

# APLICACIÓN PARA LA COMUNICACIÓN ENTRE CENTROS EDUCATIVOS Y FAMILIAS

## Anexo IV: Diseño del Sistema



VNiVERSIDAD  
D SALAMANCA

Septiembre de 2023

**Trabajo de fin de Grado**

**Tutor:**

María Navarro Cáceres

**Alumno:**

Raúl Rodríguez Villanueva

## Tabla de Contenido

1. INTRODUCCIÓN.....	4
2. PATRONES ARQUITECTONICOS.....	4
2.1. MVVM .....	4
2.2. CLIENTE-SERVIDOR.....	5
3. SUBSISTEMAS DE DISEÑO .....	6
4. REALIZACIÓN DE LOS CASOS DE USO .....	9
5. DISEÑO DE LA BASE DE DATOS .....	23
6. DISEÑO DE LA INTERFAZ.....	24
7. VISTA DE DESPLIEGUE.....	28
8. REFERENCIAS.....	29

## Tabla de Imágenes

Imagen 1. Patron MVVM.....	5
Imagen 2. Patrón Cliente-Servidor .....	6
Imagen 3. Subsistema View.....	7
Imagen 4. Subsistema Model .....	7
Imagen 5. Subsistema ViewModel .....	8
Imagen 6. Subsistema Adapter.....	8
Imagen 7. Subsistema completo .....	9
Imagen 8. UC-001 Registrarse .....	10
Imagen 9. UC-002 Iniciar sesion.....	10
Imagen 10. UC-003 Recuperar contraseña.....	11
Imagen 11. UC-004 Cerrar sesión.....	11
Imagen 12. UC-005 Ver lista de exámenes.....	12
Imagen 13. UC-006 Añadir examen.....	12
Imagen 14. UC-007 Modificar examen.....	13
Imagen 15. UC-008 Eliminar examen .....	13
Imagen 16. UC-009 Ver calendario de actividades.....	14
Imagen 17. UC-010 Avanzar semana en el calendario .....	14
Imagen 18. UC-011 Ir semana atrás en el calendario.....	14
Imagen 19. UC-012 Añadir actividad.....	15
Imagen 20. UC-013 Modificar actividad.....	15
Imagen 21. UC-014 Eliminar actividad .....	16
Imagen 22. UC-015 Ver resultados.....	16
Imagen 23. UC-016 Seleccionar jornada .....	17
Imagen 24. UC-017 Añadir jornada .....	17
Imagen 25. UC-018 Eliminar jornada .....	18
Imagen 26. UC-019 Añadir resultado .....	19
Imagen 27. UC-020 Modificar resultado .....	19
Imagen 28. UC-021 Eliminar resultado.....	20
Imagen 29. UC-022 Ver información de contacto .....	20
Imagen 30. UC-023 Ver ajustes .....	21
Imagen 31. UC-024 Modificar datos del usuario.....	21
Imagen 32. UC-025 Modificar contraseña.....	22
Imagen 33. UC-026 Asignar permisos a usuario .....	22
Imagen 34. Paleta de colores .....	24
Imagen 35. Primera Pantalla de Login.....	25
Imagen 36. Primera pantalla principal .....	26
Imagen 37. Diseño final de Login .....	27
Imagen 38. Diseño final pantalla principal .....	27
Imagen 39. Diagrama de despligue .....	28

## 1. INTRODUCCIÓN

En este anexo IV se va a tratar sobre el diseño del sistema para nuestro proyecto, siguiendo el patrón de Proceso Unificado. El diseño de nuestro sistema es una parte muy importante de cara a la futura elaboración del software, ya que aquí se plantea una solución a los requisitos extraídos para nuestra aplicación.

Vamos a seguir la siguiente estructura:

- Patrones arquitectónicos: Se explicarán los patrones utilizados para organizar la estructura de nuestro software.
- Subsistemas de diseño: Descompondremos el sistema en sistemas más pequeños para tener una mayor manejabilidad de los problemas
- Realización de casos de uso: Se van a exponer, paso a paso y siguiendo un diagrama de secuencia, la evolución de nuestros casos de uso.
- Diseño de la base de datos: Se explica cómo esta organizada la base de datos en la que se almacenarán los datos relacionados con nuestro proyecto.
- Vista de despliegue: Se muestran los artefactos relacionados con el despliegue del entorno.

## 2. PATRONES ARQUITECTONICOS

### 2.1. MVVM

El patrón de diseño Model-View-ViewModel (MVVM) es uno de los patrones de diseño arquitectónico más utilizado en el desarrollo de aplicaciones móviles, en especial para Android. Lo que busca este patrón es separar la lógica de negocio y la lógica de presentación de la interfaz de usuario, obteniendo una mejor organización del código y mejorando la mantenibilidad de la aplicación.

Este patrón consta de tres componentes:

1. Modelo (Model): Representa los datos de nuestra aplicación, siguiendo una estructuración de los mismos para poder almacenarlos en nuestra base de datos Firebase Firestore. El modelo desconoce la interfaz de usuario ni como se presentan los datos al usuario, simplemente se encarga de almacenar y proporcionar lo que el ViewModel le proporciona.
2. Vista (View): Representa la interfaz con la que el usuario interactúa, encargándose de mostrarle los datos correspondientes y recibir las notificaciones y acciones que este realiza sobre la aplicación. La vista se limita a lo anterior, sin realizar funciones de cálculo ni acceso a datos.
3. ViewModel: Es el componente central de nuestro patrón de diseño MVVM, siendo el encargado de intermediar entre la vista y el modelo. No tiene conocimiento directo de la vista, por lo que es muy independiente, solamente se encarga de realizar las operaciones correspondientes con los datos recuperados del modelo para enviárselos a la vista en caso de que ésta lo solicite, y viceversa, también se encarga de recuperar las notificaciones proporcionadas por el usuario en la vista.

Para que nuestra vista se mantenga actualizada se utiliza un tipo de dato muy común en este patrón como es el LiveData. Con esto se conectan el View y el ViewModel, siendo la vista la que se encuentra observando el LiveData, por lo que en caso de que haya alguna modificación de este por parte del ViewModel, la vista será notificada y actualizada.

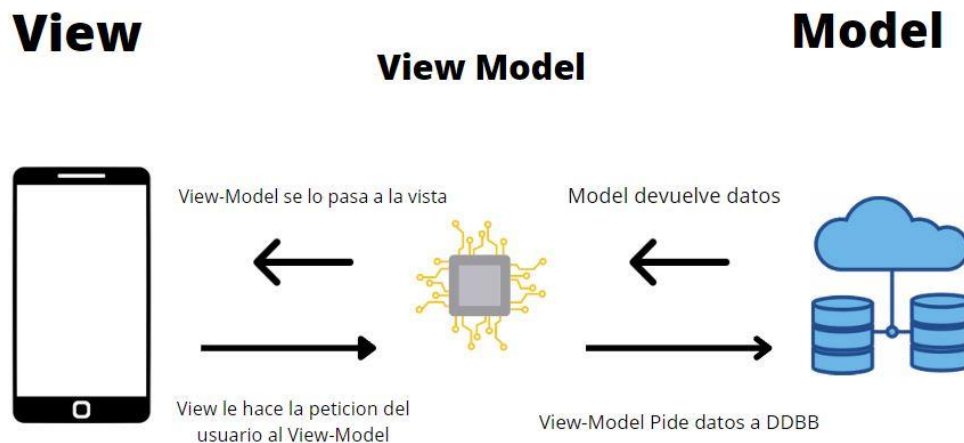


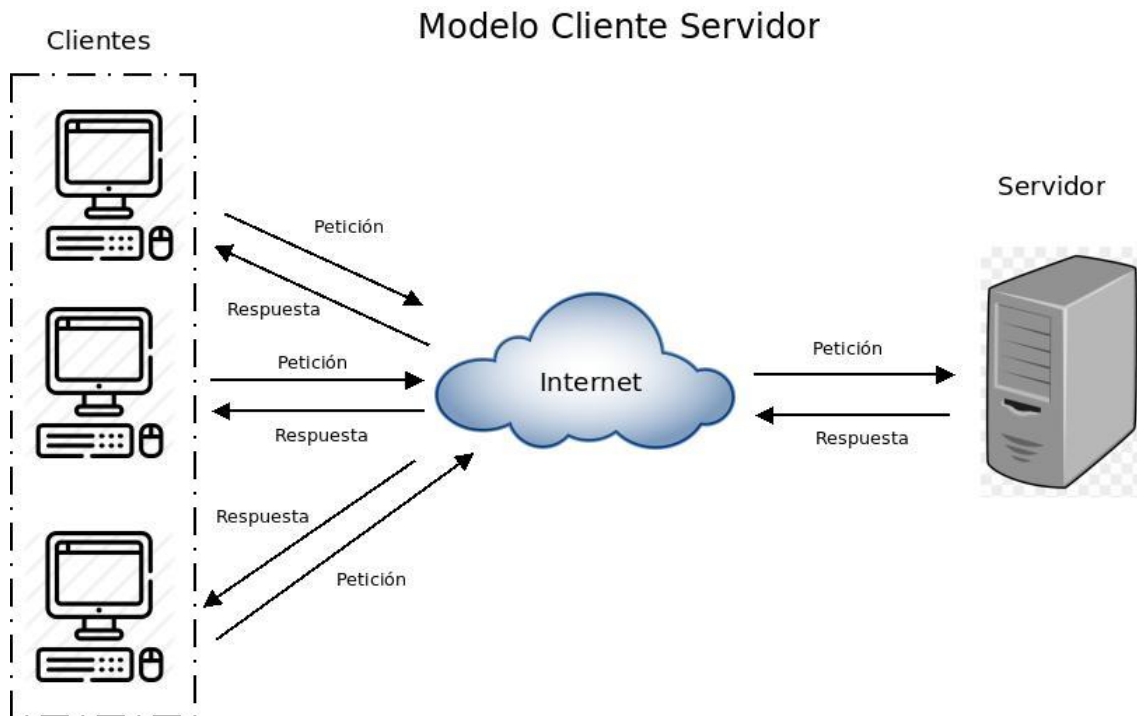
Imagen 1. Patrón MVVM

## 2.2. CLIENTE-SERVIDOR

Este patrón es utilizado por nuestra base de datos Firebase Firestore para el almacenamiento de los datos, donde se cuenta con uno o varios clientes (la aplicación) y un servidor (los servidores de Firebase donde se almacenan los datos).

A continuación se explica el funcionamiento de dicho patrón:

- **Cliente:** Es la aplicación ejecutada por el usuario, en nuestro caso San Juan Bosco. El cliente se comunicará con el servidor para acceder a los datos contenidos en este y poder verlos, modificarlos, añadir nuevos, etc.
- **Servidor:** Los servidores de Firebase, pertenecientes a Google, son los encargados de ejercer como servidor. Aquí se encuentran todas las operaciones relacionadas con el almacenamiento y recopilación del contenido del servidor.
- **Comunicación:** La comunicación entre el cliente y el servidor tiene lugar a través de la propia API que nos proporciona Firebase, permitiendo al cliente hacer una gestión correcta de los datos.
- **Escritura y Lectura de Datos:** Cuando el cliente desea leer o escribir datos en Firebase Firestore, va a enviar una solicitud al servidor a través de la API, y está será respondida por el servidor indicando éxito o error.



*Imagen 2. Patrón Cliente-Servidor*

### 3. SUBSISTEMAS DE DISEÑO

Nuestro proyecto software va a encontrarse dividido en diferentes subsistemas, cada uno de los cuales tendrá unas funcionalidades específicas, con unas funciones concretas, dividiendo así el trabajo y especializando cada componente en unas tareas determinadas.

Para dividirlo, se ha seguido el patrón arquitectónico de Model-View-ViewModel, el cual se ha explicado anteriormente. Por tanto vamos a tener una división en base a cada uno de los componentes de dicho patrón. Dentro de cada uno se encontrarán las diferentes clases con las funciones más destacadas las cuales nos van a servir para desarrollar los requisitos solicitados por el cliente.

