

# EyeSee: La aplicación que identifica con quien te cruzas

## ANEXO IV

### Diseño del sistema

Trabajo de Fin de Grado  
Grado en Ingeniería Informática



**VNiVERSiDAD  
D SALAMANCA**

Junio 2023

Autor

Silvia Requejo López

Tutor/a

Gabriell Villarrubia González

André Sales Mendes



## ÍNDICE

ÍNDICE .....	3
ÍNDICE DE FIGURAS .....	4
1. Introducción.....	6
2. Modelo de diseño.....	6
2.1 Patrones arquitectónicos.....	7
2.1.1 MVVM .....	7
2.2 Subsistemas de diseño y arquitectura .....	9
2.2 Clases de diseño.....	9
2.3 Casos de uso de diseño .....	11
2.3.1 PAQUETE USUARIOS .....	11
2.3.2 PAQUETE AMIGOS .....	16
2.3.3 PAQUETE PASEOS .....	19
3. Diseño de la base de datos.....	22
4. Diseño de la interfaz .....	22
5. Referencias .....	26

---

## ÍNDICE DE FIGURAS

Figura 1. MVVM .....	8
Figura 2. Singleton .....	8
Figura 3. Subsistemas de diseño .....	9
Figura 4. Vista .....	10
Figura 5. Modelo de vista .....	10
Figura 6. Modelo .....	11
Figura 7. UC0001 Registrarse.....	11
Figura 8. UC0002 Iniciar Sesión.....	12
Figura 9. UC0003 Ver configuración. ....	12
Figura 10. UC0004 Cambiar nombre usuario.....	13
Figura 11. UC0005 Cambiar email .....	13
Figura 12. UC0006 Cambiar contraseña .....	14
Figura 13. UC0007 Cambiar MAC.....	14
Figura 14. UC0008 Eliminar cuenta .....	15
Figura 15. UC0009 Cerrar sesión.....	15
Figura 16. UC0010 Restablecer contraseña. ....	16
Figura 17. UC0011 Ver amigos .....	16
Figura 18. UC0012 Añadir amigo .....	17
Figura 19. UC0013 Eliminar amigo .....	17
Figura 20. UC0014 Añadir descripción de amigo .....	17
Figura 21. UC0015 Añadir alias de amigo.....	18
Figura 22. UC0016 Ver detalle de amigo .....	18
Figura 23. UC0017 Ver paseos .....	19
Figura 24. UC0018 Iniciar paseo.....	19
Figura 25. UC0019 Finalizar paseo .....	20
Figura 26. UC0020 Ver paseo.....	20
Figura 27. UC0021 Eliminar paseo .....	21
Figura 28. UC0022 Añadir detalles a un paseo.....	21
Figura 29. UC0023 Registrar encuentro.....	21
Figura 30. Diagrama de clases de Firestore.....	22

## Anexo IV. Diseño del sistema

Figura 31. Logotipo del sistema .....	23
Figura 32. Pantalla de registro de la web .....	23
Figura 33. Pantalla de inicio de sesión .....	24
Figura 34. Pantalla de inicio de usuario .....	24
Figura 35. Pantalla para añadir un amigo. ....	25
Figura 36. Pantalla de configuración. ....	25

---

## 1. Introducción

Este anexo describe la parte correspondiente al diseño dentro del proceso unificado. El diseño software es una parte crucial del proceso y, en este caso, se basará en el análisis realizado en los anexos anteriores.

En este anexo encontraremos los siguientes apartados:

- Modelo de Diseño:
  - o Patrones arquitectónicos: se describirán los patrones arquitectónicos que se utilizarán durante el desarrollo software.
  - o Subsistemas de diseño: se describirán los subsistemas que componen el sistema.
  - o Clases de diseño: se mostrarán las clases de diseño que compondrán el software, incluyendo una descripción de los métodos y atributos que contienen, así como las relaciones entre ellos.
  - o Realización de casos de uso: se mostrará la evolución de los casos de uso desde el análisis al diseño, utilizando las nuevas clases del diseño.
- Diseño de la base de datos: en este apartado se recogerá la organización interna de los datos dentro del almacenamiento.
- Vista de despliegue: incluirá la representación de los artefactos y el despliegue de nodos necesario para la puesta en funcionamiento del sistema.

## 2. Modelo de diseño

En este apartado se explicarán todas las elecciones de arquitectura y herramientas utilizadas en el proceso de desarrollo como los lenguajes de programación, el framework, etc.

- Se ha utilizado el patrón arquitectónico MVC.

## Anexo IV. Diseño del sistema

- En el desarrollo del front-end de la aplicación se ha utilizado el framework Vue.js, que combina HTML, CSS y JavaScript.
- En el desarrollo del back-end se ha utilizado el lenguaje de programación Python para el reconocimiento facial.
- El almacenamiento de los datos, así como el alojamiento de la web se han llevado a cabo utilizando Firebase, una herramienta proporcionada por Google.

### 2.1 Patrones arquitectónicos

#### 2.1.1 MVVM

El patrón Modelo-Vista-Modelo de vista es un patrón arquitectónico creado específicamente para desacoplar la interfaz de usuario de la lógica de negocio en el desarrollo de webs.

A continuación, se explicarán las diferentes partes del patrón:

- Vista: componentes web construidos. Consta de las interfaces del usuario.
- Modelo: guarda los datos y el estado de la aplicación.
- Modelo de vista: implementa propiedades y comandos a los que la vista puede enlazar datos y notifica a la vista de cambios de estado mediante eventos de notificación. También aísla la vista del modelo y permite que el modelo evolucione independientemente de la vista.

El uso de este patrón supone varias ventajas.

- El modelo de vista actuará como un adaptador para las clases de modelo evitando que se realicen cambios importantes en el código del modelo.
- Se podrán crear pruebas unitarias para el modelo de vista y el modelo sin usar la vista.
- La interfaz de usuario se podrá modificar libremente sin que ello suponga cambios en el código del modelo o en modelo de vista.
- Un desarrollador podrá trabajar simultáneamente en los componentes durante el desarrollo.

En la Figura 1 se incluye una representación gráfica del sistema.

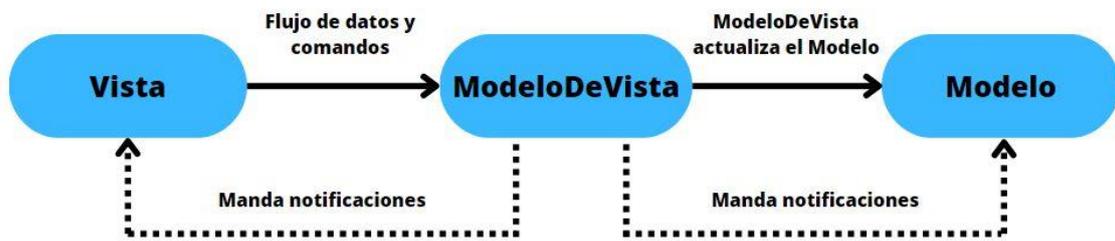


Figura 1. MVVM

### 2.1.2 Singleton

El patrón singleton es un patrón de diseño que permite restringir la creación de objetos pertenecientes a una clase. De esta forma se asegura que esa clase tenga sólo una única instancia y que se acceda a ella desde un punto global.

En la Figura 2 se incluye un diagrama del patrón Singleton.

