

Memoria del proyecto

SISTEMA DE PRÉSTAMOS DE LIBROS DE TEXTO EN CENTROS EDUCATIVOS

Trabajo de Fin de Grado
INGENIERÍA INFORMÁTICA



**VNiVERSiDAD
DSALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

Septiembre de 2021

Marta Diego Díez

André Sales Mendes

Juan Francisco de Paz Santana

Gabriel Villarrubia González

Certificado de los tutores

D. André Filipe Sales Mendes, D. Juan Francisco de Paz Santana y D. Gabriel Villarrubia González, profesores del Departamento de Informática y Automática de la Universidad de Salamanca.

CERTIFICAN:

Que el trabajo titulado “Sistema de préstamos de libros de Texto en centros educativos” ha sido realizado por Dña. Marta Diego Díez, con DNI 70914095N y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Grado de Ingeniería Informática de esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a 12 de enero de 2022.

Resumen

El presente Trabajo de Fin de Grado ha consistido en el diseño y desarrollo de una aplicación de tipo Android para facilitar la gestión del préstamo de libros de texto en centros educativos. La aplicación móvil desarrollada, que lleva por nombre BookApp, permitirá a los centros escolares contar con un sistema sencillo por el que los usuarios podrán reservar los libros que necesiten y los administradores podrán organizar más fácilmente todo lo que conlleva el préstamo, almacenando los datos de los libros y los usuarios de manera que sea fácil comprobar el estado de los mismos.

A lo largo de esta Memoria se irán presentando los pasos seguidos durante todo el proceso de creación, diseño e implementación de la aplicación.

Para el desarrollo del sistema se ha utilizado la herramienta Android Studio. Dicha herramienta que es un entorno de desarrollo integrado (IDE) es la herramienta oficial de Android para la creación de sus aplicaciones.

Durante el desarrollo de este proyecto se ha seguido el marco de trabajo del Proceso Unificado en todo lo que se ha podido tanto en el análisis, como en el diseño y el desarrollo de la aplicación final.

El objetivo que se persigue con la realización de este proyecto es la creación de un sistema que facilite la reutilización de los libros de texto a través de operaciones de préstamo y donación de los ejemplares que utilizan en los centros educativos.

Llegados al final del proyecto, el trabajo culminará con la aplicación móvil BookApp, aplicación de tipo Android, escrita con lenguaje java y cuya base de datos es Firestore Database.

Palabras clave: libros, préstamo, colegio, donaciones, reserva, peticiones.

Abstract

This end-of-degree-project has consisted of the design and development of an Android-type application to facilitate the management of the loan of textbooks in educational centers. The mobile application developed, which is called BookApp, will allow schools to have a simple system by which users can reserve the books they need and administrators can more easily organize everything that the loan entails. At the same time, they will be able to store data from books and users so it will be easier to check their status.

Throughout this document, the steps followed in the process of creating, designing and implementing the application will be presented.

For the development of the system, the Android Studio tool has been used. This tool, which is used to officially create Android applications, is the integrated development environment (IDE).

During the development of this project, the Unified Process framework has been followed as far as possible both in the analysis, as well as in the design and development of the final application.

The objective pursued with the realization of this project is the creation of a system that facilitates to re-use textbooks used in educational centers through loan and donation operations.

At the end of the project, the work will culminate with the BookApp mobile application, an Android-type application, written with Java language and whose database is Firestore Database.

Keywords: book, borrowing, school, donations, reservation, requests.

Contenido

CERTIFICADO DE LOS TUTORES	3
Resumen	5
Abstract.....	6
Índice de ilustraciones	9
Índice de tablas	11
1. Introducción.....	12
2. Objetivos.....	14
2.1. Objetivos Software	14
2.2 Objetivos técnicos	15
2.3 Objetivos personales.....	15
3. Técnicas y herramientas	17
3.1. Lenguajes utilizados	17
3.1.1 Java	17
3.1.2 XML	17
3.2 Herramientas utilizadas	17
3.2.1 Android Studio	17
3.2.2 Firebase.....	18
3.2.3 Librerías.....	19
3.3 Herramientas CASE	20
3.3.1 Visual Paradigm	20
3.3.2 REM	21
3.3.3 EZEstimate	23
3.3.4 MS Project.....	23
4. Aspectos relevantes del desarrollo	25
4.1 Marco de trabajo.....	25
4.2 Estimación de costes.....	27
4.3 Especificación de requisitos.	27
4.4 Análisis del sistema Software.....	33
4.4.1 Modelo de dominio.....	33
4.4.2 Realización de casos de uso.....	33
4.5 Diseño del sistema.....	35
4.5.1 Patrón MVC.....	35
4.5.2 Base de datos Firestore	38
4.6 Implementación	39
5. Pruebas del sistema.....	41
6. Resultado final de la aplicación.....	42

7. Conclusiones.....	51
8. Líneas de trabajo futuras.....	52
9. Referencias	53

Índice de ilustraciones

Ilustración 1: Librerías Firebase	19
Ilustración 2: Librería Awesome Validation	20
Ilustración 3: Librería escaneo de códigos	20
Ilustración 4: Diagramas realizados con Visual Paradigm	21
Ilustración 5: Herramienta REM	22
Ilustración 6: Índice del proyecto en REM.....	22
Ilustración 7: Herramienta EZEstimate	23
Ilustración 8: Tareas plan de proyecto.....	24
Ilustración 9: Fases del Proceso Unificado.....	26
Ilustración 10: Iteraciones de nuestro sistema	26
Ilustración 11: Resumen EZEstimate	27
Ilustración 12: Actores del sistema.....	29
Ilustración 13: Diagrama de casos de uso del paquete de gestión de usuarios.....	30
Ilustración 14: Diagrama de clases	33
Ilustración 15: Diagrama de secuencia Eliminar libro.	34
Ilustración 16:Arquitectura.....	35
Ilustración 17:Patrón MVC	36
Ilustración 18: Diagrama de subsistemas	36
Ilustración 19: Vista arquitectónica de capas	37
Ilustración 20: Diagrama de caso de uso registro de usuario	37
Ilustración 21: Diagrama de despliegue	38
Ilustración 22: Diseño de la base de datos.....	39
Ilustración 23: Colecciones de la base de datos	39
Ilustración 24: Paquete Activitys.....	40
Ilustración 25: Paquete adapter.....	40
Ilustración 26:Paquete modelo	41
Ilustración 27: layouts de la aplicación	41
Ilustración 28: Pantalla de inicio	42
Ilustración 29: Registro de usuario	43
Ilustración 30: Recuperar contraseña	43
Ilustración 31: Menú principal	44
Ilustración 32: Reservar libro	45

Ilustración 33: Mis peticiones.....	45
Ilustración 34: Donaciones	46
Ilustración 35:Estado de las donaciones y peticiones pendientes.....	47
Ilustración 36:Menú administradores	48
Ilustración 37: Peticiones pendientes	48
Ilustración 38: Donaciones por aprobar.....	49
Ilustración 39: Listar todos los prestamos	50

Índice de tablas

Tabla 1: OBJ-0001 Gestion de usuarios	28
Tabla 2: IRQ-0001 Datos de los usuarios registrados	30
Tabla 3: Registro de usuario	32
Tabla 4: Matriz de Rastreabilidad	32

1. Introducción

En la actualidad, todas las familias con hijos escolarizados tienen la necesidad cada año de disponer de libros de texto. La adquisición de estos libros conlleva importantes costes económicos a los que las familias tienen que hacer frente al comienzo del nuevo curso escolar.

En los últimos años son muchos los programas de préstamo e intercambio de libros que se han puesto en marcha en los colegios e institutos de nuestro país. Los beneficios de este tipo de programas son altos, no solo desde la perspectiva económica sino también desde el punto de vista del cuidado al medio ambiente. Esta aplicación surge de la importancia de facilitar a las familias el intercambio de ejemplares de texto pues pretende ser una herramienta para la reutilización de los mismos.

A través de la aplicación, las familias podrán solicitar los ejemplares que requieran sus hijos y, al mismo tiempo, podrán realizar donaciones para que los libros del curso pasado sean reutilizados por otros alumnos al año siguiente.

De aceptar estas solicitudes de reserva y donación se encargará un usuario administrador, para el que también hemos diseñado un apartado que facilite organizar todo esto.

El presente trabajo cuenta además con una serie de Anexos que lo complementan:

- **Anexo I: Planificación del proyecto** – Documento donde se expondrá la planificación temporal del proyecto, así como una estimación de los costes y el esfuerzo que será necesario para la finalización de este.
- **Anexo II: Especificación de requisitos** – Documento donde se detallarán los requisitos que ha de cumplir nuestro sistema, así como una primera definición sin demasiado detalle de las distintas funciones de nuestro sistema.
- **Anexo III: Análisis del sistema** - Documento en el que se expondrá la arquitectura que tendrá nuestro sistema, y se irán detallando más en profundidad las distintas clases.
- **Anexo IV: Diseño del sistema** – Documento en el que se detallan las especificaciones del sistema, así como las herramientas utilizadas para su desarrollo.

- **Anexo V: Documentación técnica** – Guía donde se da una visión general de las clases que componen nuestro sistema enfocadas a otro programador u otra persona que desee entender el código y la estructura de nuestra aplicación.
- **Anexo VI: Manual de usuario** – Guía para que el usuario pueda utilizar la aplicación fácilmente. Este documento contiene una explicación precisa de la funcionalidad de la aplicación.

2. Objetivos

Este apartado muestra los objetivos detallados en la propuesta de proyecto que debe cumplir nuestro sistema además de los objetivos personales que quiero cumplir con el desarrollo de esta aplicación. Se detallarán tres tipos de objetivos, de tipo software, técnicos y personales.

2.1. Objetivos Software

El objetivo principal al realizar el proyecto es el desarrollo y la implementación de una aplicación que funcione en dispositivos de tipo Android y que pueda ser gestionada por los administradores de un colegio y sus alumnos para gestionar el préstamo de libros. Para lograrlo, el proyecto debe cumplir los siguientes objetivos:

- **Gestión de usuarios:** Nuestro sistema debe permitir el registro en el mismo de nuevos usuarios, así como gestionar los datos que para ello utilicen o los que quieran añadir para completar la creación de su cuenta. También tiene que encargarse del correcto funcionamiento en caso de querer modificarlos o querer eliminarlos. Además, controlará el acceso a los distintos apartados dependiendo del rol que tenga el usuario.
- **Gestión de libros:** El sistema que realicemos debe mantener un registro de todos los libros que sean añadidos a la base de datos, ya sean por los administradores o mediante donaciones de los usuarios. También gestionará correctamente todos los datos de estos, pues el resto de los objetivos están altamente relacionados con este y depende de él y sus datos, por lo que es fundamental que funcione correctamente para no propagar errores.
- **Gestión de imágenes:** El sistema deberá proporcionar un método por el que se puedan introducir y almacenar imágenes que correspondan con las portadas de los libros.
- **Gestión de préstamos:** Es la funcionalidad principal de la aplicación, por lo que nuestro sistema debe facilitar la gestión de los libros que estén disponibles y ofrecerlos para prestarlos. Además, deberá poder gestionar todos esos préstamos y almacenar los datos de los libros y los usuarios con los que estén relacionados cada uno de ellos.

- Escaneo de códigos: Este objetivo busca que podamos localizar la información de un libro o un usuario escaneando un código de barras o un código QR.
- Gestión de peticiones: El sistema deberá gestionar toda la información que genera cada petición o reserva de un libro por parte de un usuario y controlar que no se pueda reservar un mismo libro por dos usuarios distintos.
- Gestión de donaciones: El sistema registrará las donaciones de libros que realicen los distintos usuarios y almacenará sus datos una vez hayan sido aprobadas por un administrador.

2.2 Objetivos técnicos

Los principales objetivos técnicos que se buscan al desarrollar la aplicación son los siguientes:

- La aplicación debe ser sencilla y fácil de usar.
- Tendrá una base de datos común, aunque con apartados privados que pertenezcan a cada usuario, para que todos los usuarios puedan visualizar los libros disponibles, pero no puedan ver los datos privados de los otros usuarios.
- La aplicación ha de funcionar en dispositivos de tipo Android.
- Se pretende que la aplicación sea escalable y que los tiempos de acceso sigan siendo cortos a pesar de que aumente el número de usuarios o el tamaño de la base de datos.

2.3 Objetivos personales

El primer objetivo que voy a exponer en este apartado es la finalización del grado de Ingeniería Informática, lo que conllevará la obtención del título y por tanto me permitirá trabajar en el sector de la informática disponiendo de un título.

