

QR BOOK - Memoria del proyecto



QR BOOK  
CFGS Desarrollo de Aplicaciones Multiplataforma

Sergio Plaza, David Calero  
Institut Puig Castellar  
CFGS DAM  
2



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## Resumen del proyecto:

QR BOOK es una aplicación desarrollada por APP World para fomentar la lectura entre los jóvenes estudiantes.

Su objetivo es gestionar una biblioteca de forma sencilla. Esta aplicación se encargará de la gestión de usuarios y libros almacenados en una base de datos mediante el uso de códigos QR.

Con esta aplicación el bibliotecario sólo habrá de escanear el código QR del libro y del usuario que lo alquila. La aplicación se encargará de establecer la relación en la base de datos y gestionar los tiempos de devolución avisando al usuario mediante notificaciones.

La aplicación se desarrollará en flutter con tal de que sea multiplataforma y usará una base de datos en firebase.

## Palabras clave:

*biblioteca, libros, código QR, lectores, ocio, alquiler.*

## Abstract:

QR BOOK is an application developed by APP World to promote reading among young students.

Its objective is to manage a library in a simple way. This application will be in charge of managing users and books stored in a database through the use of QR codes.

With this application the librarian will only have to scan the QR code of the book and the user who rents it. The application will be in charge of establishing the relationship in the database and managing the return times, notifying the user through notifications.

The application will be developed in flutter as long as it is cross-platform and will use a database in firebase.

## Keywords:

*library, books, QR code, readers, leisure, rental*

## Índice

<b>1 Introducción:</b>	<b>4</b>
1.1 Contexto:	4
1.2 Justificación	4
1.3 Objetivos	5
1.3.1 Objetivo general	5
1.3.2 Objetivos específicos	5
1.4 Estrategia y planificación del proyecto	5
1.5 Metodología de trabajo	6
1.6 Estudio económico y presupuestario	6
<b>2 Descripción del proyecto</b>	<b>7</b>
2.1 Análisis de requisitos	7
2.1.1 Requisitos funcionales	8
2.1.2 Requisitos no funcionales	9
2.2 Previsión de tareas de investigación	9
2.3 Tecnología	10
2.3.1 Comparativa de las tecnologías valoradas	10
2.3.2 Tecnologías escogidas	11
2.4 Estructura de proyecto	12
2.5 Definición de las funcionalidades	14
<b>3. Investigación.</b>	<b>17</b>
3.1 Investigación sobre dart.	17
3.2. Información sobre Firebase.	18
3.3. Información sobre códigos QR.	18
<b>4 Conclusiones</b>	<b>19</b>
4.1 Conclusiones generales del proyecto	19
4.2 Consecución de los objetivos	19
4.3 Valoración de la metodología y planificación	20
4.4 Visión de futuro	20
<b>5. Bibliografía</b>	<b>22</b>
<b>6 Anexos</b>	<b>22</b>

## **1 Introducción:**

QR BOOK es una aplicación desarrollada por APP World para fomentar la lectura entre los jóvenes estudiantes.

Su objetivo es gestionar una biblioteca de forma sencilla. Esta aplicación se encargará de la gestión de usuarios y libros almacenados en una base de datos mediante el uso de códigos QR.

Con esta aplicación pretendemos facilitar el trabajo de un bibliotecario, ya que este sólo habrá de escanear el código QR de un libro y del usuario que desea alquilarlo. También ayudaremos al usuario común a recordar sus requisitos como cliente de una biblioteca, notificando de los términos de devolución, mostrando publicaciones e incentivando a la lectura.

### **1.1 Contexto:**

Hemos decidido realizar esta aplicación ya que creemos que los hábitos de lectura se están perdiendo entre los jóvenes y una forma eficaz de recuperarlos sería acercarlos la lectura a las nuevas tecnologías.

Con nuestra aplicación no sólo facilitaremos las tareas del bibliotecario y el lector, sino también les incentivamos a seguir acudiendo a las bibliotecas con esta herramienta moderna.

### **1.2 Justificación**

Los jóvenes dedican muchas horas a estar con los móviles y tablets, el hecho de tener una aplicación en la biblioteca de su centro puede generar un interés y una motivación para iniciarse en la lectura.

