



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Aplicación para la suscripción a eventos deportivos

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Carlos Martin Yerle Ballara

Tutor: Pedro Joaquín Gil Vicente

2019/2020

Me gustaría dedicar el siguiente trabajo a las personas que me apoyaron y me ayudaron a realizarlo, en primer lugar, a mi madre y hermana, mis pilares fundamentales en mi vida que me motivaron y sustentaron durante toda la carrera.

También me gustaría agradecer a mi tutor Pedro Joaquín Gil Vicente por su paciencia, predisposición e infinitos consejos para lograr realizar este proyecto y también a dos profesores que fueron de gran apoyo y ayuda en la realización de todo el proyecto Juan Carlos Ruiz García y David De Andrés Martínez.

Por último, no quiero olvidarme de mis amigos que me apoyaron y me aconsejaron con el proyecto, haciendo una mención especial a Israel Sanjurjo quien fue una ayuda clave en los inicios de la aplicación.

Resumen

Hoy en día el deporte forma parte de la vida cotidiana de gran parte la población, tanto sea de forma individual o realizando deportes de equipo, por eso el objetivo de este trabajo es diseñar e implementar una aplicación móvil que permita a los usuarios practicar diferentes deportes (fútbol, vóley, pádel, básquet, etc.) aún si no disponen del número suficiente de integrantes para conformar un equipo completo. Concretamente, se diseñará una aplicación que permitirá a los usuarios la creación de diferentes eventos deportivos para que otros usuarios se puedan unir y poder practicar el deporte en cuestión.

Para el desarrollo de la aplicación móvil se utilizó Java como lenguaje de programación, y para el resto de las partes de nuestro sistema (servidor, base de datos, etc.) hemos utilizado Firebase porque nos proporciona una amplia gama de servicios y herramientas que nos ayudaron y facilitaron el trabajo a la hora de desarrollarlo, y porque es una tecnología que está siendo muy utilizada en el mundo laboral actualmente.

Palabras clave: deporte, aplicación móvil, Android, Firebase.

Abstract

Nowadays sport is part of the daily life of a large part of the society, either individually or in team sports, so the objective of this project is to design and implement a phone application that allows users to practice different sports (soccer, volleyball, paddle, basketball, etc.) even if they do not have enough members to form a complete team. Specifically, this application will be designed that will allow users to create different sporting events so that other users can join and be able to practice the sport in question.

To develop the mobile application we use Java as a programming language and for the rest of the parts of our system (server, database, etc.) we have used Firebase because it provides us with a wide range of services and tools that helped and facilitated us work when developing and also because it is a technology that is being widely used in the world of work today.

Keywords: sport, phone app, Android, Firebase.

Tabla de contenidos

Contenido

1. Introducción	11
1.1. Motivación	11
1.2. Objetivos	12
1.3. Metodología	13
1.4. Estructura de la memoria	14
2. Estado del arte	16
2.1. Trabajos relacionados	16
2.1.1. IFSports	16
2.1.2. Fubles	17
2.1.3. PadelTrack	18
2.1.4. MeetUp	19
2.1.5. Timpik	21
2.2. Análisis de aplicaciones estudiadas	23
2.3. Propuesta	26
3. Análisis del problema	27
3.1. Especificación de requisitos	27
3.1.1. Contexto de la aplicación	27
3.1.2. Modelo de dominio	28
3.1.3. Límites del sistema	30
3.1.4. Características del sistema	30
3.1.5. Restricciones y requisitos específicos	31
3.1.6. Casos de uso	32
3.1.7. Análisis de riesgos	38
3.2. Identificación y análisis de soluciones posibles	38
3.3. Solución propuesta	40
4. Diseño de la solución	42
4.1. Arquitectura del sistema	42
4.2. Diseño detallado	42
4.3. Tecnología utilizada	45
4.3.1. Firebase	45
4.3.2. Android Studio	46
4.3.3. Java	47
4.3.4. GitHub	47

5.	Desarrollo de la solución propuesta	48
5.1.	Modificaciones	52
6.	Implantación	54
7.	Pruebas	55
7.1.	Pruebas unitarias -JUnit-	55
7.2.	Pruebas de los servicios de Firebase	55
7.3.	Validación	57
8.	Manual de uso	62
8.1.	Registro y Login	62
8.2.	Perfil	63
8.3.	Eventos	65
9.	Conclusiones	69
9.1.	Relación Proyecto – Estudios cursados	69
9.2.	Trabajos futuros	69
10.	Bibliografía	71

Índice de tablas

Tabla 1. Características relevantes o no para Apps de gestión de eventos deportivos.	23
Tabla 2. Características presentes en las diferentes aplicaciones analizadas.	25
Tabla 3. Términos ontológicos relevantes del dominio.	27
Tabla 4. Términos técnicos en la implementación.	28
Tabla 5. Descripción de la clase usuario.	29
Tabla 6. Descripción de la clase evento.	29
Tabla 7. Descripción de la clase deporte.	29
Tabla 8. Descripción de la clase reserva día.	29
Tabla 9. Restricciones del sistema.	31
Tabla 10. Caso de uso registrar.	32
Tabla 11. Caso de uso iniciar sesión.	33
Tabla 12. Caso de uso recibir notificación evento nuevo.	34
Tabla 13. Caso de uso borrarse de un evento registrado.	34
Tabla 14. Caso de uso ver los eventos registrados.	34
Tabla 15. Caso de uso borrar evento creado.	35
Tabla 16. Caso de uso registrarse en evento creado.	35
Tabla 17. Caso de uso crear evento deportivo.	35
Tabla 18. Caso de uso cerrar sesión.	36
Tabla 19. Caso de uso ver eventos creados.	36
Tabla 20. Caso de uso ver perfil.	36
Tabla 21. Caso de uso editar intereses.	37
Tabla 22. Caso de uso inscribirse o desinscribirse en algún deporte.	37
Tabla 23. Creación de usuarios falsos.	38
Tabla 24. Saturación del sistema.	38
Tabla 25. Gestión de infraestructura de base de datos por parte de terceros.	38
Tabla 26. Evaluación de la efectividad	57
Tabla 27. Evaluación de la eficiencia	58
Tabla 28. Evaluación de la satisfacción	58
Tabla 29. Evaluación de la usabilidad	58
Tabla 30. Significado de la numeración en las encuestas	58

Índice de imágenes

<i>Ilustración 1. Modelo para el desarrollo del Software</i>	13
Ilustración 2. Página principal de IF7Sport	16
Ilustración 3. IF7Sports -> Inscribirse a un partido	17
Ilustración 4A. Lista de eventos disponibles	18
Ilustración 4B. Lista de partidos disponibles	18
Ilustración 5. PadelTrack Inscripción a partido	19
Ilustración 6. PadelTrack Creación de Partido	19
Ilustración 7. MeetUp selección de actividad	20
Ilustración 8. Timpik inscripción a un evento	22
Ilustración 9. Diagrama de clases	28
Ilustración 10. Límites y actores del sistema	30
Ilustración 11. Usuario no registrado	32
Ilustración 12. Usuario registrado en el sistema	33
Ilustración 13. Arquitectura del sistema	42
Ilustración 14. Esquema de la base de datos	44
Ilustración 15. Pantalla menú principal	49
Ilustración 16. Pantalla perfil de usuario	50
Ilustración 17. Pantalla crear reserva	50
Ilustración 18. Pantalla mis eventos	51
Ilustración 19. Pantalla buscar reservas	51
Ilustración 20. Pantalla mis reservas	52
Ilustración 21. Prueba Junit Usuario	55
Ilustración 22. Resultado ejecución de JUnit test	55
Ilustración 23 Resultados de la encuesta de Efectividad	59
Ilustración 24 Resultados de la encuesta de Eficiencia	60
Ilustración 25 Resultados de la encuesta de Satisfacción	60
Ilustración 26 Resultados de la encuesta de Usabilidad	61
Ilustración 27. Pantalla Editar perfil	64
Ilustración 28. Pantalla crear evento	65
Ilustración 29. Pantalla mis eventos	66
Ilustración 30. Pantalla buscar reserva	67
Ilustración 31. Pantalla mis reservas	68

1. Introducción

Hoy en día, la práctica de deportes es una actividad que se encuentra totalmente instalada en nuestra sociedad, ya sea por el hecho de realizar actividades entre amigos, socializar y fomentar un grupo con aficiones en común o bien desde el punto de vista de la salud, o una buena combinación de ambos.

Actualmente, para la realización de actividades deportivas en grupo las personas logran organizar o unirse a estas mediante la comunicación “boca a boca”. Este método resulta muy útil para mucha gente, pero también existe la problemática de todas aquellas personas que no logran tener un equipo o un grupo de conocidos el cual el permita practicar dicho deporte.

Así pues, con mi proyecto, haciendo uso de la tecnología móvil que está al alcance de todos, trataré de brindar una solución a todas aquellas personas que no son capaces o que tienen dificultades para conformar un equipo o grupo y poder practicar su deporte preferido.

1.1. Motivación

A lo largo de mi vida el deporte siempre me acompañó como medio de diversión y liberación de preocupaciones y tensiones. Cuando era niño o adolescente el deporte estaba gestionado por un club o una asociación, ya que eran competiciones y yo solamente tenía que asistir a los días y eventos que organizaban. Pero cuando ya abandoné la etapa de competición, los deportes que realizaba estaban gestionados por mí, buscaba mi grupo de amigos o conocidos para practicar esos deportes.

En general, me gustan muchos deportes, pero en concreto soy un amante del fútbol y del vóley playa, por eso cuando llegué a Valencia y a la UPV vi la gran posibilidad de poder practicar estos deportes, tanto sea por las magníficas instalaciones de la universidad como por la gran cantidad de redes de vóley que posee la playa de la Malvarrosa. Pero muchas veces las ganas de practicar un deporte se tenían que ver reprimidas debido a que no encontraba o no conocía a la suficiente cantidad de gente como para poder realizar la actividad. Este problema no solo lo sufro yo, sino que es un problema frecuente a la hora de realizar la práctica de un deporte, lo cual se puede observar en la última encuesta válida publicada por ministerio de cultura y deporte “Encuesta de hábitos deportivos en España 2015” [1] donde un 3.5% declara que no practica deporte debido a que no tiene con quien practicarlo.

Por otro lado, sabemos que las apps relacionadas a la práctica de deportes llevan en auge mucho tiempo, siendo un gran pilar en la ayuda o motivación para la práctica de deporte, aunque mayoritariamente están enfocadas al deporte individual y no al deporte en equipo.

Así que empujado por esas ganas de realizar las actividades que más me gustan, ver que hay más personas que sufren el mismo problema y que tenemos unas herramientas en la que podemos apoyarnos debido a la situación de auge de las aplicaciones relacionadas con el deporte, decidimos buscar una solución para este problema. La solución puede ser

tanto de manera específica para una organización, en este caso la UPV, como también de manera general para cualquier centro deportivo que desee implementarlo.

1.2. Objetivos

El objetivo principal de este proyecto es diseñar e implementar una aplicación móvil que permita a los usuarios crear o inscribirse en eventos deportivos de su interés. También podrán suscribirse a sus deportes de interés para ser notificados cuando se cree un nuevo evento en ese deporte.

Con el fin de desarrollar el proyecto estos son los objetivos particulares para conseguir:

- Brindar al usuario la posibilidad de gestionar (crear y editar) su perfil estableciendo sus deportes de interés.
- Tendrá un sistema de mensajes pop-up para notificar al usuario que hay nuevos eventos deportivos en sus deportes de interés.
- Permitirá a los usuarios crear eventos deportivos para sus deportes de interés en la fecha y hora que el usuario elija.
- Tendrá la opción de buscar eventos que sea del interés del usuario en los días que elija e inscribirse si es de su interés.
- También tendrá la opción de cancelar una suscripción a un evento que ya estaba inscrito.
- La aplicación tendrá un control de eventos y valoraciones para poder sancionar a los usuarios que no asistan a los eventos deportivos.
- Ofrecerá a los usuarios una interfaz sencilla e intuitiva para facilitar su uso.

1.3. Metodología

En este apartado hablaremos de los pasos a realizar para la construcción de nuestro proyecto y lo mostraremos a través de la Ilustración 1, el proceso utilizado fue el de Espiral:

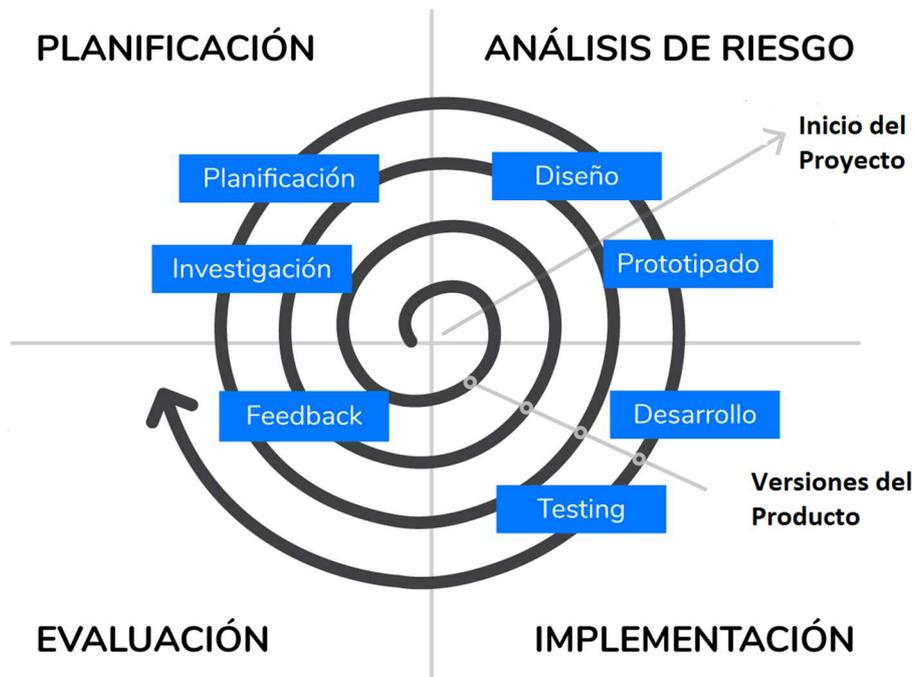


Ilustración 1. Modelo para el desarrollo del Software

Para el desarrollo de nuestro proyecto hemos utilizado el tan conocido Proceso en Espiral [1], que permite ir desarrollando poco a poco el proyecto, aumentando su complejidad en cada ciclo.

Para cada fase el procedimiento siempre fue el mismo:

- Determinar los objetivos que tenemos en el ciclo que vamos a comenzar incrementando la funcionalidad y complejidad del ciclo anterior.
- Se tiene en cuenta todo lo que puede afectar al proyecto y se diseña o prototipa lo que se pretende realizar para poder validarlo en el futuro.
- Se implementa y se valida el software según lo que se había programado.
- Una vez finalizado el desarrollo de la etapa actual, se evalúa como se desarrolló y el resultado obtenido, indicando si se cumplieron los objetivos y nos ayudará a encarrilar el siguiente ciclo.

A continuación, detallaremos algunas actividades que realizaremos a lo largo de nuestro proyecto en las diferentes etapas del ciclo:

1. Estudiamos las necesidades y características que queremos que tenga nuestra aplicación sin entrar en detalles técnicos.
2. Viendo las necesidades establecidas anteriormente, estudiaremos que tecnologías nos serían más útiles para nuestro proyecto.

3. Definiremos de manera detallada utilizando UML los diferentes casos de uso que tendrá nuestro sistema.
4. Describiremos en modo de diagrama las relaciones entre las entidades que compondrán nuestro software.
5. Estableceremos la estructura que tendrá nuestra base de datos.
6. Crearemos los mock-up de la aplicación.
7. Definiremos la arquitectura que usaremos.
8. Comenzaremos con la implementación del código teniendo en cuenta los diagramas realizados.
9. Realizaremos las pruebas de comprobación para cada módulo realizado.
10. Testeo de la aplicación por usuarios externos para comprobar el funcionamiento global.

1.4. Estructura de la memoria

Antes de continuar con nuestro proyecto nos gustaría ofrecer la estructura de este documento, presentando una breve descripción de lo que se encontrará el lector en cada uno de los capítulos.

- **Capítulo 2: Estado del arte**

Dentro de este capítulo hablaremos sobre el contexto de nuestra aplicación y enumeraremos y comentaremos diferentes alternativas a nuestra aplicación.

- **Capítulo 3: Análisis del problema**

En este capítulo, el lector encontrará la especificación de requisitos realizada para el proyecto con diferentes diagramas y explicaciones que lo situarán y le darán una idea de lo que se pretende realizar.

- **Capítulo 4: Diseño de la solución**

En el capítulo 4 podremos ver la solución que escogimos para desarrollar nuestro proyecto, el diseño que hemos elegido, la arquitectura y las tecnologías a utilizar.

- **Capítulo 5: Desarrollo de la solución propuesta**

Tendremos una explicación del desarrollo realizado utilizando como ejemplos los Mock-ups y explicándolos para facilitar el entendimiento.

- **Capítulo 6: Implantación**

Se presentará las necesidades de nuestro proyecto para realizarse la implantación de éste.

- **Capítulo 7: Pruebas**

Aquí se explicarán todas las pruebas realizadas para la verificación y validación de la aplicación.

- **Capítulo 8: Manual de Uso**

En este capítulo podremos ver una pequeña guía para poder hacer uso de nuestra aplicación. Esta guía dispondrá de las instrucciones más las imágenes que ayudarán a realizar las funciones que nos ofrece nuestra app.

- **Capítulo 9: Trabajos futuros**

En este capítulo comentaremos el resultado de nuestro proyecto, si se cumplió con el objetivo, los problemas que tuvimos y la relación del proyecto con el curso. También encontraremos una pequeña lista de mejoras o funcionalidades nuevas que nos gustaría añadir en el futuro a nuestra aplicación.

2. Estado del arte

Dentro de este apartado nos detendremos a estudiar todas aquellas aplicaciones similares al proyecto que vamos a desarrollar.