El siguiente objetivo es poner en práctica el conocimiento que se ha ido adquiriendo a lo largo de la carrera, pues, aunque haya sido evaluado, no siempre hay una forma correcta para comprobar lo aprendido y el hecho de realizar un trabajo de tal tamaño y finalizarlo correctamente implica que sí que se ha aprendido.

Por otro lado, también tengo el objetivo de lograr diseñar y desarrollar una aplicación Android, pues como usuaria habitual de un dispositivo de este tipo es un reto para mí el

poder desarrollar algo que pueda funcionar en él. Además, no es algo que se haya presentado a lo largo de la carrera por lo que implicará la búsqueda de información y manuales para que funcione de manera correcta.

También me interesa adquirir conocimientos de una base de datos NoSQL como son las que proporciona la plataforma Firebase. Hasta ahora, todo lo que sé sobre bases de datos es del tipo SQL por lo que me gustaría aprender las diferencias entre ambas y así comprobar cuál de ellas será más adecuada en futuros proyectos.

3. Técnicas y herramientas

3.1. Lenguajes utilizados

3.1.1 Java

Es el lenguaje utilizado para los archivos de código fuente que se encargarán del funcionamiento de nuestra aplicación. Se ha elegido este lenguaje porque es de el que más conocimiento tengo, además de que está presente en la mayoría de los dispositivos que utilizamos a diario.

Java es un lenguaje cuya sintaxis deriva de C y C++, y cuyas clases son compiladas a código byte por lo que se puede ejecutar siempre que se disponga de una máquina virtual Java (JVM), sin que se tenga en cuenta la arquitectura del dispositivo.

Java se basa en la programación orientada a objetos (POO) y su estructura se basa en paquetes que almacenan las distintas clases utilizadas. Las clases están formadas por métodos, variables y constantes principalmente.

Además, gracias a que está tan ampliamente extendido dispone de buenos manuales y ejemplos para muchas funcionalidades que utilizamos, así como ejemplos de cómo resolver los más típicos errores que nos puedan ir surgiendo.

3.1.2 XML

Es un metalenguaje de marcado que se utiliza para representar la estructura de los documentos del sistema y mediante él visualizar las interfaces que se presentan al usuario. Tiene gran importancia debido a que separa la información mostrada de la forma de mostrarla, por lo que permite mostrar datos dinámicos sin ser demasiado difícil.

Este lenguaje nos permite definir nuestras propias etiquetas para una mejor organización.

3.2 Herramientas utilizadas

3.2.1 Android Studio

Como se ha dicho anteriormente, el entorno de desarrollo integrado (IDE) utilizado para el desarrollo de esta aplicación ha sido Android Studio, pues el entorno oficial para

desarrollar aplicaciones Android y dispone de una amplia documentación para el manejo de algunas funcionalidades.

Además, dispone de un emulador bastante completo y fácil de poner en marcha que, sobre todo al comienzo del proyecto permite ir probando el funcionamiento de todo lo desarrollado, y la detección temprana de posibles errores y su posible causa, pues dispone de un depurador que expone por consola un código de error y a que está debido en caso de que nuestra app falle.

También hay que destacar que está enfocado de aplicaciones en dos lenguajes, Kotlin y Java, que es el que nosotros hemos elegido para nuestra aplicación.

Otro servicio que nos ofrece Android Studio y que es de gran ayuda a la hora de desarrollar una aplicación es la integración con Github, que nos permite llevar un control de versiones y almacenar en la nube nuestro código y las versiones anteriores a la actual.

Por último, resaltar que también es bastante sencillo conectar con Firebase, pues tiene un asistente que te va guiando paso a paso para que funcione correctamente.

3.2.2 Firebase

Firebase es una plataforma enfocada al desarrollo de aplicaciones que pertenece a Google y que nos ofrece distintas herramientas que mejoren y faciliten la funcionalidad de nuestra app de manera gratuita.

En nuestro caso hemos utilizado los siguientes servicios.

3.2.2.1 Cloud Firestore

Nos ofrece una base de datos de tipo NoSQL en la que almacenaremos en forma de colecciones los datos que así deseemos. Estas colecciones a su vez están definidas en documentos con un identificador único. Los datos son sincronizados en tiempo real, por lo que existirá una consistencia entre todos los datos que se le presenten a los usuarios. Firebase permitirá acceder al administrador del sistema mediante su consola, por lo que facilitará la gestión de los datos almacenados.

3.2.2.2 Firebase Authentication

Nos ofrece gestionar todo lo relacionado con la identidad de los usuarios del sistema, así como la gestión de los mismos en el registro a la aplicación. Ofrece distintos métodos para registrarse en el sistema, pero en nuestra aplicación solo está permitido el registro mediante correo y contraseña, que se verificará antes de poder entrar al sistema.

3.2.2.3 Cloud Storage

Este servicio fue creado para el almacenamiento del contenido generado por los usuarios mientras se utiliza las aplicaciones, como pueden ser videos o imágenes y que así no ocupen almacenamiento en el dispositivo. En nuestra aplicación, hacemos uso de este sistema a la hora de recibir las fotos que corresponden a las portadas de los libros, y que una vez seleccionadas por el usuario, se almacenaran en el Cloud Storage, guardando en el documento que corresponda del libro, una URL que dirija a cada imagen.

Además de estos servicios, hemos activado el servicio de Google Analytics, para que el administrador del sistema pueda visualizar mediante la consola de Firebase algunas analíticas relacionadas con el uso del sistema y que parte es la que recibe más visitas.

3.2.3 Librerías

También hemos utilizado algunas librerías externas que nos han permitido añadir distintas funcionalidades al sistema. Estas librerías hay que añadir las en el archivo “build.gradle” y sincronizar para que puedan ser usadas en todo el proyecto.

En este archivo también es donde añadimos las librerías para poder hacer uso de los servicios que nos ofrece Firebase como vemos en la Ilustración 1.

```
implementation platform('com.google.firebase:firebase-bom:28.3.0')
implementation 'com.google.firebase:firebase-analytics'
implementation 'com.google.firebase:firebase-messaging'
implementation 'com.google.firebase:firebase-auth'
implementation 'com.google.firebase:firebase-firestore'
implementation 'com.google.android.gms:play-services-auth:19.2.0'
implementation 'com.firebaseui:firebase-ui-storage:7.2.0'
```

Ilustración 1: Librerías Firebase

La siguiente librería de la que hacemos uso se llama Awesome Validation, y el sistema la utiliza para comprobar que los datos del correo y la contraseña que un usuario ha introducido para registrarse sean válidos.

```
implementation 'com.basgeekball:awesome-validation:4.3'
```

Ilustración 2: Librería Awesome Validation

Por último, la librería que aparece en la Ilustración 3 es la usada para poder implementar el lector de códigos y su correspondiente tratamiento.

```
implementation 'com.journeyapps:zxing-android-embedded:3.4.0'
```

Ilustración 3: Librería escaneo de códigos

3.3 Herramientas CASE

Las herramientas CASE (Ingeniería de Software Asistida por Computadora), son la que son utilizadas en el diseño de un sistema software para reducir su coste en tiempo de diseño, pues gracias a ellas se van desarrollando los distintos pasos que se han de seguir en las fases del ciclo de vida de desarrollo del software.

Si se siguen estos pasos a la hora de llevar a cabo el sistema se mejora la calidad del código, la planificación del proyecto o se aumenta la productividad del desarrollador entre otras cosas.

3.3.1 Visual Paradigm

Es la herramienta CASE con la que se realizan los diagramas UML, es decir el lenguaje unificado de modelado, que es el lenguaje más extendido para modelar los sistemas de software.

Con esta herramienta se realizan los distintos diagramas que componen tanto el anexo II como el anexo III. Algunos de los tipos de diagramas realizados en estos apartados son diagramas de secuencia, diagramas de casos de uso o diagramas de comunicación como vemos en la ilustración 4.

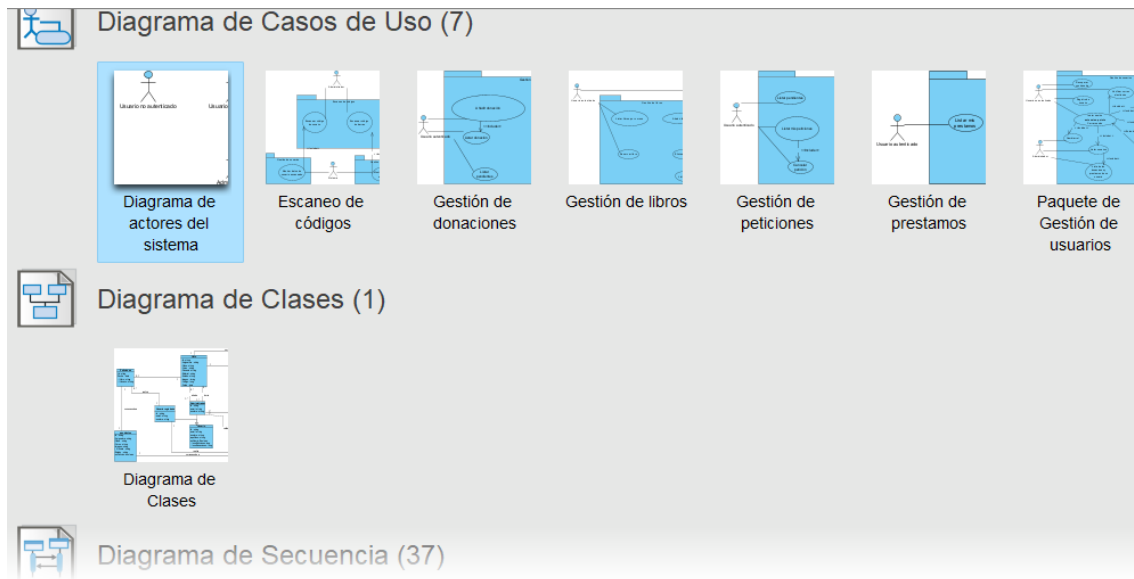


Ilustración 4: Diagramas realizados con Visual Paradigm

3.3.2 REM

La herramienta REM es una herramienta desarrollada en la Universidad de Sevilla por Amador Durán para la realización de la elicitación de requisitos según el método de Durán y Bernárdez. Gracias a esta herramienta conseguimos las tablas que componen el Anexo II, rellenando los campos que te van indicando según el tipo de tabla que sea. Las imágenes 5 y 6 corresponde a una captura de dicha herramienta.

