



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Aplicación móvil para la gestión de socios del club de  
baloncesto UPB Gandía

Trabajo Fin de Grado

Grado en Tecnologías Interactivas

AUTOR/A: Bordes Lloria, Sergi

Tutor/a: Heras Barberá, Stella María

CURSO ACADÉMICO: 2021/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA POLITÈCNICA SUPERIOR DE GANDIA

Tecnologías Interactivas

---

**“Aplicación móvil para la gestión de socios  
del club de baloncesto UPB Gandía”**

**TRABAJO FINAL DE GRADO**

**Autor/a:**

Sergi Bordes Lloria

**Tutor/a:**

Stella María Heras Barberá

**GANDÍA, 2022**

## Resumen

El presente TFG se basa en el desarrollo bajo demanda de una aplicación móvil que ayudará a un club de baloncesto (UPB Gandia) a gestionar sus socios y mantener un control de asistencia a los partidos. La aplicación tendrá como usuarios al personal del club y a sus abonados. Cada socio tendrá un pase virtual en su dispositivo móvil, y deberá presentarlo en el acceso de cada partido. El pase será leído por la aplicación y determinará, consultando con la base de datos, si el usuario está autorizado o no a acceder al evento. La aplicación también facilitará la gestión de socios (altas/bajas/modificaciones) y ofrecerá una amplia información de estadísticas en relación a los socios, como partidos más vistos, socios más concurrentes, etapas de la temporada donde más público ha habido, etc.

En cuanto a tecnologías utilizadas para implementar la aplicación, se usará Android Studio, en conexión con una base de datos Cloud Firestore. Todo el desarrollo se gestionará siguiendo la metodología Scrum.

**Palabras clave:** Android Studio, Cloud Firestore, Scrum, Git, Desarrollo de aplicaciones móviles, UX/UI, Baloncesto, Gestión de socios.

## Abstract

The actual TFG is based on the on-demand development of a mobile application that will help a basketball club (UPB Gandia) to manage its members and keep track of attendance at matches. The application will have club staff and their members as users. Each member will have a virtual pass on their mobile device, and must present it at the entrance of each match. The pass will be read by the application and it will determine, by consulting the database, whether or not the user is authorized to access the event. The application will also facilitate the management of members (registrations/deregistrations/modifications) and will offer a wide range of statistical information in relation to members, such as most watched matches, most concurrent members, stages of the season where there has been more public, etc.

As for the technologies used to implement the application, Android Studio will be used, in connection with a Cloud Firestore database. All development will be managed following the Scrum methodology.

**Key Words:** Android Studio, Cloud Firestore, Scrum, Git, Mobile application development, UX/UI, Basketball, Member management.

# Índice General

|   |    |
|---|----|
| <b>1. Introducción</b>  | 5  |
| 1.1 Motivación .....  | 5  |
| 1.2 Objetivos .....   | 6  |
| 1.3 Estructura de la memoria .....                            | 7  |
| <b>2. Marco teórico</b>                                       | 8  |
| 2.1 Estado del arte .....                                     | 8  |
| <b>3. Análisis de los requerimientos</b>                      | 10 |
| 3.1 Descripción general .....                                 | 10 |
| 3.2 Especificación de los requisitos .....                    | 10 |
| 3.3 Casos de uso .....  | 12 |
| <b>4. Diseño de la aplicación</b>                             | 16 |
| 4.1 Arquitectura de la aplicación .....                       | 16 |
| 4.2 Diseño de la interfaz y experiencia de usuario .....      | 22 |
| 4.3 Diseño de la base de datos .....                          | 22 |
| 4.4 Algoritmos .....  | 23 |
| <b>5. Implementación</b>                                      | 26 |
| 5.1 Metodología de trabajo y herramientas de desarrollo ..... | 26 |
| 5.2 Estructura de la aplicación .....                         | 27 |
| 5.3 Problemas de implementación resueltos .....               | 32 |
| <b>6. Manuales</b>  | 36 |
| 4.1 Manual de instalación .....                               | 36 |
| 4.2 Guía de uso .....   | 37 |
| <b>7. Evaluación</b>  | 44 |
| <b>8. Modelo comercial</b>                                    | 47 |
| <b>9. Conclusiones</b>  | 48 |

## Índice de figuras

- 1.1 Relación número socios por jornada
- 3.1 Casos de uso
- 4.1 Arquitectura de la aplicación
- 4.2 Estructura almacenamiento Cloud Storage
- 4.3 Diseño de la base de datos
- 4.4 Algoritmo 'acceso\_socios'
- 4.5 Algoritmo inicio de sesión
- 5.1 Estructura colección 'socios'
- 5.2 Estructura colección 'estadísticas'
- 5.3 Estructura archivos
- 5.4 Estructura *fragments*
- 5.5 Estructura Messaging Services
- 5.6 Estructura RecyclerViews
- 5.7 Estructura *front-end*
- 6.1 Instalar aplicación en dispositivo móvil
- 7.1 Frecuencia de uso de la aplicación
- 7.2 Facilidad de uso
- 7.3 Utilidad del manual
- 7.4 Funcionalidad
- 7.5 Funcionalidad del usuario regular
- 7.6 Funcionalidad para el usuario administrador

## Índice de tablas

- 2.1 Comparación entre aplicaciones similares
- 3.1 Caso de uso: Inicio de sesión
- 3.2 Caso de uso: Registro
- 3.3 Caso de uso: Menú inferior
- 3.4 Caso de uso: Información del club
- 3.5 Caso de uso: Perfil usuario 'regular'
- 3.6 Caso de uso: Estadísticas
- 3.7 Caso de uso: Alta/Baja socios
- 3.8 Caso de uso: Acceso público partidos
- 3.9 Caso de uso: Guardar estadísticas

# 1. INTRODUCCIÓN

## 1.1 Motivación

En los últimos años del club Units Pel Gandia Bàsquet, el número de socios, abonados, y espectadores ha ido aumentando considerablemente [Figura 1.1]. La población de Gandía se está aficionando cada vez más al baloncesto de la ciudad, al ser uno de los mejores equipos de la liga EBA, y estar en la lucha cada año por el ascenso a la categoría LEB Plata.

Este año, el equipo gandiense ha demostrado desde la Liga Valenciana (torneo de pretemporada), la cual ganaron, que había plantilla y cuerpo técnico suficiente para luchar por ese ascenso tan ansiado en los últimos años. Es por eso que el número de asistentes a los partidos ha aumentado exponencialmente en los últimos meses.

Número de socios por partido

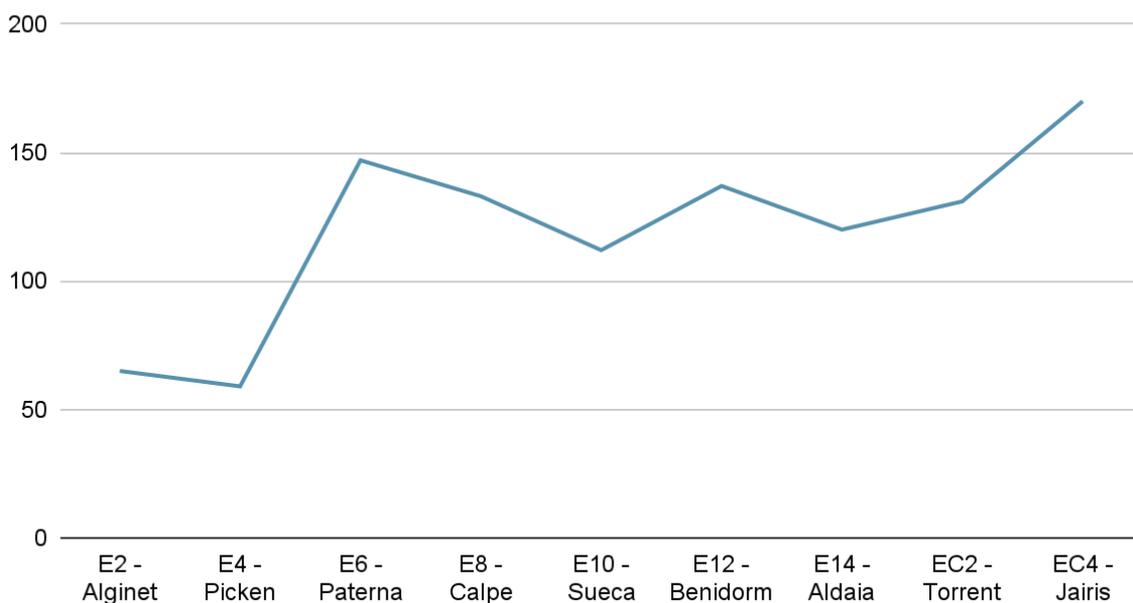


Figura 1.1. Relación número de socios por jornada.

Debido a este crecimiento de público en los encuentros jugados en casa, el club se planteó una mejor organización de socios y abonados mediante una aplicación móvil que controlase estos el aforo.

Es por eso que la motivación de este TFG es la de ofrecer una solución al club de baloncesto de la ciudad gandiense aprovechando el fácil acceso que tienen los usuarios a tecnologías como el teléfono móvil. La aplicación móvil desarrollada en este TFG, propone una solución para gestionar a los abonados del UPB Gandía, con sistemas para el control de acceso por parte de estos a los partidos jugados como locales. La aplicación permitirá dos modos, el de un usuario regular, donde se muestra información de los equipos de la base y el primer equipo, y el de administrador, donde este podrá obtener estadísticas de asistencia de los partidos y controlar el acceso a estos mismos.

## 1.2 Objetivos

El presente TFG tiene como objetivo el desarrollo de una aplicación móvil para facilitar el control de acceso por parte de socios y asistentes a los partidos jugados en Gandía y ofrecer información a los usuarios acerca del club, estadísticas de los jugadores en los partidos, y estadísticas (para los administradores) acerca de el número de asistentes a los partidos. Para alcanzar estos objetivos, primero han de cumplirse los siguientes subobjetivos:

- **Objetivo 1:** Analizar las aplicaciones ya existentes en el mercado con la misma intencionalidad que la desarrollada en este TFG.
- **Objetivo 2:** Planificar las funcionalidades que se requerirán en la aplicación a desarrollar.
- **Objetivo 3:** Diseñar la interfaz, priorizando una interfaz intuitiva y de confianza.
- **Objetivo 4:** Desarrollar la aplicación para Android.
- **Objetivo 5:** Evaluar la aplicación con usuarios reales.

### 1.3 Estructura de la memoria

El proyecto se organiza siguiendo la siguiente estructura:

En el **capítulo 1**, Introducción, explicamos de forma general el entorno del proyecto, realizamos una breve introducción identificando claramente los objetivos y relacionándolos con el entorno.

También se expone cuál ha sido la motivación principal y el porqué de la realización del proyecto.

El **capítulo 2**, Marco teórico, contiene el estado del arte, es decir, el análisis del entorno o ámbito sobre el que se va a desarrollar la aplicación, identificando los problemas de dicho entorno y realizando un estudio sobre las soluciones que ya existen en el mercado, tratando de encontrar sus problemas y comparándolas con la solución que se propone en este proyecto.

El **capítulo 3**, Análisis de los requerimientos, define en extensión y amplitud qué características queremos que tenga la aplicación.

Además, se muestran mockups de las diferentes pantallas que contienen las funcionalidades principales de la aplicación.

En el **capítulo 4**, Diseño de la aplicación, presentaremos la arquitectura de la aplicación con ayuda de esquemas, explicando los servicios que se han usado para la implementación de la aplicación.

Se explica las diferentes colecciones de la base de datos y se presentará el diseño de la misma.

Por último, se explicarán detalladamente y con sus respectivos diseños los algoritmos más importantes que se han desarrollado en el proyecto.

En el **capítulo 5**, Implementación, explicamos las metodologías ágiles de trabajo que hemos llevado a cabo durante el proyecto y las herramientas que nos han hecho falta para la implementación de éste.

Se describe la estructura de la aplicación, de carpetas y su contenido, usada para organizar el código fuente, tales como la base de datos, el *front-end*, *back-end*.

## 2. MARCO TEÓRICO

### 2.1 Estado del arte

El mercado de aplicaciones desarrolladas *ad hoc* para asociaciones culturales y deportivas ha experimentado un profundo auge en estos últimos años, donde a causa de la Covid-19, los clubes han demandado aplicaciones de este estilo para un mejor control de los socios y asistentes a los partidos, para evitar posibles contagios y cumplir las normas establecidas para frenar esta pandemia.

En este apartado, analizamos otras aplicaciones [Objetivo 1] similares o con la misma finalidad que se ofrecen en el mercado. y las comparamos con la nuestra.

La Tabla 2.1 muestra una comparativa de aplicaciones similares, atendiendo a propiedades como el precio, el objetivo, la plataforma, el lenguaje...

