



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Trabajo Fin de Grado

**PLATAFORMA DE DATOS
ENERGÉTICOS
DE VIVIENDAS Y VEHÍCULOS
ELÉCTRICOS**

Alumno: María Horna Climent

Tutores: Prof. D. Miguel Ángel García Cumbreras
Prof. D. Jesús de la Casa Hernández

Dpto: Informática

Mayo, 2020

(Página intencionalmente en blanco)



Universidad de Jaén

Escuela Politécnica Superior de Jaén
Departamento de Informática

D. Miguel Ángel García Cumbreiras y D. Jesús de la Casa Hernández, tutores del Proyecto Fin de Carrera titulado: **‘Plataforma de datos energéticos de viviendas y vehículos eléctricos’**, que presenta Dña. María Horna Climent, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, mayo de 2020

El alumno:

Los tutores:

María Horna Climent

Miguel Ángel García Cumbreiras

Jesús de la Casa Hernández

(Página intencionalmente en blanco)

Agradecimientos

En primer lugar, me gustaría agradecer a mis abuelos, a mis padres. Ellos siempre me han apoyado incondicionalmente, para las buenas y para las malas y siempre han estado ahí dándome las fuerzas que necesitaba para seguir, no solo desde que empecé la carrera, sino toda la vida.

También agradecer a mi hermana, que siempre ha sido un pilar fundamental en mi vida y una gran ayuda para superarlo todo.

Mis compañeros y amigos de la universidad, especialmente a Alberto, Tony y Elena. Los conocí en un momento en el que necesitaba a amigos como ellos en mi vida y lo que he vivido con ellos estos años no podré olvidarlo nunca.

Mis profesores de la carrera, que no solo me han ayudado a superar las asignaturas, sino que me han sabido impartir los conocimientos necesarios para la entrada en el mundo laboral.

Y, por último, pero no menos importante, a mi tutor del TFG, Miguel Ángel García Cumbreiras. Me gustaría agradecerle enormemente su labor como docente y la gran ayuda que me ha dado para convertir mi agobio en motivación para llegar hasta el final de este proyecto.

NORMAS APLICADAS	
LOCAL	
TFT-EPSJ	Criterios de evaluación y normas de estilo para TFG y TFM de la Escuela Politécnica Superior de Jaén
TFT-EPSJ:2017	Normativa sobre Trabajos Fin de Grado y Fin de Máster en la Escuela Politécnica Superior de Jaén (Normativa EPSJ aprobada en Junta de Escuela)
TFT-UJA:2017	Normativa de Trabajos Fin de Grado, Fin de Máster y otros Trabajos Fin de Título de la Universidad de Jaén (Normativa marco UJA aprobada en Consejo de Gobierno)
TFT-UJA:2020	Normativa de Trabajos Fin de Grado, Fin de Máster y otros Trabajos Fin de Título de la Universidad de Jaén (Normativa marco UJA aprobada en Consejo de Gobierno durante el estado de alarma)
NACIONAL	
UNE 157001:2014	Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico
UNE 157801:2007	Criterios generales para la elaboración de proyectos de sistemas de información
INTERNACIONAL	
ISO 2145:1978	Documentación - Numeración de divisiones y subdivisiones en documentos escritos
APA 6ª EDICIÓN	Estilo de referencias y citas de APA (American Psychological Association)

Índice

1. ESPECIFICACIÓN DEL PROYECTO	11
1.1 - Índice general.....	11
1.2 - Documentación del proyecto.....	11
1.2.1 - Introducción	11
1.2.2 - Objetivo del proyecto	12
1.2.3 - Antecedentes	12
1.2.4 - Descripción de la situación actual.....	13
1.2.5 - Normas y referencias.....	14
1.2.6 - Definiciones y abreviaturas	15
1.2.7 - Requisitos iniciales	16
1.2.8 - Alcance	18
1.2.9 - Hipótesis y restricciones	18
1.2.10 - Estudio de alternativas y viabilidad	19
1.2.11 - Descripción de la solución propuesta	21
1.2.12 - Tareas realizadas	21
1.2.13 - Resumen del presupuesto	24
1.2.14 - Orden de prioridad de los documentos básicos del proyecto	24
1.3 - Especificaciones del sistema	24
1.4 - Presupuesto	25
1.5 - A1: Documentación de entrada.....	27
1.6 - A2: Análisis y Diseño del sistema	28
1.6.1 - Metodología de desarrollo	30
1.7 - A3: Estimación del tamaño y esfuerzo	30
2. REQUISITOS DEL PROYECTO	31
2.1 - Tecnologías utilizadas.....	31
2.2 - Diseño	34
2.2.1 - Diseño arquitectónico del sistema	34
2.2.2 - Diagramas de clases	35
2.2.3 - Diagramas de casos de uso	37
2.2.4 - Diagramas de secuencia	45
2.2.5 - Diseño de la interfaz y storyboards.....	51
2.3 - Implementación.....	55

2.3.1 - Implementación de scripts en Python	55
2.3.2 - Implementación de la aplicación en Angular.....	56
2.3.3 - Conclusiones finales	59
2.4 - Pruebas finales	60
2.4.1 - Pruebas de verificación del script Python	60
2.4.2 - Pruebas de verificación de la aplicación Angular.....	60
2.4.3 - Pruebas de rendimiento del sistema.....	61
2.5 - Demostración de la aplicación Angular desarrollada	62
3 - CONCLUSIONES Y MEJORES FUTURAS.....	65
4 - APÉNDICES	66
4.1 - Instalación y configuración del sistema.....	66
4.2 - Manuales de usuario	71
5 - REFERENCIAS.....	80

Índice de ilustraciones

Ilustración 1.1. Proyectos SmartGrid abiertos en Europa desde 2008	13
Ilustración 1.2. Circuito completo SCT-013 y Arduino sin Jack hembra.	17
Ilustración 1.3. Circuito completo SCT-013 y Arduino con Jack hembra.	17
Ilustración 1.4. Cantidad de estrellas que obtienen sus repositorios de GitHub.	20
Ilustración 1.5. Planificación del proyecto.....	22
Ilustración 1.6. Diagrama de Gantt	23
Ilustración 1.7. Comparativa de planes de Cloud Firestore	26
Ilustración 1.8. Ejemplo de plataforma con Power BI	28
Ilustración 1.9. Ejemplo de plataforma ELK.....	29
Ilustración 2.1. Clases de la aplicación.....	34
Ilustración 2.2. Diagrama de clases del sistema de gestión de usuarios.....	35
Ilustración 2.3. Diagrama de clases de la visualización de los datos.	36
Ilustración 2.4. Caso de uso “Iniciar sesión (Usuario)”	38
Ilustración 2.5. Caso de uso “Iniciar sesión (Administrador)”	39
Ilustración 2.6. Caso de uso “Registrar usuario”	40
Ilustración 2.7. Caso de uso “Cerrar sesión”	41
Ilustración 2.8. Caso de uso “Visualizar perfil”	42
Ilustración 2.9. Caso de uso “Elegir opciones de la gráfica”	43
Ilustración 2.10. Caso de uso “Hacer zoom”	44
Ilustración 2.11. Caso de uso “Quitar zoom”	45
Ilustración 2.12. Diagrama de secuencia “Iniciar Sesión (Usuario)”	46
Ilustración 2.13. Diagrama de secuencia “Iniciar Sesión (Administrador)”	47
Ilustración 2.14. Diagrama de secuencia “Registrar”	48
Ilustración 2.15. Diagrama de secuencia “Cerrar sesión”	49
Ilustración 2.16. Diagrama de secuencia “Visualizar perfil”	49
Ilustración 2.17. Diagrama de secuencia “Hacer Zoom”	50
Ilustración 2.18. Diagrama de secuencia “Quitar Zoom”	51
Ilustración 2.19. Storyboard. Pantalla 1.....	52
Ilustración 2.20. Storyboard. Pantalla 2.....	52
Ilustración 2.21. Storyboard. Pantalla 3.....	53
Ilustración 2.22. Storyboard. Pantalla 4.....	54
Ilustración 2.23. Storyboard. Pantalla 5.....	54

Ilustración 2.24. Pantalla de inicio de sesión	62
Ilustración 2.25. Pantalla de registro	62
Ilustración 2.26. Pantalla del perfil del usuario	63
Ilustración 2.27. Ejemplo de visualización de datos en la gráfica	63
Ilustración 2.28. Visualización de la gráfica sin datos.....	64
Ilustración 4.1. Apertura de Node con Visual Studio Code	67
Ilustración 4.2. Apertura de la terminal de Visual Studio Code.....	67
Ilustración 4.3. Consola de Windows.....	68
Ilustración 4.5. Crear tarea automática en Windows	69
Ilustración 4.6. Pestaña General de Crear Tarea	70
Ilustración 4.7. Crear Nuevo Desencadenador	70
Ilustración 4.8. Crear una Nueva acción.....	71
Ilustración 4.9. Pantalla de inicio de sesión detallada	72
Ilustración 4.10. Pantalla de registro detallada	73
Ilustración 4.11. Pantalla de registro de ejemplo	73
Ilustración 4.12. Pantalla principal detallada.....	74
Ilustración 4.13. Pantalla de perfil detallada	74
Ilustración 4.14. Filtros detallados	75
Ilustración 4.15. Gráfica de muestra.....	76
Ilustración 4.16. Gráfica de muestra con una opción de cada filtro	77
Ilustración 4.17. Gráfica de muestra con más de una opción de cada filtro	77
Ilustración 4.18. Gráfica de muestra con más de una opción de cada filtro	78
Ilustración 4.19. Puntos clave de la gráfica	78

Índice de tablas

Tabla 2.1. Caso de uso “Iniciar sesión (Usuario)”	37
Tabla 2.2. Caso de uso “Iniciar sesión (Administrador)”	38
Tabla 2.3. Caso de uso “Registrar usuario”	40
Tabla 2.4. Caso de uso “Cerrar sesión”	41
Tabla 2.5. Caso de uso “Visualizar perfil”	42
Tabla 2.6. Caso de uso “Elegir opciones de la gráfica”	43
Tabla 2.7. Caso de uso “Hacer zoom”	44
Tabla 2.8. Caso de uso “Quitar zoom”	45

1. ESPECIFICACIÓN DEL PROYECTO

En este capítulo se hará una especificación del proyecto, siguiendo una estructura y contenido regidos según los criterios que establece la norma UNE 157801:2007 - “Criterios Generales para la elaboración de proyectos de Sistemas de Información”.

1.1 - Índice general

Para el índice de la documentación seguida por la norma UNE 157801 se utilizará una tabla con un contenido global del TFG actual, ya que los documentos contenidos en esta norma estarán dentro del documento maestro.

1.2 - Documentación del proyecto

En este apartado se proporcionarán los contenidos de la especificación del proyecto requeridos por el TFG actual. En este, se hará una descripción de todo aquello que deberá entregarse al finalizar el proyecto, así como la especificación de aquellos métodos y herramientas utilizados para la ejecución del proyecto.

1.2.1 - Introducción

En este capítulo se hará una breve introducción para la memoria del trabajo de fin de grado del Grado en Ingeniería Informática. A continuación, se definirán los objetivos básicos que han de realizarse.

El objetivo principal es el estudio y desarrollo de una plataforma web para el Equipo de Investigación de Energías de la Universidad de Jaén. Esta plataforma englobará los datos energéticos recogidos de viviendas, fotovoltaica y vehículos eléctricos.

Utilizando los datos que se registran de distintos sensores. Se hará uso de métodos y herramientas de análisis de datos para mostrar en una interfaz web el resultado de los mismos.

El proyecto está orientado al aprendizaje y estudio de tecnologías sobre las que no se ha podido profundizar durante el grado. Dicha plataforma estará limitada al uso de las personas autorizadas.

1.2.2 - Objetivo del proyecto

Los objetivos principales de este Trabajo fin de Grado son:

- Estudio y desarrollo de una plataforma para visualizar los datos, que previamente han sido obtenidos de diferentes viviendas, de la fotovoltaica y de coches eléctricos.
- Posterior visualización de los datos en una plataforma web para que puedan ser de utilidad para los usuarios finales y para la toma de decisiones.

Como objetivos secundarios se realizarán:

- Sistema de login para que las personas autorizadas puedan entrar a visualizar la gráfica.
- También será necesario realizar un script el cual recoja diariamente los datos de la base de datos utilizada, genere un JSON, los comprima y los suba a una cuenta Drive. Para este objetivo también se necesitarán realizar las correspondientes exportaciones e importaciones de un gran volumen de datos entre distintos sistemas de bases de datos.
- Por otra parte, y si es necesario, se mostrarán más gráficas, aparte de la principal que ayuden a una mejor visualización de los datos.

1.2.3 - Antecedentes

El proyecto surge como una necesidad para el Equipo de Investigación de Energías Renovables ya que los datos energéticos se estaban recogiendo y almacenando, pero no se estaban visualizando con facilidad en una plataforma web.

En este proyecto se pretende que la red de distribución eléctrica sea inteligente mediante la inclusión de tecnologías de la información, por lo tanto, es importante mencionar lo que es una Smart Grid o red inteligente. Se trata de una red que incorpora tecnología digital a las redes tradicionales y que consigue una comunicación bidireccional fluida entre las instalaciones y los usuarios. Estas redes hacen uso de herramientas informáticas, de la tecnología más puntera, de lo último en equipamiento y en domótica.



Ilustración 1.1. Proyectos SmartGrid abiertos en Europa desde 2008

Por lo tanto, este proyecto serviría para la parte informática ya que ayudaría a monitorizar tanto los consumos energéticos de un edificio como los de un conjunto de edificios dispersos, mostrando los datos de una forma sencilla a través de gráficos para facilitar la toma de decisiones a los gestores y así poder realizar previsiones de facturación. Gracias a esto, es posible tener un control exhaustivo del consumo energético, pudiendo así detectar anomalías o si se estuvieran activando consumos innecesarios fuera de los horarios con más ventajas para una facturación eléctrica.

1.2.4 - Descripción de la situación actual

Hoy en día existen algunas aplicaciones con las que se obtienen resultados similares a los de la aplicación. Sin embargo, son algo complejas y tienen funcionalidad de más, que para lo que se ha pedido no es necesario.

1.2.4.1 - Descripción del entorno actual

El proyecto está orientado en un entorno a nivel de un usuario medio, sirviendo este tanto para clientes que tengan el sistema de consumo instalado en sus casas y/o

coches, como para el equipo de investigación. Esta aplicación no tendrá un diseño complejo por lo que cualquier usuario debería poder utilizarlo sin problema.

1.2.4.2 - Principales deficiencias identificadas

Las principales deficiencias que se puede encontrar en la mayoría de las aplicaciones software que monitorizan el consumo eléctrico son la dificultad y los tecnicismos. Cuando un usuario sin conocimiento avanzado abre por primera vez una aplicación de este tipo podría agobiarse viendo la cantidad de información que se muestra en pantalla.

Un usuario de nivel medio no tiene por qué saber que significa cada tecnicismo ni cada unidad de medida del ámbito de la electrónica, por lo tanto, la situación más viable para dicho usuario sería buscar la información por internet.

Por lo tanto, lo mejor sería plantear una interfaz la cual no abrume al usuario y se sienta cómodo. Esta interfaz deberá incluir lo básico para visualizar correctamente la aplicación y un menú sencillo.

Con esto, finalmente se podrá visualizar el resultado esperado de forma intuitiva y sencilla, sin tener que agobiar a un usuario con conocimientos básicos.

1.2.5 - Normas y referencias

En este apartado de verán los reglamentos y referencias seguidos en la realización y ejecución del proyecto. La norma UNE 157801 incluirá un apartado con la bibliografía, pero al incluir el documento de especificación del proyecto en la memoria, la bibliografía de este capítulo se incluye con la bibliografía del documento maestro.

1.2.5.1 - Metodología, herramientas, modelos, métricas y prototipos

Para la realización del proyecto se va a seguir una metodología de desarrollo ágil por iteraciones, más concretamente Kanban. Esta tecnología es una de las llamadas metodologías ágiles, aquellas que buscan gestionar de manera generalizada cómo se van completando las tareas y es un proceso que ayuda a las personas a resolver problemas y terminar proyectos de forma productiva. Para organizar el proyecto,

utilizaremos Trello y se añadirán tres listas: Tareas por realizar, tareas en proceso y tareas finalizadas.

Se ha elegido este tipo de metodología para conseguir una funcionalidad básica en la implementación y aumentar un MVP, es decir, un proyecto mínimo viable.

Las herramientas para el desarrollo de la aplicación son:

- Como IDE se utilizará Visual Studio Code¹ (versión gratuita).
- Para el desarrollo de la aplicación web se analizarán con detalle cada una de las tecnologías que pueden ser más adecuadas y en posteriores capítulos se comentará de forma justificada la decisión que se ha tomado.
- Para la ingeniería del software se utilizará Visual Paradigm².

1.2.5.2 - Mecanismo de control de calidad

Para asegurar una excelente calidad en la realización del proyecto hay que verificar la integridad a la hora de documentar, garantizar que sea un documento completo, así como una correcta redacción para que sea legible y entendible por parte de todos los colaboradores y lectores del proyecto.

Este proyecto estará realizado por una sola persona y será revisado por un solo tutor, por esto no será necesario llevar un documento con el control y revisión de la documentación. Sin embargo, el autor del proyecto utilizará mecanismos básicos para verificar dicha integridad y completitud de la documentación. Estos mecanismos incluyen un control de las versiones de la documentación y especificaciones del proyecto.

1.2.6 - Definiciones y abreviaturas

Es necesario conocer algunos términos más técnicos para una mejor comprensión del proyecto. Estos términos son:

- FI: Es la abreviatura de factor de potencia y es la relación entre la potencia activa, P, y la potencia aparente, S.

¹ <https://code.visualstudio.com/>

² <https://www.visual-paradigm.com/>

- I: Es la abreviatura de intensidad de corriente eléctrica y es la cantidad de electricidad o carga eléctrica(Q) que circula por un circuito en la unidad de tiempo(t). Su unidad es el Amperio(A).
- P: Es la abreviatura de potencia activa y es la que se aprovecha como potencia útil. También se llama potencia media, real o verdadera y es debida a los dispositivos resistivos. Su unidad de medida es el vatio (W).
- S: Es la abreviatura de potencia aparente y es la potencia total consumida por la carga y es el producto de los valores eficaces de tensión e intensidad. Se obtiene como la suma vectorial de las potencias activa y reactiva y representa la ocupación total de las instalaciones debida a la conexión del receptor. Su unidad de medida es el voltamperio (VA).
- V: Es la abreviatura de la tensión eléctrica o diferencia de potencial (también denominada voltaje) y es una magnitud física que cuantifica la diferencia de potencial eléctrico entre dos puntos. También se puede definir como el trabajo por unidad de carga ejercido por el campo eléctrico sobre una partícula cargada para moverla entre dos posiciones determinadas. Su unidad es el voltio.

Magnitud	Unidad	Variable (json)	Ecuación
Tensión	Voltios	v	---
Intensidad	Amperios	i	---
Potencia activa	Vatios	p	---
Factor de potencia	Sin unidad	fi	---
Potencia reactiva	Voltios Amperios reactivos	Se obtiene mediante cálculos. (opcional)	$q=v*i*\sin(\cos(fi))$
Potencia aparente	Voltios Amperios	s	---
Hora	---	hora+minuto+segundo	---
Fecha	---	ano+mes+día	---

1.2.7 - Requisitos iniciales

Para esta aplicación los requisitos básicos que hay que seguir son:

- Como requisito inicial, siendo este el más importante, se deberán recoger correctamente los datos de entrada ya que serán los que se usen para la representación visual de estos. Para la obtención de los datos se

implementará un circuito acondicionador de señal SCT-013, cuya función es convertir la corriente en un voltaje limitado que se pueda medir con una placa Arduino. Una vez que se tengan los datos se almacenan en la base de datos.

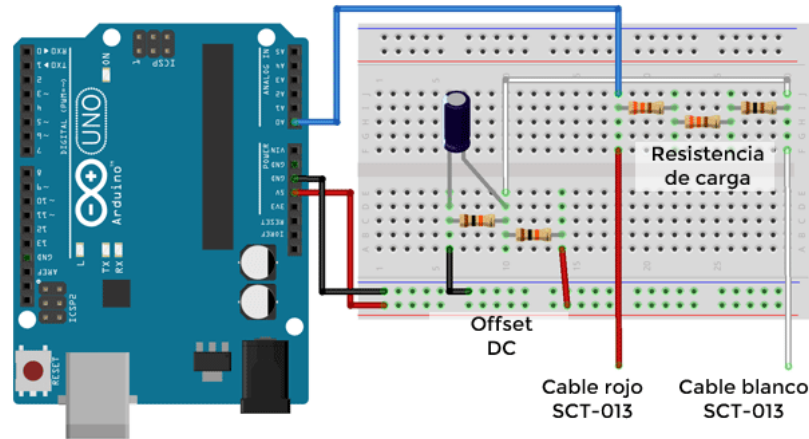


Ilustración 1.2. Circuito completo SCT-013 y Arduino sin Jack hembra.

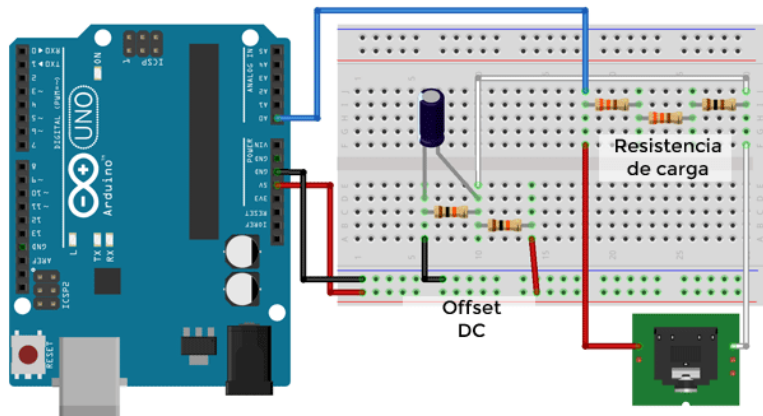


Ilustración 1.3. Circuito completo SCT-013 y Arduino con Jack hembra.

