

APLICACIÓN MÓVIL PARA LA COMUNICACIÓN ENTRE CENTROS EDUCATIVOS Y FAMILIAS

MEMORIA FINAL



VNiVERSIDAD
D SALAMANCA

Septiembre de 2023

Trabajo de fin de Grado

Tutor:

María Navarro Cáceres

Alumno:

Raúl Rodríguez Villanueva

Certificado del tutor

D./Dña. María Navarro Cáceres, profesor/a del Departamento de Informática y Automática de la Universidad de Salamanca,

HACE/N CONSTAR:

Que el trabajo titulado “Aplicación para la comunicación entre centros educativos y familias”, que se presenta, ha sido realizado por Raúl Rodríguez Villanueva, con DNI 45690196Y y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado en Ingeniería Informática en esta Universidad.

En Salamanca a 29 de agosto de 2023.

Resumen

En la actualidad, la tecnología se encuentra presente en la mayoría de los centros escolares, tanto para los alumnos, como para los profesores y padres. Esto es una buena práctica ya que están conectados sin necesidad de estar allí presencialmente. Si bien, podemos observar como la mayoría de los centros cuentan con portales web donde publican notas, noticias del centro, etc.

El trabajo desarrollado consiste en una aplicación para la comunicación entre los centros educativos y las familias, donde se va a disponer de diversos apartados para organizar todo lo relacionado con el centro. La aplicación servirá como una "agenda" o "bloc de notas", donde los profesores irán colgando fechas de actividades y fechas de exámenes, para que los padres desde sus casas puedan estar al corriente de todo. También cuenta con una sección donde se recogerán los resultados de las actividades deportivas extraescolares organizadas por jornadas.

El sistema contará con un mecanismo de registro de usuarios, donde estos deberán registrarse y matricularse en el/los curso/s en los que se encuentra/n su/s hijo/s. Esto va a servir para poder ver solamente lo necesario, y no lo del resto de alumnos.

Los usuarios van a estar organizados por roles, pudiendo diferenciar entre padre/madre/tutor de profesores y administradores. Estos dos últimos contarán con ciertos privilegios, como poder añadir, modificar o eliminar todo lo relacionado con los exámenes, actividades y resultados, entre otras funciones.

Nuestra aplicación también va a proporcionar un acceso directo desde ella a páginas web propias del centro, como el portal de Educamos, o el WordPress de noticias, además de proporcionar información sobre el contacto del centro y sus redes sociales personales.

Para el desarrollo del proyecto se va a seguir la metodología del Proceso Unificado, siguiendo un ciclo de vida, y aplicando el patrón MVVM. Todo va a estar desarrollado en los anexos de la documentación.

Palabras clave: Proceso Unificado, Aplicaciones CRUD, Comunicación, Roles.

Summary

Nowadays, technology is present in most schools, both for students, teachers and parents. This is a good practice since they are connected without the need to be there in person. However, we can observe that most schools have web portals where they publish notes, school news, etc.

The work developed consists of an application for communication between schools and families, which will have several sections to organize everything related to the center. The application will serve as an "agenda" or "notepad", where teachers will be posting dates of activities and exam dates, so that parents can be aware of everything from their homes. It also has a section where the results of extracurricular sports activities organized by days will be collected.

The system will have a user registration mechanism, where users will have to register and enroll in the course/s in which their child/children are enrolled. This will be useful to be able to see only what is necessary, and not the rest of the students.

Users will be organized by roles, being able to differentiate between parent/guardian and teachers or administrators. The latter two will have certain privileges, such as being able to add, modify or delete everything related to exams, activities and results, among other functions.

Our application will also provide direct access from it to the center's own web pages, such as the Educamos portal, or the WordPress news, as well as providing information about the center's contact and personal social networks.

For the development of the project, we will follow the Unified Process methodology, following a life cycle, and applying the MVVM pattern. Everything will be developed in the annexes of the documentation.

Keywords: Unified Process, CRUD applications, Communication, Roles.

Tabla de contenido

1.	INTRODUCCIÓN.....	14
2.	OBJETIVOS.....	15
2.1.	OBJETIVOS TÉCNICOS.....	15
2.2.	OBJETIVOS PERSONALES.....	16
3.	CONCEPTOS TEÓRICOS.....	16
3.1.	APLICACIONES CRUD.....	16
3.2.	APLICACIONES ANDROID.....	17
3.3.	APLICACIONES EDUCATIVAS DE COMUNICACIÓN.....	17
3.4.	FIREBASE.....	19
4.	TÉCNICAS Y HERRAMIENTAS.....	20
4.1.	ENTORNOS DE DESARROLLO.....	20
4.1.1.	Android Studio.....	20
4.2.	BASE DE DATOS.....	20
4.2.1.	Firebase Firestore.....	20
4.3.	LENGUAJES DE PROGRAMACIÓN.....	21
4.3.1.	Java.....	21
4.3.2.	XML.....	22
4.3.3.	UML.....	22
4.4.	HERRAMIENTAS DE CONTROL DE VERSIONES.....	23
4.4.1.	Git.....	23
4.5.	HERRAMIENTAS CASE.....	23
4.5.1.	Visual Paradigm.....	23
4.5.2.	EZEstimate.....	24
4.5.3.	Microsoft Project.....	24
4.5.4.	Javadoc.....	24
4.5.5.	Microsoft Word.....	24
5.	ASPECTOS RELEVANTES.....	25
5.1.	METODOLOGÍA.....	26
5.2.	ESTIMACIÓN DEL ESFUERZO.....	27
5.3.	PLANIFICACIÓN TEMPORAL.....	28
5.4.	ESPECIFICACIÓN DE REQUISITOS.....	29
5.5.	ANÁLISIS.....	33
5.5.1.	Modelo del Dominio.....	33
5.5.2.	Realización de casos de uso.....	34
5.5.3.	Propuesta de arquitectura.....	34

5.6.	DISEÑO	35
5.6.1.	Patrones Arquitectónicos	35
5.6.2.	SUBSISTEMAS DE DISEÑO.....	37
5.6.3.	REALIZACIÓN DE LOS CASOS DE USO	40
5.6.4.	DISEÑO DE LA BASE DE DATOS	41
5.6.5.	DISEÑO DE LA INTERFAZ.....	42
5.6.6.	VISTA DE DESPLIEGUE.....	44
5.7.	IMPLEMENTACIÓN	44
5.7.1.	Paquete Model	45
5.7.2.	Paquete View.....	46
5.7.3.	Paquete ViewModel	46
5.7.4.	Paquete Adapter.....	46
5.7.5.	Paquete Utils	46
5.8.	PRUEBAS.....	47
5.9.	FUNCIONALIDADES.....	47
5.9.1.	GESTIÓN DE USUARIOS	48
5.9.2.	GESTIÓN DE EXÁMENES	52
5.9.3.	GESTIÓN DE ACTIVIDADES.....	55
5.9.4.	GESTIÓN DE RESULTADOS	58
5.9.5.	FUNCIONES EXTERNAS	62
6.	CONCLUSIONES	64
7.	LÍNEAS FUTURAS	64
8.	Referencias	66

Figuras

Figura 1. Fases del Proceso Unificado	27
Figura 2. Ventana EZEstimate	28
Figura 3. Planificación temporal Inicio	29
Figura 4. Diagrama de casos de uso – Usuarios	32
Figura 5. Modelo del dominio - Diagrama de clases	33
Figura 6. Diagrama de secuencia - Iniciar Sesión	34
Figura 7. Propuesta de arquitectura	35
Figura 8. Patrón MVVM	36
Figura 9. Esquema Cliente-Servidor	37
Figura 10. Subsistema View	38
Figura 11. Subsistema Model	38
Figura 12. Subsistema ViewModel	39
Figura 13. Subsistema Adapter	39
Figura 14. Subsistema completo	40
Figura 15. Realización caso de uso – Diseño	41
Figura 16. Vista de despliegue	44
Figura 17. Pantalla Login	48
Figura 18. Pantalla Registro	49
Figura 19. Restablecer contraseña	50
Figura 20. Pantalla Ajustes	50
Figura 21. Cuadro cambio de contraseña	51
Figura 22. Pantalla Ajustes - Asignar Permisos	51
Figura 23. Pantalla principal	52
Figura 24. Pantalla calendario de exámenes	53
Figura 25. Pantalla Añadir Examen	53
Figura 26. Cuadro opciones examen	54
Figura 27. Pantalla Modificar Examen	55
Figura 28. Pantalla calendario de actividades	56
Figura 29. Pantalla Añadir Actividad	56
Figura 30. Cuadro Opciones de Actividad	57
Figura 31. Pantalla Modificar Actividad	58
Figura 32. Pantalla Resultados	59
Figura 33. Cuadro administración resultados	59
Figura 34. Pantalla Añadir Resultado	60
Figura 35. Cuadro Opciones de Resultados	61
Figura 36. Pantalla Modificar Resultado	61
Figura 37. Ventana Noticias	62
Figura 38. Ventana Educamos	63
Figura 39. Ventana Contacto	63

Imágenes

Imagen 1. CRUD.....	16
Imagen 2. Logo Firebase.....	19
Imagen 3. Logo Android Studio	20
Imagen 4 Esquema Firebase.....	21
Imagen 5. Logo Java	21
Imagen 6. Logo XML	22
Imagen 7. Logo UML.....	22
Imagen 8. Logo Git	23
Imagen 9. Logo Visual Paradigm	23
Imagen 10. Logo Microsoft Project	24
Imagen 11. Logo Microsoft Word.....	24
Imagen 12. Diagrama navegación funcionalidades.....	26
Imagen 13. Página web centro	42
Imagen 14. Paleta de colores	42
Imagen 15. Pantalla Login final	43
Imagen 16. Pantalla Main final.....	43

Tablas

Tabla 1. Objetivos del sistema	30
Tabla 2. Requisitos de información	30
Tabla 3. Requisitos no funcionales	30
Tabla 4. Requisitos funcionales	31
Tabla 5. Actor Padre/Madre/Tutor	31
Tabla 6. Actor Profesor	31
Tabla 7. Actor Administrador.....	32
Tabla 8. Actor Usuario no Registrado	32

1. INTRODUCCIÓN

En la actualidad, el sector educativo es uno de los campos donde se concentra la mayor parte de la actividad, ya que involucra tanto a profesores, como alumnos, como a padres y madres de dichos alumnos. Por ello ha ido evolucionando y avanzando tecnológicamente, desarrollando aplicaciones útiles que facilitan las tareas de los profesores a la hora de conectar con las familias, de colgar información, etc. Un ejemplo de aplicaciones de este estilo puede ser la propia de la USAL, la cual se puede descargar en cualquier dispositivo, y simplemente con introducir nuestras credenciales tendremos acceso a la información que los profesores han colgado, como, por ejemplo, las calificaciones de las diferentes asignaturas.

Todas las aplicaciones de este estilo ya existían, sin embargo, fue con la llegada del COVID-19 cuando tuvieron un crecimiento exponencial, debido a la falta de comunicación directa entre profesores y estudiantes o familias. Algunas de las aplicaciones que tuvieron mucho éxito durante esta crisis fueron Microsoft Teams o Google Meet, para impartir de manera virtual las clases, así como los portales educativos para mantener actualizada toda la información.

Como podemos ver, las tecnologías se encuentran presentes en prácticamente todos los sectores de nuestras vidas, incluyendo el sector educativo. La aplicación de las tecnologías en este ámbito sirve para poder establecer una comunicación mucho más directa entre los centros educativos y las familias, para que estas conozcan en todo momento la evolución del alumno.

El objetivo de nuestra aplicación va a ser que los padres / madres / tutores de los alumnos, conozcan en todo momento la programación lectiva de su hijo, para así poder tener una organización y un cierto control sobre él. Gracias a esto, los padres podrán tener conocimiento de cuando van a tener un examen y poder ayudar a preparárselo, o los días que son las actividades extraescolares en las que se encuentra apuntado, con los datos oficiales proporcionados por parte del equipo docente.

El proyecto desarrollado consiste en una aplicación que va a servir para poder anotar las fechas de los exámenes en función de los cursos en los que se encuentre el usuario matriculado, también contiene una agenda donde se irán añadiendo las actividades programadas por el centro organizadas por semanas y además se incluye una opción para incluir los resultados de los partidos relacionados con las actividades deportivas extraescolares. Además de estas funcionalidades principales, también se proporciona información de contacto y acceso a plataformas externas como la página web de noticias o el portal educativo Educamos. Esta aplicación está diseñada específicamente para el centro educativo de Salamanca San Juan Bosco, pero con unas modificaciones podría servir para cualquier otro centro, o incluso como agenda para un grupo de usuarios.

En esta memoria del Trabajo Fin de Grado se van a tratar los siguientes aspectos:

- **Objetivos:** Se va a hablar sobre las metas propuestas acerca de nuestro proyecto.
- **Conceptos teóricos:** Se explicará en detalle todo lo necesario para poder entender nuestro sistema.
- **Técnicas y herramientas:** Se van a desarrollar las técnicas y herramientas utilizadas durante el desarrollo de nuestro proyecto.
- **Aspectos relevantes:** Se van a tratar los aspectos más relevantes del proyecto.
- **Conclusiones:** Una vez terminado nuestro proyecto, se extraen unas conclusiones, las cuales expondremos en este punto.

- **Lineas futuras:** Se analizan posibles desarrollos futuros a partir del desarrollo ya existente.
- **Referencias:** Se nombran las fuentes utilizadas para la obtención de toda clase de información relacionada con nuestro proyecto.

La memoria se encuentra complementada por una serie de anexos que tratan en detalle sobre los principales aspectos clave del proyecto:

- **Anexo I - Planificación temporal y estimación de esfuerzo:** Se hace una estimación del esfuerzo que va a suponer desarrollar nuestra aplicación, y se organiza en el tiempo cómo va a desarrollarse el mismo.
- **Anexo II – Especificación de requisitos:** En este anexo se van a recoger los requisitos que nuestro sistema debe cumplir.
- **Anexo III – Análisis de requisitos:** Se documenta la fase de análisis de nuestro proyecto software.
- **Anexo IV – Diseño del sistema:** En este anexo se va a documentar el modelo de diseño del sistema.
- **Anexo V – Documentación técnica:** Recoge la documentación acerca del código generado para facilitar su comprensión de cara a futuras modificaciones.
- **Anexo VI – Manual de Usuario:** Este último anexo va a servir de guía para los usuarios, para que puedan conocer a la perfección el funcionamiento de la aplicación.

2. OBJETIVOS

2.1. OBJETIVOS TÉCNICOS

El objetivo principal de la aplicación es la comunicación entre los centros educativos y las familias, siendo el equipo docente o los administradores los encargados de publicar toda la información relacionada con exámenes, resultados y actividades del centro para que sean las familias desde sus casas quien puedan tener conocimiento de todo ello. Para ello se van a seguir los siguientes objetivos:

- **Gestión de usuarios:** El sistema deberá gestionar de forma correcta y eficiente los usuarios, permitiendo registrar usuarios nuevos para poder tener acceso a las funcionalidades de la aplicación. Habrá una diferenciación entre roles, pudiendo distinguir entre los usuarios familiares, los profesores y los administradores. También se deberá permitir la modificación de los datos personales, el cambio de contraseña, la recuperación de la contraseña y el cierre de sesión, así como a los administradores la capacidad de cambiar el rol de cualquiera de los usuarios.
- **Gestión de exámenes:** El sistema deberá ser capaz de gestionar los exámenes que puedan ser añadidos por parte de los profesores o administradores, para que los usuarios puedan ver solamente aquellos exámenes de los cursos en los que se encuentran matriculados. Para los usuarios con privilegios, se va a permitir también modificar y eliminar los exámenes.
- **Gestión de actividades:** El sistema deberá llevar a cabo la gestión de las actividades del centro organizadas por semanas, como una agenda. Los usuarios con privilegios van a tener la capacidad de añadir, modificar y eliminar actividades, para que estas puedan verse en nuestro calendario de actividades de forma correcta. En cada semana se van a

presentar las actividades que se encuentren dentro de ese intervalo de tiempo, entre lunes y domingo, pudiendo moverse entre las diferentes semanas del calendario.