Después de buscar en internet aplicaciones semejantes a la que se está desarrollando durante este TFG, se ha concluido que las más semejantes son las listadas a continuación:

| Nombre                    | Objetivo                           | Precio   | Prueba Gratuita | Plataforma      |
|---------------------------|------------------------------------|----------|-----------------|-----------------|
| <b><u>PlayOff</u></b>     | Gestión de socios                  | 25€/mes  | Sí              | Web/iOS/Android |
| <b><u>Director 11</u></b> | Gestión deportiva                  |          | Sí              | Web/iOS/Android |
| <b><u>Cluber</u></b>      | Control acceso y aforo             | 349€/año | Sí              | Web/iOS/Android |
| <b><u>Clupik</u></b>      | Administración y comunicación      | 35€/mes  | Sí              | Web/iOS/Android |
| <b><u>UPBGandia</u></b>   | Gestión socios y usuarios (ad hoc) | -        | Sí              | Android         |

Tabla 2.1. Comparación entre aplicaciones similares

*PlayOff* es una aplicación tanto móvil como web para gestionar un club deportivo de manera bastante completa. Gestiona los datos personales y documentos de los socios, admite pagos online y domiciliación de recibos en

formato SEPA. La aplicación ofrece varios precios, desde 25€ hasta 65€/mes, con posibilidad de obtener una prueba gratuita.

*Director 11* interconecta todas las áreas de gestión de un club de fútbol. Administra jugadores, entrenamientos, partidos, lesiones, etc. Ofrece una demo gratuita para probar los servicios.

*Cluber* es una aplicación para control de accesos, límite de aforo y trazabilidad Covid. Ofrece control en tiempo real de los accesos y aforo de la instalación, desde uno o más teléfonos simultáneamente. Automatiza el cobro de las cuotas del club. Ofrece sus herramientas desde 349€ hasta 599€ al año dependiendo del número de deportistas que presente el club.

*Clupik* es una app destinada a automatizar publicaciones en la web y app y conecta las redes sociales del club, envía la notificación por email, por la aplicación móvil o a través de ambos canales, publica contenido generado por todos los usuarios. Los precios oscilan entre los 39 y 105€ al mes.

Muchas de estas aplicaciones ofrecen servicios similares, como la gestión de socios del club, o el control del acceso y del aforo a los encuentros deportivos. Otras, en cambio, ofrecen servicios que no se han implementado en esta aplicación, pero que pueden ser implementados posteriormente, como la admisión de pagos online o la inscripción de jugadores a través de la aplicación.

Alguna de estas aplicaciones ha sido desarrollada para cubrir ciertas necesidades a problemas que nos ha presentado la COVID-19, como la limitación del aforo en las instalaciones deportivas. Una de las principales razones de la implementación de la aplicación desarrollada en el presente TFG, es esta misma, el control de un aforo seguro para todos, intentando minimizar los contagios..

Finalmente, *UPBGandia* es una aplicación desarrollada *ad hoc* a petición del club, esto hace que se ajuste específicamente a sus requerimientos y necesidades. Ofrece una serie de servicios para el control y gestión de los socios del club. Desde la alta y baja de socios, como el control del aforo y público de los partidos, por parte del administrador, incluyendo también noticias, estadísticas y otros tipos de información acerca de los equipos del club, por parte de los usuarios regulares.

## 3. ANÁLISIS DE LOS REQUERIMIENTOS

### 3.1 Descripción general

Para el desarrollo de este proyecto se han tenido en cuenta y estudiado las diferentes opciones en cuanto a qué funcionalidades se van a implementar en la aplicación [Objetivo 2], qué base de datos se va a utilizar y que framework o tecnología de desarrollo utilizar.

A lo largo de esta sección, se expondrán las especificaciones de requisitos de la aplicación, definiendo cada una de las funcionalidades que serán implementadas en esta misma y presentadas mediante los casos de uso, ya que es una forma muy útil de mostrar los descriptores de las acciones de la aplicación.

Seguidamente, se explicará la estructura de la aplicación en cuanto a conexiones entre APIs o servicios y el esquema general de la arquitectura de la aplicación móvil y de la base de datos, haciendo hincapié en el porqué de la elección de diseño de cada una de las partes de esta.

Se ha realizado un estudio de la oferta de tecnologías disponibles en el mercado y se han observado las diferencias más relevantes entre estas. Una vez seleccionadas las tecnologías a utilizar, se ha expuesto cada una de las usadas en este proyecto.

Se han usado los servicios de Firebase, que son los que se han estudiado a lo largo del grado, y los cuales nos permiten desarrollar aplicaciones en diferentes plataformas sin necesidad de un servidor (“serverless”) y con muy buen rendimiento.

Finalmente, se han realizado mockups con Photoshop, herramienta la cual ha permitido diseñar cada una de las interfaces que se muestran en la app.

### 3.2 Especificación de los requisitos

En esta sección vamos a especificar los requisitos de la aplicación, que se trata de las características que se espera que tenga la aplicación en relación a sus funcionalidades. Estos requisitos han sido identificados a través de entrevistas con el equipo directivo del club, que es el cliente que ha encargado el desarrollo de esta aplicación.

1. Contará con un inicio de sesión para poder usar las funcionalidades personalizadas para cada rol de usuario (administrador o usuario regular - abonado).

2. El usuario podrá registrarse con sus datos o mediante su cuenta de Google.

3. Según el tipo de usuario que se registre, se le proporcionará acceso a la aplicación con las funcionalidades de “admin” o con las de “regular”.

4. La aplicación contendrá un menú inferior donde el usuario puede navegar libremente por todas los apartados y funcionalidades de esta.

5. [Regular] El usuario podrá ver información del club desde la página inicial.

6. [Regular] El usuario tendrá acceso a una página de perfil, donde podrá ver información personal y modificarla.

7. [Admin] El administrador podrá ver estadísticas de los encuentros disputados en casa, como el número de socios que han accedido, el total de aforo que hubo, etc.

8. [Admin] El administrador podrá dar de alta nuevos socios.

9. [Admin] El administrador podrá dar de baja socios.

10. [Admin] El administrador podrá realizar un control del acceso del público a los partidos como local, usando un escaneo de QR desde la aplicación.

11. [Admin] El administrador podrá guardar las estadísticas de la asistencia al partido en la base de datos.

12. [Regular y Admin] Ambos tipos de usuario podrán recibir notificaciones por diversas causas.

### 3.3 Casos de uso

A continuación, se presentará un esquema [Figura 3.1] y unas tablas [Tablas 3.1 - 3.9] donde se verá con más claridad los distintos casos de uso siguiendo su funcionalidad.

|                         |   |
|-------------------------|---|
| <b>Caso de uso</b>      | 1. Inicio de sesión   |
| <b>Descripción</b>      | Permite al usuario acceder a la aplicación y usar las funcionalidades dependiendo del rol               |
| <b>Pre - Condición</b>  | No haberse autenticado  |
| <b>Flujo Básico</b>     | Opción 1 → Introducir correo electrónico > Introducir contraseña<br>Opción 2 → Inicio sesión con Google |
| <b>Post - Condición</b> | Se almacena las credenciales y se tiene acceso a las funcionalidades                                    |

Tabla 3.1. Caso de uso: Inicio de sesión

|                         |  |
|-------------------------|--|
| <b>Caso de uso</b>      | 2. Registro  |
| <b>Descripción</b>      | Se recogen los datos necesarios para dar de alta al usuario en la aplicación               |
| <b>Pre - Condición</b>  | Ser un usuario nuevo   |
| <b>Flujo Básico</b>     | Introducir nombre, correo electrónico, número de teléfono y contraseña                     |
| <b>Post - Condición</b> | Se da de alta el usuario en la base de datos<br>Se devuelve a la vista de inicio de sesión |

Tabla 3.2. Caso de uso: Registro

|                         |  |
|-------------------------|--|
| <b>Caso de uso</b>      | 3. Menú inferior   |
| <b>Descripción</b>      | Se muestran tabs en la parte inferior de la pantalla que permiten navegar entre las diferentes páginas y funcionalidades |
| <b>Pre - Condición</b>  | Inicio de sesión   |
| <b>Flujo Básico</b>     | Se inicia sesión > Acceso al menú  |
| <b>Post - Condición</b> | Muestra la vista predeterminada de la aplicación con dicho menú  |

Tabla 3.3. Caso de uso: Menú inferior

|                         |   |
|-------------------------|---|
| <b>Caso de uso</b>      | 4. Información del club   |
| <b>Descripción</b>      | En la página principal del usuario 'regular', podrá ver información del club como noticias, plantilla, etc. |
| <b>Pre - Condición</b>  | Iniciar sesión como usuario 'regular'   |
| <b>Flujo Básico</b>     | Inicio sesión como usuario 'regular'  |
| <b>Post - Condición</b> | Acceso a la página principal de la aplicación   |

Tabla 3.4. Caso de uso: Información del club

|                         |  |
|-------------------------|--|
| <b>Caso de uso</b>      | 5. Perfil usuario 'regular'  |
| <b>Descripción</b>      | El usuario podrá ver su información básica como el nombre, el correo, o la foto de perfil<br>Podrá modificar su información personal |
| <b>Pre - Condición</b>  | Iniciar sesión como 'regular'  |
| <b>Flujo Básico</b>     | Iniciar sesión como 'regular' > Navegar por el menú pulsando en el icono de perfil   |
| <b>Post - Condición</b> | Se visualiza la página de perfil y la información básica del usuario   |

Tabla 3.5. Caso de uso: Perfil usuario 'regular'

|                         |   |
|-------------------------|---|
| <b>Caso de uso</b>      | 6. Estadísticas   |
| <b>Descripción</b>      | El usuario administrador podrá obtener información en forma de gráficas y estadísticas sobre la asistencia a los partidos disputados como local |
| <b>Pre - Condición</b>  | Iniciar sesión como 'admin'   |
| <b>Flujo Básico</b>     | Inicio de sesión como 'admin' > Visualizar estadísticas en página principal   |
| <b>Post - Condición</b> | Se muestran gráficas y valores estadísticos de la asistencia a los partidos   |

Tabla 3.6. Caso de uso: Estadísticas

|                         |  |
|-------------------------|--|
| <b>Caso de uso</b>      | 7. Alta/Baja socios  |
| <b>Descripción</b>      | Permite dar de alta nuevos socios para el club desde la aplicación   |
| <b>Pre - Condición</b>  | Iniciar sesión como 'admin'  |
| <b>Flujo Básico</b>     | Navegar hasta página de socios > Pulsar botón 'añadir socio'<br>Navegar hasta página de socios > Pulsar botón 'borrar socio' |
| <b>Post - Condición</b> | Se da de alta o se elimina el socio  |

Tabla 3.7. Caso de uso: Alta/Baja Socios

|                         |   |
|-------------------------|---|
| <b>Caso de uso</b>      | 8. Acceso público a partidos  |
| <b>Descripción</b>      | Se controlará el acceso a los partidos mediante el escaneo de un código QR desde la aplicación  |
| <b>Pre - Condición</b>  | Iniciar sesión como 'admin'   |
| <b>Flujo Básico</b>     | Inicio sesión como 'admin' > Navegar hasta página 'game' > pulsar 'Entra socio'<br>Inicio sesión como 'admin' > Navegar hasta página 'game' > pulsar 'Sale socio' |
| <b>Post - Condición</b> | Autorizar el acceso a un socio si la aplicación lo indica<br>Denegar el acceso a un socio si la aplicación lo indica  |

Tabla 3.8. Caso de uso: Acceso público a partidos

|                         |   |
|-------------------------|---|
| <b>Caso de uso</b>      | 9. Guardar estadísticas   |
| <b>Descripción</b>      | Se almacenarán las estadísticas después de cada partido, con valores como aforo total o el número de socios que han accedido al partido |
| <b>Pre - Condición</b>  | Iniciar sesión como 'admin'   |
| <b>Flujo Básico</b>     | Inicio sesión como 'admin' > Navegar hasta página 'game' > pulsar 'Guardar estadísticas'  |
| <b>Post - Condición</b> | Las estadísticas se almacenarán en la base de datos y serán visibles en la página principal del 'admin'                                 |

