

Aplicación para la gestión de una instalación fotovoltaica

Memoria



VNiVERSIDAD
D SALAMANCA

Trabajo de Fin de Grado

Grado de Ingeniería Informática

Septiembre 2021

Tutor:

Iván Álvarez Navia

Alumno:

Miguel Sánchez González

D. Iván Álvarez Navia, profesor del Departamento de Informática y Automática de la Universidad de Salamanca

CERTIFICA:

Que el trabajo titulado “Aplicación para la gestión de una instalación fotovoltaica” ha sido realizado por D. Miguel Sánchez González, con DNI 70921138V y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Grado de Ingeniería Informática de esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a 6 de septiembre de 2021.

Firmado por ALVAREZ NAVIA
IVAN - 45426138B el día
06/09/2021 con un
certificado emitido por AC
FNMT Usuarios

D. Iván Álvarez Navia
Dpto. Informática y Automática
Universidad de Salamanca

Resumen

Este trabajo de fin de grado se ha centrado en crear una herramienta para monitorizar una instalación fotovoltaica, pudiendo ver los datos de monitorización en tiempo real y en un histórico. El trabajo constará de dos partes principales, una enfocada al diseño de los sensores de medida de la instalación y el almacenamiento de los datos obtenidos, y por otra parte la creación de una plataforma para la visualización de los datos obtenidos.

La instalación que se ha monitorizado en este trabajo consta de un panel fotovoltaico que proporciona 18,85 V y 8,5 A en el punto de máxima potencia, un inversor de corriente que genera 230 V de corriente alterna y unas baterías de plomo de 12 V con una capacidad de 600 Ah.

Los datos de la instalación que se van a registrar son la tensión y la corriente que genera el panel solar, para poder calcular la potencia generada; el voltaje de la batería, para poder calcular su porcentaje de carga; la tensión y corriente a la salida del inversor, para poder medir cuanta energía está usando el usuario.

El diseño del hardware para la obtención de los datos de la instalación consta de una Raspberry Pi 4 conectada a 2 convertidores analógico-digital, que leen, uno de ellos, el voltaje de los paneles solares y el voltaje proporcionado por el sensor de corriente del panel solar, y el otro, el voltaje de las baterías; un Arduino Nano Every, usado para la medida del voltaje y corriente a la salida del inversor - registra mediante sus convertidores analógico-digital el voltaje del sensor de corriente y el voltaje a la salida del inversor. Las comunicaciones entre los conversores analógico-digital, la placa Arduino y la Raspberry Pi se realizan a través del protocolo de comunicación I2C.

Los datos recogidos por la Raspberry Pi se envían a una base de datos en la nube llamada Firestore, que es parte del conjunto de herramientas que ofrece el servicio Firebase de Google y que usaremos en el resto de la aplicación.

La visualización de los datos recogidos se realiza en una aplicación web. Para acceder a ella se ha usado el servicio de *hosting* de Firebase. La aplicación también está conectada con la base de datos Firestore de Firebase. De esta forma podremos visualizar tanto los datos en tiempo real de la instalación como un histórico y resúmenes de estos. El registro e inicio de sesión de los usuarios en la aplicación web se realiza con el servicio de Firebase, que permite gestionar las cuentas de los usuarios de manera sencilla.

El sistema ha sido diseñado para que diferentes usuarios puedan conectar sus equipos de medición. Los datos se enviarán a la base de datos en la nube y podrán ser visualizarlos desde cualquier lugar de forma segura y privada.

Palabras clave: instalación fotovoltaica, monitorización, tiempo real, histórico, aplicación web.

Abstract

This final degree Project focuses on creating a tool to monitor a photovoltaic installation, being able to see its data in real time and in a history. The work will consist of two main parts, one based on the design of the measurement sensors of the installation and the storage of the data obtained, and another part focused on the creation of a platform for the visualization of the obtained data.

The installation that has been monitored in this project consists of a solar panel that provides 18.85 V and 8.5 A at the maximum power point, a current inverter that generates 230 V of alternating current and 12 V lead batteries with a capacity of 600 Ah.

The installation data to be recorded are the voltage and current generated by the solar panel, to be able to obtain the generated power; the battery voltage, to be able to calculate its percentage of charge; the voltage and current at the output of the inverter, to be able to measure how much energy is being used.

The hardware design to collect the installation data consists of a Raspberry Pi 4 connected to 2 analog-digital converters, that read, one of them, the solar panels voltage and the solar panel current sensor voltage, and the other, the battery voltage; an Arduino Nano Every, to measure the voltage and current at the output of the inverter - it registers through its analog-digital converters the voltage of the current sensor and the voltage at the output of the inverter. The communications between the analog-digital converters, the Arduino board and the Raspberry Pi are carried by the I2C communication protocol.

The data collected by the Raspberry Pi is sent to a cloud database called Firestore, which is part of the set of tools offered by Google's Firebase service and Will be used on the rest of the application.

The data collected is displayed on a webapp. The webapp hosting is provided by Firebase hosting service. The data is also stored in Firebase Firestore database. This setup will allow us to view both the real-time data of the installation as well as a history and summaries of these. User registration and log in on the webapp works with the Firebase service, which allows to manage user accounts easily.

The system has been designed so that different users can connect their measuring equipment. The data is then sent to the cloud database and can be accessed from anywhere in a secure and private way.

Keywords: photovoltaic installation, monitor, real time, history, web app.

Tabla de contenidos

1. Introducción	13
2. Objetivos	16
2.1. Objetivos del sistema	16
2.2. Objetivos personales	16
3. Conceptos teóricos.....	17
4. Técnicas y herramientas.....	23
4.1. Entorno de desarrollo software.....	23
4.2. Entorno de desarrollo hardware	25
4.3. Librerías y <i>frameworks</i>	29
4.4. Lenguajes de programación.....	29
4.5. Herramientas CASE	30
5. Aspectos relevantes del desarrollo del proyecto.....	31
5.1. Marco de trabajo	31
5.2. Estimación de coste y esfuerzo.....	32
5.3. Planificación temporal	33
5.4. Especificación de requisitos.....	34
5.5. Análisis del sistema	36
5.6. Diseño del sistema	38
5.7. Implementación	42
5.8. Pruebas	54
5.9. Descripción funcional.....	58
6. Conclusiones y líneas de trabajo futuras	64
6.1. Conclusiones	64
6.2. Líneas de trabajo futuras	65
7. Referencias.....	66
8. Bibliografía	68
8.1. Materiales didácticos consultados	68
8.2. Sitios web consultados.....	68
9. Agradecimientos	70

Tabla de ilustraciones

Ilustración 1: Gráfico generación solar fotovoltaica España REE	13
Ilustración 2: Funcionamiento de una célula fotovoltaica	17
Ilustración 3: Elementos básicos de una batería	18
Ilustración 4: Flujo de electrones e iones en una batería	18
Ilustración 5: Modos de operación I2C.....	19
Ilustración 6: Divisor de tensión	19
Ilustración 7: Resistencia pull-up.....	20
Ilustración 8: Amplificador operacional	21
Ilustración 9: Raspberry Pi 4	21
Ilustración 10: Raspberry Pi 4 Specs	21
Ilustración 11: Arduino Nano Every.....	22
Ilustración 12: Logo Firebase.....	22
Ilustración 13: Visual Studio Code	23
Ilustración 14: Navegador Microsoft Edge y función DevTools	24
Ilustración 15: Tinkercad circuits.....	25
Ilustración 16: CircuitJS.....	25
Ilustración 17: Multímetro digital.....	26
Ilustración 18: Amperímetro y voltímetro analógicos	26
Ilustración 19: Medidor de potencia	27
Ilustración 20: Osciloscopio	27
Ilustración 21: Fuente de alimentación.....	28
Ilustración 22: Placa de prototipos.....	28
Ilustración 23: Proceso unificado	31
Ilustración 24: Resultado estimación de costes y esfuerzo.....	32
Ilustración 25: Diagrama de Gantt y camino crítico	33
Ilustración 26: Duración fases, planificación temporal	33
Ilustración 27: Diagrama de actores.....	34
Ilustración 28: Diagrama de paquetes.....	35
Ilustración 29: Ejemplo paquete	35
Ilustración 30: Modelo de dominio	36
Ilustración 31: Vista de la arquitectura	37
Ilustración 32: Ejemplo caso de uso análisis	37
Ilustración 33: Patrón MVC.....	38
Ilustración 34: Vista arquitectónica.....	38
Ilustración 35: WebApp - Modelo	39
Ilustración 36: Ejemplo caso de uso	39
Ilustración 37: Modelo de despliegue del sistema.....	40
Ilustración 38: Cuotas Firestore.....	41
Ilustración 39: Estructura espacio de trabajo.....	42
Ilustración 40: Prototipo pantalla principal.....	44
Ilustración 41: Prototipo histórico de datos.....	44
Ilustración 42: Prototipo datos de perfil	44

Ilustración 43: Conversor analógico digital ADS1115.....	45
Ilustración 44: Sensor de corriente ACS712	45
Ilustración 45: Esquema conceptual I2C	46
Ilustración 46: Divisores de tensión	47
Ilustración 47: Amplificador operacional, Modo seguidor de voltaje.....	48
Ilustración 48: Amplificador operacional, Modo inversor	48
Ilustración 49: Esquema OPamp	48
Ilustración 50: Circuito amplificadores operacionales	48
Ilustración 51: Circuito medida a la salida del inversor.....	49
Ilustración 52: Análisis forma de onda salida del circuito del inversor.....	49
Ilustración 53: Circuito medidor de voltaje de la batería.....	50
Ilustración 54: Circuito de medición de potencia de las placas solares.....	50
Ilustración 55: Circuito de medición de consumo a la salida del inversor	51
Ilustración 56: Realización física de los circuitos de los sensores.....	51
Ilustración 57: Esquema global de la instalación	52
Ilustración 58: Entorno de pruebas de Firestore.....	54
Ilustración 59: Comando systemctl	54
Ilustración 60: Comando journalctl	55
Ilustración 61: WebApp y DevTools	55
Ilustración 62: Simulación de dispositivos.....	56
Ilustración 63: Lighthouse DevTools.....	56
Ilustración 64: Placa prototipo	57
Ilustración 65: Fuente alimentación.....	57
Ilustración 66: Osciloscopio.....	57
Ilustración 67: Circuito en placa PCB.....	57
Ilustración 68: Multímetro digital.....	57
Ilustración 69: Medidor de potencia digital	57
Ilustración 70: Visión global del sistema	58
Ilustración 71: Pantalla de acceso a la aplicación	59
Ilustración 72: Pantalla principal monitorización.....	59
Ilustración 73: Algoritmo de cálculo máxima potencia Pt1.....	60
Ilustración 74: Algoritmo de cálculo de máxima potencia Pt2.....	61
Ilustración 75: Ejemplo gráfica de histórico	62
Ilustración 76: Calendario de selección de fechas	62
Ilustración 77: Mensaje emergente	63
Ilustración 78: Ventana de datos de la cuenta.....	63

Índice de tablas

Tabla 1: Ejemplo participante.....	34
Tabla 2: Leyenda colores conexiones I2C.....	46
Tabla 3: Leyenda colores cables instalación global	52
Tabla 4: Lista de componentes y precio	53

1. Introducción

La sostenibilidad energética es un pilar fundamental de la sociedad moderna. Para lograrla se debe seguir un modelo de generación distribuida y, en la medida de lo posible, de carácter renovable. En el caso de la energía fotovoltaica hay que tener en cuenta que, las horas de mayor producción se corresponden con las horas de mayor consumo, lo que la hace una energía muy eficaz para esta tarea.

España es un país con una situación geográfica muy ventajosa en lo referido a la producción de energía fotovoltaica, por sus excelentes condiciones de irradiación solar, y gracias a ello tenemos un gran potencial para el desarrollo de este tipo de energía.

Según el informe de las energías renovables en el sistema eléctrico español del 2020 realizado por Red Eléctrica de España [1], la generación de energía solar fotovoltaica en ese año llegó casi a duplicar la de los máximos de años anteriores a 2019, tal y como se aprecia en la Ilustración 1.

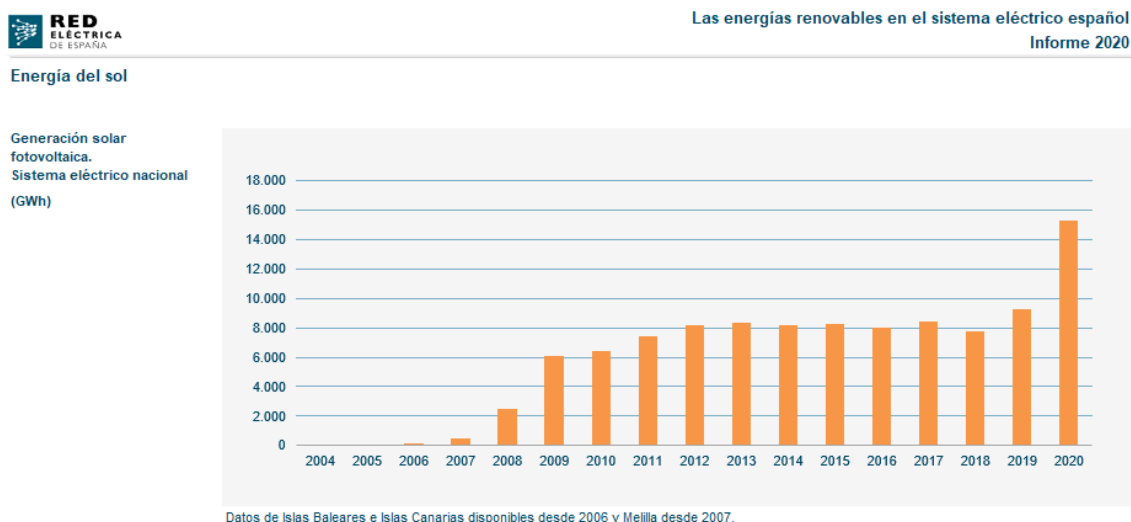


Ilustración 1: Gráfico generación solar fotovoltaica España REE

Este aumento se ha debido principalmente al abaratamiento del coste de los paneles solares, la eliminación del denominado impuesto al sol y la progresiva supresión de trabas burocráticas, propiciadas por el nuevo marco legal establecido por la actual administración en los Reales Decretos 15/2018 y 244/2019. [2]

Gracias a esto y a las metas propuestas por la unión europea para el año 2030, se prevé que España sea uno de los principales productores de energía solar del mundo.

Referido a la generación de energía solar fotovoltaica también encontramos el ámbito del autoconsumo, donde las instalaciones de paneles solares comienzan a ser una alternativa a la generación convencional, viables económicamente y por las que se está optando en muchos hogares.

Con la aplicación propuesta en este proyecto se busca crear una manera de monitorizar instalaciones fotovoltaicas nuevas y existentes, pudiendo gestionar la energía generada y consumida de una manera más fácil y eficiente.

El sistema constará de una serie de sensores que realizarán medidas en la instalación y los enviarán los datos obtenidos a una base de datos, que será accesible a través de una aplicación web desde cualquier lugar con conexión a internet.

La memoria que se verá a continuación consta de diferentes apartados donde se puede ver una descripción del proyecto realizado y la documentación técnica elaborada durante el proceso, los apartados de la memoria son los siguientes:

- **Objetivos:** En este apartado se muestran las principales metas que se pretenden alcanzar en el proyecto y se subdividen en objetivos del sistema y objetivos personales.
- **Conceptos teóricos:** En este apartado se hará una breve descripción de ciertos términos usados en el trabajo, para poder comprender esta memoria en su totalidad.
- **Técnicas y herramientas:** Se hablará entre otros temas de los entornos de desarrollo, lenguajes y recursos de programación o bibliotecas usadas durante la realización del trabajo, y que han sido fundamentales para el desarrollo.
- **Aspectos relevantes del desarrollo del proyecto:** En esta parte se recogen los aspectos más importantes de las diferentes etapas del desarrollo del sistema, centrado en el proceso unificado.
- **Conclusión y líneas de trabajo futuras:** Representan una visión del final del desarrollo del proyecto y los resultados obtenidos, así como las diferentes ramas de mejora y expansión de este proyecto.
- **Referencias y bibliografía:** Contienen las referencias a apartados de la memoria y los recursos consultados en la realización del trabajo.

A parte de la memoria, se han creado una serie de anexos, complementarios a esta memoria, que constituyen la documentación técnica del proyecto, y se dividen en:

- **Anexo I – Planificación temporal:** Recoge la planificación temporal de las tareas del proyecto para poder visualizar de una manera gráfica las actividades realizadas hasta obtener el producto final.
- **Anexo II – Especificación de requisitos:** En este anexo contienen los diferentes requisitos que el sistema debe cumplir para su correcto funcionamiento.
- **Anexo III – Análisis de requisitos:** este documento se ha elaborado un análisis de los requisitos que debe seguir el sistema, descritos en el Anexo II.
- **Anexo IV – Diseño del sistema:** En este documento, continuación del anexo anterior, se detalla, la decisión de implementación del sistema y contiene, principalmente, el diseño arquitectónico del sistema, la realización de los casos de uso y el modelo de despliegue.
- **Anexo V – Manual del programador:** Este manual enfocado a enseñar cómo funciona el sistema desde el punto de vista de un ingeniero informático, para que

otros desarrolladores e ingenieros puedan trabajar en el proyecto de una forma más cómoda.

- **Anexo VI – Manual de usuario:** En un manual de uso de la aplicación web orientado a los usuarios finales, en el que se describe la funcionalidad y la forma de uso de cada una de las partes de la aplicación.

2. Objetivos

Vamos a hablar de los objetivos que debe cumplir el sistema y los objetivos personales planteados antes del desarrollo.

2.1. Objetivos del sistema

El objetivo principal de este trabajo es desarrollar un sistema de monitorización de una instalación fotovoltaica, para ello se han planteado los siguientes objetivos:

- **Gestión de la monitorización del sistema:** Este objetivo se centra en la obtención de los datos de la instalación mediante sensores que enviarán datos a un dispositivo que será encargado de enviarlos a la base de datos.
- **Gestión de la visualización de los datos:** La visualización de datos consistirá en desarrollar un sistema para la aplicación web que haga consultas a la base de datos y cree una representación gráfica de los datos.
- **Gestión de los roles del sistema:** El sistema deberá gestionar las cuentas anónimas y registradas, los permisos de acceso a la base de datos y las funciones del hardware de monitorización del sistema.
- **Investigación sobre consumo y producción:** Se elaborará un algoritmo de cálculo de horas de máxima producción y consumo, así como unos resúmenes de la potencia generada y consumida.

2.2. Objetivos personales

Cuando se eligió la idea de este proyecto se buscaba una forma de visualizar los datos de una instalación fotovoltaica ya existente, que fuera cómoda y accesible desde cualquier lugar.

La motivación principal del proyecto era aprender nuevos conocimientos sobre desarrollo web, mejorando las habilidades básicas como podían ser HTML5 o CSS3, y adquiriendo los conocimientos necesarios para el desarrollo con el lenguaje JavaScript. El pilar principal de la aplicación web ha sido intentar seguir las directrices de la reciente tendencia de las aplicaciones web progresivas (PWA) [3] que, de forma resumida, es un paso intermedio entre una aplicación nativa de cualquier dispositivo y una aplicación web del navegador. Muchas empresas han optado por esta filosofía en los últimos años como puede ser el caso de Starbucks [4], Twitter [5], o Spotify [6] entre otras.

Como muchos de los lenguajes de programación que se han usado para este proyecto no los hemos visto durante la carrera y no he podido hacer mucho uso de los recursos aprendidos, pero si han servido las bases adquiridas para, por ejemplo, el aprendizaje de JavaScript y Python y el uso de la programación orientada a objetos (POO) [7] entre otras.

3. Conceptos teóricos

A continuación, se expondrán algunos de los conceptos teóricos usados en este trabajo para mejorar la comprensión de la memoria.

Panel solar fotovoltaico:

Es un dispositivo compuesto por una serie de células fotovoltaicas conectadas entre sí para producir corriente eléctrica.

El funcionamiento de una célula fotovoltaica reside en el principio del efecto fotoeléctrico, por el que Albert Einstein ganó un Premio Nobel en 1921, gracias a la explicación teórica en su artículo *Heurística de la generación y conversión de la luz* de 1905 [8].

El efecto fotoeléctrico, descubierto en 1887 por Heinrich Hertz, se produce cuando un material recibe radiación electromagnética y comienza a emitir electrones. Pero esto no ocurre con cualquier tipo de radiación, solo con aquella que supera un cierto nivel de energía (frecuencia).

Albert Einstein usó la explicación de Max Planck sobre los cuantos de luz, usada para resolver el problema de la Catástrofe del Ultravioleta, en su explicación del efecto fotoeléctrico. Demostró como la emisión de los electrones en un material era producida por la absorción de cuantos de luz y estaba relacionada con una frecuencia específica de cada material, a partir de la cual comenzaba a producirse este efecto.

Las células fotovoltaicas aprovechan un caso especial del efecto fotoeléctrico llamado efecto fotovoltaico. Las células fotovoltaicas están formadas por una unión de semiconductores, dopados positiva y negativamente, al recibir los fotones de los rayos de luz liberan electrones de los enlaces de los átomos de la placa semiconductor negativa, creando así una diferencia de potencial entre las placas semiconductoras. [9]

Un ejemplo de cómo funciona el efecto fotovoltaico se puede observar en la Ilustración 2, donde se aprecia la construcción de una célula fotovoltaica, con la unión p-n de dos semiconductores, de silicio en este caso, y se representa como los fotones al alcanzar la célula liberan un electrón del semiconductor dopado con fósforo (semiconductor N), generando una corriente eléctrica.

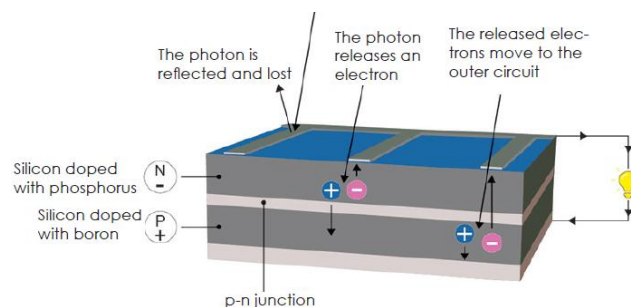


Ilustración 2: Funcionamiento de una célula fotovoltaica

Batería de plomo:

Una batería eléctrica es un dispositivo capaz de almacenar energía en forma de reacciones químicas y transformarla en una corriente eléctrica. Se componen generalmente por tres elementos el ánodo, el cátodo, y el electrolito como se puede ver en la Ilustración 3:

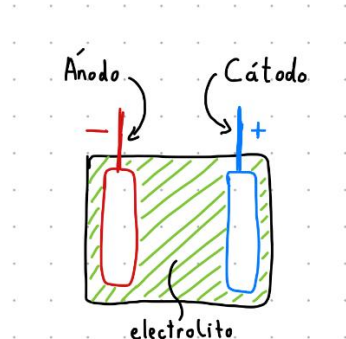


Ilustración 3: Elementos básicos de una batería

La batería genera una corriente eléctrica a partir de un proceso químico llamado reducción-oxidación, en la que el ánodo perderá electrones (se oxida), y el cátodo ganará electrones (se reduce), al perder y ganar electrones ambos elementos el ánodo y el cátodo quedarían desestabilizados, por lo que se produce un flujo de iones del ánodo al cátodo a través del electrolito, que tiene unas propiedades específicas para dejar pasar solamente los iones del ánodo al cátodo y viceversa, sin cortocircuitar la batería que es cuando entran en contacto directo el cátodo y el ánodo.

Este flujo de electrones e iones se ve representado en la Ilustración 4.

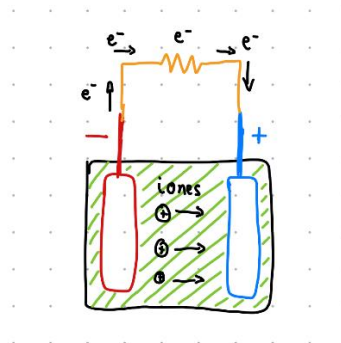


Ilustración 4: Flujo de electrones e iones en una batería

Cuando se carga la batería el proceso es el inverso, los electrones salen del cátodo y entran en el ánodo y los iones viajan a través del electrolito desde el cátodo hasta el ánodo.

Las baterías de plomo están compuestas generalmente por un ánodo de plomo, un cátodo de dióxido de plomo y como electrolito usan ácido sulfúrico. Esta configuración crea un voltaje de unos 2 voltios por celda y las baterías de plomo suelen estar compuestas por unas 3 o 6 celdas en serie proporcionando 6 o 12 voltios respectivamente.

Inversor de corriente:

Es un dispositivo electrónico encargado de transformar una corriente continua en una corriente alterna con onda sinusoidal muy parecida a la que se obtendría a la salida de un alternador. Estos dispositivos permiten usar baterías para alimentar aparatos electrónicos cotidianos que van conectados a la red (230 V - 50Hz).

Protocolo I2C:

Es un protocolo de comunicación síncrona en serie, sus siglas significan (Circuito Inter Integrado, del inglés *Inter-Integrated Circuit*). [10]

Para la transmisión de datos se usan dos líneas bidireccionales llamadas SDA y SCL y sus funciones son las siguientes:

- **Serial Data (SDA):** Los datos se transfieren a través de esta línea.
- **Serial Clock (SCL):** Lleva la señal de reloj.

Al ser un protocolo de comunicación síncrona se usa la línea de reloj para sincronizar el envío de los datos concretamente cada bit transferido por la línea SDA se sincroniza con un flanco de bajada de cada pulso de la señal de reloj. [11]

Los dispositivos conectados con este protocolo pueden operar en dos modos, representados en la Ilustración 5:

Modo maestro: Envía y recibe datos y genera la señal de reloj.

Modo esclavo: Solo envía datos cuando recibe una señal del maestro.

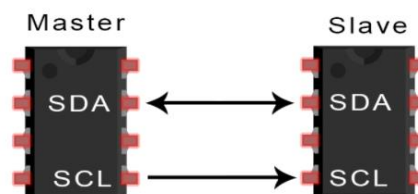


Ilustración 5: Modos de operación I2C

Divisor de tensión:

Es una configuración de un circuito eléctrico que produce un voltaje de salida menor o igual al voltaje de entrada. Este voltaje de salida se puede representar como una fracción del voltaje de entrada. Uno de los divisores de tensión más simples es un circuito compuesto por dos resistencias en serie, como se puede observar en la Ilustración 6, en el que el voltaje de salida sigue la siguiente fórmula:

$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in}$$

Formula de voltaje de salida en relación con el voltaje de entrada en un divisor de tensión.

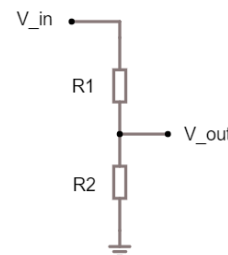


Ilustración 6: Divisor de tensión

Resistencia pull-up:

Son resistencias normales puestas de una forma determinada en un circuito eléctrico. Normalmente esta configuración se usa en el ámbito de la transmisión de datos y su función es establecer un nivel lógico de voltaje a la entrada de un circuito cuando este está en reposo reduciendo la influencia del ruido del circuito que podría dar errores en la transmisión. [12]

En la Ilustración 7 se puede observar el funcionamiento del circuito, en el que tenemos una resistencia en serie con un interruptor, que al activarse daría un voltaje lógico de salida diseñado para ese circuito y cuando está desactivado otro, si no tuviéramos la resistencia, el voltaje que leeríamos a la salida cuando el interruptor está en reposo podría verse afectado por el ruido del circuito y podríamos llegar a leer voltajes erróneos.

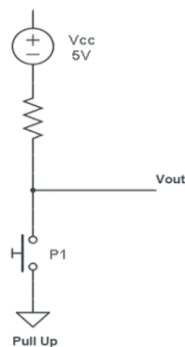


Ilustración 7: Resistencia pull-up

Convertor analógico-digital:

Es un dispositivo electrónico que transforma una señal analógica, normalmente tensión, en un valor digital. Estos aparatos funcionan dentro de un rango de voltajes, con un determinado intervalo de muestreo y una resolución determinada, que nos indicará con que precisión podrán medir el voltaje que se les facilita en la entrada. Esto quiere decir que siempre se va a perder algo de información de la señal analógica medida.

Sensor de corriente:

Es un dispositivo electrónico capaz de medir una corriente eléctrica y generar una salida con el valor medido. Funcionan usando la caída de voltaje o los campos magnéticos producidos cuando pasa una corriente por un circuito.

Generalmente se usan dos tipos de sensores de corriente, según qué tipo de detección usen pueden ser:

- **DetECCIÓN directa:** Se mide la caída de voltaje en un elemento cuando la corriente fluye a través de él.
- **DetECCIÓN indirecta:** Usa la medición de un campo magnético cuando se hace pasar una corriente eléctrica a través de un elemento.

Amplificador operacional:

Es un dispositivo electrónico lineal capaz de ampliar una señal. Estos dispositivos se acercan mucho a lo que sería una amplificación ideal de una señal de corriente continua, por lo que suelen ser usados para acondicionar señales, filtrarlas, o realizar operaciones matemáticas. [13]

Los amplificadores operacionales son generalmente dispositivos de tres terminales más dos de alimentación, como se puede observar en la Ilustración 8. Dos de ellos son entradas para la señal que se desee tratar y la otra es la salida de la señal.

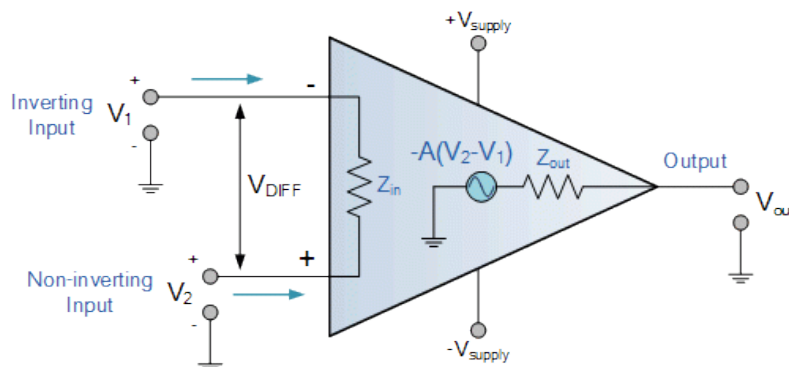


Ilustración 8: Amplificador operacional

En un amplificador operacional lineal, la salida es igual a la señal de entrada multiplicada por la ganancia del amplificador, generalmente representada con la letra A.

Raspberry Pi 4:

Es un pequeño ordenador de la marca Raspberry Pi, es del tamaño de una tarjeta de crédito. Cuenta con un SoC (Sistema en Chip, del inglés *System on a Chip*) que lleva un procesador Broadcom BCM2711, es un procesador con arquitectura ARM, de cuatro núcleos, del estilo al que llevan los teléfonos móviles. Este SoC dependiendo del modelo puede llevar 2 GB, 4 GB u 8 GB de RAM DDR4. [14] En la Ilustración 9 se ven las características comerciales más destacadas de estas placas.

Además, dispone de un módulo wifi de 2.4 GHz y 5.0 GHz, y una gran cantidad de puertos de entrada y salida contando con 40 pines GPIO. Y en la Ilustración 10 podemos ver un esquema con las especificaciones técnicas de la Raspberry Pi 4.

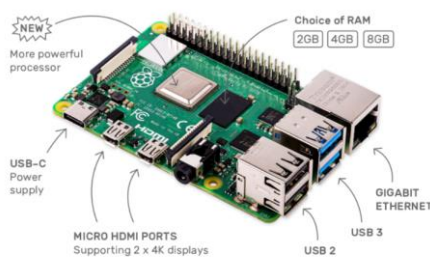


Ilustración 9: Raspberry Pi 4

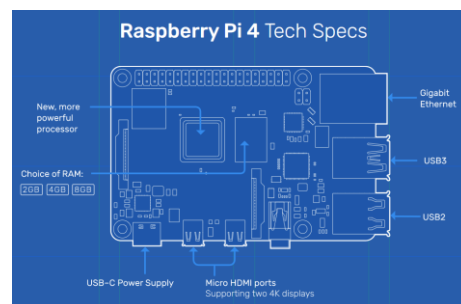


Ilustración 10: Raspberry Pi 4 Specs

Arduino Nano Every:

Es una pequeña placa de la marca Arduino, como se puede apreciar en la Ilustración 11, con un microcontrolador ATmega4809 y una gran cantidad de pines de entrada y salida. [15]

Su principal uso es la ejecución automática de tareas programadas, que se ejecutan normalmente según la información recogida a través de sensores y realiza acciones usando actuadores, ambos conectados a sus pines de entrada y salida.

Para realizar la programación de este microcontrolador se usa el lenguaje de programación de Arduino, que es un derivado del lenguaje C y C++. En la actualidad hay muchos IDEs que facilitan la programación de estos microcontroladores y la carga del código en su memoria, como pueden ser el ide de Arduino o el editor de código Visual Studio Code de Microsoft.

Entre algunas de sus características están sus 8 pines de entrada analógica conectados a convertidores analógico-digital de 10 bits y sus controladores para conexiones UART, SPI e I2C.



Ilustración 11: Arduino Nano Every

Firebase:

Es un conjunto de herramientas ofrecidas por Google para facilitar el desarrollo de aplicaciones web, móviles y videojuegos. [16]



Ilustración 12: Logo Firebase

Entre los productos que ofrece algunos de los más destacados son:

- **Cloud Firestore:** Es una base de datos NoSQL, basada en documentos con grandes capacidades de sincronización, consultas, y fácil almacenamiento de datos. [17]
- **Firebase Authentication:** Es un sistema de autenticación creado por Google, que busca gestionar de forma fácil y segura la autenticación de los usuarios de las aplicaciones de Firebase. [18]
- **Google Analytics:** Es un servicio que nos permite obtener analíticas y datos de uso de nuestras aplicaciones de Firebase. [19]
- **Firebase Hosting:** Es el servicio de alojamiento de páginas web, permite almacenar el código de la página web y obtener un enlace para acceder a ella. [20]

4. Técnicas y herramientas

4.1. Entorno de desarrollo software

Conjunto de aplicaciones usadas para facilitar la creación del código de las aplicaciones, la realización de pruebas, copias de seguridad y elaboración de la documentación.

Visual Studio Code:

El editor de código multiplataforma de Microsoft. Es un editor de código abierto que permite la programación de forma nativa en la mayoría de los lenguajes populares y en prácticamente todos los lenguajes usando extensiones. Es muy versátil y permite organizarse en espacios de trabajo para separar las fases del desarrollo de manera más cómoda. Además, permite configurar tareas para, por ejemplo, enviar el código a otros dispositivos o lanzar pruebas externas. [21]

Se ha escogido este editor de texto por la familiarización con él desde prácticamente el inicio de la carrera. En la Ilustración 13 podemos ver una captura del espacio de trabajo de Visual Studio Code durante el desarrollo del proyecto.

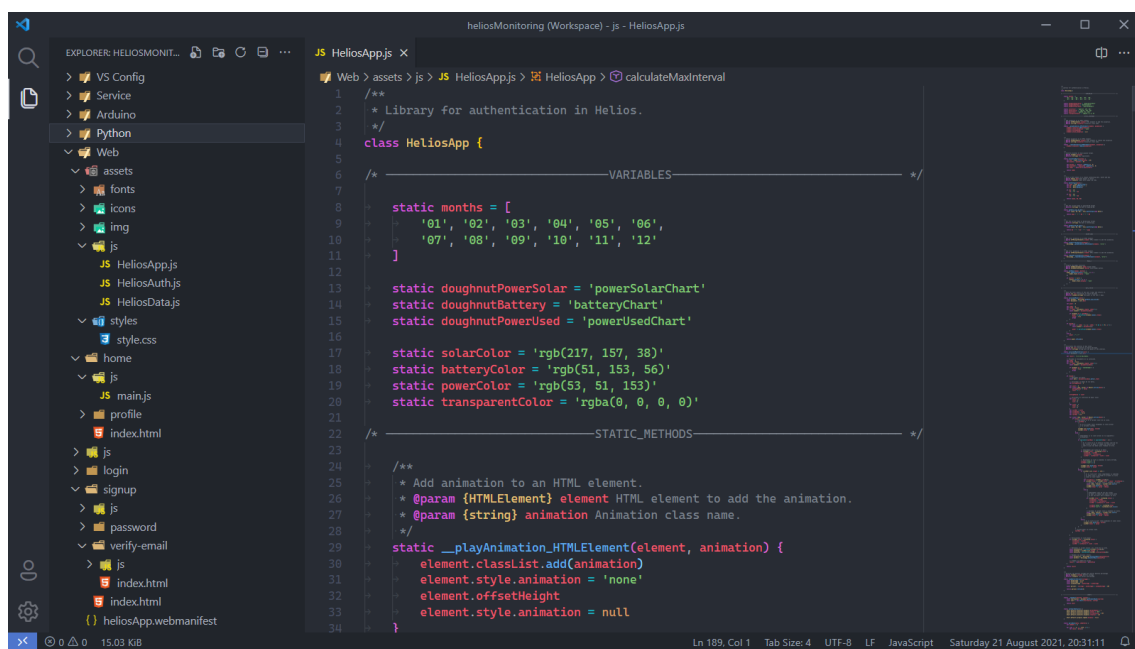


Ilustración 13: Visual Studio Code

OneDrive:

Es la nube de almacenamiento personal de Microsoft, se ha usado durante el desarrollo para mantener una copia de seguridad constante del proyecto y los documentos relacionados con este trabajo de fin de grado. [22]

Navegador Microsoft Edge:

Es el navegador predeterminado de Windows 10. Está basado en la versión de código abierto del navegador de Google Chromium. [23]

Se ha usado para la realización de pruebas de la página web, usando la función DevTools del navegador, como se puede observar en la Ilustración 14, para ver los posibles errores de la consola de JavaScript y realizar pruebas en la interfaz. [24]

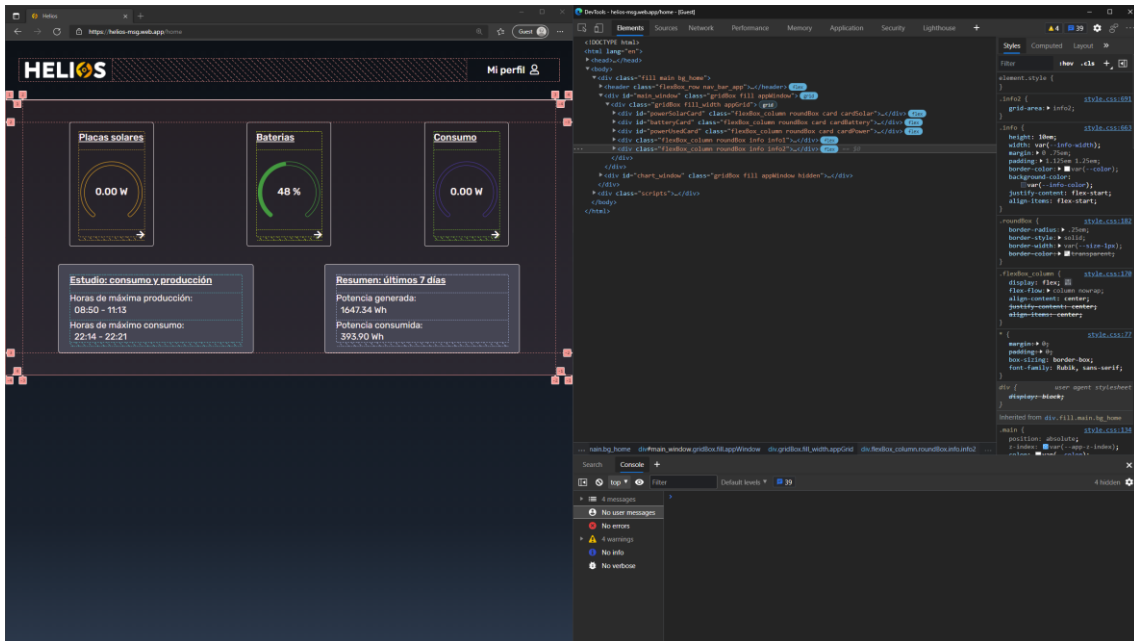


Ilustración 14: Navegador Microsoft Edge y función DevTools

PyDoc:

[PyDoc](#) es un módulo de Python para generar documentación para el código escrito en Python, permite generar archivos de documentación a partir de los comentarios del código elaborados por el programador. La documentación generada se presenta en forma de páginas web en HTML donde se pueden ver las funciones y clases documentadas. [25]

Se ha usado para generar la documentación del programa de la monitorización de la instalación.

JSDoc:

[JSDoc](#) es una API para crear documentación de los archivos de código de JavaScript, permite generar archivos de documentación a partir de los comentarios del código elaborados por el programador. La documentación generada se presenta en forma de páginas web en HTML donde se pueden ver las funciones y clases documentadas. [26]

Se ha usado para documentar el código de la aplicación web.

4.2. Entorno de desarrollo hardware

Conjunto de herramientas usadas para el desarrollo e investigación con el hardware de monitorización.

Tinkercad circuits:

[Tinkercad](#) es una herramienta gratuita de simulación de circuitos creada por Autodesk. [27]

Se ha usado durante el desarrollo del proyecto para simular ciertos circuitos y ver su viabilidad. Como se puede ver en la Ilustración 15 donde se está comprobando el voltaje de salida de unos amplificadores operacionales.

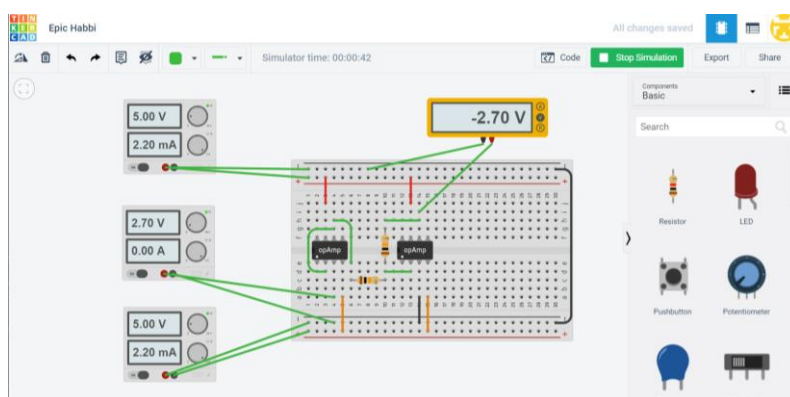


Ilustración 15: Tinkercad circuits

CircuitJs/Falstad:

[CircuitJs](#) es otro simulador gratuito de circuitos, desarrollado por Paul Falstad, con posibilidad de ver gráficas de tiempo del voltaje y corriente. Tiene dos versiones muy ligeras tanto para navegador como de escritorio. Dispone de una licencia GPL-2.0. [28]

Se ha usado con el mismo propósito que Tinkercad, pero buscando un grado más fino de precisión y simulación. En la Ilustración 16 se está analizando la salida y el tiempo que se tarda en acondicionar la señal a la salida del inversor para poder leerla.

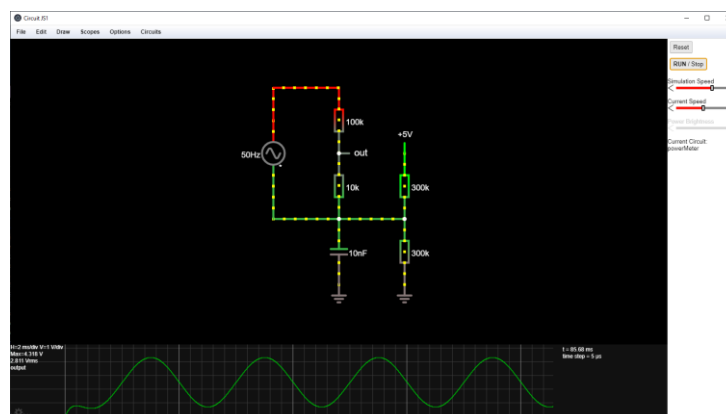


Ilustración 16: CircuitJS

Multímetro digital:

Es un aparato para realizar mediciones de diferentes magnitudes, generalmente suelen poder realizar mediciones de voltajes en corriente alterna y corriente continua, detectar la continuidad en circuitos eléctricos, medir corrientes eléctricas y frecuencias de la corriente alterna.

La realización de mediciones puntuales tanto de corriente alterna como de corriente continua se han llevado a cabo usando un multímetro, con el objetivo de verificar los resultados teóricos obtenidos con las aplicaciones. El multímetro que se ha usado para estas tareas se puede ver en la Ilustración 17.



Ilustración 17: Multímetro digital

Amperímetros y voltímetros analógicos:

Los amperímetros y voltímetros analógicos son medidores de corriente y voltaje respectivamente, que funcionan sin procesar la información de la medida de forma digital, con ello consiguen crear una representación física de la medida usando un actuador calibrado para la tarea.

Se han usado para llevar una medición paralela de los voltajes y corriente de las placas solares y comprobar que los valores leídos por los sensores eran correctos. Los medidores usados durante el proyecto son los que se pueden observar en la Ilustración 18.

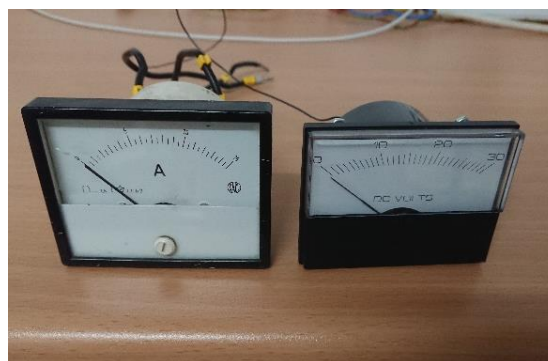


Ilustración 18: Amperímetro y voltímetro analógicos

Medidor de potencia:

Es un aparato que registra la potencia usada en un punto concreto de una instalación eléctrica. Dispone de varios modos para ver la potencia actual, la máxima el total consumido en un intervalo de tiempo e información sobre la red, como puede ser la frecuencia o el voltaje.

Se ha usado para comprobar el correcto funcionamiento del cálculo de la potencia de la instalación. El medidor de potencia usado para las pruebas es el que se ve en la Ilustración 19.



Ilustración 19: Medidor de potencia

Osciloscopio:

Es un dispositivo que muestra la forma de onda de una corriente eléctrica cuando se conectan sus terminales a dos puntos del circuito, entre sus características principales están el número de canales que pueden leer simultáneamente y el intervalo de muestreo que cuanto más alto sea mayores será la fiabilidad de la onda representada.

Su uso ha servido para la investigación de las formas de onda a la salida del inversor y ver la señal acondicionada del inversor, con el osciloscopio podíamos ver en el tiempo la forma de onda pudiendo ver los voltajes máximo y mínimo con los que íbamos a trabajar, como se puede apreciar en la Ilustración 20.

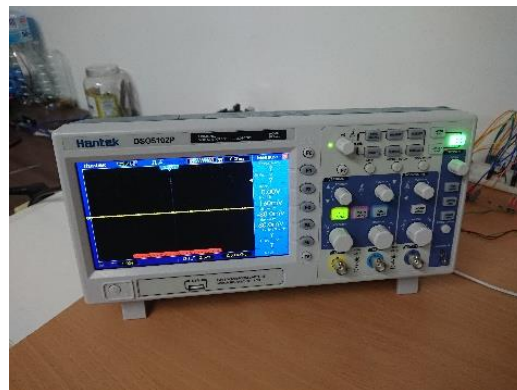


Ilustración 20: Osciloscopio

Fuente de alimentación:

Una fuente de alimentación es un dispositivo capaz de generar una variedad de tensiones y corrientes diferentes, con el objetivo de poner a prueba diferentes circuitos diseñados en el laboratorio.

Para alimentar los circuitos durante las pruebas se ha utilizado una fuente de alimentación de 30 V y 5 A, que se puede ver en la Ilustración 21.



Ilustración 21: Fuente de alimentación

Placas de prototipo:

Las placas de prototipo se usan para diseñar circuitos eléctricos, permitiendo conectar los diferentes componentes eléctricos de un circuito sin necesidad de soldaduras, para ello usan una serie de railes horizontales y verticales en los que se conectan los componentes.

Los circuitos diseñados se han montado sobre placas de prototipo, como la que se puede observar en la Ilustración 22, facilitando la modificación y nuevos diseños.

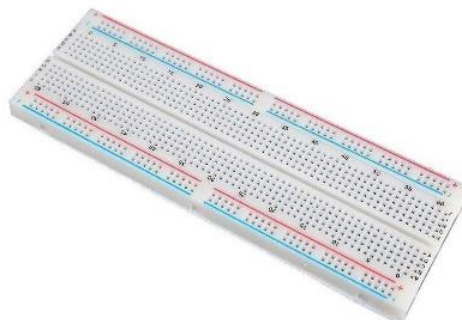


Ilustración 22: Placa de prototipos

4.3. Librerías y *frameworks*

Se han usado una librería y un *framework* para facilitar la realización de un par de tareas en la aplicación web.

ChartJS:

[ChartJS](#) es una librería gratuita de JavaScript para la creación de gráficos de datos, desarrollada por el equipo de Chart.js. Dispone de una licencia MIT. [29]

También se exploraron otras opciones como D3.js [30], pero finalmente se eligió ChartJS dado que está orientada crear representaciones gráficas más genéricas y encajaba mejor con el diseño de la aplicación.

Con esta librería se han elaborado gráficas de líneas y de donuts en la aplicación web.

MCDatepicker:

[MCDatepicker](#) es un *framework* gratuito para la creación de un calendario de selección de fechas. Ha sido desarrollado por Mike Cojocari. Dispone de licencia MIT. [31]

Su funcionalidad ha sido útil en la aplicación web para la selección de fechas de los históricos de datos.

4.4. Lenguajes de programación

Los lenguajes de programación usados tanto en la aplicación web como en la programación de los microcontroladores y la Raspberry Pi han sido los siguientes:

C:

Es un lenguaje de programación de propósito general orientado a la programación en sistemas operativos.

Se ha usado en la programación de los Arduino Nano Every, para crear el medidor de potencia a la salida del inversor y comunicación de estos datos a través de I2C.

Python:

Es un lenguaje de programación de propósito general, cuyo diseño está centrado en la legibilidad del código.

Se ha usado en la creación del programa de monitorización de la instalación y subida de datos a la base de datos Firestore.

Systemd-unit-file:

Es el lenguaje usado para la configuración de los servicios Linux, se ha usado para la creación de un servicio que lanzará el programa de monitorización de la instalación cada vez que se encienda la Raspberry Pi 4 o reiniciará el programa si hay errores.

HTML:

Lenguaje de Marcas de Hipertexto, del inglés *HyperText Markup Language*, es un lenguaje de utilizado para crear la estructura de páginas web. Se ha usado en el diseño de toda la aplicación web.

CSS:

Hojas de Estilo en Cascada, del inglés *Cascading Style Sheets*, es un lenguaje usado para aplicar estilos a un documento HTML. CSS describe como deberían mostrarse los elementos HTML. Se ha usado en la aplicación web para definir el estilo de esta.

JavaScript:

Es un lenguaje de programación orientado a objetos, implementado en la mayoría de los navegadores, por lo que es muy útil para añadir nuevas funcionalidades a páginas web.

Se ha utilizado en el proyecto para la gestión de las funcionalidades de la aplicación web.

4.5. Herramientas CASE

Son unas herramientas informáticas, cuyo nombre proviene del acrónimo en inglés de Ingeniería del Software Asistida por Computadora, se han usado durante todo el proyecto para elaborar la documentación técnica.

UML:

Lenguaje Unificado de Modelado, del inglés *Unified Modeling Language*, es un lenguaje de propósito general usado para el modelado de sistemas software. Permite crear una representación formal del sistema de forma gráfica.

EZEstimate:

Es un programa de estimación de coste y esfuerzo del desarrollo de un sistema. Para realizar estas estimaciones se usa el método UCP (Puntos de Caso de Uso, del inglés Use Case Point) a partir de los casos de uso.

Microsoft Project:

Es un software de administración del calendario de desarrollo de un proyecto, permite realizar asignación de recursos a tareas, administrar las cargas de trabajo y llevar un seguimiento del progreso. [32]

Visual Paradigm:

Es un programa que permite realizar diagramas del lenguaje UML para la creación de la documentación técnica de un proyecto informático. [33]

5. Aspectos relevantes del desarrollo del proyecto

En este apartado se hará un resumen de las partes más relevantes del desarrollo del proyecto.

5.1. Marco de trabajo

Durante el desarrollo de este proyecto de fin de grado se han seguido las indicaciones del Proceso Unificado. Es un marco de trabajo genérico y algunas de las características que presenta son:

- **Dirigido por casos de uso:** Los casos de uso se utilizan para describir las funcionalidades de algunas partes del sistema y son de gran utilidad en la realización de las iteraciones de las diferentes fases del Proceso unificado.
- **Basado en componentes y centrado en la arquitectura:** Especifica como se debe estructurar el sistema, pero puede variar a lo largo del desarrollo para adaptarse a nuevos casos de uso.
- **Iterativo e incremental:** El desarrollo es un proceso largo y tedioso, por lo que es aconsejable dividir el trabajo en varias fases e iteraciones. Al finalizar cada interacción se revisa el sistema.
- **Uso de UML:** Se usa en las representaciones y diagramas del Proceso Unificado.

El Proceso Unificado tiene cuatro fases diferentes (Inicio, Elaboración, Construcción y Transición), con ellas se organiza el desarrollo del sistema. Cada una de las fases está dividida en iteraciones cuyas tareas se agrupan en una serie de disciplinas (Modelado del negocio, Requisitos, Análisis, Diseño, Implementación y Pruebas). En la Ilustración 23 se pueden apreciar todos estos conceptos, además se observa cómo sería una distribución normal de la carga de trabajo de cada disciplina a lo largo del desarrollo del proyecto.

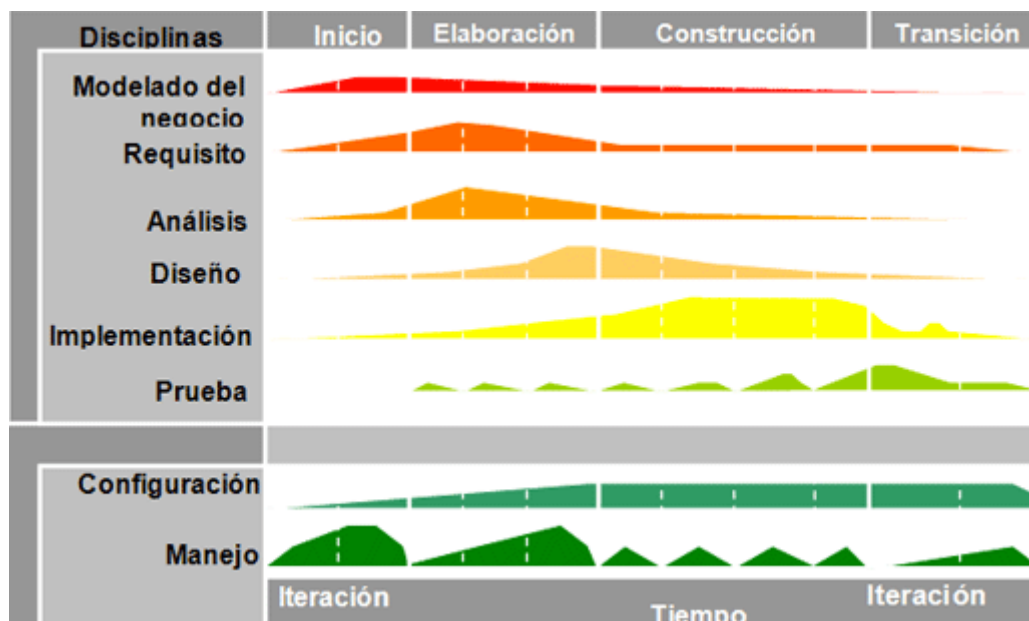


Ilustración 23: Proceso unificado

5.2. Estimación de coste y esfuerzo

La estimación de esfuerzo del proyecto se ha realizado una vez especificados los requisitos del sistema. Estos requisitos pueden variar a lo largo del desarrollo, pero como en esta fase lo que buscamos es obtener una idea del coste y duración del proyecto, los usaremos para un primer cálculo.

