6.2.- Matriz comparativa	51
2.6.2.1.- Criterios de comparación.....	52
2.6.2.2.- Matriz.....	53
2.6.3.- Método de publicación de tarifas eléctricas.....	54
2.6.4.- Alternativa: Obtención de tarifas utilizando PHP	56
3.- Especificación de Requisitos	57
3.1.- Descripción del formato.....	57
3.2.- Requisitos Funcionales.....	57
3.3.-Requisitos No Funcionales	66
4.- Diseño de la arquitectura del sistema	69
4.1.- Evaluación de alternativas de diseño.....	69
4.1.1.- Alternativa 1:.....	69
4.1.2.- Alternativa 2:.....	70
4.1.3.- Solución propuesta	71
4.2.- Diseño interfaz de la aplicación	71
4.3.- Arquitectura general. MVC	74

4.3.1.- Modelo	75
4.3.2.- Vista	77
4.3.3.- Controlador	78
4.4.- Modelo de Casos de Uso	79
4.5.- Diagramas de secuencia	80
4.5.1.- Inicio Sesión y Registro	81
4.5.2.- Consultar, crear y eliminar Tareas del Día	82
4.5.3.- Consultar, crear y eliminar electrodomésticos	83
4.5.4.- Consultar, crear y eliminar Tareas	84
4.5.5.- Consultar Gasto Semanal	85
4.6.- Diagrama de Clases	86
4.7.- Diagrama de Componentes	89
5.- Descripción de las funcionalidades de la aplicación	91
5.1.- Introducción de las funcionalidades de la aplicación	91
5.2.- Planificación de las tareas del hogar	91
5.3.- Cálculo del gasto semanal	96
5.4.- Obtención de las tarifas eléctricas	97
5.5.- Inicio de Sesión y registro	98
5.6.- Opciones: Cambiar configuración de cuenta	100
5.7.- Navegación entre páginas de la aplicación	101
6.- Plan de Pruebas del sistema	102
6.1.- Entorno de pruebas	102
6.2.- Formato de los Casos de Prueba	105
6.3.- Casos de Prueba	106
6.4.- Resultados obtenidos y pruebas completadas con éxito	115
7.- Plan de trabajo y presupuesto	117
7.1.- Diagrama de Gantt	117
7.2.- Presupuesto	120
8.- Conclusiones generales y trabajo futuro	125
9.- Bibliografía	129
10.- Acrónimos y definiciones.	131

Índice de ilustraciones

Ilustración 1. MVC diagram	15
Ilustración 2. Tabla de especificaciones para Android Studio.....	32
Ilustración 3. Consumo de Electricidad por Servicio (Fuente: Eurostat).....	37
Ilustración 4. Consumo de electrodomésticos total y medio anual por hogar (Eurostat)	38
Ilustración 5. Consumo medio anual por electrodoméstico (Eurostat).....	39
Ilustración 6. Gráficas de análisis de emisiones relacionadas con producción electricidad	40
Ilustración 7. Evolución precios tarifas últimos años (Eurostat)	42
Ilustración 8. Aplicación Boltio.....	47
Ilustración 9. Aplicación Ahorra en luz. Precio luz hora.....	48
Ilustración 10. Aplicación My Power Consumption	49
Ilustración 11. Matriz comparativa de aplicaciones encontradas	53
Ilustración 12. Captura obtención tarifas eléctricas desde página web	55
Ilustración 13. Diagrama de navegación entre páginas de la aplicación	72
Ilustración 14. Ejemplo de interfaz de aplicación	73
Ilustración 15. Diagrama MVC de aplicación.....	74
Ilustración 16. Esquema relacional de la Base de Datos de la aplicación	76
Ilustración 17. Diagrama de multiplicidades de tablas de la BD	77
Ilustración 18. Diagrama de Casos de Uso	80
Ilustración 19. Diagrama de secuencia de Inicio de sesión y registro.....	81
Ilustración 20. Diagrama de secuencia de gestión tareas del día	82
Ilustración 21. Diagrama de secuencia de gestión de electrodomésticos	83
Ilustración 22. Diagrama de secuencia de gestión de tareas del hogar.....	84
Ilustración 23. Diagrama de secuencia de gestión del gasto semanal.....	85
Ilustración 24. Diagrama de Clases	88
Ilustración 25. Ejemplo interfaz diagrama de componentes	89
Ilustración 26. Diagrama de componentes	90
Ilustración 27. Página de Agenda	92
Ilustración 28. Página de tareas del día	93
Ilustración 29. Página de Electrodomésticos	94
Ilustración 30. Página de Tareas del Hogar	95
Ilustración 31. Página de Gestión Gasto Semanal.....	96
Ilustración 32. Implementación de la solicitud cliente a la API E.sios.....	97
Ilustración 33. Página de Inicio	98
Ilustración 34. Página de Registro.....	99
Ilustración 35. Página de Opciones	100
Ilustración 36. Captura primera de emulador Android Studio.....	103

Ilustración 37. Captura segunda emulador Android Studio.....	104
Ilustración 38. Captura tercera emulador Android Studio.....	105
Ilustración 39. Diagrama de Gantt completo.....	118
Ilustración 40. Diagrama de gantt parte 1.....	119
Ilustración 41. Diagrama de Gantt parte 2.....	120
Ilustración 42. Tabla recursos utilizados por tareas y roles de empleado.....	121
Ilustración 43. Coste de personal.....	122
Ilustración 44. Coste de recursos físicos.....	122
Ilustración 45. Coste recursos software.....	123
Ilustración 46. Suma de costes.....	123
Ilustración 47. Cálculo final del presupuesto total del proyecto.....	123

Resumen

El siguiente documento representa el Trabajo de Fin de Grado correspondiente a la titulación de Grado en Ingeniería Informática, cursado en la Universidad Carlos III de Madrid.

En este documento, se expone y detalla el proceso de desarrollo de una aplicación web móvil para Android dentro del ámbito del consumo energético. Esta aplicación está orientada al consumo eléctrico en el hogar y a la planificación de tareas con el fin de ahorrar energía y dinero en los hogares llevando un seguimiento del consumo eléctrico.

Para que fuera posible este proyecto, se realizó una planificación del mismo, determinando las fases del proyecto y los objetivos que cumplir en cada una, determinando las fechas límite y disponiendo del material necesario para su desarrollo previamente.

Primeramente, se llevó a cabo la fase de planificación del proyecto, donde se determinaron las fechas límite de finalización de las distintas fases del proyecto y los hitos a alcanzar.

Posteriormente se realizó una investigación exhaustiva del mercado de las aplicaciones web móviles similares a la que atañe este proyecto con el fin de determinar, haciendo uso de un análisis comparativo, las funcionalidades que debe tener esta aplicación. Se investigó también qué herramientas y recursos se deberían utilizar para el desarrollo del proyecto, para utilizarlos en posteriores fases del proyecto.

Se determinó por tanto que la aplicación debía ser implementada en la plataforma Android haciendo uso de la herramienta Android Studio y solicitando la información sobre las tarifas eléctricas a la web de la Red Eléctrica de España (ESIOS) mediante peticiones a su API.

Después de un estudio y comprensión de los conocimientos requeridos, se comenzó el desarrollo de la aplicación, comenzando por realizar pruebas de solicitud de datos a la API de ESIOS, para lo que se tuvo que solicitar previamente un token de autorización que te permite enviar dicha solicitud y obtener respuesta.

Después se realizó la implementación de la interfaz y de la lógica de la aplicación en Android Studio, dando paso a la realización de pruebas del sistema y comprobando así qué casos de prueba eran satisfactorios.

Por último, se comenzó la fase de documentación, donde se realizó todo el documento de memoria de este proyecto con la ayuda de los conocimientos adquiridos durante el resto de fases del proyecto.

Abstract

This document explains and details the process of developing a mobile web application for Android related with electricity consumption. This application is oriented to electrical consumption at home and task planning in order to save energy and money by monitoring electricity consumption.

For this project to be possible, it was planned, determining the phases of the project and the objectives to be fulfilled in each one of the phases, determining the deadlines and previously having the material necessary for its development.

Firstly, the project planning phase was accomplished, where the deadlines for complete the different phases of the project and the milestones to be achieved were determined. Subsequently, a thorough investigation of the market for similar web applications to the one related to this project was realized in order to determine, using a comparative analysis, the functionalities that should have this application. It was also investigated which tools and resources should be used for project development, for use in later phases of the project.

It was determined that the application had to be implemented in the Android platform making use of the Android Studio tool and requesting the information on the electrical tariffs to the website of Red Eléctrica de España (E.SIOS) through requests to their API.

After a study and understanding of the required knowledge, the development of the application began, starting by realizing data request tests to the E.SIOS API, for which you had to previously request a token that allows you to send such request and get response.

Then the implementation of the interface and the logic of the application was implemented in Android Studio, giving way to the testing of the system and verifying which test cases were satisfactory.

Finally, the documentation phase was started, where the entire document of this project was made with the help of the knowledge acquired during the rest of the project phases.

Introduction

Motivation

In the last years, electricity consumption has increased considerably, and electricity market prices have maintained constant growth, almost doubling, in 2017, the price of a decade ago.

The possibility of having an application that would allow us to reduce a large part of our monthly expenditure on household bills, such as our electricity consumption, would save enough money in every home.

In addition, today there is a lot of demand in mobile applications, in fact, about 90% of people with mobile phones have smart mobile phones. In the case of Android phones, they represent 87% of smartphones in Spain.

Another factor is the environmental one. In recent years, measures to prevent pollution in capital cities have increased considerably, and it is a fact that much of the environmental pollution is related to electricity consumption in the homes.

If an application that can help to save on the electricity bill was distributed nationally, pollution would decrease, and the environment would not be so affected by it, as well as increasing the quality of air in cities.

Everyone likes to receive solutions to their daily problems and earn money. The possibility of planning the home tasks is interesting, and even more so if you save money with it, that is why I think it is a mobile application with a lot of potential to be well received in the mobile applications market.

Problem statement

In this section, we will describe the steps that must be followed to complete this project, developing the different stages of which it is composed, in a general way.

First, in order to determine the functionalities that an application should provide that helps to save on the light bill by planning the home tasks such as ours, a market study should be realized, analyzing the similar applications that exist and detect the main functionalities, or even the functionalities that would allow us to differentiate our application from the others.

Then, to develop the application, we must know the main mobile application platforms that exist, and decide on one where our application is sold, as well as find out the technologies, tools and programming languages that are used to develop applications on the platform chosen.

After this, the study of obtaining electricity tariffs is necessary. To do so, obtain the information about which websites provide this information in Spain is essential, as well as the method of obtaining such information through the web services they provide, and be able to use this information in our application.

By designing the structure, we must begin with the development of the application. For this, the previous designs are used and the application is obtained with its interface, logic and persistence, as well as the communication with the API necessary to obtain the information of the electricity tariffs.

Finally, the System Test Plan is realized, verifying that all the functionalities implemented are done correctly.

Objectives

The main purpose of this end of degree project is to develop a personal assistant for the management of domestic electricity consumption, able to plan the home tasks to save money in the light bill.

To achieve this, some goals must be complete:

1. Develop a method to obtain electricity tariffs, in order to calculate the expenditure involved in the planning of home tasks.
2. Implement an information manager that can obtain users data, such as tasks that usually carry out electricity consumption, consumption of appliances that users have at home or the agenda of daily tasks of the week and persist them.
3. With the persistent information mentioned above, be able to calculate the electrical expenditure of the week depending on the hourly variability of the price of electricity tariffs.
4. Show the user detailed information about the weekly expenditure of his home, so that he can identify in which aspects of his planning he carry out a major electrical expense, so he can correct it.
5. Development of an intuitive user interface and easy to handle, which allows the user to use the application without any difficulty or discomfort.

Work method

The methodology of work realized during the months that has lasted the development of the project has been based on the phased division of the project, each phase having a series of objectives. The phases overlap one another, that is, although there were start and end dates, these phases coincided in time and it was not necessary to finish a previous phase to start the next. In the planning phase, which is the first one of the project, the dates, with a margin of error of approximately two weeks, were established from the completion of the phases accomplish the delivery of the final project on time.

The phases of the projects are:

- **Planning phase.**

The scope of the system was established and the deadlines for completion of the remaining phases of the project were organized.

- **Investigation phase.**

This phase consisted in finding similar applications, to see alternatives in the market and to perform comparative analysis. Was also analyzed possible tools to develop the application and do the documentation.

- **Development phase.**

Phase of implementation of the application based on the design established in the previous phases of the project, both the interface and the logic of the application.

- **Test phase.**

Was realized at the end of the development phase. This is because it is necessary to have the system that you are going to test implemented in order to test it. All functionalities were tested using the System Test Plan.

- **Documentation phase.**

The final version of the memory of the end of degree work was elaborated. The decision to do the documentation after the application development was based mainly

on that it would allow more knowledge about the application, since it would be finished, and the documentation would be more complete.

Resources used

This section describes the resources that have been used for the development of the project, such as programming languages, development environments, modeling tools, operating systems and documentation tools.

- **The programming languages** in which the project has been developed have been:
 - a. Java: Used to implement application logic.
 - b. XML: For development of the user interface.
 - c. SQL: For the implementation of the Database.

- **Software Tools:**
 - a. Windows 10: Operative system on which I have worked, on which other tools have been used.

 - b. Microsoft Office Word 2016: Tool used to prepare the memory and documentation, as well as to take necessary notes to acquire the knowledge to make the project.

 - c. Mockflow: Free interface prototyping tool. It has been used to set the design of the user interface.

 - d. Android Studio: Application development environment, both interface level and application logic.

 - e. Ganttproject: Tool used to elaborate the Gantt diagram product of the project planning phase.

 - f. Eclipse JEE Neon: During the investigation phase this tool was used in order to test the Esios API call to obtain its information.

- **Hardware:**
 - Basic Enterprise Pro i5-4460

Description of the application and results

Architecture design

The general architecture of the system is organized based on the Model View Controller (MVC), which consists of dividing the functions of the system into three: the data model is responsible for persistence and how information is organized (DB), the view it is the interface of the application, and the controller, which organizes and manages the information obtained from the model and the internal business logic of the application and displays it in the interface, that is, the view.

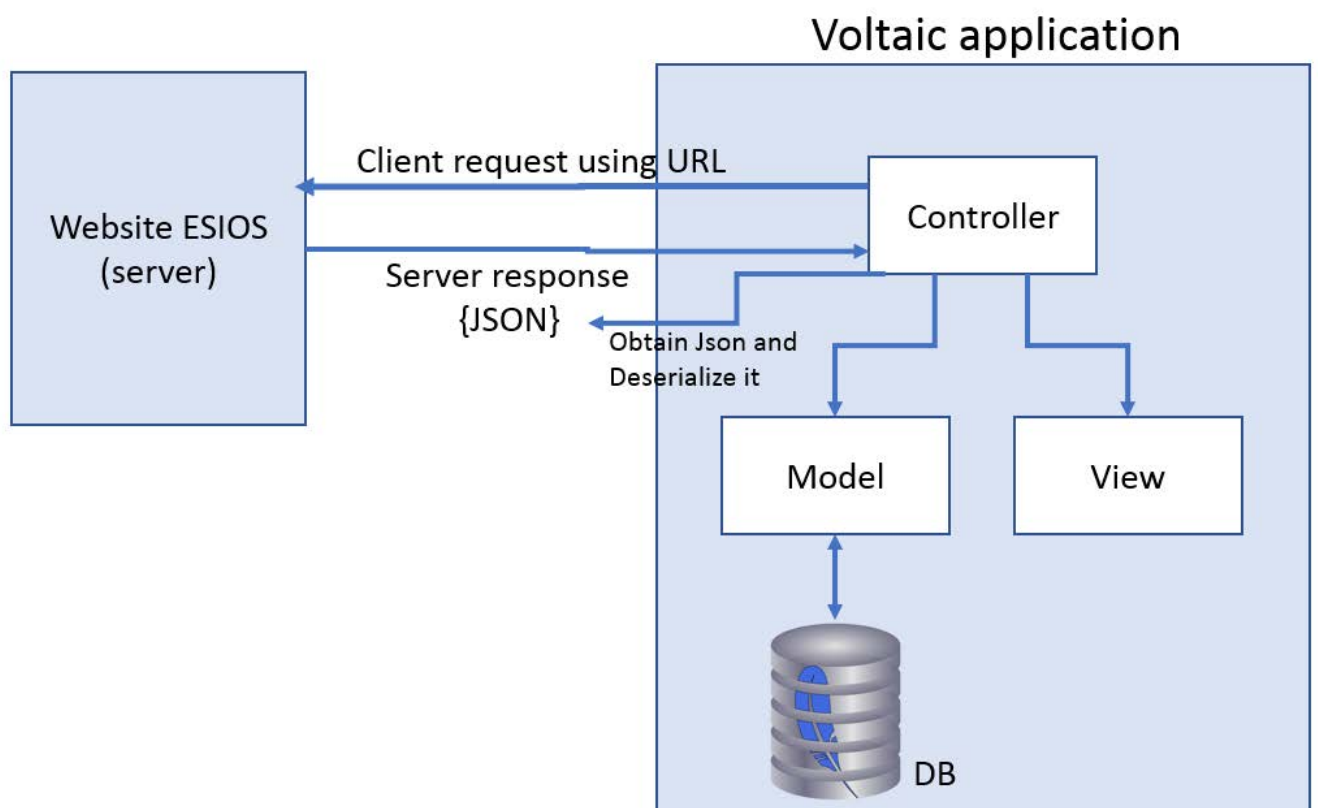


ILUSTRACIÓN 1. MVC DIAGRAM

Description of the functionalities

The application, called Voltaic, is an Android application that contains a number of features. Together, they form a personal assistant to plan the daily home tasks with the aim of saving on the light bill.

The features it contains are:

- Homework Planning: Contains a calendar that is used as an agenda to organize the tasks and schedule them the desired day and time.
- Calculation of the weekly expenditure: It is responsible for calculating the expenditure in euros of all tasks scheduled for the current week.
- Get electricity tariffs: Functionality in charge of requesting prices of electricity rates, in order to calculate the weekly expenditure in euros.
- Login and Registration: To access, first, you need to be registered, and if you are, you must log in by entering your data to access your account.
- Options. Change account settings: In order to modify your account information, such as your password or email.

Test results. System Test Plan

The test plan of the system consists of verifying that the developed application surpasses the tests on all the functionalities, in a correct, optimum, efficient way and according to the User Cases and the system requirements.

All test cases of the Test Plan are successfully completed, thus performing the required functions, as stipulated in said test cases, the inputs and outputs of them and also complying with the functional requirements related to them.

The results of the Test Plan demonstrate that the objectives initially proposed in the Project Planning Phase have been successfully completed.

During the tests, errors were detected, and from them lessons were learned that helped to achieve the optimum result.

Conclusions and future work

Finally, having done a study of the scope of the system and the problem that was raised, having analyzed the market of mobile electrical power management applications in Spain and its environment, in addition to designing the system architecture from different perspectives, as well as analyzing and describing the functionalities that have been implemented and realizing the study of the socio-economic project framework, on the one hand, to understand the repercussions of the distribution of this application in the current society and the market, as legislative, and on the other, to describe the legal perspective of the project, it can be confirmed that all the objectives proposed for the final elaboration of this project have been accomplished.

In order to achieve all these objectives, a comprehensive study of the electricity market has been realized and which websites provide the electricity tariffs, as well as the most optimal method for the project to obtain this information.

We have done a comparative analysis of the market for similar mobile applications, with the objective of studying the market and detecting the necessary functionalities for all applications of this class, as well as analyzing the other tools and resources of the project, in order to achieve the same looking for the lowest expense and the maximum effectiveness and efficiency in the work.

Much work it's been done on the implementation of a usable interface and on functionalities that meet the proposed objectives, studying the best options to achieve it with the least resources possible.

But in order to finish successfully this project, I had to go through many problems and difficulties.

In the early stages of the project, I found it difficult to understand what the development environment should be, and so I made many mistakes: libraries, project configuration, etc.

It was also difficult for me to learn how to use Android Studio, since I had never used it, but over time I learnt about how to use it.

It should be noted that what has cost me the most in general terms is the implementation of the application. It is certainly the part of the project that required more hours of work.

Regarding the documentation, it was complicated to organize the structure of memory and decide which models of the system wanted to display to describe the system, in addition to make them.

The result has been this application, called Voltaic, a mobile personal assistant for the management of the home electricity consumption on Android platform, able to plan the electrical expenditure and with the tools needed to save on the electricity bill.

I've learned a lot about programming Android applications, about the electricity market and the savings that can mean optimizing the expenditure that is produced in a home, and above all, I have learned to make a project of this magnitude and with so many hours of work that it requires.

Regarding the future work that could be done with this project, there are many options:

- We could expand the application market by implementing versions for IOS and Windows Phone, allowing the application to reach more sectors of the market and increase profits.

- Implement the Domicile functionality. It consists of that each user can have one or more addresses that he manages in his account, which each have their appliances, home tasks, and so on. With this functionality, you could share the domiciles with other people, for example, with other people you live with but do not want to have to enter all the data again.
- We could expand the obtaining of electric tariffs to other countries, not only to Spain. ESIOS is a web site of the Red Eléctrica de España and therefore only provides data of the tariffs of this country. In order to introduce our application to other countries, it would be possible to increase the functionality of requesting electricity tariffs to the electricity networks of other countries and increase the market of the application.
- Implement, through the compatibility with electric current sensors, the reception of data consumption of home appliances without the need to enter them manually by the user, which could be more convenient for a sector of users.
- Develop a sound system to facilitate the use for people with disabilities.
- In the future, and if necessary, more servers could be added. This would be necessary in case the application had a high market distribution and demand would increase fast, so we would have to be prepared to increase the capacity of the application servers with the same speed.
- Add more security to the application. Develop an RSA encryption for external operations with server and database. Taking more security measures would ensure user data and the application of malicious attacks.

1.-Introducción

1.1.- Motivación

En los últimos años, el consumo eléctrico ha aumentado considerablemente, y los precios del mercado eléctrico han mantenido un crecimiento constante, llegando casi a duplicar, en este año 2017, el precio de hace una década.

La posibilidad de tener una aplicación que nos permitiera reducir una gran parte de nuestro gasto mensual en las facturas del hogar, como es nuestro consumo eléctrico, permitiría ahorrar bastante dinero a todos los hogares.

Además, hoy en día existe mucha demanda en las aplicaciones móviles. De hecho, ya alrededor del 90% de las personas con móvil tienen móviles inteligentes. En el caso de los móviles Android, representan el 87% de los móviles inteligentes en España.

Otro factor es el medioambiental. En los últimos años, las medidas para prevenir la contaminación en las grandes ciudades han aumentado considerablemente, y es un hecho que gran parte de la contaminación medioambiental está relacionada con el consumo eléctrico en los hogares.

Si una aplicación que permitiera ahorrar en la factura de la luz se distribuyera a nivel nacional, la contaminación disminuiría, y el medioambiente no se vería tan afectado por ella, además de que aumentaría la calidad del aire en las ciudades.

A todo el mundo le gusta que le den soluciones sobre sus problemas diarios y ganar dinero. La posibilidad de planificar las tareas del hogar resulta interesante, y más aún más si ahorras dinero con ello, es por ello que pienso que es una aplicación móvil con mucho potencial para tener una buena acogida en el mercado de las aplicaciones móviles.

1.2.- Planteamiento del problema

En este apartado, se van a describir los pasos que se deberán seguir para completar este proyecto, desarrollando las distintas etapas de la que se compone, de una forma general. En el apartado 1.4- Método de trabajo y Recursos utilizados, se detallan las Fases de trabajo del proyecto y las tareas que las componen de una forma más detallada.

En primer lugar, para determinar las funcionalidades que deberá proporcionar una aplicación que ayude a ahorrar en la factura de la luz mediante la planificación de las tareas del hogar como la nuestra, se debe realizar un estudio de mercado, analizando las aplicaciones similares que existen y detectar las funcionalidades principales, o incluso, las funcionalidades que permitirían diferenciar nuestra aplicación de las demás, destacando así en el mercado.

Después, para desarrollar la aplicación, debemos saber las principales plataformas de aplicaciones móviles que existen, y decidir una en la que se venda nuestra aplicación, así como averiguar las tecnologías, herramientas y lenguajes de programación que se utilizan para desarrollar las aplicaciones en la plataforma elegida.

Tras esto, es necesario el estudio de obtención de tarifas eléctricas. Para ello, recabar información sobre que Sitios Web proporcionan esta información en España es imprescindible, así como el método de obtención de dicha información a través de los servicios web que proporcionan, y así poder utilizar esta información en nuestra aplicación.

Realizando el diseño de la interfaz, de la base de datos para persistir la información, y de la lógica de negocio de la aplicación, hay que comenzar con el desarrollo de la aplicación. Para ello, se utilizan los diseños anteriores y se obtiene la aplicación con su interfaz, lógica y persistencia, así como la comunicación con la API necesaria para obtener la información de las tarifas eléctricas.

Por último, se realiza el Plan de Pruebas del Sistema, comprobando que todas las funcionalidades implementadas se llevan a cabo correctamente.

1.3.- Objetivos

El propósito principal de este trabajo de fin de grado es el de desarrollar un asistente personal para la gestión del consumo eléctrico doméstico, capaz de planificar las tareas del hogar para ahorrar así en la factura de la luz.

Para ello, se deberán cumplir una serie de metas:

1. Desarrollar un método de obtención de las tarifas eléctricas, para poder calcular el gasto que conlleva la planificación de las tareas del hogar.
2. Implementar un gestor de información que permita obtener los datos de los usuarios, tales como tareas que suelen realizar que conllevan un consumo eléctrico, consumo de los electrodomésticos que tienen en casa o la agenda de tareas cotidianas de la semana y persistirlos.
3. Con la información persistida mencionada anteriormente, ser capaces de calcular el gasto eléctrico de la semana en función de la variabilidad por horas del precio de las tarifas eléctricas.
4. Mostrar al usuario información detallada del gasto semanal de su hogar, para que sea capaz de identificar en qué aspectos de su planificación realiza un gasto eléctrico mayor, y poder así corregirlo.
5. Desarrollo de una interfaz de usuario intuitiva y fácil de manejar, que permita al usuario utilizar la aplicación sin ningún tipo de dificultad o incomodidad.

1.4.- Método de trabajo y recursos utilizados

La metodología de trabajo llevada a cabo durante los meses que ha durado el desarrollo del proyecto se ha basado en la división por fases del proyecto, teniendo cada fase una serie de objetivos. Las fases se superponen unas a otras, es decir, aunque había fechas de inicio y fin de fase, dichas fases coincidían en el tiempo y no era necesario acabar una fase previa para empezar a siguiente. En la fase de planificación, que es la primera a del proyecto, se establecieron las fechas, con margen de error de aproximadamente dos semanas, de la finalización de las fases para llevar a cabo la entrega del proyecto final a tiempo.

Se han llevado a cabo reuniones cada tres semanas aproximadamente con el tutor del proyecto para llevar a cabo un seguimiento del TFG.

Ahora se procederá a describir las distintas fases del proyecto.

1.4.1- Fase de Planificación

Esta fase abarca a partir de la primera reunión establecida a mediados de septiembre de 2016 con el tutor de este proyecto. Durante esta etapa del proyecto, se plantearon los siguientes objetivos:

- Primero, se estableció el alcance del sistema, para determinar las funcionalidades que debía tener la aplicación y sus principales características. Esto ayudó para hacerse una idea del trabajo que iba a requerir el proyecto, aunque fuera una idea aproximada.
- Se determinaron las fechas aproximadas de finalización de cada fase, ya que en una fase tan temprana del proyecto era difícil calcular con exactitud el tiempo y trabajo que requeriría cada fase.
- Establecimiento de las herramientas y el material de trabajo de una manera aproximada y susceptible de cambios, ya que en fases posteriores se determinó en algunos casos que era más óptimo cambiar de herramientas, como fue el caso de utilizar en un principio TeamGantt para la elaboración del diagrama de Gantt para después cambiar a la herramienta GanttProject, o como en el caso de la realización del

mockup de la aplicación, que se comenzó utilizando LucidChart y después se utilizó MockFlow.

- Realización del diseño de la interfaz de usuario en un mockup, lo cual fue teniendo modificaciones a lo largo del proyecto.
- Elaboración de una primera versión de la arquitectura del sistema.

1.4.2.- Fase de Investigación

Después se llevó a cabo la fase de investigación, que consistió, a grandes rasgos, en buscar aplicaciones similares, ver alternativas en el mercado y realizar análisis comparativo. También se analizaron las posibles herramientas para desarrollar la aplicación y hacer la documentación.

- Búsqueda de aplicaciones similares en Play Store y App Store.
- Análisis comparativo de las aplicaciones encontradas.
- Estudio sobre cómo hacer posible que la aplicación obtuviera los datos de las tarifas eléctricas actualizadas diariamente.
- Análisis de las posibles herramientas de desarrollo, realizando un estudio comparativo.

