



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

**Social Paddle: APLICACIÓN ANDROID
PARA LA GESTIÓN DE EVENTOS
DEPORTIVOS**

Alumno: David Pastor Pérez

Tutor: María Rosario Abril Raymundo

Resumen

Las tecnologías móviles están en auge, en estos momentos un dispositivo móvil te permite realizar infinidad de tareas que hace años tan solo un ordenador era capaz de ello, por eso en los últimos años los móviles están sustituyendo a los ordenadores de sobremesa. Es por esto que el siguiente trabajo de fin de grado se desarrollará en el entorno de las tecnologías móviles.

En este trabajo de fin de grado se ha realizado una aplicación para dispositivos móviles, en concreto para el sistema operativo Android. La aplicación permite a un centro deportivo organizar partidos de pádel y a los clientes del centro, unirse a esos partidos, organizados principalmente en forma de torneo. La aplicación permite generar torneos con inscripción por parejas o de forma individual. En este último caso, la aplicación busca una pareja adecuada para el jugador solitario, utilizando información guardada sobre los clientes que utilizan la aplicación. Esta información se va actualizando en función de los resultados de los partidos y del nivel de los contrincantes. Esto permite un mejor emparejamiento a medida que se va utilizando la aplicación y favorece un aspecto competitivo y de superación personal en la práctica del pádel.

Abstract

Mobile technologies are booming, at the moment a mobile device allows you to perform infinite tasks that years ago only a computer was capable of, that is why in recent years mobiles are replacing desktop computers. Because of this, the following final degree project will be developed in the environment of mobile technologies.

In this project, an application has been made for mobile devices, specifically for the Android operating system. The application allows a sports center to organize paddle games and the clients of the center to join those games, organized mainly in the form of a tournament. The application allows you to generate tournaments with registration in pairs or individually. In the latter case, the application searches for a suitable match for the solitary player, using saved information about the clients that use the application. This information is updated based on the match results and the level of the opponents. This allows a better pairing as the application is used and favors a competitive and self-improvement aspect in the practice of paddle tennis

Índice general

Resumen	3
Abstract	5
1. Introducción	1
1.1. Estructura de la memoria	1
1.2. Contexto y motivación	2
1.3. Objetivos	2
1.3.1. Objetivos de la aplicación	3
1.3.2. Objetivos personales	3
2. Tecnologías y servicios utilizados	5
2.1. Android Studio	5
2.2. Java	6
2.3. Firebase	6
2.3.1. Firebase Auth	7
2.3.2. Realtime Database	7
2.3.3. Firebase Storage	7
2.4. GitHub	7
2.5. Visual Paradigm	8
2.6. Microsoft Project	8
	7

3. Planificación y gestión	9
3.1. Planificación temporal	9
3.2. Gestión de riesgos	12
3.3. Presupuesto	14
3.3.1. Hardware	14
3.3.2. Software	14
3.3.3. Personal	15
3.3.4. Coste total	15
4. Análisis y diseño	17
4.1. Requisitos funcionales	17
4.2. Requisitos de información	18
4.3. Requisitos no funcionales	20
4.4. Casos de uso	21
4.4.1. Definición de actores	21
4.4.2. Diagrama de casos de uso	22
4.4.3. Descripción de casos de uso	23
4.5. Modelo de dominio	37
4.5.1. Descripción de las clases.	37
4.6. Diagrama de secuencia	39
4.7. Diagrama de despliegue	51
4.8. Diseño base de datos	52
5. Implementación	55
5.1. Interfaz gráfica	55
5.1.1. Prototipo inicial	55
5.1.2. Aspecto final	55

5.2. Niveles de habilidad	60
5.3. Partidos igualados	61
5.3.1. Torneo en parejas	61
5.3.2. Torneo en solitario	61
5.4. Problemas en la implementación	62
6. Pruebas	65
7. Conclusiones	75
Bibliografía	77
A. Manual de instalación	79
B. Manual de usuario	81
B.1. Crear nueva cuenta.	82
B.2. Iniciar sesión	83
B.3. Cambiar contraseña	83
B.4. Perfil de usuario	84
B.5. Editar foto	85
B.6. Ver y eliminar parejas	85
B.7. Menú lateral	88
B.8. Buscar pareja	89
B.9. Peticiones de pareja	90
B.10. Unir a un torneo	90
B.11. Próximos partidos	92
B.12. Solicitud de participación en torneo	93
B.13. Historial de torneos	93
B.14. Crear torneo (Administrador)	95

ÍNDICE GENERAL

B.15. Eliminar torneo (Administrador)	96
B.16. Generar rondas (Administrador)	97
B.17. Anotar resultados (Administrador)	98
B.18. Ver todas las parejas (Administrador)	99
C. Contenidos del CD-ROM	101

Índice de figuras

2.1. Logotipo Android Studio	5
2.2. Logotipo Java	6
2.3. Logotipo Firebase	6
2.4. Logotipo GitHub	7
2.5. Logotipo Visual Paradigm	8
2.6. Logotipo Visual Microsoft Project	8
3.1. Tareas	11
3.2. Diagrama de Gantt	11
4.1. Diagrama de casos de uso	22
4.2. Modelo de dominio	37
4.3. SD-001-Crear torneo	39
4.4. SD-002-Anotar resultados	39
4.5. SD-003-Ver torneos	40
4.6. SD-004-Generar nueva ronda	41
4.7. SD-005-Eliminar torneo	42
4.8. SD-006-Registrarse en el sistema	42
4.9. SD-007-Ver perfil de usuario	43
4.10. SD-008-Activar buscar pareja	43

4.11. SD-009-Desactivar buscar pareja	43
4.12. SD-010-Subir foto de perfil	44
4.13. SD-011-Ver parejas	44
4.14. SD-012-Eliminar pareja	45
4.15. SD-013-Ver historial de torneos	45
4.16. SD-014-Ver Partidos del torneo	46
4.17. SD-015-Buscar pareja	46
4.18. SD-016-Buscar torneos en pareja	47
4.19. SD-017-Buscar torneo en solitario	47
4.20. SD-018-Salir del torneo	48
4.21. SD-019-Aceptar solicitud de pareja	48
4.22. SD-020-Rechazar solicitud de pareja	49
4.23. SD-021-Ver proximos partidos	49
4.24. SD-022-Cambiar contraseña	50
4.25. SD-023-Aceptar solicitud de torneo	50
4.26. SD-024-Rechazar solicitud de torneo	51
4.27. SD-025-Consultar equipos	51
4.28. Diagrama de despliegue	52
5.1. Prototipo inicial	56
5.2. Aspecto final usuario I	56
5.3. Aspecto final usuario II	57
5.4. Aspecto final usuario III	57
5.5. Aspecto final usuario IV	58
5.6. Aspecto final usuario V	58
5.7. Aspecto final administrador I	59
5.8. Aspecto final administrador II	59

A.1. Ejemplo activar orígenes desconocidos en móvil Xiaomi	79
A.2. Ejemplo instalación en móvil Xiaomi	80
B.1. Logo aplicación	81
B.2. Crear nueva cuenta	82
B.3. Iniciar sesión	83
B.4. Cambiar contraseña	84
B.5. Perfil de usuario	85
B.6. Editar foto de perfil	86
B.7. Ver parejas	87
B.8. Menú lateral	88
B.9. Buscar pareja	89
B.10.Solicitud de pareja	90
B.11.Unir a un torneo	91
B.12.Próximos partidos	92
B.13.Solicitudes de torneo	93
B.14.Historial de torneos	94
B.15.Crear torneo (Administrador)	95
B.16.Eliminar torneo (Administrador)	96
B.17.Generar rondas (Administrador)	97
B.18.Anotar resultados (Administrador)	98
B.19.Ver todas las parejas (Administrador)	99

Índice de cuadros

3.1. Reparto semanal fase: Codificación.	10
3.2. Estimación del tiempo y coste no realista	12
3.3. Problemas de rendimiento	12
3.4. Exceso de dificultad en el desarrollo	12
3.5. Falta de personal por baja laboral.	13
3.6. Desarrollo de interfaces erróneo.	13
3.7. Pérdida de datos.	13
3.8. Coste hardware.	14
3.9. Coste software.	14
3.10. Coste personal.	15
3.11. Coste total.	15
4.1. Requisitos funcionales.	18
4.2. FRQ-023 Información del usuario.	18
4.3. FRQ-024 Información del torneo.	19
4.4. FRQ-025 Información del partido.	19
4.5. FRQ-026 Información de las parejas.	19
4.6. NFRQ-001 Compatibilidad de la aplicación.	20
4.7. NFRQ-002 Firebase Realtime.	20
4.8. NFRQ-003 Idioma de la aplicación.	20

4.9. NFRQ-003 Autenticación con correo único.	20
4.10. NFRQ-005 Firebase Authentication.	20
4.11. NFRQ-006 Firebase Storage.	21
4.12. ACT-001 Administrador.	21
4.13. ACT-002 Jugador.	21
4.14. UC-001 Crear torneo.	23
4.15. UC-002 Anotar resultados.	24
4.16. UC-003 Ver torneos.	24
4.17. UC-004 Generar nueva ronda.	25
4.18. UC-005 Eliminar torneo.	25
4.19. UC-006 Registrarse en el sistema.	26
4.20. UC-007 Ver perfil de usuario.	26
4.21. UC-008 Activar buscar de pareja.	27
4.22. UC-009 Desactivar búsqueda de pareja.	27
4.23. UC-010 Subir foto de perfil.	28
4.24. UC-011 Ver pareja.	28
4.25. UC-012 Eliminar pareja.	29
4.26. UC-013 Ver historial de torneos.	29
4.27. UC-014 Ver partidos del torneo.	30
4.28. UC-015 Buscar pareja.	30
4.29. UC-016 Buscar torneos en pareja.	31
4.30. UC-07 Buscar torneos en solitario.	32
4.31. UC-018 Salir del torneo.	33
4.32. UC-019 Aceptar solicitud de pareja.	33
4.33. UC-020 Rechazar solicitud de pareja.	34
4.34. UC-021 Ver próximos partidos.	34

4.35. UC-022 Cambiar contraseña.	35
4.36. UC-023 Aceptar solicitud de torneo.	35
4.37. UC-024 Rechazar solicitud de torneo.	36
4.38. UC-025 Consultar equipos.	36
5.1. Correspondencia Nivel-Puntos	60
5.2. Puntos por Victoria/Derrota	61
6.1. Batería de pruebas para el caso CPSYS-001 Registrar usuario	65
6.2. Batería de pruebas para el caso CPSYS-002 Registrar usuario (2)	65
6.3. Batería de pruebas para el caso CPSYS-003 Iniciar sesión	66
6.4. Batería de pruebas para el caso CPSYS-004 Iniciar sesión (2)	66
6.5. Batería de pruebas para el caso CPSYS-005 Activar búsqueda de pareja	66
6.6. Batería de pruebas para el caso CPSYS-006 Desactivar búsqueda de pareja	66
6.7. Batería de pruebas para el caso CPSYS-007 Ver parejas	66
6.8. Batería de pruebas para el caso CPSYS-008 Ver parejas (2)	67
6.9. Batería de pruebas para el caso CPSYS-009 Eliminar pareja	67
6.10. Batería de pruebas para el caso CPSYS-010 Ver torneos	67
6.11. Batería de pruebas para el caso CPSYS-011 Ver torneos (2)	67
6.12. Batería de pruebas para el caso CPSYS-012 Unirme a torneo con pareja	68
6.13. Batería de pruebas para el caso CPSYS-013 Unirme a torneo con pareja (2)	68
6.14. Batería de pruebas para el caso CPSYS-014 Salir de un torneo con pareja	68
6.15. Batería de pruebas para el caso CPSYS-015 Cambiar foto de perfil	68
6.16. Batería de pruebas para el caso CPSYS-016 Unirse a un torneo en solitario	69
6.17. Batería de pruebas para el caso CPSYS-017 Salir de un torneo en solitario	69
6.18. Batería de pruebas para el caso CPSYS-018 Ver próximos partidos	69
6.19. Batería de pruebas para el caso CPSYS-019 Ver próximos partidos (2)	69

6.20. Batería de pruebas para el caso CPSYS-020 Enviar solicitud de pareja	70
6.21. Batería de pruebas para el caso CPSYS-021 Aceptar solicitud de pareja	70
6.22. Batería de pruebas para el caso CPSYS-022 Rechazar solicitud de pareja	70
6.23. Batería de pruebas para el caso CPSYS-023 Historial de torneos	70
6.24. Batería de pruebas para el caso CPSYS-024 Historial de torneos (2)	71
6.25. Batería de pruebas para el caso CPSYS-025 Historial de partidos	71
6.26. Batería de pruebas para el caso CPSYS-026 Cerrar sesión	71
6.27. Batería de pruebas para el caso CPSYS-027 Crear un torneo	71
6.28. Batería de pruebas para el caso CPSYS-028 Crear un torneo (2)	72
6.29. Batería de pruebas para el caso CPSYS-029 Eliminar un torneo	72
6.30. Batería de pruebas para el caso CPSYS-030 Anotar resultado	72
6.31. Batería de pruebas para el caso CPSYS-031 Anotar resultado (2)	72
6.32. Batería de pruebas para el caso CPSYS-032 Generar ronda torneo en parejas	73
6.33. Batería de pruebas para el caso CPSYS-033 Generar ronda torneo en solitario	73
6.34. Batería de pruebas para el caso CPSYS-034 Concurrencia en unir a un torneo	73
6.35. Batería de pruebas para el caso CPSYS-035 Jugador ya unido a un torneo intenta unir con otro	73

Capítulo 1

Introducción

En el presente documento, se recoge todo lo relativo al trabajo de fin de grado ‘‘ Social Paddle: aplicación Android para la gestión de eventos deportivos’’, incluyendo aspectos técnicos y metodologías hasta la elaboración de los distintos manuales para la utilización del usuario.

La aplicación que se va a desarrollar se centra en gestionar eventos deportivos, en este caso el deporte en el que nos centraremos es el pádel. La aplicación esta orientada a centros deportivos, que dispongan de pistas propias y quieran promocionar su centro.

Esta aplicación permitirá a los usuarios tener mas facilidades para encontrar nuevos compañeros de juego, e incentivará una competitividad amistosa. Se busca con esta aplicación una mayor practica del deporte, ya que en los tiempos actuales existe demasiado sedentarismo.

1.1. Estructura de la memoria

- Capitulo 1: Introducción. En el aparecen la estructura de la memoria, contexto y los objetivos del proyecto.
- Capitulo 2: Tecnologías y servicios utilizados. Descripción de las herramientas y tecnologías utilizadas para el desarrollo.
- Capitulo 3: Planificación y gestión. Información sobre como se ha planificado y gestionado el proyecto, junto a sus riesgos incluyendo un presupuesto del coste.
- Capitulo 4: Análisis y diseño. Toda la información sobre la elicitación de requisitos, casos de uso, modelo de dominio, diagramas de secuencia, diagrama de despliegue y el diseño de la base de datos.
- Capitulo 5: Implementación. Explicación de la toma de decisiones.
- Capitulo 6: Pruebas. Pruebas realizadas junto a el resultado obtenido en ellas.

- Capitulo 7: Conclusiones. Recoge las conclusiones finales sobre la realización del proyecto, tanto los objetivos cumplidos como posibles líneas futuras de trabajo.
- Bibliografía. Referencias bibliográficas consultadas durante la realización del proyecto.
- Apéndices. Contiene los manuales para poder instalar y utilizar la aplicación.

1.2. Contexto y motivación

Las tecnologías móviles están en auge, los teléfonos móviles, sin duda, ocupan una posición muy importante en la estructura personal, social y económica de nuestras vidas. Los teléfonos móviles han evolucionado bastante en los últimos años, desde el primer dispositivo ‘Motorola DynaTAC 8000x’ que salió al mercado en el año 1983, hasta el día de hoy con teléfonos como el ‘Samsung Galaxy S20’.

Debido al auge de los dispositivos móviles decidí desarrollar la aplicación para el sistema operativo Android, el cual es más accesible económicamente que su competencia más directa Apple. Solo necesitas un ordenador que tenga unas características suficientes para hacer funcionar Android Studio, y si quieres subir tu aplicación a la ‘Play Store’ solo tienes que realizar un único pago de 20€. Otra de las razones es que se dispone de mucha documentación en internet, la cual te ayuda en el desarrollo.

El interés por aprender a desarrollar una aplicación en Android me venía de varios años atrás. El cuatrimestre pasado cursé la asignatura ‘Sistemas Móviles’. En esta asignatura se realizaba una pequeña aplicación para la práctica final. Con ella aprendí conceptos básicos y quería mejorar mis conocimientos, lo que me animó a realizar una aplicación más seria para el trabajo de fin de grado.

La razón de elegir un deporte como el pádel es que esta modalidad deportiva ha sido, sin lugar a dudas, la que mayor crecimiento experimenta en los últimos años en España.[21] Las razones son varias. Una de ellas es que al tratarse de un juego por parejas es ideal para practicarlo con amigos, esto te ayudara a mejorar tu trabajo en equipo. El practicar un deporte con amigos te ayudara a mantener una constancia, y esto te traerá mejoras físicas y de salud a largo plazo.

1.3. Objetivos

El objetivo principal del proyecto es el de desarrollar una aplicación, que permita a un centro deportivo gestionar distintos tipos de torneo. Para así promocionar su centro e intentar tener un mejor uso de sus pistas de pádel.

Los usuarios de la aplicación podrán unirse a los distintos torneos junto a una pareja, la cual se podrá buscar dentro de la propia aplicación. Los usuarios también podrán participar en los denominados torneos en solitario, en los cuales se buscara generar nuevas amistades

deportivas. La aplicación se encargara de intentar nivelar lo máximo posible los enfrentamientos, ya que los usuarios en función de si ganan o pierden irán sumando puntos de experiencia, estos puntos te permitirán ir subiendo o bajando de nivel dentro de la propia aplicación. Los torneos se categorizaran en función del nivel de habilidad recomendada por el administrador de la aplicación.

Los torneos se generaran en función de los equipos que se puedan apuntar (2, 4, 8, 16, 32). A partir de 4 equipos se generaran rondas clasificatorias, hasta que un equipo se proclame vencedor del torneo. Con el objetivo de evitar una fatiga en los equipos vencedores, las siguientes rondas se generaran una semana después.

El administrador del centro deportivo podrá crear, administrar y visualizar los torneos. Se encargara de anotar los resultados de los partidos y generar las nuevas rondas clasificatorias de cada torneo. Podrá visualizar todas las parejas que existen creadas hasta el momento, con el fin de tener la información sobre los clientes.

1.3.1. Objetivos de la aplicación

- Permitir a un administrador organizar torneos.
- Permitir a los clientes unirse a torneos.
- Guardar información sobre los torneos jugados.
- Buscar nuevos compañeros para jugar al pádel.
- Mejorar el interés por el deporte.
- Incentivar una competitividad amistosa.
- Superación personal mediante niveles de experiencia.
- Utilizar las nuevas tecnologías para evitar el sedentarismo.
- Promocionar un centro deportivo.

1.3.2. Objetivos personales

- Aprender y mejorar el conocimiento sobre las tecnologías móviles.
- Realizar una aplicación completa para Android.
- Implementar bases de datos NoSQL en una aplicación.
- Aprender a redactar una memoria de proyecto.
- Ganar nueva experiencia en el desarrollo software.
- Defender una idea ante de un tribunal.

1.3. OBJETIVOS

Capítulo 2

Tecnologías y servicios utilizados

En este capítulo se van a exponer las distintas tecnologías y servicios utilizados durante el desarrollo de este trabajo de fin de grado.

2.1. Android Studio



Figura 2.1: Logotipo Android Studio

Android studio [8] es el entorno de desarrollo oficial de Google. Fue publicado en diciembre de 2014 sustituyendo al actual entorno de desarrollo para aplicaciones móviles el cual era Eclipse. Android studio esta publicado bajo la licencia de apache 2.0. Está disponible para los sistemas operativos Windows, Mac OS y GNU/Linux.

El lenguaje preferido por Google para el desarrollo de aplicaciones Android es Kotlin pero Android Studio permite el uso de otros lenguajes como Java o C++.

Características principales:

- Soporte para la construcción basada en Gradle.
- Refactorización específica de Android.

- Plantillas para crear diseños comunes de Android.
- Soporte para programar aplicaciones en Android Wear.
- Soporte integrado para Google Cloud Plataform.
- Dispositivo virtual de Android que permite ejecutar y probar aplicaciones.
- Consola de desarrollador.

2.2. Java



Figura 2.2: Logotipo Java

Java [1] es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

2.3. Firebase



Figura 2.3: Logotipo Firebase

Firebase [10] es una plataforma de desarrollo de aplicaciones web y móviles desarrollada por Google en 2014. La plataforma se ha integrado en la denominada Google Cloud Platform, que utiliza un conjunto de herramientas para crear y sincronizar proyectos de alta calidad, haciendo posible la escalabilidad y ayudándote a obtener mejores beneficios económicos.

2.3.1. Firebase Auth

Firebase Auth [11] es un servicio que puede autenticar a los usuarios utilizando solo el código del cliente. Incluye autenticación a través de proveedores de inicio de sesión (como Facebook, GitHub, Twitter, Google, Yahoo y Microsoft); y métodos clásicos de inicio de sesión por correo electrónico y contraseña. Además, incluye un sistema de administración de usuarios a través del cual los desarrolladores pueden almacenar la autenticación de usuarios para correo electrónico y contraseñas en Firebase

El servicio tiene como objetivo facilitar la creación de sistemas de verificación de identidad, al tiempo que mejora la consolidación, el acceso y la seguridad del usuario. Por lo tanto, el cliente no necesita preocuparse por desarrollar métodos de autenticación clásicos, porque Firebase les proporciona una forma simple, efectiva y segura de administrar usuarios.

2.3.2. Realtime Database

Firebase [12] proporciona una base de datos back-end en tiempo real organizada en forma de árbol JSON. El servicio proporciona a los desarrolladores de aplicaciones una API que permite que la información de la aplicación se sincronice y se almacene en la nube de Firebase. La sincronización en tiempo real de la base de datos permite a los usuarios acceder a su información de datos en tiempo real desde cualquier dispositivo, compartir una instancia de la base de datos en tiempo real y almacenar la información cada vez que el usuario la modifica.

2.3.3. Firebase Storage

Firebase Storage [6] proporciona cargas y descargas de archivos seguras para aplicaciones de Firebase. El desarrollador puede usarlo para guardar imágenes, audio, vídeo u otro contenido generado por el usuario. Firebase Storage se basa en Google Cloud Storage.

2.4. GitHub



Figura 2.4: Logotipo GitHub

GitHub [3] es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de

código fuente de programas de ordenador. El software que opera GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. Anteriormente era conocida como Logical Awesome LLC.

2.5. Visual Paradigm



Figura 2.5: Logotipo Visual Paradigm

Visual Paradigm[24] es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

2.6. Microsoft Project



Figura 2.6: Logotipo Visual Microsoft Project

Microsoft Project[20] (o MSP) es un software de gestión de proyectos y un programa de proyectos desarrollado y vendido por Microsoft. Ayuda a los gerentes de proyectos a hacer planes, asignar recursos a tareas, dar seguimiento al progreso, administrar presupuestos y analizar cargas de trabajo.

Capítulo 3

Planificación y gestión

3.1. Planificación temporal

Antes de comenzar con el proyecto se tiene que planear de manera temporal las distintas fases del mismo, para esta planificación se tiene en cuenta que el desarrollador principal trabaja a jornada completa en una empresa y solo dispondrá de tiempo durante festivos y fines de semana, para la realización de este proyecto.

Para la planificación se ha utilizado la metodología en cascada, que es un modelo lineal de diseño software empleando un proceso de diseño secuencial. Las fases de desarrollo fluyen de manera secuencial desde el punto inicial hasta la fase final del proyecto.

Teniendo en cuenta los plazos de entrega estimados que más o menos serán para la ultima semana de junio y sabiendo que el comienzo lectivo del segundo cuatrimestre este año es el día 10 febrero. Tenemos unas 20 semanas para realizar el proyecto, sabiendo de cuanto tiempo disponemos, procedemos a dividir las semanas entre las fases de nuestro proyecto.

El proyecto sera dividido en cuatro fases: Análisis de requisitos, diseño, codificación y pruebas.

Análisis de requisitos (1 Semana): En esta fase, se realiza un análisis de las necesidades del cliente para determinar las propiedades del software a desarrollar, y se especifica todo lo que el sistema tiene que hacer sin entrar en detalles técnicos. Hay que ser especialmente cuidadoso en esta primera fase, ya que en este modelo no se pueden añadir nuevos requisitos en mitad del proceso de desarrollo.

Diseño (1 Semana): Esta fase describe la estructura interna del software y las relaciones entre las entidades que lo componen.

Codificación (15 Semanas): En esta fase, los requisitos especificados se programan utilizando las estructuras de datos diseñadas en la fase anterior. Esta es una de las fases mas

3.1. PLANIFICACIÓN TEMPORAL

largas ya que requiere muchas horas de codificación del software y es la que mas problemas puede conllevar.

Pruebas (1 Semana): Esta fase como su nombre indica, se trata de realizar baterías de prueba una vez haya finalizado la codificación, el objetivo de estas pruebas es el de obtener información sobre la calidad del software y sirve para encontrar defectos o bugs en el software.

La fase mas larga es la de Codificación por eso vamos a explicar en detalle cada una de las semanas.

Semana	Descripción
Semana 1	Implementar las pantallas de login, registro de usuario y cambiar contraseña.
Semana 2	Implementar el rol de administrador, añadiendo las funciones de crear un torneo y eliminarlo.
Semana 3	Implementar la pantalla para mostrar el perfil de un usuario.
Semana 4	Implementar la funcionalidades de buscar nuevas parejas, y enviar solicitud de pareja entre usuarios.
Semana 5	Implementar la funcionalidad para que un usuario pueda unirse a un torneo junto a una pareja.
Semana 6	Implementar la funcionalidad para que un administrador pueda generar nuevas rondas de partidos.
Semana 7	Implementar la funcionalidad para que un administrador pueda anotar las puntuaciones de cada partido jugado, y que estas se reflejen en el perfil del usuario.
Semana 8	Mitad del periodo de codificación, si no se sufre ninguna incidencia en las anteriores semanas. Esta semana se implementaran los torneos solitario.
Semana 9	Implementar una pantalla para que los usuarios puedan visualizar sus próximos partidos.
Semana 10	Implementar una pantalla para que el usuario pueda visualizar un historial de los torneos y partidos jugados.
Semana 11	Implementar la funcionalidad que permita añadir, y cambiar una foto de perfil.
Semana 12	Implementar funcionalidad para eliminar parejas.
Semana 13-15	Estas semanas se dejan como flotador en el caso de que en alguna semana se produzcan incidencias, que nos lleve a retrasos de tiempo en la codificación.