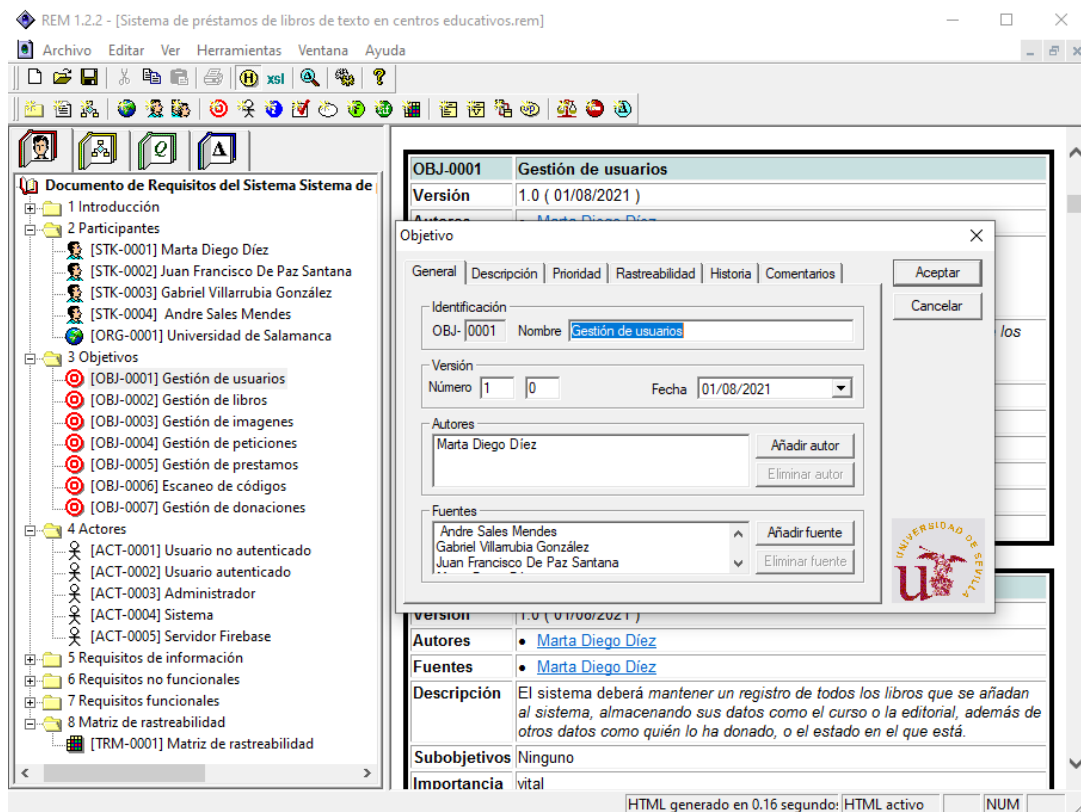


Ilustración 5: Herramienta REM

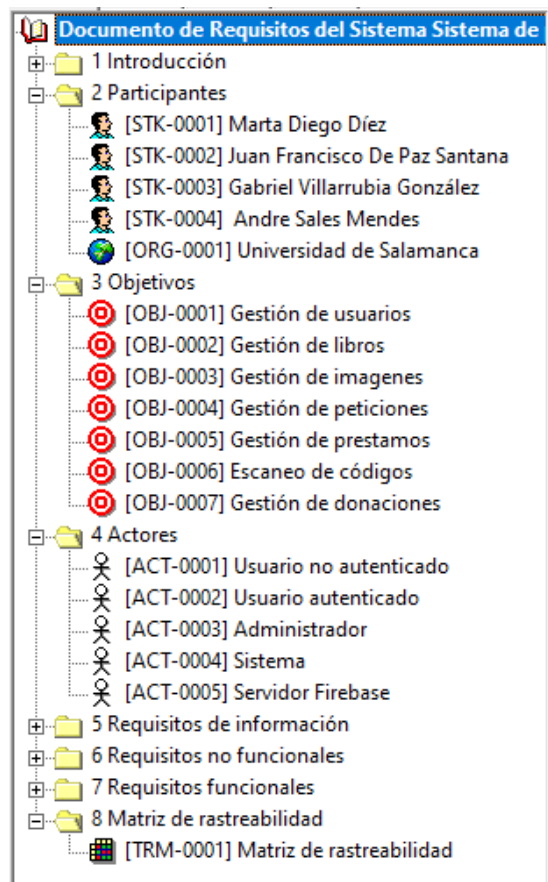


Ilustración 6: Índice del proyecto en REM

Ilustración

3.3.3 EZEstimate

Es una herramienta con la que realizamos la estimación de costes de tiempo y esfuerzo que se necesitará para desarrollar el sistema. Para hacer este cálculo utilizamos la métrica de Puntos de Casos de Uso como se explica más ampliamente en el Anexo I. Para realizar este cálculo hay que darles unos valores a los casos de uso, actores, factores técnicos y factores del entorno. La herramienta los multiplica por los pesos correspondientes, obteniendo así el número de horas que serían necesarias para finalizar el desarrollo de la aplicación.

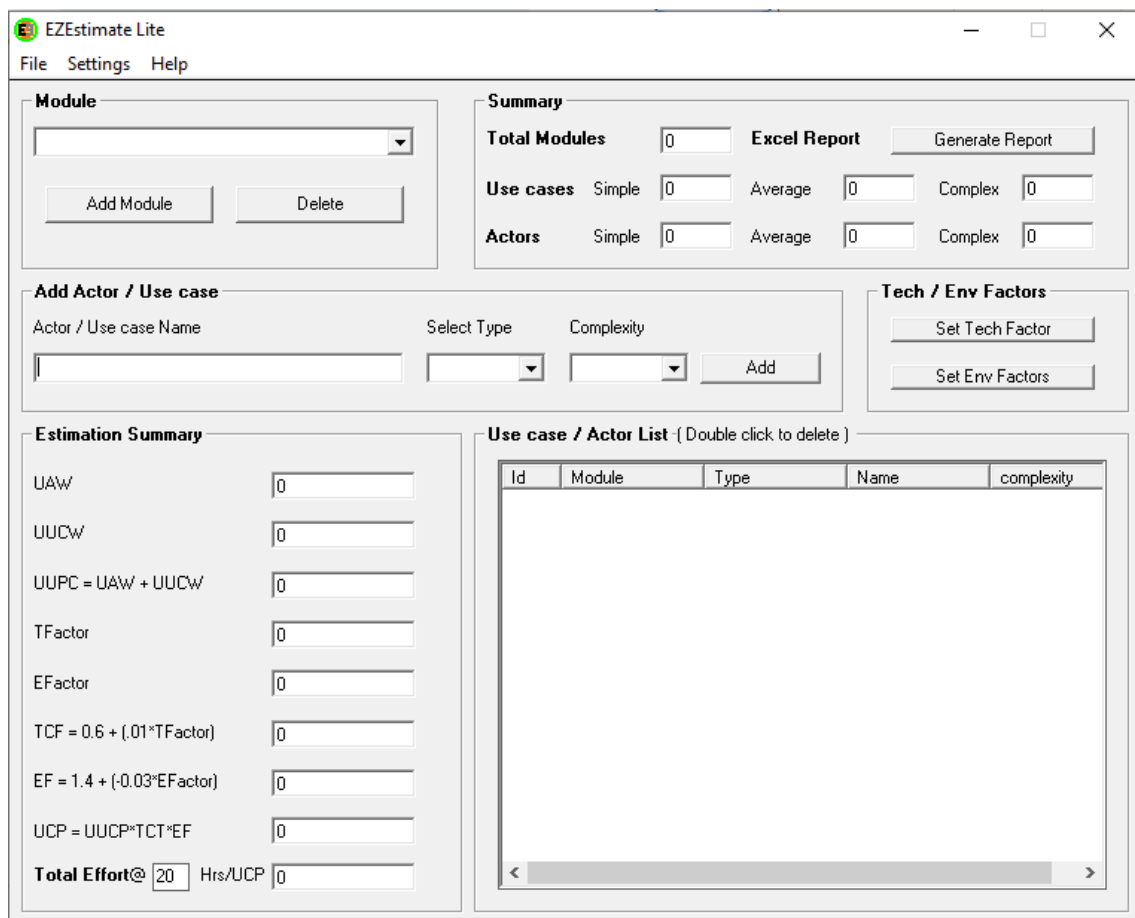


Ilustración 7: Herramienta EZEstimate

3.3.4 MS Project

Microsoft Project es un software que se usa para la administración de proyectos que es comercializado por Microsoft. Este programa, ayuda a los administradores de proyectos a organizar los mismos, planificar las tareas y asignar unos recursos a cada tarea para la correcta realización.

En nuestro caso lo hemos utilizado para realizar un diagrama de Gantt. Todo esto aparecerá más detallado en el Anexo I.

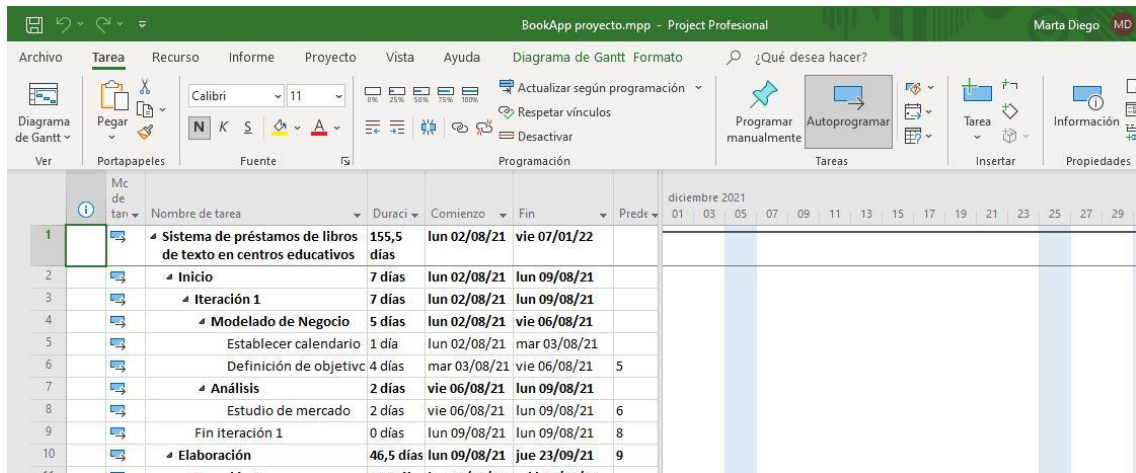


Ilustración 8: Tareas plan de proyecto

4. Aspectos relevantes del desarrollo

En este apartado se especifican las partes más importantes de las distintas fases de las que se compone el desarrollo del sistema.

4.1 Marco de trabajo

Se ha procurado utilizar el Proceso Unificado como metodología de desarrollo pues define un marco de trabajo cuyo fin es el desarrollo de un producto de manera iterativa e incremental. Las principales características del Proceso Unificado son las siguientes:

- Proceso conducido por caso de uso, que son descripciones de las acciones que se tienen que llevar a cabo para que se realice una función, y las respuestas que genera en el sistema.

- Está centrado en la arquitectura, que muestra una visión general del sistema completo y su estructura.

- Es iterativo e incremental, es decir, se divide el trabajo en tareas más pequeñas que van generando resultados y se van ampliando tras cada iteración.

Está formado por una serie de ciclos que se repiten hasta que se completa el desarrollo del sistema. Cada ciclo consta de cuatro fases:

1. Inicio: donde se define el alcance de negocio y se definen los casos de usos.
2. Elaboración: se especifican con más detalle los casos de uso y se diseña la arquitectura.
3. Construcción: en esta fase se realiza el producto.
4. Transición: se prueba el producto, se corrigen algunos errores y se incorporan nuevos requisitos para el siguiente ciclo.

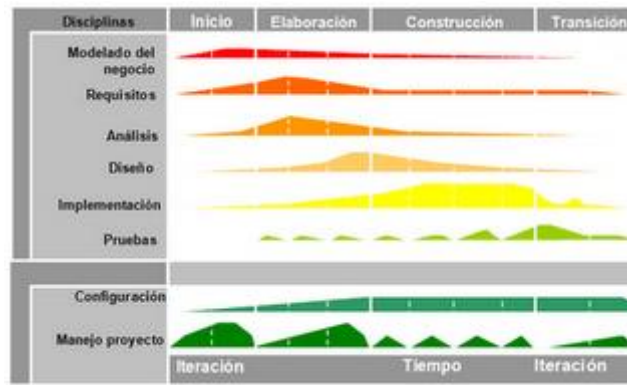


Ilustración 9: Fases del Proceso Unificado

Como nuestro sistema es una aplicación no demasiado grande, no se requieren demasiadas iteraciones para acabarlo como vemos en Ilustración 10. En el anexo I podemos ver la división que hemos hecho y los plazos de tiempo que hemos propuesto para llevarlo a cabo.

ID	Descripción	Duración	Inicio	Fin	Dependencia
1	↔ Sistema de préstamos de libros de texto en centros educativos	155,5 días	lun 02/08/21	vie 07/01/22	
2	↔ Inicio	7 días	lun 02/08/21	lun 09/08/21	
3	↔ Iteración 1	7 días	lun 02/08/21	lun 09/08/21	
4	↳ Modelado de Negocio	5 días	lun 02/08/21	vie 06/08/21	
7	↳ Análisis	2 días	vie 06/08/21	lun 09/08/21	
9	↳ Fin iteración 1	0 días	lun 09/08/21	lun 09/08/21	8
10	↔ Elaboración	46,5 días	lun 09/08/21	jue 23/09/21	9
11	↔ Iteración 2	25,5 días	lun 09/08/21	mié 01/09/21	
12	↳ Modelado de Negocio	7,5 días	lun 09/08/21	lun 16/08/21	
16	↳ Análisis	12 días	lun 16/08/21	jue 26/08/21	
20	↳ Diseño	6 días	jue 26/08/21	mié 01/09/21	
22	↳ Fin iteración 2	0 días	mié 01/09/21	mié 01/09/21	21
23	↔ Iteración 3	21 días	mié 01/09/21	jue 23/09/21	22
24	↳ Modelado de Negocio	6,5 días	mié 01/09/21	jue 09/09/21	
27	↳ Análisis	6 días	jue 09/09/21	mié 15/09/21	
30	↳ Diseño	5,5 días	mié 15/09/21	lun 20/09/21	
33	↳ Implementación	3 días	lun 20/09/21	jue 23/09/21	
35	↳ Fin iteración 3	0 días	jue 23/09/21	jue 23/09/21	34
36	↔ Construcción	72 días	jue 23/09/21	vie 03/12/21	35
37	↔ Iteración 4	33 días	jue 23/09/21	mar 26/10/21	
38	↳ Modelado de Negocio	0,5 días	jue 23/09/21	jue 23/09/21	
40	↳ Análisis	1,5 días	jue 23/09/21	vie 24/09/21	
42	↳ Diseño	4 días	vie 24/09/21	mié 29/09/21	
44	↳ Implementación	15 días	mié 29/09/21	jue 14/10/21	
47	↳ Pruebas	4 días	jue 14/10/21	lun 18/10/21	
49	↳ Documentación	8 días	lun 18/10/21	mar 26/10/21	
51	↳ Fin iteración 4	0 días	mar 26/10/21	mar 26/10/21	50
52	↔ Iteración 5	39 días	mar 26/10/21	vie 03/12/21	51
53	↳ Diseño	2 días	mar 26/10/21	mié 27/10/21	
55	↳ Implementación	22 días	mié 27/10/21	jue 18/11/21	
58	↳ Pruebas	5 días	jue 18/11/21	mié 24/11/21	
61	↳ Documentación	10 días	mié 24/11/21	vie 03/12/21	
63	↳ Fin iteración 5	0 días	vie 03/12/21	vie 03/12/21	62
64	↔ Transición	30 días	vie 03/12/21	vie 07/01/22	63
65	↔ Iteración 6	30 días	vie 03/12/21	vie 07/01/22	

Ilustración 10: Iteraciones de nuestro sistema

4.2 Estimación de costes

Esta tarea se realiza al inicio del proyecto, pues nos dará una idea de la cantidad de esfuerzo y el coste que conllevará la realización del sistema. Esta tarea es más adecuada para proyectos que dispongan de más desarrolladores, pues dará un resultado más acorde.