A medida que van pasando los años, la tecnología sigue avanzando y cada vez está más presente en nuestra vida cotidiana con la finalidad de ayudarnos en nuestro día a día en forma de herramientas. Como no podía ser de otra manera, la práctica de deportes en grupo se ve beneficiada por la implicación de la tecnología. Esto lo pudimos verificar cuando después de realizar una búsqueda por internet pudimos encontrar varias aplicaciones similares a la nuestra, algunas enfocadas a un deporte en concreto, otras a la organización de grandes eventos deportivos y algunas muy similares a la nuestra (varios deportes creando partidos específicos).

2.1. Trabajos relacionados

2.1.1. IFSports

IFSports¹ es una aplicación móvil que se dedica exclusivamente a la gestión de partidos de fútbol en Madrid. El grupo encargado de gestionar la aplicación es el que crea los partidos de fútbol para todos los días. El usuario parte de una pantalla principal como vemos en la ilustración 2, y mediante esta pantalla puede buscar partidos y elegir la opción que mejor se le adapte a él como también ver los eventos a los que está suscrito como se ve en la ilustración 3.



Ilustración 2. Página principal de IF7Sport

¹ <https://www.if7sports.com/>

En IF7Sports el usuario deberá de escoger un día del calendario y podrá ver los partidos disponibles en ese día específico (siempre en la misma dirección). Una vez que esté inscrito podrá ver cómo se van completando los equipos.

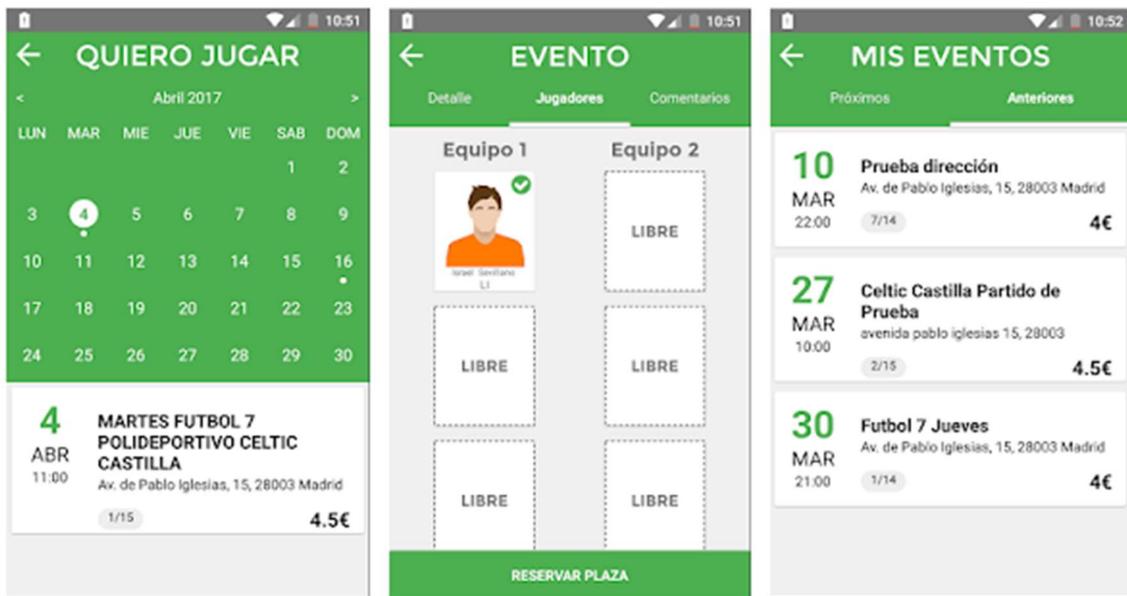


Ilustración 3. IF7Sports -> Inscribirse a un partido

2.1.2. Fubles

Fubles² es un gestor que facilita la participación en eventos deportivos creados cerca de tu zona. Esta aplicación opera mayormente dentro de Italia y se encarga de hacer llegar a los usuarios eventos deportivos creados a un precio muy bajo, facilitando la participación de equipos completos, grupos de amigos o de personas de manera individual.

En esta aplicación el usuario tiene un perfil y una puntuación que se obtiene por la valoración de los otros jugadores del partido. El usuario también tiene la opción de crear partidos en los centros deportivos que están disponibles y los otros se pueden inscribir. Como se puede apreciar en las ilustraciones 4 y 5 el usuario dispone de diferentes modalidades de búsqueda, tanto por calendario como por eventos destacados. Mayormente, está orientada a la práctica de fútbol, pero está tratando de expandirse para una gama más amplia de deportes. Los eventos aparecen en una lista diciendo donde se juega, la hora y el valor a pagar.

² <https://www.fubles.com/>

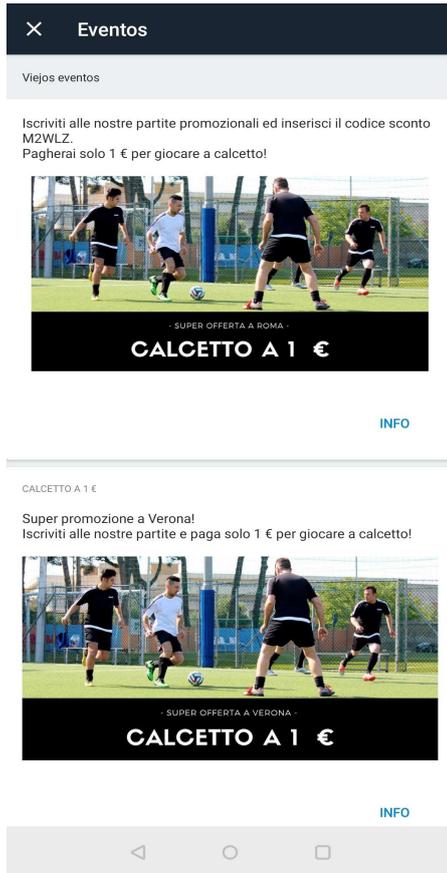


Ilustración 4A. Lista de eventos disponibles



Ilustración 4B. Lista de partidos disponibles

2.1.3. PadelTrack

PadelTrack³ es una de las aplicaciones más populares en el mundo del pádel no profesional hoy en día. Esta aplicación como podemos ver en la ilustración 5 ayuda a los usuarios a encontrar partidos de pádel que otros usuarios crean, así estos solamente tendrán que apuntarse a los partidos creados. También tiene la opción de crear partidos como se ejemplifica en la ilustración 6, aquí los usuarios fijan el día, la hora, duración y localización del partido, los otros usuarios se irán inscribiendo en las posiciones que están libres para ese partido.

³ <https://www.padeltrack.com/>



Ilustración 5. PadelTrack Inscripción a partido

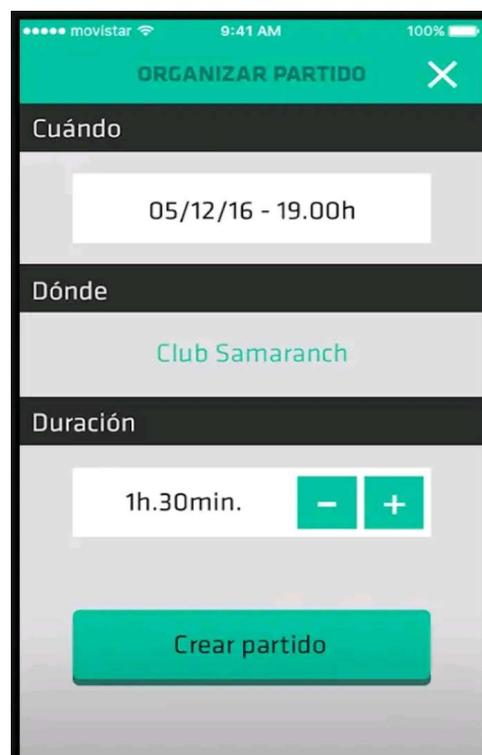


Ilustración 6. PadelTrack Creación de Partido

PadelTrack realiza una clasificación por niveles gracias a un algoritmo y a las valoraciones de los usuarios.

2.1.4. MeetUp

MeetUp⁴ es una aplicación móvil que fue creada para conectar gente con fines comunes y poder asistir a estos eventos. Estos eventos no son solo deportivos, también podemos

⁴ <https://www.meetup.com/>

encontrar cines, teatro y otras actividades. En la ilustración 7, podemos ver las diferentes ofertas de eventos que nos podemos encontrar donde no todas son deportivas. Pero nosotros vamos a centrarnos en la parte del deporte. Los usuarios deberán de pertenecer a un grupo de interés, o bien porque se inscriben o porque lo crean.

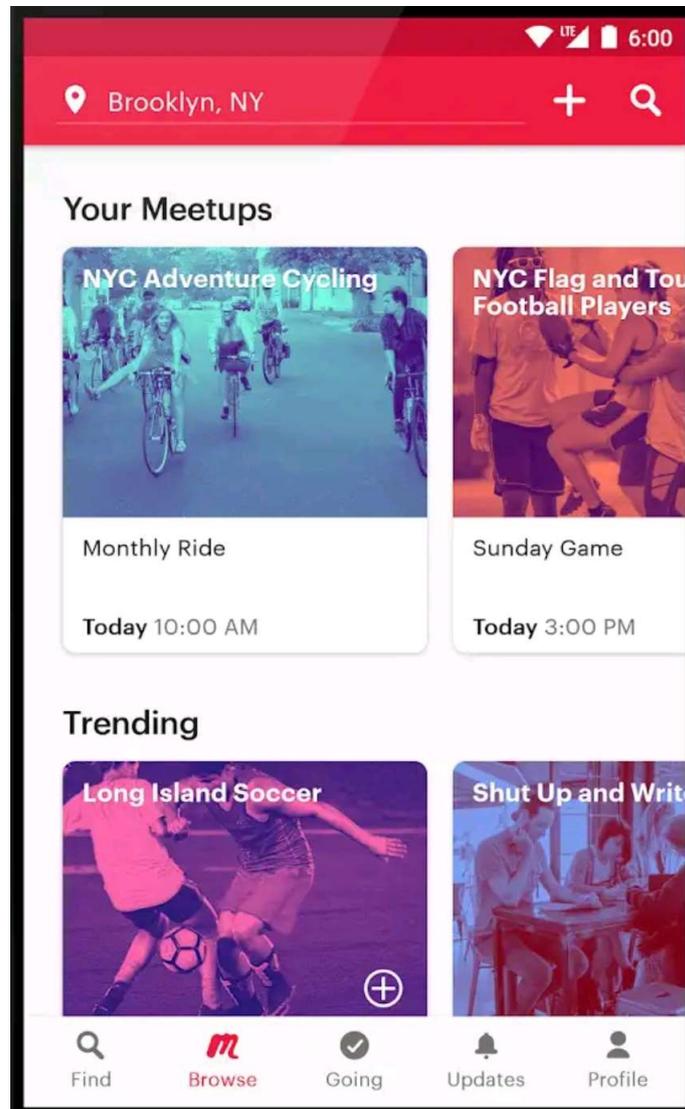


Ilustración 7. MeetUp selección de actividad

2.1.5. Timpik

Timpik⁵ es una aplicación que intenta unir deportistas en sus respectivas disciplinas siendo ésta la aplicación más similar a la nuestra. En esta ocasión, como se refleja en las ilustraciones 8 y 9 el usuario debe de tener un perfil deportivo con sus intereses y tu zona para jugar.



Ilustración 8. Timpik Perfil de Usuario



Ilustración 9. Timpik Deportes de Interés

⁵ <http://www.timpik.com/>

Para poder participar en un evento, el usuario solamente tiene que crear o buscar un evento deportivo de su interés en la zona. Después, tiene que esperar que los otros usuarios se unan o unirse a el evento seleccionado. En la ilustración 10 que aparece a continuación, vemos como aparecerían los eventos creados por los usuarios mostrando el precio, participantes, día y hora.



Ilustración 8. Timpik inscripción a un evento

En esta aplicación también encontramos diferentes niveles en cada deporte, en muchos los tienes que poner de manera manual y otros son con preguntas para poder estimar tu nivel. También ofrece la opción de mensajes entre usuarios dentro de los eventos para tener contacto entre ellos. También ofrece la posibilidad de lista de espera para los eventos, que la aplicación avisa si hay alguna baja de último momento. Una de las opciones diferenciadoras de esta aplicación es la posibilidad de crear clubs, tanto privados como públicos para que los usuarios de los clubes reciban las notificaciones de nuevos eventos en dichos clubes.

2.2. Análisis de aplicaciones estudiadas

En este apartado del documento nos centraremos en todos los proyectos mencionados anteriormente, realizando una comparación entre ellos sobre las características que deberían de presentar o no dichas aplicaciones.

Primero, nombraremos y explicaremos las características que creemos que debemos encontrar o no en este tipo de aplicaciones, estas características las encontraremos en la Tabla 1. Una vez que tengamos las características explicadas pasaremos a realizar una comparativa entre todos los proyectos mencionados viendo cuales ofrecen estas características y cuáles no, para eso tendremos la Tabla 2 que nos reflejará el resultado de la comparativa.

Tabla 1. Características relevantes o no para Apps de gestión de eventos deportivos.

Nombre	Descripción	Deseable
Interfaz amigable	La aplicación posee una interfaz sencilla e intuitiva.	Sí. Facilita el uso por parte del usuario.
Envío de notificaciones	Las aplicaciones enviarán una notificación a los usuarios que pertenezcan al evento que sufre algún cambio (creación, nuevo participante, cancelación).	Sí. Mantiene informado al usuario en todo momento evitando sorpresas.
Multideportivo	Brinda a los usuarios la posibilidad de practicar varios deportes y no centrarse solamente en uno.	Sí. Abarca un mercado mayor, ya que no se ciñe solamente a uno.
Dependencia de red social	Para poder hacer uso de la aplicación es necesario acceder o utilizar otra aplicación o una red social (Facebook, WhatsApp, Twitter, etc).	No. El usuario podrá no querer vincular sus redes sociales o no depender de otra app para practicar deporte, para eso utilizaría esa aplicación.
Creación de eventos propios	Permitir a los usuarios crear los eventos de los deportes que les apetezca a la hora que les venga bien a ellos.	Sí. El usuario dispondrá de total libertad e independencia para practicar deporte.
Publicidad	A la hora de acceder o utilizar la aplicación nos encontramos con publicidad.	No. Resulta tedioso tener que esperar a que termine la publicidad para usar una app.
Independencia de idioma o país	Permitir utilizar la aplicación para cualquier ubicación, sin restringir su uso a una ciudad en concreto o país.	Sí. Abarcaría mayor mercado y podría ser útil en cualquier parte.
Filtrado de eventos creados e inscritos	Ofrecer al usuario la posibilidad de ver de manera separada los eventos que él creó y los eventos en los que está inscrito.	Sí. Facilitaría la búsqueda al usuario y brindaría la información de manera más clara.
Estadísticas de partido	Cada usuario tendrá en su perfil unas estadísticas que irán variando, dependiendo del resultado de cada partido.	Opcional. Interesante, pero no tendría mayor trascendencia o importancia para el uso de app.
Eventos gratuitos	Los usuarios tienen la posibilidad de acceder a eventos gratuitos, sin tener que pagar para participar en ellos.	Sí. Abriría las puertas a un número mayor de usuarios.

Aplicación para la suscripción a eventos deportivos

Eventos de pago	Los usuarios deberán de pagar para poder participar en los eventos deportivos.	No. Reduciría el número de usuarios por el precio y por tener que introducir un método de pago.
Ranking	La aplicación ofrecerá un ranking (semanal o mensual) de los usuarios dependiendo sus estadísticas y valoraciones en los partidos.	Opcional. Resultaría un complemento muy atractivo, pero seguiría siendo un extra y no una necesidad.
Creaciones de grupos privados	Se podrán crear grupos privados para facilitar la organización de eventos solamente con los miembros de ese grupo.	Opcional. Facilitaría la práctica entre amigos, pero su ausencia no limitaría a la aplicación.
Permitir crear cualquier tipo de evento	A la hora de la creación de un evento el usuario podrá decidir el tipo de evento, no solo deportivo, de cualquier índole (lectura de libro, cata de vinos, etc).	No. Podría distorsionar el fin de la aplicación (practicar deporte) con eventos de otra índole que no sean deportivos.
Perfil de usuario con fotografía	El usuario dispondrá de la opción de agregar una fotografía a su perfil.	Opcional. Sería interesante, pero podría generar problemas de carga, prejuicios, racismo, etc.
Clasificación de niveles	La aplicación evalúa a los usuarios en los diferentes deportes para poder establecer su nivel de juego.	Sí. Ayudaría para facilitar la práctica de deporte entre jugadores del mismo nivel.
Valoración post-partido	Una vez acabado el partido los usuarios deberán valorar a los otros usuarios.	Opcional. Ayudaría con la clasificación de niveles, pero se podría alterar maliciosamente.
Lista de espera a los eventos	Cuando se alcance el límite de usuarios para cada evento, los usuarios podrán inscribirse como posibles reservas por si alguien cancela o hay algún imprevisto de último momento.	No. Lo que se busca es que se generen eventos deportivos, no un evento con 4 personas en reserva, sino dos eventos y que todos participen.
Chat ente usuarios	Los usuarios podrán comunicarse con otros usuarios dentro de la aplicación	Opcional. Interesante, pero se correría el riesgo de convertirse en una aplicación de ligar.

Tabla 2. Características presentes en las diferentes aplicaciones analizadas.