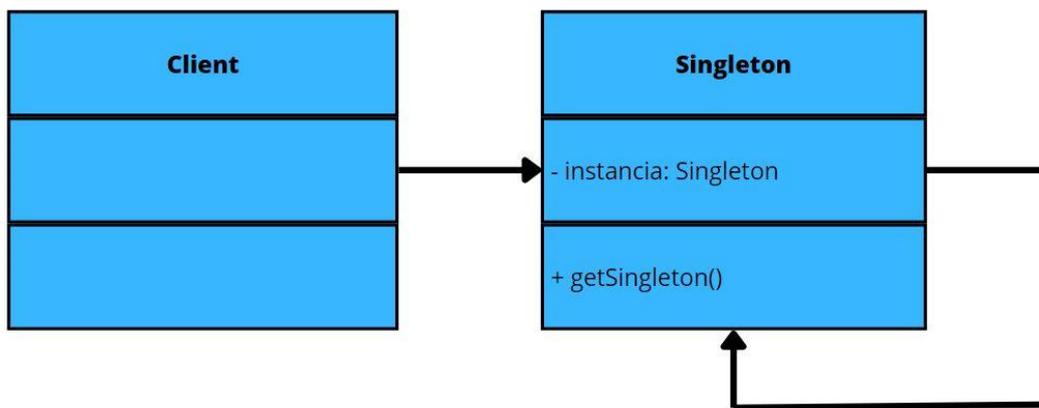


Figura 2. Singleton

## 2.2 Subsistemas de diseño y arquitectura

La realización de los subsistemas de diseño se ha desglosado en tres paquetes distintos. En primer lugar, tendremos el paquete web que constará del servicio web de la aplicación. En segundo lugar, contaremos con el paquete firebase, que constará de la API de firebase. Finalmente, contaremos con el paquete FaceRecognition que contendrá la API de reconocimiento facial.

En la Figura 3 se pueden observar los subsistemas de diseño.

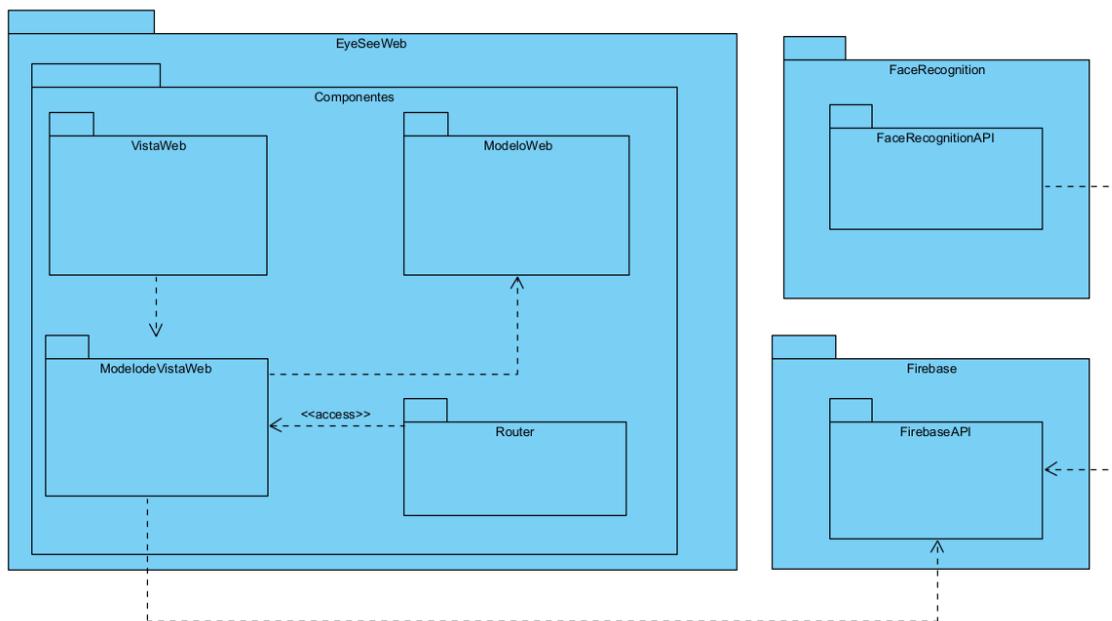


Figura 3. Subsistemas de diseño

### 2.2 Clases de diseño.

En este apartado se especificarán los contenidos del patrón Modelo-Vista-Modelo de vista. Para cada elemento se especificarán los principales métodos y atributos del mismo.

Dentro del paquete aplicación web encontraremos el paquete VistaWeb.

En la Figura 4 observamos una representación gráfica de dicho paquete con todas las vistas disponibles en la aplicación web.

## Anexo IV. Diseño del sistema

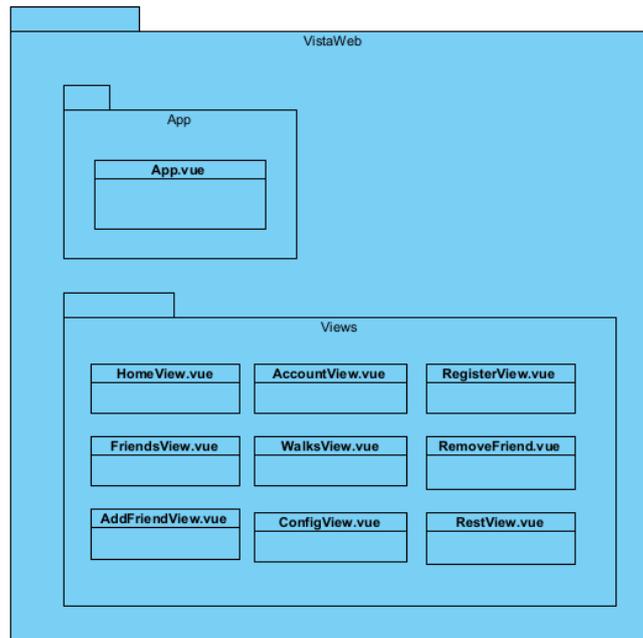


Figura 4. Vista

También nos encontraremos el paquete ModeloVista. En la Figura 5 se detallan los diferentes métodos mediante los cuales se gestiona el flujo de datos entre la vista y el modelo.

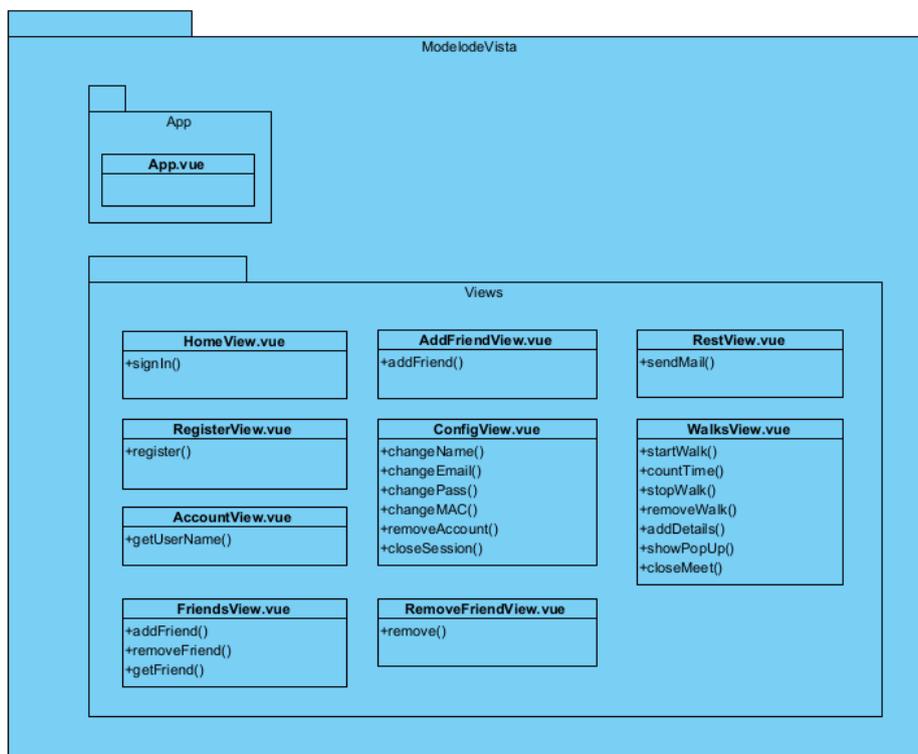


Figura 5. Modelo de vista

## Anexo IV. Diseño del sistema

Por último, tendremos el paquete Modelo que representará todas las colecciones de datos disponibles con los atributos de cada uno.

