



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

Trabajo Fin de Grado

Grado en Ingeniería Informática

(MENCIÓN TECNOLOGÍAS DE LA INFORMACIÓN)

**Cribi: Aplicación Android para la
gestión de eventos deportivos.**

Autor:

D. Abel García de Prada

Tutor:

D. Joaquín Adiego Rodríguez

Resumen

Es innegable el auge de los dispositivos móviles y las facilidades que aportan en nuestra vida. La gran evolución que han tenido estos terminales nos permite realizar prácticamente cualquier tarea desde nuestro 'Smartphone' de una manera simple, rápida y eficaz.

Además, la sociedad cada vez tiene una vida más sedentaria debido a las rutinas tan marcadas y la especialización de los trabajos. Es por ello que surge la necesidad de unir ambos mundos y utilizar el dispositivo móvil para facilitarnos la práctica de deporte.

Actualmente, si navegamos por las tiendas de aplicaciones, podemos encontrar infinidad de apps que nos permiten mantener un *tracking* de nuestros entrenamientos individuales y en gimnasios. Pero no podemos encontrar apenas aplicaciones que nos faciliten la práctica de deportes colectivos, que tan importantes son para nuestra salud física y mental.

Debido a esto, surge la idea de facilitar el acceso a estos deportes, puesto que normalmente se practican con un grupo de gente conocida y limitada, en la que si alguna persona no está disponible el partido no se puede disputar.

Cribi nace para solucionar estos problemas, ofreciendo una solución multideporte y en tiempo real que permite a los clubs ofrecer sus instalaciones libres para la práctica de partidos y a los jugadores unirse y conocer a otras personas que compartan su afición por el deporte.

Abstract

It is undeniable the rise of mobile devices and the facilities they bring to our lives. These terminals have evolved so much, that allow us to do almost any task from our 'Smartphone' in a simple, fast and efficient way.

In addition, sedentary lifestyles appear to be increasingly widespread due to the marked routines and specialization of jobs. This is why the need arises to use the mobile device to facilitate the practice of sport.

Nowadays, if we browse the app stores, we can find countless apps that allow us to keep a tracking of our workouts. But we can hardly find applications that facilitate the practice of collective sports, which are so important for our physical and mental health.

Because of this, the idea arises to facilitate access to these sports, since they are usually practiced with a group of friends and if someone is not available, the match can not be played.

Cribi was born to solve these problems, offering a multisport, realtime solution that allows clubs to offer their free fields for the practice of matches and players to join and meet other people who share their passion for sport.

Índice general

1. Introducción	1
1.1. Contexto y motivación	1
1.2. Objetivos	1
1.3. Metodología utilizada	2
1.4. Resumen del contenido de la memoria	2
2. Entorno tecnológico	3
2.1. Herramientas y tecnologías utilizadas	3
2.1.1. Android Studio	3
2.1.2. Github	3
2.1.3. Overleaf	3
2.1.4. Firebase	4
2.1.5. Android Image Cropper	5
2.1.6. Glide	5
2.2. Entorno de desarrollo	6
3. Plan de Desarrollo de Software	7
3.1. Introducción	7
3.1.1. Propósito	7
3.1.2. Alcance	7
3.2. Visión general del proyecto	8
3.2.1. Objetivos	8
3.2.2. Suposiciones y/o restricciones	8
3.2.3. Características del proyecto	8
3.2.4. Elección de la metodología de desarrollo de software	9
3.2.5. Entregables del proyecto	9
3.3. Organización del proyecto	10
3.3.1. Roles del proyecto	10
3.3.2. Reuniones, eventos y planes operativos	10
3.4. Estimaciones del proyecto	10
3.4.1. Estimación temporal	10
3.4.2. Estimación de costes	11
3.4.3. Desviación real del proyecto	11
4. Seguimiento del proyecto	12
4.1. Introducción	12
4.2. Sprint 1: 4 de Febrero - 24 de Febrero	13
4.2.1. US01 - Como Equipo quiero tener configurado el entorno de trabajo completo para poder empezar a trabajar en el proyecto	13
4.2.2. US23 - Como Product Owner quiero tener una documentación inicial del proyecto para tener una visión global del mismo	14
4.2.3. US02 - Como usuario Jugador quiero poder registrarme en la aplicación mediante email y contraseña.	14