Característica	IF7Sports	Fubles	PadelTrack	MeetUp	Timpik
Interfaz amigable	✓	✓	✓	✓	✓
Envío de notificaciones	✓	✗	✓	✓	✓
Multideportivo	✗	✗	✓	✓	✓
Dependencia de red social	✗	✗	✓	✗	✗
Creación de eventos propios	✗	✓	✓	✓	✓
Publicidad	✓	✗	✗	✗	✗
Independencia de idioma o país	✗	✗	✓	✓	✓
Filtrado de eventos creados e inscritos	✓	✗	✓	✓	✓
Estadísticas de partido	✗	✓	✗	✗	✓
Eventos gratuitos	✗	✗	✗	✓	✗
Eventos de pago	✓	✓	✓	✓	✓
Ranking	✗	✓	✗	✗	✓
Creaciones de grupos privados	✗	✗	✓	✗	✓
Permitir crear cualquier tipo de evento	✗	✗	✗	✓	✗
Perfil de usuario con fotografía	✗	✗	✗	✓	✓
Clasificación de niveles	✗	✗	✓	✗	✓
Valoración post-partido	✗	✓	✓	✗	✓
Lista de espera a los eventos	✓	✗	✓	✗	✗
Chat ente usuarios	✗	✗	✓	✓	✓

Gracias a la tabla anterior podemos observar que nos encontramos con tres tipos de características:

- I. Las características comunes a todas o casi todas las aplicaciones, aquellas que son casi imprescindibles para que este tipo de app puedan tener éxito en el mercado

y que nosotros también consideramos necesarias. Estas características son la interfaz amigable, las notificaciones, creación de eventos propios o a medida y multideportivo.

- II. Tenemos después un grupo de características que son aquellas que cada aplicación las añade o no dependiendo sus necesidades o el mercado al que desea apuntar. Independencia de país o región, valoración post-partido, chat entre usuarios, rankings, etc. Muchas de estas características permiten a la aplicación profundizar en un aspecto en concreto, pero en contra partida puede estar cerrándose mercado como el hecho de enfocar la aplicación solamente para una ciudad, ya que facilitaría mucho la práctica de deporte en dicha ciudad, pero estarías perdiendo mercado con el resto de las ciudades y usuarios.
- III. El último grupo son aquellas características que las poseen muy pocas apps o casi ninguna, donde estas características creemos que lo que hacen es limitar más que beneficiar a la aplicación. Como, por ejemplo, la introducción de publicidad o dependencia de red social donde el tener que hacer uso de una red social para utilizar la aplicación puede resultar un poco tedioso. También encontramos la permisión de crear eventos de cualquier índole, lo que puede desvirtuar claramente la naturaleza de la aplicación.

Por lo tanto, después del análisis realizado creemos que nuestra aplicación tomará varias características comunes, como las notificaciones, la interfaz amigable, multideportiva, permitiremos crear eventos propios con día y fecha a medida. También nos mantendremos al margen de colocar publicidad en la aplicación como muchas de las aplicaciones anteriores hacen. No estaremos atados a la necesidad de utilizar otra app o una red social para hacer uso de nuestra aplicación. Donde buscaremos diferenciarnos será en la parte económica. Ya que nuestra aplicación será implementada en un centro o en instalaciones deportivas, los usuarios que tengan acceso a ese centro o instalaciones no deberán pagar para hacer uso de nuestra aplicación.

2.3. Propuesta

Como hemos visto en el punto anterior, nuestra aplicación tiene similitudes y diferencias con varios proyectos del mismo espacio de conocimiento. Lo que buscamos con nuestra aplicación es poder ayudar a los usuarios a realizar sus deportes de interés, tratando de englobar todos en una sola aplicación, ya que como hemos visto hay varias que solamente se centran en un deporte en concreto. También trataremos de evitar que nuestra aplicación se convierta en una aplicación de citas para conocer gente y se mantenga en la vía de la práctica del deporte.

Para concluir, me gustaría destacar que nuestra aplicación tiene una gran similitud con Timpik a la hora de gestionar los eventos, permitiendo a los usuarios establecer la hora y la fecha de manera que más les convenga.

3. Análisis del problema

Una vez llegado a este punto, ya hemos visto las características que debería de tener nuestra aplicación y las diferencias o similitudes con otras; así que ahora nos centraremos en analizar de manera formal toda la información necesaria para el desarrollo de nuestro proyecto. Para realizar dicho análisis utilizamos la estructura de especificación y gestión de requisitos que se define en el estándar IEEE 29148:2018 [3] donde mencionaremos las diferentes características de nuestro sistema y terminaremos presentando los diagramas de casos de uso, de clases y de base de datos con sus respectivas explicaciones.

3.1. Especificación de requisitos

3.1.1. Contexto de la aplicación

En este capítulo trataremos de asegurar el entendimiento de los términos y conceptos más relevantes en el contexto de aplicación del sistema a realizar, recogeremos las ontologías del dominio y los tecnicismos utilizados a la hora de describir el sistema. Por claridad, he separado los conceptos ontológicos relevantes del dominio de los términos técnicos utilizados a nivel de implementación.

Ontología del sistema

En la tabla 3, encontraremos los términos ontológicos más relevantes del dominio:

Tabla 3. Términos ontológicos relevantes del dominio.

Término	Descripción
Usuario	Persona que accede a la aplicación teniendo una cuenta creada y puede hacer uso de las funciones de la App.
Evento	Partido organizado por un Usuario con el fin de practicar un deporte en particular y que tendrá unos participantes que se irán añadiendo.
Deporte	Nombre del deporte a practicar y una lista de reservas.
Reserva de usuario	Es una lista de reservas que el usuario tiene.

Terminología técnica

A continuación, tendremos la tabla 4 donde veremos los términos técnicos utilizados a nivel de implementación:

Tabla 4. Términos técnicos en la implementación.

Término	Descripción
Nombre de usuario	Nombre que el usuario elige para representarse en la aplicación.
Email	Correo electrónico que el usuario utilizará para registrarse en la aplicación.
Contraseña	Combinación de caracteres secretos que un usuario utiliza para ingresar en la aplicación.
Perfil	Información que tendrá el usuario en la aplicación, nombre e intereses.
Intereses	Lista de deportes que el usuario selecciona para recibir avisos y/o crear eventos sobre estos.
Reserva	Evento al que el usuario se suscribe para poder participar.
Hora	Hora a la que es usuario establece el evento.
Fecha	Fecha en la que el usuario establece el evento.
Pista	Lugar que el usuario selecciona para practicar el evento.
Miembros	Número de personas que hay inscritas en el evento.

3.1.2. Modelo de dominio

Definida así la terminología básica que se utilizará en el contexto del sistema a desarrollar, es relevante especificar también las relaciones existentes entre los conceptos. Esta relación se muestra en el modelo de dominio representado a través del diagrama de clases que aparece en la ilustración 9:

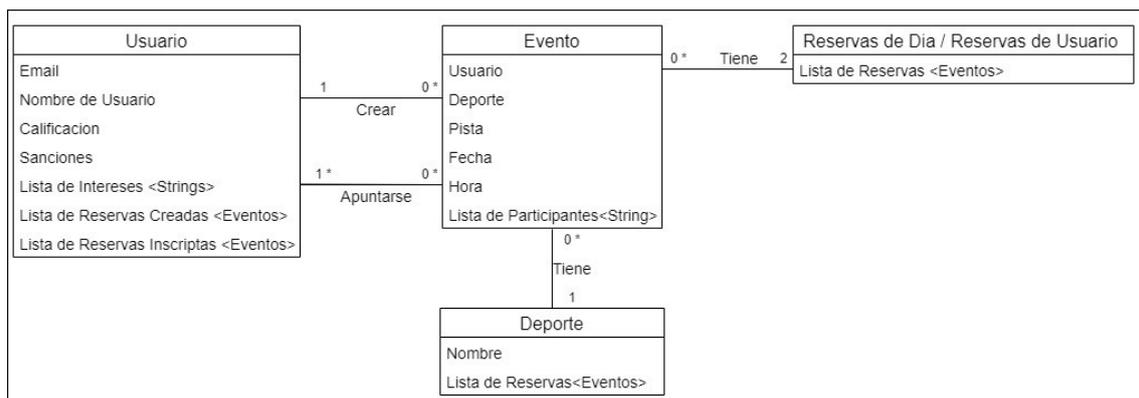


Ilustración 9. Diagrama de clases

Teniendo la imagen del diagrama como referencia pasaremos a analizar las clases y atributos que aparecen en ella, para ello expondremos una breve explicación para cada una y también tendremos las tablas 5-6-7-8 que darán una descripción más detallada de las diferentes clases.

- A. Clase usuario: representará cada uno de los usuarios creados en el sistema. Los usuarios creados dispondrán de estos atributos.

Tabla 5. Descripción de la clase usuario.

Atributo	Descripción	Tipo
Email	Email único para cada usuario.	String
Nombre de usuario	Nombre de usuario único.	String
Calificación	Valoración recibida por los otros usuarios al finalizar un evento.	Int
Sanciones	Número que determina si el usuario está sancionado o no.	Int
Lista de intereses	Lista de deportes de interés del usuario.	List <String>
Lista de reservas creadas	Lista de los eventos que ha creado el usuario.	List <Eventos>
Lista de reservas inscritas	Lista de las reservas que el usuario ha realizado.	List <Eventos>

- B. Clase evento: representará cada uno de los eventos que algún usuario cree.

Tabla 6. Descripción de la clase evento.

Atributo	Descripción	Tipo
Usuario	Es el usuario creador del evento.	String
Deporte	Nombre del deporte que describe el evento.	String
Pista	Nombre de la ubicación del evento.	String
Fecha	Fecha en la que tendrá lugar el evento.	Date
Hora	Hora en la que el evento se realizará.	Date
Lista de participantes	Lista de los usuarios que participarán.	List <String>

- C. Clase deporte: representará un deporte al que los usuarios podrán crear eventos e inscribirse.

Tabla 7. Descripción de la clase deporte.

Atributo	Descripción	Tipo
Nombre	Nombre del deporte.	String
Lista de reservas	Lista de eventos que se van creando en este deporte.	List <Evento>

- D. Clase Reserva día / Reserva Usuario: representará una reserva creada por un usuario en un deporte específico.

Tabla 8. Descripción de la clase reserva día.

Atributo	Descripción	Tipo
Lista de reservas	Lista de eventos que se añadirán a cada deporte.	List <Evento>

3.1.3. Límites del sistema

Un diagrama de contexto es una representación gráfica de alto nivel que muestra los actores y sus interfaces con el sistema. Dentro de este apartado mostraremos un diagrama de contexto que aparecerá en la ilustración 10, donde describirá los límites del sistema a implementar y se cementará a alto nivel los componentes que lo forman.

Diagrama de contexto:

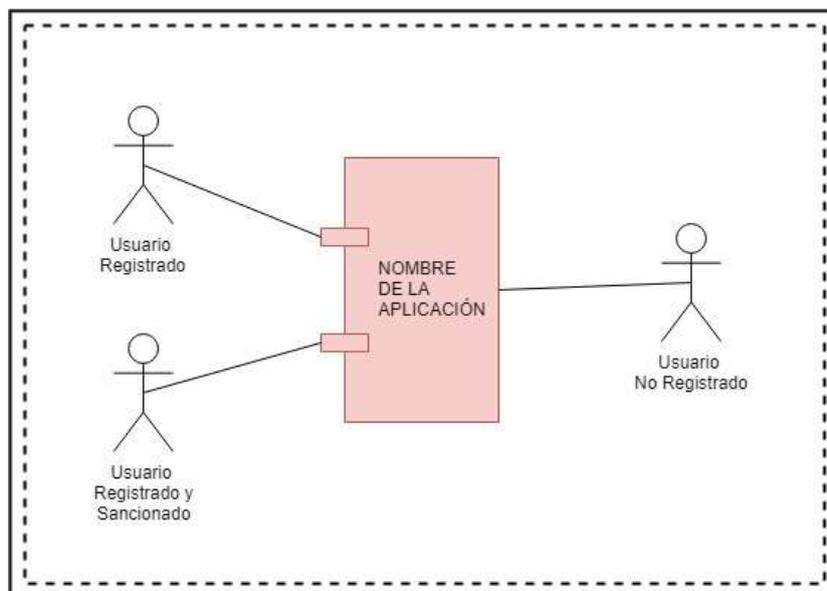


Ilustración 10. Límites y actores del sistema

Descripción de los actores:

- Usuario no registrado: es toda aquella persona que accede a la aplicación sin tener una cuenta creada, por lo tanto, solo tendrá acceso al Login y a la página de Registro.
- Usuario registrado: posee el acceso sin restricción a la aplicación, puede hacer uso de todas las funciones del sistema.
- Usuario registrado y sancionado: dicho usuario tiene limitaciones en el uso de la aplicación debido a poseer una sanción por no presentarse a un evento. No podrá crear ni inscribirse a eventos hasta que termine su sanción.

3.1.4. Características del sistema

Llegados a este punto del documento, con toda la información que tenemos podemos centrarnos en las características que el sistema debe implantar para cumplir nuestros objetivos.

Gestión de usuario:

Con lo que respecta al usuario, éstas son las demandas para cubrir por nuestro sistema:

- Crear usuario
- Editar usuario

Gestión de eventos:

Se detallan las características relacionadas con los eventos deportivos.

- Crear evento deportivo
- Borrar evento deportivo
- Consultar evento deportivo

Gestión de reservas:

- Inscribirse a una reserva
- Borrarse de una reserva

3.1.5. Restricciones y requisitos específicos

Nuestro sistema sufre una serie de restricciones que vienen provocadas por la propia esencia de la aplicación. A continuación, veremos la tabla 9 donde se detallarán las restricciones.

Tabla 9. Restricciones del sistema.

Restricción	Descripción
Conexión a internet	Para que la aplicación pueda funcionar debe de tener conexión a internet para permitir al usuario contactar con la base de datos y obtener los eventos creados que existan y añadir los posibles eventos que él cree.
Cuenta de correo electrónico	Para poder hacer uso de la aplicación es necesario que el usuario tenga una cuenta de correo electrónico para poder registrarse.

Seguiremos analizando los requisitos funcionales, no funcionales y específicos de nuestro proyecto.

Requisitos funcionales:

- Se podrán crear eventos de deportes específicos.
- Se podrán crear eventos deportivos en una fecha y hora específica.
- Se recibirán notificaciones instantáneas de nuevos eventos deportivos creados en los deportes de interés del usuario.
- El sistema actualizará automáticamente el rango de días en el que se puede crear eventos.
- Habrá la opción de búsqueda de eventos filtrando por deporte.
- Existirá la opción de listar todos los eventos en lo que se participe.
- Se permitirá borrar eventos o darse de baja de los que esté el usuario apuntado.

Requisitos no funcionales:

- Se almacenarán los datos utilizando la plataforma gestora de base de datos.
- Ofrecerá una interfaz sencilla e intuitiva para facilitar el uso.

- Se hará uso del almacenamiento interno del teléfono para guardar los datos del usuario mientras tiene iniciada la sesión.

Requisitos específicos:

- Adaptación al dispositivo: la aplicación debe de adaptarse a diferentes teléfonos con diferentes resoluciones.
- Interfaz simple e intuitiva: el usuario tiene que poder utilizar la aplicación sin tener la necesidad de perder tiempo aprendiendo como funciona.
- Disponibilidad: la aplicación deberá de estar disponible 24/7 para su uso, permitiendo realizar las acciones en cualquier momento que el usuario disponga de internet.
- Seguridad: nuestra aplicación debe garantizar que los datos que se utilizan estén seguros y no puedan ser obtenidos por terceros.

3.1.6. Casos de uso

Después de detallar las características encontradas, pasamos a estudiar los casos de usos que hemos detectado, detallaremos los actores encontrados con las actividades que realizan. Cada actividad tendrá un identificador para luego poder comentar más detalladamente su función en una tabla de especificación y en los Mock-ups.

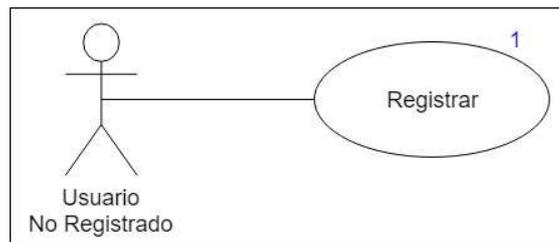


Ilustración 11. Usuario no registrado

Tabla 10. Caso de uso registrar.

Caso de uso	1
Nombre	Registrar
Descripción	El usuario debe de poder registrarse en la aplicación introduciendo los datos necesarios (nombre de usuario, email y contraseña).
Actor	Usuario no registrado.
Flujo básico	El caso de uso se inicia cuando el usuario pulsa el botón de “Registrar”.
Flujo alternativo	<ul style="list-style-type: none"> • Se comprueba que los campos nombre de usuario, email y contraseña no estén vacíos. • Se comprueba que el nombre de usuario no se esté utilizando ya. • Se comprueba que el email no esté registrado ya. • Se comprueba que la contraseña tengo al menos 8 caracteres alfanuméricos.
Precondición	Ninguna
Postcondición	Se guardarán los datos del usuario en la base de datos y en el sistema.

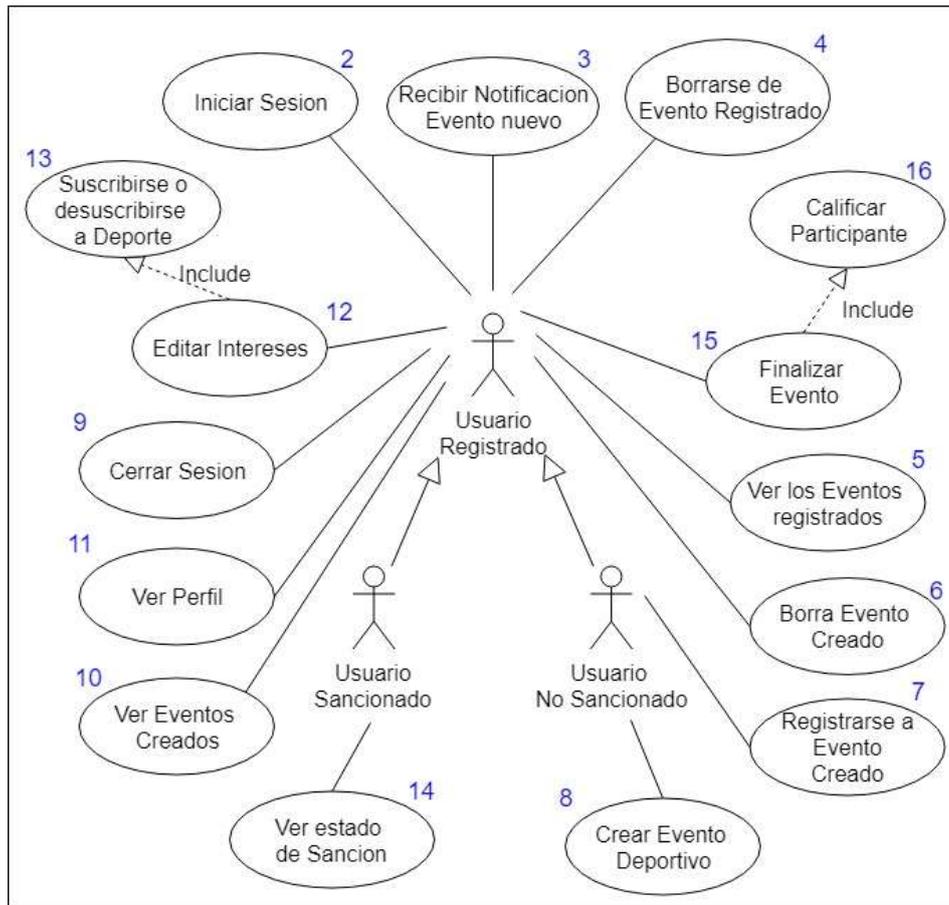


Ilustración 12. Usuario registrado en el sistema

Tabla 11. Caso de uso iniciar sesión.