## 1.3 Objetivos

### 1.3.1 Objetivo general

El objetivo general es desarrollar una aplicación con una interfaz sencilla, fácil de usar y eficaz. Queremos fomentar la lectura y facilitar el trabajo de un bibliotecario, por eso la aplicación ha de ser una herramienta eficaz para el bibliotecario y un “entretenimiento atractivo” para el usuario.

### 1.3.2 Objetivos específicos

Nuestra aplicación ha de cumplir los siguientes puntos:

**Seguridad:** La aplicación tendrá una entrada de usuario y contraseña almacenados en una base de datos. En todo momento se diferenciará y protegerá el acceso del administrador para proteger la biblioteca.

**Interfaz Gráfica:** La aplicación tendrá un aspecto moderno, atractivo y serio a la vez que colorido para fomentar su uso por los más jóvenes.

**Intuitiva:** La aplicación ha de ser sencilla y en pocos clics ha de cumplir sus funcionalidades.

## 1.4 Estrategia y planificación del proyecto

En nuestro proyecto desarrollaremos un producto nuevo, ya que vamos a crear una aplicación desde cero. Nuestra aplicación será una biblioteca virtual, existen muchas aplicaciones parecidas, pero no conocemos ninguna que integren un sistema de códigos QR para el conocimiento de usuarios y libros.

Por ello creemos que nuestra aplicación es innovadora y diferente al resto de aplicaciones de su ámbito. Sin duda, la funcionalidad de usar códigos QR nos diferenciará del resto de aplicaciones.

## 1.5 Metodología de trabajo

Hemos decidido usar la metodología Scrum ya que es una metodología de desarrollo de sistemas de información, es un tipo de metodología ágil y muy tolerante a cambios. Este tipo de metodología se suele usar en proyectos que tienen una entrega inmediata, ya que este es nuestro caso utilizaremos scrum como metodología de trabajo.

Realizaremos un único sprint el cual durará todo el desarrollo de la aplicación ya que la entrega está muy cercana y hay que trabajar rápido.

## 1.6 Estudio económico y presupuestario

Los datos que hemos usado para la factura son de primera mano, David se dedica al diseño gráfico y ha dado un valor en € a las horas de trabajo de un diseñador y Sergio ha contrarrestado varios presupuestos que ha obtenido de su puesto de trabajo y ha dado un valor en € a las horas de trabajo de un desarrollador.

El único material que necesitábamos para su desarrollo era un PC del cual ya disponíamos de otros proyectos.

Adjuntamos el presupuesto en los anexos.

## 2 Descripción del proyecto

### 2.1 Análisis de requisitos

Nuestra aplicación tendrá dos bloques de requisitos en función del rol del usuario, administrador o usuario estándar:

**Administradores:** Este usuario será el encargado de gestionar la biblioteca. Tendrá las siguientes funciones:

**Gestión de usuarios:** Crear, eliminar o modificar cualquier perfil de usuario.

**Gestión de libros:** Crear, eliminar o modificar cualquier libro, incluyendo sus datos y sus respectivos códigos QR.

**Publicar noticias:** Podrá publicar noticias tipo red social para mantener con fines lucrativos o informativos.

**Usuario estándar:** Este usuario es el que toma los servicios de la aplicación para interactuar con la biblioteca. Tendrá las siguientes funciones:

**Gestión de pedidos:** Alquilar libros y consultar información sobre sus préstamos.

**Leer publicaciones:** Aquí los usuarios pueden ponerse al día leyendo artículos y noticias relacionadas con la lectura y el centro.

La App es multiplataforma y está diseñada de una forma muy intuitiva para que cualquier usuario, con o sin conocimientos previos, pueda utilizarla. Nuestro público se divide en dos grupos:

**Bibliotecas:** todas las personas y entidades que deseen fomentar la lectura y dispongan de una biblioteca.

**Usuarios:** Todas las personas que deseen alquilar libros o recibir información relacionada con la lectura.

### 2.1.1 Requisitos funcionales

Los principales requisitos funcionales de nuestra aplicación se diferencia en función del rol de usuario que usa nuestra aplicación:

#### **Administrador:**

- Gestión de usuarios:
  - Crear perfiles de usuarios.
  - Editar perfiles de usuarios.
  - Eliminar perfiles de usuarios.
- Gestión de libros.
  - Añadir libro a la biblioteca.
  - Eliminar libro de la biblioteca.
  - Editar libro de la biblioteca.
- Publicación de noticias.
  - Crear noticias.
  - Editar noticias.
  - Eliminar noticias.