Cuadro 3.1: Reparto semanal fase: Codificación.

Para tener una visión mas global, he utilizado Microsoft Project para especificar todo el reparto de tiempo entre todas las fases. Microsoft Project me permite generar un diagrama de Gantt en función de las tareas y sus tiempos. En este diagrama se puede ver el camino critico del proyecto y vemos qué tareas son dependientes de otras.

CAPÍTULO 3. PLANIFICACIÓN Y GESTIÓN

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
4 Análisis de requisitos	1 sem	lun 10/02/20	vie 14/02/20		David Pastor
Reunión con clientes	1 día	lun 10/02/20	lun 10/02/20		David Pastor
Elaboración de requisitos	1 día	mar 11/02/20	mar 11/02/20	2	David Pastor
Modelo de dominio	1 día	mié 12/02/20	mié 12/02/20	3	David Pastor
Descripción de casos de uso	1 día	jue 13/02/20	jue 13/02/20	4	David Pastor
Diagrama de casos de uso	1 día	vie 14/02/20	vie 14/02/20	5	David Pastor
4 Diseño	1 sem	lun 17/02/20	vie 21/02/20	1	David Pastor
Diseño estructura de datos BD NOSQL	2 días	lun 17/02/20	mar 18/02/20		David Pastor
Diseño interfaz gráfica	2 días	mié 19/02/20	jue 20/02/20	8	David Pastor
4 Codificación	15 sem.	lun 24/02/20	vie 05/06/20	7	David Pastor
Pantalla login y registro de usuario	1 sem	lun 24/02/20	vie 28/02/20		David Pastor
Crear y eliminar torneos	1 sem	lun 02/03/20	vie 06/03/20	11	David Pastor
Perfil de usuario	1 sem	lun 09/03/20	vie 13/03/20	12	David Pastor
Buscar pareja y enviar solicitudes de pareja	1 sem	lun 16/03/20	vie 20/03/20	13	David Pastor
Unión a torneo con pareja	1 sem	lun 23/03/20	vie 27/03/20	14	David Pastor
Generar ronda de partidos	1 sem	lun 30/03/20	vie 03/04/20	15	David Pastor
Anotar puntuaciones	1 sem	lun 06/04/20	vie 10/04/20	16	David Pastor
Unión a torneos aleatorios	1 sem	lun 13/04/20	vie 17/04/20	17	David Pastor
Visualizar próximos partidos	1 sem	lun 20/04/20	vie 24/04/20	18	David Pastor
Visualizar historial de torneos	1 sem	lun 27/04/20	vie 01/05/20	19	David Pastor
Eliminar parejas	1 sem	lun 04/05/20	vie 08/05/20	20	David Pastor
Pruebas	1 sem	lun 08/06/20	vie 12/06/20	10	David Pastor

Figura 3.1: Tareas

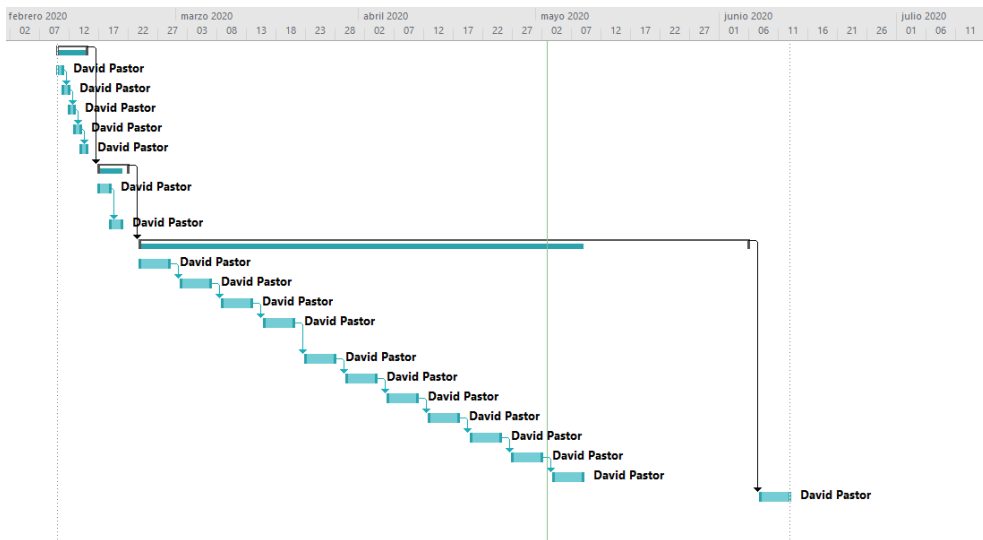


Figura 3.2: Diagrama de Gantt

3.2. Gestión de riesgos

En este apartado vamos a exponer los riesgos que puede tener la realización del proyecto. Todo proyecto tiene unos riesgos, algunos de ellos se pueden asumir, pero para otros tienes que tener un plan, el cual busque reducir el riesgo lo máximo posible, y si no es posible reducir el riesgo tienes que tener pensado cómo mitigar lo máximo posible las consecuencias del mismo. A continuación se muestra una serie de tablas en las que se recogen cada uno de los riesgos junto a su plan de contingencia.

Riesgo	Estimación del tiempo y coste no realista
Probabilidad	Media.
Causa	Mala planificación
Consecuencia	Retraso en la previsión/Aumento del presupuesto.
Impacto	Alto.
Estrategia	Mitigación del riesgo.
Plan de contingencia	Realizar la planificación dejando un tiempo extra en cada fase.

Cuadro 3.2: Estimación del tiempo y coste no realista

Riesgo	Problemas de rendimiento
Probabilidad	Baja.
Causa	Mal diseño software
Consecuencia	Aumento en la previsión.
Impacto	Alto.
Estrategia	Mitigación del riesgo.
Plan de contingencia	Realizar pruebas de carga sobre los algoritmos mas pesados.

Cuadro 3.3: Problemas de rendimiento

Riesgo	Exceso de dificultad en el desarrollo
Probabilidad	Media.
Causa	Personal inexperto
Consecuencia	Aumento en la previsión.
Impacto	Alto.
Estrategia	Mitigación del riesgo.
Plan de contingencia	Realizar una planificación dejando tiempo extra en las fases mas complejas.

Cuadro 3.4: Exceso de dificultad en el desarrollo

Riesgo	Falta de personal por baja laboral.
Probabilidad	Media.
Causa	Enfermedad/Problemas personales
Consecuencia	Aumento en la previsión.
Impacto	Alto.
Estrategia	Aceptación del riesgo.
Plan de contingencia	Se acepta el riesgo.

Cuadro 3.5: Falta de personal por baja laboral.

Riesgo	Desarrollo de interfaces erróneo.
Probabilidad	Baja.
Causa	Falta de comunicación
Consecuencia	Aumento en la previsión.
Impacto	Bajo.
Estrategia	Mitigación del riesgo.
Plan de contingencia	Realizar prototipos para tener claro las interfaces.

Cuadro 3.6: Desarrollo de interfaces erróneo.

Riesgo	Perdida de datos.
Probabilidad	Baja.
Causa	Rotura de unidades de almacenamiento
Consecuencia	Aumento en la previsión.
Impacto	Alto.
Estrategia	Supresión del riesgo.
Plan de contingencia	Realizar tres copias de seguridad después de cada jornada (almacenamiento en la nube, almacenamiento externo, almacenamiento interno).

Cuadro 3.7: Perdida de datos.

Para intentar reducir la mayoría de estos riesgos se debe de tener una buena comunicación con el cliente. También debemos de tener una revisión periódica de los avances, para intentar ajustarnos a los tiempos de la planificación. Para mitigar posibles riesgos también se tiene que tener un tiempo de margen en todas las etapas del proyecto, esto se conoce como ‘float’.

3.3. Presupuesto

Todo proyecto software tiene unos costes, los cuales se pueden dividir en tres partes: hardware, software y personal. A continuación vamos a realizar un desglose de precios de los tres componentes.

3.3.1. Hardware

En este apartado se recoge todo el equipo hardware utilizado y necesario para el desarrollo de la aplicación. Para realizar el coste por hora de los dispositivos hardware se toma como referencia la vida útil del dispositivo y que este se utilizara para un uso laboral de 8 horas al día durante cinco días a la semana. Un portátil suele tener una vida útil de unos 5 años. Los dispositivos móviles suelen tener una vida útil entre 18 y 24 meses en función de la calidad del dispositivo. Las pantallas de LED suelen tener una vida útil de unas 100.000 horas de uso.

Descripción	Coste unitario	Coste por hora	Coste
Portátil - ASUS VivoBook 15	478 €	0,049 €	14,7 €
Móvil - Xiaomi Redmi Note 4	180 €	0,046 €	13,8 €
Pantalla - Samsung S22F350FHU 22"	90 €	0,0009 €	0,27 €
Total			28,77 €

Cuadro 3.8: Coste hardware.

3.3.2. Software

En este apartado se incluye todo el coste de las herramientas software utilizadas durante el desarrollo, varias de las aplicaciones son proporcionadas por la escuela de Ingeniería Informática de Valladolid y su coste total es 0 € para los alumnos.

Descripción	Coste unitario
Android Studio	0 €
Visual paradigm 15	0 €
Firebase plan "Spark"	0 €
Licencia Windows 10	0 €
Total	0 €

Cuadro 3.9: Coste software.

3.3.3. Personal

En este apartado se tiene en cuenta todo el coste personal para el desarrollo de una aplicación, en este caso solo se dispone de un trabajador. Para el calculo se estima un salario bruto de 17000 € anual dividido en 12 pagas esto da un sueldo mensual de 1326 € al mes con pagas extra prorrateadas si dividimos esto entre 40 horas semanales nos sale a unos 11 €/H.

Descripción	Coste unitario	Horas
Salario	11 €/H	300
Total	3300 €	

Cuadro 3.10: Coste personal.

3.3.4. Coste total

En la siguiente tabla se recoge la suma de todos los presupuestos, con el fin de obtener el coste total del proyecto software.

Descripción	Coste
Coste hardware	28,77 €
Coste software	0 €
Coste personal	3300 €
Total	3328,77 €

Cuadro 3.11: Coste total.

Capítulo 4

Análisis y diseño

4.1. Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares. En algunos casos, los requisitos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer. A continuación se listan todos los requisitos funcionales.

Código	Requisito	Descripción
FRQ-001	Inicio de sesión	El sistema deberá proporcionar un sistema de autenticación para los usuarios.
FRQ-002	Crear torneos	El sistema deberá permitir a un administrador crear torneos.
FRQ-003	Registro de usuarios	El sistema deberá proporcionar un sistema de registro para los nuevos usuarios.
FRQ-004	Eliminar torneos	El sistema deberá permitir eliminar torneos ya creados
FRQ-005	Generar nuevos partidos	El sistema deberá permitir generar partidos de un torneo ya creado.
FRQ-006	Anotar puntuaciones de partidos ya jugados	El sistema deberá permitir anotar las puntuaciones de los partidos ya jugados.
FRQ-007	Visualizar los datos de un perfil usuario	El sistema deberá permitir a un usuario visualizar los datos del perfil.
FRQ-008	Subir y editar una foto de perfil	El sistema deberá permitir subir y editar la foto de perfil.
FRQ-009	Activar o desactivar el modo búsqueda de pareja	El sistema deberá permitir a un usuario activar o desactivar la búsqueda de una pareja.

Continúa en la siguiente página

4.2. REQUISITOS DE INFORMACIÓN

Código	Requisito	Descripción
FRQ-010	Buscar nuevas parejas	El sistema deberá permitir buscar nuevas parejas y enviar una solicitud de equipo.
FRQ-011	Aceptar o rechazar peticiones de pareja	El sistema deberá permitir aceptar o rechazar las peticiones de pareja.
FRQ-012	Búsqueda de torneos	El sistema deberá permitir la búsqueda de torneos nuevos.
FRQ-013	Torneo con parejas	El sistema deberá permitir unirse a un torneo con una pareja.
FRQ-014	Torneo solitario	El sistema deberá permitir unirse a un torneo de manera individual.
FRQ-015	Salir de un torneo	El sistema deberá permitir salir de un torneo el cual no haya comenzado.
FRQ-016	Próximos partidos	El sistema deberá permitir a un usuario visualizar los próximos partidos que tendrá que disputar.
FRQ-017	Historial de torneos	El sistema deberá permitir visualizar un histórico de los torneos ya jugados.
FRQ-018	Generar partidos igualados	El sistema deberá generar los partidos con la máxima equidad posible entre jugadores.
FRQ-019	Sistema de niveles	Los usuarios dispondrán de un nivel de habilidad la cual subirá o bajara en función de las victorias o derrotas.
FRQ-020	Niveles de habilidad	Los jugadores podrán elegir entre siete niveles de habilidad (Novato, Novato entrenado, Jugador casual, Jugador aficionado, Jugador experto, Jugador profesional, Jugador invicto).
FRQ-021	Eliminar vinculo de parejas	El sistema deberá permitir a los usuarios eliminar una de sus parejas.
FRQ-022	Cambiar contraseña	El sistema permitirá a un usuario modificar su contraseña.

Cuadro 4.1: Requisitos funcionales.

4.2. Requisitos de información

FRQ-023	Información del usuario
Descripción	El sistema deberá almacenar los datos de los usuarios (nombre, apellidos, correo, habilidad, puntos de habilidad, parejas, partidos jugados, partidos ganados, partidos perdidos, peticiones de pareja, estado de búsqueda de pareja)

Cuadro 4.2: FRQ-023 Información del usuario.

FRQ-024	Información del torneo
Descripción	El sistema deberá almacenar los datos de los torneos (equipos apuntados, nombre, fecha comienzo, estado del torneo, modo de emparejamiento, nivel de habilidad, numero de equipos, numero de rondas, rondas generadas, solicitudes de unión)

Cuadro 4.3: FRQ-024 Información del torneo.

FRQ-025	Información del partido
Descripción	El sistema deberá almacenar los datos de los partidos (equipo uno, equipo dos, fecha, resultado, nombre del torneo, estado del partido, numero de ronda)

Cuadro 4.4: FRQ-025 Información del partido.

FRQ-026	Información de las parejas
Descripción	El sistema deberá almacenar los datos de las parejas (jugador uno, jugador dos, tipo de pareja)

Cuadro 4.5: FRQ-026 Información de las parejas.

4.3. Requisitos no funcionales

Requisitos no funcionales, son requisitos que imponen restricciones en el diseño o la implementación como restricciones en el diseño o estándares de calidad.

NFRQ-001	Compatibilidad de la aplicación
Descripción	La aplicación debe ser compatible con la versión Android SDK 22 (Android 5.1).

Cuadro 4.6: NFRQ-001 Compatibilidad de la aplicación.

NFRQ-002	Firebase Realtime
Descripción	La aplicación tiene que utilizar el servidor de bases de datos NoSQL de Firebase

Cuadro 4.7: NFRQ-002 Firebase Realtime.

NFRQ-003	Idioma de la aplicación
Descripción	La aplicación tiene que mostrar toda la información en Español.

Cuadro 4.8: NFRQ-003 Idioma de la aplicación.

NFRQ-004	Autenticación con correo único
Descripción	La autenticación de la cuenta tiene que ser con un correo único.

Cuadro 4.9: NFRQ-003 Autenticación con correo único.

NFRQ-005	Firebase Authentication
Descripción	La aplicación tiene que usar el servicio de autenticación de Firebase.

Cuadro 4.10: NFRQ-005 Firebase Authentication.

NFRQ-006	Firestore Storage
Descripción	La aplicación tiene que utilizar los servicios de Firestore Storage para almacenar los ficheros de la aplicación

Cuadro 4.11: NFRQ-006 Firestore Storage.

4.4. Casos de uso

4.4.1. Definición de actores

La aplicación esta basada en dos roles, por un lado tenemos al administrador el cual se encargara de la gestión de los torneos, anotando las puntuaciones y generando las rondas de partidos.

El otro rol sera el jugador, el cual puede buscar a otros jugadores para añadir a su lista de parejas, o puede unirse a los torneos en pareja o en solitario. En este último caso la aplicación le buscará pareja. También podrá ir visualizando el historial de torneos en los que ya ha participado anteriormente.

ACT-001	Administrador
Autores	David Pastor
Descripción	Este actor representa al encargado de gestionar los torneos de pádel en el centro deportivo.

Cuadro 4.12: ACT-001 Administrador.

ACT-002	Jugador
Autores	David Pastor
Descripción	Este actor representa a los miembros activos de la comunidad de jugadores de pádel, pertenecientes al centro deportivo.

Cuadro 4.13: ACT-002 Jugador.

4.4.2. Diagrama de casos de uso

A continuación se muestra el diagrama de casos de uso, el cual ha sido realizado mediante la información obtenida en los pasos previos. En el diagrama se observan los distintos casos de uso que pueden realizar cada uno de los actores.



Figura 4.1: Diagrama de casos de uso

4.4.3. Descripción de casos de uso

A continuación se describe cada uno de los casos de uso que han sido representados en el diagrama anterior.

UC-001	Crear torneo
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Administrador.
Dependencias	Ninguno.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Administrador quiere crear un nuevo torneo.
Precondición	El actor Administrador tiene que estar autenticado en el sistema.
Secuencia normal	1-El actor Administrador pulsa sobre el botón añadir torneo 2-El sistema muestra una ventana para insertar los datos sobre el torneo 3-El actor Administrador introduce los datos sobre el torneo 4-El sistema pide una confirmación de los datos. 5-El actor Administrador confirma los datos 6-El sistema genera el nuevo torneo y finaliza el caso de uso.
Postcondición	Se almacenan los datos del nuevo torneo.
Excepción	2.a-El actor Administrador pulsa el botón salir. 4.a-El sistema detecta que los datos son erróneos, el caso de uso continua en el paso 2 5.a-El actor Administrador pulsa el botón salir

Cuadro 4.14: UC-001 Crear torneo.

4.4. CASOS DE USO

UC-002	Anotar resultados
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Administrador.
Dependencias	Ninguno.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Administrador quiere publicar las puntuaciones de un partido ya jugado.
Precondición	El actor Administrador tiene que estar autenticado en el sistema.
Secuencia normal	1-El actor Administrador pulsa sobre el botón anotar resultados. 2-El sistema muestra todos los partidos que existen sin anotar. 3-El actor Administrador inserta la puntuación y pulsa sobre el botón guardar. 4-El sistema comprueba si la información es correcta. 5-El sistema guarda la información y bloquea el botón guardar.
Postcondición	Se almacenan los datos sobre el partido anotado.
Excepciones	2.a-El actor Administrador pulsa el botón salir. 4.a-La información no es correcta continua el caso de uso en el paso 2.

Cuadro 4.15: UC-002 Anotar resultados.

UC-003	Ver torneos
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Administrador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Administrador quiere visualizar todos los torneos generados.
Precondición	El actor Administrador tiene que estar autenticado en el sistema.
Secuencia normal	1-El actor Administrador pulsa sobre el botón ver torneos. 2-El sistema muestra todos los torneos generados hasta el momento. 3-El actor Administrador pulsa sobre uno de los torneos. 4-El sistema muestra los datos sobre el torneo.
Postcondición	No existe una postcondición
Excepciones	3.a-El actor Administrador pulsa el botón salir.

Cuadro 4.16: UC-003 Ver torneos.

UC-004	Generar nueva ronda
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Administrador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Administrador quiere generar una nueva ronda de partidos en un torneo.
Precondición	El actor Administrador tiene que estar autenticado en el sistema.
Secuencia normal	1-Se realiza el caso de uso Ver torneos (UC-003). 2-El actor Administrador pulsa sobre el botón Generar nueva ronda. 3-El sistema comprueba el tipo de torneo y el número de ronda, acto seguido genera una nueva ronda de partidos.
Postcondición	Se almacenan los datos sobre los nuevos partidos.
Excepciones	2.a-El actor Administrador pulsa el botón salir.

Cuadro 4.17: UC-004 Generar nueva ronda.

UC-005	Eliminar torneo
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Administrador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Administrador quiere eliminar un torneo de la lista.
Precondición	El actor Administrador tiene que estar autenticado en el sistema. Tiene que existir al menos un torneo ya creado en el sistema.
Secuencia normal	1-Se realiza el caso de uso Ver torneos (UC-003). 2-El actor Administrador pulsa sobre el botón 'Eliminar torneo'. 3-El sistema elimina el torneo indicado.
Postcondición	Se eliminan los datos del torneo en la base de datos.
Excepciones	2.a-El actor administrador pulsa el botón salir.

Cuadro 4.18: UC-005 Eliminar torneo.

4.4. CASOS DE USO

UC-006	Registrarse en el sistema
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador quiere registrarse en el sistema para poder utilizar sus servicios.
Precondición	Ninguna precondición.
Secuencia normal	1-El actor Jugador pulsa sobre el botón registrar usuario. 2-El sistema muestra un formulario de registro. 3-El actor Jugador inserta los datos del registro y pulsa confirmar. 4-El sistema comprueba los datos. 5-El sistema guarda los datos y vuelve a la pantalla de inicio.
Postcondición	Se guardan los datos sobre el nuevo usuario en el sistema
Excepciones	3.a-El actor Administrador pulsa el botón salir. 4.a-Los datos introducidos son incorrectos, el caso de uso continua en el paso 2.

Cuadro 4.19: UC-006 Registrarse en el sistema.

UC-007	Ver perfil de usuario
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador quiere ver toda su información almacenada en el sistema.
Precondición	El actor Jugador tiene que estar autenticado en el sistema.
Secuencia normal	1-El actor Jugador pulsa sobre el botón Ver perfil de usuario. 2-El sistema muestra todos los datos del usuario.
Postcondición	No existe una postcondición
Excepciones	No existen excepciones.

Cuadro 4.20: UC-007 Ver perfil de usuario.

UC-008	Activar buscar pareja
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador quiere ser visible para que el resto de jugadores puedan solicitar su amistad.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. La búsqueda de pareja tiene que estar desactivada.
Secuencia normal	1-Se realiza el caso de uso Ver perfil de usuario (UC-007). 2-El actor Jugador pulsa sobre el botón Activar búsqueda de pareja. 3-El sistema activa (vuelve visible) al usuario para buscar pareja.
Postcondición	Se almacenan los datos sobre el jugador en el sistema.
Excepciones	2.a-El actor administrador pulsa el botón salir.

Cuadro 4.21: UC-008 Activar buscar de pareja.

UC-009	Desactivar buscar pareja
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador no quiere ser mostrado en la lista de jugadores que buscan pareja.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. La búsqueda de pareja tiene que estar activada.
Secuencia normal	1-Se realiza el caso de uso Ver perfil de usuario (UC-007). 2-El actor Jugador pulsa sobre el botón Desactivar búsqueda de pareja. 3-El sistema elimina al usuario de la lista para buscar pareja y muestra la información.
Postcondición	Se almacenan los datos sobre el jugador en el sistema
Excepciones	2.a-El actor administrador pulsa el botón salir.

Cuadro 4.22: UC-009 Desactivar búsqueda de pareja.

4.4. CASOS DE USO

UC-010	Subir foto de perfil
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación cuando un actor Jugador quiere subir una imagen como foto de perfil.
Precondición	El actor Jugador tiene que estar autenticado en el sistema.
Secuencia normal	1-Se realiza el caso de uso Ver perfil de usuario (UC-007). 2-El actor Jugador pulsa sobre el botón subir foto. 3-El sistema muestra el selector de archivos. 4-El actor Jugador selecciona una imagen. 5-El sistema muestra la imagen seleccionada en pantalla. 6-El actor Jugador confirma la imagen. 7-El sistema guarda la foto y finaliza el caso de uso.
Postcondición	Se almacena la imagen en el sistema
Excepción	2.a-El actor pulsa el botón salir. 4.a-El actor pulsa el botón salir. 6.a-El actor Jugador quiere cambiar de imagen y pulsa el botón de seleccionar una imagen, el caso de uso continua en el paso 3.

Cuadro 4.23: UC-010 Subir foto de perfil.

UC-011	Ver pareja
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador quiere visualizar las parejas que tiene asignadas.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. El actor Jugador tiene que tener mínimo una pareja.
Secuencia normal	1-Se realiza el caso de uso Ver perfil de usuario (UC-007). 2-El actor Jugador pulsa el botón ver parejas. 3-El sistema muestra en pantalla todas las parejas que tiene el usuario.
Postcondición	No existe una postcondición
Excepción	2.a-El actor pulsa el botón salir y finaliza el caso de uso.

Cuadro 4.24: UC-011 Ver pareja.

UC-012	Eliminar pareja
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador quiere dejar de formar equipo con otro jugador del sistema.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. El actor Jugador tiene que tener mínimo una pareja.
Secuencia normal	1-Se realiza el caso de uso Ver perfil de usuario (UC-007). 2-Se realiza el caso de uso Ver parejas (UC-011). 3-El actor Jugador pulsa sobre el botón eliminar pareja 5-El sistema elimina la pareja seleccionada y finaliza el caso de uso.
Postcondición	Se actualizan los datos en el sistema
Excepción	3.a-El actor Jugador pulsa sobre el botón salir.

Cuadro 4.25: UC-012 Eliminar pareja.

UC-013	Ver historial de torneos
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador quiere ver los torneos que ha disputado en el pasado.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. El actor Jugador tiene que haber jugado y finalizado mínimo un torneo.
Secuencia normal	1-El actor Jugador pulsa sobre el botón Historial. 2-El sistema muestra todos los torneos en los que ha participado el actor.
Postcondición	No existe una postcondición
Excepción	No existen excepciones.