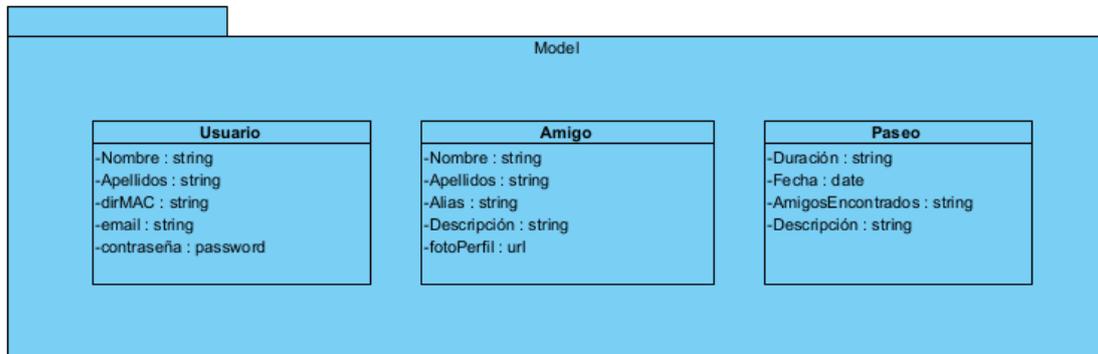


Figura 6. Modelo

## 2.3 Casos de uso de diseño

### 2.3.1 PAQUETE USUARIOS

#### UC0001 Registrarse

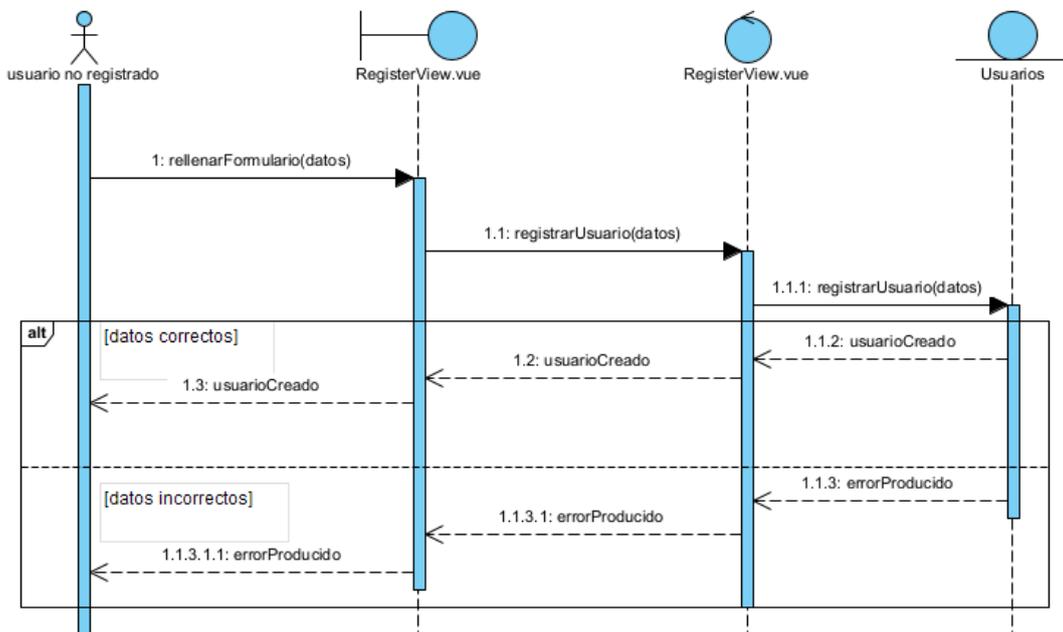


Figura 7. UC0001 Registrarse

**UC0002 Iniciar sesión**

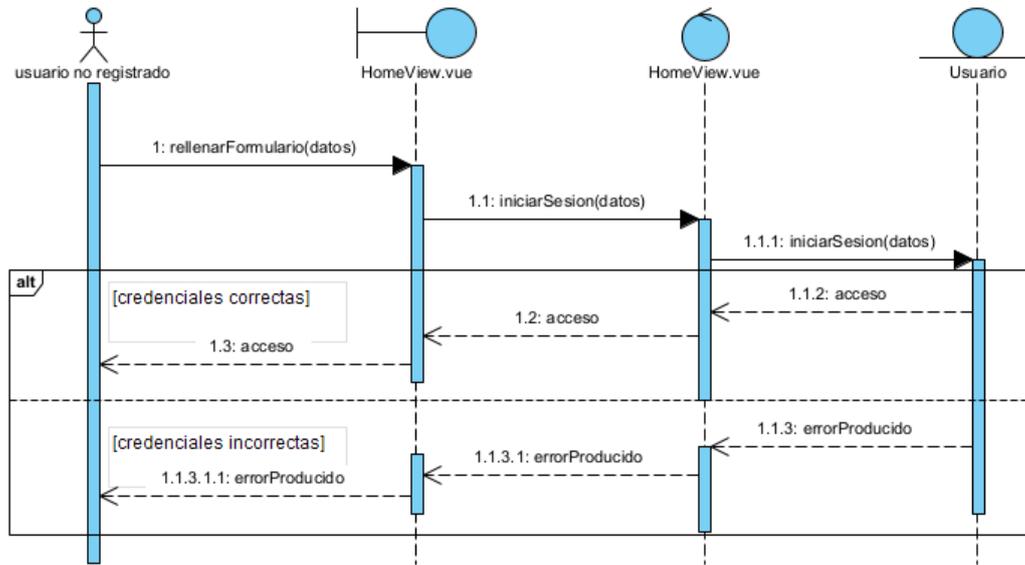


Figura 8. UC0002 Iniciar Sesión

**UC0003 Ver configuración**

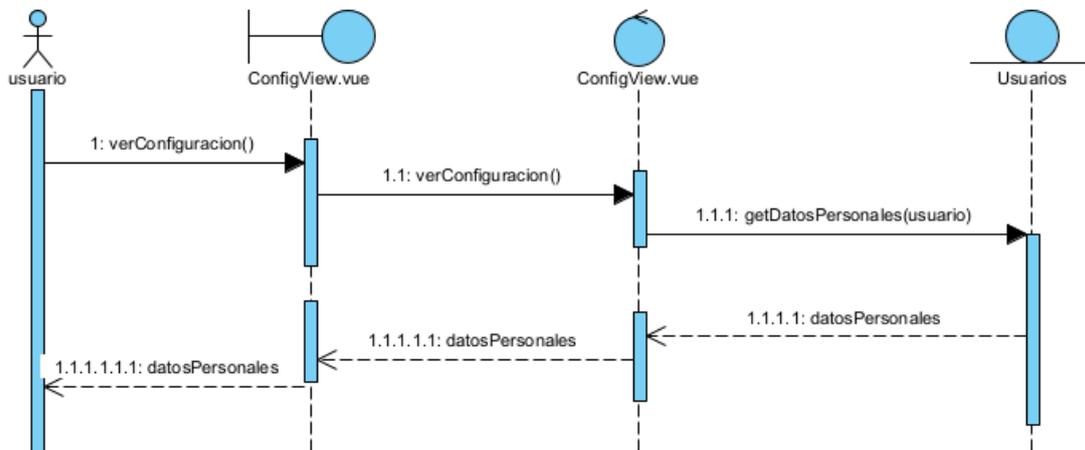


Figura 9. UC0003 Ver configuración.