#### **Usuario estándar:**

- Gestión de alquiler.
  - Alquilar libro.
  - Notificar términos de devolución.
  - Devolver libro.
- Gestión de perfil.
  - Editar perfil.
- Gestión publicaciones.
  - Leer publicaciones.
  - Comentar publicaciones.

### **2.1.2 Requisitos no funcionales**

Los requisitos no funcionales de nuestra aplicación son:

- Contraseña de los usuarios encriptadas, la idea principal era usar un email y una contraseña que se encriptará al subir a la base de datos, pero al final como hemos utilizado Inicio de sesión con google ya nos automatiza este trabajo.
- Fácil de usar e intuitiva, queremos que nuestra aplicación sea autoexplicativa y sencilla para que cualquier usuario sin conocimientos previos pueda usarla.
- Agradable para la vista, creemos que una buena aplicación ha de ser agradable a la vista para prolongar su uso, por ello jugaremos con los colores para destacar datos.

### **2.2 Previsión de tareas de investigación**

Para desarrollar nuestra aplicación debemos investigar sobre los siguientes temas:

- Tecnología de desarrollo flutter.
- Generación de códigos QR, necesarios para libros y usuarios.
- Profundizar en el sistema de almacenamiento de firebase.

## 2.3 Tecnología

### 2.3.1 Comparativa de las tecnologías valoradas

#### Tecnologías de desarrollo:

##### Android:

- Utiliza el lenguaje java y kotlin, algo positivo ya que hemos trabajado mucho en este lenguaje.
- Diferencia el diseño gráfico del código interno de la aplicación.
- El desarrollo gráfico es complejo para realizar una aplicación moderna.
- Añadir librerías externas puede complicarse ya que muchas provocan conflictos entre sí.
- Únicamente se desarrolla una aplicación para el sistema operativo Android.

##### Flutter:

- Utiliza lenguaje Dart, no conocemos mucho de este lenguaje pero lo vemos como un lenguaje simple y rápido de aprender.
- El diseño gráfico de la aplicación y su funcionalidad se pueden programar desde un mismo lugar.
- El desarrollo gráfico es muy moderno y sencillo de utilizar. De una manera ágil se puede realizar una interfaz moderna y amigable.
- Es muy fácil añadir librerías externas.
- Con flutter podemos hacer una aplicación multiplataforma ya que se puede utilizar en ios, Android y web.

#### Tecnologías de base de datos:

##### SQLite:

- SQLite es una base de datos local, se puede llevar a online pero necesitaría demasiadas horas de trabajo.

##### Firebase:

- Firebase es una base de datos desarrollada en la nube diseñada para potenciar aplicaciones colaborativas en tiempo real.