Cuadro 4.26: UC-013 Ver historial de torneos.

4.4. CASOS DE USO

UC-014	Ver partidos del torneo
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador quiere ver los partidos que se jugaron en el torneo.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. El actor Jugador tiene que haber jugado y finalizado mínimo un torneo.
Secuencia normal	1-Se realiza el caso de uso Ver historial de torneos (UC-013). 2-El actor Jugador pulsa sobre un torneo. 3-El sistema muestra todos los partidos jugados en ese torneo y finaliza el caso de uso.
Postcondición	No existe una postcondición
Excepción	2.a-El actor pulsa el botón salir y finaliza el caso de uso.

Cuadro 4.27: UC-014 Ver partidos del torneo.

UC-015	Buscar pareja
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador quiere encontrar una nueva pareja para jugar junto a ella.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. En el sistema tiene que haber un jugador que este buscando pareja.
Secuencia normal	1-El actor Jugador pulsa sobre el botón de buscar pareja. 2-El sistema muestra todos los jugadores que tienen activado el modo buscar pareja. 3-El actor Jugador pulsa sobre uno de los usuarios mostrados. 4-El sistema muestra los detalles del usuario. 5-El actor Jugador pulsa sobre el botón enviar solicitud. 6-El sistema envía la solicitud al jugador seleccionado y finaliza el caso de uso.
Postcondición	Se almacena los datos sobre la nueva petición.
Excepción	2.a-El actor Jugador pulsa el botón salir. 5.a-El actor Jugador pulsa el botón salir.

Cuadro 4.28: UC-015 Buscar pareja.

UC-016	Buscar torneos en pareja
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador quiere unirse a un torneo junto a una pareja ya formada con anterioridad.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. El actor Jugador tiene que tener al menos una pareja. 3-Tiene que existir al menos un torneo en el sistema.
Secuencia normal	1-El actor Jugador pulsa el botón Buscar torneos. 2-El sistema muestra todos los torneos disponibles. 3-El actor pulsa sobre un torneo en parejas. 4-El sistema muestra los datos sobre el torneo. 5-El actor selecciona una pareja con la que unirse. 6-El actor pulsa el botón unirse a torneo. 7-El sistema guarda los datos sobre ese usuario y su pareja.
Postcondición	Se almacenan los datos sobre la pareja en el sistema.
Excepción	2.a-El actor Jugador pulsa el botón salir y finaliza el caso de uso. 3.a-El actor Jugador pulsa el botón salir y finaliza el caso de uso. 6.a-La pareja ya esta apuntada con otra persona el caso de uso continua en el paso 4.

Cuadro 4.29: UC-016 Buscar torneos en pareja.

UC-017	Buscar torneos en solitario
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador desea unirse a un torneo para jugar con otra pareja.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. Tiene que existir al menos un torneo en el sistema.
Secuencia normal	1-El actor Jugador pulsa el botón Buscar torneos. 2-El sistema muestra todos los torneos disponibles. 3-El actor pulsa sobre un torneo en solitario. 4-El sistema muestra los datos sobre el torneo. 5-El actor pulsa el botón unirse a torneo. 6-El sistema comprueba que este usuario no este apuntado en el torneo. 7-El sistema guarda los datos sobre ese usuario.
Postcondición	Se almacenan los datos sobre el jugador en el sistema.
Excepción	2.a-El actor Jugador pulsa el botón salir y finaliza el caso de uso. 3.a-El actor Jugador pulsa el botón salir y finaliza el caso de uso. 6.a-El jugador ya esta apuntado en el torneo el caso de uso continua en el paso 4.

Cuadro 4.30: UC-07 Buscar torneos en solitario.

UC-018	Salir del torneo
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador desea salir de un torneo en el que ya estaba apuntado.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. El actor Jugador tiene que estar apuntado en un torneo. 3-El torneo no puede haber empezado
Secuencia normal	1-El actor Jugador pulsa el botón Buscar torneos. 2-El sistema muestra todos los torneos disponibles. 3-El actor pulsa sobre un torneo en el que este apuntado. 4-El sistema muestra los datos sobre el torneo. 5-El actor pulsa el botón salir del torneo. 6-El sistema elimina a ese jugador del torneo.
Postcondición	Se almacenan los datos en el sistema.
Excepción	3.a-El actor Jugador pulsa el botón salir y finaliza el caso de uso. 5.a-El actor Jugador no esta apuntado al torneo entonces finaliza el caso de uso.

Cuadro 4.31: UC-018 Salir del torneo.

UC-019	Aceptar solicitud de pareja
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador quiere aceptar una solicitud de pareja.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. El actor Jugador tiene que tener solicitudes pendientes.
Secuencia normal	1-El actor Jugador pulsa sobre el botón Ver solicitudes 2-El sistema muestra todas las solicitudes. 3-El actor pulsa sobre el botón aceptar 4-El sistema asigna esa nueva pareja a los dos jugadores y elimina de la solicitud.
Postcondición	Se almacenan los datos sobre la nueva pareja en el sistema
Excepción	3.a-El actor Jugador pulsa el botón salir y finaliza el caso de uso.

Cuadro 4.32: UC-019 Aceptar solicitud de pareja.

4.4. CASOS DE USO

UC-020	Rechazar solicitud de pareja
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador quiere rechazar una solicitud de pareja.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. El actor Jugador tiene que tener solicitudes pendientes.
Secuencia normal	1-El actor Jugador pulsa sobre el botón Ver solicitudes. 2-El sistema muestra todas las solicitudes. 3-El actor pulsa sobre el botón rechazar 4-El sistema elimina la solicitud.
Postcondición	Se actualizan los datos sobre las peticiones en el sistema.
Excepción	3.a-El actor Jugador pulsa el botón salir y finaliza el caso de uso.

Cuadro 4.33: UC-020 Rechazar solicitud de pareja.

UC-021	Ver próximos partidos.
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador quiere saber cuales van a ser los proximos partidos que tiene que jugar.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. El actor Jugador tiene que estar apuntado en un torneo cerrado.
Secuencia normal	1-El actor Jugador pulsa sobre el botón Ver próximos partidos 2-El sistema muestra los partidos siguientes a la fecha actual.
Postcondición	No existe una postcondición.
Excepción	No existen excepciones.

Cuadro 4.34: UC-021 Ver próximos partidos.

UC-022	Cambiar contraseña.
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador desea cambiar su contraseña de inicio.
Precondición	El actor Jugador tiene que estar autenticado en el sistema.
Secuencia normal	1-El actor Jugador pulsa sobre el botón cambiar contraseña 2-El sistema muestra una ventana en la que pide que introduzca su anterior contraseña y la nueva. 3-El actor jugador inserta los datos 4-El sistema comprueba que los datos sean correctos 5-El sistema cambia la contraseña y le indica al usuario que ya puede iniciar sesión con su nueva contraseña
Postcondición	Se almacena la nueva contraseña.
Excepción	4.a-Los datos son incorrectos el caso de uso continua en el paso 2.

Cuadro 4.35: UC-022 Cambiar contraseña.

UC-023	Aceptar solicitud de torneo.
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador desea aceptar una invitación para unirse a un torneo.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. El actor jugador tiene que tener una petición pendiente de otro jugador.
Secuencia normal	1-El actor Jugador pulsa sobre el botón solicitudes de torneo 2-El sistema muestra todas las solicitudes disponibles para ese jugador. 3-El actor pulsa sobre el botón aceptar 4-El sistema almacena la nueva pareja en el torneo
Postcondición	Se almacenan los datos.
Excepción	3.a-El actor pulsa el botón cancelar y finaliza el caso de uso.

Cuadro 4.36: UC-023 Aceptar solicitud de torneo.

4.4. CASOS DE USO

UC-024	Rechazar solicitud de torneo.
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Jugador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Jugador desea rechazar una invitación para unirse a un torneo.
Precondición	El actor Jugador tiene que estar autenticado en el sistema. El actor jugador tiene que tener una petición pendiente de otro jugador.
Secuencia normal	1-El actor Jugador pulsa sobre el botón solicitudes de torneo 2-El sistema muestra todas las solicitudes disponibles para ese jugador. 3-El actor pulsa sobre el botón rechazar 4-El sistema elimina la solicitud
Postcondición	Se almacenan los datos.
Excepción	3.a-El actor pulsa el botón cancelar y finaliza el caso de uso.

Cuadro 4.37: UC-024 Rechazar solicitud de torneo.

UC-025	Consultar equipos.
Autores	David Pastor
Fuentes	María Rosario Abril Raymundo
Actor principal	Administrador.
Dependencias	Ninguna.
Descripción	El sistema deberá comportarse tal como se describe a continuación, cuando un actor Administrador desea visualizar todos los equipos creados en el sistema.
Precondición	El actor Administrador tiene que estar autenticado en el sistema. Tienen que existir parejas creadas en el sistema.
Secuencia normal	1-El actor Administrador pulsa sobre el botón ver parejas. 2-El sistema muestra todas las parejas creadas hasta el momento.
Postcondición	No existe postcondición.
Excepción	No existe excepción.

Cuadro 4.38: UC-025 Consultar equipos.

4.5. Modelo de dominio

En la siguiente figura se muestra el modelo de dominio de las distintas clases.

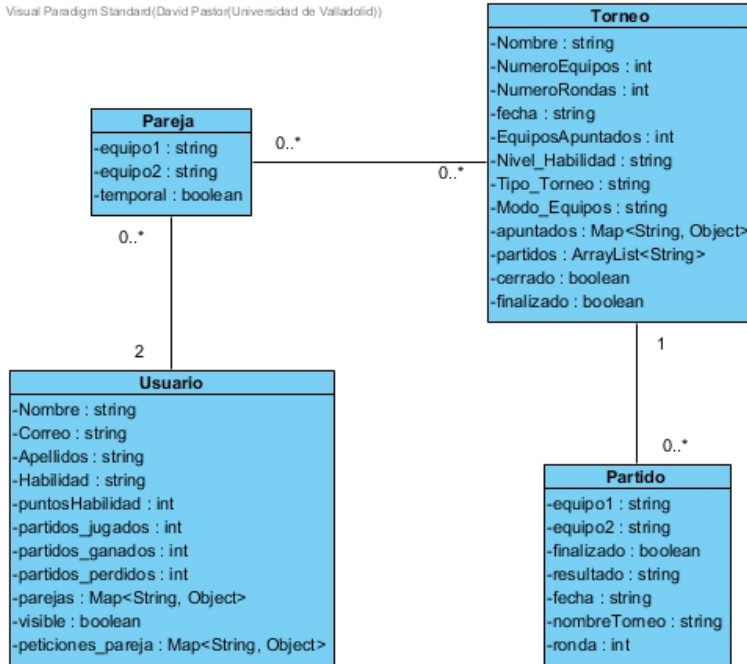


Figura 4.2: Modelo de dominio

4.5.1. Descripción de las clases.

Usuario: clase que modela a un usuario de la aplicación.

- Nombre: almacena el nombre del usuario.
- Correo: almacena la dirección de correo.
- Apellidos: apellidos del usuario.
- Habilidad: nivel de habilidad que posee el usuario.
- puntosHabilidad: almacena el número de puntos del usuario.
- partidos_jugados: número de partidos jugados.
- partidos_ganados: número de partidos ganados.
- partidos_perdidos: número de partidos perdidos.

- parejas: almacena todas las parejas de un usuario.
- visible: indica si el usuario esta activo en la búsqueda de pareja.
- peticiones_pareja: almacena todas las peticiones de pareja que tiene el usuario

Pareja: clase que modela una pareja de dos usuarios.

- equipo1: almacena el id del equipo 1.
- equipo2: almacena el id del equipo 2.
- temporal: indica si la pareja es temporal o no.

Torneo: clase que modela un torneo.

- Nombre: almacena el nombre que tiene el torneo.
- NumeroEquipos: almacena el numero de equipos máximos que se pueden apuntar.
- NumeroRondas: almacena el numero de rondas que tendra el torneo en funcion del numero de equipos.
- fecha: fecha del torneo.
- EquiposApuntados: almacena el numero de equipos apuntados en el torneo.
- Nivel_Habilidad: nivel de habilidad recomendado para el torneo.
- Tipo_Torneo: almacena el tipo de torneo(gratuito, benéfico o de pago).
- Modo_Equipos: almacena el modo de emparejamiento puede ser con pareja o solitario.
- apuntados: almacena el id de todas las parejas apuntadas.
- partidos: almacena el id de todos los partidos del torneo.
- cerrado: indica si el torneo esta empezado o no.
- finalizado: indica si el torneo ha finalizado.

Partido: clase que modela un partido.

- equipo1: id del equipo 1.
- equipo2: id del equipo 2.
- finalizado: indica si el partido ha finalizado.
- resultado: indica el resultado del partido.
- fecha: fecha del partido.
- nombreTorneo: nombre del torneo asignado a este partido.
- ronda: numero de ronda en la que se encuentra este partido.

4.6. Diagrama de secuencia

En este apartado se muestran todos los diagramas de secuencia de cada uno de los casos de uso, se omiten las clases utilizadas para el almacenamiento y la recuperación de datos en Firebase.