Estimation Summary	
UAW	8
UUCW	185
UUPC = UAW + UUCW	193
TFactor	32
EFactor	24
TCF = 0.6 + (.01*TFactor)	0,92
EF = 1.4 + (-0.03*EFactor)	0,68
UCP = UUCP*TCT*EF	120,7408
Total Effort@ <input type="text" value="10"/> Hrs/UCP	1207,408

Ilustración 11: Resumen EZEstimate

La Ilustración 11 muestra una estimación del tiempo que costaría realizar el sistema en base a los casos de uso y los actores que tenemos, y a los factores del entorno y a los factores técnicos que hemos añadido. Todo esto aparecerá más detallado en el Anexo I.

4.3 Especificación de requisitos.

En este subapartado vamos a centrarnos en la especificación de los requisitos software del sistema. Se ha seguido la metodología propuesta por Durán y Bernárdez para la realización de esta especificación. Además, se han utilizado tanto REM como Visual Paradigm.

Al existir un Anexo específico dedicado a la Especificación de requisitos software, en este documento tan solo se expondrá una tabla de cada uno de los ejemplos que servirá como ejemplo para el resto.

Así, primeramente, aparece una descripción de los objetivos que persigue el sistema como se ve en Tabla 1.

OBJ-0001	Gestión de usuarios
Versión	1.0 (01/08/2021)
Autores	<ul style="list-style-type: none"> • Marta Diego Díez
Fuentes	<ul style="list-style-type: none"> • André Sales Mendes • Gabriel Villarrubia González • Juan Francisco De Paz Santana • Marta Diego Díez
Descripción	El sistema deberá <i>permitir el registro de nuevos usuarios y la baja de los mismos, además de la modificación de sus datos una vez ha sido registrado. También gestionará la autenticación en el sistema.</i>
Subobjetivos	Ninguno
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

Tabla 1: OBJ-0001 Gestión de usuarios

Posteriormente pasamos a una concreción de los actores que intervienen en el sistema y que interactúan a través de la aplicación que podemos ver en Ilustración 12.

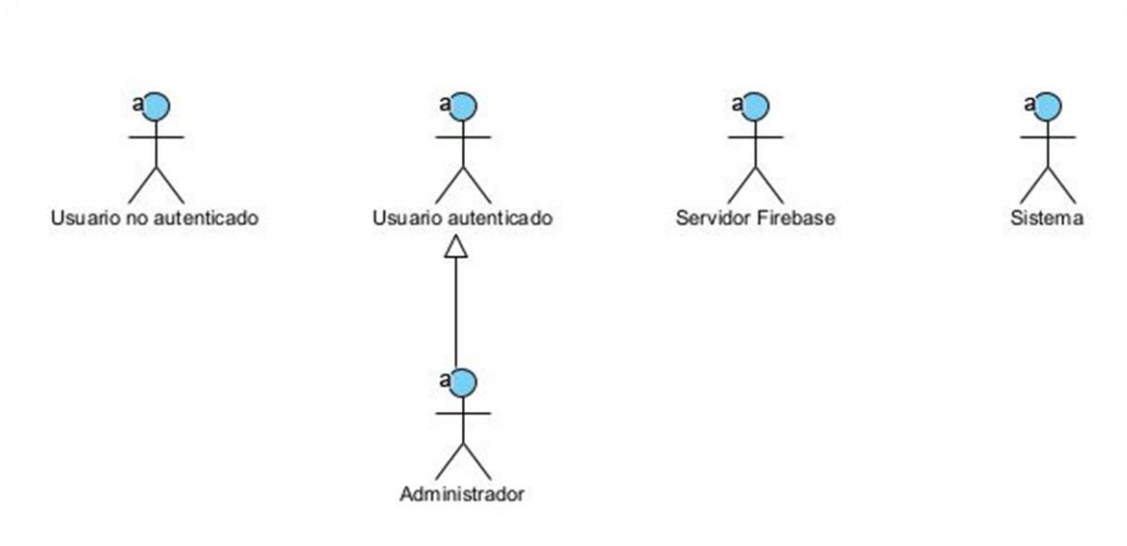


Ilustración 12: Actores del sistema

Una vez detallados los actores, aparecen ahora los requisitos de información y los requisitos no funcionales en la Tabla 2.

IRQ-0001	Datos de los usuarios registrados	
Versión	1.0 (01/08/2021)	
Autores	<ul style="list-style-type: none"> • Marta Diego Díez 	
Fuentes	<ul style="list-style-type: none"> • André Sales Mendes • Gabriel Villarrubia González • Juan Francisco De Paz Santana • Marta Diego Díez 	
Dependencias	Ninguno	
Descripción	El sistema deberá almacenar la información correspondiente a <i>los usuarios que se registren en el sistema</i> . En concreto:	
Datos específicos	<ul style="list-style-type: none"> • Identificador • Email • Nombre • Apellidos • Si es Administrador (<i>true o false</i>) • Número de donaciones • Número de prestamos • Imagen de perfil 	
Tiempo de vida	Medio	Máximo

Ocurrencias simultáneas	Medio	Máximo
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 2: IRQ-0001 Datos de los usuarios registrados

Aparecen ahora los requisitos funcionales, requisitos que describen las actividades que va a realizar el sistema.

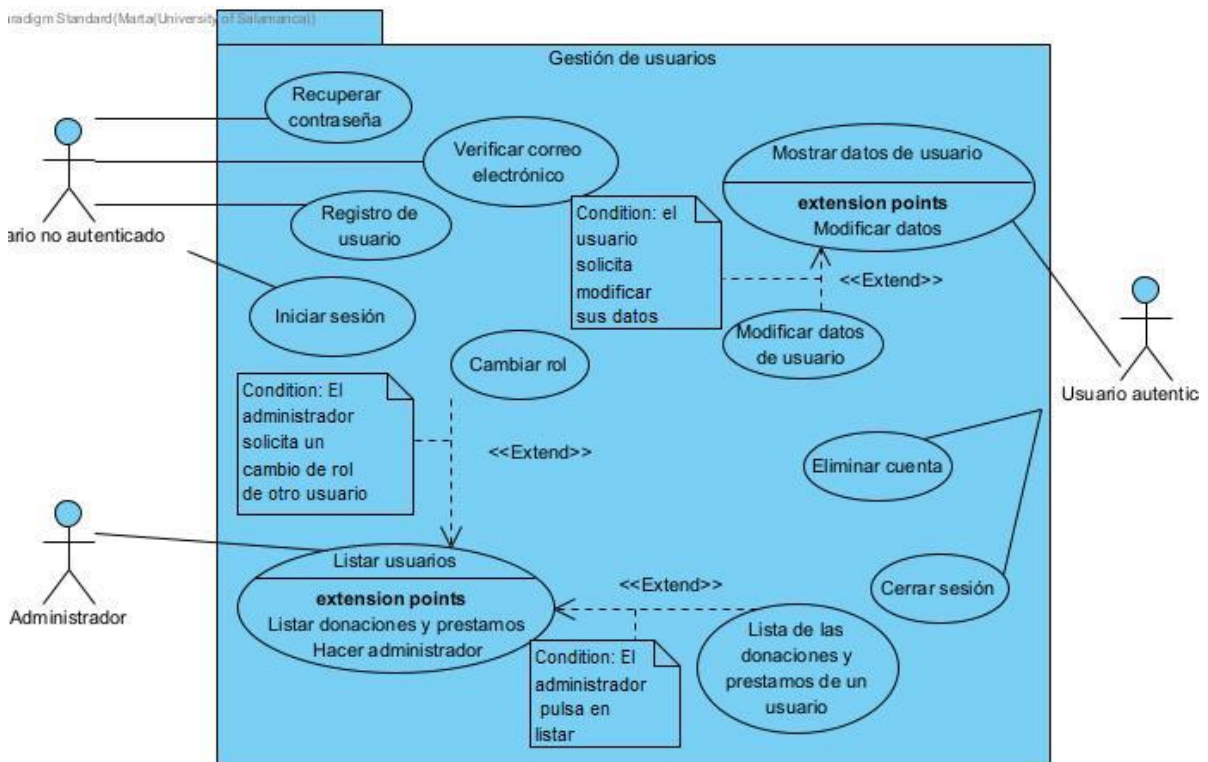


Ilustración 13: Diagrama de casos de uso del paquete de gestión de usuarios

UC-0001	Registro de usuario
Versión	1.0 (02/08/2021)
Autores	<ul style="list-style-type: none"> • Marta Diego Díez
Fuentes	<ul style="list-style-type: none"> • André Sales Mendes

	<ul style="list-style-type: none"> • Gabriel Villarrubia González • Juan Francisco De Paz Santana • Marta Diego Díez 	
Dependencias	<ul style="list-style-type: none"> • [NFR-0001] Usabilidad • [NFR-0002] Eficiencia • [OBJ-0001] Gestión de usuarios • [NFR-0003] Escalabilidad • [NFR-0005] Almacenamiento Firebase • [NFR-0004] Sistema Operativo Android 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>quiera crearse una cuenta para acceder a la aplicación.</i>	
Precondición	El usuario no tiene una cuenta de acceso a la aplicación.	
Secuencia normal	Paso	Acción
	1	El actor Usuario no autenticado (ACT-0001) <i>accede a la aplicación y decide crearse una cuenta por lo solicita registrarse.</i>
	2	El actor Sistema (ACT-0004) <i>presenta un formulario con los datos que el usuario debe rellenar para crearse una cuenta.</i>
	3	El actor Usuario no autenticado (ACT-0001) <i>rellena los campos requeridos para la creación de la cuenta solicita su registro en la aplicación.</i>
	4	El actor Sistema (ACT-0004) <i>comprueba los datos de correo y contraseña, y si son válidos los almacena.</i>
	5	El sistema <i>mostrará un mensaje de que se ha creado correctamente el usuario y envía un correo para que verifique los datos.</i>
Postcondición	El usuario queda registrado en el sistema.	
Excepciones	Paso	Acción
	3	Si <i>el correo que se ha introducido no es válido</i> , el actor Sistema (ACT-0004) <i>muestra un mensaje de error pidiendo que se introduzca un correo válido.</i> , a continuación este caso de uso <i>continúa</i>
	3	Si <i>la contraseña que ha introducido tiene menos de 6 dígitos</i> , el actor Sistema (ACT-0004) <i>muestra un error y pide que se introduzca una contraseña de más de 6 caracteres.</i> , a continuación este caso de uso <i>continúa</i>
	3	Si <i>el correo utilizado para registrarse ya dispone de una cuenta en el sistema</i> , el actor Sistema (ACT-0004) <i>muestra un mensaje de error</i>

		<i>diciendo que ya existe una cuenta registrada con ese correo y pidiendo que se utilice otra distinta, a continuación este caso de uso continúa</i>
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada		
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 3: Registro de usuario

Finalmente, hay que crear la matriz de rastreabilidad que mostrará la asociación entre los casos de uso y los diferentes objetivos planteados. A continuación, la Tabla 4 muestra un trozo de la matriz creada en el proyecto. Se puede encontrar la matriz completa en el Anexo II *Tabla 65: Matriz de rastreabilidad*.