Caso de uso	2
Nombre	Iniciar sesión
Descripción	El usuario tiene que introducir su email y su contraseña para poder entrar en la aplicación.
Actor	Usuario registrado.
Flujo básico	El caso de uso se inicia cuando el usuario pulsa el botón de "Login"
Flujo alternativo	<ul style="list-style-type: none"> Se comprueba que los campos email y contraseña no estén vacíos. Se comprueba que el email exista en la base de datos. Se comprueba que el email y contraseña corresponden a un usuario del sistema.
Precondición	El usuario debe de estar registrado.
Postcondición	<ul style="list-style-type: none"> La aplicación redirigirá a la pantalla principal. La aplicación cargará los datos del usuario. Se activarán las suscripciones de los deportes de interés para el usuario registrado.

Tabla 12. Caso de uso recibir notificación evento nuevo.

Caso de uso		3
Nombre	Recibir notificación evento nuevo	
Descripción	El usuario recibe notificaciones de la creación de nuevos eventos en sus deportes de interés.	
Actor	Usuario registrado.	
Flujo básico	El caso de uso se inicia cuando otro usuario crea un nuevo evento en un deporte del interés del usuario que está registrado en la app.	
Flujo alternativo	<ul style="list-style-type: none"> Se envía una notificación a los usuarios que tengan interés en el evento creado. 	
Precondición	<ul style="list-style-type: none"> El usuario deberá de tener seleccionado algún deporte de interés. La aplicación no debe de estar en primer plano. 	
Postcondición	<ul style="list-style-type: none"> La aplicación mostrará una notificación en la barra de notificaciones. 	

Tabla 13. Caso de uso borrarse de un evento registrado.

Caso de uso		4
Nombre	Borrarse de evento registrado	
Descripción	El usuario podrá darse de baja de los eventos que esté apuntado.	
Actor	Usuario registrado.	
Flujo básico	El caso de uso se inicia cuando el usuario hace click en el botón “Cancelar” de uno de los eventos de la lista de eventos registrado.	
Flujo alternativo	<ul style="list-style-type: none"> Ninguno 	
Precondición	<ul style="list-style-type: none"> El usuario deberá de estar registrado en algún evento. 	
Postcondición	<ul style="list-style-type: none"> Se actualizarán los datos en la base de datos del sistema. 	

Tabla 14. Caso de uso ver los eventos registrados.

Caso de uso		5
Nombre	Ver los eventos registrados	
Descripción	El usuario podrá ver todos los eventos en los que se encuentre registrado.	
Actor	Usuario registrado.	
Flujo básico	El caso de uso se inicia cuando el usuario selecciona el deporte para el cual quiere buscar sus eventos registrados y presiona el botón “buscar”.	
Flujo alternativo	<ul style="list-style-type: none"> El sistema comprueba que el usuario esté registrado en algún evento. Si el usuario no está en ningún evento se mostrará una lista vacía. 	
Precondición	<ul style="list-style-type: none"> El usuario deberá de estar registrado en algún evento. 	
Postcondición	<ul style="list-style-type: none"> La aplicación mostrará la información de los eventos en pantalla. 	

Tabla 15. Caso de uso borrar evento creado.

Caso de uso	6
Nombre	Borrar evento creado
Descripción	El usuario puede borrar los eventos que el creó anteriormente.
Actor	Usuario registrado.
Flujo básico	El caso de uso se inicia cuando el usuario hace click en el botón de “Borrar” que habrá en cada uno de los eventos creados por el usuario.
Flujo alternativo	<ul style="list-style-type: none"> • Ninguno
Precondición	<ul style="list-style-type: none"> • El usuario deberá de tener eventos creados.
Postcondición	<ul style="list-style-type: none"> • Se actualizarán los datos en la base de datos del sistema.

Tabla 16. Caso de uso registrarse en evento creado.

Caso de uso	7
Nombre	Registrarse en evento creado
Descripción	El usuario podrá buscar eventos de sus deportes de interés y registrarse en ellos para poder participar.
Actor	Usuario no registrado.
Flujo básico	El caso de uso se inicia cuando el usuario selecciona el deporte para el cual quiere buscar sus eventos registrados, el día que quiere practicar deporte y presiona el botón “buscar” en alguno de los eventos que le aparecen.
Flujo alternativo	<ul style="list-style-type: none"> • Ninguno
Precondición	<ul style="list-style-type: none"> • El usuario deberá de tener algún deporte de interés.
Postcondición	<ul style="list-style-type: none"> • Se actualizarán los datos en la base de datos del sistema.

Tabla 17. Caso de uso crear evento deportivo.

Caso de uso	8
Nombre	Crear evento deportivo
Descripción	El usuario tiene la capacidad de crear eventos deportivos estableciendo día, hora y pista.
Actor	Usuario no sancionado.
Flujo básico	El caso de uso se inicia cuando el usuario selecciona el deporte para el cual quiere buscar sus eventos registrados, el día que quiere practicar deporte, la hora y presiona el botón “buscar” en alguno de los eventos que le aparecen.
Flujo alternativo	<ul style="list-style-type: none"> • Comprobar que no tiene un evento para el mismo día. • Comprobar que no tiene más de una hora seleccionada. • Comprobar que tiene al menos una hora seleccionada.
Precondición	<ul style="list-style-type: none"> • El usuario deberá de tener algún deporte de interés.
Postcondición	<ul style="list-style-type: none"> • Se actualizarán los datos en la base de datos del sistema.

Tabla 18. Caso de uso cerrar sesión.

Caso de uso		9
Nombre	Cerrar sesión	
Descripción	Todos los usuarios tienen la opción de cerrar su sesión tanto para salir de la aplicación y entrar con otro usuario o para cerrarla simplemente.	
Actor	Usuario registrado.	
Flujo básico	El caso de uso se inicia cuando el usuario hace click en el botón de Logout.	
Flujo alternativo	<ul style="list-style-type: none"> • Ninguno. 	
Precondición	<ul style="list-style-type: none"> • El usuario deberá de tener algún deporte de interés. 	
Postcondición	<ul style="list-style-type: none"> • La aplicación te redirigirá a la pantalla de login. • Se desactivarán las suscripciones del usuario para este dispositivo. 	

Tabla 19. Caso de uso ver eventos creados.

Caso de uso		10
Nombre	Ver eventos creados	
Descripción	Todos aquellos usuarios que hayan creado eventos podrán verlos y comprobar cuantos miembros se han inscrito.	
Actor	Usuario registrado.	
Flujo básico	El caso de uso se inicia cuando el usuario escoge el deporte del cual desea ver los eventos creados y pulsa el botón “buscar”.	
Flujo alternativo	<ul style="list-style-type: none"> • Ninguno. 	
Precondición	<ul style="list-style-type: none"> • El usuario deberá de tener algún evento creado por él. 	
Postcondición	<ul style="list-style-type: none"> • La aplicación mostrará en pantalla la información cargada de la base de datos. 	

Tabla 20. Caso de uso ver perfil.

Caso de uso		11
Nombre	Ver perfil	
Descripción	El usuario tiene que ser capaz de ver su perfil con la información que tiene añadida.	
Actor	Usuario registrado.	
Flujo básico	El caso de uso se inicia cuando el usuario pulsa el icono de “perfil de usuario” en la pantalla principal.	
Flujo alternativo	<ul style="list-style-type: none"> • Ninguno. 	
Precondición	<ul style="list-style-type: none"> • Ninguna. 	
Postcondición	<ul style="list-style-type: none"> • La aplicación mostrará en pantalla la información cargada de la base de datos. 	

Tabla 21. Caso de uso editar intereses.

Caso de uso		12
Nombre	Editar intereses	
Descripción	El usuario tiene la capacidad de poder editar sus deportes de interés tanto sea añadiendo nuevos como eliminando existentes.	
Actor	Usuario registrado.	
Flujo básico	El caso de uso se inicia cuando el usuario pulsa el botón “editar” en la pantalla perfil y finaliza cuando pulsa el botón “guardar” en la pantalla editar perfil.	
Flujo alternativo	<ul style="list-style-type: none"> • Ninguno. 	
Precondición	<ul style="list-style-type: none"> • Ninguna. 	
Postcondición	<ul style="list-style-type: none"> • Se actualizarán los datos en la base de datos del sistema. • Se activarán o desactivarán las suscripciones a los deportes añadidos o borrados. 	

Tabla 22. Caso de uso inscribirse o desinscribirse en algún deporte.

Caso de uso		13
Nombre	Inscribirse o desinscribirse en algún deporte	
Descripción	Cuando el usuario modifica su perfil editando sus intereses, éste se está suscribiendo a recibir notificaciones de ese deporte o dejar de recibir notificación de aquellos deportes que ya no son de su interés.	
Actor	Usuario registrado.	
Flujo básico	El caso de uso se inicia cuando el usuario realiza algún cambio en sus deportes de interés y pulsa el botón “guardar”.	
Flujo alternativo	<ul style="list-style-type: none"> • Ninguno. 	
Precondición	<ul style="list-style-type: none"> • El usuario realizó algún cambio en los deportes de interés del usuario. 	
Postcondición	<ul style="list-style-type: none"> • Se activarán o desactivarán las suscripciones a los deportes añadidos o borrados. 	

3.1.7. Análisis de riesgos

Dentro de este apartado nos centraremos en los posibles riesgos que tiene el desarrollo de nuestro proyecto, se tendrán en cuenta los posibles riesgos que puedan afectar al usuario o a la infraestructura. Para facilitar la lectura, hemos decidido ordenar los diferentes riesgos en tablas (24-25-26) y proporcionar una descripción para cada uno de ellos.

Tabla 23. Creación de usuarios falsos.

Titulo	Creación de usuarios falsos
Descripción	Podríamos sufrir la creación de usuarios con el fin de apuntarse a un evento y no asistir a él con el objetivo de sabotear o fastidiar a otros usuarios.
Impacto	Los usuarios perjudicados empezarían a dejar de usar la aplicación, se transmitiría la mala experiencia y se crearía una mala reputación para nuestra app.
Medidas	Utilizar la cuenta de Gmail ubicada en el mismo dispositivo asegurando que son cuentas reales y no un email creado para ese fin.

Tabla 24. Saturación del sistema.

Titulo	Saturación del sistema
Descripción	El sistema recibe una gran cantidad de datos tantos en base a usuarios nuevos, creación de nuevos eventos como envío de notificaciones.
Impacto	Crea un retraso en las comunicaciones de notificaciones llegando a producir una caída del sistema.
Medidas	Utilizar las herramientas que nos brinda Firebase para notificarnos del consumo de recursos y estado del sistema.

Tabla 25. Gestión de infraestructura de base de datos por parte de terceros.

Titulo	Gestión de la infraestructura de base de datos por parte de terceros
Descripción	Nuestro gestor de la infraestructura de base de datos podría sufrir problemas tanto de mantenimiento como de errores.
Impacto	Nuestra aplicación se vería sin servicio lo que supondría la inutilidad de la aplicación.
Medidas	Utilizar los servicios de buckup para la base de datos para asegurar la disponibilidad continua de la aplicación.

3.2. Identificación y análisis de soluciones posibles

Después de haber analizado todas nuestras necesidades y analizado los posibles problemas, pasaremos a comentar las posibilidades que hemos encontrado para implementar nuestro proyecto.

Para la elección de las tecnologías a utilizar se realizó una investigación de las posibilidades que teníamos a la hora de gestionar conjuntamente o combinar el acceso a base de datos, registro, autenticación, autorización, envío de notificación push, gestión y mantenimiento de servidor.

Durante nuestra búsqueda de la tecnología adecuada, se nos planteaban tres grandes opciones, una era buscar una herramienta que nos pudiera proporcionar todos los servicios que necesitábamos, la segunda era combinar diferentes herramientas que nos proporcionaran varios servicios juntos, pero no todos. Y, por último, teníamos la opción de abordar cada necesidad que teníamos de manera separada.

Lo primero que teníamos en mente era la última opción, utilizar para cada una de nuestras necesidades una herramienta diferente, herramientas en la que ya estaba familiarizado por trabajar durante la carrera o en proyectos por mi cuenta. Para esta idea pensamos en crear un servidor utilizando la tecnología API Rest con peticiones HTTP a una base de datos en MongoDB. Dentro de este servidor se debería de implementar de manera manual la utilización del token para la autorización y la autenticación. Hasta aquí esta opción no suponía un problema en la parte de implementación, ya que era una tecnología que se conocía y eran cosas que había realizado anteriormente, pero cuando encaramos el tema del envío de notificación vimos que era un tema no muy sencillo de implementar de manera manual.

Buscando diferentes opciones para el envío de notificaciones push, encontramos las herramientas Firebase y Notificare, que te permiten hacer uso de este servicio de manera muy intuitiva y que te facilitaban toda la documentación para poder implementarlo. Con lo que respecta a la parte de satisfacer el servicio, las dos opciones cumplían con los requisitos y eran nativas para Android, la mayor diferencia fue que Firebase es gratis y que Notificare es completamente de pago.

Cuando empecé a leer la documentación de Firebase [4], a leer información de ella en internet, la documentación y ver videos sobre sus servicios, encontré que esta herramienta englobaba todas mis necesidades en una sola tecnología, trabaja de manera nativa en Android [5] y, además, muchas cosas son transparentes para el desarrollador. Firebase se encarga de manera interna de muchas tareas liberándonos de dicha labor como por ejemplo el uso de tokens para la seguridad de la app.

Firebase también nos brinda los servicios de base de datos NoSQL que era nuestra primera opción. Hay que recordar que las bases de datos NoSQL son una forma de almacenamiento no estructurado, no existe una estructura de tabla fija como en las bases de datos SQL, sino que se organizan normalmente mediante documentos. En Firebase no hay ni tablas ni filas ni esquemas, la información se almacena en documentos que se organizan en colecciones. Los documentos en Firebase son la unidad de almacenamiento, son un registro liviano que contiene campos con valores asignados y las colecciones son solamente contenedores de dichos documentos. Pero como todo, las bases de datos NoSQL tienen ventajas y desventajas [2].

Estas son las ventajas que nos brindan:

- I. Versatilidad: teniendo en cuenta que se utiliza una notación JSON en estas bases de datos, si tuviéramos que agregar un nuevo campo sobre nuestra colección simplemente se agregan sobre el documento y el sistema seguirá operando sin tener que realizar otras configuraciones.
- II. Disponibilidad de recursos: no requieren gran cantidad de recursos en los servidores para operar, se puede comenzar con una cantidad de recursos bajos y



se pueden ir aumentando, dependiendo las necesidades sin tener que detener los servicios.

- III. Crecimiento horizontal: soportan estructuras distribuidas, si es necesario instalar una mayor cantidad de nodos para ampliar la capacidad, se puede hacer sin interrumpir la usabilidad o consultas.
- IV. Optimización: las bases de datos NoSQL no requieren de algoritmo para reescribir las consultas escritas por el programador, esto se hace con el fin de no sobrecargar el rendimiento de servidores y mantener un nivel óptimo en las operaciones.

A pesar de las ventajas mencionadas, las bases de datos NoSQL tienen ciertas desventajas:

- I. Atomicidad: algunas de estas bases de datos no ofrecen la atomicidad en la información, esto puede provocar que la información no sea consistente entre nodos.
- II. Estándares en el lenguaje: no existe un estándar definido entre los diferentes motores que ofrece dicho servicio. El lenguaje o manejo de objetos varía dependiendo del tipo de base de datos (MongoDB, DB2).
- III. Documentación software: al estar hablando sobre algo relativamente nuevo, podremos encontrarnos con que las operaciones sobre la información sean limitadas debido a la falta de información sobre las herramientas.
- IV. Herramientas GUI: las herramientas que brindan soporte o administran estas bases de datos no disponen de interfaz gráfica, suelen tener acceso por consola.

3.3. Solución propuesta

Una vez llegados a este punto del documento, nos encontramos con que sabemos lo que queremos, lo que necesitamos y las opciones que tenemos para llevarlo a cabo. Así que para desarrollar este proyecto tomamos la decisión de utilizar GitHub como herramienta de control de versiones, Android Studio como IDE de programación, Java como lenguaje y Firebase como herramienta que nos proporcionará todos los servicios que necesitamos de manera práctica y nativa para Android. La manera de trabajar con la base de datos que tiene Firebase se adapta perfectamente a nuestras necesidades, las colecciones y documentos nos permiten organizar y guardar la información de una manera que nos ayudará a operar con ella, permitirá almacenar directamente objetos específicos (usuario, reservas, etc.) brindándonos sencillez a la hora de trabajar con nuestros datos. También es la mejor opción para trabajar con las notificaciones que recibirán los diferentes usuarios.