1.4.3.- Fase de Desarrollo

Referente a la fase de implementación de la aplicación basándose en el diseño establecido en las fases previas del proyecto, tanto de la interfaz como de la lógica de la aplicación.

- Desarrollo de la interfaz de la aplicación basándose en el diseño de fases previas, en Android Studio, en lenguaje XML.
- Implementación de la base de datos SQLite.
- Implementación de la lógica de negocio en Android Studio, utilizando lenguaje Java.
- Implementación de la solicitud cliente a la API de Esios para obtener los datos actualizados de las tarifas eléctricas, en lenguaje Java.

1.4.4.- Fase de pruebas

La fase de pruebas del sistema se llevó a cabo nada más terminar la fase de desarrollo. Esto es así porque es necesario tener el sistema que vas a probar implementado para poder realizarle pruebas. Las tareas que se llevaron a cabo durante esta fase fueron las siguientes:

- Elaboración del plan de pruebas del sistema.
- Comprobación de los casos de prueba.

1.4.5.- Fase de documentación

Finalmente, se llevó a cabo la fase de documentación, donde se elaboró la versión final de la memoria del Trabajo de Fin de Grado. La decisión de hacer la documentación después del desarrollo de la aplicación se basó fundamentalmente en que permitiría tener más conocimiento acerca de la aplicación, ya que estaría terminada, y la documentación sería más completa.

- Elaboración de la estructura de la memoria. Esto fue importante para comenzar a organizar la memoria y determinar el rumbo de la misma.
- Elicitación versión final de requisitos. Esta versión se realizaba en base a la hecha en la fase de planificación, y era la versión definitiva.
- Estudio del marco socioeconómico del proyecto. Este apartado se llevó a cabo realizando un estudio sobre la influencia que tendría en la sociedad y en la economía del país.
- Elaboración del presupuesto. Tras saber de primera mano las herramientas que iban a ser necesarias para la elaboración del proyecto, además de las horas necesarias para llevarlo a cabo, se realizó el presupuesto total del proyecto.
- Redacción de versión final del resto de la memoria. Esta tarea consistió en realizar las versiones finales de todos los apartados y aspectos de la memoria.

1.4.6.- Recursos utilizados

En este apartado se describen los recursos que han sido utilizados para el desarrollo del proyecto, tales como lenguajes de programación, entornos de desarrollo, herramientas de modelado, sistemas operativos y herramientas de documentación.

- Los lenguajes de programación en los que se ha desarrollado el proyecto han sido:
 - **Java**: Utilizado para implementar la lógica de la aplicación.
 - **XML**: Para desarrollo de la interfaz de usuario.
 - **SQL** : Para la implementación de la Base de Datos.
- Las herramientas utilizadas han sido:
 - **Windows 10**: Sistema operativo en el que he trabajado, sobre los que se han utilizado el resto de herramientas.
 - **Microsoft Office Word 2016**: Herramienta utilizada para elaboración de la memoria y documentación, así como para tomar apuntes necesarios para adquirir los conocimientos para llevar a cabo el proyecto.
 - **Mockflow**: Herramienta gratuita de prototipado de interfaces. Se ha utilizado para establecer el diseño de la interfaz de usuario.
 - **Notepad++**: Herramienta utilizada para visualizar y consultar código.
 - **Android Studio**: Entorno de desarrollo de la aplicación, tanto a nivel de interfaz como de lógica de la aplicación.
 - **Ganttproject**: Herramienta utilizada para elaborar el diagrama de Gantt producto de la fase de planificación del proyecto.
 - **Eclipse JEE Neon**: Durante la fase de investigación se utilizó esta herramienta con el fin de realizar pruebas de llamada a la API de Esios para solicitar su información.
- El hardware usado:
 - Portátil: Basic Enterprise Pro i5-4460

1.5.- Introducción al resto de contenidos del documento

La estructura del documento está compuesta por los siguientes capítulos:

1. **Introducción:** Como hemos visto, es el apartado que se encarga de presentar el proyecto. Se exponen la motivación que lleva a embarcarse en este proyecto, explicando las razones por las que este proyecto merece la pena llevarlo a cabo. También se han expuesto el planteamiento de la problemática, para detallar los pasos que se deberán seguir para completar el proyecto, y los objetivos del mismo, explicando de forma más precisa las metas que se deberán completar. Por último, se redacta la metodología que se ha llevado a cabo a lo largo del proyecto para realizar el trabajo de todas las tareas, así como los recursos hardware y software utilizados a lo largo del proyecto para completarlo.
2. **Estado del arte.** En este apartado se describe el entorno de la aplicación (mercado, competidores, factores limitantes, etcétera). Se comienza explicando cómo funciona, de una forma resumida, el mercado eléctrico en España y las tarifas eléctricas. Después, se describe la plataforma en la que está implementada la aplicación y el entorno de desarrollo, así como el análisis y la decisión de escoger un Sistema Gestor de Base de Datos. Posteriormente, se describen, de forma detallada y desde distintas perspectivas, el entorno socioeconómico y legal del proyecto. Por último, se redacta el proceso seguido en el proceso de búsqueda de aplicaciones similares encontradas y el análisis comparativo entre las mismas, así como la descripción precisa del método habitual de obtención de tarifas eléctricas desde un Sitio Web.
3. **Especificación de requisitos:** Consiste en la redacción, mediante el uso de un formato definido previamente, de los requisitos del sistema por medio de tablas de requisitos, para describir el sistema de forma exacta y libre de ambigüedades, para facilitar la comprensión del sistema. Se redactan tanto los requisitos Funcionales como los No Funcionales.
4. **Diseño de la arquitectura del sistema:** Primero, se exponen varias alternativas de diseño de la aplicación y se comparan con el diseño implementado, para corroborar

que la decisión de diseño tomada en este proyecto es la más óptima. Después, se describe, haciendo uso de diferentes perspectivas de diseño, y con ello, de modelos y diagramas, la estructura del sistema, tanto de la parte front-end como back-end del mismo. También se describen los Casos de Uso del sistema y las funcionalidades que lo componen de forma precisa gracias a los diagramas de secuencia.

5. **Descripción de las funcionalidades de la aplicación:** En este apartado, se exponen las distintas funcionalidades del sistema desde el punto de vista del usuario y haciendo uso de la aplicación, con explicaciones breves de cómo se ha realizado a nivel de implementación.
6. **Plan de pruebas del sistema:** Se expone el Plan de Pruebas del Sistema, encargado de comprobar que se cumplen todos los objetivos y que las funcionalidades de la aplicación se ejecutan obteniendo el resultado esperado. Descrito el entorno de pruebas y los casos de prueba con el formato indicado previamente, se describen los resultados obtenidos.
7. **Plan de trabajo y presupuesto:** Apartado en el que se detallan las tareas y las fases del proyecto a lo largo de la consecución del mismo, así como el cálculo, basándose en los gastos de recursos en dichas tareas, así como los gastos de hardware y software llevados a cabo, del presupuesto total del proyecto.
8. **Conclusiones generales y trabajo futuro:** Se extraen las enseñanzas y conclusiones finales tras la terminación del proyecto. Se exponen los pasos seguidos para conseguirlo y el resultado obtenido, además de exponer las posibles líneas de trabajo futuro en este proyecto.

2.-Estado del arte

El objetivo de este apartado es el de explicar y analizar el entorno que rodea al proyecto desde distintas perspectivas, tanto legal como social, ética, económica o tecnológica.

Primero, se explica el funcionamiento del mercado eléctrico en España, se describe la plataforma sobre la que se desarrollará esta aplicación y su IDE oficial, y se continúa realizando un estudio comparativo de aplicaciones móviles similares a la nuestra, que proporcionen información acerca de las tarifas eléctricas y puedan planificar actividades del hogar, así como recabar información sobre éstas y compararlas.

Por último, se explica el método de publicación de tarifas eléctricas en España, se describe el método utilizado para obtenerlas en este proyecto y una solución alternativa.

2.1.-Tarifas eléctricas en España

En España existen diversas centrales eléctricas y compañías que se encargan de vender la electricidad a los consumidores finales, y con ello enviarles las facturas.

El mercado eléctrico está compuesto por **dos tipos de mercado: el libre y el regulado**.

Antes, el mercado estaba regulado por los precios que establecía el Gobierno, pero a partir de 1977 se comenzó un largo proceso de liberación de dicho mercado. A partir de 2009 los consumidores son libres de elegir a quién le van a pagar su factura, pudiendo escoger actualmente entre casi 100 comercializadoras.

La liberación, sin embargo, no es completa, y por eso todavía existe el mercado regulado. En ambos mercados, los peajes de acceso y los impuestos los establece el Gobierno.

Sin embargo, lo que diferencia a ambos mercados es el precio que se cobra por producir electricidad.

En el mercado regulado se aplica la tarifa PVPC (Precio Voluntario al Pequeño Consumidor), que es un precio que cambia cada hora diariamente, dependiendo de la oferta y la demanda. Por ello, a las horas donde más se suele consumir el precio es más alto, al contrario que en las horas que menos se consume.

En el mercado libre, sin embargo, el precio se establece por contrato y es fijo.

La posibilidad de conocer las horas del día en las que cuesta menos la electricidad y aprovecharlas para consumir, aumenta el ahorro en la factura de la luz.

Este proyecto se ha implementado para aprovechar esas franjas horarias y así ahorrar, sin embargo, es posible adaptar esta funcionalidad al mercado libre. Para ello, simplemente se establecería otra consulta a la Api de Esios, que en vez de consultar el valor PVPC, consultara el precio de la tarifa del mercado libre de la compañía en cuestión.

2.2.- Android

Android es un sistema operativo, basado en Linux, diseñado para dispositivos móviles con pantalla táctil tales como móviles inteligentes, tabletas o relojes inteligentes.

Fue desarrollado por una startup llamada Android Inc en el año 2003. Esta startup era respaldada económicamente por Google, hasta que en 2005 la compró. Android se presentó en 2007 con el objetivo de dar un paso adelante en los estándares de los dispositivos móviles. El primer móvil que llevó integrado este sistema operativo fue el HTC Dream, que se vendió a partir de octubre del año 2008.

La consultora Gartner publicó en agosto de 2017 los datos de cuota de mercado referentes a smartphones. Android va aumentando la cuota y se consolida como líder de mercado, mientras que iOS de Apple se mantiene como principal alternativa.

Según este estudio, Android ha obtenido el 87,7% de cuota, siendo un total de 366,2 millones de unidades vendidas en todo el mundo durante el trimestre. Los datos también arrojan un crecimiento en ventas de un 6,7 %.

En España, la diferencia es más significativa: según los últimos datos de Kantar Worldpanel ComTech, para el trimestre que acaba en febrero de 2017, el sistema operativo de Google representa un 92,2 % de cuota de mercado, aumentando así 2,2 puntos con respecto a febrero de 2016.

2.2.1.-Características y especificaciones de Android

- **Diseño de dispositivo:** Es adaptable a tamaños de pantalla variables de alta resolución tales como VGA, bibliotecas de gráficos 2D y 3D.
- **Almacenamiento:** Hace uso de SQLite, una base de datos ligera para asegurar la persistencia de los datos.
- **Conectividad:** GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, WI-FI, LTE, HSDPA, HSPA+, NFC, WiMax, GPRS, UMTS, HSDPA+.

- Mensajería: Soporte de SMS, MMS, FCM, GCM.
- Navegador web: Basado en el motor WebKit, de código abierto.
- Soporte Java: En la plataforma no existe una máquina virtual de java, pese a que la gran mayoría de las aplicaciones estén implementadas en dicho lenguaje, sino que se utiliza (a partir de la versión 5.0) el entorno de ejecución Android Runtime (ART).
- Soporte multimedia: Soporte de WebM, H.263, H.264, MEG-4 SP, AMR, AMR-WB, AAC, HE-ACC, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, BMP.
- Soporte para Streaming: RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player.
- Soporte Hardware adicional: Cámaras de fotos y vídeos, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
- Entorno de desarrollo: Contiene un emulador de dispositivos Android, herramientas de depuración y de análisis de rendimiento. En un principio el Entorno de Desarrollo Integrado (IDE) era Eclipse con el plugin de Herramientas de desarrollo de Android, pero más tarde se estableció Android Studio como entorno oficial, del que hablaremos en el siguiente apartado y en el que se ha desarrollado este proyecto.
- Otras características: Posee también soporte multi-táctil, Bluetooth (para A2DP y AVRCP además de OPP), videollamada y multitarea (las tareas que estén en segundo plano reciben ciclos de reloj).

2.2.2.-Android Studio

Se trata del Entorno de Desarrollo Integrado (IDE) oficial para el desarrollo de aplicaciones para la plataforma Android. Está basado en IntelliJ IDEA, de JetBrains, y utiliza una licencia software libre Apache 2.0. Está programado en java y es multiplataforma.

Sus principales características son las siguientes:

- Un sistema de compilación basado en Gradle, encargada de gestionar y automatizar la construcción de proyectos, como pueden ser las tareas de testing, compilación o empaquetado.
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android.
- Instant Run para aplicar cambios mientras la aplicación que se está desarrollando se ejecuta sin la necesidad de compilar un nuevo APK.
- Integración de plantillas de código y GitHub que ayuda a compilar funciones comunes de las apps e importar ejemplos de código.
- Gran cantidad de herramientas y frameworks de prueba.

- Herramientas Lint (se encarga de detectar código no compatible entre arquitecturas diferentes o código confuso que no es capaz de controlar el compilador) para detectar problemas de rendimiento, usabilidad y compatibilidad de versiones.
- Compatibilidad con C++ y NDK.
- Soporte para programar aplicaciones para Android Wear.
- Permite la importación de proyectos realizados en el entorno Eclipse, que a diferencia de Android Studio (Gradle) utiliza ANT.
- Alertas en tiempo real de errores sintácticos, compatibilidad o rendimiento antes de compilar la aplicación.
- Editor de diseño que muestra una vista previa de los cambios realizados directamente en el archivo xml.

Especificaciones del sistema:

En la siguiente figura se detallan las especificaciones del sistema para el uso de Android Studio.

Windows	Mac OS	Linux
Microsoft Windows 8/7/Vista/2003 (32 o 64 bit)	Mac OS X 10.8.5 o superior, hasta la 10.9 (Mavericks)	GNOME o entorno de escritorio KDE
<ul style="list-style-type: none"> • Mínimo de 2 GB de RAM (recomendado: 4 GB de RAM). 		
<ul style="list-style-type: none"> • 400 MB de espacio en disco. 		
<ul style="list-style-type: none"> • Necesario 1 GB para Android SDK, emulador de imágenes del sistema, y cachés. 		
<ul style="list-style-type: none"> • Resolución mínima de pantalla de 1280 x 800 		
<ul style="list-style-type: none"> • Java Development Kit (JDK) 7 o superior. 		

ILUSTRACIÓN 2. TABLA DE ESPECIFICACIONES PARA ANDROID STUDIO

2.2.3.- Eclipse con plugin de Herramientas de Desarrollo Android

Hasta hace poco, existía una alternativa a Android Studio y era Eclipse con plugin de Herramientas de Desarrollo Android. De hecho, hace un tiempo era el entorno de desarrollo oficial para implementar aplicaciones Android, pero a finales de 2015 se anunció el fin de soporte para este plugin de entorno de desarrollo (incluso se retiró la versión Eclipse para desarrollo de Android).

Es por todo esto que no se realizó el proyecto en esta plataforma, optando por utilizar el IDE oficial para programación de aplicaciones Android, Android Studio.

2.3.- Breve Análisis de Sistemas Gestores de Bases de Datos

Para asegurar la persistencia de los datos e información que se manejan en la aplicación, y que permiten que sus funcionalidades se lleven a cabo correctamente, es necesario la utilización de un SGBD (Sistema Gestor de Bases de Datos).

Existen muchos SGBD, cada uno con sus ventajas e inconvenientes. En este apartado, se realiza un análisis para justificar la decisión de optar por SQLite para implementar nuestra Base de Datos.

Estas son las principales opciones a la hora de elegir un SGBD:

- **MySql**

Sistema gestor de bases de datos relacional, multihilo y multiusuario. Está considerada como el SGBD de código abierto más popular. Está desarrollado, en su mayor parte, en ANSI C.

Se caracteriza por la velocidad en las operaciones, su bajo coste en requerimientos para elaborar la base de datos y por su facilidad de instalación y configuración.

- **Oracle**

Es uno de los SGBD más completos y destaca por su soporte de transacciones, por su estabilidad, escalabilidad y porque es multiplataforma. Es el SGBD más utilizado a nivel mundial. Sin embargo, su precio es elevado.

- **SQLite**

Es de código abierto y compatible con ACID. Implementa la mayor parte de la librería SQL-92 y posee un alto rendimiento. Este SGBD no posee una estructura cliente-servidor, en su lugar, SQLite se enlaza con el programa para formar parte del mismo, y se comunica con el programa a través de llamadas a subrutinas y funciones, lo que reduce la latencia de acceso a la base de datos.

SQLite permite almacenar bases de datos de hasta 2 Terabytes.

2.3.1.- SGBD elegido para este proyecto

El SGBD elegido ha sido **SQLite**. Debido a su simplicidad a la hora de crear la base de datos, ya que forma parte del código de la aplicación, y con esto, debido a su rapidez a la hora de acceder a la base de datos, es la razón principal de haber escogido este SGBD para implementar la base de datos de nuestra aplicación. Resulta fácil trabajar con SQLite en Android Studio y las librerías están disponibles sin necesidad de importar librerías externas. Por todo ello, es el SGBD elegido.

2.4.- Entorno socioeconómico del proyecto

2.4.1.- Análisis del entorno socioeconómico

Ahora se procederá a describir el marco socioeconómico de este proyecto. Analizaremos la situación desde la perspectiva de los hogares y la influencia que tendría en ellos esta aplicación, así como el ahorro que podría suponer en la factura mensual para una familia media. También cabe destacar la influencia sobre el medioambiente que conllevaría el uso

masivo de esta aplicación, ya que supondría un ahorro de energía eléctrica y por tanto del uso de centrales energéticas.

Analizaremos cuánto gasto supone para una familia media la factura de la luz y cuál es el consumo medio por hogar, así como la tendencia futura de la diferencia entre ingresos en un hogar medio y el precio de las tarifas eléctricas.

En primer lugar, cabe analizar cuál es el consumo eléctrico medio por hogar en los hogares españoles. Según datos del **Eurostat** y del Ministerio de Industria, Energía y turismo, el consumo de los hogares españoles es:

1.- Consumo de los Hogares en España

- Consumo Total: 14.676 ktep
- Consumo Total Medio por hogar: 0,852 tep*
- **Consumo de Electricidad Medio por Hogar: 3.487 kWh**
- % Consumo Final: 17%
- **% Consumo de Electricidad: 25%**

(*Siendo tep Toneladas equivalentes de petróleo.)

Podemos observar que el consumo medio por hogar en España asciende a 3.487 kWh, constituyendo un 25% del consumo en el hogar.

En la tabla siguiente podemos comprobar el consumo eléctrico según los servicios:

2.- % Consumo Eléctrico según Servicios

- Calefacción: 7,4%
- ACS: 7,5%
- Cocina: 9,3%
- Refrigeración: 2,3%
- Iluminación: 11,7%
- **Electrodomésticos: 55,2%**
- Standby: 6,6%

Constituyendo más de la mitad del gasto eléctrico en electrodomésticos (55,2 %), demuestra de la importancia de saber gestionar los gastos de consumo eléctrico relacionados con las tareas del hogar que necesitan de esos mismos electrodomésticos para realizarse.

A continuación, podemos ver el porcentaje de consumo según qué electrodoméstico, siendo estos los más generales y de mayor uso, además del consumo medio de los principales electrodomésticos.

3.-% Consumo de Electrodomésticos según Tipo de Equipamiento

- Frigorífico: 30,6%
- Congeladores: 6,1%
- Lavadora: 11,8%
- Lavavajillas: 6,1%
- Secadoras: 3,3%
- Horno: 8,3%
- TV: 12,2%
- Ordenadores: 7,4%
- Standby: 10,7%
- Otros: 3,5%

4.-Consumo Medio de Iluminación y Electrodomésticos

- Iluminación: 410 kWh/hogar
- Frigorífico: 655 kWh/eq.
- Lavadora: 254 kWh/eq.
- Lavavajillas: 245 kWh/eq.
- TV: 119 kWh/eq.
- Ordenador: 145 kWh/eq.

Esto nos demuestra que las tareas cotidianas constituyen el mayor gasto eléctrico de la casa a diario, con lo que saber gestionar esos gastos supondría menos coste en la factura de la luz.

En la siguiente figura se muestra el consumo eléctrico por electrodomésticos:

6.- Consumo de Electricidad por Servicio

Unidad: GWh	España		Zona Atlántica		Zona Continental		Zona Mediterránea	
Calefacción	4.418	7,4%	691	8,8%	1.698	8,4%	2.029	6,4%
Agua caliente sanitaria	4.480	7,5%	622	8,0%	894	4,4%	2.964	9,3%
Cocina	5.572	9,3%	1.068	13,7%	2.341	11,6%	2.163	6,8%
Refrigeración	1.400	2,3%	14	0,2%	535	2,6%	851	2,7%
Iluminación	7.045	11,7%	796	10,2%	1.902	9,4%	4.346	13,6%
Electrodomésticos	33.099	55,2%	4.094	52,4%	11.521	57,0%	17.483	54,7%
Standby	3.969	6,6%	529	6,8%	1.337	6,6%	2.103	6,6%
TOTAL	59.983	100%	7.815	100%	20.229	100%	31.940	100%

Unidad: GWh	España		Pisos		Unifamiliares	
Calefacción	4.418	7,4%	3.180	7,8%	1.238	6,4%
Agua caliente sanitaria	4.480	7,5%	2.856	7,0%	1.624	8,4%
Cocina	5.572	9,3%	3.865	9,5%	1.707	8,8%
Refrigeración	1.400	2,3%	914	2,3%	486	2,5%
Iluminación	7.045	11,7%	4.805	11,8%	2.240	11,6%
Electrodomésticos	33.099	55,2%	22.122	54,5%	10.977	56,7%
Standby	3.969	6,6%	2.869	7,1%	1.101	5,7%
TOTAL	59.983	100%	40.610	100%	19.373	100%

ILUSTRACIÓN 3. CONSUMO DE ELECTRICIDAD POR SERVICIO (FUENTE: EUROSTAT)

Y a continuación el consumo de los electrodomésticos en TEP (Toneladas Equivalentes de Petróleo) así como el consumo medio anual de los electrodomésticos en el hogar, en KWh. Después, la media del consumo al año por electrodoméstico.

7.-Consumo de Electrodomésticos

Unidad: ktep	España		Zona Atlántica		Zona Continental		Zona Mediterránea	
Frigoríficos	975	30,6%	107	26,9%	341	30,9%	527	31,3%
Congeladores	193	6,1%	39	9,7%	48	4,3%	106	6,3%
Lavadoras	378	11,8%	59	14,9%	119	10,8%	199	11,8%
Lavavajillas	193	6,1%	24	6,1%	70	6,4%	98	5,8%
Secadoras	107	3,3%	10	2,5%	25	2,3%	71	4,2%
Horno	263	8,3%	42	10,6%	97	8,8%	124	7,4%
TV	388	12,2%	29	7,2%	159	14,3%	201	11,9%
Ordenadores	237	7,4%	22	5,5%	84	7,6%	131	7,8%
Standby	341	10,7%	46	11,4%	115	10,4%	181	10,7%
Resto Electrodomésticos	112	3,5%	20	5,1%	47	4,3%	45	2,7%
TOTAL	3.188	100%	398	100%	1.106	100%	1.684	100%

Unidad: ktep	España		Pisos		Unifamiliares	
Frigoríficos	975	30,6%	675	31,4%	300	28,9%
Congeladores	193	6,1%	81	3,8%	112	10,8%
Lavadoras	378	11,8%	263	12,3%	114	11,0%
Lavavajillas	193	6,1%	125	5,8%	68	6,6%
Secadoras	107	3,3%	65	3,0%	42	4,0%
Horno	263	8,3%	181	8,4%	82	7,9%
TV	388	12,2%	259	12,1%	129	12,4%
Ordenadores	237	7,4%	163	7,6%	74	7,1%
Standby	341	10,7%	247	11,5%	95	9,1%
Resto Electrodomésticos	112	3,5%	90	4,2%	22	2,2%
TOTAL	3.188	100%	2.149	100%	1.039	100%

Nota 10: Se indica de manera diferenciada el consumo de los electrodomésticos del asociado al Standby de los mismos, a efectos de mostrar la relevancia de este último consumo.

8.-Consumo Medio Anual de Electrodomésticos por Hogar

Unidad: kWh	España		Zona Atlántica		Zona Continental		Zona Mediterránea	
Frigoríficos	662	22,4%	552	18,5%	688	23,4%	673	22,8%
Congeladores	563	19,1%	664	22,3%	427	14,5%	614	20,8%
Lavadoras	255	8,6%	307	10,3%	240	8,2%	252	8,6%
Lavavajillas	246	8,3%	300	10,1%	253	8,6%	230	7,8%
Secadoras	255	8,6%	270	9,1%	237	8,1%	260	8,8%
Horno	231	7,8%	263	8,8%	258	8,8%	205	6,9%
TV	263	8,9%	149	5,0%	319	10,9%	255	8,7%
Ordenadores	172	5,8%	137	4,6%	188	6,4%	170	5,8%
Standby	231	7,8%	235	7,9%	231	7,9%	229	7,8%
Resto Electrodomésticos	76	2,6%	104	3,5%	95	3,2%	57	1,9%
TOTAL	2.953	100%	2.980	100%	2.938	100%	2.946	100%

Unidad: kWh	España		Pisos		Unifamiliares	
Frigoríficos	662	22,4%	651	22,2%	688	22,4%
Congeladores	563	19,1%	488	16,6%	631	20,5%
Lavadoras	255	8,6%	253	8,6%	261	8,5%
Lavavajillas	246	8,3%	242	8,2%	253	8,2%
Secadoras	255	8,6%	258	8,8%	252	8,2%
Horno	231	7,8%	223	7,6%	244	7,9%
TV	263	8,9%	249	8,5%	294	9,6%
Ordenadores	172	5,8%	249	8,5%	186	6,0%
Standby	231	7,8%	237	8,1%	216	7,0%
Resto Electrodomésticos	76	2,6%	86	2,9%	51	1,7%
TOTAL	2.953	100%	2.936	100%	3.076	100%

Nota 11: Medias referidas al parque de viviendas equipadas con cada tipo de electrodoméstico según la zona climática y tipo de vivienda.

ILUSTRACIÓN 4. CONSUMO DE ELECTRODOMÉSTICOS TOTAL Y MEDIO ANUAL POR HOGAR (EUROSTAT)

9.-Consumo Medio Anual por Electrodoméstico

Unidad: kWh	España	Zona Atlántica	Zona Continental	Zona Mediterránea
Frigoríficos	655	548	680	665
Congeladores	558	654	424	612
Lavadoras	254	305	239	251
Lavavajillas	245	299	252	230
Secadoras	255	270	237	260
Horno	229	262	258	202
TV	119	67	144	116
Ordenadores	145	123	160	140

Unidad: kWh	España	Pisos	Unifamiliares
Frigoríficos	655	651	662
Congeladores	558	488	623
Lavadoras	254	253	258
Lavavajillas	245	242	251
Secadoras	255	258	252
Horno	229	227	234
TV	119	119	119
Ordenadores	145	140	156

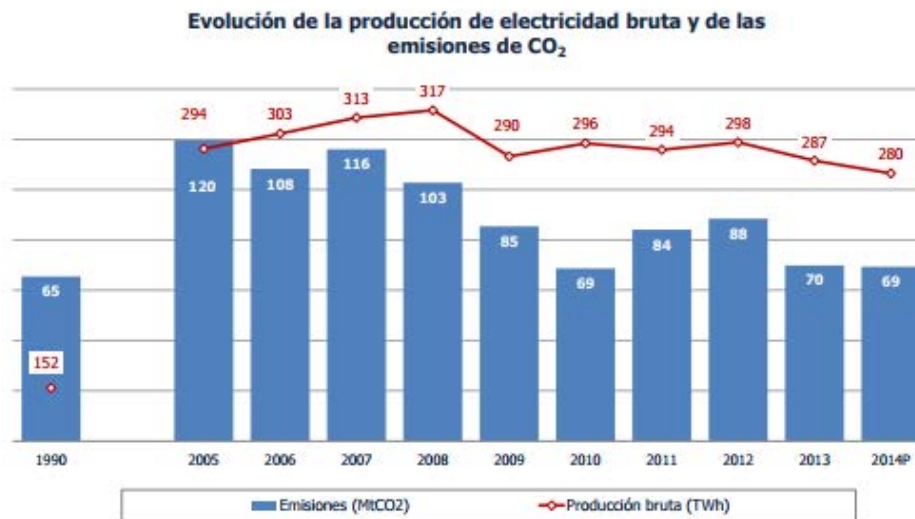
Nota 12: Medias referidas al número total de equipos según la zona climática y tipo de vivienda.