4.2.4.	US03 - Como usuario Jugador quiero poder identificarme de forma única en la aplicación de jugadores con mis credenciales.	14
4.2.5.	US04 - Como usuario Club quiero poder identificarme de forma única en la aplicación de Clubes con mis credenciales.	15
4.2.6.	US24 - Como Product Owner quiero tener un modelo de dominio inicial de ambas aplicaciones a desarrollar.	15
4.3.	Sprint 2: 25 de Febrero - 17 de Marzo	16
4.3.1.	US05 - Como usuario Jugador quiero poder crear y editar mi perfil.	16
4.3.2.	US06 - Como usuario Jugador quiero poder añadir una imagen de perfil.	16
4.3.3.	US07 - Como usuario Club quiero poder crear y editar mi perfil.	17
4.3.4.	US08 - Como usuario Club quiero poder añadir el logo de mi club a mi perfil.	17
4.4.	Sprint 3: 18 de Marzo - 7 de Abril	17
4.4.1.	US09 - Como usuario Jugador quiero poder visualizar mi perfil de usuario.	18
4.4.2.	US10 - Como usuario Jugador quiero poder cerrar mi sesión.	18
4.4.3.	US11 - Como usuario Club quiero poder visualizar mi perfil de club.	18
4.4.4.	US12 - Como usuario Club quiero poder cerrar mi sesión.	19
4.4.5.	US25 - Como Product Owner quiero tener una primera versión de la arquitectura y diseño del proyecto.	19
4.5.	Sprint 4: 8 de Abril - 28 de Abril	19
4.5.1.	US13 - Como usuario Club quiero poder crear nuevos partidos para que los jugadores puedan visualizar y acceder a ellos.	20
4.5.2.	US14 - Como usuario Jugador quiero poder ver un listado de los partidos disponibles para unirme.	20
4.5.3.	US15 - Como usuario Club quiero poder ver un listado completo de todos los partidos que he creado para tener un seguimiento en tiempo real.	20
4.6.	Sprint 5: 29 de Abril - 19 de Mayo	21
4.6.1.	US16 - Como usuario quiero consultar la información completa de un partido.	21
4.6.2.	US17 - Como usuario quiero consultar el listado de jugadores que se han inscrito a un partido.	21
4.6.3.	US18 - Como usuario Jugador quiero consultar el perfil del club que ha organizado el partido.	22
4.7.	Sprint 6: 20 de Mayo - 9 de Junio	22
4.7.1.	US19 - Como usuario Jugador quiero filtrar la lista de partidos según diferentes parámetros.	22
4.7.2.	US20 - Como usuario Club quiero filtrar la lista de mis partidos según el estado de los mismos.	23
4.7.3.	US21 - Como usuario Jugador quiero poder unirme y abandonar un partido.	23
4.7.4.	US22 - Como usuario Club quiero poder actualizar el estado de un partido.	24
4.8.	Sprint 7: 10 de Junio - 23 de Junio	24
4.8.1.	US26 - Como Product Owner quiero disponer del Plan de Desarrollo de Software final y su seguimiento.	24
4.8.2.	US27 - Como Product Owner quiero disponer de la memoria completa del proyecto.	24
5.	Plan de Gestión de Riesgos	26
5.1.	Introducción	26
5.1.1.	Gestión del riesgo	26
5.1.2.	Control de riesgos	28
6.	Requisitos	29
6.1.	Los requisitos en el desarrollo ágil	29
6.2.	Requisitos funcionales	29
6.3.	Requisitos no funcionales	30
6.4.	Requisitos de información	31