TRM-0001	OBJ-0001	OBJ-0002	OBJ-0003	OBJ-0004	OBJ-0005	OBJ-0006	OBJ-0007
<u>UC-0001</u>	↑	-	-	-	-	-	-
<u>UC-0002</u>	↑	-	-	-	-	-	-
<u>UC-0003</u>	↑	-	-	-	-	-	-
<u>UC-0004</u>	↑	-	-	-	-	-	-
<u>UC-0005</u>	↑	-	-	-	-	-	-
<u>UC-0006</u>	↑	-	-	-	-	-	-
<u>UC-0007</u>	↑	-	-	-	-	-	-

Tabla 4: Matriz de Rastreabilidad

4.4 Análisis del sistema Software

Después de realizar la especificación de requisitos que hemos explicado en el anterior apartado, hay que realizar el análisis del sistema donde realizaremos una primera visión de la arquitectura del sistema. Todo esto está definido más ampliamente en el Anexo III.

4.4.1 Modelo de dominio

Lo primero es realizar el modelo del dominio, que es donde se representaran mediante conceptos del mundo real los datos más relevantes de nuestro sistema. Para ello utilizamos un diagrama de clases.

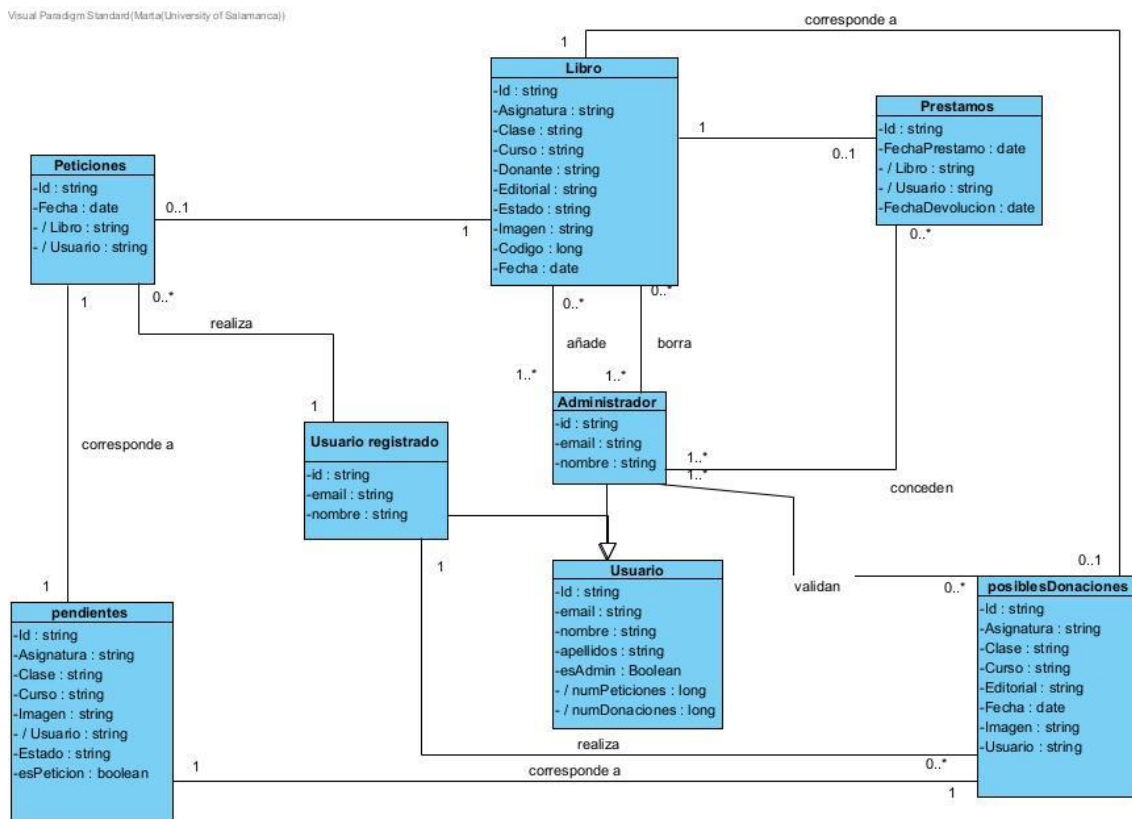


Ilustración 14: Diagrama de clases

4.4.2 Realización de casos de uso

En este apartado incluimos la realización de los diagramas de secuencia que corresponden a los distintos casos de uso por los que está compuesto el sistema. Todos estos diagramas están detallados en el Anexo III, y la Ilustración 15 es un ejemplo de ellos.

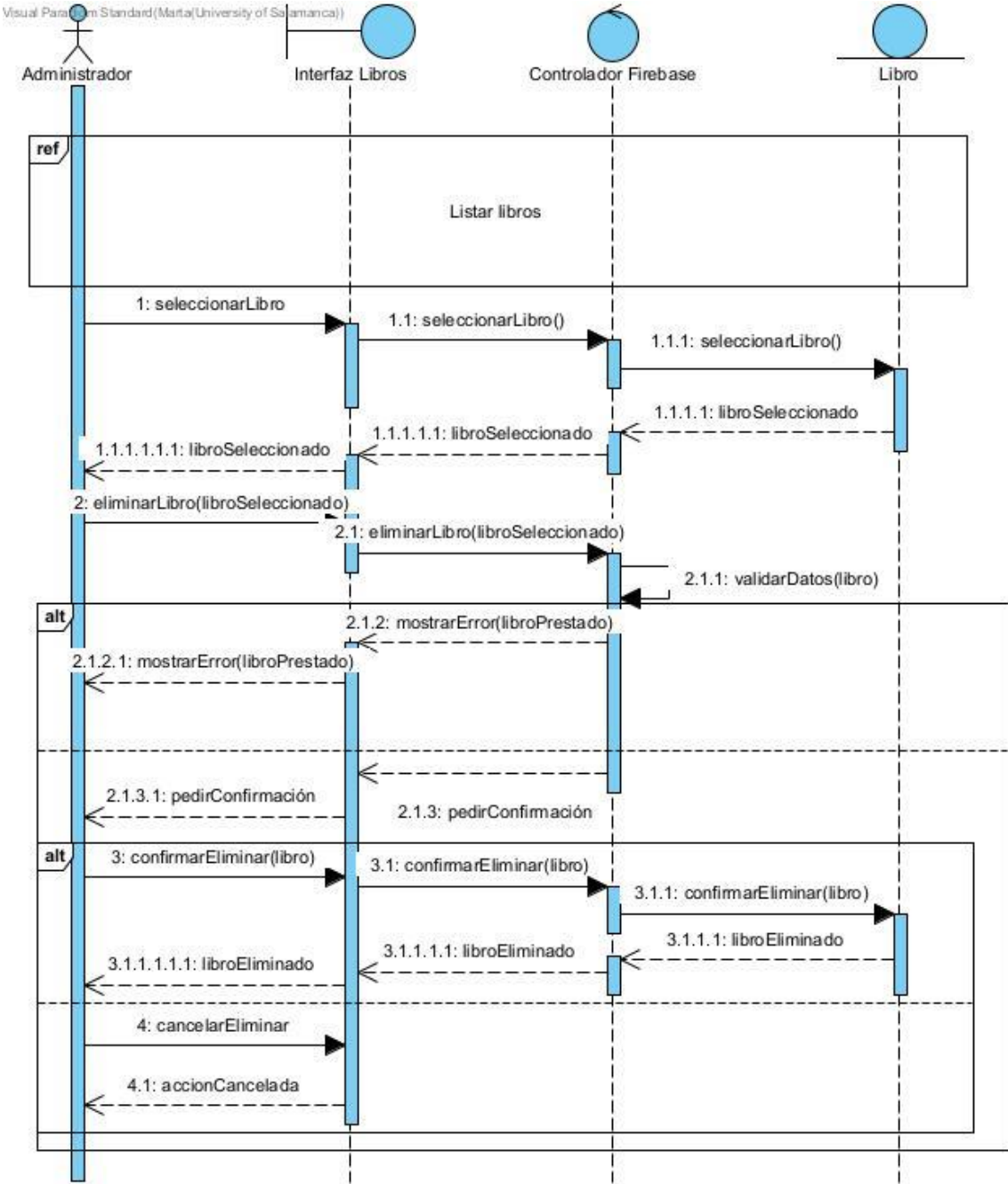


Ilustración 15: Diagrama de secuencia Eliminar libro.

Una vez realizados todos los diagramas de secuencia el siguiente paso es conseguir el primer esbozo que haremos de la arquitectura que tendrá nuestro sistema.

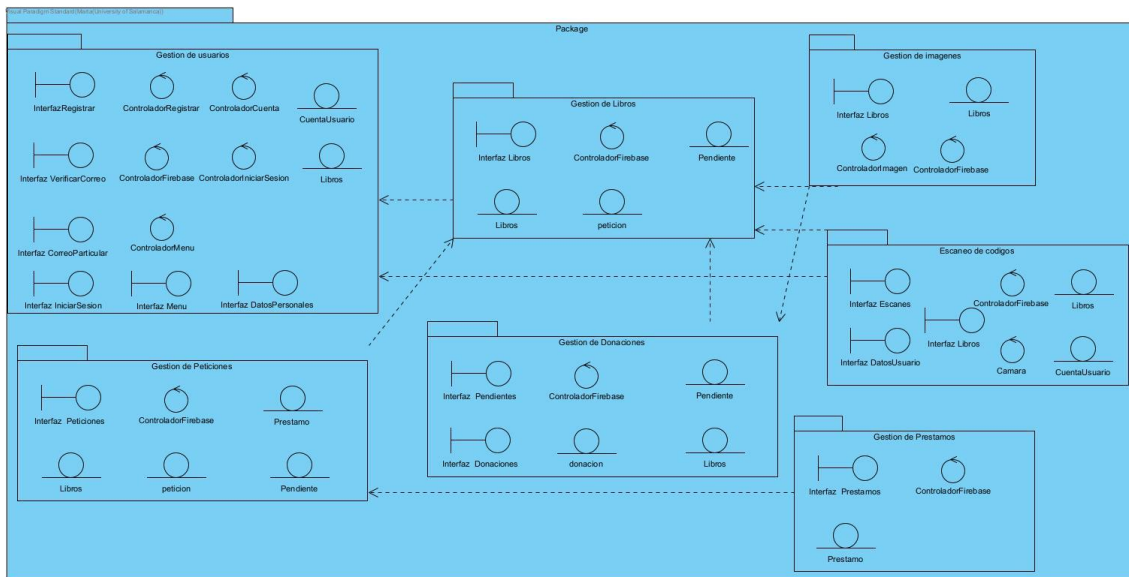


Ilustración 16:Arquitectura

4.5 Diseño del sistema

Una vez realizado tanto la especificación de requisitos como el análisis del sistema el siguiente paso es realizar lo que corresponde con la fase de diseño, donde buscaremos como realizar la implementación que nos permitirá obtener la solución final para conseguir elaborar nuestro sistema. Todo esto está explicado con más detalle en el anexo IV, donde podemos ver todos estos diagramas.

4.5.1 Patrón MVC

De acuerdo con el diseño que hemos hecho de nuestro sistema, el patrón que va más acorde y que por tanto ha sido el utilizado es el Modelo Vista Controlador (MVC). Este patrón separa en esos tres componentes los datos de la aplicación, la interfaz con la que interactúa el usuario y la lógica de control.

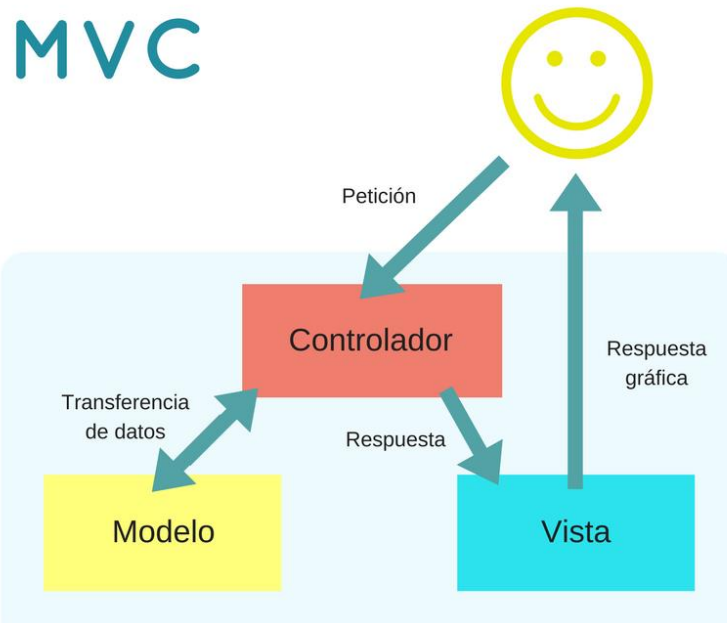


Ilustración 17: Patrón MVC

4.5.2 Modelo de diseño

En este apartado se elabora un diagrama de subsistemas que consiste en dividir el sistema en apartados más pequeños y manejables para que sea más sencillo y asequible su diseño y reutilización como podemos ver en Ilustración 18.