- Gestión de resultados: El sistema deberá gestionar correctamente los resultados de las actividades deportivas extraescolares, así como la organización de estas por jornadas. Para ello, los usuarios con privilegios irán creando las jornadas necesarias, así como añadiendo, modificando y eliminando resultados deportivos. Estos resultados se mostrarán dentro de la jornada correspondiente, y los usuarios podrán visualizar los resultados de cada una de las jornadas navegando entre ellas.

2.2. OBJETIVOS PERSONALES

El desarrollo de este proyecto es la primera toma de contacto en el desarrollo de un sistema software completo, y también mi primer desarrollo para aplicaciones móvil, en este caso, Android. Esto supone un gran reto, y además bastante exigente ya que hay cosas que me resultan novedosas y sobre las cuales tendré que realizar bastantes investigaciones.

Mi objetivo es poder aplicar todo lo visto a lo largo de los cuatro años del grado en un trabajo funcional, el cual me proporciona una motivación extra por el hecho de ser para un centro real que busca poder aplicar el sistema que yo desarrolle en su día a día.

Especialmente me gustaría poder hacer un buen uso del Proceso Unificado, ya que es la base del proyecto, la cual debo seguir estrictamente para una correcta finalización de este. También me apasiona el hecho de poder desarrollar un proyecto completo, desde inicio a fin, enfrentándome a las adversidades que se puedan ir presentando y corrigiendo todo lo necesario para mejorar de cara a líneas futuras.

3. CONCEPTOS TEÓRICOS

En este apartado se van a tratar los conceptos teóricos que puedan aparecer a lo largo del desarrollo del proyecto, para tener una idea general de todo aquello que ha sido utilizado. Los conceptos por tratar van a estar relacionados con nuestra aplicación.

3.1. APLICACIONES CRUD



Imagen 1. CRUD

Las aplicaciones CRUD (León, 2011), como la nuestra, son tipos de aplicaciones que tienen una conexión directa con una base de datos, donde se van a almacenar todos los datos. Este tipo de aplicaciones son relativamente simples, ya que solamente desarrollan cuatro funciones principales, de ahí su nombre CRUD:

- Create (Crear): Implica la creación de nuevas entradas en la base de datos, normalmente suelen realizarse a través de formularios que el usuario rellenará.
- Read (Leer): Recupera y muestra los datos de la base de datos. Puede referirse a presentar una colección de elementos.
- Update (Actualizar): Modifica los datos ya existentes en la base de datos, cambiando los valores que el usuario considere oportunos.
- Delete (Eliminar): Borra de la base de datos ese registro, por lo que no podrá volver a ser mostrado.

Este tipo de aplicaciones son muy útiles para la gestión, ya que continuamente se están añadiendo, modificando y eliminando registros.

3.2. APLICACIONES ANDROID

Las aplicaciones Android son programas software diseñados en un cierto lenguaje de programación para dispositivos móviles, concretamente aquellos cuyo sistema operativo sea Android. Estas aplicaciones, también conocidas como aplicaciones nativas, aquellas que son solamente para un sistema operativo, se centran en exprimir al máximo todas las características y funcionalidades del sistema para el que se desarrollan.

Android tiene su entorno de desarrollo personalizado, desde el cual se suelen realizar la gran mayoría de las aplicaciones para este sistema operativo, aunque no todas. Este IDE se llama Android Studio, y más adelante vamos a hablar de él, ya que ha sido el utilizado para nuestro proyecto.

Las aplicaciones Android suele tener:

- Interfaz de usuario (UI): Gracias a ella los usuarios pueden interactuar con todas las funcionalidades de la aplicación, mediante botones, formularios de entrada de texto, barras deslizantes, menús desplegables, etc.
- Conexión directa con los elementos hardware: Muchas de las aplicaciones solicitan permisos para poder acceder al hardware del dispositivo, como la cámara, micrófono, ubicación, etc. Esto va a servir para desarrollar funcionalidades concretas específicas de cada sistema.
- Conexión a internet: Hoy en día la mayoría de las aplicaciones necesitan estar conectadas a internet, bien sea para poder autenticar los usuarios, obtener datos de la nube o para permitir la comunicación en tiempo real.
- Gestión de datos: Esto es posible gracias a la utilización de bases de datos, las cuales van a permitir almacenar estructuradamente los datos que deseemos relacionados con nuestra aplicación. Nuestra aplicación va a tratar, generalmente, en la gestión de los datos, ya que está en continua comunicación con la base de datos.

3.3. APLICACIONES EDUCATIVAS DE COMUNICACIÓN

Las aplicaciones educativas de comunicación (Almenara, 2015) sirven para mejorar, como bien dice el nombre, la comunicación entre el propio centro educativo y las familias, gracias a una transmisión de información directa, para que en todo momento se tenga una información precisa y de calidad de todo lo que sucede.

Actualmente son muchas las aplicaciones que se utilizan en el día a día de cada uno de los centros de nuestro país, desde los niveles educativos más bajos hasta los más altos. Esto se debe a la facilidad existente para disponer de un dispositivo móvil con conexión a internet, además de

haber podido comprobar que un correcto uso de este tipo de aplicaciones tiene un impacto positivo, donde todas las partes salen beneficiadas.

Son muchos los tipos de aplicaciones existentes actualmente, algunas de ellas especializadas en funciones muy concretas. Algunas de los tipos de aplicaciones más utilizadas son las siguientes:

- Aplicaciones para la comunicación entre profesorado y estudiantes: Este tipo de aplicaciones suelen ser foros, un servicio de mensajería o similares, donde los alumnos pueden plantear sus dudas para que los profesores, o incluso otros alumnos, puedan resolvérselas en cualquier momento del día. Este tipo de aplicaciones suelen estar incorporadas dentro de los diferentes portales educativos, como ocurre en la plataforma utilizada por la Universidad de Salamanca, Studium.
- Aplicaciones de videoconferencia: Con la aparición de las clases online, se comenzaron a utilizar este tipo de aplicaciones donde los usuarios se pueden unir a salas en las que se transmiten, por parte de otro usuario, un contenido en concreto. Algunas de estas aplicaciones pueden ser Zoom, Microsoft Teams o Google Meet, siendo estas las más utilizadas en entornos educativos.
- Aplicaciones de mensajería: Esto sirve para una comunicación directa y mucho más rápida que por ejemplo el correo electrónico clásico. En entornos educativos, son muchos los centros donde se decantan por la creación de grupos en aplicaciones de este estilo como WhatsApp o Telegram, para poder ir transmitiendo cualquier tipo de información, cambio de planes o novedades.
- Aplicaciones de documentos: Son aplicaciones donde los usuarios pueden ir colgando en un espacio común, los diferentes documentos que vayan generando. Alguna aplicación de este estilo puede ser Google Drive, One Drive o Dropbox.

También existen aplicaciones menos conocidas, pero también muy utilizadas en entornos educativos como pueden ser:

- Aula 1: Su módulo de comunicaciones facilita información de cualquier evento, calificación o anuncio relacionado con el aula, apareciendo automáticamente en el área del alumno, familiar o docente. Dispone de medidas antispam y los comunicados pueden hacerse vía e-mail, por SMS o app.
- Alexia: Esta plataforma multi-idioma integra el área de gestión académico-administrativa, herramientas como PLE, contenidos, bibliotecas, horarios, contabilidad... También facilitar las relaciones del centro con toda la comunidad educativa, tanto empleados como alumnos y familias, ya que favorece la comunicación a través de todos los canales (web, app, email, SMS) y dispone de interesantes funcionalidades, como solicitar entrevistas, realizar cuestionarios y encuestas, etcétera.

Todo este tipo de aplicaciones se consideran de comunicación entre centros educativos y familias, sin embargo, en mi caso, la aplicación desarrollada está especializada en los requisitos solicitados por parte del centro educativo de Salamanca, San Juan Bosco. La aplicación va a tener aspecto de "agenda", donde su principal función va a ser la de almacenar información relevante como es el caso de las fechas de los exámenes de los estudiantes o las fechas de las actividades organizadas. También se va a disponer de un calendario de resultados donde se van a poder visualizar por jornadas, los resultados de los partidos deportivos que disputan los diferentes equipos de la institución. Además, se va a tener una gestión de usuarios, con diferenciación entre los administradores, profesores y familias, teniendo cada uno de estos unos privilegios específicos.

3.4. FIREBASE



Imagen 2. Logo Firebase

Firebase (Documentación de Firebase, 2023) es una plataforma de desarrollo de aplicaciones en la nube proporcionada por Google, con una amplia gama de servicios que ayudan a los desarrolladores a crear estas aplicaciones, además emplea una interfaz bastante sencilla, visual y fácil de utilizar, ofreciendo una amplia documentación a la que podemos recurrir en caso de presentar alguna dificultad o problema.

Algunos de los servicios proporcionados por Firebase más utilizados son:

- Firebase Authentication: Proporciona un servicio de autenticación segura para permitir a los usuarios registrarse y autenticarse utilizando diferentes métodos de registro.
- Cloud Firestore y Realtime Database: Son sistemas de bases de datos en tiempo real que permiten almacenar datos y sincronizarlos entre los usuarios y los dispositivos.
- Firebase Storage: Proporciona un servicio de almacenamiento de archivos y ficheros en la nube.
- Firebase Analytics: Permite obtener estadísticas acerca de los usuarios que han utilizado nuestra aplicación.

La mayoría de los servicios de esta plataforma son de acceso gratuito hasta unos ciertos niveles de uso, por lo que para aplicaciones que no son excesivamente grandes no deberíamos tener problemas.

4. TÉCNICAS Y HERRAMIENTAS

4.1. ENTORNOS DE DESARROLLO

4.1.1. Android Studio



Imagen 3. Logo Android Studio

Android Studio (Documentación de Android Studio, 2023) (Documentación de Firebase Firestore, 2023) es un IDE (Entorno de Desarrollo Integrado) para desarrollar aplicaciones móviles específicamente para el sistema operativo de Android, desarrollado por Google para remplazar a Eclipse. Este entorno de desarrollo está diseñado para exprimir a fondo las funcionalidades de los dispositivos Android, y dispone de las siguientes características:

- Soporte para la construcción basada en Gradle, siendo este un sistema de automatización de construcción de código software que construye sobre los conceptos de Apache Ant y Apache Maven e introduce un lenguaje específico del dominio basado en Groovy en vez de la forma XML utilizada por Apache Maven para declarar la configuración del proyecto.
- Incluye herramientas para detectar problemas de rendimiento y de compatibilidades.
- Integración de funciones de firma de aplicaciones.
- Incluye plantillas comúnmente utilizadas en Android.
- Fácil editor de diseño, para poder arrastrar manualmente los elementos.
- Permite la integración con Firebase.
- Dispositivos Android para poder realizar pruebas y facilidad para conectar dispositivos externos mediante la utilización de cable.
- Renderizado en tiempo real.

4.2. BASE DE DATOS

4.2.1. Firebase Firestore

La base de datos utilizada, como ya se ha explicado anteriormente, es Firebase, en concreto Firebase Firestore (Documentación de Firebase Firestore, 2023) es la que se ha empleado para el almacenamiento en tiempo real de los datos de nuestra aplicación.

La base de datos se estructura por colecciones, las cuales contienen documentos donde se encuentran almacenados los campos que queremos guardar.

Para que se pueda establecer la conexión entre los servidores de Google, que son los encargados de hostear Firebase, y nuestra aplicación, debemos añadir en nuestro proyecto el fichero proporcionado a la hora de la configuración de Firebase llamado "Google-services.json".

Para poder tener conectada nuestra aplicación con la base de datos, deberemos añadir en el fichero de configuración build.gradle (Documentación de Gradle, 2023) de nuestro proyecto, las dependencias relacionadas con la base de datos. Esto va a permitir que podamos utilizar las funcionalidades que nos proporciona Firebase en nuestras clases para la gestión de los datos.

Para realizar peticiones de información, deberemos referenciar la colección a la que queremos acceder, y en función de lo que deseemos, obtendremos toda la colección o simplemente obtendremos los documentos que deseemos realizando una consulta más elaborada.

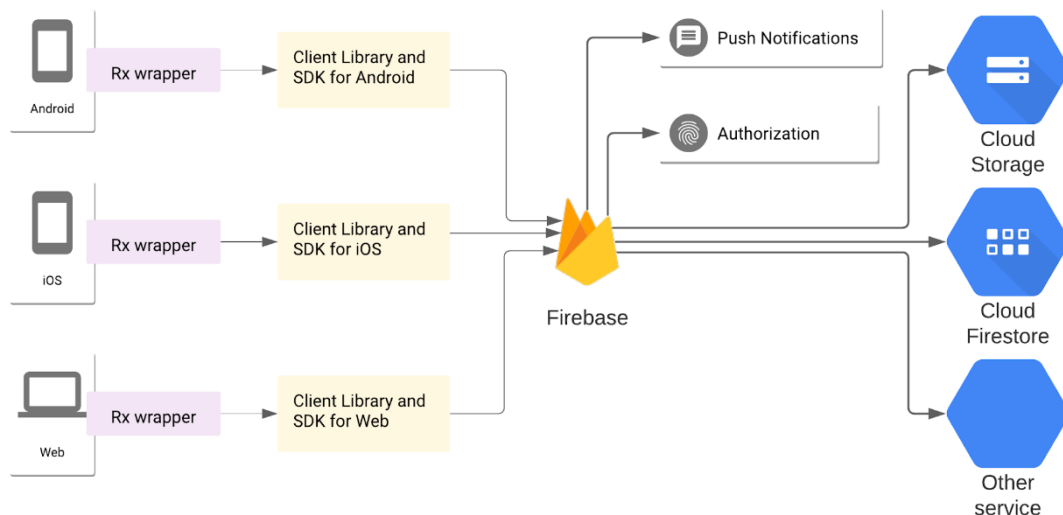


Imagen 4 Esquema Firebase.

4.3. LENGUAJES DE PROGRAMACIÓN

4.3.1. Java

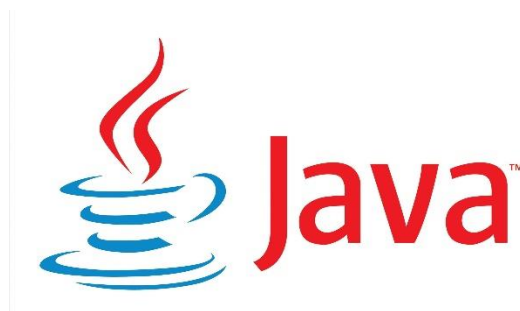


Imagen 5. Logo Java

Java (Documentación de Java, 2023) es un lenguaje de programación ampliamente utilizado para codificar aplicaciones web. Ha sido una opción popular entre los desarrolladores durante más de dos décadas, con millones de aplicaciones Java en uso en la actualidad. Java es un lenguaje multiplataforma, orientado a objetos y centrado en la red que se puede utilizar como una plataforma en sí mismo. Es un lenguaje de programación rápido, seguro y confiable para codificarlo todo, desde aplicaciones móviles y software empresarial hasta aplicaciones de macrodatos y tecnologías del servidor.