Para realizar estos cálculos se ha usado la técnica de Puntos de Caso de Uso (UCP), con la que hemos obtenido unos valores, que hemos introducido posteriormente en la herramienta EZEstimate. Esta herramienta la hemos usado en la asignatura Gestión de Proyectos para calcular estimaciones de costes y esfuerzo.

En la se puede ver una captura de la herramienta y el resultado obtenido.

The screenshot shows the EZEstimate software interface. The main window title is "EZEstimate - C:\Users\miguel\OneDrive\Miguel\Estudios\3-IngInformatica\8-Cuatri\TFG\helios\inso\ez.ezp". The interface is divided into several sections:

- Module:** A dropdown menu shows "Gestión de la monitorización del sistema". Below it are "Add Module" and "Delete" buttons.
- Summary:** Displays "Total Modules" as 4. There is an "Excel Report" button labeled "Generate Report". Below this, "Use cases" are listed: Simple (20), Average (0), and Complex (0). "Actors" are listed: Simple (0), Average (1), and Complex (2).
- Add Actor / Use case:** A form with "Actor / Use case Name", "Select Type" (dropdown set to "Usecase"), and "Complexity" (dropdown set to "Simple"). An "Add" button is present.
- Tech / Env Factors:** Two buttons: "Set Tech Factor" and "Set Env Factors".
- Estimation Summary:** A series of input fields and calculated values:
 - UAW: 8
 - UUCW: 100
 - UUCP = UAW + UUCW: 108
 - TFactor: 29
 - EFactor: 20
 - TCF = 0.6 + (.01*TFactor): 0,89
 - EF = 1.4 + (-0.03*EFactor): 0,8
 - UCP = UUCP*TCT*EF: 76,896
 - Total Effort@ 10 Hrs/UCP: 768,96**
- Use case / Actor List:** A table with columns: Id, Module, Type, Name, and complexity. It lists various use cases and actors with their respective complexities.

Ilustración 24: Resultado estimación de costes y esfuerzo

La herramienta nos ha indicado un valor de 768,96 horas de persona, que aproximadamente será el esfuerzo que se deberá realizar para completar el proyecto. La realización de los cálculos y los valores introducidos en la herramienta se pueden ver con más detalle en el *Anexo I – Planificación temporal*.

5.3. Planificación temporal

Una vez realizada la estimación de esfuerzo se ha realizado la planificación temporal del desarrollo del proyecto. En ella se puede ver para cada tarea, su duración y la carga de trabajo asignada.

La realización de la planificación temporal se ha realizado con la herramienta Microsoft Project. [32]

Con la herramienta hemos obtenido un diagrama de Gantt que nos indica la duración de las tareas con barras horizontales, además de las relaciones de precedencia entre tareas y el camino crítico del desarrollo del sistema. Un ejemplo de esto se puede observar en la Ilustración 25.

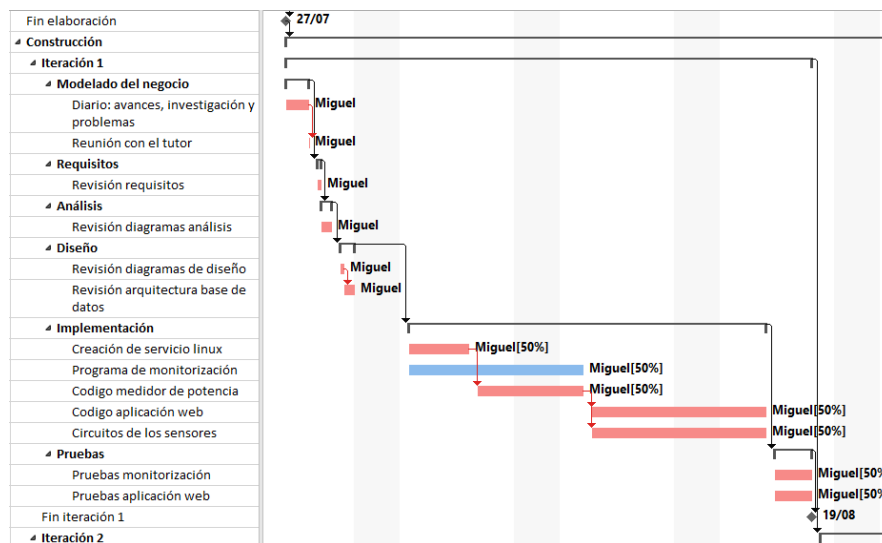


Ilustración 25: Diagrama de Gantt y camino crítico

Una vez finalizada la planificación se ha obtenido un resumen de la duración de cada fase del desarrollo del Proceso Unificado. Como se puede ver en la Ilustración 26.

▷ Inicio	15,5 days
Fin inicio	0 days
▷ Elaboración	25,25 days
Fin elaboración	0 days
▷ Construcción	36 days
Fin construcción	0 days
▷ Transición	6 days
Fin transición	0 days

Ilustración 26: Duración fases, planificación temporal

En la Ilustración 26 se observa la duración de cada fase en días con un total de 82,75 días de trabajo, que transformado a horas usando la jornada laboral definida (10 horas/día) da un total de 827,5 horas de trabajo que es un valor bastante similar al obtenido en la estimación de coste y esfuerzo.

Esta planificación temporal se encuentra más detallada en el *Anexo I – Planificación temporal*.

5.4. Especificación de requisitos

La especificación de requisitos de software se ha realizado siguiendo la metodología de Duran y Bernárdez y usando la herramienta Visual Paradigm para la elaboración de diagramas siguiendo el lenguaje UML.

A continuación, se mostrará un pequeño resumen del desarrollo de esta fase, pero si se desea profundizar más en la realización de este apartado se puede consultar el *Anexo II – Especificación de requisitos*.

5.4.1. Participantes y actores

La primera tarea que ha realizado ha sido definir los participantes del proyecto. En la Tabla 1 se puede observar un ejemplo.

Participante	Miguel Sánchez González
Organización	Universidad de salamanca
Rol	Autor
Es desarrollador	Sí
Es cliente	No
Es usuario	No
Comentarios	Ninguno

Tabla 1: Ejemplo participante

Posteriormente, se han definido los actores y los objetivos del sistema, definiendo con las tablas de la metodología de Duran y Bernárdez, los atributos de cada uno. En la Ilustración 27: Diagrama de actores podemos ver el diagrama de actores diseñado con la herramienta Visual Paradigm.

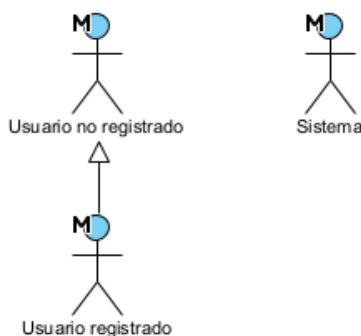


Ilustración 27: Diagrama de actores

5.4.2. Objetivos, requisitos de información, funcionales y no funcionales

Después de definir los actores y objetivos del sistema, se ha realizado la elicitación de requisitos del sistema, separándolos en requisitos de información, requisitos no funcionales y requisitos funcionales.

Los requisitos no funcionales y de información pueden describirse meramente usando las tablas de Durán y Bernárdez, pero en el caso de los requisitos funcionales se han diferenciado varios paquetes para separar las funcionalidades del sistema y poder trabajar en el proyecto de una forma más cómoda.

Los paquetes en los que se ha dividido el sistema son los de la Ilustración 28.

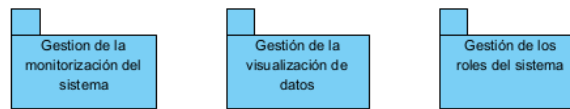


Ilustración 28: Diagrama de paquetes

Podemos ver un ejemplo de la estructura interna de uno de estos paquetes en la Ilustración 29, donde se puede apreciar los diferentes casos de uso en los que se subdivide cada paquete.

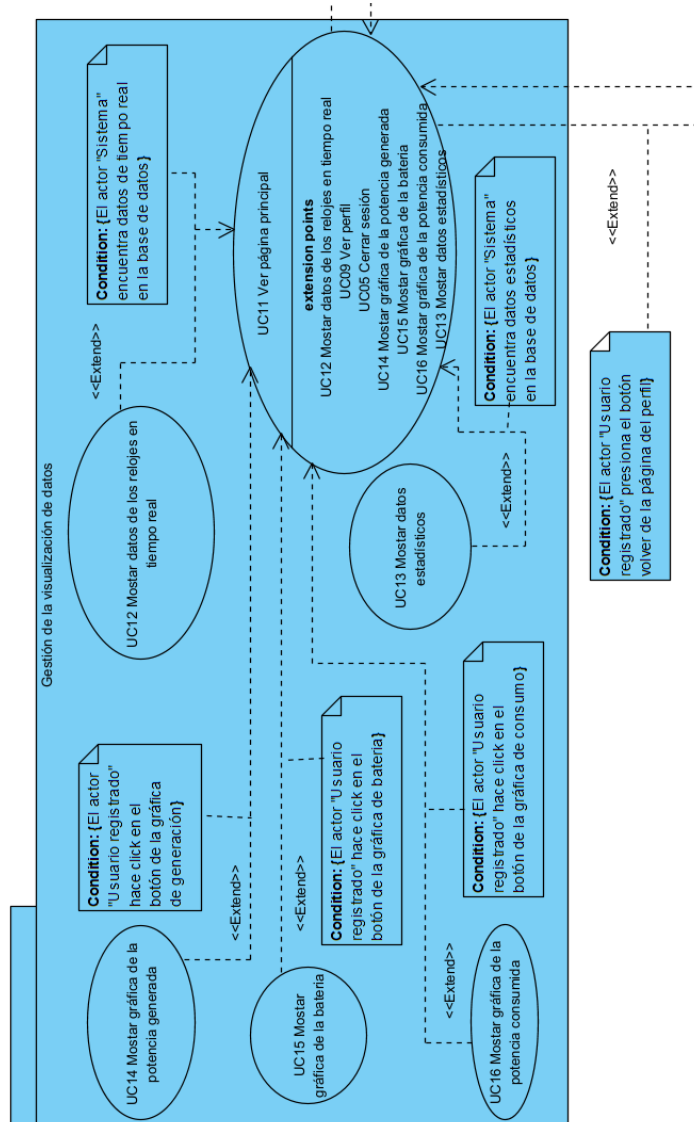


Ilustración 29: Ejemplo paquete

Una vez finalizada, la descripción de los requisitos funcionales se obtuvo un total de 20 casos de uso.

5.4.3. Matrices de rastreabilidad

Por último, se realizó la creación de las matrices de rastreabilidad para poder obtener una visión conjunta de los requisitos del sistema y sus relaciones entre sí y con los objetivos del sistema.

5.5. Análisis del sistema

Una vez realizada la especificación de requisitos, se da paso a la realización del análisis de los requisitos del sistema, que consiste en un refinamiento del trabajo realizado durante la especificación. Con el resultado del análisis se obtiene una visión más precisa del sistema, que nos servirá para realizar el diseño de este.

A continuación, se mostrará un pequeño resumen del desarrollo de esta fase, pero si se desea profundizar más en la realización de este apartado se puede consultar el *Anexo III – Análisis de requisitos*.

5.5.1. Modelo de dominio

En primer lugar, se ha desarrollado un modelo de dominio, que recoge las necesidades de almacenamiento que deberá llevar a cabo la base de datos. Con él se pueden identificar algunas de las clases conceptuales usadas en el resto del análisis. Podemos ver una imagen del modelo del dominio desarrollado en la Ilustración 30.

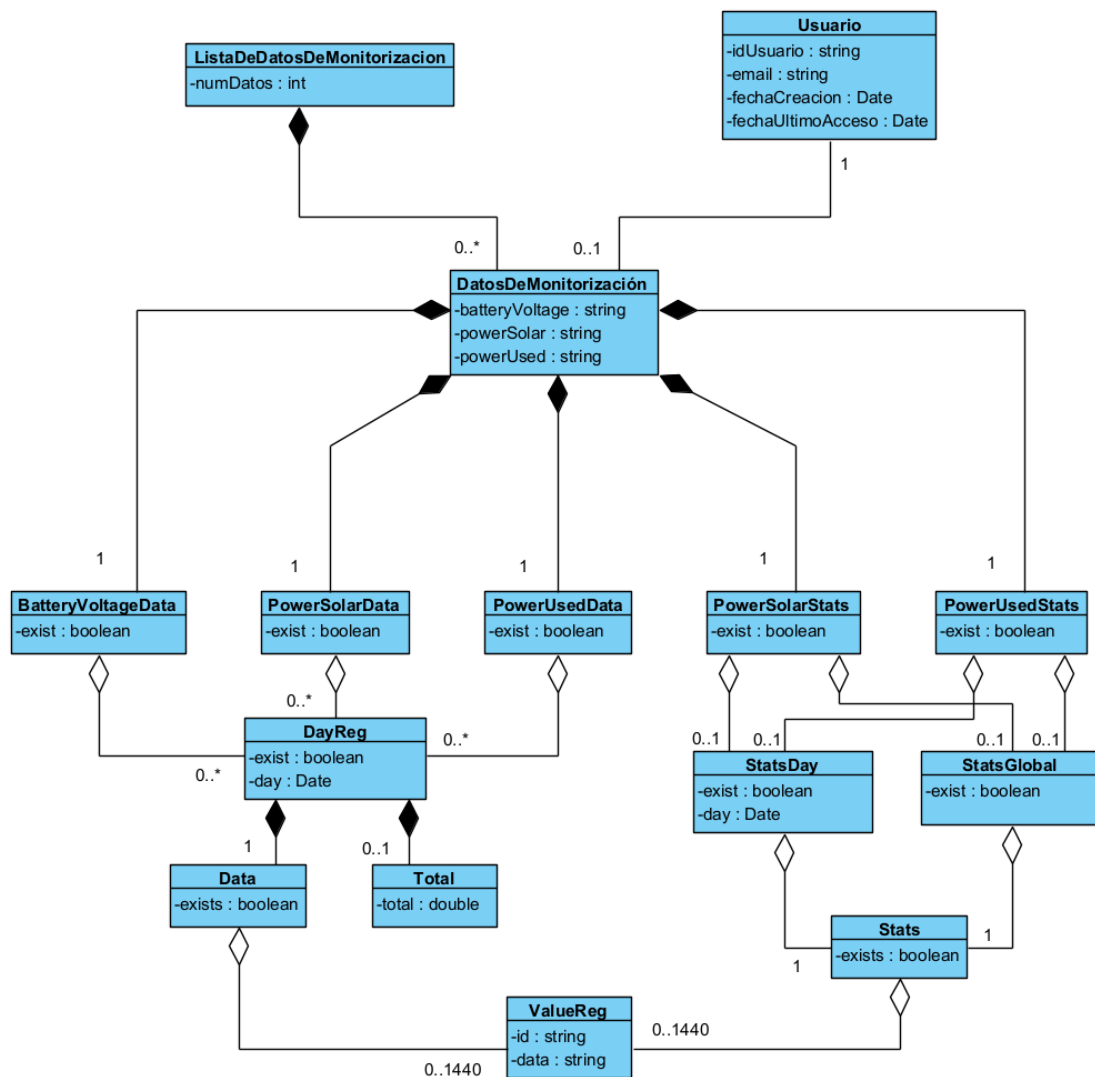


Ilustración 30: Modelo de dominio

5.5.2. Paquetes de análisis

Posteriormente se ha realizado la descomposición del sistema en paquetes de análisis, y una vez obtenidos se elabora la vista de la arquitectura del sistema que contiene todos los paquetes con sus componentes y sus relaciones. Como se puede observar en la Ilustración 31.

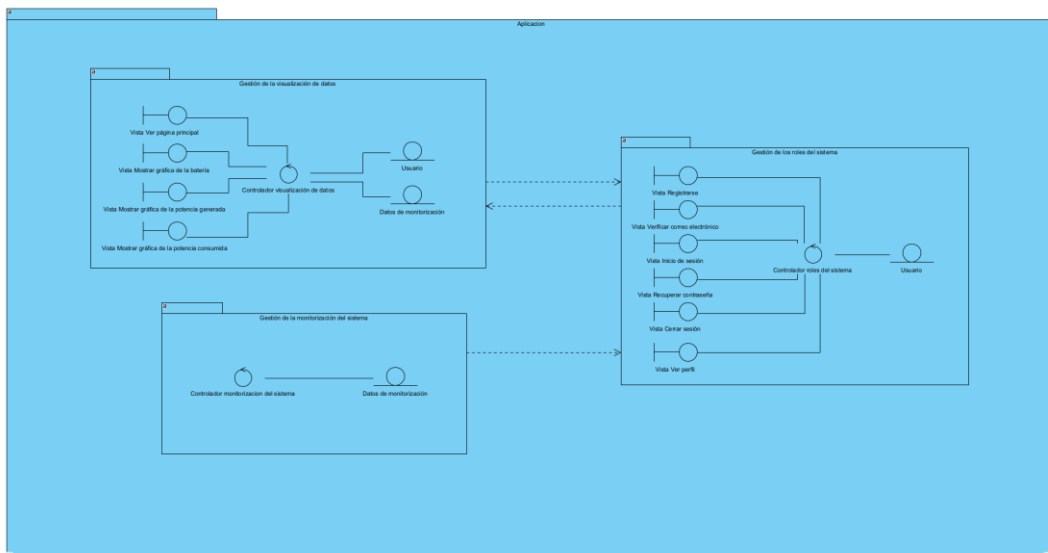


Ilustración 31: Vista de la arquitectura

5.5.3. Realización de casos de uso

Por último, se realiza la descripción de los casos de uso del análisis, donde se ve a un nivel más bajo como interactúan los diferentes objetivos entre sí, indicando los mensajes entre los componentes del sistema. Un ejemplo de esto se puede ver en la Ilustración 32.

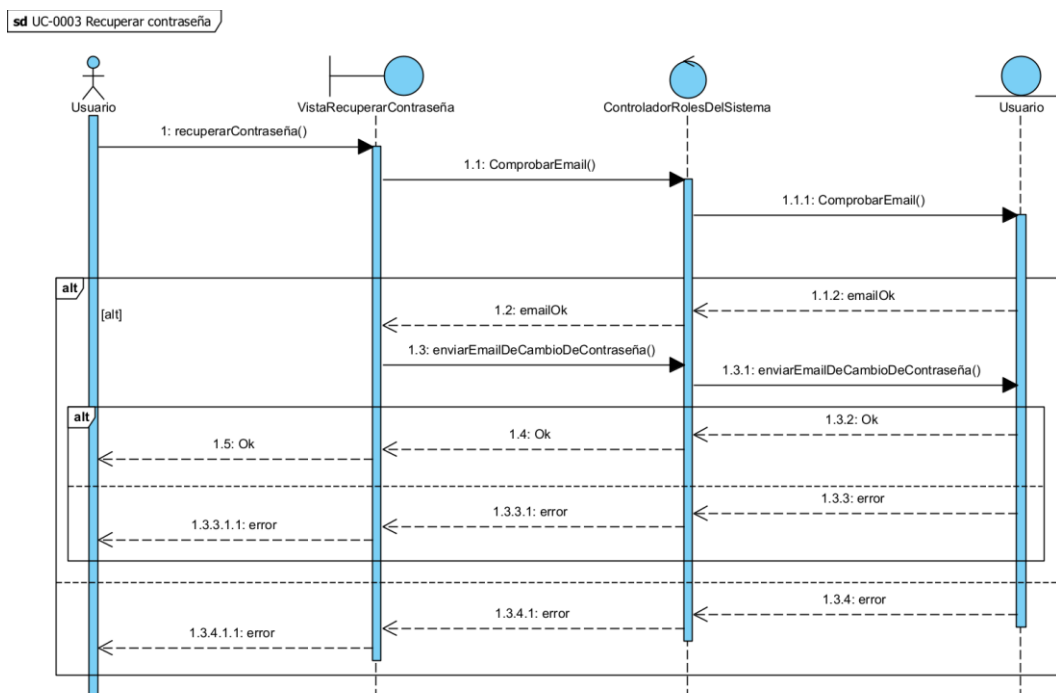


Ilustración 32: Ejemplo caso de uso análisis

5.6. Diseño del sistema

El apartado de diseño del sistema se centra en buscar una solución al problema propuesto, para, más tarde, implementarla en el proyecto. Durante esta fase se detalla la realización de los casos de uso realizados anteriormente.

A continuación, se mostrará un pequeño resumen del desarrollo de esta fase, pero si se desea profundizar más en la realización de este apartado se puede consultar el *Anexo IV – Diseño del sistema*.

5.6.1. Patrón arquitectónico

Para la realización de la fase de diseño se ha decidido usar el patrón MVC (Modelo-Vista-Controlador, del inglés Model-View-Controller), y se basa principalmente en separar la funcionalidad del sistema de la representación en la interfaz del usuario. En la Ilustración 33 se pueden ver las relaciones de los componentes de este patrón arquitectónico.

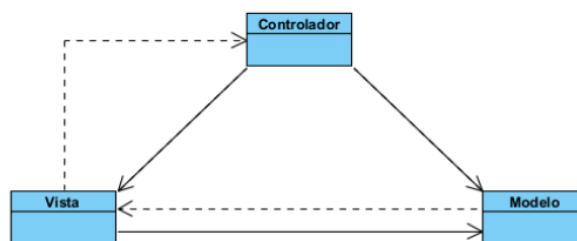


Ilustración 33: Patrón MVC

5.6.2. Vista de subsistemas y vista arquitectónica

Tras elegir el tipo de patrón arquitectónico a usar se realiza la vista de los subsistemas de diseño, en este caso siguiendo el patrón MVC, que divide el sistema en sus diferentes componentes.

Una vez conseguido esto se elabora la vista arquitectónica donde se puede ver a que parte del patrón arquitectónico pertenece cada elemento de los subsistemas de diseño, así como las relaciones entre ellos. Esto se puede ver en la Ilustración 34.

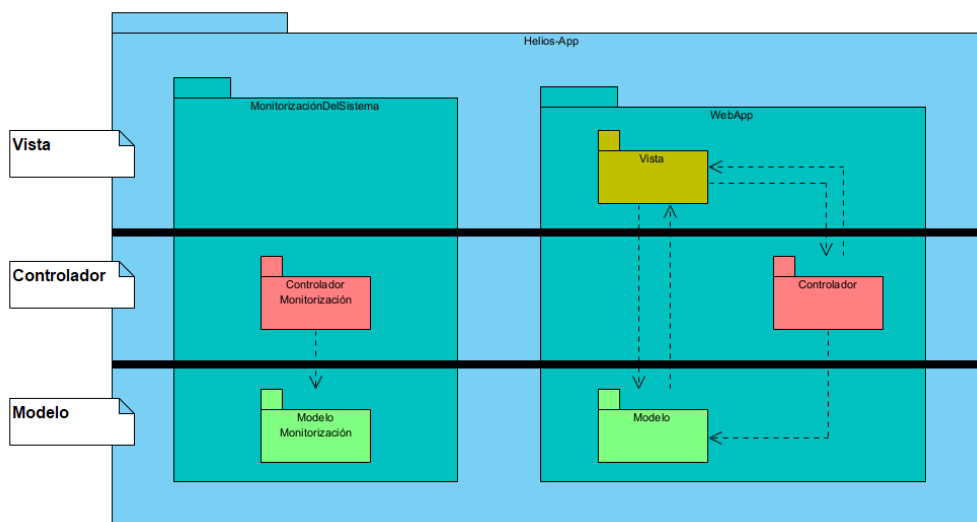


Ilustración 34: Vista arquitectónica

En la vista arquitectónica, si nos fijamos en su diagrama en la Ilustración 34, podemos diferenciar dos subsistemas principales, el de la aplicación web (WebApp), que es el encargado de interactuar con el usuario y mostrarle los datos de la monitorización y el subsistema de monitorización de la instalación (MonitorizaciónDelSistema), encargado de controlar la monitorización y subir los datos a la base de datos.

5.6.3. Clases de diseño

Con los pasos anteriores se puede ya realizar las clases de diseño que es una especificación detallada de los paquetes de cada subsistema, un ejemplo de ello es la clase de diseño del modelo del subsistema “WebApp”, visible en la Ilustración 35.

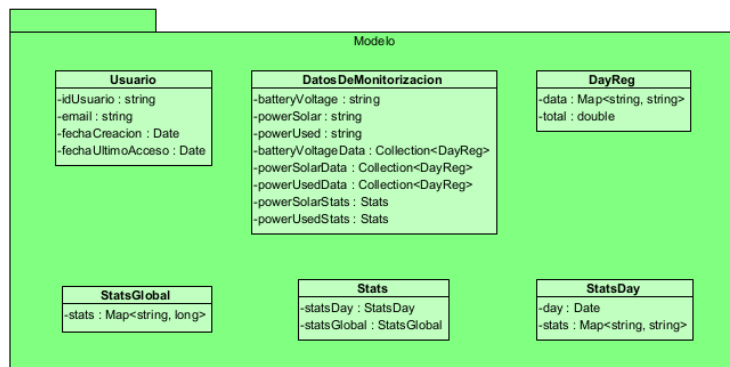


Ilustración 35: WebApp - Modelo

5.6.4. Realización de casos de uso de diseño

Tras obtener las clases de diseño del sistema, ya se tiene todo lo necesario para realizar los casos de uso en el modelo de diseño, para ello se elabora un diagrama de secuencia para cada uno de los casos de uso definidos en la especificación de requisitos. A continuación, podemos ver en la Ilustración 36 un ejemplo de uno de los muchos casos de uso realizados.

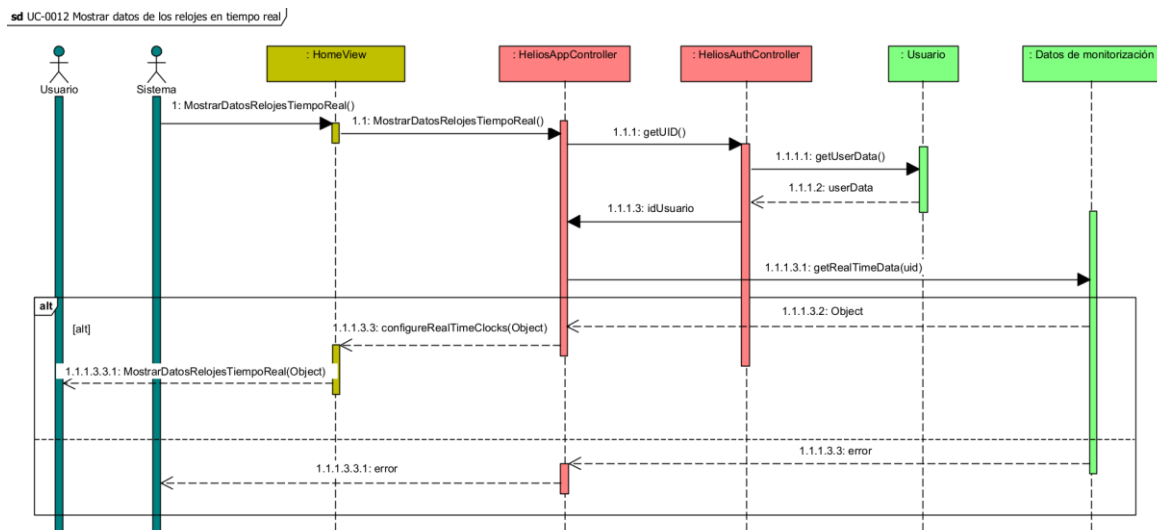


Ilustración 36: Ejemplo caso de uso

5.6.5. Modelo de despliegue

Para finalizar se realiza el modelo de despliegue del sistema donde se puede ver de forma detallada los diferentes dispositivos que componen el sistema. Cada nodo representado en la Ilustración 37 muestra un elemento hardware o software.

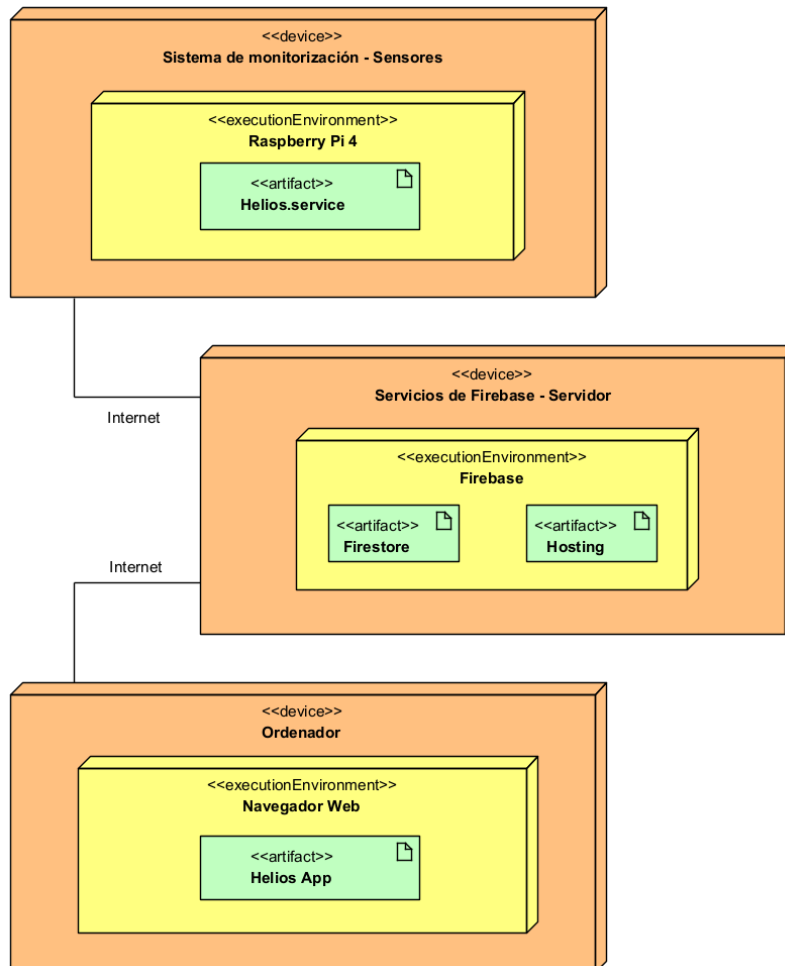


Ilustración 37: Modelo de despliegue del sistema

En el diagrama de la Ilustración 37 se aprecian nodos:

- **Sistema de monitorización - Sensores:** Es el conjunto de sensores y su controlador (Raspberry Pi 4), que realizan la monitorización de los datos de la instalación y se encarga de subirlos a la base de datos Firestore.
- **Servicios de Firebase - Servidor:** Este nodo refleja los servidores de Google que alojan nuestra base de datos y la página web y se encargan de realizar las comunicaciones de los datos.
- **Ordenador:** Es el dispositivo final del usuario, donde se visualizará la aplicación web. Puede ser cualquier tipo de ordenador, siempre y cuando disponga de un navegador web, que es el requisito mínimo para ejecutar la aplicación.

5.6.6. Diseño y estudio de la base de datos

Para el diseño de la base de datos se han tenido en cuenta una serie de factores, el primero de ellos han sido las limitaciones propias de la base de datos Firestore, que, con el plan gratuito, permite un máximo de 20k operaciones de escritura, 50k operaciones de lectura y 20k operaciones de eliminación, como se puede ver en la Ilustración 38.

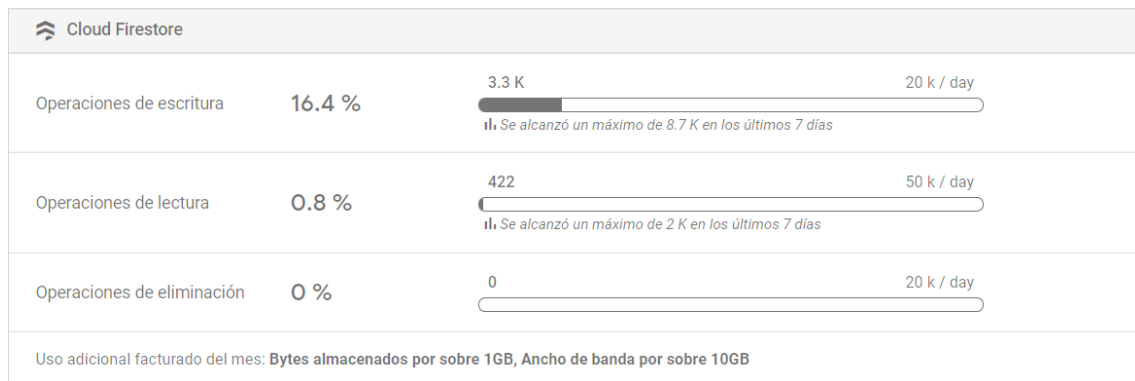


Ilustración 38: Cuotas Firestore

Inicialmente se plantea realizar las lecturas de cada sensor cada minuto y enviarlas a la base de datos, eso nos da un total de $60 \times 24 = 1440$ minutos u operaciones de escritura en un día para cada sensor, como en nuestro caso tenemos 3 sensores realizaríamos 4320 operaciones, que se queda bastante lejos del límite de Firebase. El tiempo de una lectura de los sensores por cada minuto está bastante bien para el histórico de datos, pero en cuanto a la gestión de los datos en tiempo real se queda un poco corta, por lo que se decidió realizar una lectura cada 25 segundos y enviarla a la base de datos, esto nos daba un límite de 3456 lecturas por sensor al día, que en total se traduce en unas 10000 operaciones de escritura, para los tres sensores, por día.

Adicionalmente para aumentar la precisión de algunos datos, como es el caso de el voltaje de la batería o de las placas solares, que no varía mucho en el tiempo, en el intervalo de 25 segundos, se realizan 100 lecturas y se calcula su media para finalmente enviar el valor medio a la base de datos.

5.7. Implementación

La implementación del sistema ha sido la parte que más tiempo ha llevado y a la que más recursos se han dedicado. Durante toda esta fase se han usado las herramientas descritas en el apartado 4 Técnicas y herramientas. La implementación del proyecto se ha realizado siguiendo los resultados obtenidos en la fase de Diseño del Sistema, visible en el Anexo IV – Diseño del sistema.

La implementación se ha dividido en dos grandes grupos:

- **Aplicación web:** En este apartado se ha llevado a cabo la elaboración de las páginas que componen la aplicación web, asignando funcionalidades a los elementos de las vistas e interfaces y creando las funciones para tratar los datos obtenidos de la base de datos.
- **Sistema de monitorización:** Se ha desarrollado un sistema de sensores que miden los valores de la instalación fotovoltaica, para ello se ha usado una Raspberry Pi 4 que actúa como controlador general de los sensores pidiendo datos en cada momento, y una vez adecuados sus valores los envía a la base de datos. En lo relativo a los sensores, se han diseñado los circuitos para adecuar las señales con el objetivo de poder realizar su lectura con los convertidores analógico-digital.

5.7.1. Implementación del software

Tanto el código de la aplicación web como el de la monitorización del sistema, se han desarrollado dentro de un mismo espacio de trabajo de Visual Studio Code, teniendo acceso a los diferentes programas que estaban desarrollando.

La estructura del espacio de trabajo es la que se puede apreciar en la Ilustración 39:

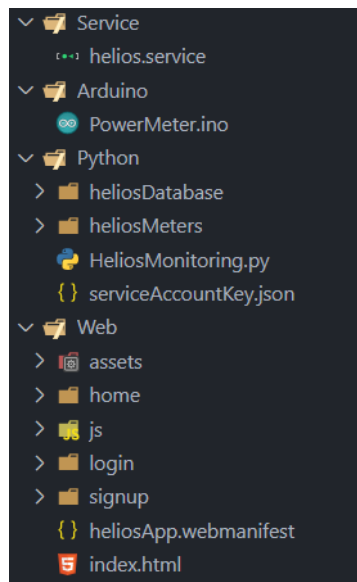


Ilustración 39: Estructura espacio de trabajo

Si se desea obtener más información sobre la estructura del espacio de trabajo y los componentes de cada paquete, se recomienda ver el *Anexo V – Manual del programador*.

Proyecto de Firebase:

El comienzo de la implementación software se inició con la configuración del proyecto de Firebase, donde se indicó que iba a ser una Aplicación Web, se realizó la configuración de la base de datos Firestore y sus reglas de acceso. Posteriormente se sincronizó el proyecto de Firebase en la carpeta “web” del espacio de trabajo de VS Code.

Software de monitorización de la instalación:

Una vez creado el proyecto de Firebase, se comenzó a desarrollar el software de monitorización de la instalación, donde se creó un primer script en Python para recoger los valores de los sensores. Dada la gran cantidad de sensores que se iban a usar, se decidió crear un módulo de Python para gestionar todos los sensores, este módulo se llamó “*heliosMeters*” y contiene una clase principal, que sirve de modelo para tres subclases, que permiten configurar los sensores de medición de corriente, tensión y potencia.

Durante el desarrollo del software de monitorización también se ha creado el código para el microcontrolador de Arduino, que registra la tensión y la corriente del sistema alrededor de unas 6000 veces por segundo, y cada 15 segundos realiza el cálculo de la potencia consumida en esos últimos 15 segundos, almacenando el valor entre lecturas para poder enviarlo cuando se pida a través del sistema I2C.

Posteriormente se realizó la última parte del software de monitorización que consistía en enviar los datos recogidos a la base de datos de Firestore desde el script de Python. Pronto, también se vio la necesidad de crear un nuevo módulo para gestionar el tratamiento de los datos antes de subirlos a la base de datos, a este módulo se le llamó “*heliosDatabase*” y su clase principal permite crear unos métodos para enviar los datos en tiempo real y almacenar los datos del histórico, esta primera clase es la clase padre de otras 3 clases del módulo que permiten gestionar la carga de datos de la batería, la potencia consumida y usada, y los datos estadísticos.

Con estos tres apartados se finaliza la parte de software respectiva al sistema de monitorización, y ya con datos disponibles en la base de datos se puede empezar a trabajar en algunos de los aspectos más relevantes de la aplicación web.

Software de la aplicación web:

La parte de implementación del software de la aplicación web, comenzó con la creación de la estructura de la aplicación web, que se subdividió en dos apartados, el primero de ellos representa la página principal de la aplicación y las páginas de registro e inicio de sesión, y el otro apartado representa las páginas de la aplicación web en sí.

En primer lugar, se elaboraron las páginas de inicio de sesión y registro y se tomaron decisiones de qué hacer cuando ya había una cuenta iniciada según en que página estuviéramos, con ello se ha llegado a un sistema que responde teniendo en cuenta el estado de la sesión.

Una vez realizadas las páginas de registro e inicio de sesión se desarrollan las funcionalidades principales de la aplicación, la gestión de los datos en tiempo real y el histórico, además de las acciones de perfil.

Para estas últimas partes de la aplicación se crearon prototipos de baja fidelidad, con ideas de cómo iba a ser la interfaz de la aplicación, para poder hacer un estudio y realizar cambios antes de tener la interfaz desarrollada.

Algunos de los prototipos creados para las interfaces se pueden ver en las ilustraciones, Ilustración 40, Ilustración 41 e Ilustración 42.

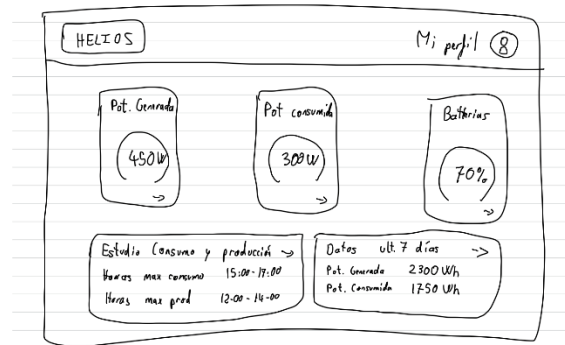


Ilustración 40: Prototipo pantalla principal

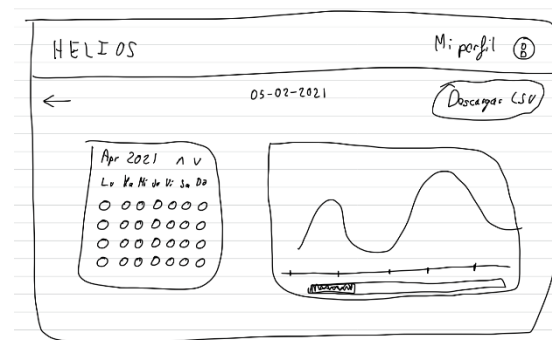


Ilustración 41: Prototipo histórico de datos

Ilustración 42: Prototipo datos de perfil

Gracias a estos prototipos se ha conseguido tener una primera visión de cómo iba a ser la interfaz además de haber podido realizar cambios que si se hubieran hecho más adelante seguramente habrían llevado una cantidad de tiempo considerable.

Con la finalización de estas tareas concluye la parte del desarrollo del software de la aplicación web.

5.7.2. Implementación del hardware

La implementación del hardware en bastantes ocasiones se ha realizado de forma paralela a la implementación del software. Ha consistido principalmente en la configuración de unos sistemas de sensores y conectarlos a la Raspberry Pi 4. Los sensores que se han construido en esta fase han sido sensores de tensión, sensores de corriente y por último un sensor de potencia.

Los dos problemas principales de este apartado eran medir voltajes y corrientes, para ello se propuso una solución para cada una de las tareas, la primera de ellas referida a la medición de voltajes, fue adecuar el voltaje que se iba a leer para poder pasarlo por un convertidor analógico-digital y obtener una medida, en cuanto a la corriente se investigaron soluciones como usar resistencias shunt para determinar la corriente, pero finalmente se optó por usar unos sensores comerciales de medición de corriente, que devolvían un voltaje en relación con la corriente que pasaba por ellos y podíamos medir también con el conversor analógico-digital.

Los convertidores analógico-digital elegidos para la tarea de medir voltajes, han sido los convertidores ADS1115, con una resolución de 15 bits y 860 muestras por segundo. [34] En la Ilustración 43 se puede apreciar uno de estos convertidores analógico-digital.

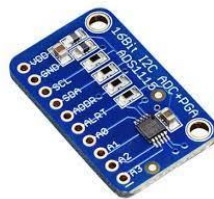


Ilustración 43: Convertor analógico digital ADS1115

Para la medición de la corriente se han elegido los sensores ACS712. Son unos sensores de efecto Hall capaces de medir corrientes en alterna y continua con un rango de -30 A a 30 A. En la Ilustración 44 se puede apreciar uno de estos sensores de corriente.

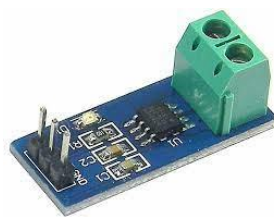


Ilustración 44: Sensor de corriente ACS712

En el caso de la medición de potencia a la salida del inversor, el tiempo de muestreo de los convertidores ADS1115 era insuficiente para poder reconstruir la onda de corriente alterna y hacer los cálculos de la potencia consumida, por lo que se optó por usar un Arduino Nano Every, con una tasa de muestreo de unas 6000 muestras por segundo y con la ventaja de poder realizar los cálculos de potencia dentro del Arduino, para enviar los datos ya procesados directamente a la Raspberry Pi 4.

Protocolo de comunicación con los sensores:

La primera tarea que se ha realizado ha sido la selección del protocolo de comunicación a usar en la comunicación de los sensores con la Raspberry Pi. Como solución a este problema se ha elegido el protocolo de comunicación I2C por varias razones, entre ellas, es un protocolo que requiere de pocos cables para la comunicación, no necesitamos una tasa de transmisión muy alta dado que la información de los sensores se lee cada pocas centésimas de segundo y en otros casos cada varios segundos, y por último los convertidores analógico-digital (ADS1115) funcionan usando este protocolo de comunicación.

En la Ilustración 45 se puede ver un esquema de las conexiones de las líneas de transmisión de datos y reloj de I2C además de su alimentación, también podemos ver en las líneas SDA y SCL las resistencias pull-up necesarias para cada una de ellas. En el esquema se observa un dispositivo I2C maestro (Raspberry Pi 4) y 3 esclavos (2 x ADS1115 y un Arduino Nano Every).

En la Tabla 2 tenemos la leyenda de los colores de los cables.

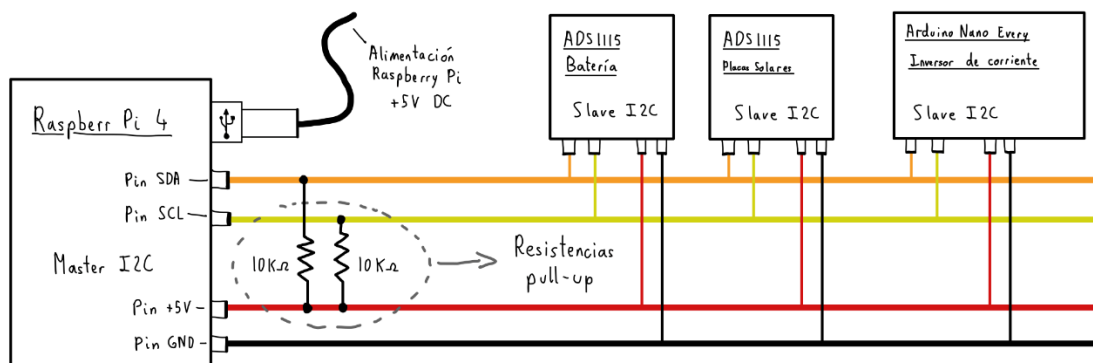


Ilustración 45: Esquema conceptual I2C

Cable	Color
GND	Negro
+5V	Rojo
Línea SDA	Naranja
Línea SCL	Verde

Tabla 2: Leyenda colores conexiones I2C

De la alimentación de los dispositivos del bus I2C y la gestión de las líneas de comunicación se ha encargado la Raspberry Pi, que además es el único dispositivo que controla el bus I2C en este proyecto.

Circuitos de adecuación de las señales:

Una vez conectados todos los dispositivos del protocolo I2C, se han comenzado a elaborar los circuitos para ajustar los voltajes para el convertidor analógico-digital ADS1115. Las limitaciones de este convertidor según la ganancia establecida es una lectura de voltajes desde -0,178 V a 4,095 V. En realidad, este es un convertidor analógico-digital de 16 bits, en el que uno de esos bits se usa para representar el signo negativo, pero dado que la amplitud de lectura de voltajes negativos es tan pequeña podemos considerar que estamos trabajando con un convertidor analógico-digital de 15 bits que puede leer de 0 V a 4,095V, eso nos da una resolución de 32768 niveles (2^{15}) y 0,125 mV por nivel.

El sensor de corriente produce una señal de 2,5 V para un valor de 0 A, en caso de que el amperaje sea mayor o menor a 0 A se multiplicarán los amperios por 0.066 V/A, y ese valor se le sumará a los 2,5 V del valor del cero. También podemos invertir esa fórmula para obtener el valor máximo que podemos medir con nuestro convertidor analógico-digital, la fórmula que usaremos para sacar el valor máximo es la siguiente:

$$\frac{V_{Max} - 2,5 V}{0,066 V / A}$$

Con ella obtenemos el valor de amperios que nos dará un cierto voltaje, introduciendo el máximo que puede leer nuestro convertidor analógico digital obtenemos un valor en torno a 24 A. Como la placa solar genera un máximo de N A no tendremos problema en este apartado.

Con la lectura de los voltajes de la batería y de las placas solares nos encontramos con que los valores que queremos leer son 4 o 5 veces mayores que el máximo valor que puede leer nuestro convertidor analógico-digital, por lo que será necesario adecuar los voltajes, para ello se han usado divisores de tensión hechos a medida con resistencias para la medida de los voltajes de la batería y las placas solares y obtener una resolución de 2 y 3 decimales respectivamente.

Los circuitos de los divisores de tensión son los que se pueden observar en la Ilustración 46, en ellos se puede ver el voltaje máximo para el cual producen un valor en la salida cercano a los 4,095 V del convertidor analógico digital, además de los valores de las resistencias.

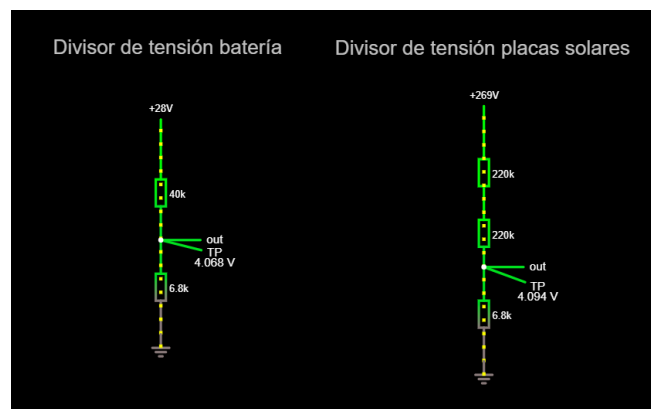


Ilustración 46: Divisores de tensión

En este punto teóricamente se podrían leer los voltajes con los convertidores analógico digital, salvo por dos cuestiones, la primera recae en que la lectura de la salida de los divisores de tensión se puede ver afectada por las cargas que haya detrás, para ello se va a “aislar” el voltaje de esa salida, usando la propiedad de los amplificadores operacionales de una impedancia de entrada infinita y una impedancia de salida nula, con esa característica colocando un amplificador operacional en modo seguidor de voltaje (Ilustración 47), se consigue que la lectura a la salida del divisor de tensión no se vea prácticamente afectada por la propia lectura.

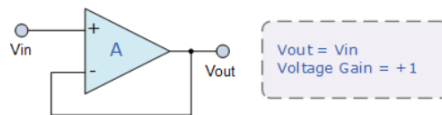


Ilustración 47: Amplificador operacional, Modo seguidor de voltaje

El otro desafío que se ha presentado ha sido que los bornes positivos de la batería y de las placas estaban conectados por el regulador de corriente, por lo que la única manera de leer los voltajes era leerlos al revés con voltajes negativos, entonces para poder introducirlos al convertor analógico-digital hace falta invertirlos, pero para eso se encontró una solución bastante sencilla usando amplificadores operacionales, configurándolos en modo inversor como se puede apreciar en la Ilustración 48.

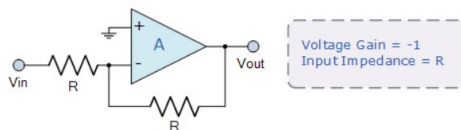


Ilustración 48: Amplificador operacional, Modo inversor

Con estas dos soluciones el circuito final para la corrección del voltaje de la batería y de las placas solares quedaría algo similar al de la Ilustración 50, donde se ve primero el amplificador operacional en modo seguidor de voltaje y después el amplificador operacional en modo inversor, siendo el esquema de un amplificador operacional el de la Ilustración 49.

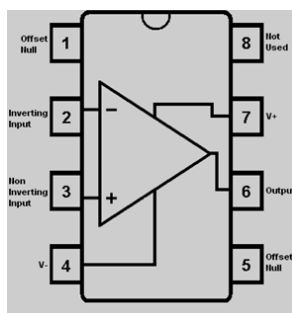


Ilustración 49: Esquema OPAMP

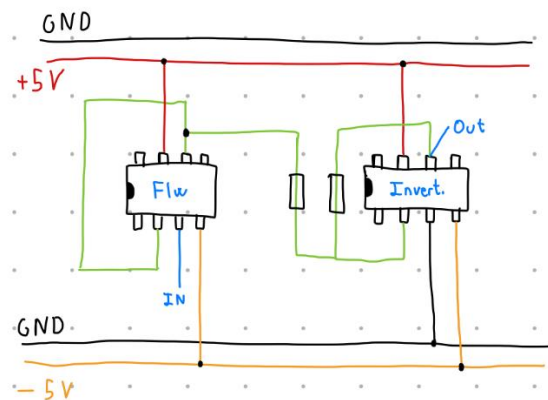


Ilustración 50: Circuito amplificadores operacionales

Una vez terminada la lectura de los voltajes de la batería y placas solares, se ha realizado el circuito para analizar la onda de corriente a la salida del inversor para poder realizar el cálculo de la potencia consumida. La onda de voltaje a la salida del inversor es una onda como la de los enchufes de casa, tiene forma senoidal y sus voltajes pico a pico van de ~ -310 V a ~ 310 V, esta onda es terrible para medirla con nuestra Raspberry Pi, que tiene un rango de medida en su conversor analógico-digital de 0 V a 5 V. Entonces lo que buscamos es pasar esa onda que tenemos a una cuyos voltajes pico a pico estén dentro del intervalo 0-5 V, para ello necesitaremos reducir el voltaje de la onda con un divisor de tensión y añadirle un offset para eliminar los voltajes negativos.

El circuito que se ha construido para solucionar este problema es bastante simple, y se puede observar en la Ilustración 51. Consta de dos divisores de tensión, uno para reducir el voltaje del inversor y otro para crear un offset de 2,5 V a partir de la alimentación general de 5V.

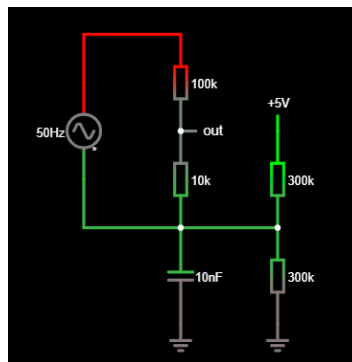


Ilustración 51: Circuito medida a la salida del inversor

Con la herramienta CircuitJS y posteriormente con el osciloscopio se ha comprobado la forma de onda en la salida del circuito, como se puede apreciar en la Ilustración 52, donde se ve los valores que toma la onda una vez se alimenta el circuito.

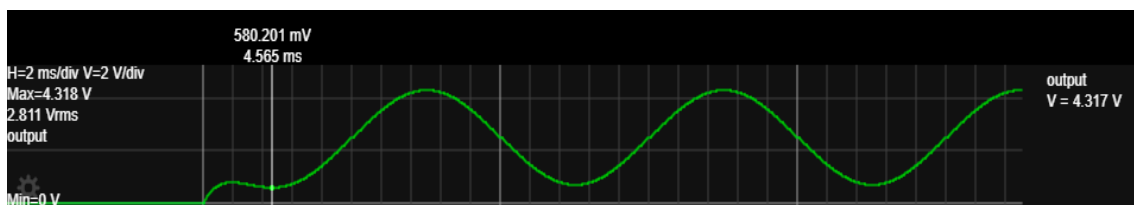


Ilustración 52: Análisis forma de onda salida del circuito del inversor

Finalmente se ha añadido un amplificador operacional a la salida del circuito de la Ilustración 51, para poder leer de manera correcta la señal.

Para la lectura de los voltajes de los sensores de corriente no es necesario usar un amplificador operacional dado que ya llevan uno integrado.

Con esto quedaría acabada la creación de los circuitos de los sensores y solo faltaría juntar las piezas y montar todo el circuito como se verá más adelante.

Se ha decidido hacer un poco más esquemática la representación de este circuito usando una serie de cajas negras que representan cada uno de los circuitos de adecuación de señales vistos anteriormente.