### 2.3.2 Tecnologías escogidas

#### **Tecnologías de desarrollo:**

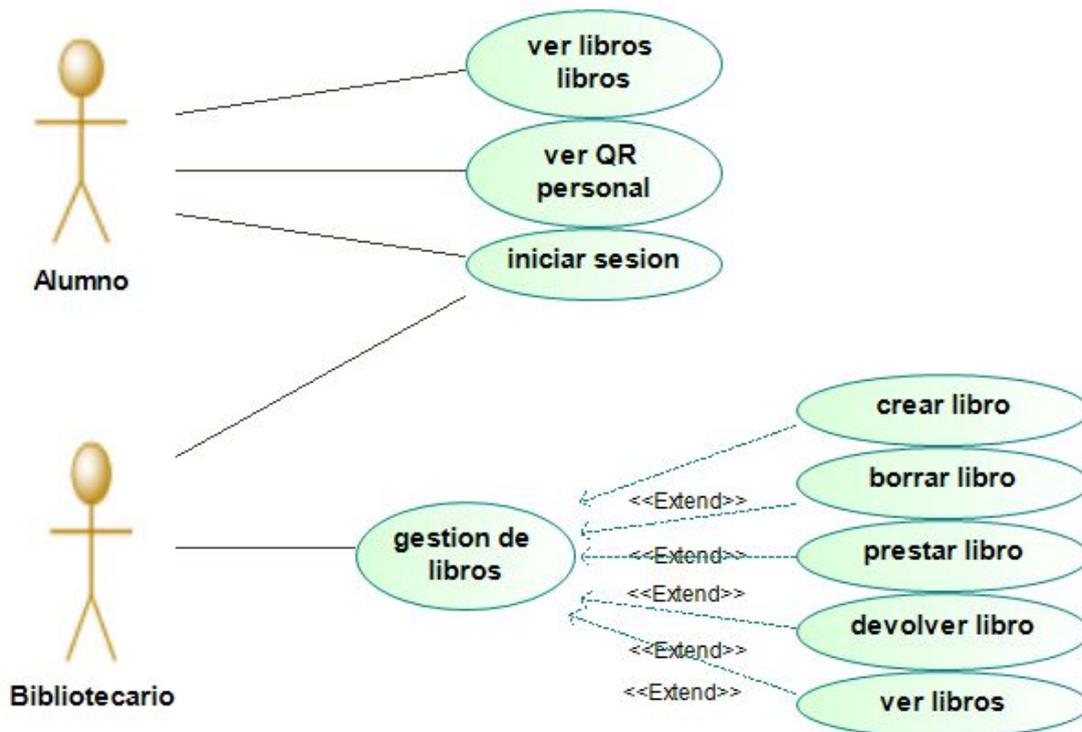
Finalmente hemos decidido decantarnos por flutter ya que es un lenguaje muy potente, multiplataforma y orientado al futuro. Aunque conocemos más a fondo android, y el lenguaje java, el proyecto nos parece una oportunidad ideal para aprender sobre esta nueva tecnología. Además sabemos que con flutter conseguiremos una aplicación más moderna e intuitiva.

#### **Tecnologías de bases de datos:**

Finalmente nos hemos decantado por utilizar una base de datos firebase, puesto que creemos que es mucho más potente y fácil de implementar dentro de nuestra aplicación. Además, con firebase nuestra aplicación estará 100% online, solo necesitas una cuenta de correo y acceso a internet.

## 2.4 Estructura del proyecto

Hemos decidido realizar un diagrama de casos de usos, ya que en flutter no nos ha hecho falta implementar las clases, porque hemos trabajado con la base de datos de firebase. Consideramos que con un diagrama de casos de uso nos ayuda resumir mejor nuestra aplicación.



El diagram de casos de usos que mostramos corresponde a la versión final de la aplicación.

Para realizar el diseño de nuestra aplicación nos hemos apoyado en un prototipo en Figma, con la idea inicial que tuvimos al empezar con el desarrollo:

Enlace a Figma:

<https://www.figma.com/file/EVk071lphVy5b3ox8W2LRz/qrbook?node-id=0%3A1>

Y también hemos diseñado una paleta de colores para crear una armonía visual:



## 2.5 Definición de las funcionalidades

En este punto vamos a explicar qué funcionalidades hemos implementado finalmente en nuestra aplicación.

Como hemos explicado anteriormente, nuestra aplicación se divide en dos grandes bloques dependiendo el tipo de usuario, ya sea bibliotecario (administrador) o usuario común. Al abrir la aplicación nos pide que iniciemos sesión con google, una vez iniciemos sesión si somos el bibliotecario tendremos un menú con las siguientes opciones:

**Bibliotecario/Administrador:** Este usuario será el encargado de gestionar la biblioteca. Tendrá las siguientes funciones:

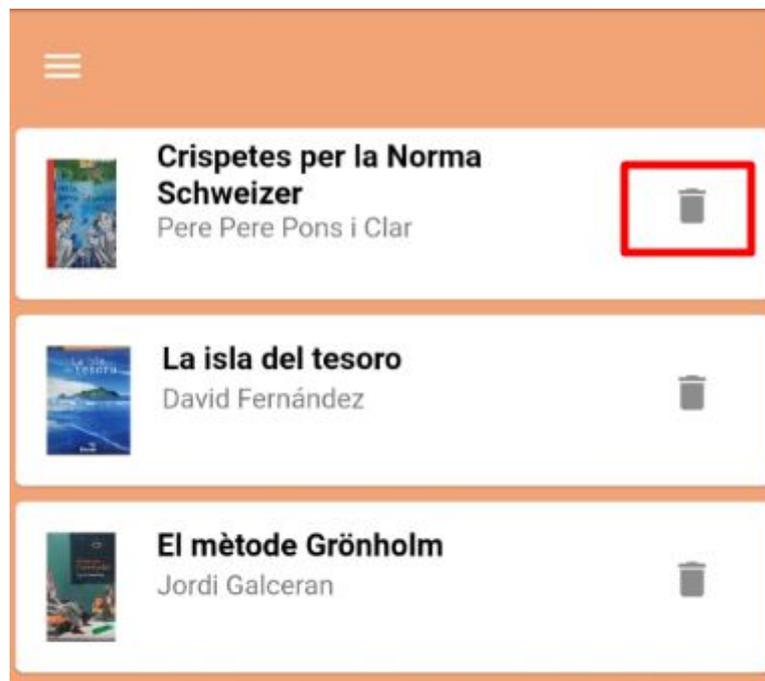
**Gestión de usuarios:** Tras debatirlo con uno de nuestros tutores de proyecto, Gerard, hemos decidido que por seguridad y por facilidad, trabajaremos con usuarios de google. Por lo tanto esta función ha sido eliminada del proyecto, ya que usando usuarios de google era innecesaria. En caso de tener que bloquear algún usuario, el administrador podrá hacerlo desde la consola de firebase:



**Gestión de libros:** Crear y eliminar.

**Crear:** Para crear un libro el bibliotecario dispondrá de una lista de códigos qr (formato imagen) que serán entregados junto a la aplicación. Dichos códigos deberán ser escaneados a la hora de añadir un libro junto a su determinada información. Este código QR guardará como un string que actuará como ID del libro.

**Eliminar:** Para eliminar un libro el bibliotecario sólo tendrá que clicar en la icono de la papelera que aparece junto al título de un libro en el listado de de su página inicio.



**Publicar noticias:** Este apartado no lo hemos implementado, pero nos parece interesante desarrollarlo próximamente.

Consistiría en añadir un pantalla al administrador en la que pudiera redactar noticias sobre el centro o curiosidades relacionadas con el ámbito de la lectura.

En cambio, si iniciamos sesión y no somos el bibliotecario nos saldrá un menu con las siguientes opciones:

**Usuario estándar:** Este usuario es el que tomas los servicios de la aplicación para interactuar con la biblioteca. Tendrá las siguientes funciones:

**Gestión de préstamos:** Consultar información sobre sus préstamos.

El alumno dispondrá de una pantalla en la que podrá visualizar una lista de los libros que tiene actualmente en préstamo. Para una futura versión, nos gustaría poder mostrar las fechas de devolución y poder añadir unas notificaciones para avisar al usuario de los términos de devolución.

**Visualizar libros:** Visualizar los libros de la biblioteca.

El alumno dispondrá de una pantalla dónde podrá visualizar todos los libros que no estén en préstamos. De esta manera puede consultar la disponibilidad de los libros para finalmente ir a buscar un libro a la biblioteca.

**Mostrar código QR:** Visualización del código qr del UID del alumno.



Este código qr sirve para establecer la relación de los préstamos.

## 3. Investigación.

### 3.1 Investigación sobre dart.

Para poder realizar nuestra aplicación con flutter hemos aprendido a trabajar con el lenguaje de programación dart. Al principio, nos costó acostumbrarnos a trabajar con su nomenclatura pero rápidamente descubrimos un lenguaje muy potente y sencillo de utilizar.