ILUSTRACIÓN 5. CONSUMO MEDIO ANUAL POR ELECTRODOMÉSTICO (EUROSTAT)

Todo esto nos indica de la importancia de saber ahorrar electricidad en el hogar, especialmente relacionado con el gasto de los electrodomésticos, que como se ha visto antes, constituye el 55,2 % del consumo eléctrico en el hogar. De este gasto, los electrodomésticos que más contribuyen al mismo son los frigoríficos y congeladores, aunque en nuestro caso, no nos interesan ya que no son electrodomésticos que se vayan a planificar con nuestra aplicación, ya que están siempre encendidos. Por tanto, sin contar con frigoríficos ni congeladores, los que más gastan electricidad son las lavadoras, lavavajillas y secadoras. Todos ellos son electrodomésticos de uso prácticamente diario, y cuyos gastos eléctricos se pueden planificar y gestionar para garantizar un mayor ahorro en nuestra factura de la luz.

Respecto al **ámbito medioambiental**, el ahorro de electricidad que supondría que la inmensa mayoría de los consumidores tuviera instalada nuestra aplicación ayudaría a reducir las emisiones de CO2 para generar electricidad.

En la siguiente figura se muestran dos gráficas. La primera representa las emisiones de CO2 generadas por la producción de electricidad (en bruto) en España, y la segunda muestra el porcentaje de emisión por sectores.



Distribución por sectores de las emisiones de CO₂ equivalente - Año 2013

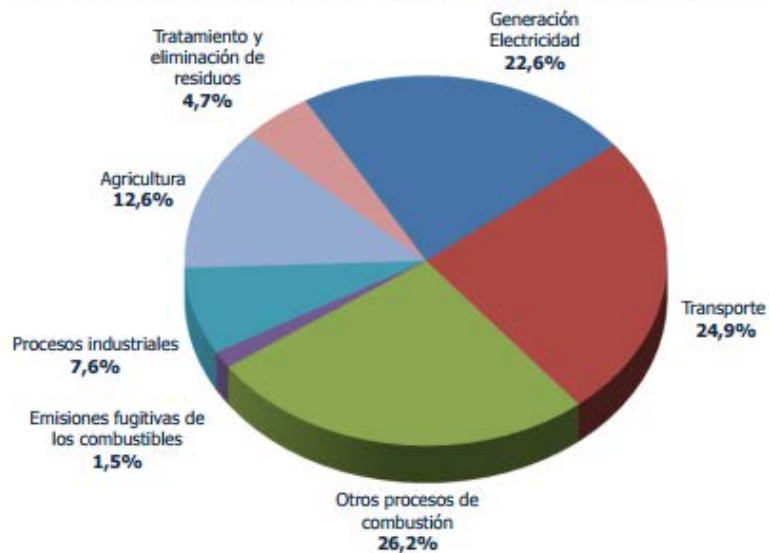


ILUSTRACIÓN 6. GRÁFICAS DE ANÁLISIS DE EMISIONES RELACIONADAS CON PRODUCCIÓN ELÉCTRICA

De estos datos se puede extraer que la generación de electricidad supone un 22,6 % de las emisiones de CO₂, y que la distribución de nuestra aplicación podría disminuir considerablemente las emisiones de ese sector, ya que el consumo eléctrico de los hogares

españoles supone una parte importante de esas emisiones, y provocaría un beneficio para el medioambiente al reducir así la contaminación atmosférica.

2.4.2.- Evolución de las tarifas eléctricas en los últimos años y tendencia futura

En la actualidad, España tiene en vigor la tarifa de la luz por horas desde que el Ministerio de Industria realizó la reforma del mercado eléctrico. Pese a esta medida para favorecer al pequeño consumidor, **España sigue siendo uno de los países europeos con mayor precio por kilovatio hora (kWh).**

En España, el precio del kilovatio hora (kWh) ha disminuido en los últimos años debido a que se ha reducido el peaje de acceso. Pero el precio del kilovatio ha aumentado. En España, el precio del kWh es de 0,182 euros, uno de los más altos de Europa.

En la siguiente gráfica se nos muestra el precio del kWh consumido.

La línea en azul representa los precios en España para el período 2004 – 2015 mientras que la línea verde representa la media de los precios en Europa.

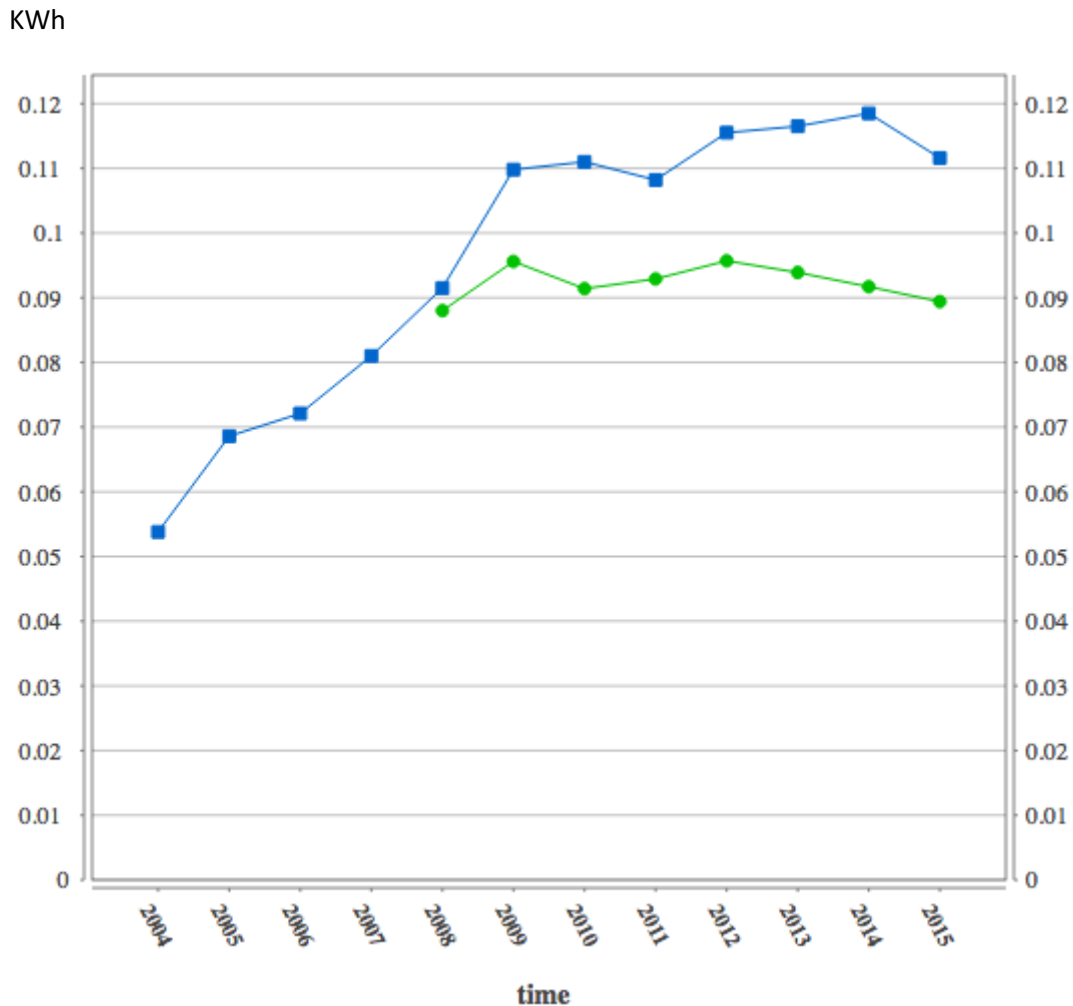


ILUSTRACIÓN 7. EVOLUCIÓN PRECIOS TARIFAS ÚLTIMOS AÑOS (EUROSTAT)

2.4.3- Conclusión al análisis del entorno socioeconómico

La distribución de nuestra aplicación a la gran mayoría de los hogares en España supondría un ahorro energético notable a nivel nacional. Las familias españolas serían capaces de ahorrar bastante en su factura de la luz y con ello su economía mejoraría a medio y largo plazo, por tanto, la estabilidad económica del país se consolidaría, provocando mayor bienestar social y calidad de vida.

Además, el ahorro energético que supondría a nivel nacional provocaría una disminución notable en las emisiones de CO2, beneficiando así al medio ambiente, por no olvidar que las

disminuciones de CO2 también mejorarían la calidad de vida en las ciudades al evitar mayor contaminación.

2.5.- Marco regulador

Este proyecto debe atenerse a la legislación española. En este apartado se describen las leyes y condiciones de uso que debe cumplir la aplicación que nos atañe, tanto en el marco de aplicaciones informáticas como en el del uso de los servicios proporcionados por la página web de ESIOS.

2.5.1.- Aspectos legales generales

Los aspectos legales que debe cumplir esta aplicación son:

- **Funcionalidades:** Siempre se debe solicitar al usuario su permiso para ejecutar determinadas funcionalidades que puedan influir en sus derechos como usuario, siempre que estas funcionalidades sean lícitas y estén en regla dentro de la legalidad y marco legal.
- **Privacidad:** Si se pretende obtener datos personales de los usuarios deberán obtenerse únicamente los datos de usuario estrictamente necesarios. El usuario deberá poder controlar su configuración de privacidad.
- **Derechos propios y de terceros:** Toda aplicación móvil está compuesta por una serie de elementos como librerías, bases de datos, elementos gráficos, textos, imágenes... Para poder publicarlos y/o utilizarlos, se deben tener las licencias necesarias que nos lo permitan.
- **Condiciones de uso:** Se deberán proporcionar un texto de condiciones legales, de uso y de licencia para poder utilizar nuestra aplicación, y requerir al usuario que para poder utilizar nuestra aplicación deberá aceptar dichos términos. Siempre que dichos términos sean correctos dentro de la legalidad, serán la mejor defensa en caso de posibles reclamaciones futuras.
- **Información y permisos:** Se deberá aclarar al usuario el requerimiento de utilización de recursos personales como acceso a agenda de contactos, contenidos del móvil, cesión de datos, o pagos, y deberá aceptar explícitamente el acceso por nuestra parte a esta información.

- **Informar al usuario:** Se deberá informar a los usuarios de los aspectos legales relacionados con la utilización de nuestra aplicación, indicándolo en textos de condiciones legales.
- **Publicidad:** En caso de incluir publicidad en nuestra aplicación, se deberá identificar como tal y se deberá cumplir la Ley 34/1988, de 11 de noviembre, General de Publicidad.
- **Condiciones de Google Play:** Se deberán aceptar las condiciones impuestas por Google Play a la hora de comercializar nuestra aplicación en su mercado de aplicaciones.

2.5.2.- Legislación concreta

Así mismo, la aplicación debe cumplir con las siguientes leyes:

1. **Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal:** Establece la ley que busca garantizar y proteger los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, así como su honor e intimidad personal y familiar.
2. **Ley Orgánica 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y de Comercio Electrónico:** Establece las obligaciones que tienen los prestadores de servicios informáticos para con los consumidores. Se debe proporcionar información a los usuarios sobre quiénes somos, como nuestro nombre, domicilio, etc. y las responsabilidades que tenemos.
3. **Ley 34/1988, de 11 de noviembre, General de Publicidad:** En caso de albergar publicidad en nuestra aplicación, se deberá identificar como tal y deberá ser lícita, según lo indicado en el artículo 3 de esta ley.
4. **Ley 23/2006, de 7 de julio, por la que se modifica el texto refundido de la Ley de la Propiedad Intelectual, aprobado por el Real Decreto Legislativo 1/1996, de 12 de abril:** Se deberá atener a esta ley, para proteger la originalidad de esta aplicación.
5. **Ley 24/2015, de 24 de julio, de Patentes:** En caso de querer establecer una patente para este proyecto.

2.5.3.- Herramientas utilizadas

Esta aplicación deberá cumplir con los estándares técnicos establecidos para el uso de las herramientas utilizadas, así como con los estándares para la implementación y desarrollo de la tecnología de la misma.

2.5.4.- Servicios de la web de Esios

La aplicación deberá garantizar que el uso de la API de ESIOS es lícito y no se compartirá el token personal con terceros. Además, se hará un uso correcto de la funcionalidad de la API proporcionada por la Red Eléctrica de España.

Como usuarios de la API de Esios, ya que utilizamos sus servicios para consultar las tarifas eléctricas, deberemos cumplir con las condiciones de acceso y uso de los datos, información y funcionalidades que proporciona.

En estas condiciones, se establece en calidad de usuario del sitio web de Esios como todo aquél que acceda al mismo y el cual acepta, desde el mismo momento del acceso al sitio, las condiciones de acceso y uso en relación con los servicios que prestan. Se establece por tanto un vínculo del usuario a estas condiciones por el simple hecho de utilizar los datos, información, funcionalidad y documentos contenidos en el sitio web.

Las condiciones de acceso y uso establecen que al acceder al sitio web se da por hecha la lectura previa de las condiciones de uso del sitio, y que en el caso de atenerse a dichas condiciones, el usuario deberá abstenerse de acceder al sitio web. Se establece el contenido como gratuito, salvo para aquellos en los que explícitamente se establezcan como de pago o suscripción. Además, asegura que el usuario se compromete a utilizar los servicios que proporcionan sin fines ilícitos o a contravenir la legislación vigente.

Respecto a la Política de Privacidad, se establece la navegación anónima por el sitio web, conservando de los usuarios únicamente:

- El nombre del dominio del proveedor (ISP) y/o dirección IP que le da acceso a la red, para elaborar estadísticas sobre los países y servidores que más utilizan sus servicios.
- Fecha y hora de acceso. Esto lo utilizan para evitar problemas de saturación de solicitud a sus servidores.
- La dirección de internet desde la que partió el enlace que dirige a su sitio. Esto es para conocer la efectividad de sus enlaces y poder mejorar sus servicios.
- El número de visitantes diarios de cada sección, para conocer las áreas de mayor éxito y mejorar sus servicios.

También hablan de la política que establecen con los Enlaces, donde afirman que la web proporciona enlaces a contenidos de terceros, y que la Red Eléctrica no se hace responsable de los usos que se puedan llevar a cabo por los usuarios, ni de los contenidos que hay en dichos enlaces.

También establecen la política en caso de querer crear un enlace de un sitio web a la de la Red Eléctrica, estableciendo como la página principal del sitio web del que se quiere crear el enlace

en Esios como la página a la que se redirija al introducir la URL absoluta. También se prohíben el uso de etiquetas iFrame ni la inclusión de marcas comerciales en la URL de la web de ESIOS.

En las condiciones respecto a las condiciones respecto al contenido en el Sitio Web, establecen la multimedia, software y diseño gráfico como su propiedad, salvo el contenido que se indica explícitamente como de otro titular.

Nosotros, como usuarios, estamos ligados a las condiciones (respecto a la reproducción, comunicación pública, distribución, modificación o cualquier otra acción similar) siguientes:

- Mencionar de forma clara y explícita a Red Eléctrica como fuente de la información y de los datos.
- No alterar ni desnaturalizar el sentido de la información ni de los datos.
- Mencionar la fecha de la última actualización de la información y los datos.
- No se podrá sugerir o indicar que el titular de la información patrocina o apoya la reutilización que se realice.

El usuario, según estas condiciones, se compromete a hacer un buen uso de los datos, contenidos y/o servicios presentes en el sitio web de ESIOS conforme a la ley vigente. Se prohíbe expresamente la utilización de estos datos y servicios con fines comerciales. La reutilización de estos datos y contenidos está sujeta a las condiciones establecidas para licencias-tipo en el artículo 4.2. b de la Ley 37/2007 de 16 de noviembre.

La Red Eléctrica de España no se hace responsable de las consecuencias de la utilización de los servicios que proporcionan, de forma directa o indirecta sobre los usuarios y determina que la prestación de sus servicios es indefinida, reservándose el derecho a interrumpir, suspender o terminar la prestación a cualquiera que integra el sitio web.

2.6.- Búsqueda de aplicaciones similares

La búsqueda de estas aplicaciones se ha realizado en Google Play, Apple store y en cualquier entorno que pudiera proporcionar ejemplos de aplicaciones que realicen funcionalidades similares a las que se pretende conseguir con el desarrollo de nuestra aplicación. Se ha buscado en función de si éstas tenían implementado un planificador de tareas del hogar que dependiendo del gasto energético y de las tarifas eléctricas organizara las tareas a realizar para minimizar el gasto eléctrico. La búsqueda se realizó en etapas tempranas del proyecto con el objetivo de recabar información acerca de aplicaciones similares y de la posibilidad, tanto de

incorporar características importantes que se nos hubieran pasado por alto, como de corregir errores encontrados en dichas aplicaciones.

Las aplicaciones encontradas se listan a continuación, junto con una descripción general de cada una, y analizando las distintas funcionalidades que ofrecen. Las aplicaciones son las siguientes:

- Boltio** Se trata de una aplicación para Android que es capaz de indicar el precio por hora de la tarifa eléctrica a cada hora del día, e indicar cuáles son las horas en las que la tarifa es más barata a lo largo del día. También, como se pueden ver en las capturas a la derecha, puede informar del coste de utilizar un aparato eléctrico durante una hora, en función de la hora del día y el aparato a utilizar.

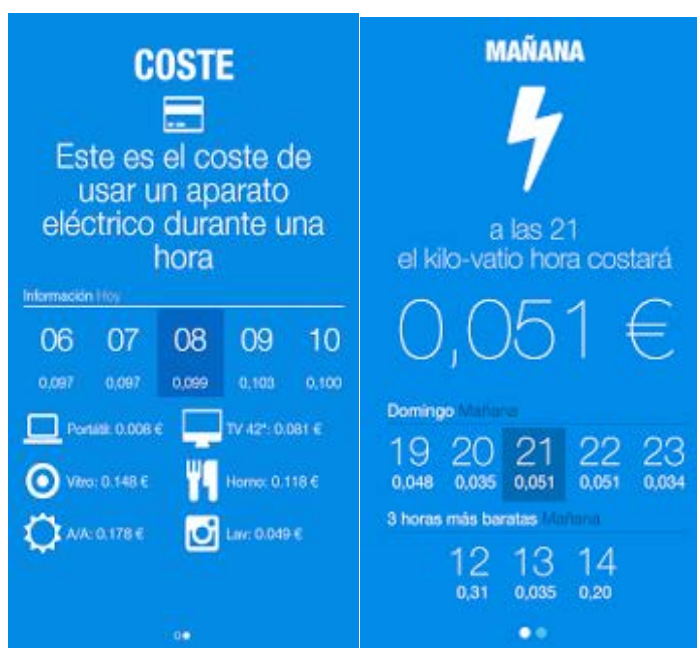


ILUSTRACIÓN 8. APLICACIÓN BOLTIO

- **Ahorra en luz. Precio luz hora**

Esta aplicación, que se puede encontrar en Google Play, permite seleccionar entre tarifa nocturna y general, y además de que se pueden obtener los datos actualizados de las tarifas diariamente (20:30 todos los días) permite consultar el precio medio y si a una determinada hora, las tarifas están por encima o por debajo de dicho precio. Además, proporciona avisos al usuario para que a las horas de menor coste pueda utilizar sus electrodomésticos y así ahorrar dinero.

Dispone de un apartado de consejos donde el usuario puede consultar qué electrodomésticos utilizar y las razones para hacerlo.

- **Tarifazo**

Tarifazo. Tarifa de la Luz es una aplicación encontrada en Google Play que proporciona múltiples funcionalidades. Entre ellas, y aparte de proporcionar el precio de la tarifa eléctrica por hora, permite comparar los precios entre distintas horas del día para así asesorar al usuario a la hora de elegir el momento del día en el que utilizar un electrodoméstico.



ILUSTRACIÓN 9. APLICACIÓN AHORRA EN LUZ. PRECIO LUZ HORA

- **Lupbak**

Se trata de una aplicación para Android que dispone de un seguimiento diario de los precios de las tarifas eléctricas por hora. Los precios se actualizan diariamente a las 20:30 y permite crear notificaciones para que el usuario sepa a qué hora de qué día le interesa estar atento, ya sea para consumir, o todo lo contrario. Se trata por tanto de una planificación manual y por alertas, lo cual guarda relación con los criterios de

búsqueda para el análisis comparativo de aplicaciones similares, y que se expondrán en el siguiente apartado, de ahí la elección de esta aplicación en el proceso de búsqueda.

- **My Power Consumption**

Esta aplicación está disponible para Android y permite introducir electrodomésticos y otros aparatos eléctricos del hogar y, para cada uno de ellos, calcular el coste por hora, día, mes o año. Se puede saber, introduciendo los datos del electrodoméstico, el coste medio de éste, lo cual permite planificar al usuario y decidir el electrodoméstico cuyo uso es más barato en un momento dado y a largo plazo. Todo esto se calcula en función de las tarifas eléctricas que son actualizadas cada día.



ILUSTRACIÓN 10. APLICACIÓN MY POWER CONSUMPTION

- **Energy Manager**

Energy Manager es una aplicación on-line gratuita con la que puedes realizar un seguimiento por hora o por día del consumo del gasto eléctrico. Permite indicar un objetivo de gasto eléctrico y establecer así una serie de objetivos a cumplir por parte del usuario para alcanzar la meta. Las tarifas se actualizan diariamente.

2.6.1.- Análisis comparativo de las aplicaciones encontradas

Se ha reunido información de las aplicaciones que muestran información sobre el consumo eléctrico con el fin de realizar un análisis comparativo.

La comparación de las distintas aplicaciones encontradas se ha realizado en función de:

- Si tiene algún tipo de planificador de tareas del hogar.
- Cada cuánto actualiza la tarifa eléctrica.

- **Boltio:** Se muestra el precio del kilovatio por hora de cada tipo de electrodoméstico para elegir el momento adecuado de encender cada uno, lo cual se trata de un planificador de uso de electrodomésticos. Las tarifas eléctricas se actualizan cada día.
- **Ahorra en luz. Precio luz hora:** Se actualizan las tarifas todos los días a las 20:30. Dispone de un apartado de consejos donde se indican los electrodomésticos que se deben usar.
- **Tarifazo:** Muestra la tarifa eléctrica por horas y el precio actual. Esta aplicación compara el coste con el resto de horas para recomendar al usuario cuando utilizar los electrodomésticos que tienen un mayor consumo eléctrico. Los datos son actualizados diariamente.
- **Lupbak:** Sólo se descarga los precios una vez al día para evitar generar tráfico innecesario al usuario. Los precios son publicados todos los días a las 20:30. Recomienda cuándo consumir electricidad, pero no tiene planificador de uso de los distintos electrodomésticos o aparatos del hogar como tal, si bien se trata de solo un aviso general dependiendo de los precios en cada hora del día.
- **My Power Consumption:** Aplicación de Google Play, permite capturar todos los consumidores de electricidad del hogar y buscar específicamente las mejores oportunidades de ahorro de energía. Permite calcular el gasto por hora, día, semana, mes o año. Aconseja, por así decir, qué electrodoméstico utilizar en un momento dado, lo cual se puede parecer a un planificador de tareas en tanto que en un momento determinado te dice la tarea a realizar que menos electricidad gasta. La tarifa se actualiza diariamente.

- **Energy Manager:** Es una aplicación que permite establecer una meta de ahorro, elaborar una lista de acciones que al usuario le gustaría tomar para alcanzar su objetivo y le muestra el progreso hacia su objetivo cada vez que inicie sesión. Es, por tanto, otro tipo de planificador de tareas, pero a largo plazo, haciendo que el usuario obtenga hábitos de uso eléctrico que le ayuden a ahorrar. No parece que sea el tipo de planificador de tareas del hogar que se busca, aunque podría tener una aplicación similar. Las tarifas son actualizadas cada día.

Estas son las aplicaciones encontradas basándonos en el periodo de publicación de las tarifas eléctricas y en si tienen algún tipo de planificador de tareas o lo que es lo mismo, planificador de uso de electrodomésticos y otros aparatos eléctricos del hogar.

Todas las aplicaciones encontradas actualizan los datos diariamente. Estos datos se publican alrededor de las 20:30 horas en la mayoría de las aplicaciones.

Respecto a si poseen planificador de tareas o no, por un lado, se han encontrado aplicaciones como **Ahorra en luz**, **Precio luz hora**, **Tarifazo**, **Boltio** o **Lupbak** que, dadas las tarifas por hora en un mismo día, recomiendan el consumo de electricidad en general, de cualquier aparato o electrodoméstico, a unas horas del día, pero no especifican qué aparatos eléctricos utilizar o en qué orden.

My Power Consumption y **Energy Manager** poseen planificadores de tareas más específicos.

My Power Consumption determina en qué momento del día es mejor utilizar un electrodoméstico u otro. Se trata, por tanto, de un planificador de utilización de electrodomésticos, o, dicho de otro modo, de tareas del hogar.

Energy Manager tiene otro estilo de planificación enfocada más al largo plazo del consumo eléctrico del usuario, al cual se le recomiendan hábitos y usos de aparatos a unas determinadas horas para que poco a poco el usuario vaya teniendo costumbres que le ayuden a ahorrar en el gasto del consumo eléctrico.

2.6.2.- Matriz comparativa

A continuación, se expone la matriz comparativa entre las distintas aplicaciones similares encontradas en el mercado actual. También se detallan los criterios de evaluación a la hora de determinar cuáles son las aplicaciones y funcionalidades que más se acercan a las características de la aplicación que atañe a este Trabajo de Fin de Grado.

2.6.2.1.- Criterios de comparación

- **Actualización diaria de las tarifas:** Indica si la aplicación en cuestión actualiza de forma diaria la información de las tarifas eléctricas.
- **Introducción manual de tareas y/o electrodomésticos:** Si requiere o permite que el usuario introduzca de forma manual las tareas a realizar, así como los electrodomésticos a utilizar.
- **Planificación manual (alertas):** Si la aplicación permite al usuario establecer alarmas o eventos para un momento determinado (hora, día) en función de lo que el usuario considere que es el momento de mayor ahorro para realizar una tarea.
- **Planificación automática (Selección de mejor opción):** Si la aplicación, dados los datos disponibles, es capaz de establecer una tarea o electrodoméstico a utilizar en un momento dado del día que permita ahorrar lo máximo posible.
- **Planificación en función de cada electrodoméstico de manera unitaria:** Si la aplicación permite planificar en función del gasto de cada aparato eléctrico o electrodoméstico por separado, y determinar así cuál es el momento del día para utilizarlo obteniendo un mayor ahorro en función de las tarifas eléctricas de cada momento.

2.6.2.2.- Matriz

La matriz comparativa es la que sigue:

	Actualización diaria de las tarifas	Introducción manual de tareas y/o electrodomésticos	Planificación manual (alertas)	Planificación automática (Selección de mejor opción)	Planificación en función de cada electrodoméstico de manera unitaria
Boltio	Sí	No	No	Sí	Sí
Ahorra en Luz. Precio luz hora	Sí	No	Sí	Sí	No
Tarifazo	Sí	No	Sí	No	No
Lupbak	Sí	No	Sí	No	No
My Power Consumption	Sí	Sí	No	Sí	Sí
Energy Manager	Sí	Sí	No	Sí	Sí

ILUSTRACIÓN 11. MATRIZ COMPARATIVA DE APLICACIONES ENCONTRADAS

De estos datos podemos observar que, en todos los casos, las aplicaciones contienen la funcionalidad de obtener las tarifas de forma diaria y actualizada, como es el caso de nuestra aplicación, que también obtendrá los valores de todas las tarifas del día con una consulta a la API de ESIOS, realizando una petición del PVPC con un intervalo de horas entre las 00:00 y las 23:59, para así obtener todas las tarifas que haya en ese intervalo de horas.

Respecto a la introducción manual de electrodomésticos y/o tareas que consuman electricidad, tan solo **MyPowerConsumption** y **Energy Manager** ofrecen esta funcionalidad, lo que nos demuestra que nuestra aplicación no cuenta con competidores que ofrezcan estas funcionalidades y nos diferencia dentro del mercado. Respecto a la planificación manual y de alertas, existe variedad en los resultados. En el caso de nuestra aplicación, toda la planificación de tareas será manual, con lo que dentro del mercado nos encontraremos prácticamente el mismo número de competidores que tengan como que no tengan este tipo de planificación.

También existen aplicaciones que planifican de forma automática el consumo, de forma que el sistema elige la mejor opción de consumo, indicando a qué horas debe consumir electricidad el usuario. Este tipo de funcionalidad no la implementamos, ya que solo implementamos el cálculo de gastos y de representación de gráficas evolutivas del gasto, pero no se realiza la planificación de las tareas de forma automática, con lo que tendremos que diferenciarnos de este tipo de aplicaciones para ganar notoriedad en el mercado.