### UC0004 Cambiar nombre usuario

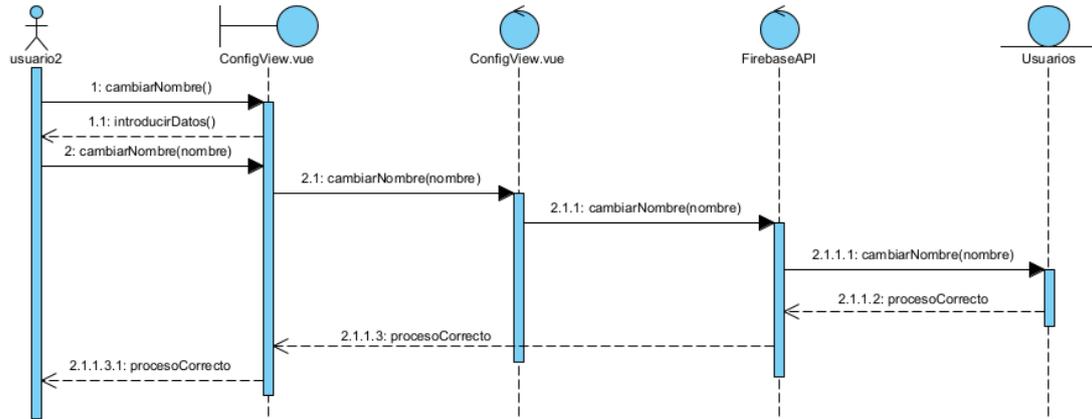


Figura 10. UC0004 Cambiar nombre usuario.