Como se ha visto en algunos de los esquemas anteriores es necesario tener una alimentación de -5 V, como la alimentación de 5 V nos la proporciona la Raspberry Pi hemos usado cargador de móvil normal y corriente, conectando su cable positivo a la tierra del circuito para conseguir así un voltaje de -5 V, que se observará en los circuitos mostrados a continuación.

Las mediciones realizadas se pueden dividir en **tres** bloques principales:

- **Circuito de medición del voltaje de la batería:** En el esquema de este circuito, visible en la Ilustración 53, tenemos un convertidor analógico-digital (ADS1115), conectado al circuito de inversión de voltaje y seguimiento de voltaje, que a su vez está conectado al divisor de tensión conectado al borne negativo de la batería. El borne positivo de la batería está conectado a tierra para poder medir el voltaje relativo en el borne negativo.

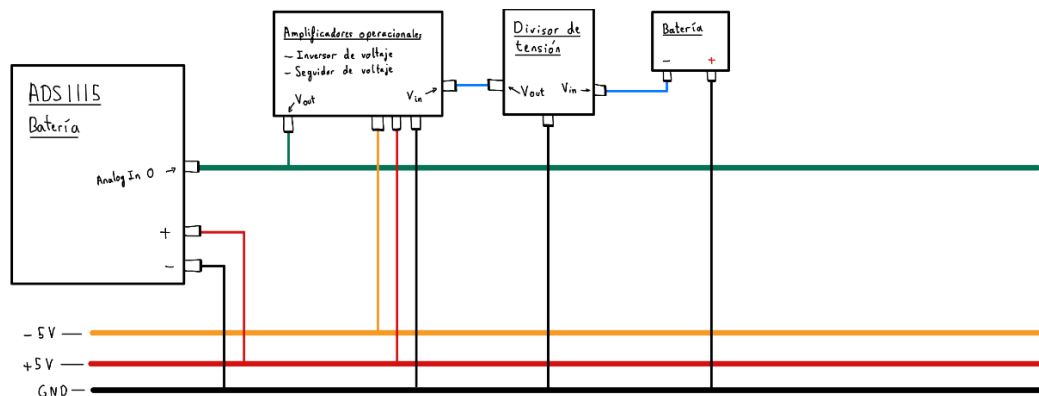


Ilustración 53: Circuito medidor de voltaje de la batería

- **Circuito de medición de potencia de las placas solares:** En el esquema de la Ilustración 54 se pueden observar los componentes de este circuito, tenemos un convertidor analógico-digital (ADS1115), al que se le conectan un sensor de corriente (ACS712), conectado en serie con las placas solares, y el circuito de amplificadores operacionales en modo seguidor de voltaje e inversión de voltaje, que está conectado a un divisor de tensión conectado al cable negativo de la placa solar. El contacto con el cable positivo de la placa solar se conecta a la tierra del circuito para establecer la referencia de voltaje, como en el caso del circuito de medición del voltaje de la batería.

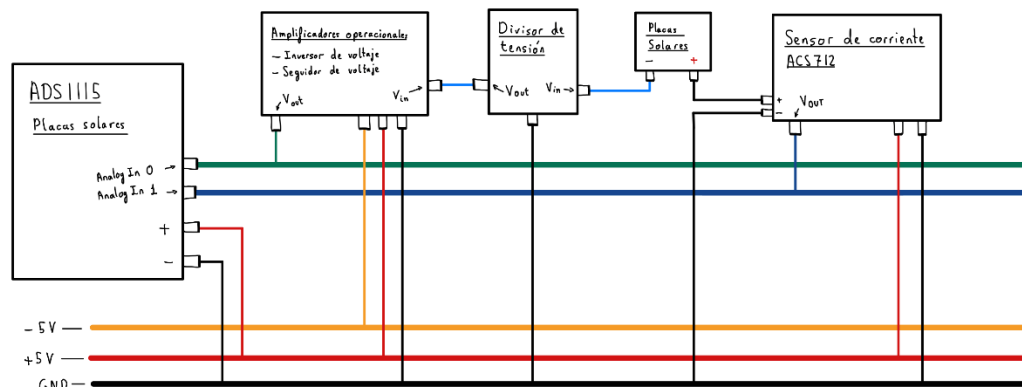


Ilustración 54: Circuito de medición de potencia de las placas solares

- Circuito de medición de consumo a la salida del inversor:** En el esquema de la Ilustración 55 se observan los componentes que forman este circuito, el esquema es muy similar al del circuito de medición de potencia de las placas solares, salvo por unos detalles, en este caso tenemos un Arduino Nano Every), al que se le conectan un sensor de corriente (ACS712), conectado en serie con la salida del inversor de corriente, y un amplificador operacional en modo seguidor de voltaje, que está conectado a un divisor de tensión con un offset de 2,5 V, como vimos en el esquema de la Ilustración 51, a su vez conectado a uno de los cables de la salida del inversor.

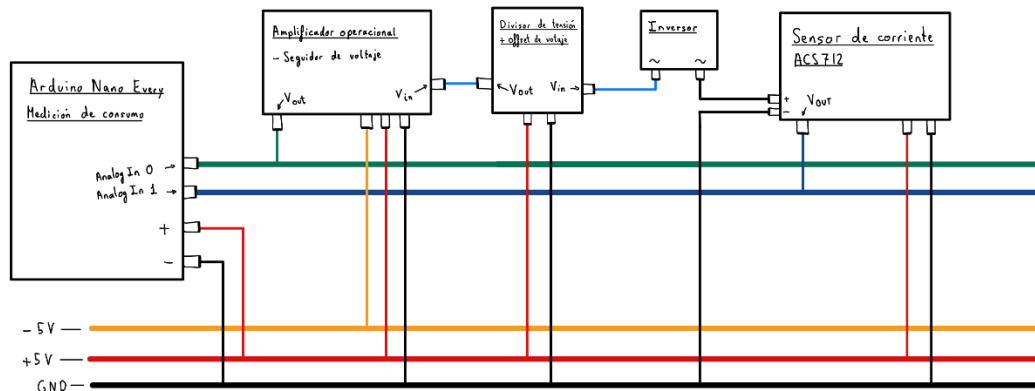


Ilustración 55: Circuito de medición de consumo a la salida del inversor

A continuación, en la Ilustración 56, podemos ver la realización física de estos circuitos:

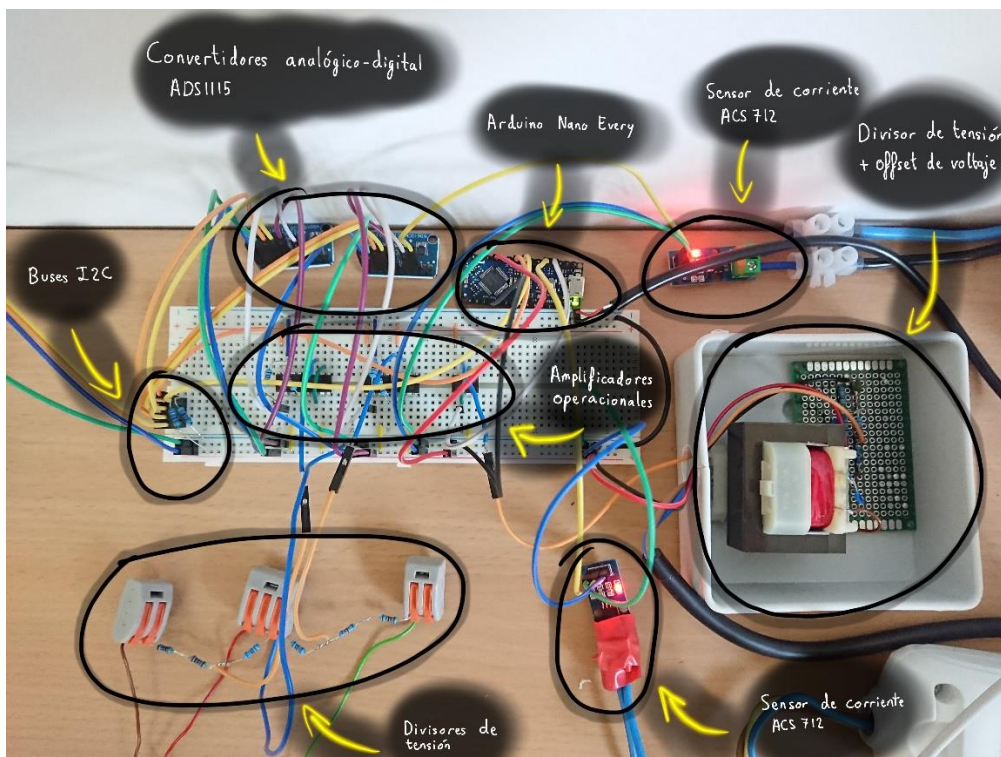


Ilustración 56: Realización física de los circuitos de los sensores

Cabe destacar que, en la Ilustración 56, en el divisor de tensión con el offset de voltaje, se ve un transformador de corriente, pero la única función de este transformador es aislar la corriente del inversor de la corriente del sistema de sensores.

Para finalizar el apartado de implementación del hardware, vamos a ver un esquema global de la instalación, donde se representarán mediante cajas negras y conexiones simples las relaciones entre los diferentes elementos del sistema.

El esquema de la Ilustración 57, muestra como la energía solar llega a las placas solares, allí se transforma en una corriente eléctrica que va hasta el controlador de carga de las baterías, esa primera corriente eléctrica la medimos con el sensor de las placas solares, tanto el voltaje como el amperaje. Desde el controlador de carga se distribuye la energía hacia las baterías y el inversor. En las baterías se conecta el sensor de carga de las baterías que mide el voltaje de estas. A la salida del inversor se conecta el sensor de consumo que medirá la tensión y la corriente que pasan por ese punto, para poder calcular el consumo de potencia que está realizando el usuario. Todos los sensores del esquema están conectados con la Raspberry Pi 4 a través del Bus I2C por el que comunican los datos obtenidos.

En la Tabla 3 se encuentra la leyenda de los colores y las conexiones del esquema.

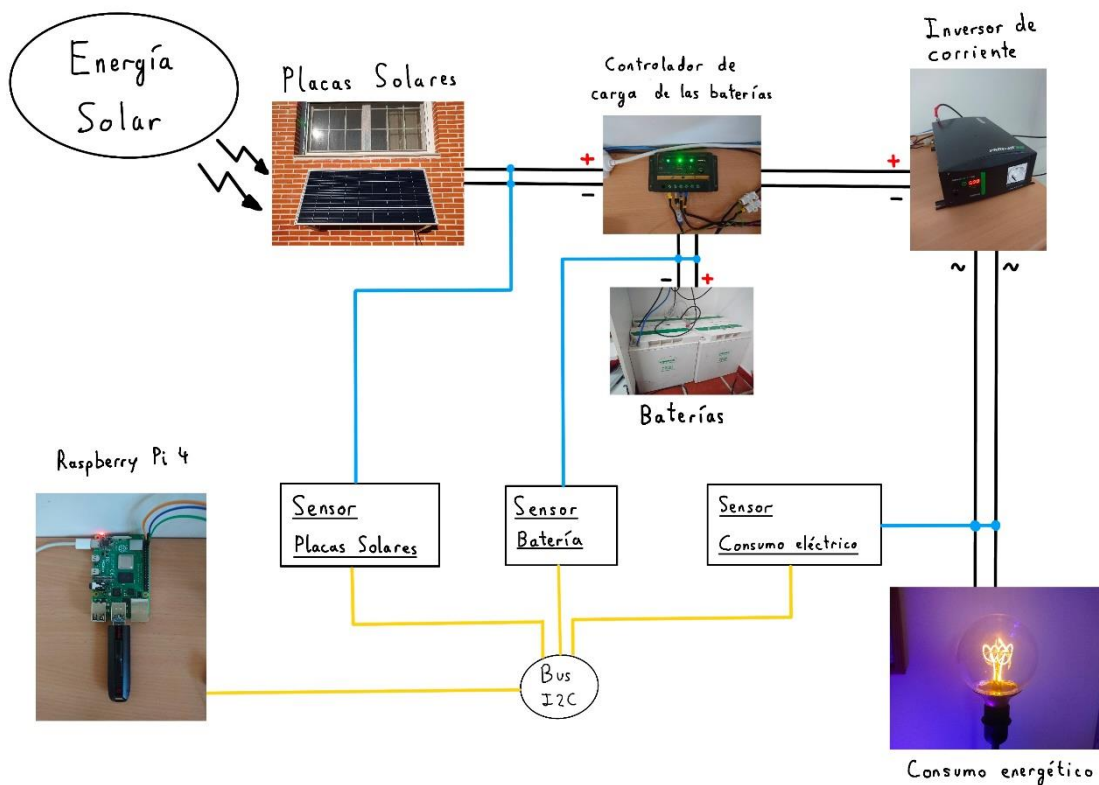


Ilustración 57: Esquema global de la instalación

Conexión	Color
Cables instalación	Negro
Bus I2C (SDA y SCL)	Amarillo
Cables de sensor	Azul

Tabla 3: Leyenda colores cables instalación global

5.7.3. Costes del hardware:

A continuación, se detallan los precios de los componentes usados en el sistema de sensores del proyecto, indicando también la tienda donde se han adquirido. La lista de componentes usados se puede observar en la Tabla 4.

Componente	Tienda	Precio
1 x Raspberry Pi 4 (4 GB)	Kubii	50€
3 x Arduino Nano Every	Arduino	22€
5 x Sensores de corriente ACS712	Amazon	13€
5 x Conversores analógico digital ADS1115	Amazon	17€
Kit de amplificadores operacionales	Amazon	9€
Kit de resistencias (tolerancia 1%)	Amazon	9€
10 x Placas PCB universales	Amazon	4€

Tabla 4: Lista de componentes y precio

5.8. Pruebas

Durante todo el desarrollo del proyecto se han ido realizando pruebas de los componentes implicados en el sistema, para poder comprobar el correcto funcionamiento y verificar que se cumplieran los objetivos marcados, usando algunas de las herramientas descritas en el apartado 4 Técnicas y herramientas.

Las pruebas que se han ido realizando son las siguientes:

Pruebas sobre la base de datos Firestore de Firebase:

Estas han sido unas de las primeras pruebas que se realizaron, cuando se creó el proyecto de Firebase y se configuró la base de datos, se crearon una serie de reglas para controlar el acceso a la base de datos y con ello se realizaron unas pruebas de acceso para verificar el funcionamiento de las reglas, para ello se usó el entorno de pruebas de Firestore, como se puede ver en la Ilustración 58.

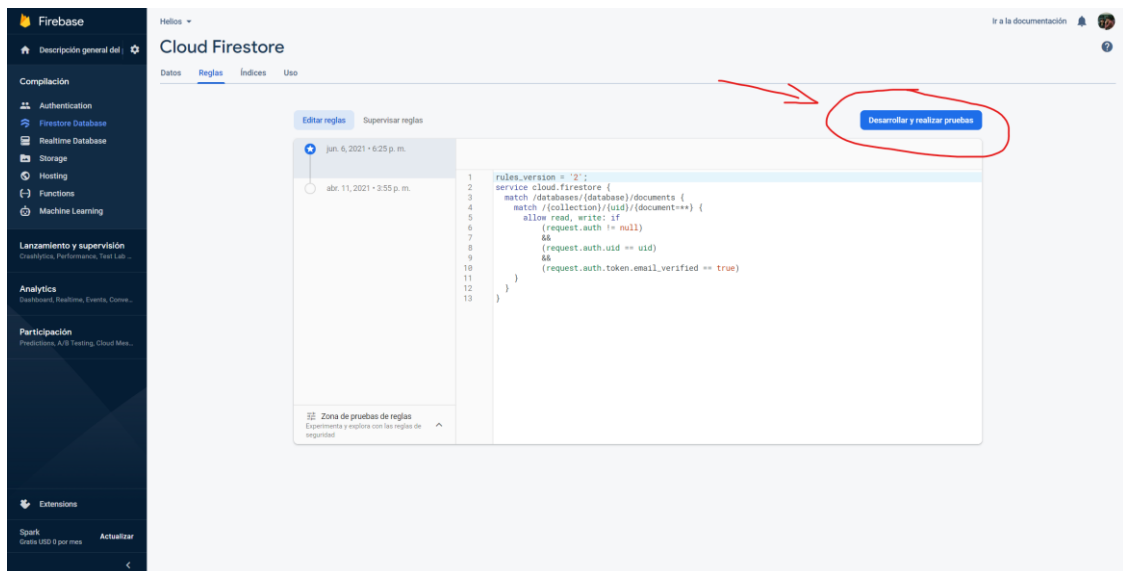


Ilustración 58: Entorno de pruebas de Firestore

Pruebas con el servicio Linux:

Una vez creado el servicio Linux para la ejecución del programa de monitorización, se realizaron varias pruebas de ejecución y restablecimiento del servicio frente a fallos, comprobando el estado del sistema con las órdenes “sudo systemctl status helios.service” (Ilustración 59) y “sudo journalctl -u helios.service” (Ilustración 60).

```

pi@RP4-Miguel: /home/pi [00:42:20]
$ sudo systemctl status helios.service
● helios.service - Helios monitoring service.
   Loaded: loaded (/lib/systemd/system/helios.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-08-30 20:32:54 CEST; 3 days ago
     Main PID: 27738 (python3)
        Tasks: 5 (limit: 4915)
      CGroup: /system.slice/helios.service
             └─27738 python3 /usr/bin/HeliosMonitoring.py

Aug 30 20:32:54 RP4-Miguel systemd[1]: Started Helios monitoring service..

```

Ilustración 59: Comando systemctl

```

pi@RP4-Miguel: /home/pi [00:42:33]
$ sudo journalctl -u helios.service
-- Logs begin at Thu 2021-08-19 16:29:21 CEST, end at Fri 2021-09-03 00:47:31 CEST. --
Aug 19 16:29:37 RP4-Miguel systemd[1]: Started Helios monitoring service..
Aug 23 10:52:57 RP4-Miguel env[561]: Traceback (most recent call last):
Aug 23 10:52:57 RP4-Miguel env[561]:   File "/usr/bin/HeliosMonitoring.py", line 172, in <module>
Aug 23 10:52:57 RP4-Miguel env[561]:     main()
Aug 23 10:52:57 RP4-Miguel env[561]:   File "/usr/bin/HeliosMonitoring.py", line 119, in main
Aug 23 10:52:57 RP4-Miguel env[561]:     vBatteryVoltage += batteryVoltage.readRaw()
Aug 23 10:52:57 RP4-Miguel env[561]:   File "/usr/share/helios/heliosMeters/HeliosVoltage.py", line 22, in readRaw
Aug 23 10:52:57 RP4-Miguel env[561]:     return self._channel.value

```

Ilustración 60: Comando journalctl

La creación del servicio Linux aparte de comprobar el estado del sistema ha servido para obtener información de los fallos de la ejecución como también se puede ver en la Ilustración 60.

Pruebas de la aplicación web:

Para realizar pruebas en la aplicación web se ha usado el navegador Microsoft Edge con la herramienta DevTools del navegador que se inicia pulsando la tecla F12.

En la Ilustración 61 podemos ver la aplicación web del proyecto (a la izquierda) junto con la herramienta DevTools del navegador (a la derecha).

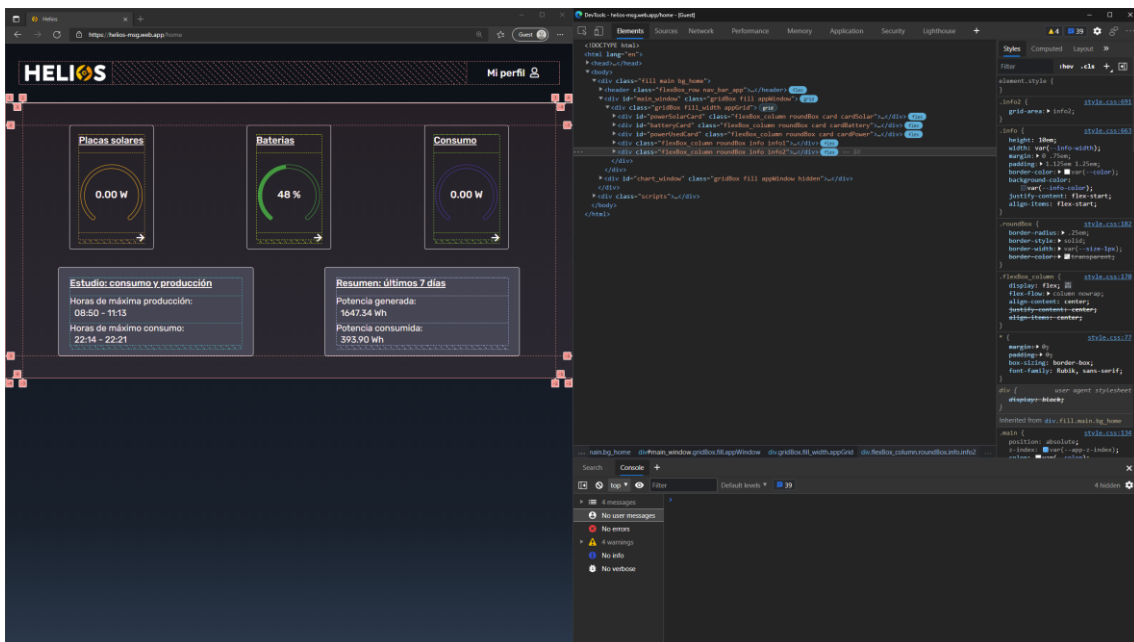


Ilustración 61: WebApp y DevTools

Generalmente, las pruebas realizadas con esta herramienta han sido visualizar en la consola de JavaScript los mensajes de depuración introducidos en el código, además de visualizar algunos datos de objetos de JavaScript para estudiar su funcionamiento, como ha sido el caso de los datos de los objetos leídos de la base de datos Firestore.

También se ha usado la herramienta DevTools para comprobar la responsabilidad de la aplicación web, ya que esta herramienta dispone de un entorno de simulación de dispositivos, desde teléfonos móviles hasta *tablets*. En la Ilustración 62 se puede ver una captura de la herramienta de simulación representando como se vería la interfaz en un teléfono Moto G4.

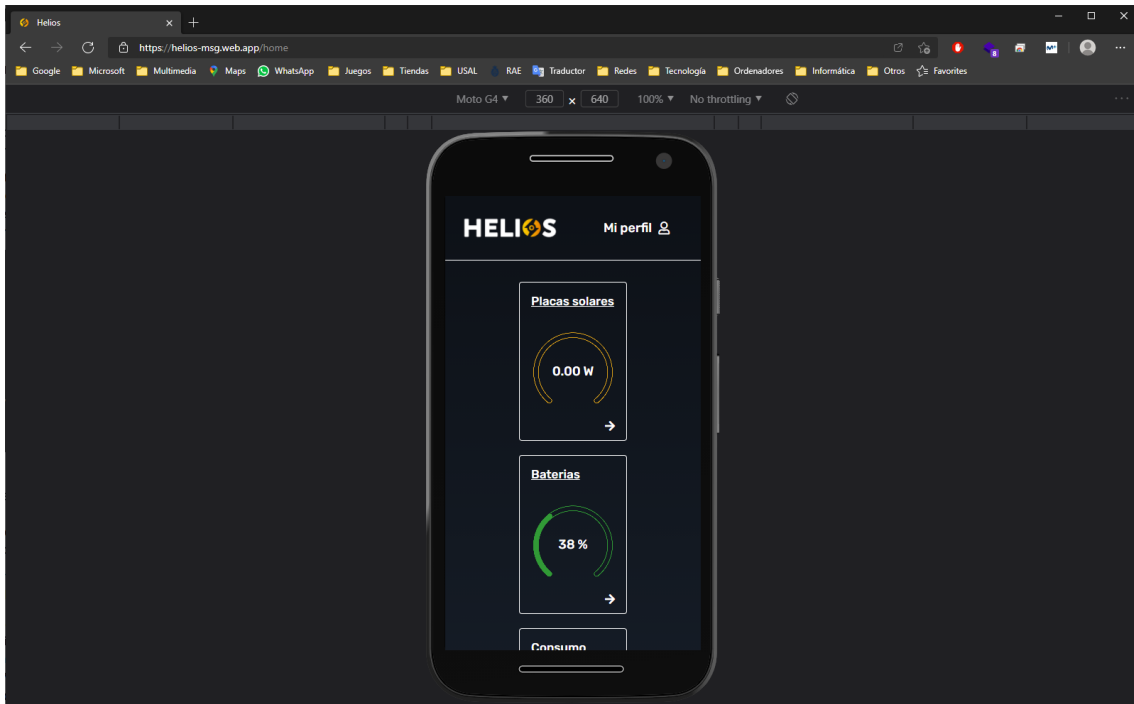


Ilustración 62: Simulación de dispositivos

Por último, se ha usado la opción *Lighthouse* para generar un reporte del funcionamiento de la aplicación web y ver que cosas se podrían mejorar o habría que corregir. Una de las pruebas realizadas se puede observar en la Ilustración 63.

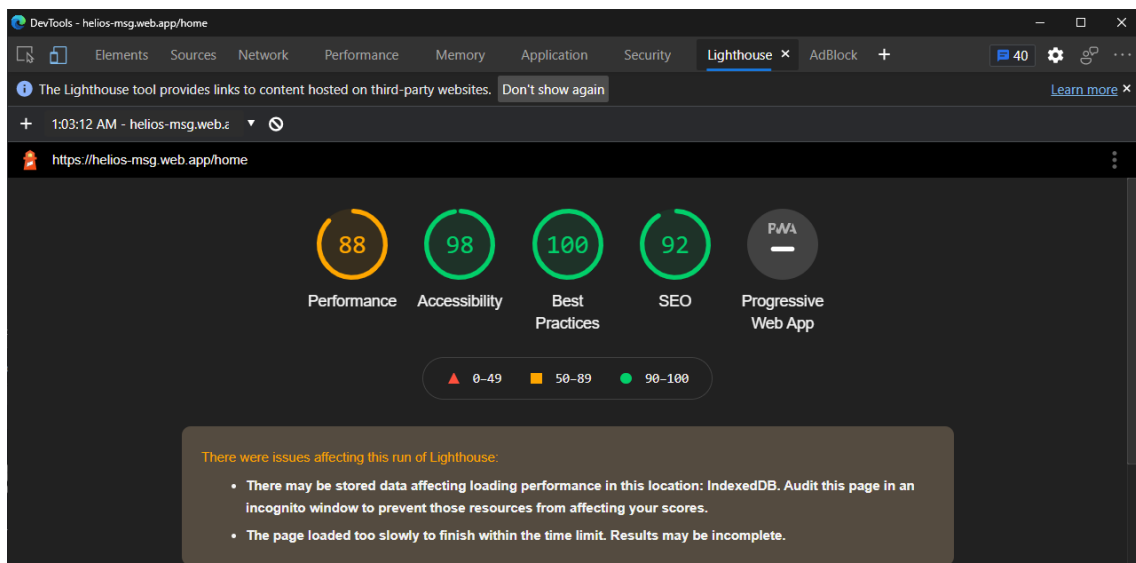


Ilustración 63: Lighthouse DevTools

Pruebas de circuitos:

Los circuitos diseñados en este proyecto se han puesto a prueba usando placas de prototipo (Ilustración 64) para montar los componentes temporalmente, una fuente de alimentación (Ilustración 65) para probar algunos de los circuitos y un osciloscopio (Ilustración 66) para visualizar las formas de onda con las que se iba a trabajar.

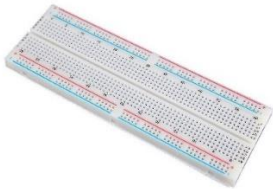


Ilustración 64: Placa prototipo



Ilustración 65: Fuente alimentación



Ilustración 66: Osciloscopio

También se han usado las herramientas Tinkercad Circuits y CircuitJs para realizar simulaciones de circuitos antes de realizar las pruebas en la vida real.

En el caso del circuito de acondicionamiento de señal a la salida del inversor, una vez realizadas las pruebas usando una placa de prototipo se ha usado una placa PCB universal para montar los componentes del circuito y que queden unidos de manera permanente. Esto se ha hecho ya que este circuito trabaja con corriente alterna de alto voltaje y es mejor que esté lo más protegido posible, además de eso se ha introducido en una caja para proporcionar mayor aislamiento, tal y como se observa en la Ilustración 67.



Ilustración 67: Circuito en placa PCB

Pruebas de la monitorización y lectura de sensores:

Para comprobar el correcto funcionamiento de los sensores y los datos recogidos en la Raspberry Pi 4, se han verificado los valores recibidos en la base de datos con los valores observados experimentalmente usando aparatos de medida como un multímetro digital (Ilustración 68) o un medidor de potencia digital (Ilustración 69).



Ilustración 68: Multímetro digital



Ilustración 69: Medidor de potencia digital

5.9. Descripción funcional

En este apartado se describen las funcionalidades principales del sistema y se explican que aspectos hacen posibles que se cumplan.

Para ver otros detalles de uso del sistema se recomienda ver el *Anexo VI – Manual de usuario*.

5.9.1. Visión global del sistema

En la Ilustración 70 se puede observar una visión global del sistema donde se ven los dos elementos principales que interactúan con los servicios de Firebase, lugar en el que se aloja la base de datos, el servicio de hosting y el servicio de autenticación. Los sensores de la instalación se comunican con la base de datos para enviar los datos recogidos y para poder realizar los cálculos estadísticos. Por otro lado, la aplicación web desarrollada, “Helios app”, se comunica con el servicio de hosting para obtener las páginas de la aplicación web, también maneja los inicios de sesión comunicándose con el servicio de autenticación y por último obtiene la información de la monitorización de la base de datos y la procesa para poder mostrarla al usuario.



Ilustración 70: Visión global del sistema

5.9.2. Instalación y sistema de sensores

La instalación como bien se explicó en la Ilustración 57, consta de una placa solar, un regulador de carga, unas baterías y un inversor de corriente.

El sistema de sensores se encarga de recoger, procesar y enviar los datos de la instalación a la base de datos, el procesamiento que realiza el sistema de sensores en la mayoría de los casos queda en dar un formato a los datos recogidos para cargarlos a la base de datos, pero en el caso de los datos estadísticos de potencia, el algoritmo lleva una cuenta de los registros apuntando cuales son los de mayor valor y seleccionando los 144 primeros por cada día, dejando constancia de ellos en un registro global en la base de datos. Estos datos se procesarán más tarde en la aplicación web para obtener las horas de máximo consumo y producción.

5.9.3. Servicios de Firebase

Los servidores de Google son los encargados de alojar los datos del proyecto, los datos de monitorización, almacenados en su base de datos Firestore; los datos de las cuentas de los usuarios de la aplicación, gestionados por el servicio Firebase Auth; y por último los datos del código de la aplicación web que mediante el servicio de hosting se sirven al usuario para que pueda ejecutar la aplicación en su navegador.

5.9.4. Helios App

La aplicación web permite la visualización los datos de la monitorización, de una manera cómoda e intuitiva.

Los usuarios en la pantalla principal pueden registrarse o iniciar sesión, como se ve en la Ilustración 71.

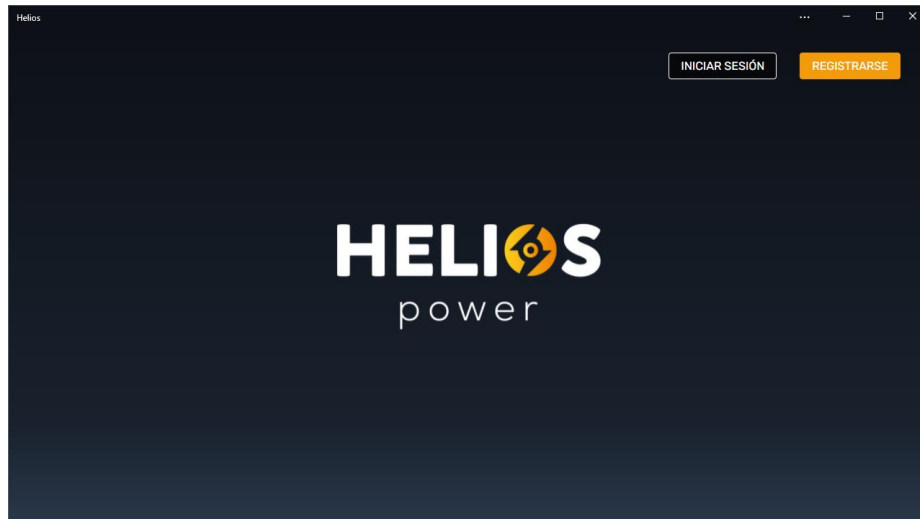


Ilustración 71: Pantalla de acceso a la aplicación

Una vez inician sesión acceden a la pantalla principal de la monitorización (Ilustración 72), donde se encuentran con los relojes de datos en tiempo real y los datos estadísticos. Los relojes tiempo real están enlazados con la base de datos, de forma que cuando se actualice un dato en el sistema de sensores se verá reflejado en su respectivo reloj. Los datos estadísticos de los últimos 7 días se calculan haciendo la suma de los totales de los últimos 7 registros si los hubiera. En cuanto, a los datos del estudio de consumo y producción, el algoritmo recoge los datos que había procesado el sistema de monitorización y busca un intervalo de 144 valores, en el que la suma de sus valores sea máxima, este intervalo representa en el mayor de los casos un espacio temporal de 2h y 24 minutos (144 minutos).



Ilustración 72: Pantalla principal monitorización

El algoritmo del cálculo estadístico de máximo consumo y producción es el que se puede ver en las ilustraciones Ilustración 73 e Ilustración 74. De forma resumida este algoritmo lo que hace es recoger los datos de la base de datos y una vez los tiene crea dos listas que irá llenando simultáneamente y comparándolas hasta encontrar el intervalo con un valor máximo, a estas listas solo se añadirán valores que superen la media de todos los valores de la gráfica y las listas solo estarán compuestas por valores consecutivos. Finalmente se toma la hora del primer y último valor de la lista con mayor valor y se formatea la cadena que se imprimirá por pantalla.

```
1  /**
2   * Calculates the interval of max value.
3   * @param {*} coll Collection to fetch the data.
4   * @returns {string} String with the dates of the interval.
5   */
6   static calculateMaxInterval(coll) {
7     const documents = coll.docs
8
9     let result = 'Aún no hay datos.'
10
11    // Buscamos el documento en la colección.
12    let index = 0
13    let found = false
14    for (; index < documents.length; index++) {
15      const element = documents[index];
16
17      if (element.id == 'statsGlobal') {
18        found = true
19        break
20      }
21    }
22
23    if (found) {
24      // Obtenemos los datos.
25      const data = documents[index].data().stats
26
27      // Calculamos la media de los datos.
28      let averageValue = 0
29
30      let count = 0
31      for (const [key, value] of Object.entries(data)) {
32        averageValue += value
33        count += 1
34      }
35
36      averageValue /= count
37
38      // Calculamos el intervalo de mayor valor.
39      let list1 = {
40        total: 0,
41        list: []
42      }
43      let list2 = {
44        total: 0,
45        list: []
46      }
47      let listRef = list1
48      let listSaved = list2
49      let listSwitch = false
50      let lastKey = null
51
52      for (const [key, value] of Object.entries(data)) {
53        if (value ≥ averageValue) {
54          // Comprobamos si es el primer valor de la lista.
55          if (!lastKey) {
56            /*
57             * Si es el primer valor asignamos la clave actual
58             * a la variable lastKey
59             */
60            listRef.list.push([key, value])
61            listRef.total += value
62          }
```

Ilustración 73: Algoritmo de cálculo máxima potencia Pt1


```

63         else {
64             /*
65              * Comprobamos si la clave actual es la siguiente a
66              * la anterior.
67              */
68             if (parseInt(lastKey) !== (parseInt(key) - 1)) {
69                 /*
70                  * Si la clave no es la anterior miramos cual de las
71                  * dos listas es mayor y creamos una nueva lista
72                  * sobre la que sea menor para trabajar en ella.
73                  */
74
75                 // Comprobamos que lista es la menor.
76                 if (listRef.total > listSaved.total) {
77                     listSaved = listRef
78                     listSwitch = !listSwitch
79                     listRef = listSwitch ? list2 : list1
80                 }
81
82                 // Reseteamos la lista y añadimos la nueva entrada.
83                 listRef['total'] = 0
84                 listRef['list'] = []
85
86                 listRef.list.push([key, value])
87                 listRef.total += value
88             }
89             else {
90                 if (listRef.list.length == 144) {
91                     /*
92                      * Si la lista está llena comprobamos si añadiendo
93                      * el nuevo valor y quitando el primero la lista
94                      * tiene un valor mayor.
95                      */
96                     let firstValue = listRef.list[0][1]
97                     if (listRef.total < listRef.total + value - firstValue) {
98                         // Quitamos el primer valor y añadimos el nuevo.
99                         const [lKey, lValue] = listRef.list.shift()
100                         listRef.list.push([key, value])
101                         listRef.total += value - lValue
102                     }
103                     else {
104                         /*
105                          * Comprobamos cual de las dos listas
106                          * es mayor y creamos una nueva lista sobre
107                          * la que sea menor para trabajar en ella.
108                          */
109                         if (listRef.total > listSaved.total) {
110                             listSaved = listRef
111                             listSwitch = !listSwitch
112                             listRef = listSwitch ? list2 : list1
113
114                             listRef['total'] = listSaved.total
115                             listRef['list'] = listSaved.list.slice()
116                         }
117
118                         // Quitamos el primer valor y añadimos el nuevo.
119                         const [lKey, lValue] = listRef.list.shift()
120                         listRef.list.push([key, value])
121                         listRef.total += value - lValue
122                     }
123                 }
124                 else {
125                     // Si la lista no está llena añadimos el nuevo valor.
126                     listRef.list.push([key, value])
127                     listRef.total += value
128                 }
129             }
130         }
131         // Actualizamos la última clave.
132         lastKey = key
133     }
134 }
135
136 // Seleccionamos la lista mayor.
137 if (listRef.total < listSaved.total) {
138     listSaved = listRef
139     listSwitch = !listSwitch
140     listRef = listSwitch ? list2 : list1
141 }
142
143 // Obtenemos el primer minuto y el último de la lista.
144 const minutoInicio = parseInt(listRef.list[0][0])
145 const lastItem = listRef.list.length - 1
146 const minutoFin = parseFloat(listRef.list[lastItem][0])
147
148 // Pasamos los minutos a horas.
149 const horaInicio = HeliosApp.minutesToHour(minutoInicio)
150 const horaFin = HeliosApp.minutesToHour(minutoFin)
151
152 // Creamos la cadena de salida.
153 result = `${horaInicio} - ${horaFin}`
154 }
155
156 return result
157 }
158

```

Ilustración 74: Algoritmo de cálculo de máxima potencia Pt2

Si el usuario hace clic en cualquiera de los relojes de tiempo real, se desplegará una gráfica formada por los datos de la monitorización, en la que podrá ver para cada minuto del día el valor que se ha registrado en la base de datos.

Esta gráfica es compartida por todas las representaciones de datos de la aplicación, pero dependiendo de en qué reloj se haya pulsado se mostrarán unos datos u otros.

En la Ilustración 75 tenemos un ejemplo de la gráfica del histórico de datos, en la que se ha centrado la visualización a las 11:00.



Ilustración 75: Ejemplo gráfica de histórico

Dentro de la ventana del histórico si el usuario pulsa sobre la fecha podrá seleccionar cualquier otra fecha de las que haya registros para visualizar los datos en ese día. En la Ilustración 76 podemos ver el calendario de selección de fechas con las fechas disponibles. Estas fechas se toman a partir de los registros de la base de datos, marcando como válido el intervalo de fechas del primer al último registro existente.

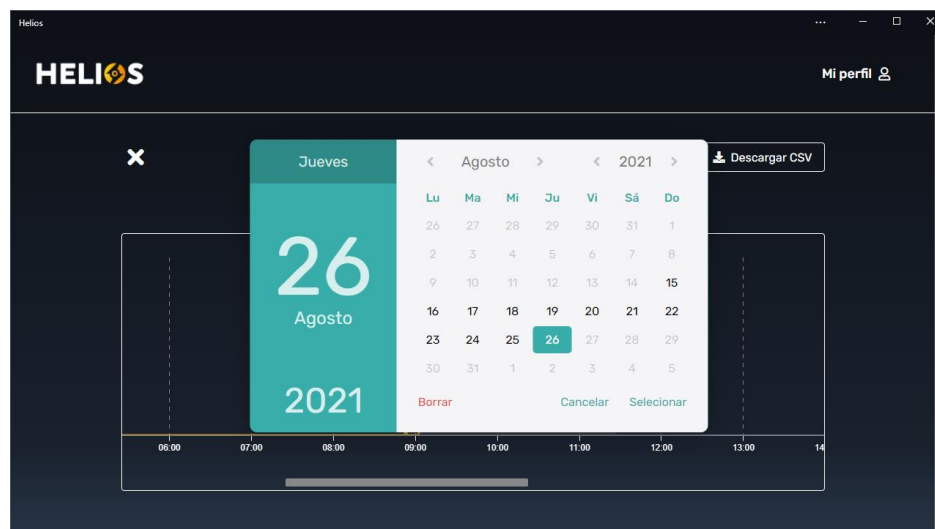


Ilustración 76: Calendario de selección de fechas

En toda la aplicación se ha diseñado un sistema de mensajes emergentes para comunicar los estados de la aplicación al usuario o dar mensajes informativos, un ejemplo de ello es la Ilustración 77, donde se puede ver el mensaje emergente una vez hemos hecho clic en enviar el correo electrónico de verificación.

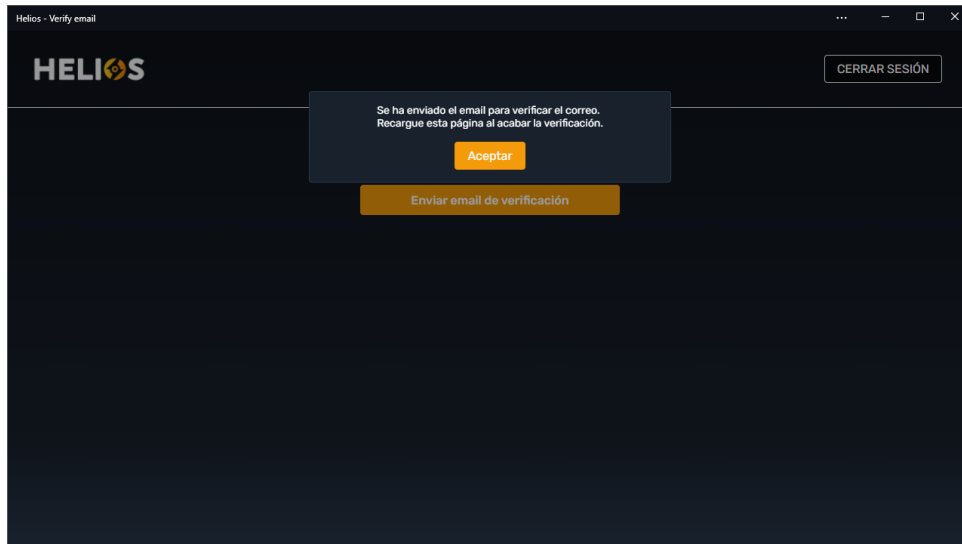


Ilustración 77: Mensaje emergente

Por último, el usuario tiene la oportunidad de editar sus datos de perfil, en la ventana de datos de la cuenta, donde podrá editar los formularios para cambiar sus datos, además de poder eliminar su cuenta previa confirmación de la operación, para algunas modificaciones como cambiar el email o borrar la cuenta es necesario haber iniciado sesión recientemente, por temas de seguridad. La ventana de visualización y edición de los datos de perfil se puede ver en la Ilustración 78.

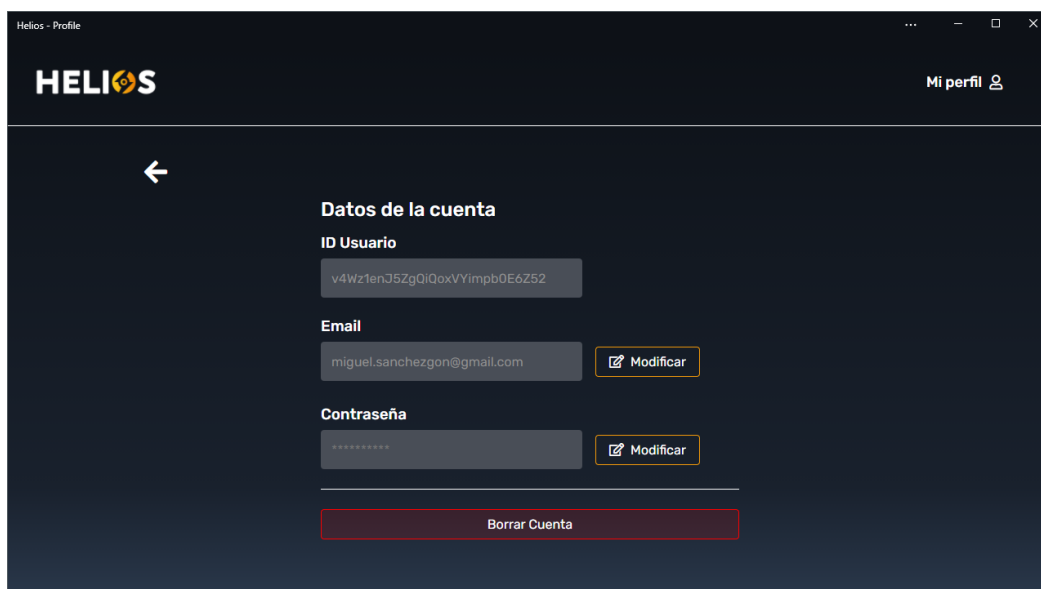


Ilustración 78: Ventana de datos de la cuenta

6. Conclusiones y líneas de trabajo futuras

Finalmente vamos a hacer una revisión sobre las conclusiones a las que se ha llegado al finalizar este proyecto. Además, se verán una serie de posibles caminos de trabajo futuro con el objetivo de ampliar y mejorar el proyecto.

6.1. Conclusiones

La finalización de este proyecto ha desencadenado una retrospectiva sobre los objetivos inicialmente marcados, pudiendo comprobar que todos se han cumplido satisfactoriamente, aunque algunos con pequeñas variaciones a las pensadas inicialmente:

- **Gestión de la monitorización del sistema:** Se ha conseguido monitorizar satisfactoriamente toda la instalación, desde la energía generada en las placas solares, pasando por la almacenada en las baterías y finalmente la consumida por el usuario.
- **Gestión de la visualización de los datos:** En cuanto a la visualización de los datos se ha creado una aplicación que permite ver los datos en tiempo real e históricos de los datos de la instalación, además de estadísticas y resúmenes de consumo y producción, como se indicaba en los objetivos iniciales.
- **Gestión de los roles del sistema:** La gestión de los roles del sistema se ha llevado a cabo usando los servicios que ofrecía la aplicación Firebase de Google, en un principio el método para llevar esto a cabo no estaba definido pero la solución alcanzada cumple los requisitos de manera satisfactoria además de haber proporcionado nuevos conocimientos.
- **Investigación sobre consumo y producción:** Este apartado es el más controvertido en cuanto al cumplimiento de los objetivos que marcaba ya que no se había definido bien del todo su funcionalidad ni como se vería reflejado en la aplicación. La solución propuesta es sencilla y permite hacerse una idea de las predicciones de consumo y producción, pero admite un buen margen de mejora e investigación futura.

Aparte de las conclusiones sobre los objetivos principales propuestos, estos son otros aspectos relevantes de la finalización de este proyecto:

- Durante el desarrollo del proyecto se han adquirido gran cantidad de conocimientos. En el campo de la electrónica se ha conseguido adecuar señales para su medición con sensores. En el campo del desarrollo web se han ampliado mucho los conocimientos de HTML y CSS para la creación de interfaces además de haber aprendido a usar el lenguaje JavaScript y crear multitud de funcionalidades en la aplicación. Con Python y C ya se había trabajado anteriormente y en menor medida este proyecto ha ayudado a reforzar los conocimientos de estos lenguajes.
- Con la realización de este trabajo de fin de grado se han puesto en práctica gran cantidad de conocimientos adquiridos durante la carrera en asignaturas como

Periféricos, Ingeniería del Software, Interacción persona-ordenador o administración de sistemas entre otras. Con estos conocimientos se ha facilitado el desarrollo del proyecto y la búsqueda de soluciones incluso cuando no estaban directamente relacionadas con lo visto en las asignaturas.

- Este proyecto ha servido como acercamiento a lo que podría ser un proyecto de cualquier empresa, pero con la carga de realizarlo una sola persona. Con él se ha aprendido a gestionar el tiempo y obtener una mejor consciencia del esfuerzo que requieren algunas implementaciones.

Como resumen de la conclusión, el proyecto se ha llevado a cabo con éxito cumpliendo todos los requisitos propuestos y ha servido tanto para ampliar los conocimientos adquiridos durante la carrera como para reforzarlos.

6.2. Líneas de trabajo futuras

Durante la realización del trabajo y en su finalización han ido surgiendo diferentes ideas de expansión del proyecto o mejora de sus características que por falta de tiempo no han podido realizarse pero que en un futuro podrían llevarse a cabo, algunas de estas ideas son las siguientes:

- Creación de menú para gestionar varias instalaciones desde una misma cuenta.
- Predicción de desarrollo de los valores de las gráficas en el tiempo.
- Mostrar más estadísticas en los apartados de los resúmenes como pueden ser el total de energía producido y consumido, los valores totales del día actual, el número de días registrados de la instalación...
- Opción de multilinguaje para que usuarios de diferentes idiomas puedan usar la aplicación.
- Creación de sensores inalámbricos usando placas ESP32 para poder deslocalizar la monitorización de la instalación.
- Mostar los valores de generación de cada panel solar en vez del total de los paneles solares.

Algunas de estas tareas podrían suponer unos cambios bastante grandes y requerir un rediseño de la interfaz o creación de interfaces nuevas, pero basándose en el modelo actual no sería una tarea muy ardua dado que se puede reutilizar gran parte del trabajo realizado.

7. Referencias

Estilo de referencias y citas según las indicaciones de IEEE.

- [1] «Las energías renovables en el sistema eléctrico español 2020,» [En línea]. Available: <https://www.ree.es/es/datos/publicaciones/informe-de-energias-renovables/informe-2020>.
- [2] «La situación actual de la energía fotovoltaica en España,» [En línea]. Available: https://www.energiza.org/index.php?option=com_k2&view=item&id=1345.
- [3] «Aplicaciones Web Progresivas, PWA,» [En línea]. Available: <https://web.dev/what-are-pwas>.
- [4] «Starbucks PWA,» [En línea]. Available: <https://app.starbucks.com>.
- [5] «Twitter PWA,» [En línea]. Available: <https://mobile.twitter.com>.
- [6] «Spotify PWA,» [En línea]. Available: <https://open.spotify.com>.
- [7] «Programación orientada a objetos, POO,» [En línea]. Available: <https://www.educative.io/blog/object-oriented-programming>.
- [8] «Youtube - ¿Sabes qué es el EFECTO FOTOELÉCTRICO?,» [En línea]. Available: <https://youtu.be/5CLj9uJPQKg>.
- [9] «How a photovoltaic cell works,» [En línea]. Available: <https://www.planete-energies.com/en/medias/close/how-does-photovoltaic-cell-work>.
- [10] «I2C communication protocol,» [En línea]. Available: <https://www.geeksforgeeks.org/i2c-communication-protocol>.
- [11] «Basics of the I2C communication protocol,» [En línea]. Available: <https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol>.
- [12] «Resistencias pull up,» [En línea]. Available: <https://programarfacil.com/blog/arduino-blog/resistencia-pull-up-y-pull-down>.
- [13] «Amplificadores operacionales,» [En línea]. Available: https://www.electronicstutorials.ws/opamp/opamp_1.html.
- [14] «Raspberry Pi 4,» [En línea]. Available: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications>.
- [15] «Arduino Nano Every,» [En línea]. Available: <https://store.arduino.cc/arduino-nano-every>.

- [16] «Firebase,» [En línea]. Available: <https://firebase.google.com>.
- [17] «Firestore,» [En línea]. Available: <https://firebase.google.com/products/firestore>.
- [18] «Firebase Auth,» [En línea]. Available: <https://firebase.google.com/products/auth>.
- [19] «Google Analytics,» [En línea]. Available: <https://firebase.google.com/products/analytics>.
- [20] «Firebase Hosting,» [En línea]. Available: <https://firebase.google.com/products/hosting>.
- [21] «Visual Studio Code,» [En línea]. Available: <https://code.visualstudio.com/>.
- [22] «OneDrive,» [En línea]. Available: <https://www.microsoft.com/es-es/microsoft-365/onedrive/online-cloud-storage>.
- [23] «Microsoft Edge,» [En línea]. Available: <https://www.microsoft.com/en-us/edge>.
- [24] «Microsoft Edge DevTools,» [En línea]. Available: <https://docs.microsoft.com/en-us/microsoft-edge/devtools-guide-chromium>.
- [25] «PyDoc,» [En línea]. Available: <https://docs.python.org/3/library/pydoc.html>.
- [26] «JSDoc,» [En línea]. Available: <https://github.com/jsdoc/jsdoc>.
- [27] «Tinkercad,» [En línea]. Available: <https://www.tinkercad.com/learn/circuits>.
- [28] «CircuitJs,» [En línea]. Available: <https://www.falstad.com/circuit/circuitjs.html>.
- [29] «ChartJS,» [En línea]. Available: <https://www.chartjs.org/>.
- [30] «D3.JS,» [En línea]. Available: <https://d3js.org>.
- [31] «MCDatepicker,» [En línea]. Available: <https://github.com/mikecoj/MCDatepicker>.
- [32] «Microsoft Project,» [En línea]. Available: <https://www.microsoft.com/es-es/microsoft-365/project/project-management-software>.
- [33] «Visual Paradigm,» [En línea]. Available: <https://www.visual-paradigm.com/>.
- [34] «Adafruit-ADS1115,» [En línea]. Available: <https://www.adafruit.com/product/1085>.

8. Bibliografía

En este apartado se mostrarán los materiales usados durante el estudio y desarrollo del trabajo.

8.1. Materiales didácticos consultados

- Prácticas de Administración de sistemas, Gabriel Villarrubia González, Informática y Automática, Universidad de Salamanca (USAL), curso 2019-2020.
- Teoría UML de Ingeniería del software I, Francisco José García Peñalvo y María N. Moreno García, Informática y Automática, Universidad de Salamanca (USAL), curso 2019-2020.
- Teoría UML de Ingeniería del software II, María N. Moreno García, Informática y Automática, Universidad de Salamanca (USAL), curso 2019-2020.
- Prácticas de Gestión de proyectos, María N. Moreno García, Informática y Automática, Universidad de Salamanca (USAL), curso 2020-2021.

