

Trabajo de Fin de Grado

Desarrollo de Aplicaciones Multiplataforma

# MyFavBook



*Sergio Blázquez Fernández*

*David Matías Mota*

*Iván Pérez Martínez*

*Tutora: Raquel Cerdá*

# Índice

1. Introducción	3
2. Palabras clave	4
3. Asignaturas aplicadas en el trabajo	5
4. Herramientas / Lenguajes utilizados	6
5. Componentes del equipo y aportación realizada	7
6. Fases del proyecto	8
7. Conclusiones y mejoras del proyecto	26
8. Bibliografía	28
9. Agradecimientos	29

## **1. Introducción**

MyFavBook es una herramienta con la que puedes llevar un seguimiento de los libros que has leído, poder calificarlos con una puntuación de 0 a 10 y hacer un resumen de lo que te ha parecido, con el fin de tener una pequeña base de datos la cual poder consultar en cualquier momento.

Mediante un buscador escribimos el título del libro o el autor y podemos añadirlo a nuestra base de datos, en la pestaña MyBooks.

La aplicación está enfocada, en gran parte, para personas que leen muchos libros y desean llevar un registro con su toque personal. Creado sin ánimo de lucro con el fin de poder ayudar a los lectores.

La idea surgió tras conocer a varias personas que se leen varios libros al mes y en ocasiones no son capaces de recordar a lo largo de los años si cierto libro se lo han leído o recordar vagamente de qué trata, por lo que hemos ideado una aplicación que les pueda facilitar ambas cosas.

## 2. Palabras clave

**Activity:** Es una referencia al punto de entrada de nuestra aplicación, que se relaciona con las acciones que puede realizar un usuario.

**API:** En castellano se denomina como interfaz de programación de aplicaciones (Application Programming Interface). La podemos entender como un código que indica a las aplicaciones cómo pueden mantener una comunicación entre sí.

**APP:** También denominada aplicación, es un programa de software que está diseñado para hacer una función determinada.

**Authentication:** Servicio de Firebase para comprobar que los usuarios que accedan a nuestra aplicación estén registrados.

**Firebase:** Es una plataforma de desarrollo de apps de Google, que ayuda a compilar y desarrollar las apps.

**Firestore Database:** Es la base de datos de Firebase, que almacena los datos en formato JSON.

**Interfaz:** Es una conexión funcional entre dos componentes, dispositivos, programas o ventana gráfica de un programa con el que interactuamos.

**Jackson:** Librería de Java que sirve para serializar.

**JSON:** JavaScript Object Notation, es un formato ligero de intercambio de datos.

**Layout:** Hace referencia a la disposición que toman los elementos dentro de una ventana, como botones, texto...

**Picasso:** Es una librería de Android que se usa para cargar imágenes en nuestra aplicación

**ResquestQueue:** Cola de peticiones que usamos para no crear varios hilos.

**REST:** Deriva de 'REpresentational State Transfer'. Es una interfaz para conectar varios sistemas basados en el protocolo HTTP, y sirve para obtener y generar datos y operaciones, devolviendo esos datos en formatos como XML o JSON.

**Volley:** Biblioteca HTTP que facilita y agiliza el uso de redes en apps para Android.

**XML:** Proviene de ‘Extensible Markup Lenguaje’, que se traduce como ‘Lenguaje de Marcado Extensible’, permite definir un conjunto de reglas de codificación de documentos en un formato legible tanto por el hombre como por la máquina.

### **3. Asignaturas aplicadas en el trabajo**

**Inglés:** Se ha decidido hacer todos los comentarios de código y los commit en GitHub en inglés.

**Lenguajes de Marcas:** Nuestra base de datos funciona en formato JSON y las estructuras de las actividades de Android Studio se crean mediante XML.

**Programación:** Se ha programado íntegramente en Java.

**Bases de Datos:** Usamos la base de datos de Google para almacenar los datos y poder realizar consultas.

**Programación Multimedia y Dispositivos Móviles:** Se ha utilizado Android Studio para la elaboración íntegra del proyecto.

**Entornos de Desarrollo:** Hemos usado el control de versiones GitHub para ir subiendo el código que hemos hecho individualmente y poder fusionarlo con el de los demás integrantes del proyecto. También se han usado los conocimientos adquiridos en cuanto a depuración y optimización.

**Programación de Servicios y Procesos:** Se ha implementado un servicio REST.

#### **4. Herramientas / Lenguajes utilizados.**

**Java:** El lenguaje que hemos decidido usar para crear nuestro proyecto, ya que es el lenguaje el cual hemos trabajado los dos años y en el que nos desenvolvemos mejor.

**Android Studio:** El entorno de trabajo que hemos usado, ya que nos hemos decantado por hacer una aplicación Android.

**Postman:** Se ha utilizado para probar las conexiones con la API de Google Books.

**app.diagrams.net:** Con esta herramienta online hemos hecho el diagrama de clases y el modelo de datos.

**Picasso:** Librería Android que hemos usado para cargar imágenes al hacer las peticiones desde Google Books.

**Volley:** Lo usamos para hacer una cola de peticiones, que trae por defecto una memoria caché y una conexión de red básica. También podemos personalizarlas, pero en nuestro caso no ha sido necesario.

**Firestore:** Hemos utilizado Firestore como gestor de nuestra base de datos. Contiene dos tipos de bases de datos y nosotros nos hemos decantado por Firestore Database, ya que los datos que vamos a almacenar son muy simples.

## **5. Componentes del equipo y aportación realizada.**

### **- David Matías**

David ha creado el NavigationBar junto con sus estilos y la lógica. La clase 'Search' con su lógica e implementando las llamadas a la API. Ha dado la lógica de los botones add, delete, buy, preview, review y rate de la clase 'BookDetails', diseño de un Rating bar que al final se ha descartado por simplificar. Ha creado la plantilla de las clases BookDetails, Book\_rv\_item y el adapter. Ha realizado también la conexión a la BBDD Firebase junto con las operaciones de lectura y escritura. Ha implementado las librerías Volley y Picasso. Crear clase 'books' y 'book' y crear el fondo para el logo del escritorio. También ha creado el README. Ha trabajado en las ramas 'david' y 'API' de Git.