### UC0005 Cambiar email

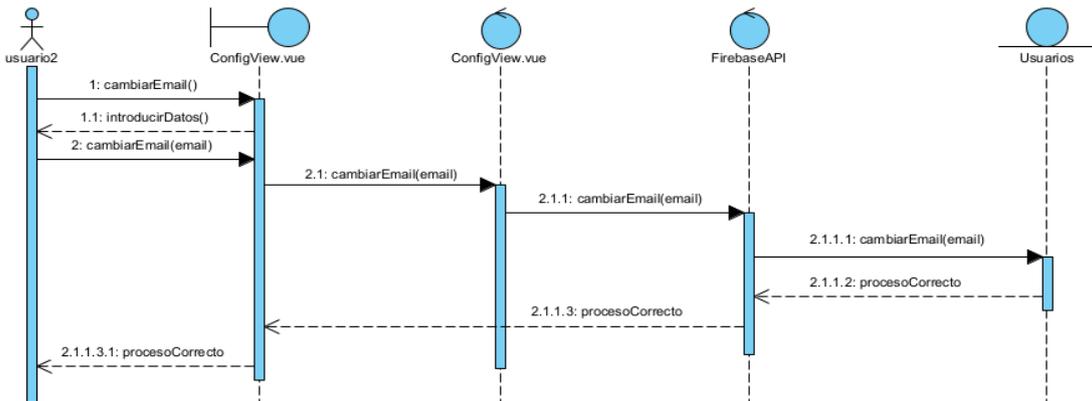


Figura 11. UC0005 Cambiar email

**UC0006 Cambiar contraseña**

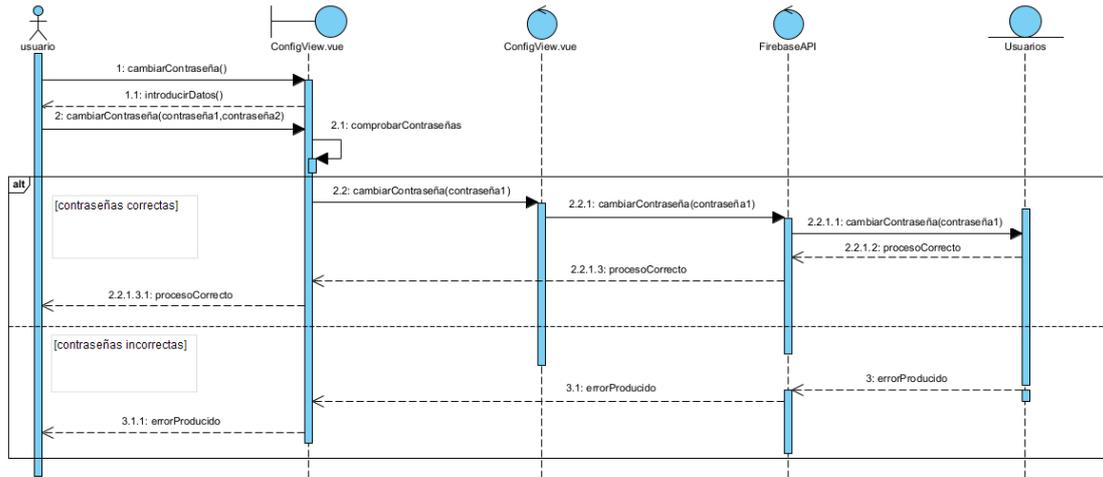


Figura 12. UC0006 Cambiar contraseña

**UC0007 Cambiar MAC**

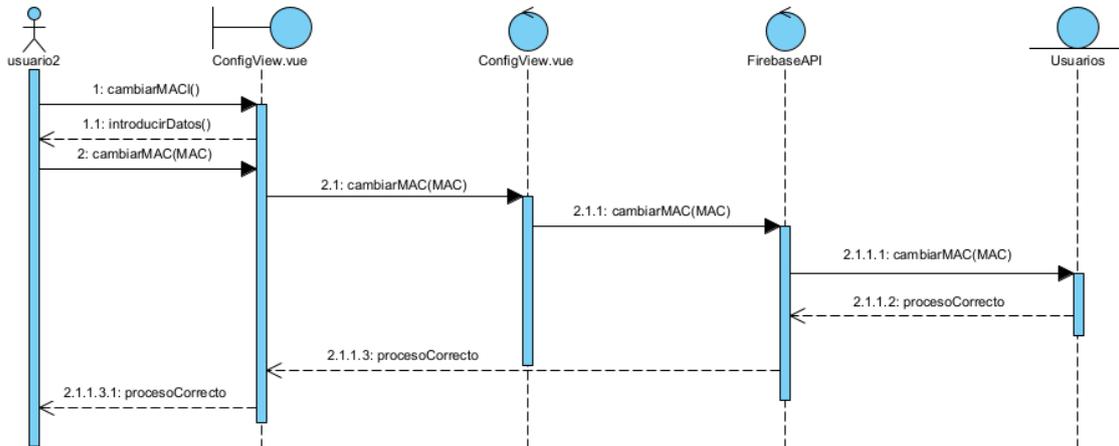


Figura 13. UC0007 Cambiar MAC

### UC0008 Eliminar cuenta

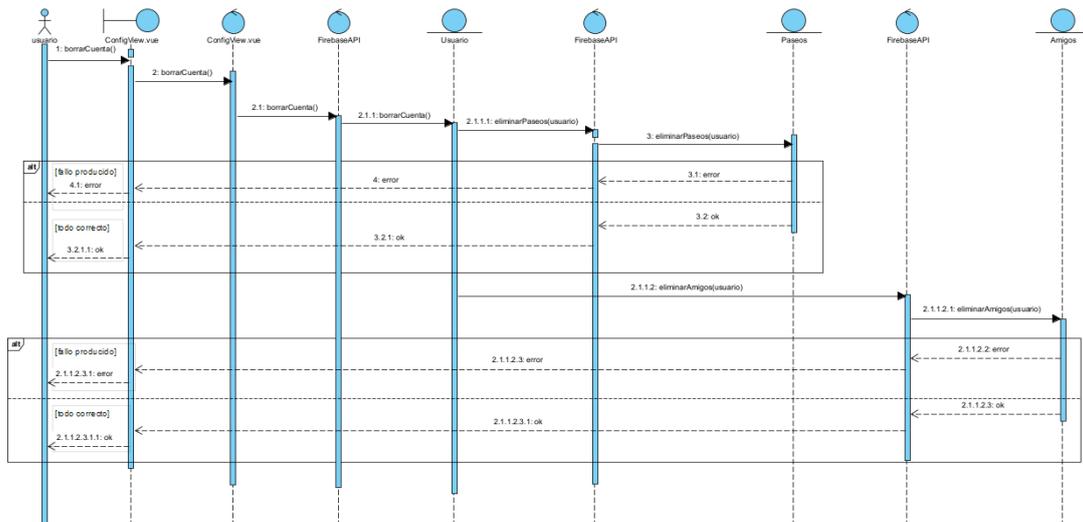


Figura 14. UC0008 Eliminar cuenta

### UC0009 Cerrar sesión

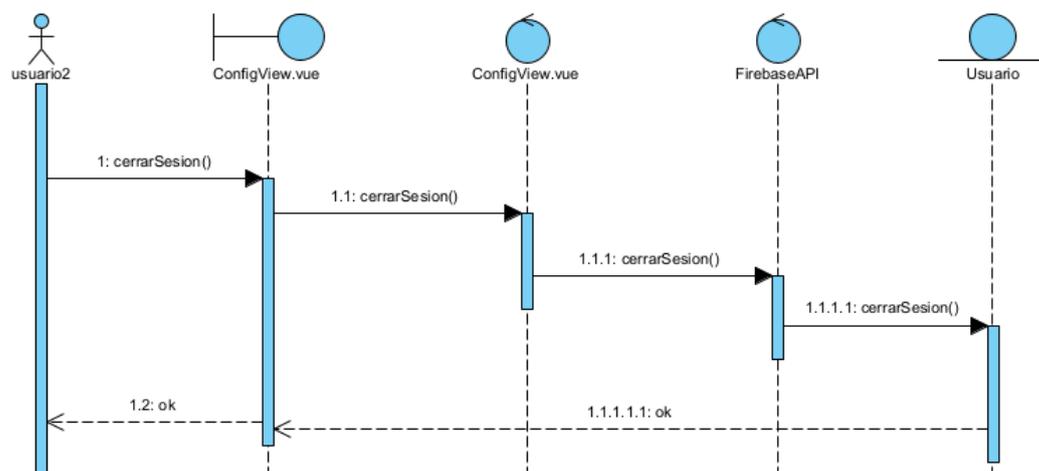


Figura 15. UC0009 Cerrar sesión.

### UC0010 Restablecer contraseña

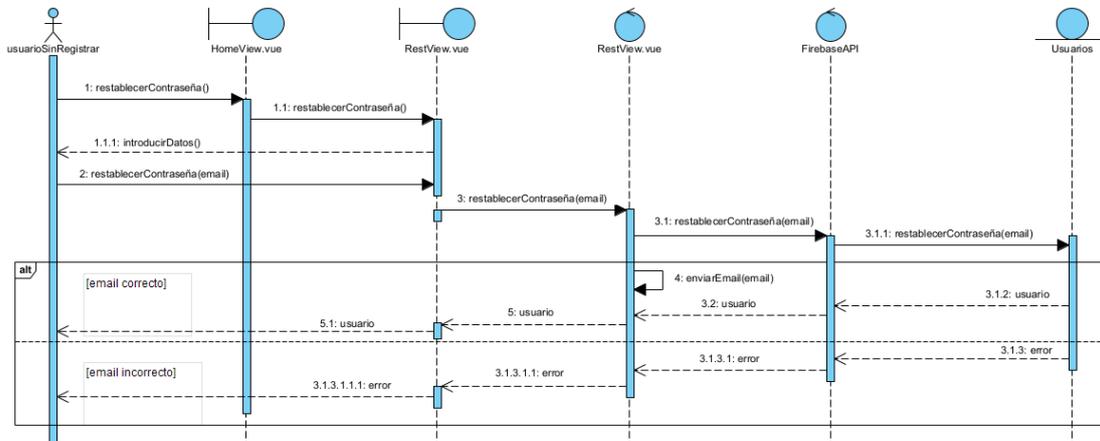


Figura 16. UC0010 Restablecer contraseña.