8.2. Sitios web consultados

Desarrollo web:

- **Aplicaciones web progresivas:**
<https://web.dev/progressive-web-apps>
- **CSS-Flexbox:**
<https://css-tricks.com/snippets/css/a-guide-to-flexbox>
- **CSS-Grid:**
<https://css-tricks.com/snippets/css/complete-guide-grid>
- **Chart.js:**
<https://www.chartjs.org>
- **Tutorial Chart.js:**
<https://youtu.be/sE08f4iuOhA>
- **MCDatepicker:**
<https://youtu.be/sE08f4iuOhA>
- **MCDatepicker docs:**
<https://mcdaterangepicker.netlify.app/docs>

I2C y otros protocolos de comunicación:

- **UART vs SPI vs I2C:**
<https://www.seeedstudio.com/blog/2019/09/25/uart-vs-i2c-vs-spi-communication-protocols-and-uses>

- **Diferencias entre I2C y SPI:**
<https://aticleworld.com/difference-between-i2c-and-spi>
<https://articles.saleae.com/logic-analyzers/spi-vs-i2c-protocol-differences-and-things-to-consider>
- **Pines I2C Raspberry PI:**
<https://pinout.xyz/pinout/i2c>
- **Programación I2C Python:**
<https://www.abelectronics.co.uk/kb/article/1094/i2c-part-4---programming-i-c-with-python>
- **Librería Python I2C:**
<https://pypi.org/project/smbus2>
- **Conectando dos Arduino Nano Every con I2C:**
<https://docs.arduino.cc/tutorials/nano-every/i2c#circuit>

Servicios Linux:

- **Bases de Systemd, Services, Units and the Journal:**
<https://www.digitalocean.com/community/tutorials/systemd-essentials-working-with-services-units-and-the-journal>
- **Creación de un servicio Linux, Systemd:**
<https://medium.com/@benmorel/creating-a-linux-service-with-systemd-611b5c8b91d6>
- **Entendiendo las Systemd Units y los archivos Unit:**
<https://www.digitalocean.com/community/tutorials/understanding-systemd-units-and-unit-files>

Electrónica:

- **Amplificadores operacionales:**
<https://www.electronics-tutorials.ws/opamp/op-amp-building-blocks.html>
- **Medición de potencia corriente alterna usando ESP32:**
<https://youtu.be/PSzkaSy5IHY>
- **Medición de potencia corriente alterna con Arduino:**
<https://learn.openenergymonitor.org/electricity-monitoring/ac-power-theory/arduino-maths>

Firestore:

- **Firestore web:**
<https://firebase.google.com>
- **Firestore JavaScript SDK:**
<https://firebase.google.com/docs/reference/js>
- **Firestore Admin Python SDK:**
<https://firebase.google.com/docs/reference/admin/python>

9. Agradecimientos

Personas que han facilitado el desarrollo de este proyecto:

- Ayuda en el diseño gráfico y la identidad visual de la aplicación:
 - María Sánchez González

Aplicación para la gestión de una instalación fotovoltaica

Anexo I – Planificación temporal



VNiVERSiDAD
D SALAMANCA

Trabajo de Fin de Grado

Grado de Ingeniería Informática

Septiembre 2021

Tutor:

Iván Álvarez Navia

Alumno:

Miguel Sánchez González

Tabla de contenidos

1. Introducción	7
2. Estimación de costes y esfuerzo	8
2.1. Puntos de Caso de Uso Desajustados (UUCP)	8
2.1.1. Complejidad de los actores (UAW).....	8
2.1.2. Complejidad de Casos de Uso (UUCW)	8
2.2. Cálculo de los TCF	10
2.3. Cálculo de los ECF	11
2.4. Resultados.....	13
3. Planificación temporal	14
3.1. Distribución de actividades.....	14
3.2. Calendario de trabajo	15
3.3. Planificación de tareas	16
3.4. Distribución de recursos	19
3.5. Diagrama de Gantt.....	20
3.6. Resultado final	23

Tabla de ilustraciones

Ilustración 1: Programa EZEstimate	13
Ilustración 2: Proceso unificado	14
Ilustración 3: Calendario de trabajo	15
Ilustración 4: Planificación de tareas Inicio	16
Ilustración 5: Planificación de tareas Elaboración It1	17
Ilustración 6: Planificación de tareas Elaboración It2	17
Ilustración 7: Planificación de tareas Construcción.....	18
Ilustración 8: Planificación de tareas Transición	19
Ilustración 9: Gráfico de distribución de recursos.....	19
Ilustración 10: Diagrama de Gantt Inicio.....	20
Ilustración 11: Diagrama de Gantt Elaboración It1	21
Ilustración 12: Diagrama de Gantt Elaboración It2	21
Ilustración 13: Diagrama de Gantt Construcción It1	22
Ilustración 14: Diagrama de Gantt Construcción It2	22
Ilustración 15: Diagrama de Gantt Transición	23
Ilustración 16: Resultado estimación de esfuerzo	23

Índice de tablas

Tabla 1: Tabla de complejidad de los actores del sistema	8
Tabla 2: Tabla de complejidad de Gestión de los roles del sistema.....	9
Tabla 3: Tabla de complejidad de Gestión de la visualización de datos	9
Tabla 4: Tabla de complejidad de Gestión de la monitorización del sistema.....	10
Tabla 5: Factores de complejidad técnica	11
Tabla 6: Factores de complejidad de entorno.....	12

1. Introducción

En este anexo se va a presentar la planificación temporal que se debe llevar a cabo para el desarrollo del proyecto.

Este documento consta de los apartados siguientes:

- Estimación de costes y esfuerzo, nos dará una idea de los recursos y el tiempo que han de ser empleados para llevar a cabo el proyecto. La estimación se realiza usando los casos de uso del sistema y la complejidad de los actores.
- La planificación temporal se realiza con los datos del apartado anterior, asignando tareas en un calendario y los recursos que se usarán para cada tarea. Una vez finalizada esta planificación se obtiene un diagrama de Gantt que permite llevar a cabo un seguimiento del proyecto, así como tener una visión general del desarrollo.

2. Estimación de costes y esfuerzo

Al realizar el cálculo de los costes y esfuerzo obtenemos unos valores aproximados de cuantos recursos se van a necesitar para el desarrollo de un proyecto.

Los cálculos realizados en este apartado se han hecho usando los diagramas del Anexo II – Especificación de requisitos.

La estimación se realiza usando UCP (Puntos de caso de uso). Sirven para medir la funcionalidad de los casos de uso y se calculan a partir de los puntos de caso de uso desajustados (UUCP), el factor de complejidad técnica (TCF) y el factor de complejidad de entorno (ECF), con el uso de la siguiente fórmula:

$$UCP = UUCP \cdot TCF \cdot ECF$$

2.1. Puntos de Caso de Uso Desajustados (UUCP)

Los puntos de caso de uso sin ajustar (UUCP) se calculan obteniendo el peso la complejidad de los actores sin ajustar (UAW) y multiplicándolo por el peso de la complejidad de los casos de uso sin ajustar (UUCW):

$$UUCP = UUCW \cdot UAW$$

2.1.1. Complejidad de los actores (UAW)

Para determinar la complejidad de los actores se usan las siguientes reglas:

- **Simple:** 3 transacciones o menos
- **Medio:** de 4 a 7 transacciones
- **Complejo:** el actor es una persona y tendrá que usar una interfaz gráfica.

Actor	Complejidad
ACT-0001 Usuario no registrado	Complejo
ACT-0002 Usuario registrado	Complejo
ACT-0003 Sistema	Medio

Tabla 1: Tabla de complejidad de los actores del sistema

2.1.2. Complejidad de Casos de Uso (UUCW)

La clasificación de los casos de uso según su complejidad se realiza contabilizando transacciones:

- **Simple:** 3 transacciones o menos
- **Medio:** de 4 a 7 transacciones
- **Complejo:** más de 7 transacciones

Paquete Gestión de los roles del sistema:

Caso de uso	Número de transacciones	Complejidad
UC-0001 Registrarse	2	Simple
UC-0002 Validar correo electrónico	1	Simple
UC-0003 Recuperar contraseña	2	Simple
UC-0004 Iniciar sesión	2	Simple
UC-0005 Cerrar sesión	1	Simple
UC-0006 Modificar correo electrónico	1	Simple
UC-0007 Modificar contraseña	1	Simple
UC-0008 Eliminar cuenta	1	Simple
UC-0009 Ver perfil	1	Simple
UC-0010 Enviar correo electrónico de cambio de contraseña	1	Simple
UC-0017 Obtener UID	1	Simple

Tabla 2: Tabla de complejidad de Gestión de los roles del sistema

Paquete Gestión de la visualización de datos:

Caso de uso	Número de transacciones	Complejidad
UC-0011 Ver página principal	1	Simple
UC-0012 Mostrar datos de los relojes en tiempo real	1	Simple
UC-0013 Mostrar datos estadísticos	1	Simple
UC-0014 Mostrar gráfico de la potencia generada	1	Simple
UC-0015 Mostrar gráfico de la batería	1	Simple
UC-0016 Mostrar gráfico de la potencia consumida	1	Simple

Tabla 3: Tabla de complejidad de Gestión de la visualización de datos

Paquete Gestión de la monitorización del sistema:

Caso de uso	Número de transacciones	Complejidad
UC-0018 Inicialización de la monitorización	1	Simple
UC-0019 Leer información de los sensores	1	Simple
UC-0020 Enviar información a la base de datos	1	Simple

Tabla 4: Tabla de complejidad de Gestión de la monitorización del sistema

2.2. Cálculo de los TCF

Para realizar el cálculo de los factores de complejidad técnica (TCF) se aplica un valor del 0 al 5 indicando mayor relevancia cuanto mayor es el número.

Para su cálculo se aplica la siguiente fórmula:

$$TCF = C_1 + C_2 \cdot \sum (W_i \cdot F_i)$$

Factor	Complejidad	Explicación
Sistemas distribuidos	2	La aplicación en si se desarrollará en un entorno distribuido, pero gran parte de este trabajo recaerá en la aplicación de Firebase por lo que solo se deberá realizar un pequeño aprendizaje del uso de la plataforma.
Rendimiento	2	El sistema debe responder rápidamente a las peticiones de los usuarios, la mayor carga de trabajo de este apartado será al tratar los datos, en los dispositivos del usuario tanto móviles como
Eficiencia del usuario final	3	Se desea hacer una interfaz eficiente e intuitiva para que el usuario pueda realizar las tareas de forma satisfactoria.
Procesamiento interno complejo	4	El procesamiento de los datos obtenidos de la base de datos no es trivial y requerirá de ciertos algoritmos y funciones para su representación.
Reusabilidad	3	Se crearán ciertas librerías que permitirán reusar código para diferentes funciones.
Facilidad de instalación	1	Al ser una aplicación web el usuario no tendrá que realizar ningún tipo de instalación, salvo ejecutarla en local como PWA.

Facilidad de uso	3	Se intentará obtener una herramienta fácil de usar para cualquier usuario, pero dado que la cantidad de elementos no es muy grande no será muy difícil.
Portabilidad	0	Al tratarse de una aplicación web debería ser soportada en cualquier plataforma con un navegador.
Facilidad de cambio	3	Se intentará facilitar en la medida de lo posible la adición de nuevas funciones al sistema.
Concurrencia	3	Se deberá idear una forma para gestionar la concurrencia de lectura de los sensores de la instalación.
Características especiales de seguridad	2	Las únicas características de seguridad que se deberán implementar serán las autorizaciones de acceso a los datos de los usuarios.
Acceso directo a terceras partes	0	No se facilitará acceso directo a terceras partes.
Se requiere entrenamiento especial del usuario	1	Al ser una aplicación web con una interfaz gráfica intuitiva, solo serán necesarios unos pequeños conocimientos sobre la materia.

Tabla 5: Factores de complejidad técnica

2.3. Cálculo de los ECF

Al igual que con los (TCF), para realizar el cálculo de los factores de complejidad de entorno (ECF) se aplica un valor del 0 al 5 indicando mayor relevancia cuanto mayor es el número.

Para su cálculo se aplica la siguiente fórmula:

$$ECF = C_1 + C_2 \cdot \sum (W_i \cdot F_i)$$

Factor	Complejidad	Explicación
Familiaridad con UML	3	Durante la carrera se han tenido ya algunos contactos con el lenguaje UML, obteniendo los conocimientos necesarios para el desarrollo de este trabajo.
Trabajadores a tiempo parcial	2	El proyecto cuenta con un único desarrollador que dedicará la mayor parte de su tiempo a este proyecto.
Capacidad de los analistas	2	Durante los estudios se ha realizado varios trabajos que han proporcionado experiencia como analista, pero en el fondo no es una experiencia real que

		pueda facilitar mucho el desarrollo del proyecto.
Experiencia en la aplicación	2	Durante la carrera se han trabajado en algunas partes similares a la aplicación que se va a desarrollar, pero por otra parte hay algunas que nunca se han tratado.
Experiencia en orientación a objetos	4	El desarrollador cuenta ya con bastante experiencia en lenguajes de programación orientados a objetos por lo que la creación de clases y gestión de los programas no debería suponer mucha dificultad.
Motivación	5	La motivación inicial de este proyecto es bastante alta ya que es el último trabajo de la carrera.
Dificultad del lenguaje de programación	3	Durante la carrera se ha trabajado con una parte de los lenguajes de programación con los que se va a trabajar en esta aplicación y los que no son lenguajes de relativo alto nivel por lo que no debería ser muy difícil su aprendizaje.
Estabilidad de los requisitos	5	Los requisitos inicialmente propuestos difícilmente van a cambiar a lo largo del desarrollo ya que se han planteado pensando en muchos de sus aspectos y sabiendo que su cambio puede conllevar una pérdida de tiempo significativa.

Tabla 6: Factores de complejidad de entorno

2.4. Resultados

He utilizado el programa EZEstimate, junto con los resultados obtenidos anteriormente para realizar la estimación de costes y esfuerzo:

The screenshot shows the EZEstimate software interface with the following sections:

- Module:** A dropdown menu showing "Gestión de la monitorización del sistema" with "Add Module" and "Delete" buttons.
- Summary:** Fields for "Total Modules" (4), "Use cases" (Simple: 20, Average: 0, Complex: 0), and "Actors" (Simple: 0, Average: 1, Complex: 2). Includes an "Excel Report" button.
- Add Actor / Use case:** Fields for "Actor / Use case Name", "Select Type" (Usecase), and "Complexity" (Simple), with an "Add" button.
- Tech / Env Factors:** "Set Tech Factor" and "Set Env Factors" buttons.
- Estimation Summary:** A list of calculation fields:
 - UAW: 8
 - UUCW: 100
 - UUCP = UAW + UUCW: 108
 - TFactor: 29
 - EFactor: 20
 - TCF = 0.6 + (.01*TFactor): 0.89
 - EF = 1.4 + (-0.03*EFactor): 0.8
 - UCP = UUCP*TCF*EF: 76,896
 - Total Effort@ 10 Hrs/UCP: 768.96**
- Use case / Actor List:** A table listing project items with columns for Id, Module, Type, Name, and complexity.

Id	Module	Type	Name	complexity
1	Actores	Actor	Usuario no regi...	Complex
10	Gestion de los r...	Usecase	UC-0007 Modifi...	Simple
11	Gestion de los r...	Usecase	UC-0008 Elimina...	Simple
12	Gestion de los r...	Usecase	UC-0009 Ver p...	Simple
13	Gestion de los r...	Usecase	UC-0010 Enviar...	Simple
14	Gestion de los r...	Usecase	UC-0017 Obten...	Simple
15	Gestión de la vi...	Usecase	UC-0011 Ver p...	Simple
16	Gestión de la vi...	Usecase	UC-0012 Mostr...	Simple
17	Gestión de la vi...	Usecase	UC-0013 Mostr...	Simple
18	Gestión de la vi...	Usecase	UC-0014 Mosta...	Simple
19	Gestión de la vi...	Usecase	UC-0015 Mosta...	Simple
2	Actores	Actor	Usuario registra...	Complex
20	Gestión de la vi...	Usecase	UC-0016 Mosta...	Simple
21	Gestión de la m...	Usecase	UC-0018 Iniciali...	Simple
22	Gestión de la m...	Usecase	UC-0019 Leer i...	Simple
23	Gestión de la m...	Usecase	UC-0020 Enviar...	Simple
3	Actores	Actor	Sistema	Average
4	Gestion de Ins r	Usecase	UC-0001 Regist...	Simple

Ilustración 1: Programa EZEstimate

Los resultados obtenidos muestran una puntuación de 76.896 UCP al que se le ha aplicado un factor de corrección de 10 dado que no es un proyecto muy grande, para proyectos medianos por defecto se aplica un factor de 20. Con esto hemos obtenido que serán necesarias un total de 768.96 horas de persona para poder llevar a cabo este proyecto.

3. Planificación temporal

En este apartado se va a ver la realización de la planificación temporal, detallando la división del proyecto en tareas teniendo en cuenta su tiempo de desarrollo.

Para realizar esta planificación se ha usado el software Microsoft Project.

3.1. Distribución de actividades

Para comentar como se han distribuido las diferentes tareas, nos vamos a apoyar en un gráfico de los apuntes de la asignatura Gestión de proyectos de este grado. Este gráfico está basado en el Proceso Unificado.

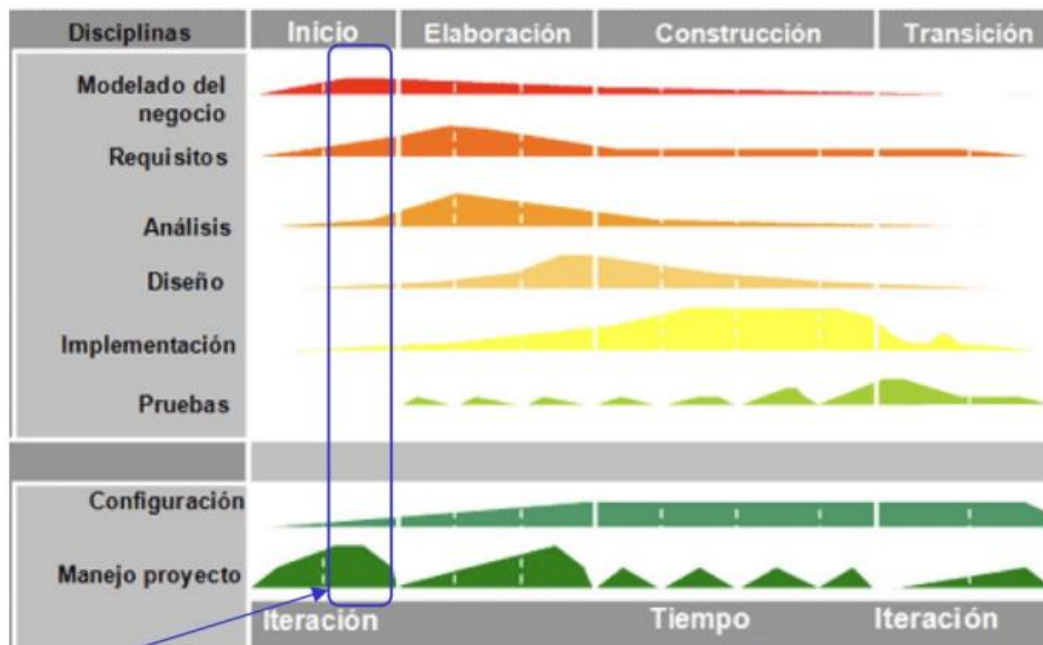


Ilustración 2: Proceso unificado

Como se puede ver hay diferentes fases en el proyecto, dentro de cada fase habrá una serie de iteraciones que se subdividirán a su vez en 6 disciplinas. Se ha intentado en la medida de lo posible repartir la carga del trabajo para que se asemeje al reparto del gráfico anterior.

Las fases que aparecen en el gráfico son:

- **Inicio:** es el comienzo del proyecto, en él se define el alcance del proyecto y se desarrolla el modelo de negocio.
- **Elaboración:** en esta fase se planifica el proyecto especificando la mayor parte de los casos de uso y definiendo la arquitectura del sistema.
- **Construcción:** se desarrolla el proyecto.
- **Transición:** se realiza una revisión general del sistema completo y se asegura el correcto funcionamiento de este.

3.2. Calendario de trabajo

La herramienta Microsoft Project ofrece un calendario laboral sobre el que realizar la planificación de las tareas, se ha propuesto un modelo de trabajo de 10 horas diarias con descansos los sábados y domingos. Tal y como se puede ver en la Ilustración 3.

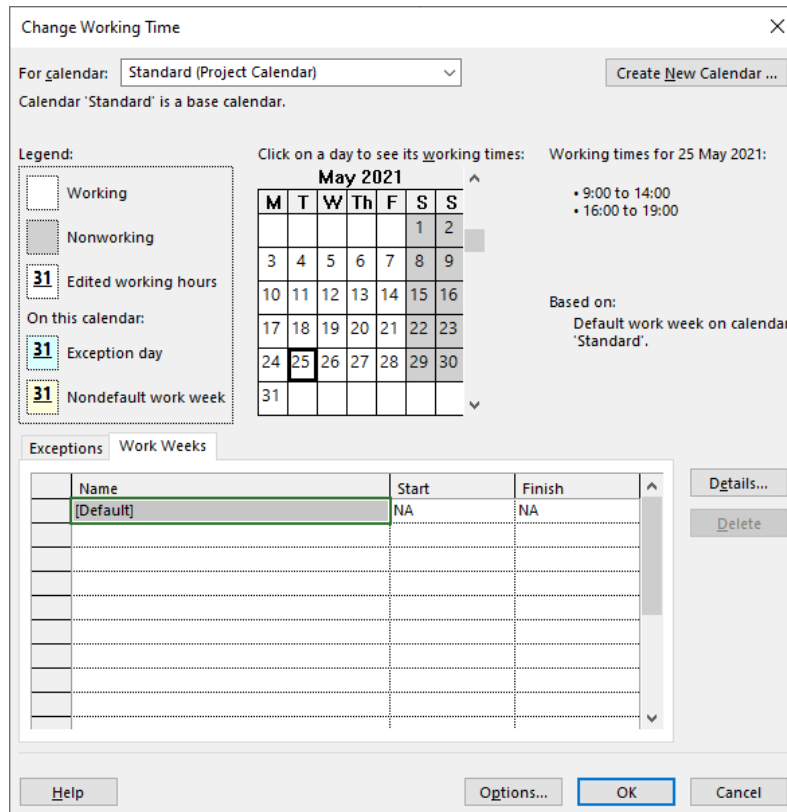


Ilustración 3: Calendario de trabajo

3.3. Planificación de tareas

En este apartado se verá la lista de tareas del proyecto, con su duración y relaciones de precedencia entre ellas. Todo ello siguiendo una planificación guiándose por el Proceso Unificado.

	i	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1			Inicio	15,5 days	Tue 01/06/21	Tue 22/06/21		
2			Iteración 1	15,5 days	Tue 01/06/21	Tue 22/06/21		
3			Modelado del negocio	2,25 days	Tue 01/06/21	Thu 03/06/21		
4			Planificación provisional del modelo de trabajo	2 days	Tue 01/06/21	Wed 02/06/21		Miguel[50%]
5			Estudio de mercado	2 days	Tue 01/06/21	Wed 02/06/21		Miguel[50%]
6			Reunión con el tutor	0,25 days	Thu 03/06/21	Thu 03/06/21	5	Miguel
7			Requisitos	5,25 days	Thu 03/06/21	Thu 10/06/21	3	
8			Definición de actores	0,25 days	Thu 03/06/21	Thu 03/06/21		Miguel
9			Establecimiento de los Objetivos del sistema	1 day	Thu 03/06/21	Fri 04/06/21	8	Miguel
10			Requisitos de almacenamiento de información	1 day	Fri 04/06/21	Mon 07/06/21	9	Miguel
11			Requisitos no funcionales	1 day	Mon 07/06/21	Tue 08/06/21	10	Miguel
12			Requisitos funcionales	2 days	Tue 08/06/21	Thu 10/06/21	11	Miguel
13			Análisis	1 day	Thu 10/06/21	Fri 11/06/21	7	
14			Creación de posibles interfaces	1 day	Thu 10/06/21	Fri 11/06/21		Miguel
15			Diseño	2 days	Fri 11/06/21	Tue 15/06/21	13	
16			Diseño de la arquitectura de la base de datos	2 days	Fri 11/06/21	Tue 15/06/21		Miguel
17			Implementación	3 days	Tue 15/06/21	Fri 18/06/21	15	
18			Elección e instalación de herramientas	1 day	Tue 15/06/21	Wed 16/06/21		Miguel
19			Configuración inicial del ordenador de sensores	1 day	Wed 16/06/21	Thu 17/06/21	18	Miguel
20			Creación del espacio de trabajo en VSCode	1 day	Thu 17/06/21	Fri 18/06/21	19	Miguel
21			Pruebas	2 days	Fri 18/06/21	Tue 22/06/21	17	
22			Pruebas de lectura de sensores	2 days	Fri 18/06/21	Tue 22/06/21		Miguel
23			Fin iteración 1	0 days	Tue 22/06/21	Tue 22/06/21	2	
24			Fin inicio	0 days	Tue 22/06/21	Tue 22/06/21	1	

Ilustración 4: Planificación de tareas Inicio

	i	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
25			Elaboración	25,25 days	Tue 22/06/21	Tue 27/07/21	24	
26			Iteración 1	16 days	Tue 22/06/21	Wed 14/07/21		
27			Modelado del negocio	2,25 days	Tue 22/06/21	Thu 24/06/21		
28			Diario: avances, investigación y problemas	1 day	Tue 22/06/21	Wed 23/06/21		Miguel
29			Estimación de esfuerzo	1 day	Wed 23/06/21	Thu 24/06/21	28	Miguel[50%]
30			Estimación temporal	1 day	Wed 23/06/21	Thu 24/06/21	28	Miguel[50%]
31			Reunión con el tutor	0,25 days	Thu 24/06/21	Thu 24/06/21	29;30	Miguel
32			Requisitos	2,75 days	Thu 24/06/21	Tue 29/06/21	27	
33			Refinamiento de requisitos	0,5 days	Thu 24/06/21	Fri 25/06/21		Miguel
34			Diagrama de paquetes	0,25 days	Fri 25/06/21	Fri 25/06/21	33	Miguel
35			Diagrama de casos de uso	1 day	Fri 25/06/21	Mon 28/06/21	34	Miguel
36			Descripciones de casos de uso	1 day	Mon 28/06/21	Tue 29/06/21	35	Miguel
37			Análisis	3 days	Tue 29/06/21	Fri 02/07/21	32	
38			Primer modelo de dominio	1,5 days	Tue 29/06/21	Wed 30/06/21		Miguel
39			Primer modelo de análisis	1,5 days	Thu 01/07/21	Fri 02/07/21	38	Miguel
40			Diseño	3 days	Fri 02/07/21	Wed 07/07/21	37	
41			Primer modelo de arquitectura	1 day	Fri 02/07/21	Mon 05/07/21		Miguel
42			Primer paquete de análisis	1 day	Mon 05/07/21	Tue 06/07/21	41	Miguel
43			Revisión del diseño de la base de datos	1 day	Tue 06/07/21	Wed 07/07/21	42	Miguel
44			Implementación	3 days	Wed 07/07/21	Mon 12/07/21	40	
45			Creación del proyecto de firebase	3 days	Wed 07/07/21	Mon 12/07/21		Miguel
46			Pruebas	2 days	Mon 12/07/21	Wed 14/07/21	44	
47			Pruebas de sensores en la Raspberry Pi	2 days	Mon 12/07/21	Wed 14/07/21		Miguel[50%]
48			Pruebas en el proyecto de firebase	2 days	Mon 12/07/21	Wed 14/07/21		Miguel[50%]
49			Fin iteración 1	0 days	Wed 14/07/21	Wed 14/07/21	26	

Ilustración 5: Planificación de tareas Elaboración It1

	i	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
49			Fin iteración 1	0 days	Wed 14/07/21	Wed 14/07/21	26	
50			Iteración 2	9,25 days	Wed 14/07/21	Tue 27/07/21	49	
51			Modelado del negocio	1,25 days	Wed 14/07/21	Thu 15/07/21		
52			Diario: avances, investigación y problemas	1 day	Wed 14/07/21	Thu 15/07/21		Miguel
53			Reunión con el tutor	0,25 days	Thu 15/07/21	Thu 15/07/21	52	Miguel
54			Requisitos	1,5 days	Thu 15/07/21	Mon 19/07/21	51	
55			Revisión de requisitos	0,5 days	Thu 15/07/21	Fri 16/07/21		Miguel
56			Revisión de casos de uso y descripciones	1 day	Fri 16/07/21	Mon 19/07/21	55	Miguel
57			Análisis	3 days	Mon 19/07/21	Thu 22/07/21	54	
58			Revisión modelo de dominio y análisis	0,5 days	Mon 19/07/21	Mon 19/07/21		Miguel
59			Diagramas de secuencia	2 days	Mon 19/07/21	Wed 21/07/21	58	Miguel
60			Vista de arquitectura	0,5 days	Wed 21/07/21	Thu 22/07/21	59	Miguel
61			Diseño	1 day	Thu 22/07/21	Fri 23/07/21	57	
62			Revisión Modelo de arquitectura	0,5 days	Thu 22/07/21	Thu 22/07/21		Miguel
63			Revisión paquete de análisis	0,5 days	Thu 22/07/21	Fri 23/07/21	62	Miguel
64			Implementación	1,5 days	Fri 23/07/21	Mon 26/07/21	61	
65			Configuración reglas de la base de datos	0,5 days	Fri 23/07/21	Fri 23/07/21		Miguel
66			Inicio del programa de monitorización	1 day	Fri 23/07/21	Mon 26/07/21	65	Miguel
67			Pruebas	1 day	Mon 26/07/21	Tue 27/07/21	64	
68			Pruebas subir datos a la base de datos	1 day	Mon 26/07/21	Tue 27/07/21		Miguel
69			Fin iteración 2	0 days	Tue 27/07/21	Tue 27/07/21	50	
70			Fin elaboración	0 days	Tue 27/07/21	Tue 27/07/21	25	

Ilustración 6: Planificación de tareas Elaboración It2

	i	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
71			Construcción	36 days	Tue 27/07/21	Wed 15/09/21	70	
72			Iteración 1	17,25 days	Tue 27/07/21	Thu 19/08/21		
73			Modelado del negocio	1,25 days	Tue 27/07/21	Wed 28/07/21		
74			Diario: avances, investigación y problemas	1 day	Tue 27/07/21	Wed 28/07/21		Miguel
75			Reunión con el tutor	0,25 days	Wed 28/07/21	Wed 28/07/21	74	Miguel
76			Requisitos	0,5 days	Thu 29/07/21	Thu 29/07/21	73	
77			Revisión requisitos	0,5 days	Thu 29/07/21	Thu 29/07/21		Miguel
78			Análisis	0,5 days	Thu 29/07/21	Thu 29/07/21	76	
79			Revisión diagramas análisis	0,5 days	Thu 29/07/21	Thu 29/07/21		Miguel
80			Diseño	1 day	Fri 30/07/21	Fri 30/07/21	78	
81			Revisión diagramas de diseño	0,5 days	Fri 30/07/21	Fri 30/07/21		Miguel
82			Revisión arquitectura base de datos	0,5 days	Fri 30/07/21	Fri 30/07/21	81	Miguel
83			Implementación	12 days	Mon 02/08/21	Tue 17/08/21	80	
84			Creación de servicio linux	3 days	Mon 02/08/21	Wed 04/08/21		Miguel[50%]
85			Programa de monitorización	6 days	Mon 02/08/21	Mon 09/08/21		Miguel[50%]
86			Código medidor de potencia	3 days	Thu 05/08/21	Mon 09/08/21	84	Miguel[50%]
87			Código aplicación web	6 days	Tue 10/08/21	Tue 17/08/21	86	Miguel[50%]
88			Circuitos de los sensores	6 days	Tue 10/08/21	Tue 17/08/21	86	Miguel[50%]
89			Pruebas	2 days	Wed 18/08/21	Thu 19/08/21	83	
90			Pruebas monitorización	2 days	Wed 18/08/21	Thu 19/08/21		Miguel[50%]
91			Pruebas aplicación web	2 days	Wed 18/08/21	Thu 19/08/21		Miguel[50%]
92			Fin iteración 1	0 days	Thu 19/08/21	Thu 19/08/21	89	
93			Iteración 2	18,75 days	Fri 20/08/21	Wed 15/09/21	72	
94			Modelado del negocio	1,25 days	Fri 20/08/21	Mon 23/08/21		
95			Diario: avances, investigación y problemas	1 day	Fri 20/08/21	Fri 20/08/21		Miguel
96			Reunión con el tutor	0,25 days	Mon 23/08/21	Mon 23/08/21	95	Miguel
97			Requisitos	0,5 days	Mon 23/08/21	Mon 23/08/21	94	
98			Revisión requisitos	0,5 days	Mon 23/08/21	Mon 23/08/21		Miguel
99			Análisis	0,5 days	Mon 23/08/21	Tue 24/08/21	97	
100			Revisión diagramas análisis	0,5 days	Mon 23/08/21	Tue 24/08/21		Miguel
101			Diseño	0,5 days	Tue 24/08/21	Tue 24/08/21	99	
102			Revisión diagramas de diseño	0,5 days	Tue 24/08/21	Tue 24/08/21		Miguel
103			Implementación	14 days	Tue 24/08/21	Mon 13/09/21	101	
104			Programa de monitorización	4 days	Tue 24/08/21	Mon 30/08/21		Miguel
105			Código aplicación web	7 days	Mon 30/08/21	Wed 08/09/21	104	Miguel
106			Refinamiento circuitos de los se	3 days	Wed 08/09/21	Mon 13/09/21	105	Miguel
107			Pruebas	2 days	Mon 13/09/21	Wed 15/09/21	103	
108			Pruebas monitorización	2 days	Mon 13/09/21	Wed 15/09/21		Miguel[50%]
109			Pruebas aplicación web	2 days	Mon 13/09/21	Wed 15/09/21		Miguel[50%]
110			Fin iteración 2	0 days	Wed 15/09/21	Wed 15/09/21	93	
111			Fin construcción	0 days	Wed 15/09/21	Wed 15/09/21	71	

Ilustración 7: Planificación de tareas Construcción

	i	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
112			Transición	6 days	Wed 15/09/21	Thu 23/09/21	111	
113			Iteración 1	6 days	Wed 15/09/21	Thu 23/09/21		
114			Modelado del negocio	0,25 days	Wed 15/09/21	Wed 15/09/21		
115			Reunión con el tutor	0,25 days	Wed 15/09/21	Wed 15/09/21		Miguel
116			Requisitos	0,25 days	Thu 16/09/21	Thu 16/09/21	114	
117			Revisión final Requisitos	0,25 days	Thu 16/09/21	Thu 16/09/21		Miguel
118			Análisis	0,25 days	Thu 16/09/21	Thu 16/09/21	116	
119			Revisión final diagramas de aná	0,25 days	Thu 16/09/21	Thu 16/09/21		Miguel
120			Diseño	0,25 days	Thu 16/09/21	Thu 16/09/21	118	
121			Revisión final diagramas de dise	0,25 days	Thu 16/09/21	Thu 16/09/21		Miguel
122			Implementación	3 days	Thu 16/09/21	Tue 21/09/21	120	
123			Refinamiento de la interfaz de l	1 day	Thu 16/09/21	Fri 17/09/21		Miguel
124			Documentación final de la aplic	2 days	Fri 17/09/21	Tue 21/09/21	123	Miguel
125			Pruebas	2 days	Tue 21/09/21	Thu 23/09/21	122	
126			Pruebas generales del proyecto	2 days	Tue 21/09/21	Thu 23/09/21		Miguel
127			Fin iteración 1	0 days	Thu 23/09/21	Thu 23/09/21	113	
128			Fin transición	0 days	Thu 23/09/21	Thu 23/09/21	112	

Ilustración 8: Planificación de tareas Transición

3.4. Distribución de recursos

Al asignar los recursos se pretende que la productividad sea máxima (cercana al 100%), según nuestro calendario de trabajo una jornada laboral de 10h/día, pero sin sobrecargar los recursos.

Como se puede ver en la Ilustración 9 se ha distribuido la carga de trabajo intentando obtener ese máximo rendimiento.

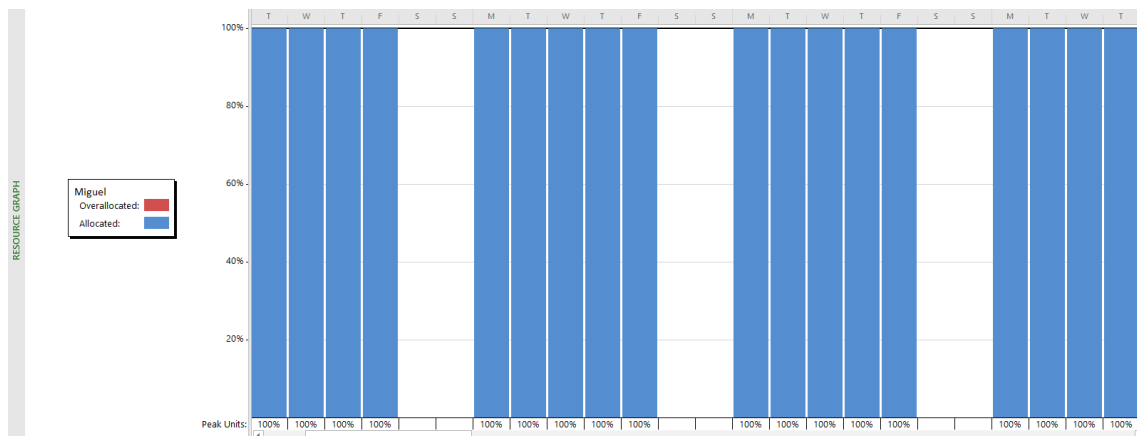


Ilustración 9: Gráfico de distribución de recursos

3.5. Diagrama de Gantt

El diagrama de Gantt representa todas las tareas propuestas en relación con su duración y las dependencias entre ellas. Esta representación se realiza mediante barras que indican la duración y flechas que indican las relaciones entre las tareas, además se muestra con barras de color rojo el camino crítico que debe seguir el desarrollo del proyecto, indicando las tareas que no pueden retrasarse ya que implicarían un retraso en todo el proyecto.

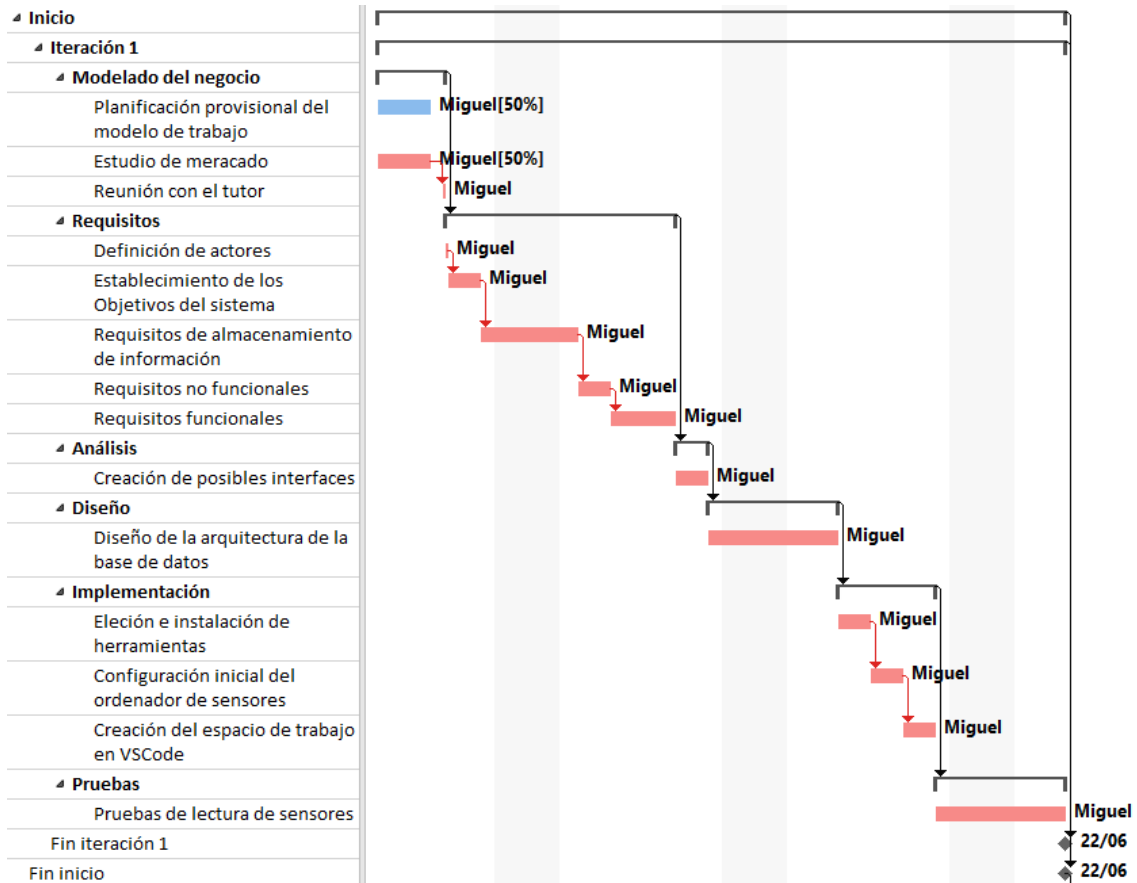


Ilustración 10: Diagrama de Gantt Inicio

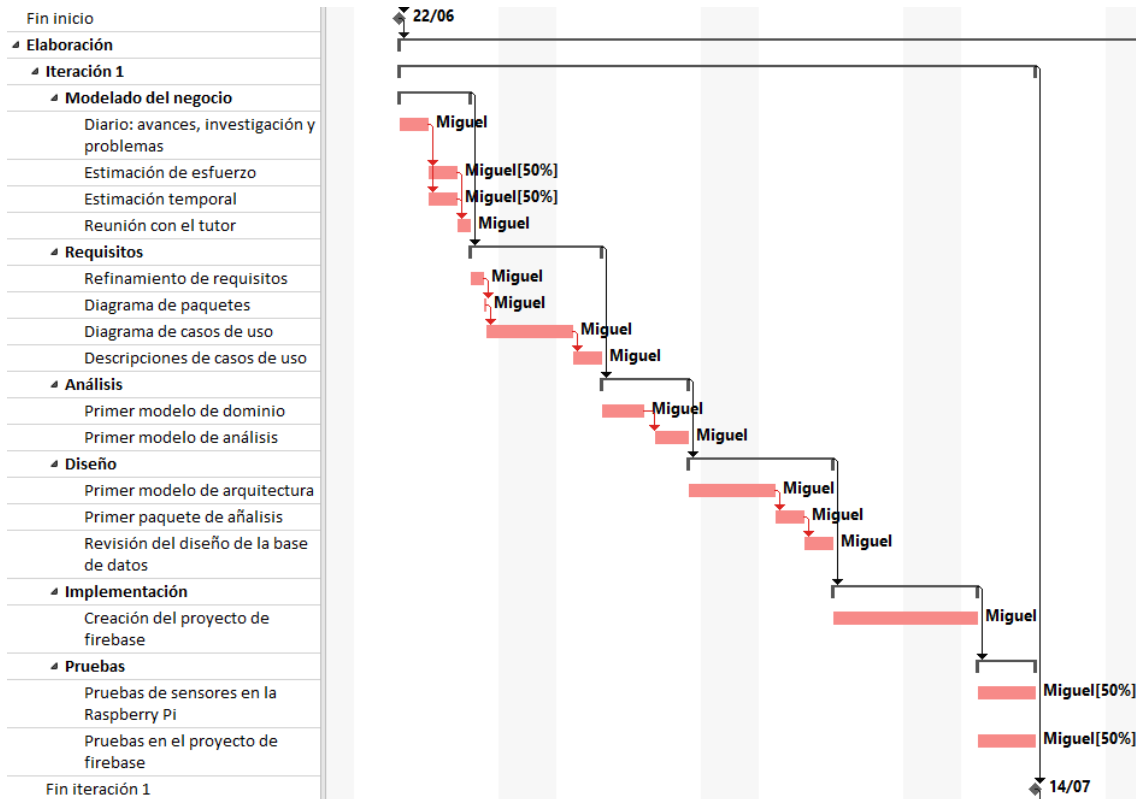


Ilustración 11: Diagrama de Gantt Elaboración It1

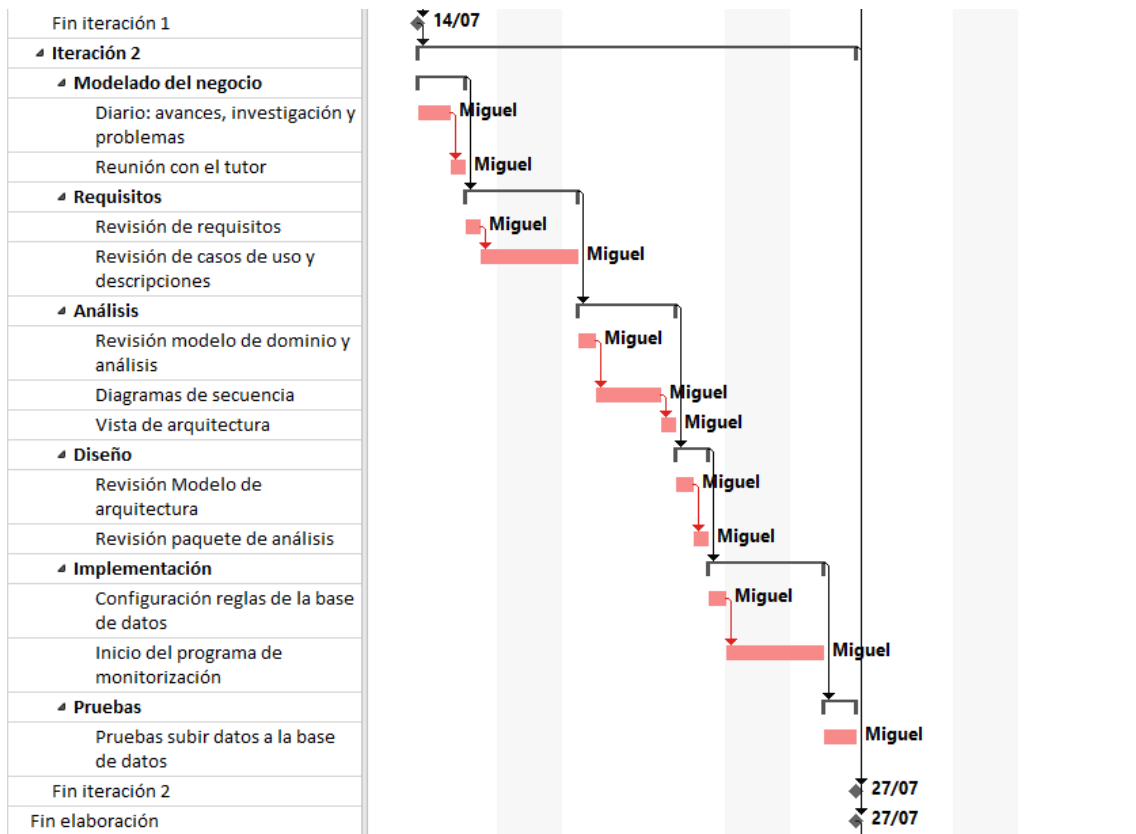


Ilustración 12: Diagrama de Gantt Elaboración It2

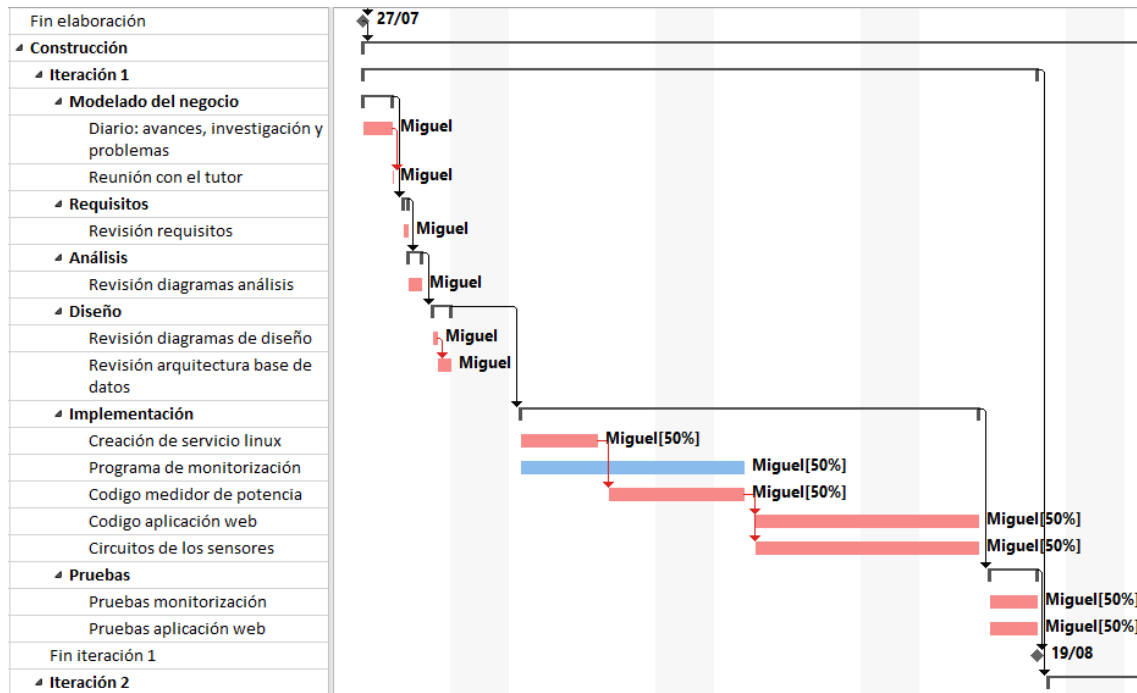


Ilustración 13: Diagrama de Gantt Construcción It1

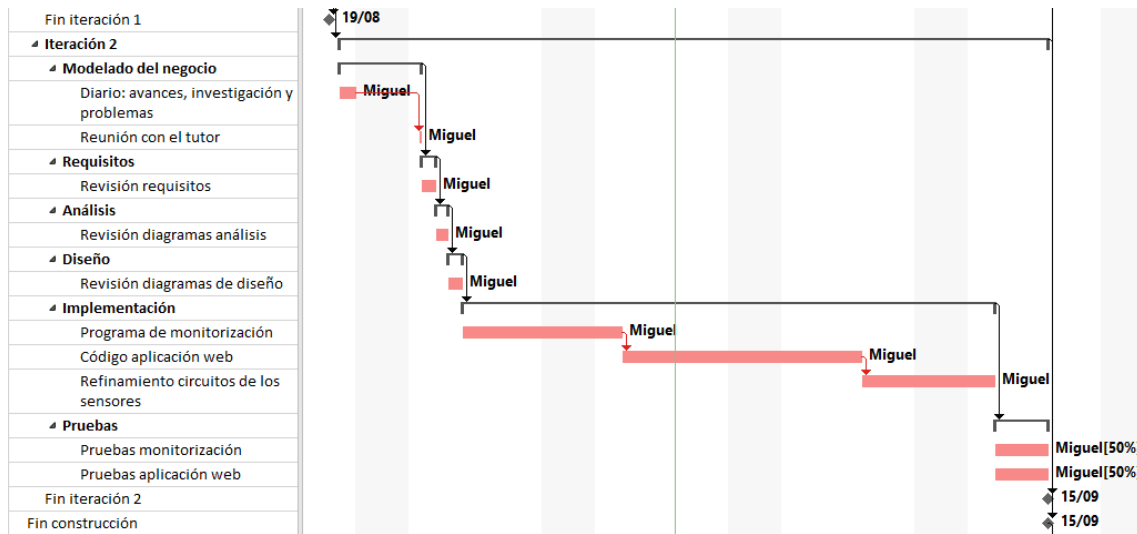


Ilustración 14: Diagrama de Gantt Construcción It2

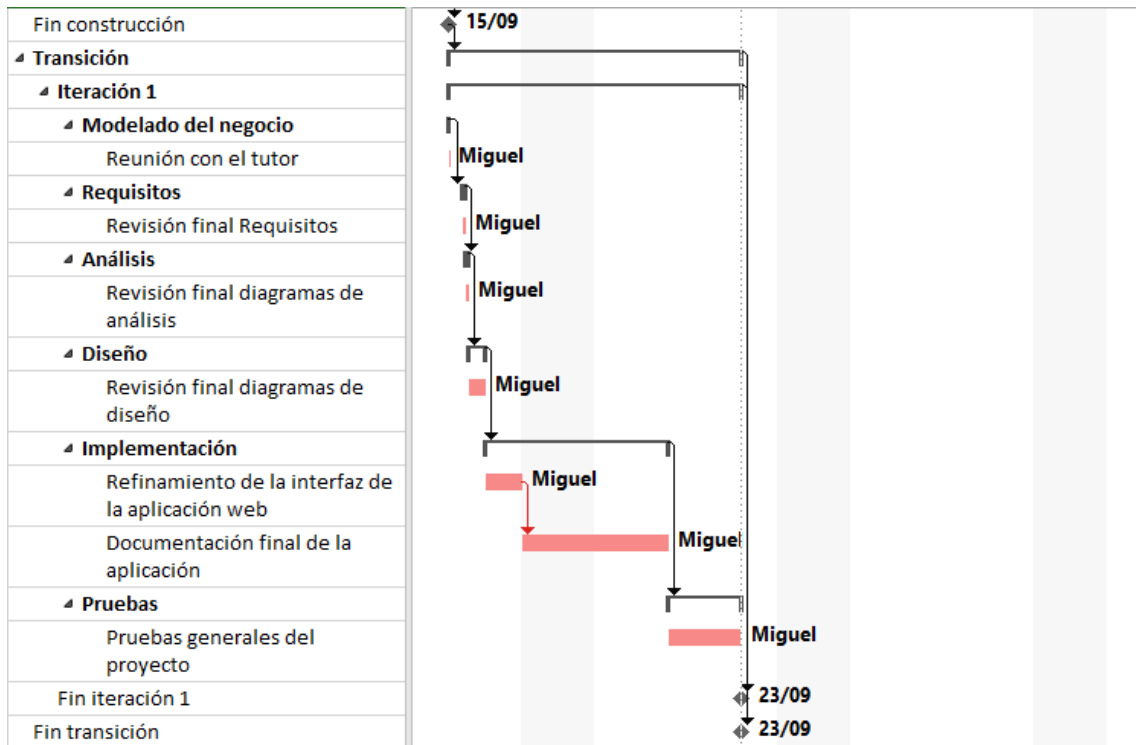


Ilustración 15: Diagrama de Gantt Transición

3.6. Resultado final

Para acabar, se muestra el resultado de la estimación temporal obtenida con el programa Microsoft Project.

▷ Inicio	15,5 days
Fin inicio	0 days
▷ Elaboración	25,25 days
Fin elaboración	0 days
▷ Construcción	36 days
Fin construcción	0 days
▷ Transición	6 days
Fin transición	0 days

Ilustración 16: Resultado estimación de esfuerzo

En la Ilustración 16 se puede observar el resultado de la estimación en la que se puede ver para cada fase el número de días que llevará su desarrollo.

En total se ha obtenido una suma de 82,75 días de trabajo, que multiplicados por las 10 horas de la jornada laboral dan un total de 827,5 horas. Es un resultado bastante similar al obtenido en la estimación de esfuerzo, en el que obtuvimos un valor de 768.96 horas, por lo que se puede deducir que se ha realizado una estimación correcta.