### **- Sergio Blázquez**

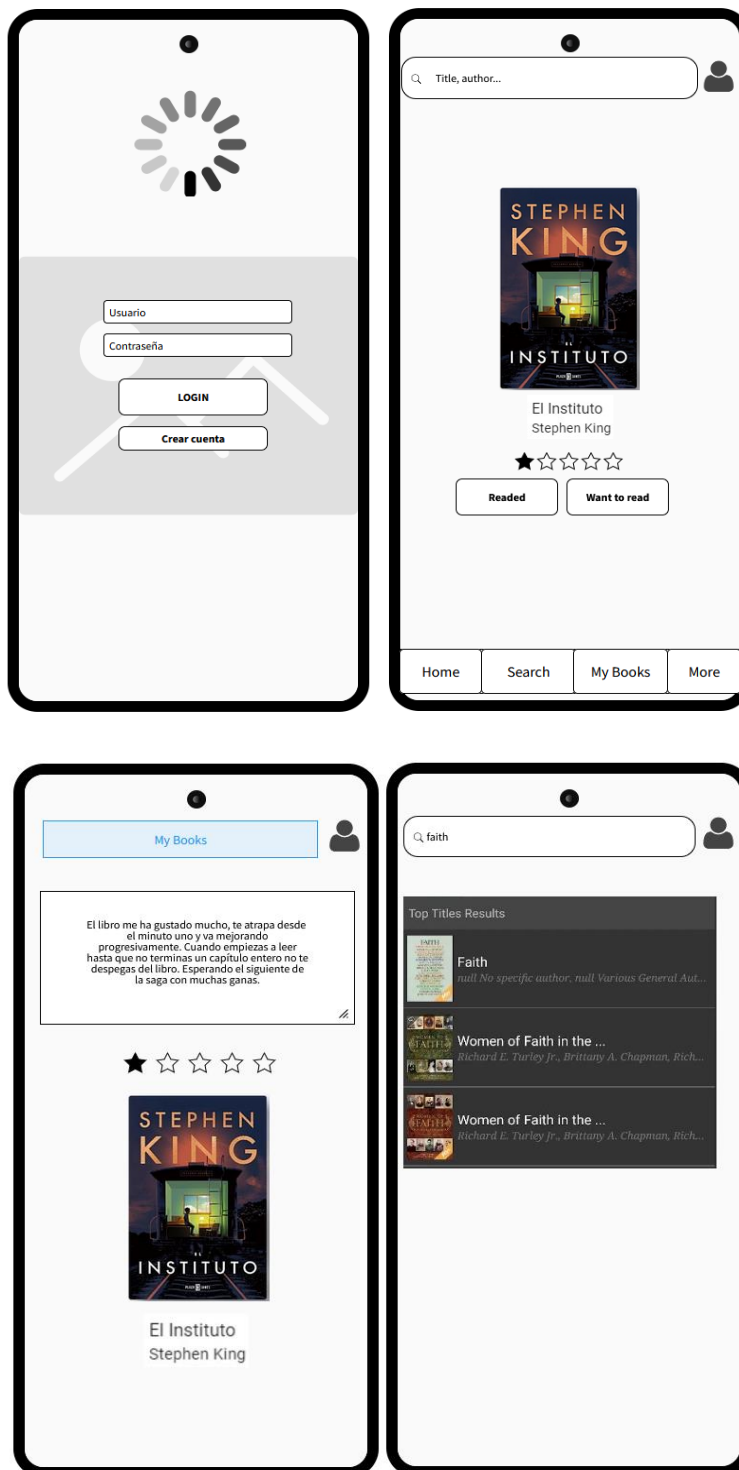
Sergio ha diseñado y dado estilo a las cajas de texto de la pantalla Login y Register, ha creado el botón add de la clase 'bookDetails' y también ha creado la vista 'Faq' donde ha implementado un FAQ con respuesta oculta que nos da información sobre nuestra aplicación. Ha trabajado en la rama 'sergio' de Git.

### **- Iván Pérez**

Iván ha aportado la idea del proyecto, diseñando los mockups, dando los estilos a la aplicación, logo de MyFavBook junto con el background. Ha implementado todo el sistema de Autentificación junto con las credenciales, hecho el 'logout', ha ayudado a David con la parte de base de datos, ha refactorizado el código, creado la memoria del proyecto, el esquema de datos, comentado métodos para describir su funcionalidad, corregido bugs, rehecho la lógica del almacenamiento en base de datos para poder almacenarlo en una única librería y que los usuarios puedan coger de ahí los libros. Ha trabajado en la rama 'main' y en 'LastChanges'.

## 6. Fases del proyecto.

Estos son los mockups que hemos utilizado como plantilla para trabajar:



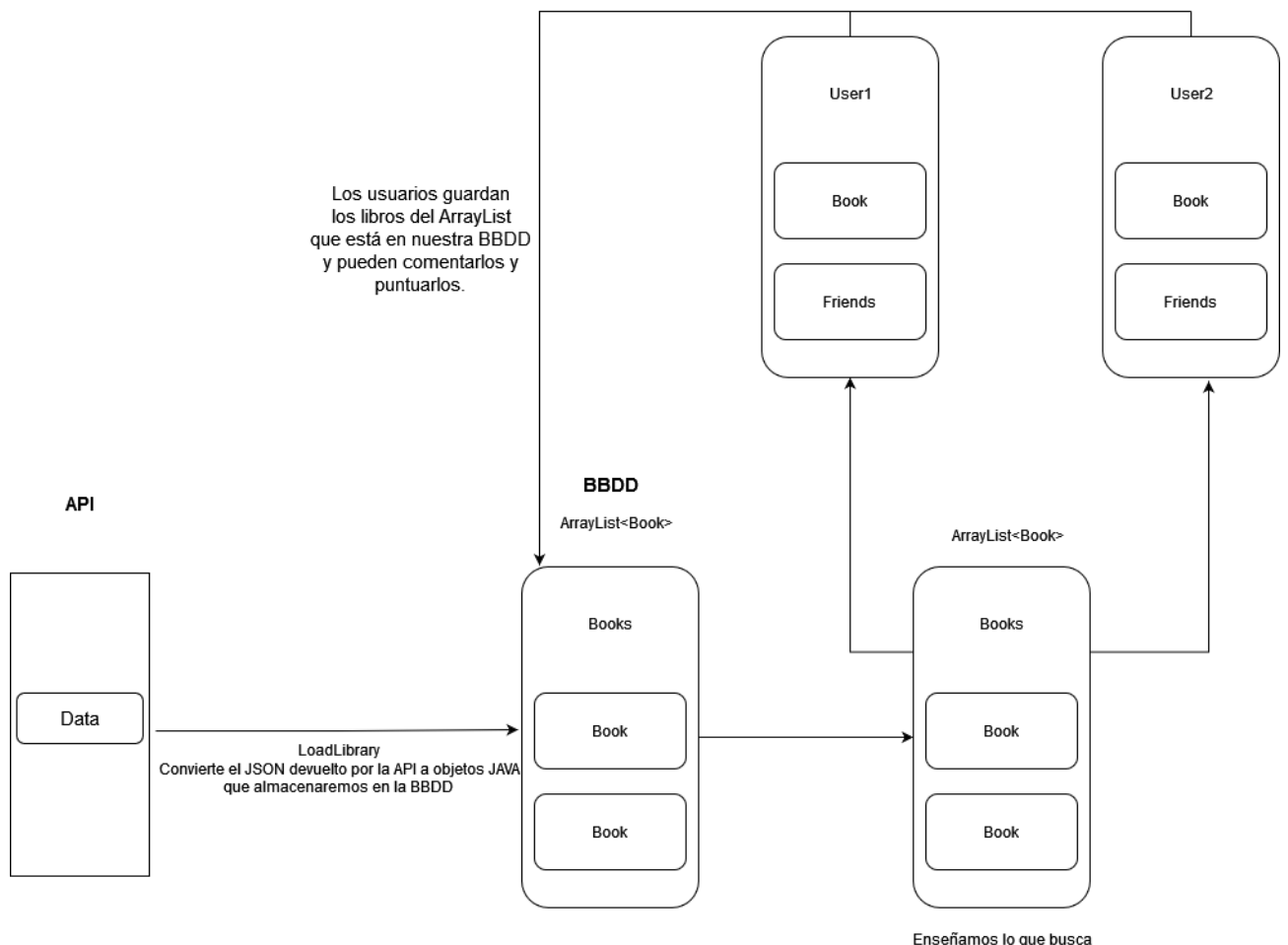
### 6.1 Mockups

Lo hemos adaptado un poco a lo que pensábamos que era lo óptimo.