7. Análisis	32
7.1. Modelo de Casos de Uso	32
7.1.1. Actores	32
7.1.2. Diagrama de casos de uso	32
7.1.3. Especificación de casos de uso	33
7.1.4. Modelo de Dominio	40
7.1.5. Descripción de las clases del modelo de dominio	41
7.1.6. Diagramas de actividad Cribi	42
7.1.7. Diagramas de actividad Cribi Clubs	53
8. Diseño	58
8.1. Diseño de la arquitectura	58
8.1.1. Patrón arquitectónico MVVM	58
8.1.2. Arquitectura general	59
8.1.3. Estructura del proyecto	61
8.1.4. Interacción entre componentes	61
8.2. Diseño de la base de datos no relacional	62
9. Pruebas	63
9.1. Introducción	63
9.2. Casos de prueba	63
9.2.1. Aplicación para jugadores Cribi	63
9.2.2. Aplicación para organizadores Cribi Clubs	67
10. Conclusiones	69
10.1. Valoración personal	69
10.2. Líneas futuras	69
11. Bibliografía	70
A. Manual de usuario	72
A.1. Introducción	72
A.2. Cribi - Aplicación para los jugadores	72
A.2.1. Identificación de usuarios	72
A.2.2. Registro de nuevos usuarios	73
A.2.3. Página principal	74
A.2.4. Home	74
A.2.5. Filtrar partidos	75
A.2.6. Ver información completa del partido	76
A.2.7. Ver listado de jugadores inscritos	77
A.2.8. Ver perfil de club organizador	78
A.2.9. Unirse a partido	79
A.2.10. Abandonar partido	80
A.2.11. Perfil de usuario	81
A.2.12. Editar perfil de usuario	81
A.2.13. Cerrar sesión	82
A.2.14. Listado de partidos inscritos	82
A.3. Cribi Clubs - Aplicación para Organizadores	83
A.3.1. Identificación de Clubs	83
A.3.2. Página principal Cribi Clubs	83
A.3.3. Ver información del partido	84
A.3.4. Cambiar estado del partido	85
A.3.5. Crear nuevo partido	85
A.3.6. Ajustes	86

A.3.7. Editar Perfil	87
B. Manual de instalación	88
C. Contenido del soporte digital	89

Índice de figuras

2.1. Logotipo de Android Studio	3
2.2. Logotipo de Firebase	4
2.3. Jerarquía de Firestore	5
2.4. Logotipo de Glide	6
4.1. Flujo de trabajo de Scrum	12
7.1. Diagrama de casos de uso actor Jugador	32
7.2. Diagrama de casos de uso actor Organizador	33
7.3. Modelo de dominio de la aplicación.	40
7.4. Diagrama de actividad CU01.	42
7.5. Diagrama de actividad CU02.	43
7.6. Diagrama de actividad CU03.	44
7.7. Diagrama de actividad CU04.	45
7.8. Diagrama de actividad CU05.	46
7.9. Diagrama de actividad CU06.	47
7.10. Diagrama de actividad CU07.	48
7.11. Diagrama de actividad CU08.	49
7.12. Diagrama de actividad CU09.	50
7.13. Diagrama de actividad CU10.	51
7.14. Diagrama de actividad CU11.	52
7.15. Diagrama de actividad CU12.	53
7.16. Diagrama de actividad CU16.	54
7.17. Diagrama de actividad CU17.	55
7.18. Diagrama de actividad CU20.	56
7.19. Diagrama de actividad CU21.	57
8.1. Arquitectura MVVM	58
8.2. Arquitectura MVVM en conjunto con Firestore	59
8.3. Paquetes aplicación Cribi	61
8.4. Paquetes aplicación Cribi Clubs	61
8.5. Interacción componentes Club.	62
8.6. Interacción componentes Match.	62
A.1. Página de identificación.	72
A.2. Fallo en identificación.	72
A.3. Página de registro de usuarios.	73
A.4. Registrando nuevo usuario.	73
A.5. Fallo al registrarse.	73
A.6. Configurar perfil de usuario.	73
A.7. Página principal.	74
A.8. Menú de navegación.	74
A.9. Listado de partidos.	75
A.10. Filtrar partidos.	75
A.11. Modal de selección de filtros.	75