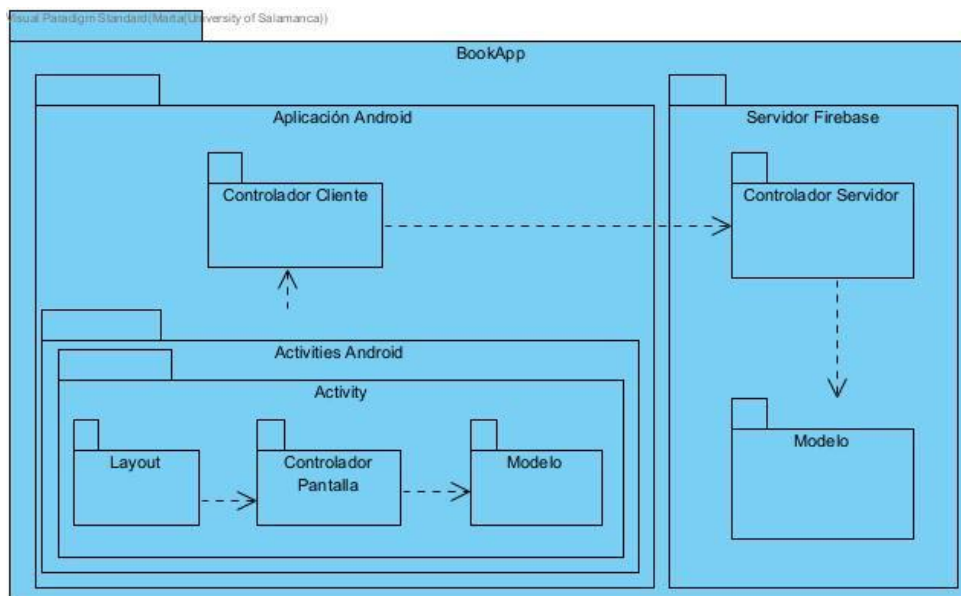


Ilustración 18: Diagrama de subsistemas

También se realiza un diagrama de capas siguiendo la arquitectura de nuestro patrón MVC, que se muestra en la Ilustración 19.

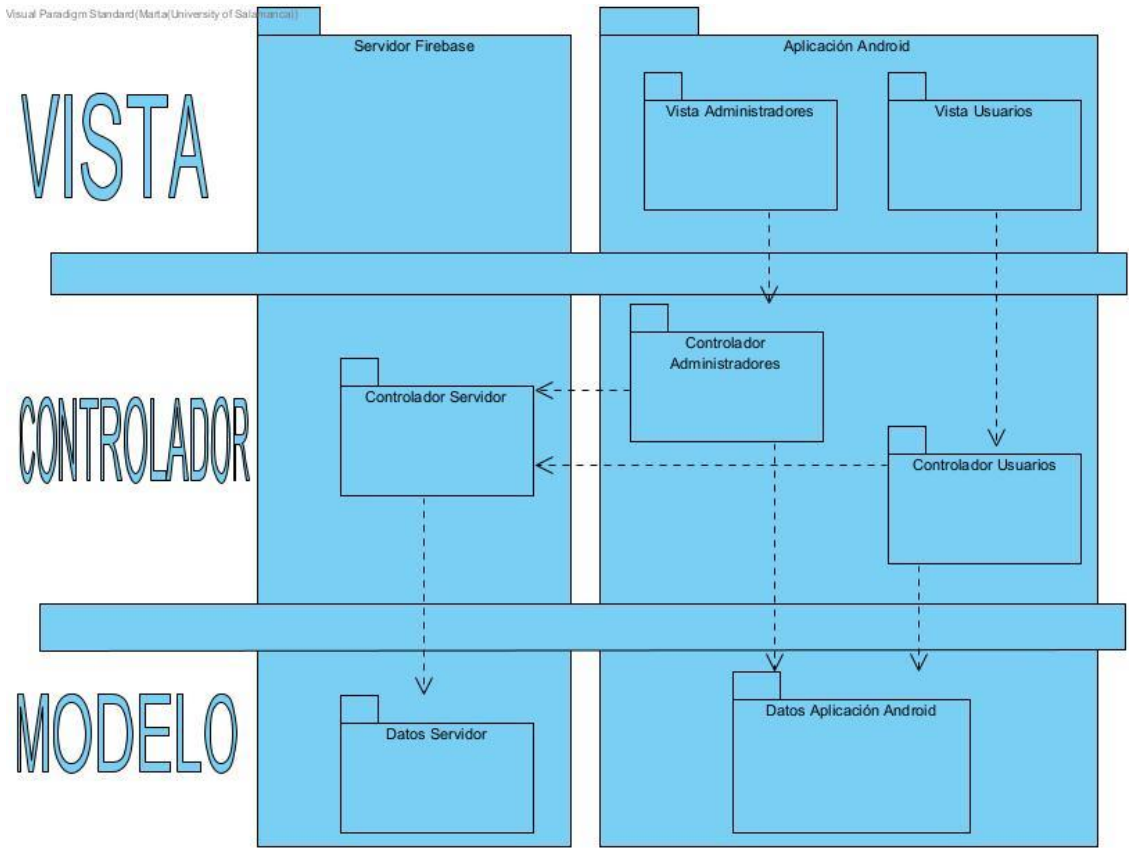


Ilustración 19: Vista arquitectónica de capas

En este apartado también se lleva a cabo la realización de los diagramas de casos de uso. Como ejemplo, el que se muestra en Ilustración 20.

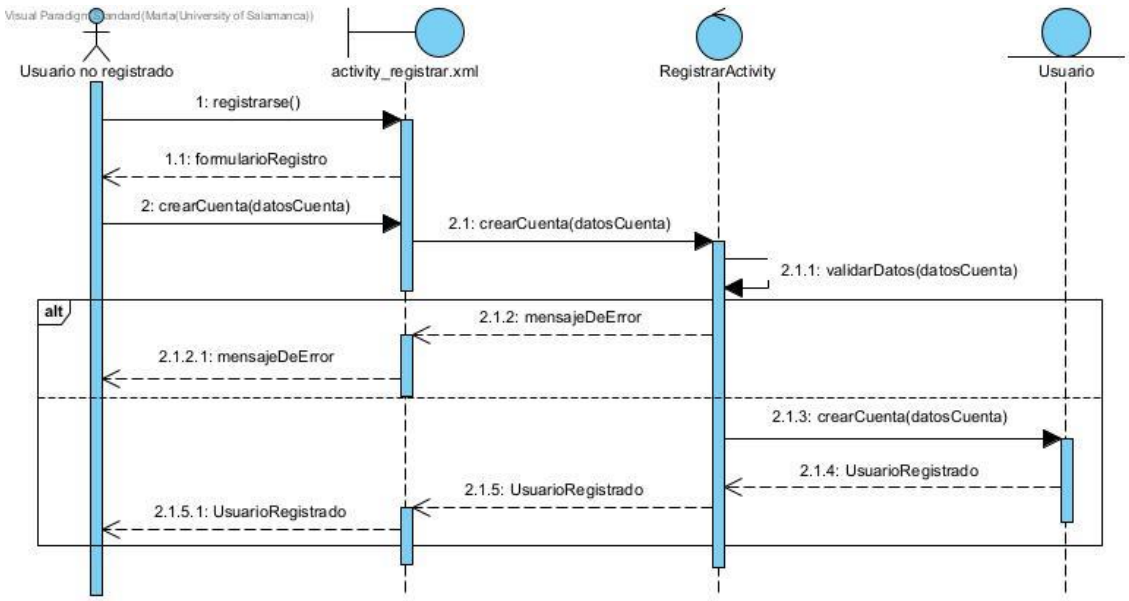


Ilustración 20: Diagrama de caso de uso registro de usuario

Por último se muestra un modelo de despliegue que muestra los nodos que componen el sistema y la estructura de los distintos artefactos que son los que vemos en la Ilustración 21 . Todo esto se puede ver con más detalle en el anexo IV.

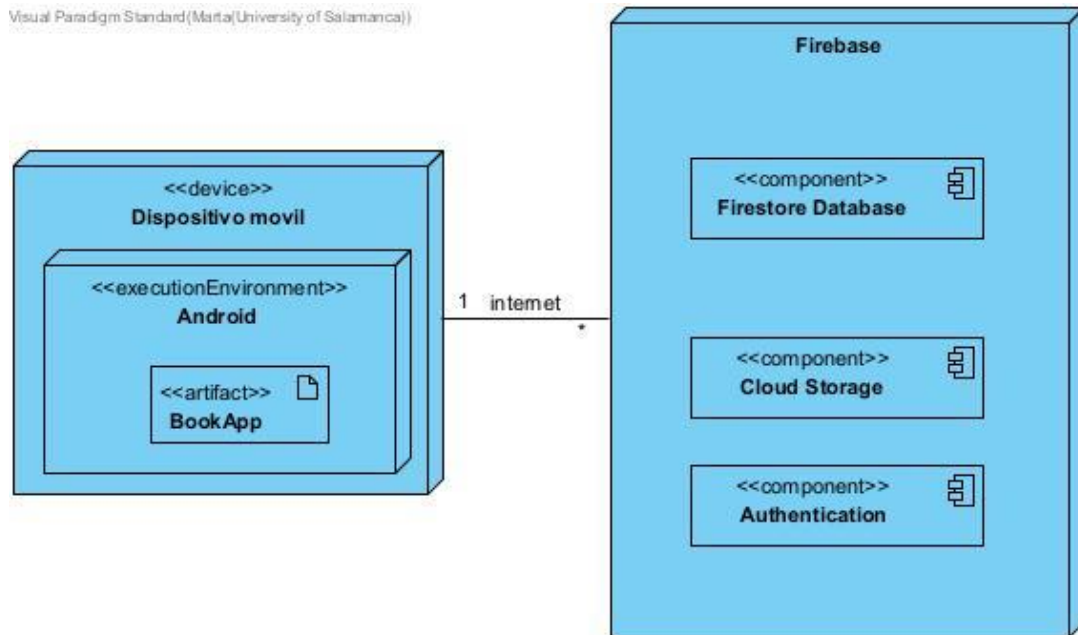


Ilustración 21: Diagrama de despliegue

4.5.2 Base de datos Firestore

Para almacenar nuestros datos, hacemos uso del servicio de Cloud Firestore de la plataforma Firebase, que nos permite guardar los datos en distintas colecciones a las que tienen acceso todos los usuarios de nuestro sistema, aunque hay datos que no pueden leer. El diagrama correspondiente a dicha base de datos es el que nos muestra la Ilustración 22.

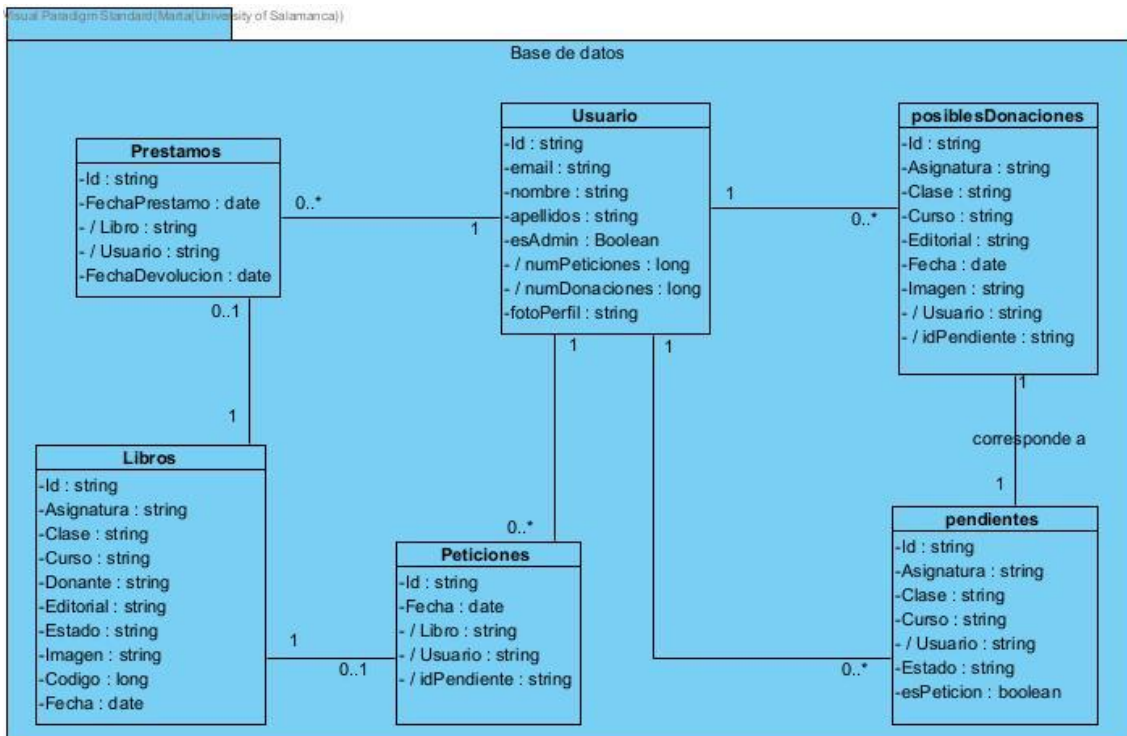


Ilustración 22: Diseño de la base de datos

La especificación de como almacenamos los datos y sus tipos se encuentra en el Anexo IV, pero en la Ilustración 23 muestro una captura de los mismos.