Lo primero que haremos será crear nuestro proyecto tanto en Android Studio, Firebase como GitHub y vincular los tres. Una vez esté todo conectado, empezaremos por crear la base de datos y ver cómo realizar las peticiones y trabajar con los datos recibidos. El siguiente paso será implementar el registro y la autenticación en la aplicación. A continuación, tendrá lugar la programación de la recepción de notificaciones en los dispositivos. También iremos implementando funcionalidades de la aplicación de

manera paralela, tratando de ir dejando terminado cada caso de uso. Por último, una vez que toda la funcionalidad esté implementada y la hayamos ido validando a medida que se desarrollaba, nos centraremos en implementar todo lo que va relacionado con el diseño, colores y tipografía, que estará definido en los Mockup realizados previamente. Una vez esté todo terminado, se realizarán unas pruebas finales, estas pruebas consistirán en la utilización de la aplicación por parte de diferentes usuarios para poder recibir sus comentarios y experiencias con el producto.

4. Diseño de la solución

En este capítulo del documento expondremos la arquitectura utilizada en el proyecto y el diseño realizado tanto para la base de datos como los módulos o clases utilizados. También tendremos un apartado para hablar sobre la tecnología utilizada.

4.1. Arquitectura del sistema

La arquitectura software que hemos utilizado en nuestro proyecto es la que aparece en la ilustración 13 que veremos a continuación:

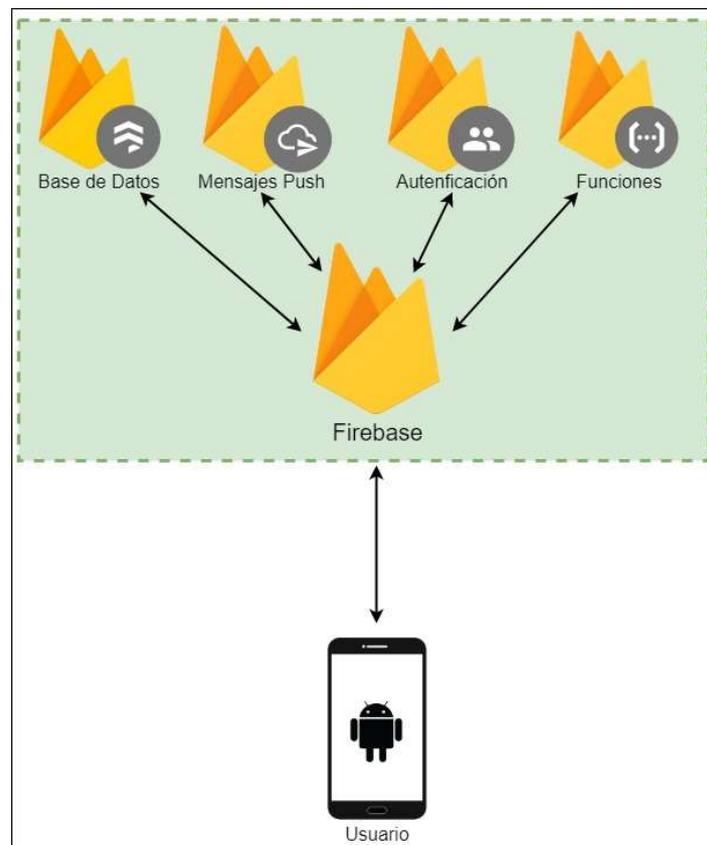


Ilustración 13. Arquitectura del sistema

Como podemos ver en la imagen, nuestra aplicación se divide en dos partes, el dispositivo móvil, donde tendremos nuestra aplicación instalada y será lo que el usuario puede ver y, por otra parte, tenemos todos los servicios que nos brinda Firebase, que se incluirán en el SDK de los dispositivos con lo que se gestionará la autenticación y los mensajes push que estarán condicionados por las funciones que se programen en Firebase. También el usuario tendrá acceso a la información de la base de datos utilizando Firebase.

4.2. Diseño detallado

Una vez llegados a este punto nos vamos a dedicar a comentar el diseño interno de nuestra aplicación, desde el patrón elegido, las diferentes clases y las relaciones entre ellas, así como también el esquema de nuestra base de datos.

Empezaremos por el patrón utilizado, en nuestro caso usamos el MVC, Modelo Vista Controlador con una arquitectura de tres capas. Las capas las hemos dividido de la siguiente manera:

- Vista: aquí se encuentra todo lo que está relacionado con la parte visible de la aplicación, conocido como frontend, lo que el usuario ve cuando usa nuestro programa (botones, textos, imágenes, listas, etc.) y donde se muestra la información que el usuario solicitó al controlador.
- Controlador: encargado de recibir las peticiones del usuario, las atiende y las procesa. Solicita los datos necesarios al Modelo para obtener los resultados que se le enviarán a la Vista.
- Modelo: es el encargado de gestionar, tratar y actualizar los datos. Es el que se comunica a la base de datos realizando las consultas, búsquedas, filtrados y actualizaciones.

La elección de este patrón fue debido a que nos da una gran libertad entre módulos, ya que podemos realizar modificaciones de una misma capa sin que el resto sufra cambios y, además, permite a nuestro proyecto ser más escalable.

En segundo lugar, hablaremos acerca de nuestras clases y de la relación que hay entre ellas. En nuestro proyecto podemos encontrar las siguientes clases:

- Activity: son todas aquellas clases que muestran la interfaz al usuario (HomeActivity, LoginActivity, etc.). Dentro de estas clases solamente encontraremos cosas referentes a la interfaz (botones, Labels, ListView, etc) y llamadas a los diferentes controladores (UsuarioControler, EventoControler)
- Controladores: aquí tenemos los controladores de Eventos y Usuarios, que se encargan de recibir las peticiones y devolver las respuestas haciendo las consultas necesarias a los Modelos (Día, Evento, Deporte) y a la base de datos (FirebaseInteraction)
- Modelo: aquí tenemos las entidades de Dominio (Usuario, Deporte, Evento) que también serán usadas a la hora de configurar la base de datos.



Para terminar, veremos en la ilustración 14 un esquema de nuestra base de datos con su respectiva explicación de las partes que la componen. Debemos tener en cuenta que es una base de datos NoSQL.

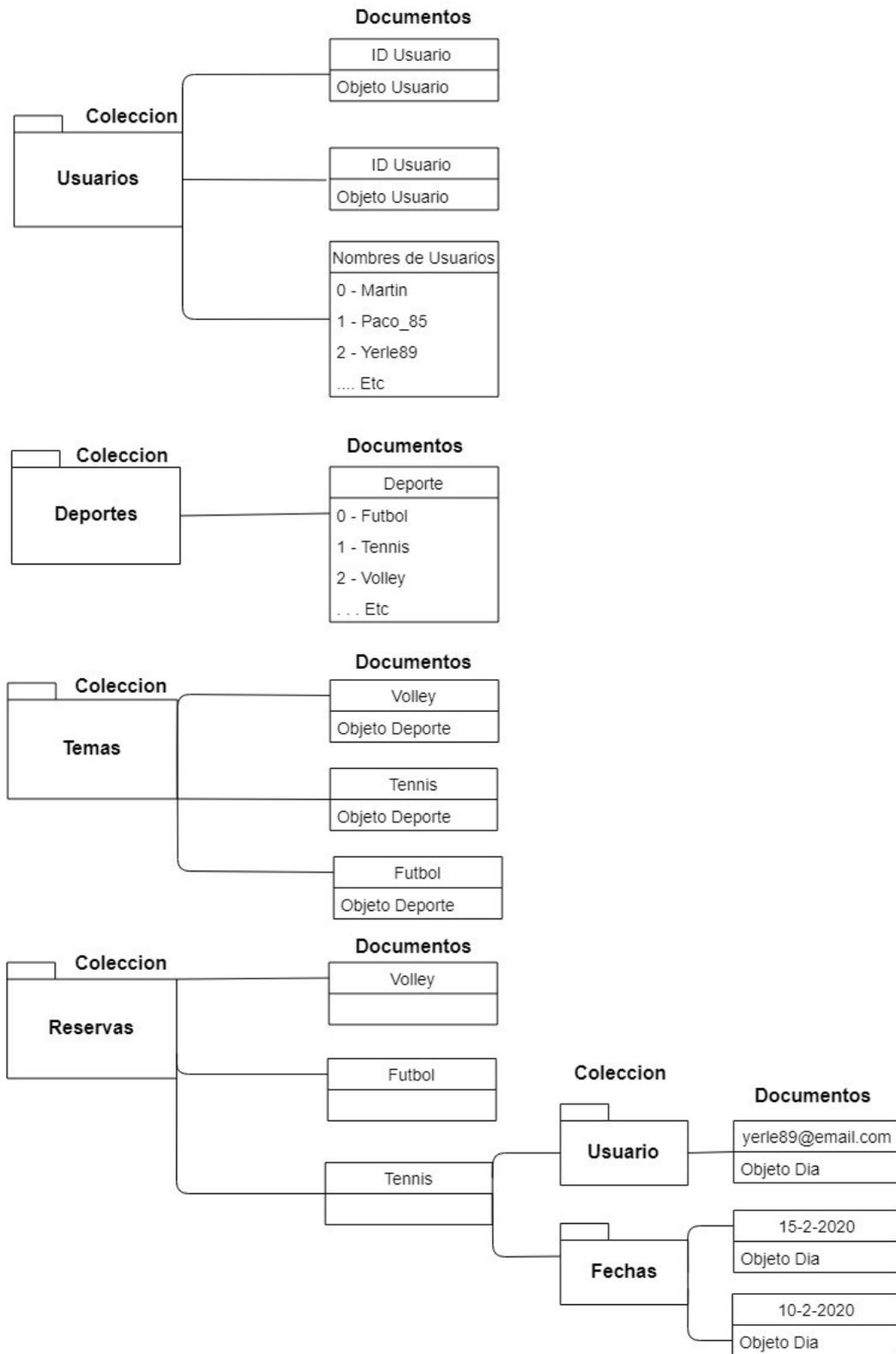


Ilustración 14. Esquema de la base de datos

Una vez visto el esquema de nuestra base de datos en la imagen anterior pasamos a explicarla para poder entenderla mejor, hemos dividido nuestro proyecto 4 colecciones:

- Usuarios: dentro de esta colección tendremos un documento que contenga todos los nombres de usuarios que estén registrados y además un documento para cada usuario creado, el nombre de este documento será el email del usuario y contendrá un Objeto de tipo Usuario con toda la información.
- Deportes: en esta colección encontraremos un documento denominado deporte que contendrá los nombres de todos los deportes que se podrán practicar.
- Temas: la colección temas almacenará documentos de tipo Deporte con el nombre de cada deporte.
- Reservas: con nuestra última colección gestionaremos las reservas, donde tendremos un documento para cada deporte, pero dentro de estos documentos haremos una división para facilitar y agilizar las búsquedas. La división que haremos será:
 - Fechas: esta colección ordenará las reservas realizadas en un día concreto facilitando la búsqueda por día.
 - Usuario: aquí vamos a encontrar la lista de reservas que un usuario realizó. Esto nos agiliza mucho a la hora de traer las reservas de cada usuario.

4.3. Tecnología utilizada

Dentro de este apartado presentaremos todas las herramientas utilizadas para llevar a cabo nuestro proyecto.

4.3.1. Firebase



Firebase es una plataforma digital desarrollada por Google que se usa en el desarrollo de aplicaciones web o móviles. Esta herramienta facilita el desarrollo de las aplicaciones de manera rápida, sencilla y efectiva. El objetivo principal de Firebase es mejorar el rendimiento de las aplicaciones gracias a las diversas funcionalidades que ofrece y que veremos a continuación.

- Es una herramienta multiplataforma, podemos integrarla con aplicaciones web, Unity, IOS y como en nuestro caso Android.
- Permite crear proyectos sin necesidad de un servidor, ya que los servicios que ofrece se incluyen en los SDK de los dispositivos móviles.
- Plataforma gratuita para la iniciación del proyecto y en una fase media, solamente se empieza a cobrar cuando el proyecto coge dimensiones avanzadas.

- Analytics: proporciona datos sobre el comportamiento y la experiencia de usuarios en nuestra aplicación.
- Ofrece dos diferentes tipos de base de datos en tiempo real (Cloud Firestore y RealTime Database), ambas son NoSQL, pero con pequeñas diferencias adaptándose según las exigencias del desarrollador, en nuestro caso utilizamos Cloud Firestore [5].
- Autenticación [3]: permite realizar la autenticación de la aplicación con código solamente del lado del cliente utilizando diferentes proveedores (Facebook, Google, email, etc.), permitiendo al desarrollador liberarse de la carga de crear métodos de autenticación.
- Ofrece los servicios tales como envío de mensajes [3] y notificaciones para los dispositivos [4], permitiendo notificar cambios en la base de datos, nuevos correos electrónicos de contacto, etc.
- Testeo de aplicaciones: Firebase nos da la herramienta Test Lab [5] que nos permite testear nuestras aplicaciones en la nube seleccionando diferentes configuraciones y diferentes datos, obteniendo los resultados en un registro y con capturas.

4.3.2. Android Studio

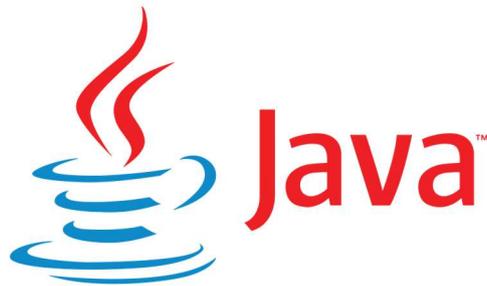


Para nuestro proyecto hemos elegido utilizar Android Studio como entorno de desarrollo. Este IDE está basado en IntelliJ IDEA y es el entorno de desarrollo oficial para aplicaciones Android [12]. Android Studio brinda muchas funciones que aumentan la productividad de los desarrolladores a la hora de crear aplicaciones, en nuestro caso las que más aprovechamos fueron:

- Emulador rápido que nos permite comprobar rápidamente cada cambio o implementación realizada.
- Sistema Flexible basado en Gradle.
- Permite integrar GitHub en tu proyecto para poder gestionar las versiones de tu proyecto.
- Ofrece la opción de aplicar cambios para insertar cambios de códigos y recursos a la aplicación en ejecución sin tener que reiniciar la aplicación.
- Compatibilidad total e integrada con Google Cloud Platform, que facilita en nuestro caso la utilización de Google Cloud Messaging.

Aparte de las funciones mencionadas, de todas las otras que nos ofrece y, porque es el IDE por excelencia para las apps en Android, también hemos elegido este IDE debido a que ya teníamos experiencia trabajando con él en diversos proyectos personales, universitarios y un curso de Android de la UPV.

4.3.3. Java



Para desarrollar nuestro proyecto, hemos elegido Java como lenguaje de programación, nuestra elección se basó básicamente en el conocimiento que poseemos sobre dicho lenguaje, tanto por la carrera como por cuenta propia y por las ventajas y facilidades que tendremos con el IDE a utilizar para dispositivos Android.

Java es un lenguaje de programación orientado a objetos, interpretado, compilado y tipado estáticamente. Hemos elegido Java como lenguaje para nuestro proyecto por los conocimientos y experiencia que tenemos en trabajar con él y además porque es simple, distribuido, robusto, seguro, multihilo y con garbage collector.

4.3.4. GitHub



A pesar de que en nuestro proyecto haya un solo desarrollador, siempre es conveniente disponer de un sistema de control de versiones, en nuestro caso elegimos GitHub. GitHub te permite almacenar en la nube tu proyecto y administrar tu código al igual que llevar un registro y control de cualquier cambio sobre dicho código. GitHub también posee otras funciones que, en nuestro caso, por la simpleza del equipo de desarrollo (1 persona) no fue necesario usar, pero no hay que olvidar todas las herramientas que brinda para poder llevar a cabo un proyecto utilizando la metodología ágil, con Kanban, sprint, tareas asignables, etc.

5. Desarrollo de la solución propuesta

En este capítulo del documento mostraremos como queda confeccionada la solución final representándola con sus respectivos Mock-ups y explicando cómo se implementaron las funcionalidades.

Empezamos con las funcionalidades de registro y login en la aplicación que las encontraremos en las ilustraciones 15 y 16, respectivamente.

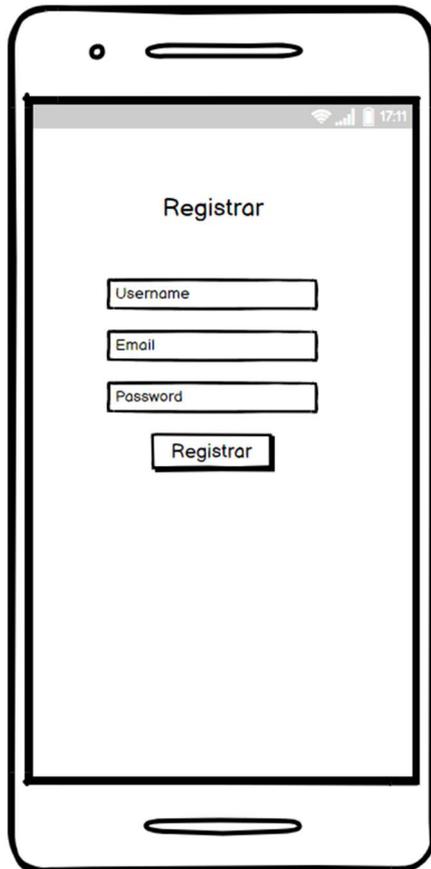


Ilustración 15. Pantalla registro



Ilustración 16. Pantalla login

Con estas dos imágenes vemos el diseño de la interfaz para ambas vistas y podemos observar que en las dos imágenes tenemos un botón que será el encargado de enviar los datos a Firebase. Firebase recibirá los datos que el usuario escribió en los diferentes campos. Si todo va correctamente, el usuario podrá registrarse o loguearse dependiendo de la ventana. En caso de que el usuario intente registrarse con un email ya utilizado o introduzca mal los datos para el login, el sistema le mostrará un cartel advirtiéndole del error.