Aplicación para la gestión de una instalación fotovoltaica

Anexo II – Especificación de requisitos



VNiVERSIDAD
D SALAMANCA

Trabajo de Fin de Grado

Grado de Ingeniería Informática

Septiembre 2021

Tutor:

Iván Álvarez Navia

Alumno:

Miguel Sánchez González

Tabla de contenidos

1. Introducción	7
2. Herramientas.....	7
3. Participantes.....	7
4. Captura de requisitos	8
4.1. Actores del sistema.....	8
4.2. Objetivos	9
4.3. Requisitos de almacenamiento de información	10
4.4. Requisitos no funcionales	14
4.5. Requisitos funcionales	15
4.5.1. Diagrama de paquetes.....	15
4.5.2. Diagrama de casos de uso	16
4.5.3. Gestión de los roles del sistema	17
4.5.4. Gestión de la visualización de datos.....	24
4.5.5. Gestión de la monitorización del sistema	28
4.6. Matrices de rastreabilidad	30
4.6.1. Matriz de objetivos.....	30
4.6.2. Matriz de requisitos.....	31

Tabla de ilustraciones

Ilustración 1: Diagrama de actores.....	8
Ilustración 2: Diagrama de paquetes.....	15
Ilustración 3: Diagrama de casos de uso	16
Ilustración 4: Paquete Gestión de los roles del sistema.....	17
Ilustración 5: Paquete Gestión de la visualización de datos	24
Ilustración 6: Paquete Gestión de la monitorización del sistema	28

Índice de tablas

Tabla 1: Participante Miguel Sánchez González	7
Tabla 2: Participante Iván Álvarez Navia	7
Tabla 3: Actor usuario no registrado	8
Tabla 4: Usuario registrado	8
Tabla 5: Actor Sistema	8
Tabla 6: Objetivo Gestión de los roles del sistema	9
Tabla 7: Objetivo Gestión de la visualización de datos	9
Tabla 8: Gestión de la monitorización del sistema.....	10
Tabla 9: IRQ-0001 Datos de los usuarios	10
Tabla 10: IRQ-0002 Datos de tiempo real	11
Tabla 11: IRQ-0003 Histórico potencia generada	11
Tabla 12: IRQ-0004 Histórico de uso de batería.....	12
Tabla 13: IRQ-0005 Histórico de potencia consumida	12
Tabla 14: IRQ-0006 Estadísticas de potencia generada	13
Tabla 15: IRQ-0007 Estadísticas de potencia consumida	13
Tabla 16: NFR-0001 Escalabilidad y concurrencia	14
Tabla 17: NFR-0002 Sistema responsivo multiplataforma.....	14
Tabla 18: NFR-0003 Usabilidad y eficiencia.....	14
Tabla 19: NFR-0004 Privacidad y seguridad de los datos.....	15
Tabla 20: NFR-0005 Mantenibilidad	15
Tabla 21: UC-0001 Registrarse	18
Tabla 22: UC-0002 Validar correo electrónico	18
Tabla 23: UC-0003 Recuperar contraseña.....	19
Tabla 24: UC-0004 Iniciar sesión	19
Tabla 25: UC-0005 Cerrar sesión	20
Tabla 26: UC-0006 Modificar correo electrónico	21
Tabla 27: UC-0007 Modificar contraseña.....	21
Tabla 28: UC-0008 Eliminar cuenta	22
Tabla 29: UC-0009 Ver perfil	22
Tabla 30: UC-0010 Enviar correo electrónico de cambio de contraseña.....	23
Tabla 31: UC-0017 Obtener UID	23
Tabla 32: UC-0011 Ver página principal	24
Tabla 33: UC-0012 Mostrar datos de los relojes en tiempo real	25
Tabla 34: UC-0013 Mostrar datos estadísticos	25
Tabla 35: UC-0014 Mostrar gráfico de la potencia generada	26
Tabla 36: UC-0015 Mostrar gráfico de la batería	26
Tabla 37: UC-0016 Mostrar gráfico de la potencia consumida.....	27
Tabla 38: UC-0018 Inicialización de la monitorización.....	28
Tabla 39: UC-0019 Leer información de los sensores	29
Tabla 40: UC-0020 Enviar información a la base de datos	29
Tabla 41: Matriz de objetivos	30
Tabla 42: Matriz de requisitos	31

1. Introducción

En los apartados siguientes se verá la especificación de requisitos llevada a cabo para el proyecto.

El sistema sobre el que se ha trabajado está compuesto por una aplicación web, que muestra a los usuarios una interfaz para visualizar los datos de una instalación.

2. Herramientas

Durante todo este proceso, se usará el Proceso Unificado de Desarrollo Software, un modelo de desarrollo de software centrado en la arquitectura, iterativo e incremental, y se caracteriza por estar dirigido por casos de uso.

La documentación de requisitos se ha realizado haciendo uso del Lenguaje Unificado de Modelado (UML) y siguiendo el método de Duran y Bernárdez. Para crear los diagramas de casos de uso del proyecto se ha usado la herramienta Visual Paradigm.

3. Participantes

A continuación, se muestran los miembros del proyecto y su papel asociado.

Participante	Miguel Sánchez González
Organización	Universidad de Salamanca
Rol	Autor
Es desarrollador	Sí
Es cliente	No
Es usuario	No
Comentarios	Ninguno

Tabla 1: Participante Miguel Sánchez González

Participante	Iván Álvarez Navia
Organización	Universidad de Salamanca
Rol	Tutor
Es desarrollador	No
Es cliente	No
Es usuario	No
Comentarios	Ninguno

Tabla 2: Participante Iván Álvarez Navia

4. Captura de requisitos

En este apartado se muestran la especificación de los requisitos y objetivos del sistema.

4.1. Actores del sistema

Actores que interaccionan con la aplicación.

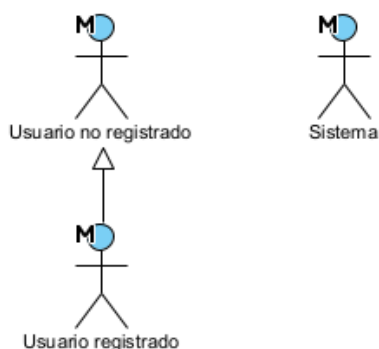


Ilustración 1: Diagrama de actores

ACT-0001	Usuario no registrado
Versión	1.0
Autores	Miguel Sánchez González
Descripción	Este actor representa a los usuarios que no han iniciado sesión en el sistema.
Comentarios	Ninguno

Tabla 3: Actor usuario no registrado

ACT-0002	Usuario registrado
Versión	1.0
Autores	Miguel Sánchez González
Descripción	Este actor representa a los usuarios registrados y a los que han iniciado sesión en el sistema.
Comentarios	Ninguno

Tabla 4: Usuario registrado

ACT-0003	Sistema
Versión	1.0
Autores	Miguel Sánchez González
Descripción	Este actor representa todas las acciones que llevará a cabo el sistema de forma automática.
Comentarios	Ninguno

Tabla 5: Actor Sistema

4.2. Objetivos

Las tablas que se ven a continuación son los objetivos principales del sistema y ofrecen una visión a muy alto nivel de la aplicación. Estos objetivos representan aquellos requisitos que el producto debe tener para satisfacer la funcionalidad final deseada.

OBJ-0001	Gestión de los roles del sistema
Versión	1.0
Autores	Miguel Sánchez González
Fuentes	
Descripción	El sistema deberá ser capaz de realizar el alta y baja de cuentas de usuarios, facilitar métodos de inicio de sesión y una forma de visualizar los datos de la cuenta.
Subobjetivos	Ninguno
Importancia	Alta
Urgencia	Alta
Estado	Activo
Estabilidad	Alta
Comentarios	Ninguno

Tabla 6: Objetivo Gestión de los roles del sistema

OBJ-0002	Gestión de la visualización de datos
Versión	1.0
Autores	Miguel Sánchez González
Fuentes	
Descripción	El sistema deberá de ser capaz de crear una representación gráfica de los datos de usuarios almacenados en la base de datos a través de la aplicación web.
Subobjetivos	Ninguno
Importancia	Alta
Urgencia	Alta
Estado	Activo
Estabilidad	Alta
Comentarios	Ninguno

Tabla 7: Objetivo Gestión de la visualización de datos

OBJ-0003	Gestión de la monitorización del sistema
Versión	1.0
Autores	Miguel Sánchez González
Fuentes	
Descripción	El sistema deberá de ser capaz de gestionar la monitorización de la instalación fotovoltaica y enviar los datos recogidos a la base de datos.
Subobjetivos	Ninguno
Importancia	Alta
Urgencia	Alta
Estado	Activo
Estabilidad	Alta
Comentarios	Ninguno

Tabla 8: Gestión de la monitorización del sistema

4.3. Requisitos de almacenamiento de información

El almacenamiento de la información y la forma en la que se almacena esta es una tarea vital dentro de cualquier proyecto, dado que un buen tratamiento de la información puede facilitar el desarrollo del proyecto, el rendimiento de las aplicaciones y su seguridad.

IRQ-0001	Datos de los usuarios
Versión	1.0
Autores	Miguel Sánchez González
Fuentes	
Dependencias	[OBJ-0001]
Descripción	El sistema deberá almacenar la información correspondiente a los datos de cada usuario.
Datos específicos	<ul style="list-style-type: none"> · UID · Correo electrónico · Contraseña
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 9: IRQ-0001 Datos de los usuarios

IRQ-0002	Datos de tiempo real
Versión	1.0
Autores	Miguel Sánchez González
Fuentes	
Dependencias	[OBJ-0002] [OBJ-0003]
Descripción	El sistema deberá almacenar la información correspondiente a los datos en tiempo real de la instalación fotovoltaica.
Datos específicos	<ul style="list-style-type: none"> · Datos en tiempo real del voltaje de la batería · Datos en tiempo real de la potencia generada · Datos en tiempo real de la potencia consumida
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 10: IRQ-0002 Datos de tiempo real

IRQ-0003	Histórico de potencia generada
Versión	1.0
Autores	Miguel Sánchez González
Fuentes	
Dependencias	[OBJ-0002] [OBJ-0003]
Descripción	El sistema deberá almacenar la información correspondiente al histórico de potencia generada de la instalación fotovoltaica.
Datos específicos	· Datos de la potencia generada de cada día de los minutos registrados.
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 11: IRQ-0003 Histórico potencia generada

IRQ-0004	Histórico de uso de batería
Versión	1.0
Autores	Miguel Sánchez González
Fuentes	
Dependencias	[OBJ-0002] [OBJ-0003]
Descripción	El sistema deberá almacenar la información correspondiente al histórico de uso de la batería de la instalación fotovoltaica.
Datos específicos	· Datos del voltaje de la batería cada día de los minutos registrados.
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 12: IRQ-0004 Histórico de uso de batería

IRQ-0005	Histórico de potencia consumida
Versión	1.0
Autores	Miguel Sánchez González
Fuentes	
Dependencias	[OBJ-0002] [OBJ-0003]
Descripción	El sistema deberá almacenar la información correspondiente al histórico de potencia consumida de la instalación fotovoltaica.
Datos específicos	· Datos de la potencia consumida de cada día de los minutos registrados.
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 13: IRQ-0005 Histórico de potencia consumida

IRQ-0006	Estadísticas de potencia generada
Versión	1.0
Autores	Miguel Sánchez González
Fuentes	
Dependencias	[OBJ-0002] [OBJ-0003]
Descripción	El sistema deberá almacenar la información correspondiente a las estadísticas de generación de potencia de la instalación fotovoltaica.
Datos específicos	<ul style="list-style-type: none"> · Datos estadísticos de la potencia generada de cada día. · Datos del estado actual de las estadísticas de potencia generada del día.
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 14: IRQ-0006 Estadísticas de potencia generada

IRQ-0007	Estadísticas de potencia consumida
Versión	1.0
Autores	Miguel Sánchez González
Fuentes	
Dependencias	[OBJ-0002] [OBJ-0003]
Descripción	El sistema deberá almacenar la información correspondiente a las estadísticas de uso de potencia de la instalación fotovoltaica.
Datos específicos	<ul style="list-style-type: none"> · Datos estadísticos de la potencia consumida de cada día. · Datos del estado actual de las estadísticas de potencia consumida del día.
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 15: IRQ-0007 Estadísticas de potencia consumida

4.4. Requisitos no funcionales

En este apartado se van a estudiar algunos requisitos no fundamentales para la funcionalidad final del sistema pero que son igual de importantes en su realización.

NFR-0001	Escalabilidad y concurrencia
Versión	1.0
Autores	Miguel Sánchez González
Dependencias	Todo el sistema
Descripción	El sistema debe ser capaz de adaptarse a cualquier instalación y además poder dar servicio a una gran cantidad de usuarios.
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 16: NFR-0001 Escalabilidad y concurrencia

NFR-0002	Sistema responsivo multiplataforma
Versión	1.0
Autores	Miguel Sánchez González
Dependencias	Todo el sistema
Descripción	El sistema debe ser capaz de funcionar en varios dispositivos y adaptarse a sus dimensiones.
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 17: NFR-0002 Sistema responsivo multiplataforma

NFR-0003	Usabilidad y eficiencia
Versión	1.0
Autores	Miguel Sánchez González
Dependencias	Todo el sistema
Descripción	El sistema debe ser fácil de usar y eficiente en las transacciones y funciones de la aplicación.
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 18: NFR-0003 Usabilidad y eficiencia

NFR-0004	Privacidad y seguridad de los datos
Versión	1.0
Autores	Miguel Sánchez González
Dependencias	Todo el sistema
Descripción	El sistema debe ser capaz de mantener a salvo los datos de cada usuario y facilitar solo aquellos datos importantes a los usuarios autorizados.
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 19: NFR-0004 Privacidad y seguridad de los datos

NFR-0005	Mantenibilidad
Versión	1.0
Autores	Miguel Sánchez González
Dependencias	Todo el sistema
Descripción	El sistema debe diseñarse pensando en minimizar los esfuerzos futuros para realizar cambios y mejoras.
Importancia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 20: NFR-0005 Mantenibilidad

4.5. Requisitos funcionales

En este apartado se detallarán aquellos requisitos referentes a la funcionalidad y objetivos del sistema, vitales para cumplir con todas las necesidades

A continuación, se detallarán los diagramas de paquetes y sus casos de uso.

4.5.1. Diagrama de paquetes

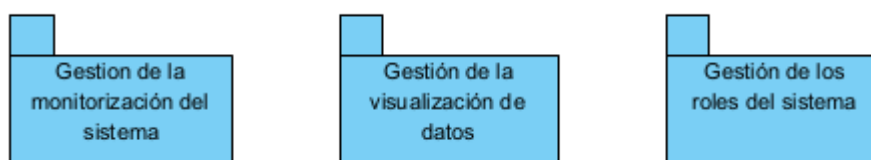


Ilustración 2: Diagrama de paquetes

4.5.3. Gestión de los roles del sistema

Este módulo recoge todas aquellas funciones requeridas para la gestión de los usuarios del sistema.

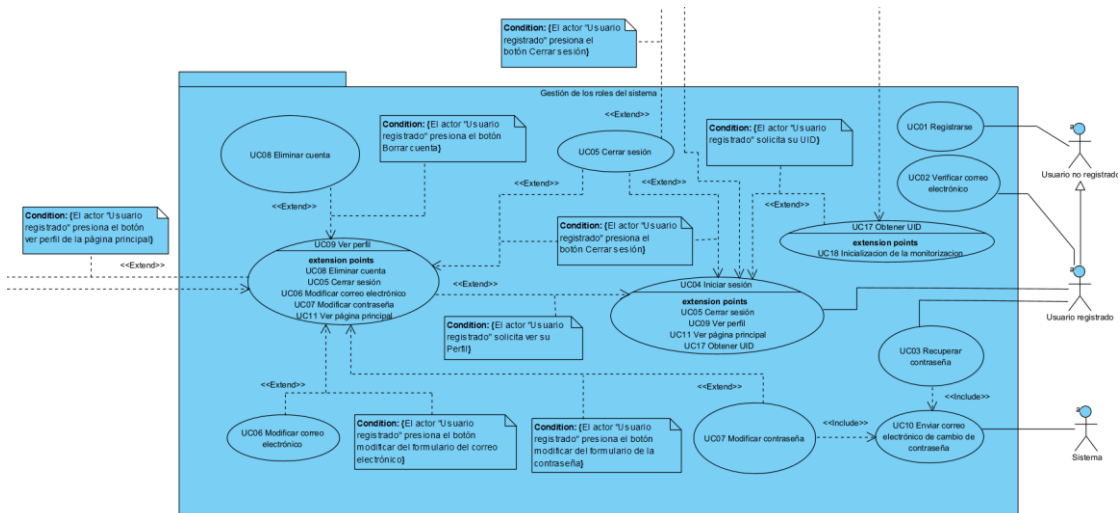


Ilustración 4: Paquete Gestión de los roles del sistema

UC-0001		Registrarse
Versión	1.0	
Autores	Miguel Sánchez González	
Fuentes		
Dependencias	Ninguna	
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario no registrado” [ACT-0001], solicite registrarse.	
Precondición	El usuario no está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	EL actor Usuario no registrado [ACR-0001] solicita registrarse.
	2	El sistema muestra la página de registro
	3	El actor Usuario no registrado [ACT-0001] rellena los campos de registro y pulsa en confirmar.
	4	El sistema comprueba que los datos introducidos sean únicos.
	5	El sistema crea la cuenta del usuario.
	6	El sistema redirige al usuario a la página principal
Postcondición	El usuario queda registrado no verificado.	
Excepciones	Paso	Acción
	4	Si los datos introducidos no son únicos se muestra un mensaje de error y se solicitan los datos de nuevo.

Importancia	Alta
Urgencia	Alta
Estado	Activo
Estabilidad	Alta
Comentarios	Ninguno

Tabla 21: UC-0001 Registrarse

UC-0002		Validar correo electrónico	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	Ninguna		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario registrado” [ACT-0002], solicite validar su correo electrónico.		
Precondición	El usuario está registrado no verificado.		
Secuencia normal	Paso	Acción	
	1	El usuario solicita la verificación de su correo electrónico.	
	2	El sistema emite el correo de verificación para el usuario.	
Postcondición	Ninguna		
Excepciones	Paso	Acción	
	1	Ninguna	
Importancia	Alta		
Urgencia	Alta		
Estado	Activo		
Estabilidad	Alta		
Comentarios	Ninguno		

Tabla 22: UC-0002 Validar correo electrónico

UC-0003		Recuperar contraseña	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	Ninguna		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario registrado” [ACT-0002], solicite recuperar su contraseña.		
Precondición	No hay ninguna sesión iniciada.		
Secuencia normal	Paso	Acción	
	1	El usuario solicita la recuperación de su contraseña	
	2	El sistema muestra la página de recuperación de contraseña.	

	3	El usuario rellena el campo de recuperación de contraseña y pulsa en recuperar.
	4	El sistema comprueba que el email introducido exista.
	5	Se ejecuta el caso de uso [UC-0010]
Postcondición	Ninguna	
Excepciones	Paso	Acción
	4	El correo introducido no existe
Importancia	Alta	
Urgencia	Alta	
Estado	Activo	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 23: UC-0003 Recuperar contraseña

UC-0004	Iniciar sesión	
Versión	1.0	
Autores	Miguel Sánchez González	
Fuentes		
Dependencias	Ninguna	
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario registrado” [ACT-0002] o el “Usuario no registrado” [ACT-0001], solicite iniciar sesión.	
Precondición	No hay ninguna sesión iniciada.	
Secuencia normal	Paso	Acción
	1	El usuario solicita iniciar sesión.
	2	El sistema redirige al usuario a la página de inicio de sesión.
	3	El usuario rellena los campos de registro y pulsa en confirmar.
	4	El sistema comprueba que los datos introducidos sean correctos.
	5	El sistema inicia sesión con la cuenta del usuario.
	6	El sistema redirige al usuario a la página de inicio de sesión.
Postcondición	Hay una sesión iniciada.	
Excepciones	Paso	Acción
	1	Si los datos no son correctos se muestra un mensaje de error y pide volver a introducir los datos.
Importancia	Alta	
Urgencia	Alta	
Estado	Activo	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 24: UC-0004 Iniciar sesión

UC-0005		Cerrar sesión	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	[UC-0004] Iniciar sesión [UC-0009] Ver perfil		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario registrado” [ACT-0002], solicite cerrar su sesión.		
Precondición	Hay una sesión iniciada.		
Secuencia normal	Paso	Acción	
	1	El usuario solicita cerrar sesión.	
	2	El sistema cierra la sesión del usuario.	
Postcondición	No hay ninguna una sesión iniciada.		
Excepciones	Paso	Acción	
	1	Ninguna	
Importancia	Alta		
Urgencia	Alta		
Estado	Activo		
Estabilidad	Alta		
Comentarios	Ninguno		

Tabla 25: UC-0005 Cerrar sesión

UC-0006		Modificar correo electrónico	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	[UC-0009] Ver perfil		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario registrado” [ACT-0002], solicite modificar su correo electrónico.		
Precondición	Hay una sesión iniciada.		
Secuencia normal	Paso	Acción	
	1	El usuario pulsa el botón modificar del formulario de su correo electrónico.	
	2	El usuario rellena el campo de correo electrónico y pulsa guardar.	
	3	El sistema comprueba que el correo introducido sea único.	
	4	El sistema cambia el correo electrónico.	
Postcondición	Ninguna		
Excepciones	Paso	Acción	
	3	Si los datos no son correctos el sistema muestra un mensaje de error y pide volver a introducir los datos.	

Importancia	Alta
Urgencia	Alta
Estado	Activo
Estabilidad	Alta
Comentarios	Ninguno

Tabla 26: UC-0006 Modificar correo electrónico

UC-0007		Modificar contraseña	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	[UC-0009] Ver perfil		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario registrado” [ACT-0002], solicite modificar su contraseña.		
Precondición	Hay una sesión iniciada.		
Secuencia normal	Paso	Acción	
	1	El usuario pulsa el botón modificar del formulario de su contraseña.	
	2	Se ejecuta el caso de uso [UC-0010]	
Postcondición	Ninguna		
Excepciones	Paso	Acción	
	1	Ninguna	
Importancia	Alta		
Urgencia	Alta		
Estado	Activo		
Estabilidad	Alta		
Comentarios	Ninguno		

Tabla 27: UC-0007 Modificar contraseña

UC-0008		Eliminar cuenta	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	[UC-0009] Ver perfil		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario registrado” [ACT-0002], solicite eliminar su cuenta.		
Precondición	Hay una sesión iniciada.		
Secuencia normal	Paso	Acción	
	1	El usuario solicita la eliminación de su cuenta.	
	2	El sistema muestra un dialogo de confirmación.	
	3	El usuario pulsa en confirmar.	
	4	El sistema elimina la cuenta del usuario.	

Postcondición	Ninguna	
Excepciones	Paso	Acción
	1	Ninguna
Importancia	Alta	
Urgencia	Alta	
Estado	Activo	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 28: UC-0008 Eliminar cuenta

UC-0009	Ver perfil	
Versión	1.0	
Autores	Miguel Sánchez González	
Fuentes		
Dependencias	[UC-0004] Iniciar sesión	
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario registrado” [ACT-0002], solicite ver su perfil de usuario.	
Precondición	Hay una sesión iniciada.	
Secuencia normal	Paso	Acción
	1	El usuario solicita ver su información de Cuenta.
	2	El sistema redirige al usuario a la página del perfil.
Postcondición	Ninguna	
Excepciones	Paso	Acción
	1	Ninguna
Importancia	Alta	
Urgencia	Alta	
Estado	Activo	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 29: UC-0009 Ver perfil

UC-0010	Enviar correo electrónico de cambio de contraseña	
Versión	1.0	
Autores	Miguel Sánchez González	
Fuentes		
Dependencias	[UC-0007] Modificar contraseña [UC-0003] Recuperar contraseña	
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el Sistema [ACT-0003] detecte que un “Usuario registrado” [ACT-0002] ha solicitado un cambio de contraseña.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El sistema envía el email de restablecimiento de contraseña y muestra un dialogo de información.
Postcondición	Ninguna	

Excepciones	Paso	Acción
	1	Ninguna
Importancia	Alta	
Urgencia	Alta	
Estado	Activo	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 30: UC-0010 Enviar correo electrónico de cambio de contraseña

UC-0017		Obtener UID	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	[UC-0004] Iniciar sesión		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el Sistema [ACT-0003] detecte que el “Usuario registrado” [ACT-0002] está en la página de su perfil de usuario.		
Precondición	Ninguna		
Secuencia normal	Paso	Acción	
	1	El sistema muestra el UID del usuario.	
Postcondición	Ninguna		
Excepciones	Paso	Acción	
	1	Ninguna	
Importancia	Alta		
Urgencia	Alta		
Estado	Activo		
Estabilidad	Alta		
Comentarios	Ninguno		

Tabla 31: UC-0017 Obtener UID

4.5.4. Gestión de la visualización de datos

En este módulo aparecen aquellas funciones relacionadas con la representación de los datos en la aplicación web.

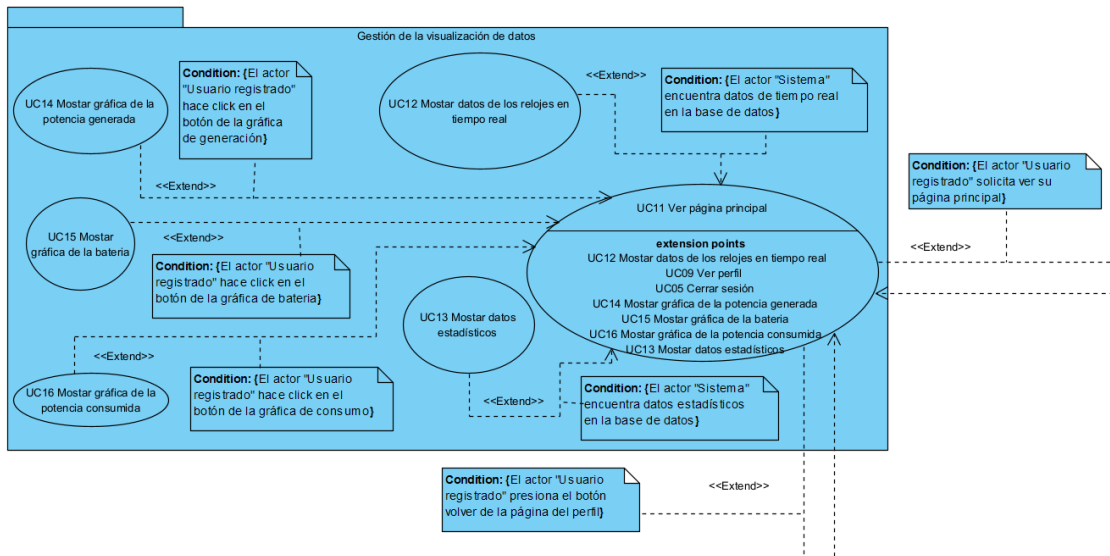


Ilustración 5: Paquete Gestión de la visualización de datos

UC-0011	Ver página principal	
Versión	1.0	
Autores	Miguel Sánchez González	
Fuentes		
Dependencias	[UC-0004] Iniciar sesión [UC-0009] Ver perfil	
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario registrado” [ACT-0002], solicite ver la página principal.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El usuario solicita ver la página principal.
	1	El sistema redirige al usuario a la página principal.
Postcondición	Ninguna	
Excepciones	Paso	Acción
	1	Ninguna
Importancia	Alta	
Urgencia	Alta	
Estado	Activo	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 32: UC-0011 Ver página principal

UC-0012		Mostrar datos de los relojes en tiempo real	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	[UC-0011] Ver página principal		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el Sistema [ACT-0003] detecte que el “Usuario registrado” [ACT-0002] está en la página principal.		
Precondición	Ninguna		
Secuencia normal	Paso	Acción	
	1	El sistema carga y muestra los datos de tiempo real	
Postcondición	Ninguna		
Excepciones	Paso	Acción	
	1	Ninguna	
Importancia	Alta		
Urgencia	Alta		
Estado	Activo		
Estabilidad	Alta		
Comentarios	Ninguno		

Tabla 33: UC-0012 Mostrar datos de los relojes en tiempo real

UC-0013		Mostrar datos estadísticos	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	[UC-0011] Ver página principal		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el Sistema [ACT-0003] detecte que el “Usuario registrado” [ACT-0002] está en la página principal.		
Precondición	Ninguna		
Secuencia normal	Paso	Acción	
	1	El sistema carga y muestra los datos estadísticos.	
Postcondición	Ninguna		
Excepciones	Paso	Acción	
	1	Ninguna	
Importancia	Alta		
Urgencia	Alta		
Estado	Activo		
Estabilidad	Alta		
Comentarios	Ninguno		

Tabla 34: UC-0013 Mostrar datos estadísticos

UC-0014		Mostrar gráfico de la potencia generada	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	[UC-0011] Ver página principal		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario registrado” [ACT-0002], solicite ver los históricos de potencia generada.		
Precondición	Ninguna		
Secuencia normal	Paso	Acción	
	1	El usuario solicita ver el gráfico de potencia generada.	
	2	El sistema pide los datos de la potencia generada.	
	3	El sistema muestra los datos de la potencia generada.	
Postcondición	Ninguna		
Excepciones	Paso	Acción	
	1	Ninguna	
Importancia	Alta		
Urgencia	Alta		
Estado	Activo		
Estabilidad	Alta		
Comentarios	Ninguno		

Tabla 35: UC-0014 Mostrar gráfico de la potencia generada

UC-0015		Mostrar gráfico de la batería	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	[UC-0011] Ver página principal		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario registrado” [ACT-0002], solicite ver los históricos del estado de la batería.		
Precondición	Ninguna		
Secuencia normal	Paso	Acción	
	1	El usuario solicita ver el gráfico de la batería.	
	2	El sistema pide los datos de la batería.	
	3	El sistema muestra los datos de la batería.	
Postcondición	Ninguna		
Excepciones	Paso	Acción	
	1	Ninguna	
Importancia	Alta		
Urgencia	Alta		
Estado	Activo		
Estabilidad	Alta		
Comentarios	Ninguno		

Tabla 36: UC-0015 Mostrar gráfico de la batería

UC-0016		Mostrar gráfico de la potencia consumida	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	[UC-0011] Ver página principal		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario registrado” [ACT-0002], solicite ver los históricos de potencia consumida.		
Precondición	Ninguna		
Secuencia normal	Paso	Acción	
	1	El usuario solicita ver el gráfico de potencia consumida.	
	2	El sistema pide los datos de la potencia generada.	
	3	El sistema muestra los datos de la potencia consumida.	
Postcondición	Ninguna		
Excepciones	Paso	Acción	
	1	Ninguna	
Importancia	Alta		
Urgencia	Alta		
Estado	Activo		
Estabilidad	Alta		
Comentarios	Ninguno		

Tabla 37: UC-0016 Mostrar gráfico de la potencia consumida

4.5.5. Gestión de la monitorización del sistema

Este módulo representa las funciones de monitorización de la instalación fotovoltaica.

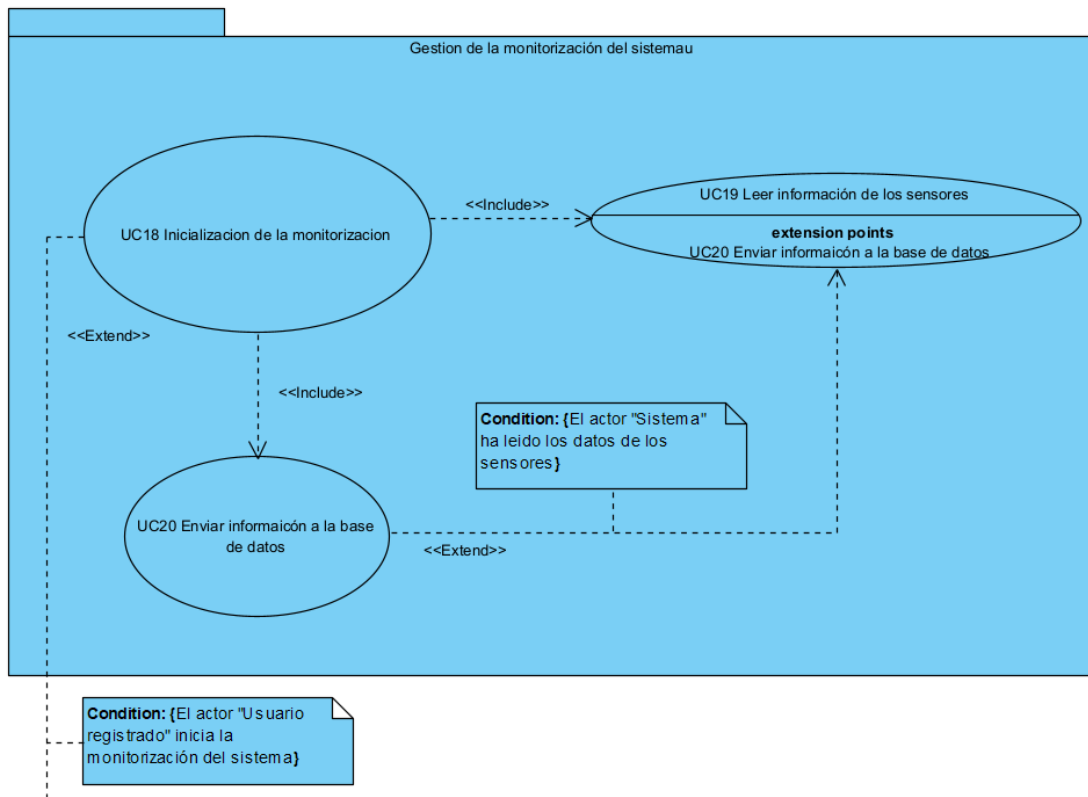


Ilustración 6: Paquete Gestión de la monitorización del sistema

UC-0018		Iniciación de la monitorización	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	[UC-0017] Obtener UID		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el “Usuario registrado” [ACT-0002], inicie la monitorización de la instalación.		
Precondición	Ninguna		
Secuencia normal	Paso	Acción	
	1	El sistema configura los sensores y las bases de datos para subir los datos	
Postcondición	Ninguna		
Excepciones	Paso	Acción	
	1	Ninguna	
Importancia	Alta		
Urgencia	Alta		
Estado	Activo		
Estabilidad	Alta		
Comentarios	Ninguno		

Tabla 38: UC-0018 Iniciación de la monitorización

UC-0019		Leer información de los sensores	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	[UC-0018] Inicialización de la monitorización		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el Sistema [ACT-0003], quiera leer los datos a la salida de los sensores.		
Precondición	Ninguna		
Secuencia normal	Paso	Acción	
	1	El sistema lee los datos de los sensores de las placas solares.	
	2	El sistema lee los datos del sensor de la batería.	
	3	El sistema lee los datos del sensor de potencia a la salida del inversor.	
Postcondición	Ninguna		
Excepciones	Paso	Acción	
	1	Ninguna.	
Importancia	Alta		
Urgencia	Alta		
Estado	Activo		
Estabilidad	Alta		
Comentarios	Ninguno		

Tabla 39: UC-0019 Leer información de los sensores

UC-0020		Enviar información a la base de datos	
Versión	1.0		
Autores	Miguel Sánchez González		
Fuentes			
Dependencias	[UC-0018] Inicialización de la monitorización [UC-0019] Leer información de los sensores		
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso cuando el Sistema [ACT-0003], quiera enviar los datos recogidos a la base de datos.		
Precondición	Hay información leída de los sensores.		
Secuencia normal	Paso	Acción	
	1	El sistema envía la información de los sensores a la base de datos.	
Postcondición	Ninguna		
Excepciones	Paso	Acción	
	1	Ninguna	
Importancia	Alta		
Urgencia	Alta		
Estado	Activo		
Estabilidad	Alta		
Comentarios	Ninguno		

Tabla 40: UC-0020 Enviar información a la base de datos

4.6. Matrices de rastreabilidad

4.6.1. Matriz de objetivos

	OBJ-0001	OBJ-0002	OBJ-0003
UC-0001	x		
UC-0002	x		
UC-0003	x		
UC-0004	x		
UC-0005	x		
UC-0006	x		
UC-0007	x		
UC-0008	x		
UC-0009	x		
UC-0010	x		
UC-0011		x	
UC-0012		x	
UC-0013		x	
UC-0014		x	
UC-0015		x	
UC-0016		x	
UC-0017	x		
UC-0018			x
UC-0019			x
UC-0020			x

Tabla 41: Matriz de objetivos

4.6.2. Matriz de requisitos

	IRQ-01	IRQ-02	IRQ-03	IRQ-04	IRQ-05	IRQ-06	IRQ-07	NRF-01	NRF-02	NRF-03	NRF-04	NRF-05
UC-0001	X							X	X	X	X	X
UC-0002	X							X	X	X	X	X
UC-0003	X							X	X	X	X	X
UC-0004	X							X	X	X	X	X
UC-0005	X							X	X	X	X	X
UC-0006	X							X	X	X	X	X
UC-0007	X							X	X	X	X	X
UC-0008	X							X	X	X	X	X
UC-0009	X							X	X	X	X	X
UC-0010	X							X	X	X	X	X
UC-0011		X	X	X	X	X	X	X	X	X	X	X
UC-0012		X						X	X	X	X	X
UC-0013						X	X	X	X	X	X	X
UC-0014			X					X	X	X	X	X
UC-0015				X				X	X	X	X	X
UC-0016					X			X	X	X	X	X
UC-0017	X							X	X	X	X	X
UC-0018		X	X	X	X	X	X	X	X	X	X	X
UC-0019		X	X	X	X	X	X	X	X	X	X	X
UC-0020		X	X	X	X	X	X	X	X	X	X	X

Tabla 42: Matriz de requisitos

Aplicación para la gestión de una instalación fotovoltaica

Anexo III – Análisis de requisitos



VNiVERSiDAD
D SALAMANCA

Trabajo de Fin de Grado

Grado de Ingeniería Informática

Septiembre 2021

Tutor:

Iván Álvarez Navia

Alumno:

Miguel Sánchez González

Tabla de contenidos

1. Introducción	6
2. Modelo de dominio	6
3. Paquetes de análisis	7
4. Vista de la arquitectura	9
5. Diagramas de secuencia: realización de casos de uso	10
5.1. Gestión de la visualización de datos	10
5.2. Gestión de los roles del sistema	13
5.3. Gestión de la monitorización de la instalación	19

Tabla de ilustraciones

Ilustración 1: Modelo del dominio	6
Ilustración 2: Diagrama de paquetes de análisis	7
Ilustración 3: Paquete de análisis Gestión de la visualización de datos	7
Ilustración 4: Paquete de análisis Gestión de los roles del sistema	8
Ilustración 5: Paquete de análisis Gestión de la monitorización del sistema	8
Ilustración 6: Vista de arquitectura del modelo de análisis	9
Ilustración 7: Diagrama de secuencia Ver página principal	10
Ilustración 8: Diagrama de secuencia Mostrar datos de los relojes en tiempo real.....	10
Ilustración 9: Diagrama de secuencia Mostrar datos estadísticos.....	11
Ilustración 10: Diagrama de secuencia Mostrar gráfica de la potencia generada.....	11
Ilustración 11: Diagrama de secuencia Mostrar gráfico de la batería.....	12
Ilustración 12: Diagrama de secuencia Mostrar gráfico de la potencia consumida	12
Ilustración 13: Diagrama de secuencia Registrarse.....	13
Ilustración 14: Diagrama de secuencia Verificar correo electrónico	14
Ilustración 15: Diagrama de secuencia Recuperar contraseña	14
Ilustración 16: Diagrama de Secuencia Iniciar sesión	15
Ilustración 17: Diagrama de secuencia Cerrar sesión	15
Ilustración 18: Diagrama de secuencia Modificar correo electrónico	16
Ilustración 19: Diagrama de secuencia Modificar contraseña	16
Ilustración 20: Diagrama de secuencia Eliminar Cuenta	17
Ilustración 21: Diagrama de secuencia Ver Perfil.....	17
Ilustración 22: Diagrama de secuencia Enviar correo electrónico de cambio de contraseña.....	18
Ilustración 23: Diagrama de secuencia Obtener UID	18
Ilustración 24: Diagrama de secuencia Inicialización de la monitorización	19
Ilustración 25: Diagrama de secuencia Leer información de los sensores	19
Ilustración 26: Diagrama de secuencia Enviar información a la base de datos	20

1. Introducción

En este anexo se va a realizar el análisis de los requisitos del sistema, a partir de los datos obtenidos en el Anexo II – Especificación de requisitos. El resultado nos permitirá tener una visión de mas bajo nivel, más detallada, del sistema.

En este anexo se podrán ver los siguientes apartados:

- Modelo de dominio
- Paquetes de análisis y servicio
- Vista de arquitectura
- Realización de casos de uso

Todos los diagramas que se ven en este anexo se han realizado usando la herramienta Visual Paradigm.

2. Modelo de dominio

El modelo de dominio permite crear una diferenciación de cada una de las clases que almacena el sistema. Teniendo por cada clase del modelo unos atributos y unas relaciones con otras entidades.

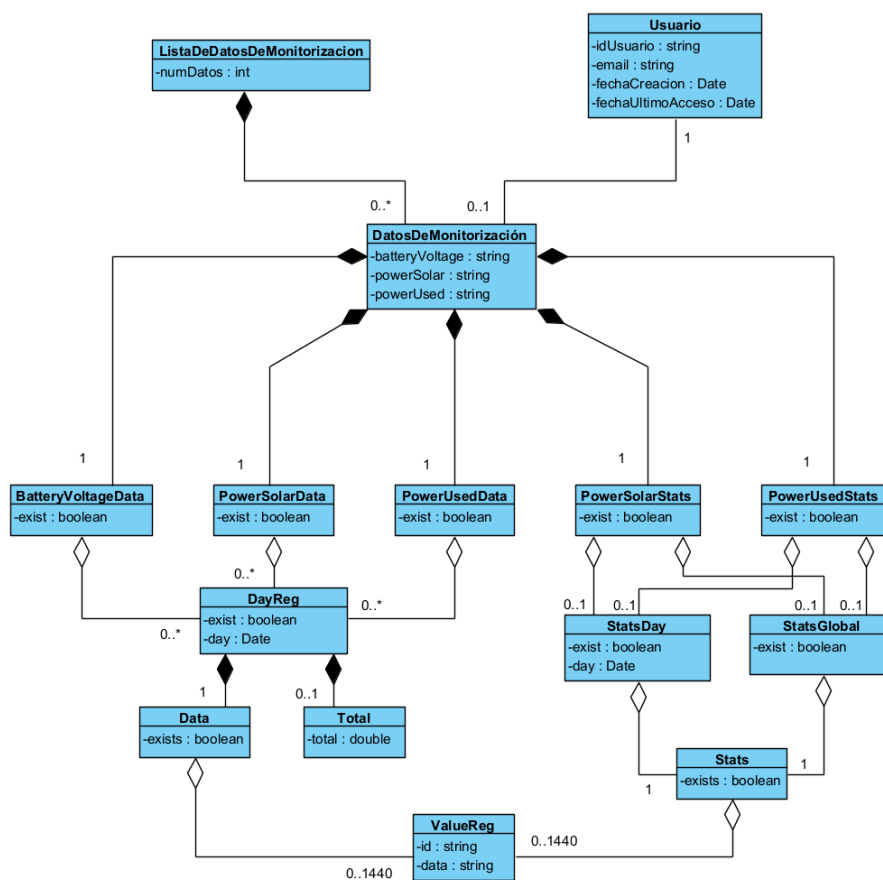


Ilustración 1: Modelo del dominio

3. Paquetes de análisis

En este apartado se realizará la descomposición del sistema en paquetes de análisis y servicio. A continuación, se puede ver la distribución de los paquetes de análisis:

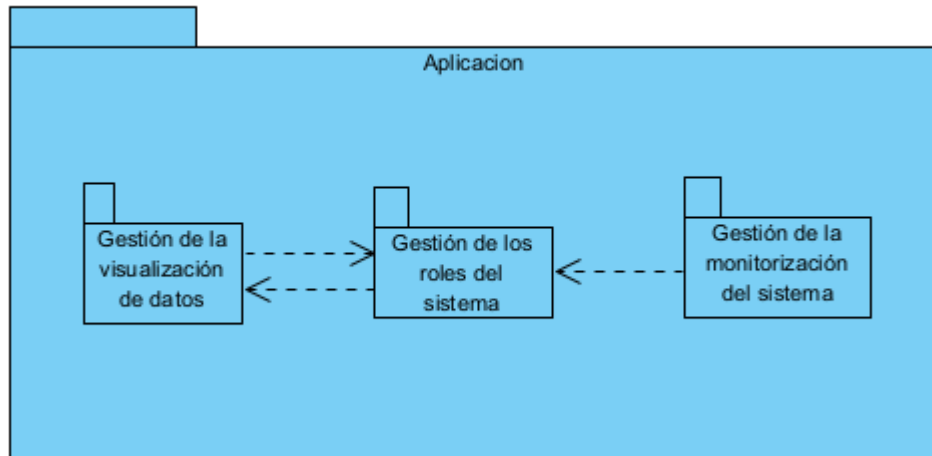


Ilustración 2: Diagrama de paquetes de análisis

La descripción de los paquetes es la siguiente:

- **Gestión de la visualización de datos:** Realiza las tareas de representación de los datos almacenados en la base de datos.
- **Gestión de los roles del sistema:** Realiza la gestión de los usuarios del sistema, además de las acciones de modificación de los datos de las cuentas.
- **Gestión de la monitorización del sistema:** Realiza la gestión de la lectura de los sensores del sistema y el envío de los datos a la base de datos.

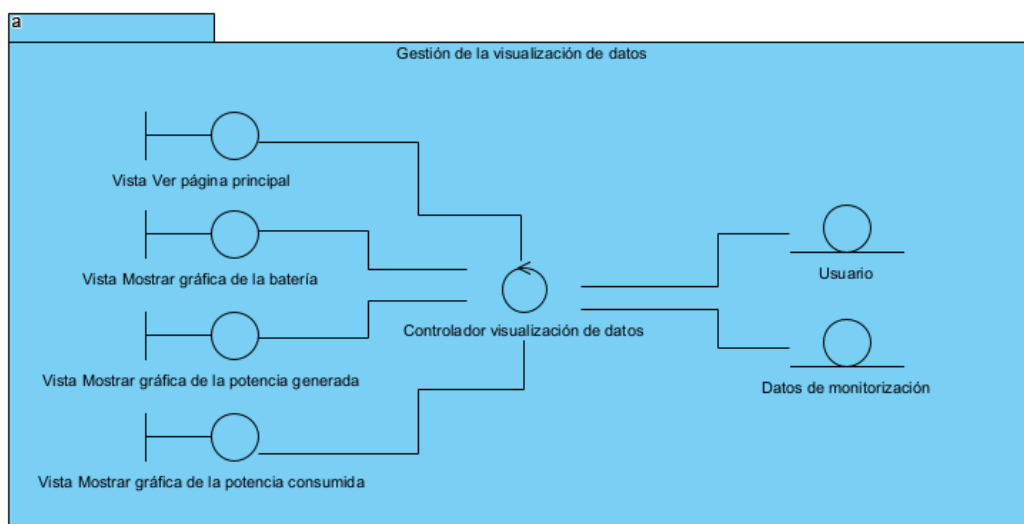


Ilustración 3: Paquete de análisis Gestión de la visualización de datos

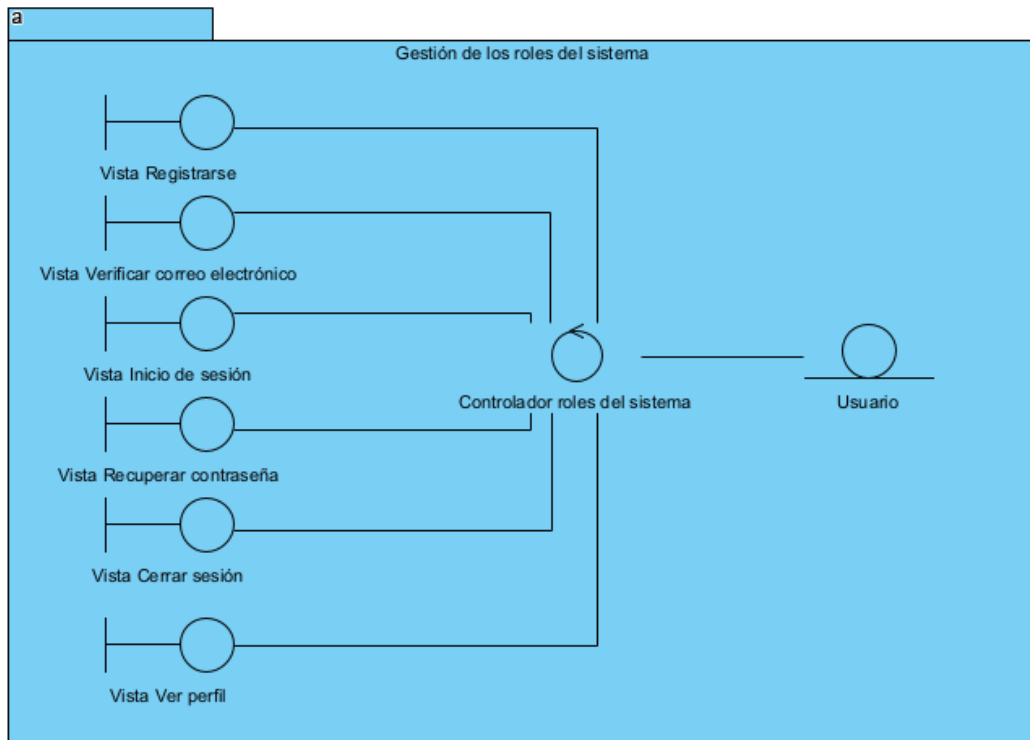


Ilustración 4: Paquete de análisis Gestión de los roles del sistema

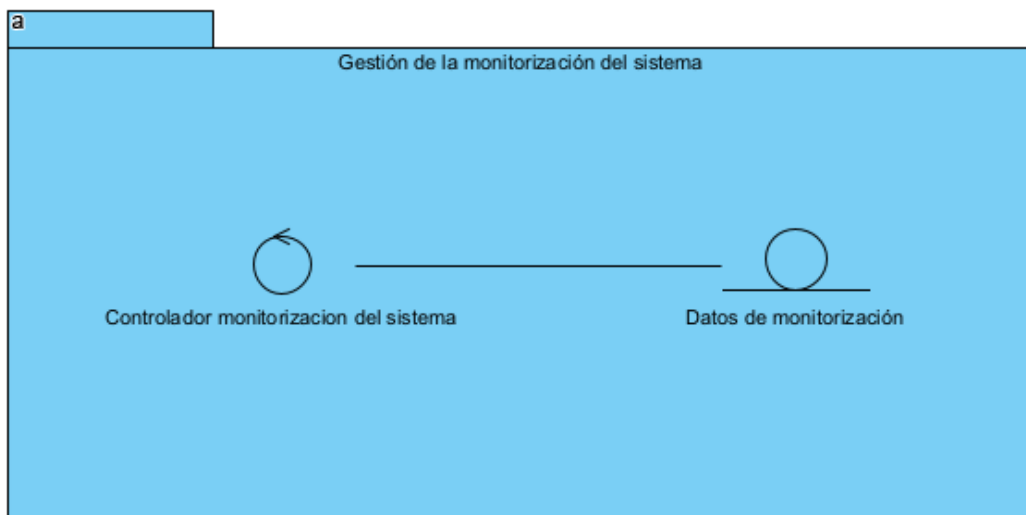


Ilustración 5: Paquete de análisis Gestión de la monitorización del sistema

4. Vista de la arquitectura

En este apartado, se observa una primera versión de la vista de arquitectura del sistema, cuyos paquetes se verán con mas detalla en el Anexo IV – Diseño del sistema.