Tendremos los siguientes subsistemas:

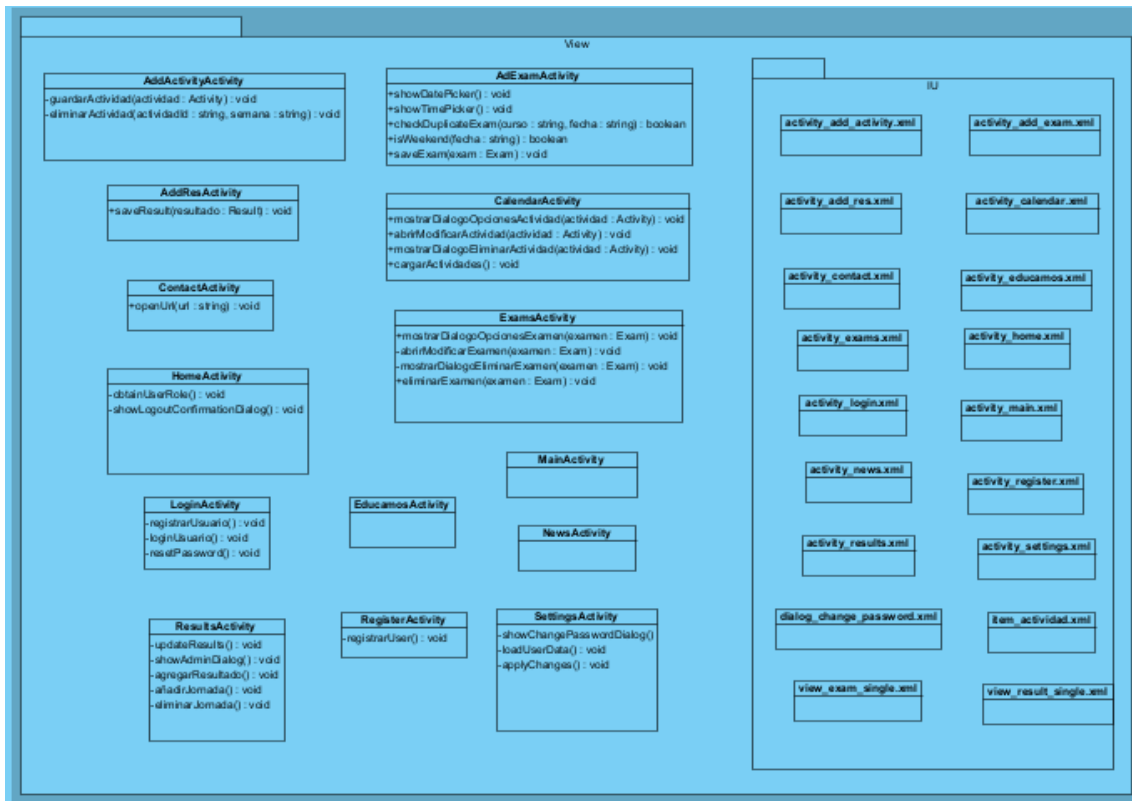


Imagen 3. Subsistema View

En primer lugar tenemos la vista, con todas sus clases, tanto las XML como las Java. En las clases XML se encuentra el código relacionado con la interfaz gráfica, por ello se ha hecho una división en la vista, colocando todos los código XML dentro de la interfaz del usuario.

También se pueden ver las clases Java donde se inicializan los componentes de la vista y se hacen peticiones al ViewModel para presentar los datos, almacenar datos, etc.

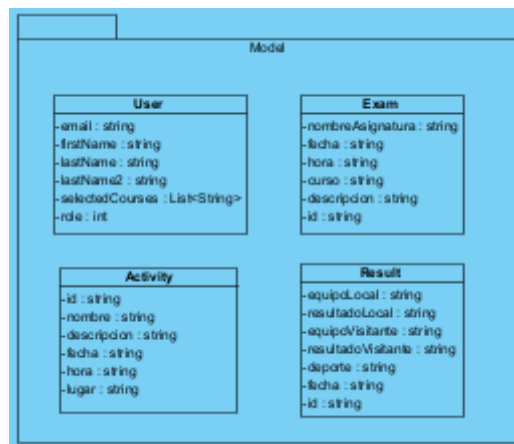


Imagen 4. Subsistema Model

En cuanto al modelo, tenemos las clases relacionadas con los datos, con sus parámetros más significativos. Este tipo de datos será el que se almacenará en Firebase Firestore cuando hagamos una petición de inserción de datos.

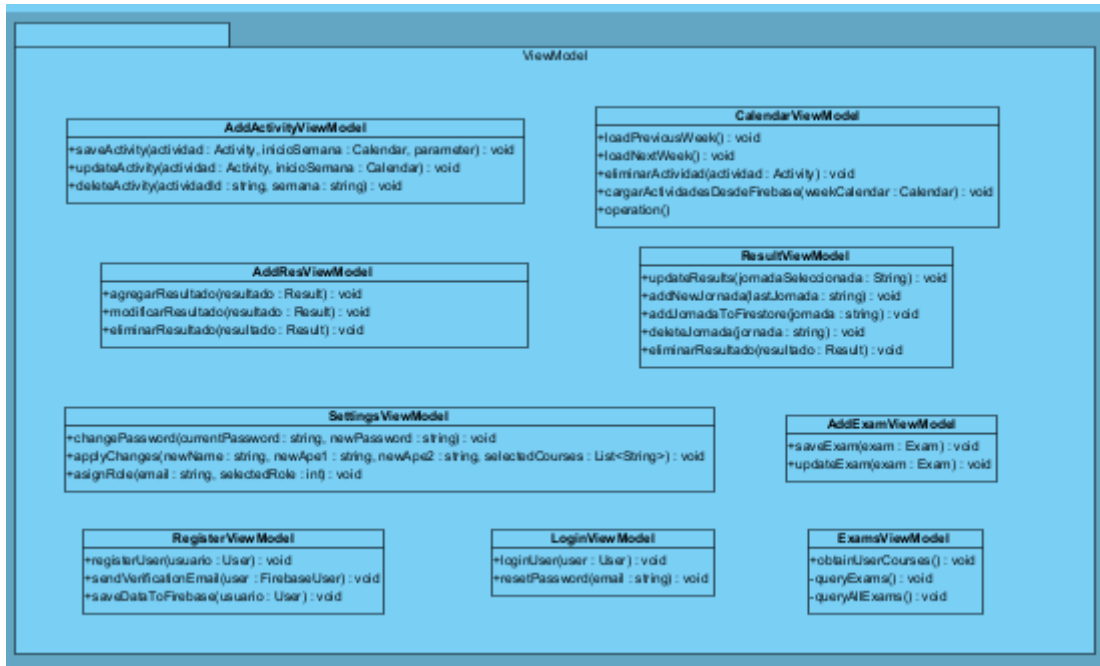


Imagen 5. Subsistema ViewModel

En el ViewModel se encuentran las funciones encargadas de conectar la vista proporcionada al usuario con el modelo de datos, siendo el responsable de realizar todos los cálculos, operaciones, etc.

Cada clase está conectada con su correspondiente vista, y en ella se encuentran las funciones más destacadas para el procesamiento de los datos.

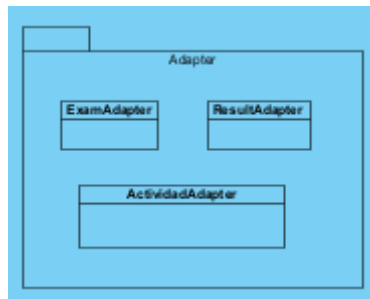


Imagen 6. Subsistema Adapter

El adapter es el encargado de presentar en los RecyclerView de las respectivas clases, los diferentes elementos contenidos en la base de datos, dándoles un formato y colocación determinado en base a un fichero XML asociado a él.

El subsistema final completo es el siguiente:



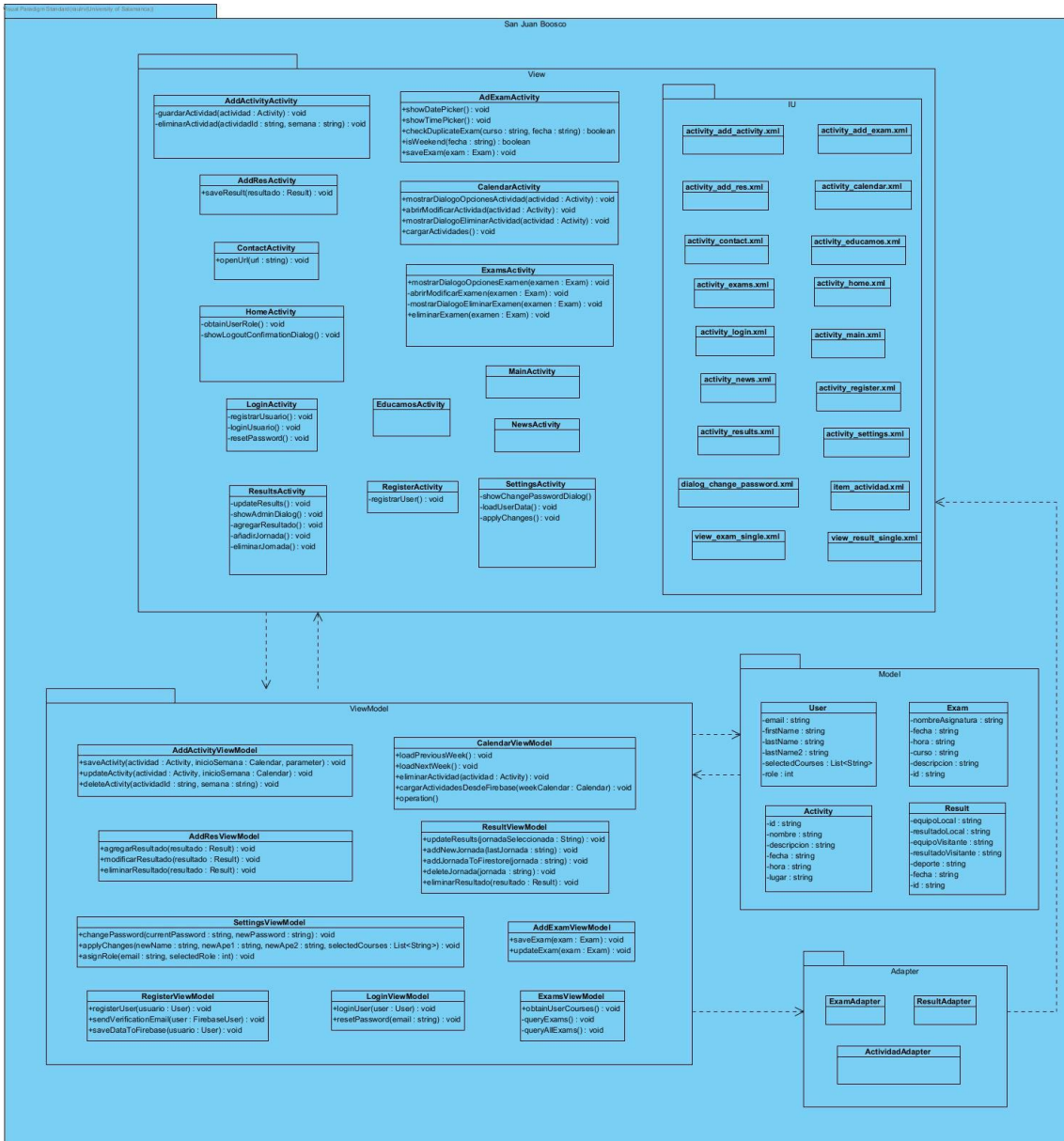


Imagen 7. Subsistema completo

## 4. REALIZACIÓN DE LOS CASOS DE USO

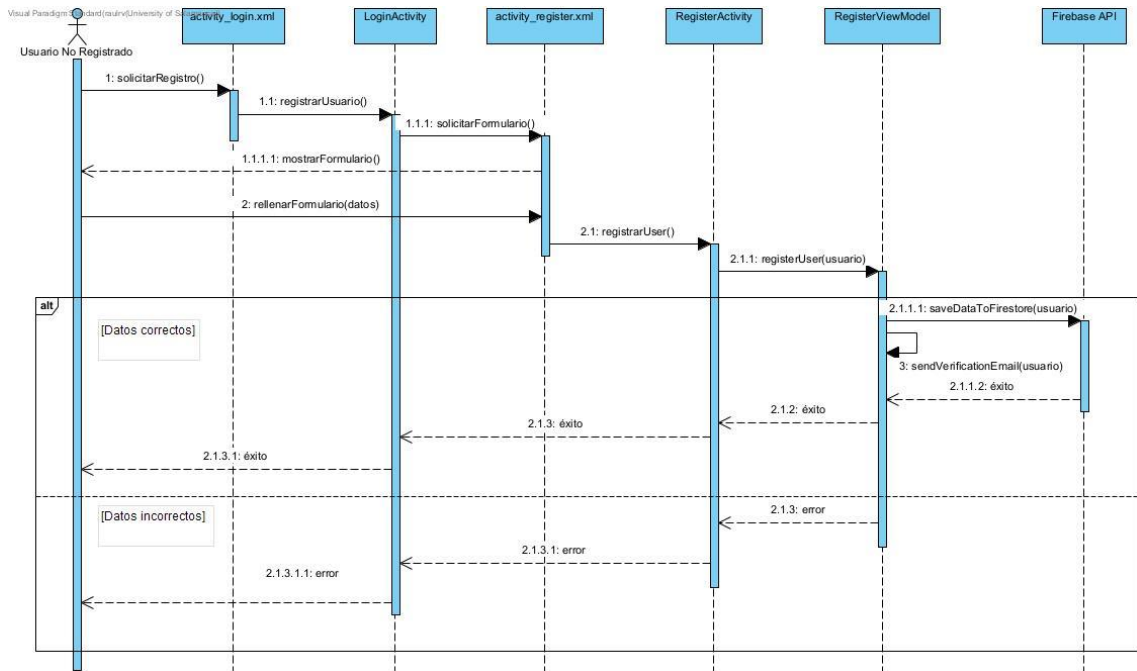


Imagen 8. UC-001 Registrarse

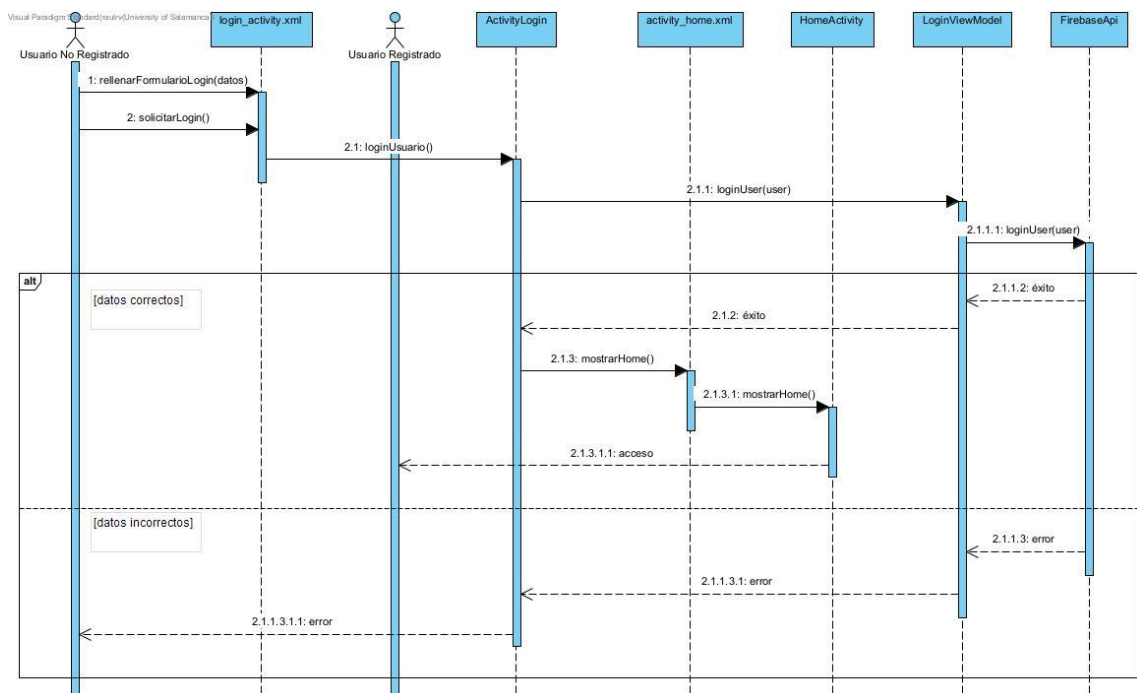


Imagen 9. UC-002 Iniciar sesion

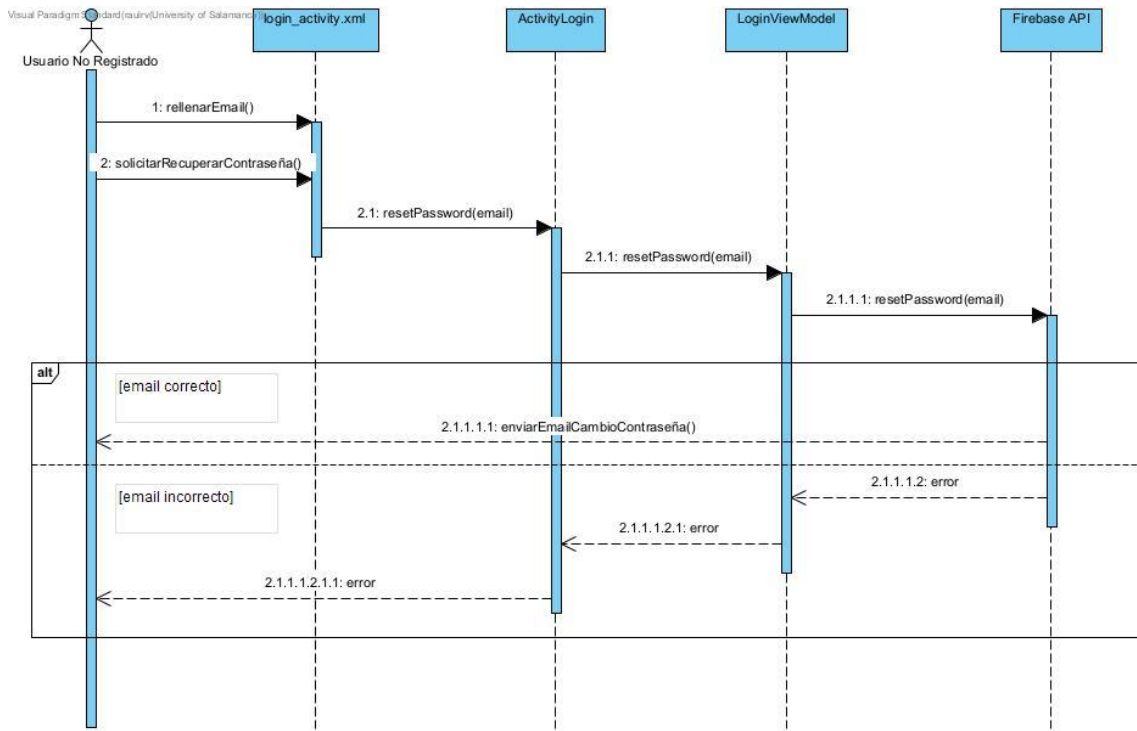


Imagen 10. UC-003 Recuperar contraseña

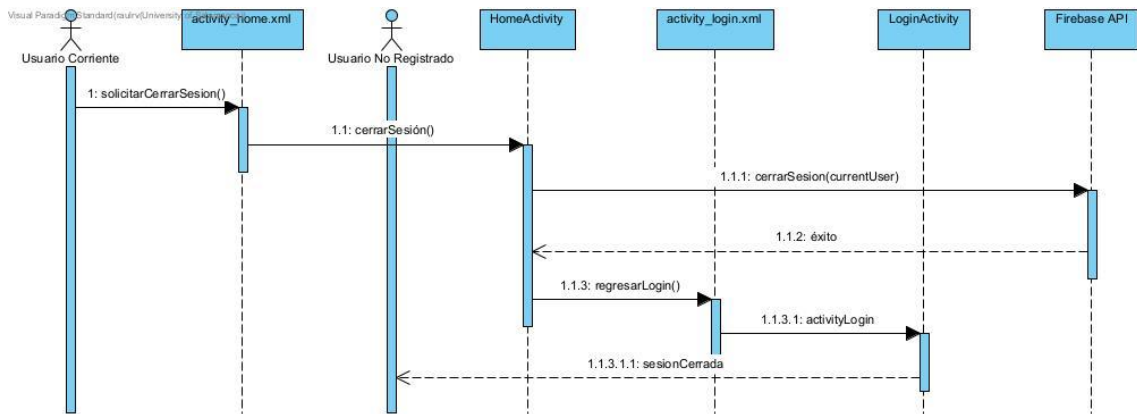


Imagen 11. UC-004 Cerrar sesión

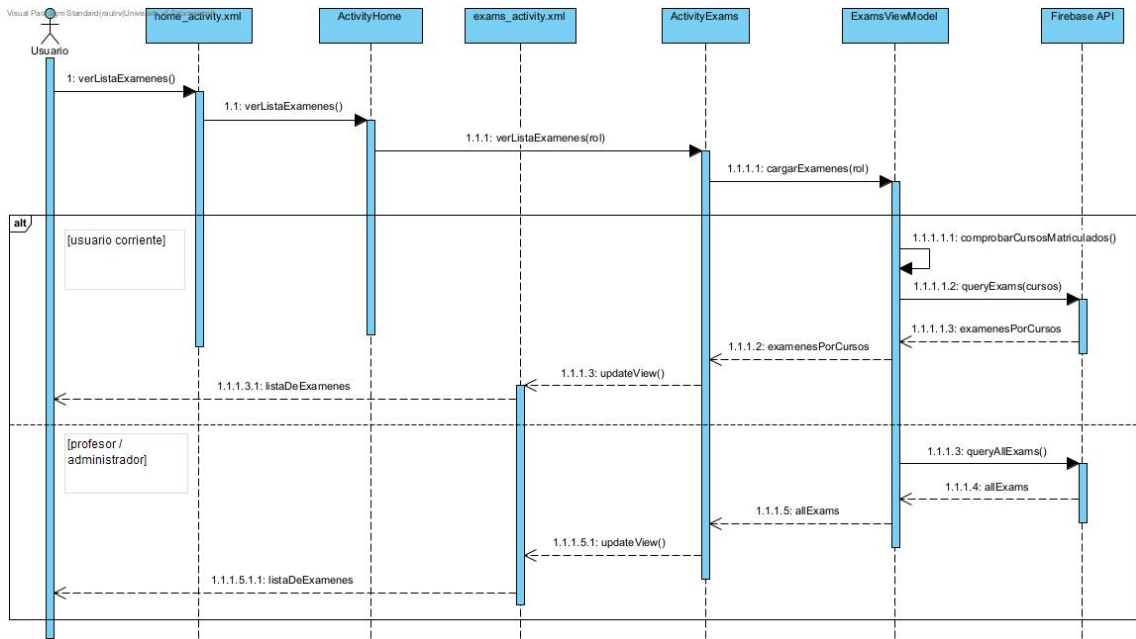


Imagen 12. UC-005 Ver lista de exámenes

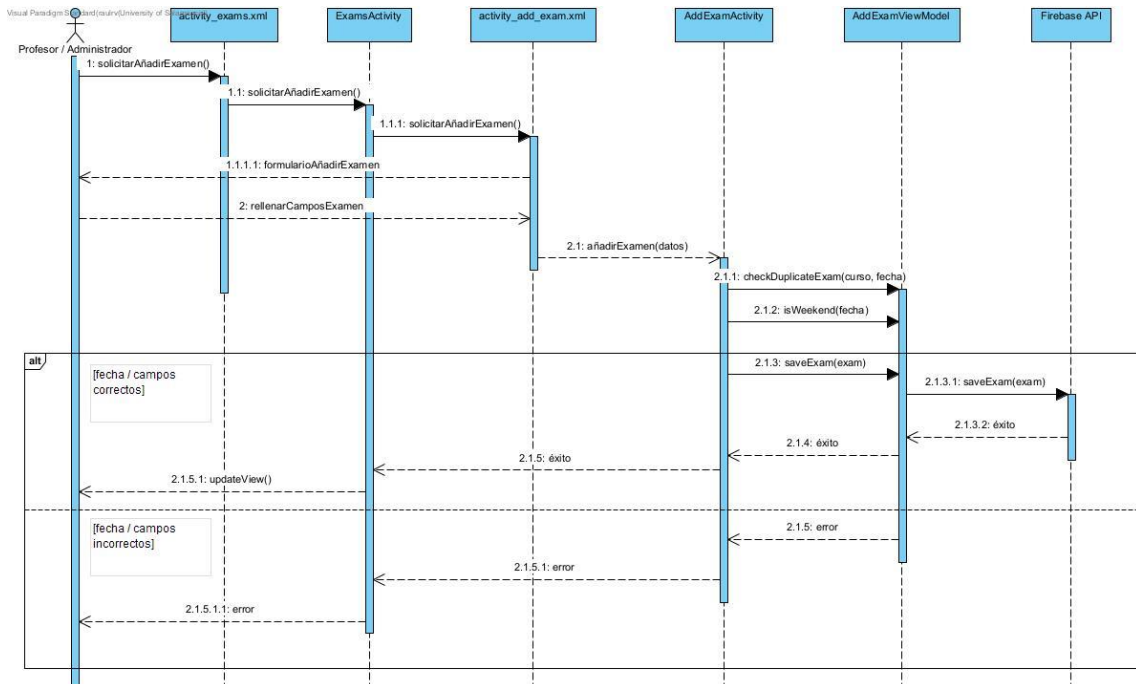


Imagen 13. UC-006 Añadir examen

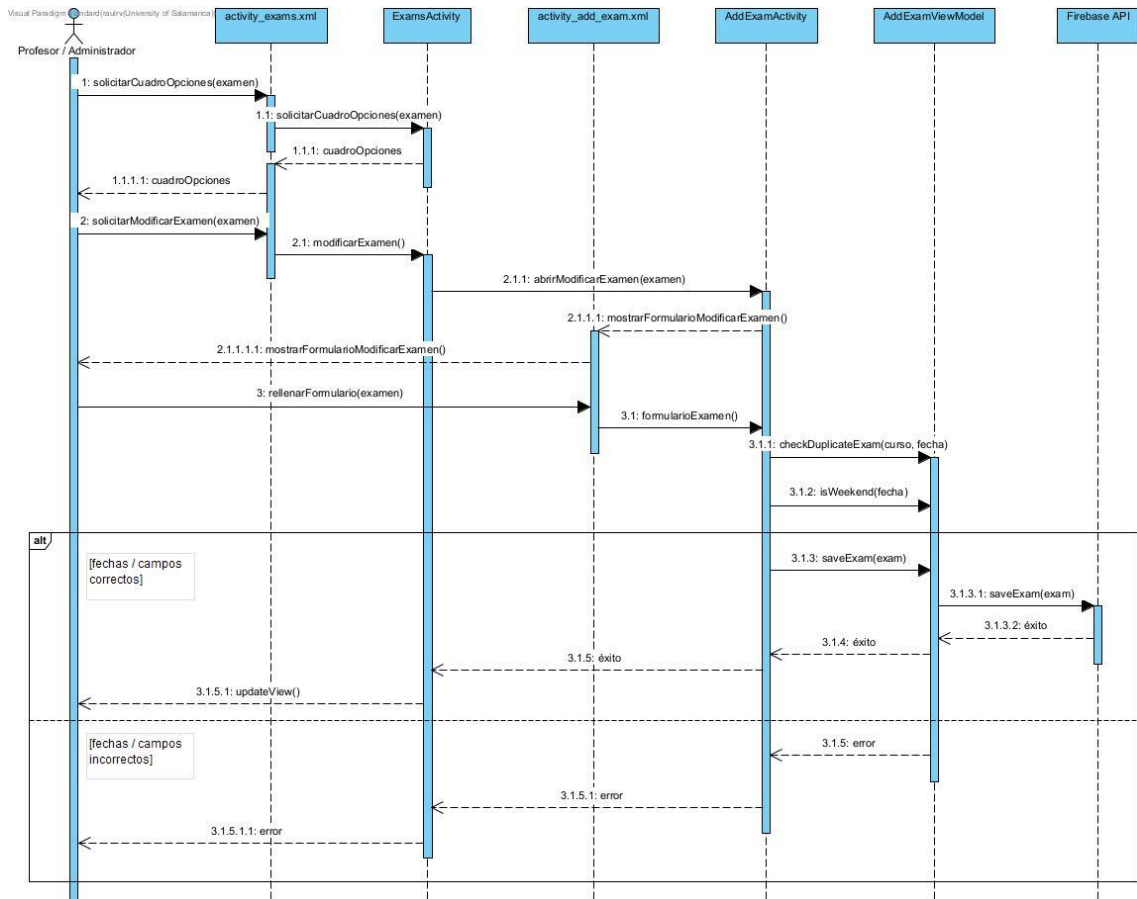


Imagen 14. UC-007 Modificar examen

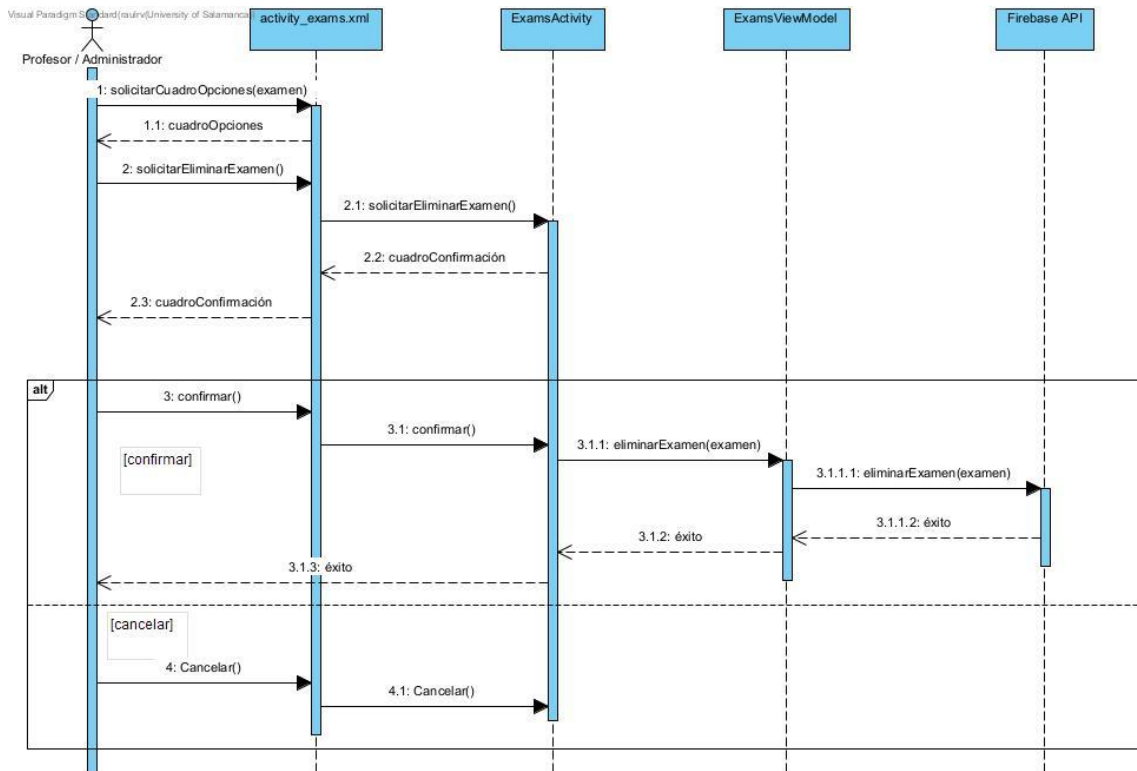


Imagen 15. UC-008 Eliminar examen

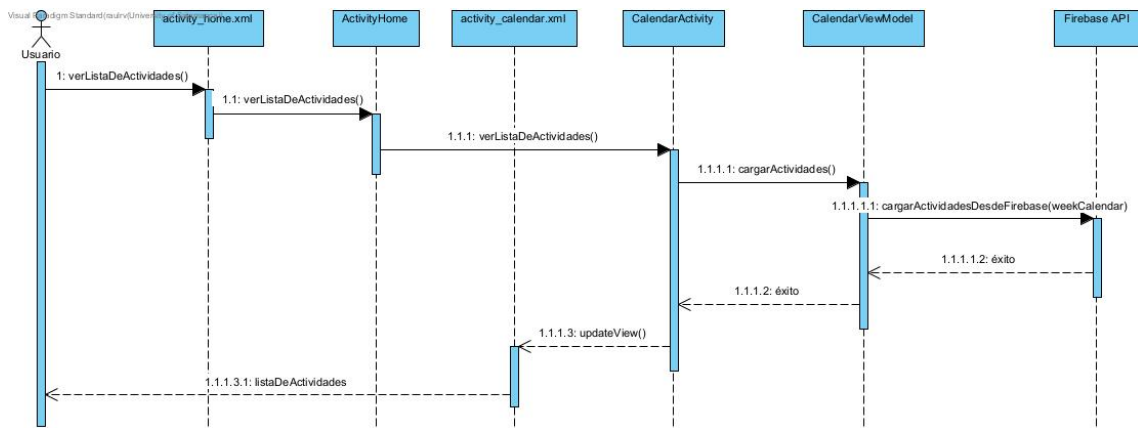


Imagen 16. UC-009 Ver calendario de actividades

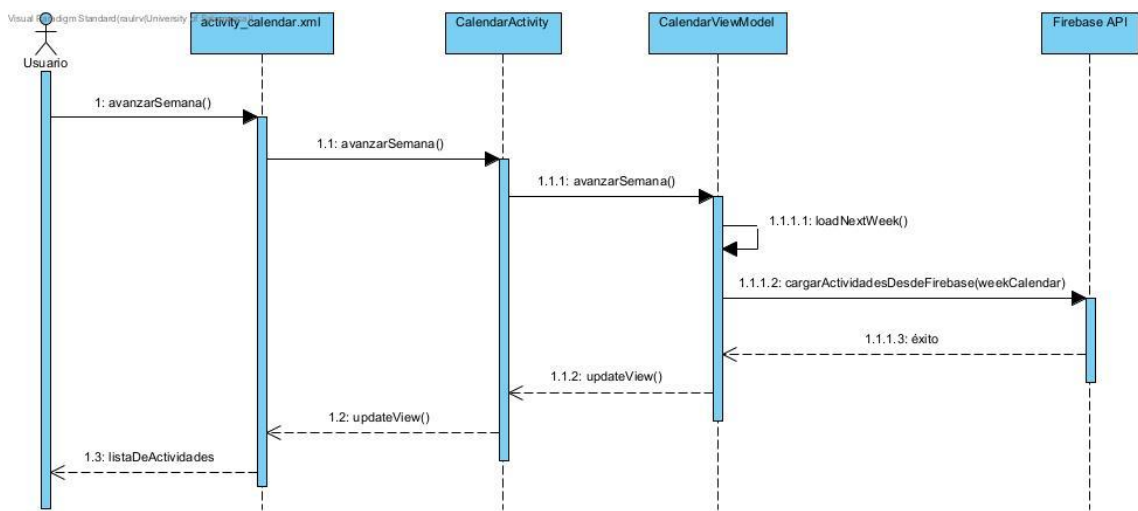


Imagen 17. UC-010 Avanzar semana en el calendario

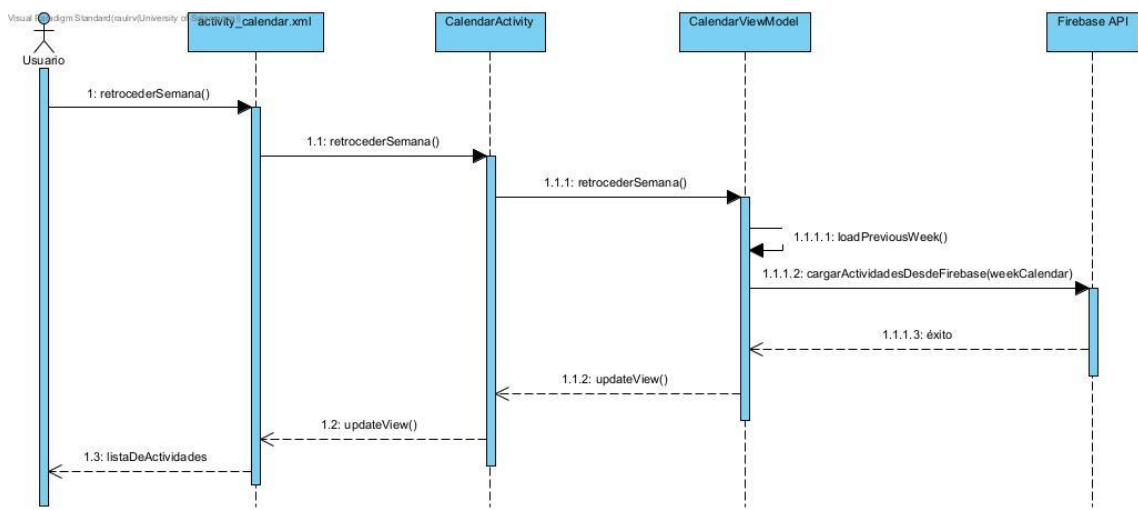


Imagen 18. UC-011 Ir semana atrás en el calendario

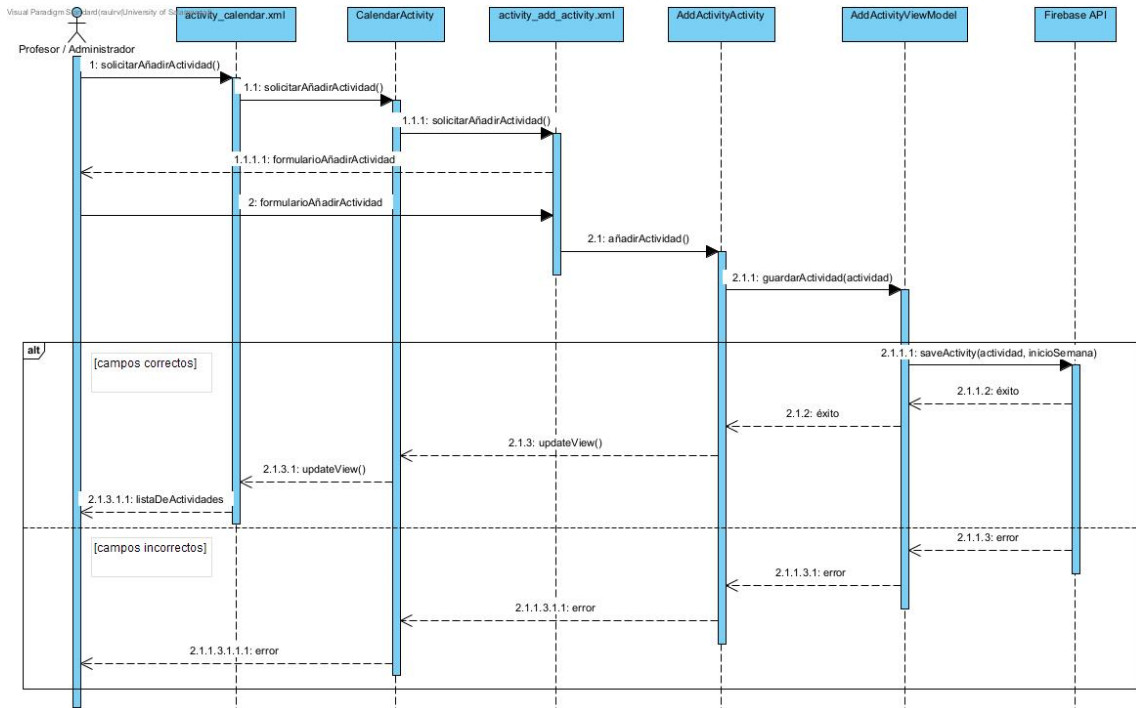


Imagen 19. UC-012 Añadir actividad

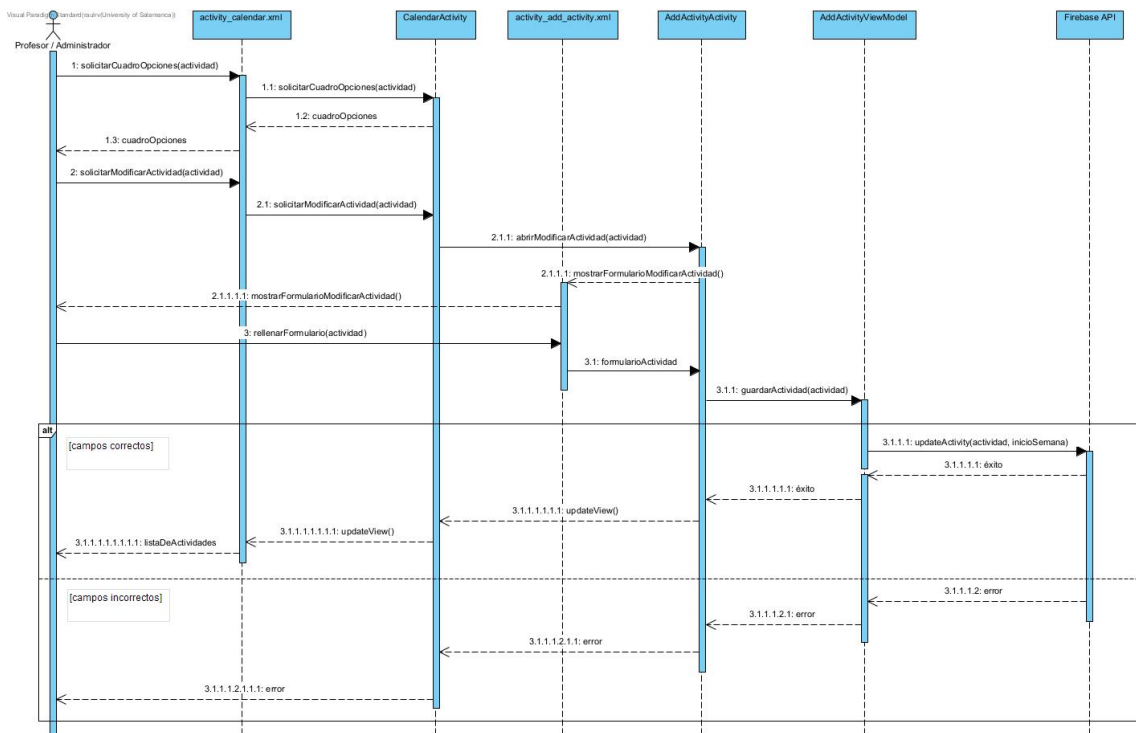


Imagen 20. UC-013 Modificar actividad

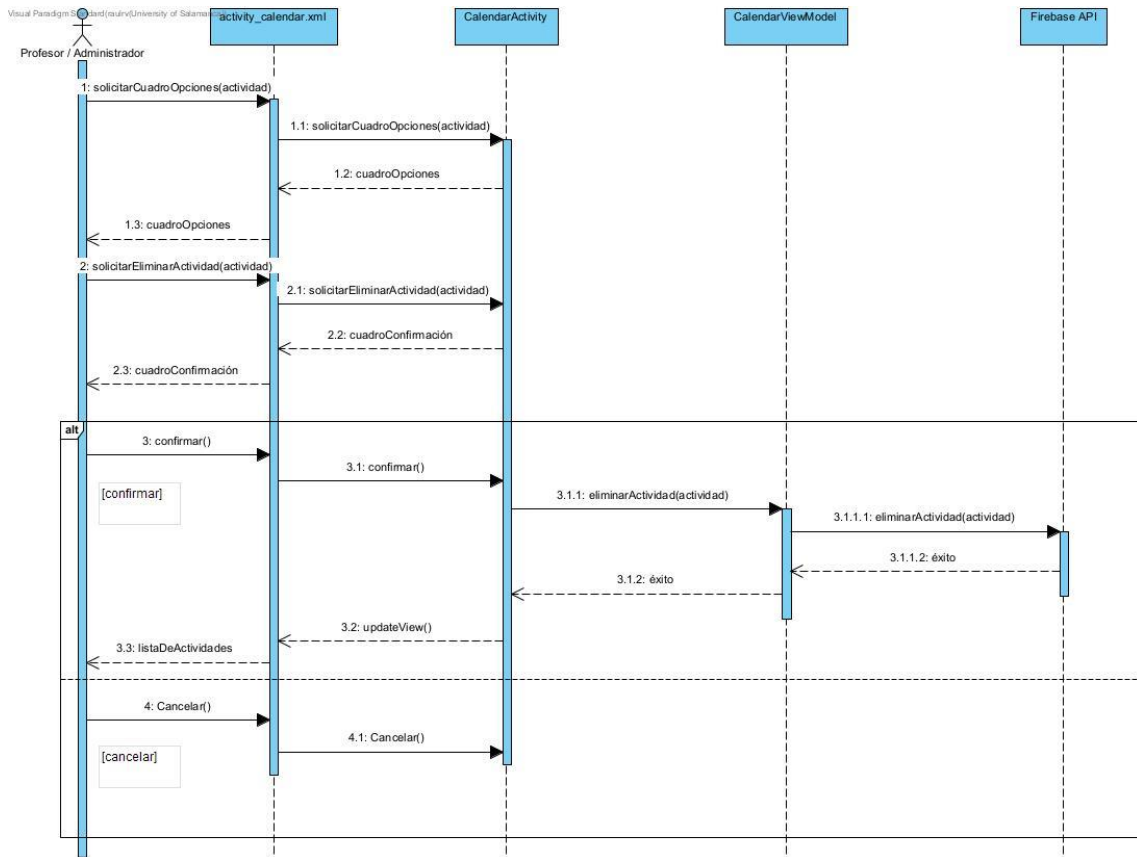


Imagen 21. UC-014 Eliminar actividad

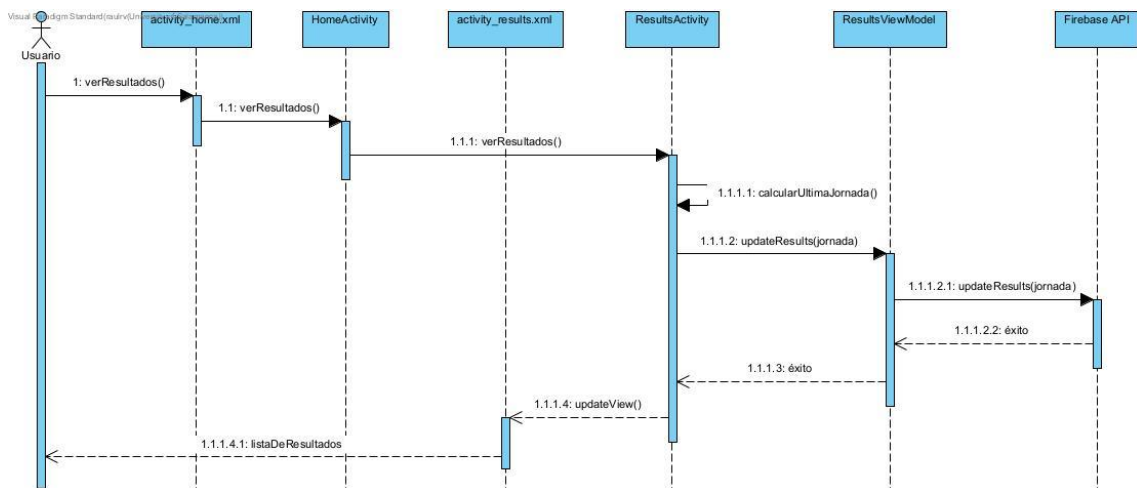


Imagen 22. UC-015 Ver resultados



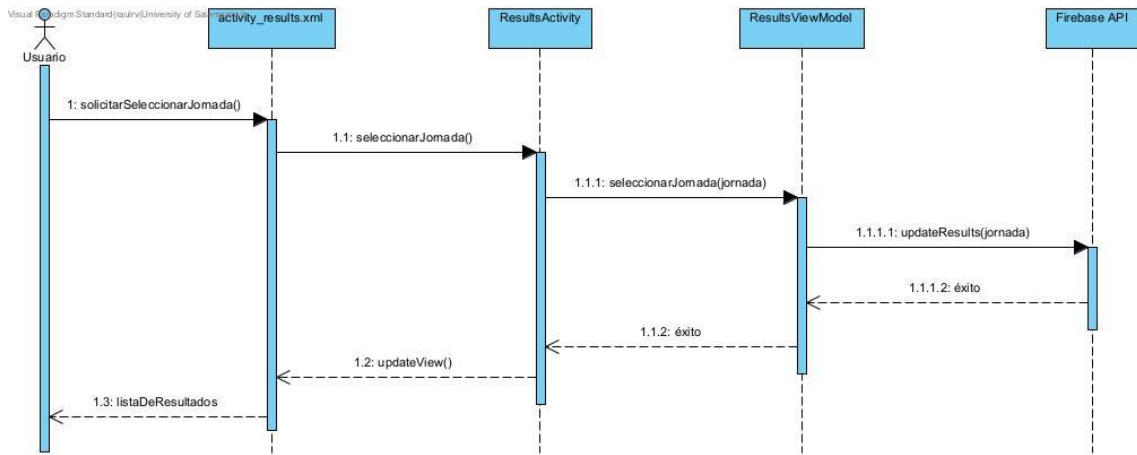


Imagen 23. UC-016 Seleccionar jornada

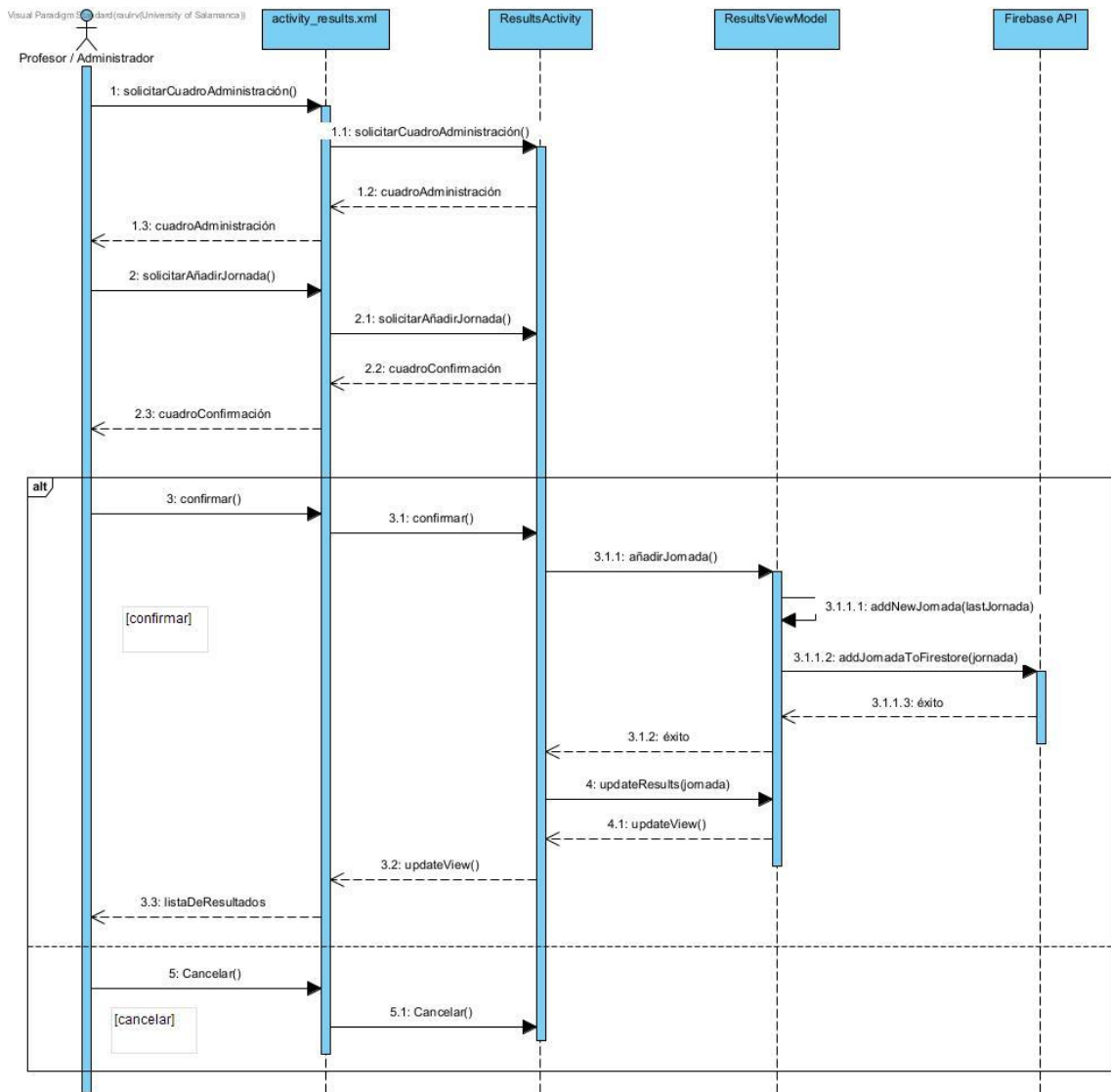