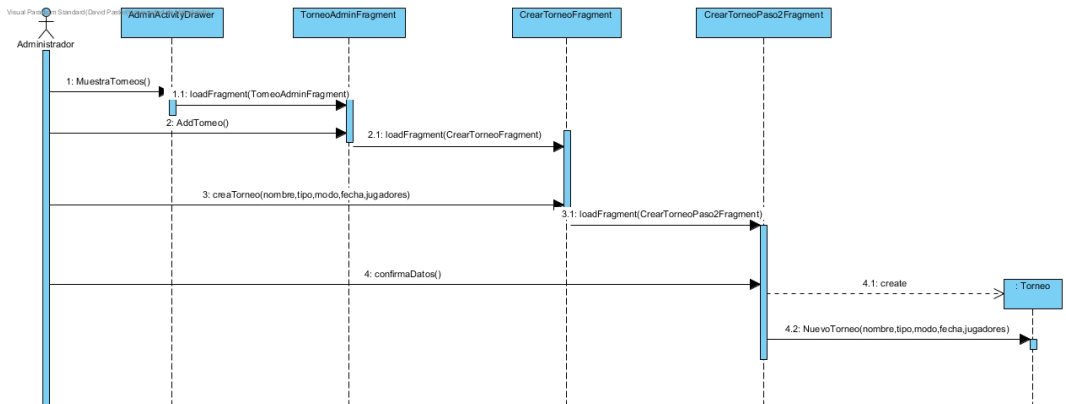


Figura 4.3: SD-001-Crear torneo

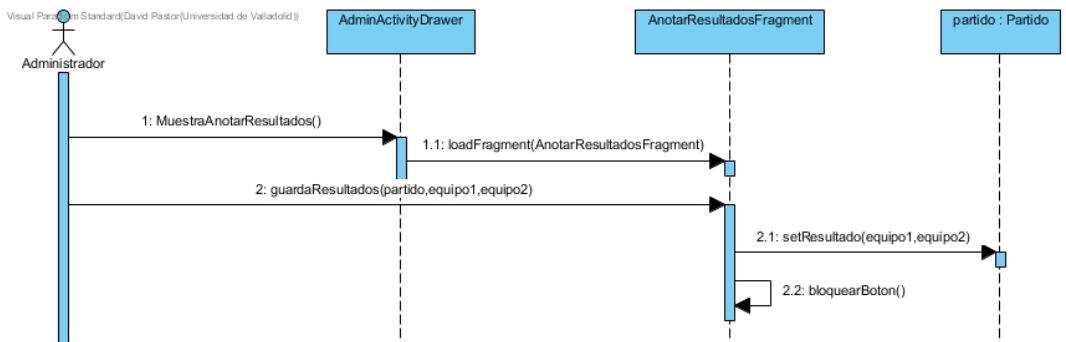


Figura 4.4: SD-002-Anotar resultados

4.6. DIAGRAMA DE SECUENCIA

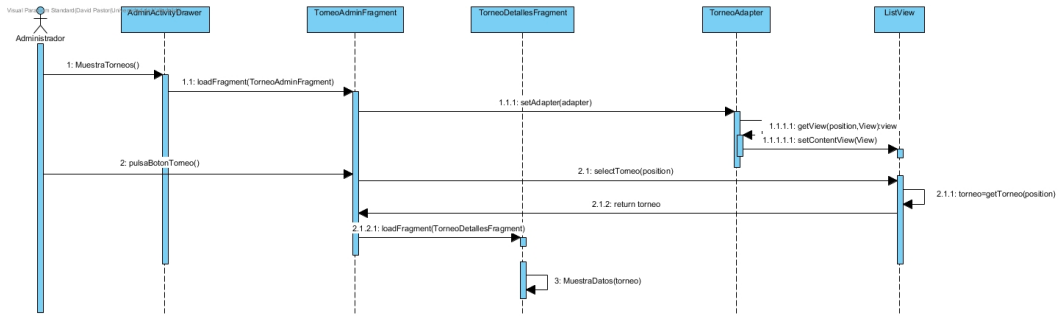


Figura 4.5: SD-003-Ver torneos

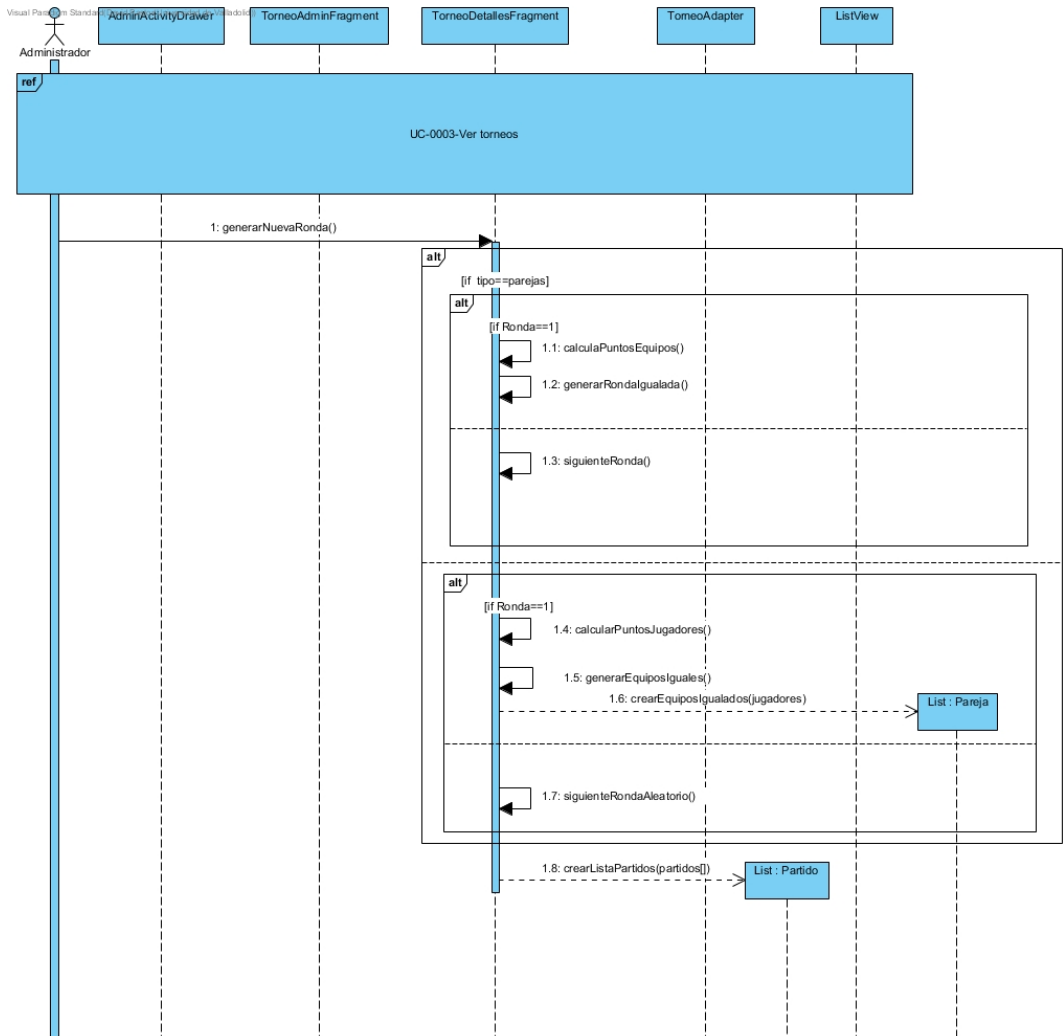


Figura 4.6: SD-004-Generar nueva ronda

4.6. DIAGRAMA DE SECUENCIA

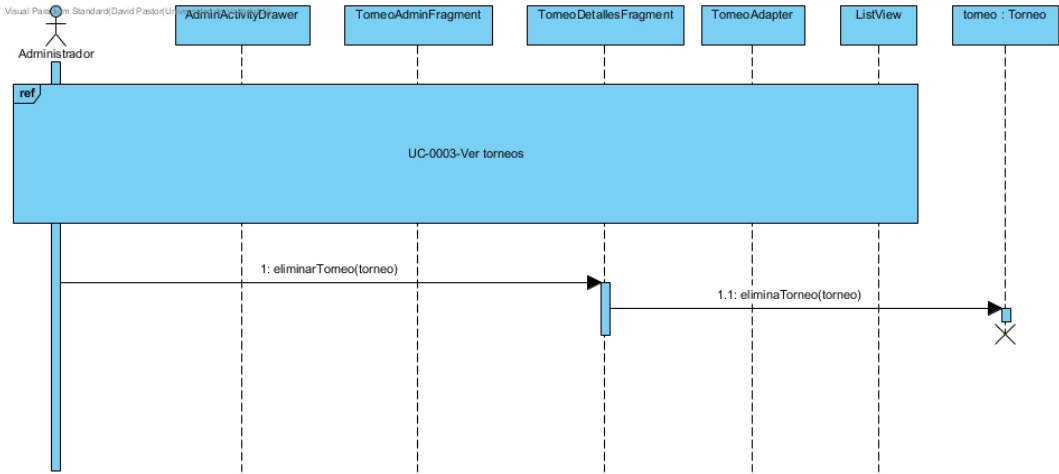


Figura 4.7: SD-005-Eliminar torneo

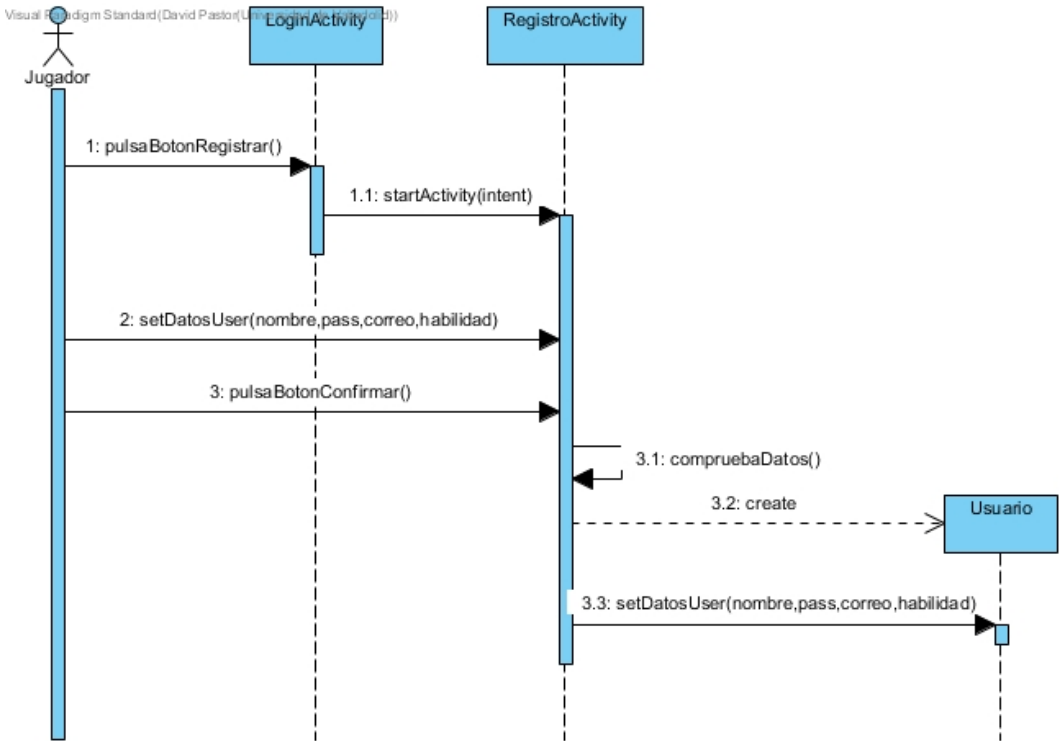


Figura 4.8: SD-006-Registrarse en el sistema

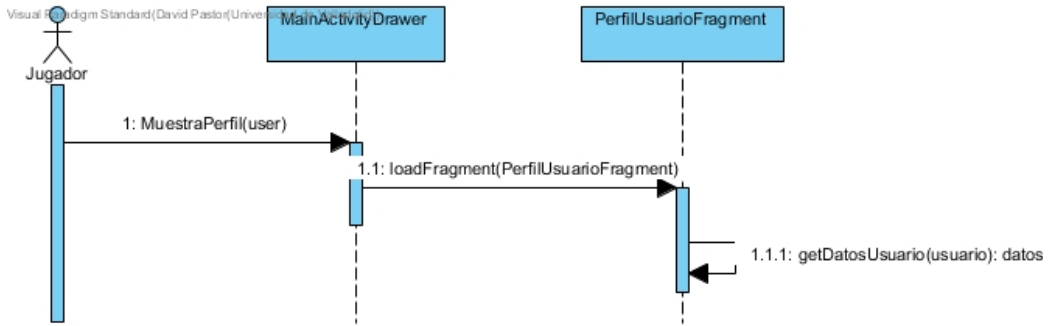


Figura 4.9: SD-007-Ver perfil de usuario

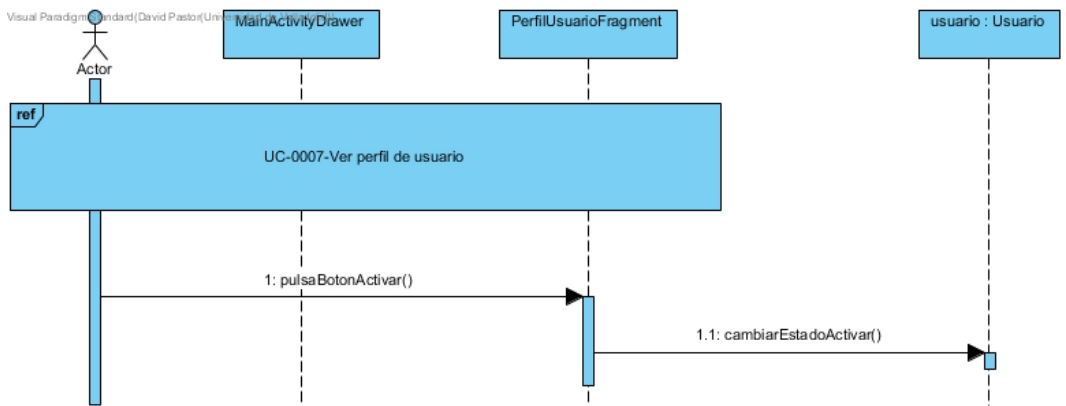


Figura 4.10: SD-008-Activar buscar pareja

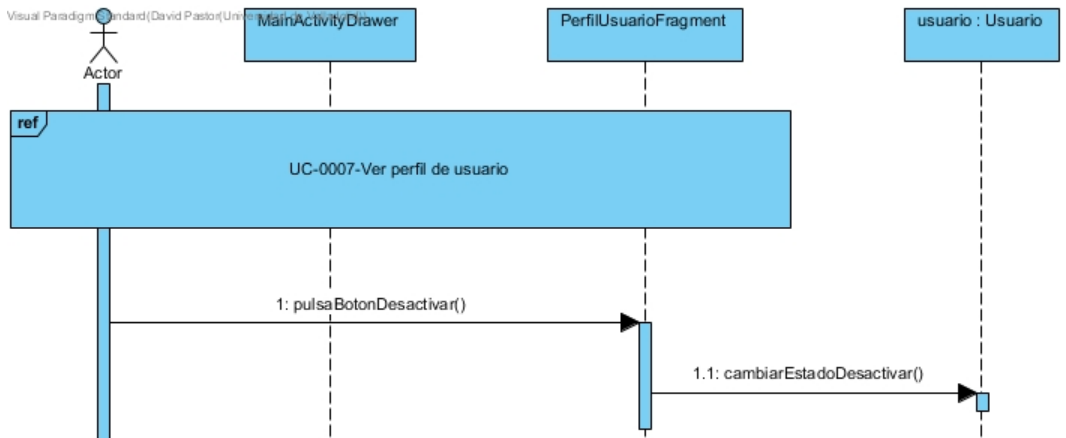


Figura 4.11: SD-009-Desactivar buscar pareja

4.6. DIAGRAMA DE SECUENCIA

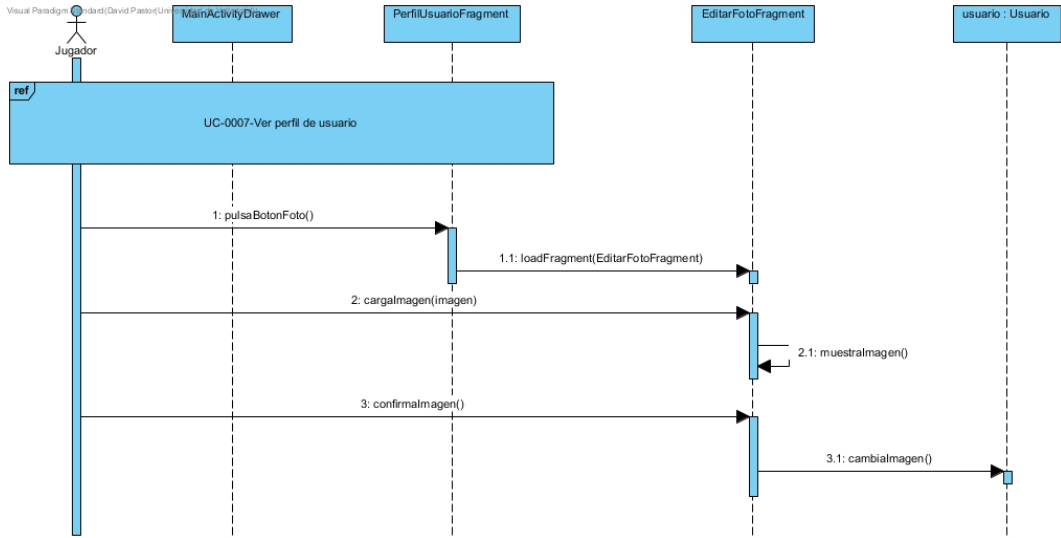


Figura 4.12: SD-010-Subir foto de perfil

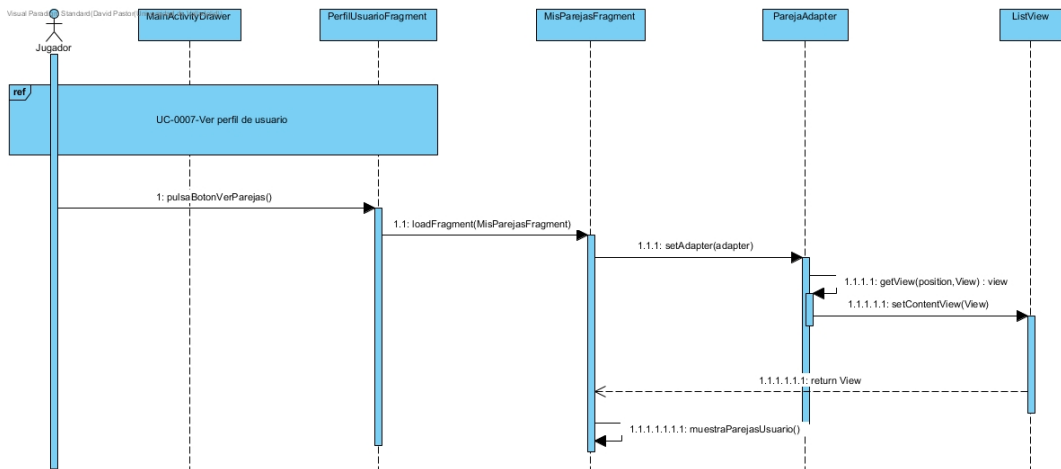


Figura 4.13: SD-011-Ver parejas

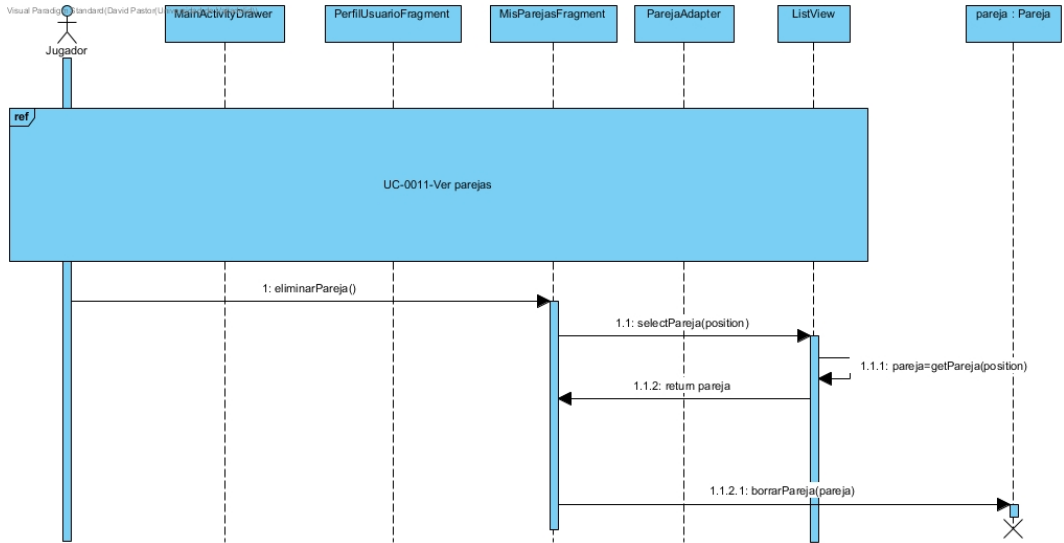


Figura 4.14: SD-012-Eliminar pareja

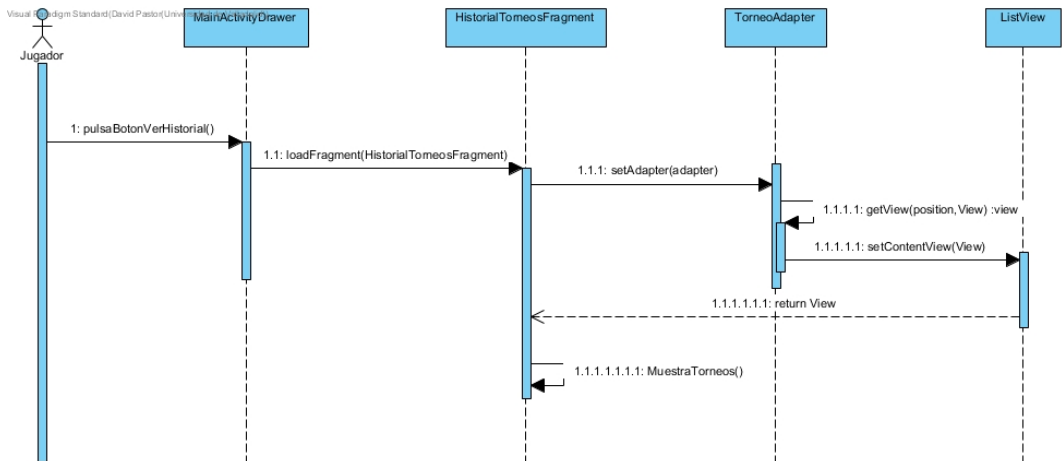


Figura 4.15: SD-013-Ver historial de torneos

4.6. DIAGRAMA DE SECUENCIA

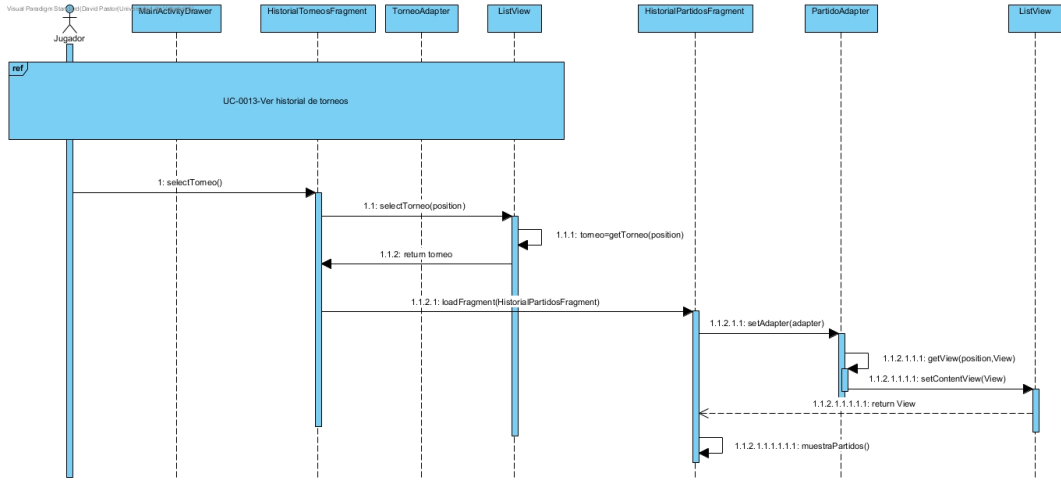


Figura 4.16: SD-014-Ver Partidos del torneo

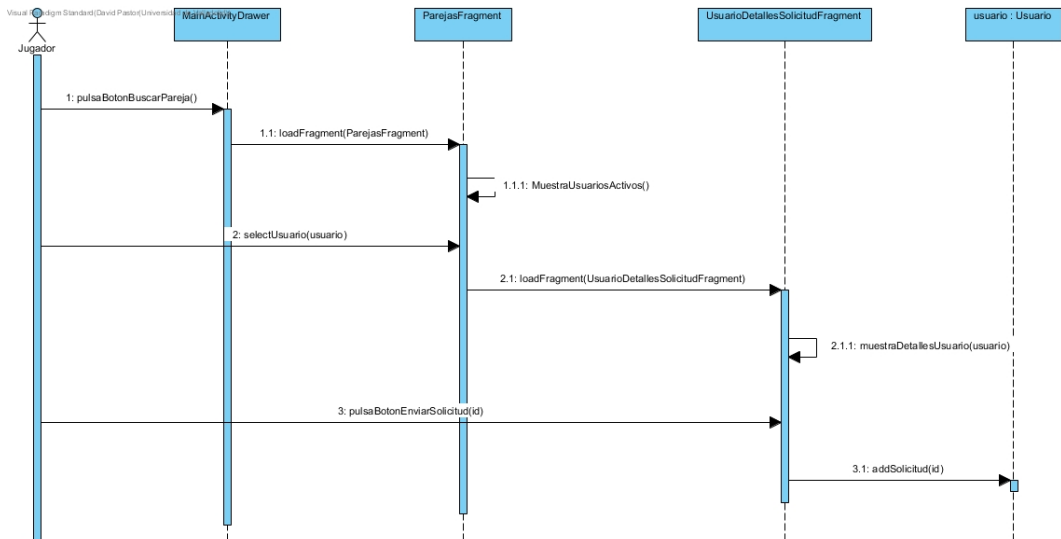


Figura 4.17: SD-015-Buscar pareja

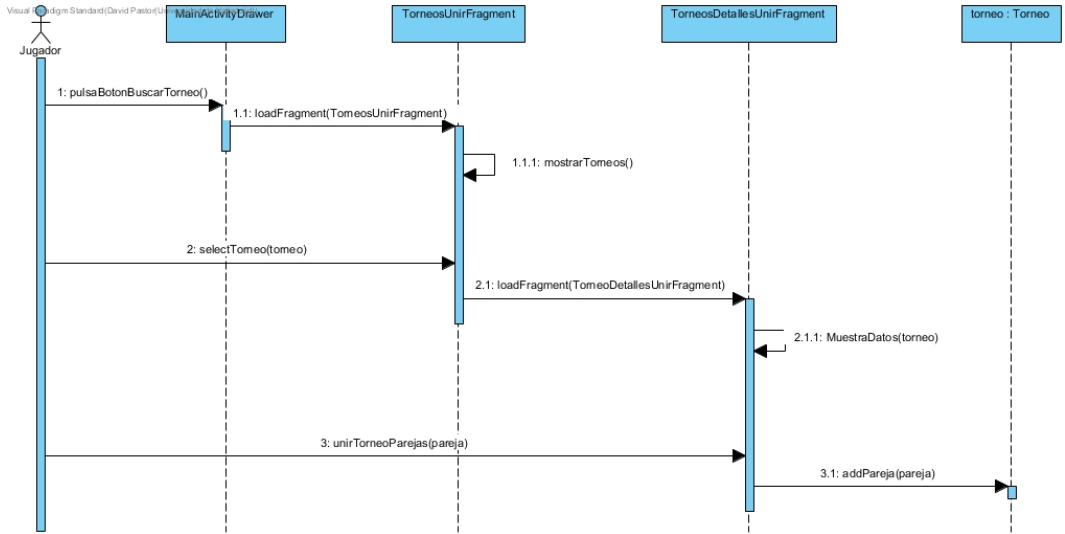


Figura 4.18: SD-016-Buscar torneos en pareja

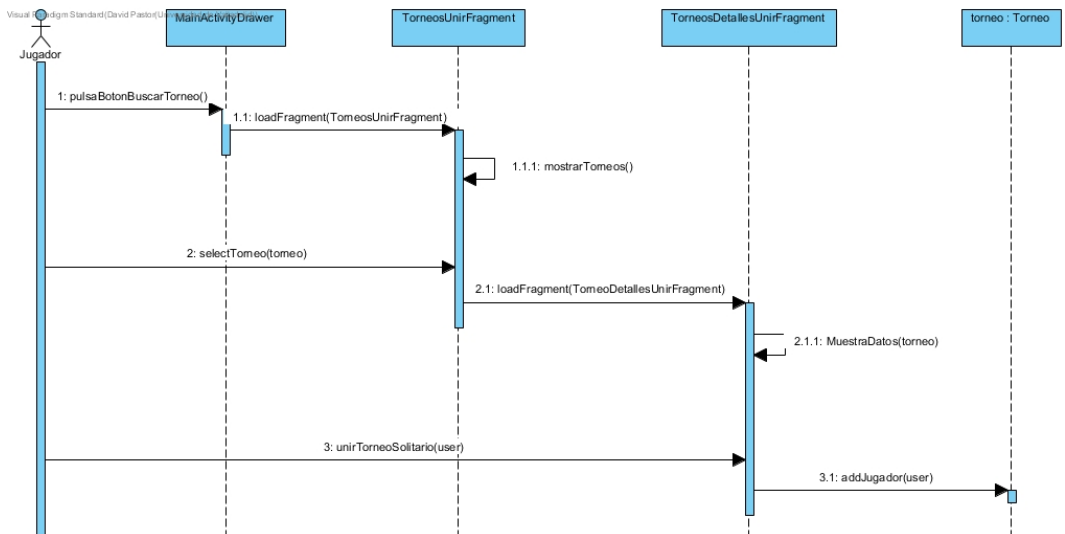


Figura 4.19: SD-017-Buscar torneo en solitario

4.6. DIAGRAMA DE SECUENCIA

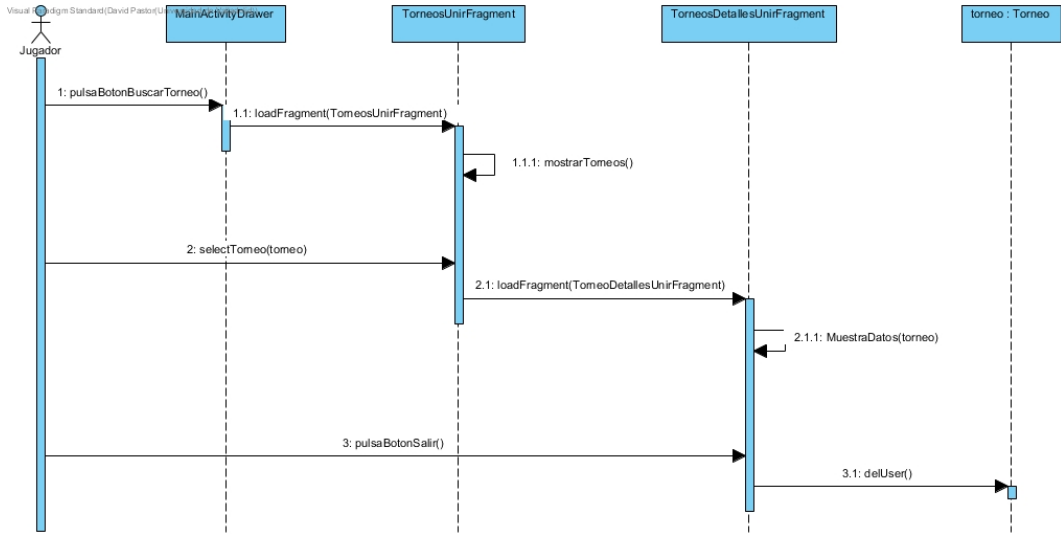


Figura 4.20: SD-018-Salir del torneo

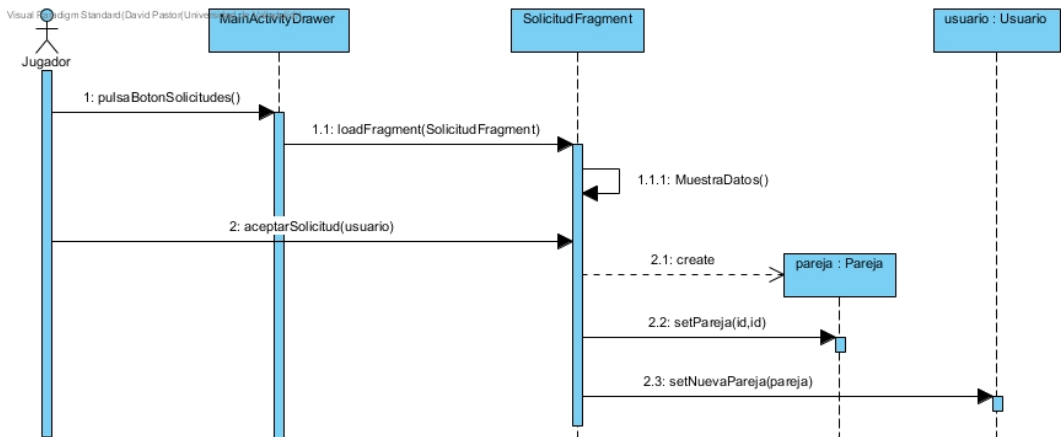


Figura 4.21: SD-019-Aceptar solicitud de pareja

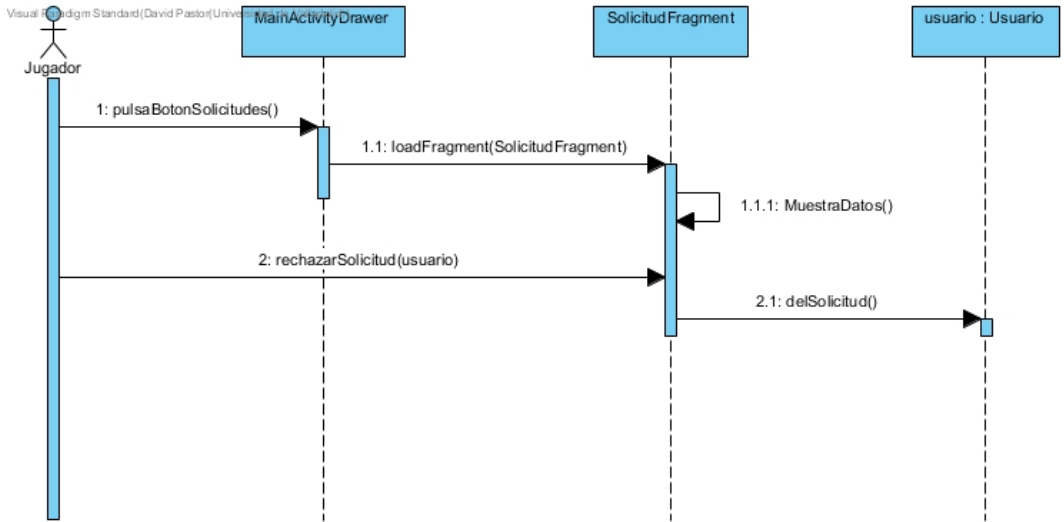


Figura 4.22: SD-020-Rechazar solicitud de pareja

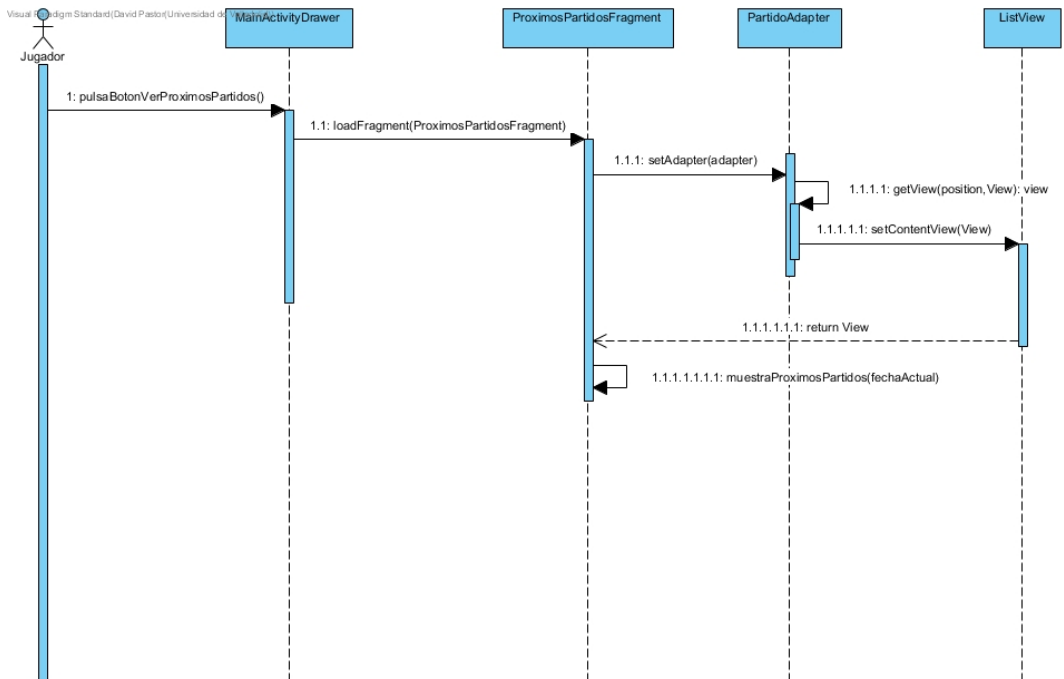


Figura 4.23: SD-021-Ver proximos partidos

4.6. DIAGRAMA DE SECUENCIA

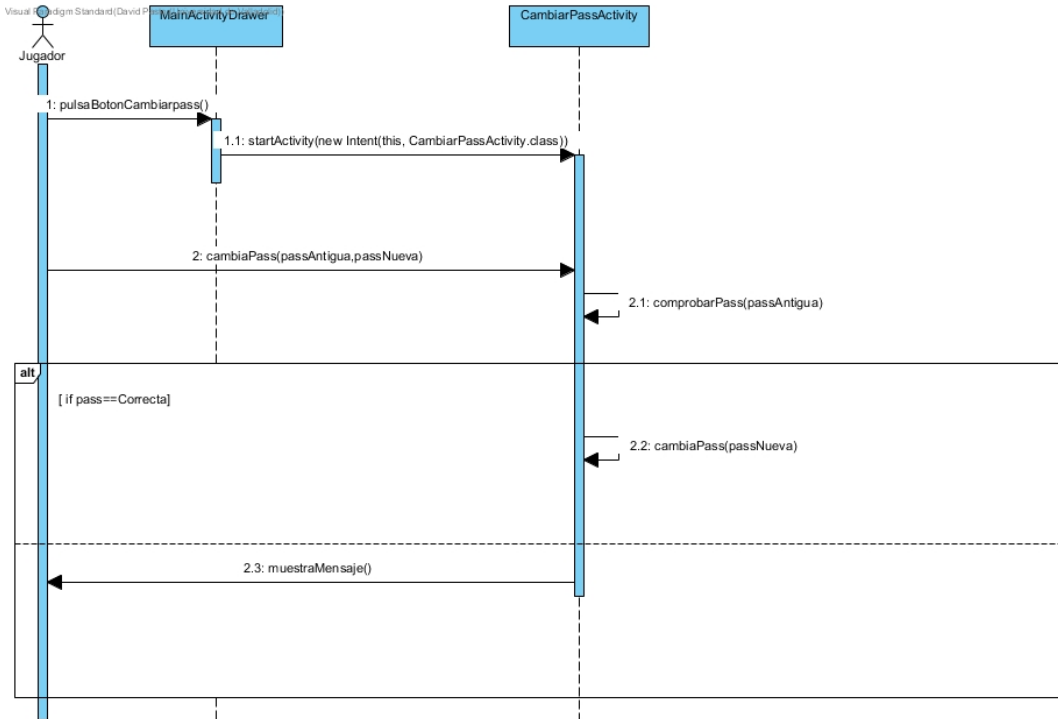


Figura 4.24: SD-022-Cambiar contraseña

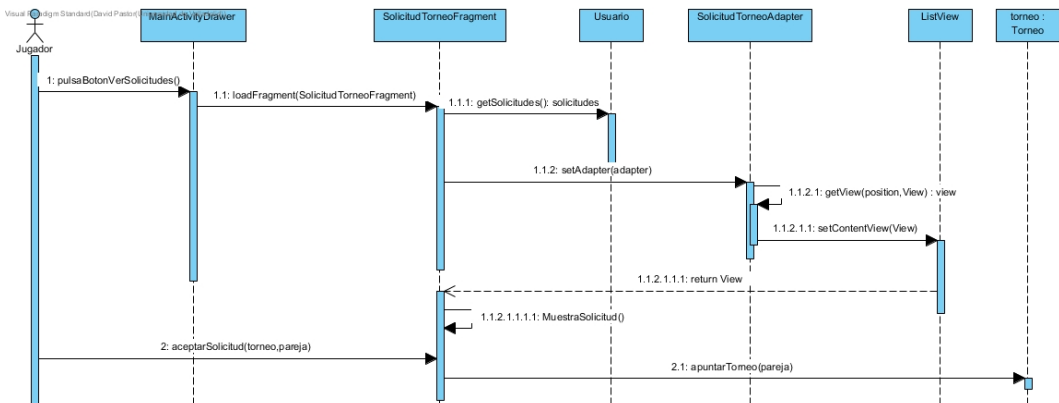


Figura 4.25: SD-023-Aceptar solicitud de torneo

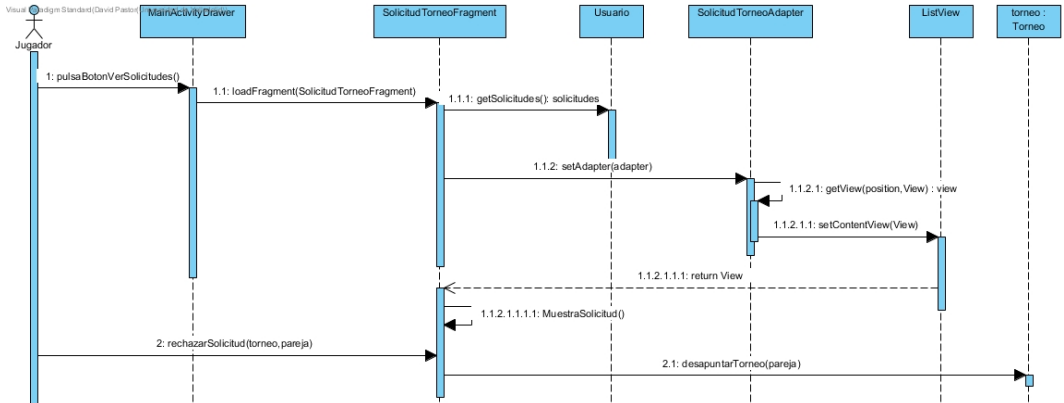


Figura 4.26: SD-024-Rechazar solicitud de torneo

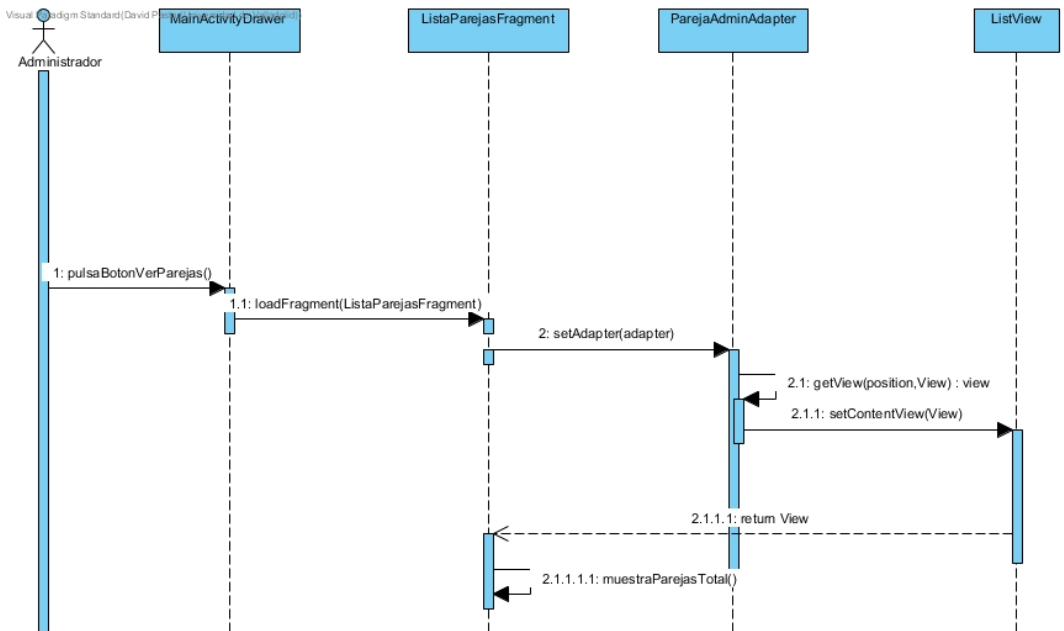


Figura 4.27: SD-025-Consultar equipos

4.7. Diagrama de despliegue

A continuación se muestra el diagrama de despliegue, el cual se utiliza para representar la distribución física de los componentes software en diferentes nodos físicos de la red. En este caso vemos cómo se conecta el dispositivo Android con el servidor Firebase tanto para acceder a la base de datos, como para obtener las imágenes almacenadas de todos los usuarios.

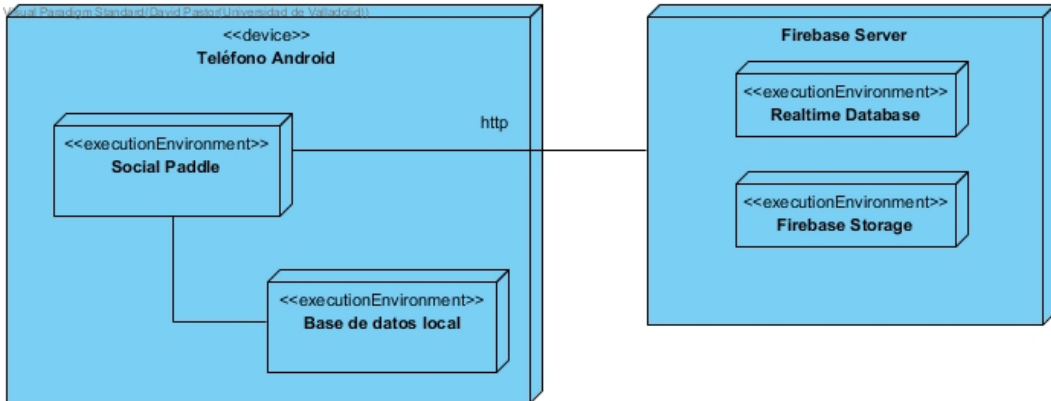


Figura 4.28: Diagrama de despliegue

4.8. Diseño base de datos

Para almacenar los datos se ha utilizado Firebase el cual utiliza bases de datos no relacionales, estos datos se estructuran en formato JSON. A diferencia de las bases de datos SQL, no hay tablas ni registros. Cuando se agregan datos al árbol JSON, se convertirá en un nodo de la estructura JSON existente con una clave asociada.

Para este proyecto se han utilizado 4 grandes nodos los cuales son: Usuario, Pareja, Partido, Torneo. Con estos grandes nodos es suficiente para almacenar toda la información necesaria para el correcto funcionamiento de la aplicación. Cuando se necesita relacionar dos nodos, se utiliza la clave única que dispone cada nodo. A continuación se muestra unos ejemplos de almacenamiento en formato JSON.

En el siguiente ejemplo podemos ver que cada usuario tiene un id único, y dentro de ese id tiene todos los atributos del usuario. Para poder saber qué parejas tiene, se almacena el id de la pareja dentro del propio nodo. Las solicitudes de pareja se almacena dentro del usuario al que van dirigidas, lo que se almacena en cada solicitud es el id del usuario que realizó la solicitud.

```

"Usuarios" : {
  "-M0xA8j8oi2q_bUJAgm" : {
    "apellidos" : "Pastor",
    "correo" : "pastor@gmail.com",
    "habilidad" : "Jugador Aficionado",
    "nombre" : "David",
    "parejas" : {
      "-M1MbQWrCrI7iHFc220m" : "-M1F6ILZYrRiYmvTVHUA",