Al no ser un lenguaje nuevo, Java cuenta con gran cantidad de documentación que pueden permitir realizar todo lo que uno se proponga. Además, dispone de bibliotecas ya creadas con infinitas funciones, por lo que no hay que hacer las cosas desde cero una y otra vez.

Al ser uno de los lenguajes más utilizados, tiene un gran apoyo por parte de la comunidad. También destaca por la seguridad proporcionada por Java.

4.3.2. XML



Imagen 6. Logo XML

XML (Documentación de XML, 2023) es un lenguaje de marcado extensible, que permite definir y almacenar datos de forma compatible. Admite el intercambio de información entre diferentes sistemas como webs, bases de datos y aplicaciones.

XML no puede realizar operaciones de computación por sí mismo, sin embargo, se puede implementar cualquier software o lenguaje de programación para la administración de la base de datos.

Algunos de sus beneficios son:

- Respaldo para las transacciones interempresariales, ya que pueden compartir información de negocio entre las dos empresas sin necesidad de la intervención humana.
- Conservación de la integridad de los datos, permitiendo transferir datos junto con la descripción de estos.
- Mejora de la eficiencia de búsqueda, ya que los programas pueden ordenar los archivos XML más fácilmente que cualquier otro tipo de documentos.
- Diseño de aplicaciones flexibles, pudiendo actualizar o modificar el diseño de nuestra aplicación de manera rápida y sencilla.

4.3.3. UML



Imagen 7. Logo UML

El Lenguaje Unificado de Modelado (UML) (Larman, 1999) fue creado para forjar un lenguaje de modelado visual común a la hora de desarrollar un producto software, tanto en arquitectura como en diseño e implementación.

Se utiliza en el desarrollo de la documentación para la creación de diferentes diagramas con una estructura y una forma común. Es el lenguaje principal que se utiliza en Visual Paradigm en un proceso software.

Fue creado por tres amigos en los años 90, en concreto en 1996, pero no fue hasta 1999 cuando lanzaron una Guía para el Usuario.

4.4. HERRAMIENTAS DE CONTROL DE VERSIONES

4.4.1. Git



Imagen 8. Logo Git

Git (Git, 2023) es un sistema de control de versiones, el cual nos permite clonar en un repositorio virtual toda la información de nuestro proyecto. Esto es útil, además de para tener todas las versiones registradas, para poder trabajar varios integrantes de un mismo equipo de manera conjunta gracias al uso de ramas adicionales.

Permite la confirmación de los cambios, la visualización de estos y también poder retomar versiones anteriores en caso de haber sufrido algún tipo de error.

En el caso de nuestro proyecto, se encontraba integrado dentro del entorno de desarrollo de Android Studio, sin embargo, hay diversas herramientas con interfaces más visuales que la línea de comandos para la gestión de los proyectos.

4.5. HERRAMIENTAS CASE

4.5.1. Visual Paradigm



Imagen 9. Logo Visual Paradigm

Visual Paradigm (Visual Paradigm, 2023) es la herramienta software utilizada para la realización de todos los diagramas durante el proceso software, tales como diagramas de clases, diagramas de casos de uso, diagramas de secuencia, etc.

Estos diagramas han sido realizados en UML, a pesar de que Visual Paradigm admite también otros lenguajes diferentes, pero UML es el más utilizado y el que ya conocíamos anteriormente.

Esta herramienta permite exportar los diagramas que vamos creando dentro de nuestro proyecto a diferentes formatos para poder utilizarlos en nuestra documentación del proyecto.

4.5.2. EZEstimate

Esta herramienta permite calcular la estimación del esfuerzo de nuestro proyecto software a partir de ciertos valores que le iremos asignando. Debemos crear los módulos necesarios, ir añadiendo los actores y casos de uso y posteriormente ajustar los factores técnicos y los factores de entorno. También deberemos ajustar el esfuerzo en función de la dificultad de los casos de uso que debemos desarrollar, y nos dará un valor aproximado en horas.

4.5.3. Microsoft Project



Imagen 10. Logo Microsoft Project

Microsoft Project (Visual Paradigm, 2023) es una herramienta de organización y gestión de proyectos, utilizada para administrar las tareas que se deberán realizar dentro de un proceso software. Cabe la posibilidad de añadir los días festivos, el horario de trabajo, los integrantes del equipo y demás factores que nos van a servir para tener todo el desarrollo bajo control.

Es una aplicación muy sencilla y fácil de usar, por lo que con unos pocos clicks podemos tener una correcta gestión de nuestros proyectos.

4.5.4. Javadoc

Javadoc (Javadoc, 2023) es la utilidad diseñada por Oracle para generar documentación en formato HTML a partir del código Java. Esta herramienta reconoce los comentarios, y además reconoce las etiquetas con palabras reservadas, las cuales comienzan por una “@”.

Por ejemplo, para indicarle los parámetros que se le pasan a una función se hace uso de “@param”, y para indicar lo que devuelve dicha función se utiliza “@return”.

Esta herramienta viene implementada en Android Studio.

4.5.5. Microsoft Word



Imagen 11. Logo Microsoft Word

Microsoft Word (Microsoft Word, 2023) es un procesador de textos diseñado por Microsoft, a través del cual se ha desarrollado toda la documentación de nuestro proyecto fin de grado.

5. ASPECTOS RELEVANTES

En esta sección se van a tratar los aspectos más relevantes del desarrollo de nuestra aplicación, la cual va a consistir en una aplicación para que los centros educativos, en este caso más concretamente el San Juan Bosco, de Salamanca, y las familias de los alumnos puedan mantenerse comunicados, teniendo a su disposición toda la información necesaria.

Las principales funcionalidades de este sistema software van a ser las siguientes:

- Gestión de usuarios, donde dispondremos de un sistema capaz de registrar usuarios, pudiendo diferenciar entre diferentes roles, otorgando unos privilegios u otros en función del rol asignado.
- Gestión de exámenes, permitiendo organizar los exámenes por cursos y fechas para tener disponible una “agenda” con los días que éstos van a tener lugar.
- Gestión de actividades, pudiendo añadir las diferentes actividades relacionadas con el centro en la semana correspondiente. Se va a poder visualizar un calendario con las diferentes actividades planificadas.
- Gestión de resultados, donde se van a ir añadiendo los resultados de los partidos deportivos que hayan tenido lugar, organizado por jornadas para poder tener un seguimiento de los diferentes equipos del centro.

Estas son las funcionalidades más destacadas sobre las que trata nuestra aplicación, las cuales vamos a poder ir viendo en detalle.

En la Imagen 12 podemos ver cómo, a través de la aplicación, podemos navegar entre las diferentes funcionalidades de nuestro sistema. Hay flechas unidireccionales, que indican que solamente podemos acceder en ese sentido, y flechas bidireccionales que indican que podemos acceder a la información en ambos sentidos.

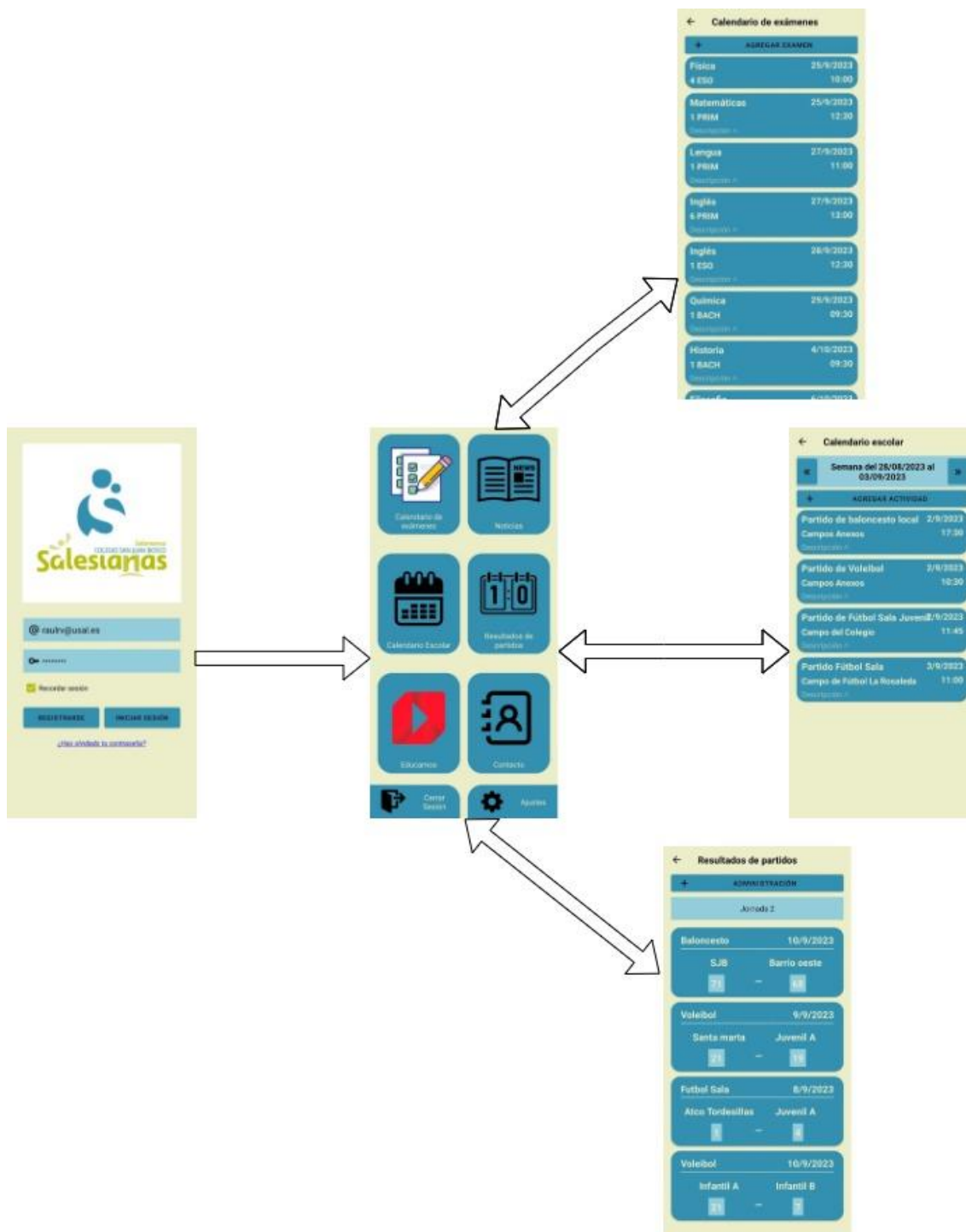


Imagen 12. Diagrama navegación funcionalidades

5.1. METODOLOGÍA

Para el desarrollo de nuestro proyecto software se va a utilizar la metodología del Proceso Unificado, el cual ya se ha explicado previamente y se han hablado de sus características.

Para este proceso de desarrollo se va a seguir el ciclo de vida del Proceso Unificado, el cual consta de las siguientes fases:

- Inicio: Es la fase inicial del proyecto, donde se definen las metas y se realiza una introducción a este.

- Elaboración: Es en esta fase en donde se comienzan a diseñar y refinar los casos de uso, así como la arquitectura que va a tener nuestro sistema.
- Construcción: En esta fase, quizá la más extensa, es donde se lleva a cabo el desarrollo del producto de manera más gruesa, dejándolo prácticamente terminado a falta de pequeñas correcciones.
- Transición: El sistema es llevado a un periodo de pruebas por parte del cliente, corrigiendo los fallos que puedan ir surgiendo.

Cada fase puede tener una o varias iteraciones dependiendo de la dificultad que tenga la misma.

Dentro de cada una de las fases, hay unas disciplinas que tienen lugar en prácticamente todas las iteraciones:

- Modelado de negocio
- Requisitos
- Análisis
- Diseño
- Implementación
- Pruebas

A continuación podemos ver una imagen con un resumen donde se puede ver la duración aproximada de cada fase, sus subfases y las iteraciones.

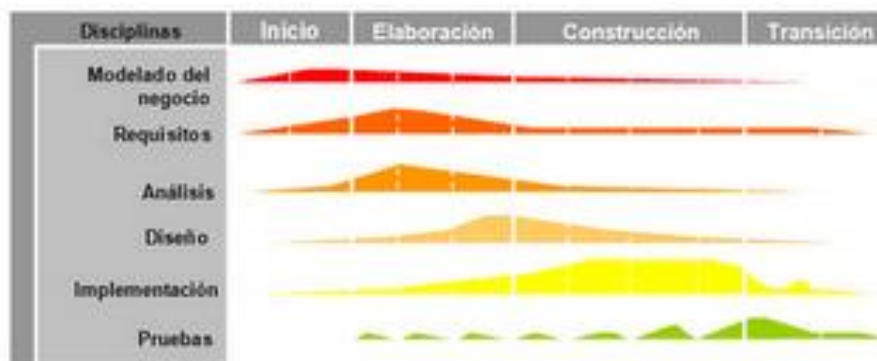


Figura 1. Fases del Proceso Unificado

Las dos primeras fases de Inicio y Elaboración suelen hacer referencia a un proceso de ingeniería, donde se suele tratar más acerca de los requisitos, refinamiento de casos de uso etc. Las dos fases siguientes de Construcción y Transición suelen tratar acerca de la elaboración del sistema software, incluyendo fases de pruebas. Esto es una estimación, ya que, en todas las fases, se hace prácticamente algo de cada una de las subfases.

5.2. ESTIMACIÓN DEL ESFUERZO

La tarea de estimación de esfuerzo consiste en calcular el esfuerzo necesario en horas por persona para realizar nuestro sistema completo.

Para realizar esta estimación, en nuestro proyecto se ha utilizado la métrica basada en Puntos de Casos de Uso (UCP), a través de la herramienta ya explicada de EZEstimate, donde como se comentó se introducen los actores, casos de uso, factores técnicos, factores de entorno y se define un valor de esfuerzo, y a partir de eso se obtiene un cálculo final.