Una vez que el usuario realizó el login, éste se encontrará en la pantalla principal, la cual tendrá un aspecto como el que aparece en la ilustración 17 y le dará acceso al resto de pantallas con sus respectivas funcionalidades.



Ilustración 15. Pantalla menú principal

En el menú principal el usuario podrá acceder o salir de la aplicación (desloguearse). También tendrá acceso a las pantallas de las funcionalidades restantes (Crear Evento, Mis Eventos Creados, Buscar Eventos, Mis Reservas Inscriptas) que las veremos en las siguientes imágenes.

- Perfil de usuario

En la ilustración 18 tendremos lo que el usuario verá cuando acceda a su perfil, donde tendrá su nombre de usuario y los intereses que haya elegido. Estos intereses son los que determinarán sobre qué deportes recibirá las notificaciones. Si el usuario desea añadir o eliminar algún deporte de su lista de interés, solamente deberá de ir a editar y seleccionar o deseleccionar el deporte deseado.

- Crear reserva

Para representar la funcionalidad de “Crear Reserva” podremos ver la ilustración 19 donde el usuario dispondrá de la siguiente pantalla donde podrá elegir el deporte a practicar, la pista, la fecha, hora y día donde tendrá lugar. Una vez tenga todo

seleccionado deberá de hacer click en el botón “Reservar” y mediante Firebase se añadirá el evento a la base de datos y se notificará a todos los usuarios que tengan el deporte del evento elegido como deporte de interés.

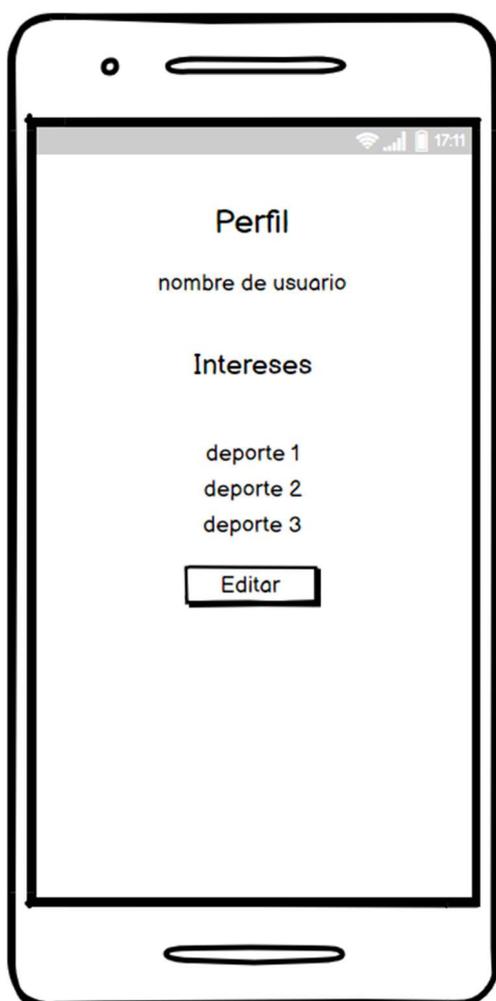


Ilustración 16. Pantalla perfil de usuario

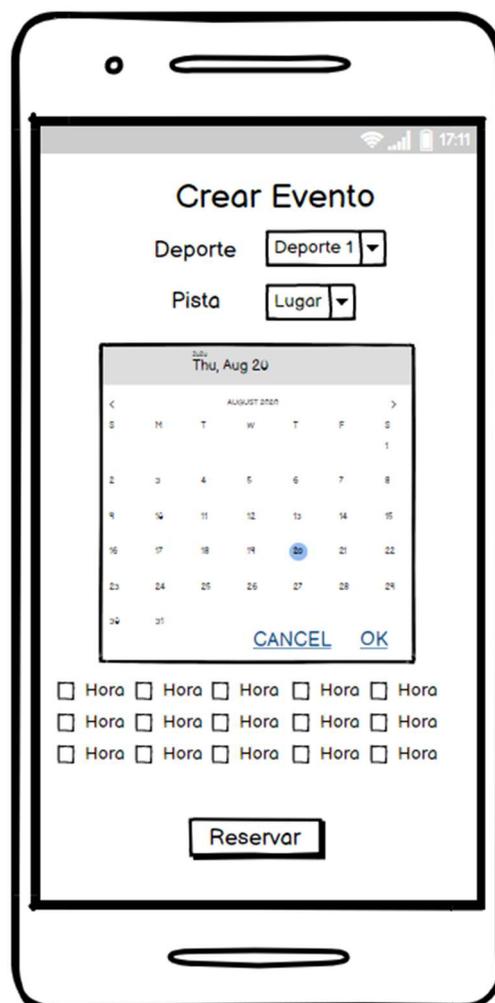


Ilustración 17. Pantalla crear reserva

- Mis eventos

Esta pantalla tendrá un aspecto similar al que aparece en la ilustración 20 que permitirá al usuario ver todos sus eventos creados, el usuario deberá escoger el deporte del cual quiere ver los eventos que tiene creados y pulsar el botón “Buscar”, ahí se le emitirá una solicitud a Firebase que nos enviará todos los Eventos creados por el Usuario en dicho deporte.

- Buscar reservas

El usuario dispondrá de una pantalla para poder buscar e inscribirse en eventos que otros usuarios crearon previamente. Esta pantalla será como aparece en la ilustración 21 donde el usuario tendrá que elegir el deporte, el día y pulsar el botón “Buscar”. Nuestra aplicación se comunicará con Firebase y así obtener todos los eventos creados en esa fecha con ese deporte en particular. La aplicación mostrará todos los resultados en forma de lista ofreciendo al usuario la posibilidad de inscribirse en los eventos haciendo click

en el botón “Inscribirse.” Si el usuario decide inscribirse en el evento, inmediatamente el sistema envía a Firebase la acción para que actualice la base de datos en los lugares correspondientes.

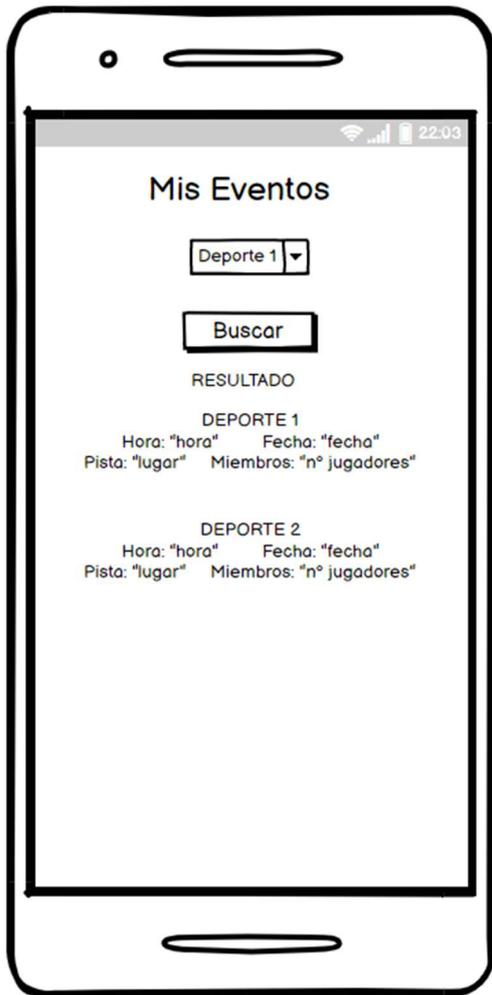


Ilustración 18. Pantalla mis eventos

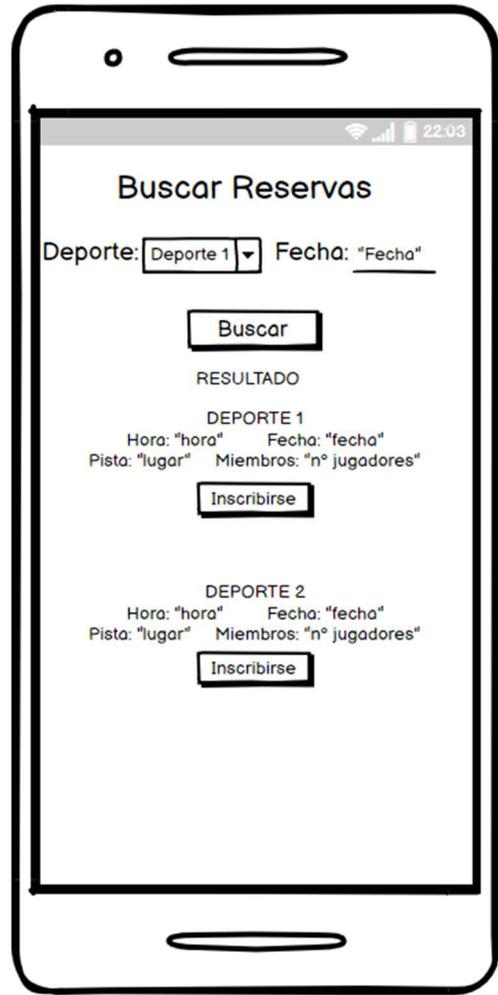


Ilustración 19. Pantalla buscar reservas

- Mis reservas

Esta es la pantalla encargada de ofrecer al usuario la posibilidad de ver en cuantos eventos se inscribió. Encontraremos el ejemplo visual en la ilustración 22, aquí el usuario solamente deberá de elegir el deporte del cual quiera saber sus inscripciones y pulsar “Buscar”, después de eso, el sistema se comunicará con la base de datos en Firebase y traerá todos los eventos en los que dicho usuario se encuentre inscrito. También, si el usuario lo desea puede darse de baja del evento pulsando el botón “Cancelar” que aparece en cada evento inscrito.

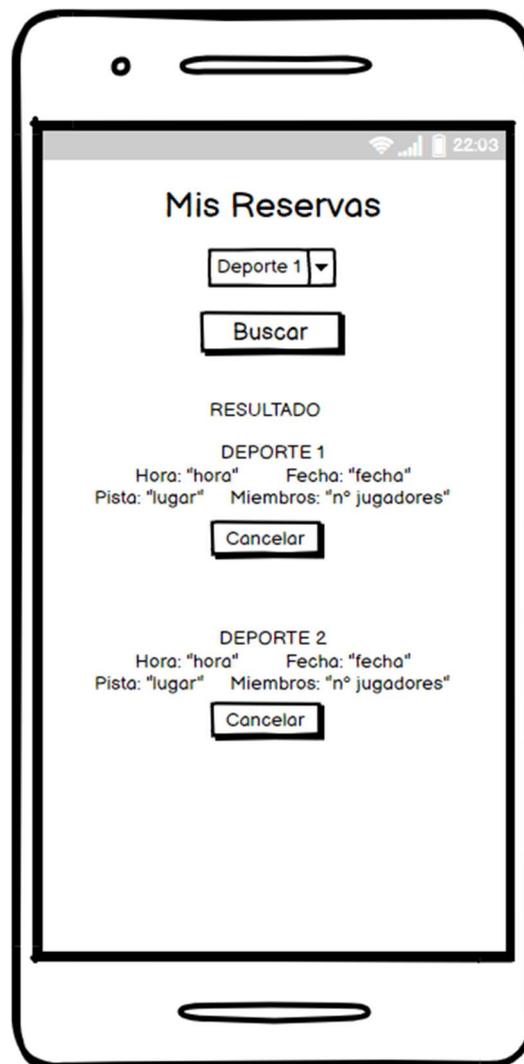


Ilustración 20. Pantalla mis reservas

5.1. Modificaciones

Dentro de este apartado hablaremos sobre los cambios que sufrió la especificación original debido a los problemas encontrados, que, como en la gran mayoría de desarrollo de aplicaciones, siempre aparecen problemas, tanto sean por problemas de tiempo de realización, problemas de implementación por falta de conocimiento, etc.

En nuestro caso los problemas encontrados afectaron a la función que el usuario tendría de valorar a los participantes del evento no pudimos implementarla. Cuando iniciamos el proyecto propusimos un gran abanico de funcionalidades utilizando tecnologías nuevas, lo que conllevó una gran dedicación de tiempo. Durante todo el proyecto y para todas las funcionalidades fuimos realizando tareas de investigación e implementación conjuntamente, pero cuando llegamos a la funcionalidad de “Valoración de participante” nos dimos cuenta de que el tiempo que nos restaba era reducido, que nos conllevaría una dedicación mayor que con las otras y que además no era una funcionalidad básica o primordial, ya que la catalogamos como opcional al inicio.

Así que teniendo en cuenta todo esto decidimos tomar la decisión de posponer dicha funcionalidad para el futuro y añadirla a la lista de trabajo futuros que mencionaremos más adelante.

6. Implantación

En nuestro caso en particular, tratamos con un proyecto que está enfocado a diferentes organizaciones o centros deportivos los cuales tienen sus propios horarios y pistas para realizar un determinado número de deportes. Para implantar nuestro sistema en los centros deportivos no hace falta realizar ninguna instalación de software o hardware en el establecimiento, solamente se deben de tomar los siguientes datos:

- Nombre de los deportes que se pueden practicar.
- Tipo de canchas/pistas en las que se practican los deportes.
- Días de la semana que se podrá practicar los deportes.
- Franja horaria para cada día que el establecimiento ofrecerá los servicios para practicar los deportes.

Una vez que tenemos estos datos, solamente hay que introducirlos en la base de datos y los usuarios podrán comenzar a utilizar nuestra aplicación.

Para poder realizar este proyecto y presentar una demostración hemos utilizado los datos del servicio de deporte de la UPV. Hemos seleccionado algunos deportes con sus diferentes pistas y el horario de disponibilidad de la UPV, de lunes a viernes con un horario de 8:00 hasta 23:00.

7. Pruebas

Como en todo proceso software es necesario realizar las pruebas pertinentes para comprobar que el desarrollo realizado en el ciclo cumple con las especificaciones. En nuestro proyecto hemos utilizado diversas pruebas dependiendo lo que se quería probar y dependiendo de en qué ciclo del proyecto nos encontramos.

7.1. Pruebas unitarias -JUnit-

Para realizar las pruebas unitarias hemos utilizado JUnit, que nos permite escribir métodos que cuando se ejecuten comprobarán que el resultado obtenido o el proceso realizado es el esperado. Nosotros hemos utilizado esta herramienta para testear las clases de los Modelos de nuestro proyecto (Usuario, Deporte, Evento.).

A continuación, veremos unas series de imágenes que muestran los tests realizados, la ilustración 23 muestra el test para la clase Usuario y la ilustración 24 muestra el resultado de ejecutar los tests de las diferentes clases.

```

@Test
public void usuario()
{
    String[] intereses = {"futbol", "tennis"};
    User usuario1 = new User("martin@email.com", "martin89", Arrays.asList(intereses));
    assertEquals(usuario1.getUserName(), "martin89");
    assertEquals(usuario1.getEmail(), "martin@email.com");
    assertEquals(usuario1.getUserInteres().get(0), "futbol");
    User usuario2 = new User();
    usuario2.setUserName("yerle89");
    usuario2.setEmail("yerle@email.com");
    usuario2.setUserInteres(Arrays.asList(new String[]{"futbol", "tennis"}));
    assertEquals(usuario2.getUserName(), "yerle89");
    assertEquals(usuario2.getEmail(), "yerle@email.com");
    assertEquals(usuario2.getUserInteres().get(1), "tennis");
}

```

Ilustración 21. Prueba Junit Usuario

Test Name	Result	Execution Time
ExampleUnitTest (com.example.tfgproject)	Pass	0 ms
evento	Pass	0 ms
usuario	Pass	0 ms
sport	Pass	0 ms

Ilustración 22. Resultado ejecución de JUnit test

7.2. Pruebas de los servicios de Firebase

Una vez probada las clases de los Modelos con todos sus métodos pasamos a comprobar que la conexión de nuestra aplicación con Firebase y los servicios que usamos. Para realizar todas las pruebas hemos usado el registro de la consola de Firebase, la base de datos, las nuevas funciones desarrolladas y un emulador para ejecutar las pruebas.

- **Comprobar registro y login:**
Esta comprobación se realizó para saber si la aplicación accedía correctamente a la base de datos para obtener los nombres de usuarios ya existentes, la lista de email ya identificados y si añadía correctamente el usuario insertando todos los datos en sus respectivos lugares.
- **Comprobar el acceso a la base de datos:**
La siguiente comprobación se realizó para ver si se obtenían correctamente los datos de los usuarios para los perfiles y que se podían editar correctamente. También se probó la gestión de eventos, cuando se creaban, se buscaban o se actualizaban.
- **Comprobación de funciones y Cloud Messaging:**
En estas pruebas lo que se busca es comprobar el correcto funcionamiento de las funciones escritas en JavaScript para el envío de notificaciones a los diferentes usuarios dependiendo de sus deportes de interés.

Pruebas de calidad

Antes de enviar la aplicación para que la validen los usuarios realizamos las pruebas de calidad que sugiere Android en su página oficial [7]. Estas pruebas aseguran que las aplicaciones cumplan con un conjunto de criterios que aseguran que funcionarán bien en diferentes dispositivos, que cumplen las normas de Android para la navegación y diseño, y que están preparadas para publicarse en Google Play Store.

Algunos de los criterios usados y de las pruebas realizadas fueron las siguientes:

I. Diseño visual e interacción del usuario:

- a. Navegación estándar del sistema con el “botón Atrás”: desde cada una de las pantallas de la app, presionar el botón Atrás.
- b. Botón de inicio navega a la pantalla principal del dispositivo: desde cada pantalla presionar la tecla de inicio y luego reinicia la app desde la pantalla Todas las apps.
- c. Las notificaciones indica un cambio de contexto relacionado con el usuario: iniciar la app y observar en el panel de notificaciones todos los tipos de notificaciones que se muestran.

II. Funcionalidad:

- a. Solicitud mínimos de permisos: revisar los permisos que requiere la aplicación en tiempo de ejecución en la configuración del dispositivo.
- b. Servicios en ejecución mientras está en 2º plano: cambiar a otra app para colocar nuestra aplicación en segundo plano y comprobar qué servicios usa la app.
- c. Preservación del estado de usuario luego de estar en 2º plano: presiona tecla de inicio y luego abre nuevamente la app.