2.6.3.- Método de publicación de tarifas eléctricas

Existen sitios web donde se publican los precios de las tarifas eléctricas e incluso se realizan análisis de estos datos, y en los que puedes exportar dichos precios en distintos formatos (CSV, JSON, EXCEL). Esta url lleva a uno de estos sitios web, donde te permite exportar los datos:

https://www.esios.ree.es/es/analisis/1013?vis=1&start_date=28-01-2016T00%3A00&end_date=28-01-2016T23%3A00&compare_start_date=27-01-2016T00%3A00&groupby=hour&compare_indicators=1014,1015

En la página mostrada en la siguiente ilustración, abajo a la izquierda te permite exportar en alguno de los tres formatos anteriormente citados pulsando “Exportación”.

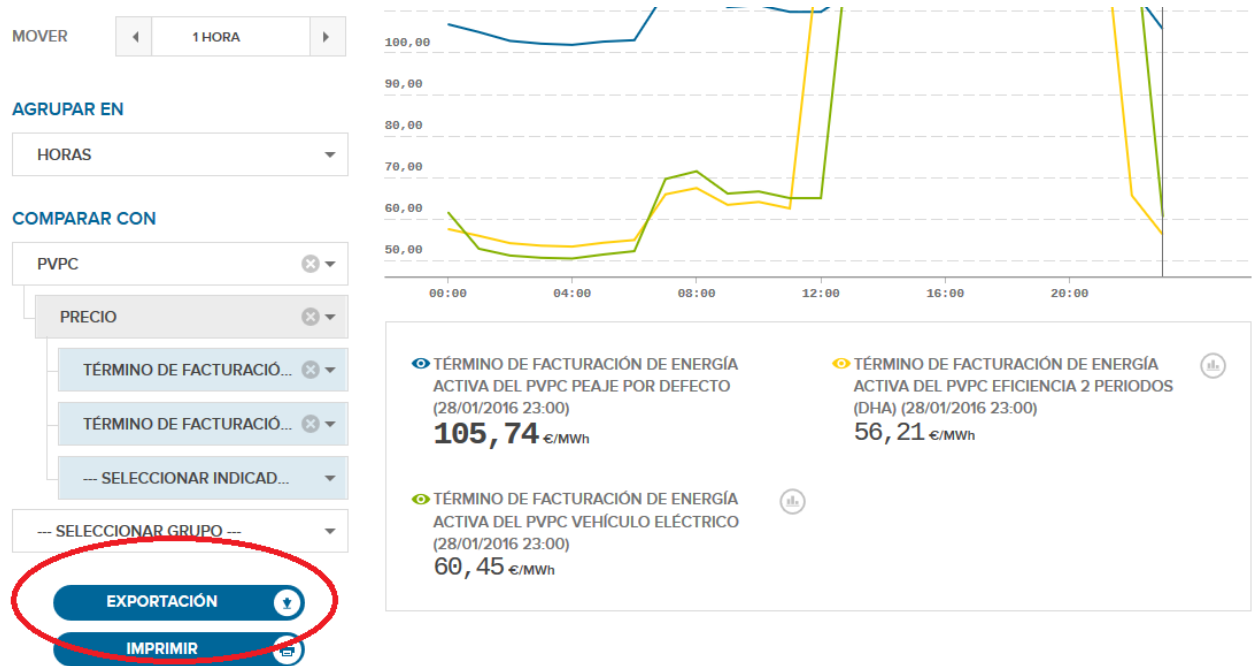


ILUSTRACIÓN 12. CAPTURA OBTENCIÓN TARIFAS ELÉCTRICAS DESDE PÁGINA WEB

También existen distintas API REST para poder tener acceso al precio de la electricidad de una forma más manual. Estas API son solo para acceder a los datos y no se puede añadir, modificar, ni borrar los datos ya existentes. Para acceder a ello se hace por medio de unas URLs que se pueden invocar por método GET modificando los ficheros php (en caso de tratarse de una aplicación web móvil) u obteniendo del response de dicha solicitud cliente el json y deserializándolo mediante el uso de la librería org.json en java.

En este proyecto, sin embargo, no se ha necesitado utilizar PHP, y en líneas generales, la implementación de esta funcionalidad ha sido simple pero eficaz, en tanto que cumple con la función de obtener la tarifa dada una fecha y hora determinadas.

El método utilizado consiste en realizar una llamada a la API de ESIOS por URL consultando el intervalo de hora del que necesitemos saber la tarifa, mediante una solicitud cliente pasándole los valores de dicho intervalo de hora y el tipo de tarifa que queremos saber en la URL de la petición. La respuesta de la API nos proporciona un JSON, en el que aparecen listados una serie de datos, y de los cuales solo nos importa el de la tarifa o las tarifas del intervalo de hora que le hemos pasado en la consulta. Esta respuesta de la API se puede almacenar como un String de java, y gracias a las librerías de org.json se pueden obtener los objetos y arrays de objetos que hay contenidos en el mensaje de respuesta, que es el código de un json. Leemos entonces

el json y obtenemos el valor numérico de la tarifa, lo cual lo almacenamos en la aplicación para utilizarlo en las operaciones de cálculo de gasto eléctrico en el hogar.

2.6.4.- Alternativa: Obtención de tarifas utilizando PHP

Durante la fase de investigación para conseguir implementar esta funcionalidad, se encontraron diversas formas de hacerlo, por ello, y con el fin de mostrar alternativas posibles a lo realizado en este proyecto, se describe la siguiente metodología.

En el caso de realizar el desarrollo de una aplicación web móvil (no Android), se requeriría el uso de otros recursos. Se podría crear una aplicación en **PHP** que leyera el **JSON** obtenido a partir del servicio web proporcionado y después extraer la información útil de este JSON.

Habría que crear un **script en PHP que leyera el JSON generado por la API** y mostrara la información útil en la interfaz web.

Podría crearse un formulario que enviara los datos de un input a sí mismo mediante el **método GET**. Estos datos serían procesados por PHP, comprobando primero que los datos del input no vienen vacíos. Si es así, obtendremos el JSON que genera la API gracias a la función *file_get_contents*. *El contenido del JSON lo almacenaremos en la variable \$html*.

Una vez que hemos obtenido todo el código **html**, es decir, el **JSON** que nos envía la API, decodificaremos el **JSON** con la función *json_decode*.

Una vez que tenemos almacenado en la variable *\$json* toda la información del JSON obtenido de la API extraeremos la información útil de éste, para ello almacenaríamos en variables la información del JSON.

Por último, mostraríamos en la interfaz web la información proporcionada de la forma y diseño que queramos.

Siguiendo todos estos pasos se podrían publicar los datos de las tarifas eléctricas en la aplicación web del solicitante.

3.- Especificación de Requisitos

3.1.- Descripción del formato

- **Identificación de requisito:** Valor alfanumérico que identifica el requisito. Los requisitos funcionales comienzan por RF y los no funcionales por RNF. Seguido de los identificativos de funcional o no funcional (RF o RNF respectivamente) viene dado un número de dos cifras.
- **Nombre del requisito:** Valor que identifica el requisito en lenguaje natural.
- **Descripción:** Explicación breve de la funcionalidad del sistema que abarca el requisito.

3.2.- Requisitos Funcionales

Los requisitos funcionales han sido organizados en relación a las distintas funcionalidades del sistema que especifican. Éstas son tres:

1. **Funcionalidades Generales:** Representan requisitos de funciones del sistema genéricas y que solo sirven para el funcionamiento básico de la aplicación, no cumplen ninguna función específica del sistema. Algunos ejemplos son inicio de sesión, registro o cerrar sesión.
2. **Estadísticas:** Se corresponde con los requisitos relacionados con funcionalidades referentes a la representación gráfica de los gastos semanales de los usuarios.
3. **Planificación de tareas:** Se tratan de los requisitos de las funcionalidades que se encargan de planificar y organizar las tareas del hogar.

Funcionalidades Generales:

Identificación de requisito	RF01
Nombre del requisito	Iniciar sesión de usuario.
Descripción del requisito	El sistema permitirá iniciar sesión en su cuenta al usuario siempre que introduzca correctamente sus credenciales.

Identificación de requisito	RF02
Nombre del requisito	Registrar usuario.
Descripción del requisito	El sistema permitirá registrarse al usuario en la aplicación siempre que introduzca correctamente los datos de usuario en el formulario de registro.

Identificación de requisito	RF03
Nombre del requisito	Abrir menú principal.
Descripción del requisito	El sistema desplegará el menú principal de la aplicación siempre que se pulse el símbolo de menú en la parte superior izquierda de la aplicación.

Identificación de requisito	RF04
Nombre del requisito	Abrir menú secundario.
Descripción del requisito	El sistema desplegará el menú secundario de la aplicación siempre que se pulse el símbolo de menú en la parte superior derecha de la aplicación.

Identificación de requisito	RF05
Nombre del requisito	Cerrar menú.
Descripción del requisito	Si el usuario, una vez abierto cualquier menú, vuelve a pulsar el botón del menú o pulsa en otro lugar de la pantalla que no sea el menú, el menú de la aplicación se cerrará.

Identificación de requisito	RF06
Nombre del requisito	Cambiar opciones de cuenta.
Descripción del requisito	Si el usuario, en la ventana de opciones de cuenta, introduce nuevos datos de cuenta, los datos del usuario se modificarán y se tendrá que iniciar sesión de nuevo con los datos establecidos nuevamente.

Identificación de requisito	RF07
Nombre del requisito	Cancelar cambio de opciones de cuenta.
Descripción del requisito	El sistema permitirá al usuario cancelar los cambios en la configuración de la cuenta en caso de cambiar de opinión en el proceso y antes de aplicar los cambios.

Identificación de requisito	RF08
Nombre del requisito	Cerrar sesión.
Descripción del requisito	El sistema permitirá al usuario cerrar sesión en su cuenta.

Identificación de requisito	RF09
Nombre del requisito	Salir de la aplicación.
Descripción del requisito	El sistema permitirá al usuario salir de la aplicación.

Planificación de tareas:

Identificación de requisito	RF10
Nombre del requisito	Crear electrodoméstico.
Descripción del requisito	El sistema permitirá crear un electrodoméstico nuevo con un nombre y consumo determinados.

Identificación de requisito	RF11
Nombre del requisito	Cancelar creación de electrodoméstico.
Descripción del requisito	El sistema permitirá al usuario cancelar la adición de un electrodoméstico.

Identificación de requisito	RF12
Nombre del requisito	Consultar electrodomésticos guardados.
Descripción del requisito	El sistema permitirá al usuario consultar los electrodomésticos que tiene guardados.

Identificación de requisito	RF13
Nombre del requisito	Eliminar electrodoméstico guardado.
Descripción del requisito	El sistema permitirá eliminar un electrodoméstico guardado que haya elegido el usuario.

Identificación de requisito	RF14
Nombre del requisito	Crear tarea.
Descripción del requisito	El sistema permitirá crear una tarea nueva con un nombre y electrodoméstico asociado.

Identificación de requisito	RF15
Nombre del requisito	Cancelar añadir tarea.
Descripción del requisito	El sistema permitirá al usuario cancelar la adición de una tarea.

Identificación de requisito	RF16
Nombre del requisito	Consultar tareas guardadas.
Descripción del requisito	El sistema permitirá al usuario consultar las tareas que tiene guardadas.

Identificación de requisito	RF17
Nombre del requisito	Eliminar una tarea guardada.
Descripción del requisito	El sistema permitirá eliminar una tarea del hogar guardada que haya elegido el usuario.

Identificación de requisito	RF18
Nombre del requisito	Añadir tarea a una fecha.
Descripción del requisito	El sistema permitirá planificar una tarea para un día determinado.

Identificación de requisito	RF19
Nombre del requisito	Consultar tareas planificadas para un día.
Descripción del requisito	El sistema permitirá consultar las tareas planificadas para un día determinado.

Identificación de requisito	RF20
Nombre del requisito	Eliminar una tarea planificada para un día determinado.
Descripción del requisito	El sistema permitirá eliminar una tarea de un día determinado que decida el usuario.

Identificación de requisito	RF21
Nombre del requisito	Navegar entre meses en calendario de la agenda.
Descripción del requisito	El sistema permitirá al usuario navegar por los meses del calendario más allá del actual.

Identificación de requisito	RF22
Nombre del requisito	Consultar tareas de la semana.
Descripción del requisito	El sistema permitirá al usuario consultar un listado de las tareas que se han planificado para la semana actual.

Estadísticas:

Identificación de requisito	RF23
Nombre del requisito	Mostrar gráfica consumo semanal previsto.

Descripción del requisito	El sistema mostrará al usuario los datos de consumo semanal previsto en una gráfica.
---------------------------	--

Identificación de requisito	RF24
Nombre del requisito	Mostrar gráfica consumo semanal real.
Descripción del requisito	El sistema mostrará al usuario los datos de consumo semanal real en una gráfica.

Identificación de requisito	RF25
Nombre del requisito	Comparación entre valores de gráficas de consumo real y previsto semanales.
Descripción del requisito	El sistema permitirá al usuario comparar los valores de las gráficas de gastos reales y previstos semanales.

Identificación de requisito	RF26
Nombre del requisito	Datos de la gráfica de gasto previsto actualizados.
Descripción del requisito	Si los datos de los costes totales del día de la semana previstos se modifican, estos valores se actualizarán en la gráfica de gasto previsto semanal.

Identificación de requisito	RF27
Nombre del requisito	Datos de la gráfica de gasto real actualizados.

Descripción del requisito	Si los datos de los costes totales del día de la semana reales se modifican, estos valores se actualizarán en la gráfica de gasto real semanal.
---------------------------	---

Identificación de requisito	RF28
Nombre del requisito	Mostrar gasto en función de los electrodomésticos.
Descripción del requisito	El sistema permitirá consultar al usuario el gasto que llevan a cabo semanalmente todos los electrodomésticos guardados.

Identificación de requisito	RF29
Nombre del requisito	Mostrar horas de utilización prevista de los electrodomésticos guardados.
Descripción del requisito	El sistema permitirá consultar al usuario el número de horas previstas para ser utilizados los electrodomésticos guardados para la semana actual.

Identificación de requisito	RF30
Nombre del requisito	Mostrar horas de utilización real de los electrodomésticos guardados.
Descripción del requisito	El sistema permitirá consultar al usuario el número de horas reales utilizadas de los electrodomésticos guardados para la semana actual.

3.3.-Requisitos No Funcionales

Identificación de requisito	RNF01
Nombre del requisito	Aprendizaje de usuario.
Descripción del requisito	El tiempo que tarde un usuario en aprender a utilizar el sistema deberá ser inferior a 1 hora.

Identificación de requisito	RNF02
Nombre del requisito	Eficiencia.
Descripción del requisito	Toda funcionalidad del sistema deberá ser completada en un tiempo inferior a 4 segundos.

Identificación de requisito	RNF03
Nombre del requisito	Disponibilidad.
Descripción del requisito	El sistema deberá estar siempre disponible y contar con un plan de contingencia en caso de no ser posible.

Identificación de requisito	RNF04
Nombre del requisito	Avisos a usuario.
Descripción del requisito	El sistema avisará al usuario con mensajes por pantalla en caso de que el usuario cometa algún error en alguna operación, y dichos mensajes serán claros e informativos.

Identificación de requisito	RNF05
Nombre del requisito	Realización de las funcionalidades pese a acceso a registros.
Descripción del requisito	El sistema deberá garantizar la realización de las funcionalidades del propio sistema, no viéndose alterado el tiempo de respuesta si se consultaran o actualizaran la información y registros almacenados.

Identificación de requisito	RNF06
Nombre del requisito	Seguridad.
Descripción del requisito	El sistema garantizará la seguridad y confidencialidad de los datos que se manejan, tales como documentos, archivos y datos de usuario.

Identificación de requisito	RNF07
Nombre del requisito	Permisos.
Descripción del requisito	Los permisos de acceso al sistema solo podrán ser modificados por un administrador del sistema.

Identificación de requisito	RNF08
Nombre del requisito	Interfaz adaptable.
Descripción del requisito	El sistema deberá tener una interfaz adaptable a todo tipo de dimensiones de pantalla y dispositivos.

Identificación de requisito	RNF09
Nombre del requisito	Sistema Operativo de móvil usuario.
Descripción del requisito	El sistema operativo del móvil del usuario deberá ser Android v5.0 (Lollipop) o superior.

Identificación de requisito	RNF10
Nombre del requisito	Base de datos.
Descripción del requisito	El sistema utilizará Sqlite 3.20.1 como base de datos de la aplicación.

Identificación de requisito	RNF11
Nombre del requisito	Servidor.
Descripción del requisito	El sistema utilizará un servidor GlassFish 4.1.2

4.- Diseño de la arquitectura del sistema

El diseño de la arquitectura del sistema será expuesto desde diferentes perspectivas. Cada una aportará una visión diferente de la aplicación con el fin de explicar de la forma más comprensiva, detallada y congruente la estructura del sistema, tanto a nivel de interfaz de usuario como de lógica de negocio de la aplicación.

Primero se evaluarán las alternativas de diseño del sistema, y después las diferentes vistas del mismo, que estarán compuestas por el diseño de la interfaz, el modelo vista controlador, el modelo de Casos de Uso (CDU) en el que se explicarán los diferentes escenarios de ejecución del sistema dependiendo de la situación, los diagramas de Secuencia, en los que se explican de forma más detallada que en los CDU la secuencia de acciones que se lleva a cabo en los diferentes escenarios de uso de la aplicación, el modelo Conceptual, en el que se detallará el flujo de datos de la aplicación, y el modelo de Implementación, donde se exponen la comunicación entre los diferentes componentes del sistema.

4.1.- Evaluación de alternativas de diseño

A continuación, se describen y evalúan las alternativas de diseño más importantes que se podrían haber dado en vez de seguir con el diseño establecido en el proyecto. Primero, se describe la decisión de diseño que se habría tomado para cambiar la aplicación, después se analiza, descubriendo las ventajas e inconvenientes de llevar a cabo esta decisión, y, por último, a modo de conclusión, se determinan las razones de no optar por estas alternativas de diseño y si hacerlo por la solución que se ha llevado a cabo en este proyecto.

4.1.1.- Alternativa 1:

Una alternativa de diseño en nuestra aplicación sería que **las tareas del hogar se crearan el mismo día en el que se establecen dentro de la semana**. Es decir, al introducir la tarea en el día de la semana para planificar el gasto, también se crearía y por tanto no haría falta crear la tarea previamente y luego introducirla en el día de la semana. Por otro lado, esto provocaría

que la tabla de la base de datos de tareas del día (de la semana) no hiciera falta crearla, y por tanto mejoraría el rendimiento de acceso a base de datos y con ello la velocidad de carga de la aplicación, al no tener que almacenar tanto volumen de datos. **Sin embargo, la duración de cada tarea sí que habría que guardarla, y al insertar la duración de la tarea habría que hacerla persistente de alguna forma, lo que nos llevaría a crear otra tabla en base de datos.**

Por ello, esta opción de diseño no es tan buena, porque el diseño relacional y **la estructura de la base de datos no sería óptima, pese a que sería mucho más cómodo para el usuario al no tener que introducir tanta información para planificar su gasto semanal**, en este caso las Tareas del Hogar.

4.1.2.- Alternativa 2:

La otra alternativa de diseño sería **eliminar la introducción de electrodomésticos**. En su lugar, a la hora de introducir una Tarea del Hogar, introduciríamos el consumo por horas, es decir, en kilovatios por hora, pero sin indicar el nombre del electrodoméstico, únicamente el consumo en kilovatios por hora.

Esto nos **permitiría eliminar la ventana de Electrodomésticos de nuestra aplicación y hacer más fácil al usuario planificar el gasto semanal**, ya que no tendría que introducir tanta información para planificar su gasto.

Sin embargo, esto provocaría que el usuario **no podría planificar su gasto en función de sus electrodomésticos**, ya que no estarían identificados. Esta funcionalidad le sería privada, y **resulta muy útil saber qué electrodoméstico consume más a lo largo de la semana** para poder planificar mejor el gasto eléctrico.

En consecuencia, no se optó por esta opción alternativa de diseño de la aplicación, ya que consideramos más óptima la posibilidad de conocer el consumo semanal por cada electrodoméstico.

4.1.3.- Solución propuesta

Para la realización de este proyecto, se desecharon las anteriores alternativas por los motivos expuestos previamente en sus subsecciones. Ahora explicaremos porque se llevó a cabo la solución propuesta.

Primero, las tareas que se registran en la agenda para ser planificadas debían tener una referencia en base de datos que permitiera al usuario saber de qué recursos dispone para planificar, por ello se crearon electrodomésticos y tareas primero, y después las tareas del día como unión entre ambos recursos a la hora de planificar. Al unir los recursos utilizando un tercero, se pueden gestionar las estadísticas que se muestran al usuario por separado: por un lado, puedes mostrar al usuario el consumo por electrodoméstico, y por otro, por tarea del hogar que más consume.

Por ello, el modelo de datos es más sencillo de implementar, y más comprensible para el usuario. Además, permite que las operaciones del cálculo de estadísticas sean más eficientes.

En los siguientes apartados, se explicará el diseño de la arquitectura del sistema de la solución propuesta que hemos analizado.

4.2.- Diseño interfaz de la aplicación

Para realizar un análisis completo del sistema desarrollado, también es necesario explicar la interfaz de usuario. En nuestro caso, es una parte esencial del sistema, al tratarse de una aplicación móvil. La interfaz va a ser el medio por el que interactúen los usuarios con la aplicación, y por tanto deberá ser fácil de utilizar e intuitiva.

En el siguiente diagrama, se muestra el mapa de navegación de la aplicación. Este diagrama explica cómo se podrá mover el usuario por la aplicación. Cada nodo es una página y cada flecha indica el sentido de una página a otra a la que puede ir el usuario, pudiendo ser bidireccional, ya que puede irse desde ambas páginas a la otra.

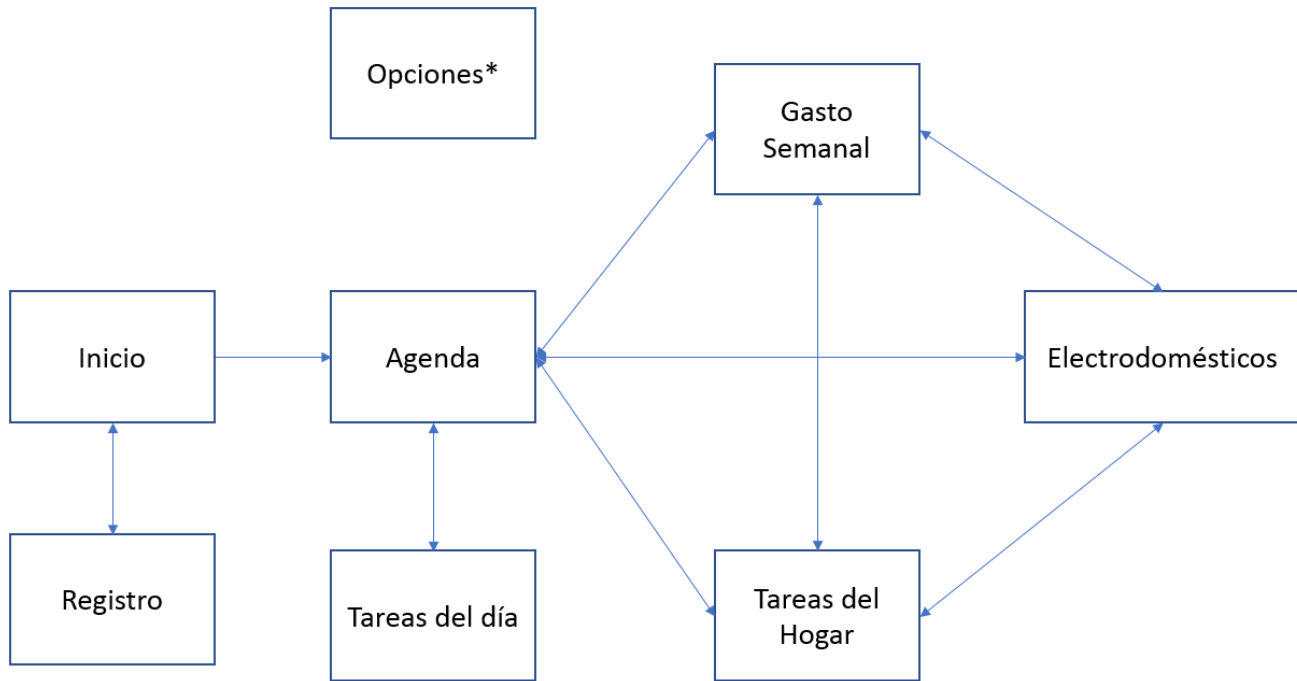


ILUSTRACIÓN 13. DIAGRAMA DE NAVEGACIÓN ENTRE PÁGINAS DE LA APLICACIÓN

*A la página de opciones se llega desde cualquier otra página una vez se ha iniciado sesión.

La aplicación tiene implementada un menú principal siempre presente en todas las páginas de la aplicación una vez inicias sesión desde la página de Inicio. Esto facilita que el usuario no se pierda en la navegación y le resulte más fácil utilizarla.

También está el menú secundario, que al igual que el principal, está en todas las páginas. En él, se encuentra la página de Opciones, y por eso es accesible desde todas las páginas como se muestra en la aclaración en el pie de la ilustración.

ILUSTRACIÓN 14. EJEMPLO DE INTERFAZ DE APLICACIÓN

A la derecha, tenemos una imagen que muestra un ejemplo de la interfaz utilizada para gestionar la planificación de recursos (electrodomésticos, tareas del hogar, etcétera).

En todos los gestores de recursos se utiliza el mismo esquema estructural.

Era necesario que se implementara la posibilidad de consultar, crear y eliminar los recursos, por ello, se decidió esta estructura.

Lo primero, se muestra el apartado para crear el recurso, ya que es lo primero que busca el usuario. Después, la consulta de los recursos que tiene, ya que después de crearlos, el usuario necesita consultar de qué dispone para planificar. Por último, hemos situado el apartado para eliminar el recurso, justo debajo de la tabla donde se consulta, para que pueda comprobar el recurso, en el caso de la imagen la tarea del día, que desea eliminar.

Por todos estos motivos, se ha tomado la decisión de implementar la interfaz de esta forma.



El resto de páginas no se comentan ya que no han requerido decisiones que merezcan la pena resaltar, ya sea por falta de complejidad o importancia dentro de la aplicación.

4.3.- Arquitectura general. MVC

La arquitectura general del sistema está organizada basada en el Modelo Vista Controlador (MVC), que consiste en dividir las funciones del sistema en tres: el modelo de datos se encarga de la persistencia y de cómo se organiza la información (BD), la vista se trata de la interfaz de la aplicación, y el controlador, que organiza y maneja la información obtenida del modelo y de la lógica de negocio interna de la aplicación y la muestra en la interfaz, es decir, la vista.