## 2.3.2 PAQUETE AMIGOS

### UC0011 Ver amigos

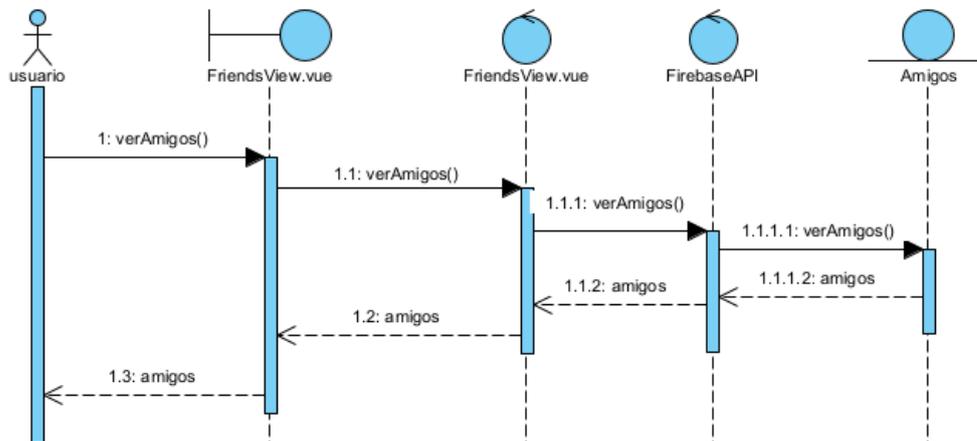


Figura 17. UC0011 Ver amigos

**UC0012 Añadir amigo**

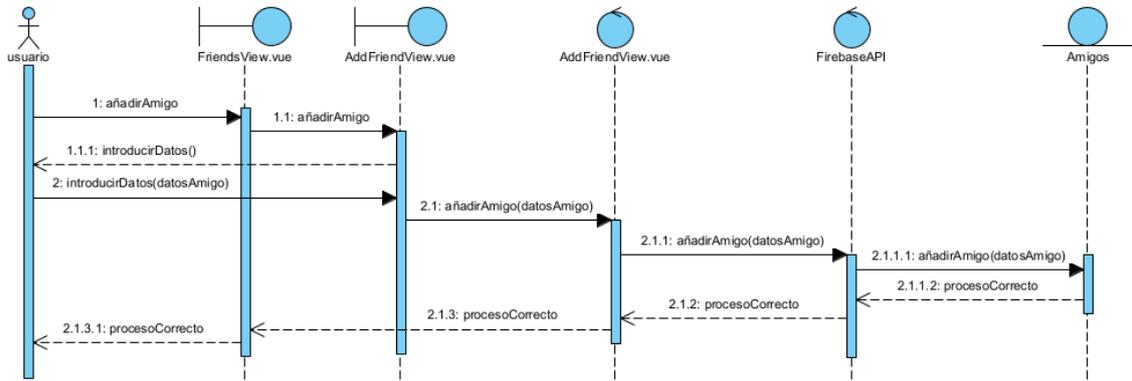


Figura 18. UC0012 Añadir amigo

**UC0013 Eliminar amigo**

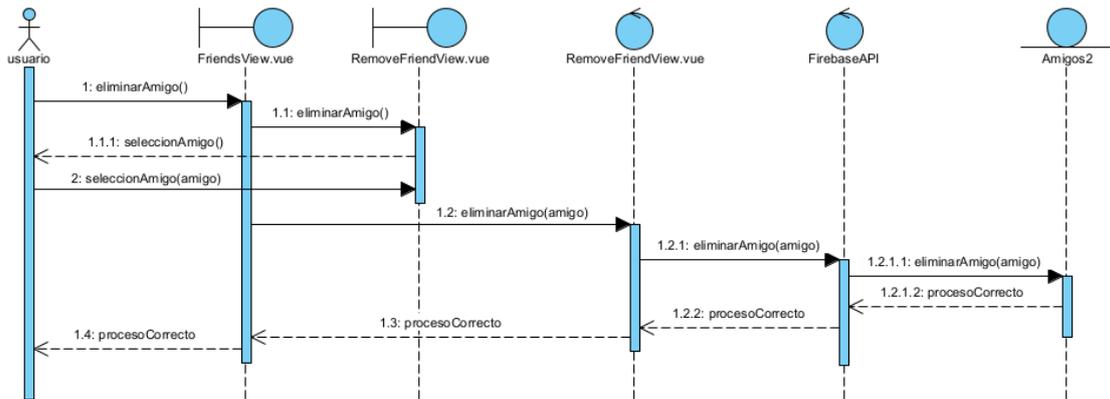


Figura 19. UC0013 Eliminar amigo

**UC0014 Añadir descripción de amigo**

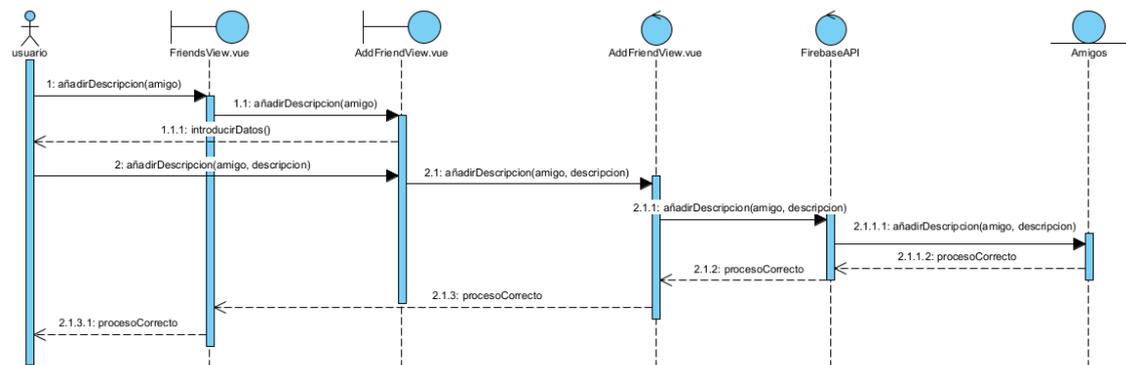


Figura 20. UC0014 Añadir descripción de amigo

**UC0015 Añadir alias de amigo**

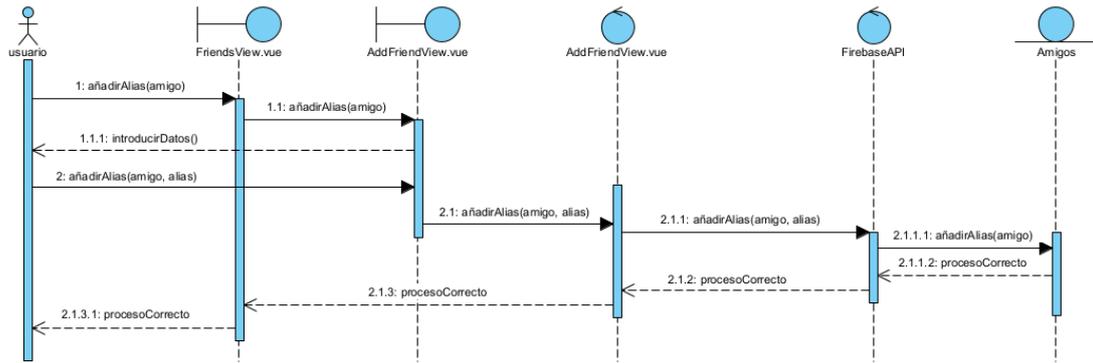


Figura 21. UC0015 Añadir alias de amigo

**UC0016 Ver detalle de amigo**

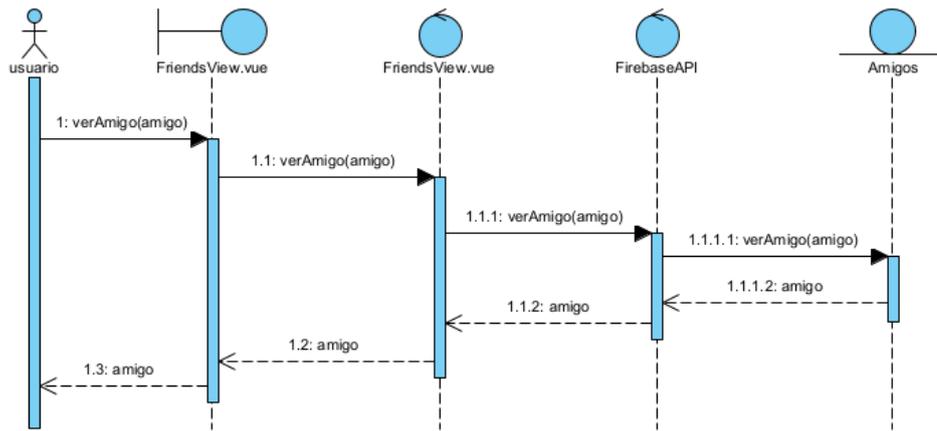


Figura 22. UC0016 Ver detalle de amigo

### 2.3.3 PAQUETE PASEOS

#### UC0017 Ver paseos

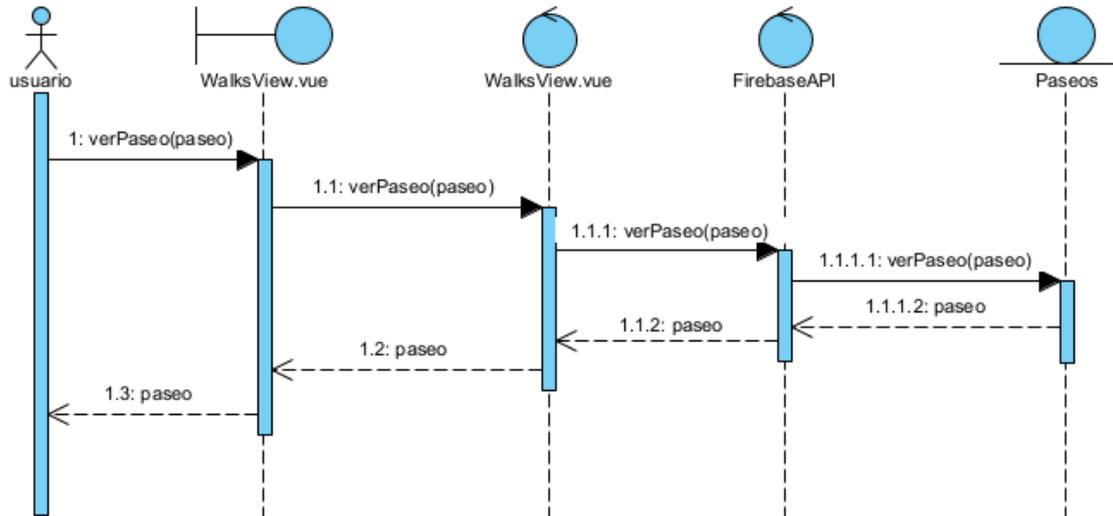


Figura 23. UC0017 Ver paseos

#### UC0018 Iniciar paseo

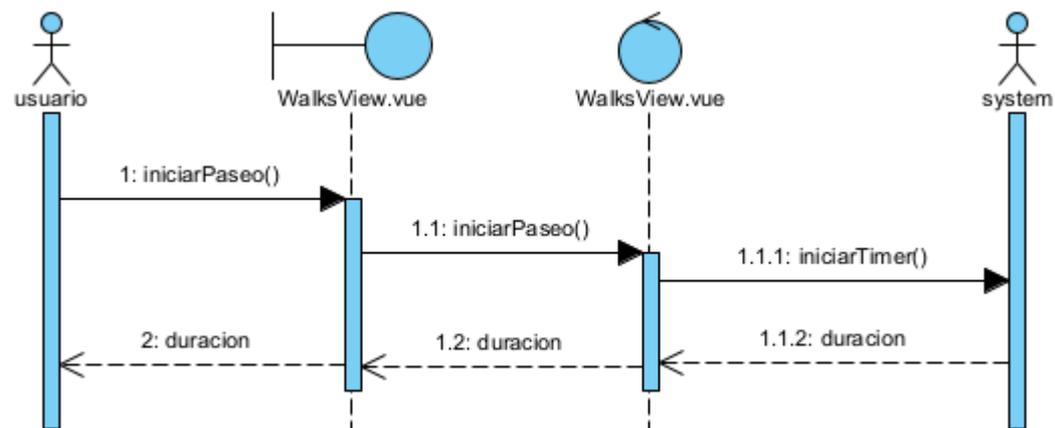


Figura 24. UC0018 Iniciar paseo

**UC0019 Finalizar paseo**

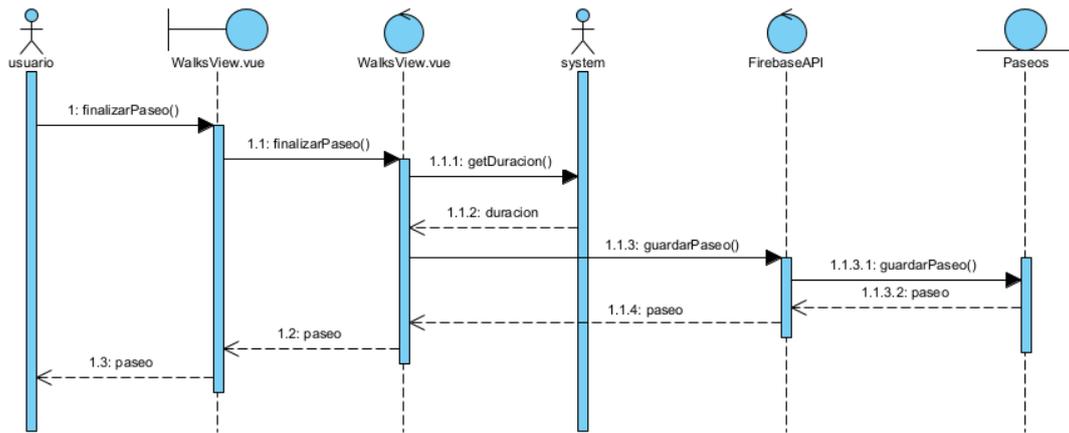


Figura 25. UC0019 Finalizar paseo

**UC0020 Ver paseo**

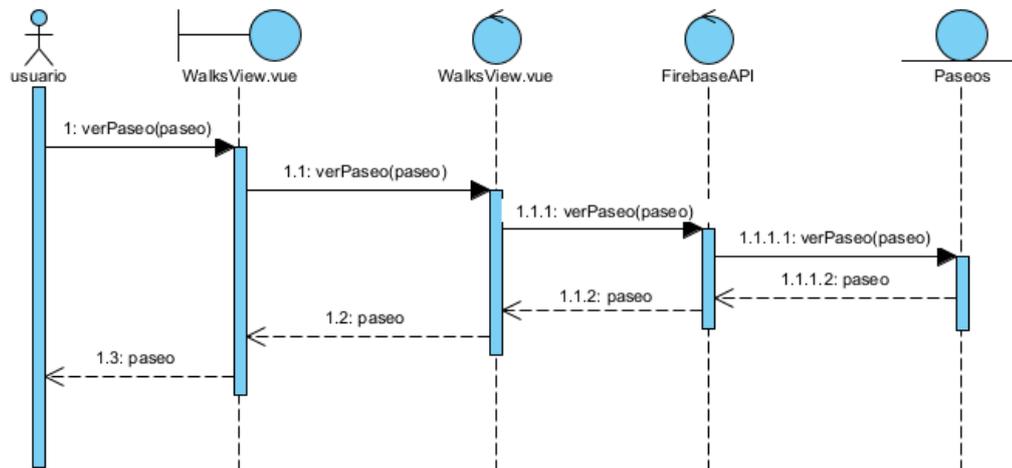


Figura 26. UC0020 Ver paseo

**UC0021 Eliminar paseo**

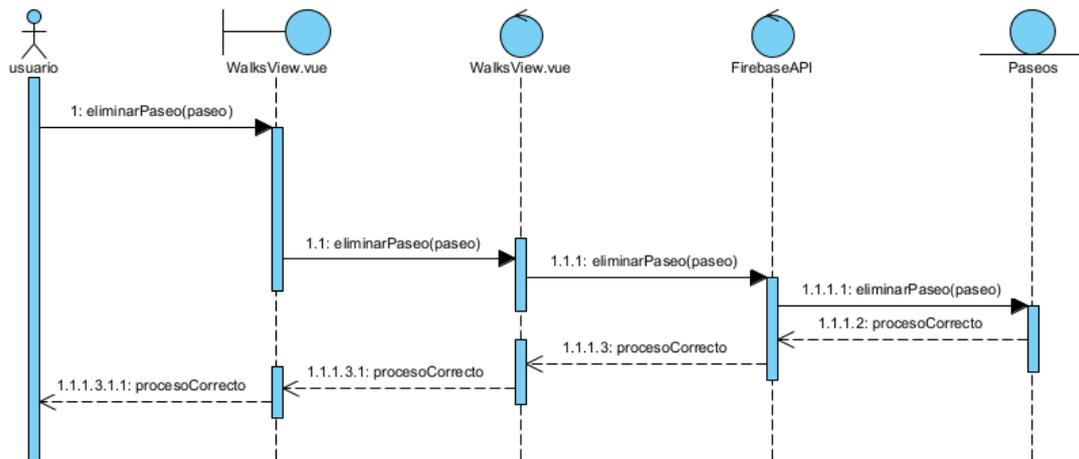


Figura 27. UC0021 Eliminar paseo

**UC0022 Añadir detalles a un paseo**

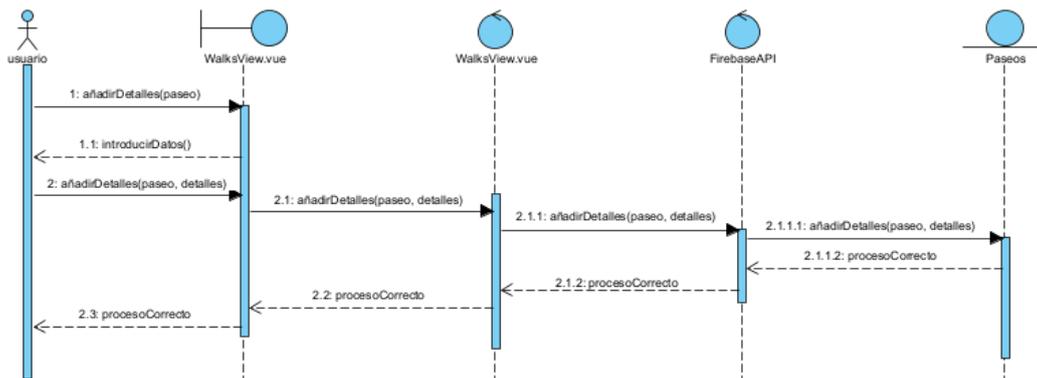


Figura 28. UC0022 Añadir detalles a un paseo

**UC0023 Registrar encuentro**

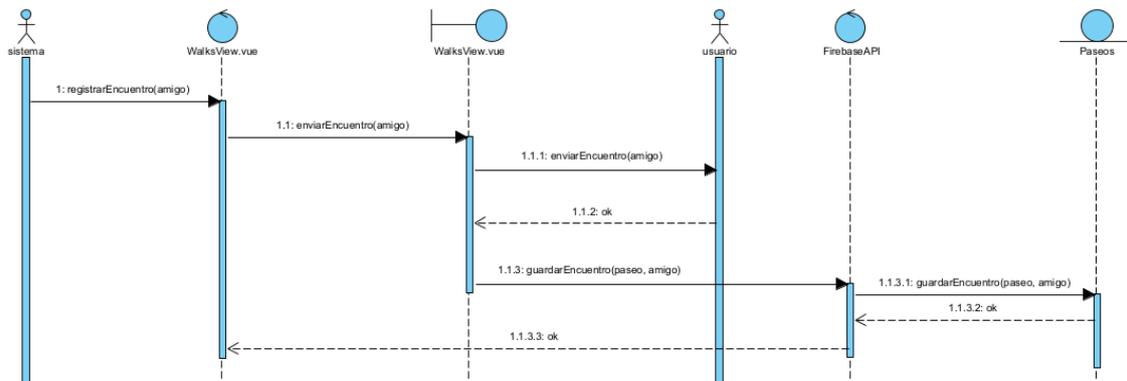