Un inconveniente que hemos encontrado es que algunas funciones muy concretas carecen de documentación ya que dart es un lenguaje muy nuevo. Por otro lado, firebase carece de una documentación de implementación en Dart “no está en la web oficial”.

```
body: StreamBuilder(
  stream: Firestore.instance.collection("books").snapshots(),
  builder: (context, snapshot) {
    if (snapshot.hasError)
      return Text('Error: ${snapshot.error}');
    switch(snapshot.connectionState){
      case ConnectionState.waiting:
        return const CircularProgressIndicator();
      default:
        return
          ListView.builder(
            itemCount: snapshot.data.documents.length,
            itemBuilder: (context, index){
              var books = snapshot.data.documents.elementAt(index);
              return Card(
                child: ListTile(
                  leading: books['imagenURL'] !=null ? Image.network(books['imagenURL']) : const Icon(Icons.image),
                  title: Text(books['titulo'],style: TextStyle(color: Colors.black,fontWeight: FontWeight.bold)),
                  subtitle: Text(books['autor']),
                  trailing: Column(
                    children: <Widget>[
                      Container(
                        child: IconButton(
                          icon: Icon(Icons.delete),
                          onPressed: () async {
                            await Firestore.instance.runTransaction((Transaction myTransaction) async {
                              await myTransaction.delete(snapshot.data.documents[index].reference);
                            });
                          }
                        );
                    ]
                  );
                );
            );
          );
    }
  });
```

En esta imagen mostramos un trozo de código de Dart que sería equivalente a un recyclerView de Java. Podemos apreciar que en Dart el código es mucha más corto y conciso.

### **3.2. Información sobre Firebase.**

Hemos tenido que investigar cómo implementar firebase en flutter, para ello nos hemos basado en una práctica de Gerard.

Una dificultad que hemos topado es entender cómo funciona una base de datos noSQL ya que nosotros estamos acostumbrado a trabajar con base de datos relacional. Esto quiere decir que en una base de datos que la base de datos utilizada es diferente el modelo clásico SGBDR(Sistema de Gestión de Base de Datos Relacionales), lo más destacado es que no usa lenguaje SQL como sistema principal de consultas.

Como positivo, una vez aprendido, hemos observado que trabajar con una base de datos en firebase es muy fácil y rápido, además funciona de maravilla realiza las tareas muy rápido.

### **3.3. Información sobre códigos QR.**

Como nuestra aplicación se basa principalmente en la lectura y generación de códigos QR hemos tenido que estudiar a fondo esta tecnología.

Una de las razones principales de usar Flutter y Dart es la facilidad que tienen para implementar la tecnología QR en su código.

Para ello hemos utilizado el widget QrCamera.

## 4 Conclusiones

### 4.1 Conclusiones generales del proyecto

Gracias a este proyecto hemos aprendido mucho sobre una nueva tecnología de programación como son Dart y flutter. Creemos que son la tecnología más potente con la que se puede programar aplicaciones actualmente.

También hemos profundizado en el uso de firebase y hemos podido comprobar que es una base de datos muy versatil, potente y veloz. Sólo con su versión gratuita, nuestra aplicación funciona online perfectamente.

### 4.2 Consecución de los objetivos

En el apartado 2.4 hemos profundizado en la explicación de las funcionalidades finales de la aplicación. A continuación mostramos un listado de las funciones que nos planteamos al principio explicando si se han realizado o no.

#### **Bibliotecario/Administrador:**

- **Inicio de sesión:** Funciona correctamente y además te dirige a la parte de la aplicación para el bibliotecario.
- **Gestión de usuarios:** Este apartado finalmente no se ha implementado en la aplicación porque trabajaremos con usuarios de google.
- **Gestión de libros.**
  - **Añadir libro a la biblioteca:** Funciona correctamente.
  - **Eliminar libro de la biblioteca:** Funciona correctamente.
  - **Editar libro de la biblioteca:** Esta función no ha sido implementada. Actualmente solo se pueden modificar datos desde la consola de firebase.
- **Publicación de noticias.** Este apartado finalmente no ha sido implementado en la aplicación, consideramos que sería una mejora para una futura versión.
- **Gestión de préstamo**
  - **Préstamo de libro:** Funciona correctamente
  - **Devolución de libro:** Funciona correctamente
  - **Visualización de libros prestados:** Funciona correctamente