III. Compatibilidad, rendimiento y estabilidad:

- a. La aplicación no falla o no impone un cierre brusco: realizar todas las acciones anteriores más todas las otras pruebas ofrecidas por Android.

- b. La app carga o muestra la información rápidamente: realizar todas las pruebas anteriores y esperar que la respuesta sea menos de 2 segundos.
- c. La app muestra gráficos, textos, imágenes de la IU sin distorsión: realizar todas las pruebas anteriores y se observa todos los elementos en alta calidad ni alteraciones en botones o menús.

IV. Seguridad:

- a. Todos los datos privados se guardan en el almacenamiento interno.
- b. No se registran datos personales del usuario en el sistema o registro: ejecutar la app y verificar que no aparecen datos personales en el registro.
- c. La aplicación declara una configuración de seguridad de red.
- d. La app no carga dinámicamente el código desde afuera del apk.

7.3. Validación

Cuando completamos todos los ciclos en el desarrollo de nuestro software, pasamos a comprobar que el sistema que hemos realizado cumple con todas las especificaciones iniciales y con los objetivos trazados. Para realizar dichas validaciones nos basamos en unas encuestas obtenidas en un artículo [8] publicado en internet que nos facilitó mucho el trabajo. El procedimiento en cada ciclo era el mismo, crear la apk (archivo instalador) y enviárselo a 7 amigos para que la utilicen entre ellos y así recibir el feedback de sus experiencias.

Para realizar las pruebas tuvimos que dividirlos en dos grupos para que eligieran los mismos deportes y así se ver si se podían comunicar entre ellos y no enterarse de la actividad que pasaba entre los usuarios con otros intereses. Después de 5 días de que los usuarios probaran nuestra aplicación, tenían que rellenar un formulario con unas preguntas para poder ver los resultados y analizarlos. Para cada test, los usuarios encargados de validar recibían un pequeño resumen de las nuevas implementaciones en el último ciclo y cuatro tests diferentes, efectividad (Tabla 27), eficiencia (Tabla 28), satisfacción (Tabla 29) y usabilidad (Tabla 30).

I. Test de efectividad:

Evaluación de efectividad					
A) El programa que se está evaluando es fácil de aprender a usar.	1	2	3	4	5
B) Los elementos mostrados en la interfaz son necesarios para su uso.	1	2	3	4	5
C) El programa presenta una consistencia visual de los elementos (iconos, barras de estados, menú, tipografía).	1	2	3	4	5

Tabla 26. Evaluación de la efectividad

II. Test de eficiencia:

Evaluación de eficiencia					
A) El programa presenta de forma clara si se presenta un error.	1	2	3	4	5
B) El programa proporciona información del proceso actual.	1	2	3	4	5
C) La obtención de resultados satisfactorios por medio de este programa es fácil.	1	2	3	4	5
D) Solo se debe de introducir la información una sola vez para obtener los resultados.	1	2	3	4	5

Tabla 27. Evaluación de la eficiencia

III. Test de satisfacción:

Evaluación de satisfacción					
A) Usted recomienda el uso del programa.	1	2	3	4	5
B) El programa funciona de manera adecuada.	1	2	3	4	5
C) La interfaz de usuario de la aplicación es agradable.	1	2	3	4	5
D) El programa estimula su uso prolongado.	1	2	3	4	5

Tabla 28. Evaluación de la satisfacción

IV. Test de usabilidad:

Evaluación de usabilidad					
A) Fue simple utilizar la aplicación.	1	2	3	4	5
B) Soy capaz de realizar la tarea que se me pide rápidamente.	1	2	3	4	5
C) Me siento cómodo utilizando la aplicación.	1	2	3	4	5
D) Pienso que la aplicación se puede utilizar sin necesidad de un manual de instrucciones.	1	2	3	4	5
E) Las funciones que ofrece la aplicación están bien integradas.	1	2	3	4	5
F) Cuando cometo un error puedo resolverlo rápidamente.	1	2	3	4	5

Tabla 29. Evaluación de la usabilidad

Además de los test que aparecen en las tablas anteriores, el usuario también recibe una escala estableciendo el significado de los números para que pueda realizar la votación.

Nunca	Pocas veces	A veces	Casi siempre	Siempre
1	2	3	4	5

Tabla 30. Significado de la numeración en las encuestas

La mayor parte de las valoraciones de los tests fueron positivas, obtuvimos unos resultados positivos reflejando que las nuevas implementaciones hacían lo que se esperaba y que la aplicación seguía siendo útil, agradable e intuitiva. Pero varios de los usuarios añadieron ciertos puntos que se podrían incluir en el futuro para mejorar nuestro programa o que no les gustaba cómo funcionaba la aplicación actualmente. Todas estas mejoras serán comentadas más adelante donde se presentarán futuras mejoras para la aplicación que, por motivos de complejidad o porque no las habíamos pensado, no tuvimos tiempo para desarrollarlas.

A continuación, mostraremos una serie de gráficas que representarán los resultados obtenidos reflejando así los porcentajes de las valoraciones. Hay que tener en cuenta para entender estas graficas que hemos realizado tres ciclos de desarrollo, que el número de usuarios que validaban nuestra aplicación era 7 y que el número de preguntas de preguntas varía dependiendo la característica a evaluar. En el eje y tendremos la referencia a las preguntas de cada característica (A, B, C, etc.) y en el eje x tendremos las valoraciones obtenidas para cada pregunta en porcentaje a la cantidad de usuarios validadores. Así que por ejemplo en la Ilustración 23 tendremos un total de 21 valoraciones (3 ciclos * 7 testers) que se repartirán entre las 5 opciones posibles (1, 2, 3, 4, 5).

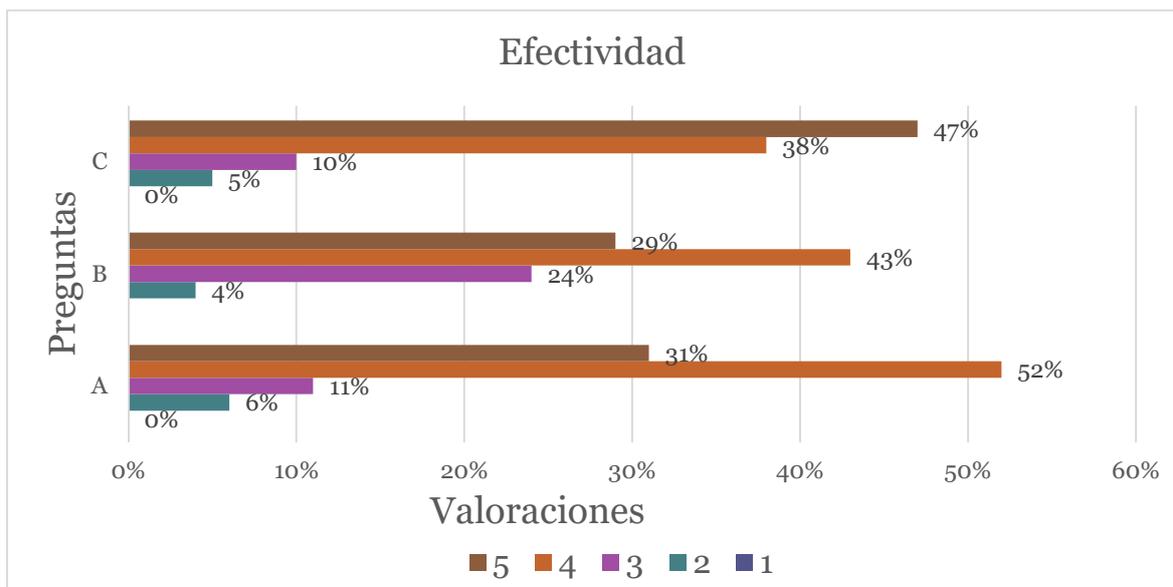


Ilustración 23 Resultados de la encuesta de Efectividad

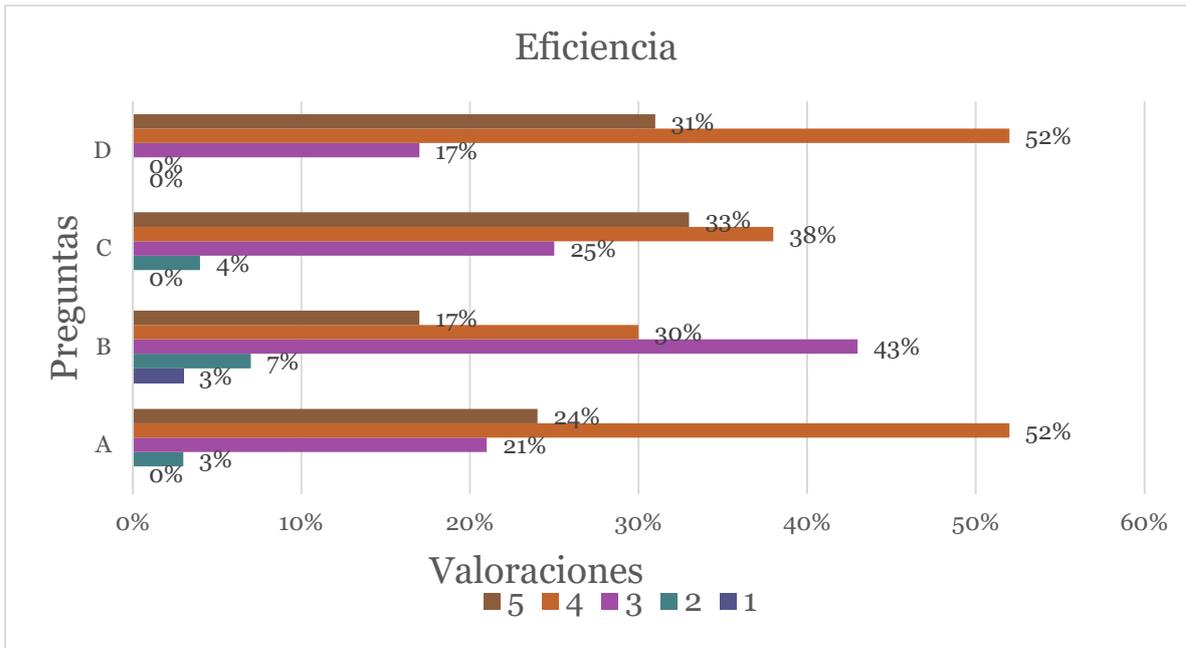


Ilustración 24 Resultados de la encuesta de Eficiencia

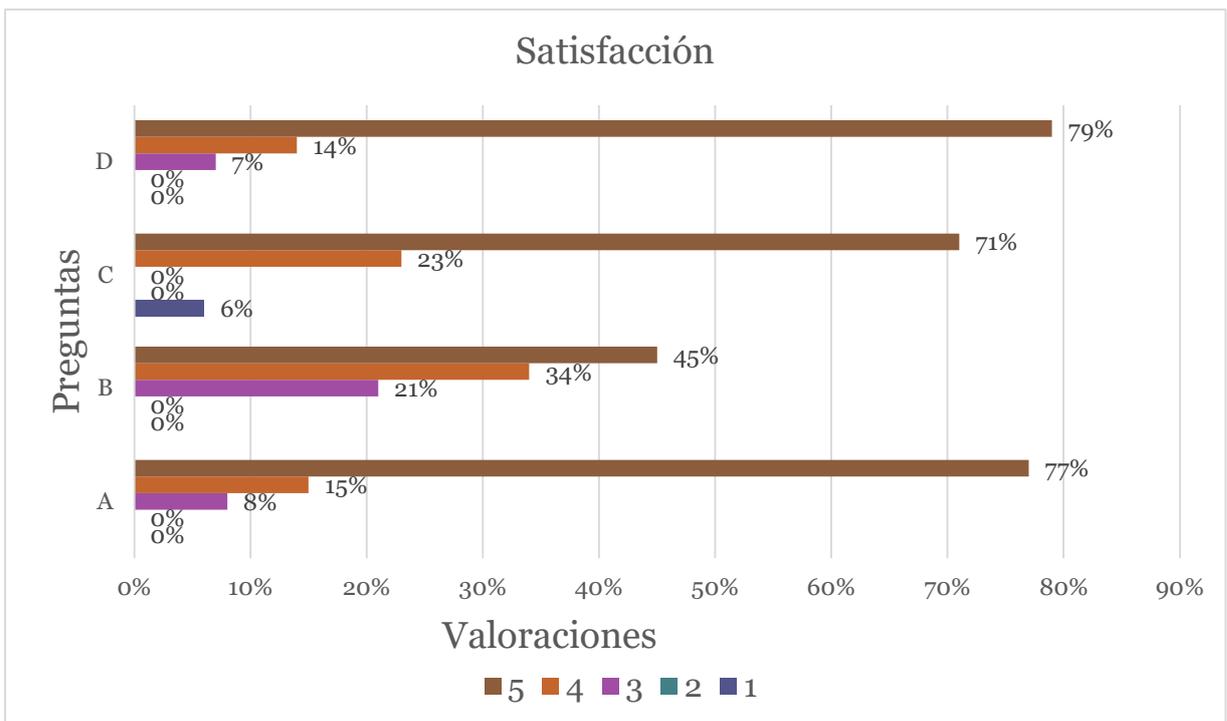


Ilustración 25 Resultados de la encuesta de Satisfacción

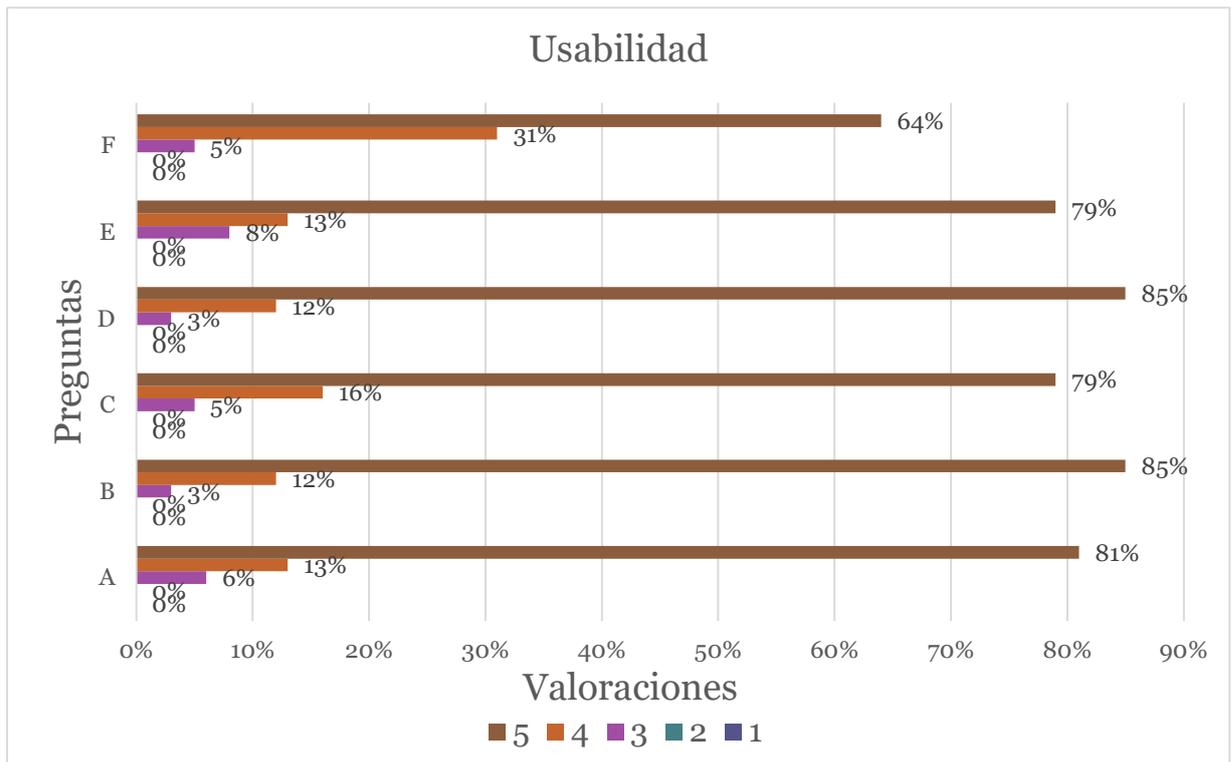


Ilustración 26 Resultados de la encuesta de Usabilidad

Como podemos observar en las Ilustraciones anteriores nuestras encuestas obtuvieron mayormente valoraciones entre 4-5, lo que significa que nuestro trabajo tuvo una buena repercusión entre los usuarios. Pero no podemos dejar escapar la pequeña “llamada de atención” que sufrimos en la encuesta de Eficiencia en la pregunta B “El programa proporciona información del proceso actual.”. Los usuarios valoraron mayormente con una nota de 3, lo que significa que el programa no provee al usuario de la información del proceso actual. Así que esto deberá de incluirse como mejora para los desarrollos futuros de nuestro proyecto.

8. Manual de uso

En este apartado vamos a detallar los requisitos que hay que tener para usar nuestra aplicación y los pasos a seguir para utilizarla.

8.1. Registro y Login

Para comenzar a utilizar nuestra aplicación es necesario disponer de una cuenta de correo. Debemos de acceder a la pantalla de Registro (ilustración 25) y cumplimentar los campos que ahí aparecen. Se deberá de elegir una contraseña alfanumérica de mínimo 8 caracteres. Una vez que dispongamos de nuestra cuenta, podremos acceder a la aplicación introduciendo el email y la contraseña en la pantalla de Login (Ilustración 26).