      "-M4cNY8E4xF4Qu1HuLWH" : "-M1F6rIzMf6GKn9vswqn",
      "-M4cNhu6oJvQAO1IM-wz" : "-M1F6elsyvQHfq_aH1aS",
      "-M4cNo2bhIWK_nkbMXLP" : "-M1FTeJhGn6sp51T-5i6",
    }
  }
}
    
```

```

    "-M4cNuKtQWPSl25sGSfK" : "-M1oYSQTyFKwWq8Aj_A2"
  },
  "partidos_ganados" : 2,
  "partidos_jugados" : 3,
  "partidos_perdidos" : 1,
  "peticiones" : {
    "-M6iV4kdv6jvWZcwClfr" : "-M6iV4kdv6jvWZcwClfr"
  },
  "puntosHabilidad" : 3120,
  "solicitudes_torneo":{
    "-M4cNhu6oJvQAO1IM-wzUNION-M6oH59RMSoqTSxpgmRZ:" -M6oH59RMSoqTSxpgmR
  },
  "visible" : true
}
}

```

En el nodo parejas se almacena cada pareja con un id único, dentro del nodo tenemos 3 atributos los cuales son los id de los dos jugadores y un atributo que indica si la pareja es temporal o no.

```

" Parejas" : {
  "-M1KRhlglBQY00HUcHi2" : {
    "jugador1" : "-M0xkA8j8oi2q_bUJAgm",
    "jugador2" : "-M1F6elsyvQHfq_aH1aS",
    "temporal" : false
  },
}

```

En el nodo de partidos se almacenan los distintos datos sobre el partido como la fecha, nombre del torneo, resultado, numero de ronda, el id de los equipos que disputaran el partido y si el partido esta finalizado o no.

```

" Partidos" : {
  "-M2SgNSRiRafoAHARMTb" : {
    "equipo1" : "-M1q0SYa0N521cWJk0WG",
    "equipo2" : "-M1MbQWrCrI7iHFc220m",
    "fecha" : "8/5/2020",
    "finalizado" : true,
    "nombreTorneo" : "Torneo4",
    "resultado" : "0-3",
    "ronda" : 1
  }
}

```

En el nodo de torneos se almacenan todos los datos principales. En función del estado en que se encuentre el torneo podemos encontrar distintos atributos. Si el torneo todavía no ha comenzado tenemos el atributo apuntados en el cuál se almacenan los id de todas las parejas. Si los partidos ya se han generado tenemos un atributo de partidos en el que almacena el id

de todos los partidos que tiene este torneo. El atributo finalizado nos indica si este torneo ya genero todas sus rondas.

```
"Torneos" : {
  "-M1pztsY_KBy-DptI-UX" : {
    "apuntados" : {
      "-M1MbQWrCrI7iHFc220m" : "-M1MbQWrCrI7iHFc220m",
      "-M1q0CbGx0rniRPbgK_2" : "-M1q0CbGx0rniRPbgK_2",
      "-M1q0SYa0N521cWJk0WG" : "-M1q0SYa0N521cWJk0WG",
      "-M1q0dythc8jPZ68LKJc" : "-M1q0dythc8jPZ68LKJc"
    },
    "cerrado" : true,
    "equiposApuntados" : 4,
    "fecha" : "8/5/2020",
    "finalizado" : false,
    "modo_Equipos" : "Parejas",
    "nivel_Habilidad" : "Jugador Aficionado",
    "nombre" : "Torneo4",
    "numeroEquipos" : 4,
    "numeroRondas" : 2,
    "solicitudes":{
      "-M4cNhu6oJvQAO1IM-wz:" -M4cNhu6oJvQAO1IM-wz"
    },
    "partidos" : [ "-M2SgNSRiRafoAHARMTb", "-M2SgNSUX1bk3IPWEdZ" ],
    "tipo_Torneo" : "Pago"
  }
}
```


Capítulo 5

Implementación

En este capítulo se abordan las distintas partes de la implementación, la interfaz gráfica, algoritmos principales y problemas durante el desarrollo de la aplicación.

5.1. Interfaz gráfica

En este apartado se expone el desarrollo de la interfaz gráfica de la aplicación, desde el punto inicial en papel hasta su puesta a punto en producción.

5.1.1. Prototipo inicial

Antes de comenzar con el desarrollo de la aplicación, se plasmó en papel un prototipo (5.1) mostrando las distintas pantallas y las acciones donde te llevaba cada uno de ellas. Con esto se buscaba tener un concepto inicial sobre el aspecto que tenía que tener la aplicación cuando terminase.

5.1.2. Aspecto final

A continuación se muestra cómo quedó el aspecto final de la aplicación. Como podemos observar el diseño de la aplicación se ha centrado en el color azul. Para dar un mejor aspecto visual se han redondeado todos los botones y campos de texto. A fin de dar más facilidad de uso al usuario se intenta indicar con texto qué función realiza cada botón. Algunos botones utilizan los iconos propios que proporciona Android Studio. Se busca que los iconos indiquen de la mejor manera la función del icono, como por ejemplo una "papelera" para borrar algo. Alguno de los iconos se han obtenido de internet [13]. Se ha intentado dar un aspecto simple a la aplicación para intentar no saturar al usuario con demasiados botones o iconos.

5.1. INTERFAZ GRÁFICA

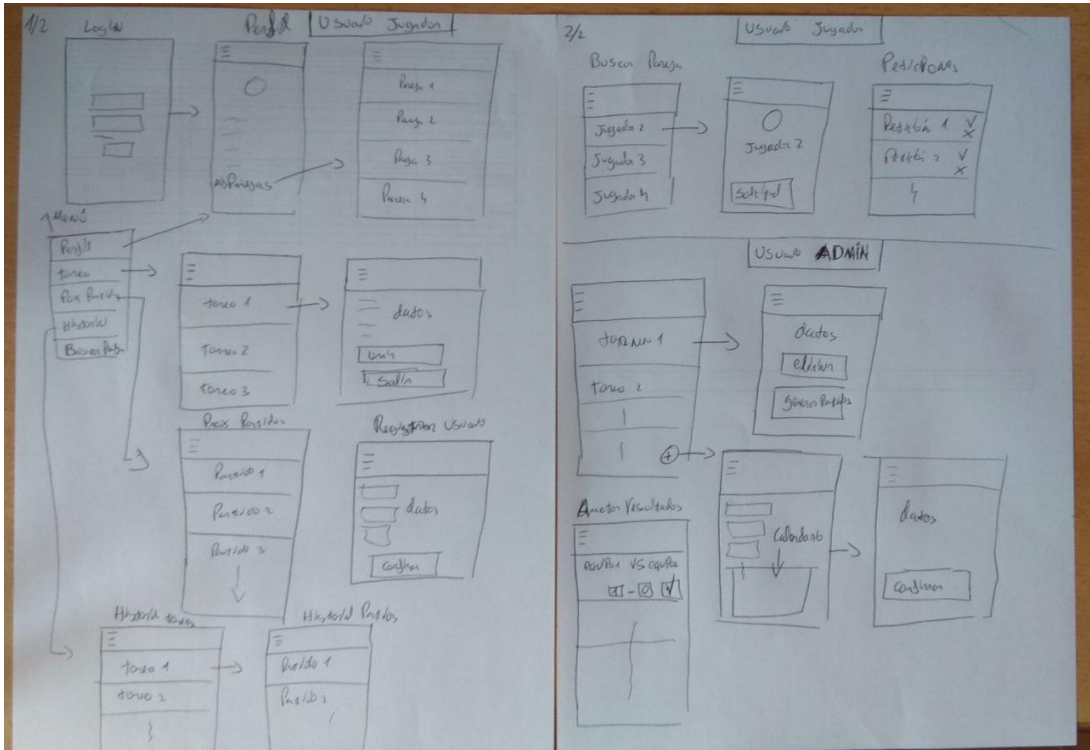
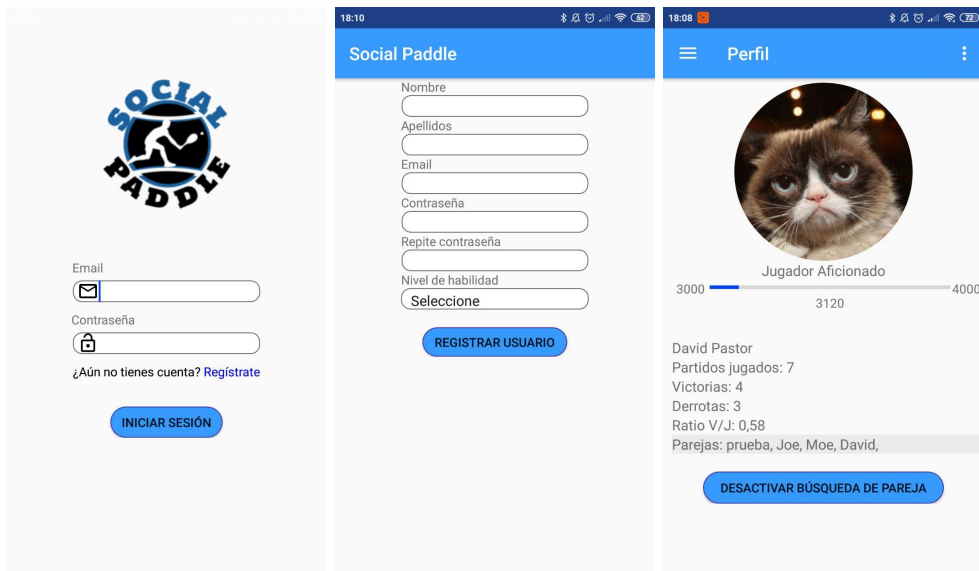