- Para que se puedan generar las gráficas, previamente se deben elegir correctamente unas opciones. Esto hará un filtrado de los parámetros que se mostrarán en las gráficas. Este filtrado se hará para que solo se muestren aquellos datos que se necesiten realmente.
- De los filtros citados anteriormente se podrán distinguir tres:

- o **Rango:** Este filtro será la fecha en la que estarán comprendidos los valores de la gráfica, por lo que se tiene que elegir una fecha inicial y otro final. Este filtro se hará para que no se muestren todas las fechas que se tengan disponibles sino la que se necesite.
- o **Dispositivos:** Este filtro tendrá todos los dispositivos donde se esté llevando la monitorización del consumo eléctrico. Este filtro se utiliza para que cada usuario elija solo sus dispositivos, que vendrán marcados con un nombre del dispositivo y el id.
- o **Unidad de medida:** Este filtro será cada una de las unidades de medida recogidas y almacenadas en la base de datos. Este filtro se utiliza para que el usuario pueda elegir todas las unidades que quiera y haya disponibles.

Habrà que hacer una elecci3n de cada uno de los filtros, sino no se podrà visualizar la gràfica.

- Por ùltimo, se podràn visualizar los datos en la interfaz gràfica de usuario.

1.2.8 - Alcance

Una vez que se finalice el proyecto se tendrà disponible:

- La aplicaci3n web para uso del usuario.
- Scripts programados para acciones de conversi3n de datos y de importaci3n/exportaci3n.
- Una documentaci3n final o memoria.
- Un manual de usuario y los anexos necesarios.

1.2.9 - Hip3tesis y restricciones

Este proyecto es una asignatura que se compone de 12 cr3ditos, por lo que serían unas 300 horas totales de trabajo. Estas horas deberàn incluir todas las etapas del ciclo de vida del proyecto. Por lo tanto, las principales restricciones que se aplican a la hora de la realizaci3n del trabajo son:

- La limitación de las horas de trabajo.
- La limitación del acceso a los datos reales.
- La limitación en cuanto a información sobre el TFG en la época de estado de alarma.

1.2.10 - Estudio de alternativas y viabilidad

Para empezar, se estudió la posibilidad de realizar la aplicación web con Angular³, con React ⁴o con Vue⁵.

Lo primero que tuve en cuenta fue la curva de aprendizaje ya que eran tres tecnologías totalmente desconocidas para mí y al disponer de un tiempo limitado, consideré que era importante con cuál de estas tecnologías obtendría un beneficio antes. En primer lugar, Angular puede ser una tecnología que toma más tiempo dominar y que para sentirte cómodo con ella, tome más tiempo. Por otro lado, React, que tiene librerías muy buenas y su aprendizaje podría ser sencillo. Y, por último, Vue, que, aunque es similar a React, tiene ventajas como que es un ecosistema más uniforme y sus librerías son muy similares por lo que es fácil sentirse productivo.

Lo segundo que estudié fue la robustez, ya que es una parte importante en las aplicaciones, sobre todo si futuramente se quiere hacer a gran escala. La robustez de Vue es la peor con respecto a las otras dos. Por otra parte, Angular incluye herramientas de la terminal, Angular Universal, librerías de interfaz gráfica entre otras muchas cosas y aunque React, que es una librería que cuenta con una gran comunidad detrás, Angular sería la ganadora porque desde el core está hecha para interactuar con todos estos componentes.

Lo tercero que habría que tener en cuenta sería las herramientas de desarrollo ya que, con el tiempo, dichas herramientas te pueden permitir programar más rápido, detectar errores de forma más sencilla, depurar problemas, etc. Esto es una gran ventaja en cuanto a ahorro de tiempo y eficiencia en el trabajo. Para empezar, en Angular todos los frameworks se crean usando Typescript y este lenguaje mejora considerablemente la experiencia de desarrollo de aplicaciones JavaScript. React

³ <https://angular.io/>

⁴ <https://es.reactjs.org/>

⁵ <https://vuejs.org/>

desarrolla herramientas como las React Developer Tools, una extensión para el navegador que tiene soporte para depurar aplicaciones. Por lo que, consideré que Vue era la que peores herramientas de desarrollo tenía en comparación.

En conclusión, considerando las ventajas y desventajas de las tres tecnologías llegué a la conclusión de que como tendría que diseñar y crear la interfaz desde cero y que todo pasaría por el framework, la mejor opción en este caso sería escoger Angular. El lenguaje de programación viene predefinido con la propia tecnología. No es una tecnología con la que haya trabajado previamente, sin embargo, hay una comunidad grande que la respalda y eso afianzó mi decisión a la hora de escogerla.

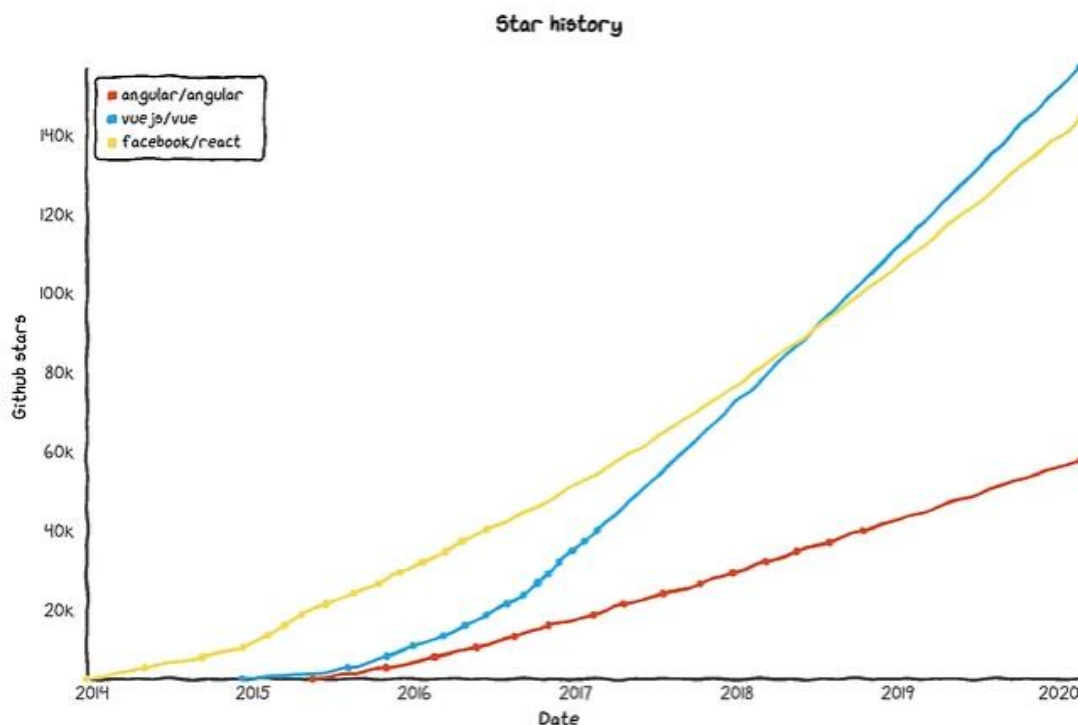


Ilustración 1.4. Cantidad de estrellas que obtienen sus repositorios de GitHub.

En esta aplicación, el principal problema lo ha presentado la base de datos. La conexión con esta no ha sido demasiado compleja pero sí que se han presentado problemas a la hora de manejar los datos. Más adelante se explicará cómo se ha desarrollado finalmente.

La viabilidad de este proyecto puede ser concebida desde el punto de vista de una organización como un software gratuito para dar a conocer dicha organización. Si

tenemos en cuenta esto, este proyecto sería viable para darle prestigio a la empresa. Sin embargo, este software tendrá una distribución gratuita, por lo tanto, no tendrá ningún beneficio económico.

1.2.11 - Descripción de la solución propuesta

En cuanto a la implementación:

- En primer lugar, se implementará un sistema de login, registro y perfil para que cada usuario pueda entrar bien con su cuenta Gmail o con un email y contraseña.
- Después, se implementarán unos scripts para el procesamiento de los datos y la posterior carga de los mismos en la base de datos.
- Se implementará una interfaz con la que el usuario podrá interactuar con los valores recogidos de la base de datos y podrá visualizarlos de forma correcta en una gráfica.
- No es necesaria la visualización de todos los datos, solo los que el usuario necesite por lo que una vez obtenidos los datos se implementará un sistema de filtrado.

1.2.12 - Tareas realizadas

Se prevé que este proyecto se podría implementar en aproximadamente cinco meses. Hay que indicar que los días en los que se ha hecho el desarrollo son días laborales. Para marcar las principales etapas de desarrollo se ha seguido una metodología ágil. Por lo tanto, el proyecto se podría dividir en iteraciones de menos tiempo:

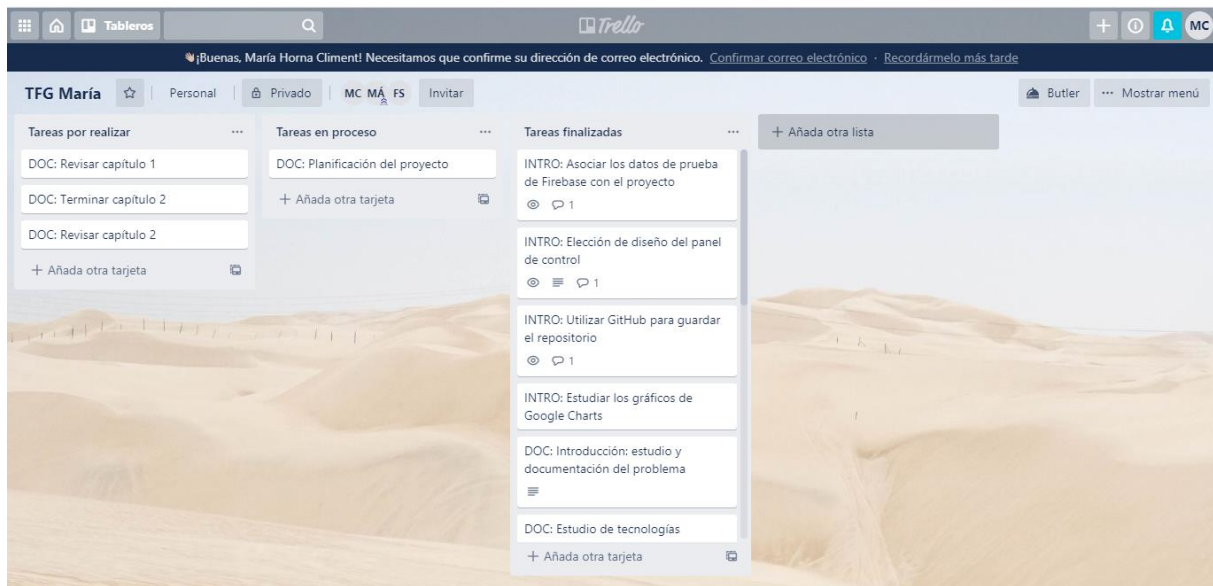


Ilustración 1.5. Planificación del proyecto

Primera iteración

Esta iteración sería más introductoria y estaría compuesta por la búsqueda de información, el análisis, una planificación del proyecto, asociación de GitHub con el proyecto y, por último, el estudio del sistema de gráficos que se va a utilizar.

- **Búsqueda de información:** Para empezar, habrá que buscar información acerca de las tecnologías que van a usarse, en este caso sobre Angular y Firebase⁶. Esta parte es muy importante ya que se estudiará la documentación de estas tecnologías y se realizarán pequeñas pruebas previas antes de empezar el proyecto para familiarizarse y saber qué posibilidades ofrecen.
- **Análisis:** Se realizará un estudio en el que se diseñarán los casos de uso y los diagramas de flujo necesarios. También se tratarán los requisitos funcionales y los no funcionales.
- **Planificación:** Se hará una estimación de los tiempos, un diagrama de Gantt donde se verá el desarrollo en el tiempo del proyecto y por último un análisis de costes estimado.

⁶ <https://firebase.google.com/>

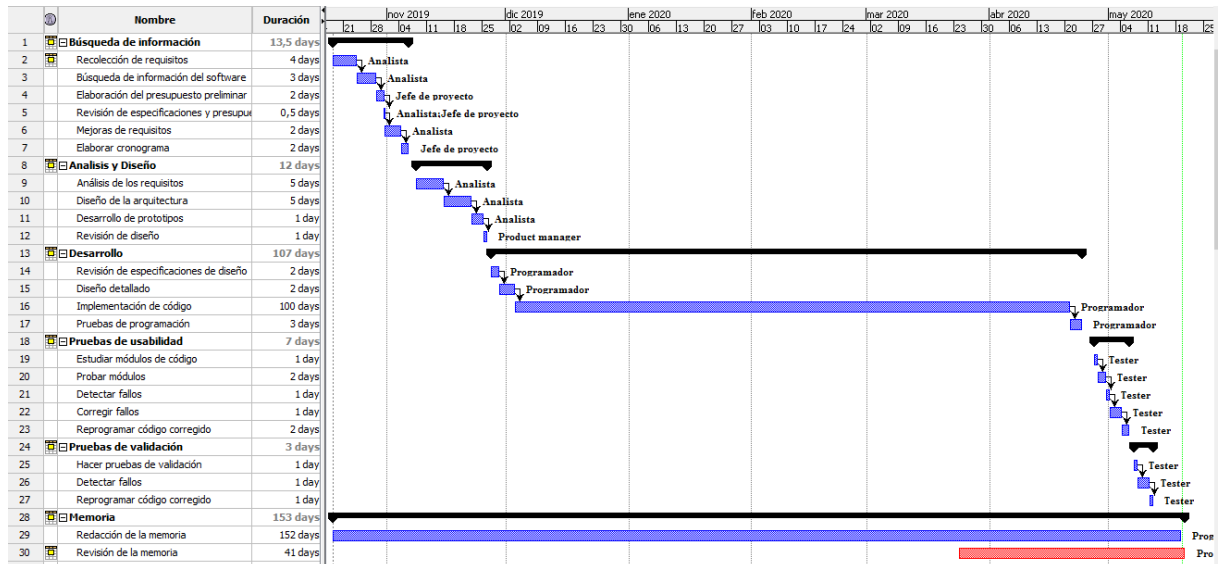


Ilustración 1.6. Diagrama de Gantt

Segunda iteración

Esta iteración sería la parte del desarrollo y estaría compuesta por la elección de un diseño para la aplicación, asociación de los datos de la base de datos con el proyecto, la creación de un sistema de gestión de usuarios, creación de scripts en Python para la importación/exportación de los datos, implementación de un sistema de filtrado de datos, implementación de las gráficas para la visualización y, por último, una serie de pruebas de validación.

- **Diseño e implementación:** Se hará un diseño de la interfaz inicial y otro con la interfaz final, la interacción que habrá con esta interfaz y, por último, una integración de los datos usados de la plataforma Firebase a la plataforma.
- **Pruebas:** Una vez que ya se haya completado la implementación del proyecto, se realizarán una serie de pruebas para comprobar que todo funciona como debería.

Tercera iteración

Esta iteración sería la parte de la documentación y estaría compuesta por la redacción de la memoria y la posterior revisión de esta.

- **Memoria:** Para finalizar, se hará una documentación paralela a la realización del proyecto y deberá de ser lo más detallada posible.

1.2.13 - Resumen del presupuesto

Para realizar un coste aproximado del proyecto tendremos que tener en cuenta tanto la parte software como la parte hardware. El coste total ascendería a unos 3000 euros. Posteriormente se detalla todo lo que da lugar a esta cifra.

1.2.14 - Orden de prioridad de los documentos básicos del proyecto

Este proyecto está integrado dentro de este capítulo del documento global con los contenidos que se especifican en la norma UNE 157801. El orden de prioridad que se ha utilizado es:

1. Documentación del proyecto.
2. Especificaciones del sistema.
3. Presupuesto.
4. Estudios con entidad propia.
5. Anexos especificados en la norma (numerados como A1...AN).

1.3 - Especificaciones del sistema

Requisitos funcionales

Llamamos requisitos funcionales a aquellos que se deben cumplir para satisfacer las necesidades de los usuarios. Dichos requisitos serían:

- El usuario podrá conectarse a la plataforma mediante correo electrónico y contraseña y también mediante Gmail. Tanto el correo electrónico como el Gmail deberá estar registrado previamente en la base de datos que se está utilizando.
- Una vez que el usuario se haya conectado:
 - Podrá seleccionar correctamente los parámetros necesarios para la visualización de los datos en la gráfica. Los datos no se verán

reflejados en la gráfica hasta que no se haya seleccionado al menos un campo de cada opción del menú y un rango de fechas.

- Una vez que se hayan visualizado los datos de la gráfica, se podrá hacer zoom de los datos en distintas franjas de tiempo.
- No se podrán descargar los datos directamente desde la aplicación, sin embargo, cuando se ejecute el script para la carga de los datos en la base de datos, también se descargará un archivo JSON con todos los datos y dicho archivo se almacenará en Drive, pudiendo consultar los datos en cualquier momento.

Requisitos no funcionales

Llamamos requisitos no funcionales a aquellos que no es necesario cumplir estrictamente, pero ayudan a tener un mejor funcionamiento del sistema. Dichos requisitos serían:

- Una interfaz sencilla e intuitiva para que los usuarios no tengan ninguna complicación a la hora de usarla.
- La interfaz deberá ser responsive, es decir, que deberá adaptarse a las distintas resoluciones de las diferentes pantallas, ya sean ordenadores, tablets o móviles.
- El funcionamiento de la plataforma debe ser fluido a la hora de la realización de cada una de las funciones que haga internamente.

1.4 - Presupuesto

Es necesario la realización de un presupuesto estimado de los costes. Este presupuesto incluirá tanto el presupuesto del personal que realizará el trabajo como el presupuesto de los materiales que se utilizarán para la realización del proyecto. Aparte de esto, también se tendrá en cuenta las horas de trabajo del personal y el hardware y software que se necesita. El conjunto de todo esto nos dará una estimación final del coste del proyecto.

Para la parte de software solo será necesaria la adquisición licencias software

- Microsoft Windows 8.1: 0 € (Adquirido con la licencia de estudiante)
- GitHub: 0 € (Plataforma gratuita ilimitada de desarrollo colaborativo)
- Angular: 0 € (Código abierto)
- Visual Paradigm: 0 € (Adquirido con la licencia de estudiante)
- Balsamic Mockups 3: 0 € (Versión de prueba gratuita de 30 días)

Todo el software que se va a utilizar es gratuito, por lo que por esta parte no habrá ningún coste.

Se va a trabajar con Firebase. Esta plataforma incluye herramientas como el sistema de autenticación que será gratuito. Otras herramientas tendrán cuotas de pago que saltarán una vez se supere el límite del plan gratuito por día. Dentro de éstas últimas están incluídas Firestore, usado para almacenar la información de la aplicación y Storage, usado para almacenar las imágenes de los usuarios.

Productos	Plan Spark Límites generosos para aficionados Sin cargo	Plan Flame Precios fijos para apps en expansión USD 25 por mes	Plan Blaze Calcula los precios de las apps a gran escala. Pago por uso ✓ Se incluye el uso gratuito del plan Spark*
Cloud Firestore			
Datos almacenados	1 GiB en total	2.5 GiB en total	USD 0.18 por GiB
Network egress	10 GiB por mes	20 GiB por mes	Google Cloud pricing
Operaciones de escritura de documentos	20,000/día	100,000 por día	USD 0.18 por cada 100,000
Operaciones de lectura de documentos	50,000/día	250,000/día	USD 0.06 por cada 100,000
Operaciones de eliminación de documentos	20,000/día	100,000 por día	USD 0.02 por cada 100,000

Ilustración 1.7. Comparativa de planes de Cloud Firestore

En cuanto a la parte del hardware serán necesarios dos dispositivos: un ordenador portátil que será necesario para el desarrollo, tanto de la aplicación web como de la aplicación móvil posterior, y para probar la aplicación web y un móvil con un sistema operativo Android para poder probar la aplicación móvil.

El ordenador portátil está valorado en unos 700 € con una vida útil de 5 años. También habría que sumarle el ratón que está valorado aproximadamente en unos 15 € con una vida útil de 8 años y un disco duro Samsung 860 EVO 250 GB de unos 80 € con una vida útil de unos 2 años. El móvil está valorado en unos 350 € con una vida útil de 1 año. El total del hardware ascendería a unos 1145 € pero estaría totalmente amortizado a lo largo de la vida útil.

Este proyecto va a ser realizado por una única persona que se encargará del análisis, programación, testeo de la aplicación, etc. Por lo tanto, teniendo en cuenta lo que dice la web Indeed ⁷sobre que el sueldo medio bruto anual en España:

- Para un jefe de proyecto sería de 35.718 € al año. Entonces, estimando que la parte del análisis será aproximadamente unas 5 horas, el presupuesto sería de unos 20 €.
- Para un analista sería de 22.021 € al año. Entonces, estimando que la parte del análisis será aproximadamente unas 100 horas, el presupuesto sería de unos 251 €.
- Para un programador junior sería de 19.133 € al año. Entonces, estimando que la parte de la programación será aproximadamente unas 200 horas, el presupuesto sería de unos 436 €.
- Para un tester sería de unos 25.949 €. Entonces, estimando que la parte de testeo serán unas 10 horas, el presupuesto sería de unos 29 €

Para la puesta en marcha del proyecto será necesaria la adquisición de un plan de pago para el uso de Firestore, ya que el almacenamiento de los datos superará el límite del almacenamiento.