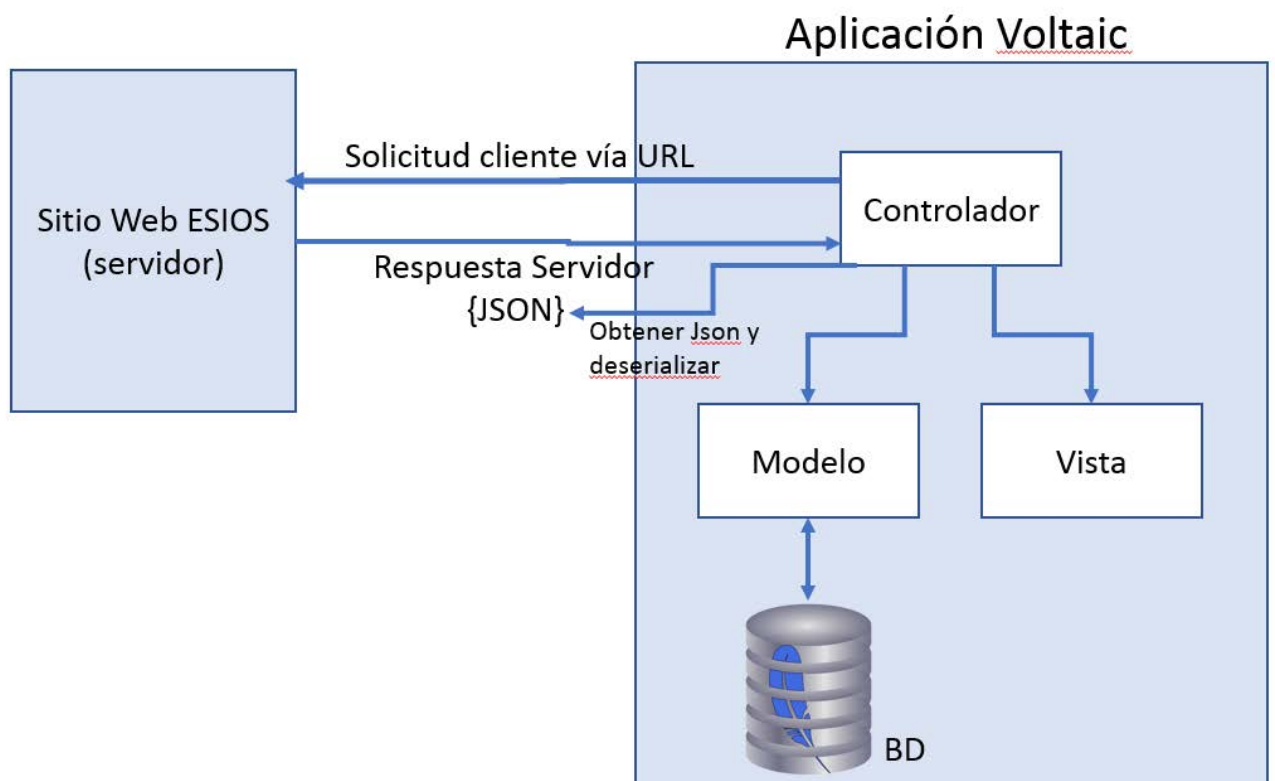


ILUSTRACIÓN 15. DIAGRAMA MVC DE APLICACIÓN

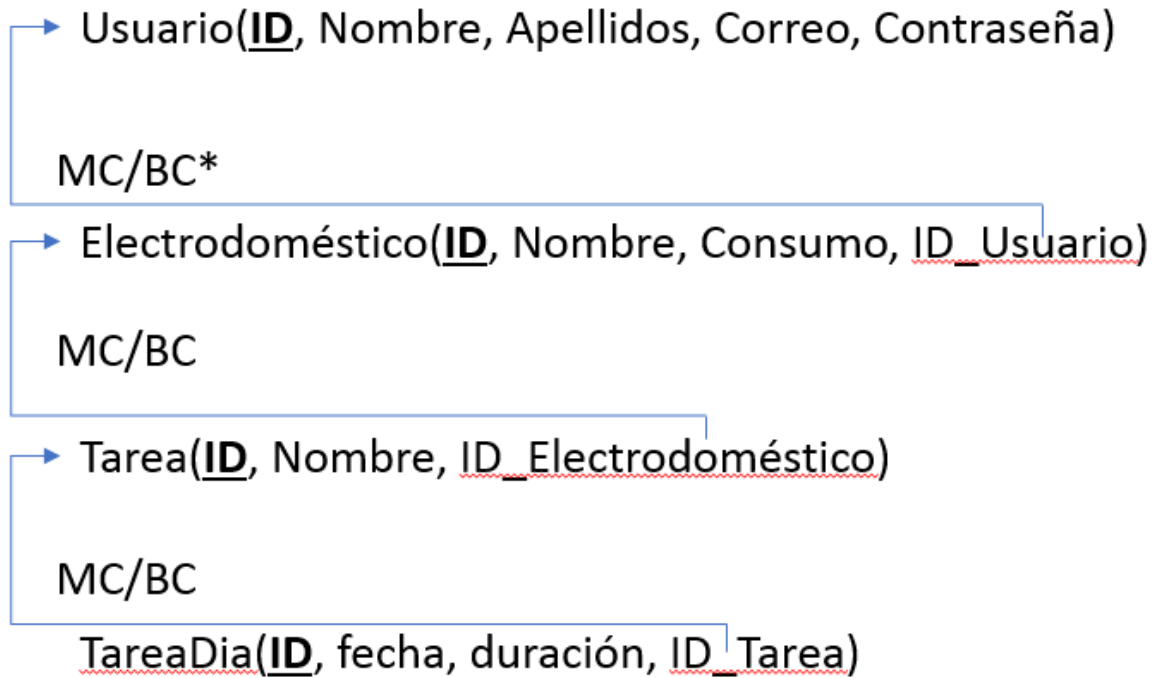
4.3.1.- Modelo

Para asegurar la persistencia de los datos que introduce el usuario para planificar su gasto y la información que se utiliza para calcular el consumo semanal se requiere de una base de datos.

Existen varias formas de implementar esta funcionalidad y tipos de bases de datos, en este proyecto se ha optado por utilizar una base de datos **SQLite**, por su facilidad de implantación en una aplicación Android y por su facilidad de uso en Android Studio. Por así decir, era la base de datos más compatible y cómoda de implementar.

Para implementar la base de datos, primero se realizó un diseño del esquema relacional, que indica la estructura de las tablas y atributos de éstas, así como las relaciones que guardan entre sí las tablas para almacenar información de la aplicación, tales como datos de usuario, tareas, electrodomésticos, etcétera.

El esquema relacional es el que sigue:



*MC: Modificación en Cascada; BC: Borrado en Cascada.

ILUSTRACIÓN 16. ESQUEMA RELACIONAL DE LA BASE DE DATOS DE LA APLICACIÓN

Las multiplicidades de las tablas se representan en el siguiente diagrama:

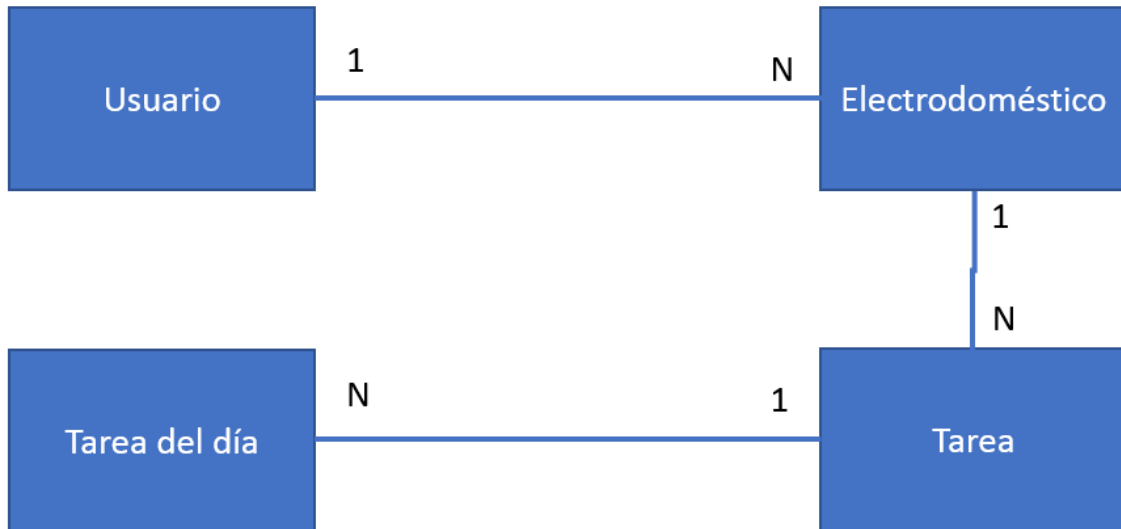


ILUSTRACIÓN 17. DIAGRAMA DE MULTIPLICIDADES DE TABLAS DE LA BD

Puede observarse las relaciones de clave foránea entre las tablas. Esto es así porque es necesario que los datos sean consistentes. Pongamos un ejemplo: si introduzco una tarea, en un día determinado, pero el electrodoméstico que estoy introduciendo para realizar esta tarea no existe, entonces no se podrá realizar un seguimiento del gasto, y la planificación sería imposible. Es por ello importante la consistencia de los datos que se manejan en esta aplicación, para poder utilizar los datos almacenarlos y así mostrar al usuario un análisis correcto del gasto eléctrico de su hogar.

El borrado y la modificación serán en cascada, ya que asegura la consistencia de la información almacenada. Un ejemplo de esto es si queremos borrar un electrodoméstico que tiene asignado unas cuantas tareas para unos días de la semana. Si borras ese electrodoméstico también se deberían borrar las tareas ya que resultaría imposible hacerlas sin dicho electrodoméstico. También favorece la fiabilidad de los cálculos de los gastos semanales al no existir tareas con un gasto vacío, por así decir.

4.3.2.- Vista

Para implementar la vista del sistema, se ha programado en lenguaje XML, y mediante el uso de etiquetas se han implementado las Activities del proyecto. Básicamente, una Activity es como una página de la aplicación, en la que se muestra el contenido de la interfaz de la aplicación.

En total, se han implementado 8 activities, que son:

- **activity_main.xml**: Página de inicio.
- **activity_registro.xml**: Página de registro.
- **activity_calendario.xml**: Página de Agenda con el calendario.
- **activity_opciones.xml**: Página de Opciones de cuenta.
- **activity_gasto_semanal.xml**: Página donde se muestran las estadísticas del gasto semanal.
- **activity_costes_elect.xml**: Página donde se añaden y consultan los electrodomésticos.
- **activity_crear_tarea.xml**: Página donde se añaden y consultan las tareas del hogar.
- **activity_tareas_dia.xml**: Página de cada día del calendario de la Agenda.

A cada una de estas activities se les integró el menú de la aplicación, para que el usuario pueda navegar por la aplicación de forma cómoda e intuitiva.

Todas estas activities conforman la interfaz de la aplicación, encargada de mostrar la aplicación al usuario, junto con la información de la base de datos, es decir, el modelo, y cuya navegación entre activities está controlada por el controlador.

4.3.3.- Controlador

Conforma la lógica de negocio de la aplicación, que se encarga de manejar la interacción entre activities y su navegación, la obtención de la información proporcionada por el modelo y el manejo de la información interna de la aplicación, así como de solicitar recursos externos, como es el caso de la solicitud de información de las tarifas eléctricas a la Web de ESIOS.

Los componentes del controlador son clases Java, y para cada activity de la capa de la vista, se ha implementado una clase java que se encarga de controlar la información que se muestra y cómo lo hace. Se encargan de cómo interactúan las activities en función de las entradas a la aplicación, como pulsar un botón o escribir en un campo.

Además, se han creado clases para conformar el modelo de datos, el cual está compuesto por las clases:

- **Electrodoméstico**: Se encarga de crear un objeto Electrodoméstico.
- **Tarea**: Crea un objeto de tarea del hogar.
- **TareaDia**: Crea un objeto de una tarea del día.
- **Usuario**: Para crear un objeto usuario.

Estos objetos son utilizados para extraer la información de la base de datos desde la capa de Controlador. Así, es posible obtener los datos concretos de las tablas de la base de datos.

Sin embargo, el principal componente de la capa de Control para administrar la base de datos y su información es la clase **DatabaseHelper**, que se encarga de almacenar los métodos de consulta a la base de datos. Cada vez que se quiere acceder a la base de datos, se crea un objeto de esta clase, que sirve como puente entre la capa de control y el modelo, y así es posible obtener la información de la base de datos desde la capa de controlador. Es posible por tanto enviar órdenes desde la capa de controlador a la base de datos, tales como almacenar un electrodoméstico, borrar una tarea, o mostrar una lista de las tareas de un día determinado.

4.4.- Modelo de Casos de Uso

A continuación, se representa el diagrama que explica gráficamente las secuencias de acción que lleva a cabo un usuario al operar con el sistema. Este diagrama muestra los diferentes cursos de acción que puede llevar a cabo el usuario al utilizar el sistema.

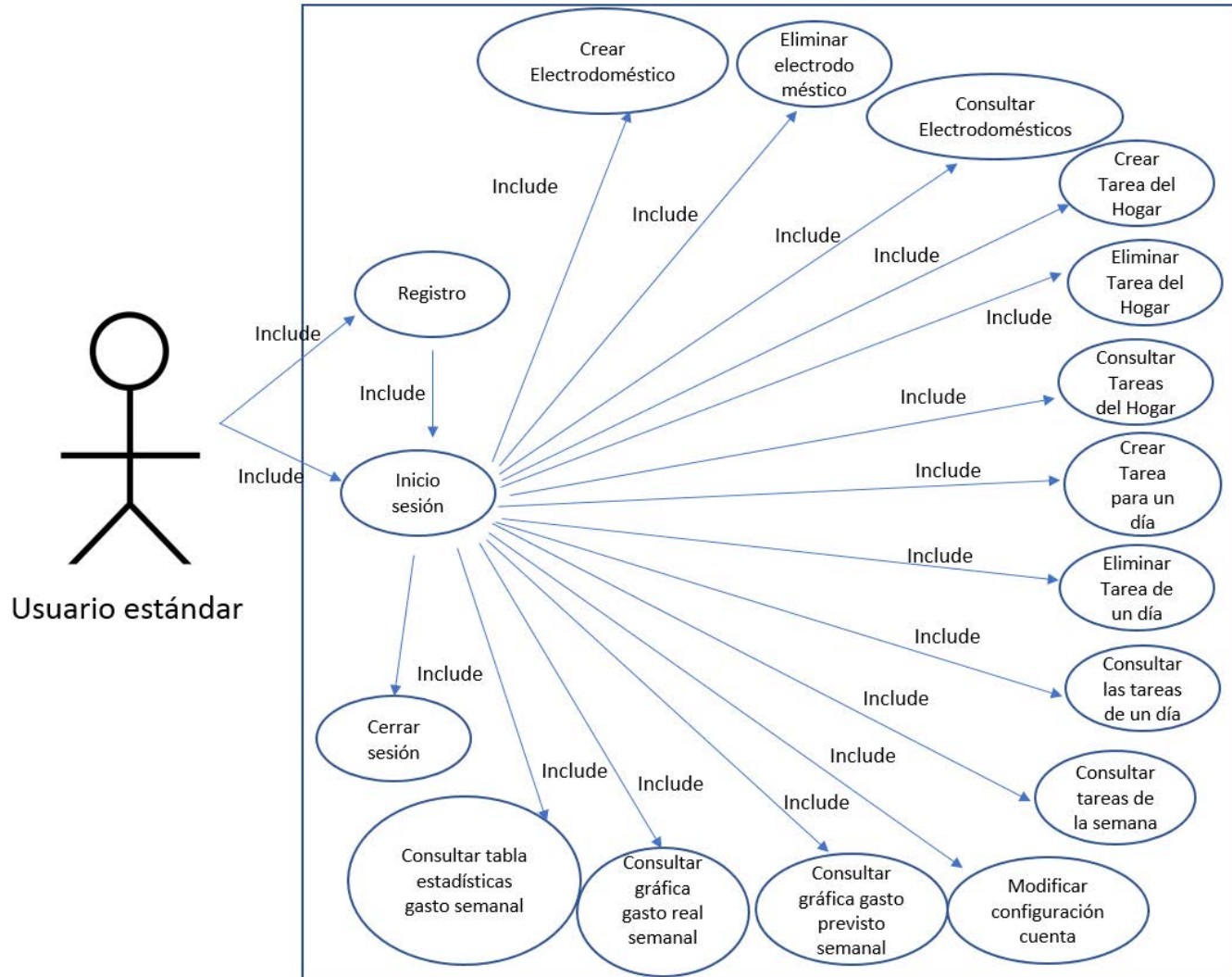


ILUSTRACIÓN 18. DIAGRAMA DE CASOS DE USO

4.5.- Diagramas de secuencia

Para explicar en mayor detalle los Casos de Uso mostrados en el apartado anterior, y para describir de forma más detallada las funcionalidades del sistema, desde una perspectiva más cercana a los métodos y clases que se utilizan, vamos a hacer uso de los diagramas de secuencia.

Dichos diagramas están agrupados por funcionalidades, que son las siguientes:

4.5.1.- Inicio Sesión y Registro

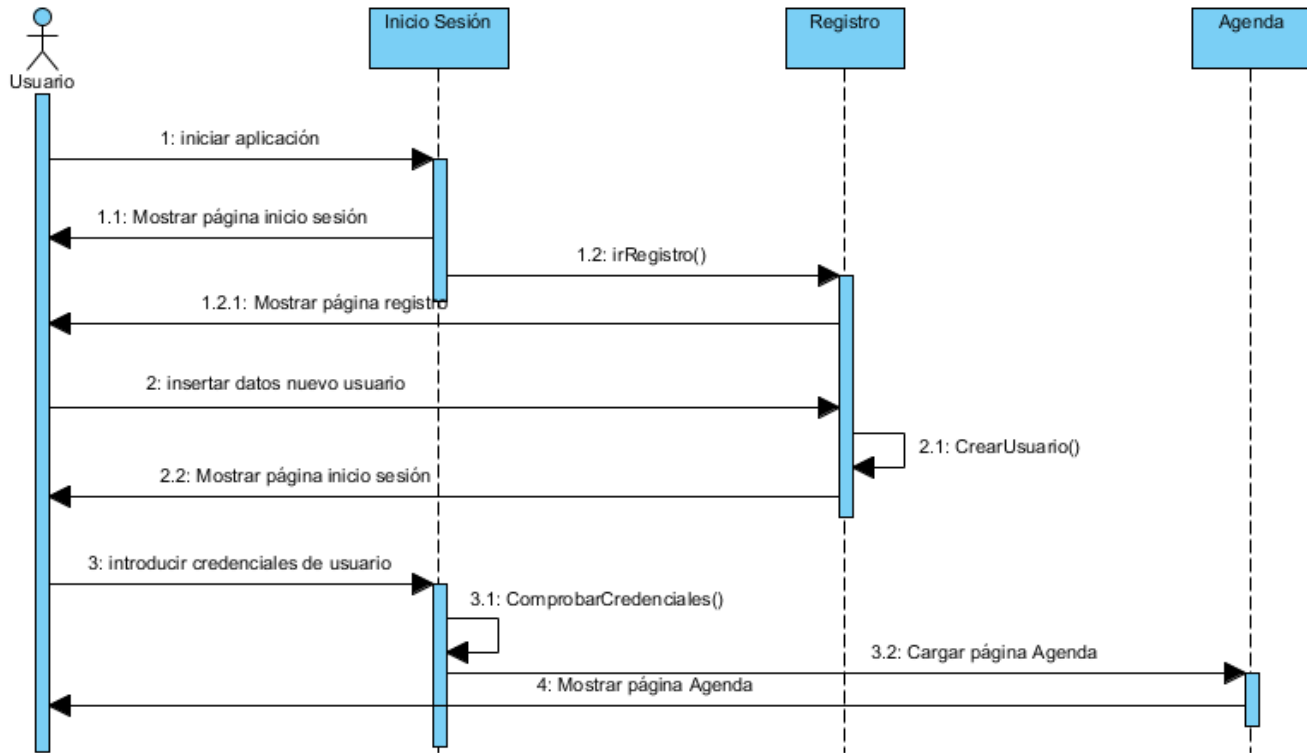


ILUSTRACIÓN 19. DIAGRAMA DE SECUENCIA DE INICIO DE SESIÓN Y REGISTRO

4.5.2.- Consultar, crear y eliminar Tareas del Día

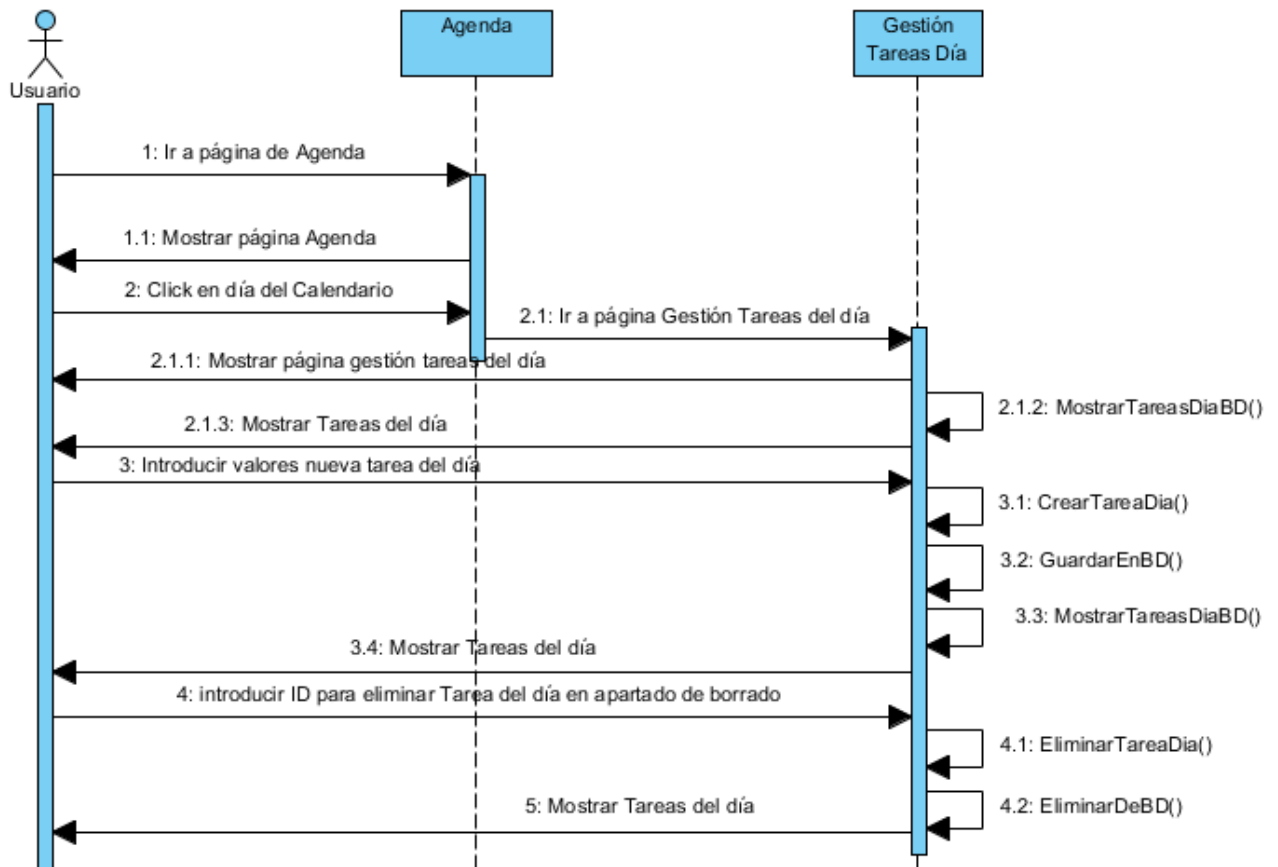


ILUSTRACIÓN 20. DIAGRAMA DE SECUENCIA DE GESTIÓN TAREAS DEL DÍA

4.5.3.- Consultar, crear y eliminar electrodomésticos

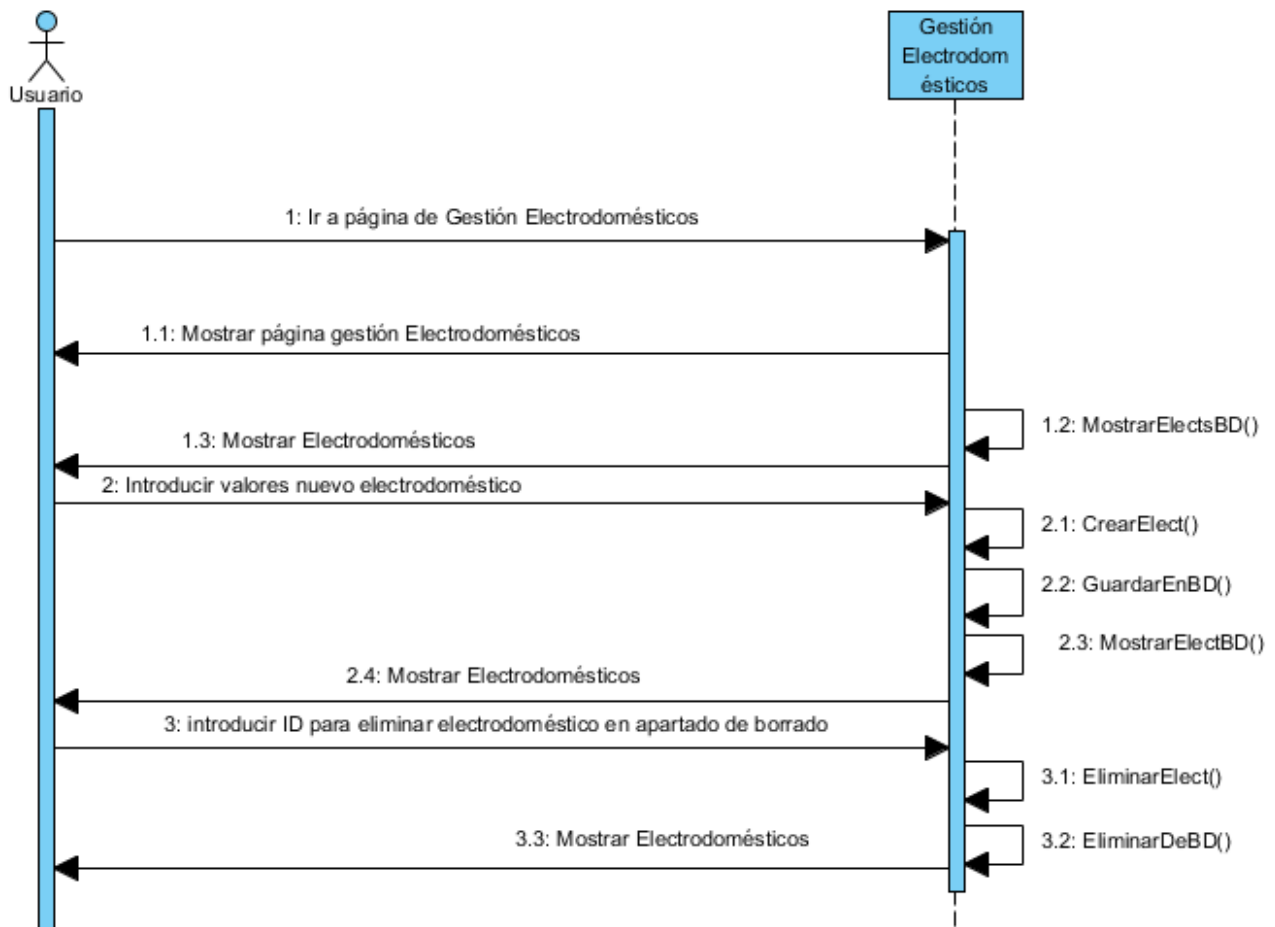


ILUSTRACIÓN 21. DIAGRAMA DE SECUENCIA DE GESTIÓN DE ELECTRODOMÉSTICOS

4.5.4.- Consultar, crear y eliminar Tareas

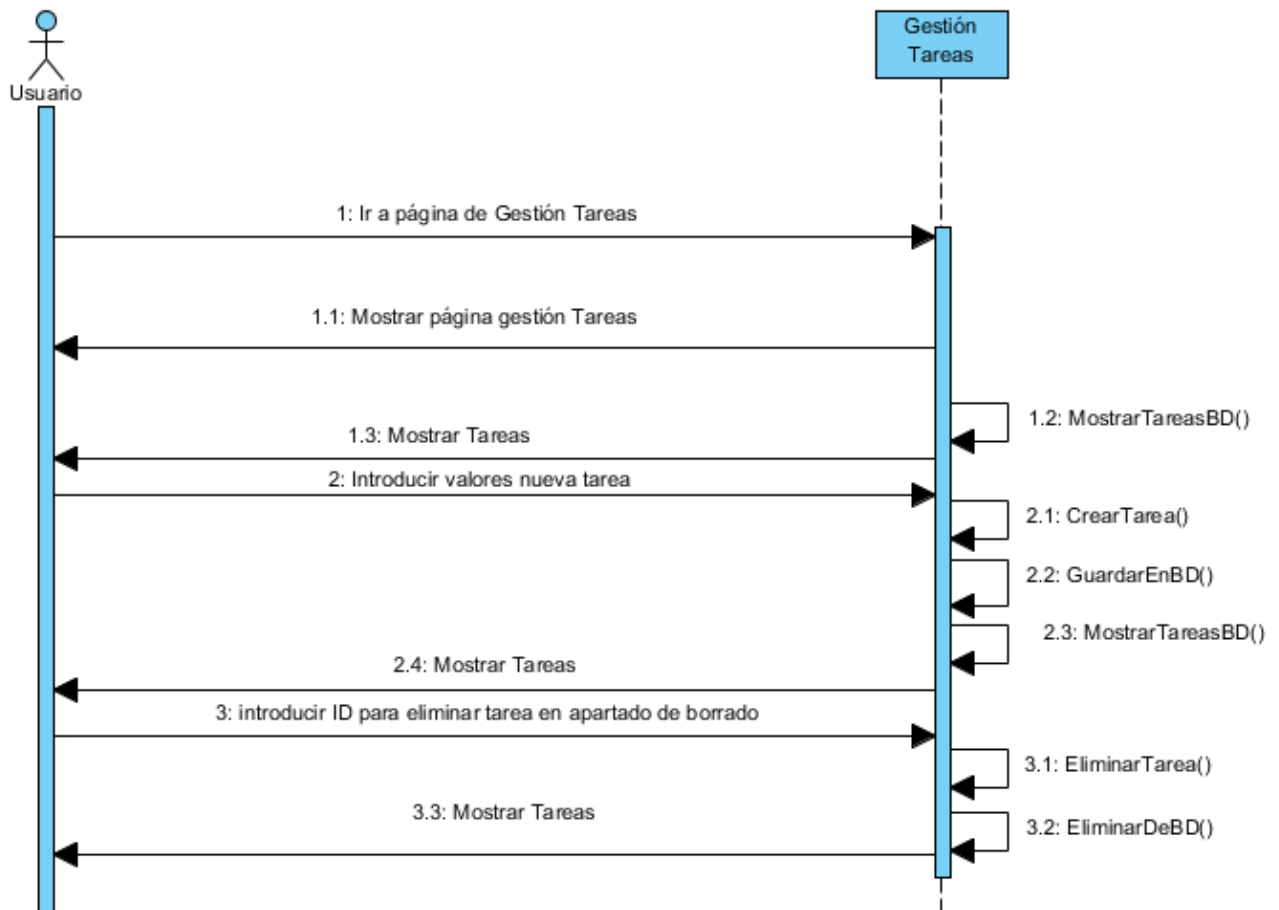


ILUSTRACIÓN 22. DIAGRAMA DE SECUENCIA DE GESTIÓN DE TAREAS DEL HOGAR

4.5.5.- Consultar Gasto Semanal

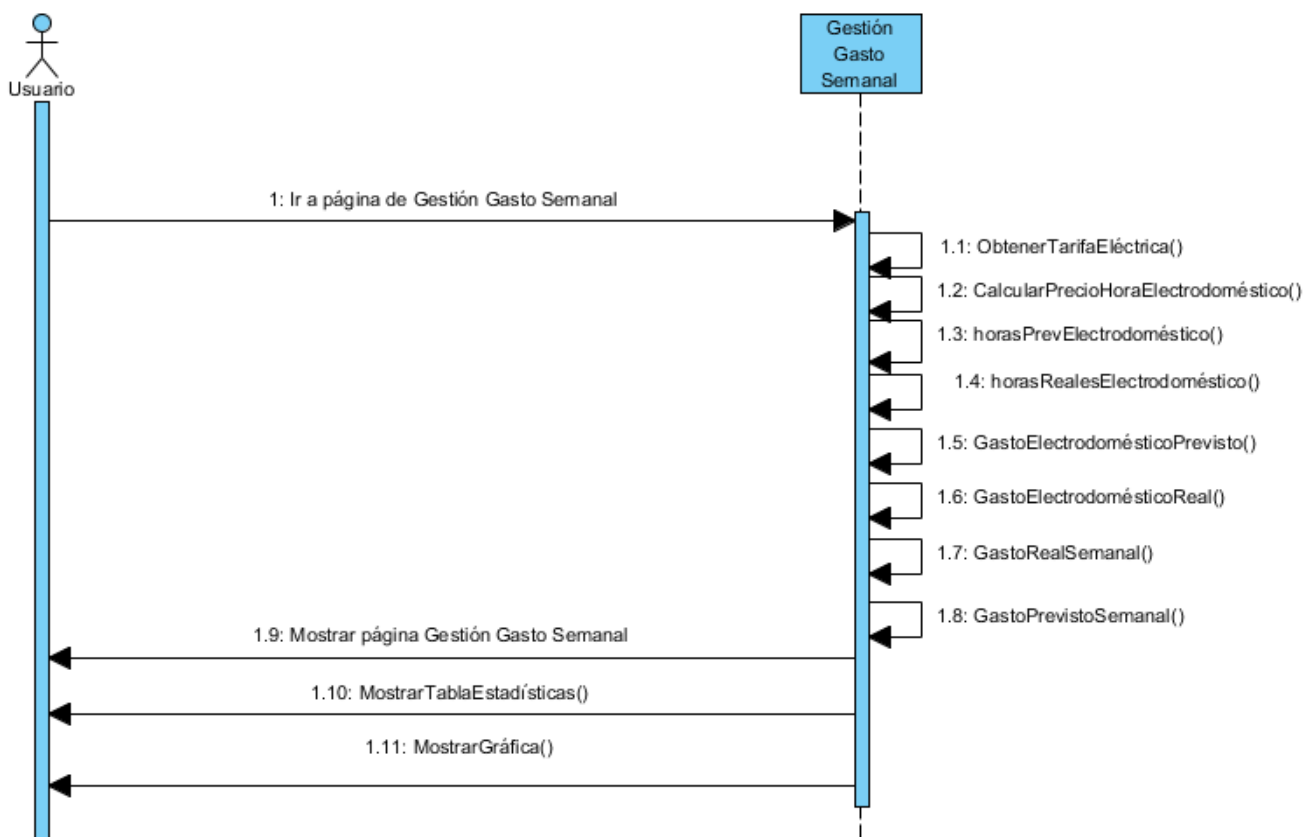


ILUSTRACIÓN 23. DIAGRAMA DE SECUENCIA DE GESTIÓN DEL GASTO SEMANAL

4.6.- Diagrama de Clases

En este modelo, se muestra la arquitectura del sistema mostrando la información con la que se trabaja en el mismo. Su elemento principal son las clases, que se asimilan bastante a las clases java de una aplicación, en tanto que representan un molde para crear objetos de información. Entre las clases existen relaciones, con multiplicidades determinadas.

Antes de mostrar el diagrama, explicaremos en detalle cuál es la función de cada clase y método:

- **Clase Usuario:** Se trata del objeto de usuario que utiliza la aplicación, con todos los datos necesarios del mismo.
 - eliminarUsuario(): Método encargado de eliminar un usuario del sistema.
- **Inicio Sesión:** Actividad de la aplicación encargada de iniciar sesión en la aplicación.
 - ComprobarCredenciales(): Prueba que los datos introducidos para iniciar sesión por parte del usuario son correctos.
 - IniciarSesión(): Inicia una nueva sesión para el usuario que lo ha solicitado.
 - irRegistro(): Redirige hacia la actividad de Registro de la aplicación.
- **Registro:** Actividad para que un usuario se pueda registrar en la aplicación.
 - CrearUsuario(): Crea un nuevo usuario.
 - ComprobarDatos(): Comprueba que los datos introducidos para registrarse son correctos.
 - GuardarEnDB(): Almacena en BD el nuevo usuario.
- **Opciones de Cuenta:** Esta actividad permite a los usuarios cambiar la configuración de su cuenta, como su correo o contraseña. Esta actividad, sólo puede generarse en caso de haber iniciado una sesión.
 - ComprobarDatos(): Comprueba que los nuevos datos sean correctos.
 - ModificarUsuarioBD(): Se encarga de modificar los datos del usuario sustituyéndolos por los nuevos introducidos.
- **Gestión Electrodomésticos:** Actividad encargada de gestionar los electrodomésticos del usuario.
 - MostrarElectsBD(): Muestra los electrodomésticos que hay guardados en la Base de Datos.
 - CrearElect(): Crea un nuevo electrodoméstico.
 - GuardarEnBD(): Almacena en Base de Datos un nuevo electrodoméstico.
 - EliminarElect(): Borra el objeto de un electrodoméstico.
 - EliminarDeBD(): Borra de la Base de Datos un electrodoméstico.

- **Gestión Tareas:** Actividad encargada de gestionar las tareas del hogar del usuario.
 - MostrarTareasBD(): Muestra las tareas del hogar que hay guardadas en la Base de Datos.
 - CrearTarea(): Crea una nueva tarea del hogar.
 - GuardarEnBD(): Almacena en Base de Datos una nueva tarea.
 - EliminarTarea(): Borra el objeto de una tarea.
 - EliminarDeBD(): Borra de la Base de Datos una tarea.
- **Agenda:** Esta actividad es la encargada de mostrar la agenda con el calendario. Se trata de la página principal de la aplicación una vez el usuario ha iniciado sesión.
 - MostrarCalendario(): Método encargado de mostrar al usuario el calendario donde podrá introducir sus tareas.
 - MostrarTareasSemana(): Muestra las tareas del día programadas para la semana actual.
 - MostrarDia(): Método encargado de, una vez el usuario hace click en un día del calendario, que sea redirigido a la página de gestión de tareas del día en cuestión.
- **Gestión Tareas del día:** Actividad encargada de gestionar las tareas del día del usuario. Esta actividad solo se muestra desde la Agenda, ya que para llegar a la gestión de las tareas del día hay que seleccionar un día del calendario que hay en la Agenda para poder ver así las tareas que hay planificadas para el día seleccionado.
 - MostrarTareasDiaBD(): Muestra las tareas del día que hay guardadas en la Base de Datos.
 - CrearTareaDia(): Crea una nueva tarea del día.
 - GuardarEnBD(): Almacena en Base de Datos una nueva tarea del día.
 - EliminarTareaDia(): Borra el objeto de una tarea del día.
 - EliminarDeBD(): Borra de la Base de Datos una tarea del día.
- **Electrodoméstico:** Clase objeto de electrodomésticos de usuario.
- **Tarea:** Clase objeto de tareas del hogar.
- **TareaDia:** Clase objeto de tareas del día.
- **Gestión Gasto Semanal:** Esta actividad es la encargada de mostrar la página de Gasto Semanal de la aplicación.
 - obtenerTarifaEléctrica(): Se encarga de obtener el precio de la tarifa eléctrica solicitándolo a la API de E.sios.
 - calcularPrecioHoraElectrodoméstico(): Calcula el precio por hora de uso de los electrodomésticos almacenados en el sistema en euros.
 - horasPrevElectrodoméstico(): Calcula el número de horas previstas de uso para la semana actual de todos los electrodomésticos almacenados.
 - horasRealesElectrodoméstico(): Calcula el número de horas reales de uso para la semana actual de todos los electrodomésticos almacenados.
 - gastoElectrodomésticoPrev(): Se encarga de calcular el gasto previsto de todos los electrodomésticos.
 - gastoElectrodomésticoReal(): Se encarga de calcular el gasto real de todos los electrodomésticos.

- gastoRealSemanal(): Se encarga de calcular el gasto total real de la semana actual.
- gastoPervistoSemanal(): Se encarga de calcular el gasto total previsto de la semana actual.
- MostrarTablaEstadísticas(): Muestra la tabla con todas las estadísticas de gastos en la página de Gasto Semanal.
- MostrarGráfica(): Muestra la gráfica comparativa entre el gasto previsto y real de la semana actual.

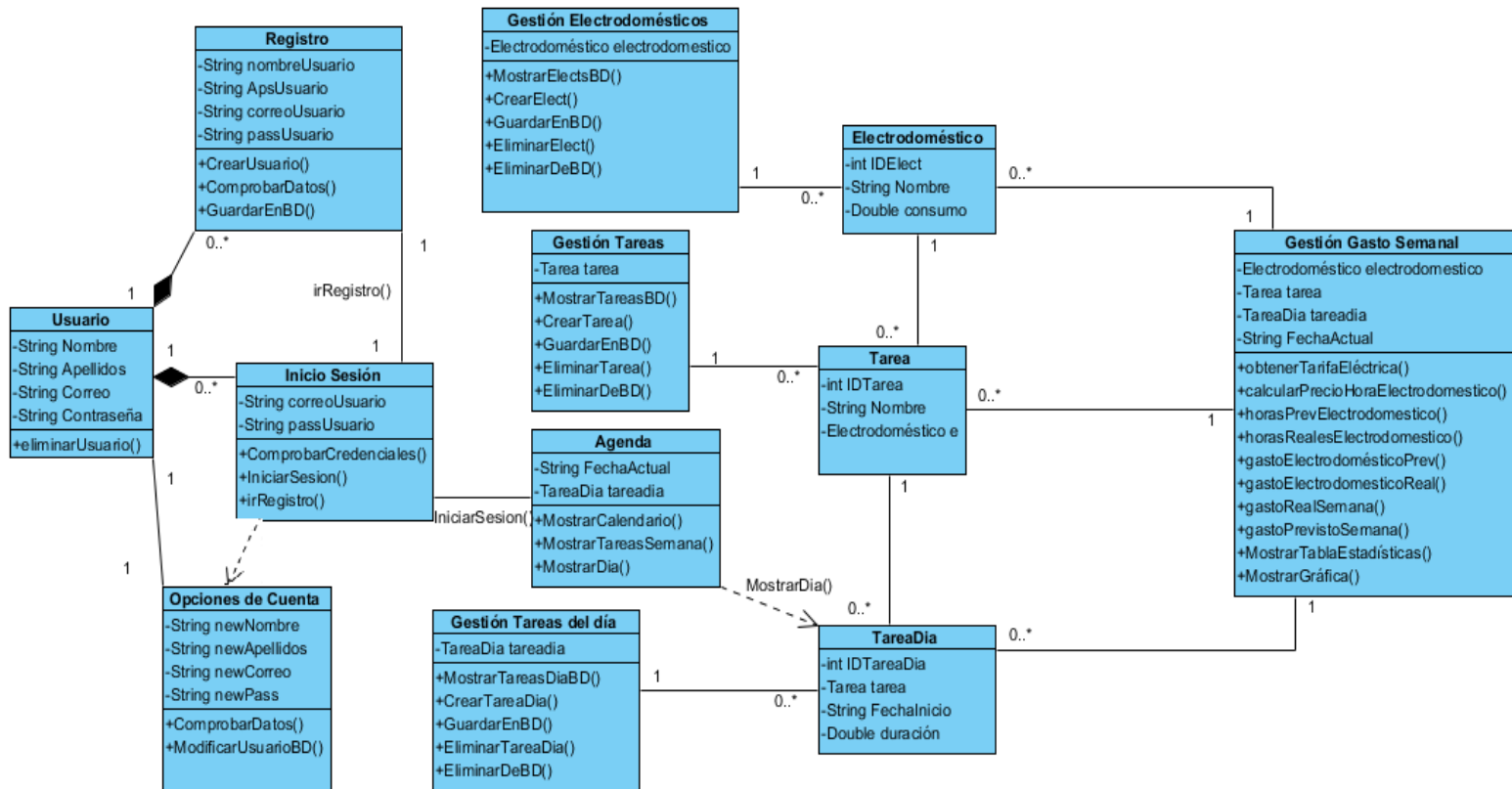


ILUSTRACIÓN 24. DIAGRAMA DE CLASES

4.7.- Diagrama de Componentes

En el siguiente modelo se explica la arquitectura del sistema desde la perspectiva de los componentes que integran la aplicación y cómo se comunican entre ellos. Cada componente se encarga de realizar una serie de funcionalidades del sistema, y lo que permite que se compartan información son las interfaces. Estas interfaces permiten el flujo de información entre componentes y permiten asimismo que el sistema funcione en su conjunto, permitiendo que todas las funcionalidades se cumplan.

Este diagrama, por tanto, busca explicar cómo interactúan los diferentes componentes entre sí.

Para que los componentes se comuniquen, se requiere de interfaces. Un ejemplo de interfaz se ve en la siguiente figura:



ILUSTRACIÓN 25. EJEMPLO INTERFAZ DIAGRAMA DE COMPONENTES

Un componente proporciona la interfaz, que en el ejemplo es el de la izquierda, y el otro es el que la utiliza, es decir, el de la derecha de la imagen anterior. La interfaz proporciona los métodos y procedimientos del componente que lo proporciona.

El diagrama de componentes se muestra a continuación:

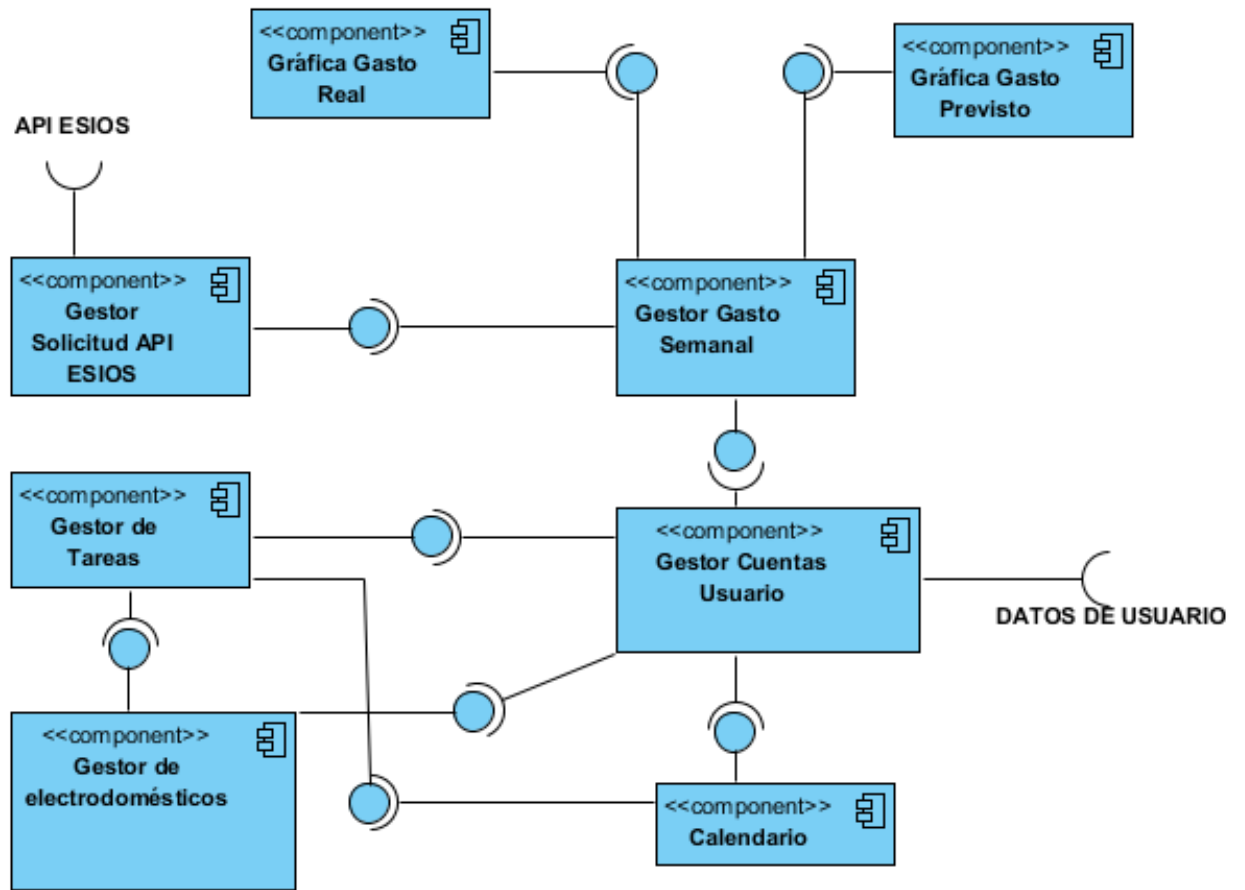


ILUSTRACIÓN 26. DIAGRAMA DE COMPONENTES

5.- Descripción de las funcionalidades de la aplicación

En este capítulo se procede a desarrollar los aspectos técnicos de la solución obtenida. Para ello, se comenzará realizando una introducción de la aplicación desarrollada, describiendo de forma general las principales características de ésta. Después, se irán describiendo las funcionalidades contenidas en la aplicación una a una y con mayor detalle.

5.1.- Introducción de las funcionalidades de la aplicación

La aplicación, llamada **Voltaic**, es una aplicación Android que contiene una serie de funcionalidades. En su conjunto, forman un asistente personal para planificar las tareas cotidianas del hogar con el objetivo de ahorrar en la factura de la luz.

Las funcionalidades que contiene son:

1. **Planificación de las tareas del hogar:** Contiene un calendario que se utiliza como agenda para organizar las tareas y programarlas el día y hora deseados.
2. **Cálculo del gasto semanal:** Se encarga de calcular el gasto en euros de todas las tareas programadas para la semana actual.
3. **Obtención de las tarifas eléctricas:** Funcionalidad encargada de solicitar los precios de las tarifas eléctricas, para poder calcular el gasto semanal en euros que conlleva la realización de las tareas programadas de la semana.
4. **Inicio de sesión y registro:** Para poder acceder, primero, es necesario estar registrado, y si lo estás, debes iniciar sesión introduciendo tus datos para acceder a tu cuenta.
5. **Opciones. Cambiar configuración de cuenta:** Se encarga de modificar información de tu cuenta, como tu contraseña o correo electrónico.

5.2.- Planificación de las tareas del hogar

Esta funcionalidad es la encargada de permitir al usuario programar las tareas del hogar en el calendario. Funciona como una agenda: en el calendario, pueden seleccionar cualquier día del mes, y en el día que seleccionen, te saldrá un listado con todas las tareas que hay

programadas. También, para poder añadir más tareas ese día, hay una serie de campos que hay que rellenar para añadir la tarea para ese día. Por último, hay un apartado que permite al usuario eliminar tareas introduciendo el ID de la tarea de ese día que quiera eliminar.

En la imagen de la derecha, se puede observar la ventana de la aplicación de la agenda. Debajo del calendario se puede comprobar que hay un apartado que informa de las tareas de la semana, en este caso hay dos tareas programadas.

Como se ha mencionado, cada día del calendario se puede seleccionar para planificar las tareas de ese día, lo que nos llevaría a una ventana diferente, exclusiva de cada día, que se puede ver en la siguiente imagen.

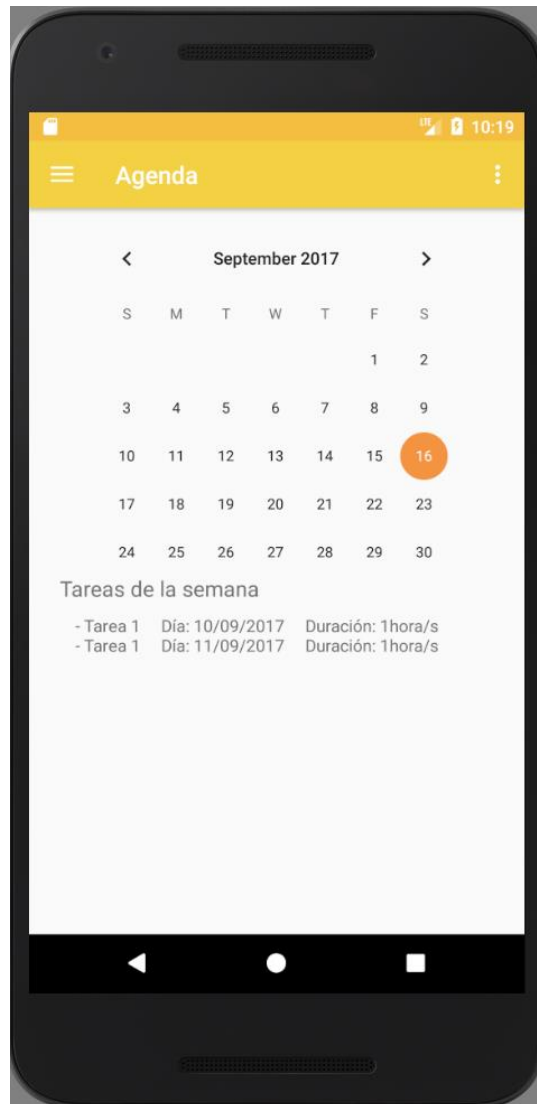


ILUSTRACIÓN 27. PÁGINA DE AGENDA

Tras hacer click en un día del calendario, se nos muestra la ventana que tenemos en la figura de la derecha. En este caso, estamos a día 11 de septiembre de 2017, y tenemos una tarea programada.

Para introducir una nueva tarea, deberemos rellenar los campos que hay en la parte superior y pulsar “Añadir”.

Si se desea eliminar una tarea de ese día, iremos al apartado inferior, “Eliminar tarea por ID” e introduciremos el número identificador de la tarea que queramos eliminar de las listadas en la tabla de la parte media de la ventana, donde se sitúa la lista de tareas programadas.



ILUSTRACIÓN 28. PÁGINA DE TAREAS DEL DÍA

Pero para poder introducir una tarea en la agenda, previamente se deben de haber creado los electrodomésticos que se utilizan y la tarea del hogar en cuestión.

Para introducir un electrodoméstico, tenemos la ventana de que se encarga de ello, que se muestra más abajo.

Como se puede observar a la derecha, para crear un electrodoméstico con el que realizar las tareas del hogar deberemos introducir un nombre y un consumo eléctrico expresado en kilovatios por hora (kWh) y pulsar el botón “Añadir”.

Debajo, podemos ver la tabla de electrodomésticos, ordenados por el identificador (ID).

Por último, se muestra la funcionalidad de borrado de electrodomésticos. Para eliminar un electrodoméstico, basta con introducir el ID del electrodoméstico y pulsar “Eliminar”.



ILUSTRACIÓN 29. PÁGINA DE ELECTRODOMÉSTICOS

Después, para crear una tarea, iremos a la página “tareas del Hogar”, que se muestra en la siguiente figura.

Para crear las tareas que después planificaremos en nuestra agenda, debemos ir a la ventana de Tareas del Hogar.

Para crear una nueva tarea, deberemos introducir un nombre de la tarea y el ID del electrodoméstico que se utiliza en la misma. Por último, pulsar “Añadir”.

Debajo, veremos la tabla, como en el resto de ventanas, para consultar las tareas que tenemos en base de datos.

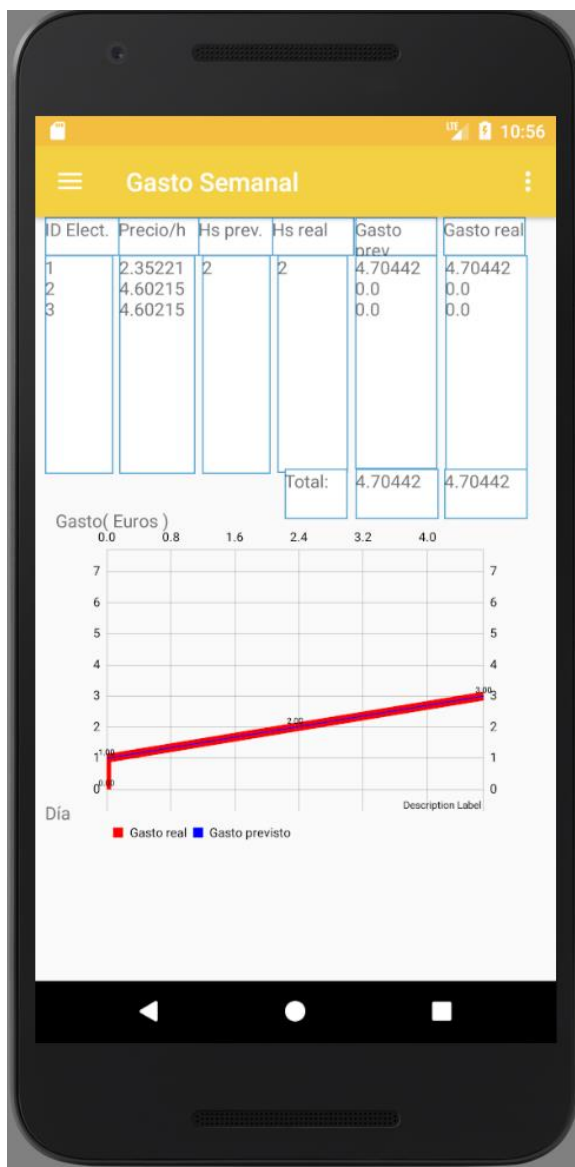
Para eliminar, seguimos el mismo procedimiento introduciendo el ID de la tarea y pulsando “Eliminar”.



ILUSTRACIÓN 30. PÁGINA DE TAREAS DEL HOGAR

5.3.- Cálculo del gasto semanal

A continuación, se describe la funcionalidad que se encarga de realizar los cálculos pertinentes con la información proporcionada por el usuario sobre la planificación de sus tareas y gracias también a los datos proporcionados de las tarifas eléctricas por ESIOS.



A la izquierda, se puede ver la ventana donde se consultan los gastos de la semana resultantes de la planificación.

La tabla que hay en la parte superior disgrega el gasto en función de los electrodomésticos, para acabar calculando el gasto real total y el gasto previsto total.

Cada electrodoméstico está identificado por su ID, y se calcula su consumo por horas.

Después, gracias a una consulta a base de datos se obtienen las horas previstas y las horas reales que se han utilizado dichos electrodomésticos. **Cabe señalar que las horas previstas son las horas que se han planificado, pero que aún no se han ejecutado, por no haber llegado la hora de la tarea aún. Respecto a las horas reales, se tratan de las horas que efectivamente se han ejecutado y que por tanto están por detrás en el tiempo de la hora actual.**

Debajo, se muestra la gráfica, que obtiene el gasto total y previsto de cada día de la semana y los muestra. En color azul, se muestra la gráfica del gasto previsto de la semana, y en rojo, el gasto real.

ILUSTRACIÓN 31. PÁGINA DE GESTIÓN GASTO SEMANAL

5.4.- Obtención de las tarifas eléctricas

Para implementar esta funcionalidad, tuve que investigar en la documentación de la Api de Esios, ya que tenía que encontrar el indicador que, al enviar la solicitud por URL, me permitiera obtener el PVPC, el cual es el 1013.

Por tanto, para solicitar a Esios el precio del PVPC, tenía que enviar una solicitud vía URL pasándole los parámetros adecuados. Estos eran el intervalo de fechas del cual necesitaba saber el precio de las tarifas PVPC, que vendrían en una lista.