Figura 29. UC0023 Registrar encuentro.

### 3. Diseño de la base de datos

En este proyecto se ha utilizado la base de datos Firebase de Google. Dicha base de datos se denomina Firestore. Firestore es una base de datos NoSQL flexible, escalable y en la nube pensada para almacenar y sincronizar datos para el desarrollo del lado del cliente y del lado del servidor. Los datos se mantienen sincronizados a través de objetos de escucha en tiempo real.

La estructura diseñada para la aplicación web se describe en la figura que encontramos a continuación. Contaremos con una colección de usuarios denominada “users” y otra denominada MACs.

En la colección denominada users guardaremos toda la información relativa a los usuarios, como sus amigos, paseos asociados, datos personales, etc.

Además, contaremos con la colección MACs que guardará los encuentros registrados durante un paseo.

En la siguiente figura se puede observar un diagrama de clases del sistema.

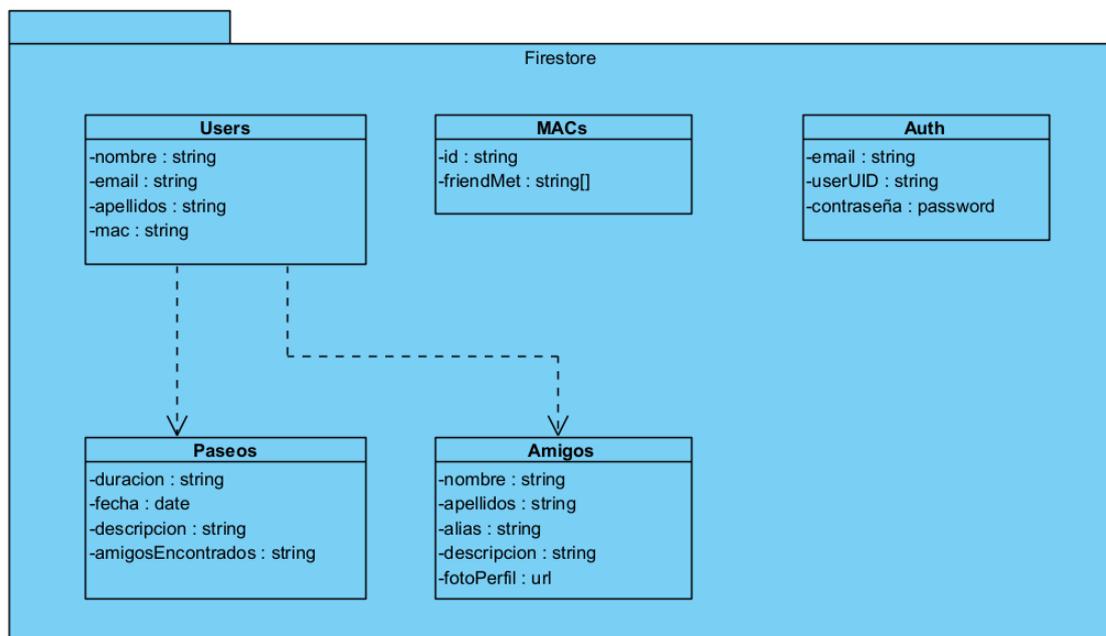


Figura 30. Diagrama de clases de Firestore

### 4. Diseño de la interfaz

El diseño de la interfaz se ha llevado a cabo buscando el máximo contraste y sencillez. El desarrollador es consciente de que la aplicación está pensada para personas con un cierto grado de discapacidad visual y por tanto se ha

tratado de utilizar combinaciones de colores que introdujeran gran contraste entre el fondo y el texto. Así mismo, se ha intentado presentar los elementos en pantalla con un tamaño superior para facilitar el uso de la aplicación.