Module
 Resultados
 Add Module Delete

Summary
Total Modules 5 **Excel Report** Generate Report
Use cases Simple 26 Average 0 Complex 0
Actors Simple 0 Average 0 Complex 4

Add Actor / Use case
 Actor / Use case Name Select Type Complexity
 Usecase Simple Add

Tech / Env Factors
 Set Tech Factor
 Set Env Factors

Estimation Summary

UAW	12
UUCW	130
UUPC = UAW + UUCW	142
TFactor	32
EFactor	14
TCF = 0.6 + (.01*TFactor)	0,92
EF = 1.4 + (-0.03*EFactor)	0,98
UCP = UUCP*TCF*EF	128,0272
Total Effort@ 5 Hrs/UCP	640,136

Use case / Actor List { Double click to delete }

Id	Module	Type	Name	complexity
1	Actores	Actor	Usuario No Re...	Complex
10	Usuarios	Usecase	Ver ajustes	Simple
11	Usuarios	Usecase	Modificar datos ...	Simple
12	Usuarios	Usecase	Modificar contr...	Simple
13	Usuarios	Usecase	Asignar rol a us...	Simple
14	Exámenes	Usecase	Ver calendario ...	Simple
15	Exámenes	Usecase	Agregar examen	Simple
16	Exámenes	Usecase	Modificar examen	Simple
17	Exámenes	Usecase	Eliminar examen	Simple
18	Actividades	Usecase	Ver calendario ...	Simple
19	Actividades	Usecase	Avanzar seman...	Simple
2	Actores	Actor	Padre / Madre	Complex
20	Actividades	Usecase	Ir atras semana ...	Simple
21	Actividades	Usecase	Agregar actividad	Simple
22	Actividades	Usecase	Modificar activi...	Simple
23	Actividades	Usecase	Eliminar actividad	Simple
24	Resultados	Usecase	Ver resultados ...	Simple
25	Resultados	Usecase	Seleccionar inr	Simple

Figura 2. Ventana EZEstimate

Como podemos ver en la imagen 14, el esfuerzo calculado por nuestra herramienta es de 640 horas, colocando un Total Effort con valor 5, debido a la simplicidad de algunos de los casos de uso.

En base a las 640 horas obtenidas, decidí organizarme el tiempo de tal manera que pudiese dedicarme prácticamente a tiempo completo al proyecto, siguiendo una rutina de trabajo para poder tener una progresión adecuada. Para ello, esperé a la finalización de las prácticas externas del Grado, y posterior a esto, pude compaginar la realización del sistema con la asignatura restante de Sistemas Distribuidos.

En proyectos de tales dimensiones, con gran cantidad de información novedosa la cual supone una gran inversión de tiempo en labores de investigación, la organización del tiempo es primordial para poder desarrollar un buen trabajo final.

El resultado final de horas obtenidas estuvo cerca de las 600, sumando todo el tiempo dedicado a la investigación, visualización de tutoriales, pruebas de la base de datos antes de comenzar con su utilización dentro de la aplicación, refinamiento de la documentación, etc. Es decir, el valor obtenido en la aplicación de EZEstimate, para ser un valor aproximado que se proporciona, estuvo bastante acertado.

5.3. PLANIFICACIÓN TEMPORAL

En la planificación temporal de nuestro proyecto lo que vamos a buscar es organizar las diferentes actividades que tenemos que realizar en una línea del tiempo, indicando el tiempo aproximado que desarrollaremos en cada una de ellas dependiendo del calendario de trabajo

que hayamos definido. Estableceremos dependencias entre ciertas tareas, las cuales no pueden comenzar hasta que su actividad respectivamente anterior no hay finalizado.

Para llevar a cabo esta planificación temporal vamos a hacer uso de la herramienta Microsoft Project, donde iremos añadiendo las diferentes tareas a las fases del ciclo de vida de proceso unificado, terminando con hitos cada una de las iteraciones y de fases.

Tendremos un total de 7 iteraciones, repartidas entre las diferentes fases. Las 3 primeras iteraciones van a pertenecer a las fases de Inicio y Elaboración, y por tanto al proceso de Ingeniería, mientras que las 4 fases restantes se corresponden con las fases de Construcción y Transición, y, por tanto, hacen referencia al proceso de desarrollo software, donde se crea y prueba el producto final.

En la siguiente figura se va a poder observar la planificación temporal para la primera iteración dentro de la fase de Inicio.

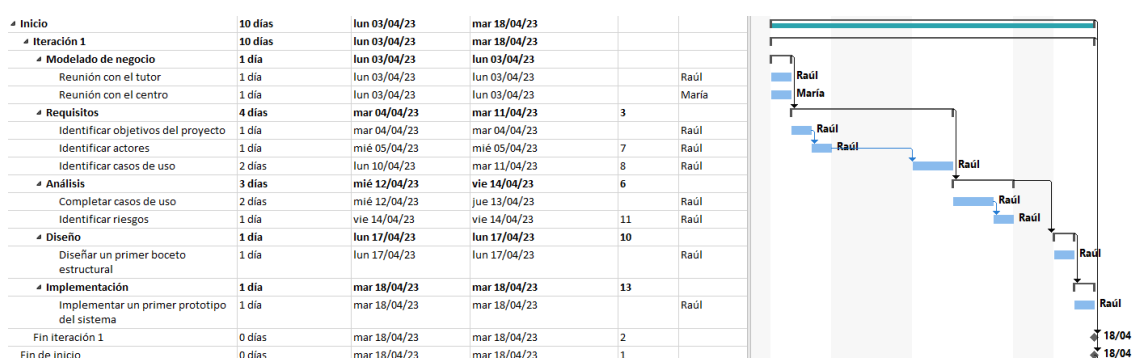


Figura 3. Planificación temporal Inicio

El resto de información acerca de la planificación temporal puede observarse dentro del Anexo I – Planificación y estimación del esfuerzo.

5.4. ESPECIFICACIÓN DE REQUISITOS

Para la especificación de requisitos de nuestro sistema software, se deberán tomar del cliente las necesidades que este tiene acerca de la aplicación a desarrollar, y se procederá a la elaboración de una serie de requisitos, explicados en detalle, que serán el paso previo a un posterior diseño antes de una implementación.

Para el diseño de estos requisitos se ha seguido una metodología basada en las tablas de Duran y Bernárdez.

Lo primero que vamos a desarrollar van a ser los objetivos que queremos que nuestro sistema cumpla, y por ello, para cada objetivo, hemos desarrollado una tabla similar a la siguiente:

OBJ-001	Gestión de usuarios
Autor	Raúl Rodríguez Villanueva
Fuentes	María Navarro Cáceres
Descripción	El sistema debe permitir el registro, inicio de sesión, cierre de sesión y recuperación de contraseña para los usuarios de la aplicación. Además, permitirá la modificación de los datos de los usuarios ya registrados (nombre, apellidos, cursos matriculados,

	contraseña), así como la asignación de roles específicos para ciertos usuarios con privilegios.
Importancia	Alta
Urgencia	Alta
Estabilidad	Alta
Comentarios	Ninguno

Tabla 1. Objetivos del sistema

A continuación, se van a definir los requisitos de la información, siendo estos los que van a hacer referencia a las necesidades y características que debe tener la información que vamos a manejar dentro del sistema, y para ello se han desarrollado unas tablas similares a la Tabla 2.

IRQ-002	Información sobre exámenes
Autor	Raúl Rodríguez Villanueva
Fuentes	
Descripción	El sistema deberá almacenar la información correspondiente a los exámenes añadidos. En concreto:
Datos específicos	<ul style="list-style-type: none"> - Nombre de la asignatura - Fecha - Hora - Curso - Descripción - ID del examen
Importancia	Alta
Urgencia	Alta
Estabilidad	Alta
Comentarios	Ninguno

Tabla 2. Requisitos de información

Lo siguiente que se va a tratar van a ser los requisitos no funcionales, siendo este tipo de requisitos los que hacen referencia a como deberá comportarse nuestro sistema en diferentes situaciones, en términos de calidad, seguridad, etcétera, sin estar relacionados directamente con las funcionalidades. Para ello se han desarrollado una serie de tablas con el contenido de este tipo de requisitos, con aspecto similar a la Tabla 3.

NFR-002	Usabilidad
Autor	Raúl Rodríguez Villanueva
Descripción	El tiempo de aprendizaje del sistema por un usuario debe ser menor a 2 horas. El sistema debe poseer interfaces gráficas sencillas y bien formadas.
Importancia	Alta
Comentarios	Ninguno

Tabla 3. Requisitos no funcionales

A continuación, se van a especificar los casos de uso o requisitos funcionales, los cuales sí que tratan acerca de la funcionalidad que va a tener nuestra aplicación. Estos se han elaborado a partir de las especificaciones recibidas por parte del centro escolar que va a utilizar la aplicación, es decir, nuestro cliente. Cada una de las tablas desarrolladas va a tener un aspecto diferente, en función del requisito que estén tratando, sin embargo, van a tener una estructura similar a la Tabla 5.

UC-007	Modificar examen
---------------	-------------------------

Autor	Raúl Rodríguez Villanueva	
Objetivos Asociados	- OBJ-002	
Requisitos Asociados	- IRQ-002	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario quiera modificar un examen añadido previamente	
Precondición	El usuario debe ser profesor o administrador	
Secuencia normal	Paso	Acción
	1.	El actor Profesor (ACT-002) o Administrador (ACT-003) mantendrá pulsado sobre el examen que desea modificar
	2.	El sistema le mostrará un cuadro de diálogo con varias opciones
	3.	El actor Profesor (ACT-002) o Administrador (ACT-003) pulsará en la opción de “Modificar Examen”
	4.	El sistema le mostrará al usuario una ventana con los datos del examen cargados en sus respectivos campos.
	5.	El actor usuario realizará los cambios necesarios y pulsará en “Guardar”.
	6.	El sistema actualizará el examen en la base de datos con la nueva información.
Postcondición		
Excepciones	Paso	Acción
	6.	La nueva fecha del examen es fin de semana o coincide con otro examen ese día para ese mismo curso
Frecuencia	10 veces / día	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 4. Requisitos funcionales

Por último, solamente quedan los actores, que van a ser los diferentes tipos de usuario de los que van a interactuar con el sistema. Los actores son 4 y son los siguientes:

ACT-001	Padre / Madre / Tutor
Autor	Raúl Rodríguez Villanueva
Fuentes	María Navarro Cáceres
Descripción	Este actor representa un usuario logeado, con funcionalidad de padre / madre dentro de la aplicación
Comentarios	Ninguno

Tabla 5. Actor Padre/Madre/Tutor

ACT-002	Profesor
Autor	Raúl Rodríguez Villanueva
Fuentes	María Navarro Cáceres
Descripción	Este actor representa un profesor del centro
Comentarios	Ninguno

Tabla 6. Actor Profesor

ACT-003	Administrador
Autor	Raúl Rodríguez Villanueva
Fuentes	María Navarro Cáceres

Descripción	Este actor representa al administrador de la aplicación
Comentarios	Ninguno

Tabla 7. Actor Administrador

ACT-004	Usuario No Registrado
Autor	Raúl Rodríguez Villanueva
Fuentes	María Navarro Cáceres
Descripción	Este actor representa un usuario no registrado en la aplicación
Comentarios	Ninguno

Tabla 8. Actor Usuario no Registrado

También se van a crear diagramas de casos de uso para cada uno de los paquetes de nuestra aplicación, donde se va a mostrar las funcionalidades que van a tener cada uno de los actores y la relación entre ellos. En la Figura 5 se va a poder observar el diagrama de casos de uso para el paquete Usuarios, siendo este paquete el encargado de las funcionalidades relacionadas con los diferentes usuarios del sistema, así como su registro, inicio de sesión, diferenciación de roles, recuperación de contraseña, etc.

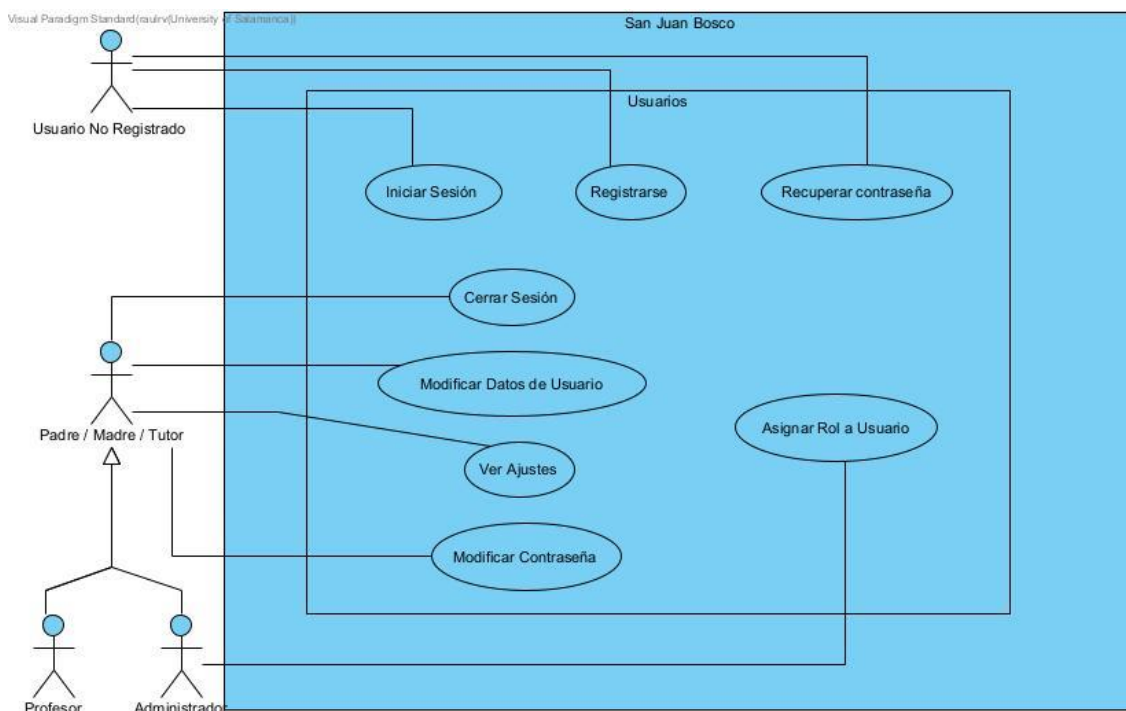


Figura 4. Diagrama de casos de uso – Usuarios

Los casos de uso se han organizado en función de las funcionalidades con las que estuviesen asociadas, pudiendo diferenciar, además del paquete de Usuarios, los siguientes paquetes:

- Exámenes: Contiene los casos de uso relacionados con toda la gestión de los exámenes, diferenciando entre los cursos, y otorgando las funcionalidades especiales a los usuarios con privilegios.
- Actividades: Encargada de la gestión de las actividades, organizándolas por semanas dentro del calendario.
- Resultados: Contiene todos los casos de uso relacionados con los resultados, así como la organización de las jornadas incluida.

El resto de los diagramas y tablas de Duran y Bernárdez se encuentran contenidos en el Anexo II – Especificación de Requisitos.

5.5. ANÁLISIS

Para desarrollar un correcto análisis debemos recopilar la información más importante y correcta de nuestros requisitos, para poder tener una base previa a la implementación. En esta fase de análisis de requisitos se van a desarrollar tres tareas:

- Modelo del dominio: Representación de las clases que componen nuestro sistema y las relaciones que existen entre ellas. Para esto se utilizará un diagrama de clases.
- Realización de casos de uso en el modelo de análisis: Describe a través de diagramas la ejecución de los diferentes casos de uso redactados. Para ello se utilizarán diagramas de secuencia.
- Propuesta de arquitectura: Representación de la arquitectura seguida en el proyecto. Para esto se utilizará un diagrama de paquetes.

5.5.1. Modelo del Dominio

En esta sección se buscan las clases más importantes del sistema, con los atributos más significativos, y se establecen relaciones entre estas clases para que el sistema pueda desarrollar sus funcionalidades de manera correcta. En la Figura 6 se puede ver el diagrama de clases implementado.