Ilustración 25. Pantalla registrar

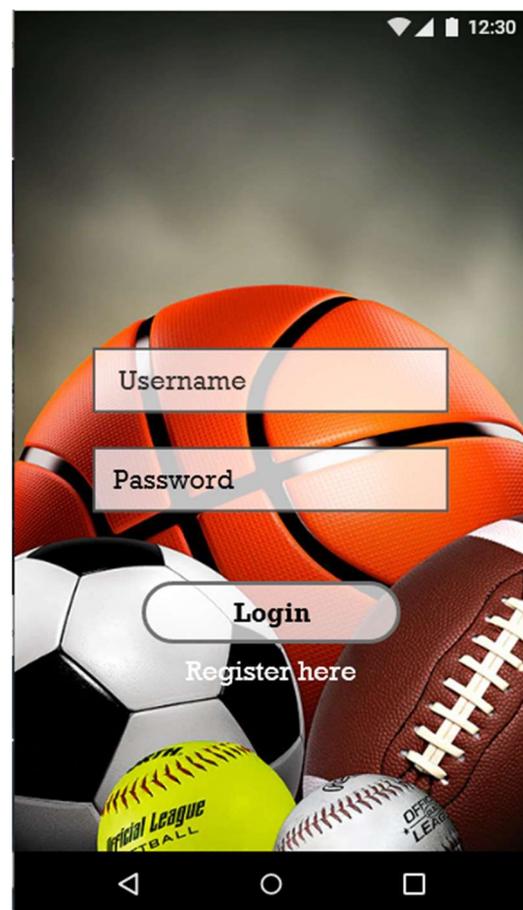


Ilustración 26. Pantalla login

8.2. Perfil

Una vez estamos dentro de nuestra aplicación en la pantalla principal (Ilustración 27) tendremos que configurar los intereses de nuestro perfil. Esto nos permitirá crear y/o buscar eventos para practicar deporte. Para esto será necesario pulsar el icono de Perfil y así acceder a nuestro perfil (Ilustración 28). Una vez dentro de nuestro perfil debemos de pulsar el botón de editar para poder ir a la pantalla de edición de nuestro perfil (Ilustración 29) y añadir o quitar deportes de la lista de nuestros intereses. Una vez tengamos ya finalizada nuestra elección de deportes solamente debemos de pulsar el botón de “guardar”.

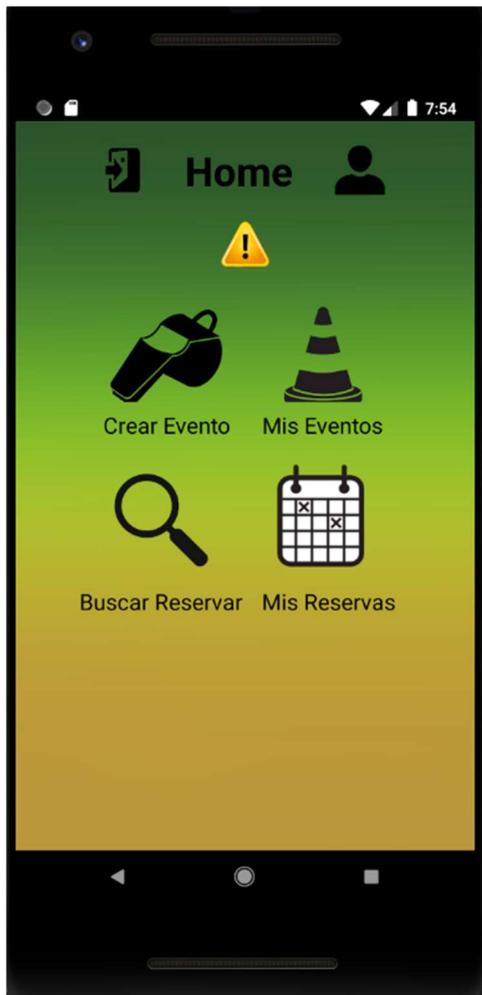


Ilustración 27. Pantalla principal



Ilustración 28. Pantalla perfil

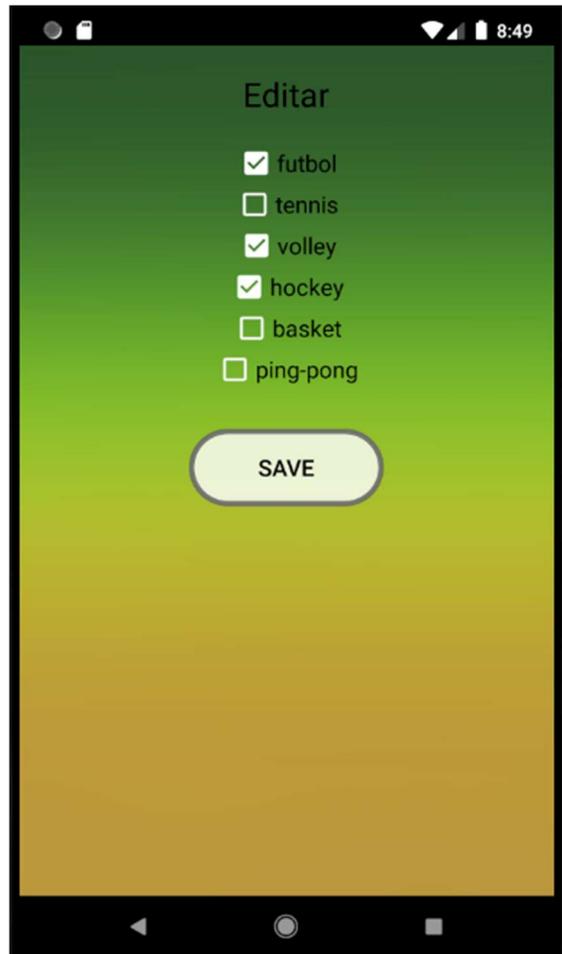


Ilustración 27. Pantalla Editar perfil

8.3. Eventos

Esta es la parte principal de nuestra aplicación, la gestión de los eventos.

- Crear una reserva:

Para crear un evento y esperar que otros usuarios se unan, el usuario debe de ir a la pantalla de “Crear Evento” (Ilustración 30), una vez en dicha pantalla el usuario deberá de elegir el deporte del evento, la pista en la que se disputará el evento, la fecha y la hora entre las que estén disponibles. Ahora solamente necesitamos confirmar la creación de la reserva pulsando el botón de “Reservar”.



Ilustración 28. Pantalla crear evento

- Visualizar los eventos creados:

El usuario deberá de acceder a la pantalla de “Mis Eventos”, una vez en esa pantalla se deberá seleccionar el deporte y pulsar “Buscar”. Ahora el usuario podrá visualizar el resultado de la búsqueda viendo todos los eventos creados del deporte elegido.

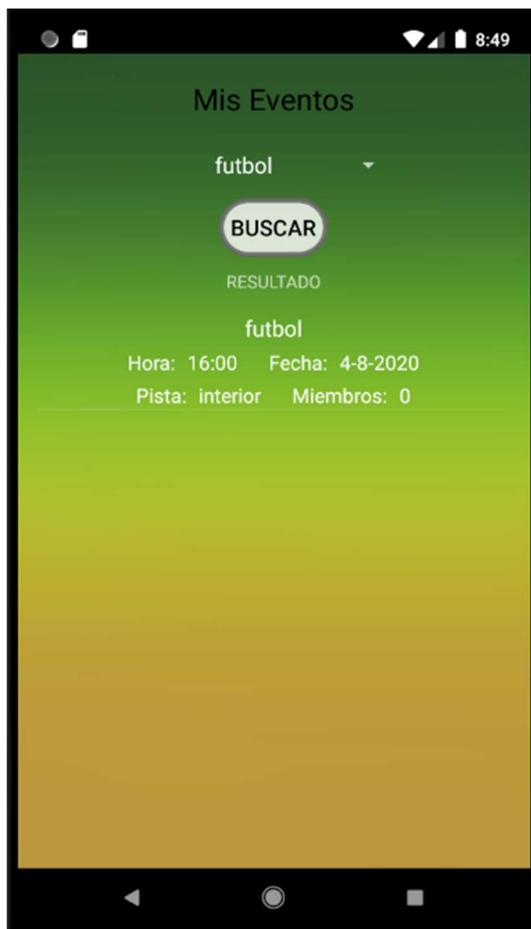


Ilustración 29. Pantalla mis eventos

- Buscar una reserva:

El usuario podrá unirse a un evento creado por otro usuario yendo a la pantalla de “Buscar Reservas” (Ilustración 32), una vez ahí, el usuario deberá seleccionar el deporte y el día en el que le interesa encontrar el evento. Cuando el usuario pulse el botón “Buscar” le aparecerán una lista de eventos a los cuales el usuario podrá unirse solamente pulsa el botón de “Inscribirse” en cada evento.



Ilustración 30. Pantalla buscar reserva

- Visualizar las reservas que en las que el usuario se inscribió:
El usuario puede ir a la pantalla “Mis Reservas” (Ilustración 33) seleccionar el deporte del que desea ver sus inscripciones y pulsar el botón “Buscar”. El sistema mostrará la lista de todos los eventos en el que el usuario se encuentre inscrito. Quedará a elección del usuario el darse de baja del evento, solamente tendrá que pulsar el botón “Cancelar” en el evento.

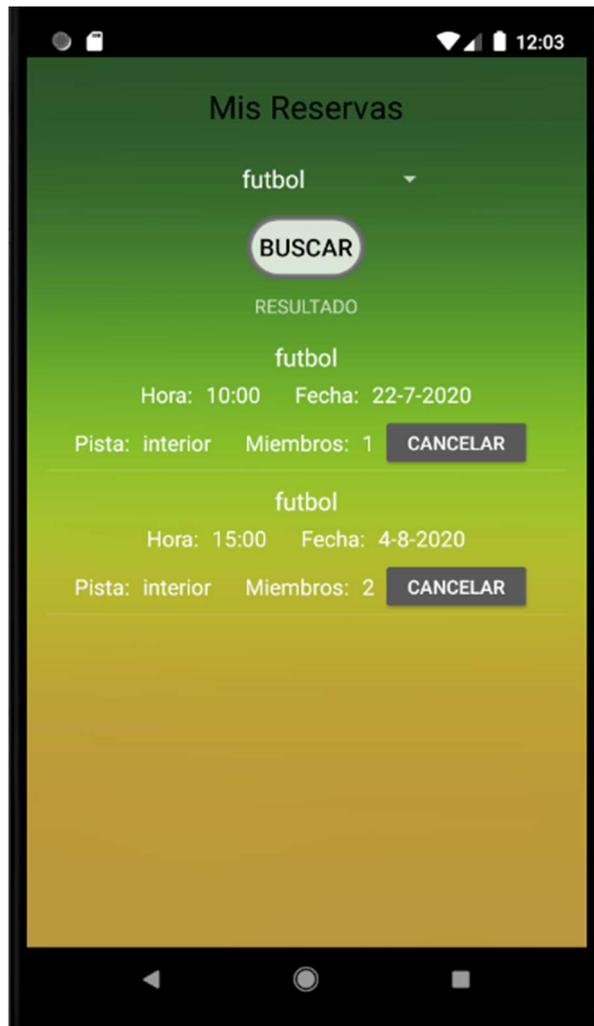


Ilustración 31. Pantalla mis reservas

9. Conclusiones

Si nos paramos a analizar cómo se desarrolló el proyecto podemos ver como hemos podido cumplir el objetivo global, pero si analizamos el viaje que hemos realizado a lo largo del proyecto nos encontramos con el gran desafío que supuso utilizar Firebase, ya que era la encargada de ofrecernos todos los servicios que necesitábamos y que era totalmente nueva para nosotros. Por fortuna, Firebase dispone de una gran cantidad de información en forma de documentación y videos que pudimos utilizar. Con diferencia, la parte que requirió más tiempo fue la de la configuración de las notificaciones. Otro gran apoyo para la implementación de los servicios de Firebase en nuestro proyecto fue el sitio web “Stackoverflow”⁶, que nos proporcionó infinidad de ejemplos capaces de ayudarnos a comprender como funciona Firebase. El resto de las tecnologías utilizadas Java, IDE y GitHub no supusieron un gran problema, ya que disponíamos de los conocimientos necesarios para llevar adelante nuestro proyecto.

Así que una vez llegados a este punto y analizado toda la aplicación podemos estar orgullosos de nuestro resultado, hemos conseguido crear una aplicación móvil que permite conectar diferentes usuarios con los mismos intereses deportivos para poder practicar deporte. Estos usuarios podrán crear o unirse a los eventos deportivos el día y a la hora que a ellos les convengan, también estarán informados constantemente de cualquier cambio relacionado con los deportes que les interesa.

9.1. Relación Proyecto – Estudios cursados

En este apartado veremos cómo y dónde se pusieron en práctica los conocimientos que aprendimos en la carrera. Empezaremos hablando del lenguaje que hemos utilizado, Java, dicho lenguaje lo hemos aprendido durante la carrera viéndolo en varias asignaturas como por ejemplo PRG. Si hablamos de la parte de la estructura del proyecto y de la arquitectura, debemos de hacer alusiones a asignaturas como Ingeniería del Software (ISW) o Diseño de Software (DDS). A la hora de hablar de herramientas para controlar el proceso software usamos los conocimientos aprendidos en Proceso del Software (PSW) o Proceso de Ingeniería Software (PIN). Finalizaremos con lo relevante a la interfaz de la aplicación, para esto, hemos aplicado lo adquirido en Interfaces Persona Computador (IPC). Con lo relacionado a la base de datos o todo lo relacionado con Firebase fue un área más desconocida para mí, como se mencionó en el punto anterior tuve que adquirir los conocimientos por mi cuenta.

9.2. Trabajos futuros

Una vez completado el proyecto y recibido el feedback de nuestros usuarios pasaremos a comentar nuestras ideas de trabajos futuros, aquellas cosas que o bien no pudimos desarrollarla como queríamos o bien mejoras o ideas que han surgido mediante la validación con el usuario.

⁶ <https://es.stackoverflow.com/>

- Establecer un sistema de niveles para los usuarios así poder realizar eventos deportivos entre jugadores del mismo nivel. Dichos niveles se obtendrían al inicio mediante unas preguntas que el usuario debe de contestar y, posteriormente, por las valoraciones de los compañeros.
- Establecer un sistema de chat para que los usuarios se puedan comunicar entre ellos por si hubiera algún imprevisto o cambio de última hora.
- Crear un sistema de estadísticas para cada usuario independiente para cada deporte. Esto indicará el número de partidos jugados, ganados, posición, nivel, etc.
- Poder brindar una opción a usuarios que deseen ofrecer sus servicios de profesores en algunos deportes. Ayudaríamos a algunos usuarios a poder impartir clases y a otros a desarrollar sus habilidades en ese deporte.
- El usuario deberá valorar a los participantes de los eventos, confirmando su asistencia. Si no asisten al evento recibirán una sanción que les impediría crear o inscribirse a eventos por un determinado periodo de tiempo.
- Brindar a los usuarios más información sobre el estado actual del programa, ej:” Cargando datos”, “creando evento”, “buscando evento”, etc.

10. Bibliografía

- [1] M. d. E. C. y. Deporte., «Encuesta de hábitos deportivos en España 2015,» 12 2015. [En línea]. Available: http://www.culturaydeporte.gob.es/dam/jcr:398d6e55-7bc5-4b8e-b0f3-95dd49c4997f/Encuesta_de_Habitos_Deportivos_2015.pdf. [Último acceso: 7 2020].
- [2] A. gems, «Metodología de desarrollo de software (III) – Modelo en Espiral,» 5 4 2019. [En línea]. Available: <https://aspgems.com/metodologia-de-desarrollo-de-software-iii-modelo-en-espinal/>. [Último acceso: 15 5 2020].
- [3] IEEE, «International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering,» 30 11 2018. [En línea]. Available: <https://ieeexplore.ieee.org/document/8559686>. [Último acceso: 27 2 2020].
- [4] J. Tomás, V. Carbonell, J. Bataller y J. Lloret, Firebase: trabajar en la nube (Spanish Edition), Marcombo S.A , 2019.
- [5] L. Moroney, The Definitive Guide to Firebase. Build Android Apps on Google's Mobile Platform, 2017.
- [6] M. Manzo, «SQL vs NoSQL; Ventajas y Desventajas,» 17 5 2018. [En línea]. Available: <https://medium.com/@marlonmanzo/sql-vs-nosql-ventajas-y-desventajas-849ccc9db3d4>. [Último acceso: 25 5 2020].
- [7] Firebase, «Primeros pasos con Cloud Firestore,» 19 1 2020. [En línea]. Available: <https://firebase.google.com/docs/firestore/quickstart?hl=es-419>. [Último acceso: 4 3 2020].
- [8] Firebase, «Autentica con Firebase mediante cuentas con contraseña en Android,» 21 1 2020. [En línea]. Available: <https://firebase.google.com/docs/auth/android/password-auth?hl=es-419>. [Último acceso: 27 2 2020].
- [9] Firebase, «Envía mensajes a varios dispositivos,» 17 1 2020. [En línea]. Available: <https://firebase.google.com/docs/cloud-messaging/android/send-multiple>. [Último acceso: 10 2 2020].
- [10] Sgoliver, «Notificaciones Push en Android: Firebase Cloud Messaging,» 20 12 2016. [En línea]. Available: <https://www.sgoliver.net/blog/notificaciones-push-android-firebase-cloud-messaging-1/>. [Último acceso: 19 2 2020].
- [11] Firebase, «Descripción general de Firebase Test Lab,» 19 1 2020. [En línea]. Available: <https://firebase.google.com/docs/test-lab/android/overview>. [Último acceso: 21 2 2020].

- [12] I. F. Darwin, *Android Cookbook: Problems and Solutions for Android Developers*, O'Reilly.
- [13] G. developers, «Calidad de las apps principales,» 27 12 2019. [En línea]. Available: <https://developer.android.com/docs/quality-guidelines/core-app-quality>. [Último acceso: 10 5 2020].
- [14] M. A. Barajas B., R. M. Reyes M., A. A. Maldonado, A. I. García A. y J. De la Rivera R., «Análisis de cuestionarios para la evaluación de la usabilidad,» 4 1 2018. [En línea]. Available: <http://148.202.248.167/ojs/index.php/egnosis/article/viewFile/206/158>. [Último acceso: 9 5 2020].
- [15] Google, «Documentación para desarrolladores de apps,» [En línea]. Available: <https://developer.android.com/docs>. [Último acceso: 17 3 2020].