La idea pasaba por tener un sistema de autenticación, una llamada a la API para poder guardar el libro en nuestra base de datos y una vez guardado poder hacer una reseña y puntuarlo.

Al usar la base de datos de Firebase, que es no relacional, hemos seguido este modelo:



## 6.2 Esquema de datos

El funcionamiento es el siguiente; hacemos una llamada a la API con una búsqueda del libro y todos los resultados de la llamada se nos guardan en nuestra base de datos, en una colección llamada Books. Desde esta colección el usuario accede a los libros a los que podrá añadir una reseña y una puntuación. Si otro usuario buscara el mismo libro, la app miraría primero en BBDD si el libro existe, si no, haría la llamada a la API para repetir el proceso de guardar los resultados de la búsqueda en nuestra BBDD.

Comenzamos a configurar nuestra App con Android Studio.

El proyecto se programó en Java, ya que fue el lenguaje usado durante el curso. Hemos escogido el emulador con una API 30, y el Nexus 10 como dispositivo.

En el archivo build.gradle (project) añadimos los servicios de Google.

```
dependencies {  
    // ...  
    classpath 'com.google.gms:google-services:4.3.14'  
}
```

### *6.3 Dependencias project*

También los añadimos en el archivo build.gradle (app).

```
plugins {  
    id 'com.android.application'  
    id 'com.google.gms.google-services'  
}
```

### *6.4 Plugins app*

En este mismo archivo añadimos todas las dependencias que vamos a usar, como librerías, el sistema de autenticación, base de datos...

```
dependencies {  
  
    implementation 'androidx.appcompat:appcompat:1.5.1'  
    implementation 'com.google.android.material:material:1.7.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
    implementation 'com.google.android.material:material:1.7.0'  
    implementation 'androidx.annotation:annotation:1.5.0'  
    implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.5.1'  
    implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.5.1'  
    implementation 'com.google.firebase:firebase-auth:21.1.0'  
    implementation 'com.google.firebase:firebase-firestore:24.4.1'  
    implementation 'androidx.navigation:navigation-fragment:2.5.3'  
    implementation 'androidx.navigation:navigation-ui:2.5.3'  
    implementation 'com.android.volley:volley:1.2.1'  
    implementation 'com.squareup.picasso:picasso:2.71828'  
    implementation 'com.google.firebase:firebase-database:20.1.0'  
    implementation 'com.fasterxml.jackson.core:jackson-databind:2.10.3'  
    testImplementation 'junit:junit:4.13.2'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.4'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.0'  
}
```

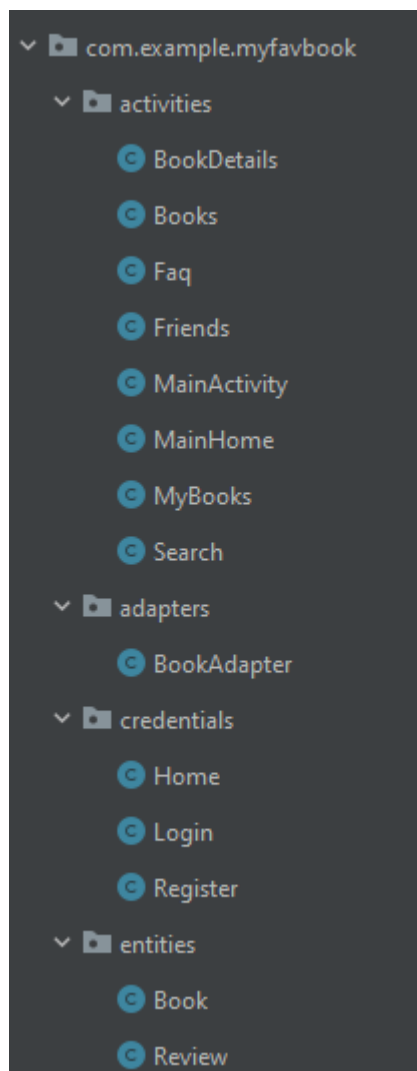
### *6.5 Dependencias*

Para terminar la configuración del proyecto, damos permisos para la conexión a internet en el AndroidManifest.xml:

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

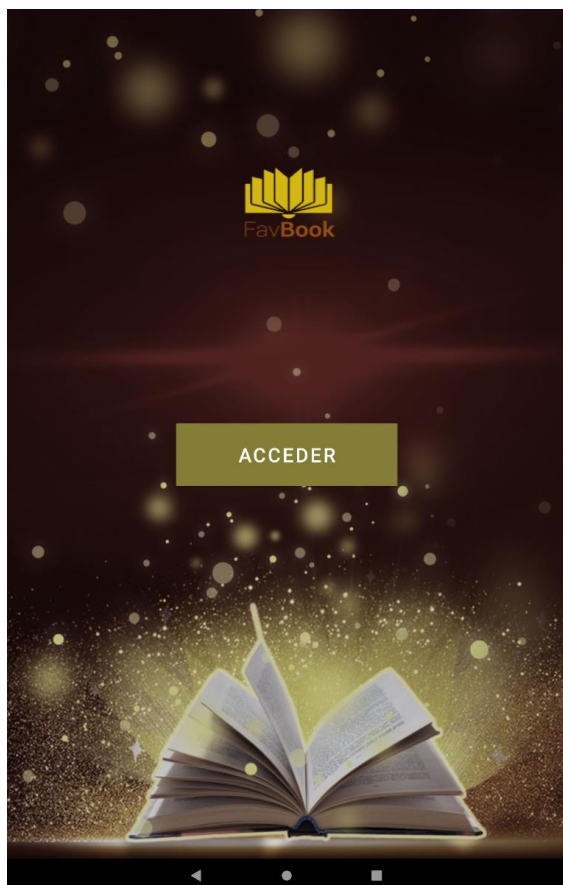
## 6.6 Permisos

La estructura del proyecto sería la siguiente:



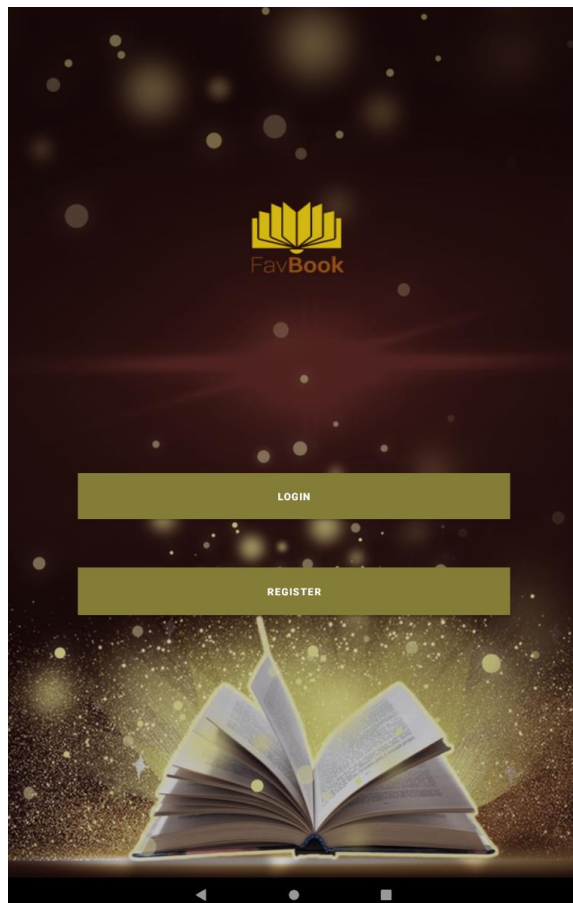
## 6.7 Estructura del proyecto

Cuando arrancamos la app tenemos la vista activity\_main:



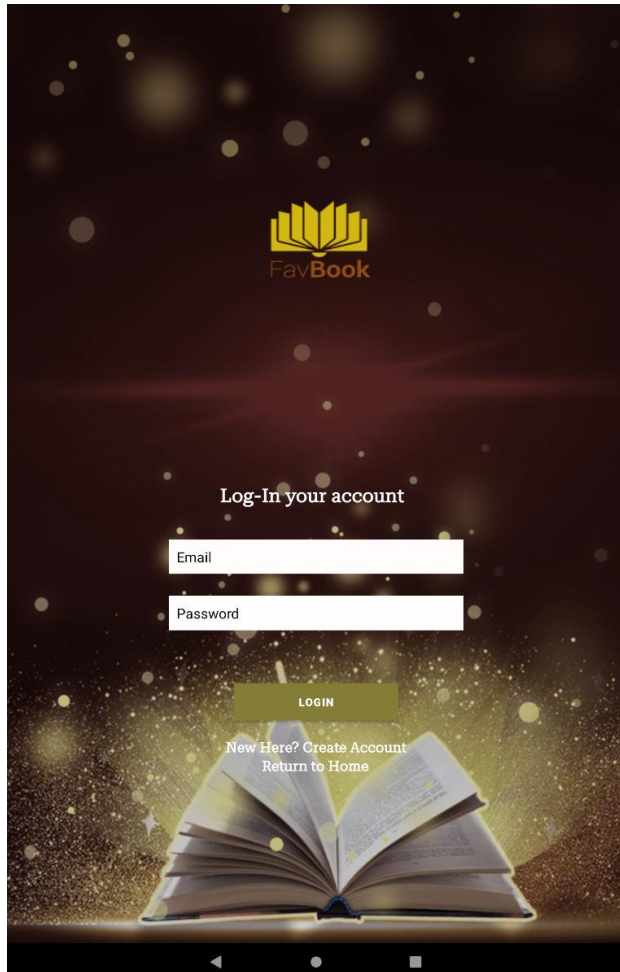
*6.8 Activity\_main*

Al pulsar en el botón acceder vamos a la ventana activity\_home:



### *6.9 Activity\_home*

En esta ventana podemos o bien loguearnos o bien registrarnos si aún no tenemos una cuenta creada.

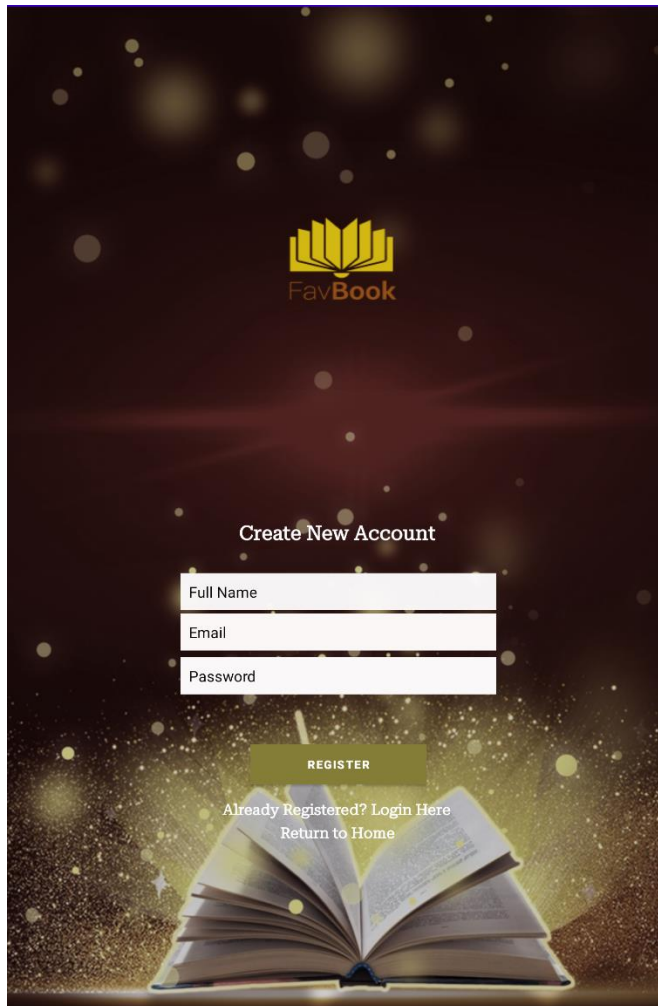


### *6.10 Activity\_login*

Si pulsamos el botón Login nos redirige a la vista `activity_login`, la cual tendremos que introducir un correo electrónico y un password. También nos puede redirigir a la pantalla de `activity_main` pinchando en; 'New here? Create account return to home'.

Si nos logueamos correctamente saldrá un toast diciendo que nos hemos logueado con éxito, si la contraseña no se ha escrito correctamente nos dará otro aviso con el problema que ha detectado.

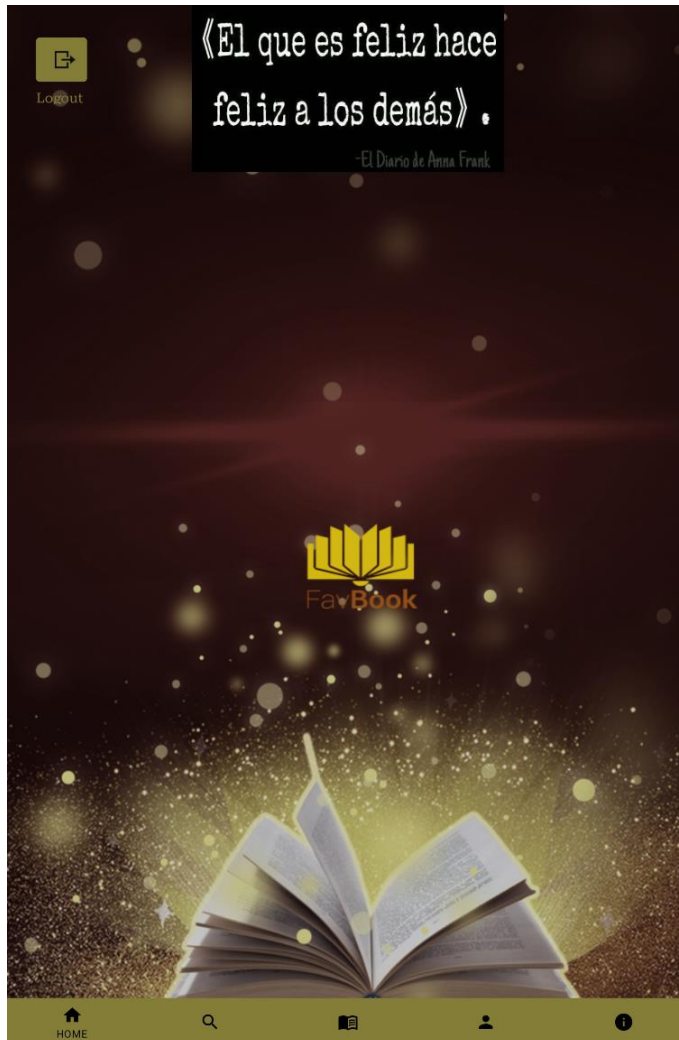
Si aún no tenemos cuenta y queremos registrarnos, pulsamos el botón Register y nos redirige a la `activity_register`, que nos muestra lo siguiente:



### *6.11 Activity\_register*

Rellenamos el nombre completo, correo electrónico y el password y si todo ha ido bien nos saldrá un toast diciendo que el usuario se ha creado. Si ha habido un problema nos saldrá otro avisándonos de lo que ha ocurrido.