Imagen 24. UC-017 Añadir jornada

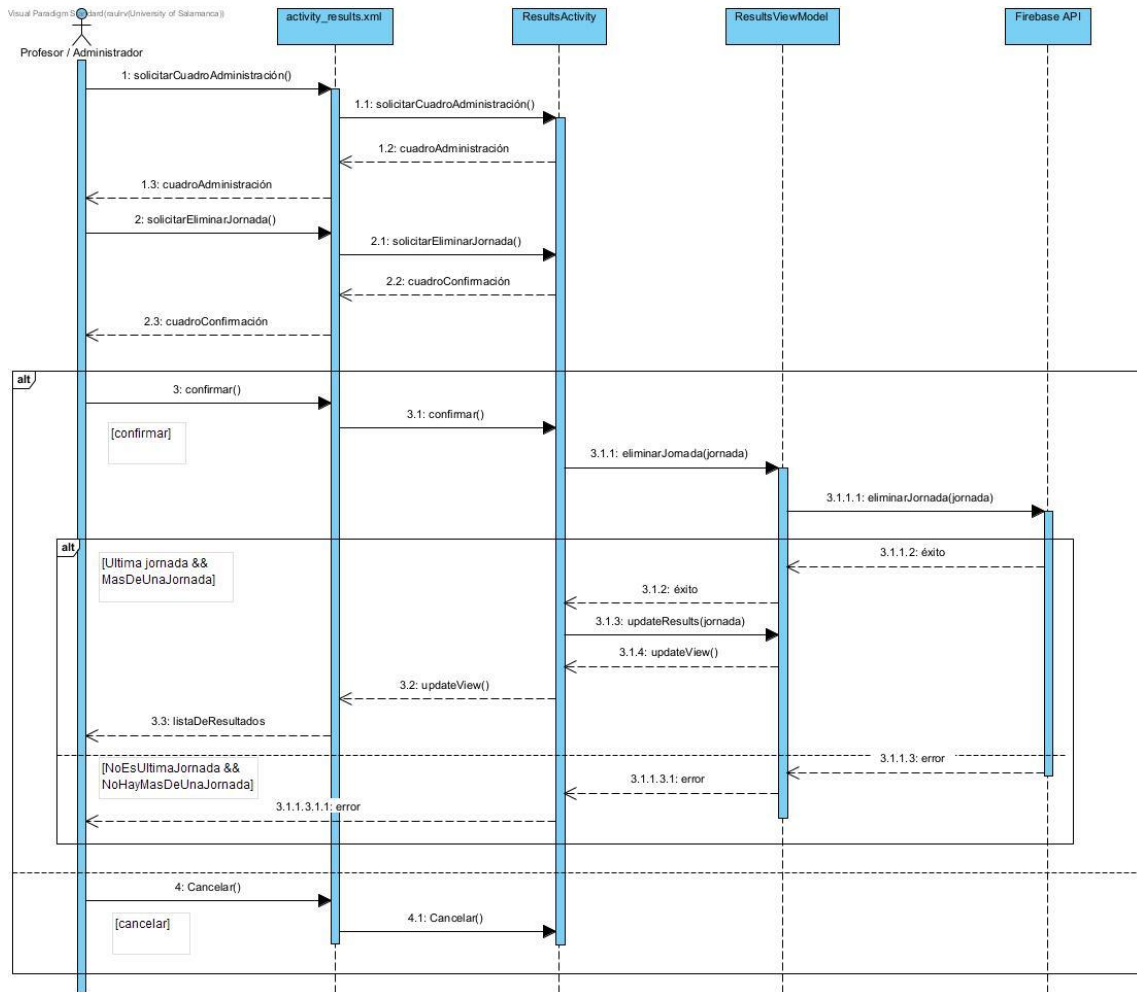


Imagen 25. UC-018 Eliminar jornada

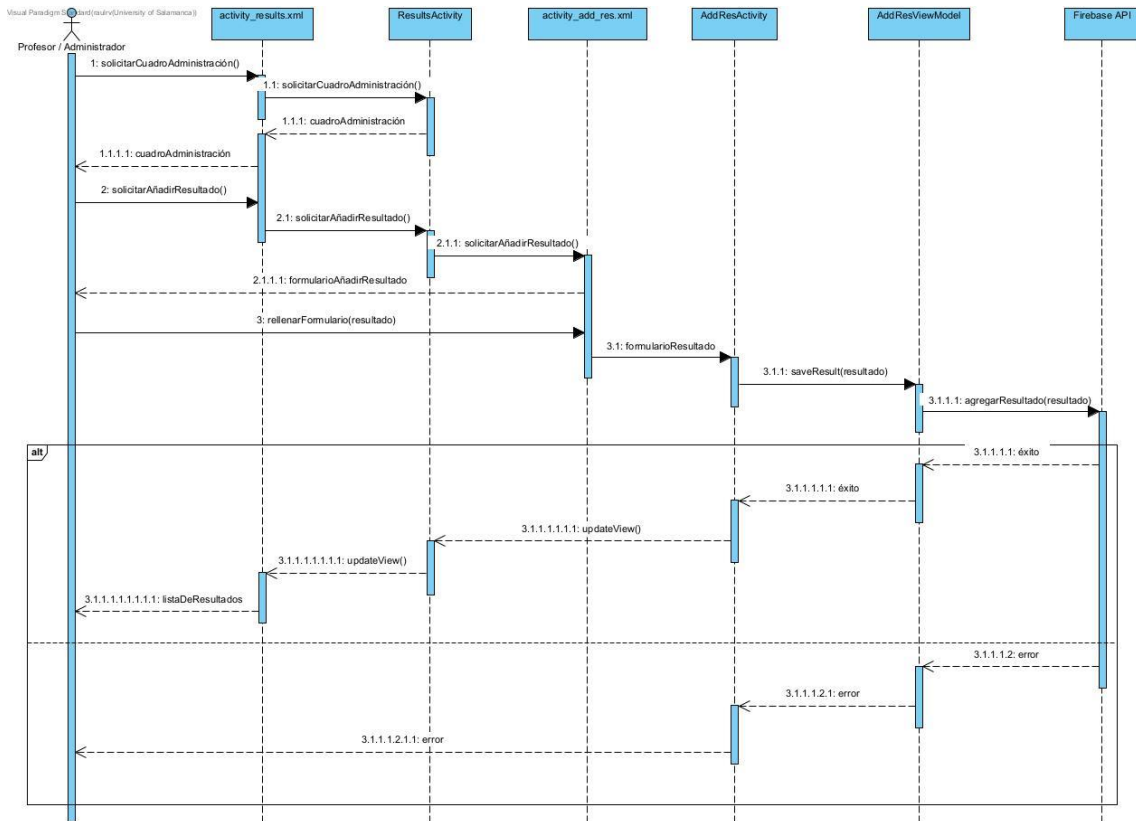


Imagen 26. UC-019 Añadir resultado

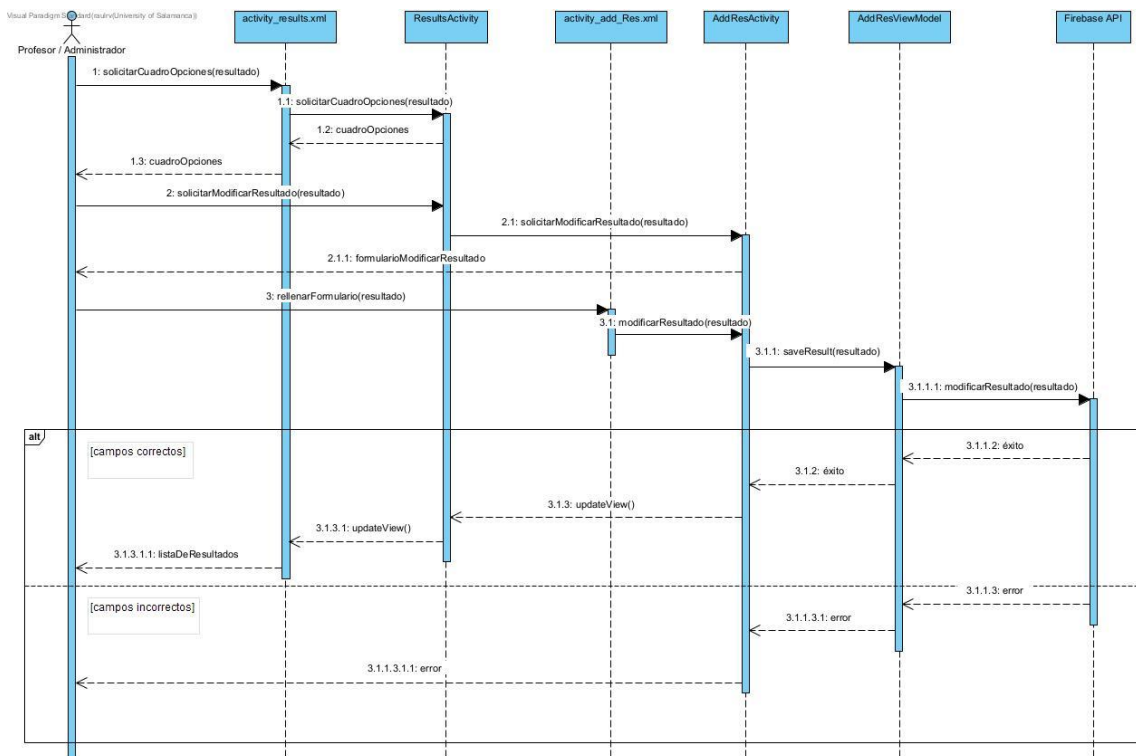


Imagen 27. UC-020 Modificar resultado

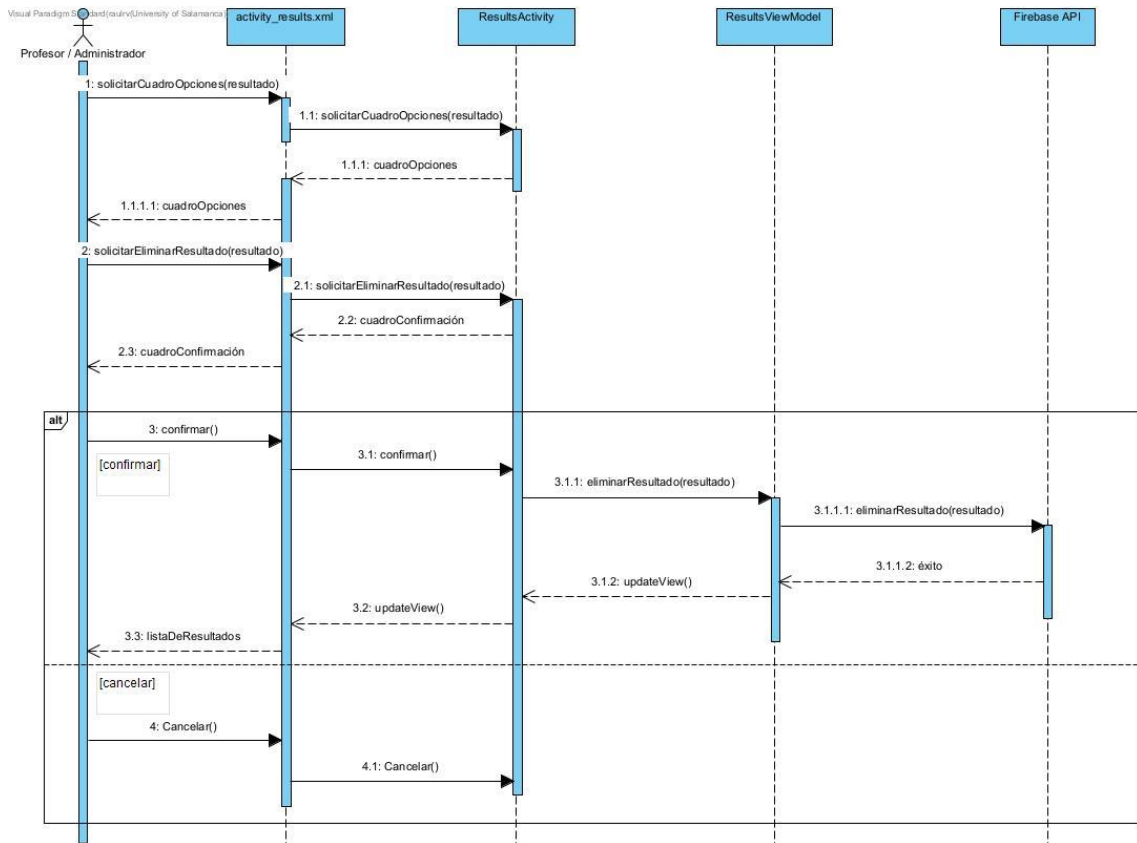


Imagen 28. UC-021 Eliminar resultado

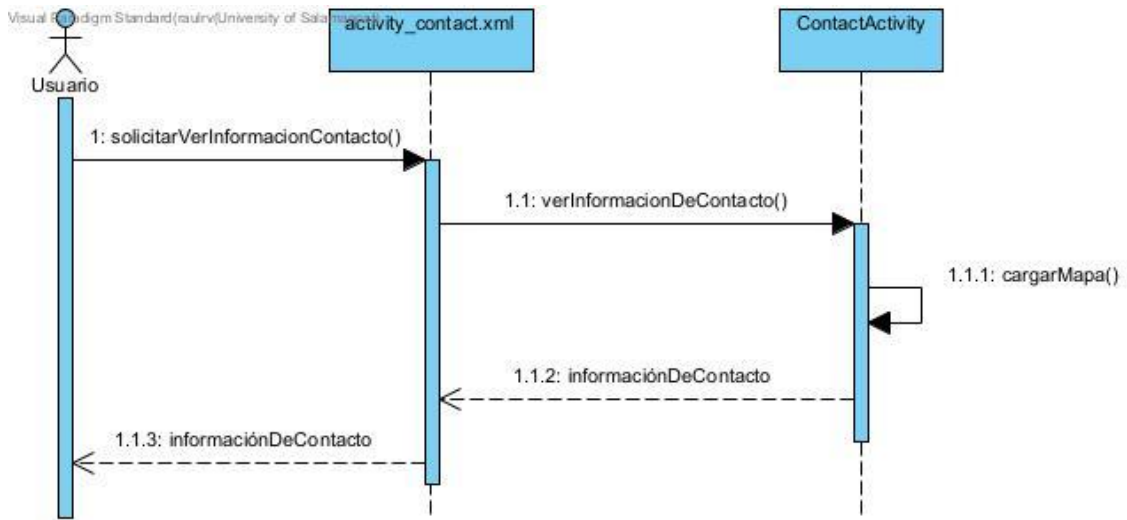


Imagen 29. UC-022 Ver información de contacto

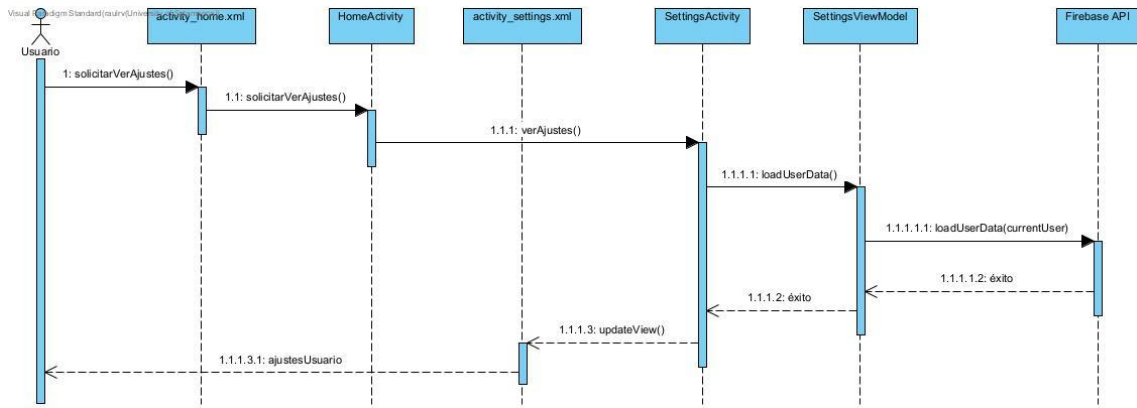


Imagen 30. UC-023 Ver ajustes

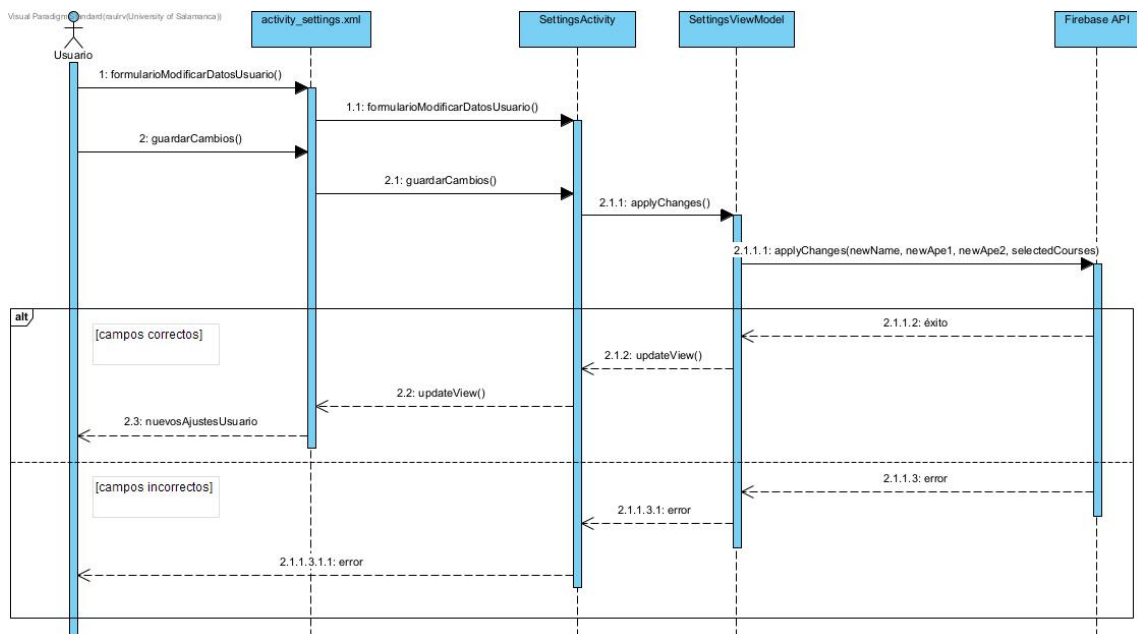


Imagen 31. UC-024 Modificar datos del usuario

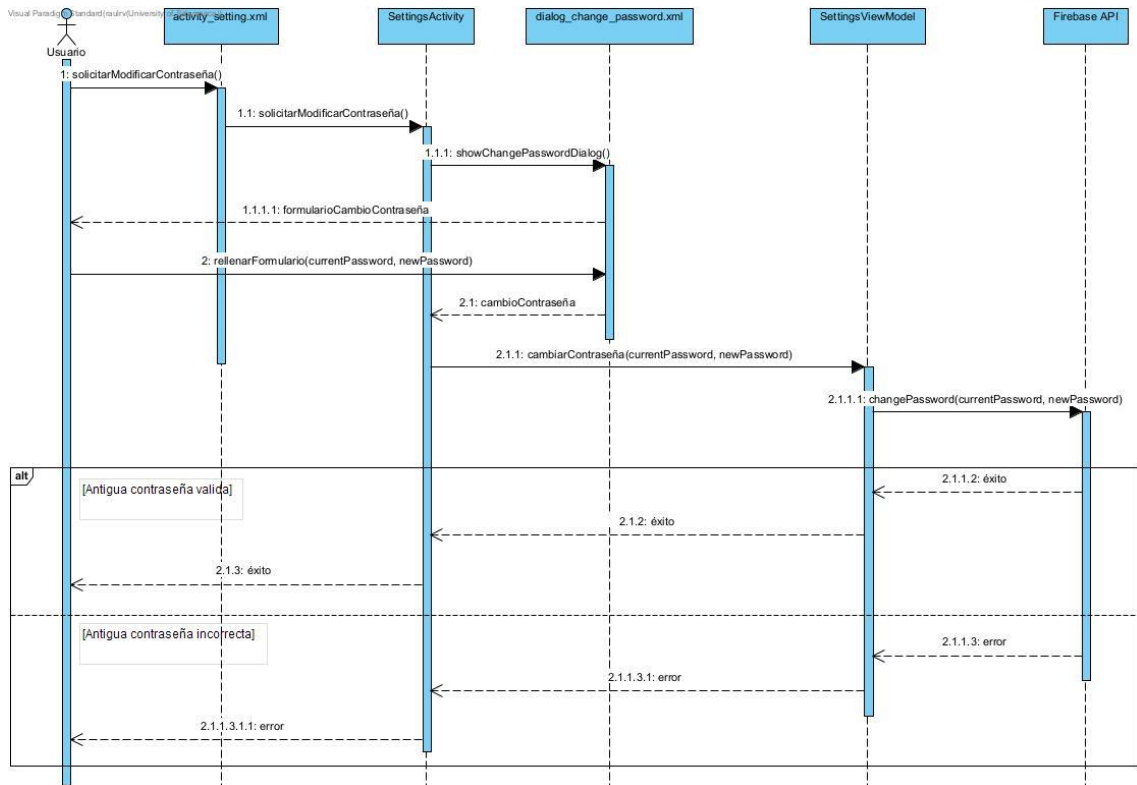


Imagen 32. UC-025 Modificar contraseña

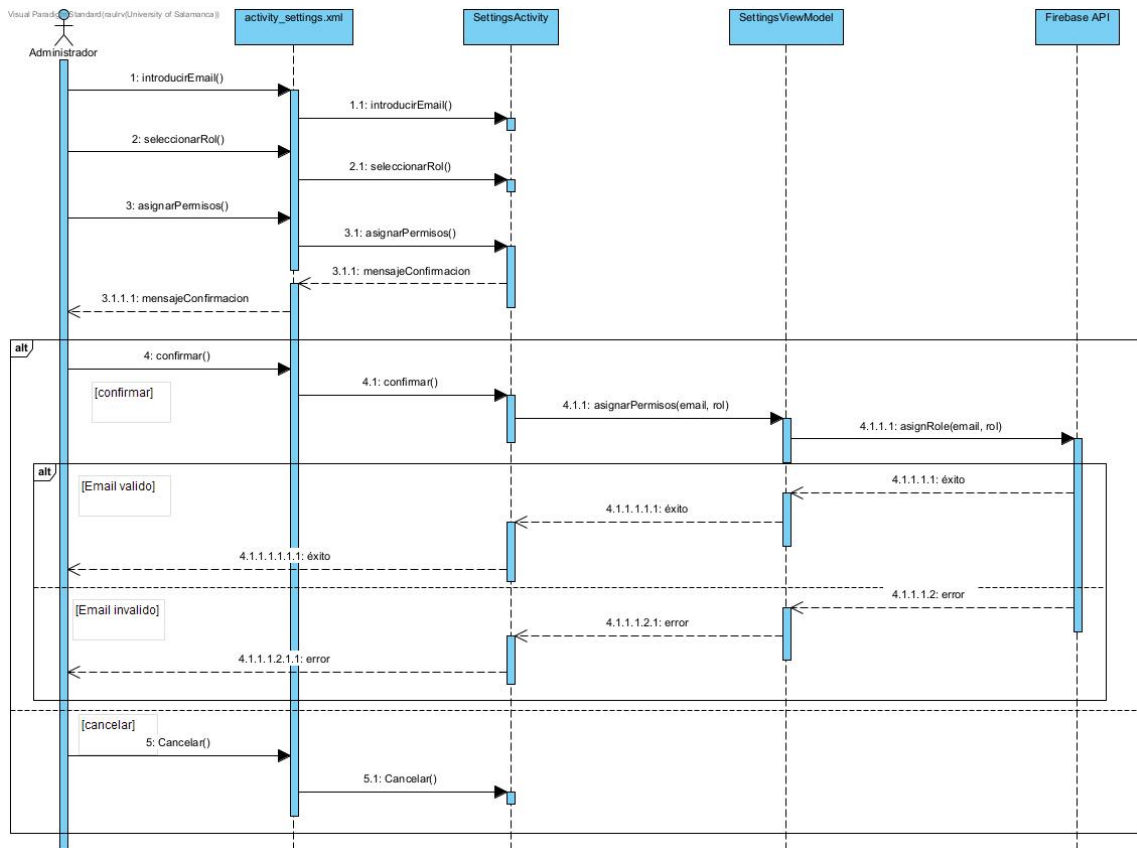


Imagen 33. UC-026 Asignar permisos a usuario

## 5. DISEÑO DE LA BASE DE DATOS

Para nuestra aplicación hemos utilizado la base de datos que nos ofrece Google, conocida como Firebase. Esta base de datos es muy potente y poderosa ya que cuenta con una interfaz muy sencilla y visual, además de herramientas propias como su API, que nos permite acceder a los datos, añadir, modificar y eliminar de manera relativamente sencilla desde nuestro entorno. Además ofrece gran cantidad de servicios como pueden ser Firebase Auth, utilizado para registrar usuarios de forma segura, con diferentes métodos para llevar a cabo el registro y logeo de éstos; Firebase Analytics, para llevar a cabo un seguimiento de las estadísticas de nuestro