1.5 - A1: Documentación de entrada

En equivalencia al pliego de condiciones se incluye la documentación de entrada que se especifica en el Apéndice Guía original del TFT.

⁷ <https://www.indeed.es/>

1.6 - A2: Análisis y Diseño del sistema

Análisis

Antes de empezar a realizar la aplicación que se me ha pedido es necesario buscar información sobre otras plataformas que hacen esto, es decir, que analicen grandes cantidades de información y generen paneles de control con gráficos de tiempo, etc.

Una plataforma que hace esto podría ser Microsoft Power BI⁸. Se trata de una solución de análisis empresarial que permite visualizar los datos y compartir información con toda la organización, o insertarla en su aplicación o sitio web. Esta plataforma se conectará a cientos de orígenes de datos para después dar una visualización a sus datos con los paneles e informes dinámicos.

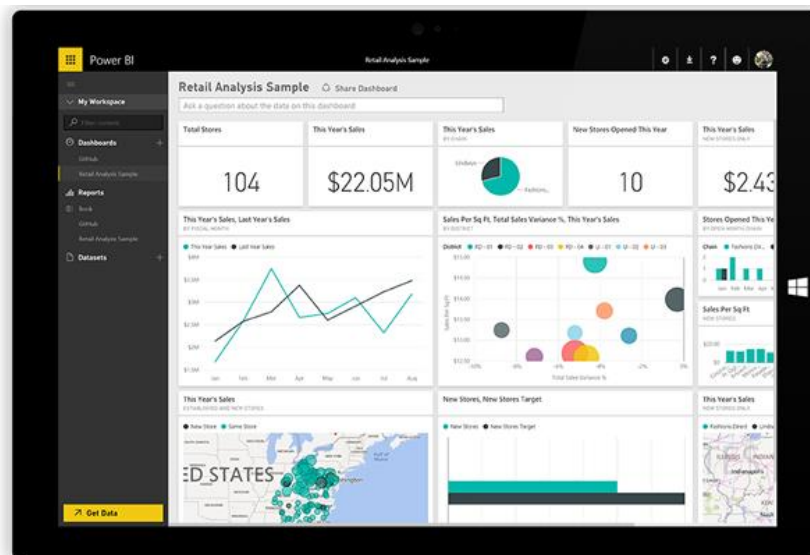


Ilustración 1.8. Ejemplo de plataforma con Power BI

Por otro lado, están Elasticsearch y Kibana⁹. Elasticsearch es un motor de analítica y análisis distribuido y open source para todos los tipos de datos, incluidos textuales, numéricos, geospaciales, estructurados y desestructurados. Conocido por sus API REST simples, naturaleza distribuida, velocidad y escalabilidad, Elasticsearch es el componente principal del Elastic Stack, un conjunto de herramientas open source para la ingesta, el enriquecimiento, el almacenamiento, el análisis y la visualización

⁸ <https://powerbi.microsoft.com/es-es/what-is-power-bi/>

⁹ <https://www.elastic.co/es/>

de datos. Elastic Stack ahora incluye una gran colección de agentes de envío conocidos como Beats para enviar los datos a Elasticsearch. Kibana es una aplicación de frontend open source que se encuentra sobre el Elastic Stack y proporciona capacidades de visualización de datos y de búsqueda para los datos indexados en Elasticsearch. Normalmente, funciona como la herramienta de representación para el Elastic Stack aunque también actúa como la interfaz de usuario para monitorear, administrar y asegurar un cluster del Elastic Stack; además de cómo concentrador centralizado de las soluciones integradas desarrolladas en el Elastic Stack.

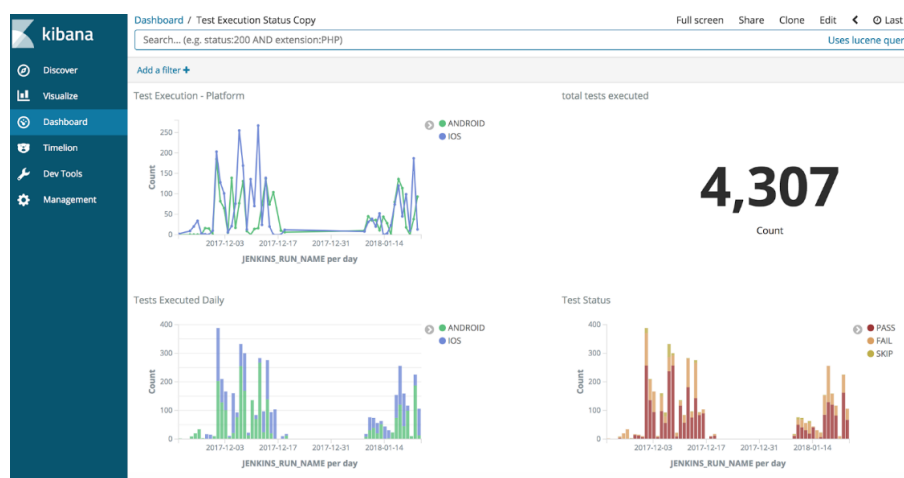


Ilustración 1.9. Ejemplo de plataforma ELK

Una vez que se ha obtenido la suficiente información de aplicaciones como estas, será el momento de analizar lo necesario para hacer la aplicación que se ha pedido.

El usuario debe marcar los filtros correspondientes para la correcta visualización de los datos. Este requisito es esencial para que se puedan visualizar los datos.

Para la correcta visualización de los datos es necesario procesar bien dichos datos. Para ello será necesario un script que se encargue de procesar los datos de la base de datos Realtime y convertirlos en el formato que tiene Firestore. El script también se encargará de comprimir los archivos JSON y subir dichos datos a una cuenta Drive. Cada cierto tiempo se borrará la base de datos, por la limitación de espacio.

Con respecto a los párrafos anteriores, el sistema se encargará de procesar los datos de la base de datos y generar la gráfica representativa.

La interfaz debe contener un menú que incluya los filtros (rango, dispositivos, unidad de medida). Estos filtros ya fueron descritos en el apartado de requisitos iniciales.

Diseño

Al igual que en los ejemplos que se vieron en el análisis, la aplicación que se ha pedido contará con un sistema de gestión de los usuarios (login, registro y perfil) como pantalla inicial. Una vez que se haya iniciado la sesión, habrá una pantalla principal que contará con la visualización gráfica de los datos y un menú a la izquierda que incorpore los filtros que haya que seleccionar.

El diseño que se ha pedido para esta aplicación será mucho más sencillo que los que se han visto anteriormente, por lo que se partirá desde cero para hacerla.

Más adelante se verá de forma más gráfica un diseño, del cual se partirá como base y al que se le podrán realizar posteriores cambios si es necesario.

1.6.1 - Metodología de desarrollo

Como metodología de desarrollo se ha usado una metodología ágil que está compuesta por:

- Primera iteración: búsqueda de información, análisis y planificación.
- Segunda iteración: diseño, implementación y pruebas.
- Tercera iteración: memoria.

1.7 - A3: Estimación del tamaño y esfuerzo

Debido a que este proyecto se trata de un TFG, no hay restricciones económicas. Sin embargo, sí que habrá restricciones de tiempo, ya que las horas que hay para realizarlo están limitadas. Por lo tanto, tamaño del proyecto estará también limitado a dicho tiempo.

En este caso, el esfuerzo de este proyecto sólo lo llevará a cabo el autor del proyecto.

2. REQUISITOS DEL PROYECTO

2.1 - Tecnologías utilizadas

- Lenguajes:
 - TypeScript: lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un conjunto de JavaScript y ECMAScript, que esencialmente añade tipos estáticos y objetos basados en clases.
 - HTML: lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet.
 - CSS: lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Se usa mayoritariamente para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML. Junto con HTML y JavaScript, CSS es una tecnología que se usa para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles.
 - Python: lenguaje de programación multiparadigma, es decir, que soporta orientación a objetos, programación imperativa y programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma. Posee una licencia de código abierto, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.
- Framework:
 - Angular 8: framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con

capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript.

Está basado en clases tipo "Componentes", cuyas propiedades son las usadas para hacer el binding de los datos. En dichas clases tenemos propiedades (variables) y métodos (funciones a llamar).

- Bootstrap: biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.
- Base de datos:
 - Realtime Firebase: base de datos en tiempo real, back-end que está organizada en forma de árbol JSON. Se proporciona a los desarrolladores de aplicaciones una API que permite que la información de las aplicaciones sea sincronizada y almacenada en la nube de Firebase. Es compatible la integración con aplicaciones Android, iOS, JavaScript, Java, Objective-C, Swift y Node.js. La base de datos es también accesible a través de una REST API e integración para varios sistemas de Javascript como AngularJS, React, Ember.js y Backbone.js. La REST API utiliza el protocolo SSE que es una API para crear conexiones de HTTP para recibir notificaciones push de un servidor.

Con esta base de datos se permite a los usuarios acceder a la información de sus datos desde cualquier dispositivo en tiempo real, compartiendo una instancia de Realtime Database, por lo que cada vez que un usuario realice una modificación en esta, se almacenará la

información en la nube y se notificará simultáneamente al resto de dispositivos.

Gracias a lo mencionado anteriormente, si un usuario realiza cambios y pierde a la vez su conexión a Internet, el SDK de la plataforma usa una caché local en el dispositivo donde guarda estos cambios, así, cuando se vuelva a tener conexión, automáticamente se sincronizan los datos locales.

- Firebase Storage: proporciona cargas y descargas seguras de los archivos para aplicaciones Firebase, sin importar la calidad de la red. El desarrollador lo puede utilizar para almacenar contenido audiovisual o cualquier otro contenido generado por el usuario.
- Firebase Cloud Firestore: servicio de almacenamiento de datos derivado de Google Cloud Platform, adaptado a la plataforma de Firebase. Al igual que Realtime Database, es una base de datos NoSQL, pero tiene varias diferencias. Está organizado mediante documentos agrupados en colecciones, y en ellos se pueden incluir tanto campos de diversos tipos (cadenas de texto, números, arrays, etc.) como otras subcolecciones.

No soporta las búsquedas de subcadenas del texto almacenado y tampoco puede filtrar las búsquedas con condiciones que impliquen más de un campo, si no es por búsquedas por el texto exacto.

- Plataformas:
 - Github: plataforma de desarrollo colaborativo que sirve para el alojamiento de proyectos utilizando un sistema de control de versiones llamado Git¹⁰. Utilizado principalmente para la creación de código fuente de programas de ordenador. El código de los proyectos se almacena típicamente de forma pública, aunque utilizando una cuenta de pago, también permite hospedar repositorios privados.

¹⁰ <https://git-scm.com/downloads>

2.2 - Diseño

2.2.1 - Diseño arquitectónico del sistema

Al igual que se mencionó en apartados anteriores, esta aplicación surge como una necesidad para mostrar de forma gráfica los valores reales de los datos energéticos obtenidos de casas y coches de usuarios, ya que no se estaban visualizando con claridad y un gráfico podría facilitar el estudio del conjunto de estos datos con mayor precisión. Por ello, el diseño está basado en una plataforma llamada Pecan Street | Dataport¹¹. Sin embargo, aunque nos basemos en dicha aplicación, en la plataforma que se va a realizar no se implementarán aquellas consultas a datos con los que no se trabaja, es decir, consultas que no sean de electricidad, ya que la plataforma que se toma de base trabaja con también con datos de agua y gas. Por lo tanto, el diseño se centraría en un diseño y funcionalidad más sencilla que la que incorpora la plataforma mencionada.

Para hacer la aplicación ha sido necesaria la división del código en clases las cuales tendrán una funcionalidad específica. Esta división se realizará para saber con mayor facilidad para lo que es y lo que hace cada clase.

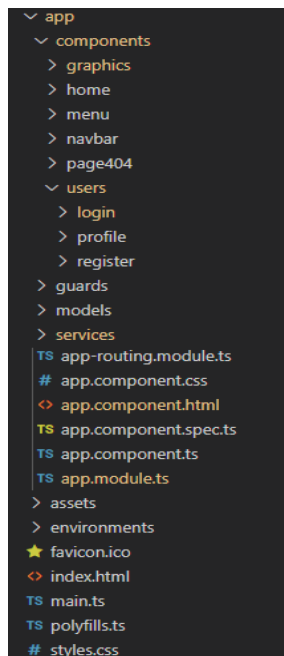


Ilustración 2.1. Clases de la aplicación

¹¹ <https://dataport.pecanstreet.org/>

Dicha clasificación se explicará con detalle más adelante en el apartado de implementación.

2.2.2 - Diagramas de clases

Un diagrama de clases es un tipo de diagrama con una estructura estática que describe cómo se organiza un sistema mediante las clases del sistema, atributos, funciones o métodos, y las relaciones entre los objetos.

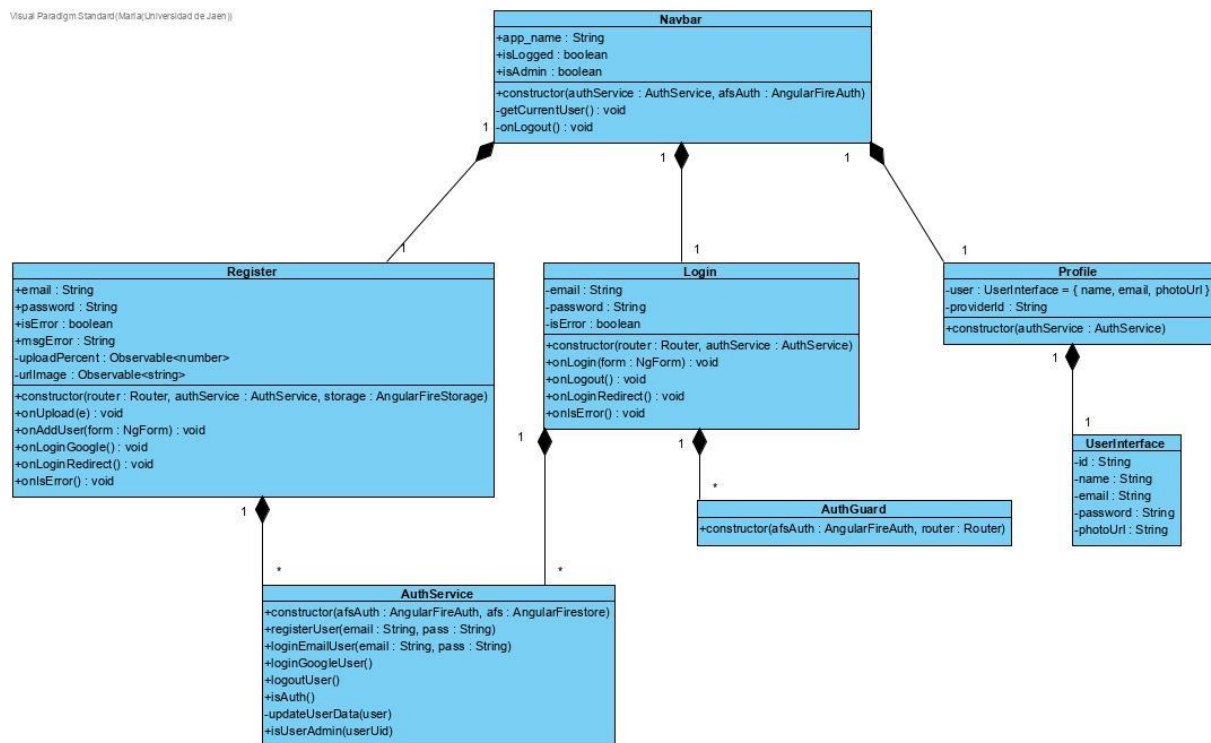


Ilustración 2.2. Diagrama de clases del sistema de gestión de usuarios.

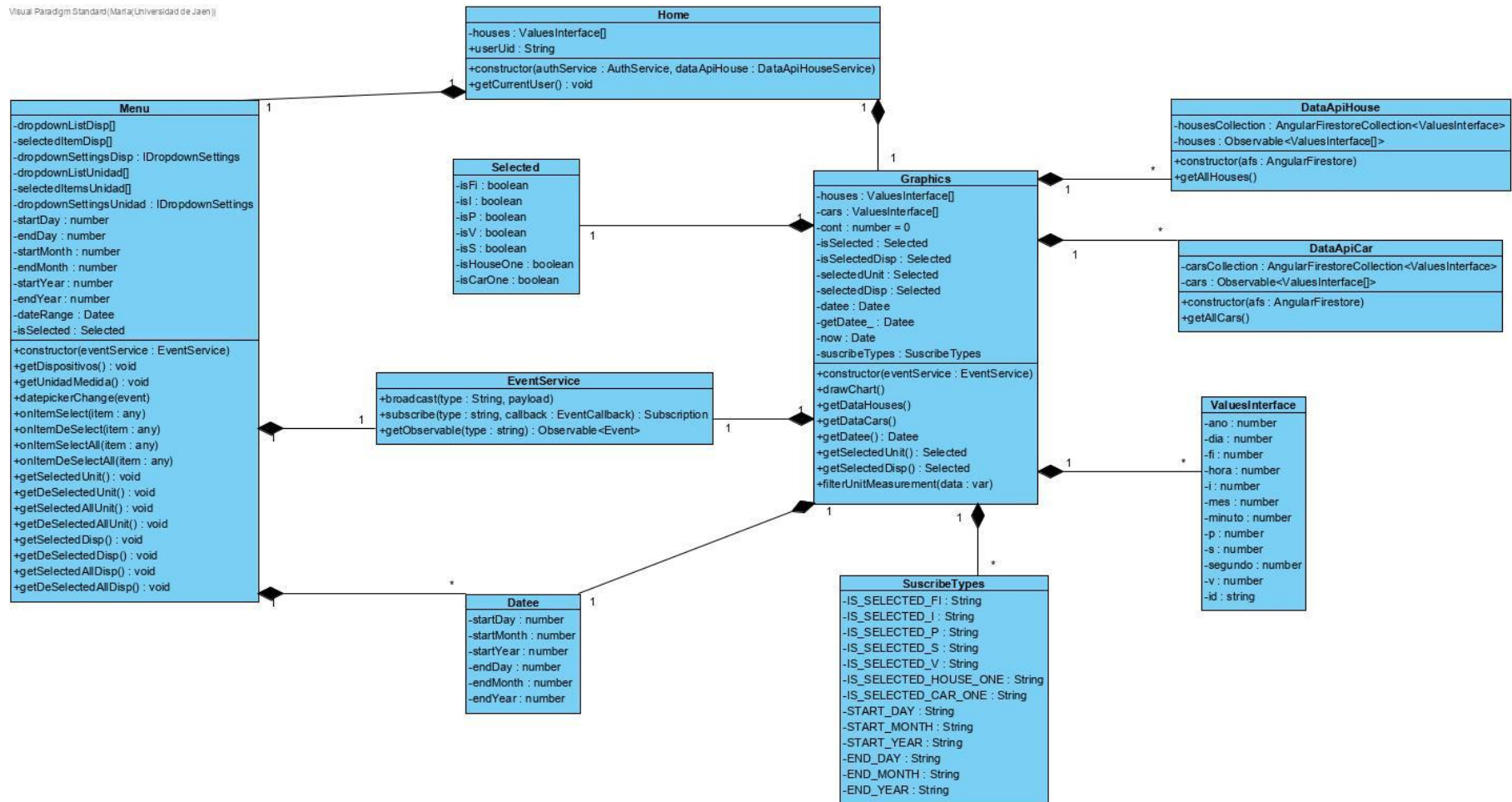


Ilustración 2.3. Diagrama de clases de la visualización de los datos.

Para realizar los diagramas se ha dado mayor relevancia a las clases que tienen mayor interés en el objetivo de esta aplicación. Las clases que no se han detallado son parte de la aplicación vacía que se genera al principio y, aunque son indispensables, no requieren tanto nivel de detalle como estas.

2.2.3 - Diagramas de casos de uso

Los diagramas de casos de uso son diagramas que se comportan como UML, pero más gráficos. Sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas.

Iniciar sesión (Usuario)

Caso de uso	Iniciar sesión
Descripción	El usuario introduce su Gmail para entrar con su perfil
Actores	Usuario, Sistema
Condición previa	Ninguna
Curso normal	1. Introducir Gmail 2. Pulsar el botón de Iniciar Sesión con Gmail
Curso alternativo	- El Gmail con el que se inicia la sesión es incorrecto. Por lo tanto hay que introducir uno correcto.
Condición de salida	El Gmail es correcto y se inicia correctamente la sesión.

Tabla 2.1. Caso de uso “Iniciar sesión (Usuario)”

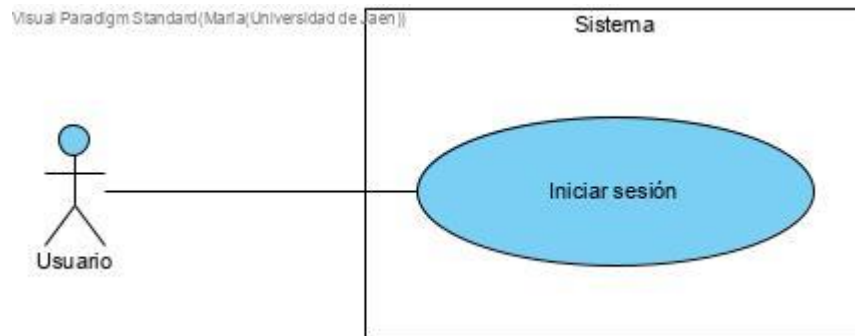


Ilustración 2.4. Caso de uso “Iniciar sesión (Usuario)”

Iniciar sesión (Administrador)

Caso de uso	Iniciar sesión
Descripción	El administrador introduce su correo y su contraseña para entrar con su perfil
Actores	Administrador, Sistema
Condición previa	Ninguna
Curso normal	1. Introducir correo 2. Introducir contraseña 3. Pulsar el botón de Iniciar Sesión
Curso alternativo	- No se introduce correo y/o contraseña. Hay que introducir ambos campos. - El correo y/o la contraseña son incorrectos. Volvemos a introducir el correo y/o la contraseña.
Condición de salida	El correo y la contraseña son correctas y se inicia correctamente la sesión.