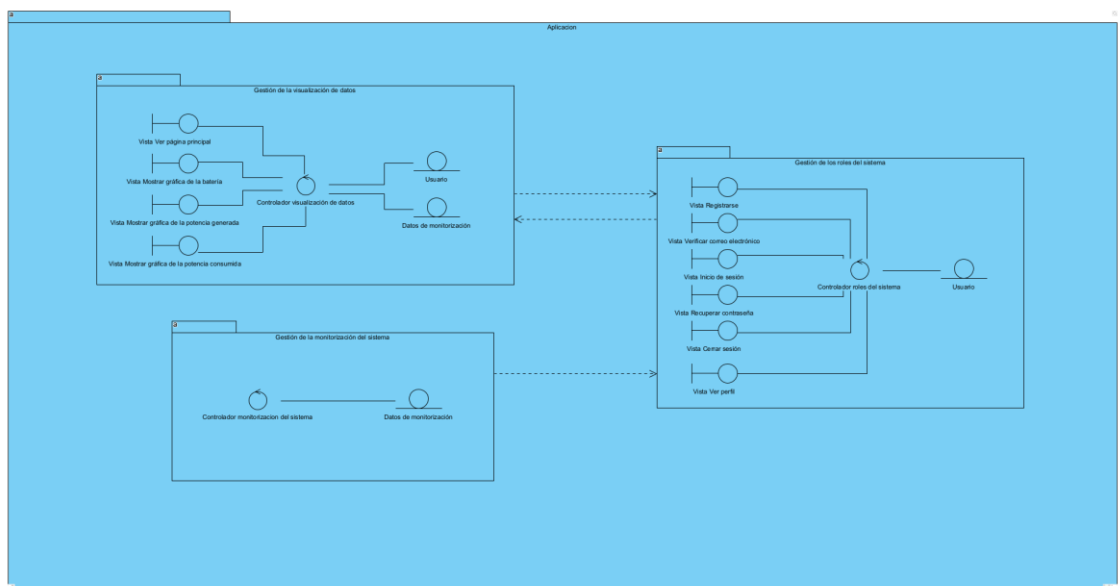


Ilustración 6: Vista de arquitectura del modelo de análisis

5. Diagramas de secuencia: realización de casos de uso

5.1. Gestión de la visualización de datos

sd UC-0011 Ver página principal

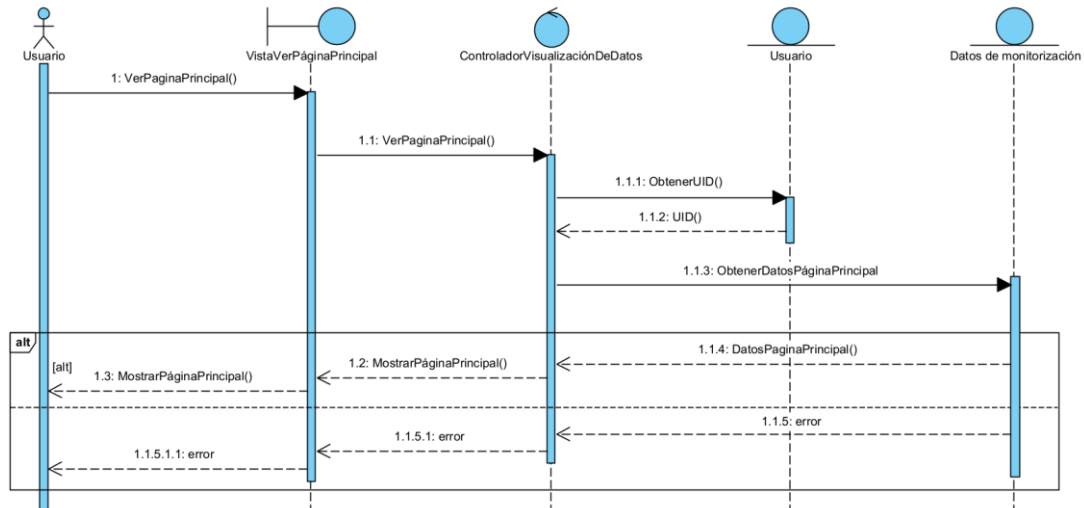


Ilustración 7: Diagrama de secuencia Ver página principal

sd UC-0012 Mostrar datos de los relojes en tiempo real

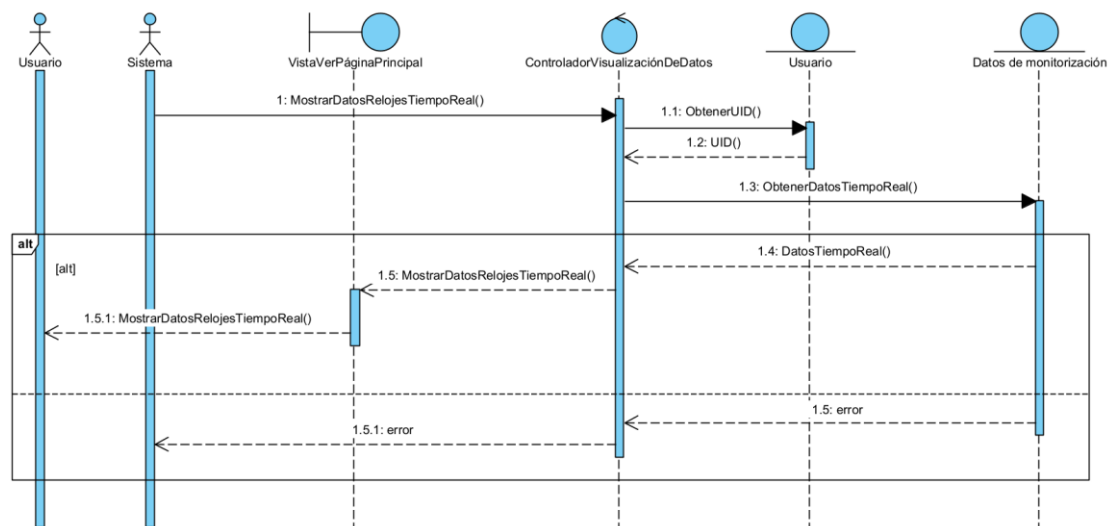


Ilustración 8: Diagrama de secuencia Mostrar datos de los relojes en tiempo real

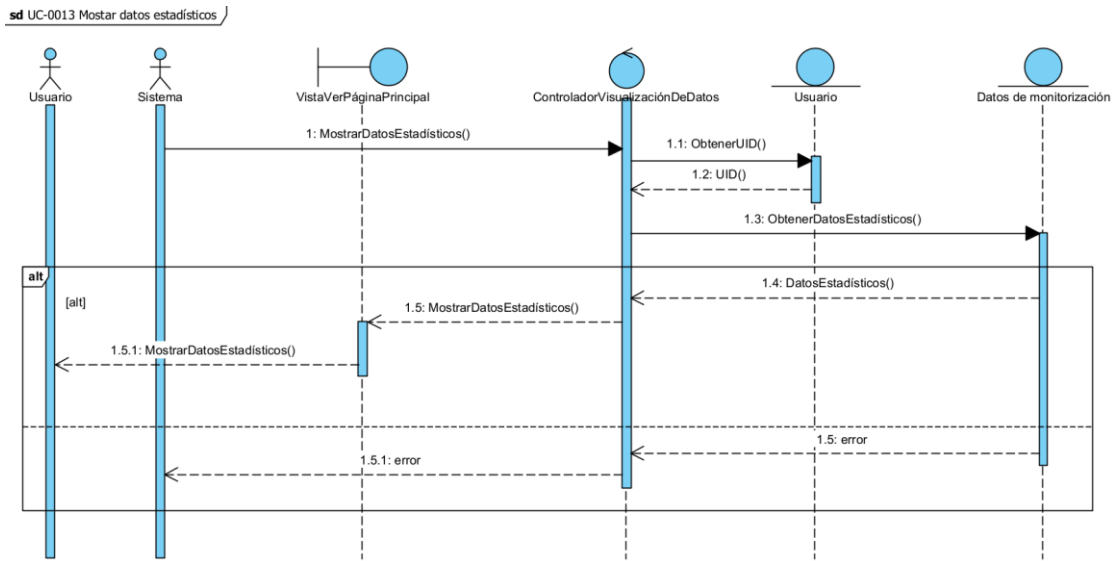


Ilustración 9: Diagrama de secuencia Mostrar datos estadísticos

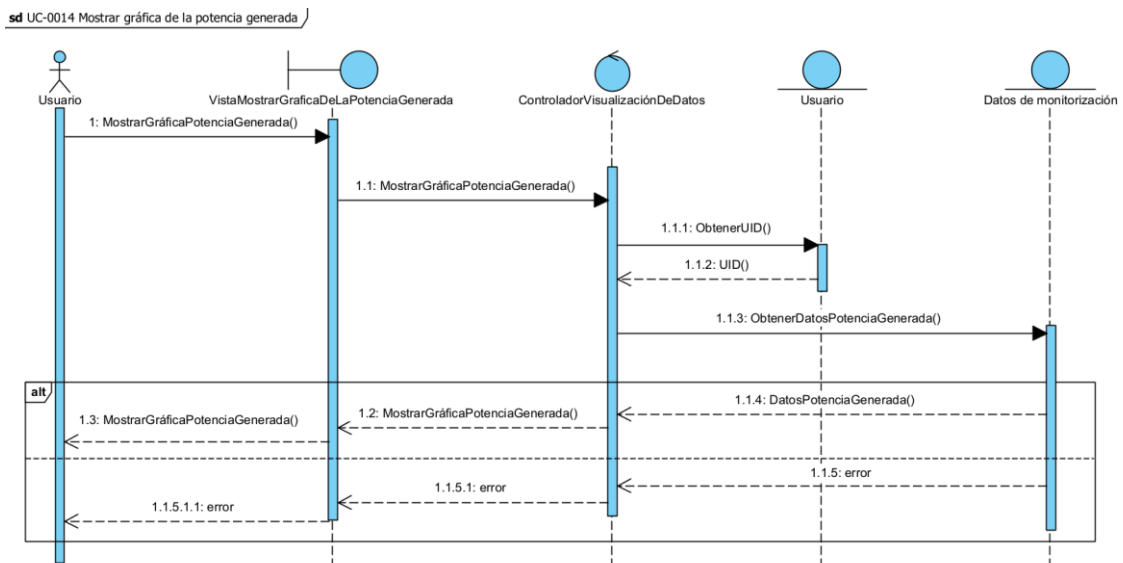


Ilustración 10: Diagrama de secuencia Mostrar gráfica de la potencia generada

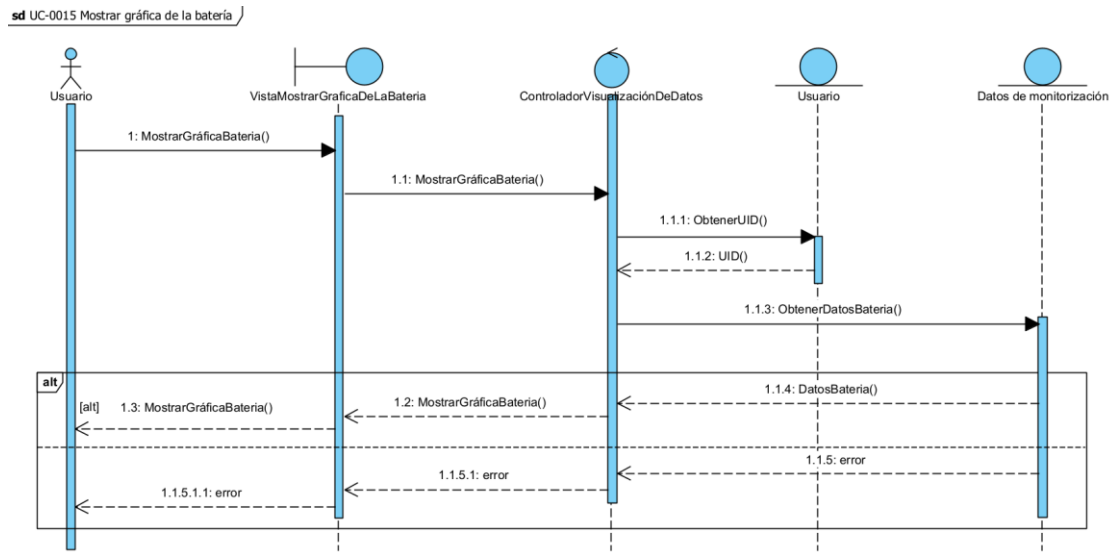


Ilustración 11: Diagrama de secuencia Mostrar gráfico de la batería

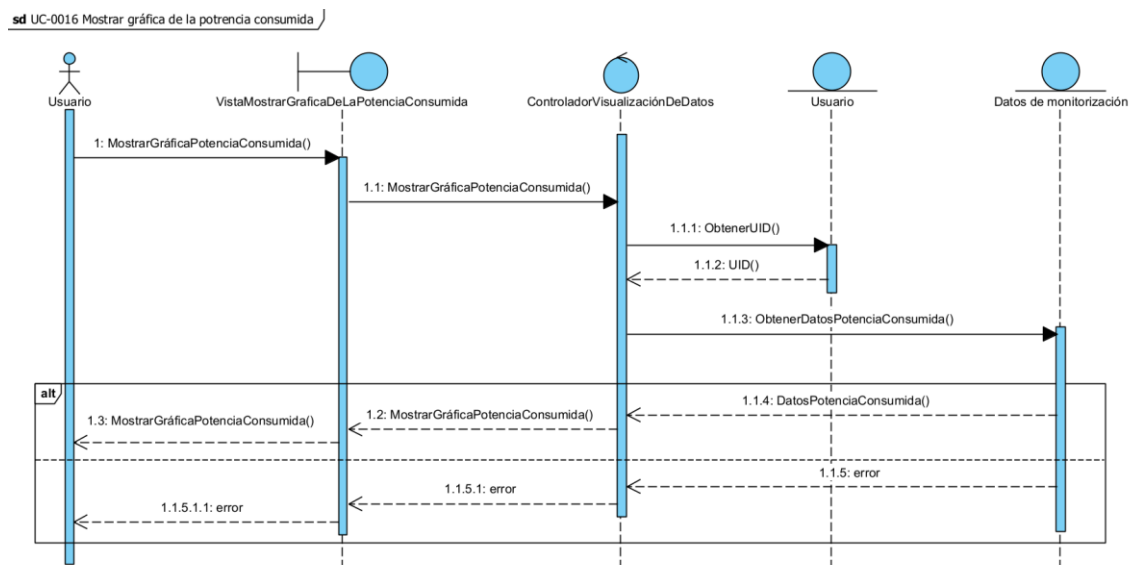


Ilustración 12: Diagrama de secuencia Mostrar gráfico de la potencia consumida

5.2. Gestión de los roles del sistema

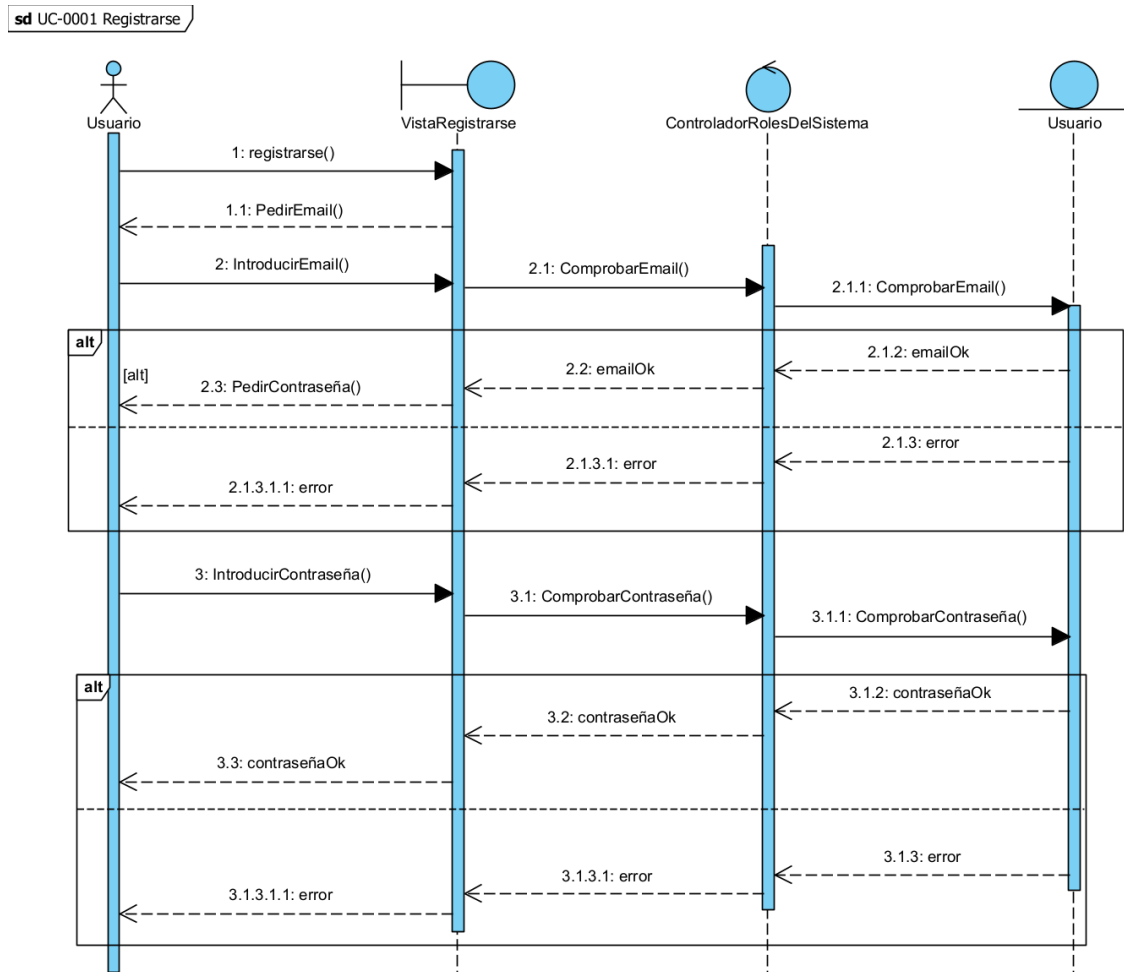


Ilustración 13: Diagrama de secuencia Registrarse

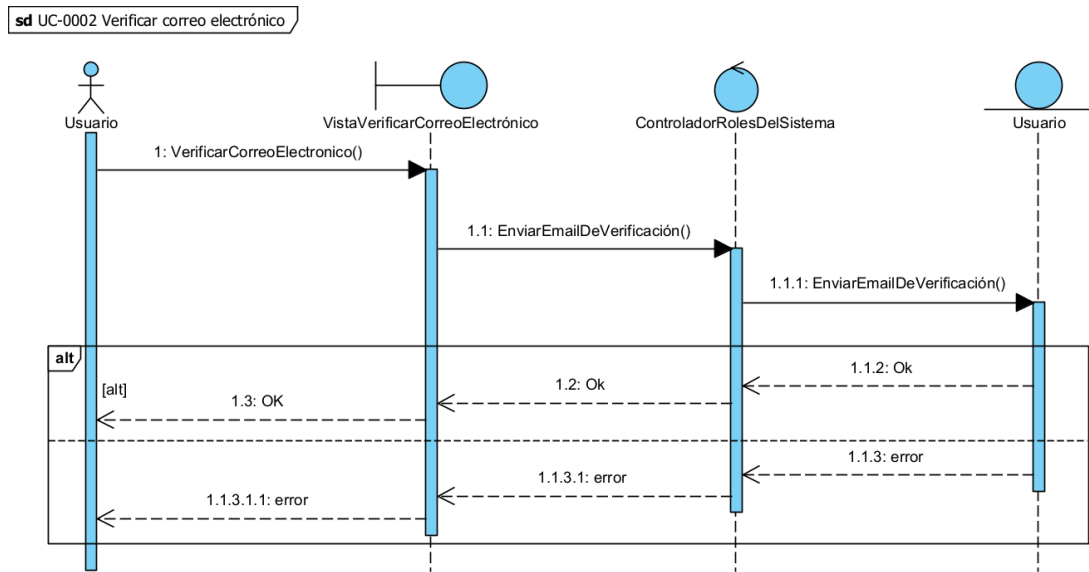


Ilustración 14: Diagrama de secuencia Verificar correo electrónico

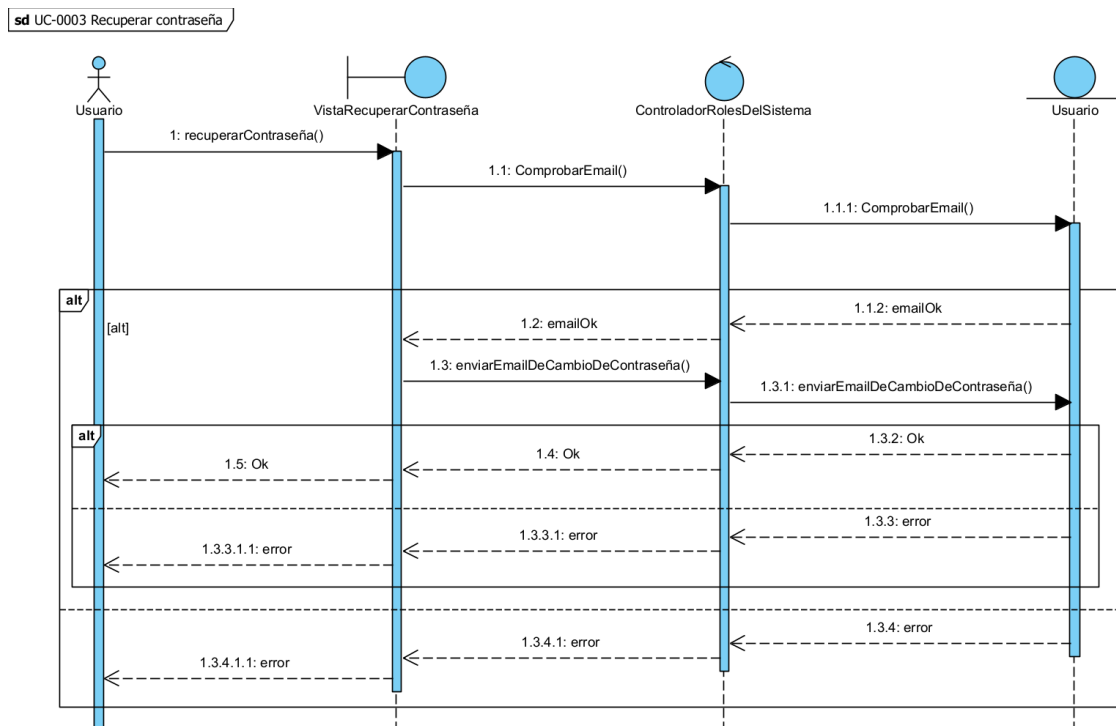


Ilustración 15: Diagrama de secuencia Recuperar contraseña

sd UC-0004 Iniciar sesión

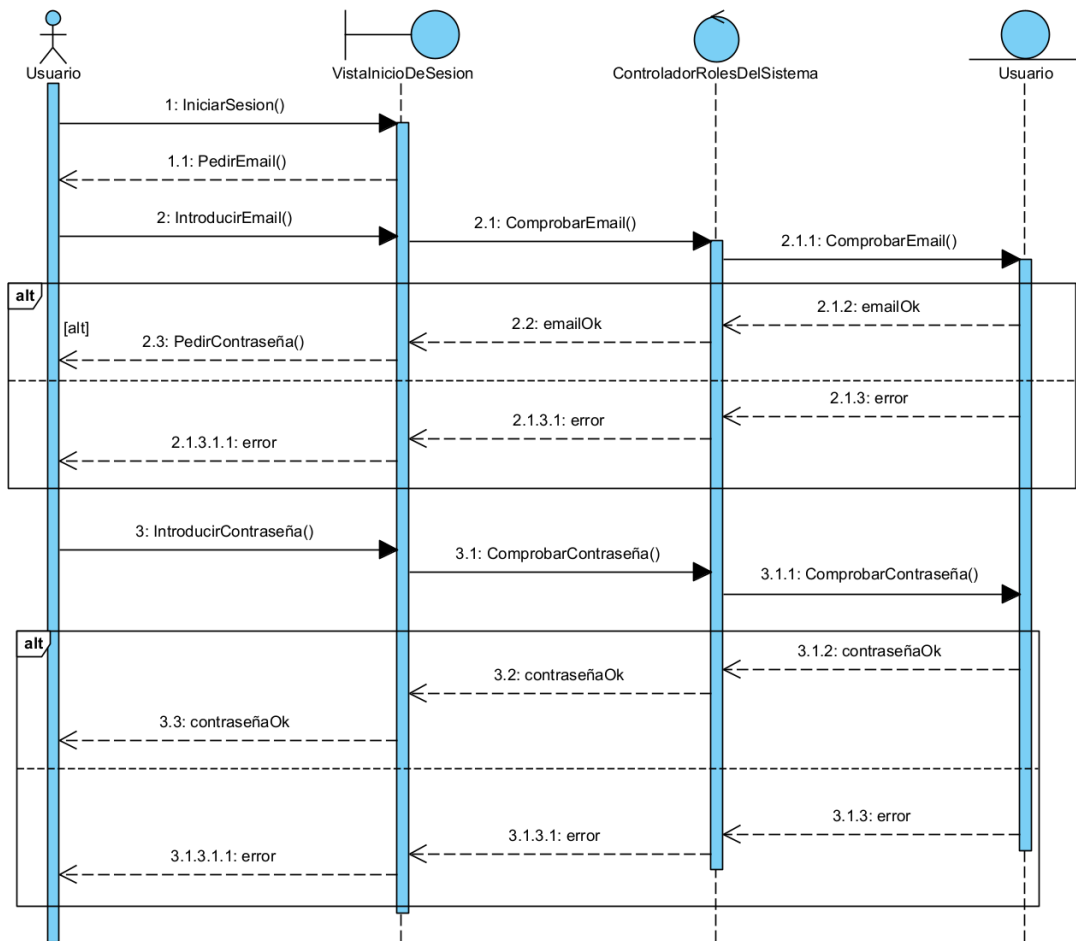


Ilustración 16: Diagrama de Secuencia Iniciar sesión

sd UC-0005 Cerrar Sesión

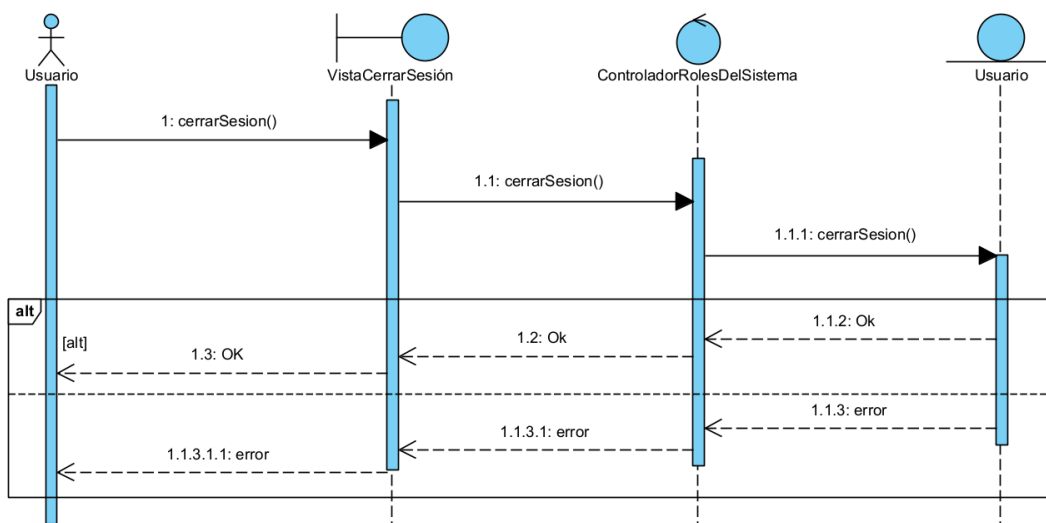


Ilustración 17: Diagrama de secuencia Cerrar sesión

sd UC-0006 Modificar correo electrónico

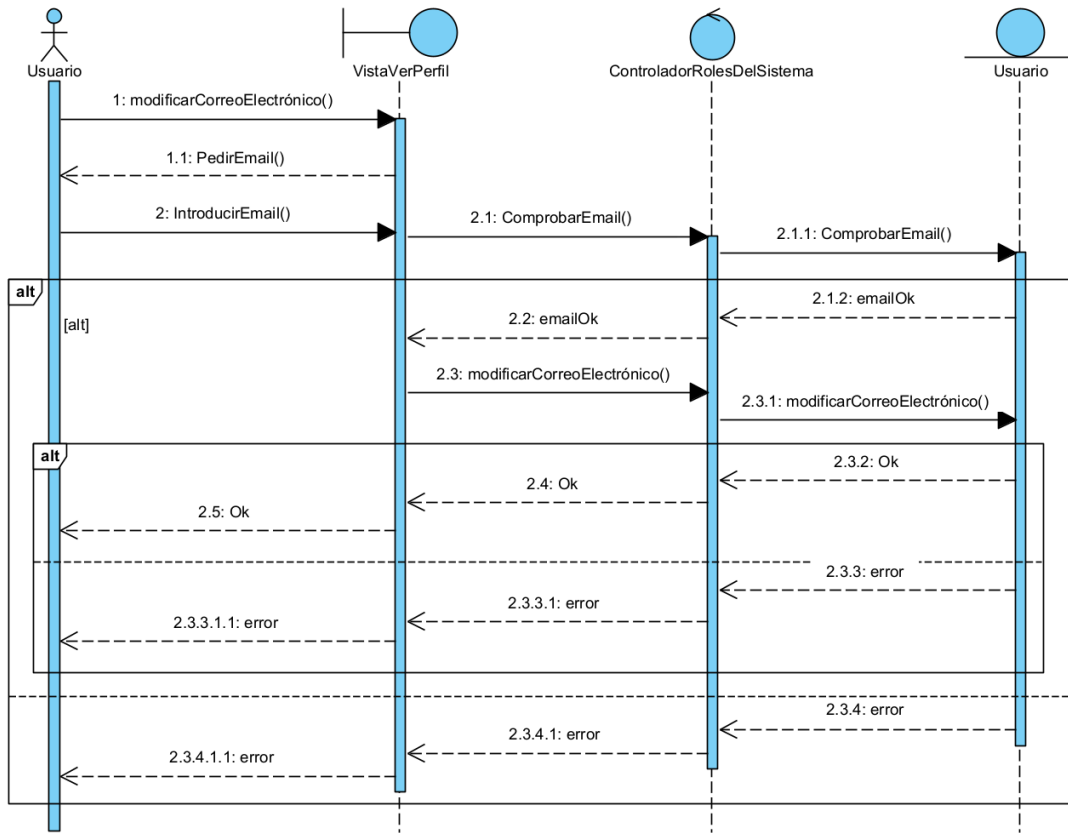


Ilustración 18: Diagrama de secuencia Modificar correo electrónico

sd UC-0007 Modificar contraseña

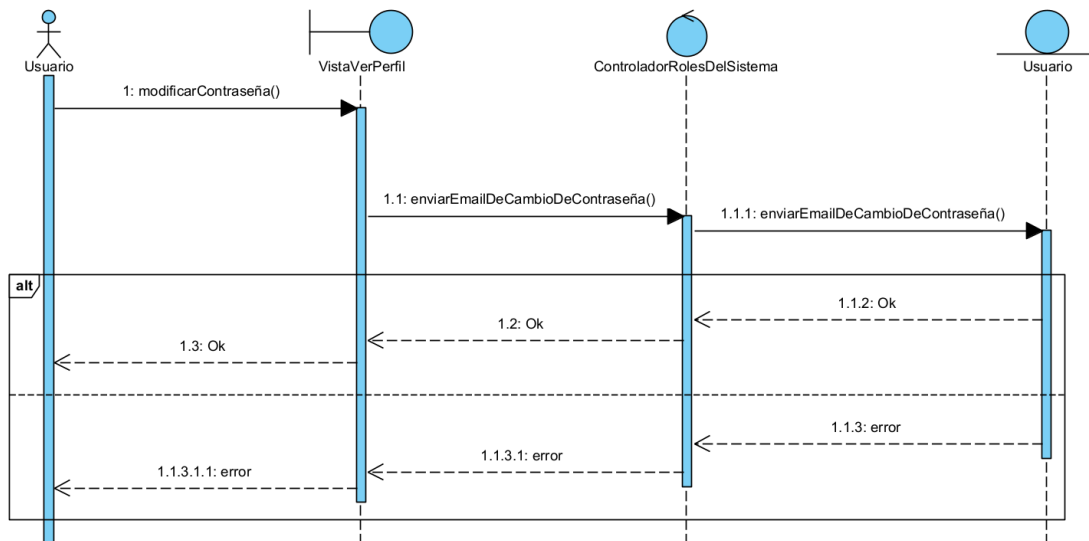


Ilustración 19: Diagrama de secuencia Modificar contraseña

sd UC-0008 Eliminar cuenta

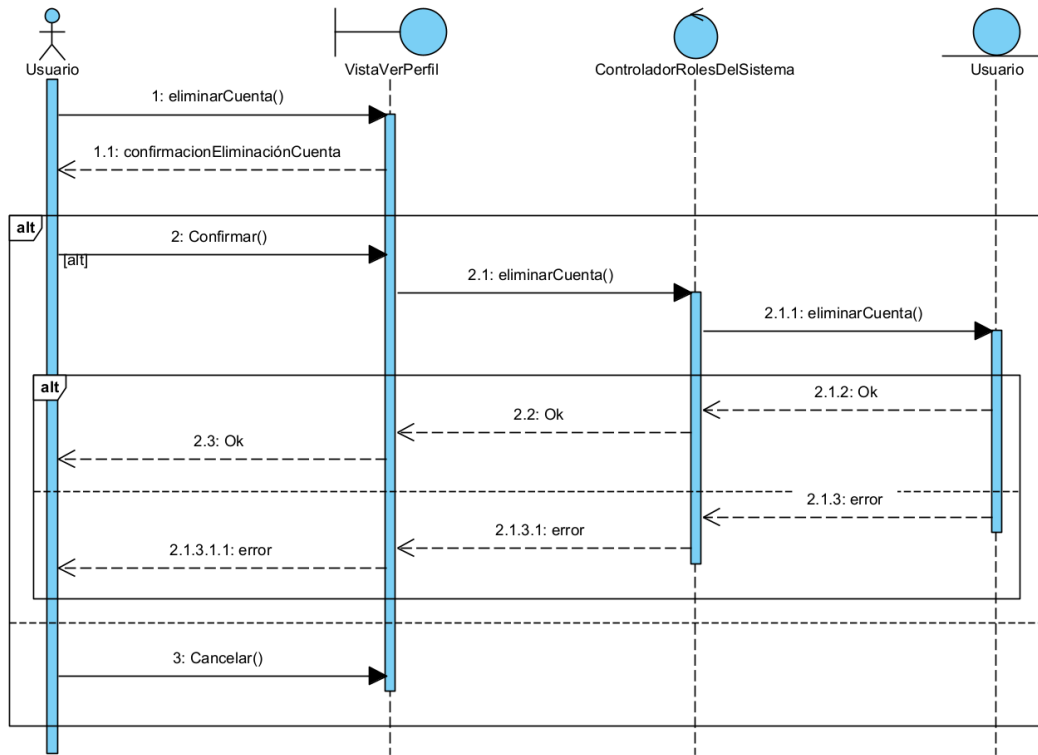


Ilustración 20: Diagrama de secuencia Eliminar Cuenta

sd UC-0009 Ver Perfil

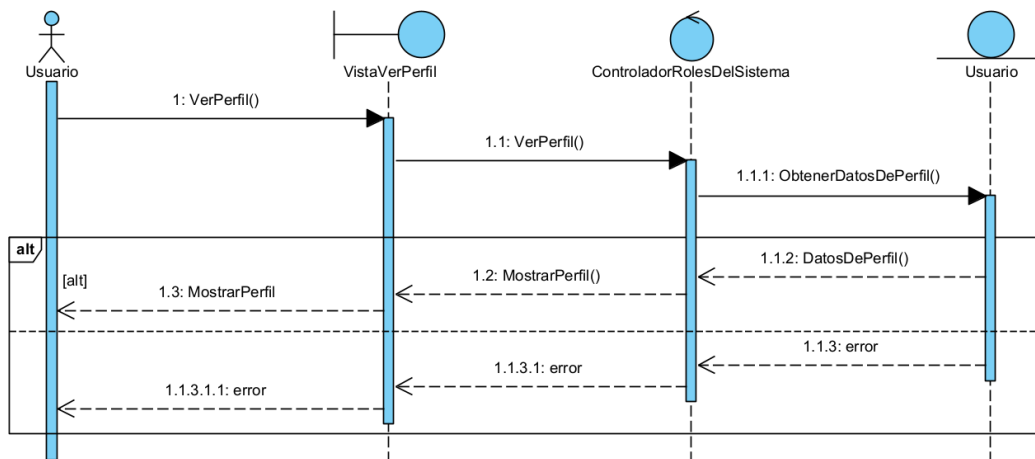


Ilustración 21: Diagrama de secuencia Ver Perfil

sd UC-0010 Enviar correo electrónico de cambio de contraseña

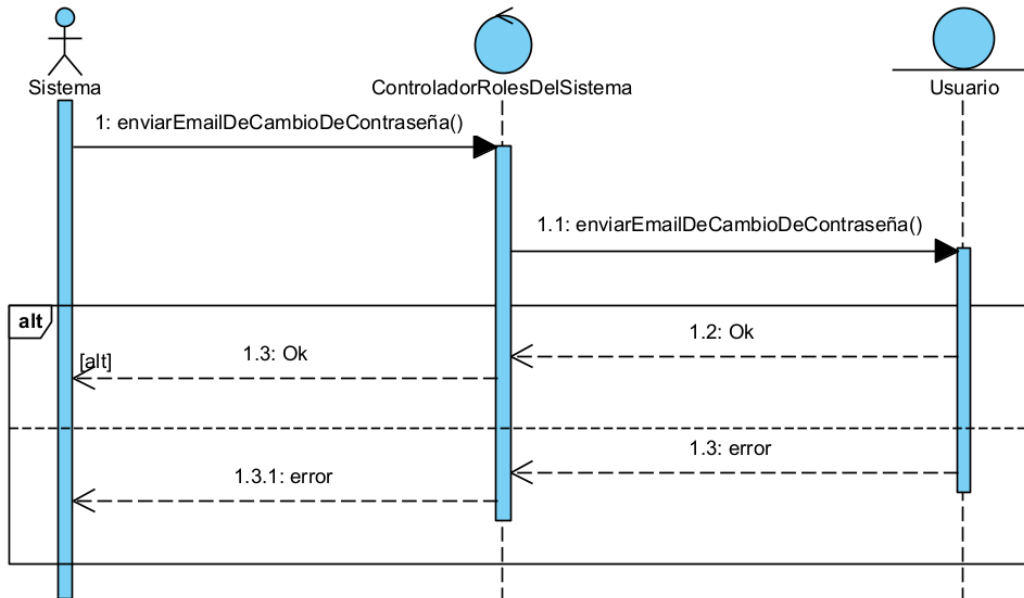


Ilustración 22: Diagrama de secuencia Enviar correo electrónico de cambio de contraseña

sd UC-0017 Obtener UID

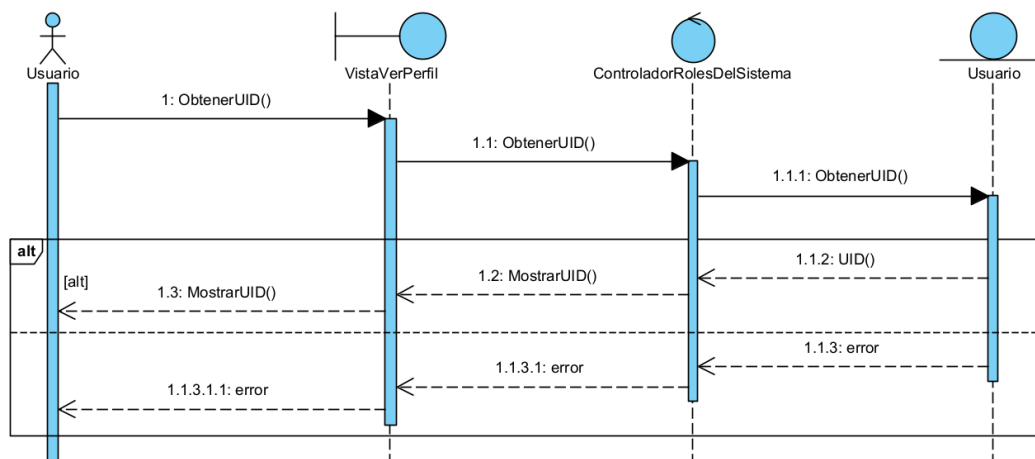


Ilustración 23: Diagrama de secuencia Obtener UID

5.3. Gestión de la monitorización de la instalación

sd UC-0018 Inicialización de la monitorización

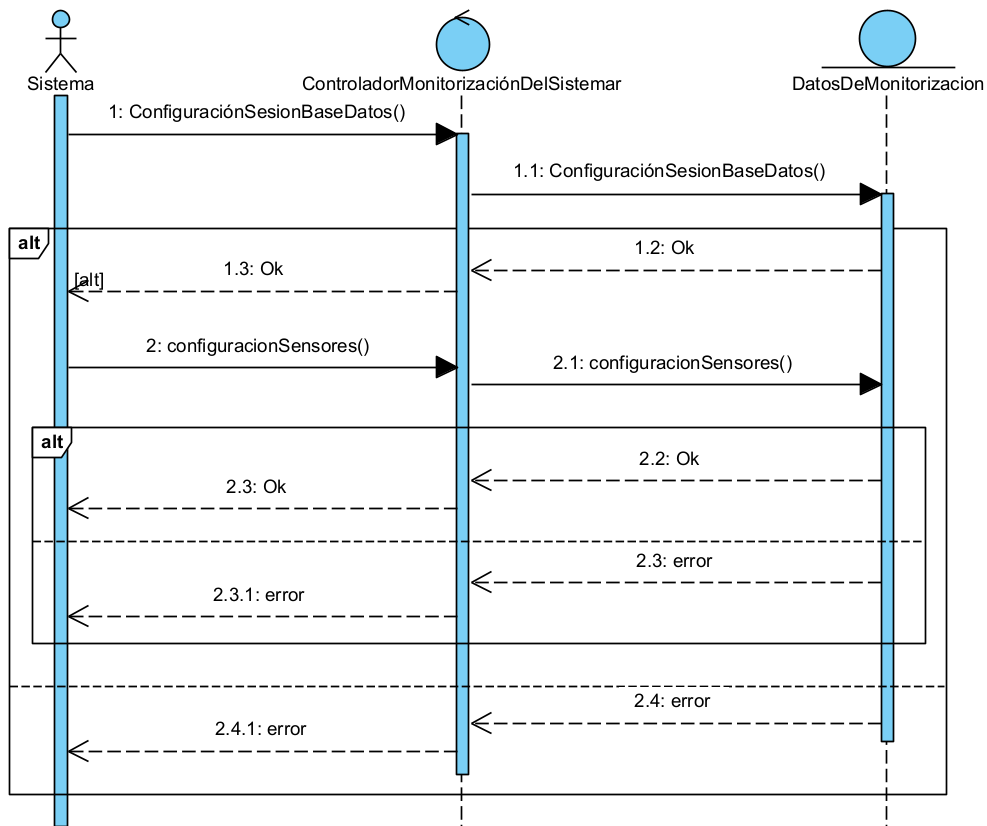


Ilustración 24: Diagrama de secuencia Inicialización de la monitorización

sd UC-0019 Leer información de los sensores

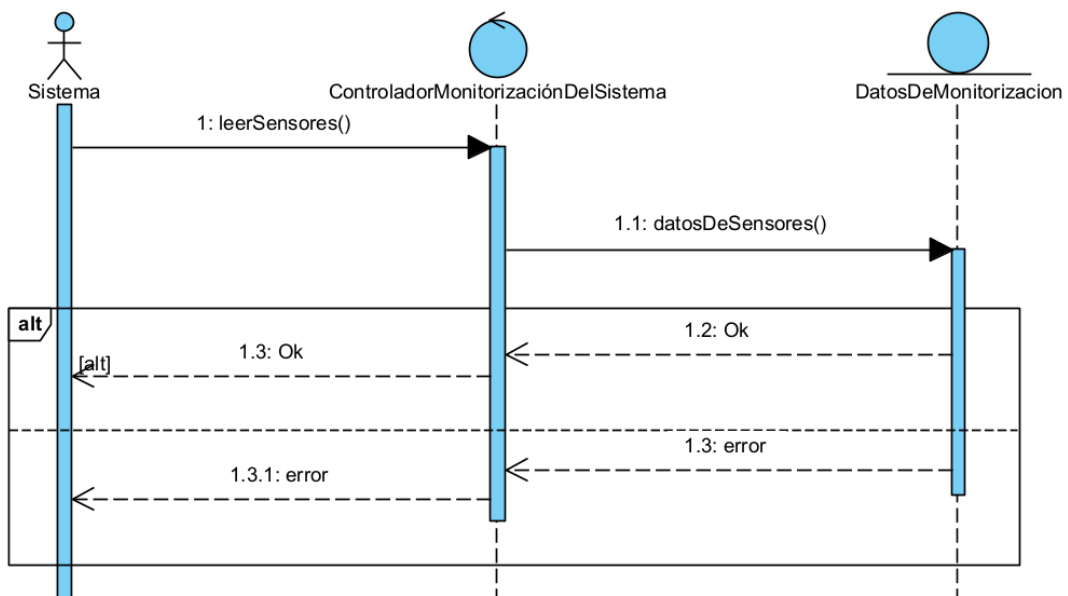


Ilustración 25: Diagrama de secuencia Leer información de los sensores

sd UC-0020 Enviar información a la base de datos

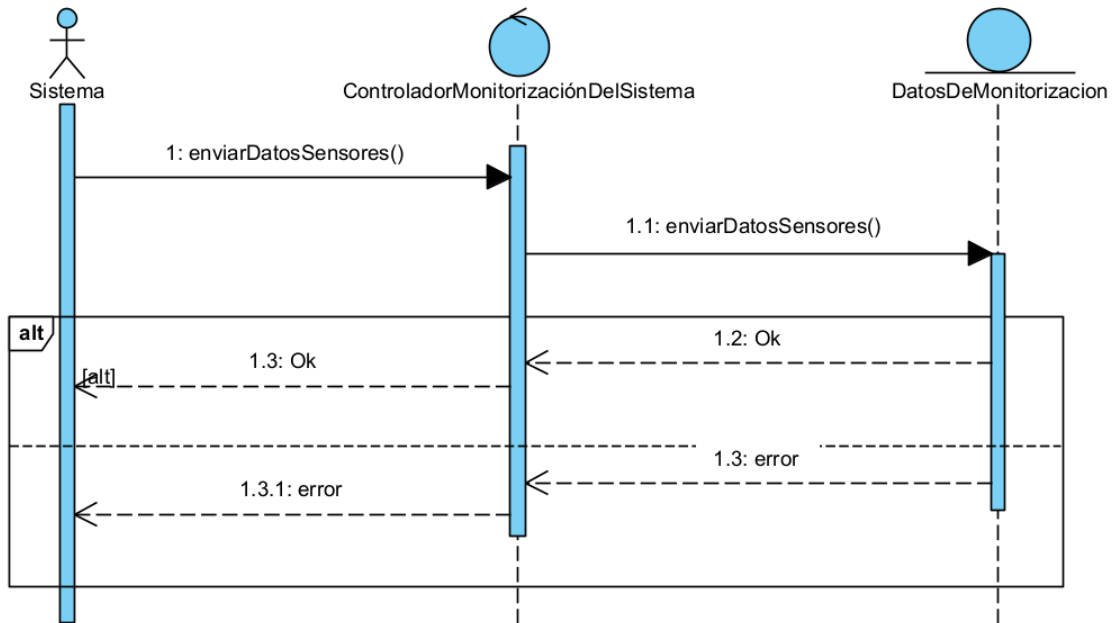


Ilustración 26: Diagrama de secuencia Enviar información a la base de datos

Aplicación para la gestión de una instalación fotovoltaica

Anexo IV – Diseño del sistema



VNiVERSiDAD
D SALAMANCA

Trabajo de Fin de Grado

Grado de Ingeniería Informática

Septiembre 2021

Tutor:

Iván Álvarez Navia

Alumno:

Miguel Sánchez González

Tabla de contenidos

1. Introducción	6
2. Diseño arquitectónico	7
2.1. Patrón arquitectónico	7
2.2. Subsistemas de diseño.....	8
2.3. Vista arquitectónica	8
2.4. Clases de diseño.....	9
3. Realización de casos de uso	10
3.1. Gestión de la visualización de datos	10
3.2. Gestión de los roles del sistema	16
3.3. Gestión de la monitorización de la instalación.....	21
4. Modelo de despliegue.....	23

Tabla de ilustraciones

Ilustración 1: Patrón MVC.....	7
Ilustración 2: Vista de subsistemas de diseño.....	8
Ilustración 3: Vista arquitectónica.....	8
Ilustración 4: WebApp y Monitorización - Modelo.....	9
Ilustración 5: WebApp - Vista.....	9
Ilustración 6: Monitorización - Controlador.....	9
Ilustración 7: WebApp - Controlador.....	9
Ilustración 8: Diagrama de secuencia Ver página principal.....	10
Ilustración 9: Diagrama de secuencia Mostrar datos de los relojes en tiempo real.....	11
Ilustración 10: Diagrama de secuencia Mostrar datos estadísticos.....	12
Ilustración 11: Diagrama de secuencia Mostrar gráfica de la potencia generada.....	13
Ilustración 12: Diagrama de secuencia Mostrar gráfica de la batería.....	14
Ilustración 13: Diagrama de secuencia Mostrar gráfica de la potencia consumida.....	15
Ilustración 14: Diagrama de secuencia Registrarse.....	16
Ilustración 15: Diagrama de secuencia Verificar correo electrónico.....	16
Ilustración 16: Diagrama de secuencia Recuperar contraseña.....	17
Ilustración 17: Diagrama de secuencia Iniciar Sesión.....	17
Ilustración 18: Diagrama de secuencia Cerrar Sesión.....	18
Ilustración 19: Diagrama de secuencia Modificar correo electrónico.....	18
Ilustración 20: Diagrama de secuencia Modificar contraseña.....	19
Ilustración 21: Diagrama de secuencia Eliminar cuenta.....	19
Ilustración 22: Diagrama de secuencia Ver perfil.....	20
Ilustración 23: Diagrama de secuencia Enviar correo electrónico de cambio de contraseña.....	20
Ilustración 24: Diagrama de secuencia Obtener UID.....	21
Ilustración 25: Diagrama de secuencia Inicialización de la monitorización.....	21
Ilustración 26: Diagrama de secuencia Leer información de los sensores.....	22
Ilustración 27: Diagrama de secuencia Enviar información a la base de datos.....	22
Ilustración 28: Modelo de despliegue del sistema.....	23

1. Introducción

Este anexo muestra la documentación del modelo de diseño del sistema, en el que se lleva a cabo una solución para el problema, que servirá como base en la fase de implementación.

Los apartados que se van a ver son los siguientes:

- **Diseño arquitectónico:** en este apartado se explicará el patrón arquitectónico usado, los subsistemas de diseño de los que está compuesto, la vista arquitectónica del sistema y el desarrollo de las clases de diseño junto con sus atributos.
- **Realización de casos de uso:** aquí se tratan la realización de los casos de uso, estos son una evolución de los que se especificaron en la parte de análisis.
- **Modelo de despliegue:** muestra las interacciones de los elementos del sistema y su distribución a nivel de software.

De nuevo, se ha usado la herramienta Visual Paradigm para la elaboración de los diagramas que se ven en este documento.

2. Diseño arquitectónico

2.1. Patrón arquitectónico

Para el desarrollo del sistema se ha elegido el patrón MVC (Modelo-Vista-Controlador, del inglés Model-View-Controller). Es un tipo de patrón de diseño en el que se separa la parte visible del sistema de las funcionalidades, con el objetivo de facilitar el desarrollo y mantenimiento de aplicaciones, además de facilitar la reutilización del código.

Este patrón se subdivide en tres componentes:

- **Modelo:** contiene la información del sistema. Se comunica con el “controlador” para realizar operaciones de manipulación de datos y envía a la “vista” la información requerida. En nuestro caso los datos del sistema se almacenan en una base de datos.
- **Vista:** se encarga de la representación de la información que recibe del “modelo” o del “controlador”, y de comunicar las acciones del usuario para su posterior procesamiento.
- **Controlador:** en el se encuentran las funcionalidades principales del sistema. Responde a los eventos del usuario a través de la “vista” y realiza peticiones al “modelo” para obtener la información con la que trabajar.

En la Ilustración 1 se pueden ver las relaciones entre los componentes del patrón MVC.

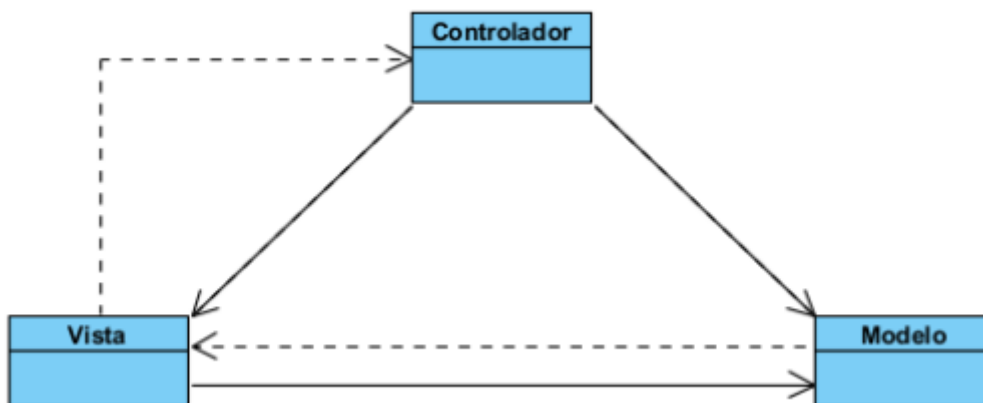


Ilustración 1: Patrón MVC

2.2. Subsistemas de diseño

Se realizará la división del sistema en sus diferentes componentes siguiendo el patrón Modelo-Vista-Controlador.

En la **¡Error! No se encuentra el origen de la referencia.** se puede ver cada uno de los subsistemas del proyecto.

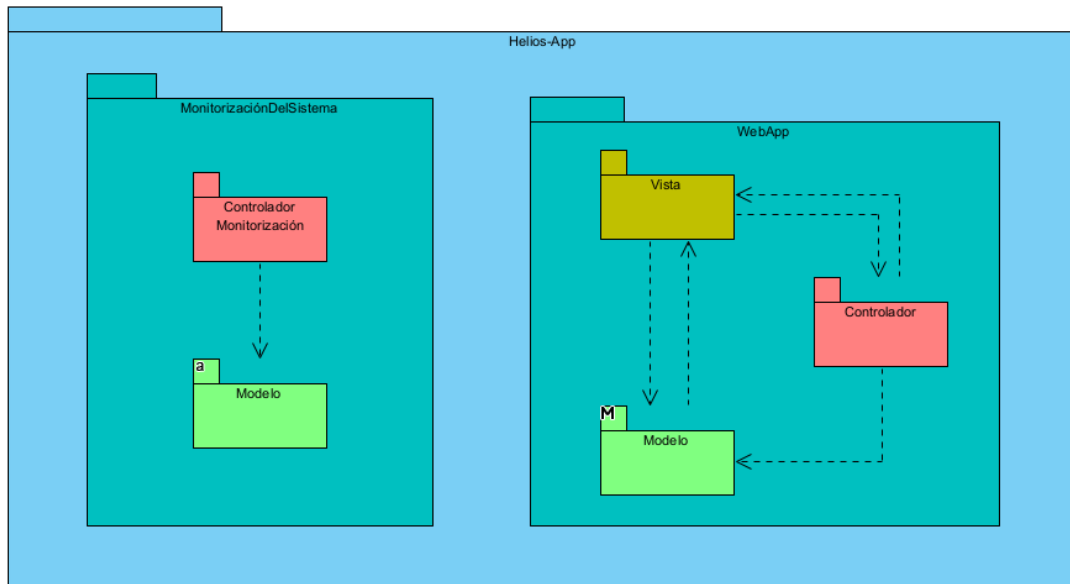


Ilustración 2: Vista de subsistemas de diseño

2.3. Vista arquitectónica

Esta vista representa una evolución de la que se obtuvo en el Anexo III – Análisis de requisitos, pero implementando el patrón arquitectónico Modelo-Vista-Controlador.

En la figura de la Ilustración 3 se puede observar en detalle las capas del patrón arquitectónico usado (MVC) en la arquitectura del sistema.

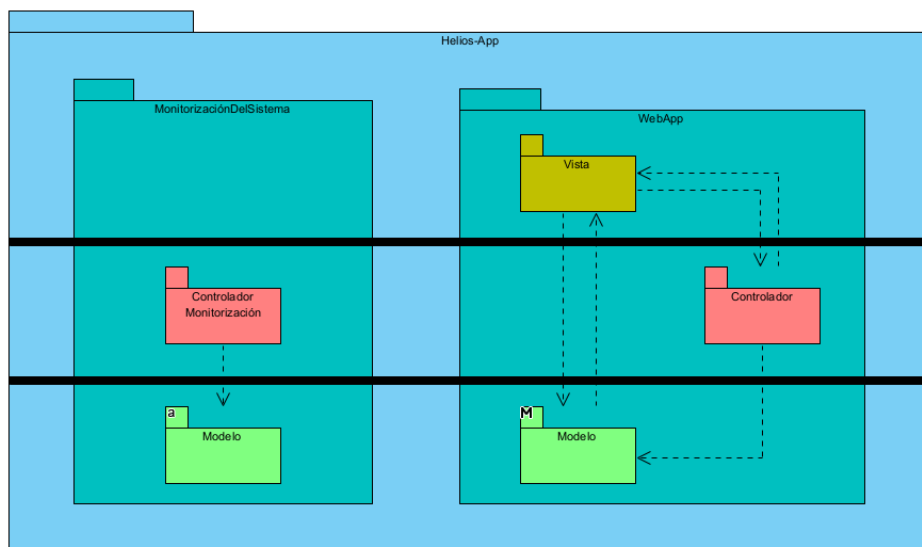


Ilustración 3: Vista arquitectónica

2.4. Clases de diseño

Son una evolución del modelo de dominio del Anexo III – Análisis de requisitos. En ellas se detallan los paquetes de cada subsistema.

Los diagramas de las clases de diseño se muestran en las ilustraciones Ilustración 4, Ilustración 5, Ilustración 6 e Ilustración 7.