proyecto, los usuarios activos, el pico de usuarios, etc; Firebase Storage, para el almacenamiento de ficheros, sirviendo de una especie de disco duro, donde podremos acceder desde nuestro entorno a los ficheros que se encuentren allí almacenados.

En nuestro caso, lo que mayormente hemos utilizado ha sido Firebase Firestore, la propia base de datos, la cual nos permite almacenar todo lo relacionado con la aplicación en colecciones, las cuales son accedidas cuando consideremos necesario.

En cuanto a las colecciones, la división la he realizado en base a los diferentes bloques del sistema, por lo que tendremos nuestra base de datos fraccionada en usuarios, exámenes, actividades y resultados.

En primer lugar, comenzando por los usuarios, dentro de esta colección se encuentran todos los usuarios registrados en el sistema, ya que además de registrarse en Firebase Auth, se añadirán a la base de datos, donde se almacenarán todos los datos (sin la contraseña por motivos de seguridad), añadiéndole un rol por defecto de usuario corriente. Esto ha sido creado para poder modificar los datos de cada usuario y asignar permisos a usuarios específicos, como los profesores del centro y los administradores encargados de la gestión.

En cuanto a los exámenes, dentro de esta colección se van a almacenar todos los exámenes, con los campos específicos de cada uno, cada uno de ellos con un id específico, el cual nos va a permitir poder modificarlos y eliminarlos sin necesidad de eliminarlo y volver a añadirlo. Los exámenes tendrán campos específicos como el curso, el cual va a servir de filtro a la hora de presentarlos dentro de la aplicación.