**Usuario estándar:**

- **Inicio de sesión:** Funciona correctamente y además te dirige a la parte de la aplicación para el usuario
- **Gestión de alquiler:** Finalmente hemos decidido que este apartado solo lo tendrá disponible el bibliotecario.
- **Gestión de perfil:** Este apartado finalmente no se ha implementado en la aplicación porque trabajaremos con usuarios de google. Los datos de perfil corresponden a los datos de la cuenta de google. De esta manera el usuario puede editar sus datos desde su cuenta de google.
- **Gestión de publicaciones:** Este apartado finalmente no ha sido implementado en la aplicación, consideramos que sería una mejora para una futura versión.

### 4.3 Valoración de la metodología y planificación

Hemos realizado el trabajo de forma conjunta en todo momento. Nos hemos complementado asignando unos horarios por las tardes, ya que Sergio trabaja por las mañanas.

Para facilitar el desarrollo de la aplicación hemos utilizado un solo ordenador para evitar conflictos de versiones. David se ha conectado remotamente al ordenador de Sergio.

### 4.4 Visión de futuro

En este apartado listamos funcionalidades a desarrollar y mejorar:

**Inicio de sesión:** Nos gustaría añadir una preguntas sobre los gustos del usuario para hacerle recomendaciones de libros.

**Gestión de usuarios:** Nos gustaría poder implementar esta funcionalidad para poder modificar datos de los usuarios desde la propia aplicación.

**Gestión de libros.**

**Editar libro de la biblioteca:** Nos gustaría añadir esta funcionalidad, además de poder añadir datos adicionales sobre el libro.

**Publicación de noticias:** Nos gustaría añadir esta función en una futura versión.

**Visualización de libros prestados:** Nos gustaría poder añadir las fechas de devolución y el usuario que tiene prestado del libro.

**Usuario estándar:**

- **Inicio de sesión:** Funciona correctamente y además te dirige a la parte de la aplicación para el usuario
- **Gestión de alquiler:** Finalmente hemos decidido que este apartado solo lo tendrá disponible el bibliotecario.
- **Gestión de perfil:** Este apartado finalmente no se ha implementado en la aplicación porque trabajaremos con usuarios de google. Los datos de perfil corresponden a los datos de la cuenta de google. De esta manera el usuario puede editar sus datos desde su cuenta de google.
- **Gestión de publicaciones:** Este apartado finalmente no ha sido implementado en la aplicación, consideramos que sería una mejora para una futura versión.

## 5. Bibliografía

Documentación oficial de flutter:

<https://flutter.dev/docs>

Documentación oficial de firebase:

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

Consultas periódicas a:

<https://stackoverflow.com/>

Documentación oficial de:

[gerard@elpuig.xeill.net](mailto:gerard@elpuig.xeill.net)

## 6 Anexos

Adjuntamos un documento en pdf con un esquema completo de la aplicación.

<https://drive.google.com/file/d/1idNv8bESmIFwlyPJchkbq1Za4xrlfztU/view?usp=sharing>

Hemos creado un video demostrativo de nuestra aplicación que estará disponible para el día 31/05/2020 junto a la defensa de la aplicación.

Adjuntamos el código de la aplicación en github:

<https://github.com/splaza97/qrbook>

Adjuntamos la APK del proyecto:

<https://drive.google.com/file/d/1LTqatOqfkE4JcABTKTvrWIOrrXvIBcnA/view?usp=sharing>

Adjuntamos PDF demostración de las páginas QR.

[https://drive.google.com/file/d/1k2L-KKdlo-5KGZa5aFQjzVU\\_81EXpBe\\_/view?usp=sharing](https://drive.google.com/file/d/1k2L-KKdlo-5KGZa5aFQjzVU_81EXpBe_/view?usp=sharing)

Adjuntamos el presupuesto del desarrollo del proyecto en un excel:

[https://docs.google.com/spreadsheets/d/1sA7V3c7dOc\\_7VoalUEQNM0kHZE49WPAPgTIMZ7-BUf4/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1sA7V3c7dOc_7VoalUEQNM0kHZE49WPAPgTIMZ7-BUf4/edit?usp=sharing)