Tabla 2.2. Caso de uso “Iniciar sesión (Administrador)”

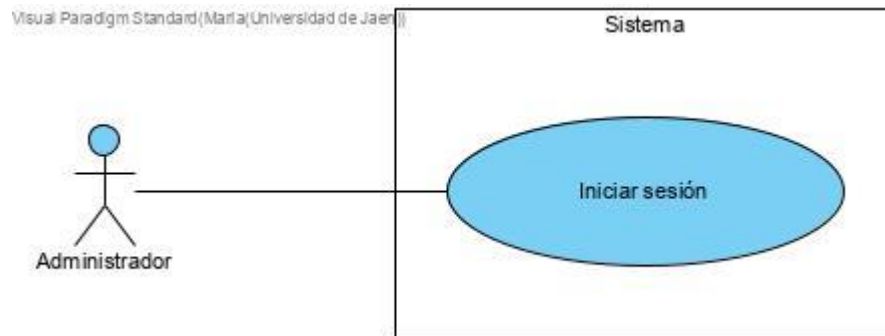


Ilustración 2.5. Caso de uso “Iniciar sesión (Administrador)”

Registrar usuario

Caso de uso	Registrar usuario
Descripción	El usuario introduce su correo y contraseña para que se registre en la base de datos.
Actores	Usuario, Administrador, Sistema
Condición previa	Que el usuario no esté registrado previamente
Curso normal	1. Introducir correo 2. Introducir contraseña 3.1. Pulsar el botón de Seleccionar archivo y seleccionar una imagen que será la imagen de perfil 3.1.1. Pulsar el botón de Registrar 3.2. Pulsar el botón de Registrar con Google para que el registro se haga directamente con el Gmail.
Curso alternativo	- No se escribe correo o contraseña válido. Hasta que no se escriban correctamente ambos, el usuario no se podrá registrar en el sistema.
Condición de salida	El correo y la contraseña son correctos o si se hace el registro con

	Gmail y es válido, el usuario se registra con éxito.
--	--

Tabla 2.3. Caso de uso “Registrar usuario”

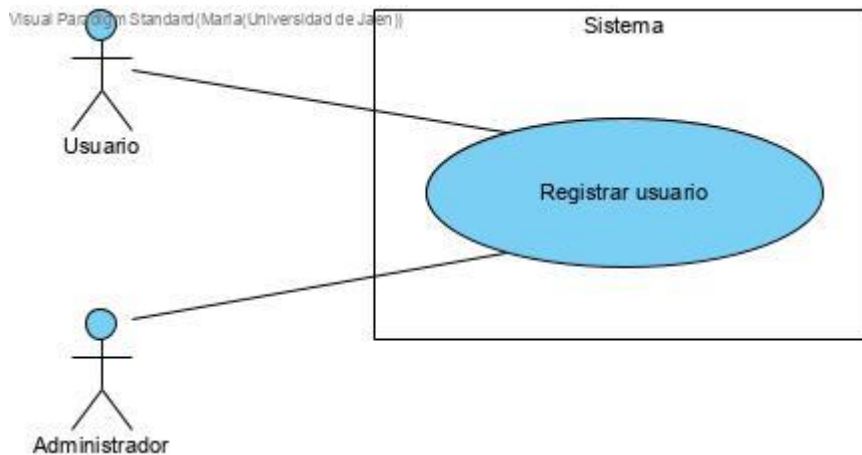


Ilustración 2.6. Caso de uso “Registrar usuario”

Cerrar sesión

Caso de uso	Elegir opciones gráfica
Descripción	El usuario cierra la sesión abierta
Actores	Usuario, Administrador, Sistema
Condición previa	Estar logueado previamente
Curso normal	1. Pulsar el botón de “cerrar sesión”
Curso alternativo	No hay curso alternativo
Condición de salida	Se pulsa el botón y se redirige a la página inicial de inicio de sesión

Tabla 2.4. Caso de uso “Cerrar sesión”

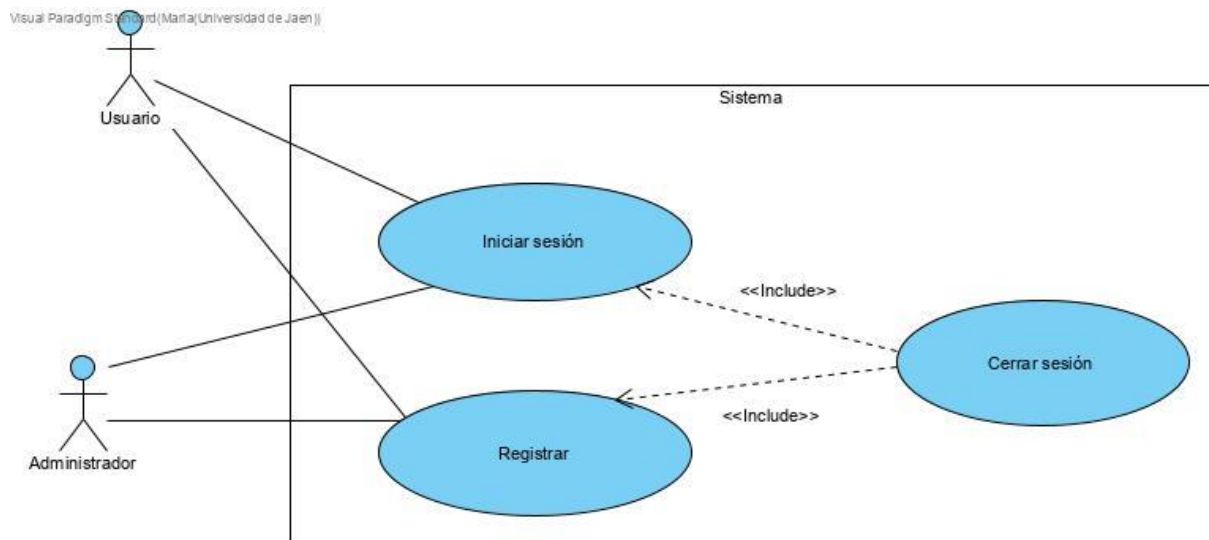


Ilustración 2.7. Caso de uso “Cerrar sesión”

Visualizar perfil

Caso de uso	Elegir opciones gráfica
Descripción	El usuario visualiza su perfil con sus datos
Actores	Usuario, Administrador, Sistema
Condición previa	Estar logueado previamente
Curso normal	1. Pulsar el botón de “perfil”
Curso alternativo	No hay curso alternativo
Condición de salida	Se pulsa el botón y se redirige a la página donde se encuentra la información del usuario

Tabla 2.5. Caso de uso “Visualizar perfil”

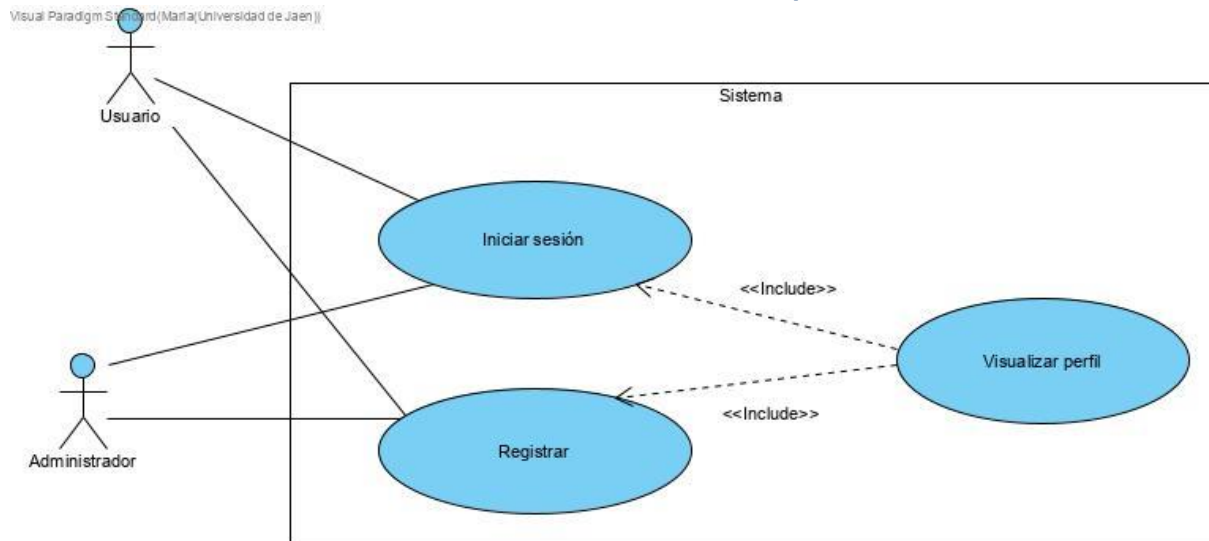


Ilustración 2.8. Caso de uso “Visualizar perfil”

Elegir opciones de la gráfica

Caso de uso	Elegir opciones gráfica
Descripción	El usuario marca las opciones que desee visualizar en la gráfica
Actores	Usuario, Administrador, Sistema
Condición previa	Estar logueado previamente
Curso normal	2. Seleccionar rango de fechas 3. Seleccionar los dispositivos 4. Seleccionar el tiempo 5. Seleccionar las unidades de medida
Curso alternativo	- No se marca al menos un valor de cada opción y/o el rango. Hay que marcarlos para que funcione la visualización.
Condición de salida	Se marcan todas las opciones y se visualiza correctamente.

Tabla 2.6. Caso de uso “Elegir opciones de la gráfica”

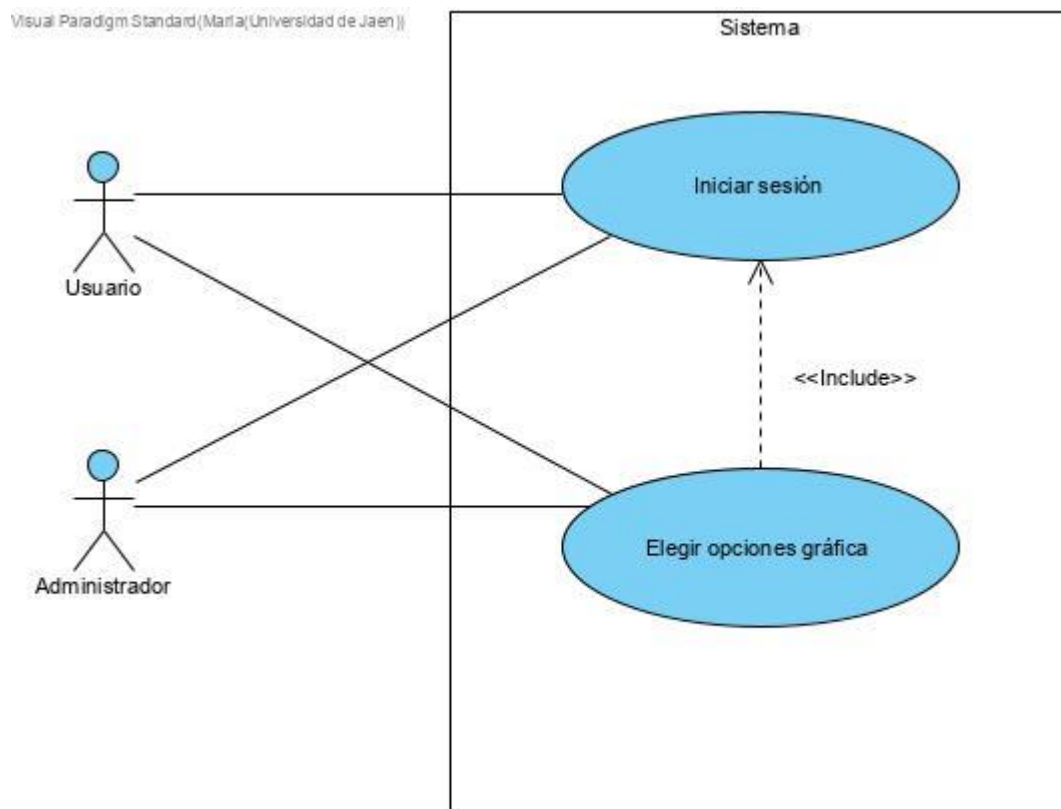


Ilustración 2.9. Caso de uso “Elegir opciones de la gráfica”

Hacer zoom

Caso de uso	Hacer zoom
Descripción	El usuario amplía la zona que desee visualizar en la gráfica
Actores	Usuario, Administrador, Sistema
Condición previa	Debe haber valores en la gráfica
Curso normal	1. Por defecto se muestra la gráfica con los valores de un día. 2. Seleccionar un botón entre los disponibles que hay en la parte superior izquierda de la gráfica, siempre y cuando sean valores menores del que

	haya seleccionado en ese momento.
Curso alternativo	No hay curso alternativo
Condición de salida	Se selecciona algún botón de zoom y se visualiza la franja de tiempo correspondiente a ese botón

Tabla 2.7. Caso de uso “Hacer zoom”

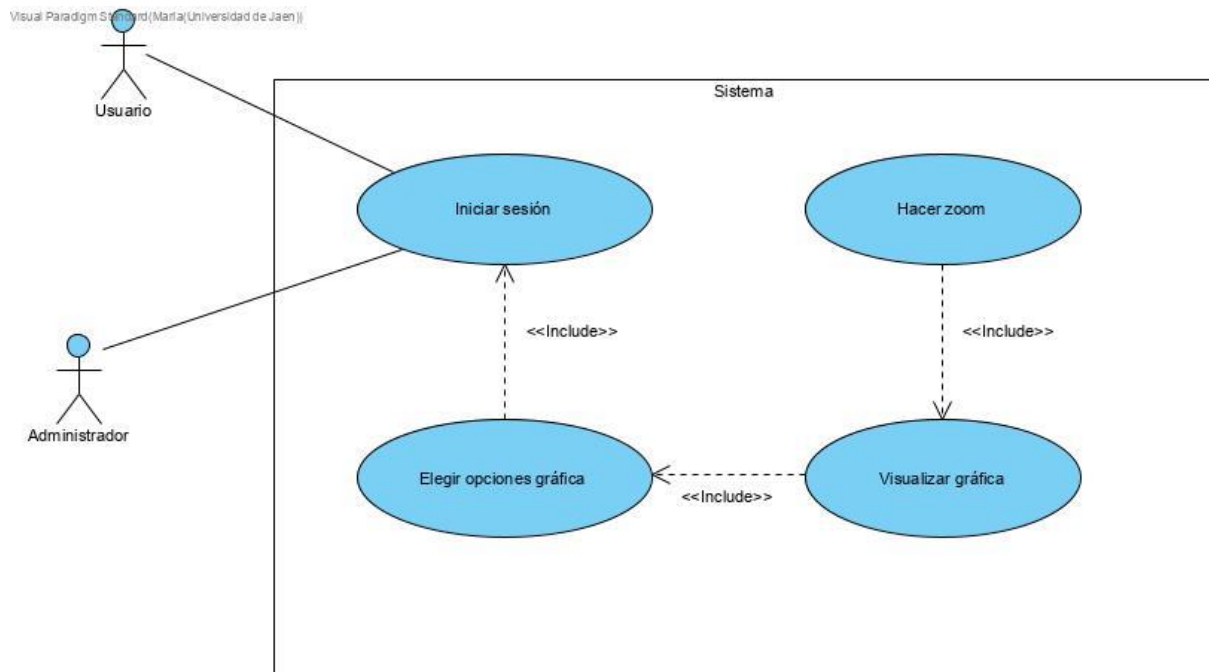


Ilustración 2.10. Caso de uso “Hacer zoom”

Quitar zoom

Caso de uso	Hacer zoom
Descripción	El usuario disminuye la zona que desee visualizar en la gráfica
Actores	Usuario, Administrador, Sistema
Condición previa	Debe haber valores en la gráfica
Curso normal	1. Seleccionar un botón entre los disponibles que hay en la parte superior izquierda de la

	gráfica, siempre y cuando sean valores mayores del que haya seleccionado en ese momento.
Curso alternativo	No hay curso alternativo
Condición de salida	Se selecciona algún botón de zoom y se visualiza la franja de tiempo correspondiente a ese botón

Tabla 2.8. Caso de uso “Quitar zoom”

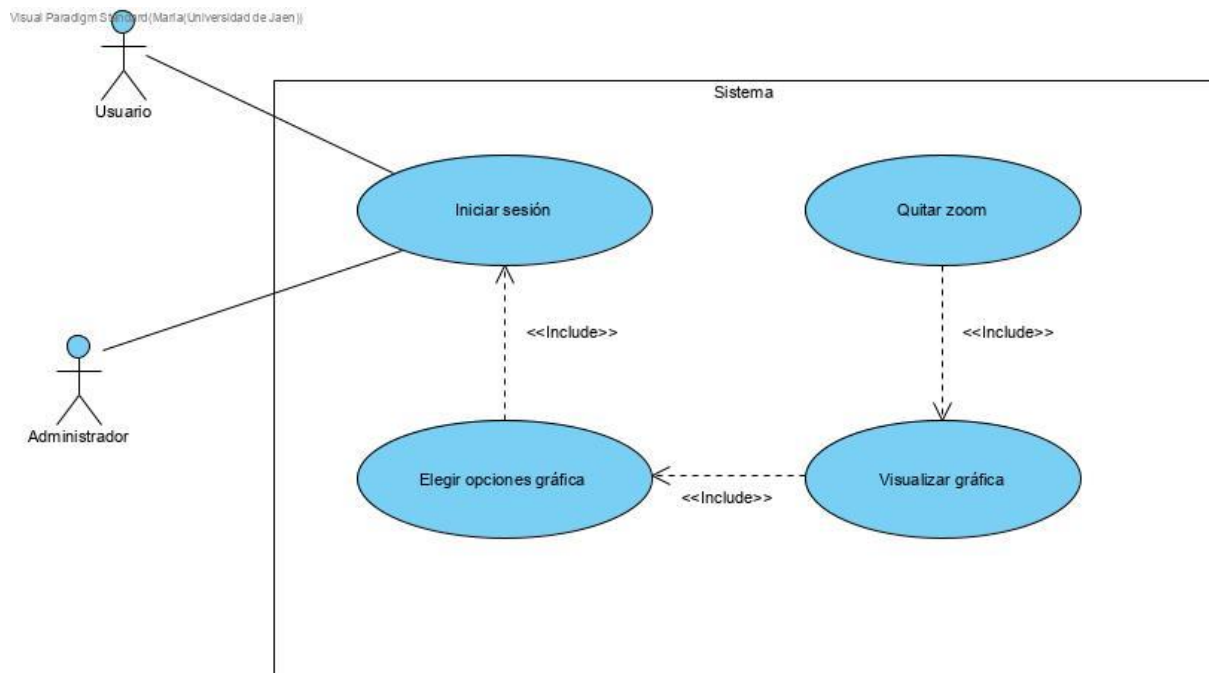


Ilustración 2.11. Caso de uso “Quitar zoom”

2.2.4 - Diagramas de secuencia

Un diagrama de secuencia es un tipo de diagrama que se usa para modelar la interacción que hay entre los objetos de un sistema según UML. Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.