Respecto a las actividades, tendremos las actividades organizadas dentro de cada una de la semana a la que corresponden, teniendo diferentes colecciones con las actividades que se encuentran contenidas dentro de cada semana.

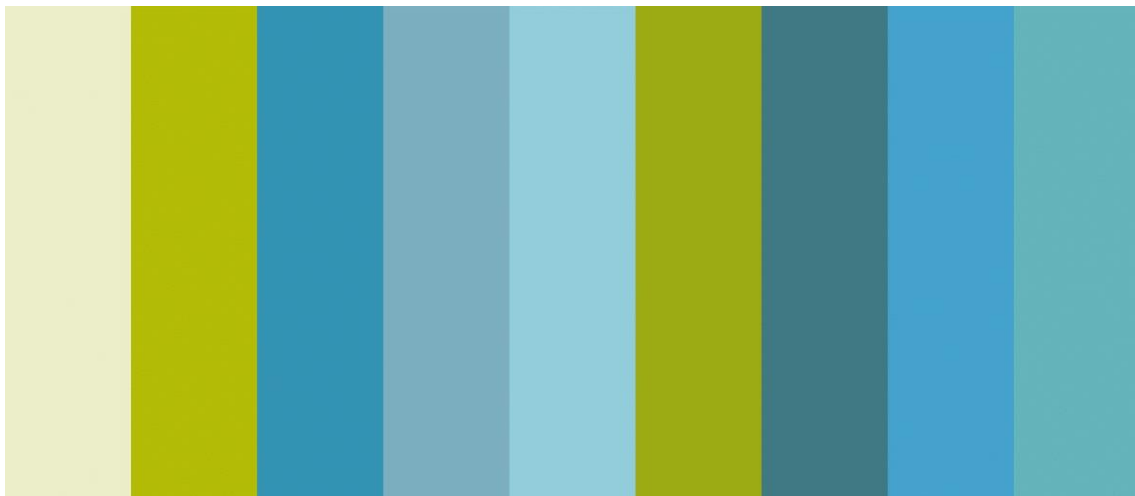
Por último, los resultados van a estar organizados por jornadas, por lo que almacenaremos por un lado las jornadas en la base de datos (solamente el nombre de la jornada, por ejemplo: "Jornada 1") y por otro lado los resultados dentro de cada jornada.

De esta manera y con una organización de la base de datos estructurada nos va a resultar más sencillo el acceso a los datos en caso de que la aplicación crezca exponencialmente.

## 6. DISEÑO DE LA INTERFAZ

Para el diseño de la interfaz de nuestra aplicación, se ha seguido el criterio de usabilidad, para que los usuarios tuviesen un manejo más sencillo de esta, sin que se les puedan presentar grandes problemas a la hora de navegar. Esto se ha llevado a cabo así debido a que el público de nuestra aplicación es un público con una media de edad mayor, quizá no todos tengan amplios conocimientos de informática, por lo que se deberá seguir una simplicidad para facilitar el trabajo a todos estos usuarios.