Tabla 3.9. Caso de uso: Guardar estadísticas

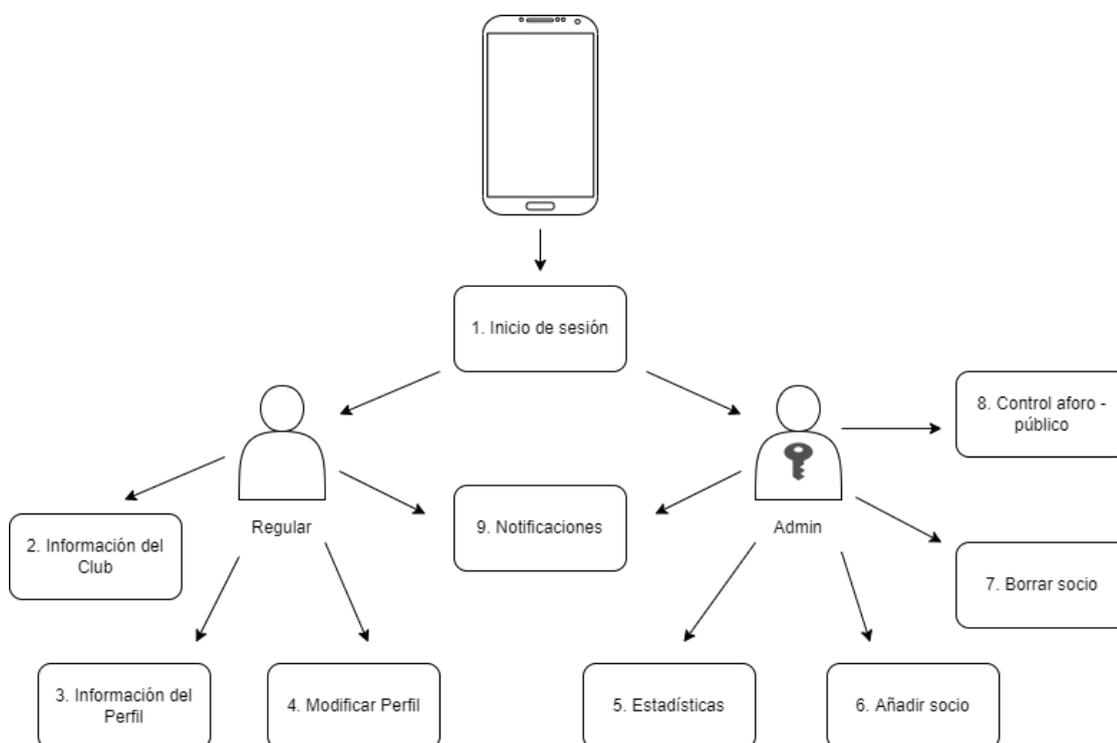


Figura 3.1. Casos de uso

## 4. DISEÑO DE LA APLICACIÓN

### 4.1 Arquitectura de la aplicación

A continuación, se puede observar en el siguiente esquema [Figura 4.1] la arquitectura que presenta la aplicación.

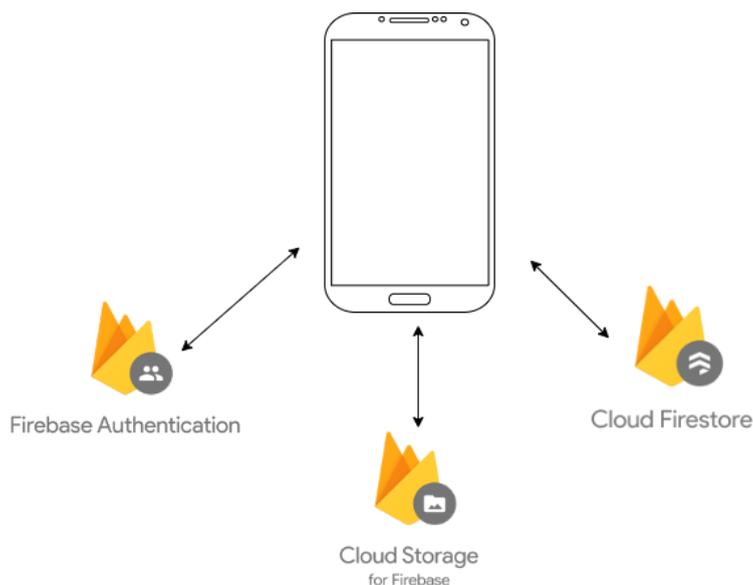


Figura 4.1. Arquitectura de la aplicación

La aplicación realizada en este proyecto usará los servicios de Firebase, usados habitualmente durante los estudios del grado y que se adaptan perfectamente a las necesidades de la aplicación.

Para la autenticación de usuarios y la gestión de credenciales se ha usado el servicio de Firebase Authentication. Cuando el usuario se dirige a la página de inicio de sesión e introduce sus credenciales, la aplicación hace una consulta en tiempo real en Firebase y nos indica si el usuario se ha registrado anteriormente o no, permitiendo a este acceder a las funcionalidades de la aplicación o negándole el acceso. Gracias a esta librería de Firebase, también hemos podido implementar el registro y el inicio de sesión a través de la cuenta personal de Google del usuario, facilitando el acceso a los usuarios que no quieren crearse una cuenta nueva, sino usar una que ya tienen.

Para el almacenamiento de archivos se ha usado Cloud Storage, un servicio de almacenamiento de objetos potente, simple y rentable construido para el escalamiento de Google donde se almacena, principalmente, imágenes en .jpg [Figura 4.2] que serán usadas en las imágenes de perfil de los usuarios regulares.

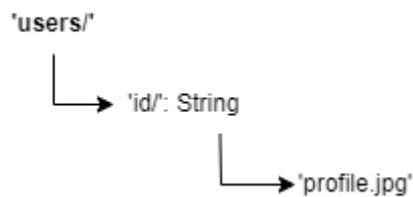


Figura 4.2. Estructura almacenamiento Cloud Storage

Finalmente, como base de datos de la aplicación, se ha usado Cloud Firestore, una base de datos NoSQL flexible, escalable y en la nube a fin de almacenar y sincronizar datos para el desarrollo tanto del lado del cliente como del servidor. Mantiene los datos sincronizados entre las aplicaciones a través de objetos de escucha en tiempo real. Se ha usado para almacenar colecciones de los diferentes usuarios de la aplicación, los socios del club, las estadísticas de los partidos, etc.

## 4.2 Diseño de la interfaz y experiencia de usuario

A continuación, se realizará un recorrido a través de la aplicación definiendo en extensión y amplitud cada una de las características y funcionalidades de esta, algunas de las cuales (calendario de eventos) quedan previstas para ser desarrolladas como trabajo futuro a este proyecto. Para ello se presentan croquis esquemáticos [Objetivo 3] previos a la implementación para explicar y definir dichas pantallas.



#### 4.2.1 Inicio de sesión

En el apartado de inicio de sesión, el usuario tendrá la posibilidad de autenticarse para poder acceder a las funcionalidades de la aplicación.

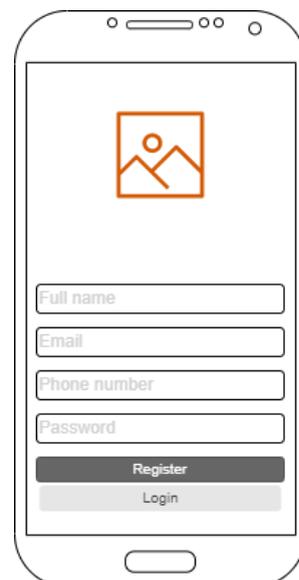
Deberá introducir el correo y la contraseña que haya establecido anteriormente para acceder a la app, o también puede acceder a la aplicación a través de su cuenta de Google, pulsando en el icono e iniciando sesión a través de Google.

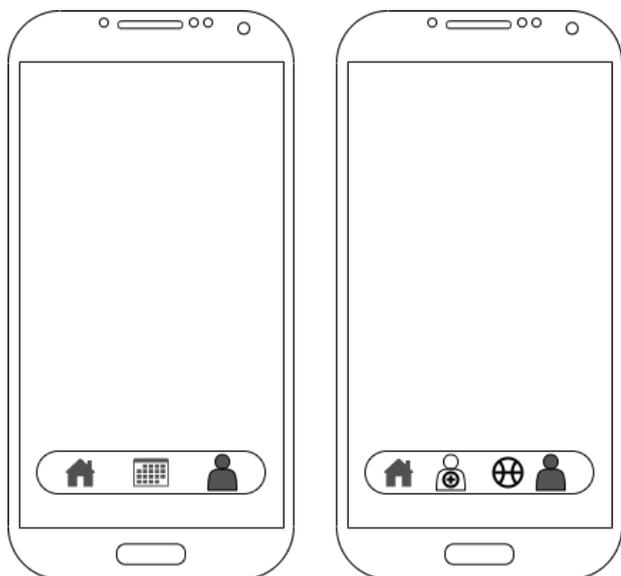
Si el usuario no tiene ninguna cuenta aún, deberá pulsar sobre 'Registrar' y seguir los pasos.

#### 4.2.2 Registro de usuario

Cuando el usuario quiere crear una cuenta en nuestra aplicación, deberá acceder a esta pantalla e introducir los datos que se le piden: Nombre completo, correo electrónico, número de teléfono, y la contraseña.

Una vez el usuario esté registrado, se le redireccionará a la pantalla de inicio de sesión para que acceda con las credenciales del usuario que acaba de crear.





### 4.2.3 Menú inferior

La aplicación presenta dos tipos de menú inferior, el menú para usuarios 'regular' (izquierda), y el menú para usuarios 'admin' (derecha).

El usuario 'regular' podrá navegar por las pantallas: home, eventos, perfil.

Mientras que el 'admin' podrá navegar por las pantallas: home, socios, partido, panel de administrador.

### 4.2.4 [Regular] Página principal 'home'.

La página principal del usuario regular tendrá la información de la página web del club, para que el usuario pueda navegar por esta y acceder a diferentes lugares de la web, como visitar la plantilla del primer equipo, sacarse el pase de socio, visitar la galería de fotos, etc.

También podrá acceder a las crónicas y estadísticas individuales de los partidos.



### 4.2.5 [Regular] Página eventos.

El usuario podrá ver los próximos eventos que organice el club en esta sección, con información como el título, la descripción, y la fecha.

Tendrá dos posibilidades de observar la información, en forma de lista, pulsando sobre el icono de la lista, o en forma de calendario, pulsando sobre el icono del calendario.



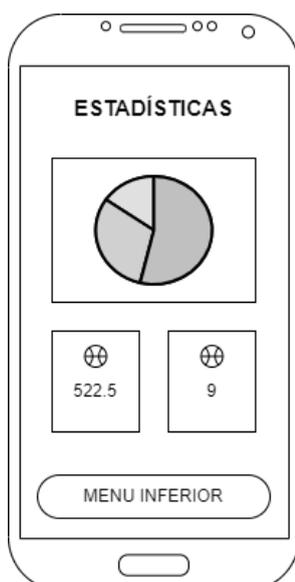
#### 4.2.6 [Regular] Página perfil.

El usuario regular podrá acceder a una sección de perfil para ver diferente información acerca de su cuenta como la foto de perfil, el nombre, el correo electrónico, o diferente información estadística sobre su asistencia a los partidos.

El usuario también podrá modificar esta información. Tiene la posibilidad de cambiar la foto de perfil por alguna de su galería así como su nombre y teléfono.



#### 4.2.7 [Admin] Página estadísticas 'home'.



El administrador tendrá como página principal una pantalla donde podrá ver información estadística acerca de la asistencia a los partidos que se disputen durante la temporada.

Podrá ver en un *PieChart* la comparación de asistencia de cada partido de forma gráfica y visual, con el número de asistentes totales a los encuentros.

También podrá ver la media de asistentes socios y la media de asistentes que han comprado la entrada en taquilla (no socios).

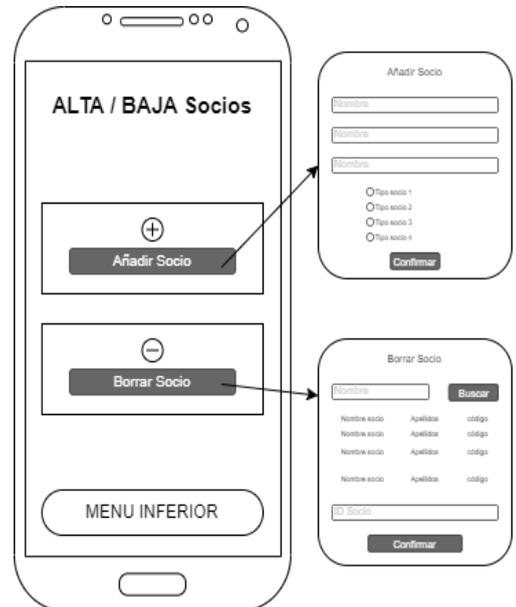
#### 4.2.8 [Admin] Página socios.

En esta sección el admin tendrá la capacidad de dar de alta o baja a socios pulsando en los respectivos botones.