A.12.Listado de partidos filtrado.	75
A.13.Información del partido.	76
A.14.Acciones disponibles.	76
A.15.Partido completo.	77
A.16.Partido confirmado por el club.	77
A.17.Partido jugado.	77
A.18.Partido cancelado.	77
A.19.Lista de jugadores vacía.	78
A.20.Lista de jugadores inscritos.	78
A.21.Perfil del Organizador.	78
A.22.Diálogo de unión a partido.	79
A.23.Uniéndose al partido.	79
A.24.Éxito al unirse al partido.	79
A.25.Fallo al unirse al partido.	79
A.26.Diálogo para abandonar un partido.	80
A.27.Abandonando partido.	80
A.28.Éxito al abandonar el partido.	80
A.29.Fallo al abandonar partido.	80
A.30.Perfil de usuario.	81
A.31.Editar perfil de usuario.	81
A.32.Añadir imagen de perfil.	81
A.33.Cerrar sesión de usuario.	82
A.34.Cierre de sesión completo.	82
A.35.Listado de partidos inscritos.	82
A.36.Página de identificación de clubes.	83
A.37.Fallo en identificación de club.	83
A.38.Página principal Cribi Clubs.	83
A.39.Menú de navegación Cribi Clubs.	83
A.40.Información del partido	84
A.41.Cambiar estado de partido.	85
A.42.Cambio de estado exitoso.	85
A.43.Crear nuevo partido.	86
A.44.Creación de partido con éxito.	86
A.45.Ajustes Cribi Clubs.	86
A.46.Editar perfil de club.	87
A.47.Selector de logo del club.	87

Índice de tablas

2.1. Características del ordenador portátil utilizado para el desarrollo	6
3.1. Asignación de roles del proyecto	10
3.2. Eventos de Scrum	10
3.3. Iteraciones del proyecto	11
4.1. Historias de usuario del Sprint 1	13
4.2. Historias de usuario del Sprint 2	16
4.3. Historias de usuario del Sprint 3	17
4.4. Historias de usuario del Sprint 4	19
4.5. Historias de usuario del Sprint 5	21
4.6. Historias de usuario del Sprint 6	22
4.7. Historias de usuario del Sprint 7	24
6.1. Requisitos funcionales Cribi	30
6.2. Requisitos funcionales Cribi Clubs	30
6.3. Requisitos no funcionales	31
6.4. Requisitos no funcionales de información	31
7.1. Descripción de CU01	33
7.2. Descripción de CU02	34
7.3. Descripción de CU03	34
7.4. Descripción de CU04	34
7.5. Descripción de CU05	35
7.6. Descripción de CU06	35
7.7. Descripción de CU07	35
7.8. Descripción de CU08	35
7.9. Descripción de CU09	36
7.10. Descripción de CU10	36
7.11. Descripción de CU11	36
7.12. Descripción de CU11	37
7.13. Descripción de CU13	37
7.14. Descripción de CU14	37
7.15. Descripción de CU15	38
7.16. Descripción de CU16	38
7.17. Descripción de CU17	38
7.18. Descripción de CU18	39
7.19. Descripción de CU19	39
7.20. Descripción de CU20	39
7.21. Descripción de CU21	40
7.22. Descripción de CU22	40
9.1. Descripción del CP01	63
9.2. Descripción del CP02	63
9.3. Descripción del CP03	64