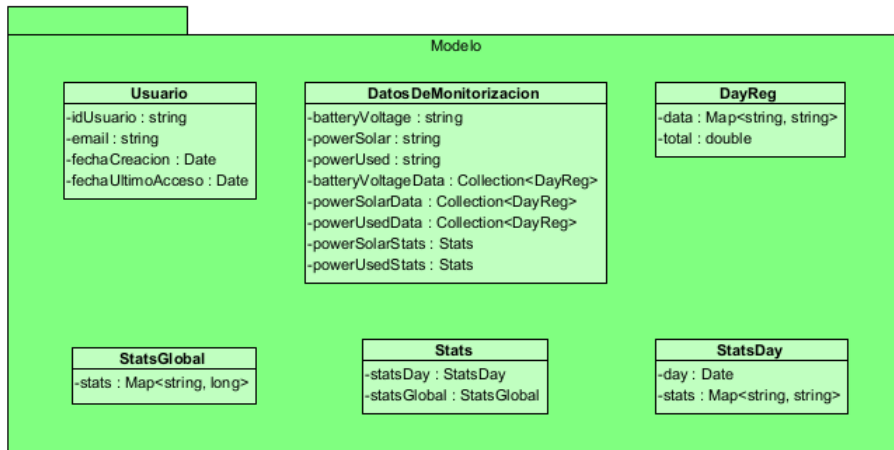


Ilustración 4: WebApp y Monitorización - Modelo

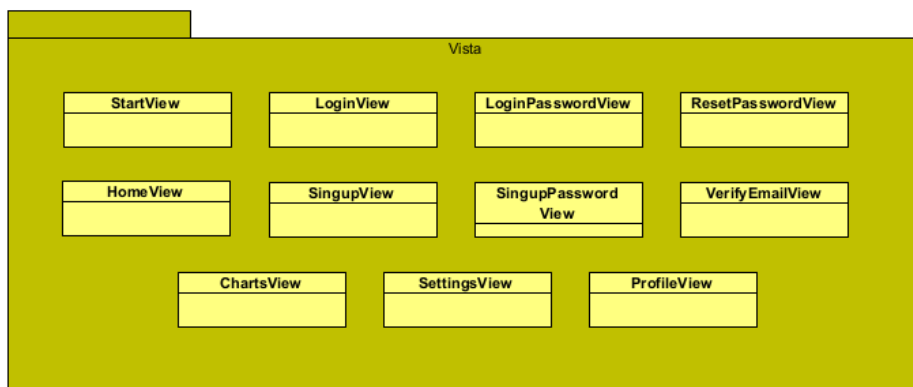


Ilustración 5: WebApp - Vista

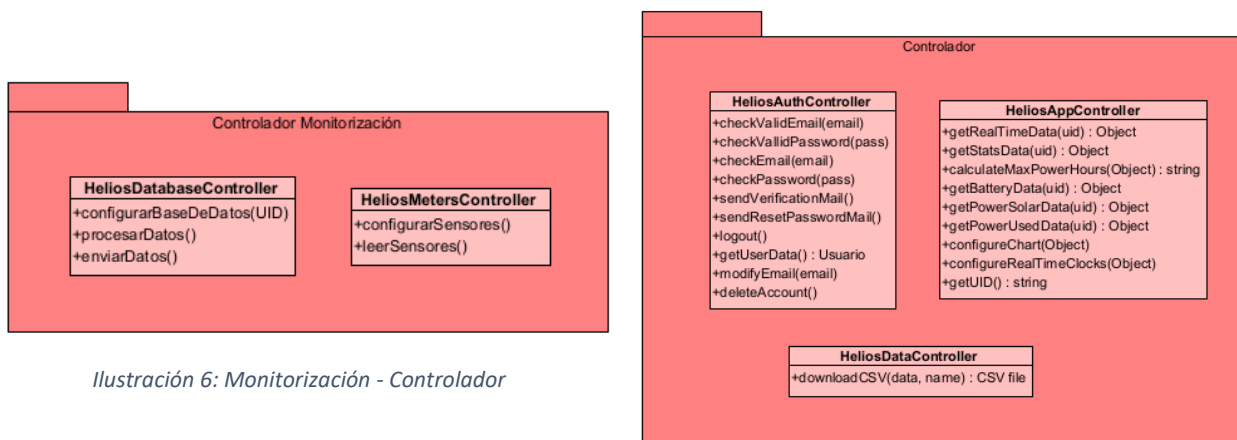


Ilustración 6: Monitorización - Controlador

Ilustración 7: WebApp - Controlador

3. Realización de casos de uso

3.1. Gestión de la visualización de datos

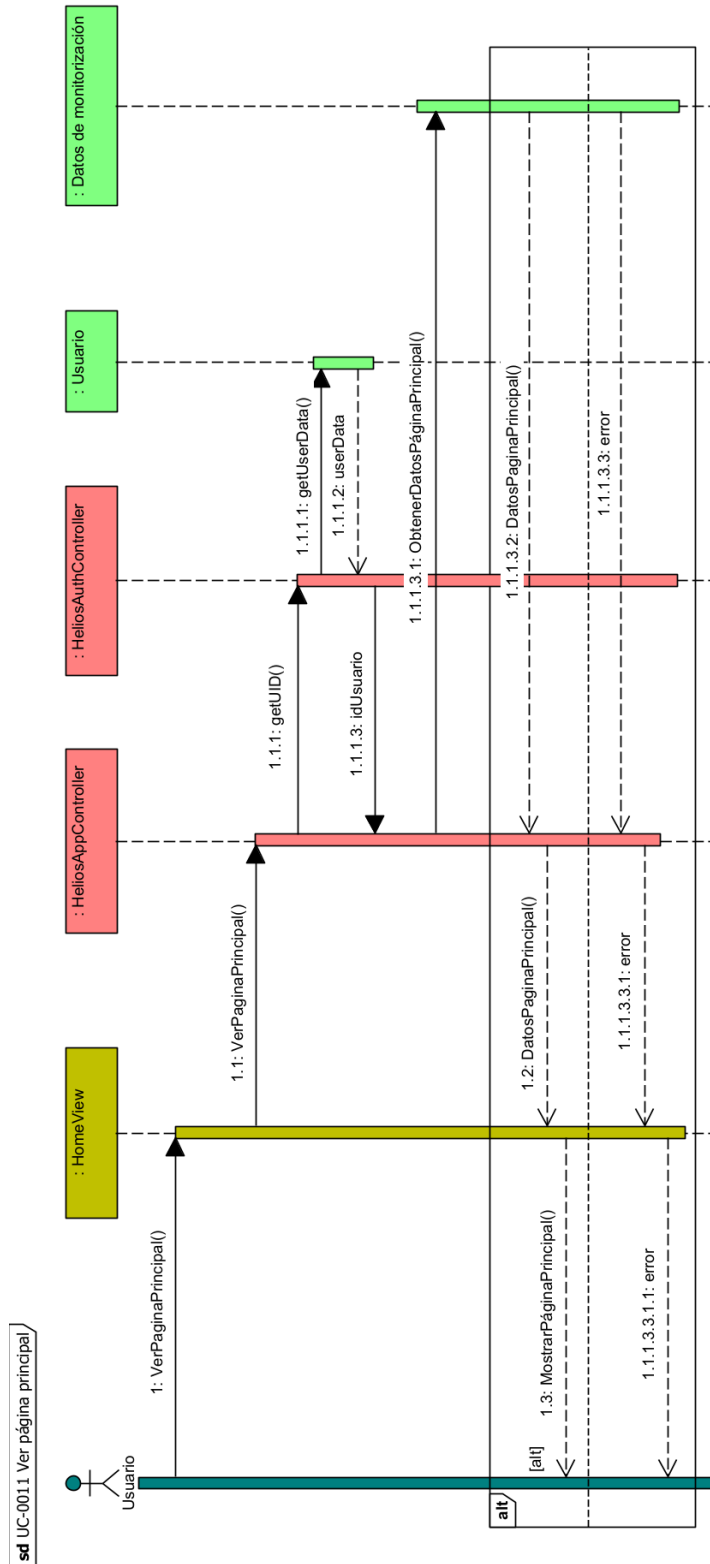


Ilustración 8: Diagrama de secuencia Ver página principal

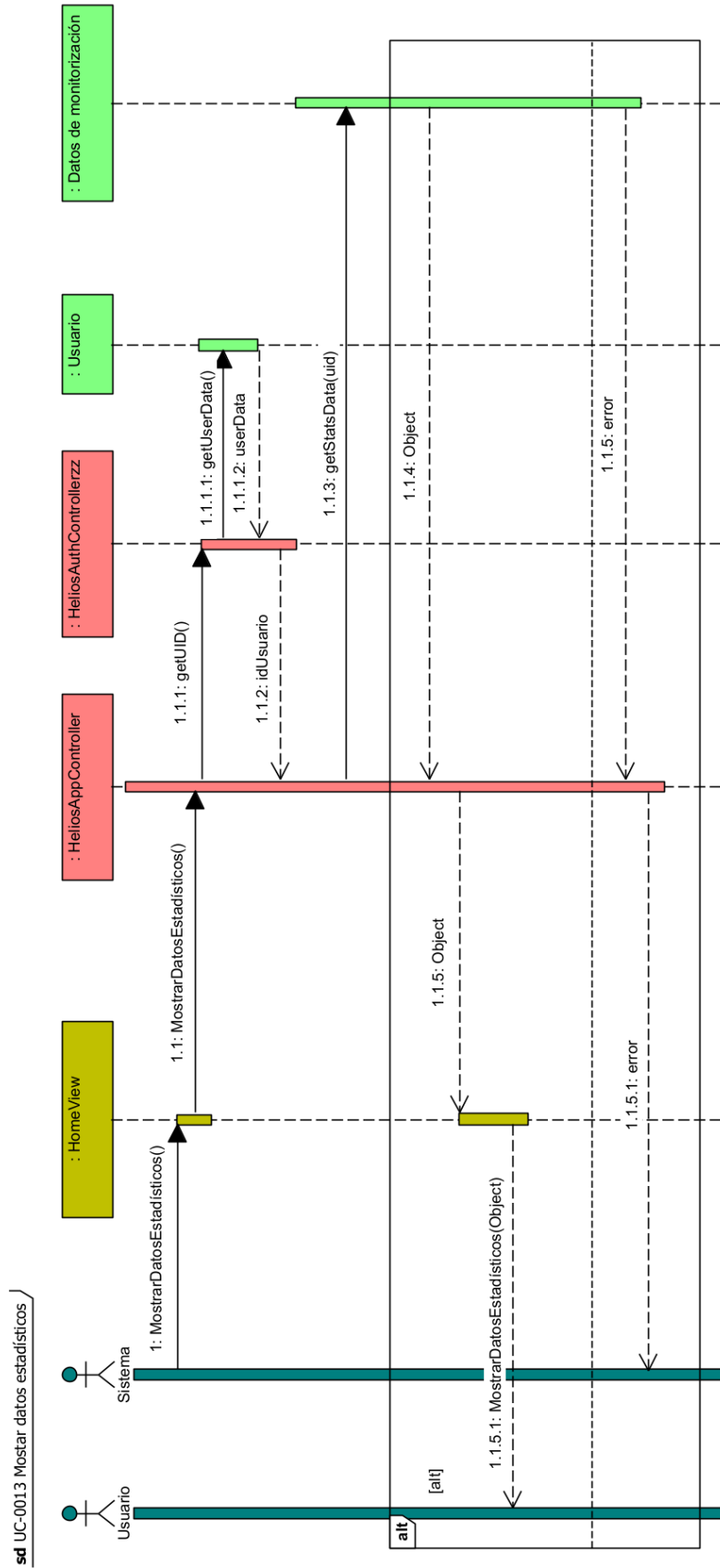


Ilustración 10: Diagrama de secuencia Mostrar datos estadísticos

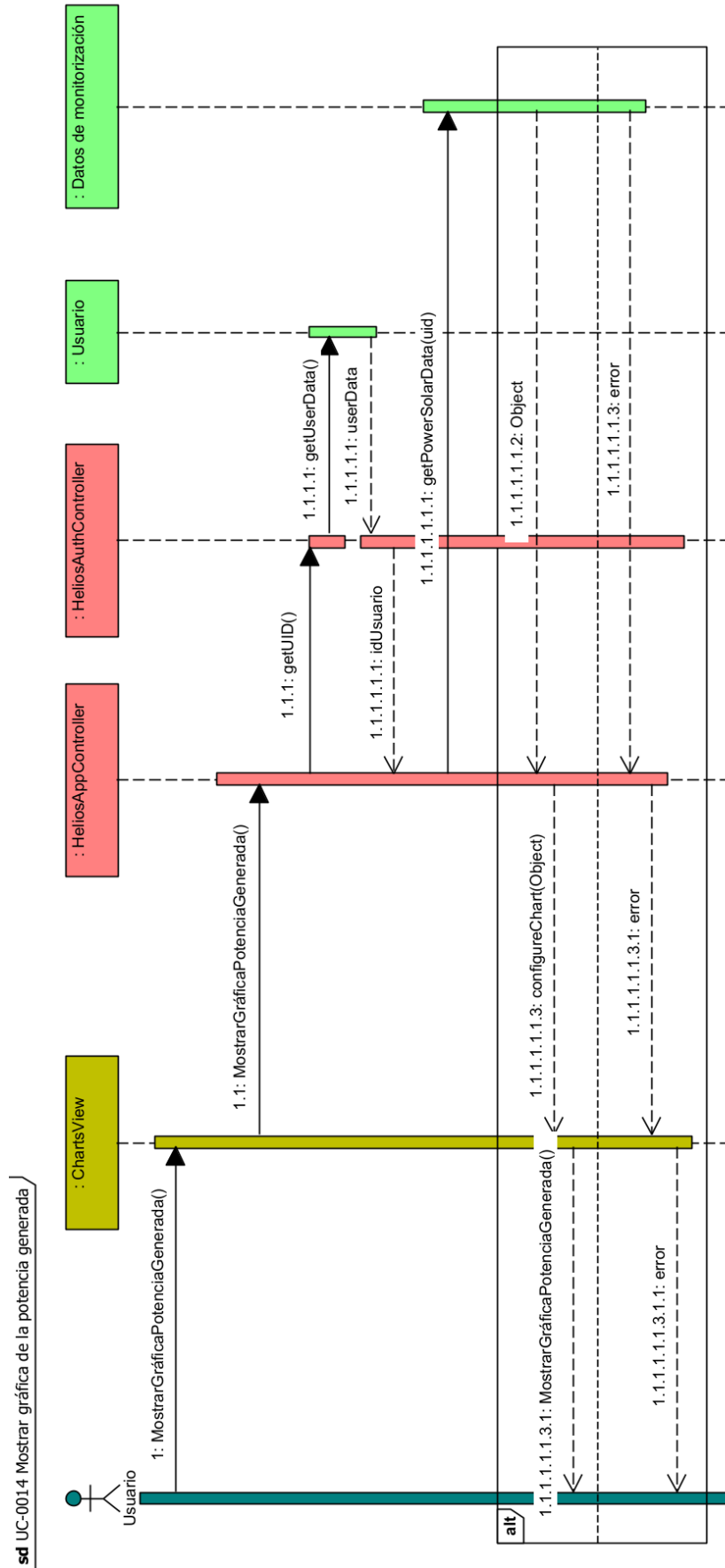


Ilustración 11: Diagrama de secuencia Mostrar gráfica de la potencia generada

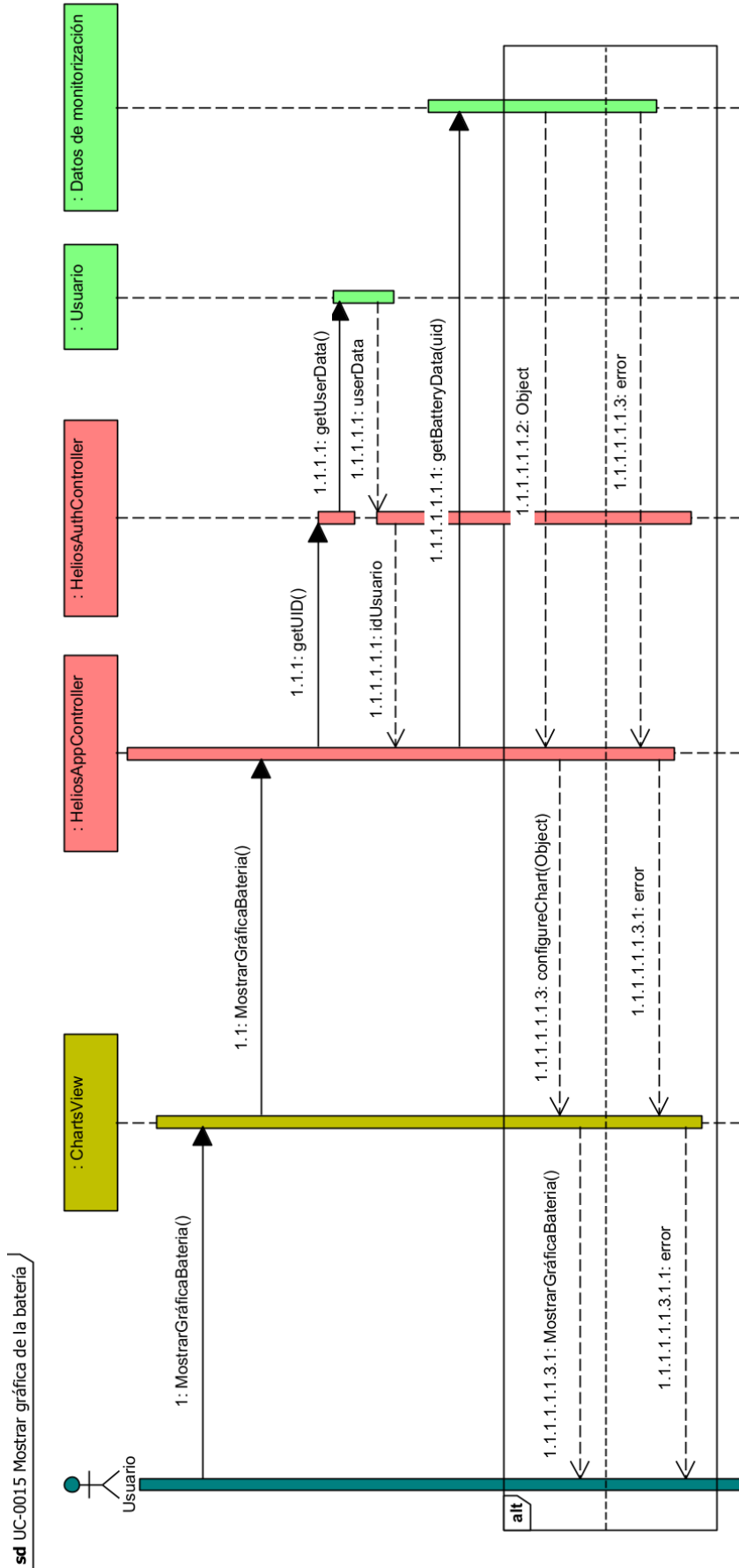


Ilustración 12: Diagrama de secuencia Mostrar gráfica de la batería

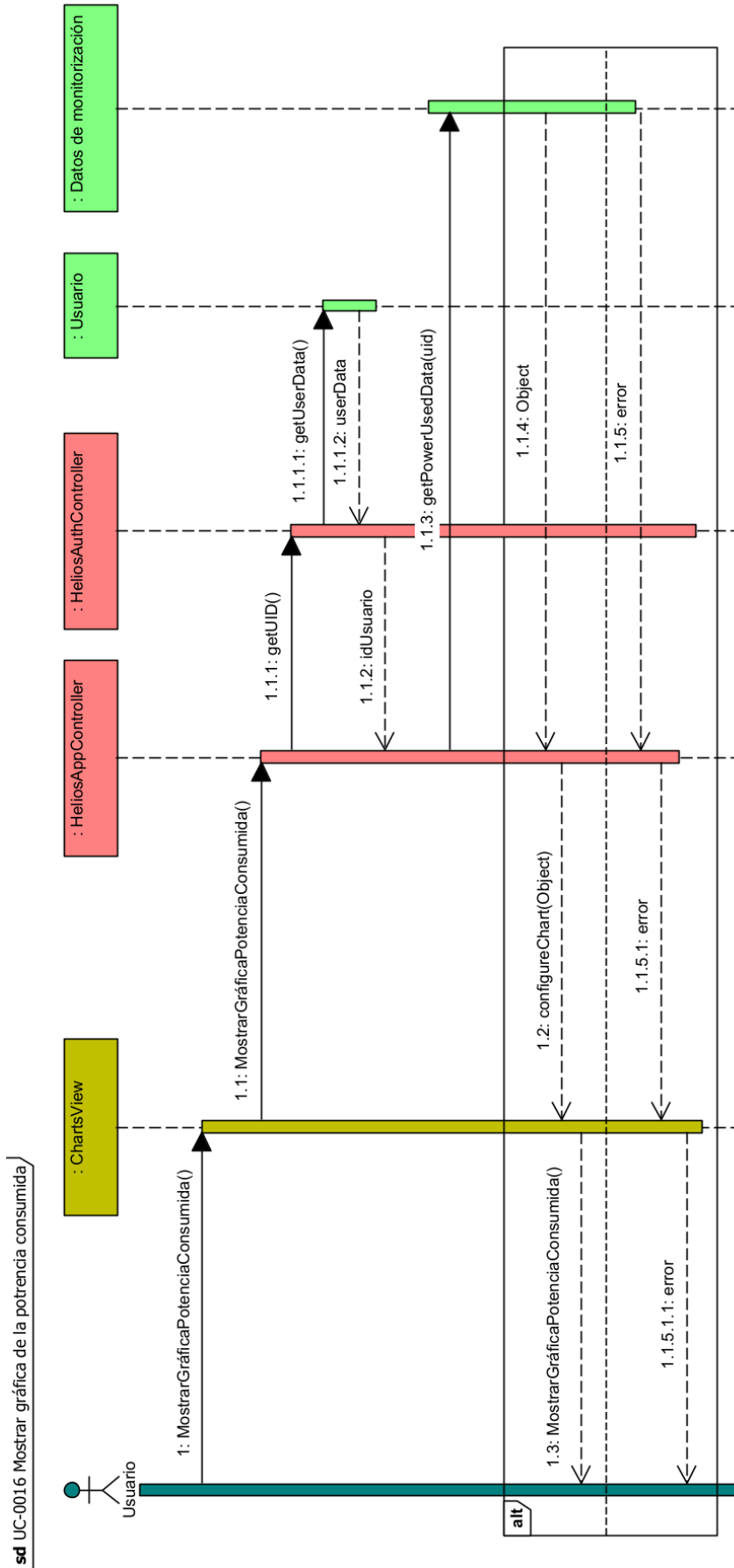


Ilustración 13: Diagrama de secuencia Mostrar gráfica de la potencia consumida

3.2. Gestión de los roles del sistema

sd UC-0001 Registrarse

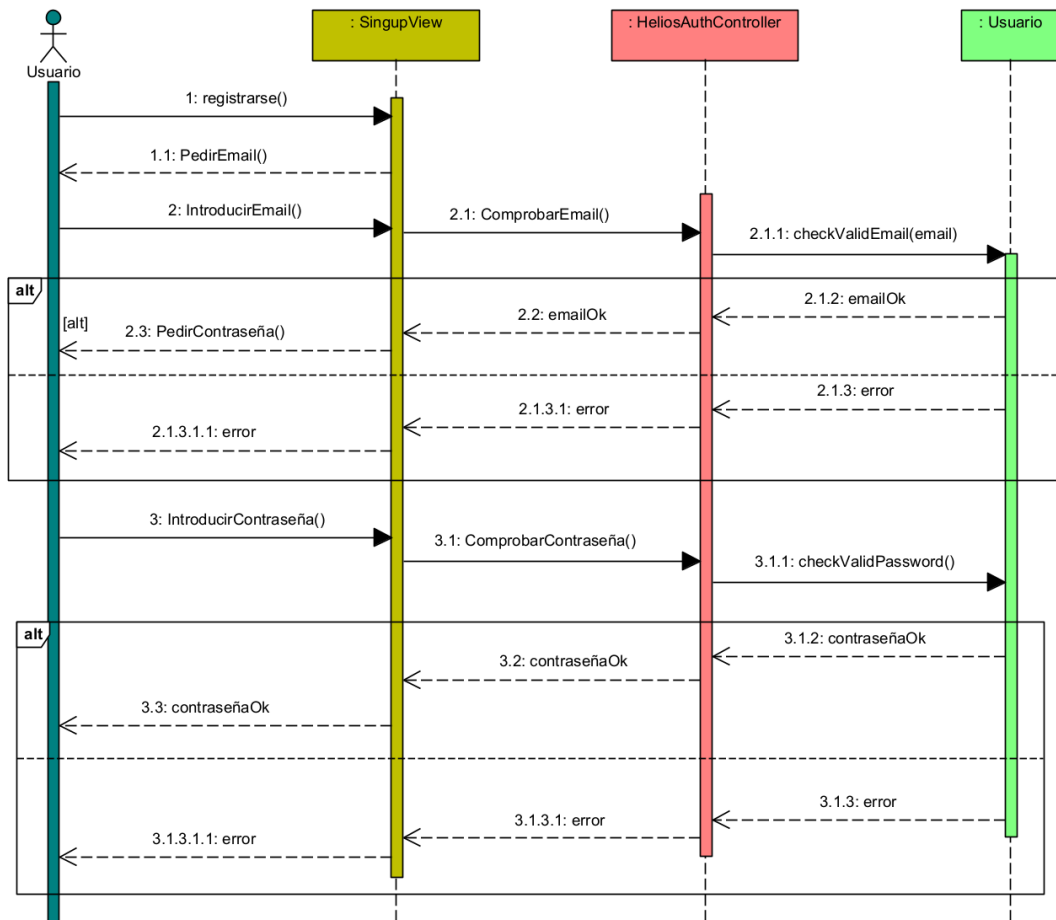


Ilustración 14: Diagrama de secuencia Registrarse

sd UC-0002 Verificar correo electrónico

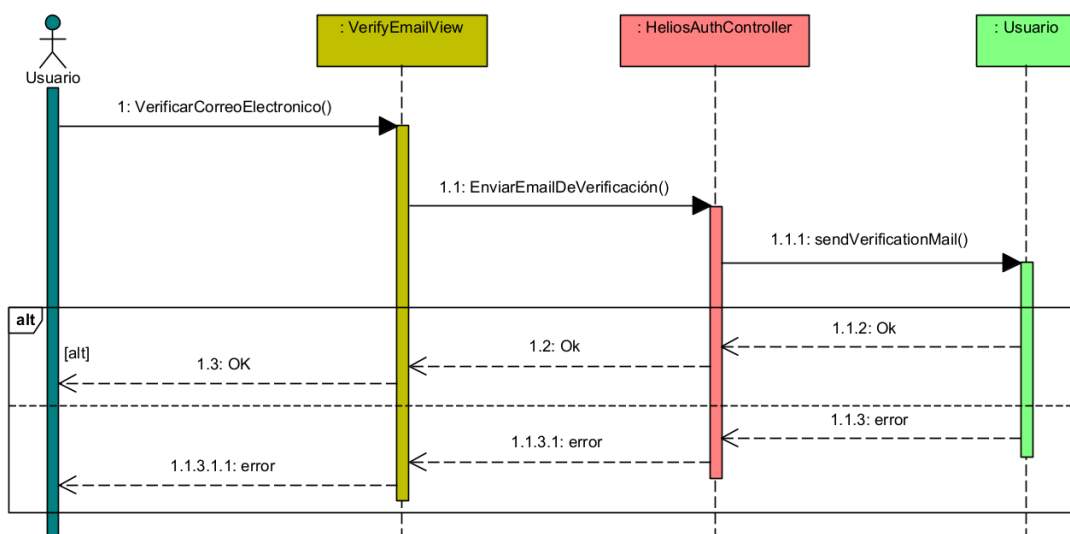


Ilustración 15: Diagrama de secuencia Verificar correo electrónico

sd UC-0003 Recuperar contraseña

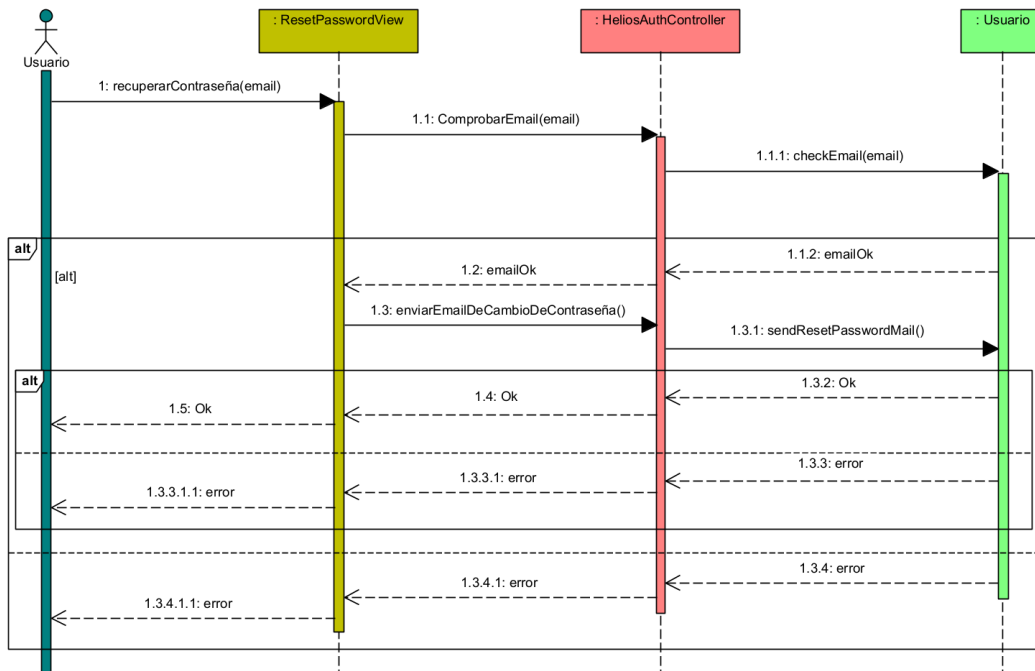


Ilustración 16: Diagrama de secuencia Recuperar contraseña

sd UC-0004 Iniciar sesión

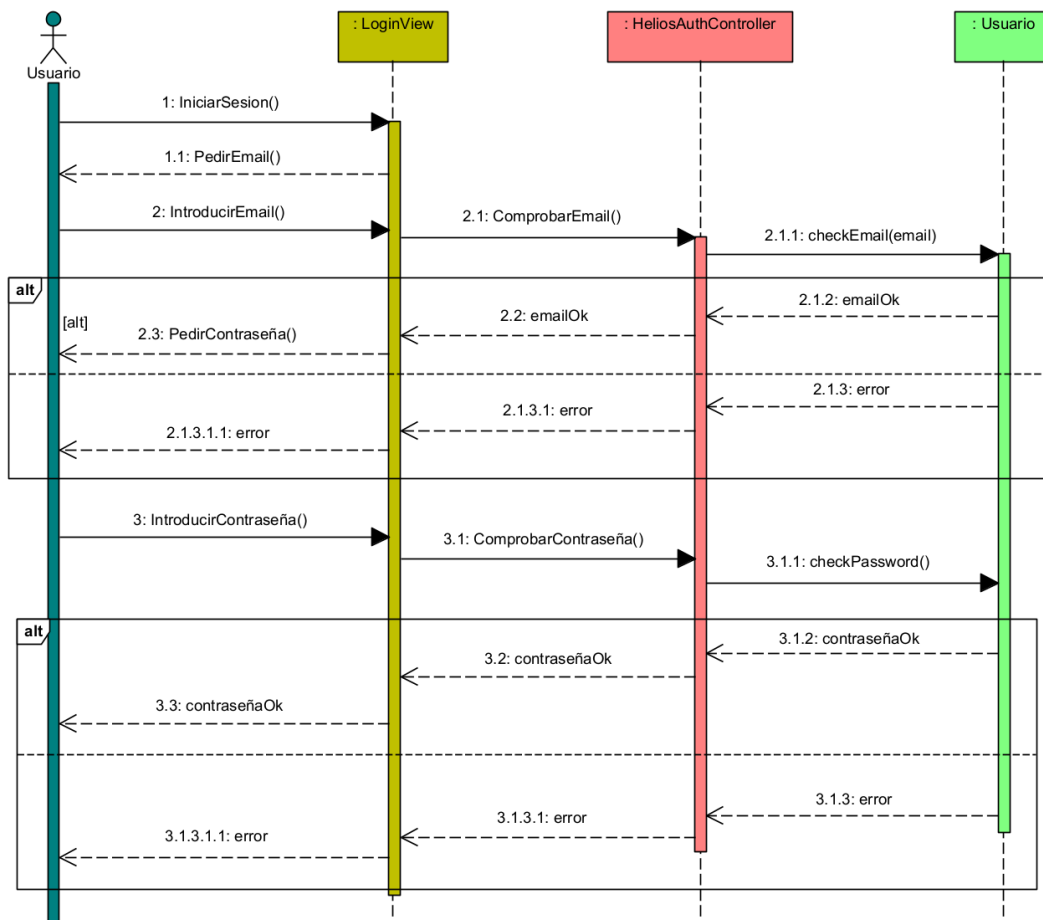


Ilustración 17: Diagrama de secuencia Iniciar Sesión

sd UC-0005 Cerrar Sesión

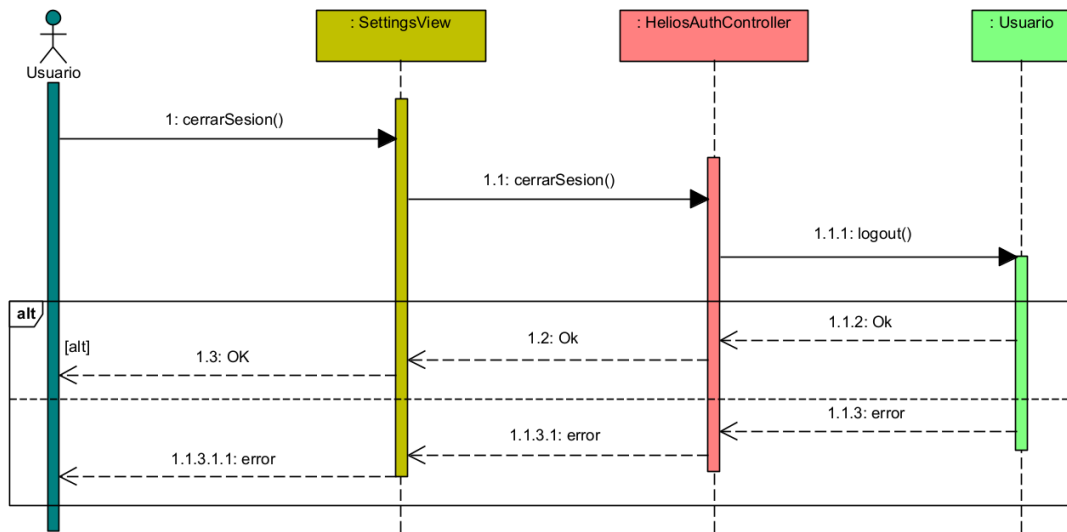


Ilustración 18: Diagrama de secuencia Cerrar Sesión

sd UC-0006 Modificar correo electrónico

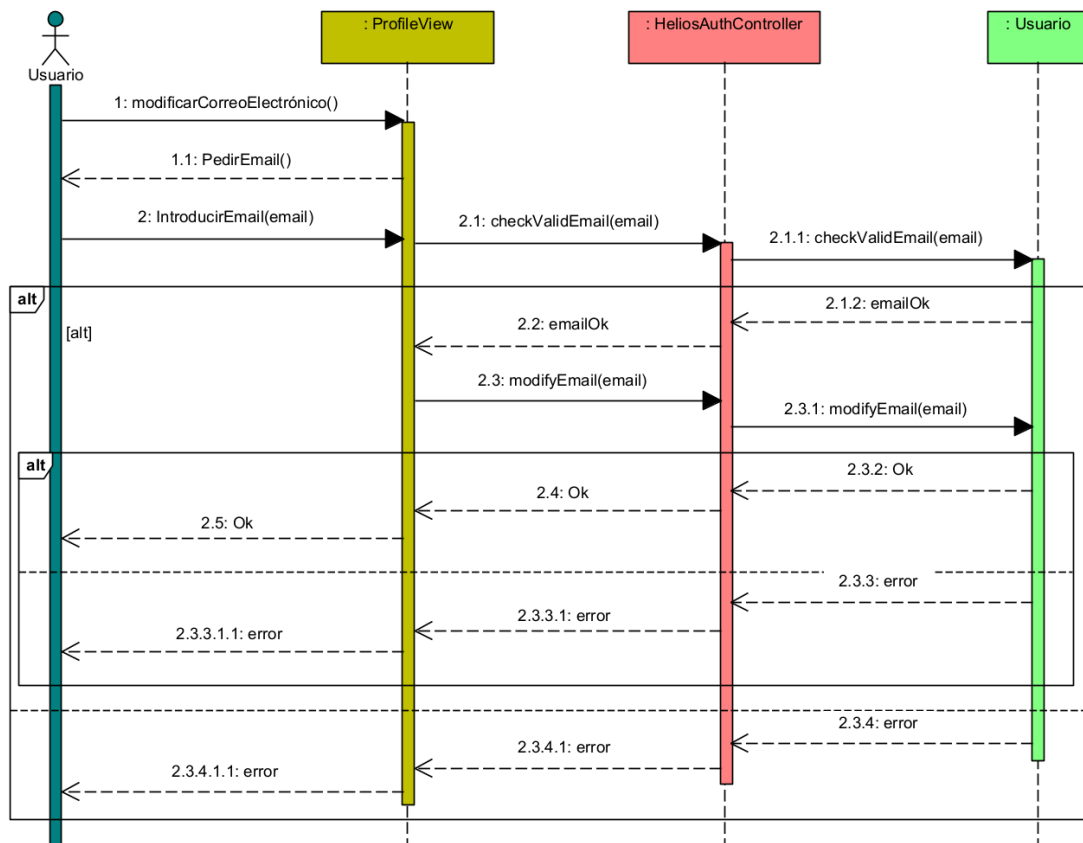


Ilustración 19: Diagrama de secuencia Modificar correo electrónico

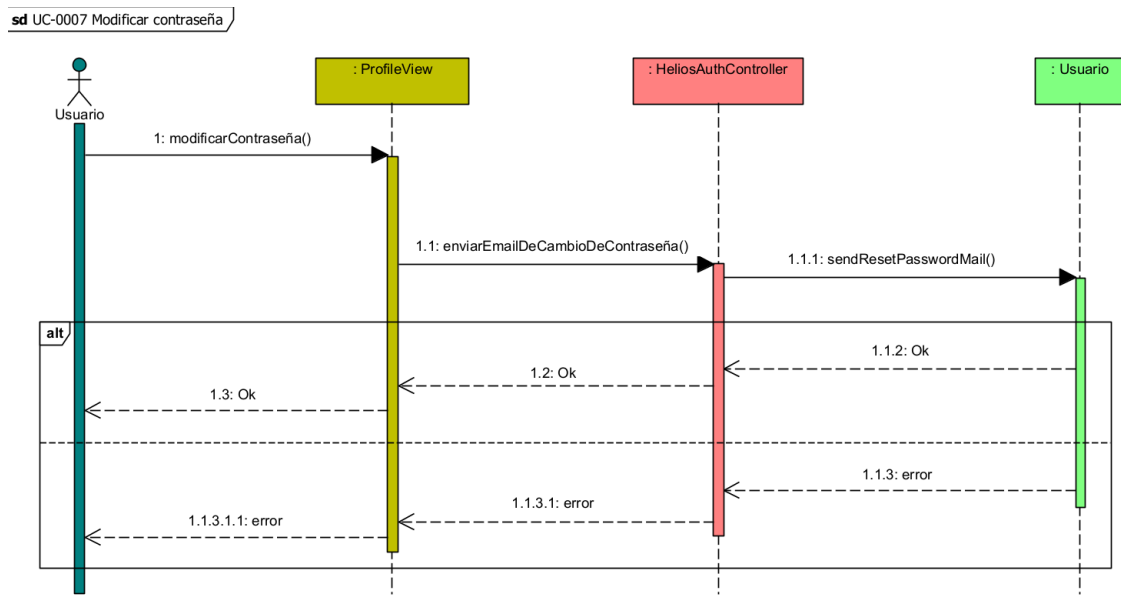


Ilustración 20: Diagrama de secuencia Modificar contraseña

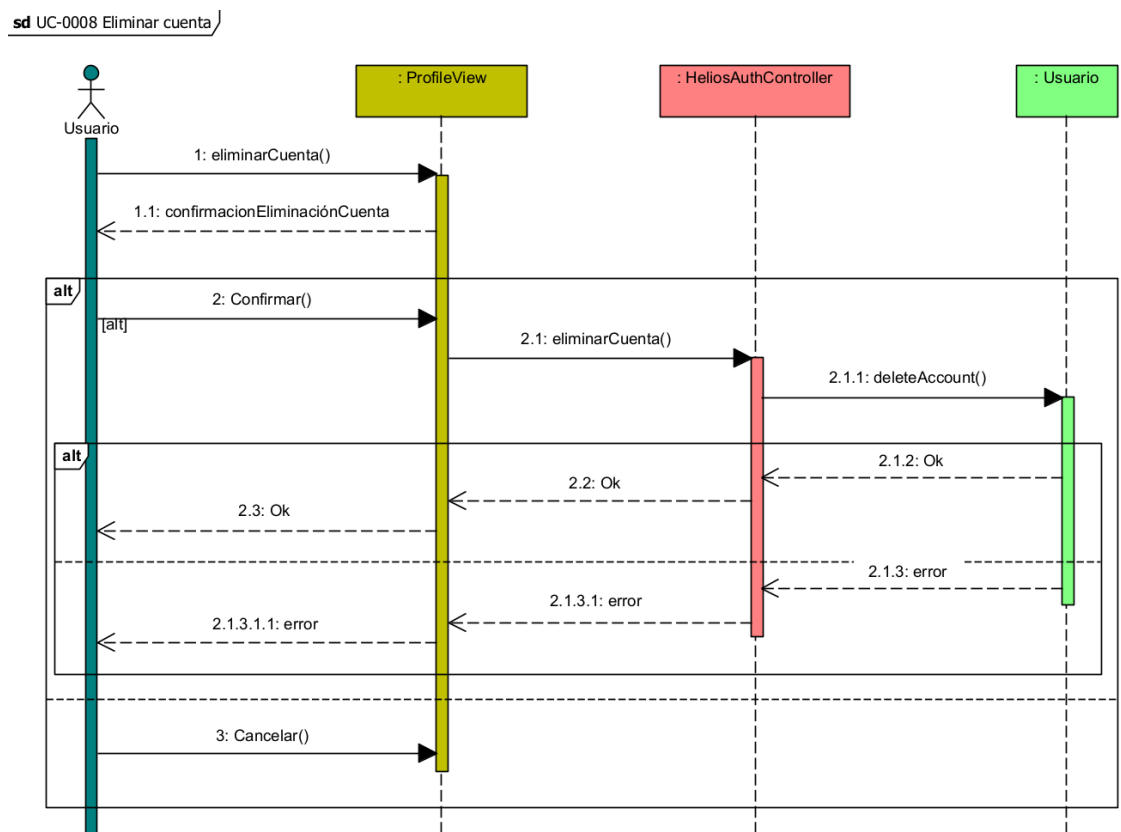


Ilustración 21: Diagrama de secuencia Eliminar cuenta

sd UC-0009 Ver Perfil

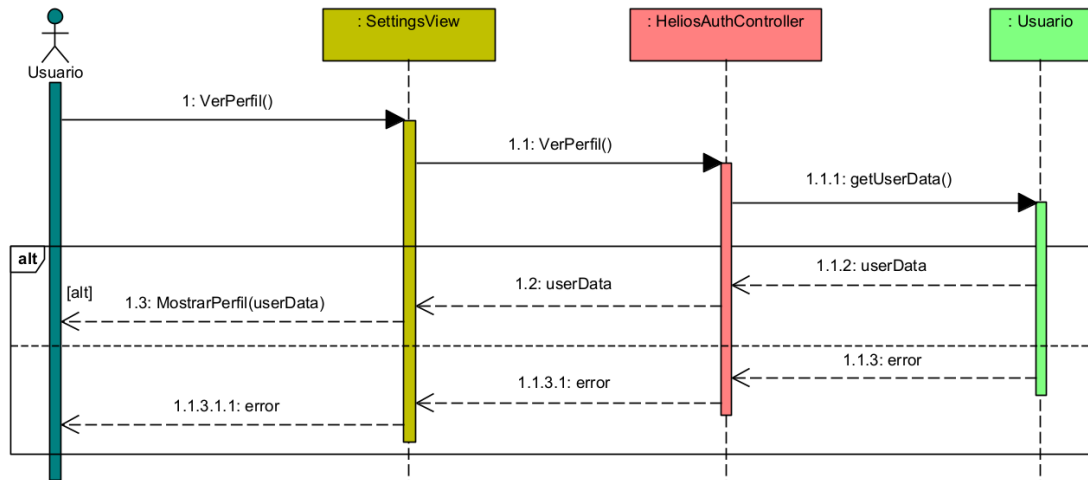


Ilustración 22: Diagrama de secuencia Ver perfil

sd UC-0010 Enviar correo electrónico de cambio de contraseña

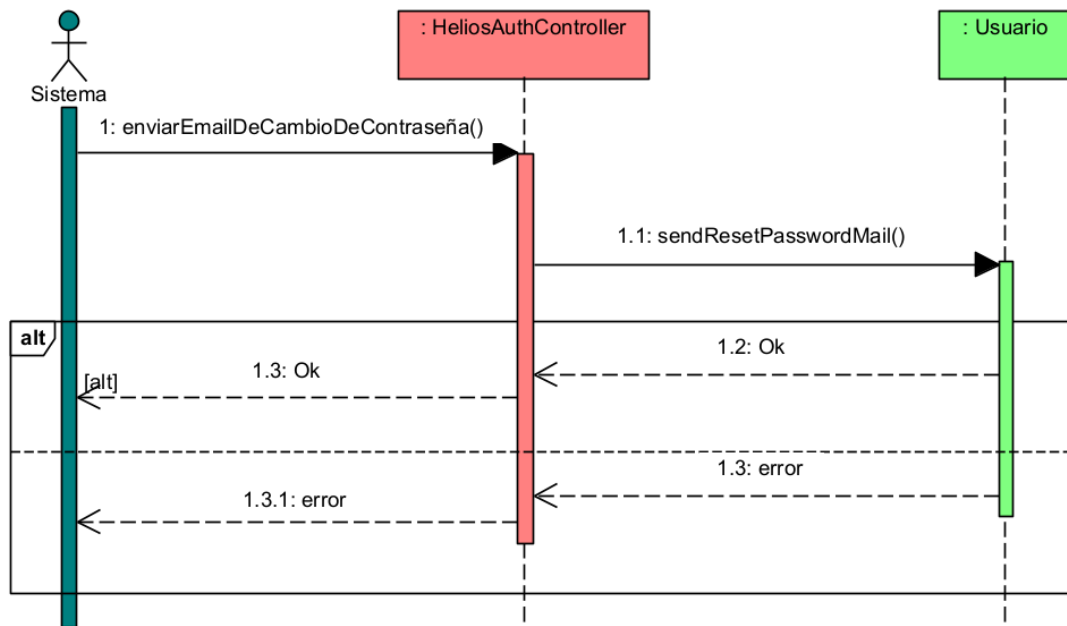


Ilustración 23: Diagrama de secuencia Enviar correo electrónico de cambio de contraseña

sd UC-0017 Obtener UID

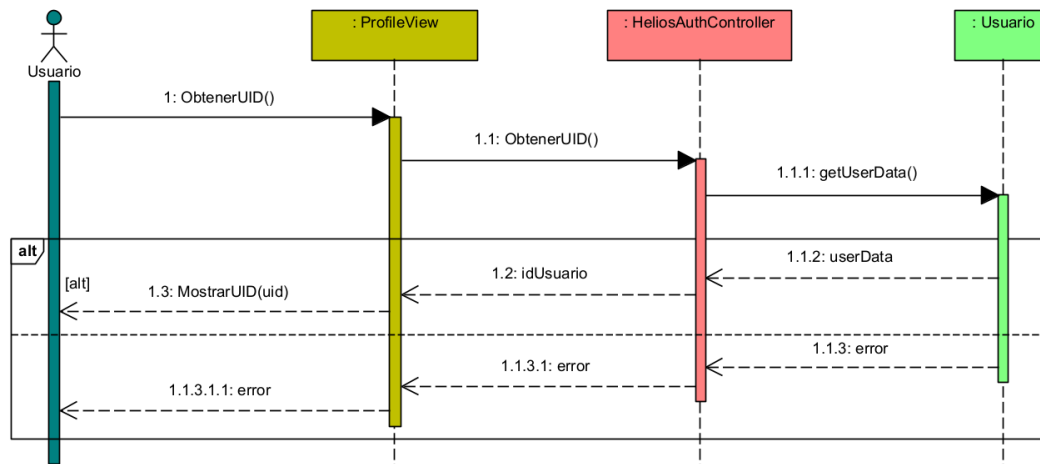


Ilustración 24: Diagrama de secuencia Obtener UID

3.3. Gestión de la monitorización de la instalación

sd UC-0018 Inicialización de la monitorización

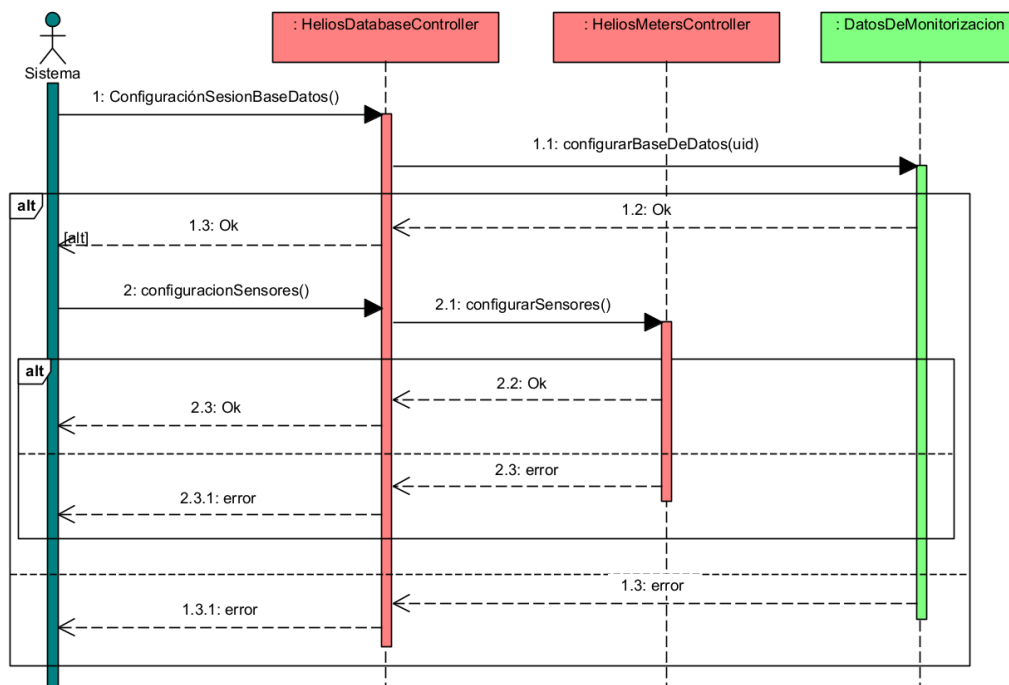


Ilustración 25: Diagrama de secuencia Inicialización de la monitorización

sd UC-0019 Leer información de los sensores

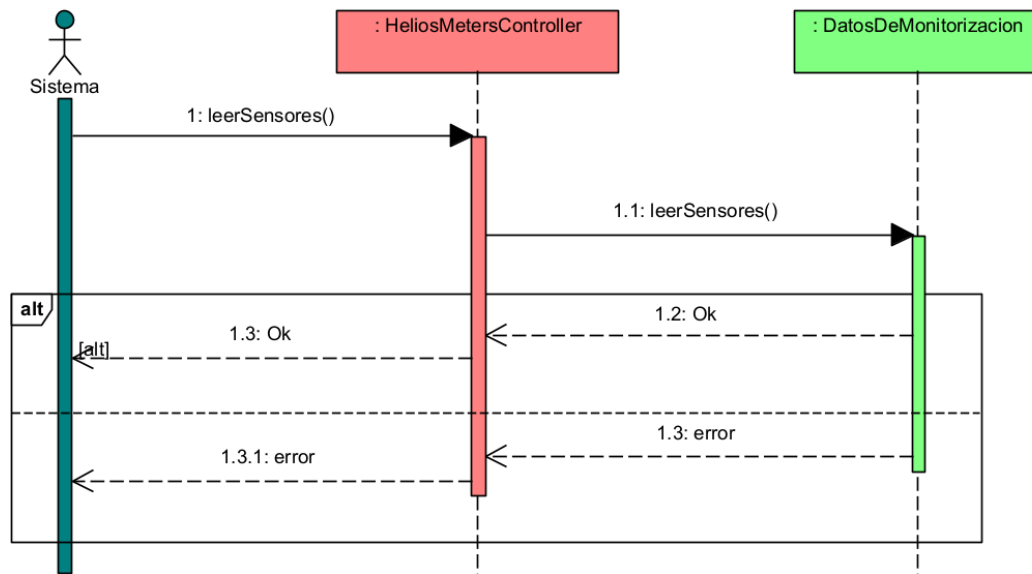


Ilustración 26: Diagrama de secuencia Leer información de los sensores

sd UC-0020 Enviar información a la base de datos

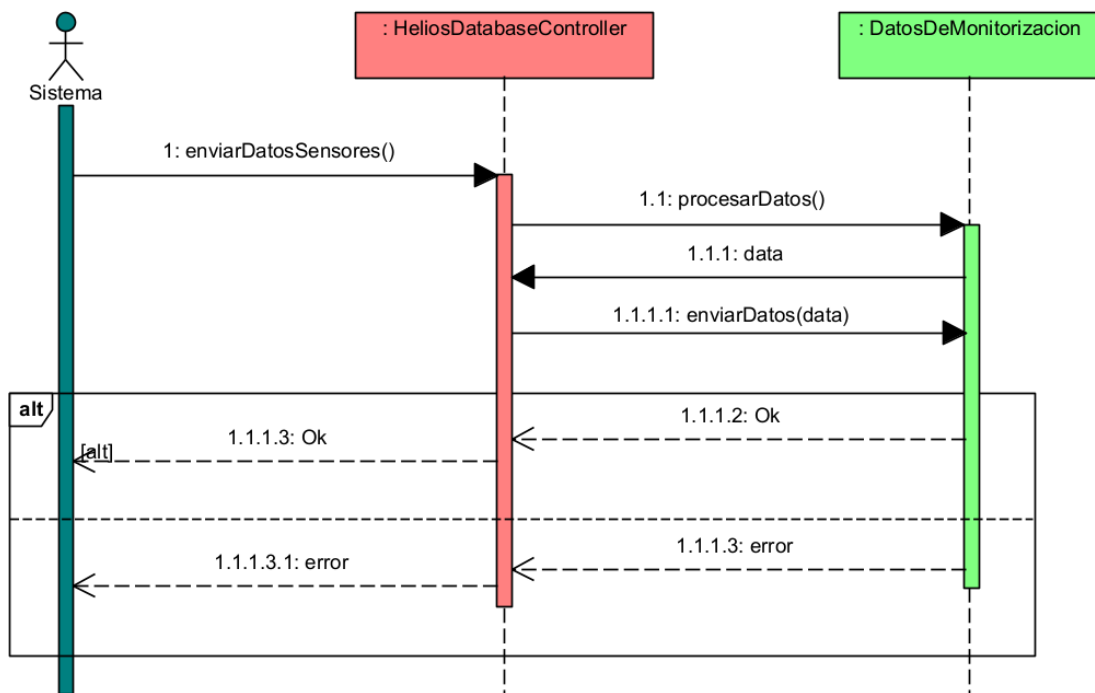


Ilustración 27: Diagrama de secuencia Enviar información a la base de datos

4. Modelo de despliegue

El modelo de despliegue muestra la distribución física de los distintos elementos del sistema. Cada nodo representado en la Ilustración 28 muestra un elemento hardware o software.