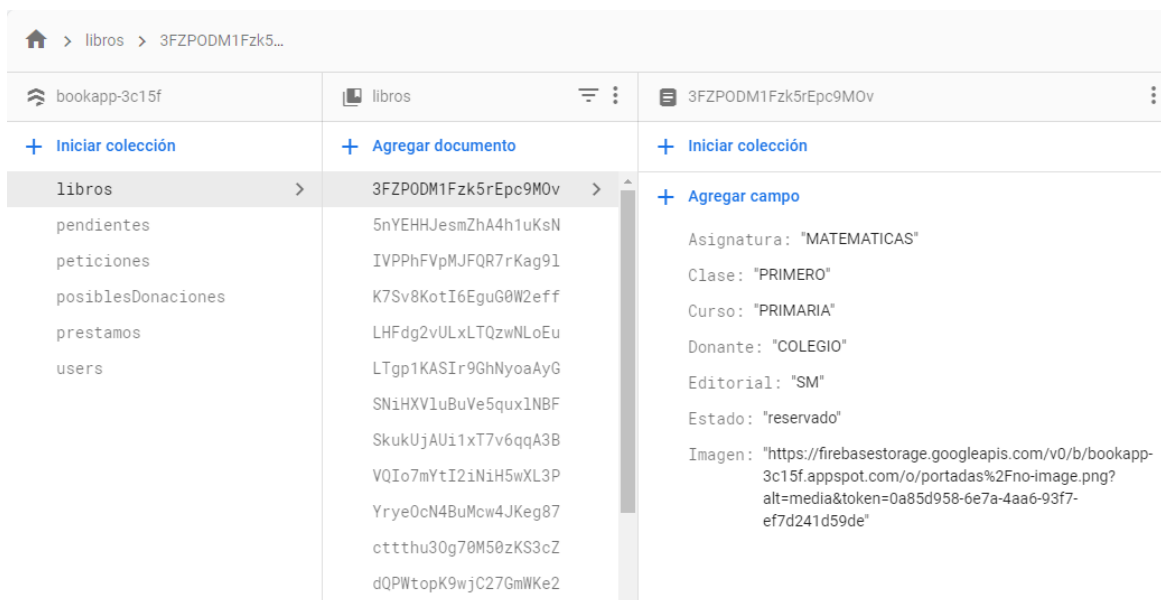


Ilustración 23: Colecciones de la base de datos

4.6 Implementación

Una vez hechos los apartados anteriores, el siguiente paso es pasar a la implementación del sistema, para lo que como hemos explicado anteriormente utilizamos el IDE de

Android Studio, utilizando como lenguaje para el código java, y XML para las distintas interfaces que contiene el sistema. Todo esto está explicado con detalle en el Anexo V. Las siguientes imágenes corresponden con la estructura general que tiene nuestra aplicación.

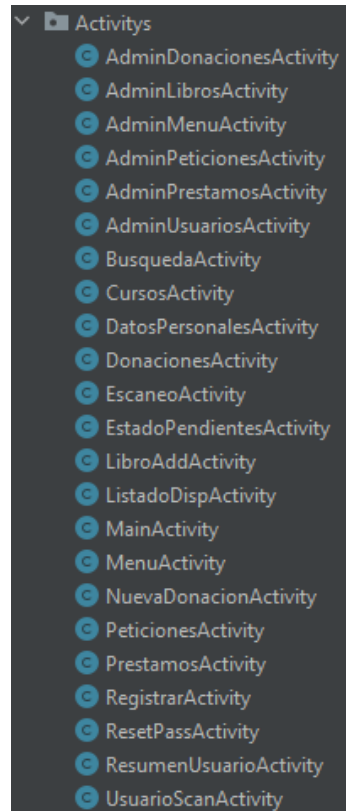


Ilustración 24: Paquete Activitys

Tanto en la Ilustración 244 como en la Ilustración 255 se ve la estructura del paquete de clases java, que corresponden al código que forma parte de la capa controlador y que es la que tiene la lógica de control del sistema.

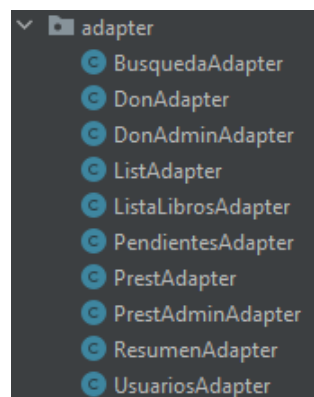


Ilustración 25: Paquete adapter

Este paquete contiene las clases que se utilizarán para la obtención de los datos desde la base de datos y que permitirá al resto de clases utilizarlo.

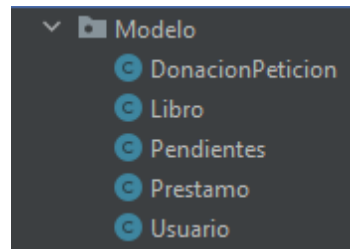


Ilustración 26: Paquete modelo

Por último, también hay que destacar el paquete layout, donde están los archivos XML que representan las diferentes interfaces de nuestro sistema.

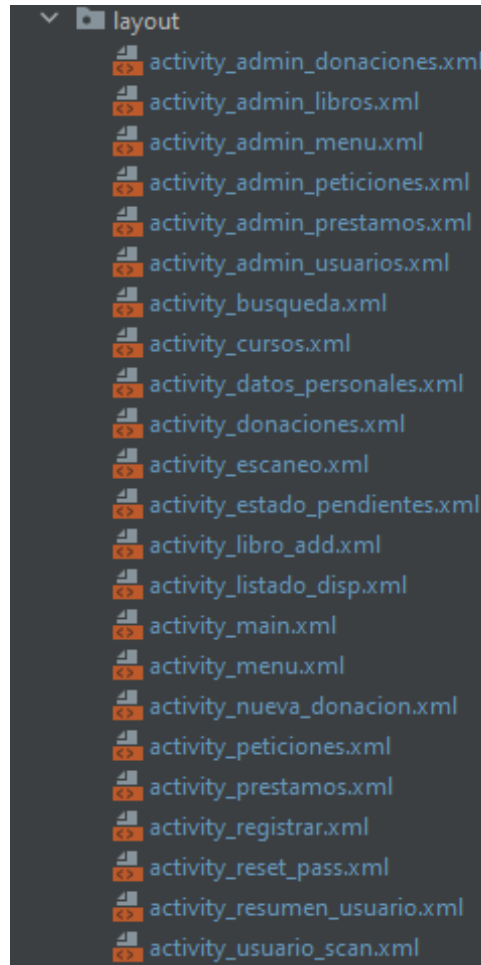


Ilustración 27: layouts de la aplicación

5. Pruebas del sistema

A lo largo de todo el proceso de desarrollo e implementación de la aplicación ha sido necesario ir haciendo constantes pruebas para comprobar el correcto funcionamiento de esta.

A medida que se iba avanzando en las sucesivas fases de creación del Proceso Unificado iban surgiendo distintos fallos a subsanar. Para encontrar dichos errores utilicé el depurador de Android Studio. Esta herramienta facilita notablemente el trabajo ya que, a

la vez que notifica el error encontrado, muestra una pequeña descripción del motivo sobre cuál es el error.

Una vez se consideró que la aplicación estaba terminada, se realizó una última prueba final. Para esta, se contó con la ayuda de terceras personas que hicieron uso de la aplicación solicitando préstamos de libros y proponiendo donaciones al sistema.

6. Resultado final de la aplicación

A pesar de que se adjunta una guía para la utilización del sistema por el usuario en el Anexo VI, en esta Memoria final explicaremos de manera breve el funcionamiento de la aplicación.

Una vez que el usuario tiene en su dispositivo móvil la aplicación descargada, deberá presionar el icono para acceder a ella. Al presionar sobre el mismo, el usuario entrará directamente a la pantalla inicial del sistema que vemos en la Ilustración 288.

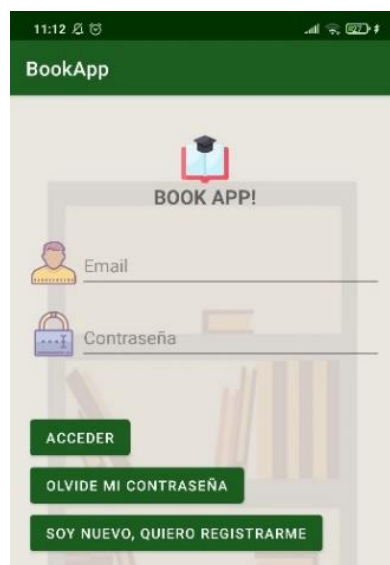


Ilustración 28: Pantalla de inicio

En el caso de que el usuario ya estuviera registrado, tan solo tendría que completar los campos vacíos de la pantalla de inicio y pulsar sobre acceder.

Si el usuario es nuevo, entonces deberá registrarse a través de la tercera opción que nos proporciona la pantalla de inicio y que vemos en la Ilustración 299. Al introducir correctamente sus datos en la pantalla siguiente, el usuario quedará correctamente registrado y ya podrá hacer uso de la aplicación una vez verifique su correo.

Ilustración 29: Registro de usuario

En el caso de que el usuario registrado hubiera perdido su contraseña, hará clic sobre la opción “Olvidé mi contraseña” de la pantalla de inicio y se le muestra la Ilustración 30.

Ilustración 30: Recuperar contraseña

Entonces se le redirigirá a una página de recuperación de contraseña donde tan solo con introducir el email de registro, el sistema le hará llegar un correo para la configuración de una nueva contraseña de acceso a BookApp.

El usuario entonces ya estará dentro de la aplicación y podrá hacer uso de la misma. A partir del menú de inicio que vemos en la Ilustración 3131 y tendrá la capacidad de hacer préstamos y donaciones de libros.



Ilustración 31: Menú principal

En la opción de “Libros disponibles” el usuario podrá consultar el listado de libros que puede solicitar para un préstamo. Tan solo tendrá que seleccionar el curso que desee y se le mostrarán los libros disponibles. Si selecciona uno y presiona la opción “Reservar”, se enviará la petición a los administradores como nos muestra la Ilustración 3232.



Ilustración 32: Reservar libro

De esta manera, habrá realizado correctamente la reserva que desea. Si quisiera deshacer la misma, en el menú inicial presionará “Mis peticiones” y le aparecerá la opción para ello que vemos en la Ilustración 33.



Ilustración 33: Mis peticiones

En el caso de que el usuario quisiera realizar una donación de un libro de texto que posee previamente, irá al menú inicial y presionará sobre la opción “Donaciones”. Ahí le aparecerán los libros que tiene y que, por tanto, puede donar al sistema como nos muestra la Ilustración 34.