9.4. Descripción del CP04	64
9.5. Descripción del CP05	64
9.6. Descripción del CP06	64
9.7. Descripción del CP07	64
9.8. Descripción del CP08	65
9.9. Descripción del CP09	65
9.10. Descripción del CP10	65
9.11. Descripción del CP11	65
9.12. Descripción del CP12	65
9.13. Descripción del CP13	66
9.14. Descripción del CP14	66
9.15. Descripción del CP15	66
9.16. Descripción del CP16	66
9.17. Descripción del CP17	67
9.18. Descripción del CP18	67
9.19. Descripción del CP19	67
9.20. Descripción del CP20	67
9.21. Descripción del CP21	67
9.22. Descripción del CP22	67
9.23. Descripción del CP23	68
9.24. Descripción del CP24	68
9.25. Descripción del CP25	68
9.26. Descripción del CP26	68
9.27. Descripción del CP27	68

Capítulo 1

Introducción

1.1. Contexto y motivación

El ejercicio físico es una pieza fundamental para mantener una vida sana. La práctica de un deporte de manera regular favorece no solo a la salud física, sino también a la salud mental, y permite mantener un estilo de vida sano, disminuyendo así problemas de salud derivados del sedentarismo.

El sedentarismo es uno de los estilos de vida más comunes en la actualidad. España es uno de los diez países más sedentarios de Europa [1]. Esto es debido, principalmente, a la facilidad que nos proporcionan las ciudades actualmente y al auge de trabajos cada vez más técnicos. El sedentarismo conlleva riesgos para la salud física como el sobrepeso, la propensión a enfermedades cardiovasculares o el aumento de riesgo de algunos tipos de cáncer. Pero también afecta a la salud mental, pudiendo producir ansiedad, estrés o depresión.

El auge de las nuevas tecnologías y de los dispositivos móviles abre un mundo de posibilidades para el desarrollo de aplicaciones que permitan a los usuarios abandonar este estilo de vida. Actualmente en la Play Store de Google podemos encontrar gran cantidad de aplicaciones para realizar ejercicio físico con rutinas individuales para abandonar el sedentarismo y mantenerse activo. Pero normalmente son eso, individuales.

Es por ello que surge la idea de crear una aplicación que proporcione la opción de organizar partidos de deportes colectivos, permitiendo así a los usuarios mantenerse activos y entablar relaciones sociales con otras personas que compartan la misma afición por el deporte.

1.2. Objetivos

El objetivo principal de este Trabajo de Fin de Grado consiste en el desarrollo de una aplicación Android que sea de utilidad tanto para los organizadores que ofrecen sus pistas, como para los jugadores que buscan otros jugadores para disputar un partido.

A partir de este objetivo principal, se derivan unos objetivos secundarios que se detallan a continuación:

- Afianzar los conocimientos del lenguaje de programación Java para el desarrollo de aplicaciones móviles android nativas.
- Aprender a planificar un proyecto software.
- Manejar con cierta soltura herramientas de desarrollo de software Android como Android Studio.
- Aprender a estructurar una aplicación Android con una arquitectura MVVM.
- Elaboración de una memoria de un proyecto software real.

1.3. Metodología utilizada

Debido al auge de las metodologías ágiles para el desarrollo de software, se ha utilizado Scrum para llevar a cabo este proyecto.

Este modelo de desarrollo se caracteriza principalmente por lo siguiente:

- Desarrollo incremental en *sprints*.
- Cada iteración aporta valor al producto.
- Flexibilidad ante cambios.
- Permite solapar diferentes etapas del desarrollo software.

1.4. Resumen del contenido de la memoria

A continuación se describe la estructura de la memoria del proyecto:

- **Capítulo 1. Introducción:** se expone el contexto, los objetivos y la metodología utilizada.
- **Capítulo 2. Entorno tecnológico:** se describen las herramientas utilizadas para el desarrollo del proyecto.
- **Capítulo 3. Plan de Desarrollo de Software:** se muestra la visión global del proyecto.
- **Capítulo 4. Seguimiento del proyecto:** se muestra el seguimiento completo del desarrollo del proyecto.
- **Capítulo 5. Plan de Gestión de Riesgos:** se exponen los riesgos del desarrollo, se fijan planes de acción y se detallan los riesgos producidos.
- **Capítulo 6: Requisitos:** se identifican los requisitos funcionales, no funcionales y de información.
- **Capítulo 7: Análisis:** se enumeran actores del sistema, los casos de uso, el modelo de dominio y los diagramas de actividad.
- **Capítulo 8: Diseño:** se muestran los patrones arquitectónicos y de diseño utilizados y el modelo general de la arquitectura.
- **Capítulo 9: Pruebas:** se muestran las pruebas realizadas para comprobar el correcto funcionamiento de toda la funcionalidad desarrollada.
- **Capítulo 10: Conclusiones:** se exponen los objetivos alcanzados, la valoración general del alumno y las líneas futuras del proyecto.
- **Bibliografía:** se enumera la documentación utilizada para la realización del proyecto.
- **Anexos:** se muestra el manual de usuario, el manual de instalación y el contenido del soporte digital.

Capítulo 2

Entorno tecnológico

2.1. Herramientas y tecnologías utilizadas

En esta sección se detallarán las herramientas y tecnologías principales utilizadas para el desarrollo de la aplicación. Esto incluye qué es y cómo funciona Android, el uso de tecnologías como Firebase y las APIs externas utilizadas.

2.1.1. Android Studio

Android Studio[2] es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado en 2013 y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android.



Figura 2.1: Logotipo de Android Studio

Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, macOS y GNU/Linux. Para este proyecto se ha utilizado la versión 3.4.0.

2.1.2. Github

Github[3] es una plataforma de desarrollo colaborativo de software que se ha utilizado para gestionar el control de versiones de Git de la aplicación y alojar el código fuente de la misma.

2.1.3. Overleaf

Overleaf es un servicio de LaTeX colaborativo en línea que se ha utilizado para la realización de esta memoria. Cuenta con una detallada documentación [4] que se ha consultado de manera frecuente para la consecución de la estructura de la memoria del proyecto.

2.1.4. Firebase

Firebase[5] es una plataforma que permite a los desarrolladores crear aplicaciones relativamente fácil y sencillo. Ofrece una gran variedad de servicios ya construidos y preparados para utilizar, así como herramientas para determinar la experiencia de los usuarios en la aplicación.

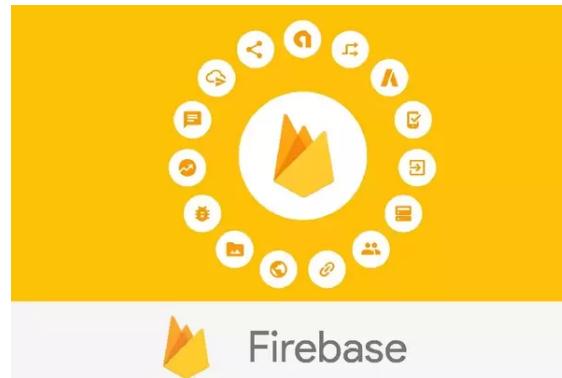


Figura 2.2: Logotipo de Firebase

Es parte de la creciente tendencia conocida como "Back end como un servicio". No es necesario crear y configurar un servidor. No se necesita comprobar si ha habido cambios en el servidor cada vez. No es indispensable crear una API para recuperar los datos. Básicamente Firebase libera a los desarrolladores de la gestión del backend y permite centrar el esfuerzo en la experiencia de usuario.

Razones por las que se eligió Firebase para el desarrollo del proyecto:

- Firebase ofrece, como se verá posteriormente, la posibilidad de añadir un mecanismo de autenticación en la aplicación para un acceso seguro a la misma.
- Firebase facilita un backend en tiempo real para la aplicación de fácil integración que permite centrar los esfuerzos en el desarrollo de la misma.
- Firebase ofrece almacenamiento en la nube para poder almacenar cierta cantidad de archivos dependiendo sus planes.
- Firebase proporciona estadísticas en tiempo real de los usuarios en la aplicación y de cómo es su experiencia en la app.