Una vez logueados nos redirige al activity\_main\_home:



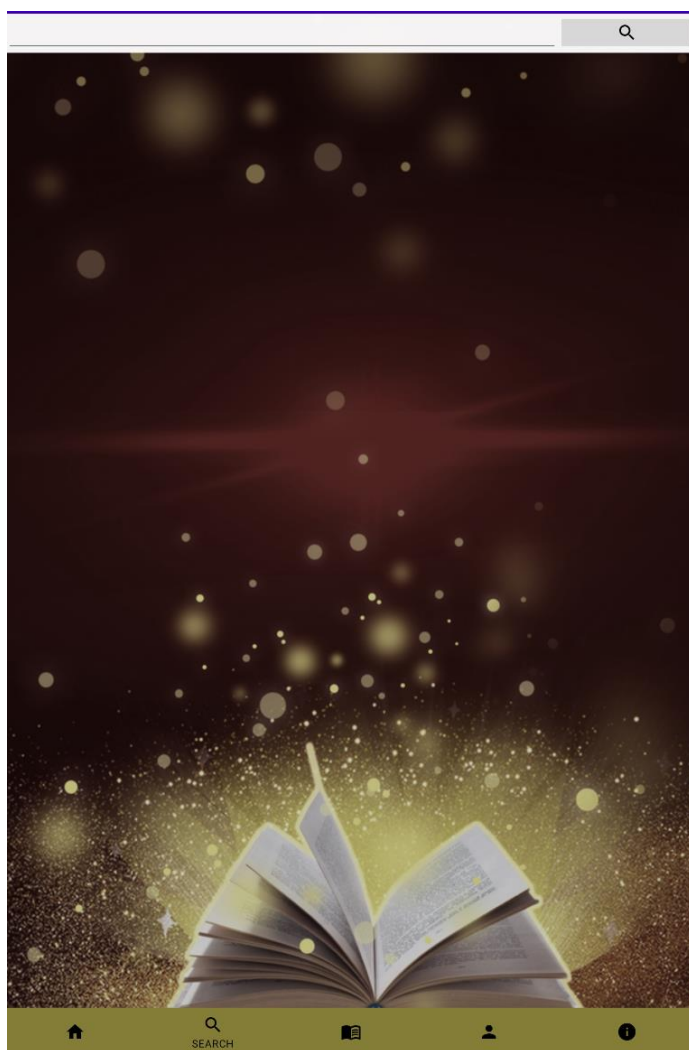
### 6.12 Activity\_main\_home

Aquí tenemos el botón del logout y en la parte inferior la barra de navegación con 5 botones. De izquierda a derecha son el home, search, mybooks, friends y faq.

La vista actual corresponde con el botón home.



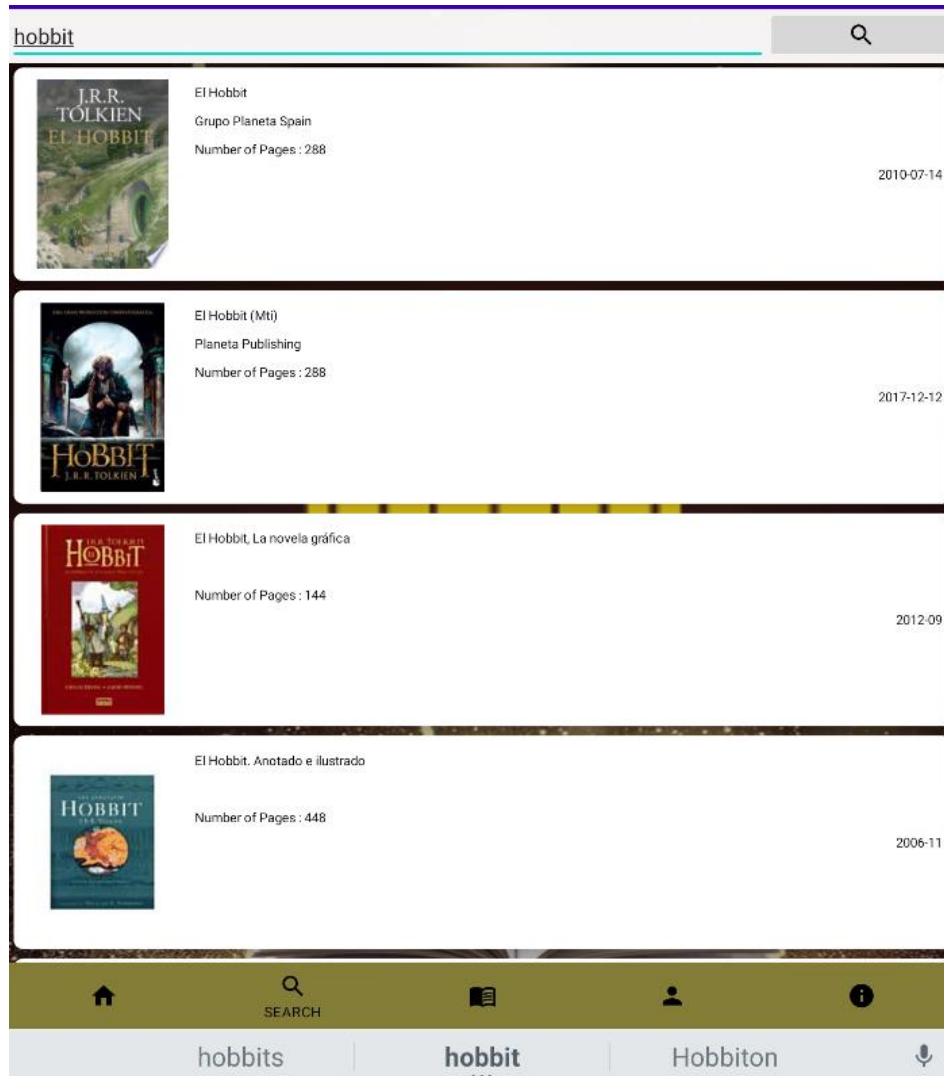
Si pulsamos el botón de search nos redirige a su vista:



### *6.13 Activity\_search*

En la parte superior se nos habilita la barra de búsqueda, la cual hace una llamada a la API de Google Books para buscar el libro que queramos.