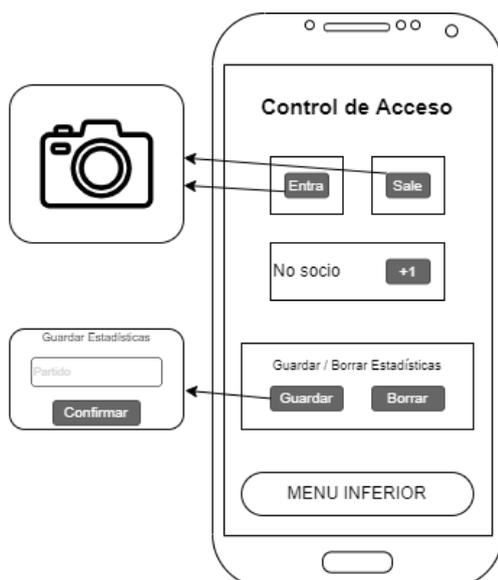
Al pulsar sobre 'Añadir socio' aparecerá un popup inferior con el formulario que se indica. Para añadir al socio a la base de datos, habrá que indicar información básica sobre la persona y el tipo de pase (Individual, Familiar, etc.) con el cual se va a dar de alta.

'Borrar socio' también hace aparecer un popup inferior con una barra de buscador, donde tras indicar el nombre, se listan los socios con ese nombre, y al mantener presionado sobre un socio concreto se podrá borrar este.

Otra manera de eliminar los socios es indicando directamente el ID del socio y pulsando sobre 'Confirmar'.



#### 4.2.9 [Admin] Página Control de acceso.



Esta es la página más importante de la aplicación, que constituye la principal demanda del club, una aplicación para el control de acceso a los partidos.

Con los botones 'Entra' y 'Sale', se accederá a la cámara para el escaneo de los códigos QR proporcionados por el club que presentarán los socios. Al escanear, la pantalla se volverá verde o roja, indicando así si el socio puede pasar o no.

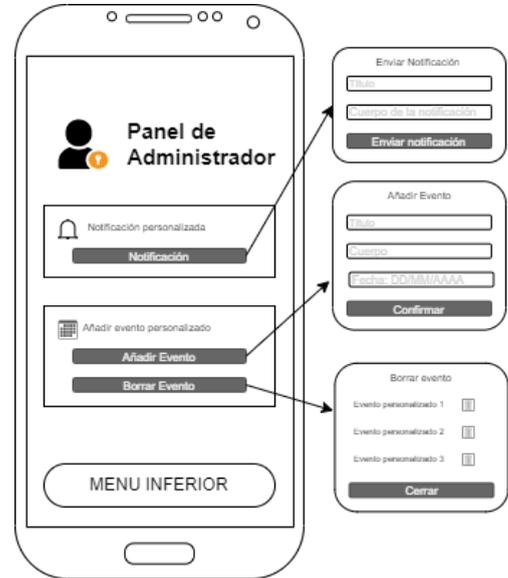
El botón '+1' sirve para registrar el acceso de los espectadores no socios que han comprado la entrada en taquilla, para que posteriormente, con el botón 'Guardar' se registre todo el acceso al partido en la base de datos. Se podrá eliminar las estadísticas del partido actual, guardadas hasta el momento, pulsando el botón 'Borrar' (previo a un popup de confirmación).

#### 4.2.10 [Admin] Página Control de acceso.

En el panel del administrador, este será capaz de realizar ciertas funciones que vincularán a todos aquellos usuarios que posean la aplicación.

Podrá crear una notificación para todos los usuarios de la aplicación, llenando un pequeño formulario con el título y cuerpo de la notificación, una manera muy rápida y eficaz de enviar recordatorios.

También podrá crear eventos que aparecerán en la página de eventos de la aplicación, los cuales podrán ver los usuarios regulares en forma de lista o calendario.



### 4.3 Diseño base de datos

La aplicación presenta diferentes colecciones [Figura 4.3] en la base de datos:

- 'socios': Se almacenan los socios que el administrador ha dado de alta, con información como el código de socio, o el tipo de socio.
- 'users': Donde se almacenan todos los usuarios que se han dado de alta en la aplicación, con sus respectivas credenciales e información básica.
- 'estadísticas': Información como el número de socios que han accedido por partido o el número de asistentes totales al encuentro.
- 'eventos': Contiene una lista detallada de los próximos eventos que sucederán en el club.

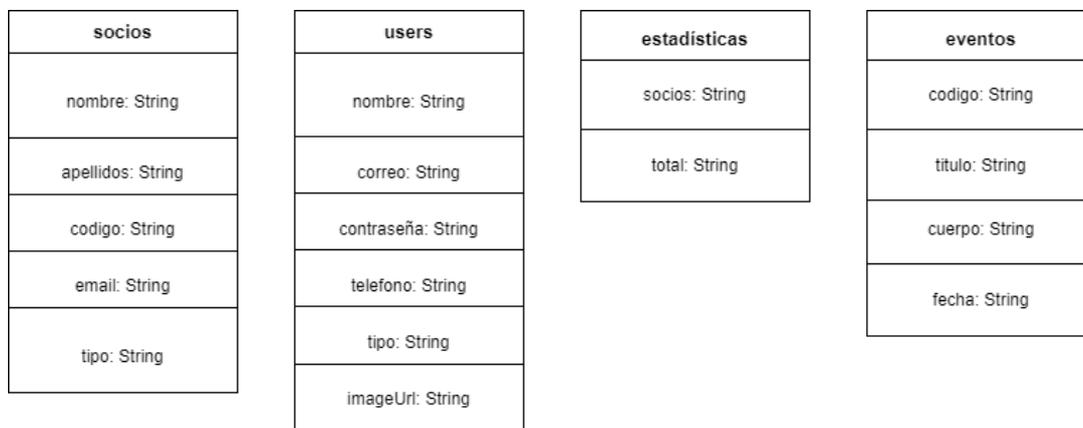


Figura 4.3. Diseño base de datos

## 4.4 Algoritmos

Uno de los algoritmos más importantes que presenta la aplicación, y función por la cual la aplicación fue solicitada por el club, es la del control de acceso a los partidos.

El algoritmo de la función funciona de la siguiente forma: al pulsar un botón de la página de 'Control de acceso', el teléfono móvil abre su cámara y busca códigos QR que escanear. Una vez capta este código QR, obtiene el número de ID que presenta el QR y consulta en la base de datos, buscando un documento en la colección 'socios' que presente el siguiente ID: "Socio\_" + "Código QR". Si después de revisar la base de datos, la aplicación no encuentra ninguna coincidencia con ningún documento de la colección, la pantalla se volverá roja, indicando así al administrador y al usuario que ha presentado el código QR que no puede acceder al partido.

Si el ID del documento de la colección de la base de datos coincide con el código del QR, significa que el socio sí que está dado de alta, y la aplicación iniciará una nueva búsqueda en otra colección, 'socios\_partido', colección en la cual están los socios que ya han presentado el carnet y por lo tanto ya han accedido al partido. Si también hay coincidencia en esta colección, la pantalla se volverá roja, negando así el acceso al partido al socio, ya que ha intentado reusar el carnet.

Finalmente, si tras buscar en la colección 'socios\_partido', no se ha encontrado ninguna coincidencia, significa que el socio está presentando el carnet por primera vez en ese partido, y por lo tanto, la pantalla de la aplicación se volverá verde, indicando al administrador y al socio que puede acceder al encuentro.

En la Figura 4.4 el diseño del algoritmo de la función 'acceso socios':

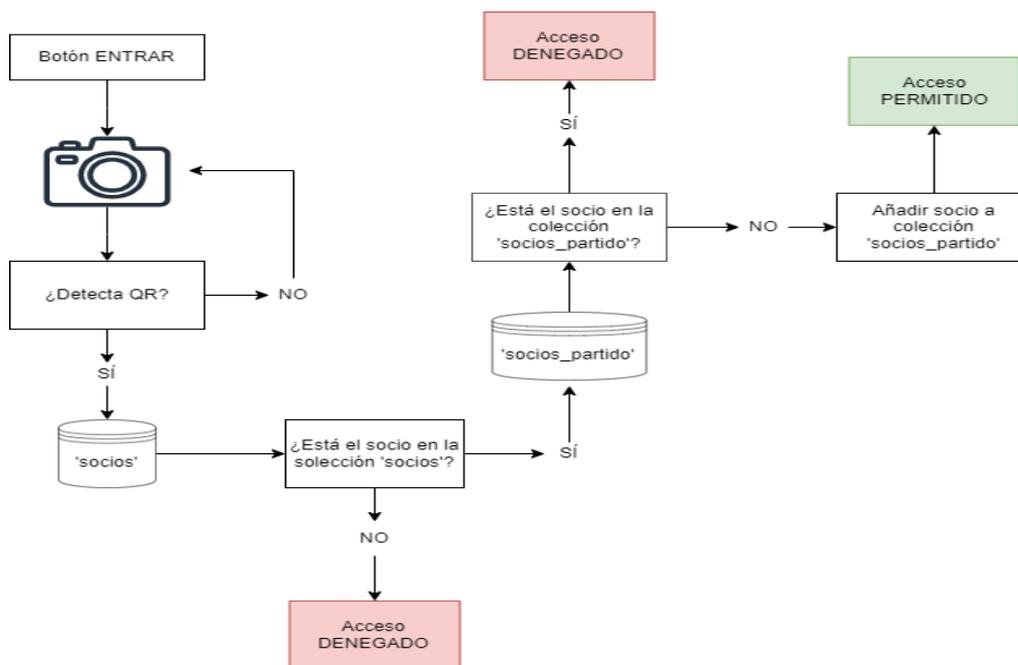


Figura 4.4. Algoritmo 'acceso socios'.

Otro algoritmo importante de la aplicación sería el inicio de sesión, ya que presenta un condicionante dependiendo del tipo de usuario que se registre, es decir, dependiendo de si el usuario es 'regular' o 'admin'.

La aplicación por defecto clasifica a los nuevos usuarios cómo 'regular', es por esto que solo los administradores que tienen acceso a la base de datos son los únicos que pueden decidir qué cuentas serán 'admin'.

Para acceder a la aplicación, el usuario deberá rellenar el formulario de acceso introduciendo sus credenciales, y posteriormente pulsar el botón 'Iniciar sesión'. La aplicación realizará una búsqueda en Firebase Authentication para ver si el usuario se ha registrado anteriormente, y si es así, la aplicación buscará en la colección 'users' y apuntará al parámetro 'tipo', el cual indica el rol del usuario.

Si el tipo es 'regular', la aplicación redirigirá al usuario a las funcionalidades para 'regular', por otro lado, si el tipo es 'admin', el usuario será redirigido a las funcionalidades del administrador.

La figura 4.5 muestra el diseño del inicio de sesión de la aplicación:

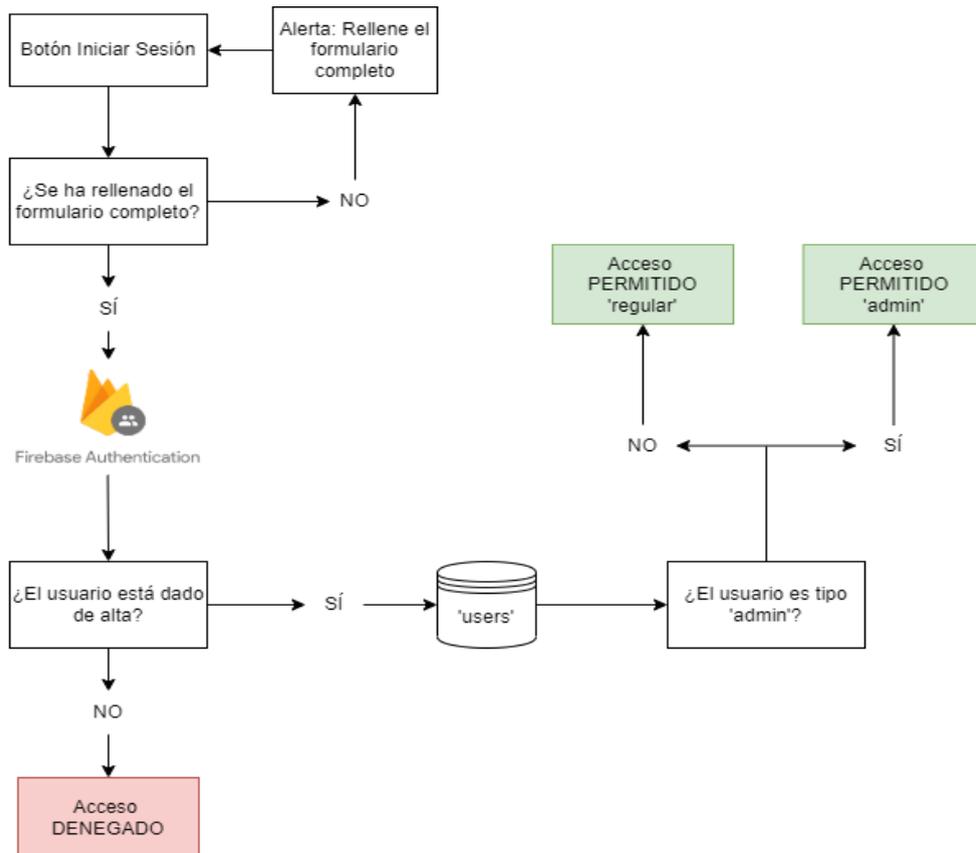


Figura 4.5. Algoritmo inicio de sesión.