Figura 5.1: Prototipo inicial



(a) Login

(b) Registro

(c) Perfil usuario

Figura 5.2: Aspecto final usuario I

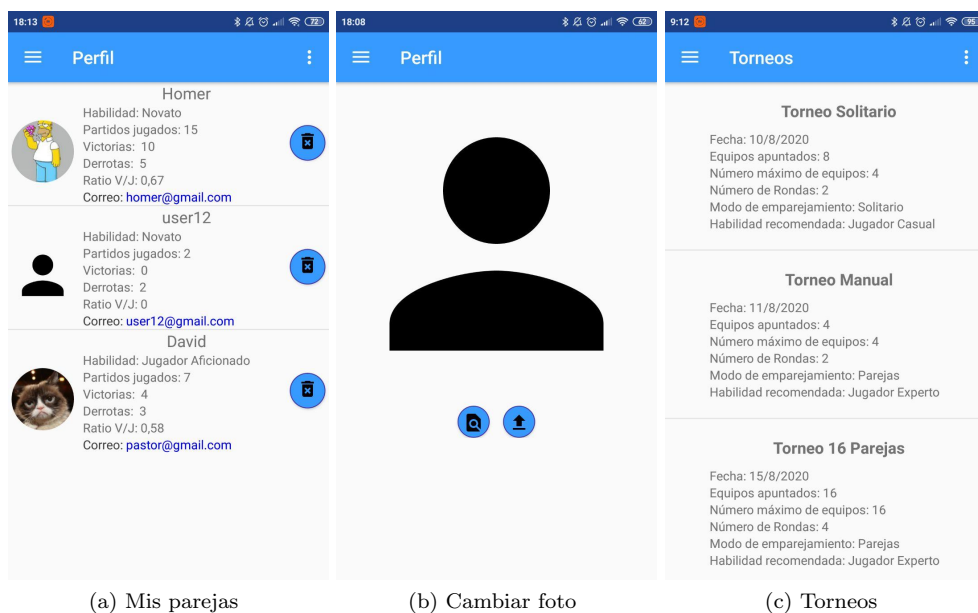


Figura 5.3: Aspecto final usuario II

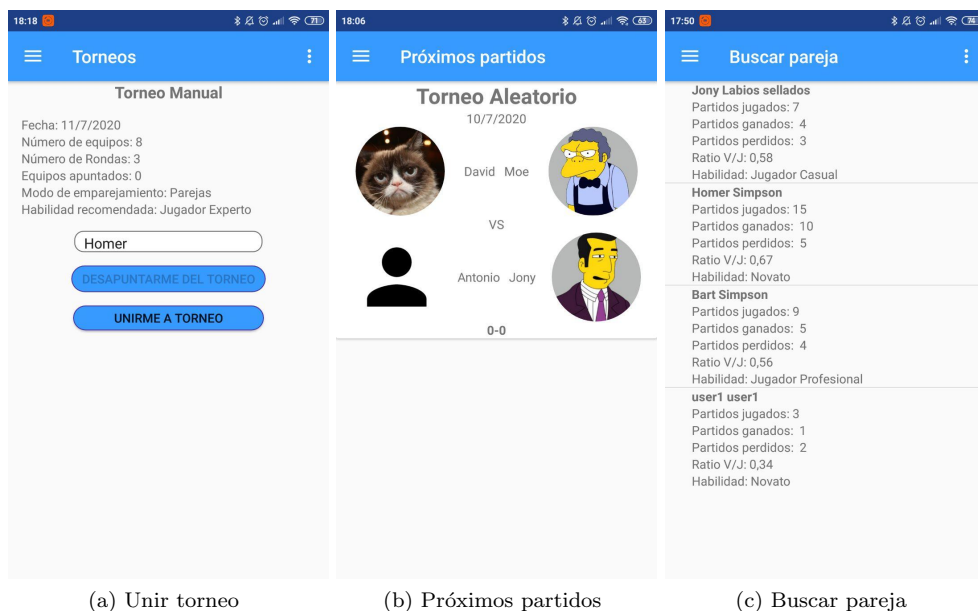


Figura 5.4: Aspecto final usuario III

5.1. INTERFAZ GRÁFICA

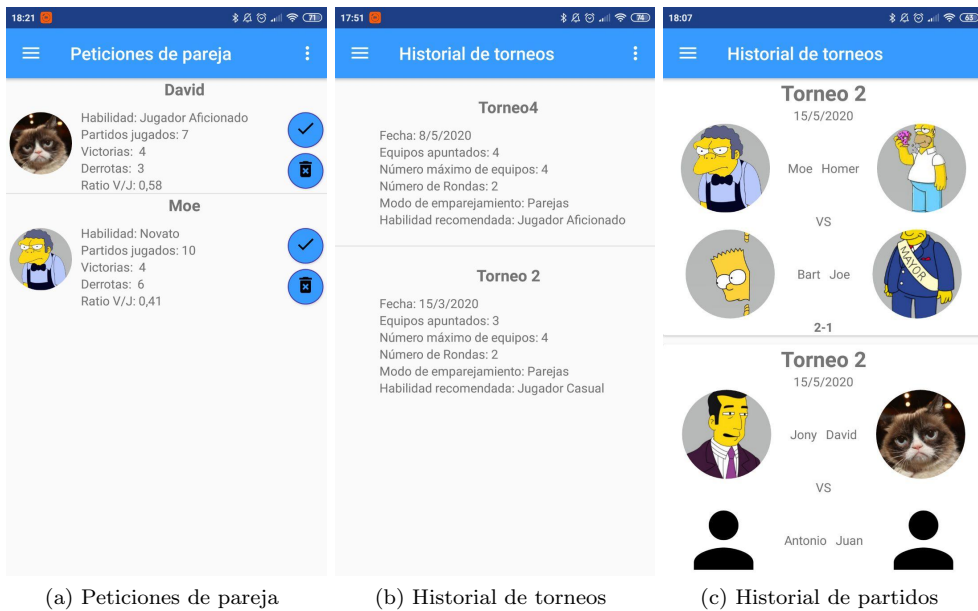


Figura 5.5: Aspecto final usuario IV

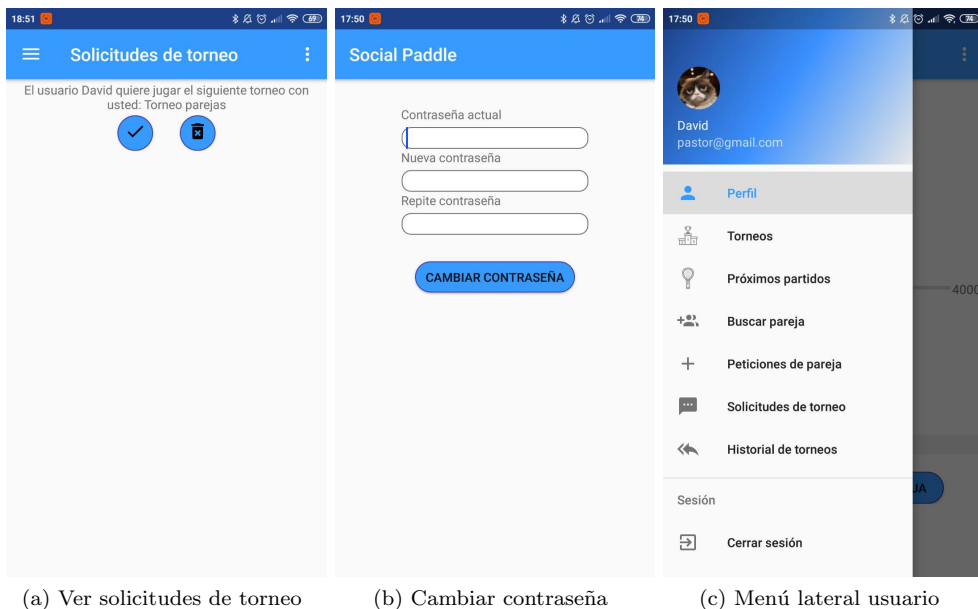


Figura 5.6: Aspecto final usuario V

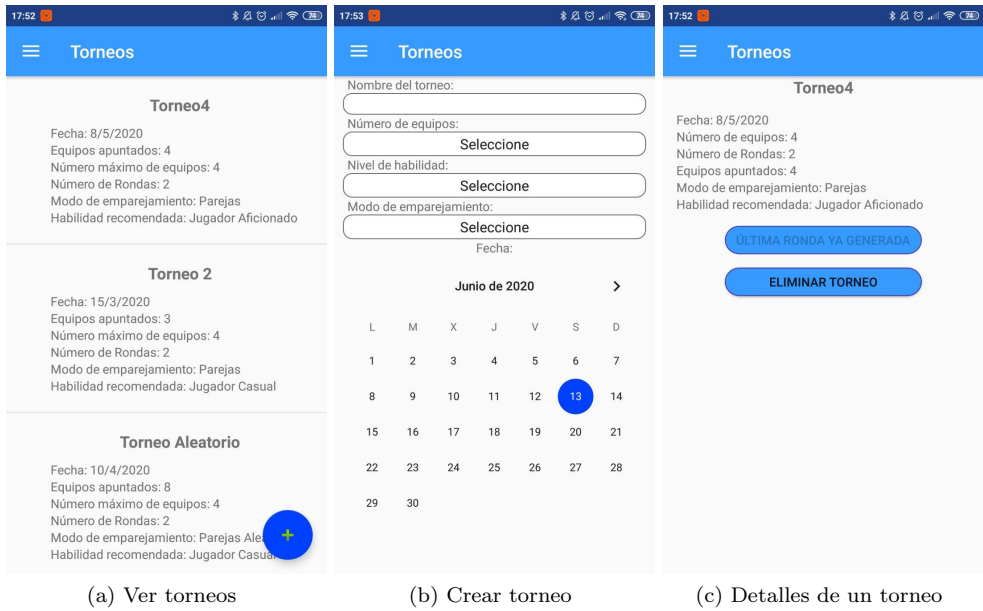


Figura 5.7: Aspecto final administrador I

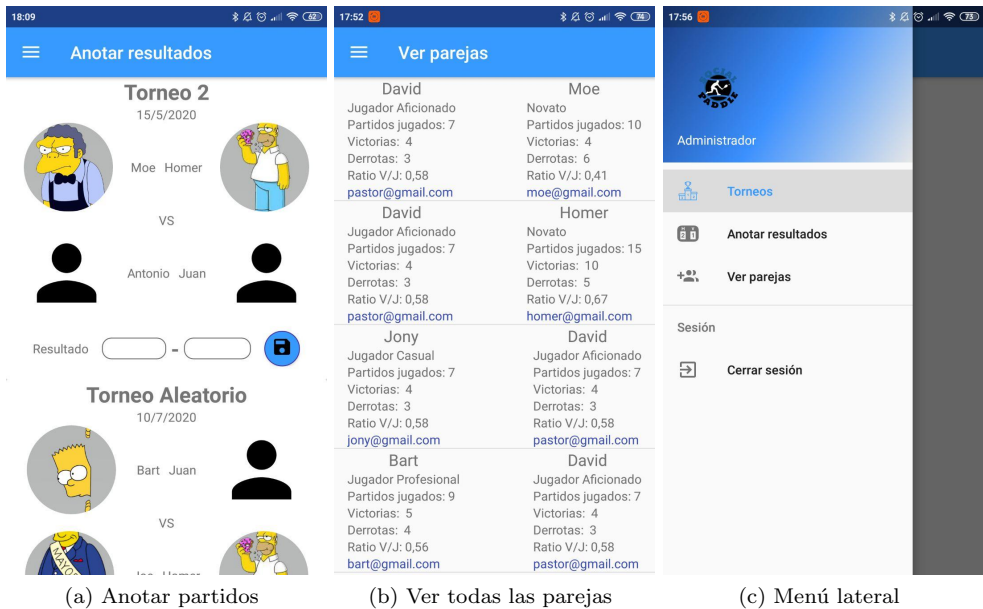


Figura 5.8: Aspecto final administrador II

5.2. Niveles de habilidad

Para los niveles de habilidad se crearon 7 categorías. El nivel de habilidad se asigna cuando un jugador se crea la cuenta. El jugador tiene que seleccionar un nivel de habilidad. Para este paso se tiene confianza en que el usuario sea crítico con su nivel de habilidad. En el momento que el jugador ya tiene su cuenta con el nivel de habilidad seleccionado, a este nivel se le asignan unos puntos de experiencia (5.1).

Nombre del nivel	Número de puntos
Novato	0-999
Novato entrenado	1000-1999
Jugador casual	2000-2999
Jugador aficionado	3000-3999
Jugador experto	4000-4999
Jugador profesional	5000-5999
Jugador invicto	6000-7000

Cuadro 5.1: Correspondencia Nivel-Puntos

Todos estos puntos irán cambiando durante el uso de la aplicación, en función de si se ganan los partidos o se pierden. Los puntos aumentarán o disminuirán de manera proporcional al nivel del contrincante, si tu contrincante es muy superior y pierdes, se te quitaran menos puntos que si el contrincante fuera de un nivel equitativo. Con esto se busca nivelar los enfrentamientos, ya que cuanto más partidos juegues tu nivel se ira ajustando de una mejor manera a tu nivel real. Para calcular la diferencia de nivel entre las parejas que se enfrentan, se suman los puntos de los jugadores de cada equipo, y se compara la suma de los puntos, para saber cuántos niveles existen de diferencia entre los dos equipos. A continuación se muestra una tabla (5.2) con el número de puntos que se gana o se pierde, en función de los niveles de diferencia con el contrincante. El número de niveles de diferencia se calcula en función de los puntos que distan a las dos parejas, no de los niveles de habilidad. Un nivel de habilidad equivale a 1000 puntos.

Niveles de diferencia	Puntos por victoria	Puntos por derrota
6	10	-190
5	20	-180
4	30	-160
3	50	-150
2	60	-140
1	80	-120
0	100	-100
-1	120	-80
-2	140	-60
-3	150	-50
-4	160	-30
-5	180	-20
-6	190	-10

Cuadro 5.2: Puntos por Victoria/Derrota

5.3. Partidos igualados

Para generar torneos lo mas equitativos posibles, cada torneo tiene una recomendación de nivel de habilidad, pero puede entrar cualquier jugador en el torneo. El sistema para intentar igualar los partidos varía en función de si el torneo es con parejas o es un torneo en solitario.

5.3.1. Torneo en parejas

Para poder generar los partidos necesitamos que un torneo esté completo. Cuando el torneo esta completo se calculara la suma total de puntos de cada pareja, estos cálculos se realizan sobre un tipo de dato ‘Map’. El primer paso es ordenar este tipo de dato, pero no tiene ninguna función definida que permita ordenarlo, para ordenar esto utilizamos la interfaz Java ‘Comparator’[16] la cual nos permite con la ayuda del tipo de dato ‘TreeMap’ ordenar nuestro ‘Map’. Ahora para generar los partidos recorreremos nuestro ‘Map’ generando los partidos juntando los primeros de la lista con los últimos, con esto conseguimos que los mejores equipos se enfrenten en la final, y a la vez penalizamos a los jugadores que entraron en el torneo con un nivel de habilidad muy distinto al real.

5.3.2. Torneo en solitario

Para generar los partidos de un torneo en solitario, antes se realiza un paso previo el cual es generar parejas lo mas igualadas posible, para que luego los partidos que se generen sean lo mas equitativos. Lo primero que se hace es ordenar los niveles de habilidad de menor a mayor, de una manera muy similar a cuando se ordenan los puntos de habilidad en los torneos por

pareja. Las parejas se irán generando obteniendo el primer jugador de la lista con el ultimo, y así hasta generar todas las parejas. Ahora ya tenemos todo lo que necesitamos para generar los partidos. En la base de datos estas parejas se almacenaran con un atributo que indica que son temporales para que no se mezclen con las parejas que ya tiene el propio usuario.

5.4. Problemas en la implementación

En este apartado se van a exponer los distintos problemas encontrados durante la implementación de la aplicación, podemos dividir los problemas en dos grandes bloques: problemas con la tecnología Android y problemas con el sistema de almacenamiento en Firebase.

Donde se han concentrado la mayoría de problemas ha sido en la tecnología Android, aunque uno de los lenguajes de programación en Android sea Java, tenemos una gran cantidad de funcionalidades que nos ofrece muchas posibilidades, estas funcionalidades pueden ser un poco complejas de entender al principio y te den bastantes problemas a la hora de implementar una aplicación. A continuación listare los principales problemas que encontré con la tecnología Android:

- **Fragments:** son componentes que funcionan dentro del alcance de la "Activity". Su propósito es extender una parte de la lógica utilizada para la navegación entre pantallas o "Activities" para que se puedan definir múltiples "Fragments" en la misma actividad. Al principio puede ser un poco complejo entender su ciclo de vida.
- **Adapters:** el adaptador es un objeto de una clase que implementa la interfaz "Adapter". Esto actúa como un enlace entre el conjunto de datos y el adaptador de vista. El crear un adaptador personalizado para mostrar los datos que obtenía de Firebase, me dio algunos problemas al principio de la implementación.
- **ListView:** es un objeto que nos permite mostrar una lista de elementos. Los ListView se usan junto a los adaptadores para mostrar los datos en forma de lista. Construir un ListView personalizado para mostrar los datos obtenidos en Firebase fue complejo al principio, pero en el momento que creas uno los demás se parecen bastante.
- Una de las cosas más complejas es la de adaptar una aplicación a la multitud de pantallas que existen en el mercado.

Todos estos problemas principalmente se originaron al principio del desarrollo, la principal causa de ellos fue la falta de experiencia. En función del avance del proyecto los problemas con el entorno de Android Studio fueron disminuyendo.

El otro gran bloque de problemas en la implementación fue en el uso de Firebase para el almacenamiento de datos. El principal problema fue el uso de bases de datos "NoSQL", Este sistema de bases de datos nunca lo había utilizado para realizar una aplicación, entonces tuve que cambiar mi manera de pensar al diseñar la base de datos y para obtener los datos.

Para realizar las consultas a Firebase se generan receptores asíncronos para recuperar la información, estos receptores están escuchando principalmente a cinco eventos que genera

el árbol de datos JSON los eventos son los siguientes: Added, Changed, Removed, Moved, Cancelled. Estos eventos están escuchando en segundo plano a no ser, que tu indiques que los eventos dejen de escuchar. Esto me trajo problemas al principio ya que no paraba los eventos y si otro usuario modificaba el nodo JSON mi pantalla se actualizaba generando datos duplicados o errores. Esto pasa principalmente cuando se están utilizando ‘Adapters’ junto a ‘ListView’.

Otro problema con Firebase fue anidar consultas asíncronas de distintas ramas del árbol JSON. Para solucionar este problema se hizo lo siguiente: para la consulta principal se utiliza un ‘Event listener’ de Firebase llamado ‘addChildEventListener’ el cual va recuperando hijos del nodo indicado hasta finalizar pero si a esa consulta le añades otro ‘Event listener’ llamado ‘addValueEventListener’ el cual se ejecuta justo después del primero con esto conseguimos que en la finalización de la primera consulte inmediatamente arranque la segunda. Esto también se puede utilizar para indicar gráficamente cuando acaba de ejecutarse la consulta.

Capítulo 6

Pruebas

En este capítulo se expone la batería de pruebas realizada para comprobar que todas las funcionalidades están implementadas de manera correcta, con esto se busca saber la calidad del software y corregir algún posible fallo antes de la puesta en producción de la herramienta.

Durante la codificación de la aplicación se fueron realizando pruebas unitarias de cada parte realizada, pero con esta batería de pruebas se busca una prueba completa de todas las funcionalidades listas ya que con las pruebas durante la codificación se pueden escapar muchos fallos.

CPSYS-001	Registrar usuario
Descripción	La prueba consiste en que un usuario se registre en la aplicación
Resultado esperado	El usuario queda registrado en la aplicación.
Resultado de la prueba	Correcto

Cuadro 6.1: Batería de pruebas para el caso CPSYS-001 Registrar usuario

CPSYS-002	Registrar usuario (2)
Descripción	La prueba consiste en que un usuario deje un campo sin completar al registrarse en el sistema.
Resultado esperado	Se muestra un mensaje de error en la pantalla.
Resultado de la prueba	Correcto

Cuadro 6.2: Batería de pruebas para el caso CPSYS-002 Registrar usuario (2)

CPSYS-003	Iniciar sesión
Descripción	La prueba consiste en que un usuario inicie sesión en la aplicación
Resultado esperado	La aplicación inicia sesión y muestra la pantalla principal del usuario.
Resultado de la prueba	Correcto

Cuadro 6.3: Batería de pruebas para el caso CPSYS-003 Iniciar sesión

CPSYS-004	Iniciar sesión (2)
Descripción	La prueba consiste en que un usuario inicie sesión en la aplicación con unas credenciales erróneas
Resultado esperado	La aplicación muestra un mensaje de error.
Resultado de la prueba	Correcto

Cuadro 6.4: Batería de pruebas para el caso CPSYS-004 Iniciar sesión (2)

CPSYS-005	Activar búsqueda de pareja
Descripción	La prueba consiste en que un usuario pulsa sobre el botón activar buscar pareja en la pantalla de su perfil de usuario.
Resultado esperado	La aplicación actualiza el estado en la base de datos y cambia el mensaje del botón indicando desactivar búsqueda
Resultado de la prueba	Correcto

Cuadro 6.5: Batería de pruebas para el caso CPSYS-005 Activar búsqueda de pareja

CPSYS-006	Desactivar búsqueda de pareja
Descripción	La prueba consiste en que un usuario pulsa sobre el botón desactivar buscar pareja en la pantalla de su perfil de usuario.
Resultado esperado	La aplicación actualiza el estado en la base de datos y cambia el mensaje del botón indicando activar búsqueda
Resultado de la prueba	Correcto

Cuadro 6.6: Batería de pruebas para el caso CPSYS-006 Desactivar búsqueda de pareja

CPSYS-007	Ver parejas
Descripción	La prueba consiste en que un usuario pulsa sobre los nombres de sus parejas en el perfil de usuario.
Resultado esperado	La aplicación cambia de pantalla y muestra todas las parejas que tiene el usuario.
Resultado de la prueba	Correcto

Cuadro 6.7: Batería de pruebas para el caso CPSYS-007 Ver parejas

CPSYS-008	Ver parejas (2)
Descripción	La prueba consiste en que un usuario pulsa sobre los nombres de sus parejas en el perfil de usuario y el usuario no dispone de ninguna pareja.
Resultado esperado	La aplicación cambia de pantalla y muestra un mensaje indicando que no tienes parejas.
Resultado de la prueba	Correcto

Cuadro 6.8: Batería de pruebas para el caso CPSYS-008 Ver parejas (2)

CPSYS-009	Eliminar pareja
Descripción	La prueba consiste en que un usuario pulsa sobre el botón con un icono en forma de papelera para eliminar una pareja.
Resultado esperado	La aplicación elimina la pareja de la base de datos y actualiza la pantalla sin mostrar la pareja eliminada recientemente.
Resultado de la prueba	Correcto

Cuadro 6.9: Batería de pruebas para el caso CPSYS-009 Eliminar pareja

CPSYS-010	Ver torneos
Descripción	La prueba consiste en que un usuario pulsa sobre el menú lateral sobre la pestaña torneos.
Resultado esperado	La aplicación muestra una pantalla con todos los torneos disponibles en la fecha actual.
Resultado de la prueba	Correcto

Cuadro 6.10: Batería de pruebas para el caso CPSYS-010 Ver torneos

CPSYS-011	Ver torneos (2)
Descripción	La prueba consiste en que un usuario pulsa sobre el menú lateral sobre la pestaña torneos y no existe ningún torneo en ese momento.
Resultado esperado	La aplicación muestra un mensaje en la pantalla indicando que no existen torneos.
Resultado de la prueba	Correcto

Cuadro 6.11: Batería de pruebas para el caso CPSYS-011 Ver torneos (2)

CPSYS-012	Unirme a torneo con pareja
Descripción	La prueba consiste en que un usuario pulsa sobre un torneo y este le lleva a una pantalla en la que puede unirse con una pareja, para esto tiene que seleccionar una pareja y después pulsar el botón unir.
Resultado esperado	La aplicación añade a esta pareja al torneo y muestra un mensaje confirmándote que estas unido al torneo.
Resultado de la prueba	Correcto

Cuadro 6.12: Batería de pruebas para el caso CPSYS-012 Unirme a torneo con pareja

CPSYS-013	Unirme a torneo con pareja (2)
Descripción	La prueba consiste en que un usuario pulsa sobre un torneo y este le lleva a una pantalla en la que puede unirse con una pareja, para esto tiene que seleccionar una pareja pero el usuario no dispone de pareja.
Resultado esperado	La aplicación bloquea el botón de unir al torneo y muestra un mensaje indicando que necesitas una pareja.
Resultado de la prueba	Correcto

Cuadro 6.13: Batería de pruebas para el caso CPSYS-013 Unirme a torneo con pareja (2)

CPSYS-014	Salir de un torneo con pareja
Descripción	La prueba consiste en que un usuario pulsa sobre un torneo y este le lleva a una pantalla en la que puede salirse del torneo, el usuario pulsa el botón para salir del torneo.
Resultado esperado	La aplicación elimina a esta pareja del torneo y muestra un mensaje confirmándote que fuera del torneo.
Resultado de la prueba	Correcto

Cuadro 6.14: Batería de pruebas para el caso CPSYS-014 Salir de un torneo con pareja

CPSYS-015	Cambiar foto de perfil
Descripción	La prueba consiste en que un usuario cambia la foto de perfil seleccionando una nueva de su galería de fotos
Resultado esperado	La aplicación muestra la nueva foto de perfil en la pantalla principal del usuario
Resultado de la prueba	Correcto

Cuadro 6.15: Batería de pruebas para el caso CPSYS-015 Cambiar foto de perfil

CPSYS-016	Unirse a un torneo en solitario
Descripción	La prueba consiste en que un usuario selecciona un torneo en solitario y pulsa sobre el botón unir.
Resultado esperado	La aplicación añade al usuario en el torneo y muestra en pantalla que el usuario ya está unido al torneo.
Resultado de la prueba	Correcto

Cuadro 6.16: Batería de pruebas para el caso CPSYS-016 Unirse a un torneo en solitario

CPSYS-017	Salir de un torneo en solitario
Descripción	La prueba consiste en que un usuario selecciona un torneo en solitario y pulsa sobre el botón salir.
Resultado esperado	La aplicación elimina al usuario del torneo y muestra en pantalla que el usuario ya está fuera del torneo.
Resultado de la prueba	Correcto

Cuadro 6.17: Batería de pruebas para el caso CPSYS-017 Salir de un torneo en solitario

CPSYS-018	Ver próximos partidos
Descripción	La prueba consiste en que un usuario selecciona en el menú lateral la opción de ver próximos partidos.
Resultado esperado	La aplicación cambia de pantalla y muestra todos los partidos que están próximos a la fecha actual.
Resultado de la prueba	Correcto

Cuadro 6.18: Batería de pruebas para el caso CPSYS-018 Ver próximos partidos

CPSYS-019	Ver próximos partidos (2)
Descripción	La prueba consiste en que un usuario selecciona en el menú lateral la opción de ver próximos partidos y no existe ningún partido.
Resultado esperado	La aplicación cambia de pantalla y muestra un mensaje indicando que no existen partidos.
Resultado de la prueba	Correcto

Cuadro 6.19: Batería de pruebas para el caso CPSYS-019 Ver próximos partidos (2)

CPSYS-020	Enviar solicitud de pareja
Descripción	La prueba consiste en que un usuario selecciona en el menú lateral la opción de buscar pareja se le muestra una lista de usuarios, pulsa sobre uno de ellos y dentro de los detalles del usuario pulsa sobre el botón enviar solicitud.
Resultado esperado	La aplicación almacena en la base de datos la petición y bloquea el botón de enviar solicitud para indicar al usuario que la solicitud ya esta mandada.
Resultado de la prueba	Correcto

Cuadro 6.20: Batería de pruebas para el caso CPSYS-020 Enviar solicitud de pareja

CPSYS-021	Aceptar solicitud de pareja
Descripción	La prueba consiste en que un usuario selecciona en el menú lateral la opción de peticiones de pareja y se le muestra una lista de peticiones, el usuario pulsa sobre el botón con el símbolo ‘‘Check‘‘.
Resultado esperado	La aplicación almacena en la base de datos la nueva pareja y actualiza la pantalla.
Resultado de la prueba	Correcto

Cuadro 6.21: Batería de pruebas para el caso CPSYS-021 Aceptar solicitud de pareja

CPSYS-022	Rechazar solicitud de pareja
Descripción	La prueba consiste en que un usuario selecciona en el menú lateral la opción de peticiones de pareja y se le muestra una lista de peticiones, el usuario pulsa sobre el botón con el símbolo ‘‘Papelera‘‘.
Resultado esperado	La aplicación elimina de la base de datos petición y actualiza la pantalla.
Resultado de la prueba	Correcto

Cuadro 6.22: Batería de pruebas para el caso CPSYS-022 Rechazar solicitud de pareja

CPSYS-023	Historial de torneos
Descripción	La prueba consiste en que un usuario selecciona en el menú lateral la opción de historial de torneos y se le muestra una lista de los torneos que ha jugado.
Resultado esperado	La aplicación muestra una pantalla con la lista de torneos.
Resultado de la prueba	Correcto

Cuadro 6.23: Batería de pruebas para el caso CPSYS-023 Historial de torneos

CPSYS-024	Historial de torneos (2)
Descripción	La prueba consiste en que un usuario selecciona en el menú lateral la opción de historial de torneos y no existe ningún torneo.
Resultado esperado	La aplicación muestra un mensaje en la pantalla indicando que no existen torneos.
Resultado de la prueba	Correcto

Cuadro 6.24: Batería de pruebas para el caso CPSYS-024 Historial de torneos (2)

CPSYS-025	Historial de partidos
Descripción	La prueba consiste en que un usuario selecciona en el menú lateral la opción de historial de torneos y pulse sobre uno de los torneos.
Resultado esperado	La aplicación muestra una pantalla con todos los partidos de ese torneo.
Resultado de la prueba	Correcto

Cuadro 6.25: Batería de pruebas para el caso CPSYS-025 Historial de partidos

CPSYS-026	Cerrar sesión
Descripción	La prueba consiste en que un usuario selecciona en el menú lateral la opción de cerrar sesión.
Resultado esperado	La aplicación elimina la sesión del usuario y muestra la pantalla de login.
Resultado de la prueba	Correcto

Cuadro 6.26: Batería de pruebas para el caso CPSYS-026 Cerrar sesión

CPSYS-027	Crear un torneo
Descripción	La prueba consiste en que un usuario administrador pulsa sobre el símbolo “+” para crear un torneo y introduce todos los datos de manera correcta.
Resultado esperado	La aplicación añade un nuevo torneo a la base de datos y muestra la pantalla principal al administrador.
Resultado de la prueba	Correcto

Cuadro 6.27: Batería de pruebas para el caso CPSYS-027 Crear un torneo

CPSYS-028	Crear un torneo (2)
Descripción	La prueba consiste en que un usuario administrador pulsa sobre el símbolo '+' para crear un torneo y introduce los datos de manera errónea.
Resultado esperado	La aplicación muestra un mensaje indicando al usuario que rellene todos los campos.
Resultado de la prueba	Correcto

Cuadro 6.28: Batería de pruebas para el caso CPSYS-028 Crear un torneo (2)

CPSYS-029	Eliminar un torneo
Descripción	La prueba consiste en que un usuario administrador pulsa sobre un torneo y cuando se muestran los datos del mismo pulsa sobre el botón eliminar.
Resultado esperado	La aplicación elimina el torneo de la base de datos y muestra la pantalla principal.
Resultado de la prueba	Correcto

Cuadro 6.29: Batería de pruebas para el caso CPSYS-029 Eliminar un torneo

CPSYS-030	Anotar resultado
Descripción	La prueba consiste en que un usuario administrador abre el menú lateral, pulsa sobre anotar resultados y anota un partido.
Resultado esperado	La aplicación actualiza la base de datos con el resultado y bloquea el botón de guardar.
Resultado de la prueba	Correcto

Cuadro 6.30: Batería de pruebas para el caso CPSYS-030 Anotar resultado

CPSYS-031	Anotar resultado (2)
Descripción	La prueba consiste en que un usuario administrador abre el menú lateral, pulsa sobre anotar resultados pero no existe ningún partido para anotar.
Resultado esperado	La aplicación muestra un mensaje indicando que no existen partidos para anotar.
Resultado de la prueba	Correcto

Cuadro 6.31: Batería de pruebas para el caso CPSYS-031 Anotar resultado (2)

CPSYS-032	Generar ronda torneo en parejas
Descripción	La prueba consiste en que un usuario administrador abre un torneo en parejas y pulsa sobre el botón generar ronda.
Resultado esperado	La aplicación genera los partidos en la base de datos y muestra en pantalla un mensaje indicando que están generados.
Resultado de la prueba	Correcto

Cuadro 6.32: Batería de pruebas para el caso CPSYS-032 Generar ronda torneo en parejas

CPSYS-033	Generar ronda torneo en solitario
Descripción	La prueba consiste en que un usuario administrador abre un torneo en solitario y pulsa sobre el botón generar ronda.
Resultado esperado	La aplicación genera las parejas temporales en la base de datos y los partidos, después de eso muestra un mensaje en pantalla y bloquea el botón.
Resultado de la prueba	Correcto

Cuadro 6.33: Batería de pruebas para el caso CPSYS-033 Generar ronda torneo en solitario

CPSYS-034	Concurrencia en unir a un torneo
Descripción	La prueba consiste en dos usuarios se unen simultáneamente a un torneo.
Resultado esperado	La aplicación almacena los dos usuarios en la base de datos.
Resultado de la prueba	Correcto

Cuadro 6.34: Batería de pruebas para el caso CPSYS-034 Concurrencia en unir a un torneo

CPSYS-035	Jugador ya unido a un torneo intenta unir con otro
Descripción	La prueba consiste en que un usuario el cual ya esta en un torneo con una pareja intenta unirse con otra pareja.
Resultado esperado	La aplicación bloquea el botón para unirse y muestra un mensaje indicando al usuario que no puede unirse.
Resultado de la prueba	Correcto

Cuadro 6.35: Batería de pruebas para el caso CPSYS-035 Jugador ya unido a un torneo intenta unir con otro

Capítulo 7

Conclusiones

Una vez finalizado el proyecto se han conseguido los objetivos marcados al principio del desarrollo en el tiempo establecido.