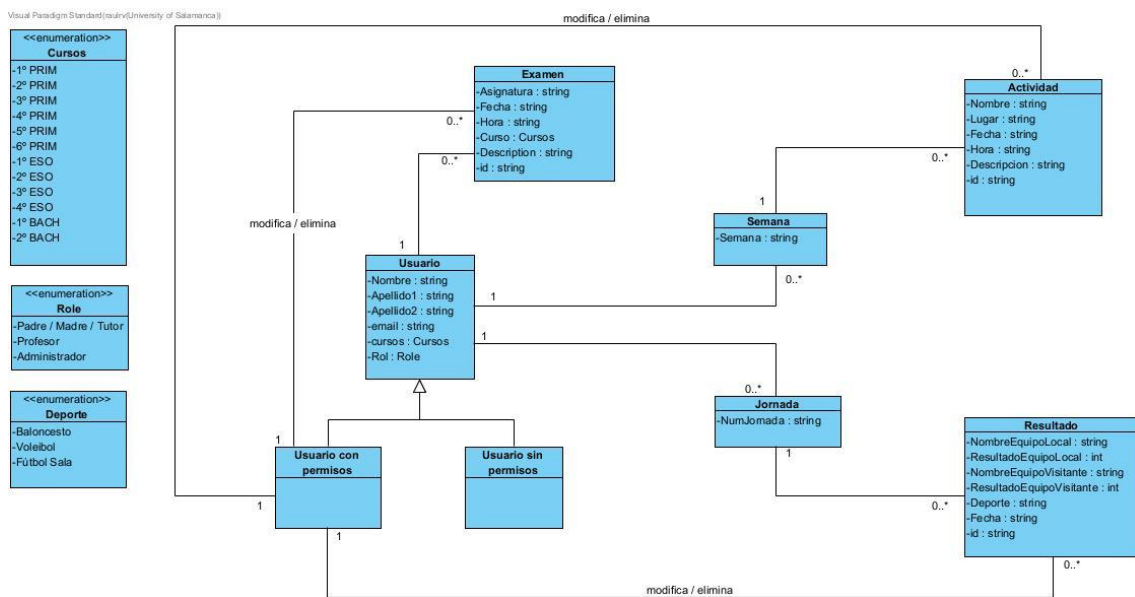


Figura 5. Modelo del dominio - Diagrama de clases

En la parte izquierda del diagrama se encuentran las enumeraciones, las cuales se van a utilizar, sobre todo, en los Spinners de Android Studio, siendo estos unos desplegables con diferentes opciones las cuales pueden ser seleccionadas para establecer opciones predeterminadas.

También se pueden ver como en la parte más central tenemos una clase “Usuario”, que tiene dos clases que dependen de él, el Usuario sin permisos y el Usuario con permisos. Este último usuario va a poder hacer lo mismo que el “Usuario”, con la diferencia que tiene funcionalidades añadidas como es el caso de poder añadir, modificar y eliminar los exámenes, resultados y actividades.

5.5.2. Realización de casos de uso

Nuestros casos de uso van a estar realizados mediante diagramas de secuencia, donde, además de los actores, va a haber tres tipos de clases:

- Clases de interfaz: Esta es la clase que está en contacto con el usuario, la que se encarga de la comunicación con él.
- Clases de control: Se encargan de realizar las funcionalidades del sistema, a partir de lo que el usuario le ha solicitado a la interfaz.
- Clases de entidad: Son las clases que se almacenan en el sistema, conteniendo la información.

Uno de los casos de uso que podemos ver puede ser el de la Figura 7. Este caso de uso hace referencia al inicio de sesión de un usuario que aún no se encuentra registrado. Este diagrama se ha seleccionado ya que el inicio de sesión es uno de los requisitos más importantes y utilizados a lo largo de cualquier aplicación que incluye una gestión de usuarios.

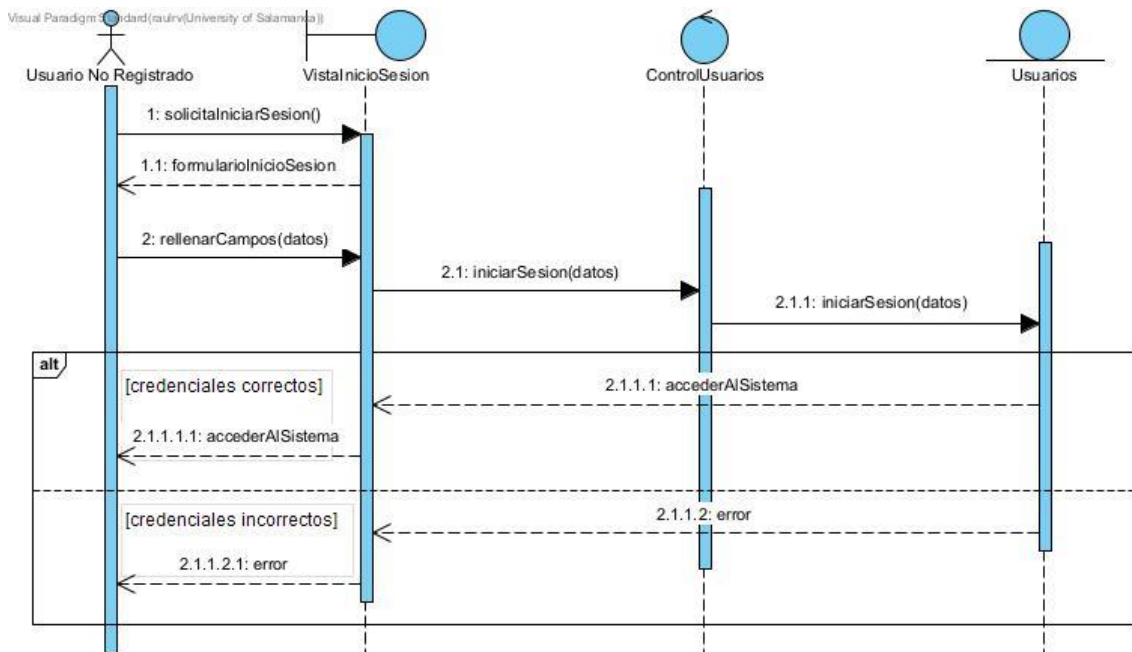


Figura 6. Diagrama de secuencia - Iniciar Sesión

5.5.3. Propuesta de arquitectura

La propuesta de arquitectura definida para nuestro sistema es la que podemos observar en la Figura 8.

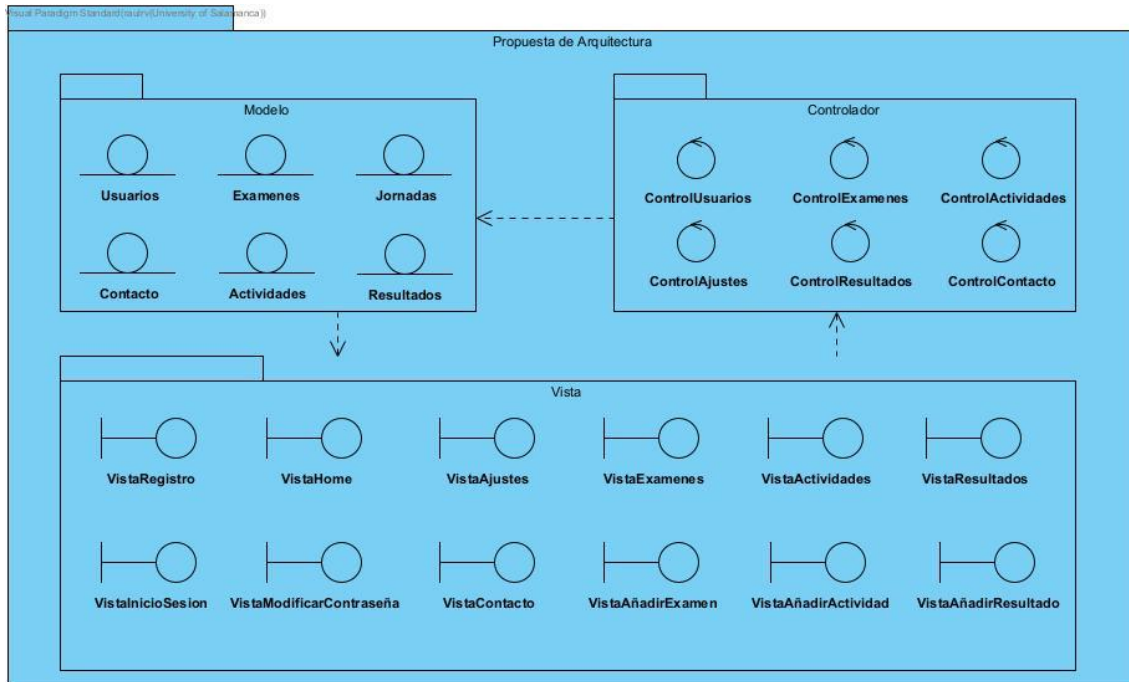


Figura 7. Propuesta de arquitectura

5.6. DISEÑO

En el diseño del sistema software se van a modelar los casos de una forma más parecida a la realidad, con clases existentes.

Una extensión de este apartado se puede encontrar en el Anexo IV – Diseño del Sistema.

5.6.1. Patrones Arquitectónicos

Se explicarán los patrones utilizados para organizar la estructura de nuestro software.

5.6.1.1. MVVM

El patrón de diseño llamado Model-View-ViewModel es uno de los patrones más utilizados en el desarrollo de aplicaciones, en especial en aplicaciones Android, separando la lógica de negocio y la lógica de presentación de la interfaz de usuario.

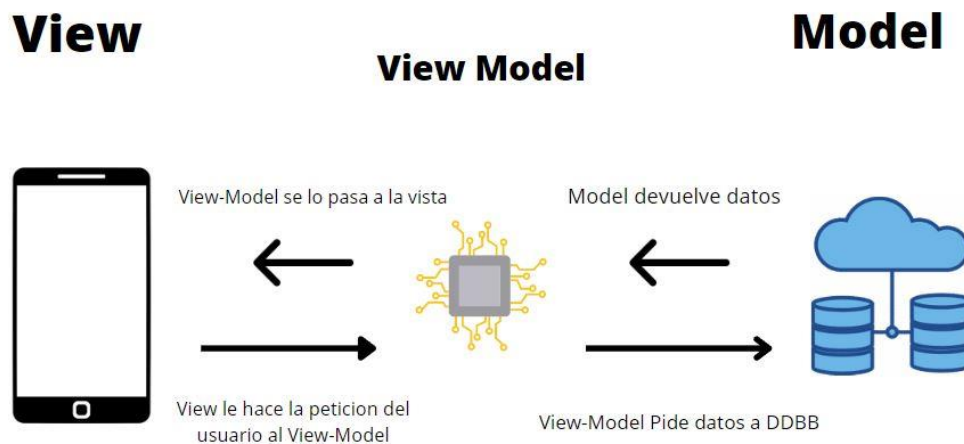


Figura 8. Patrón MVVM

Este patrón MVVM consta de tres componentes:

1. **Modelo (Model):** Representa los datos de nuestra aplicación, siguiendo una estructuración de estos 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.

5.6.1.2. Cliente-Servidor

Este patrón Cliente-Servidor es uno de los más utilizados comúnmente en aplicaciones no muy grandes donde tenemos bases de datos. En aplicaciones con mayor escala puede resultar más problemático ya que una posible caída del servidor podría dejar colgadas todas las funcionalidades de la aplicación.

En nuestro caso, lo vamos a utilizar para la conexión con la base de datos Firebase Firestore, donde el almacenamiento se encuentra en un servidor propio de Google (proveedor de los servidores de Firebase). Por otro lado, los clientes van a ser los usuarios de la aplicación, que cuando hagan ciertas operaciones, van a necesitar la respuesta del servidor con los datos solicitados. La comunicación entre estos va a tener lugar gracias a la propia API proporcionada por Firebase.

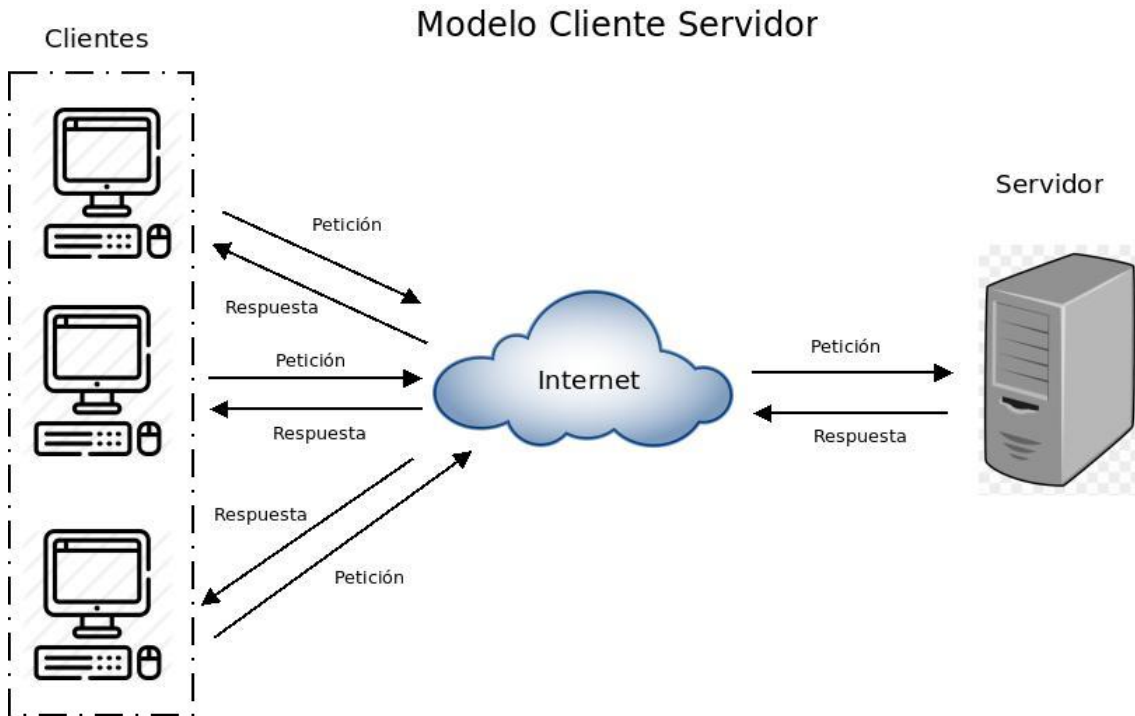


Figura 9. Esquema Cliente-Servidor

5.6.2. SUBSISTEMAS DE DISEÑO

Para dividir el trabajo y simplificar por tanto las funcionalidades, he decidido dividir el sistema en diferentes subsistemas, siguiendo la estructura basada en el patrón de diseño MVVM. Dentro de cada subsistema se van a poder encontrar las funciones más destacadas que nos van a servir para poder desarrollar los requisitos solicitados por el cliente.