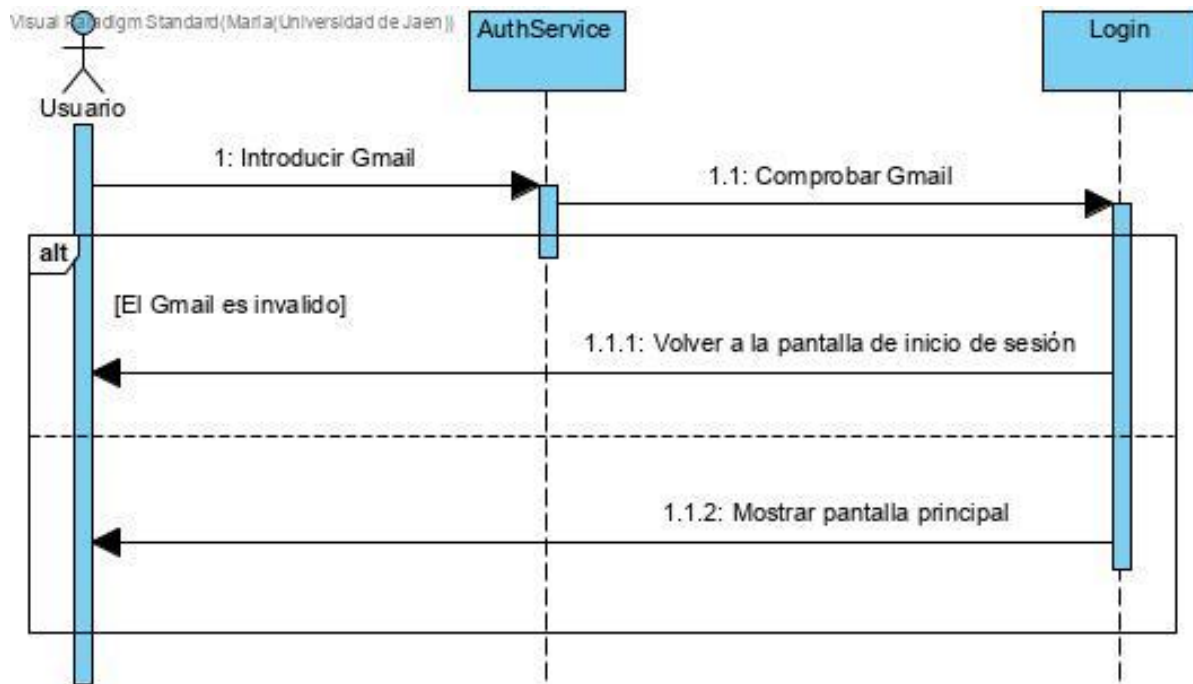


Ilustración 2.12. Diagrama de secuencia “Iniciar Sesión (Usuario)”

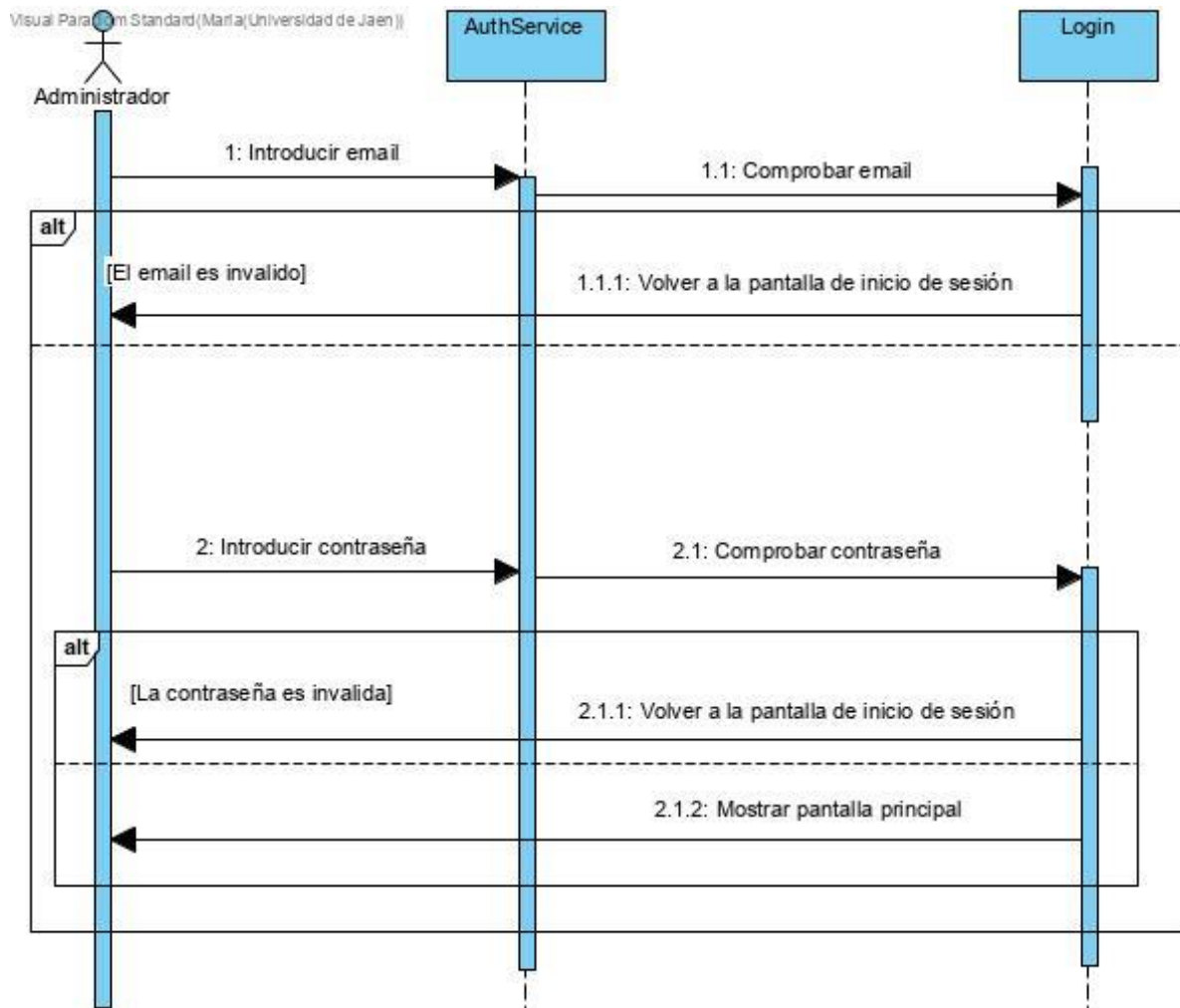


Ilustración 2.13. Diagrama de secuencia “Iniciar Sesión (Administrador)”

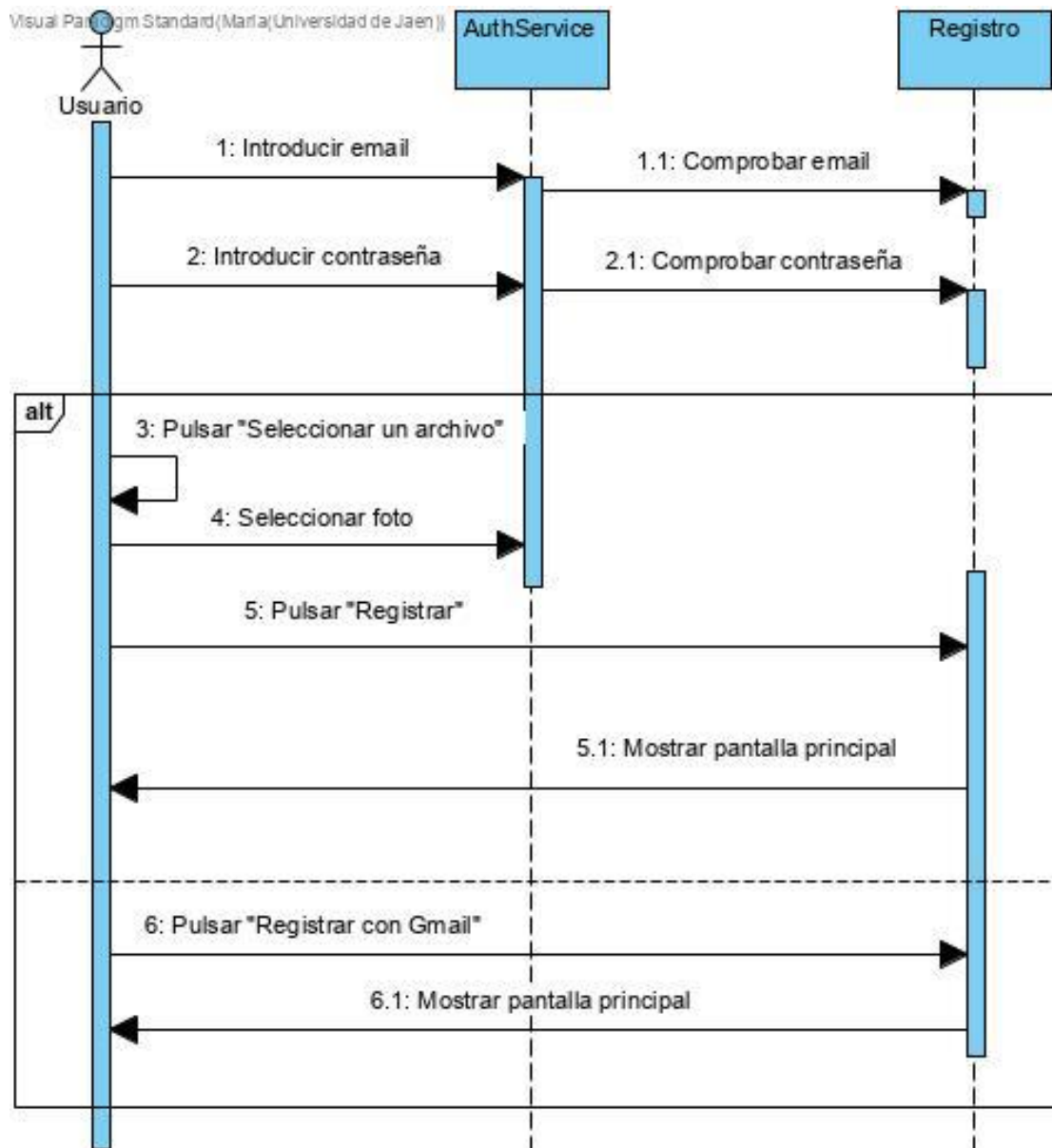


Ilustración 2.14. Diagrama de secuencia “Registrar”

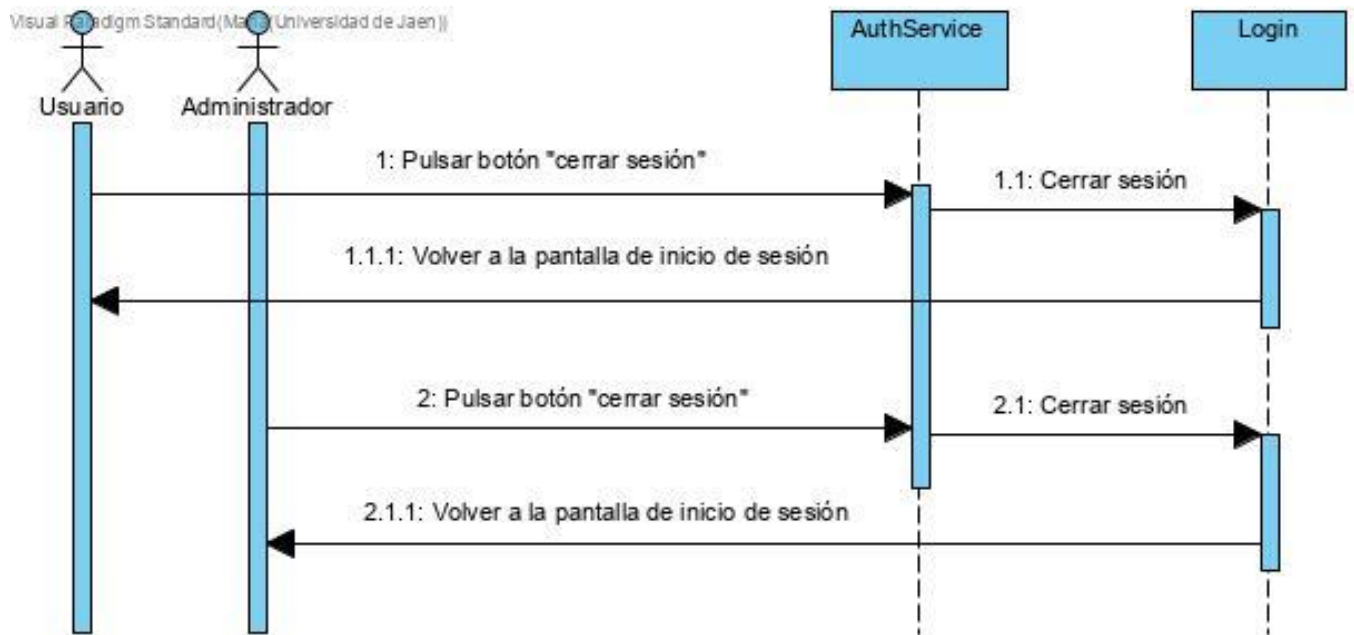


Ilustración 2.15. Diagrama de secuencia "Cerrar sesión"

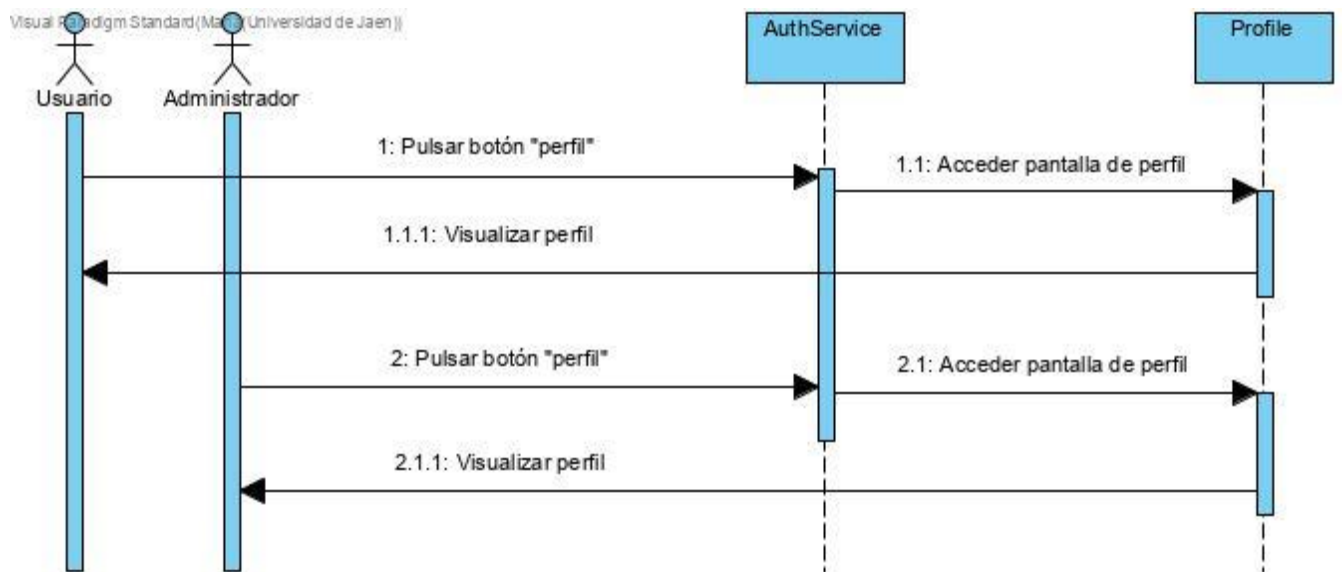


Ilustración 2.16. Diagrama de secuencia "Visualizar perfil"

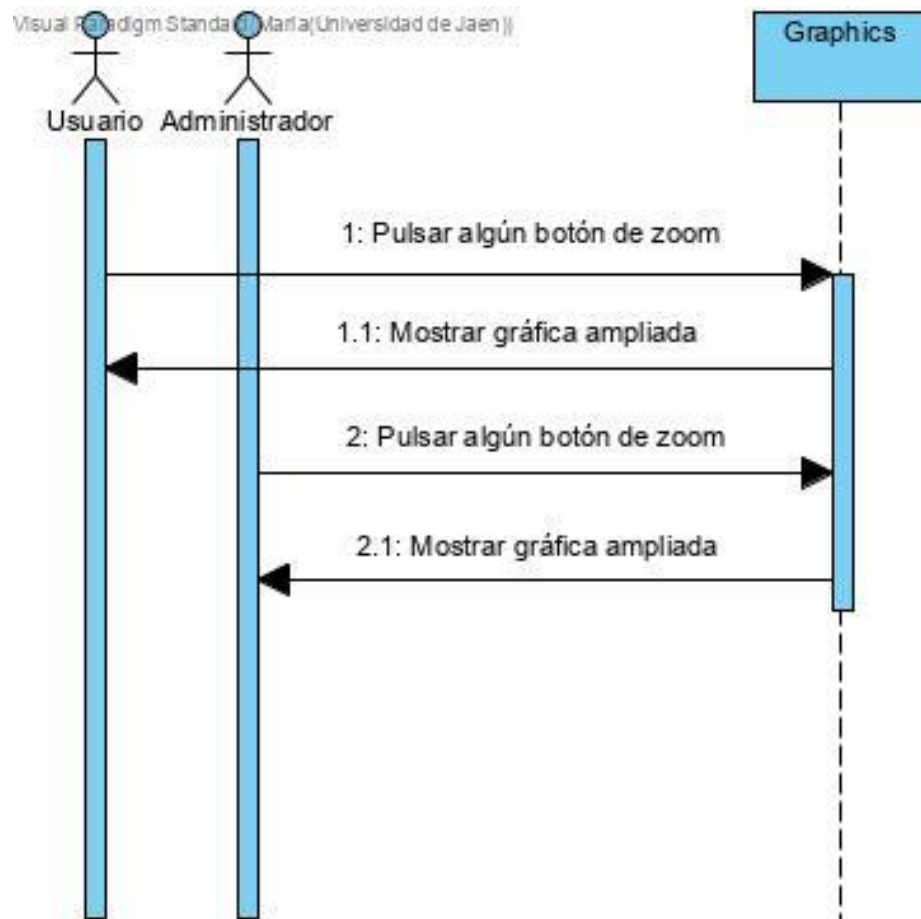


Ilustración 2.17. Diagrama de secuencia “Hacer Zoom”

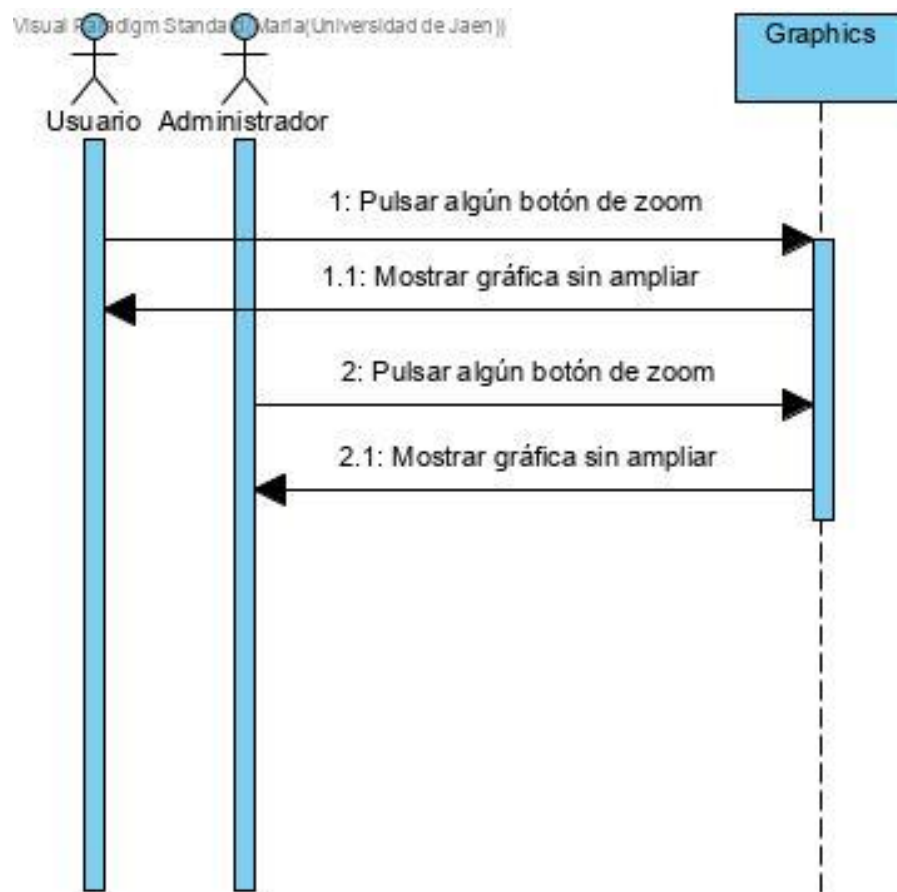


Ilustración 2.18. Diagrama de secuencia “Quitar Zoom”

2.2.5 - Diseño de la interfaz y storyboards

Un guion gráfico o storyboard es un conjunto de ilustraciones mostradas en secuencia con el objetivo de servir de guía para entender una historia. Se hace antes de empezar a programar la aplicación, ya que nos sirve de guía para saber cómo queremos que nos quede la posible interfaz.

La función que tiene principalmente es la de narrar la historia con pocas imágenes secuenciadas acompañadas de unos números que servirán de guía para saber a qué página vamos cuando realizamos una acción.

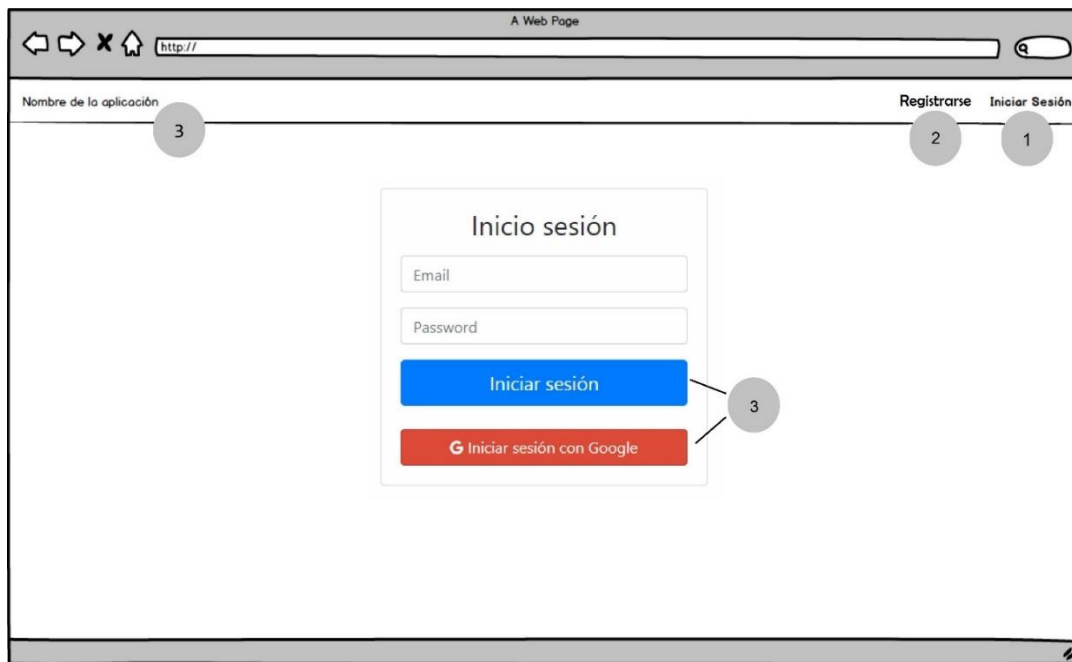


Ilustración 2.19. Storyboard. Pantalla 1.