Probamos a buscar con la palabra 'hobbit y nos aparece lo siguiente:

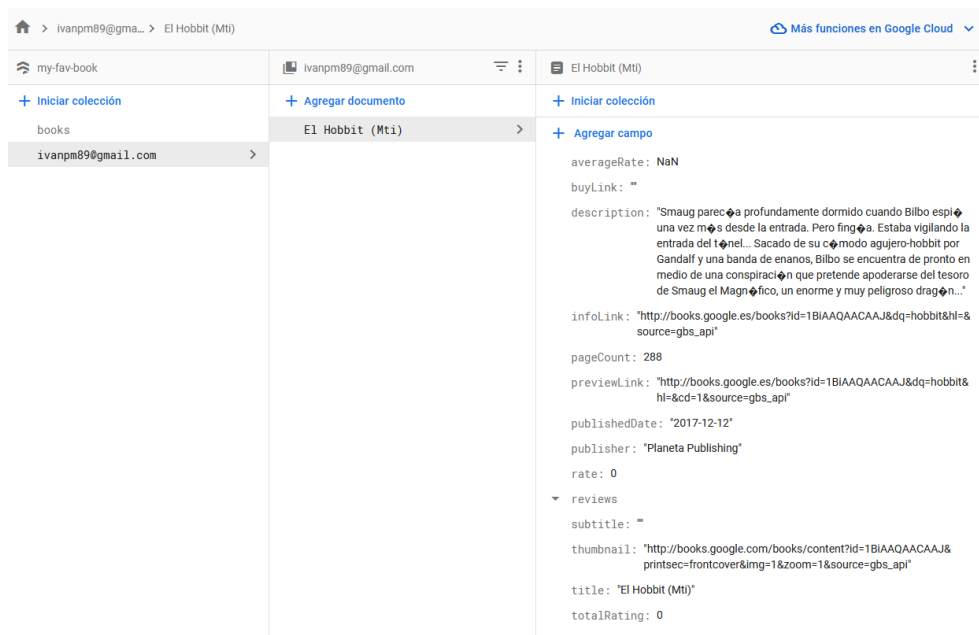


#### 6.14 Resultados de búsqueda

Una vez hecha la búsqueda se nos ha quedado guardada en nuestra base de datos. Para comprobarlo vamos a la consola de firebase <https://console.firebase.google.com/u/0/project/my-fav-book/firestore> y vemos que ha creado la colección books y ha añadido la búsqueda íntegra.

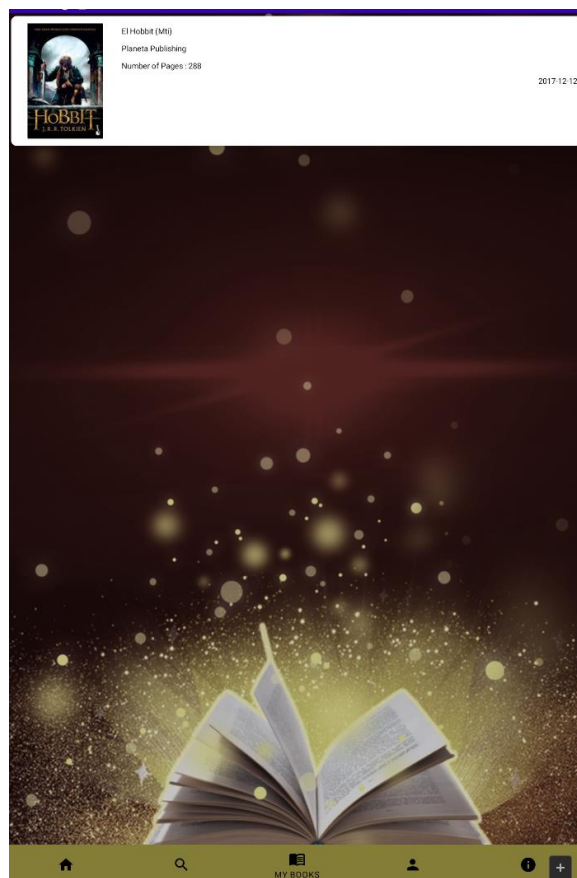


Y vemos que se nos ha guardado en nuestro usuario.




### 6.17 Almacenamiento user en Firebase

También lo tenemos en la pestaña 'mybooks'.



### 6.18 Libro añadido a mybooks

## Le Podemos añadir un review y una nota



Planeta Publishing  
No Of Pages : 288  
Published On : 2017-12-12

El Hobbit (Mti)

Smaug parecía profundamente dormido cuando Bilbo espió una vez más desde la entrada. Pero fingía. Estaba vigilando la entrada del tónel... Sacado de su cómodo agujero-hobbit por Gandalf y una banda de enanos, Bilbo se encuentra de pronto en medio de una conspiración que pretende apoderarse del tesoro de Smaug el Magnífico, un enorme y muy peligroso dragón...

Preview

Buy

Add

Delete

Rating: 0/10

Me ha parecido muy bueno


Review

9

Rate

### 6.19 Vista libros para añadir reseña y nota

Y los datos se quedan almacenados



Planeta Publishing  
No Of Pages : 288  
Published On : 2017-12-12

El Hobbit (Mti)

Smaug parecía profundamente dormido cuando Bilbo espió una vez más desde la entrada. Pero fingía. Estaba vigilando la entrada del tónel... Sacado de su cómodo agujero-hobbit por Gandalf y una banda de enanos, Bilbo se encuentra de pronto en medio de una conspiración que pretende apoderarse del tesoro de Smaug el Magnífico, un enorme y muy peligroso dragón...

Preview

Buy

Add

Delete

Rating: 9/10

Review

Review

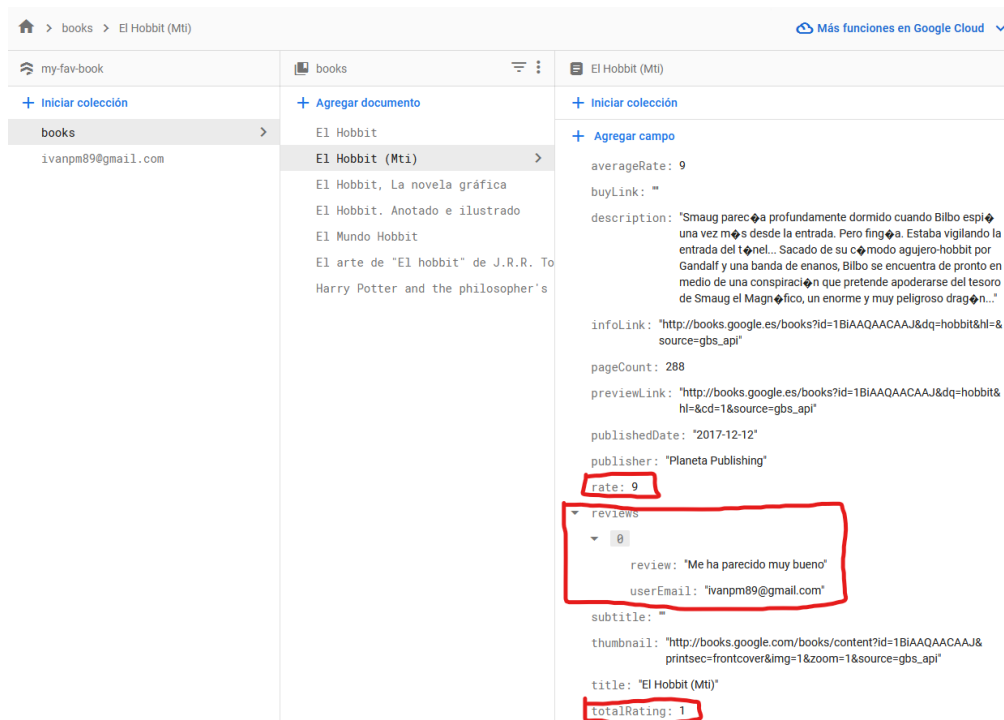
Rate

Rate

Me ha parecido muy bueno - ivanpm89@gmail.com

### 6.20 Vista libros con su reseña y nota añadidos

Comprobamos en firebase que los datos también se han almacenado en nuestra colección de libros.