La solicitud cliente a la API de Esios es la siguiente:

```
try {
    URL url = new URL("https://api.esios.ree.es/indicators/1013?start_date=2017-09-02T09:30:00&end_date=2017-09-02T10:00:00");

    HttpURLConnection urlConnection = (HttpURLConnection) url
        .openConnection();

    urlConnection.setRequestMethod("GET");
    urlConnection.setRequestProperty("Accept", "application/json; application/vnd.esios-api-v1+json");
    urlConnection.setRequestProperty("Content-Type", "application/json");
    urlConnection.setRequestProperty("Host", "api.esios.ree.es");
    urlConnection.setRequestProperty("Authorization", "Token token=\"
    urlConnection.setRequestProperty("Cookie", " ");

    InputStream in = urlConnection.getInputStream();

    InputStreamReader isw = new InputStreamReader(in);
}
```

ILUSTRACIÓN 32. IMPLEMENTACIÓN DE LA SOLICITUD CLIENTE A LA API E.SIOS

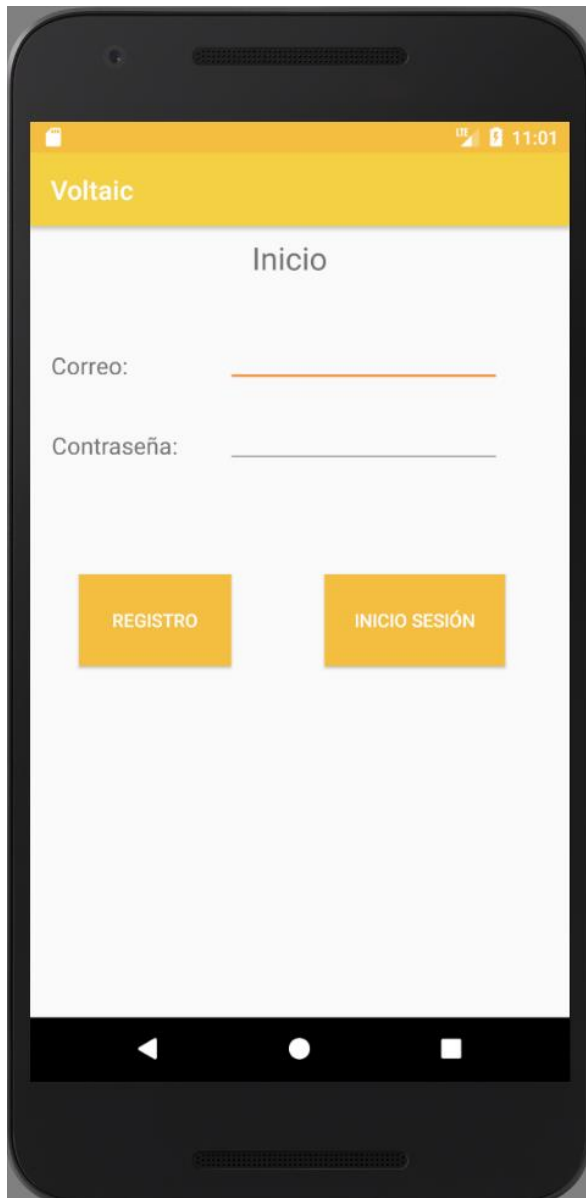
Como se puede ver en la URL, esta consulta está pidiendo las tarifas PVPC para el intervalo de tiempo entre el **2017-09-02** a las 9:30 y el **2017-09-02** a las 10:00.

Lo demás, son los parámetros de la consulta, como el método GET, **o el token de autorización, que es personal e intransferible, por ese motivo se ha borrado de la captura.**

La respuesta que envía Esios a esta consulta es un JSON, con los datos solicitados. Para obtenerlos, solo hay que extraer el texto del JSON y mediante las librerías de json.org deserializar el json, y así obtener la información que contiene, entre ella el precio de la tarifa de ese intervalo de horas, que como indica en el ejemplo consta entre las 9:30 y las 10:00.

5.5.- Inicio de Sesión y registro

ILUSTRACIÓN 33. PÁGINA DE INICIO



La función de inicio de sesión se encarga de permitir al usuario entrar en la aplicación con sus credenciales. También impide que alguien que no esté registrado pueda acceder a la aplicación.

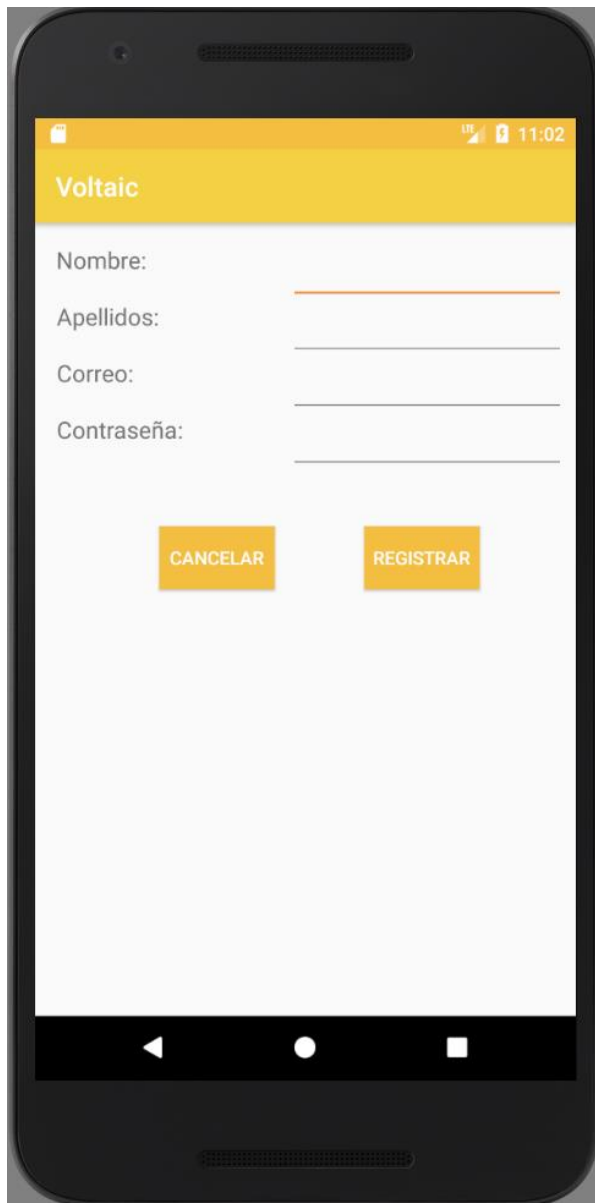
Es la primera ventana que se muestra al entrar en la aplicación, y pasar por ella es obligatorio, además de identificarse correctamente, para poder acceder a nuestra cuenta de usuario.

Pulsando el botón “Inicio sesión”, y habiendo introducido nuestras credenciales correctamente, entramos en nuestra cuenta.

Si nuestras credenciales son incorrectas, ya sea el correo, la contraseña o ambas, saltará un mensaje indicando el error. En caso contrario, se nos abrirá la ventana de Agenda de nuestra cuenta.

Sin embargo, si no tenemos cuenta todavía, podemos registrarnos. Para ello, pulsamos

el botón “Registro”.



Respecto a la funcionalidad de registro, que puede verse a la izquierda, es necesaria para poder crear una cuenta en caso de no tenerla aún.

Para registrarse, hay que introducir un Nombre y apellidos del usuario, su correo y una contraseña. Por último, pulsar botón registrar.

Para volver a la página de Inicio, basta con pulsar el botón “Cancelar”.

ILUSTRACIÓN 34. PÁGINA DE REGISTRO

5.6.- Opciones: Cambiar configuración de cuenta.

Para modificar la configuración de la cuenta, como el correo del usuario o la contraseña, deberemos ir a la ventana de Opciones, donde podremos introducir los nuevos datos y pulsar “Aplicar” para realizar los cambios. Si cambiamos de opinión, podemos pulsar cancelar.

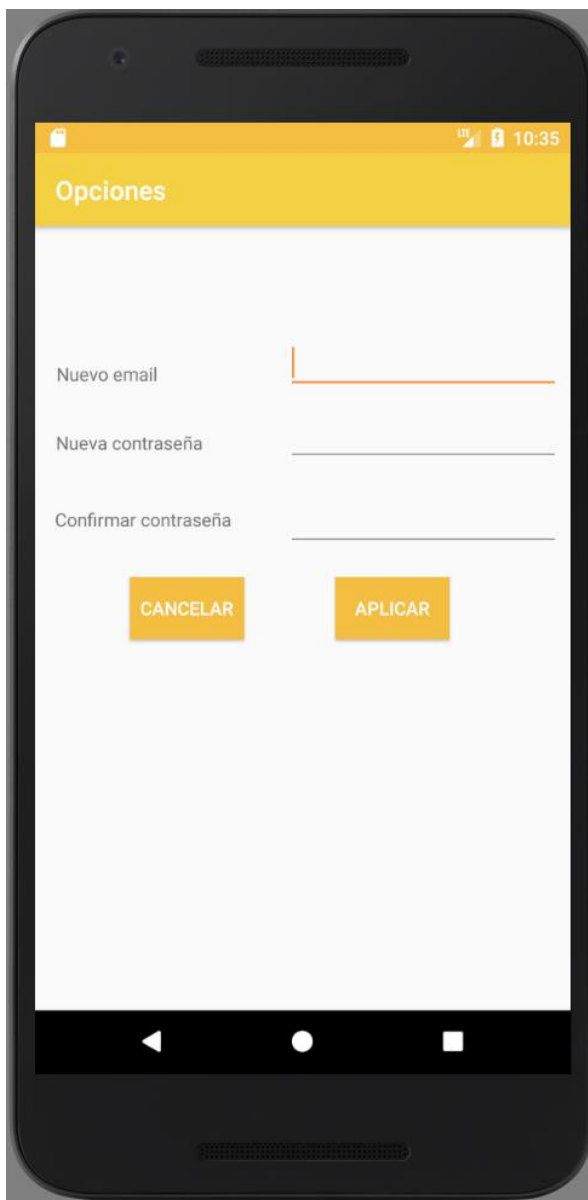


ILUSTRACIÓN 35. PÁGINA DE OPCIONES

5.7.- Navegación entre páginas de la aplicación.

Para proporcionar una navegación cómoda, sencilla e intuitiva para el usuario se optó por implementar menús que estuvieran disponibles en todo momento. Con ello, el usuario será capaz de ubicarse mejor en la aplicación y no se perderá en la navegación.

Se decidió separar el menú que organizaba las funcionalidades de las Opciones y de Salir de la app porque así, el usuario puede entender mejor dónde está su planificador, y donde están las opciones de su cuenta, de forma diferenciada. También evita errores al pulsar en una opción no elegida, como Salir queriendo ir a Tareas del Hogar, y con ello evita frustraciones al usuario.

Se ha buscado por tanto una navegación fluida y elegante, sin giros complicados de la interfaz, para que sea lo más cómoda posible.

6.- Plan de Pruebas del sistema

El plan de pruebas del sistema consiste en la fase del proyecto en la que se comprueba que la aplicación desarrollada supera las pruebas sobre todas las funcionalidades, de una forma correcta, óptima, eficiente y de acuerdo con los Casos de Uso y los Requerimientos del sistema.

Este Plan de Pruebas del Sistema se corresponde con el trabajo y las tareas realizadas durante la Fase de planificación del proyecto, Fase de Pruebas.

Cada Caso de Prueba se corresponde con la comprobación de una funcionalidad del sistema diferente, pudiendo haber casos en los que los pasos a seguir para comprobar que se lleva a cabo correctamente dos funcionalidades distintas sean los mismos, al ser funcionalidades complementarias o con secuencias de acción por parte del usuario similares.

6.1.- Entorno de pruebas

Para realizar las pruebas del sistema, se ha utilizado un emulador de móviles Android. Dicho emulador, está integrado en el entorno de desarrollo del proyecto, Android Studio.

Se trata de **Android Emulator**, y es capaz de simular la aplicación Android que se está desarrollando, simplemente, al ejecutar la aplicación, hay que seleccionar un tipo de dispositivo en el que será la simulación, el cual tendrá una serie de características como el tamaño de pantalla, y se mostrará en el dispositivo que hayamos seleccionado como si fuera un teléfono móvil en nuestro ordenador.

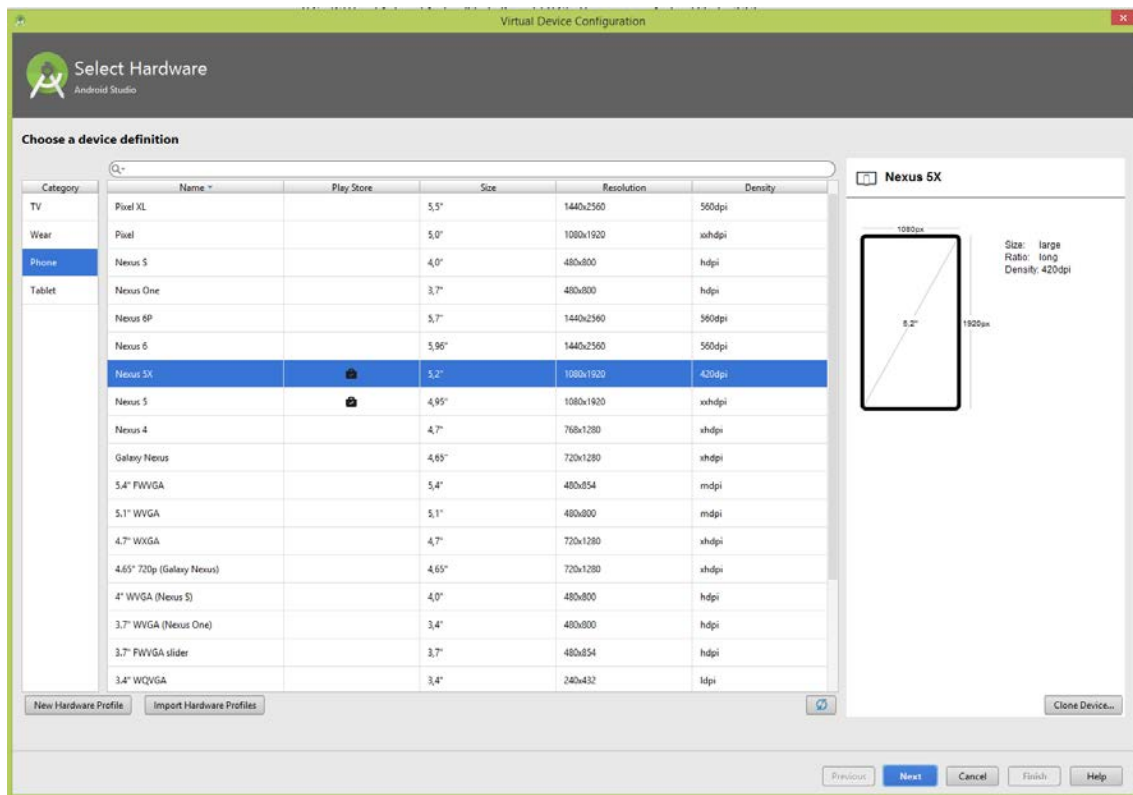


ILUSTRACIÓN 36. CAPTURA PRIMERA DE EMULADOR ANDROID STUDIO

Esta simulación es exactamente igual en términos de funcionalidad y usabilidad a la que se realizaría en un teléfono móvil de verdad, con la salvedad de que, al ser en un ordenador, la forma de interactuar con la pantalla del móvil no es táctil y tendremos que utilizar nuestro ratón para tocar la pantalla del emulador.

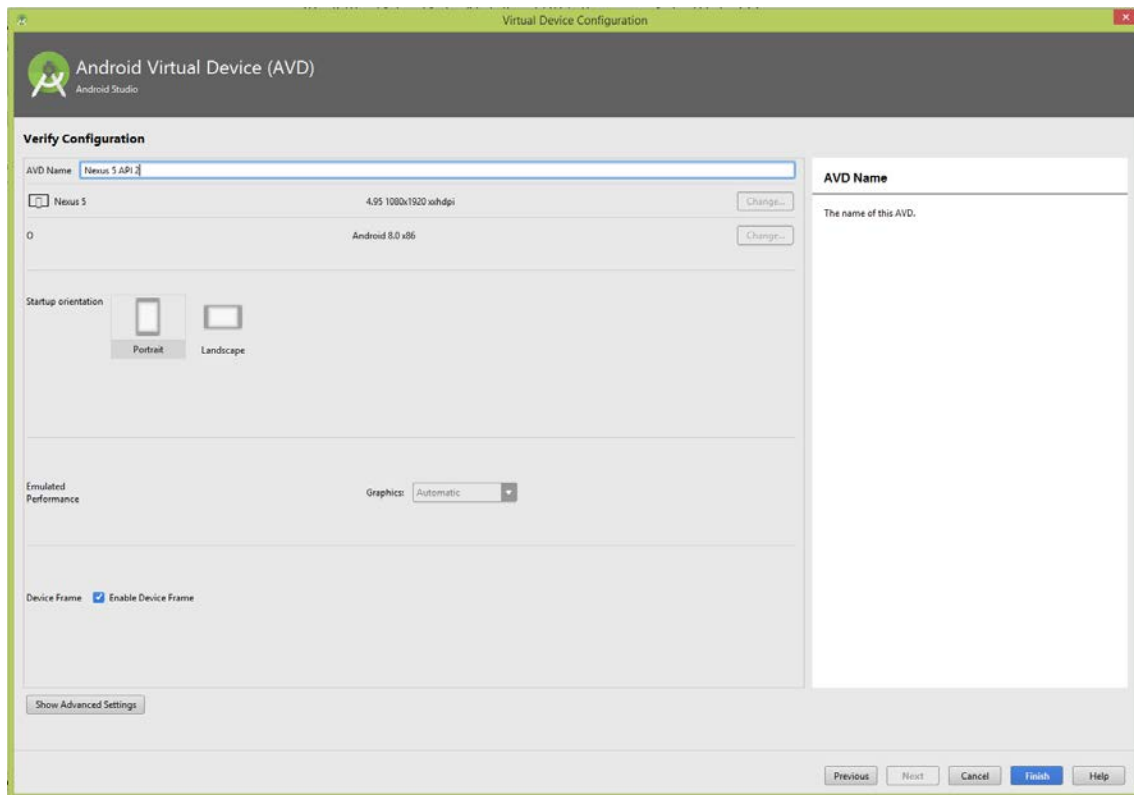


ILUSTRACIÓN 37. CAPTURA SEGUNDA EMULADOR ANDROID STUDIO

Este emulador ha sido probado previamente a la realización de las pruebas del sistema para comprobar que efectivamente, se trata de un entorno de pruebas óptimo y similar respecto a un entorno de pruebas como un teléfono móvil real respecto a rendimiento y usabilidad. Por tanto, todas las pruebas que se han realizado en este entorno y sus resultados son veraces y componen unos resultados que prueban correctamente la realización de las funcionalidades del sistema.

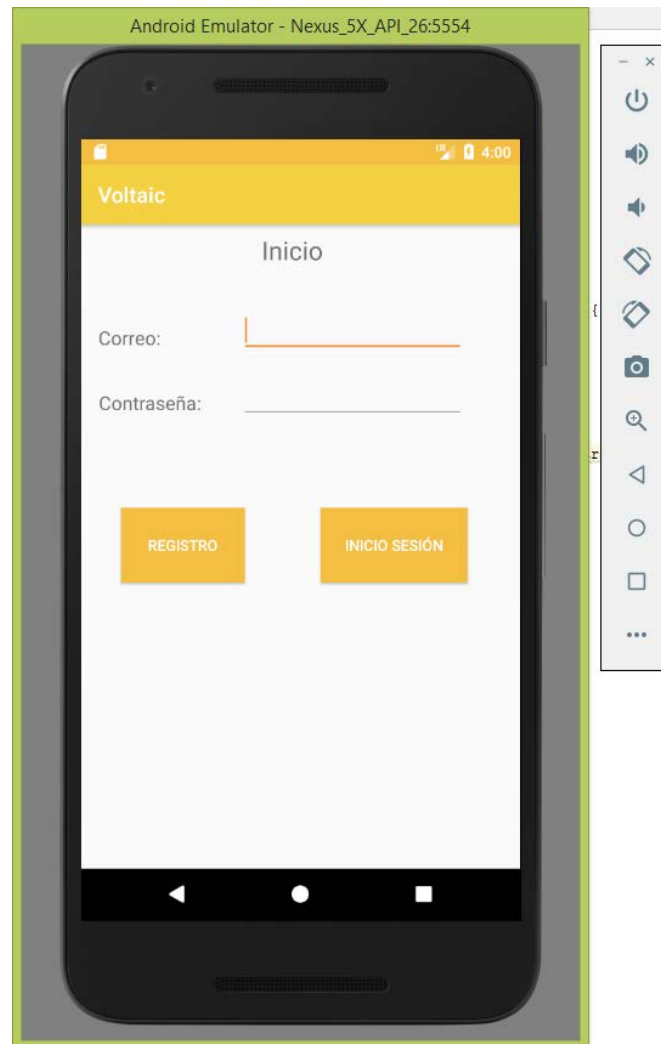


ILUSTRACIÓN 18. CAPTURA TERCERA EMULADOR ANDROID STUDIO

6.2.- Formato de los Casos de Prueba

Ahora procederemos a describir el formato de los Casos de Prueba correspondientes al Plan de Pruebas del sistema. Para ello, nombraremos los distintos campos de las tablas de los casos de prueba y los explicaremos en detalle. Todos los campos son obligatorios.

- **Identificador de prueba:** Código que identifica unívocamente al caso de prueba, compuesto siempre por las siglas CP (Caso de Prueba) seguido por un número de dos dígitos.
- **Requisitos Funcionales:** Se refiere a los requerimientos funcionales del sistema relacionados con este caso de prueba y con la funcionalidad que comprueba.
- **Descripción de prueba:** Breve descripción del caso de prueba, indicando qué va a comprobar del sistema.
- **Entradas:** Se trata de lo que el comprobador del caso de prueba introduce en el sistema, ya sea una acción, interacción con el sistema, información o datos.
- **Salidas:** Lo que produce/proporciona al usuario el sistema después de introducir la entrada correspondiente. Toda salida tiene una entrada que la convoca.

6.3.- Casos de Prueba

A continuación, se redactan los casos de prueba del sistema en el formato anteriormente explicado.

Identificador Prueba	CP01
Requisitos Funcionales	RF02
Descripción prueba	El sistema registra con éxito a un usuario.
Entradas	Salidas
1. Click en botón de registrar	1. Aparece la pantalla de registro de usuario.
2. Introducir nombre de usuario, Apellidos de usuario, Correo de usuario y contraseña de cuenta y hacer click en botón registrarse	2. Mensaje: "Se ha registrado correctamente". 3. Aparece la pantalla de inicio de sesión.

Identificador Prueba	CP02
Requisitos Funcionales	RF01, RF02
Descripción prueba	El sistema no permite iniciar sesión a usuario no registrado.
Entradas	Salidas
1. Introducir correo no registrado y contraseña errónea.	1. Pantalla de inicio.
2. Pulsar botón Iniciar sesión.	2. Pantalla de inicio. 3. Mensaje de error: "Email y/o contraseña incorrectos".

Identificador Prueba	CP03
Requisitos Funcionales	RF01, RF02
Descripción prueba	Inicio de sesión con usuario ya registrado.
Entradas	Salidas
1. Introducir correo registrado y contraseña correcta.	1. Pantalla de inicio.
2. Pulsar botón Iniciar sesión.	2. Pantalla de Agenda. 3. Mensaje: "Inicio de sesión correcto".

Identificador Prueba	CP04
Requisitos Funcionales	RF10, RF12
Descripción prueba	El sistema crea un electrodoméstico correctamente.
Entradas	Salidas
1. Introducir Nombre de electrodoméstico y consumo por horas (en kW/h).	1. Pantalla de gestión de electrodomésticos.
2. Pulsar botón Añadir.	2. Pantalla de gestión de electrodomésticos con el electrodoméstico introducido mostrado en la tabla de electrodomésticos existentes.

Identificador Prueba	CP05
Requisitos Funcionales	RF14, RF16
Descripción prueba	El sistema crea una tarea correctamente.
Entradas	Salidas
1. Introducir Nombre de tarea e ID del electrodoméstico asociado a esa tarea que esté registrado previamente.	1. Pantalla de gestión de Tareas del Hogar.
2 Pulsar botón Añadir.	2 Pantalla de gestión de Tareas del Hogar con la tarea introducida mostrado en la tabla de tareas del hogar existentes.

Identificador Prueba	CP06
Requisitos Funcionales	RF18, RF19
Descripción prueba	Añadir una tarea en un día de la semana.
Entradas	Salidas
1. Introducir ID de tarea existente, fecha, y duración de la tarea	1- Pantalla del día de la semana.
2. Pulsar botón Añadir.	2. Pantalla del día de la semana con la tarea introducida añadida en la tabla de tareas pendientes para ese día.

Identificador Prueba	CP07
Requisitos Funcionales	RF23, RF24, RF25, RF26, RF27, RF28
Descripción prueba	Obtener tarifa de la API de ESIOS.
Entradas	Salidas
1. Abrir ventana de consulta de Gastos de la semana.	1. Ventana de Gasto Semanal, con todos los gastos de la semana actualizados con las tarifas correspondientes.

Identificador Prueba	CP08
Requisitos Funcionales	RF23, RF24, RF25, RF26, RF27
Descripción prueba	Generar gráfica de Gasto Semanal.
Entradas	Salidas
1. Abrir ventana de Gasto Semanal.	1. Ventana de Gasto Semanal, con todos los gastos de la semana. 2. Debajo de la tabla con los datos, se genera la gráfica comparativa entre consumo previsto y real.

Identificador Prueba	CP09
Requisitos Funcionales	RF04, RF06
Descripción prueba	Cambiar opciones de cuenta.
Entradas	Salidas
1. Abrir ventana de Opciones	1. Ventana de Opciones.
2. Introducir nuevo correo y contraseña.	2. Ventana de Opciones.
3. Pulsar botón Aplicar.	3. Ventana de Agenda.
4. Salir de la aplicación pulsando botón Salir.	4. Ventana de Inicio.
5. Introducir nuevo correo y contraseña.	5. Mensaje: "Inicio de sesión correcto". 6. Ventana de Agenda.

Identificador Prueba	CP10
Requisitos Funcionales	RF04, RF08
Descripción prueba	Cerrar sesión.
Entradas	Salidas
1. Pulsar botón Salir.	1. Ventana de Inicio.

Identificador Prueba	CP11
Requisitos Funcionales	RF28, RF29, RF30
Descripción prueba	Mostrar información en la tabla de gastos.
Entradas	Salidas
1. Abrir ventana de Gasto Semanal.	1. Ventana de Gasto Semanal. 2. Se muestra la tabla con toda la información referente al consumo del hogar actualizada.

Identificador Prueba	CP12
Requisitos Funcionales	RF12, RF13
Descripción prueba	Eliminar un electrodoméstico guardado.
Entradas	Salidas
<ol style="list-style-type: none"> 1. Abrir ventana de gestión de electrodomésticos. 	<ol style="list-style-type: none"> 1. Ventana de gestión de electrodomésticos. 2. Se muestra la tabla con todos los electrodomésticos guardados con sus IDs correspondientes.
<ol style="list-style-type: none"> 2. Dentro de la misma ventana, en apartado de "Eliminar electrodoméstico por ID" introducir ID del electrodoméstico que se desea eliminar. 	<ol style="list-style-type: none"> 3. El electrodoméstico seleccionado para eliminar ya no se muestra en la tabla de electrodomésticos guardados.

Identificador Prueba	CP13
Requisitos Funcionales	RF16, RF17
Descripción prueba	Eliminar una tarea del hogar guardada.
Entradas	Salidas
1. Abrir ventana de gestión de tareas del hogar.	1. Ventana de gestión de tareas del hogar. 2. Se muestra la tabla con todas las tareas del hogar guardadas con sus IDs correspondientes.
2. Dentro de la misma ventana, en apartado de "Eliminar tarea por ID" introducir ID de la tarea que se desea eliminar.	3. La tarea seleccionada para eliminar ya no se muestra en la tabla de tareas guardadas.

Identificador Prueba	CP14
Requisitos Funcionales	RF19, RF20
Descripción prueba	Eliminar una tarea del día guardada.
Entradas	Salidas
1. Abrir ventana de Agenda.	1. Ventana de Agenda.
2. Hacer click en un día del calendario donde esté la tarea que deseamos eliminar.	2. Se muestra la ventana de tareas del día seleccionado, con un listado de todas las tareas planificadas para ese día.
3. Dentro de la ventana de tareas del día, en apartado de "Eliminar tarea por ID" introducir ID de la tarea del día que se desea eliminar.	3. La tarea del día seleccionada para eliminar ya no se muestra en la tabla de tareas del día guardadas.

6.4.- Resultados obtenidos y pruebas completadas con éxito.

Todos los casos de prueba del Plan de Pruebas se cumplen con éxito, realizando así las funciones requeridas, tal y como están estipuladas en dichos casos de prueba, las entradas y salidas de los mismos y cumpliendo asimismo con los requisitos funcionales relacionados con ellos.

Este plan de pruebas busca comprobar que todas las funcionalidades estipuladas en los requisitos se llevan a cabo correctamente y sin errores de ningún tipo.

Los resultados del Plan de Pruebas demuestran que se han cumplido los objetivos propuestos inicialmente en la Fase de Planificación del proyecto.