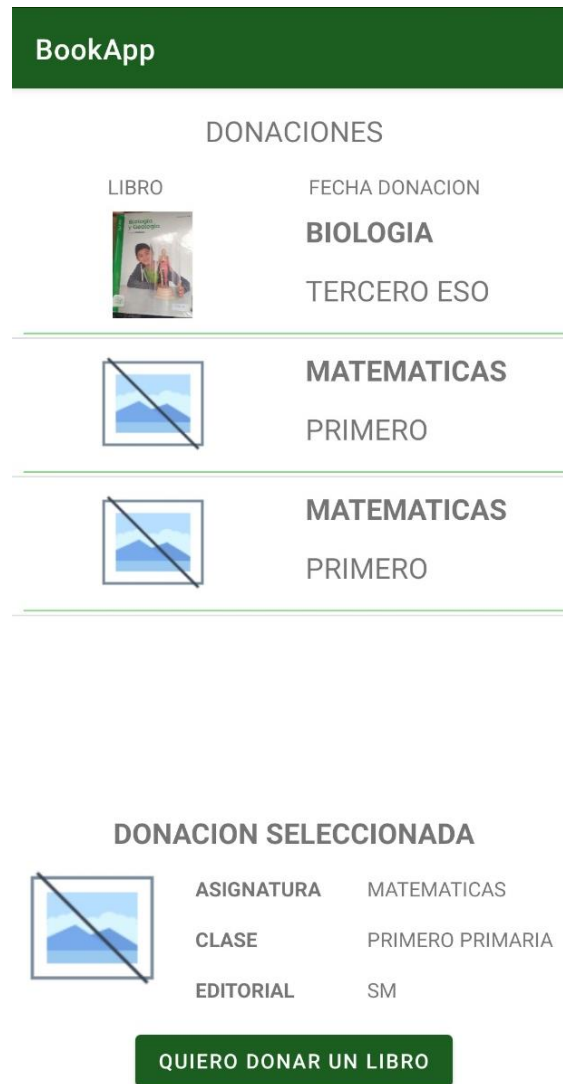


Ilustración 34: Donaciones

Tras pulsar sobre el botón que aparece en la parte inferior de la página, la aplicación le mostrará un formulario a rellenar con los datos del libro y le dará la opción de seleccionar una portada desde la galería o elegir la portada por defecto.

Para comprobar el estado tanto de las peticiones como de las donaciones propuestas, en el menú inicial el usuario accederá a la opción “Estado de las donaciones y peticiones pendientes” que muestra la Ilustración 35. Una vez el administrador le conceda el libro

requerido o le autorice a donar el ejemplar propuesto, el estado de la petición y de la donación aparecerá como “Aceptada”.

BookApp		
ESTADO DE LAS DONACIONES Y PETICIONES PENDIENTES		
TIPO	LIBRO	ESTADO
DONACION	EDUCACION FISICA TERCERO PRIMARIA	PENDIENTE
DONACION	SOCIALES PRIMERO PRIMARIA	RECHAZADA
PETICION	MATEMATICAS SEGUNDO BACHILLERATO	PENDIENTE
DONACION	MATEMATICAS PRIMERO PRIMARIA	ACEPTADA
DONACION	MATEMATICAS PRIMERO PRIMARIA	ACEPTADA
DONACION	SOCIALES SEGUNDO BACHILLERATO	PENDIENTE
PETICION	MATEMATICAS PRIMERO PRIMARIA	PENDIENTE
DONACION	MATEMATICAS PRIMERO PRIMARIA	PENDIENTE

Ilustración 35: Estado de las donaciones y peticiones pendientes

Por otra parte, el administrador accederá a un menú de inicio diferente al del usuario y es el que vemos en la Ilustración 36.

Desde este menú podrá aprobar las peticiones de préstamo y las donaciones propuestas por los usuarios. En primer lugar, para aprobar o denegar las peticiones de libros de texto que hayan realizado los distintos usuarios, el administrador hará clic sobre “Peticiones pendientes” y le aparecerá una lista con los libros solicitados.



Ilustración 36: Menú administradores

El administrador podrá aceptar todas las peticiones pendientes a la vez, pero también podrá ir aprobando o denegando individualmente cada una de ellas haciendo clic sobre la petición que vemos en la Ilustración 37.



Ilustración 37: Peticiones pendientes

Para el caso de aprobar o denegar las donaciones, el procedimiento a seguir por el administrador es prácticamente idéntico al que se acaba de exponer. En el menú de inicio clicará sobre “Donaciones por aprobar” y le aparecerá una pantalla como la de la Ilustración 38 con una lista de las donaciones que han propuesto los distintos usuarios.

LIBRO	USUARIO	FECHA RESERVA
SOCIALES SEGUNDO BACHILLERATO EDELVIVES	martuchi.7 .95@gmail.com	Fri Dec 31 18:25:26 GMT+01:00 2021
BIOLOGIA PRIMERO BACHILLERATO SM	martuchi.7 .95@gmail.com	Sun Jan 02 20:04:28 GMT+01:00 2022
MATEMATICAS PRIMERO PRIMARIA SM	anita1915 _dd@hotmail.com	Tue Sep 07 03:18:17 GMT+02:00 2021
MATEMATICAS PRIMERO PRIMARIA SM	martuchi.7 .95@gmail.com	Wed Dec 22 11:17:02 GMT+01:00 2021
EDUCACION FISICA TERCERO PRIMARIA SM	martuchi.7 .95@gmail.com	Wed Dec 29 19:59:20 GMT+01:00 2021

ACEPTAR TODAS LAS DONACIONES

Ilustración 38: Donaciones por aprobar

Como en el caso anterior, podrá ir aceptando o denegando cada una de las solicitudes de manera individual o podrá aceptarlas todas conjuntamente.

Para que pueda llevar un mayor control sobre la aplicación, al administrador se le ha proporcionado la opción de comprobar el listado de libros y su estado. Para ello, hará clic sobre “Lista de libros” del menú de inicio y elegirá cómo quiere que aparezcan ordenados los libros.

Para ejemplificar esta opción, se ha elegido la opción de “Listar todos”. Como se observa en la Ilustración 39, en la pantalla le aparecerá la información del libro, el usuario que lo ha solicitado y la fecha en la que debe devolver el ejemplar que se le ha prestado.

BookApp

PRESTAMOS ACTUALES

LISTAR TODOS
 LISTAR POR CLASES
LISTAR

USUARIO	LIBRO	FECHA DEVOLUCION
tediego@ucm.es	MATEMATICAS PRIMERO PRIMARIA	25 / 6 / 2022
anita1915_dd@hotmail.com	SOCIALES PRIMERO PRIMARIA	25 / 6 / 2022
martuchi.7.95@gmail.com	BIOLOGIA TERCERO ESO	23 / 4 / 2022
anita1915_dd@hotmail.com	MATEMATICAS PRIMERO PRIMARIA	25 / 6 / 2022
tfgmartadd@gmail.com	LENGUA PRIMERO PRIMARIA	2 / 9 / 2021
ester@gmail.com	MATEMATICAS PRIMERO PRIMARIA	6 / 9 / 2021
martuchi_7_95@hotmail.com	MATEMATICAS PRIMERO PRIMARIA	25 / 6 / 2022
anita1915_dd@hotmail.com	BIOLOGIA PRIMERO PRIMARIA	25 / 6 / 2022
tfgmartadd@gmail.com	MATEMATICAS PRIMERO PRIMARIA	25 / 6 / 2022

Ilustración 39: Listar todos los prestamos

Como se comentó, en este apartado tan solo se explicarían las operaciones más relevantes que permite la aplicación tanto al usuario como al administrador. Para entrar en más detalle, se adjunta el Anexo VI: Guía de usuario.

7. Conclusiones

En este último apartado, se recogen las conclusiones que se han obtenido tras la realización del presente trabajo de final de carrera. La intención es hacer una breve evaluación del trabajo respecto a los objetivos que se perseguían.

A modo de recordatorio, al principio del trabajo nos propusimos tres tipos de objetivos: tipo software, técnicos y personales. Así, las conclusiones irán referidas a cada uno de estos objetivos

- Respecto a los objetivos software que eran: gestión de usuarios, gestión de libros, gestión de imágenes, gestión de préstamos, escaneo de códigos, gestión de peticiones y gestión de donaciones, creo que se han logrado satisfacer. Se ha conseguido desarrollar e implementar satisfactoriamente la aplicación móvil para dispositivos Android. Estaría lista para poder ser utilizada en centros escolares de manera que facilitara el préstamo de libros entre los administradores y los alumnos. Permite el registro de nuevos usuarios y es capaz de gestionar correctamente tanto los datos de los usuarios como de los distintos libros e imágenes. También gestiona los préstamos y donaciones de libros de texto y permite el escaneo de códigos de barras o QR para la localización de un libro o usuario. Por tanto, se cumplen los objetivos software que se pusieron al inicio del proyecto.

- Respecto a los objetivos técnicos, la aplicación resultante es fácil de entender y de usar. Cuenta con una base de datos común pero que no permite a los usuarios acceder a los datos privados de otros usuarios. Por último, aún no es comprobable si la aplicación es escalable o no, ya que el número de usuarios todavía no es muy alto. A pesar de ello, todos los indicadores parecen señalar que sí lo será.

- Respecto a los objetivos personales, considero que este trabajo me ha permitido poner en práctica los conocimientos que he ido adquiriendo progresivamente a lo largo de mis años de estudio en este grado. Me ha servido también para desarrollar mis capacidades de aprendizaje autodidacta, sobre todo respecto a la utilización de la base de datos Firebase. Por otra parte, considero que este trabajo ha sido un reto bastante grande a nivel personal puesto que nunca antes había tenido que desarrollar por mí misma una aplicación de este tamaño. Durante todo el proceso me he enfrentado a muchas situaciones de bloqueo o de saturación que he tenido que aprender a gestionar.

Por todo esto, se concluye que la aplicación BookApp ha sido completada correctamente cumpliendo en líneas generales con todos los objetivos que se plantearon al inicio del trabajo.

8. Líneas de trabajo futuras

De cara a ampliar y mejorar las funcionalidades de la aplicación desarrollada, se podrían poner en práctica distintas acciones como las que se presentan a continuación:

- Mejorar los datos que se almacenan en el perfil del usuario: de esta manera, la información que se presenta de cada perfil creado sería más interesante y completa.

- Introducir un chat en la aplicación: una vez el administrador aprobara las donaciones propuestas por los usuarios, el chat permitiría que los donantes fueran capaces de ponerse en contacto directamente con los usuarios que hubieran solicitado el préstamo de dichos ejemplares a través de la aplicación.

- Posibilidad de mandar notificaciones a los usuarios cuando tanto sus préstamos como sus donaciones propuestas hayan sido aprobadas por el administrador, de manera que no tuviera que estar permanentemente pendiente de entrar en la aplicación para comprobar el estado de sus peticiones.

- Crear alertas que avisen al usuario de la cercanía del vencimiento del préstamo solicitado para que antes de que el plazo venza, el libro haya sido devuelto.

- Por último, incrementar la carga de datos posible para que, en vez de un solo colegio, el rango de colegios que puedan participar en el sistema a la vez se amplíe.

9. Referencias

- Java para Windows(versión 8) [Software] (2021). Recuperado de https://www.java.com/es/download/ie_manual.jsp
- Android Studio (2020.3.1 for Windows 64-bit) [Software] Recuperado de https://developer.android.com/studio?hl=es&gclid=CjwKCAiAlfqOBhAeEiwAYi43Fx-6zTWhLwry1Yh6V2HIuDzp-vptASKbiBROuuqAS03jHLV2y4mGWxoCd8UQAvD_BwE&gelsrc=aw.ds [Accedido : 12-09-2021]
- Agrega Firebase al proyecto de Android (s. f.) Recuperado de <https://firebase.google.com/docs/android/setup> [Accedido: 16-09-2021]
- Elige una base de datos: Cloud Firestore o Realtime Database Primeros pasos con Firebase Authentication en Android (s. f.) Recuperado de <https://firebase.google.com/docs/auth/android/start?hl=es> [Accedido: 19-09-2021]
- Cloud Storage para Firebase (s. f.) Recuperado de <https://firebase.google.com/docs/storage?authuser=1> [Accedido: 10-10-2021]
- Librería AwesomeValidation (2021) Recuperado de <https://github.com/thyrlan/AwesomeValidation> [Accedido: 17-10-2021]
- ZXing Android Embedded (2021) Recuperado de <https://github.com/journeyapps/zxing-android-embedded> [Accedido: 17-11-2021]
- Visual Paradigm (Versión 16.3) [Software] Recuperado de <https://ap.visual-paradigm.com/university-of-salamanca/>
- Durán Toro, A., Bernárdez Jiménez, B. (2002). Metodología para la Elicitación de Requisitos de Sistemas Software (versión 2.3).
- Herramienta REM (2004) [Software] Recuperado de http://www.lsi.us.es/descargas/descarga_programas.php?id=3
- Microsoft Project (Versión 2111) [Software] (2021). Recuperado de <https://www.microsoft.com/es-es/microsoft-365/project/project-management-software> [Accedido: 12-09-2021]
- Modelo vista controlador (s. f.) Recuperado de <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html> [Accedido: 20-10-2021]