La pantalla de login será la pantalla inicial, que permitirá conectarse a la plataforma mediante un email y contraseña o bien mediante Gmail. Para poder acceder a la visualización de los datos en la gráfica, previamente habrá que loguearse.

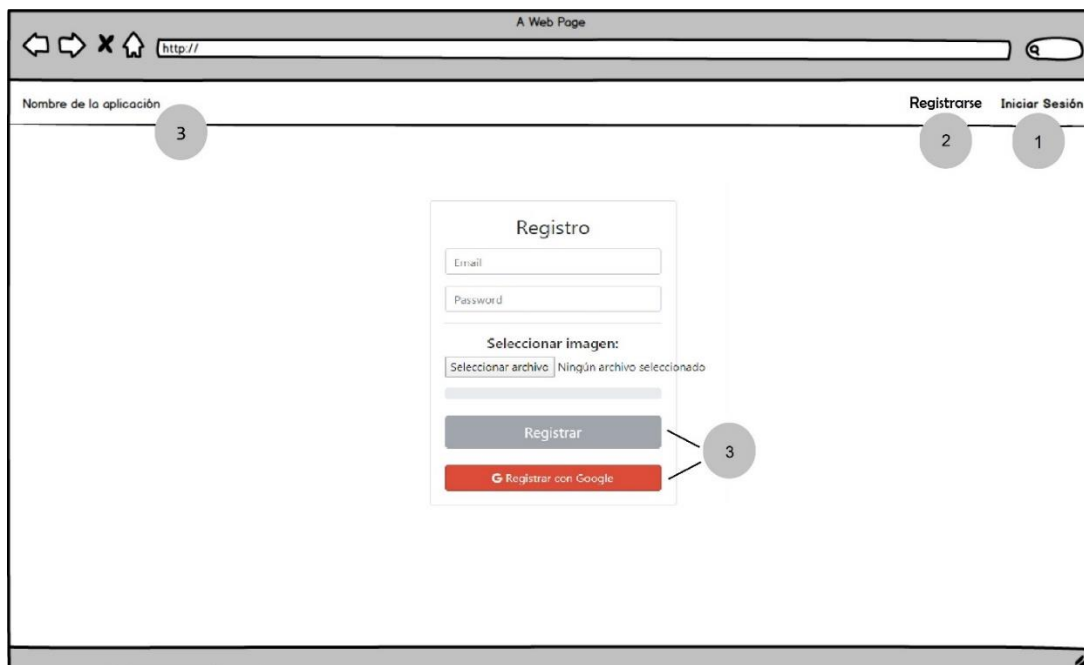


Ilustración 2.20. Storyboard. Pantalla 2.

Si el usuario es nuevo, tendrá que registrarse. Para ello tendrá que ingresar un email y contraseña válidos y seleccionar una imagen que será la que se muestra en el perfil. También tendrá la opción de hacer el registro con un Gmail ya existente.

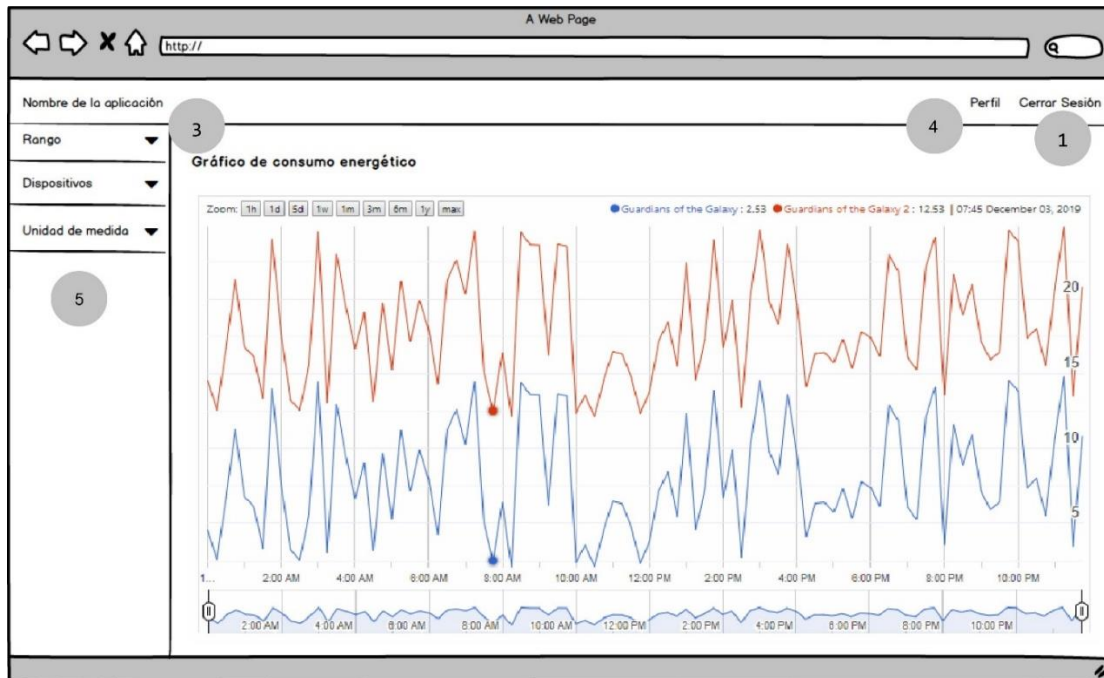
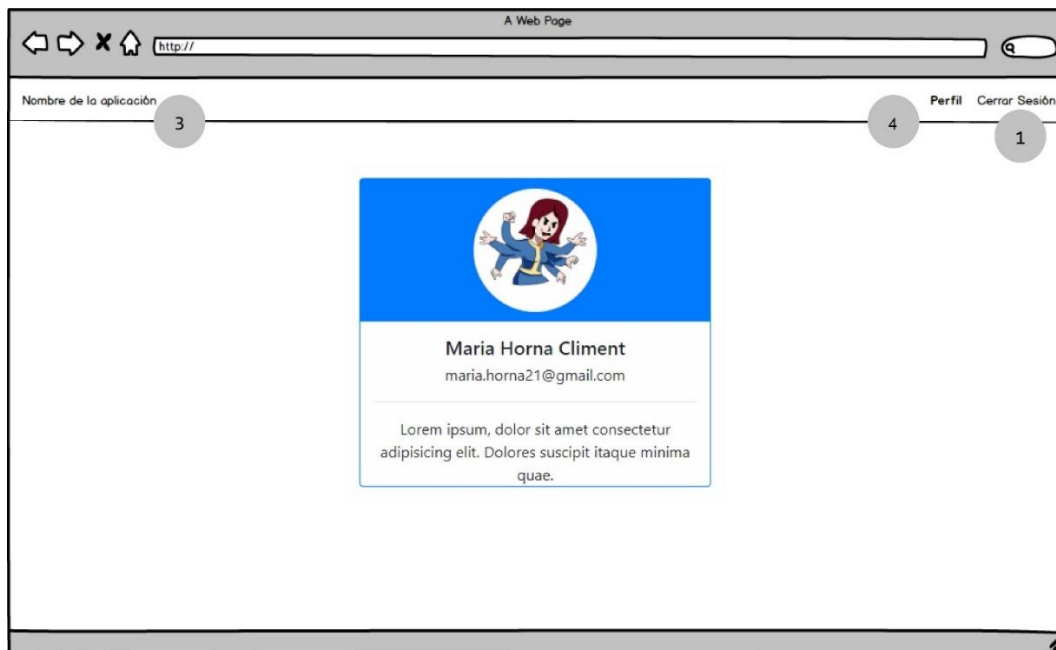
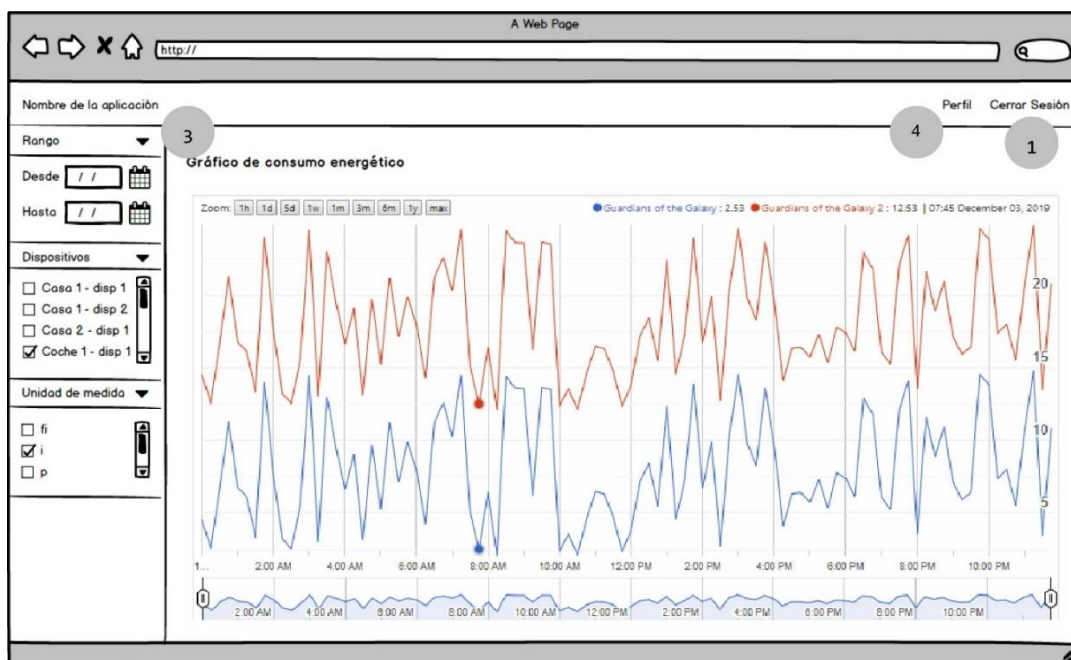


Ilustración 2.21. Storyboard. Pantalla 3.

Esta es la pantalla principal en la que aparecerá un menú desplegable a la izquierda y una gráfica en la que inicialmente se verán pintados unos datos iniciales. Estos datos que aparecerán inicialmente serán aquellos que sean más representativos.

**Ilustración 2.22. Storyboard. Pantalla 4.**

Una vez que el usuario se haya logueado podrá acceder a su perfil, pudiendo ver su foto, su nombre, su correo y una breve descripción. Por otro lado, el navbar habrá cambiado, dando la opción de cerrar la sesión. Si le damos a esta última opción volverá a la pantalla de login.

**Ilustración 2.23. Storyboard. Pantalla 5.**

En esta pantalla se muestra el menú desplegado con cada una de las opciones.

- En la primera opción se muestra el rango de fechas en la que se quieran visionar los datos en la gráfica.
- La segunda opción serán los diferentes dispositivos que tendrá cada casa. En esta opción cada usuario elegirá el dispositivo que quiera visionar.
- La tercera opción será cada una de las unidades de medida recogidas por los sensores.

Para que se puedan mostrar los datos en la gráfica tiene que haber una fecha de inicio y fin seleccionada y al menos un dispositivo y una unidad de medida seleccionada.

2.3 - Implementación

Como ya mencioné en anteriores apartados, esta aplicación principalmente está centrada en la visualización gráfica de los datos almacenados en la base de datos Firebase. Por lo que la aplicación está compuesta por dos partes, la primera parte sería la del procesamiento de los datos de la base de datos que se hará mediante scripts de Python y la segunda parte sería para la posterior visualización de dichos datos una vez que han sido procesados, que se implementará con Angular.

2.3.1 - Implementación de scripts en Python

Para subir datos a la base de datos hay que subirlos a Realtime Firebase y yo los datos los iba a coger de Firestore, por lo que era necesaria alguna forma de pasar los datos de un sitio a otro. Por lo tanto, tuve que hacer un script en Python.

Lo que hace este script es conectarse con el proyecto de Firebase, una vez conectado procesa los datos de la Realtime Firebase y los transforma para que se puedan almacenar en Firestore. Tuve que hacer esto ya que la organización que tiene Firestore es distinta a la de Realtime. Por lo tanto, se crea una colección con el nombre que se elija, y cada id que contenía el archivo json almacenado en la Realtime se convertirá en un documento con sus correspondientes datos. Esto lo hace con cada uno de ellos. Una vez que se terminan de transformar los datos, se almacenan los datos en un archivo JSON, se comprimen todos los archivos que tengan la terminación JSON y se envían a una cuenta Drive a la que previamente

hayamos tenido que dar unos permisos para poder hacer la conexión con Drive. Por último, subimos los datos a Firestore y así tendremos que hacerlo cada vez que queramos pasar los datos desde Realtime a Firestore.

2.3.2 - Implementación de la aplicación en Angular

En primer lugar, comencé buscando algunas plantillas que tuvieran gráficas parecidas a lo que me pidieron para partir de una base. Sin embargo, dichas plantillas eran demasiado complejas y contenían muchas cosas que eran totalmente innecesarias para lo que se me estaba pidiendo, por lo que decidí empezar la aplicación desde cero.

Primera parte – Conexión de la base de datos con la aplicación

Para esto, es necesario crear un nuevo proyecto en Firebase y asociarlo con la aplicación. Una vez que se haya hecho esto, se proporcionará un fragmento de código que habrá que poner en la clase `environment.ts` y otro fragmento que habrá que poner en la clase `app.module.ts` para inicializar la app y poder conectarlo correctamente con Firestore.

Segunda parte – Inicio de sesión

Una vez que pude conectar correctamente la base de datos con la aplicación, hice el sistema de login. Para ello cree una clase `login` que contuviera las funciones necesarias para poder conectarme con un email y contraseña, otra función para conectarme desde Gmail, y otra función para cerrar la sesión. Cada una de estas funciones incluían a su vez otra función que servía para que cada vez que iniciamos la sesión, la aplicación nos lleve directamente a la página principal. Si algo falla en el sistema de login, nos saltará un error que nos avisa de que algo anda mal.

Para la parte del inicio de sesión es necesaria la activación de algunos proveedores en la parte de autenticación de Firebase. Por lo que he tenido que activar el inicio de sesión con email y con Gmail.

Tercera parte – Registro de usuario

Después, hice el sistema de registro. Para ello, cree una clase llamada `register` y le incluí funciones para que se pudiera registrar mediante email y contraseña o mediante Gmail (al igual que con el sistema de login).

Sin embargo, pensé en incluir una página para el perfil, por lo que a la parte del registro le añadí para que cuando nos registramos con el email y la contraseña, incluyera una parte para añadir una fotografía. Para esta parte era necesario conectarnos con una parte de Firebase que se llama Firebase Storage. Esta parte sirve para incluir imágenes entre otras cosas. Para conectar con esta parte es necesario añadir otro fragmento de código en `environment.ts` y otro fragmento en `app.module.ts`. Una vez que conseguí conectarlo también con esta parte, añadí una función para que cada vez que le diéramos al botón de registrarnos, se añadiera el usuario a la base de datos de Firebase, en el apartado de los usuarios. Cuando se añade tanto el email como el Gmail a la autenticación de Firebase, se añaden con un id único. Si el usuario existe en este sistema de autenticación, no se podrá volver a añadir en el sistema.

Cuarta parte – Perfil de usuario

Luego, hice una pantalla para el perfil, por lo que cree una `case profile`. En esta clase incluí una función para que una vez que el usuario se haya logueado con éxito, este pueda ver su foto de perfil, su nombre y su correo y una breve descripción.

Quinta parte – Navbar

En esta parte se incluiría el nombre de la aplicación, el login, el registro y el perfil. Por lo que cada vez que se pincha en alguna de estas opciones llevaría a la correspondiente pestaña. Sin embargo, había que poner restricciones. Lo primero fue que la pantalla principal fuera la de login y que, si pulsabas en el nombre la aplicación sin estar logueado, volviera a la pantalla del login. También, mientras no estuvieras logueado, en el navbar solo aparecerá la opción de pulsar en el nombre de la aplicación, la opción de login y la opción de registro, pero no aparecería la de visualizar el perfil del usuario. Una vez que el usuario se ha logueado o registrado, el navbar cambiará y cuando se pulse en la opción del nombre de la aplicación, esta vez llevará a la pantalla principal que será la que incluirá la gráfica y ya no aparecerán las opciones de login o registro, sino que aparecerán las opciones de perfil y cerrar sesión. Esta parte se ha hecho así para que si no estás logueado no puedas acceder a ciertas opciones.

Sexta parte – Retoques visuales al sistema de gestión

Modifiqué los .css y mediante esto y bootstrap conseguí que la interfaz se viera mejor y que fuera responsive. Esto último, significa que cuando la página se decida abrir desde un móvil o desde una tablet, no se descoloquen las opciones, sino que se adapte a la pantalla.

Septima parte – Pantalla principal y menú de selección

Seguidamente cree una clase home que sería la que contendría los elementos que se visualizarán en la pantalla principal, es decir, la pantalla que incluiría la gráfica y su correspondiente menú. Al principio, puse un ejemplo para ver si se mostraba bien, sin embargo, había un problema y es que inicialmente me mostraba esta página sin haberme logueado. Esto era así porque en la clase app-routing.module.ts hay que poner las rutas correspondientes a cada pestaña de forma ordenada.

Una vez que hice esto, ya se visualizaban correctamente todas las pestañas por lo que pasé a obtener los datos de la base de datos.

Cree tres clases, la clase services, la clase menu y la clase graphics.

Dentro de la clase services se creará un servicio para cada dispositivo que se vaya a hacer, por lo que cree un servicio para la casa 1 y un servicio para el coche 1. Cada servicio tiene una función que se encargará de coger la colección correspondiente para posteriormente poder acceder a cada uno de los atributos que tenga cada documento de la colección.

Dentro de la clase menu se hará un menú desplegable con cada uno de los filtros necesarios para aplicarlos en la gráfica. En primer lugar, tendremos el filtro de la fecha que se realizará con un datepicker y que en tiempo real irá almacenando el día, el mes y el año, tanto inicial como final. También tendremos el filtro para los dispositivos en el que cada vez que se encuentre pulsado un dispositivo se pondrá a true, en señal de que está siendo pulsado y si se deselectiona se pondrá a false. Por último, haremos el filtro de las unidades de medida que se hará de igual forma que el filtro de los dispositivos.

Mandaremos en tiempo real si están seleccionados o no a la clase graphics.module.ts. Esta clase contendrá la gráfica en la que vamos a visualizar los datos. La forma en la que he realizado la gráfica es añadiendo todas las columnas,

que serán cada uno de los dispositivos con su correspondiente unidad de medida y todas las filas, que serán los valores de los mencionados anteriormente.

En el caso de que no esté seleccionado un dispositivo o una unidad de medida, se eliminará la correspondiente columna y la correspondiente fila. Esta selección y desección de filtros se hará con unos condicionales para saber cuándo eliminar las columnas y las filas.

En el caso de que no se seleccione ningún filtro, la gráfica tendrá valores nulos para que no de error en la gráfica.

2.3.3 - Conclusiones finales

Por un lado, el script que había que realizar en Python permite:

- Transformar los datos JSON de la base de datos Realtime en datos para Firestore.
- Exportar esos datos a una carpeta.
- Importar la carpeta a una cuenta Drive.

Cuando se ha terminado el desarrollo de la aplicación, esta permite:

- Iniciar sesión mediante email y contraseña para el administrador.
- Iniciar sesión mediante Gmail para los usuarios.
- Registro mediante email y contraseña o bien mediante Gmail.
- Visualización del perfil una vez que se haya iniciado la sesión.
- Selección del filtro de “Rango” donde se podrán seleccionar las fechas en las que queremos que estén comprendidos los datos.
- Selección del filtro de “Dispositivos”.
- Selección del filtro de “Unidad de medida”.
- Visualización de los datos en la gráfica siempre y cuando se hayan seleccionado los tres filtros.
- Visualización vacía en el caso de no seleccionar algún filtro.
- Hacer zoom de los datos de la gráfica
- Disminuir el zoom de los datos de la gráfica.

Por lo tanto, se puede concluir que se han completado los objetivos funcionales de la aplicación.

2.4 - Pruebas finales

2.4.1 - Pruebas de verificación del script Python

Se ha empezado probando el script de Python, para ello se ha importado un archivo JSON en la base de datos Realtime y se ha ejecutado el script para comprobar si hacía bien todo.

Después de realizar la primera prueba se detectó que se producía un error en la subida de los datos a Firestore. Esto ocurría porque en el plan gratuito de Firebase, que es el que se está usando, solo son posibles 20,000 operaciones de escritura y 50,000 operaciones de lectura al día.

Para intentar solucionar esta situación sin contratar un sistema de pago, se modificó el script para que solo se exportaran en Firestore datos de 10000 en 10000. Tras realizar la segunda prueba se comprobó que tampoco era posible esta solución ya que, aunque solo se escriban esos datos, tiene que seguir haciendo la lectura de todos los datos.

Como solución final, se ha optado por escribir solamente unos 10 datos, para comprobar que efectivamente el script hacía lo que se pedía.

Por lo tanto, el script lee los datos desde Realtime y los sube a Firestore sin problema. También se genera un archivo JSON, se comprime y se sube a una cuenta Drive y se hace sin ningún problema.