Vamos a tener los siguientes subsistemas:

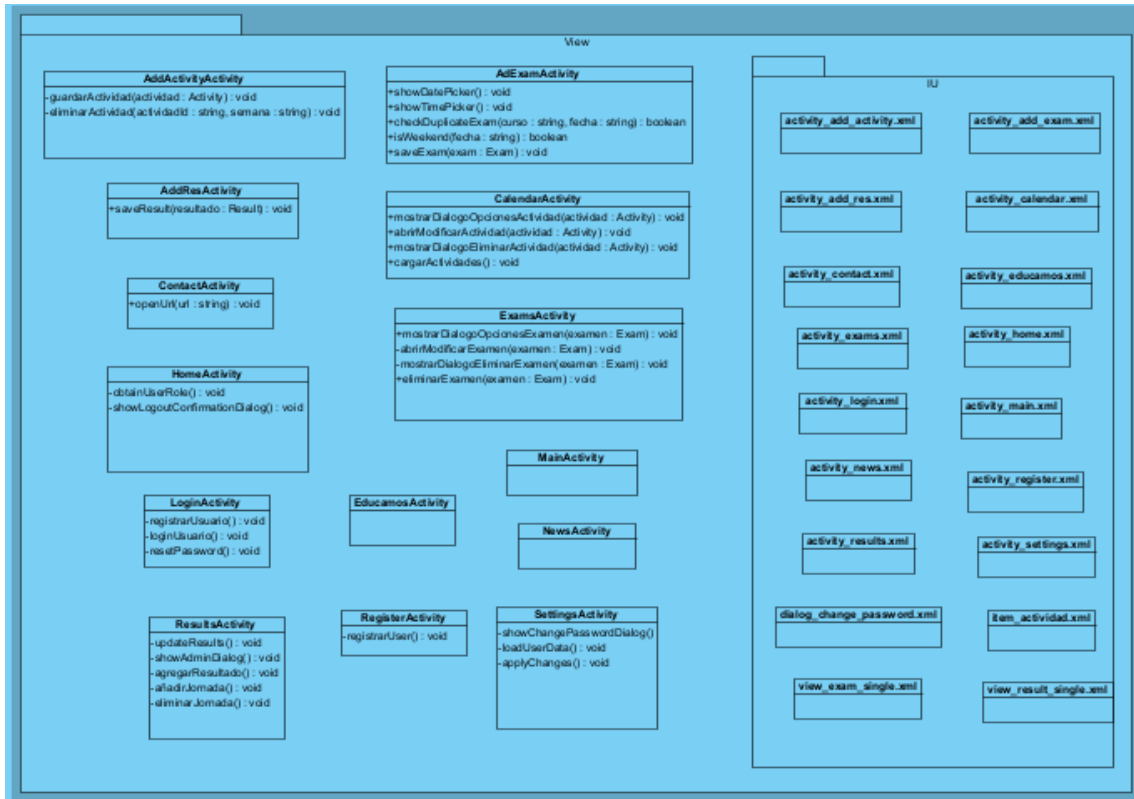


Figura 10. Subsistema View

Dentro de este primer subsistema se encuentra todo lo relacionado con la vista, tanto la parte más gráfica como es el caso de los ficheros XML como los ficheros Java donde se inicializan todos los componentes. Es la vista la encargada de interactuar con el ViewModel para transmitir lo que el usuario le proporciona.

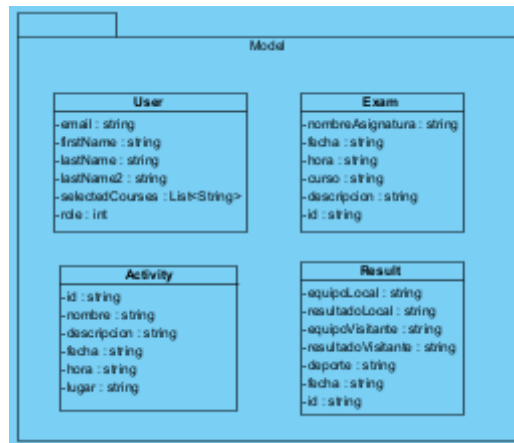


Figura 11. Subsistema Model

En cuanto al subsistema Model, es aquel que se va a almacenar en la base de datos Firebase Firestore cuando hagamos una petición de inserción de datos. También va a ser lo que se recupere cuando queramos ver cierta información.

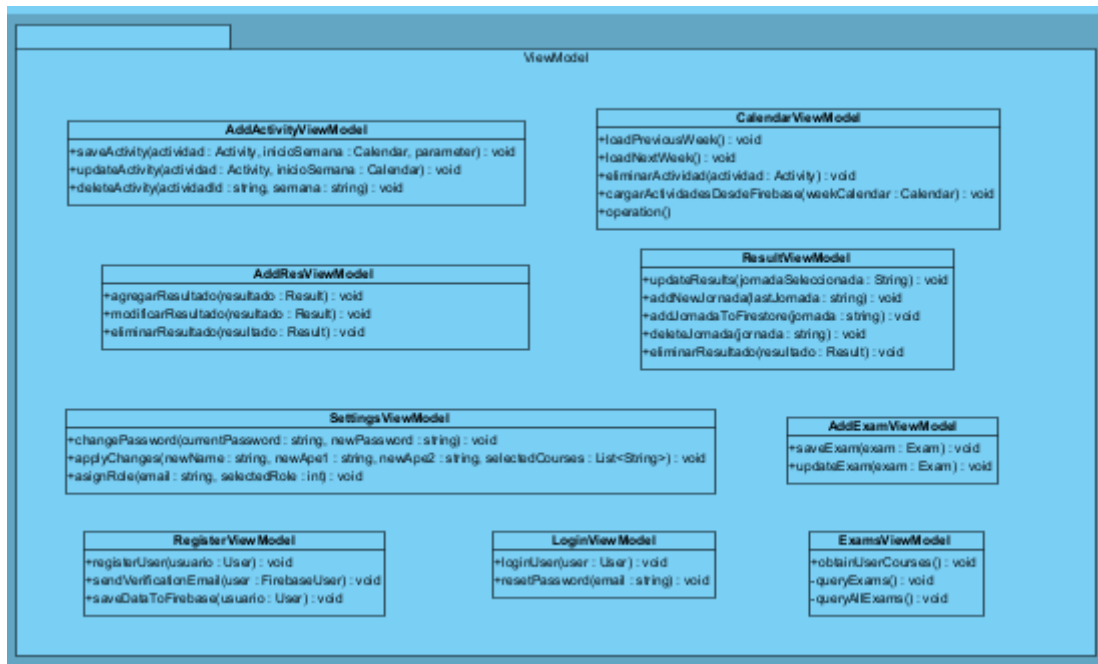


Figura 12. Subsistema ViewModel

Es el “corazón” de la aplicación, el encargado de la comunicación entre la vista del usuario y la base de datos, encargándose de realizar todos los cálculos, operaciones, etc.

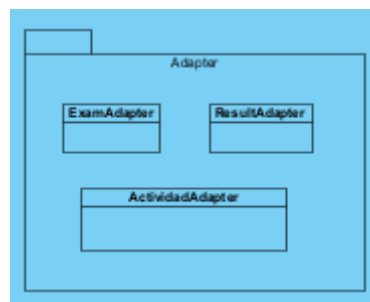


Figura 13. Subsistema Adapter

Este último subsistema es el encargado de presentar cada uno de los objetos recuperados de la base de datos en la vista con un formato específico, dentro de la ventana correspondiente.

Uniendo todos estos subsistemas, podemos observar el siguiente subsistema completo:

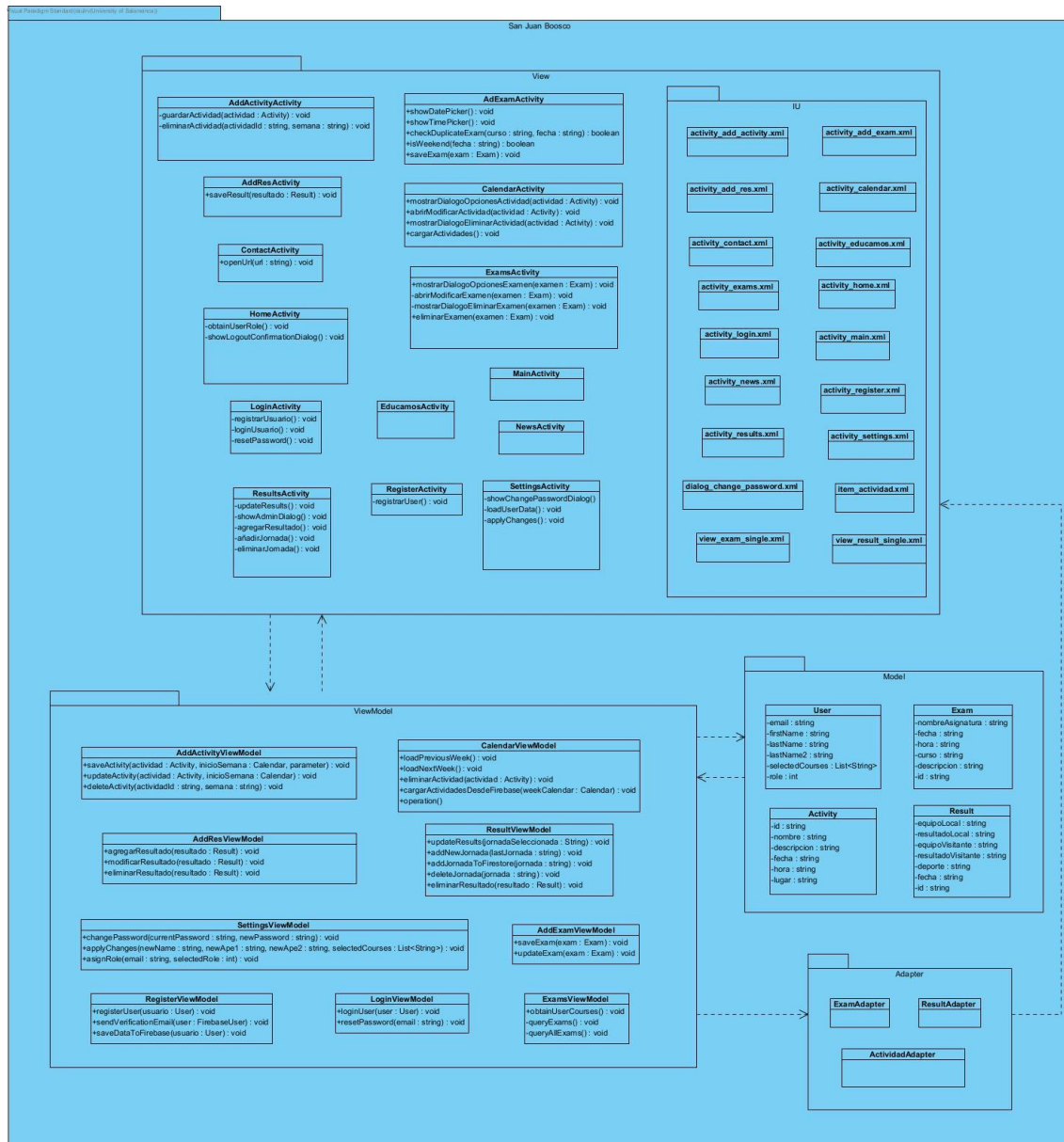


Figura 14. Subsistema completo

5.6.3. REALIZACIÓN DE LOS CASOS DE USO

Una vez tenemos todo organizado en subsistemas, podemos realizar los diferentes casos de uso con las diferentes clases. Estos casos de uso van a aproximarse mayormente al entorno real, con las clases y funciones necesarias.

Un ejemplo de un caso de uso realizado es el de la Figura 16, este caso de uso trata de la funcionalidad de modificar un examen el cual se encuentra ya añadido. Esta es una de las funcionalidades más complejas, que involucra a una mayor cantidad de clases que cualquier otra funcionalidad.

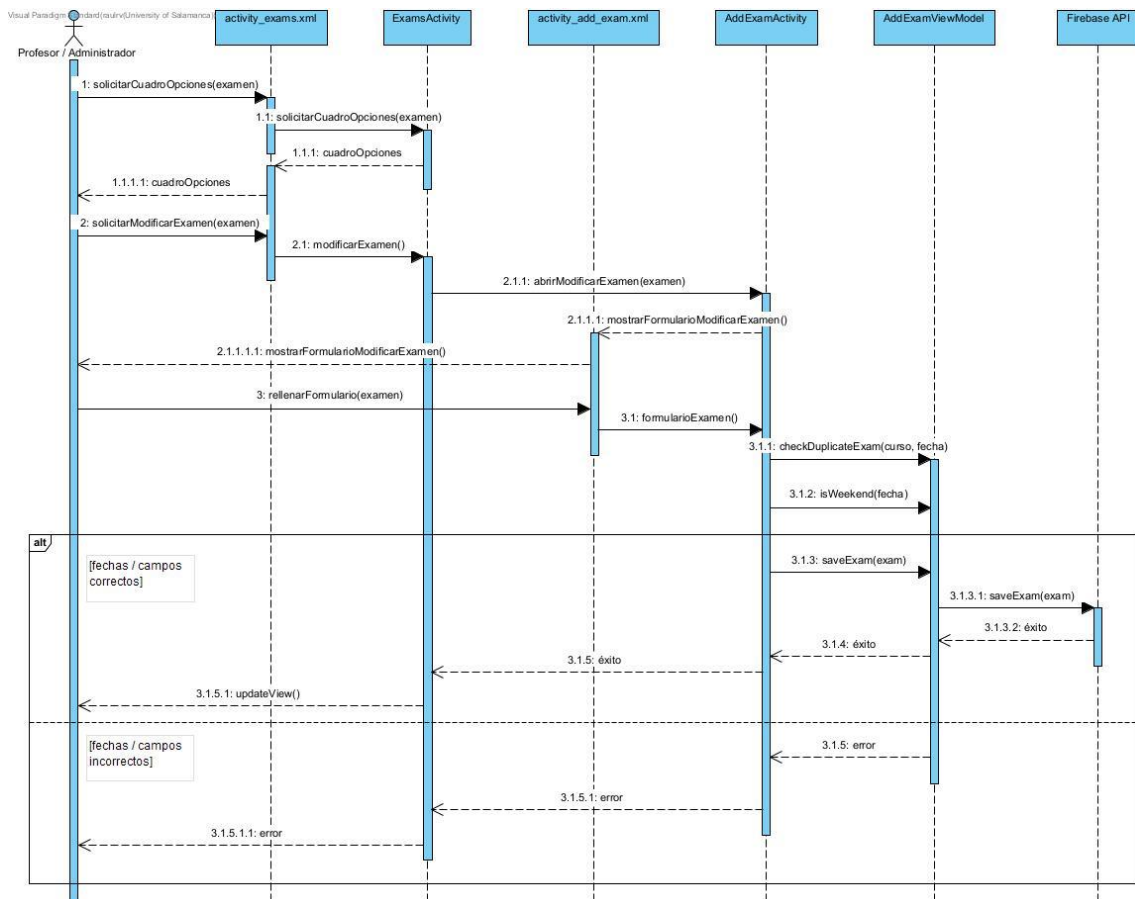


Figura 15. Realización caso de uso – Diseño

El resto de los casos de uso pueden verse en detalle en el Anexo IV – Diseño del Sistema.

5.6.4. DISEÑO DE LA BASE DE DATOS

La base de datos que hemos utilizado, como ya hemos comentado, es Firebase Firestore, y la organización de esta es por colecciones, dentro de las cuales pueden contenerse una serie de documentos donde se encuentren almacenados nuestros datos.

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.

5.6.5. DISEÑO DE LA INTERFAZ

Para el diseño de nuestra interfaz, he buscado un diseño sencillo para potenciar la usabilidad en todos los usuarios, y evitar así que puedan tener problemas a la hora de la navegación por la aplicación. Esto se debe a que la aplicación está orientada a toda clase de públicos, incluyendo usuarios con conocimientos escasos o incluso nulos de informática. Por ello, gracias a un sencillo diseño, vamos a reducir la dificultad de navegación por la misma, cumpliendo con todas las funcionalidades propuestas y solicitadas por el cliente.