Se ha apostado por la sencillez ya que se tiene en cuenta que en algunos casos el usuario final podría necesitar ayuda de algún familiar o amigo para poder usar la app y por ello se ha buscado que sea un diseño sencillo y fácil de comprender para usuarios de cualquier edad.

A continuación, se muestra el diseño inicial construido en Adobe xd como prototipo digital. Con este prototipo se han llevado a cabo pruebas de usuario de las cuales se han extraído las conclusiones pertinentes para decidir el diseño final del sistema.

## EyeSee



Figura 31. Logotipo del sistema

## EyeSee



Nombre

Apellidos

Email

Contraseña

Repetir contraseña

Registrarse

Figura 32. Pantalla de registro de la web



Figura 33. Pantalla de inicio de sesión



Figura 34. Pantalla de inicio de usuario



Figura 35. Pantalla para añadir un amigo.

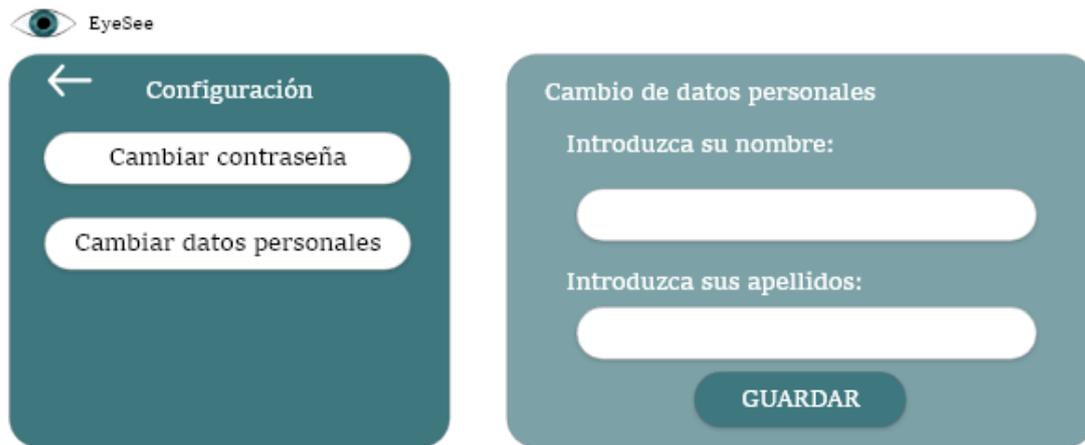


Figura 36. Pantalla de configuración.

## 5. Referencias

- [1] Modelo-Vista-Modelo de vista. (n.d.). Microsoft.com. Retrieved June 20, 2023, from <https://learn.microsoft.com/es-es/dotnet/architecture/maui/mvvm>
- [2] ¿Qué es el patrón singleton? (2022, November 10). KeepCoding Bootcamps. <https://keepcoding.io/blog/el-patron-singleton/>
- [3] Firestore. (n.d.). Firebase. Retrieved June 20, 2023, from <https://firebase.google.com/docs/firestore?hl=es-419>