### 6.21 Comprobación en Firebase de datos añadidos

Los reviews se guardan en un ArrayList de reviews, el cual nos muestra tanto el user que lo ha hecho como como el string del review.

Hemos probado a añadir el libro de El Hobbit a nuestro usuario ([ivanpm89@gmail.com](mailto:ivanpm89@gmail.com)) ahora lo que vamos a hacer es con otro usuario añadir el mismo libro y ponerle otro review y una nota distinta. En este caso le hemos puntuado con un 10.



Planeta Publishing  
No Of Pages : 288  
Published On : 2017-12-12

El Hobbit (Mti)

Smaug parecía profundamente dormido cuando Bilbo espió una vez más desde la entrada. Pero fingía. Estaba vigilando la entrada del tónel... Sacado de su cómodo agujero-hobbit por Gandalf y una banda de enanos, Bilbo se encuentra de pronto en medio de una conspiración que pretende apoderarse del tesoro de Smaug el Magnífico, un enorme y muy peligroso dragón...

Preview

Buy

Add

Delete

Rating: 9.5/10

Review

Review

Rate

Rate

Me ha parecido muy bueno - ivanpm89@gmail.com

Es una maravilla - test@gmail.com

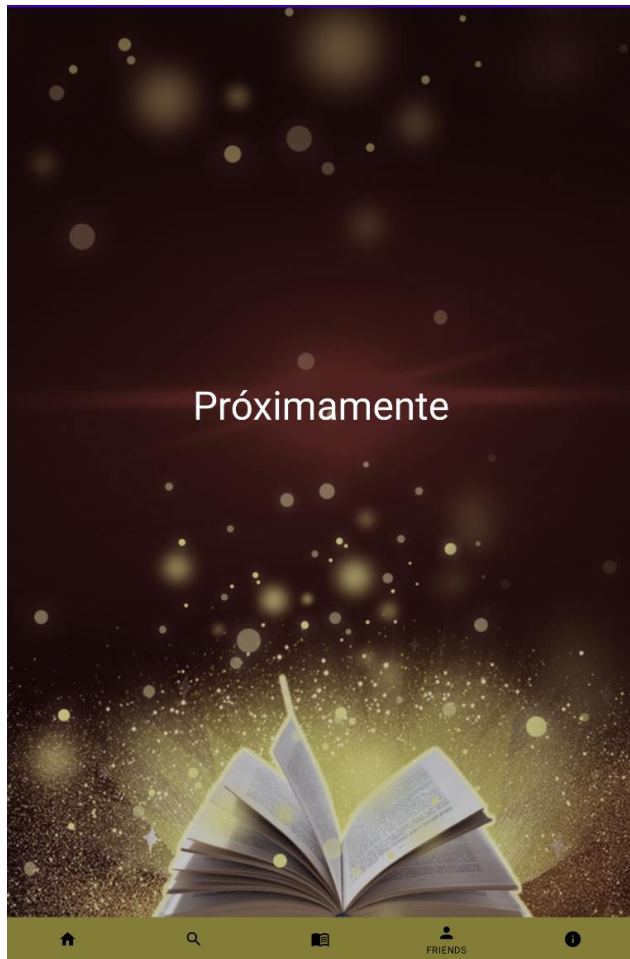
## 6.22 Vista libros con doble reseña y nota

Al haberlo puntuado con un 10, y ya tenía un 9, suma las dos puntuaciones y lo divide entre las dos por lo que nos da 9.5 de media. Mostramos la parte de firebase a continuación.

```
{
  "books": {
    "El Hobbit (Mti)": {
      "averageRate": 9.5,
      "buyLink": "",
      "description": "Smaug parec\u00eda profundamente dormido cuando Bilbo espi\u00f3 una vez m\u00e1s desde la entrada. Pero fing\u00eda. Estaba vigilando la entrada del t\u00f3nel... Sacado de su c\u00f3modo agujero-hobbit por Gandalf y una banda de enanos, Bilbo se encuentra de pronto en medio de una conspiraci\u00f3n que pretende apoderarse del tesoro de Smaug el Magn\u00edfico, un enorme y muy peligroso drag\u00f3n...",
      "infoLink": "http://books.google.es/books?id=1BIAQAACAAJ&dq=el+hobbit&hl=&source=gbv_api",
      "pageCount": 288,
      "previewLink": "http://books.google.es/books?id=1BIAQAACAAJ&dq=el+hobbit&hl=&cd=1&source=gbv_api",
      "publishedDate": "2017-12-12",
      "publisher": "Planeta Publishing",
      "rate": 10,
      "reviews": [
        {
          "review": "Me ha parecido muy bueno",
          "userEmail": "ivanpm89@gmail.com"
        },
        {
          "review": "Es una maravilla",
          "userEmail": "test@gmail.com"
        }
      ],
      "subTitle": "",
      "thumbnail": "http://books.google.com/books/content?id=1BIAQAACAAJ&printsec=frontcover&img=1&zoom=1&source=gbv_api",
      "title": "El Hobbit (Mti)",
      "totalRating": 2
    }
  }
}
```

## 6.23 Base de datos con Array de rese\u00f1as y doble nota

La parte de 'friends' pretendemos implementarla en un futuro próximo, ya que tenemos la idea clara de lo que hacer. Será una especie de chat para interactuar con otros usuarios. En la parte de mejoras del proyecto adjuntaremos el mockup que hemos hecho.