Respecto a los colores utilizados en la interfaz, me he basado en los colores que ha utilizado el propio centro en su [página web personal](#).



Imagen 13. Página web centro

A partir de estos colores, gracias a la siguiente herramienta: <https://colorpalette.imageonline.co/es/>, generé una paleta de colores, la cual he utilizado para el diseño de todas las interfaces, mezclando los diferentes colores.



Imagen 14. Paleta de colores

El resultado de la interfaz final es el de las imágenes 15 y 16.



Imagen 15. Pantalla Login final



Imagen 16. Pantalla Main final

5.6.6. VISTA DE DESPLIEGUE

En este último apartado del diseño del sistema software se van a mostrar los nodos y los artefactos que necesitamos para que el sistema se ponga en ejecución.

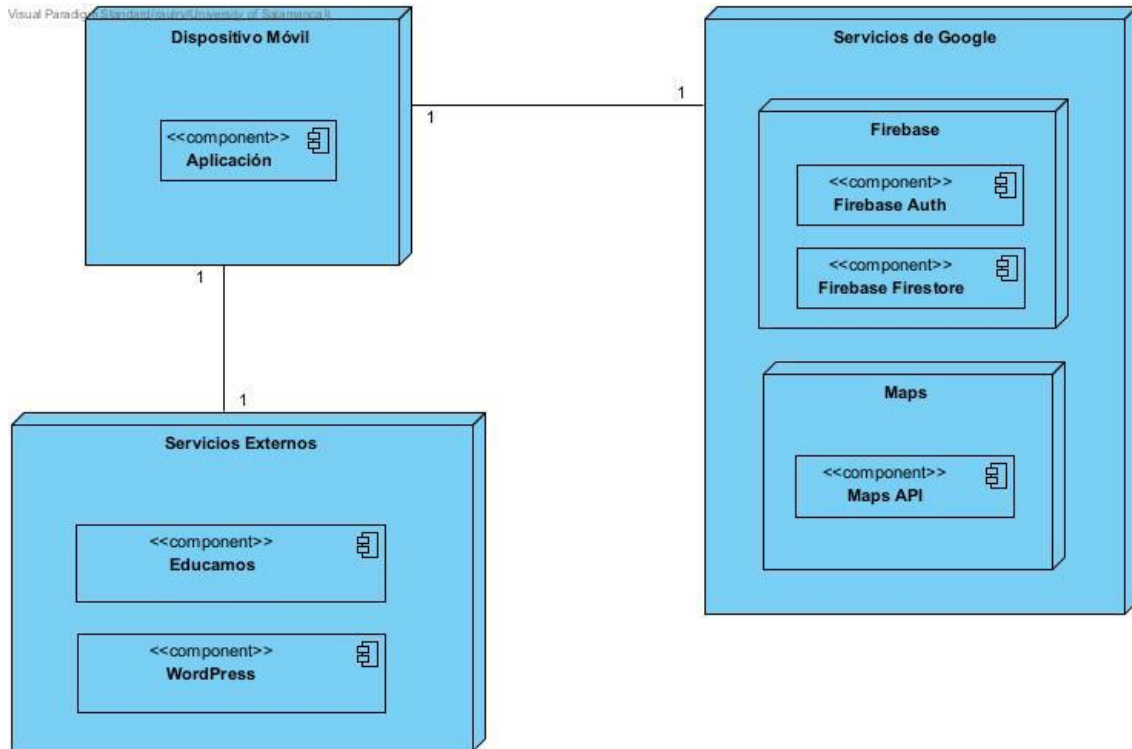


Figura 16. Vista de despliegue

En nuestra vista de despliegue podemos ver 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.

5.7. IMPLEMENTACIÓN

Para la implementación de nuestro sistema software he utilizado Java como lenguaje principal y XML para la parte más gráfica, la que el usuario puede ver directamente, desarrollando en el entorno de trabajo de Android Studio.

Para la división de nuestro proyecto, he creado diferentes paquetes siguiendo el patrón Model-View-ViewModel, dentro de cada cual se van a encontrar las clases correspondientes con sus respectivas funciones.

A partir de las diferentes clases de cada paquete se ha generado una documentación adicional gracias a la herramienta Javadoc, la cual, gracias a los comentarios de las clases, explica lo que hace cada una de ellas, proporciona información sobre las funciones tal como sus parámetros, valores de retorno, etc. Esta información puede encontrarse en el archivo HTML contenido en la carpeta Javadoc.

A pesar de seguir una estructuración basada en el modelo MVVM, el desarrollo del sistema ha sido siguiendo las funcionalidades, e implementando una por una hasta tener todo el proyecto completo. Estas funcionalidades son las siguientes:

- Gestión de usuarios: En esta funcionalidad, siendo la primera de todas ellas en desarrollar ya que considero que es la base de la aplicación, he utilizado Firebase Auth, la cual permite registrar usuarios y autenticarlos de manera más segura. Además, para poder diferenciar los usuarios por roles, también he almacenado sus datos (excluyendo la contraseña) en la base de datos de Firebase Firestore, asignando un campo que va a indicar el rol que tienen asignado. Gracias a estar almacenados en la base de datos de Firebase, pueden recuperarse y modificarse los datos en caso de que sea necesario.
- Gestión de exámenes: En este apartado he conseguido que se permitan añadir, modificar y eliminar los exámenes que los profesores deseen. Además, los usuarios de la familia podrán ver solamente los exámenes de los cursos en los cuales se encuentran sus hijos matriculados, evitando tener una carga excesiva de información.
- Gestión de actividades: Este es sin duda el apartado más complicado, que más tiempo me ha llevado y problemas me ha generado, debido a que mi primera idea era la de implementar un calendario ya desarrollado como el de Google, donde se pudiesen ir añadiendo actividades, sin embargo, esta opción la descarté debido a las dificultades generadas, y me decante por crear mi propio calendario por semanas. Con esto se consigue poder navegar entre semanas viendo las actividades (sobre todo actividades extraescolares) que los profesores o administradores han ido añadiendo. Es similar a una agenda.
- Gestión de resultados: Esta funcionalidad también resultó bastante compleja, ya que busque separar los diferentes resultados por jornadas, teniendo que almacenar no solo los resultados en la base de datos, sino también las jornadas. La funcionalidad desarrollada cumple con lo propuesto por el centro, obteniendo un resultado bastante visual y funcional.

La organización del código se ha hecho en paquetes, siguiendo el patrón MVVM, donde se han ido añadiendo las clases generadas para el correcto desarrollo del sistema completo.

5.7.1. Paquete Model

En este paquete se encuentran las clases con el modelo de la aplicación:

- Actividad.java
- Exam.java
- Resultado.java
- User.java

5.7.2. Paquete View

En el paquete de la vista vamos a tener las clases Java solamente, ya que las clases XML se encuentran dentro del paquete res/layout. Estas clases Java van a ser las encargadas de inicializar los componentes de las clases XML y de ponerse en contacto con el ViewModel:

- AddActivityActivity.java
- AddExamActivity.java
- AddResActivity.java
- CalendarActivity.java
- ContactActivity.java
- EducamosActivity.java
- ExamsActivity.java
- HomeActivity.java
- LoginActivity.java
- MainActivity.java
- NewsActivity.java
- RegisterActivity.java
- ResultsActivity.java
- SettingsActivity.java

5.7.3. Paquete ViewModel

Este paquete es el encargado de intermediar entre la vista y el modelo, realizando todas las operaciones y conexiones necesarias. Está compuesto por las siguientes clases:

- AddActivityViewModel.java
- AddExamViewModel.java
- AddResViewModel.java
- CalendarViewModel.java
- ExamsViewModel.java
- LoginViewModel.java
- RegisterViewModel.java
- ResultsViewModel.java
- SettingsViewModel.java

5.7.4. Paquete Adapter

El paquete Adapter contiene aquellas clases encargadas de presentar los elementos individuales recuperados de la base de datos en la vista, otorgándoles una estructura definida en un archivo XML. Las clases Adapter son las siguientes:

- ActividadAdapter.java
- ExamAdapter.java
- ResultAdapter.java

5.7.5. Paquete Utils

El paquete Utils contiene solamente una clase encargada de realizar diferentes funciones relacionadas con los objetos Calendar y todo lo relacionado con las fechas:

- CalendarUtils.java

5.8. PRUEBAS

Las pruebas del software desarrollado se han ido realizando a medida que se iban implementando nuevas funcionalidades, de forma incremental.

Cada vez que una funcionalidad se desarrollaba y se ponía en funcionamiento dentro de la aplicación, esta se probaba de manera local para comprobar que todo estaba correctamente, evitando así posibles fallos futuros en iteraciones más avanzadas del ciclo de vida, los cuales podrían ocasionar situaciones mucho más problemáticas.

Una vez se implementaron todas las funcionalidades, se probó el subsistema por completo, seleccionando una serie de usuarios y asignándoles unos roles para comprobar que todos ellos cumplían con las funciones que se les habían otorgado.

Cuando se terminó de probar y corregir nuestro sistema software, se le pasó una versión al centro educativo, para que pudiesen probarlo ellos y comprobar que el funcionamiento de la aplicación era el esperado. Una vez realizaron sus propios tests, me proporcionaron un informe con los fallos detectados y algunos detalles que se podrían corregir, sobre todo cambios visuales, buscando una interfaz más visual.

A partir de las sugerencias propuestas por el centro, realicé una reestructuración de la interfaz, creando un diseño con mayor organización y más detallista, donde todo estaba centrado, los bordes de los iconos tenían una forma redondeada, añadí opciones para contraer y expandir las descripciones en caso de que estas fuesen excesivamente largas...

Tras estas correcciones, se lanzó una versión final de la aplicación, la cual se probó también, comprobando correctamente la sincronización con la base de datos, el funcionamiento de los roles de usuarios, la modificación de todos los campos, etc.

5.9. FUNCIONALIDADES

En este apartado de la memoria se va a explicar todas y cada una de las funcionalidades de nuestra aplicación para que los diferentes usuarios puedan hacer uso de ella sin tener ninguna dificultad.

En sí, las interfaces de nuestra aplicación son muy sencillas para evitar que los usuarios sin mucha experiencia en el sector tecnológico puedan hacer uso de la misma sin que se les presenten dificultades.

Algunas de las funcionalidades solamente van a estar disponibles para usuarios con ciertos privilegios, es decir, profesores y administradores.

Todas las funcionalidades están explicadas con más detalle en el Anexo VI – Manual del Usuario.

Las principales funcionalidades son las cuatro mencionadas anteriormente en el apartado de implementación, por lo que vamos a ver que se hace dentro de cada una de ellas en detalle.

5.9.1. GESTIÓN DE USUARIOS

5.9.1.1. LOGIN

En la pantalla de login o de inicio de sesión es donde el usuario va a poder introducir sus credenciales para acceder al contenido de la aplicación. También va a tener la posibilidad de registrarse en caso de que no lo esté y de recuperar la contraseña en caso de que se le haya olvidado introduciendo su correo electrónico.



Figura 17. Pantalla Login

5.9.1.2. REGISTRO

En esta ventana es donde el usuario deberá introducir sus datos, completando todos los campos para poder registrarse en el sistema. Una vez hecho esto, se le enviará una verificación al correo electrónico, la cual deberá confirmar para poder tener acceso completo a su cuenta.

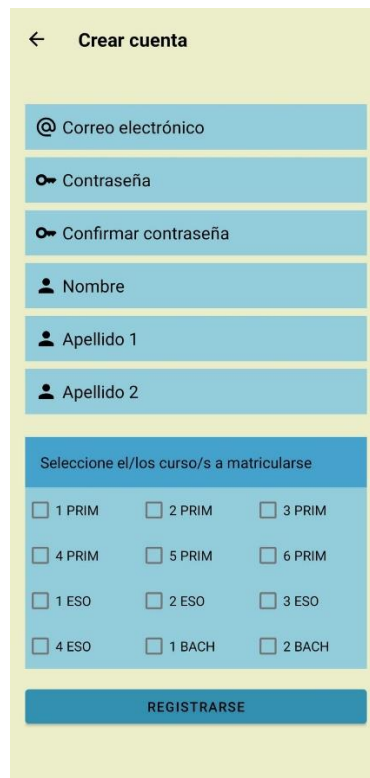


Figura 18. Pantalla Registro

5.9.1.3. OLVIDO DE CONTRASEÑA

Al pulsar en “¿Ha olvidado la contraseña?” desde la pantalla de Login, se nos va a mostrar el siguiente mensaje, en caso de haber introducido un correo en la pantalla de Login, sino mostrará un error. Si pulsamos en aceptar se nos enviará un correo electrónico para restablecer la contraseña y poner una nueva.

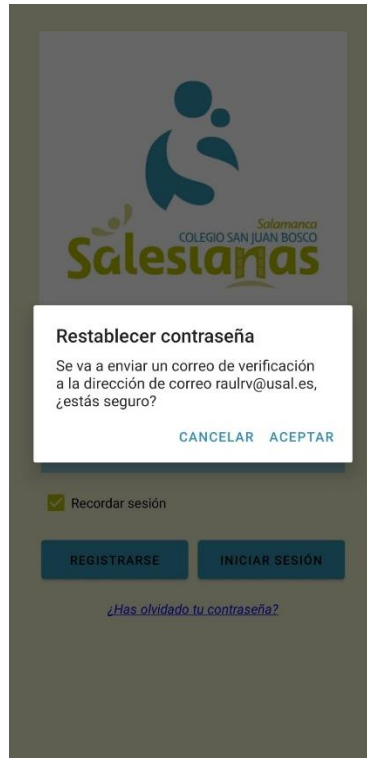


Figura 19. Restablecer contraseña

5.9.1.4. AJUSTES

La ventana de ajustes nos va a permitir modificar los datos de registro de un usuario (Nombre, apellidos y cursos matriculados) como podemos ver en la Figura 35, además de una opción para modificar la contraseña, como se puede ver en la Figura 36.

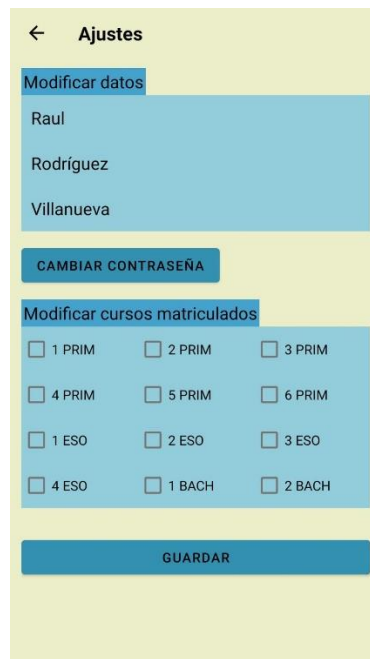


Figura 20. Pantalla Ajustes



Figura 21. Cuadro cambio de contraseña

5.9.1.5. ASIGNAR PERMISOS

Esta funcionalidad solamente va a estar disponible para administradores, encargados de asignar permisos a los profesores una vez estos hayan sido registrados. También pueden asignar el resto de los roles (padre/madre o administrador) en caso de que fuese necesario.