Firestore Authentication

Como la mayoría de aplicaciones, Cribi también necesita la identificación de usuarios para proporcionar una experiencia de usuario óptima. Para ello, se ha utilizado Firebase Authentication [6]. Esta funcionalidad de Firebase permite registrar y autenticar a los usuarios de forma segura, además de integrarla con otras funcionalidades de la plataforma.

Firestore Firestore

Para el almacenamiento de datos, Firebase ofrece dos opciones. Ambas son NoSQL, flexibles, escalables y en la nube. La primera es Realtime Database [7], que ofrece un almacenamiento en formato JSON y cuyas limitaciones se encuentran, principalmente, en el número de usuarios concurrentes.

En segundo lugar, Firebase Firestore [8], que proporciona un almacenamiento de datos en documentos que contienen campos que se asignan a valores. Estos documentos se almacenan en colecciones, que son contenedores para los documentos y que podemos usar para organizar los datos y compilar consultas. Esta jerarquía se muestra en la Figura 2.3. La principal limitación de este tipo de almacenamiento es que ofrece una cantidad determinada y limitada de lecturas, escrituras y borrados al día/mes.

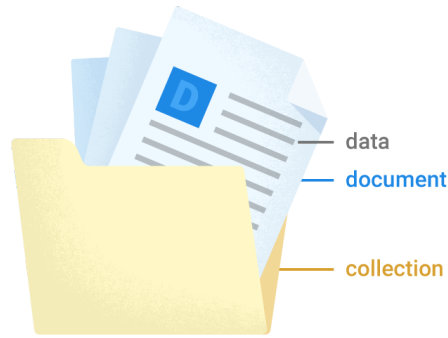


Figura 2.3: Jerarquía de Firestore

Las diferencias entre estos almacenamientos están resaltadas de forma acertada en esta referencia [9]. Finalmente, se utilizó Firestore para el almacenamiento por las siguientes razones:

- **Escalabilidad:** La jerarquía de documentos y colecciones permite un almacenamiento más sencillo y más escalable. Con Realtime Database, aparece cierto límite en número de conexiones concurrentes y número de escrituras por segundo en la misma base de datos. A partir de ahí, tendremos que fragmentar los datos en distintas bases de datos. En Firestore ese problema desaparece, puesto que escala de forma automática sin importar el número de usuarios que tenga la aplicación.
- **Consultas:** Firestore ofrece consultas indexadas con ordenamiento y filtrado compuestos. A diferencia de Realtime Database, se puede encadenar filtros y combinar distintos tipos de filtros por colecciones, subcolecciones o documentos. También dispone de operaciones atómicas de escritura y transacciones que se repiten automáticamente hasta completarse.
- **Coste:** En Realtime Database se cobra por número de usuarios concurrentes, por ancho de banda y almacenamiento. En cambio, en Firestore el coste viene dado por operaciones de lectura, escritura y borrado en la base de datos y con una tarifa inferior por ancho de banda y almacenamiento. Es por ello, que con una buena estructura, se pueden optimizar y reducir los costes de Firestore frente a Realtime.

Firestore Storage

Para el almacenamiento de archivos se ha usado Cloud Storage [10].

Cloud Storage para Firebase es un servicio de almacenamiento de objetos simple y gratuito hasta cierta cuota construido para almacenar contenido generado por el usuario. Esto, añadido a la sencilla integración con los demás servicios de Firebase, lo convierten en una buena opción para el almacenamiento. En este caso, será de utilidad para almacenar imágenes de perfil.

2.1.5. Android Image Cropper

Se ha utilizado esta librería [11] para poder obtener las imágenes de perfil de los usuarios. De esta forma, con los permisos necesarios, se puede generar un selector de imágenes que el usuario tendrá en el dispositivo. Con esto se pueden recortar las imágenes para reducir su tamaño y limitar sus dimensiones (formato cuadrado) y obtener así una mejor disposición en la interfaz.

2.1.6. Glide

Glide [12] es una librería de código abierto que permite una carga rápida y eficiente de imágenes en la interfaz. Ofrece una API de fácil uso para la carga de imágenes tanto locales como remotas. Además, dispone de la posibilidad de cargar imágenes por defecto en caso de que falle la obtención de la imagen por cualquier motivo.