Se ha conseguido una aplicación Android que permite a un centro deportivo gestionar sus eventos deportivos, a la vez que sus clientes tienen la facilidad de poder buscar nuevas parejas con las que pueden unirse a los torneos existentes.

La realización de este proyecto me ha ayudado a adquirir nuevos conocimientos sobre el desarrollo en Android, estos conocimientos son bastante útiles a día de hoy, ya que Android es una tecnología con mucha fuerza actualmente. Otro de los conocimientos que he mejorado ha sido el desarrollo de un proyecto software completo, ya que he realizado los distintos roles que pueden existir en un proyecto (analista, programador, director de proyecto, diseñador) todo esto me han otorgado una experiencia que espero utilizar en el mundo laboral.

En el aspecto académico el desarrollo de este proyecto me ha ayudado a afianzar todas las habilidades que he adquirido durante mi formación académica. Puede que no haya podido utilizar todo lo aprendido, pero he utilizado pequeñas partes de cada una de las asignaturas.

Como última conclusión decir que el proyecto ha sido un poco complicado, la razón principal ha sido compaginarlo con un trabajo a jornada completa, el cual te absorbe mucho tiempo y energía, pero con constancia y trabajo diario el proyecto fue cogiendo forma hasta llegar a su finalización.

Como trabajo futuro en las siguientes versiones de la aplicación había pensado en añadir las siguientes funcionalidades.

- Sistema de mensajería entre usuarios.
- Sistema de notificaciones.
- Soporte de la aplicación en otros idiomas.

-
- Mejoras de aspecto visual.
 - Filtros para la búsqueda de torneos.
 - Filtros para la búsqueda de usuarios.

Bibliografía

- [1] *¿Qué es Java y para qué es necesario?* URL: https://www.java.com/es/download/faq/whatis_java.xml (visitado 25-04-2020).
- [2] *Android*. es. Page Version ID: 125478456. Abr. de 2020. URL: <https://es.wikipedia.org/w/index.php?title=Android&oldid=125478456> (visitado 25-04-2020).
- [3] *Build software better, together*. en. URL: <https://github.com> (visitado 25-04-2020).
- [4] *bumptech/glide*. original-date: 2013-07-08T22:52:33Z. Abr. de 2020. URL: <https://github.com/bumptech/glide> (visitado 25-04-2020).
- [5] Alberto Ceruelo Andrés. “Rehabilitación funcional de extremidades basada en Android”. spa. En: (2018). URL: <http://uvadoc.uva.es/handle/10324/33261> (visitado 13-05-2020).
- [6] *Cloud Storage*. es-419. URL: <https://firebase.google.com/docs/storage?hl=es-419> (visitado 13-05-2020).
- [7] *Documentación*. es-419. URL: <https://firebase.google.com/docs?hl=es> (visitado 25-04-2020).
- [8] *Download Android Studio and SDK tools — Android Studio*. en. URL: <https://developer.android.com/studio?hl=es> (visitado 05-05-2020).
- [9] editor. *¿Por qué el pádel se ha convertido en el deporte de moda?* es. Abr. de 2019. URL: <https://periodistas-es.com/por-que-el-padel-se-ha-convertido-en-el-deporte-de-moda-122289> (visitado 10-05-2020).
- [10] *Firebase*. es-419. URL: <https://firebase.google.com/?hl=es-419> (visitado 05-05-2020).
- [11] *Firebase Authentication*. es-419. URL: <https://firebase.google.com/docs/auth?hl=es-419> (visitado 05-05-2020).
- [12] *Firebase Realtime Database — Firebase Realtime Database*. es-419. URL: <https://firebase.google.com/docs/database?hl=es-419> (visitado 05-05-2020).
- [13] *Flaticon, the largest database of free vector icons*. en. URL: <https://www.flaticon.com/> (visitado 25-04-2020).
- [14] Héctor García Cabezón. “SensorLab: Aplicación móvil para la recopilación de mediciones sobre sensores en dispositivos Android”. spa. En: (2019). URL: <http://uvadoc.uva.es/handle/10324/38730> (visitado 13-05-2020).

- [15] *GitHub*. es. Page Version ID: 125073702. Abr. de 2020. URL: <https://es.wikipedia.org/w/index.php?title=GitHub&oldid=125073702> (visitado 25-04-2020).
- [16] *java - Sort a Map Key, Value by values*. URL: <https://stackoverflow.com/questions/109383/sort-a-mapkey-value-by-values> (visitado 25-04-2020).
- [17] *Java (lenguaje de programación)*. es. Page Version ID: 125248290. Abr. de 2020. URL: [https://es.wikipedia.org/w/index.php?title=Java_\(lenguaje_de_programaci%C3%B3n\)&oldid=125248290](https://es.wikipedia.org/w/index.php?title=Java_(lenguaje_de_programaci%C3%B3n)&oldid=125248290) (visitado 25-04-2020).
- [18] *Logo Maker - Create Your Own Logo, It's Free!* es. URL: <https://www.freelogodesign.org> (visitado 25-04-2020).
- [19] Marcos López Rodríguez. “Aplicación móvil para Android sobre un guía automático por el yacimiento de Pintia”. spa. En: (2019). URL: <http://uvadoc.uva.es/handle/10324/38705> (visitado 13-05-2020).
- [20] *Microsoft Project*. es. Page Version ID: 122186169. Dic. de 2019. URL: https://es.wikipedia.org/w/index.php?title=Microsoft_Project&oldid=122186169 (visitado 05-05-2020).
- [21] *Pádel, "deporte de moda"*. es. URL: <https://mundoentrenamiento.com/padel-deporte-de-moda/> (visitado 10-05-2020).
- [22] *Stack Overflow - Where Developers Learn, Share, & Build Careers*. URL: <https://stackoverflow.com/> (visitado 25-04-2020).
- [23] Alex Mendoza Ventura. *Android — Primeros pasos con Glide*. en. Ago. de 2016. URL: <https://medium.com/@alexmendozaventura/android-primeros-pasos-con-glide-f26a9db303ee> (visitado 25-04-2020).
- [24] *Visual Paradigm*. en. Page Version ID: 896993014. Mayo de 2019. URL: https://en.wikipedia.org/w/index.php?title=Visual_Paradigm&oldid=896993014 (visitado 05-05-2020).
- [25] Hai Zhang. *zhanghai/MaterialProgressBar*. original-date: 2015-06-06T14:31:56Z. Abr. de 2020. URL: <https://github.com/zhanghai/MaterialProgressBar> (visitado 25-04-2020).

Apéndice A

Manual de instalación

Para la instalación de la aplicación se necesita disponer de un teléfono móvil Android con una versión igual o superior a Android 5.1. La aplicación no está disponible en la tienda de aplicaciones de ‘Play Store’, por lo cual necesitarás almacenar el fichero apk en tu dispositivo móvil. Los requisitos que debes cumplir para poder instalar una aplicación desde el almacenamiento del dispositivo son los siguientes: activar ‘Orígenes desconocidos’ en tu dispositivo (A.1) y disponer de al menos 15MB de almacenamiento. En la figura (A.2) se muestra un ejemplo de los pasos para instalar la aplicación en un dispositivo móvil Xiaomi

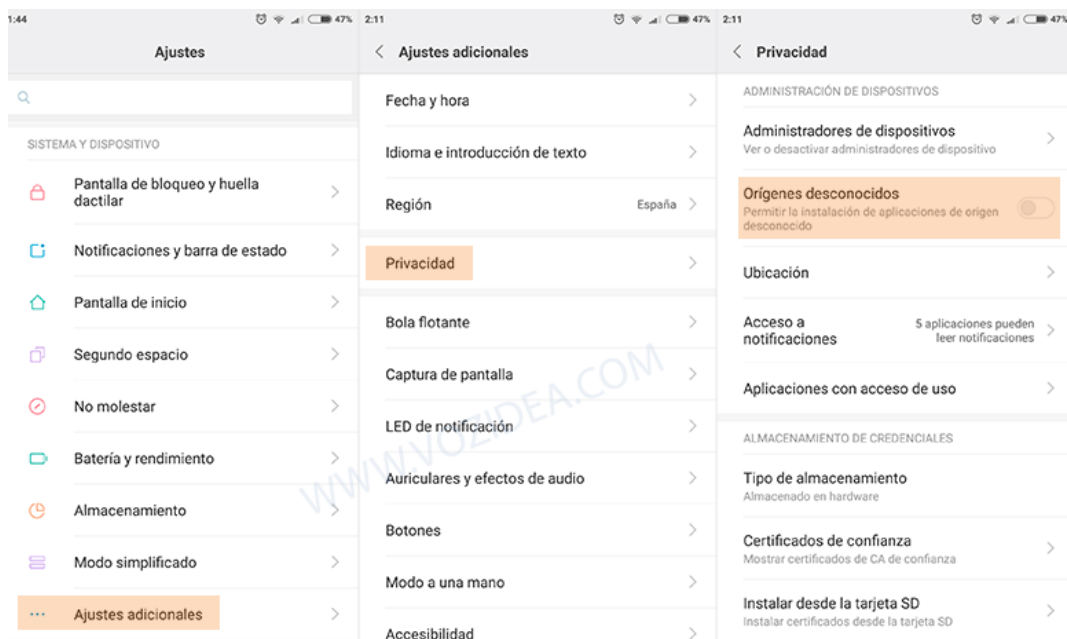
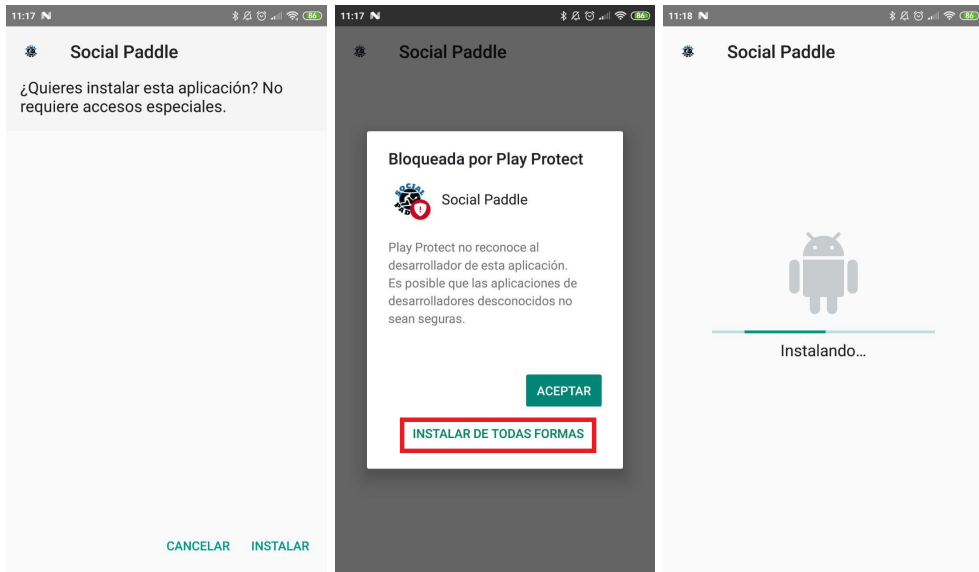


Figura A.1: Ejemplo activar orígenes desconocidos en móvil Xiaomi



(a) Paso I

(b) Paso II

(c) Paso III

Figura A.2: Ejemplo instalación en móvil Xiaomi

Apéndice B

Manual de usuario

El objetivo de este manual es mostrar, mediante una serie de imágenes y descripciones todas las funcionalidades de la aplicación.

Las imágenes mostradas a continuación han sido realizadas con un dispositivo móvil Xiaomi Redmi Note 4 el cual dispone de una pantalla de 5,5 pulgadas. Es posible que el aspecto de la aplicación pueda cambiar si el dispositivo es de una pantalla de menor tamaño. En cambio el funcionamiento es el mismo en todos los dispositivos.

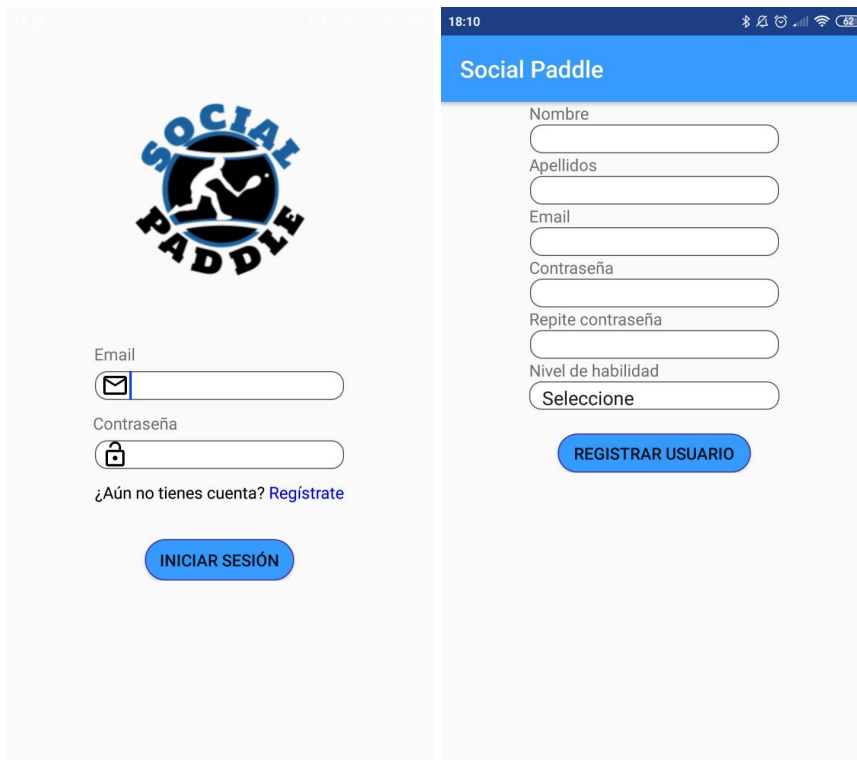
El primer paso de todos es abrir la aplicación la cual ya hemos instalado en nuestro dispositivo para esto pulsamos sobre el icono de la aplicación(B.1).



Figura B.1: Logo aplicación

B.1. Crear nueva cuenta.

El primer paso es ejecutar la aplicación, acto seguido se nos muestra la ventana de login (B.2a), para crear una cuenta pulsamos sobre el texto que dice “¿Aún no tienes cuenta? Regístrate” esto nos abre una nueva ventana (B.2b). En esta ventana introducimos nuestros datos de usuario, en el momento que tengamos todos nuestros datos pulsamos sobre el botón “Registrar usuario”. Si todos los datos son correctos, ya tenemos nuestra cuenta de usuario.



(a) Paso I

(b) Paso II

Figura B.2: Crear nueva cuenta

B.2. Iniciar sesión

En el momento que tenemos creada nuestra cuenta de usuario, ya podemos iniciar sesión en la aplicación (B.3), tan solo tenemos que introducir nuestro correo y contraseña y pulsar sobre el botón iniciar sesión.

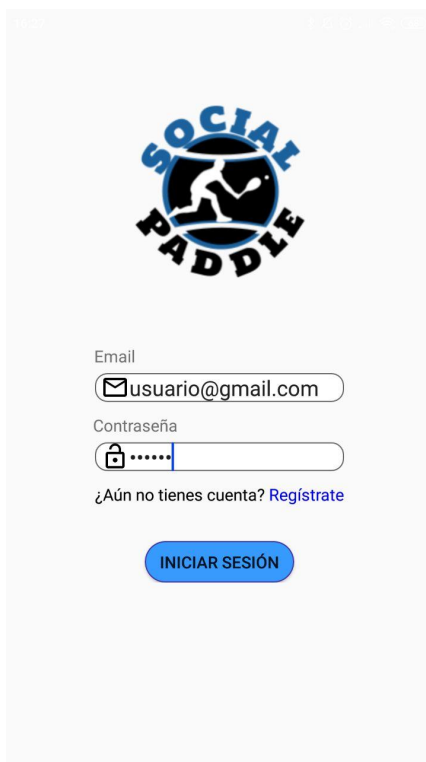


Figura B.3: Iniciar sesión

B.3. Cambiar contraseña

Para modificar nuestra contraseña de acceso a la aplicación, tenemos que pulsar sobre los tres puntos de la esquina superior derecha. Se nos mostrara un desplegable que nos ofrece la opción de cambiar la contraseña. Pulsando sobre el nos abrirá una nueva ventana (B.4b), en esta ventana indicaremos nuestra anterior contraseña y la nueva contraseña dos veces y pulsaremos sobre el botón "Cambiar contraseña". Si la contraseña cumple las características se cambiara y se cerrara sesión para que podamos iniciar sesión con nuestra nueva contraseña.

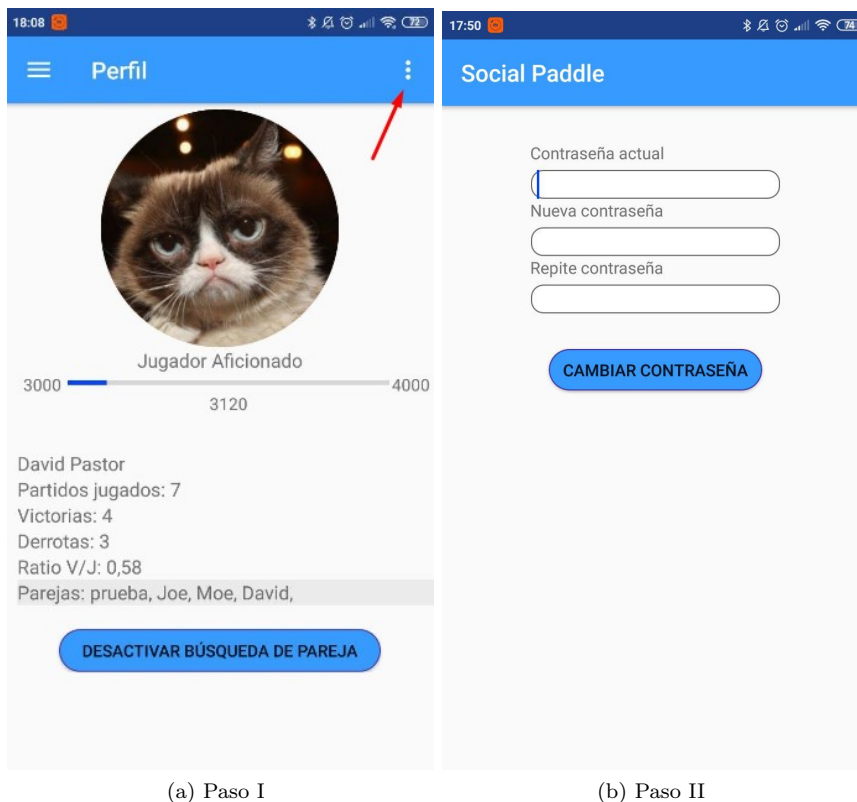


Figura B.4: Cambiar contraseña

B.4. Perfil de usuario

Para ver nuestro perfil de usuario tan solo tenemos que iniciar sesión con nuestra cuenta, y acto seguido se nos muestra la pantalla del perfil de usuario (B.5). En esta pantalla se nos indica todos los datos de nuestro usuario: nombre, partidos jugados, victorias, derrotas, ratio, foto de perfil, nivel de habilidad, estado de búsqueda de pareja y nuestras parejas.

En esta pantalla podemos realizar distintas acciones la primera de ellas es cambiar nuestra foto de perfil, para esto tenemos que pulsar sobre nuestra imagen y esto nos abrirá una nueva ventana. Otra acción en esta ventana es visualizar nuestras parejas. Para eso pulsamos sobre el texto que pone "Parejas:..." acto seguido nos llevara a una ventana en la cual vemos nuestras parejas y podremos eliminarlas si lo deseamos. La ultima acción que podemos realizar aquí es Activar o Desactivar la búsqueda de pareja.

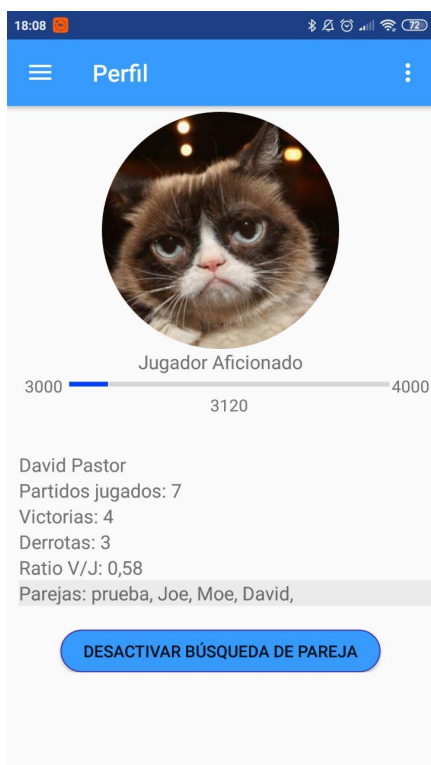


Figura B.5: Perfil de usuario

B.5. Editar foto

Si en la pantalla de perfil de usuario pulsamos sobre la imagen de perfil se nos muestra una pantalla con este aspecto (B.6). Para seleccionar una foto de nuestro gestor de archivos pulsamos sobre el icono con forma de lupa, esto nos abre el gestor de archivos de nuestro dispositivo, aquí podemos elegir la foto que mas nos guste. Cuando ya tenemos la foto seleccionada se nos muestra en la pantalla, para que podamos decidir si nos gusta el aspecto final. Si queremos confirmar esta foto basta con pulsar sobre el botón con el icono de una flecha hacia arriba el cual sube la imagen al servidor.

B.6. Ver y eliminar parejas

Si en la pantalla de perfil de usuario pulsamos sobre las parejas nos llevara a una pantalla con el siguiente aspecto (B.7). En esta pantalla veremos todas nuestras parejas y también podremos eliminar las parejas que no deseamos tener tan solo tendremos que pulsar sobre el icono de la papelera.

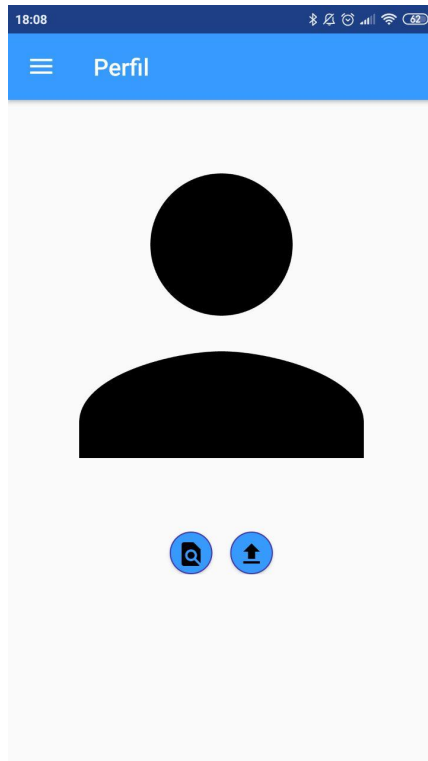


Figura B.6: Editar foto de perfil

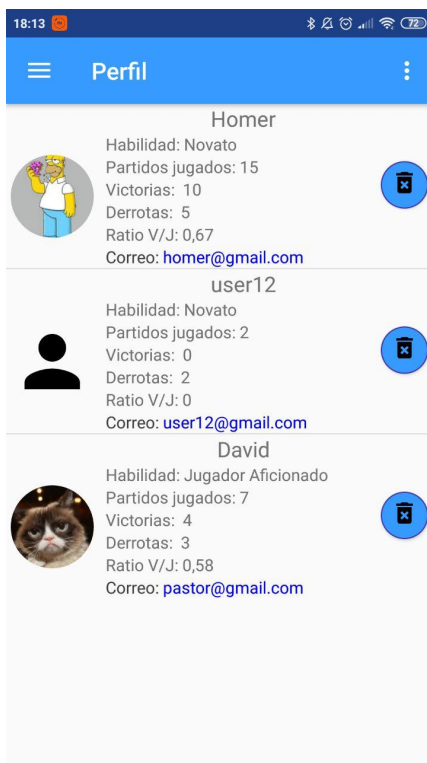


Figura B.7: Ver parejas

B.7. Menú lateral

Si pulsamos sobre el menú lateral se nos mostrara la siguiente pantalla (B.8) en la cual podremos ir cambiando de ventanas para realizar las distintas acciones que se explican a continuación.