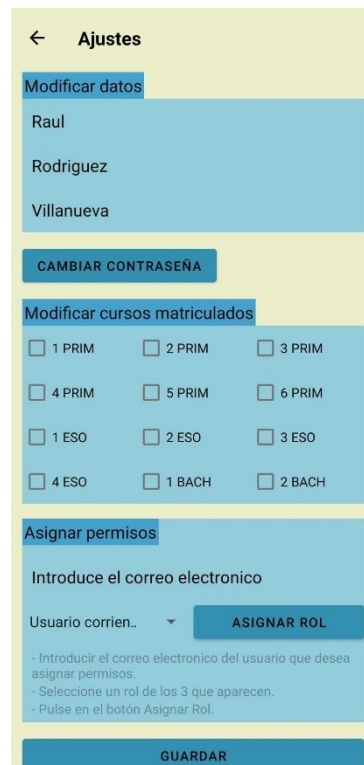


Figura 22. Pantalla Ajustes - Asignar Permisos

5.9.1.6. PANTALLA PRINCIPAL

Esta es la pantalla a través de la cual se va a poder acceder al resto de funcionalidades de la aplicación con un simple click. También existe la posibilidad de acceder a los ajustes de la cuenta o cerrar la sesión en la cual nos encontramos.



Figura 23. Pantalla principal

5.9.2. GESTIÓN DE EXÁMENES

5.9.2.1. VISUALIZAR EXÁMENES

En esta pantalla vamos a poder ver los exámenes almacenados en la base de datos, filtrados por los cursos en los cuales se encuentre matriculado el usuario. En caso de ser profesor o administrador, se van a poder visualizar todos los exámenes.



Figura 24. Pantalla calendario de exámenes

5.9.2.2. AÑADIR EXAMEN

Esta funcionalidad solamente estará disponible para profesores y administradores, y sirve para añadir exámenes a la base de datos, rellenando el formulario de la Figura 23.

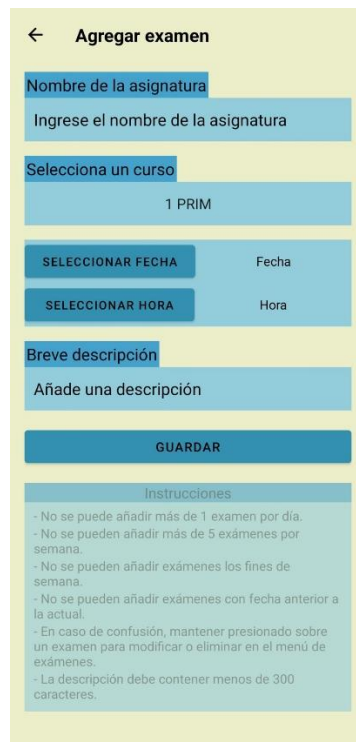


Figura 25. Pantalla Añadir Examen

5.9.2.3. MODIFICAR EXAMEN

Para modificar un examen, un usuario con privilegios va a mantener presionado sobre el examen que desee modificar, y le aparecerá un cuadro con las opciones de la Figura 24.

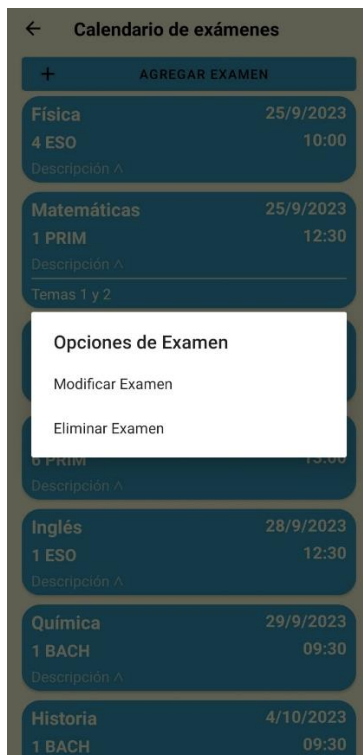


Figura 26. Cuadro opciones examen

Al presionar sobre Modificar Examen se va a abrir la ventana de Añadir Examen con el contenido del examen seleccionado ya cargado, permitiendo al usuario modificarlo. Una vez haya terminado de modificar, pulsará en el botón inferior de "MODIFICAR".

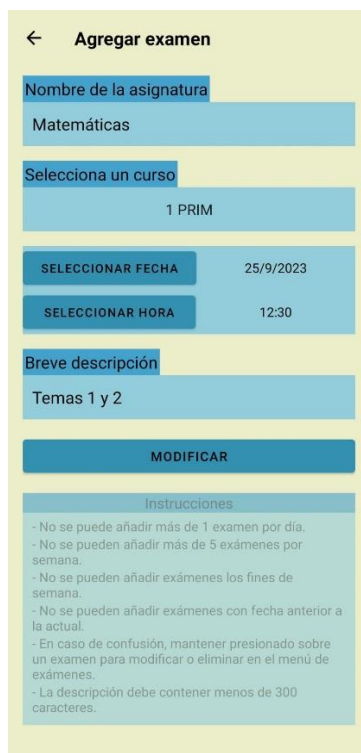


Figura 27. Pantalla Modificar Examen

5.9.2.4. ELIMINAR EXAMEN

Esta opción, al igual que la de Añadir Examen y la de Modificar Examen es solamente para usuarios con privilegios. Para eliminar un examen, el usuario deberá mantener presionado sobre el cuadro de opciones de la Figura 24 y seleccionar sobre “Eliminar Examen”. A continuación, se le mostrará un mensaje de confirmación, y en caso de aceptar, el examen será borrado de la base de datos y de la vista.

5.9.3. GESTIÓN DE ACTIVIDADES

5.9.3.1. VISUALIZAR CALENDARIO ESCOLAR

En esta ventana se va a poder ver el calendario de actividades organizado por semanas, pudiendo avanzar o retroceder entre las semanas para ir viendo cada una de las actividades programadas para las fechas que deseemos ver.



Figura 28. Pantalla calendario de actividades

5.9.3.2. AÑADIR ACTIVIDAD

Esta funcionalidad, activa solamente para los usuarios con privilegios, va a permitir añadir al calendario actividades rellenando el formulario que podemos visualizar en la Figura 27.

Figura 29. Pantalla Añadir Actividad

5.9.3.3. MODIFICAR ACTIVIDAD

Al igual que para los exámenes, tenemos la opción de modificar una actividad que hemos añadido a nuestro calendario. Para ello mantendremos presionado sobre la actividad que queramos modificar y podremos ver el cuadro de opciones de la Figura 28.



Figura 30. Cuadro Opciones de Actividad

Si pulsamos en “Modificar Actividad” se nos abrirá la ventana de “Añadir Actividad” con la actividad que hemos seleccionado cargada, y nos va a permitir modificar los campos que queramos. Una vez realizados todos los cambios pulsaremos en “GUARDAR”.



Figura 31. Pantalla Modificar Actividad

5.9.3.4. ELIMINAR ACTIVIDAD

Para eliminar una actividad, funcionalidad solamente activa para usuarios con privilegios, deberemos mantener presionado sobre la actividad que queramos eliminar y en el cuadro de la Figura 28 deberemos seleccionar “Eliminar Actividad”. De esta forma se nos presentará un mensaje de confirmación, el cual deberemos aceptar si queremos que la actividad se elimine.

5.9.4. GESTIÓN DE RESULTADOS

5.9.4.1. VISUALIZAR RESULTADOS DE PARTIDOS

En esta pantalla vamos a poder ver los resultados de los partidos en función de la jornada que tengamos seleccionada.



Figura 32. Pantalla Resultados

Los usuarios con privilegios van a tener un cuadro de administración como el de la Figura 31, donde van a poder añadir un resultado, añadir una jornada o eliminar una jornada.



Figura 33. Cuadro administración resultados

5.9.4.2. AÑADIR RESULTADO

Para añadir un resultado, desde el cuadro de administración de resultados deberemos pulsar en “Agregar Resultado”, y se nos abrirá una nueva pantalla con un formulario como el de la Figura 32, el cual deberemos rellenar para poder añadir un nuevo resultado.

Figura 34. Pantalla Añadir Resultado

5.9.4.3. MODIFICAR RESULTADO

Esta opción también es para usuarios con privilegios, y para poder modificar un resultado, al igual que en los casos anteriores, deberemos mantener presionado sobre uno de los resultados, y veremos de nuevo un cuadro de opciones como el de la Figura 33.



Figura 35. Cuadro Opciones de Resultados

Pulsaremos en “Modificar Resultado”, y se nos abrirá la ventana de “Añadir Resultado” con el resultado que queremos modificar cargado como podemos ver en la Figura 34.

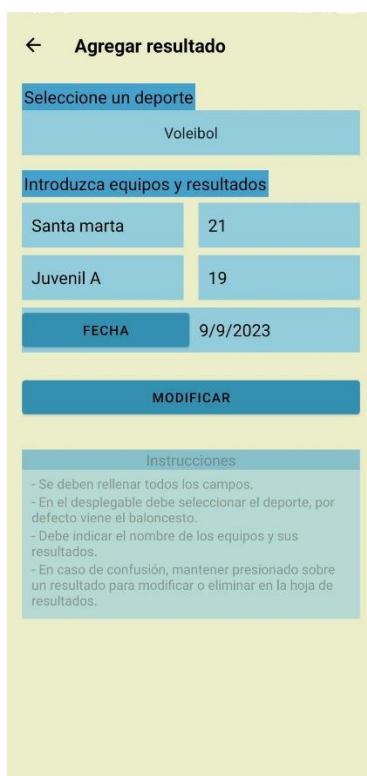


Figura 36. Pantalla Modificar Resultado

Tras realizar los cambios correspondientes, pulsaremos en “MODIFICAR” para establecer los cambios.

5.9.4.4. ELIMINAR RESULTADO

Esta funcionalidad, al igual que la de añadir y modificar un resultado, es solamente para usuarios con privilegios. Para eliminarlo mantendremos pulsado sobre el resultado y en el cuadro de la Figura 33 pulsaremos sobre “Eliminar Resultado”, aceptando la confirmación que se nos mostrará antes de que el resultado se elimine. En caso de Cancelar, se volverá al estado anterior.

5.9.5. FUNCIONES EXTERNAS

Estas funcionalidades son las que no dependen de nuestra aplicación, sino que derivan en otras aplicaciones externas ya creadas.

Vamos a encontrarnos un WordPress con las noticias del centro (Figura 38), un portal educativo llamado Educamos donde se publicarán las notas de los exámenes e información confidencial más en detalle (Figura 39) y una página de contacto donde se deriva al resto de redes sociales del centro, además de poder ver su ubicación en Google Maps, teléfono y correo electrónico (Figura 40).



Figura 37. Ventana Noticias

The screenshot shows a mobile browser interface with the URL `sso2.educamos.com`. At the top, there is a home icon, a lock icon, and a microphone icon. The main content features the logo of 'Colegio San Juan Bosco' with the text 'Salesianas' and 'COLEGIO SAN JUAN BOSCO'. Below the logo, it says 'Código de centro: 1281'. There are two input fields: 'Nombre de usuario' and 'Contraseña'. A blue button labeled 'ACCEDER' is positioned below the password field. Underneath the button, there is a link that says 'He olvidado mis claves de acceso'. At the bottom, it says 'Powered by sm educamos'.

Figura 38. Ventana Educamos



Figura 39. Ventana Contacto

6. CONCLUSIONES

El proyecto desarrollado ha cumplido con los objetivos que se propusieron al inicio del mismo, tanto funcionalmente, donde se han desarrollado todas las funcionalidades que se definieron inicialmente, como personalmente, donde he aprendido mucho a lo largo de un proceso largo y lento, donde no he dejado de aprender cosas nuevas a medida que avanzaban los días.

Antes de comenzar con el proyecto, nunca había realizado un proyecto completo, desde la toma de requisitos hasta la realización de una documentación final, siguiendo un modelo basado en el Proceso Unificado, por lo que primeramente he tenido que tomarme bastante tiempo organizándome y buscando todo tipo de información para consolidar una base. Tampoco había realizado nunca trabajos para dispositivos móvil, por lo que la instalación y configuración del entorno de desarrollo también fue algo novedoso, a pesar de ser bastante sencillo. Cabe mencionar que dentro de Android Studio hay dos posibles lenguajes para el desarrollo del código, y elegí Java ya que era aquel que ya conocía gracias a estos años en el Grado.

Me ha resultado una experiencia muy buena por el hecho de poder ampliar aquellos conocimientos que se han ido viendo a lo largo de los diferentes años en las diferentes asignaturas del Grado, además me ha servido también para refrescar aquellos que se habían quedado un poco olvidados con el paso del tiempo. También me ha resultado positivo saber que la aplicación desarrollada es para un cliente real, con requisitos reales que se han ido tomando a lo largo del tiempo y refinando, con pruebas funcionales, etc.

Por último, quiero mencionar que me gustaría poder seguir desarrollando la aplicación para poder proporcionar al centro unas nuevas mejoras de cara a futuros cursos o futuras versiones de la aplicación, para tener así una aplicación más completa.

7. LÍNEAS FUTURAS

Una vez finalizado el proyecto, se me han ido ocurriendo nuevas funcionalidades que podrían ser incorporadas de cara a próximas actualizaciones:

- Mejora en el sistema de usuarios: Poder añadir una opción para administradores donde se observen todos los usuarios del sistema, pudiendo filtrarlos, y así poder hacer una gestión más sencilla y directa de estos. Actualmente al no estar diseñada para un número excesivo de usuarios no es necesario, pero en caso de que la aplicación crezca sería un cambio bastante notable.
- Mejora en las funcionalidades de exámenes, automatizando el relleno de los campos a la hora de añadir un nuevo examen a la lista, con una lista de asignaturas diseñada por los profesores, cursos extra, etc.
- Implementación de notificaciones, proceso el cual puede ser muy útil si la aplicación comienza a usarse de manera continua, ya que los padres podrían ver todas las nuevas notificaciones sin necesidad de acceder al contenido de la aplicación.
- Integración de las páginas externas dentro de la propia aplicación, como Educamos y el WordPress de las noticias. Este proceso quizá sea más complicado, pero puede servir para concentrar todo lo relacionado con el centro en una misma aplicación, algo que va a permitir a los padres disponer de una mayor organización.

8. Referencias

Almenara, J. C. (2015). *Reflexiones educativas sobre las tecnologías de la información y la comunicación (TIC)*. Revista Tecnología, Ciencia y Educación, 19-27.

Documentación de Android Studio. (2023). Obtenido de <https://developer.android.com/docs?hl=es-419>

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

Documentación de Gradle. (2023). Obtenido de <https://docs.gradle.org/current/userguide/userguide.html>

Documentación de Java. (2023). Obtenido de <https://aws.amazon.com/es/what-is/java/>

Documentación de XML. (2023). Obtenido de <https://aws.amazon.com/es/what-is/xml/>

Git. (2023). Obtenido de <https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>

Jacobson, I. B. (1999). *The unified process*. IEEE Software, 16(3), 96.

Javadoc. (2023). Obtenido de <https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>

Larman, C. R. (1999). *UML y Patrones: Introducción al análisis y diseño orientado a objetos (Vol. 2)*. Prentice Hall.

León, A. R. (2011). *CRUD-PG*. Revista Cubana de Ciencias Informáticas, 5(1), 1-8.

Microsoft Project. (2023). Obtenido de <https://www.microsoft.com/es-es/microsoft-365/project/project-management-software>

Microsoft Word. (2023). Obtenido de <https://learn.microsoft.com/en-us/office/client-developer/word/word-home>

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*.

Visual Paradigm. (2023). Obtenido de <https://www.visual-paradigm.com/>