# 5. IMPLEMENTACIÓN

## 5.1 Metodología de trabajo y herramientas de desarrollo

Para desarrollar el presente TFG se han seguido dos metodologías, una para definir y realizar las fases de desarrollo de la aplicación móvil (metodología de desarrollo de las etapas del ciclo de vida del desarrollo de una aplicación móvil [Desarrollo de aplicaciones móviles]) y otra para la gestión del trabajo diario (metodología ágil de gestión de proyectos), utilizada mientras se ha realizado la aplicación y la memoria.

### **Etapas del ciclo de vida del desarrollo de una aplicación móvil**

- Inicio: La aplicación se inicia con una idea, perfeccionada en una base sólida y analizando el mercado.
- Diseño: Definir una experiencia de usuario (UX) mediante mockups y definir funcionalidades finales de la aplicación.
- Desarrollo: Una vez concluidas las dos fases anteriores, se pasa a la creación real de la aplicación.
- Estabilización: Cuando la aplicación ya ha avanzado lo suficiente, se empieza a probar la aplicación mediante tests para corregir errores.
- Distribución: Tras estabilizar la aplicación y corregir todos los posibles errores, se publica la aplicación.

### **Metodología ágil de gestión de proyectos**

La metodología seguida para la gestión del proyecto y del trabajo diario, ha sido la de *Scrum*, una metodología seguida durante todo el grado, la cual nos ayuda a tener una mejor organización del trabajo, eficiente y ordenada.

Esta metodología está pensada para proyectos en equipo, pero también puede ser usada en individuales.

Con *Scrum*, se ejecutan ciclos temporales donde se debe presentar un producto mínimo viable, normalmente cada dos semanas.

La metodología Scrum [*Metodología scrum*] se desarrolla a través de las siguientes fases:

- Planificación de cada Sprint: Se proponen unos objetivos para las siguientes semanas.
- Etapa de desarrollo: Realización de las tareas propuestas en el Sprint dentro del plazo de tiempo marcado.
- Revisión del Sprint: Evaluación de los resultados obtenidos al final del sprint.
- Retroalimentación: Se entrega el producto para plantear el siguiente Sprint en base al feedback obtenido.

### **Herramientas de desarrollo**

La herramienta principal de desarrollo que se ha utilizado en el proyecto para la implementación de una aplicación móvil ha sido Android Studio [Objetivo 4].

Se eligió esta herramienta ya que la propuesta de trabajo del club era la creación de una aplicación para el control de acceso a los partidos disputados en casa, y los teléfonos móvil que posee el club para realizar este control utilizan Android como sistema operativo.

Además, es la principal herramienta usada durante el grado para el desarrollo de aplicaciones móviles y con la que más familiarizado se está, ampliando así el abanico de posibilidades de las funcionalidades a implementar.

## **5.2 Estructura de la aplicación**

La estructura de la aplicación se divide principalmente en tres apartados, la base de datos, el *front-end*, y el *back-end*.

Para la **base de datos** se han usado los servicios de Cloud Firestore de Firebase, y se ha estructurado de la siguiente manera:

Se han usado nombres que definan a las colecciones y sean identificables fácilmente, así como ‘socios’, ‘estadísticas’ o ‘eventos’. Dentro de cada una de estas colecciones se ha usado un ID generado aleatoriamente, excepto para la colección de ‘estadísticas’, que se usa el nombre del partido como ID. Además, los ID de los socios empiezan por “Socio\_”, seguido de un número aleatorio.

Para el ID de los documentos en la colección ‘users’, se ha usado los ID generados por Firebase Authentication con los que almacena a los usuarios que se dan de alta en la aplicación.

La figura 5.1 y figura 5.2 un ejemplo de la estructura de la base de datos, con las colecciones ‘estadísticas’ y ‘socios’:

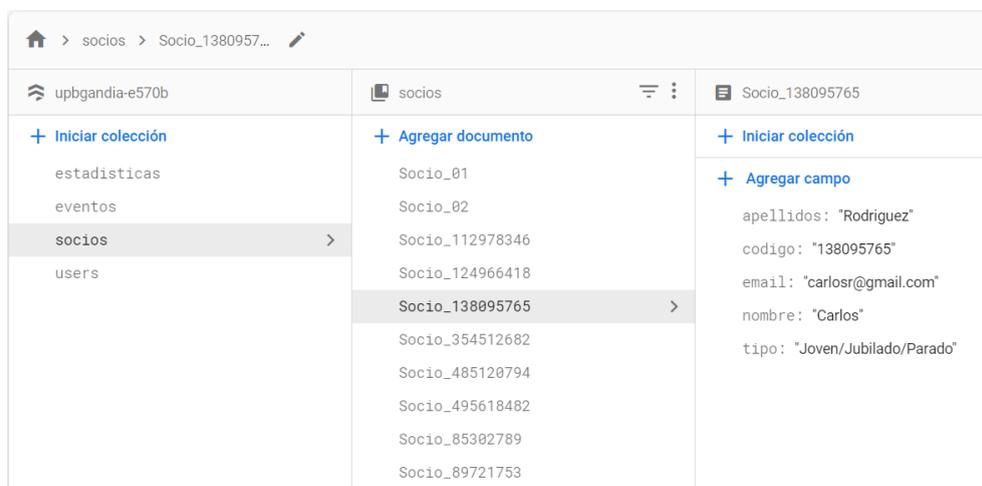


Figura 5.1. Estructura colección ‘socios’

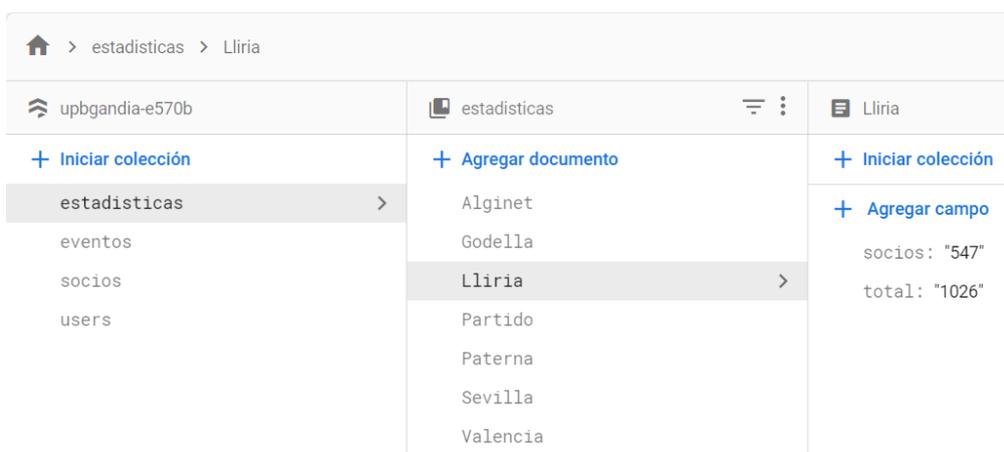


Figura 5.2. Estructura colección ‘estadísticas’

En segundo lugar, el **back-end** de la aplicación ha sido estructurado de la siguiente forma:

Dentro de la carpeta 'java' y seguidamente del paquete de la aplicación 'upbgandia' se han creado las principales clases de la aplicación, y las que conforman la base funcional de esta.

- La clase RegisterActivity, la cual contiene todas las funciones para el correcto registro de los usuarios en la aplicación.
- La clase AuthActivity, mantiene el correcto funcionamiento de la página de autenticación de usuarios.
- La clase HomeActivity, donde todos los usuarios que inician sesión son redirigidos, para que en esta clase, vuelvan a ser redirigidos donde corresponda dependiendo de su rol en la aplicación (regular o admin).
- La clase RegularActivity, contiene todas las funciones que puede realizar el usuario regular, desde visualizar información de la web del club hasta modificar su imagen de perfil, ver los próximos eventos, etc.
- Y finalmente, la clase AdminActivity, donde recae la mayoría de las funcionalidades de la aplicación y donde el usuario administrador podrá controlar a los socios, realizar controles de acceso, enviar notificaciones, añadir eventos, visualizar estadísticas, etc.

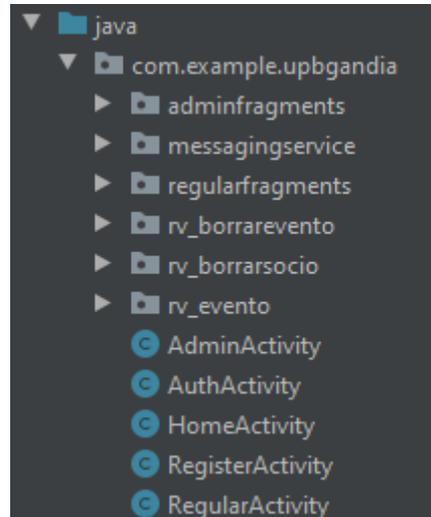


Figura 5.3. Estructura archivos

Dentro de esta misma carpeta ('java/upbgandia/') y como se ve en la figura 5.3, se han creado varias carpetas para organizar mejor las diferentes clases y funciones de la aplicación.

Empezando por los *fragments*, los cuales han sido creados para evitar tiempos de espera largos entre página y página para una mejor navegación por dentro de las páginas principales de la aplicación (regular y admin), se clasifican en dos carpetas: *adminfragments* (para el cambio de pestañas de la parte del administrador) y *regularfragments* (para el cambio de pestañas de la parte del usuario regular).

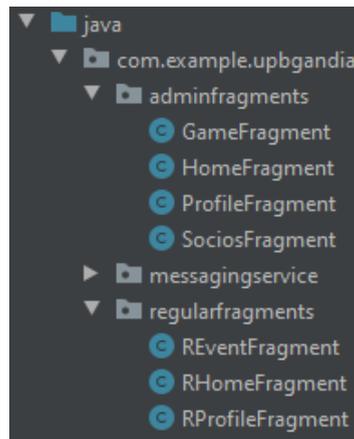


Figura 5.4. Estructura fragments

Siguiendo con los servicios de mensajería para el envío de notificaciones a los usuarios de la aplicación, se ha creado una carpeta llamada 'messagingservice', donde se almacena la clase 'MyFirebaseMessagingService', donde se hace uso de la API de Firebase Cloud Messaging, la cual nos permite realizar envíos de alertas y notificaciones.

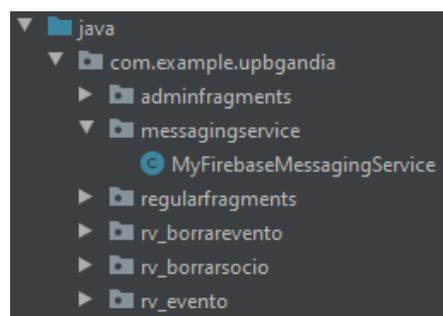


Figura 5.5. Estructura MessagingService

Finalmente, a lo largo de la aplicación se han usado diferentes *RecyclerView* para diferentes funciones como listar los socios a borrar, listar los eventos y listar los eventos a borrar.

Para esto también se han creado diferentes carpetas, ya que para la implementación de uno de estos *RecyclerView* se necesitan como mínimo tres clases: un *Model*, un *Adapter*, y un *ViewHolder*.

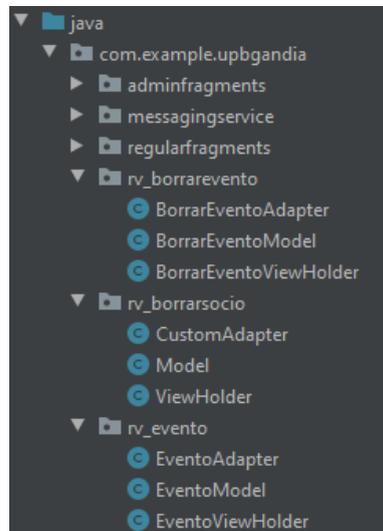


Figura 5.6. Estructura RecyclerViews

Finalmente, para la organización de archivos del **front-end**, se ha seguido la estructura predeterminada que ofrece Android Studio.

Dentro de la carpeta 'res/' se han creado diferentes directorios para los diversos archivos como iconos, fuentes, etc. Los cuales se han organizado de la siguiente manera:

- La carpeta 'drawable' contiene todos los iconos e imágenes que se usan a lo largo de la aplicación.
- En 'font' se almacenan las fuentes que hemos usado durante la aplicación, siendo *righteous* la principal.
- En la carpeta 'layout' se han almacenado todas las diferentes pantallas de la aplicación, así como los fragments, los modelos de *RecyclerView*, las principales *activity*, y los popups.
- La carpeta 'menu' contiene los archivos *xml* con el diseño de los menús de la aplicación, uno para el administrador, y otro para el usuario regular.

- En 'mipmap' están almacenados los iconos que se le dan a la aplicación en la pantalla del móvil.