2.4.2 - Pruebas de verificación de la aplicación Angular

En primer lugar, se ha empezado probando el sistema de gestión de usuarios. El usuario podía iniciar sesión correctamente mediante el email y la contraseña, sin embargo, si no se escribía o no se escribía bien alguno de los dos campos no aparecía ningún aviso, por lo que se tuvo que corregir esto. Del mismo modo, para el registro también se tuvo que incluir un aviso por si no se escribía alguno de los campos. Después de registrar a varios usuarios ficticios tanto con Gmail como con email y contraseña, se probó a iniciar sesión y se ingresó a la página principal sin problema.

En segundo lugar, se probó el funcionamiento de la gráfica cuando se aplicaban los distintos filtros. La primera prueba fue solo con el filtro de unidades de medida. En esta prueba se detectó un error en el que si no se seleccionaba ninguna opción de

este filtro aparecía un mensaje en el que decía que la gráfica debía tener al menos una opción seleccionada. Para solucionar este error, se implementó que cada vez que no hubiera ninguna opción seleccionada, apareciera una gráfica con datos vacíos.

La segunda prueba fue para el filtro de los dispositivos y tras realizar dicha prueba no se detectó ningún error.

La tercera prueba fue para el filtro del rango, este filtro tampoco presentó ningún error, sin embargo, cuando se entra por primera vez a la página principal, el calendario que aparece en este filtro tiene seleccionado la fecha actual en la que estamos y en la gráfica no se mostraba bien la fecha, sino que cogía la primera fecha que hubiera disponible en el sistema. Para solucionar esto se puso que, por defecto, la fecha inicial y final que se coge sería la actual. Después se comprobó que el filtro estaba correcto y funcionaba como debía.

La cuarta prueba que se hizo fue para comprobar que funcionaba bien la gráfica cuando se combinaban los distintos filtros.

Una vez que se hicieron todas estas pruebas, se pudo concluir que el sistema de gestión de usuarios funciona y redirige correctamente a la página principal donde se encuentra la gráfica y que los filtros hacen su función en la gráfica.

2.4.3 - Pruebas de rendimiento del sistema

El primer requisito funcional que tenía que cumplir el sistema era la usabilidad, de esta forma la plataforma web es sencilla de usar incluyendo textos fáciles y legibles, proporcionando la información en tiempos que sean imperceptibles para el usuario y con menús sencillos. Por lo tanto, una vez que se ha cumplido esta condición, el usuario encontrará satisfechas sus consultas y se sentirá cómodo con el uso de dicha plataforma. En la aplicación se visualiza la información que desea el usuario en cada momento, por lo que es intuitiva y sencilla.

Como se ha mencionado antes, la usabilidad también necesita de tiempos de respuesta cortos e imperceptibles para los usuarios. Por lo tanto, cuando se produzcan cambios en las vistas de la plataforma o cuando se seleccionen las distintas opciones del menú, los resultados de dichas acciones se verán reflejados de forma inmediata ya que, si esto no ocurriera así, el usuario se cansaría de esperar.

En conclusión, un usuario medio que no haya utilizado este tipo de aplicaciones podría aprender rápidamente el funcionamiento de esta aplicación y no tendría problema en poder utilizarla en un periodo corto de tiempo.

2.5 - Demostración de la aplicación Angular desarrollada

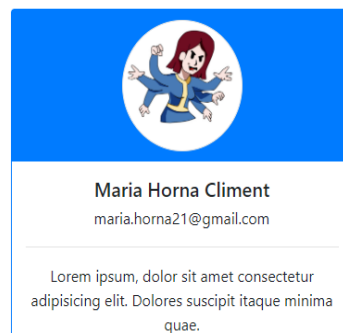
Estos serían los resultados obtenidos de la aplicación:

The screenshot shows the login page of the ElectricUJA application. At the top, there is a blue header bar with the text "ElectricUJA" on the left and "Registro Login" on the right. The main content area is white and contains a centered login form. The form has a title "Inicio sesión". Below the title are two input fields: "Email" and "Password". Below these fields is a blue button labeled "Iniciar sesión". At the bottom of the form is a red button labeled "G Iniciar sesión con Google".

Ilustración 2.24. Pantalla de inicio de sesión

The screenshot shows the registration page of the ElectricUJA application. At the top, there is a blue header bar with the text "ElectricUJA" on the left and "Registro Login" on the right. The main content area is white and contains a centered registration form. The form has a title "Registro". Below the title are two input fields: "Email" and "Password". Below these fields is a section titled "Seleccionar imagen:". Under this section is a button labeled "Seleccionar archivo" and the text "Ningún archivo seleccionado". Below this is a grey button labeled "Registrar". At the bottom of the form is a red button labeled "G Registrar con Google".

Ilustración 2.25. Pantalla de registro

**Ilustración 2.26. Pantalla del perfil del usuario**

En la siguiente imagen se muestra la pantalla principal en la que se encuentra la gráfica para la visualización de los datos procedentes de la base de datos. Estos datos que se muestran están restringidos a los filtros que se encuentran seleccionados en el menú que aparece a la izquierda. En el primer filtro se ha seleccionado esa fecha como ejemplo, pero se podrá seleccionar la fecha que se desee mientras haya datos en la fecha seleccionada. Los últimos filtros son un ejemplo, sin embargo, se pueden combinar de la forma que se desee.

**Ilustración 2.27. Ejemplo de visualización de datos en la gráfica**

Esta imagen muestra cómo se visualizaría la gráfica en el caso de que no se seleccione ningún filtro.



Ilustración 2.28. Visualización de la gráfica sin datos

3 - CONCLUSIONES Y MEJORES FUTURAS

Para concluir con los apartados anteriores, se puede decir que la aplicación y los scripts han conseguido lograr los objetivos principales que se pidieron.

Los objetivos que se han conseguido superar son:

- El estudio y desarrollo de una plataforma que sirva para la visualización de los datos obtenidos de diferentes viviendas, de la fotovoltaica y de coches eléctricos.
- Posterior visualización de los datos en dicha plataforma web para que puedan ser de utilidad para los usuarios finales y para la toma de decisiones.
- Sistema de gestión de usuarios para que las personas autorizadas puedan entrar a la visualización de los datos.
- Realización de scripts que faciliten la lectura/escritura de los datos de la base de datos, ya que estos datos serán los necesarios para conseguir la visualización final.

Por lo tanto, esta aplicación aportaría una ayuda extra para el Equipo de Investigación de Energías Renovables.

En el futuro se podrían incluir mejoras al sistema, como serían:

- Inclusión de más gráficas, aparte de la principal que ayuden a una mejor visualización de los datos.
- Una opción para la exportación de los datos en formato JSON o CSV.
- Roles de usuario, es decir, administrador y usuario. De esta forma, el administrador sería el único que controlara el registro de los usuarios y podría ver opciones que los usuarios comunes no pudieran.

Debido a la limitación de tiempo, la plataforma ha quedado simple en cuanto a contenido, pero cumple con los objetivos principales que se pidieron en un principio y al ser así de sencilla, no debería de presentar mayor complicación a los usuarios. Aun así, se podrían realizar distintas pruebas a los usuarios para ver cómo interactúan con la aplicación y así poder ver donde podrían tener dudas o cometer errores para poder solucionarlos.

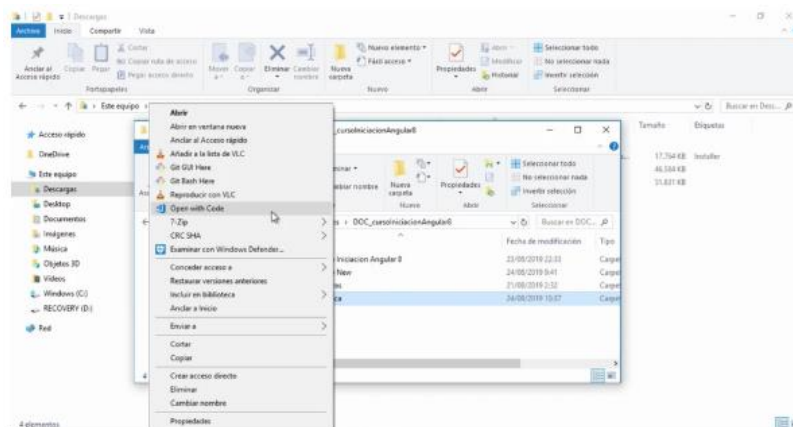
4 - APÉNDICES

4.1 - Instalación y configuración del sistema

Aplicación Angular

Para poder ejecutar la aplicación es necesaria la instalación de Angular 8. Sin embargo, antes de poder realizar la instalación de Angular 8, necesitamos instalar previamente:

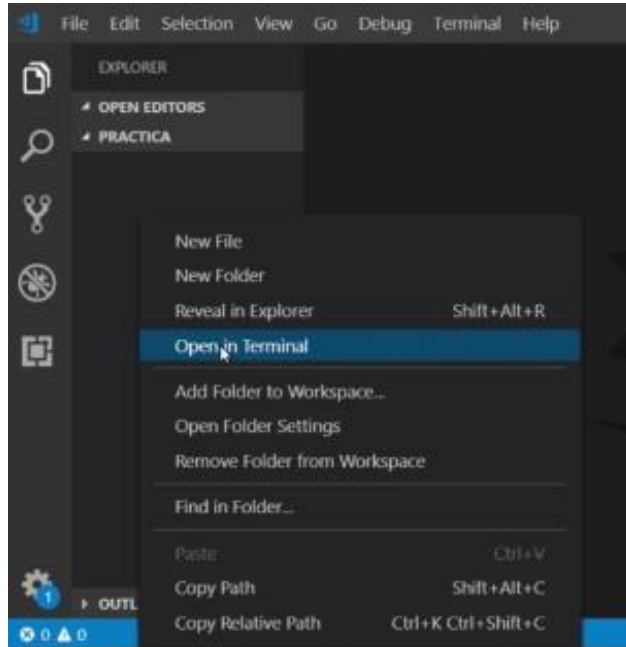
- Visual Studio Code:
 - La descarga se puede realizar desde su página web y pulsando en el botón de descargar.
- Git:
 - Realizamos la descarga del instalador desde su página web eligiendo nuestro sistema operativo y lo ejecutamos.
- Node y npm:
 - Para descargar Node accedemos a la web de nodejs¹², dónde aparecerán dos opciones de descarga. En este caso elegiremos la opción LTS que es la opción estable y la más recomendada para la mayoría.
 - Al finalizar la descarga, se ejecuta el instalador y se sigue el proceso de instalación con las opciones marcadas por defecto.
 - Una vez instalado Node, abrimos la carpeta del proyecto con la opción “Open with Code”.



¹² <https://nodejs.org/es/>

Ilustración 4.1. Apertura de Node con Visual Studio Code

- De esta forma cargamos la carpeta en Visual Studio Code, y en la zona izquierda pulsamos con el botón derecho para abrir la terminal.

**Ilustración 4.2. Apertura de la terminal de Visual Studio Code**

- Una vez se ha abierto la terminal, se procede a comprobar si Node se ha instalado correctamente, ejecutando el comando “node -v”. Esto nos debe mostrar cual es la versión de Node que se ha instalado. En el caso de que se mostrara un mensaje de error, nos indicaría que la instalación no se ha hecho correctamente.
- CLI Angular 8:
 - En la misma terminal que se ha mencionado antes se ejecutará el comando “npm install -g @angular/cli”. De esta forma, se instalará la última versión estable que exista.
 - Para comprobar si se ha instalado correctamente se ejecutará el comando “ng -v”, que nos devolverá la información de la versión instalada.

Una vez que se haya instalado todo, lo siguiente será clonar el proyecto almacenado en Github, para ello:

- Abrir una consola Windows con la combinación de teclas Windows + R.
- Escribir "cmd" en el recuadro que aparece al escribir la combinación de teclas anterior.
- Escribir en la terminal "git clone https://github.com/Ahriel21/PaginaWeb.git"
- Una vez clonado el proyecto tendremos que ejecutar en la terminal de Visual Studio Code el comando "npm -i".
- Una vez finalizado lo anterior, se lanzará ejecutando en la terminal el comando "ng serve -o"

Scripts Python

Para la ejecución manual del script de Python es necesario seguir los siguientes pasos:

- Ubicar nuestro script en la carpeta correspondiente.
- Abrir una consola Windows con la combinación de teclas Windows + R.
- Escribir "cmd" en el recuadro que aparece al escribir la combinación de teclas anterior.
- Se abrirá una consola como la siguiente:

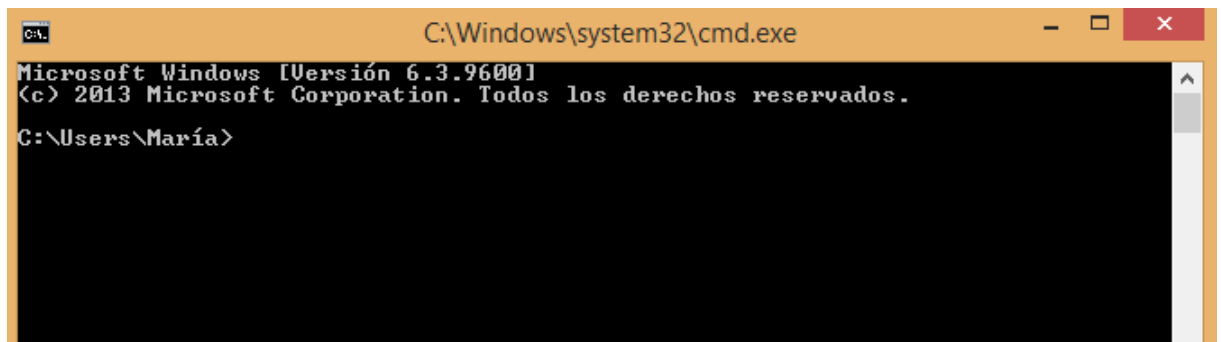


Ilustración 4.3. Consola de Windows

- Para ejecutar el script basta con escribir "python nombreDelScript.py"

Para la ejecución automática del script de Python es necesario seguir los siguientes pasos:

- Abrir una consola Windows con la combinación de teclas Windows + R.

- Escribir "taskschd.msc" en el recuadro que aparece al escribir la combinación de teclas anterior.
- Después, nos vamos al menú Acción y seleccionamos Crear tarea...:

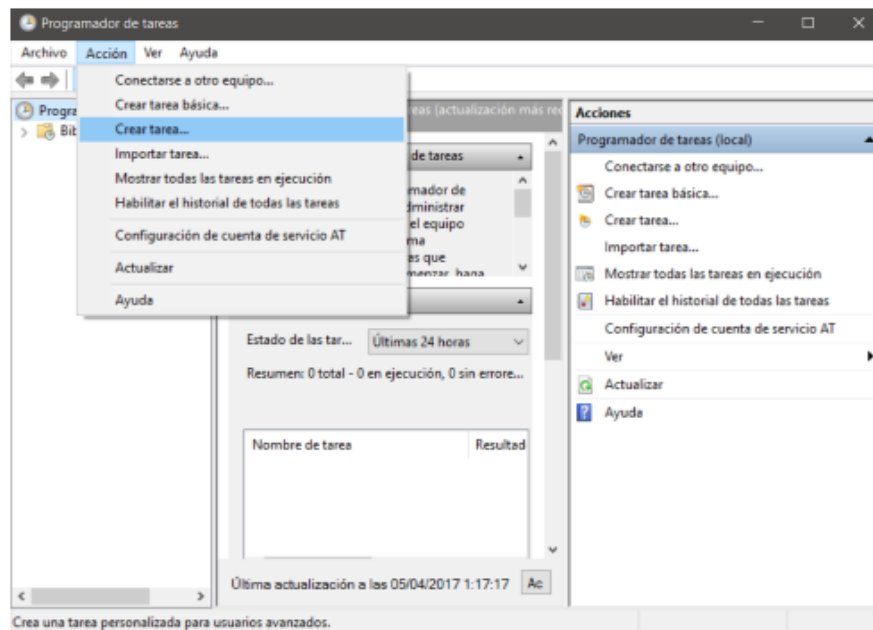


Ilustración 4.5. Crear tarea automática en Windows

- En la subventana que aparece en la pestaña General introducimos el nombre de la tarea (es obligatorio y no puede haber dos tareas con el mismo nombre):

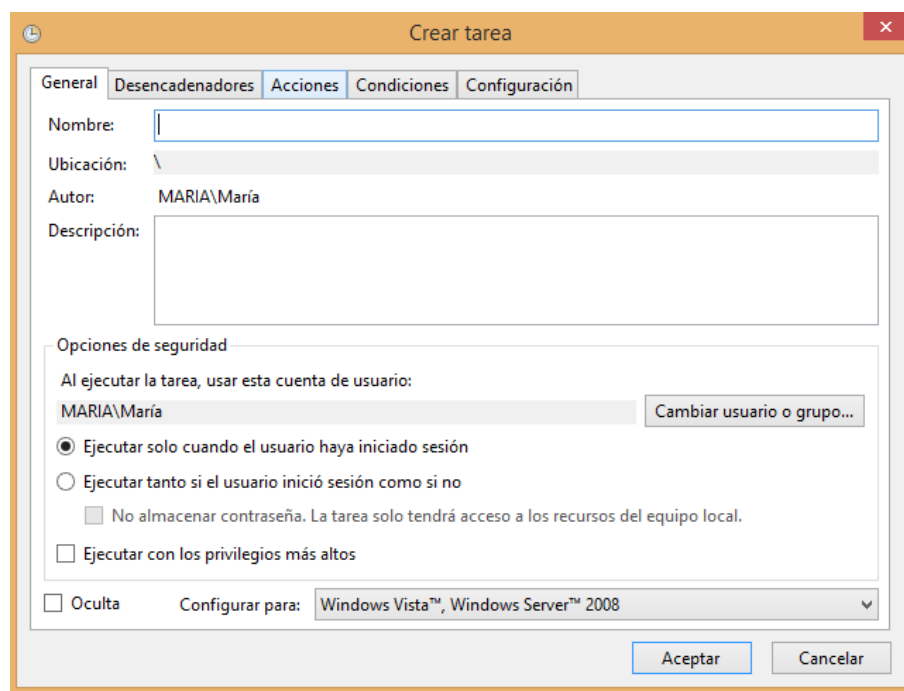
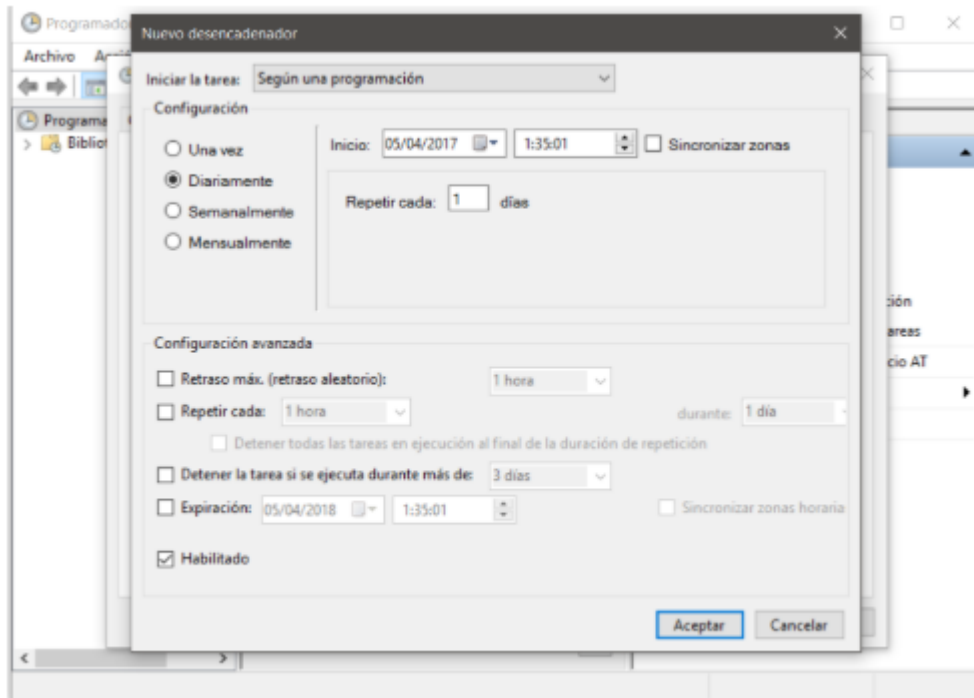


Ilustración 4.6. Pestaña General de Crear Tarea

- En la pestaña Desencadenadores hacemos clic en Nuevo y aparecerá una nueva subventana donde elegimos la hora, fecha, periodo, etc. en la que se quiere ejecutar la tarea, una vez seleccionada pulsamos Aceptar:

**Ilustración 4.7. Crear Nuevo Desencadenador**

- Ahora nos vamos a la pestaña Acciones y pulsamos en Nueva... En la ventana que se abre se podrá introducir la ruta del script a ejecutar. Esto funcionará si tenemos bien configuradas las variables de entorno (si al hacer doble clic sobre un archivo .py se ejecuta directamente en la CMD). De no ser así o querer usar otro intérprete podemos poner la ruta al intérprete, seguida de un espacio y la ruta del script.