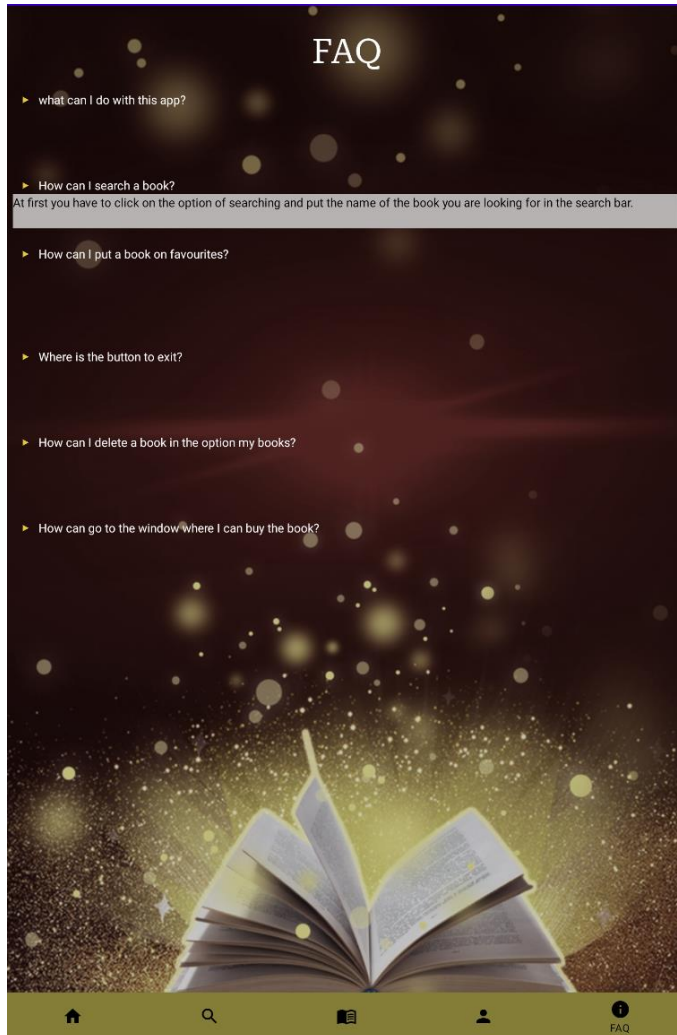


#### *6.24 Activity\_friends*

Por el momento lo tenemos con un ViewText avisando de que próximamente estará la implementación.



Y por último la view de FAQ, la cual mostramos unas preguntas básicas del manejo de la aplicación. Al clicar la pregunta nos muestra la respuesta.

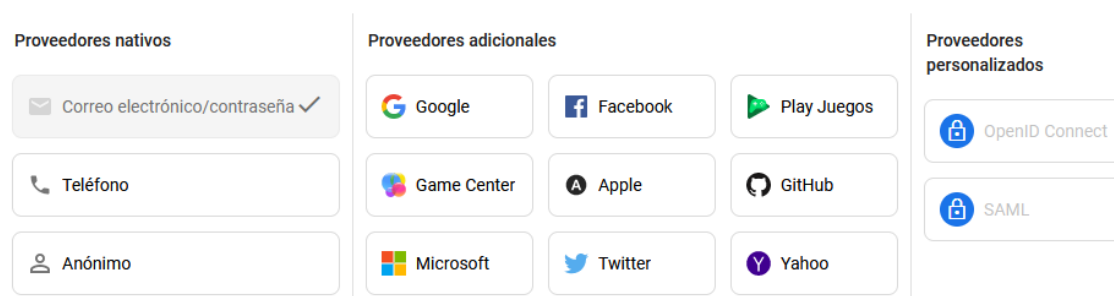


### 6.25 Activity\_faq

Siempre es recomendable hacer un logout, ya que si no lo hacemos se quedan nuestras credenciales almacenadas y no nos dejaría, por ejemplo, darle a register por si quisiésemos crear una nueva cuenta.

## 7. Conclusiones y mejoras del proyecto.

Hemos logrado hacer una aplicación 100% funcional acorde a lo mostrado en el anteproyecto, que tiene un sistema de logueo, con su registro, el cual va apoyado en un sistema de autenticación. Nos hemos decantado por ponerlo en exclusiva con email/contraseña, pero aparte podríamos haber añadido cualquiera de estos:



### 7.1 Sistemas de login

En un principio la idea iba a ser haber usado una base de datos SQL, al haber trabajado con ella durante todo el curso, pero Raquel nos dio la idea de mirar Firebase y nos pareció bastante interesante. Al principio nos costó hacernos a ella, ya que es una base de datos no relacional, y no habíamos trabajado con ellas antes.

Firebase tiene un potencial muy grande, ya que podemos implementar anuncios en nuestra app y llevar un registro monetario, ver los usuarios activos por día/mes, a través del Analytics Dashboard ver información de todo tipo, como por ejemplo el tiempo de interacción medio de los usuarios, ingresos publicitarios, ingresos por usuario por compra, usuarios por país...

La llamada a la API que usamos es a Google Books. Nos hubiese gustado hacerla con GoodReads, pero desde diciembre de 2020 ya no dan nuevas claves para su API pública.

Diseñamos varios logos para nuestra aplicación y nos decantamos por el actual. Los mostramos a continuación.



### 7.2 Logos

Como mejoras del proyecto tenemos pendientes las siguientes cosas.

- Implementar botón para editar la review y el rate.
- Implementar RatingBar para que muestre la nota conjunta con rate.
- 



### *7.3 RatingBar*

- Implementar funcionalidad de la vista Friends.

User 1
¡Hola amigo! ¿Me podrías comentar qué te ha parecido el libro de El Hobbit?

User 2
¡Hola!, pues la verdad que me ha encantado, Tolkien mima cada detalle al máximo y te atrapa en su mundo de fantasía. Te recomiendo que te lo leas cuanto antes.

### *7.4 Mockup futura implementación friends*

## 8. Bibliografía.

Android Studio <https://developer.android.com/studio/>

API [https://developers.google.com/books/docs/v1/getting\\_started](https://developers.google.com/books/docs/v1/getting_started)

Firebase <https://firebase.google.com/>

<https://firebase.google.com/docs?>

GitHub <https://github.com>

Icons <https://www.flaticon.com/> <https://icons8.com/icons/android>

Material <http://materialcolors.com/>

Mocks <https://wireframepro.mockflow.com/>

Picasso <https://square.github.io/picasso/>

Pics <https://www.freepik.es/>

Stackoverflow <https://stackoverflow.com/>

Udemy <https://www.udemy.com/>

Volley <https://google.github.io/volley/>

YouTube <https://www.youtube.com/>

## 9. Agradecimientos

David Matías:

*Llega a su fin esta etapa en la que he aprendido mucho a cerca de un sector que no era el mío. Han sido dos años duros por tener que compaginarlos con otro trabajo, pero también bonitos porque al final se ve el fruto de todo el esfuerzo. Me gustaría agradecer en primer lugar a mi familia que ha estado apoyándome en todo momento durante todo este tiempo y a mi novia con quien en muchas ocasiones no he podido pasar todo el tiempo que quería por tener que estudiar o hacer trabajos y aun así siempre ha estado apoyándome. Quería dar las gracias también a una persona que ha sido imprescindible durante este curso, mi amigo Gerard que me ha enseñado todo lo que sabe acerca de la programación y me ha ayudado en todo lo que he necesitado. Por último dar las gracias a los profesores y a los compañeros de los que he aprendido mucho y espero seguir aprendiendo en un futuro.*

Sergio Blázquez:

*Dar muchas gracias a todos los profesores por hacernos aprender cada asignatura tan amena y fácil y sobre todo un gran agradecimiento a Tomás por su gran empeño al alumnado y tanto entusiasmo al dar clase y explicar.*

Iván Pérez:

*Dar las gracias a todos y cada uno de los profesores en estos dos años de aprendizaje que nos han puesto a prueba en todo momento. Ha sido una etapa dura pero muy fructífera, ya que todo lo aprendido compensa con creces los momentos de haber querido tirar todo por la borda. También dar las gracias a todos los compañeros, en especial a Laurence, que su ayuda a lo largo de todo el curso ha sido imprescindible para poder llegar a este punto. A toda la familia, sin su apoyo nada de esto hubiese sido posible. Y como no, a Brody, mi amigo más incondicional.*

Repositorio: <https://github.com/ivaperma/myFavBook>