-Finalmente, en 'values' se almacenan los valores constantes de los colores principales de la aplicación, los *strings* y los estilos customizados.

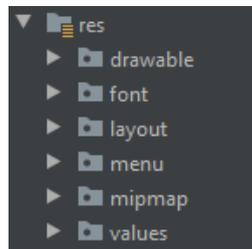


Figura 5.7. Estructura front-end.

### 5.3 Problemas de implementación resueltos

Durante el proceso de implementación de las funcionalidades de la aplicación han ido surgiendo dudas y problemas, algunas de estas se pudieron resolver fácilmente, otras conllevaron días e incluso semanas para poder resolverse.

En esta sección, vamos a explicar los dos problemas principales que hemos tenido durante el proyecto y el proceso de resolución de los mismos.

#### 5.3.1 Actualización de una gráfica cada vez que se pulse un fragment

En la página de estadísticas hay una gráfica circular, la cual indica cuántos socios y no socios espectadores han asistido a los encuentros disputados en Gandía, haciendo una consulta a nuestra base de datos de Firebase, concretamente a la colección 'estadísticas'.

Lo que se hizo en un principio para comprobar el correcto funcionamiento de la gráfica, es decir, que se hiciese la consulta a la colección, y plasmar los datos en la gráfica, fue crear un botón de prueba donde cada vez que fuese pulsado, la aplicación recuperaría los datos de la base de datos y mostraría la gráfica.

El siguiente paso fue pensar cómo quitar el botón y hacer este proceso automáticamente, cada vez que se encendiera la aplicación.

Rápidamente pensamos en el método 'onCreate' de la clase del administrador, clase donde se había implementado la función que creaba la gráfica circular. Es decir, íbamos a hacer una llamada a esta función desde el método 'onCreate', para que cada vez que se iniciara la aplicación hiciese una llamada automáticamente a la creación de la gráfica. Después de probarlo e iniciar la aplicación, la gráfica fue creada automáticamente.

Sin embargo el problema apareció cuando después, tras estar navegando por la aplicación, nos dimos cuenta de que la gráfica desaparecía tras cambiar de fragment, es decir, cuando cambiábamos de fragment, y volvíamos al fragment de la gráfica, esta ya no aparecía, sino que solo se veía una gráfica vacía.

Tras analizar el error, vimos que podía estar en la llamada a la función, que tan solo se realizaba una vez al iniciar la aplicación, cuando había que hacer la llamada cada vez que se iniciase la vista del fragment (no la aplicación).

La solución estaba en hacer la llamada en la clase del fragment donde estaba la gráfica, no en la clase principal.

Otro error que tuvimos fue llamar a la función desde la función 'onCreateView' de la clase del fragment. Esto tampoco funcionaba ya que cuando se ejecuta esta función, los objetos aún no han sido declarados.

Por lo tanto, y después de buscar por internet, la solución final estaba en crear una nueva función en el fragment llamada 'onViewCreated' (la cuál no está implementada por defecto, como la anterior) y llamar a la función para crear la gráfica desde ahí, ya que esta función se ejecutaba inmediatamente después de crear el fragment, y por lo tanto los objetos ya habían sido declarados.

### 5.3.2 Imagen de perfil para usuario regular

Otro problema que hemos tenido durante el proyecto, ha sido el de dejar que el usuario pudiese poner una foto de perfil, en el apartado de perfil de usuario regular.

La idea desde el principio era clara, queríamos que el usuario, al pulsar sobre el icono de la imagen de perfil, abriese sus archivos y eligiese una imagen para establecer como imagen de perfil.

El primer fallo estuvo a la hora de declarar los permisos de la aplicación, ya que no le habíamos otorgado los permisos para buscar en el almacenamiento externo del teléfono, por lo tanto la aplicación se cerraba instantáneamente cuando se pulsaba sobre el icono de imagen de perfil.

Una vez otorgados los permisos, y el teléfono abría correctamente el almacenamiento externo, el siguiente paso era recuperar la imagen y ponerla donde anteriormente estaba el icono de imagen de perfil. Para esto, y tras mucha búsqueda por internet, vimos que podíamos hacer uso de la librería *Picasso*, la cual nos permitía cargar una imagen en un *ImageView* con una *Uri* o *Url*.

Ahora lo que buscábamos era que la imagen saliese en el perfil del usuario cada vez que éste entrase en su perfil. Para ello, había que subir la imagen a alguna parte donde se almacenase y posteriormente se consultase para volver a cargarla en la página de perfil.

Tras mucha búsqueda e investigación, dimos con la solución. Había que hacer uso de la herramienta de *Cloud Storage*, donde se podría subir la imagen que había elegido el usuario.

El procedimiento era el siguiente, había que crear una carpeta en *Cloud Storage* con el ID del usuario en la base de datos, y almacenar dentro de esta la imagen, la cual después, la aplicación consultaría y pondría en la imagen de perfil de la aplicación.

Esto produjo errores, ya que había que recuperar la *URL* o la *URI* de la imagen, y no el archivo como tal para poder usarse en la librería *Picasso*.

Para resolver esto, decidimos hacer lo siguiente: Cuando el usuario elige la imagen de su galería, la aplicación almacena la URI de la imagen elegida en la clase `onActivityResult` en el parámetro 'data'. Lo que íbamos a hacer era recuperar este dato y subirlo a la base de datos del usuario en un nuevo campo llamado 'imageUrl', donde se almacenaría la URL que apunta a la imagen en Cloud Storage, y sí que se podría utilizar en la librería *Picasso*.

Finalmente esto funcionó, pero volvía a surgir el mismo problema que había aparecido en el anterior apartado. Sin embargo, con la experiencia del anterior, pudimos resolver el problema.



## 6.2 Guía de uso.

Durante esta sección, se explicará el proceso que hay que recorrer para realizar cada una de las funciones implementadas en el proyecto.



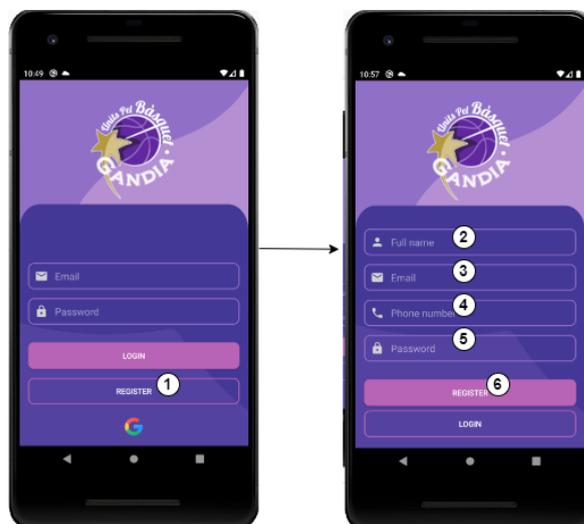
### 6.2.1 Inicio de sesión.

Para iniciar sesión, el usuario deberá rellenar el formulario que se indica (1) y (2) y seguidamente pulsar el botón Login (3).

Una vez hecho esto, la aplicación comprobará que el usuario ha sido dado de alta y le dará acceso a la aplicación.

### 6.2.2 Registro de usuarios.

Cuando se abre la aplicación, el usuario deberá pulsar el botón Register (1) y rellenar el formulario de la página que se abrirá (2), (3), (4), (5), y seguidamente pulsar el botón Register (6), lo cual hará que el usuario se dé de alta en la aplicación y sea redirigido a la pantalla de inicio de sesión.





### 6.2.3 [Regular] Visualizar web del club.

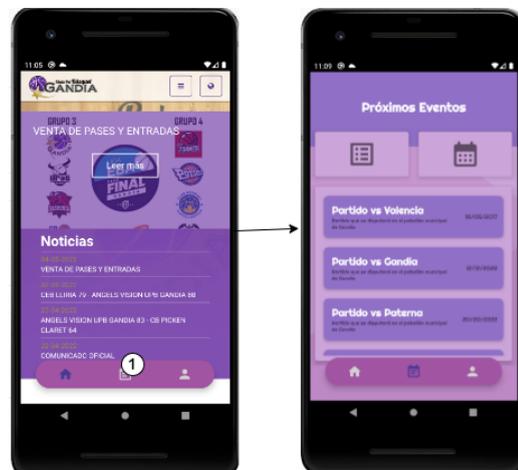
Para visualizar la web del club, el usuario simplemente deberá iniciar sesión en la aplicación y tener el rol de 'regular'.

Si el usuario ha navegado por la aplicación, deberá pulsar el icono de 'Home' (1), para volver a visualizar la web del club.

### 6.2.3 [Regular] Ver los próximos eventos del club.

El usuario podrá ver los próximos eventos que se van a realizar en el club pulsando la segunda opción del menú inferior, el calendario (1).

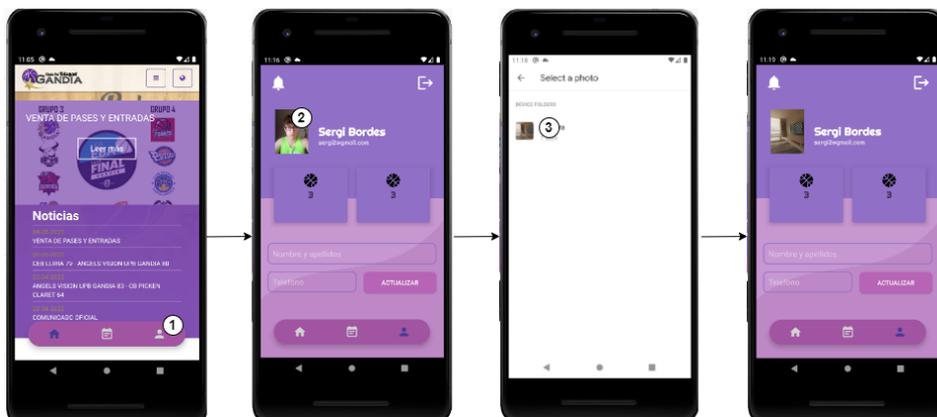
Una vez en esta página, podrá deslizar la lista hacia abajo para ver todos los eventos.



### 6.2.3 [Regular] Cambiar foto de perfil.

Para cambiar la imagen de perfil, el usuario deberá navegar hasta la página de perfil de usuario, en el icono de la persona (1).

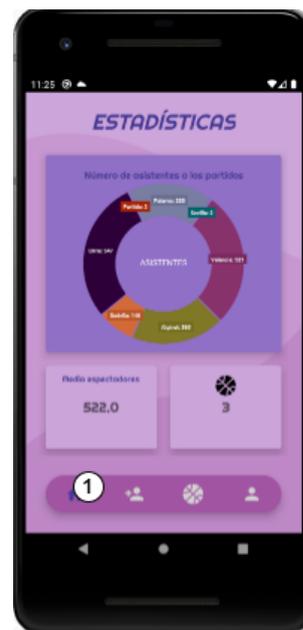
Seguidamente, deberá pulsar sobre la foto de perfil (2), la cual abrirá la galería del teléfono, y seleccionar la foto (3) que desea poner como imagen de perfil.



### 6.2.4 [Admin] Consultar estadísticas.

El usuario administrador podrá ver las estadísticas con tan solo haber iniciado sesión en la aplicación.

Si ya ha navegado por la aplicación y quiere volver a visualizar las estadísticas, deberá pulsar sobre el icono 'home' (1).

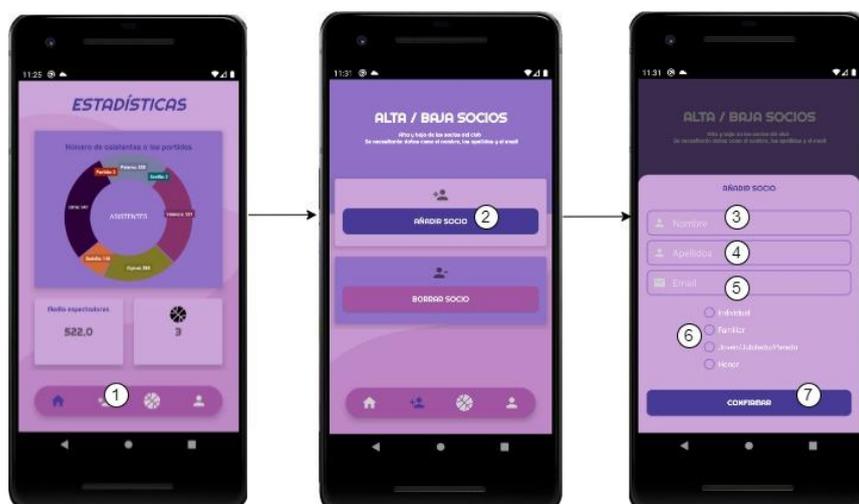


### 6.2.5 [Admin] Añadir socio.

Si un usuario administrador quiere añadir un nuevo socio a la base de datos, deberá navegar hasta la pantalla de socios pulsando el botón (1), y posteriormente pulsar en 'añadir usuario' (2).