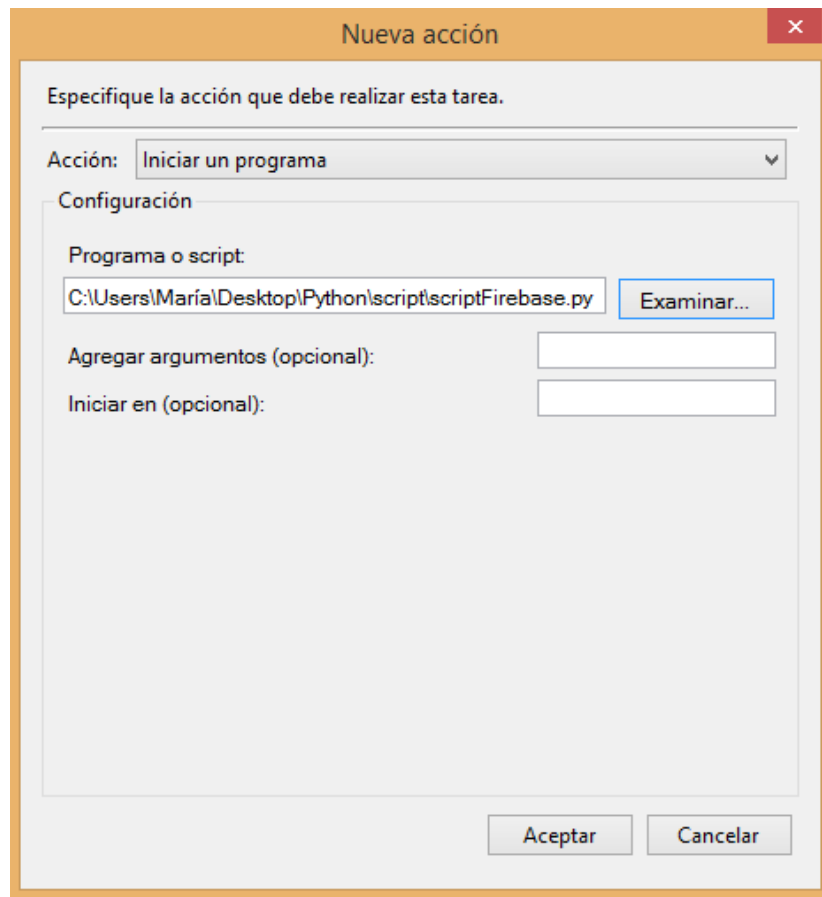


Ilustración 4.8. Crear una Nueva acción

4.2 - Manuales de usuario

En este apartado se describirá con detalle y paso a paso la funcionalidad de la aplicación.

Antes de empezar a utilizar la aplicación se tendrán que convertir los datos de la base de datos Firebase Realtime al formato de Cloud Firestore. Esto lo hará el script, por lo tanto, habrá que ejecutar el script manualmente o bien configurarlo para que se ejecute cada día a una hora determinada. Esta segunda forma sería la más recomendada.

Una vez hecho esto, es necesario conocer las distintas ventanas de la aplicación.

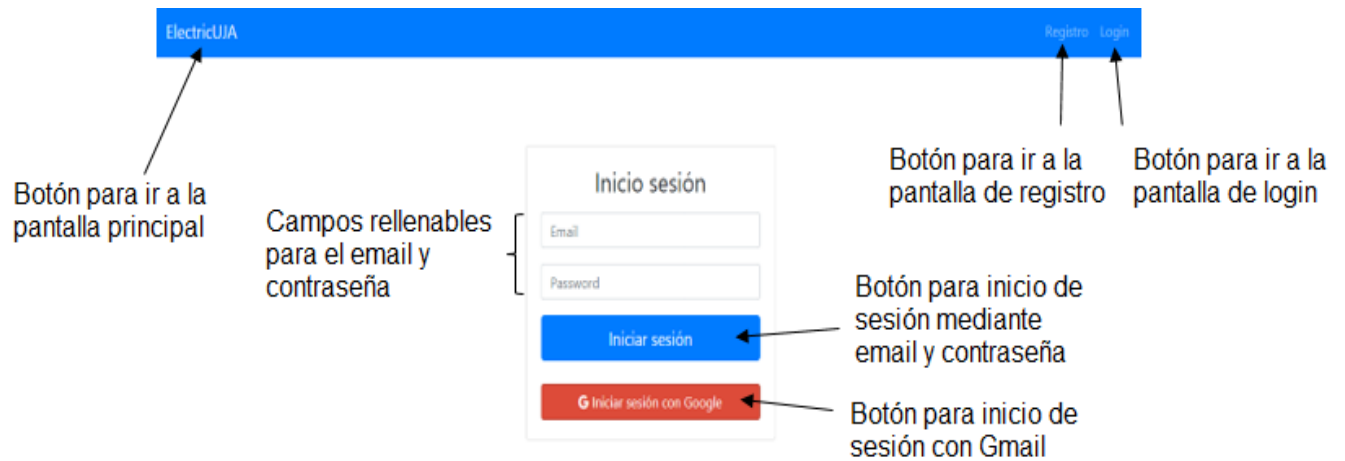


Ilustración 4.9. Pantalla de inicio de sesión detallada

En primer lugar, aparecerá esta ventana en la que habrá que loguearse mediante una de las dos opciones dependiendo de si eres administrador o un usuario.

En la pantalla de registro, estas serán las opciones que aparezcan en el navbar, sin embargo, hasta que no se completa el login o el registro, el botón del nombre la aplicación nos seguirá llevando hacia esta pantalla.

En el caso de que el usuario no tenga cuenta, se podrá registrar pulsando el botón de registro.

**Ilustración 4.10. Pantalla de registro detallada**

En este caso, si el usuario quiere registrarse mediante email y contraseña, el botón de registro estará bloqueado hasta que el usuario seleccione una foto. Esto se ha hecho para que aparezca una foto en la pantalla del perfil.

Ejemplo de la pantalla de registro:

- Título: Registro
- Formulario de Email: email ejemplo@red.ujaen.es
- Formulario de Password:|
- Sección "Seleccionar imagen:"
 - Botón "Seleccionar archivo"
 - Texto: Screenshot_20...road.lite.jpg
- Botón "Registrar" (azul)
- Botón "Registrar con Google" (rojo)

Ilustración 4.11. Pantalla de registro de ejemplo

No será necesario subir una foto en el caso de que el usuario se quiera registrar con Gmail ya que cogerá la foto que puesta en su cuenta de Gmail.

Una vez que se haya completado el login o el registro aparecerá la pantalla principal.

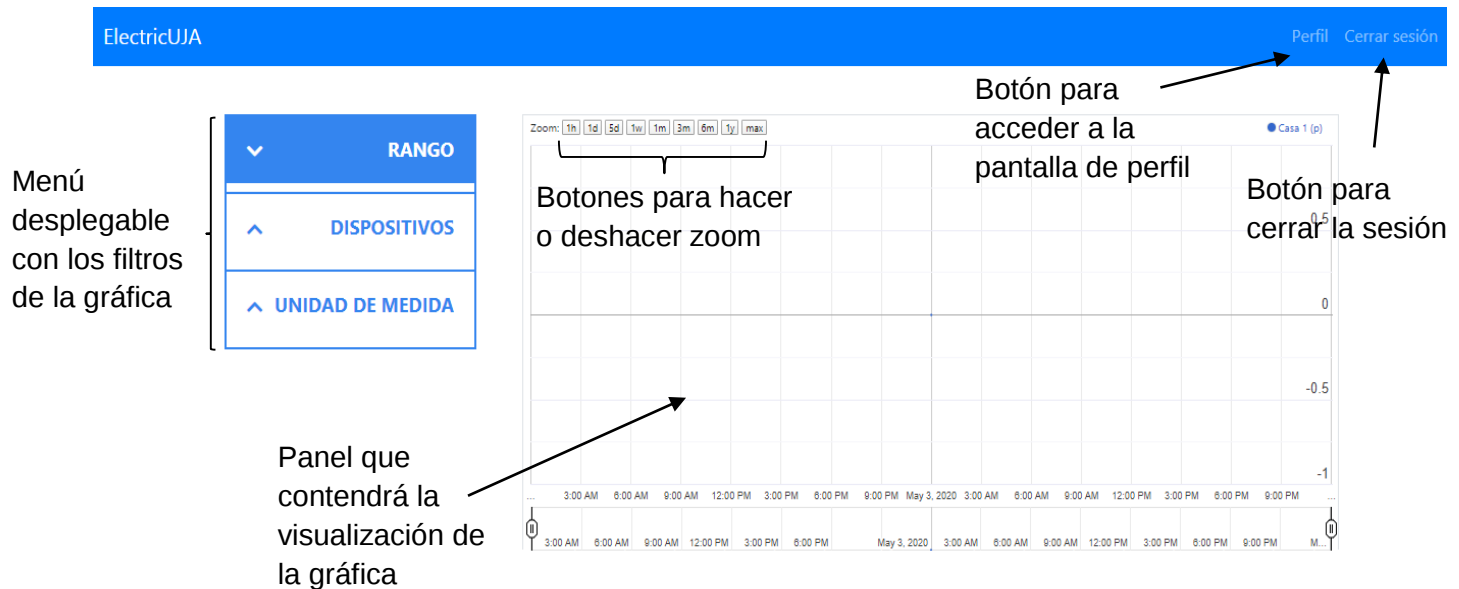


Ilustración 4.12. Pantalla principal detallada

En esta pantalla el navbar habrá cambiado pudiendo acceder a la pantalla de perfil o pudiendo cerrar la sesión. Esta vez, si pulsamos en el nombre de la aplicación nos llevará a esta página principal ya que estamos logueados.

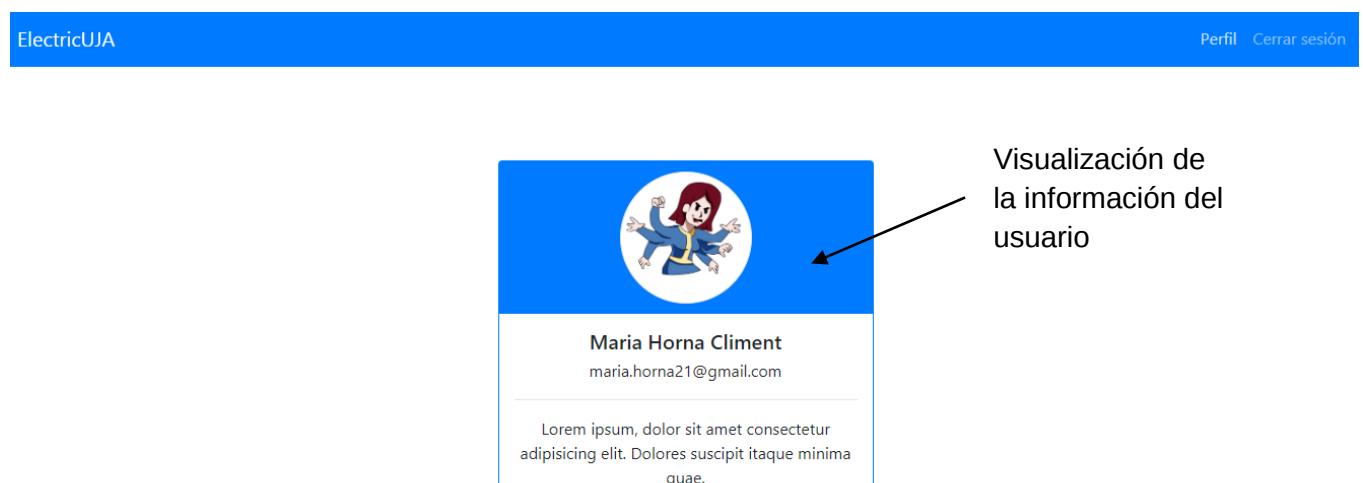


Ilustración 4.13. Pantalla de perfil detallada

Esta pantalla contendrá la información del usuario y la foto que aparecerá será la que se haya seleccionado en el momento en el que el usuario se registró. Para volver a la pantalla principal, bastaría con pinchar en el nombre de la aplicación.

RANGO

03-05-2020 - 03-05-2021

May 2020

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Jun 2020

Su	Mo	Tu	We	Th	Fr	Sa
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

OK

DISPOSITIVOS

Casa 1 x

☐ Seleccionar todo

Buscar

☒ Casa 1

☐ Coche 1

☐ Fotovoltaica 1

UNIDAD DE MEDIDA

i x p x

☐ Seleccionar todo

Buscar

☐ fi

☒ i

☒ p

☐ v

☐ s

Ilustración 4.14. Filtros detallados

Por defecto, los tres filtros del menú tendrán algo seleccionado. En este caso, como la fecha por defecto es la del día actual y en este día no se han subido datos, este panel aparecerá vacío. Sin embargo, se supone que todos los días se subirán los datos y por lo tanto no aparecerá vacío.



Ilustración 4.15. Gráfica de muestra

Este sería un ejemplo de cómo debería de aparecer la gráfica. Como se ha dicho antes, cada filtro tendrá algo seleccionado por defecto, por lo que he decidido poner por defecto la Casa 1 en los dispositivos, p como unidad de medida y la fecha del día actual.

Los valores “x” de la gráfica será el tiempo, es decir, horas, minutos y segundos y la “y” serán los valores que tenga cada una de las unidades de medida.

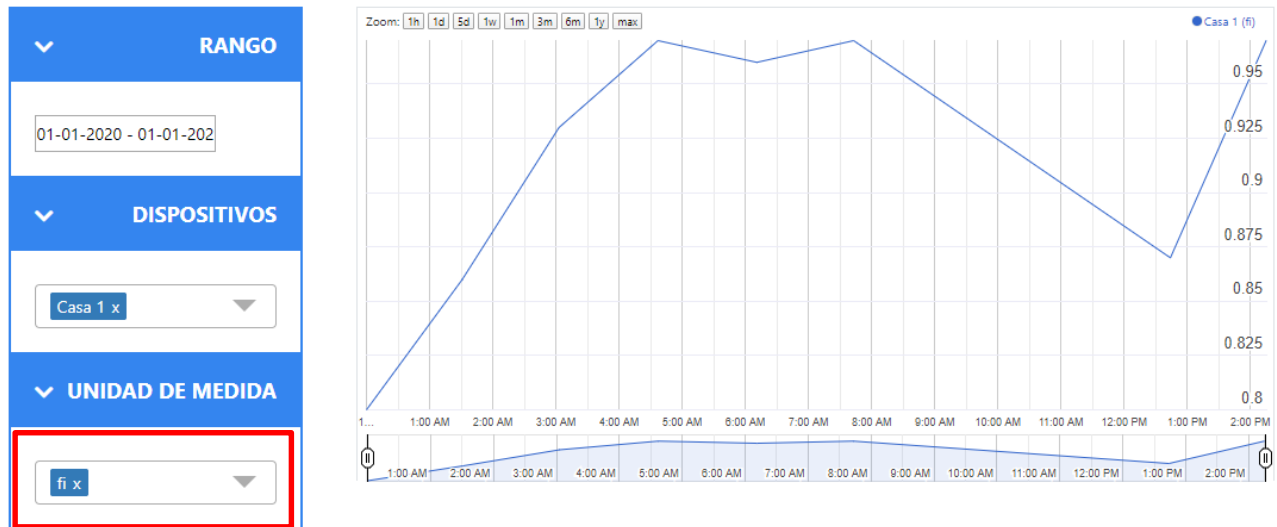


Ilustración 4.16. Gráfica de muestra con una opción de cada filtro

En el caso de cambiar la unidad de medida, al tener esta un valor más pequeño, los valores de la “y” cambiarán.

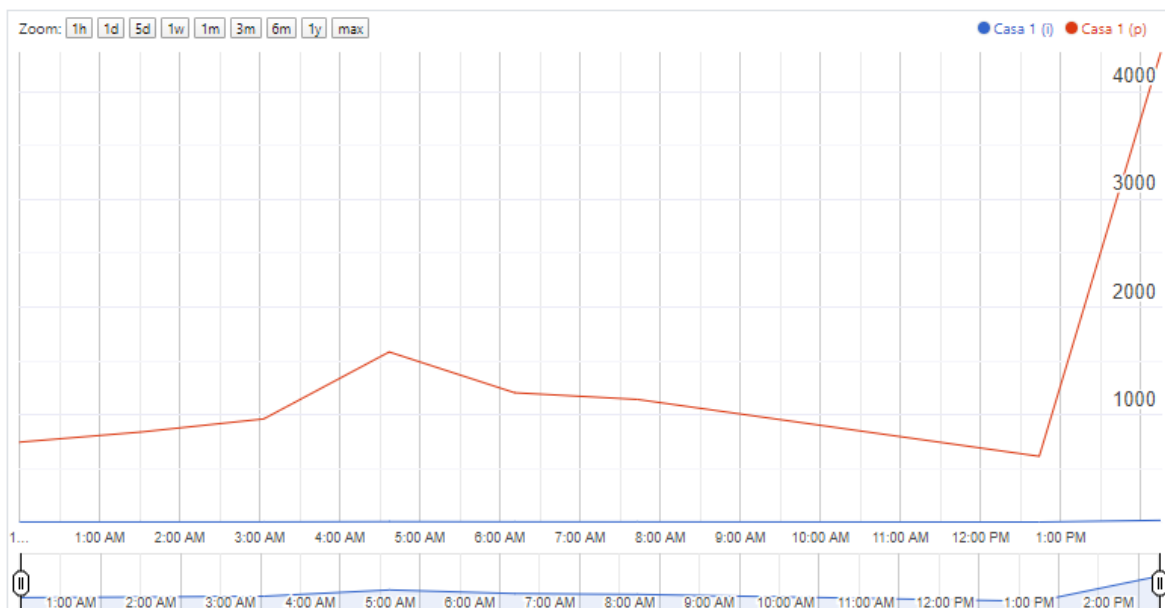


Ilustración 4.17. Gráfica de muestra con más de una opción de cada filtro

También es posible seleccionar varios dispositivos y varias unidades de medida. De esta forma, la “y” dependerá de la unidad de medida que tenga los valores mayores. Como se aprecia aquí, el valor de i es muy pequeño en comparación con el de p .

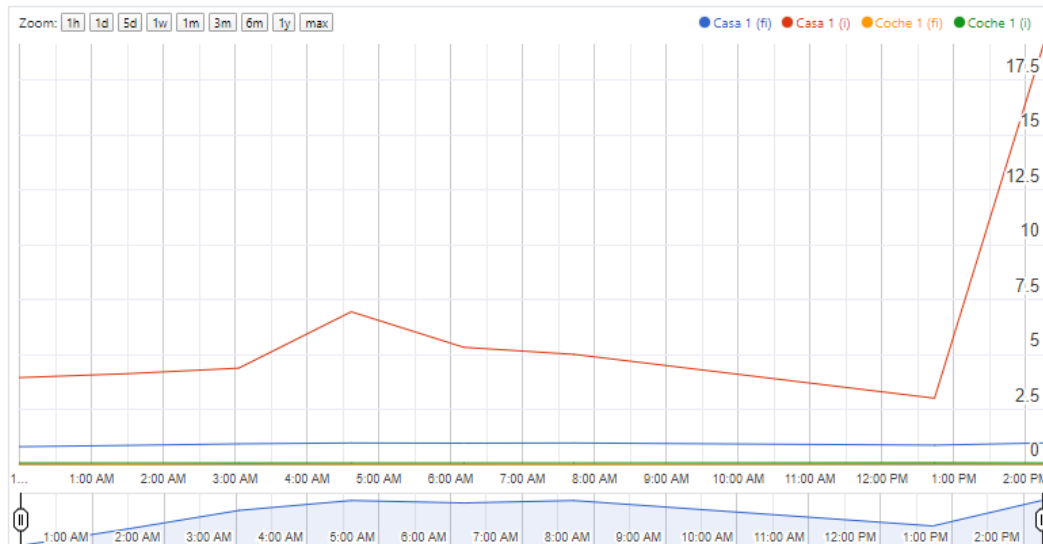


Ilustración 4.18. Gráfica de muestra con más de una opción de cada filtro

Sin embargo, aquí el valor de i sería el mayor y el resto de unidades de medida tendrían valores mucho más pequeños.



Ilustración 4.19. Puntos clave de la gráfica

Para ver con mejor precisión la comparación de valores, la gráfica dispone de una opción que consiste en posicionar el ratón encima de la línea pudiendo de esta forma observar un punto en ese determinado instante y así poder comparar los valores que tiene cada unidad de medida. Los valores de cada unidad de medida se podrán observar en la leyenda de la gráfica.

5 - REFERENCIAS

Tutoriales instalación de Angular: <https://www.youtube.com/watch?v=itsrbSq4Rx0>

<https://www.techomoro.com/how-to-install-and-setup-angular-8-on-windows-10/>

Tutorial instalación Python y Pycharm:

<https://www.youtube.com/watch?v=btrTN0q5SFY>

Tutoriales para sistema de login, registro y perfil:

https://www.youtube.com/watch?v=UiA8QQ5modE&list=PL_9MDdjVuFjHgSCW7_15yoCAFUE5NAEjm

Google Charts: <https://developers.google.com/chart/interactive/docs/gallery>

Salarios medios en España usados en el proyecto (Indeed):

<https://www.indeed.es/salaries/jefe-de-proyecto-Salaries>

<https://www.indeed.es/salaries/analista-programador-Salaries>

<https://www.indeed.es/salaries/programador-junior-Salaries>

<https://www.indeed.es/salaries/tester-Salaries>

Smart Grid: <https://www.fundacionendesa.org/es/recursos/a201908-smart-grid>

<https://www.thinksmartgrids.fr/en/actualites/france-leads-smart-grids-investments-in-europe-2>

Kanban: <https://leadsfac.com/marketing/que-es-la-metodologia-kanban-y-como-utilizarla/>

Trello: <https://trello.com/b/we67Nh09/tfg-mar%C3%ADa>

Funcionamiento de SCT-013 y Arduino: <https://programarfacil.com/blog/arduino-blog/sct-013-consumo-electrico-arduino/>

Comparativa de tecnologías: <https://codigofacilito.com/articulos/angular-react-vue>

<https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>