En cuanto a los colores utilizados, son los propios de la página web del centro, San Juan Bosco, con tonos azules claros y marrones, además del blanco. A partir de aquí, hemos generado una paleta con los colores más utilizados y las transiciones de colores, teniendo así un mayor abanico de colores, todos ellos relacionados entre sí y con un significado específico.



*Imagen 34. Paleta de colores*

Los primeros prototipos de la interfaz los realicé en papel, con diseños sencillos y ubicando los elementos más o menos donde quería que fuesen colocados. Esto lo llevé directamente a la aplicación debido a la sencillez del mismo, y obtuvimos un primer diseño bastante sencillo pero visual, donde estaba claro que había que hacer cambios.





Imagen 35. Primera Pantalla de Login



*Imagen 36. Primera pantalla principal*

Tras probarlo con algún usuario, y pedir opiniones sobre el diseño, en numerosos casos se propusieron cambios, ya que era una interfaz muy poco visual y poco expresiva, por lo que se llevó a cabo un proceso de rediseño completo de la interfaz, modificando, sobre todo, los ficheros con código XML, donde se encuentra la parte visual de nuestra aplicación.



Imagen 37. Diseño final de Login



Imagen 38. Diseño final pantalla principal

Por ello, a partir de un primer boceto a papel, llevado a cabo a la aplicación, se ha ido modificando la interfaz en base a las peticiones del usuario, dejando una interfaz final mucho más visual y bonita, empleando los colores que identifican al centro.