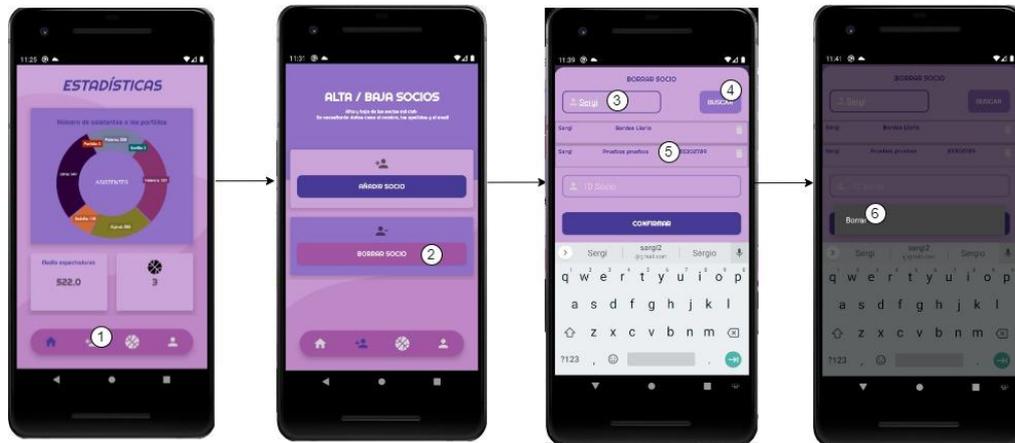
Simplemente deberá rellenar el formulario indicando el nombre (3), los apellidos (4), el email (5) y el tipo de pase (6).

Finalmente, pulsando en confirmar (7), el socio será añadido a la base de datos.



### 6.2.6 [Admin] Borrar socio.

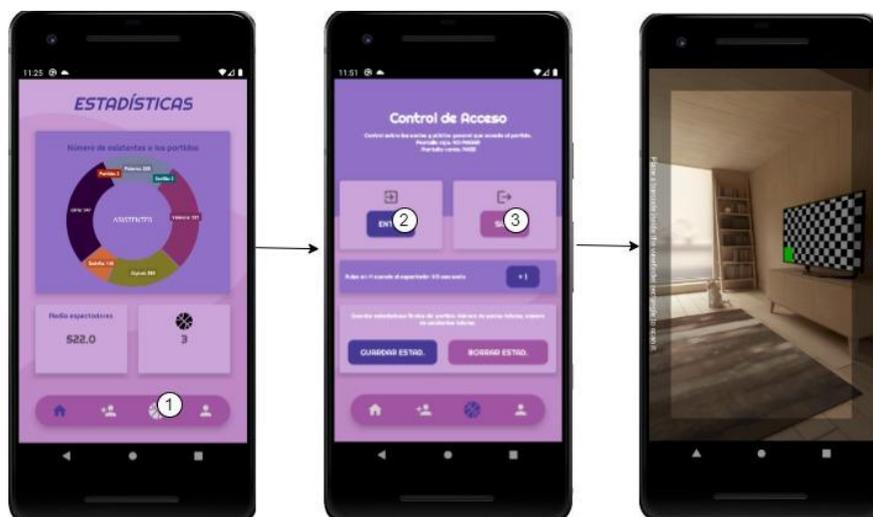
Para borrar el socio, el administrador deberá navegar hasta al apartado socios y pulsar sobre 'Borrar socio' (2), seguidamente escribir en el buscador (3) el nombre del socio que quiere borrar y pulsar en buscar (4). Aparecerá una lista con los socios que coincidan con el nombre escrito, para borrar, deberá mantener (5) en el nombre del socio y pulsar 'Borrar' (6)



### 6.2.7 [Admin] Control de acceso.

El usuario podrá realizar un control de acceso a los encuentros de la siguiente manera: Deberá navegar hasta el apartado 'Partido' (1), y seguidamente pulsar uno de los dos botones 'Entrar' o 'Salir', dependiendo de la función que quiera.

La cámara se abrirá y buscará un QR que escanear, cuando lo encuentre, la aplicación volverá a la página Control de Acceso indicando con una pantalla verde o roja si el usuario puede acceder al partido o no.



### 6.2.8 [Admin] Añadir espectador anónimo.

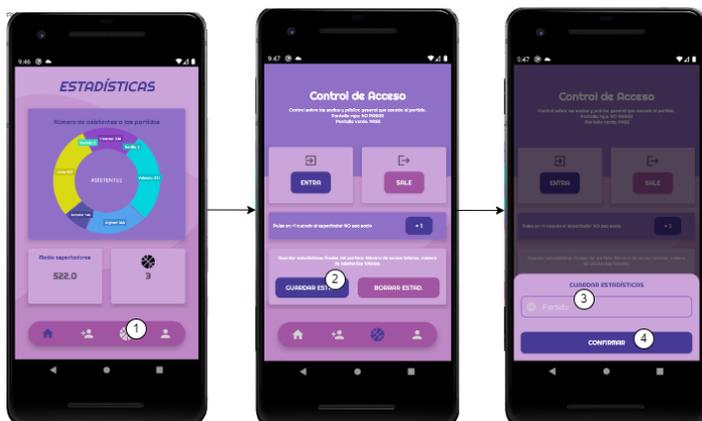
Cuando un espectador no socio compre la entrada y acceda al partido, también debe ser registrado, para un control estadístico.

El administrador deberá navegar hasta la pantalla Control de Acceso (1) y pulsar sobre el botón '+1' (2), y seguidamente confirmar (3).



### 6.2.9 [Admin] Guardar estadísticas de acceso.

El usuario deberá navegar hasta el apartado 'Partido' (1) y cuando acabe el partido y se hayan registrado todos los accesos, pulsará sobre 'Guardar estadísticas' (2), en el popup escribirá el nombre del equipo (3) y pulsará sobre 'confirmar' para subir las estadísticas a la base de datos.



### 6.2.10 [Admin] Guardar estadísticas de acceso.

Como en el paso anterior, el usuario viajará hasta 'Partido' (1) y pulsará en 'Borrar estadísticas' (2) una vez las haya guardado anteriormente.

Tendrá que confirmar (3) para borrarlas definitivamente.

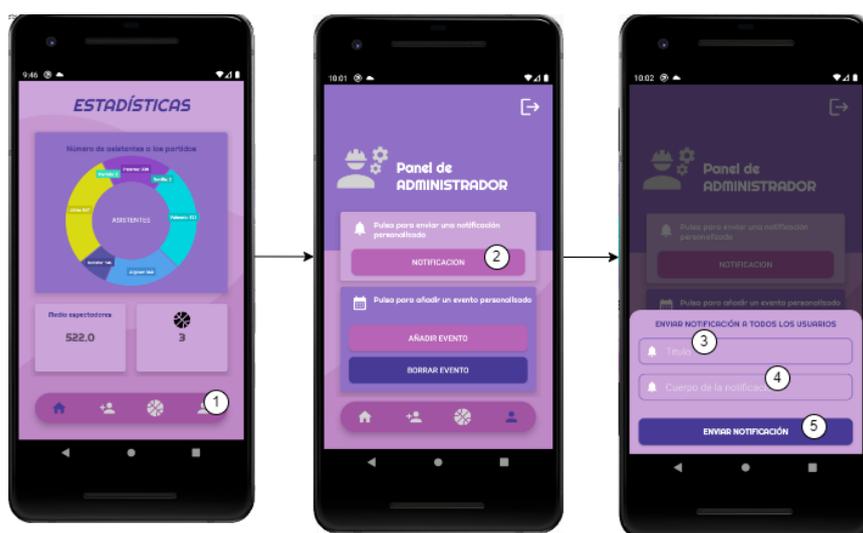


### 6.2.11 [Admin] Guardar estadísticas de acceso.

Para el envío de notificaciones, el administrador deberá acceder al 'Panel de Administrador' (1), y seguidamente pulsar sobre el botón (2).

Una vez acceda al formulario del envío de la notificación, deberá declarar un título (3) y una descripción detallada (4).

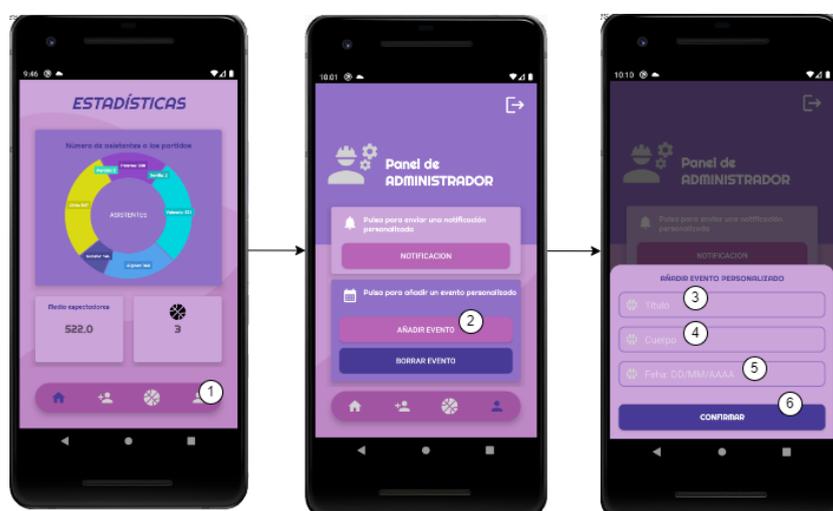
Finalmente, pulsará en 'Enviar notificación' (5) para que llegue el aviso a todos los dispositivos que tengan la aplicación en sus teléfonos móviles.



### 6.2.12 [Admin] Añadir evento de club.

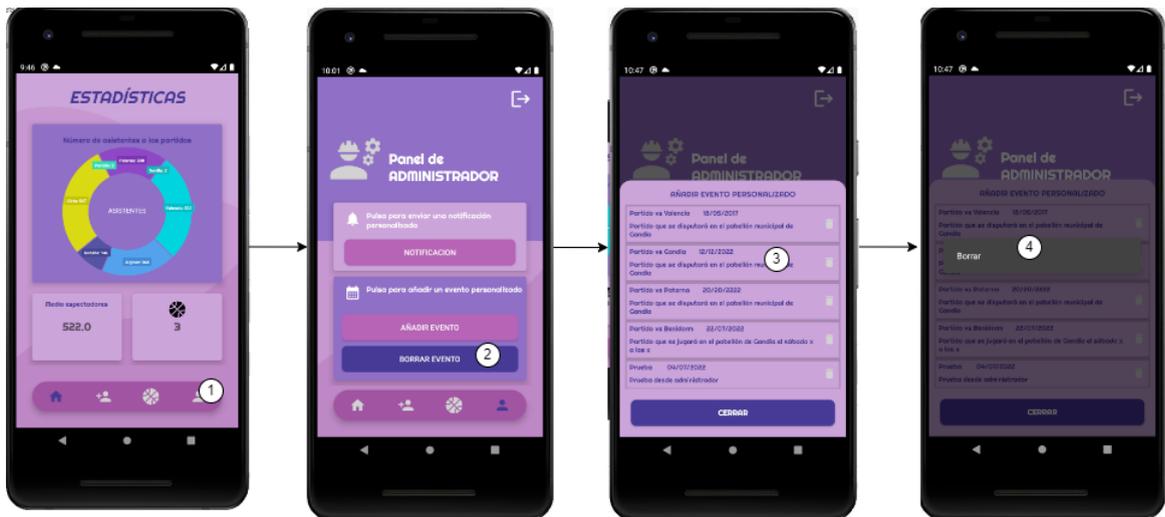
El administrador podrá añadir un evento de club para que le aparezca a los usuarios regulares en el apartado de 'Eventos'.

Para esto, deberá dirigirse al panel de administrador (1) y pulsar sobre 'Añadir evento' (2) y rellenar el formulario con un título (3), un cuerpo (4), y la fecha del evento (5). Tras completar todos los campos, el usuario pulsará 'confirmar' (6).



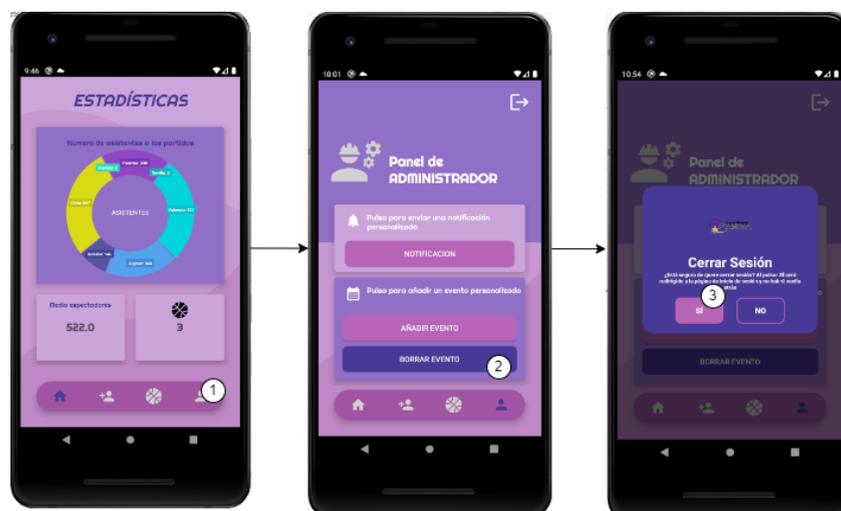
### 6.2.14 [Admin] Borrar evento de club.