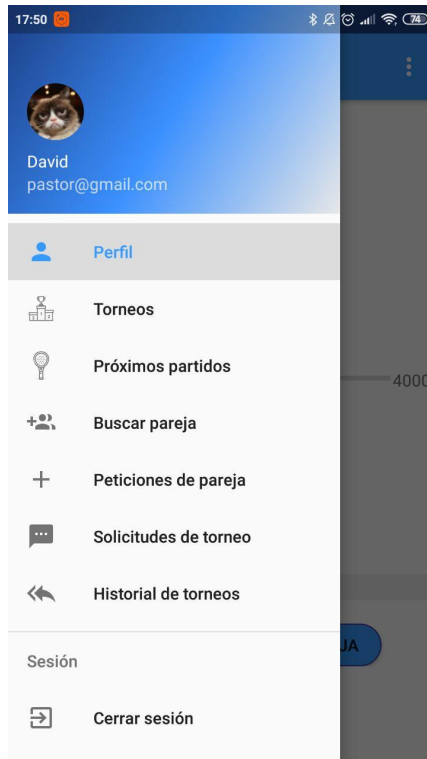
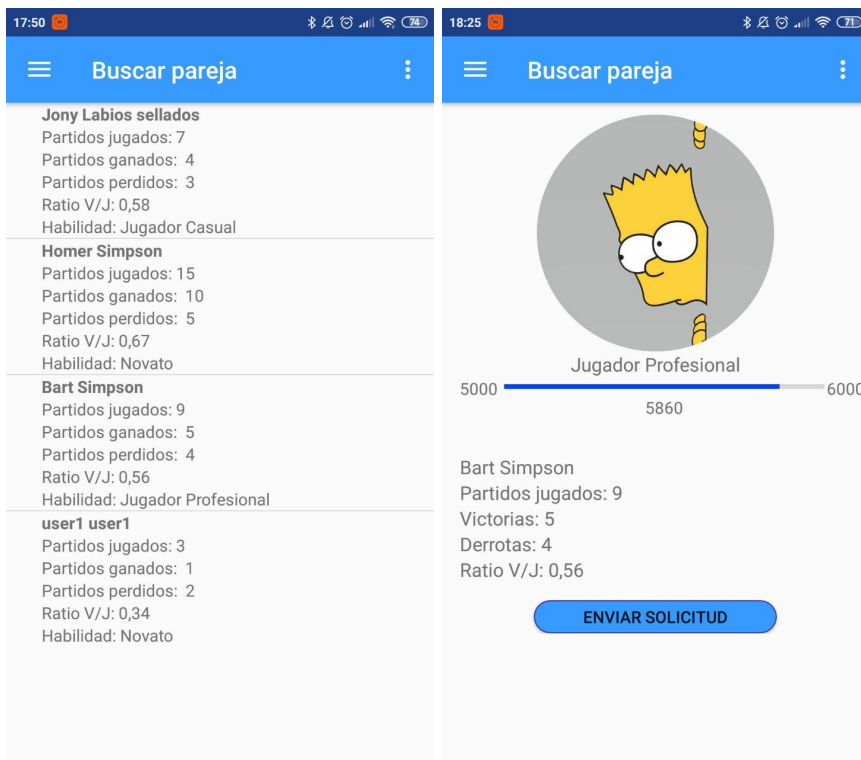


Figura B.8: Menú lateral

B.8. Buscar pareja

En esta ventana (B.9a) podemos seleccionar uno de los usuarios que se muestran en la lista y pedirle ser nuestra pareja de juego. Para esto primero tienes que pulsar sobre el usuario que deseas, acto seguido se te mostrara una pantalla, en esta pantalla (B.9b) puedes ver las características del usuario y si lo deseas puedes pulsar sobre el botón “Enviar solicitud”. Después de esto tan solo tienes que esperar a que este usuario acepte o rechace tu solicitud.



(a) Paso I

(b) Paso II

Figura B.9: Buscar pareja

B.9. Peticiones de pareja

Si queremos visualizar las peticiones que tenemos activas, pulsamos sobre la opción ‘‘Peticiones de pareja’’ en el men lateral. Se nos mostrara una pantalla similar a esta (B.10), en cada peticin de pareja podemos aceptar pulsando sobre el icono con forma de ‘‘Check’’ o podemos rechazar la peticin pulsando sobre el icono con forma de papelera.

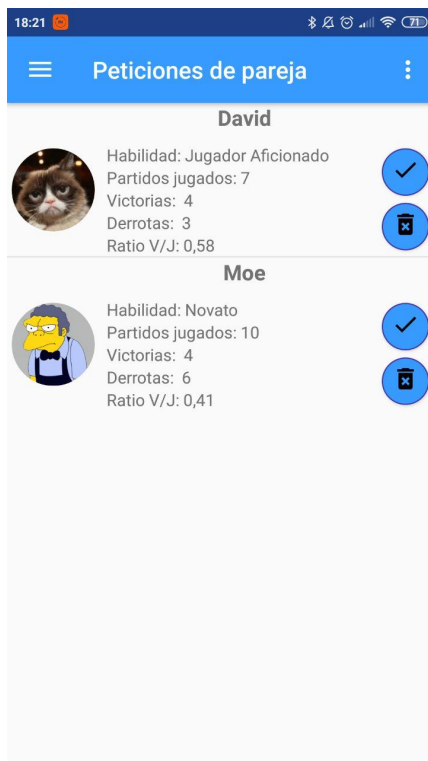


Figura B.10: Solicitud de pareja

B.10. Unir a un torneo

Para poder unirnos a un torneo primero tenemos que pulsar sobre ‘‘Torneos’’ en el men lateral, acto seguido se nos mostrara en una pantalla similar a esta (B.11a) todos los torneos que estn disponibles con una fecha posterior a la actual. Tenemos dos tipos de torneos en funcin de si queremos unirnos con una pareja, o queremos que el torneo nos asigne una pareja de manera aleatoria. Despus de seleccionar el tipo de torneo que deseamos se nos muestra una pantalla similar a esta (B.11b). En esta ventana podemos ver todas las caractersticas del torneo, si el torneo elegido es con pareja podremos seleccionar nuestra pareja en el ‘‘Spinner’’ y despus pulsamos sobre el botn ‘‘Unirme al torneo’’, en cambio si el torneo elegido es sin pareja (B.11c) tan solo tenemos que pulsar sobre el botn ‘‘Unirme al

torneo⁴. Si deseamos salir de un torneo el cual no ha comenzado podemos hacerlo pulsando sobre el botón “Desapuntarme del torneo”.

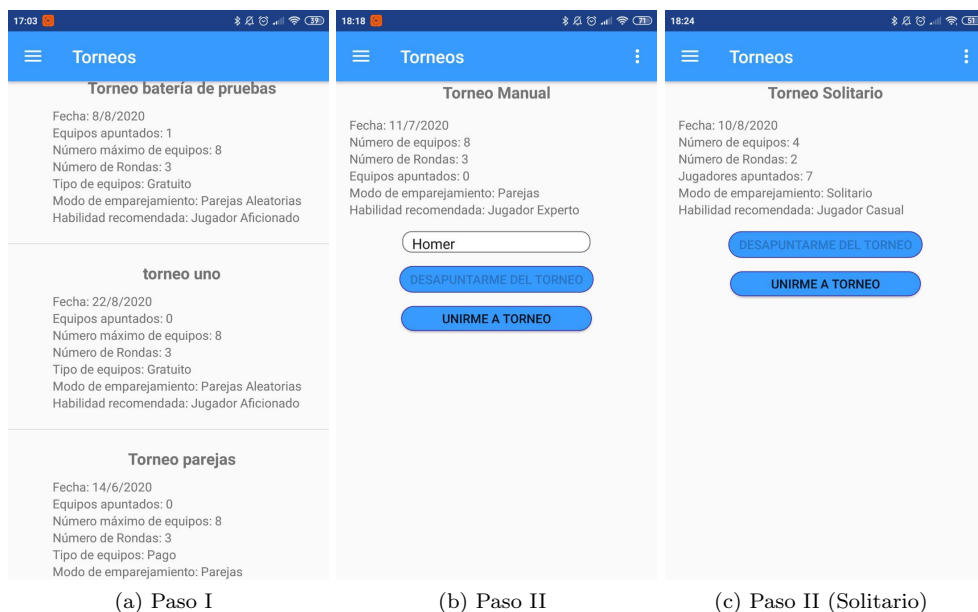


Figura B.11: Unir a un torneo

B.11. Próximos partidos

Para poder visualizar los próximos partidos primero tenemos que pulsar sobre el menú lateral y abrir esta opción. En esta ventana (B.12) podremos ver los próximos partidos que vamos a disputar, se nos mostrara la pareja contra la que jugaremos y la fecha del enfrentamiento.

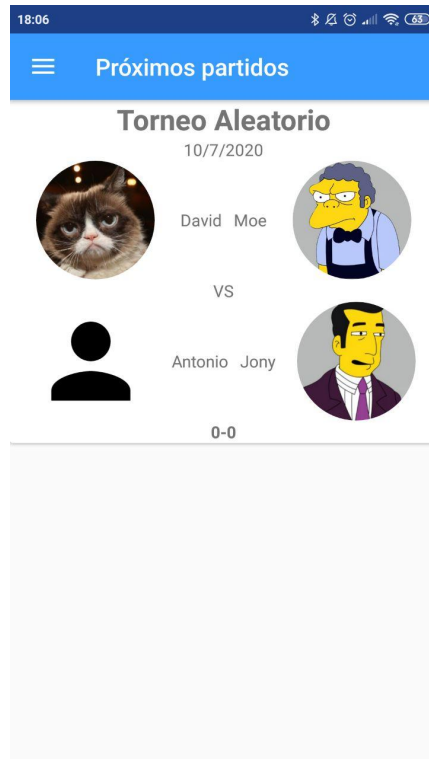


Figura B.12: Próximos partidos

B.12. Solicitud de participación en torneo

Para poder ver nuestras solicitudes para unirnos a un torneo, pulsamos sobre el menú lateral y seleccionamos la opción de ‘Solicitudes de torneo’, si disponemos de alguna solicitud se nos mostraran en pantalla, aquí tenemos la opción de aceptar o rechazar esta solicitud.

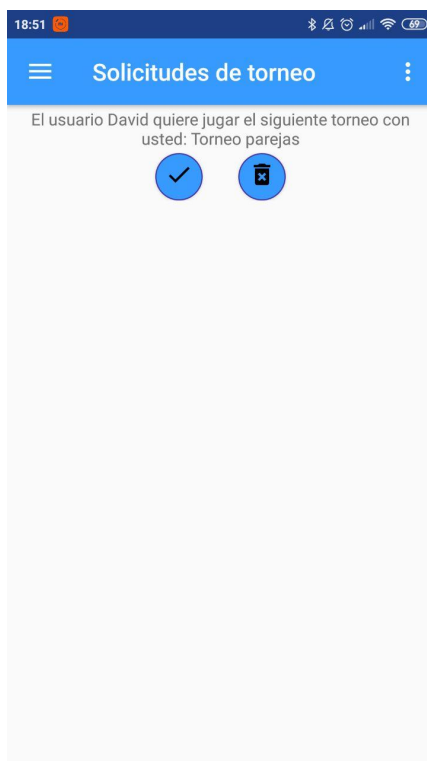


Figura B.13: Solicitudes de torneo

B.13. Historial de torneos

Para poder visualizar nuestro historial primero tenemos que pulsar sobre el menú lateral y abrir esta opción. Una vez dentro en esta ventana (B.14a) pulsamos sobre el torneo que deseamos visualizar y se nos mostrara una pantalla (B.14b) en la cual están todos los partidos que se jugaron en ese torneo.

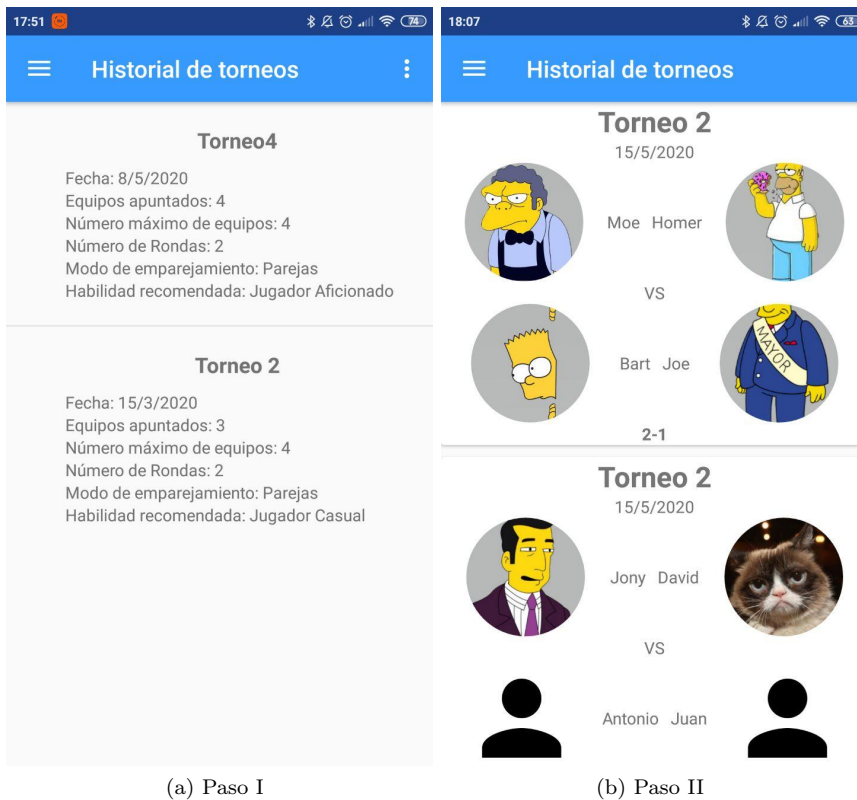


Figura B.14: Historial de torneos

B.14. Crear torneo (Administrador)

Para poder crear un nuevo torneo tienes que tener una cuenta de administrador, en la pantalla principal (B.15a) tienes que pulsar sobre el botón de la esquina inferior derecha, el cual es un icono con el símbolo “+”. Acto seguido se te abrirá una nueva ventana (B.15b), aquí tendremos que indicar todos los datos del torneo y cuando tengamos todo listo seleccionamos la fecha que deseemos, en este momento se nos abrirá una nueva ventana (B.15c) la cual recoge todos los datos indicados anteriormente, este paso es para confirmar si esta todo correcto, si pulsas sobre el botón confirmar se generara el torneo indicado.

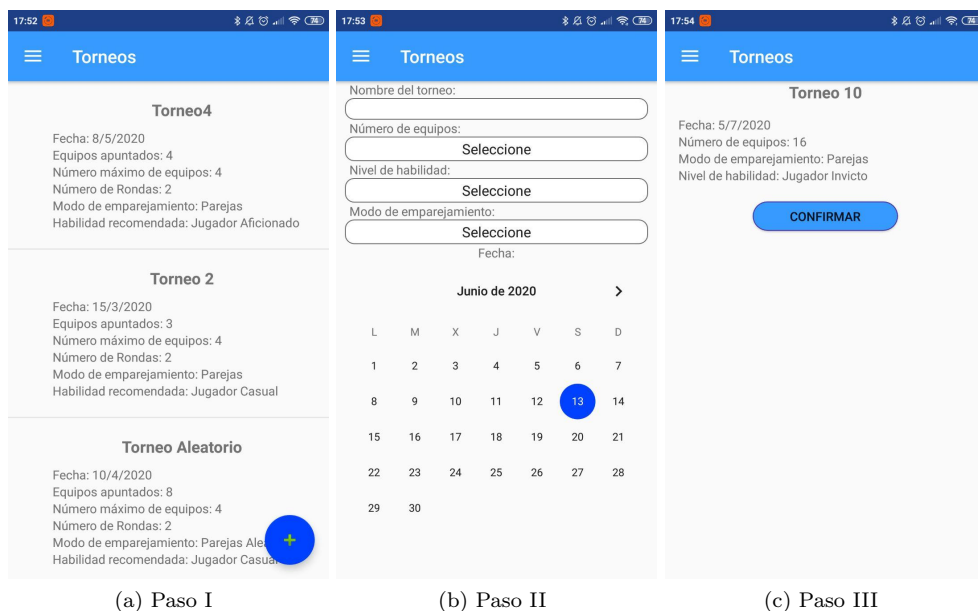
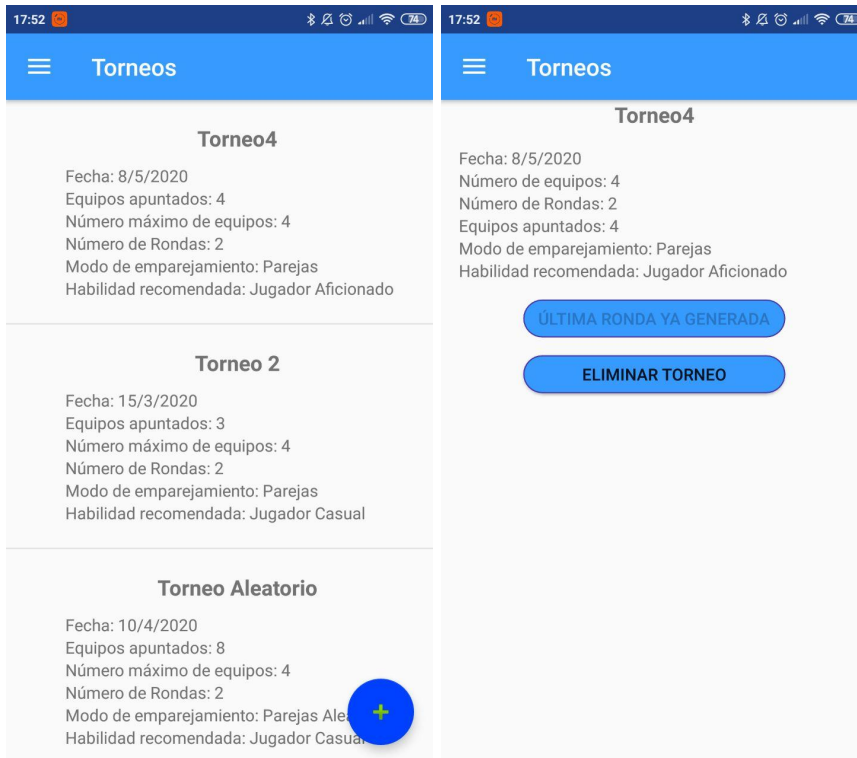


Figura B.15: Crear torneo (Administrador)

B.15. Eliminar torneo (Administrador)

Para eliminar un torneo lo primero que tenemos que hacer es buscarlo en la lista de torneos (B.16a), cuando pulsemos sobre el torneo que deseamos eliminar se nos mostrara una pantalla similar a esta (B.16b), en esta pestaña pulsamos sobre el botón “Eliminar torneo”, se nos mostrara un mensaje de confirmación para preguntarnos si estamos seguros y en el momento que pulsemos confirmar el torneo quedara eliminado.



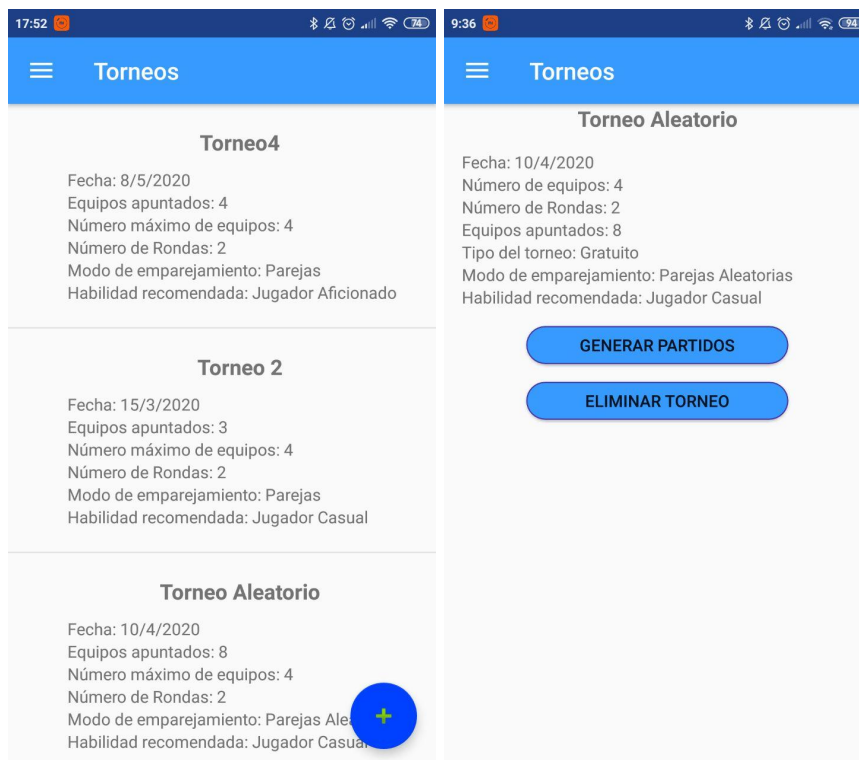
(a) Paso I

(b) Paso II

Figura B.16: Eliminar torneo (Administrador)

B.16. Generar rondas (Administrador)

Se pueden generar rondas en el momento que un torneo se completa de participantes o cuando una ronda termina. Para poder generar una ronda tenemos que seleccionar el torneo indicado en la ventana principal (B.17a), dentro de la ventana del torneo (B.17b) pulsamos sobre el botón ‘Generar partidos’, en ese mismo instante se generaran todos los partidos de la siguiente ronda.



(a) Paso I

(b) Paso II

Figura B.17: Generar rondas (Administrador)

B.17. Anotar resultados (Administrador)

Cuando generemos una ronda de partidos estos mismos partidos aparecerán en la ventana de anotar partidos, para acceder a esta ventana pulsamos sobre el menú lateral (B.18a) y seleccionamos esta opción, una vez dentro de la ventana (B.18b) veremos todos los partidos encabezados por el nombre del torneo al que pertenecen, para puntuar un partido pulsamos sobre los campos de texto y insertamos los valores, cuando los tengamos pulsamos sobre el botón con un icono de “Disquete”.

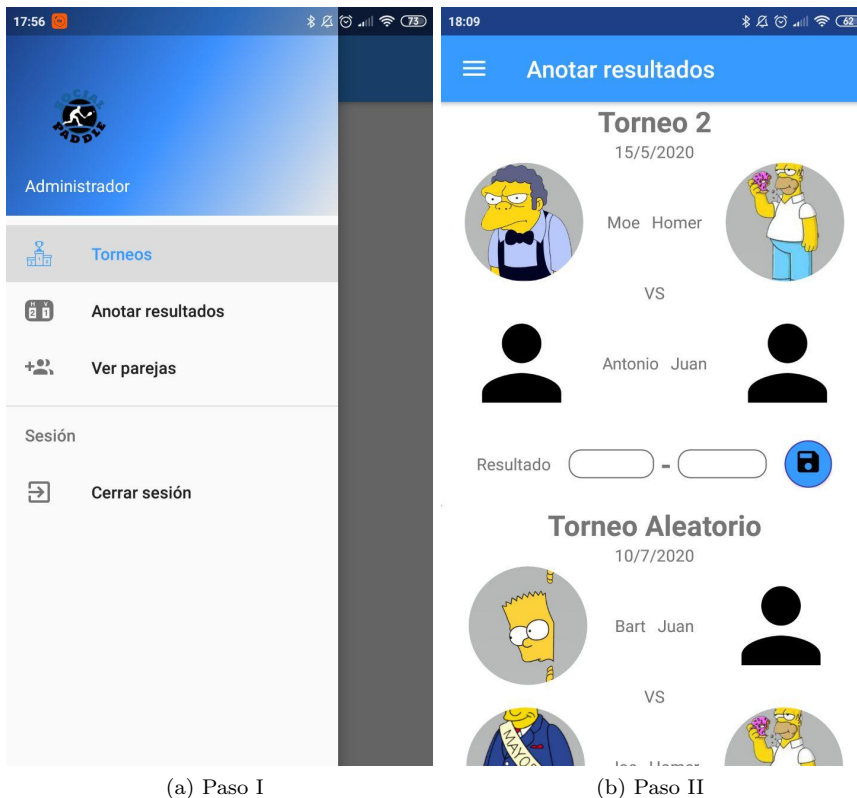


Figura B.18: Anotar resultados (Administrador)

B.18. Ver todas las parejas (Administrador)

El administrador puede visualizar todas las parejas que existan hasta el momento, para ver estas parejas tan solo tiene que pulsar sobre el botón “Ver parejas” en el menú lateral y se le abrirá una ventana en la cual se muestran todas las parejas.

Ver parejas	
David Jugador Aficionado Partidos jugados: 7 Victorias: 4 Derrotas: 3 Ratio V/J: 0,58 pastor@gmail.com	Moe Novato Partidos jugados: 10 Victorias: 4 Derrotas: 6 Ratio V/J: 0,41 moe@gmail.com
David Jugador Aficionado Partidos jugados: 7 Victorias: 4 Derrotas: 3 Ratio V/J: 0,58 pastor@gmail.com	Homer Novato Partidos jugados: 15 Victorias: 10 Derrotas: 5 Ratio V/J: 0,67 homer@gmail.com
Jony Jugador Casual Partidos jugados: 7 Victorias: 4 Derrotas: 3 Ratio V/J: 0,58 jony@gmail.com	David Jugador Aficionado Partidos jugados: 7 Victorias: 4 Derrotas: 3 Ratio V/J: 0,58 pastor@gmail.com
Bart Jugador Profesional Partidos jugados: 9 Victorias: 5 Derrotas: 4 Ratio V/J: 0,56 bart@gmail.com	David Jugador Aficionado Partidos jugados: 7 Victorias: 4 Derrotas: 3 Ratio V/J: 0,58 pastor@gmail.com

Figura B.19: Ver todas las parejas (Administrador)

Apéndice C

Contenidos del CD-ROM

Los contenidos que acompaña a la memoria entregada son los siguientes:

- MemoriaTFGDavidPastorPerez.pdf: copia de la memoria en formato pdf.
- Social-Paddle.apk: instalador para dispositivos Android.
- TournamentGenerator: código fuente de la aplicación.
- Imágenes: imágenes utilizadas en el desarrollo del proyecto.