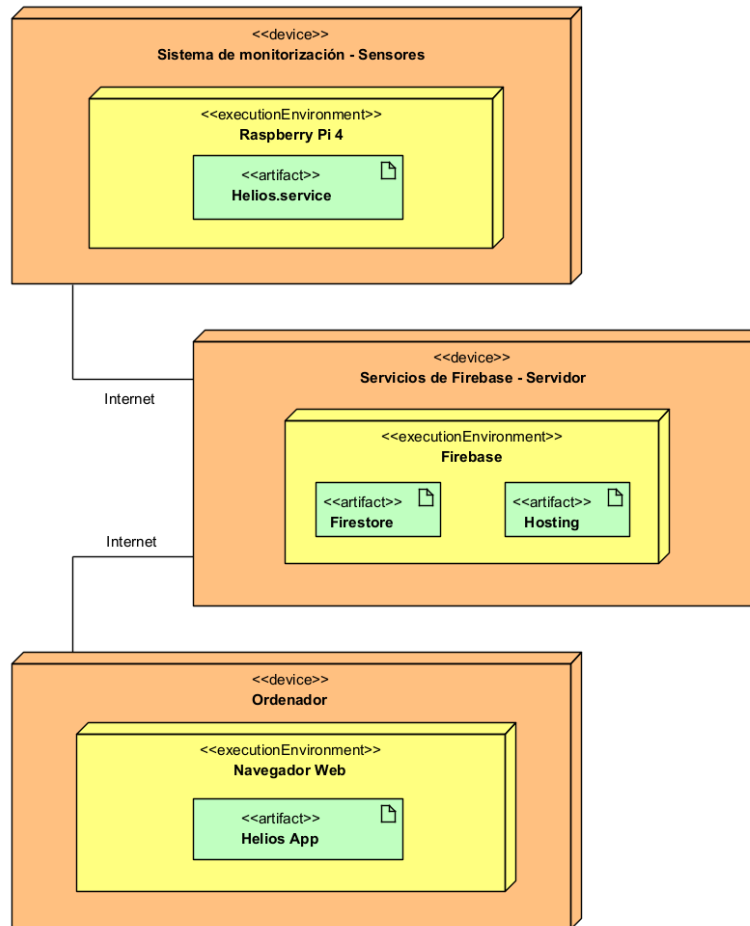


Ilustración 28: Modelo de despliegue del sistema

En el diagrama de la Ilustración 28 se aprecian nodos:

- **Sistema de monitorización - Sensores:** Es el conjunto de sensores y su controlador (Raspberry Pi 4), que realizan la monitorización de los datos de la instalación y se encarga de subirlos a la base de datos Firestore.
- **Servicios de Firebase - Servidor:** Este nodo refleja los servidores de Google que alojan nuestra base de datos y la página web y se encargan de realizar las comunicaciones de los datos.
- **Ordenador:** Es el dispositivo final del usuario, donde se visualizará la aplicación web. Puede ser cualquier tipo de ordenador, siempre y cuando disponga de un navegador web, que es el requisito mínimo para ejecutar la aplicación.

Aplicación para la gestión de una instalación fotovoltaica

Anexo V – Manual del programador



VNiVERSIDAD
D SALAMANCA

Trabajo de Fin de Grado

Grado de Ingeniería Informática

Septiembre 2021

Tutor:

Iván Álvarez Navia

Alumno:

Miguel Sánchez González

Tabla de contenidos

1. Introducción	6
2. Carpeta Service.....	7
3. Carpeta Arduino	8
4. Carpeta Python.....	9
5. Carpeta web	10
6. Documentación	11

Tabla de ilustraciones

Ilustración 1: estructura general del código de la aplicación.....	6
Ilustración 2: Servicio Linux	7
Ilustración 3: código Arduino	8
Ilustración 4: Módulo heliosDatabase.....	9
Ilustración 5: Módulo heliosMeters	9

1. Introducción

En este anexo se va a hablar de la estructura del código de la aplicación y la manera de desplegar el proyecto, que podemos encontrar en la carpeta de código fuente, adjunta con este anexo, con el propósito de que las personas encargadas del futuro mantenimiento de la aplicación puedan realizar esta tarea de una forma más cómoda y sencilla.

Las carpetas en las que se divide el espacio de trabajo son las siguientes:

- **Service:** contiene el código del servicio Linux para lanzar la monitorización automáticamente.
- **Arduino:** contiene el código para el microcontrolador de Arduino que realiza la medición de potencia en corriente alterna.
- **Python:** aquí se encuentra el código del programa que controla la monitorización del sistema y envía los datos a la base de datos.
- **Web:** aquí se encuentra el código de la aplicación web.

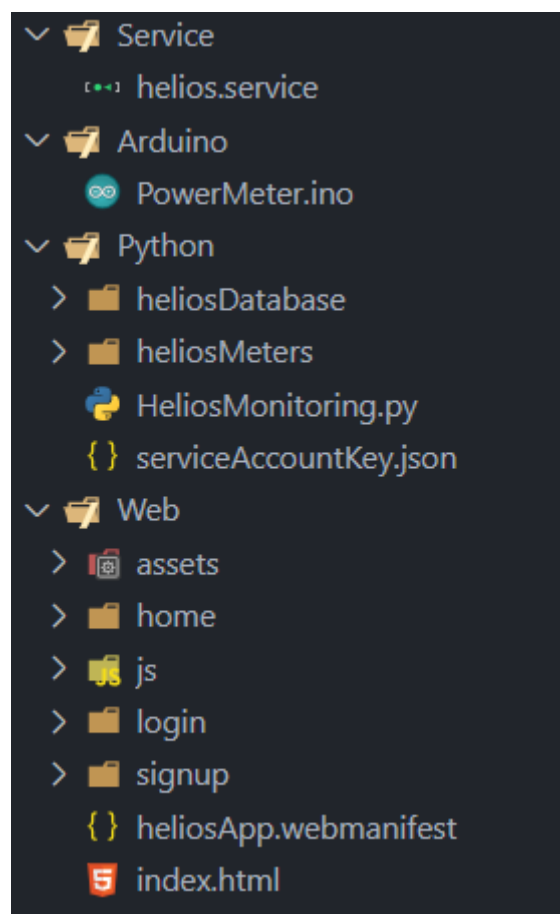


Ilustración 1: estructura general del código de la aplicación

2. Carpeta Service

El código de esta carpeta deberá instalarse como servicio Linux en la Raspberry Pi, para que cada vez que se reinicie el ordenador, o aparezca algún fallo se reinicie el programa de la monitorización.

```
1  [Unit]
2  Description=Helios monitoring service.
3  After=network.target
4  StartLimitIntervalSec=0
5
6  [Service]
7  Type=simple
8  Restart=always
9  RestartSec=20
10 User=pi
11 ExecStart=/usr/bin/env python3 /usr/bin/HeliosMonitoring.py
12
13 [Install]
14 WantedBy=multi-user.target
```

Ilustración 2: Servicio Linux

Para habilitar el servicio Linux deberemos copiar el archivo “helios.service” en la ruta “/lib/systemd/system” como administrador y ejecutar el comando “systemctl enable helios.service” como administrador.

3. Carpeta Arduino

En esta carpeta se encuentra el código del microcontrolador de Arduino que deberá cargarse en un Arduino Nano Every para poder ejecutarse, usando cualquier IDE de Arduino.

La función principal de este programa es el lazo de lectura de los sensores de corriente y voltaje y su código es el siguiente:

```
1 void loop() {
2     /* Leemos los valores del sensor de corriente y el medidor de voltaje. */
3     voltajeADC = convertValueVoltage(analogRead(PIN_VOLTAGE));
4     currentADC = convertValueCurrent(analogRead(PIN_CURRENT));
5     power += voltajeADC * currentADC;
6
7     /* Incrementamos el contador de lecturas. */
8     reads += 1;
9
10    /*
11     * Si ha pasado el tiempo de lectura calculamos la potencia
12     * y establecemos una nueva marca para el contador de tiempo.
13     */
14    if ((millis() - time) > READ_TIME) {
15        lastPower = power / reads;
16        power = 0;
17        reads = 0;
18
19        /* Actualizamos el contador de tiempo. */
20        time = millis();
21    }
22 }
```

Ilustración 3: código Arduino

4. Carpeta Python

En esta carpeta tenemos el código principal del programa de monitorización y el código de los módulos auxiliares.

Para ejecutar el código de esta carpeta deberemos instalar previamente los paquetes de Python “firebase-admin”, “smbus2”, “busio” y “adafruit-circuitpython-ads1x15”, y cargar el código de Python en la carpeta de la Raspberry Pi “/usr/share/helios”.

Los módulos auxiliares de la carpeta del programa de monitorización son:

- **heliosDatabase:** contiene una serie de clases para tratar y subir los datos a la base de datos de Firebase.
- **heliosMonitoring:** las clases de este módulo tienen la función de leer los diferentes datos de los sensores y del Arduino.

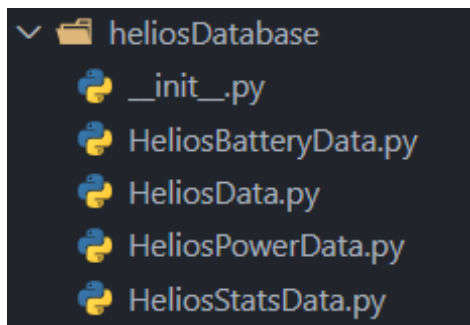


Ilustración 4: Módulo heliosDatabase

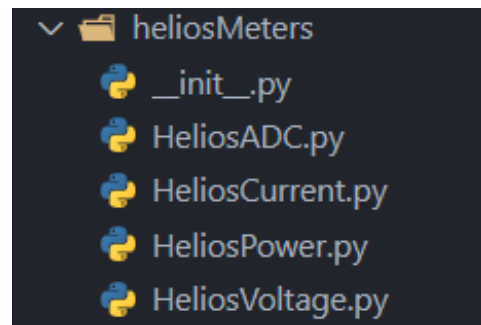


Ilustración 5: Módulo heliosMeters

5. Carpeta web

Esta es la carpeta principal de la aplicación web y del proyecto, donde su código será el encargado de recoger los datos y mostrarlos al usuario.

Esta carpeta contiene el código de las páginas web que se muestran en la aplicación web y una carpeta general llamada “assets” donde se encuentran los recursos generales de la aplicación y los scripts que usaremos para todas las páginas web.

Las carpetas principales de páginas web son:

- **login:** esta carpeta contiene el código de las páginas de inicio de sesión, contraseña de inicio de sesión y recuperar contraseña.
- **signup:** en esta carpeta podemos ver las páginas de registro, contraseña de registro y verificación de cuenta.
- **home:** es la última de las carpetas de este apartado, contiene la plantilla general de la aplicación, aquí se encuentran la página principal de la ventana de monitorización y la página de visualización del perfil de la cuenta.

Los scripts principales de la aplicación se encuentran dentro de la carpeta “js” en la carpeta “assets”, y son 3:

- **HeliosData.js:** este script contiene una clase con funciones para gestionar los datos locales y de sesión de la aplicación, también contiene funciones para pasar los datos de las gráficas a formato csv para que el usuario pueda usarlos.
- **HeliosAuth.js:** en esta clase tendremos funciones para gestionar el inicio y cierre de sesión, controlar las sesiones activas, modificar los datos de la cuenta y eliminar la cuenta entre otros.
- **HeliosApp.js:** es el script principal de la aplicación donde encontraremos las funciones para la gestión de los datos de la monitorización y la representación de las gráficas, el uso del widget del calendario, el control de las ventanas modales y las funciones de recogida y adecuación de datos de la base de datos de Firestore.

Para realizar el despliegue de la aplicación web será necesario tener instalado [Node.js](https://nodejs.org/), para poder descargar el SDK para JavaScript de Firebase, se instala con el comando:

```
npm install -g firebase-tools
```

Una vez instalado en la misma carpeta donde se encuentra la aplicación web podremos ejecutar el comando

```
firebase serve -o 0.0.0.0
```

para crear un servidor local al que podremos acceder con nuestra ip o ejecutar el comando

```
firebase deploy
```

para subir el código de la página web al servicio de alojamiento de Firebase, Firebase Hosting.

6. Documentación

Se ha documentado el código de los archivos Python y JavaScript de la aplicación, usando las herramientas Pydoc y JsDoc.

Esta documentación se puede encontrar dentro de la carpeta código fuente, en la carpeta “0-Docs” y es accesible desde el navegador, esta documentación se ha elaborado a partir de los comentarios del código.

Aplicación para la gestión de una instalación fotovoltaica

Anexo VI – Manual de usuario



VNiVERSIDAD
D SALAMANCA

Trabajo de Fin de Grado

Grado de Ingeniería Informática

Septiembre 2021

Tutor:

Iván Álvarez Navia

Alumno:

Miguel Sánchez González

Tabla de contenidos

1. Introducción	6
2. Acceso a la aplicación.....	7
2.1. Iniciar sesión	8
2.2. Registrarse	9
2.3. Verificar cuenta.....	10
2.4. Recuperar contraseña.....	12
3. Aplicación principal	14
3.1. Cerrar sesión	15
3.2. Mostrar perfil.....	15
3.3. Elementos principales.....	16
4. Histórico de datos	17

Tabla de ilustraciones

Ilustración 1: Ventana de acceso a la aplicación.....	7
Ilustración 2: Pantalla de correo de inicio de sesión.....	8
Ilustración 3: Pantalla de contraseña de inicio de sesión.....	8
Ilustración 4: Pantalla de correo electrónico de registro.....	9
Ilustración 5: Pantalla de contraseña del registro.....	9
Ilustración 6: Pantalla de verificación de cuenta.....	10
Ilustración 7: Mensaje de verificación de correo electrónico.....	10
Ilustración 8: Correo de verificación de cuenta.....	11
Ilustración 9: Mensaje de cuenta verificada.....	11
Ilustración 10: pantalla de recuperación de contraseña.....	12
Ilustración 11: Mensaje de recuperación de contraseña.....	12
Ilustración 12: correo electrónico de recuperación de contraseña.....	13
Ilustración 13: Dialogo de cambio de contraseña.....	13
Ilustración 14: Pantalla principal de la monitorización.....	14
Ilustración 15: Ventana emergente para ver perfil y cerrar sesión.....	15
Ilustración 16: Pantalla de perfil.....	15
Ilustración 17: Gráficos de tiempo real.....	16
Ilustración 18: Consumo y producción máximos.....	16
Ilustración 19: Resumen últimos 7 días.....	16
Ilustración 20: Gráfico de histórico de datos.....	17
Ilustración 21: Calendario de selección de fecha.....	17

1. Introducción

En este anexo se va a explicar de una forma más visual el funcionamiento de la aplicación para que aquellos usuarios poco experimentados puedan hacer un buen uso de ella.

2. Acceso a la aplicación

Para entrar en la aplicación web del proyecto, hay que poner la URL <https://helios-msg.web.app> en el navegador, con eso accederemos a la pantalla principal de la aplicación web, donde se nos mostraran dos botones, uno para iniciar sesión y otro para crearse una cuenta en la aplicación.

Hay un usuario de prueba de la aplicación cuyos datos de inicio de sesión son:

Correo: miguel.sanchezgon@gmail.com

Contraseña: 12345678

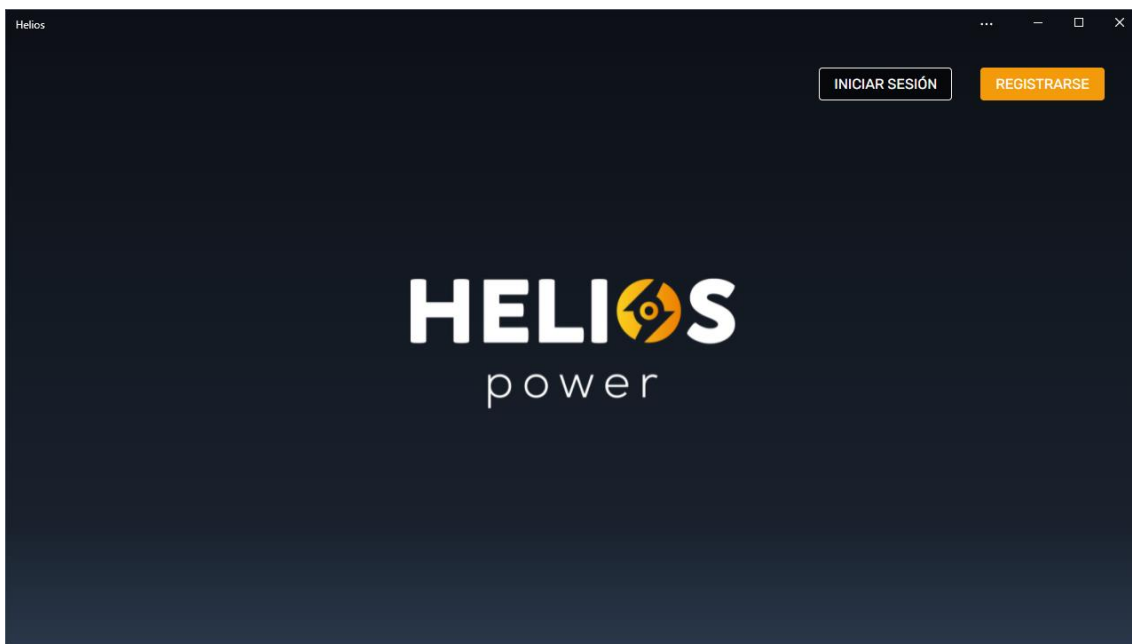


Ilustración 1: Ventana de acceso a la aplicación.

2.1. Iniciar sesión

Al pulsar el botón “Iniciar sesión” de la pantalla principal, nos llevará a la ventana de inicio de sesión donde nos pedirá introducir nuestro correo electrónico y una vez hecho deberemos darle al botón “Continuar”. Además, tenemos una opción para registrarnos en la aplicación pulsando el enlace llamado “Registrarse”.

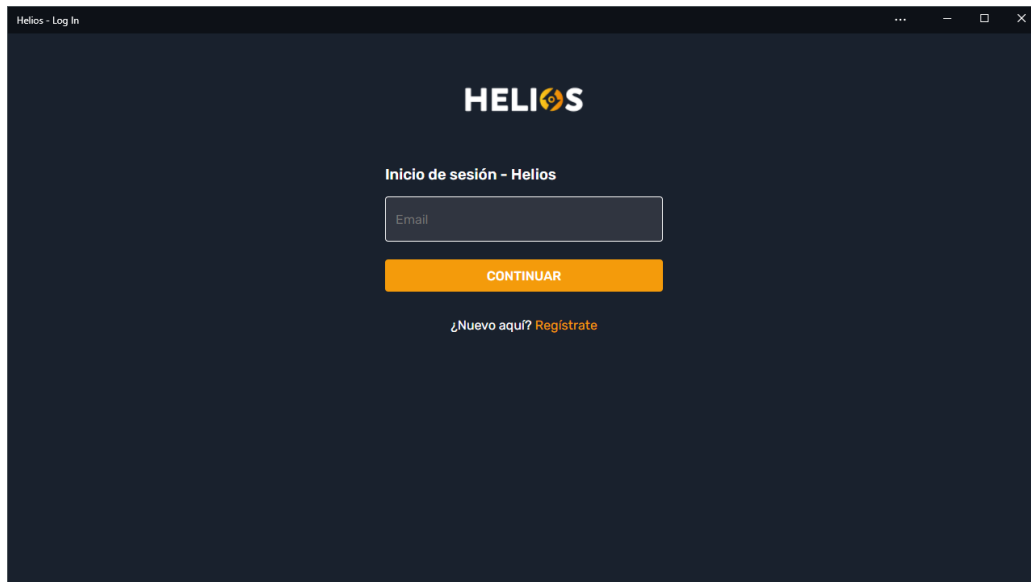


Ilustración 2: Pantalla de correo de inicio de sesión

Una vez pulsado el botón “Continuar”, aparecerá un campo para introducir nuestra contraseña, la pondremos y en caso de que sea correcta pulsaremos el botón “Iniciar sesión” y nos llevará a la página de inicio de nuestra monitorización, en el apartado de la contraseña también tendremos un enlace al que podremos acceder en caso de que hayamos olvidado nuestra contraseña, el nombre de “¿Olvidaste tú contraseña?”.

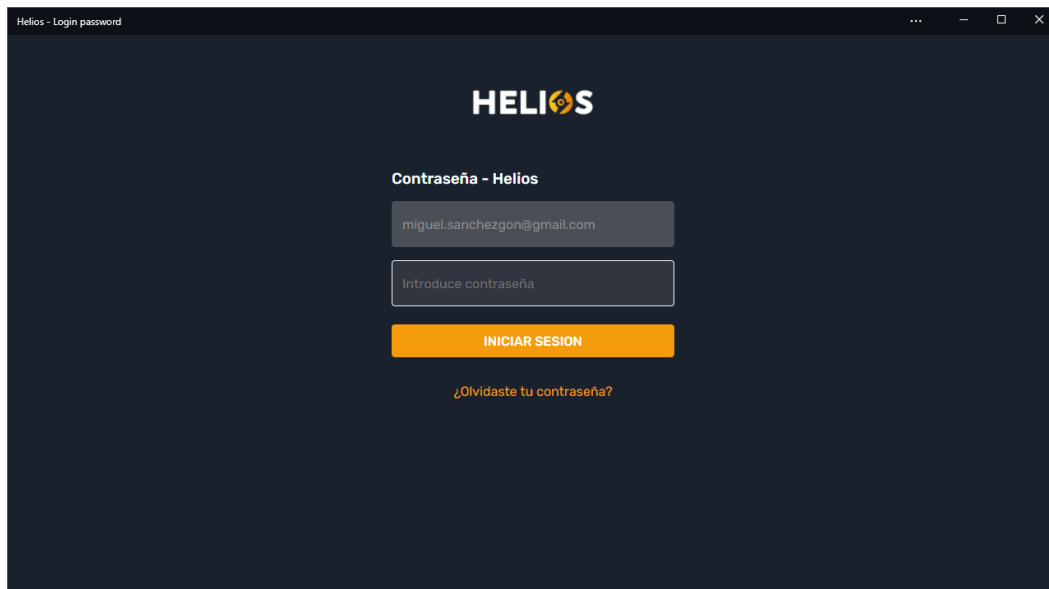


Ilustración 3: Pantalla de contraseña de inicio de sesión

2.2. Registrarse

Para crear una cuenta en la aplicación, una vez pulsado el botón “registrarse” de la pantalla principal, accederemos a la ventana de registro de la aplicación.

En ella se nos pedirá que introduzcamos nuestro email, para nuestra cuenta, una vez hecho eso pulsaremos el botón “Continuar”.

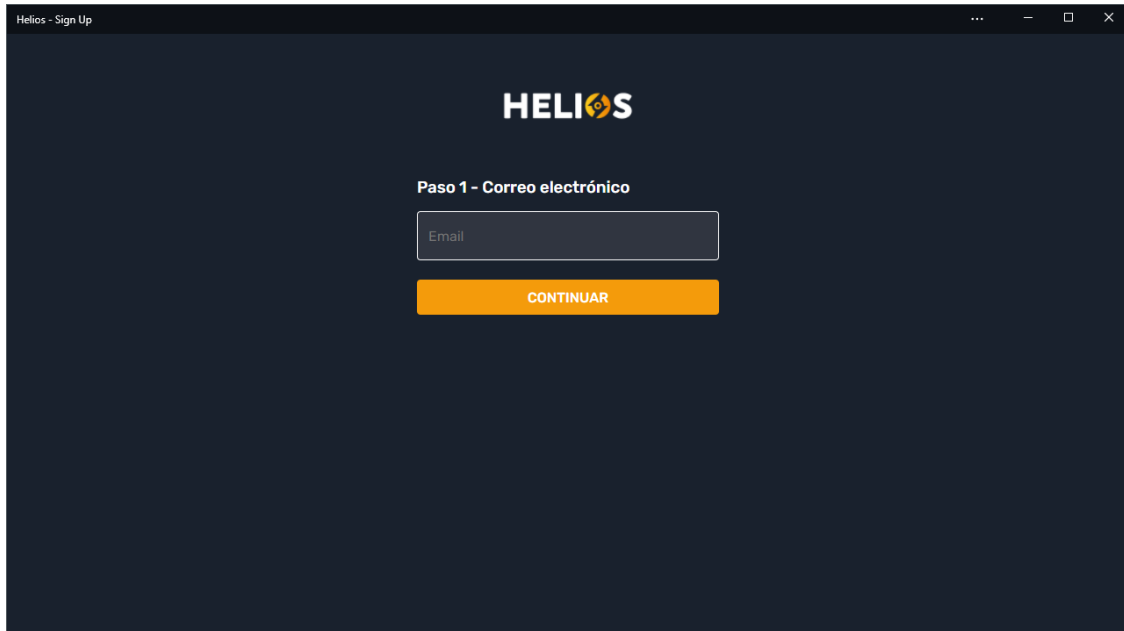
The screenshot shows a dark-themed web browser window titled "Helios - Sign Up". At the top center is the "HELIOS" logo. Below it, the text "Paso 1 - Correo electrónico" is displayed. There is a text input field labeled "Email" and a prominent orange button labeled "CONTINUAR".

Ilustración 4: Pantalla de correo electrónico de registro

El paso dos del registro será introducir una contraseña que usaremos para iniciar sesión en nuestra aplicación, una vez introducida pulsaremos el botón “Registrarse”.

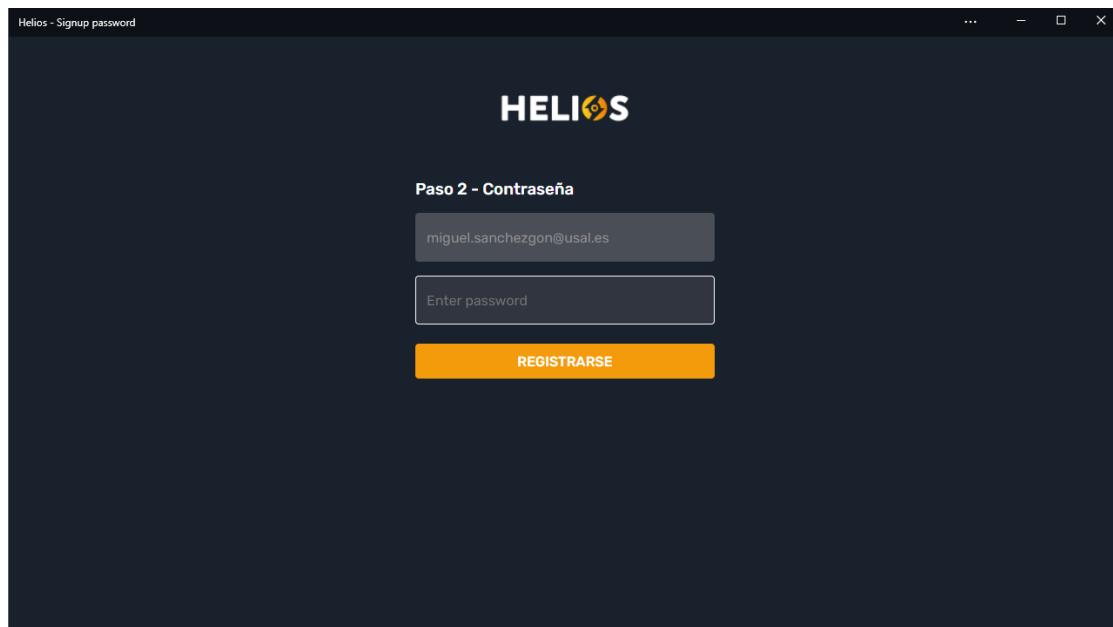
The screenshot shows a dark-themed web browser window titled "Helios - Signup password". At the top center is the "HELIOS" logo. Below it, the text "Paso 2 - Contraseña" is displayed. There are two text input fields: the first contains the email address "miguel.sanchezgon@usal.es" and the second is labeled "Enter password". Below the fields is a prominent orange button labeled "REGISTRARSE".

Ilustración 5: Pantalla de contraseña del registro

2.3. Verificar cuenta

Para verificar nuestra nueva cuenta, una vez registrados nos encontraremos en la pantalla de verificación, deberemos pulsar el botón “Enviar email de notificación”.

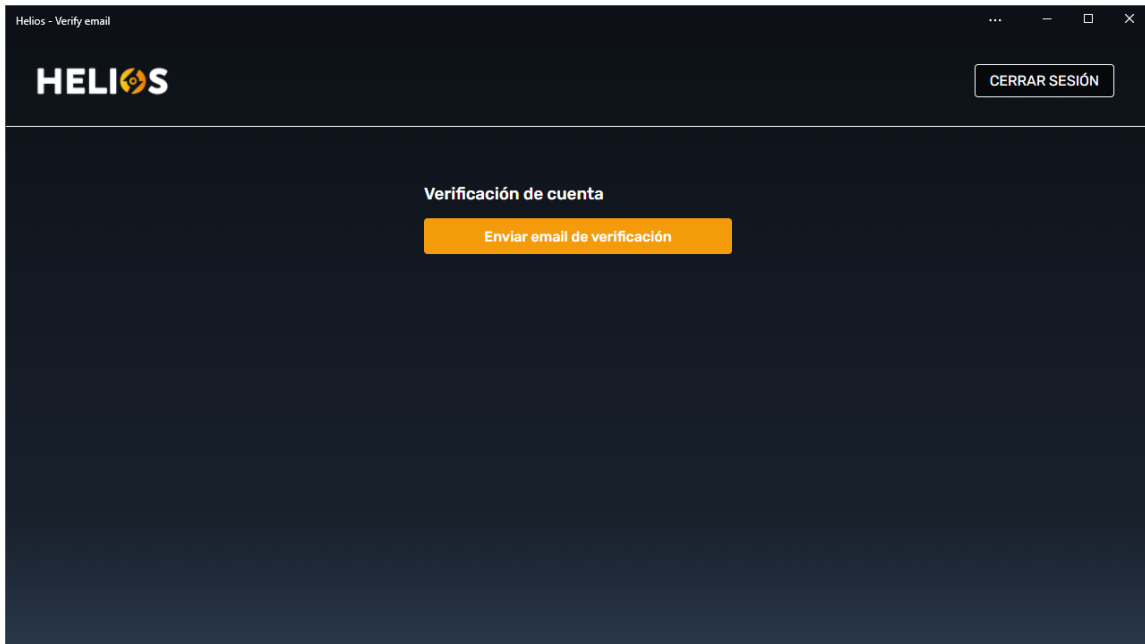


Ilustración 6: Pantalla de verificación de cuenta

Cuando hayamos pulsado el botón nos debería aparecer una ventana emergente que nos dirá que comprobemos el correo electrónico.

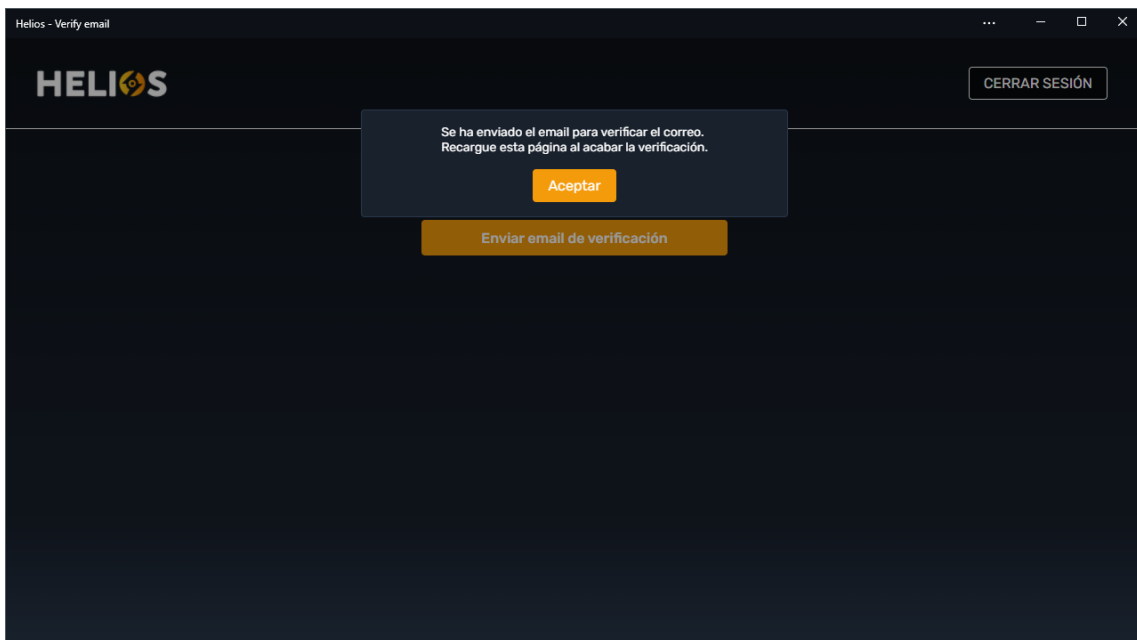


Ilustración 7: Mensaje de verificación de correo electrónico

En nuestro correo electrónico encontraremos un mensaje de la aplicación donde podremos ver que nos dice que hagamos clic en un enlace para verificar nuestra cuenta.

Verifica tu dirección de correo electrónico de Helios Power



noreply@helios-msg.firebaseio.com

para mí ▾

Hola:

Haz clic en este enlace para verificar tu dirección de correo electrónico.

https://helios-msg.firebaseio.com/_/auth/action?mode=verifyEmail&oobCode=hz6-kTFKyguU3kDc

Si no has emitido esta solicitud, ignora este mensaje.

Gracias,

El equipo de Helios Power

Ilustración 8: Correo de verificación de cuenta

Una vez echo clic en ese enlace aparecerá una ventana que nos dirá que hemos verificado nuestra cuenta satisfactoriamente.

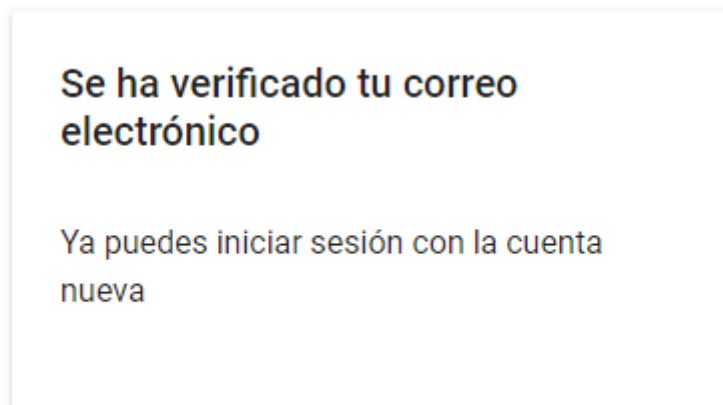


Ilustración 9: Mensaje de cuenta verificada

Y si recargamos la página de la aplicación entraremos a la ventana principal de nuestra monitorización.

2.4. Recuperar contraseña

Para recuperar nuestra contraseña deberemos pulsar el botón iniciar sesión e introducir nuestro correo electrónico asociado a nuestra cuenta de la aplicación y pulsaremos el botón “Continuar” y cuando aparezca el campo para introducir nuestra contraseña haremos clic en el enlace “¿Olvidaste tu contraseña?” que nos llevará a la ventana de restablecimiento de la contraseña.

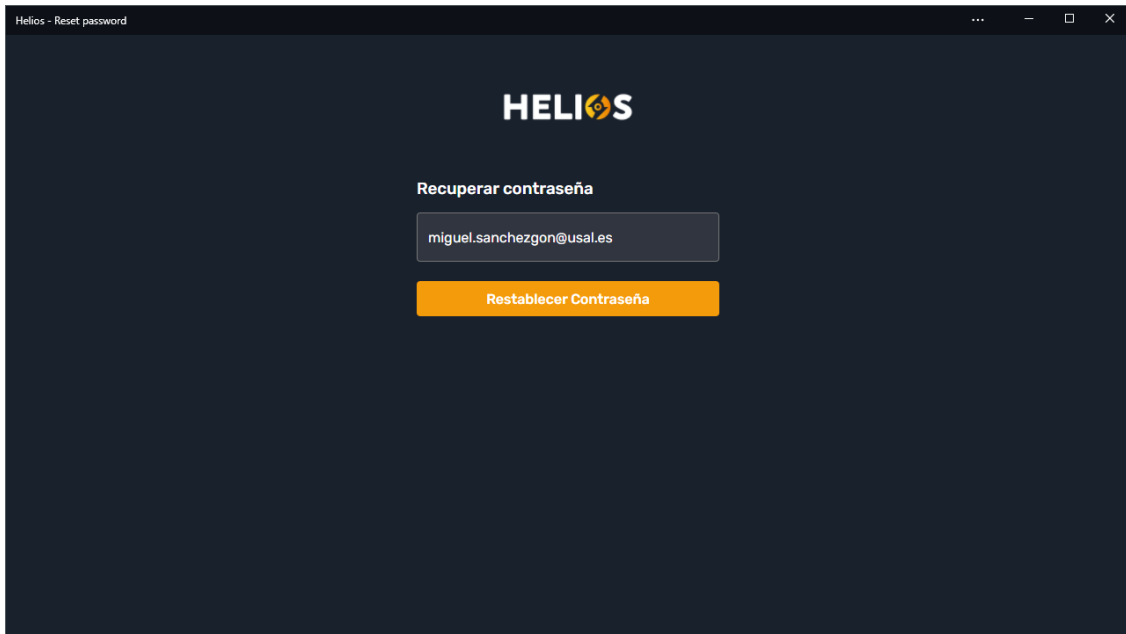


Ilustración 10: pantalla de recuperación de contraseña

Aquí podremos comprobar que el correo que hemos introducido es el que queremos, si es así pulsaremos el botón restablecer contraseña y nos aparecerá una ventana emergente comunicándonos que se ha enviado un email para restablecer la contraseña.

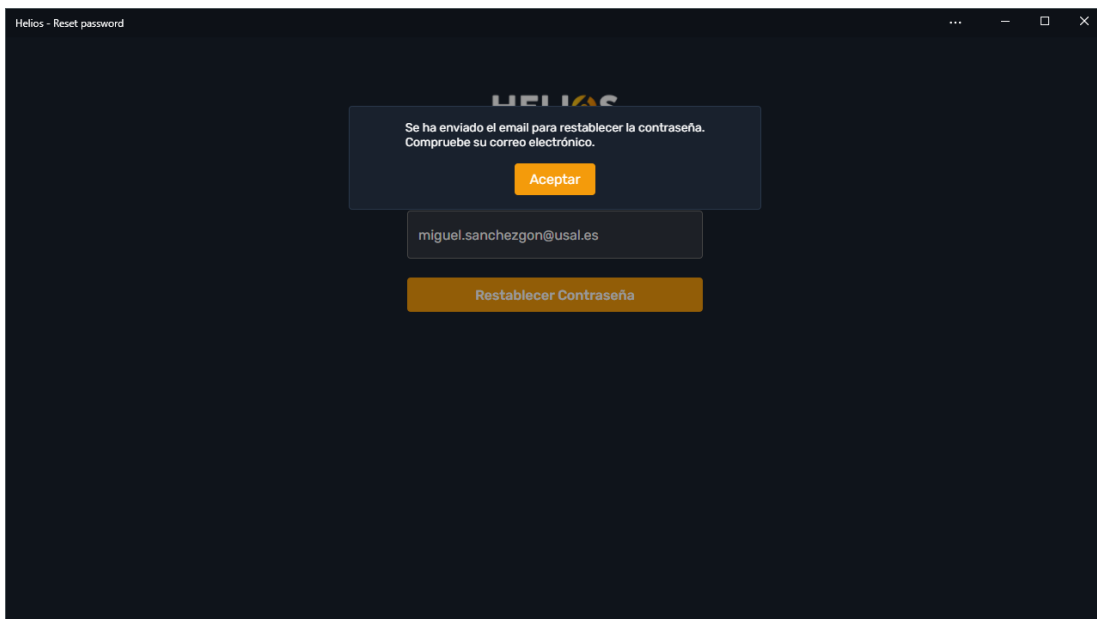


Ilustración 11: Mensaje de recuperación de contraseña

En ese email tendremos un enlace al igual que en el apartado de verificación de la cuenta, y si hacemos clic en el enlace nos llevará a una página para introducir nuestra nueva contraseña.

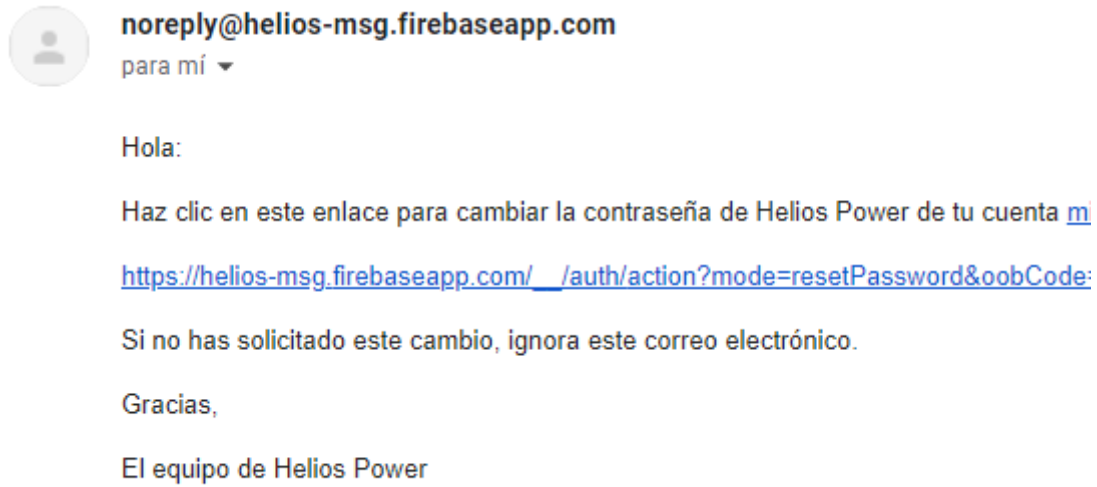


Ilustración 12: correo electrónico de recuperación de contraseña

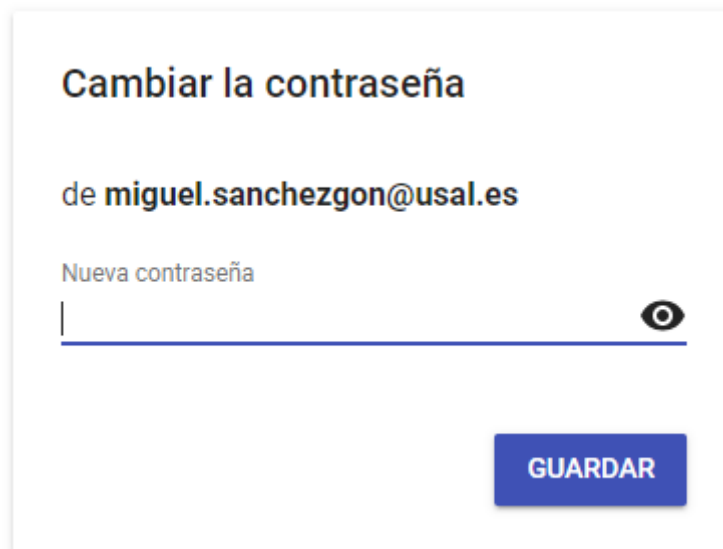
The image shows a white dialog box with a light grey border. The title is 'Cambiar la contraseña' in bold black text. Below the title, it says 'de miguel.sanchezgon@usal.es'. There is a text input field labeled 'Nueva contraseña' with a blue underline and a blue eye icon to its right. At the bottom right of the dialog box is a blue button with the white text 'GUARDAR'.

Ilustración 13: Dialogo de cambio de contraseña

3. Aplicación principal

Una vez iniciado sesión con nuestra cuenta se nos mostrará la ventana principal de la monitorización desde la que podemos realizar todas las acciones de la aplicación.



Ilustración 14: Pantalla principal de la monitorización.

3.1. Cerrar sesión

Si pulsamos el botón “mi perfil” de la ventana principal de la monitorización, se mostrarán dos botones el de acceso a “mi perfil” y el de “cerrar sesión”, si pulsamos el botón de “cerrar sesión” acabará nuestra sesión en la cuenta.



Ilustración 15: Ventana emergente para ver perfil y cerrar sesión

3.2. Mostrar perfil

Si pulsamos el botón “mi perfil” de la ventana principal de la monitorización, se mostrarán dos botones el de acceso a “mi perfil” y el de “cerrar sesión”, si pulsamos el botón de “mi perfil” se mostrará la ventana de visualización del perfil.

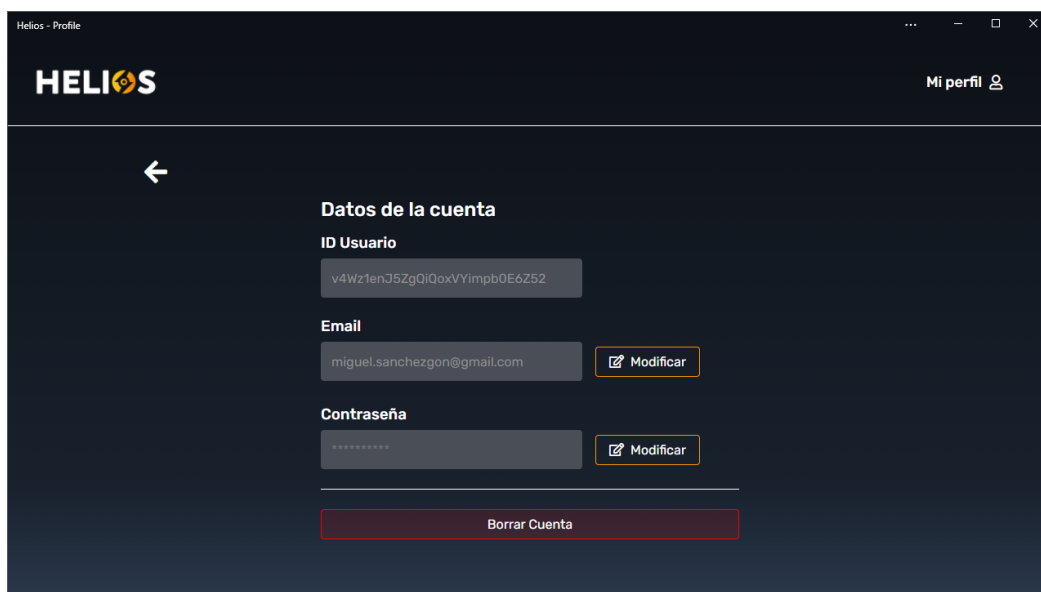


Ilustración 16: Pantalla de perfil

3.3. Elementos principales

En la ventana principal tenemos 3 gráficas en tiempo real que nos muestran los datos del estado de la batería, el consumo y la producción de energía, si hacemos clic en ellas nos llevará al histórico de datos de ese gráfico.

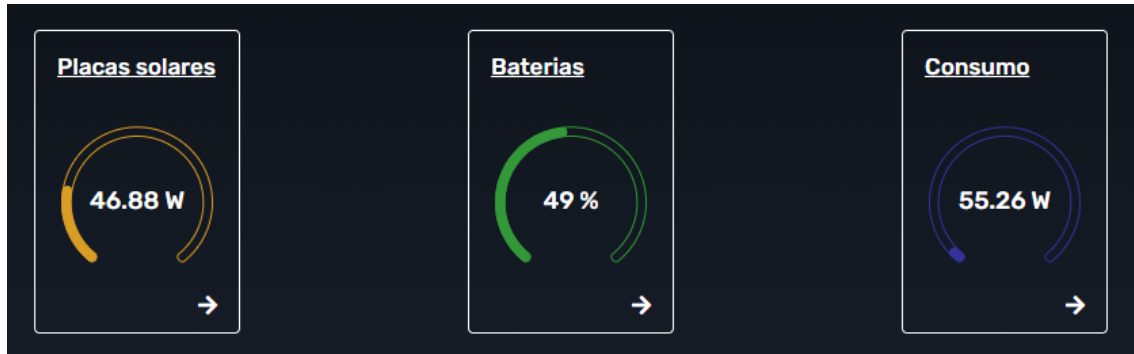


Ilustración 17: Gráficos de tiempo real

También está el estudio de consumo y producción que nos muestra las horas de máxima producción y máximo consumo de los datos registrados.

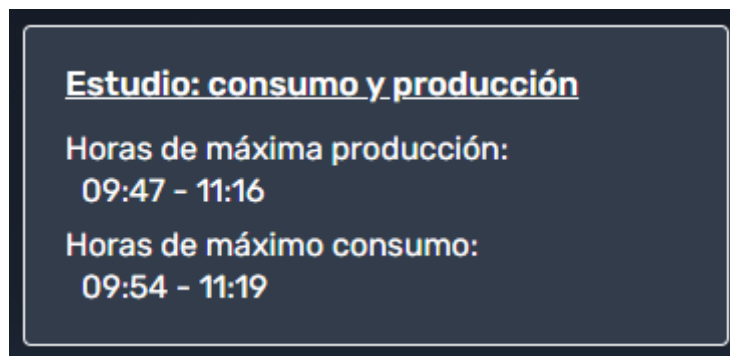


Ilustración 18: Consumo y producción máximos

Por último, tenemos el resumen de los últimos 7 días, en el que aparecen los datos de la potencia generada y consumida en los últimos 7 días de datos.

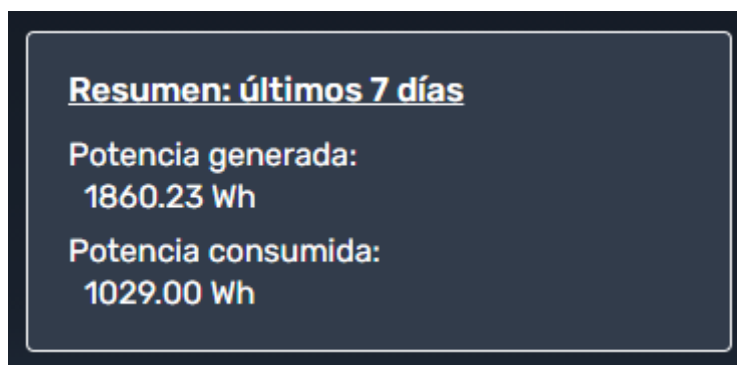


Ilustración 19: Resumen últimos 7 días

4. Histórico de datos

Si hacemos clic en cualquiera de los gráficos de tiempo real de la pantalla principal de monitorización, nos llevará a una pantalla en la que podremos ver un histórico de datos de ese gráfico.

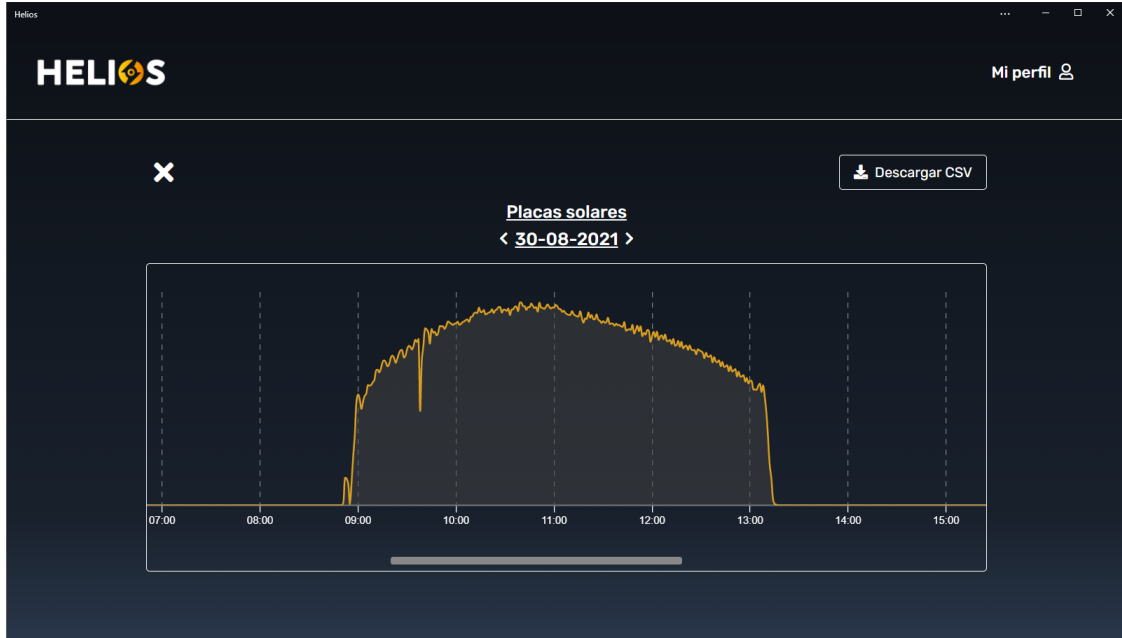


Ilustración 20: Gráfico de histórico de datos

La pantalla consta de un botón para volver a la pantalla principal de monitorización, un botón para descargar los datos del día seleccionado en formato CSV, el título del gráfico donde podremos pulsar para cambiar la fecha que queremos visualizar y por último la gráfica de datos del día, en la que podremos ver los valores de cada minuto del día y podremos desplazarnos para ver los datos a lo largo del día.

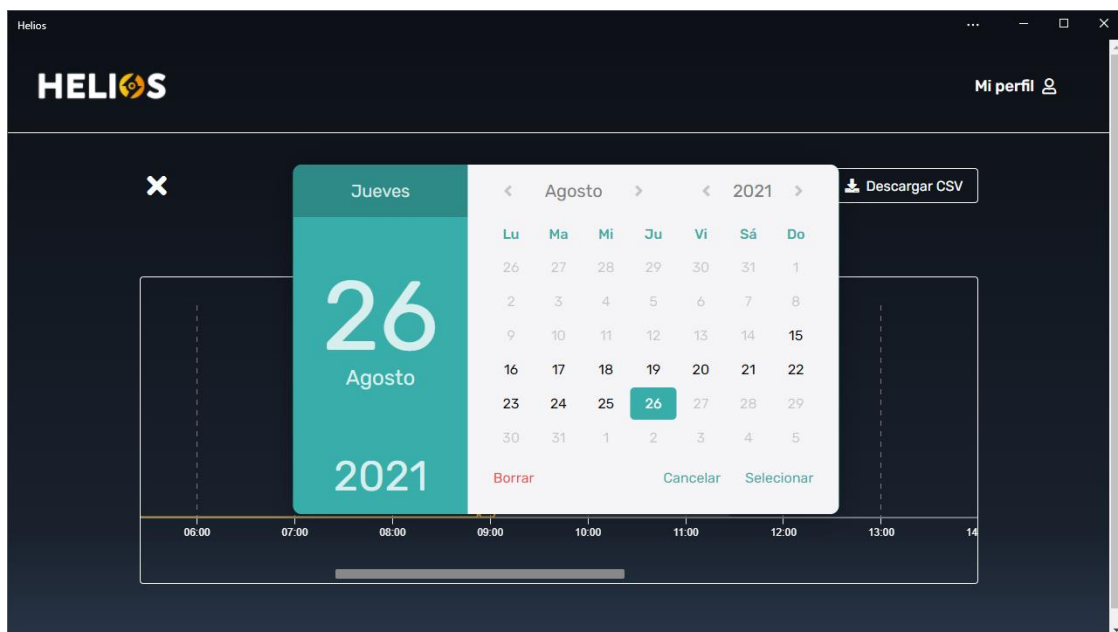


Ilustración 21: Calendario de selección de fecha