Para que el administrador pueda borrar un evento, deberá dirigirse al panel de administrador (1) y pulsar sobre 'borrar evento' (2), donde se abrirá un popup con un listado de los eventos programados, habrá que mantener pulsado sobre el evento que queremos borrar (3) y seguidamente pulsar en la opción 'borrar' (4)



### 6.2.15 [Admin y Regular] Cerrar sesión.

Para salir de la aplicación y cerrar sesión, tanto el admin como el usuario regular deberán dirigirse a la pantalla de perfil (1) y pulsar sobre el icono de 'logout' (2) de arriba a la derecha. Seguidamente confirmar (3) para cerrar sesión. El usuario será redirigido a la pantalla de inicio de sesión.



## 7. EVALUACIÓN

En esta sección del proyecto, valoraremos la usabilidad de la aplicación, es decir, hemos preguntado a 7 personas <sup>1</sup> de entre 16 y 60 años de edad sobre la facilidad de uso que han experimentado con nuestra aplicación. Esto nos indicará como de consistente es el diseño de la aplicación.

Para valorar este aspecto, hemos realizado las siguientes preguntas, donde el valor 1 hace referencia a “Muy en desacuerdo”, y el valor 5 “Muy de acuerdo”:

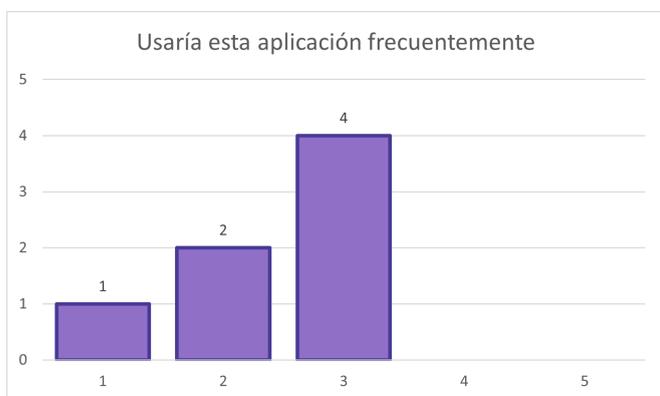


Figura 7.1 Frecuencia de uso de la aplicación

### 7.1 Usaría esta aplicación frecuentemente

Como se ve en la figura 7.1, la mayoría de los usuarios que han testeado la aplicación no la usarían de manera frecuente. Esto puede ser debido a que no se trata de una aplicación de uso diario, sino una app dedicada a una entidad deportiva en concreto.

### 7.2 Es una aplicación fácil de usar.

La mayoría de testadores están muy de acuerdo en que la aplicación tiene un uso fácil. Esto nos confirma que el diseño es consistente.



Figura 7.2 Facilidad de uso.

---

1

aunque el número de usuarios es todavía pequeño, no se ha podido alcanzar un número más elevado y actualmente se siguen realizando estas pruebas

### 7.3 He podido usar todas las funciones con ayuda del manual.

Según las personas que han probado la aplicación, coinciden en que han podido hacer uso de todas las funciones con ayuda del manual.

Esto nos indica que el manual cumple correctamente su función.



Figura 7.3 Utilidad del manual.

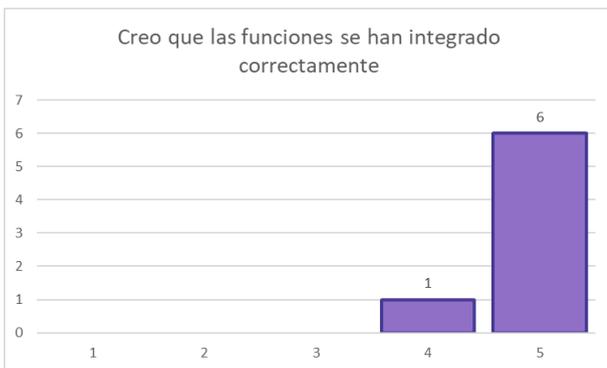


Figura 7.4 Funcionalidad

### 7.4 Creo que las funciones se han integrado correctamente.

La gran mayoría creen que las funciones han sido integradas correctamente.

### 7.5 [Regular] Creo que la aplicación cumple su función.

Los usuarios que han probado la aplicación creen que las funciones del rol regular cumplen la función deseada.

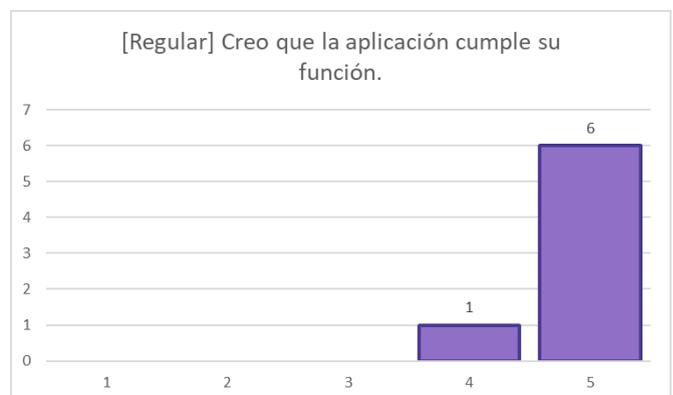


Figura 7.5 Funcionalidad para el usuario regular.

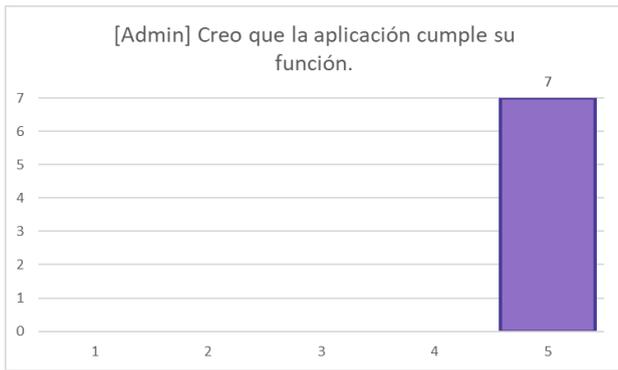


Figura 7.6 Funcionalidad para el usuario administrador.

### 7.6 [Admin] Creo que la aplicación cumple su función.

Los usuarios que han probado la aplicación creen que las funciones del rol admin cumplen la función deseada.

Finalmente, después de realizar las preguntas a todos los voluntarios, creemos que la aplicación ha sido desarrollada con éxito y que cumple todas las funciones previstas de manera correcta.

## 8. MODELO COMERCIAL

Para desarrollar esta sección, empezaremos definiendo qué es un modelo comercial. Según ([Clark, Osterwalder, & Pigneur, 2012](#)) se puede definir Modelo de Negocio “como la lógica que subyace en el sustento económico de las corporaciones, es decir, la lógica que sigue una empresa para obtener ganancias.”

Nuestra lógica para obtener ganancias es la de vender la aplicación al club que nos ofreció implementarla, y por la cual se ha creado la aplicación ad hoc con todas las funcionalidades que demandó el club.

En un futuro, otros modelos comerciales que podríamos usar para obtener ganancias con nuestra aplicación, podrían ser los siguientes:

- **Pago por descarga:** Pagar por adelantado para poder descargar la aplicación, sin tener prueba gratuita previamente.
- **Publicidad:** Podemos ofrecer la aplicación gratuita a los usuarios pero añadiendo publicidad y anuncios dentro de la aplicación, obteniendo beneficios de las empresas que se publiciten.
- **Freemium:** La aplicación será gratuita, pero las funcionalidades serán limitadas, si el usuario quiere obtener funciones extra deberá de pagar una pequeña cantidad.

## 9. CONCLUSIONES

Una vez finalizado y completado el proyecto, realizaremos en esta sección una conclusión, haciendo un resumen de todo el trabajo que se ha realizado, recalcando qué hitos y méritos se han conseguido, valorar también el TFG desde un punto de vista personal incluyendo el aprendizaje y las dificultades superadas.

En el presente TFG, se ha desarrollado bajo demanda, una aplicación móvil que ayuda al club de baloncesto (UPB Gandía) a gestionar sus socios y mantener un control de asistencia a los partidos. La aplicación tiene como usuarios administradores al personal del club, y como usuarios regulares a cualquier persona que decida descargar la aplicación.

Se han desarrollado diferentes funciones para los diferentes roles (admin y regular), cubriendo las necesidades propuestas por la entidad deportiva. Se han implementado funciones para el control de acceso a los partidos, la posibilidad de dar de alta y baja a socios, o la de crear eventos del club, entre muchas otras, para los administradores. En cuanto a los usuarios regulares, se han implementado funciones como la de observar un listado de los eventos anteriormente mencionados, o la posibilidad de navegar por la web del club e informarse de este mismo a través de nuestra aplicación.

Se han cumplido todos los requisitos que el club había solicitado, incluyendo algunas funcionalidades extra.

### Punto de vista personal:

Durante el proceso de este proyecto, he conseguido desarrollar los conocimientos adquiridos durante el Grado de Tecnologías Interactivas, los cuales me han permitido dar una solución informática funcional a un problema propuesto por una empresa, o en este caso, un club deportivo.

Los conocimientos necesarios para el desarrollo de este TFG han sido adquiridos en asignaturas como las siguientes:

- **Programación 2, Proyecto de diseño y programación Web:** Lenguaje de programación JavaScript.

- **Aplicaciones para dispositivos móviles y Proyecto de Internet de las cosas y Aplicaciones móviles:** Para el desarrollo de una aplicación móvil en Android Studio y el manejo de los servicios de Firebase.

- **Todos los proyectos realizados cada cuatrimestre:** Uso de la metodología Scrum y buena organización, haciendo uso de la herramienta Trello para la organización de las tareas.

- **Diseño de interfaces y experiencia de usuario:** Para la creación del diseño de las diferentes pantallas de la aplicación.

Además, se han ejercitado las siguientes competencias transversales <sup>2</sup> del grado de Tecnologías Interactivas (siguiendo el orden):

- **Comprensión e integración.** En especial del código de programación.

- **Aplicación y pensamiento práctico.** Aplicar los conocimientos teóricos y establecer el proceso a seguir para alcanzar determinados objetivos.

- **Análisis y resolución de problemas:** Capacidad de análisis de problemas y el planteamiento de la solución.

- **Innovación, creatividad y emprendimiento:** Responder satisfactoriamente y de forma original a las necesidades y demandas solicitadas por el club.

- **Diseño y proyecto:** Diseñar de manera efectiva el proyecto.

- **Aprendizaje permanente:** Seguir aprendiendo detrás de un cierto objetivo.

- **Planificación y gestión del tiempo:** Orden y organización para una correcta planificación del proyecto.

Finalmente y como experiencia personal en el desarrollo de este proyecto, he sido capaz de resolver problemas de forma autosuficiente, desarrollando la capacidad de saber cómo buscar la posible solución por mi mismo o buscando ayuda por internet, sabiendo detectar el problema y adaptando las posibles soluciones encontradas a este proyecto. También he sido capaz de organizarme de manera efectiva gracias a las habilidades adquiridas durante todos los proyectos realizados en cada cuatrimestre del grado de Tecnologías Interactivas.

---

2

[https://www.upv.es/titulaciones/GTI/menu\\_1015443c.html](https://www.upv.es/titulaciones/GTI/menu_1015443c.html)

## BIBLIOGRAFÍA

Clark, T., Osterwalder, A., & Pigneur, Y. (2012). *Business model you: a one-page method for reinventing your career*. John Wiley & Sons.

Units Pel Bàsquet Gandia. (29 noviembre de 2019). Gandia Basquet. <https://www.upbgandia.com/>

Cluber. (8 octubre de 2019). Cluber - Impulsamos clubes deportivos. <https://www.cluber.es/>

Director11. (9 noviembre de 2014). Director11 - FOOTBALL CLUB MANAGEMENT REVOLUTION. <https://www.director11.com/>

Clupik. (10 mayo de 2017). Clupik - Herramienta de gestión de clubes. <https://clupik.com/>

PlayOff. (8 febrero de 2017). PlayOff - Programa de gestión de socios online | Playoff Informática. <https://playoffinformatica.com/>

Trigás Gallego, M. (2012). Metodología scrum.

Garrido Cobo, J. (2013). Desarrollo de aplicaciones móviles.