Durante la realización de las pruebas, se detectaron errores, y de ellos se extrajeron enseñanzas que ayudaron a conseguir el resultado óptimo.

La Fase de Pruebas es importante para asegurar que el sistema realiza las funcionalidades correctamente y con total fiabilidad, evitando que en el momento en que el producto llegue al cliente, éste no encuentre errores.

7.- Plan de trabajo y presupuesto

En el siguiente apartado, se mostrará primero la planificación del proyecto y cómo se ha organizado el trabajo a lo largo del proyecto mediante un diagrama de Gantt, detallando las tareas que se han realizado.

Después, se expondrá el presupuesto del proyecto, obteniendo los recursos utilizados en cada una de las tareas realizadas en el proyecto, por cada uno de los roles de trabajadores necesarios para la consecución del proyecto. Por último, se calculará, en base a los datos anteriormente citados y a la identificación de los diferentes gastos, el presupuesto total de este proyecto.

7.1.- Diagrama de Gantt

Para representar la planificación del proyecto, se ha utilizado un diagrama de Gantt. En él se muestran las tareas del proyecto y el intervalo de tiempo en el que fueron realizadas, con una fecha de inicio de trabajo en la tarea y otra fecha de fin.

Las tareas del proyecto están agrupadas en Fases, las cuales son: Fase de Planificación, Fase de Investigación, Fase de Desarrollo, Fase de Pruebas y Fase de Documentación.

El proyecto abarca las fechas desde el 19 de septiembre de 2016, que es cuando se realiza la primera reunión con el tutor para determinar los objetivos y el alcance del proyecto, hasta el 26 de septiembre de 2017, fecha de entrega de la memoria del Trabajo de Fin de Grado y, por tanto, fecha de finalización del proyecto.

Se ha optado por una planificación por fases principalmente porque las tareas se agrupaban en distintos objetivos generales, y con el fin de optimizar tiempo y recursos, se optó por realizar las tareas que iban en línea con el mismo objetivo en fechas cercanas.

Las tareas dentro de una Fase se pueden realizar de forma simultánea, pero las Fases entre sí no se pueden superponer en la planificación de este proyecto, ya que para comenzar un

bloque de tareas de una fase es necesario que termine el bloque de tareas de la fase anterior, guardando una relación de dependencia la fase posterior de la anterior.

En la figura que se muestra a continuación se muestra el diagrama de Gantt del proyecto.

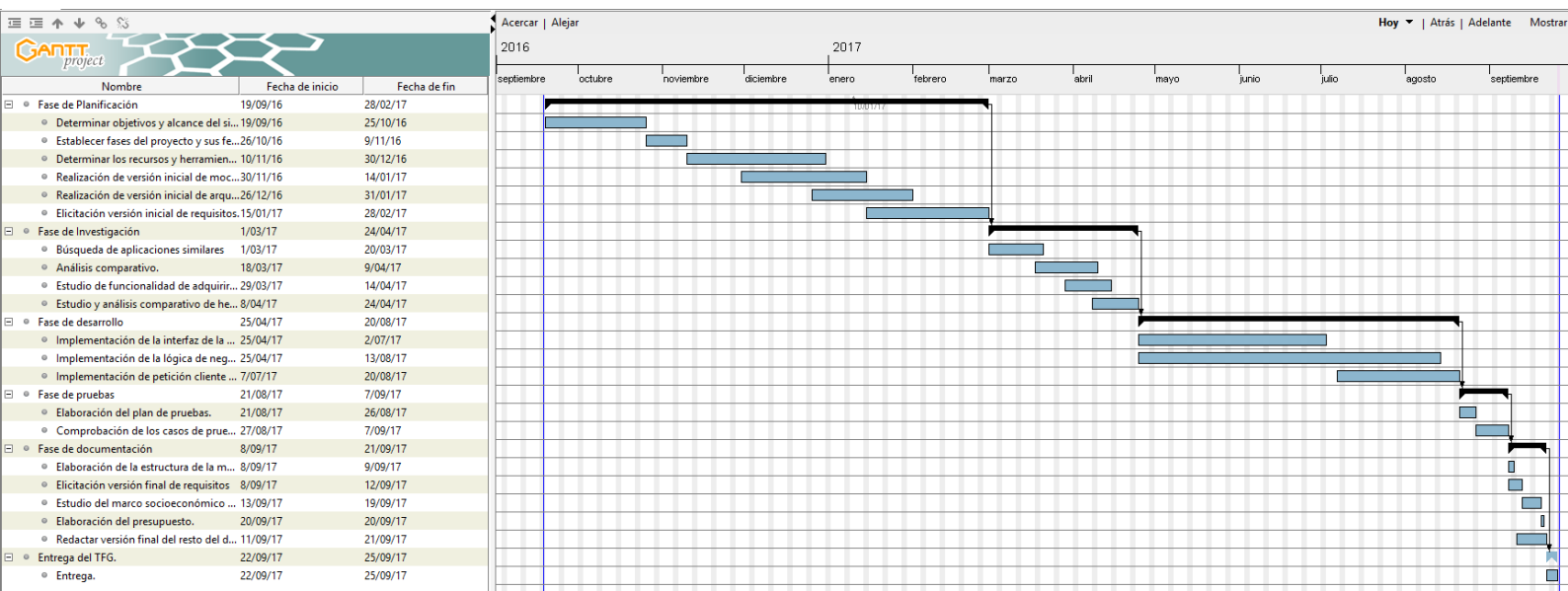


ILUSTRACIÓN 39. DIAGRAMA DE GANTT COMPLETO

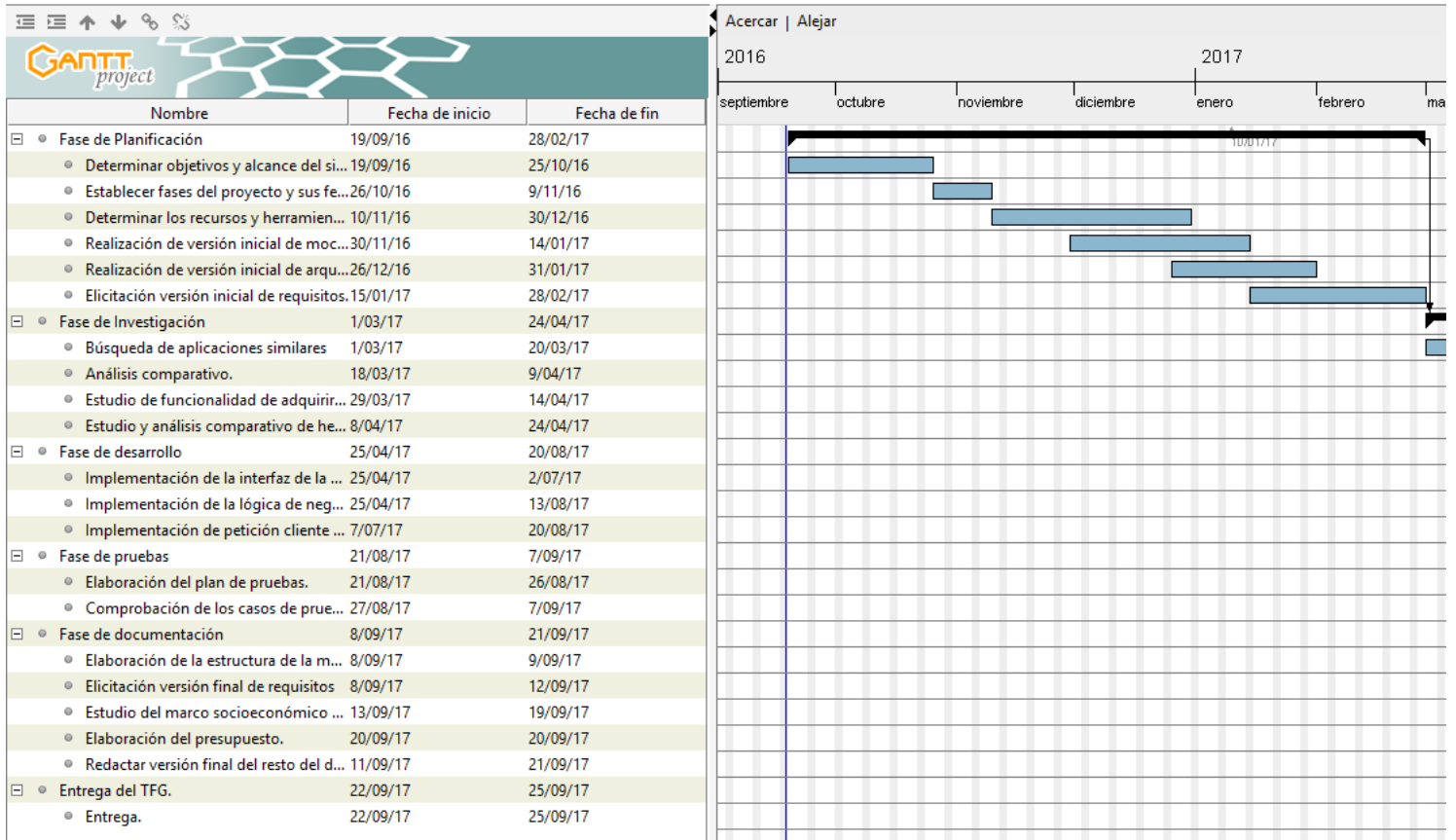


ILUSTRACIÓN 40. DIAGRAMA DE GANTT PARTE 1

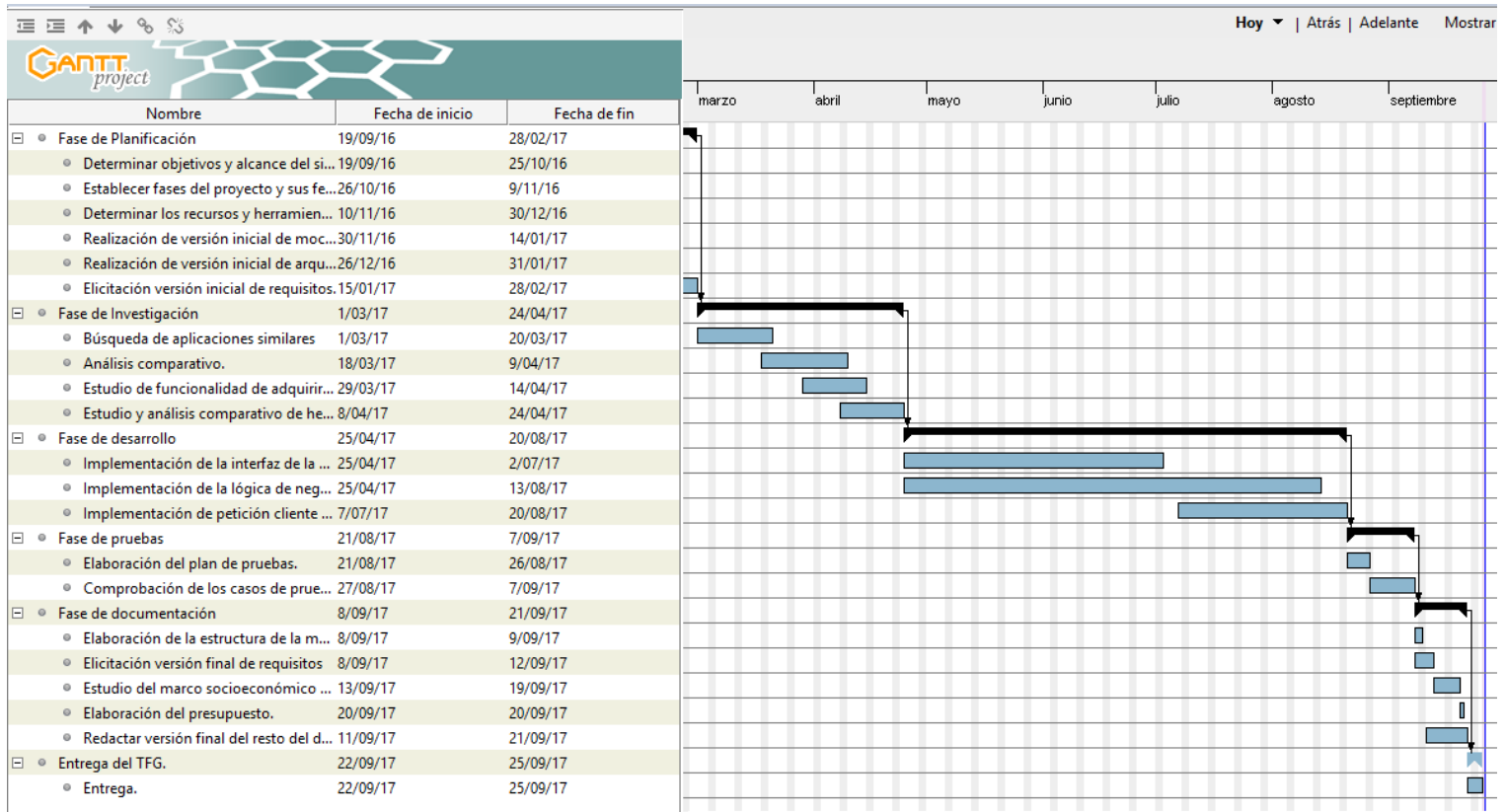


ILUSTRACIÓN 41. DIAGRAMA DE GANTT PARTE 2

7.2.- Presupuesto

A continuación, se detalla el presupuesto del proyecto. En la siguiente tabla se expone el coste total en euros por cada cargo de empleado. También se detalla el número de horas que se han requerido para llevar a cabo cada una de las tareas del proyecto. Cada tarea tiene asignada una serie de horas de trabajo para cada rol de compañero de equipo de trabajo, habiendo casos en los que alguno de ellos no participa en alguna tarea, generalmente porque no pertenece a su campo de especialización.

Actividad	Jefe de Proyecto	Analista	Programador	Gestor de calidad y pruebas
Determinar objetivos y alcance	4	10	10	5
Establecer fases del proyecto	2	20	5	5
Determinar recursos	2	10	5	-
Versión inicial Mockup	4	10	10	-
Versión inicial arquitectura	2	15	5	-
Versión inicial requisitos	2	20	5	-
Búsqueda de aplicaciones similares	4	5	5	-
Análisis comparativo	4	5	5	-
Estudio funcionalidad Esios	3	10	20	-
Estudio de herramientas de desarrollo	2	10	20	-
Implementación interfaz	2	5	60	-
Implementación lógica	1	10	80	-
Implementación petición Esios	2	5	45	-
Elaborar plan de pruebas	2	-	-	20
Comprobar casos de prueba	2	-	-	35
Estructura memoria	4	10	-	5
Versión final de requisitos	2	5	5	-
Marco socioeconómico	2	5	-	-
Presupuesto	2	5	-	-
Versión final del resto de documentación	2	5	5	5
TOTAL	50	165	285	75

ILUSTRACIÓN 42. TABLA RECURSOS UTILIZADOS POR TAREAS Y ROLES DE EMPLEADO

Cargo	Coste por hora	Número de horas	Coste total
Jefe de Proyecto	60 euros	50	3000 euros
Analista	30 euros	165	4950 euros
Programador	25 euros	285	7125 euros
Gestor de calidad y pruebas	15 euros	75	1125 euros
Total	130 euros	575	16.200 euros

ILUSTRACIÓN 43. COSTE DE PERSONAL

Lo que nos deja un presupuesto de personal total igual a 16.200 euros.

Ahora calcularemos el coste total de los recursos físicos del proyecto.

Componente	Descripción	Uds.	Precio Ud.	Total
Ordenadores	Basic Enterprise Pro i5-4460 (teclado y ratón incluidos)	4	529,00 €	2.116,00 €
Servidores	Western Digital Nas CloudWD Ex4	1	349,00 €	349,00 €
Sistema Operativo	Microsoft Windows 10 Pro 64 bits	4	279,00 €	1.116,00 €
			Total	3.581,00 €

ILUSTRACIÓN 44. COSTE DE RECURSOS FÍSICOS

Y el coste total de los recursos software para la realización del proyecto.

Componente	Descripción	Uds.	Precio Ud.	Total
Entorno de desarrollo	<ul style="list-style-type: none"> Android Studio 	4	0 €	0 €
Herramienta de documentación	<ul style="list-style-type: none"> Paquete Windows Office 	4	149,00 €	596,00 €
Herramientas de modelado	<ul style="list-style-type: none"> Mockflow Visual Paradigm GanttProject 	4 4 4	0 € 0 € 0 €	0 €
Total				596,00 €

ILUSTRACIÓN 45. COSTE RECURSOS SOFTWARE

- La amortización de equipos de trabajo:** $[(\text{Ordenadores} + \text{Sistema Operativo}) * \text{duración del proyecto en meses}] / \text{vida útil en meses} = [(2.116 + 1116) * 12] / 36 = \mathbf{1077,33 \text{ €}}$

Tipo coste recurso	Coste
Físico	3.581,00 €
Software	596,00 €
Total	4.177,00 €

ILUSTRACIÓN 46. SUMA DE COSTES

Lo que nos arroja que el coste total de las herramientas necesarias para llevar a cabo el proyecto es un total de 4.177 euros.

ILUSTRACIÓN 47. CÁLCULO FINAL DEL PRESUPUESTO TOTAL DEL PROYECTO

Tipo de coste	Coste
Personal	16.200,00 €
Recursos físicos y software	4.177,00 €
Total	20.377,00 €
Beneficio*	5.094,25 €
Riesgo**	4.177,00 €

**Consideramos el riesgo como la cobertura total de los costes del proyecto.

*Beneficio del 25%. Vendiendo la aplicación a 0,80 €, y con un mercado en el que hay aproximadamente 500.000 usuarios, para obtener el 25 % de beneficios bastaría con realizar ventas en un 1,3 % (siendo exactos, un 1,2735 %).

Lo que nos indica que el coste total del proyecto asciende a un total de 20.377 euros, sumando gastos de personal y gastos en las herramientas necesarias.

8.- Conclusiones generales y trabajo futuro

Finalmente, habiendo realizado un estudio de del alcance del sistema y del problema que se planteaba, habiendo analizado el mercado de las aplicaciones móviles gestoras de consumo eléctrico en España y su entorno, además de realizar el diseño de la arquitectura del sistema desde distintas perspectivas, así como analizar y describir las funcionalidades que se han implementado y realizando el estudio del marco del proyecto tanto socioeconómico, por un lado, para entender las repercusiones de la distribución de esta aplicación en la sociedad y el mercado actuales, como regulador y legislativo, por otro, para describir la perspectiva legal del proyecto, se puede confirmar que se han cumplido todos los objetivos propuestos para la elaboración final de este proyecto.

Por todo ello, los objetivos generales logrados han sido:

1. Se ha conseguido desarrollar un método de obtención de las tarifas eléctricas, para poder calcular el gasto que conlleva la planificación de las tareas del hogar, y se ha logrado implantar con éxito en la aplicación.
2. También se ha logrado implementar un gestor de información que permita obtener los datos de los usuarios, tales como tareas que suelen realizar que conllevan un consumo eléctrico, consumo de los electrodomésticos que tienen en casa o la agenda de tareas cotidianas de la semana y persistirlos utilizando una base de datos.
3. Con la información almacenada de la base de datos, se ha desarrollado una aplicación capaz de calcular el gasto eléctrico de la semana en función de la variabilidad por horas del precio de las tarifas eléctricas sin problemas de rendimiento ni de tiempos de respuesta de la aplicación debido a acceso a base de datos, gracias a la elección de un SGBD tan ligero como es SQLite.
4. Se ha logrado mostrar al usuario información detallada del gasto semanal de su hogar, para que sea capaz de identificar en qué aspectos de su planificación realiza un gasto eléctrico mayor, y poder así corregirlo a gusto del propio usuario.
5. Se ha conseguido desarrollar una interfaz de usuario intuitiva y fácil de manejar, que permite al usuario utilizar todas las funcionalidades de la aplicación sin ningún tipo de problemas.

6. Además, se ha logrado realizar un análisis exhaustivo del mercado, del marco socioeconómico y legal del proyecto, gracias a los cuales será más fácil distribuir la aplicación en el mercado de las aplicaciones móviles Android.

Para llegar a cumplir todos estos objetivos, se ha realizado un estudio exhaustivo del mercado eléctrico y de qué sitios web proporcionan las tarifas eléctricas, además del método más óptimo para el proyecto de obtener esta información.

Se han realizado análisis comparativos del mercado de las aplicaciones móviles similares, con el objetivo de estudiar el mercado y detectar las funcionalidades imprescindibles para toda aplicación de esta clase, así como del análisis del resto de herramientas y recursos del proyecto, para conseguir la consecución del mismo buscando el menor gasto y la máxima eficacia y eficiencia en el trabajo.

También se ha trabajado mucho en la implementación de una interfaz usable y en unas funcionalidades que cumplan los objetivos propuestos, estudiando las mejores opciones para conseguirlo con el menor gasto de recursos posible.

El resultado ha sido esta aplicación, llamada Voltaic, un asistente personal móvil para la gestión del consumo eléctrico en el hogar en plataforma Android, capaz de planificar el gasto eléctrico de los hogares y con las herramientas necesarias para ahorrar en la factura de la luz.

Pero para acabar finalizando con éxito este proyecto, hubo que pasar por muchos problemas y dificultades.

En las etapas tempranas del proyecto, me resultó complicado entender cuál debía ser el entorno de desarrollo, y en consecuencia cometí muchos errores: de librerías, de configuración del proyecto, etc.

Cuando por fin conseguí averiguar la configuración correcta de mi entorno de desarrollo, comencé a investigar cómo conseguir las tarifas eléctricas, y me costó mucho, la verdad. Fue una ardua tarea de búsqueda de información, y fue complicado encontrarla ya que es un problema muy concreto, y del que hay pocos ejemplos o documentación al respecto. Fue lo que más me costó averiguar, pero aprendí mucho.

También me fue complicado aprender a usar Android Studio, ya que nunca lo había utilizado, pero con el tiempo me fui haciendo a ello.

Cabe destacar que lo que más me ha costado en líneas generales, y aunque resulte obvio, es la implementación de la aplicación. Sin duda es la parte del proyecto que requirió más horas de trabajo.

Respecto a la documentación, fue complicado organizar la estructura de la memoria y decidir que modelos del sistema quería mostrar para describir el sistema, además de realizarlos.

Pero, con todas estas complicaciones, he aprendido mucho sobre programación de aplicaciones Android, sobre el mercado eléctrico y el ahorro que puede suponer saber optimizar el gasto que se produce en un hogar, y, sobre todo, he aprendido a afrontar un proyecto de semejante envergadura y con tantas horas de trabajo que requiere.

Respecto **al trabajo futuro** que se podría realizar con este proyecto, hay bastantes opciones:

- Se podría expandir el mercado de la aplicación implementando versiones para IOS y Windows Phone, permitiendo que llegue la aplicación a más sectores del mercado y pudiendo aumentar los beneficios.
- Implementar la **funcionalidad Domicilios**. Consiste en que cada usuario puede tener uno o más domicilios que gestiona en su cuenta, los cuales cada uno tienen sus electrodomésticos, tareas del hogar, etcétera. Con esta funcionalidad, se podrían compartir los domicilios con otras personas, por ejemplo, con otra gente con la que vives pero que no quiere tener que introducir todos los datos de nuevo. Esto provocaría que el potencial de extensión por el mercado de la aplicación aumentara.
- Implementación de chat para que los usuarios pertenecientes a un domicilio pudieran comunicarse entre ellos, facilitando así la planificación de las tareas del hogar en equipo.
- Se podría expandir la obtención de tarifas eléctricas a otros países, no solo a España. ESIOS es un Sitio Web de la Red Eléctrica de España y por tanto solo proporciona datos de las tarifas de este país. Para poder introducir nuestra aplicación a otros países se podría aumentar la funcionalidad de solicitud de tarifas eléctricas a las redes eléctricas de otros países y así aumentar el mercado de la aplicación.
- Implementar, mediante la compatibilidad con sensores de corriente eléctrica, el recibimiento de datos de consumo de los electrodomésticos del hogar sin necesidad de introducirlos manualmente por el usuario, lo cual podría resultar más cómodo para un sector de los usuarios.
- Desarrollar un sistema de sonido para para facilitar el uso de personas con deficiencias.

- En un futuro, y en caso de ser necesario, se podrían añadir más servidores. Esto sería necesario en caso de que la aplicación tuviera una distribución elevada en el mercado y la demanda aumentara rápidamente, por tanto, tendríamos que estar preparados para aumentar la capacidad de los servidores de la aplicación con la misma rapidez.
- Añadir más seguridad a la aplicación. Elaborar un encriptado RSA para operaciones externas con servidor y base de datos. Tomar mayores medidas de seguridad aseguraría los datos de los usuarios y la aplicación de ataques maliciosos.

9.- Bibliografía

- academiaandroid.com. (2017). Obtenido de <https://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/>
- BOE. (s.f.). Obtenido de <https://www.boe.es/buscar/doc.php?id=BOE-A-1999-23750>
- BOE. (s.f.). Obtenido de <https://www.boe.es/buscar/act.php?id=BOE-A-2002-13758>
- BOE. (s.f.). Obtenido de <https://www.boe.es/buscar/doc.php?id=BOE-A-1988-26156>
- BOE. (s.f.). Obtenido de <https://www.boe.es/buscar/doc.php?id=BOE-A-2006-12308>
- BOE. (s.f.). Obtenido de https://www.boe.es/diario_boe/txt.php?id=BOE-A-2015-8328
- BOE, J. d. (s.f.). Obtenido de <https://www.boe.es>
- developer.android.com. (s.f.). Obtenido de <https://developer.android.com/studio/intro/index.html?hl=es-419>
- developer.android.com. (2017). Obtenido de <https://developer.android.com/guide/topics/media/media-formats.html>
- E.sios, R. E. E. (2017). Obtenido de <https://www.esios.ree.es/es/aviso-legal-y-politica-de-privacidad>
- eldiario.es. (2017). Obtenido de www.energiaoscura.es/como-se-fijan-los-precios/mercado-diario/
- Eurostat. (2015). Obtenido de <http://ec.europa.eu/eurostat/web/energy/overview>
- Ezequiel, A. (2014). Obtenido de <http://www.programandoamedianoche.com/2014/06/como-crear-una-aplicacion-desde-cero-con-android-studio/>
- muycomputer.com. (2016). Obtenido de http://www.muycomputer.com/2017/08/28/android-cuota/?utm_content=buffer7d02a&utm_medium=social&utm_source=facebook.com&utm_campaign=buffer
- S.A., R. E. E. (2017). Obtenido de <https://api.esios.ree.es/>
- Tabian, M. (8 de Mayo de 2016). Obtenido de <https://www.youtube.com/watch?v=TNeE9DJJoOMY>
- Technologies, N. (4 de Enero de 2016). Obtenido de <https://www.numericlabz.com/android-line-chart-using-mpandroidchart-tutorial/>
- Topolsky, J. (2007). Obtenido de <https://www.engadget.com/2007/11/12/googles-android-os-early-look-sdk-now-available/>
- Wikipedia. (2017). Obtenido de <https://es.wikipedia.org/wiki/Android>

Wikipedia. (2017). Obtenido de <https://es.wikipedia.org/wiki/SQLite>

Wikipedia. (2017). Obtenido de
<https://es.wikipedia.org/wiki/Modelo%2%80%93vista%2%80%93controlador>

10.- Acrónimos y definiciones.

A

- Android: Sistema Operativo para dispositivos móviles.
- Android Studio: Herramienta de desarrollo de aplicaciones Android.
- API: Interfaz de Programación de Aplicaciones.
- App: Abreviatura de aplicación.

B

- BD: Base de Datos.
- BBDD: Bases de Datos.

C

- CDU: Caso de Uso.

D

- Diagrama: Representación gráfica del sistema.

E

- E.SIOS: Sistema de Información del Operador del Sistema

G

- Gantt: Tipo de diagrama utilizado para mostrar los plazos de las fases y tareas de un proyecto, así como los recursos utilizados o los empleados que lo llevan a cabo.

I

- IDE: Entorno de Desarrollo Integrado.
- IOS: Sistema Operativo para dispositivos móviles.

J

- Java: Lenguaje de programación.
- JSON: Tipo de archivo.

k

- kWh: Kilovatio por hora.

M

- Modelo: Prototipo de referencia para construir un producto.
- MVC: Modelo Vista Controlador

P

- Persistencia: Almacenamiento en una base de datos.
- PHP: Lenguaje de programación para desarrollo web.
- PVPC: Precio Voluntario al Pequeño Consumidor.

S

- SGBD: Sistema Gestor de Bases de Datos.
- Smartphone: Teléfono inteligente.
- SQL: Lenguaje de programación.

T

- Tarifa eléctrica: Precio establecido del consumo de kilovatio por hora.
- TEP: Toneladas equivalentes de Petróleo.
- TFG: Trabajo de Fin de Grado.

U

- URL: Dirección de un sitio web, mediante la cual se puede acceder a él.

W

- Windows Phone: Sistema Operativo para dispositivos móviles.

X

- XML: Lenguaje de programación.