## 7. VISTA DE DESPLIEGUE

En el siguiente diagrama se va a ver la distribución física del hardware y software en nuestro sistema, así como las conexiones entre los mismo.

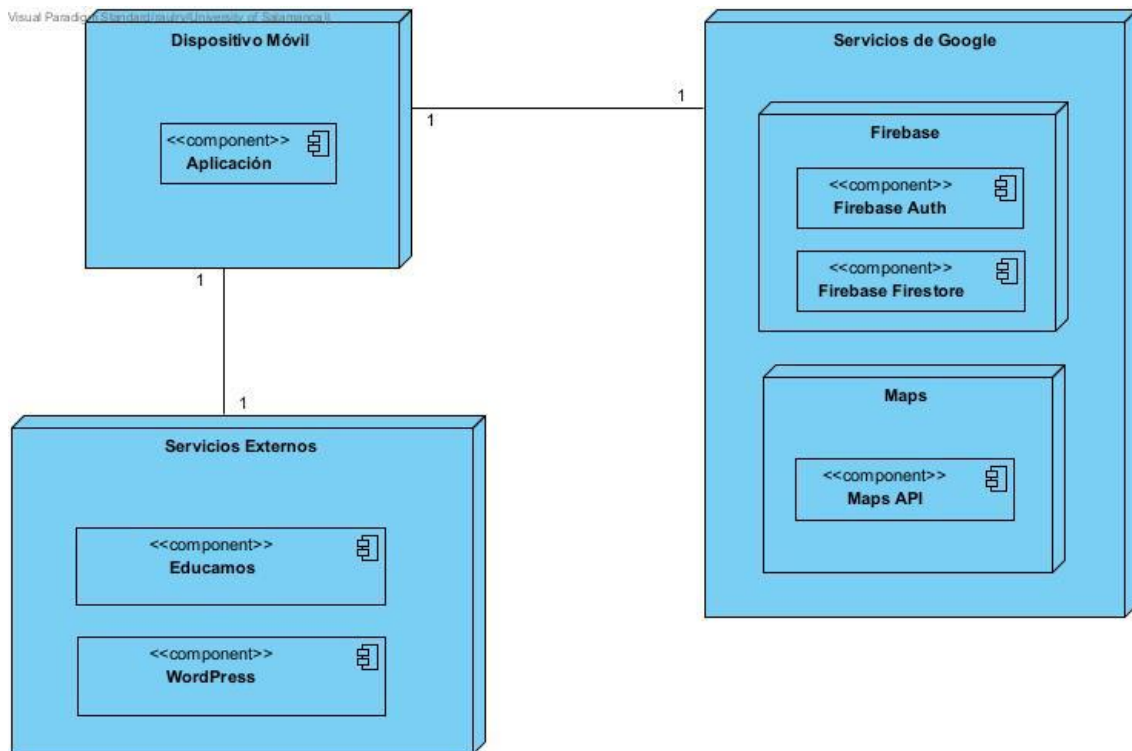


Imagen 39. Diagrama de despliegue

El diagrama se encuentra compuesto por tres nodos:

- Dispositivo móvil: hace referencia al dispositivo empleado por el usuario para el lanzamiento de nuestra aplicación.
- Servicios de Google: divididos a su vez en Firebase (nuestra base de datos utilizada para almacenar todos los datos de la aplicación) y Maps (el servicio de mapas de Google).
- Servicios externos: son los servicios que hemos implementado dentro de nuestra aplicación, pero que son ajenos a ésta. En este caso es la página web de noticias del centro, alojada en WordPress, y la página web de Educamos, donde se pueden ver las calificaciones de cada usuario del centro.

## 8. REFERENCIAS

*Documentación Cliente-Servidor.* (2023). Obtenido de <https://es.wikipedia.org/wiki/Cliente-servidor>

*Documentación Firebase.* (2023). Obtenido de <https://firebase.google.com/docs/guides?hl=es-419>

*Documentación MVVM.* (2023). Obtenido de <https://learn.microsoft.com/es-es/dotnet/architecture/maui/mvvm>

Moreno García M. N., Z. D. (2021). Transparencias de Ingeniería del Software I.

Moreno García M. N., Z. D. (2022). Transparencias de Gestión de Proyectos.

Moreno García M. N., Z. D.-A. (2022). Transparencias de Ingeniería del Software II.