Figura 2.4: Logotipo de Glide

2.2. Entorno de desarrollo

El entorno de trabajo durante el desarrollo del proyecto ha sido un ordenador portátil Acer con las siguientes características:

Equipo	
Marca	Acer
Modelo	Aspire F5-573G
Hardware	
Procesador	Intel Core i7-6500U
Memoria	16Gb DDR4
Disco duro	1 Tb 7200 rpm.
Tarjeta gráfica	Nvidia GeForce GTX 950M
Pantalla	15,6" 1920 x 1080
Software	
OS	Windows 10 Home

Tabla 2.1: Características del ordenador portátil utilizado para el desarrollo

Además del dispositivo anterior, se han utilizado diferentes dispositivos móviles para realizar las pruebas.

Entre ellos se encuentran:

- **Xiaomi Redmi Note 4x:** Android 7.0.
- **Xiaomi Redmi Note 5:** Android 8.1.0.
- **Honor 9 Lite:** Android 9.

Con la selección de estos dispositivos, la aplicación se ha probado en diferentes versiones de android y diferentes tamaños y resoluciones de pantalla.

Capítulo 3

Plan de Desarrollo de Software

3.1. Introducción

En esta sección de la memoria se detallará la visión global del proyecto. Además se comentará la decisión de la utilización del modelo de desarrollo ágil Scrum para este proyecto Software.

3.1.1. Propósito

El propósito del Plan de Desarrollo de Software es proporcionar la información detallada para realizar un control sobre el proyecto a realizar.

Como usuarios de este documento se encuentran:

- **El alumno:** se encargará de analizar, diseñar, planificar, desarrollar y documentar el proyecto.
- **El tutor del TFG:** se encargará de revisar la implementación y documentación del proyecto, así como su avance mediante la aportación de nuevas ideas y soluciones a los problemas que puedan surgir.

3.1.2. Alcance

En esta memoria se incluye una visión general del proyecto, se identifican los participantes y sus roles y se desarrolla un plan de trabajo en iteraciones. Además, se añadirá más información que pueda ser relevante para su desarrollo.

Resumen del plan de desarrollo

El plan de desarrollo de software se puede dividir en las siguientes secciones:

- **Visión general del proyecto:** se subdivide en las siguientes subsecciones:
 - Objetivos del proyecto.
 - Suposiciones y/o restricciones.
 - Características del proyecto.
 - Elección de Scrum como modelo de desarrollo.
 - Elección del tipo de desarrollo a utilizar.
 - Entregables del proyecto.
- **Organización del proyecto:** incluye la estructura del proyecto, los roles en el mismo y las reuniones que se han llevado a cabo.
- **Estimaciones del proyecto:** se muestran tanto las estimaciones temporales como de costes del proyecto, así como las desviaciones del mismo.

3.2. Visión general del proyecto

3.2.1. Objetivos

El objetivo principal de este proyecto consiste en el desarrollo de una plataforma android que permita realizar la gestión de eventos y pistas deportivas desde el smartphone. De esta manera, los organizadores publicarán sus pistas libres y las pondrán a disposición de los jugadores. Por su parte, los jugadores podrán unirse a otros jugadores para completar un partido y practicar un deporte de forma rápida y eficiente.

Puesto que el desarrollo corresponde a la realización de dos aplicaciones, a continuación se muestra brevemente la funcionalidad de cada una de ellas:

- Aplicación de gestión para los clubes. A través de esta aplicación, los organizadores y responsables del club podrán crear partidos y gestionarlos. Además, podrán actualizar su perfil de club para que los jugadores puedan encontrar las instalaciones o ponerse en contacto sin ningún tipo de inconveniente.
- Aplicación para los jugadores. A través de esta aplicación, los usuarios podrán encontrar partidos a los que apuntarse. Para ello dispondrán de diferentes filtros: Deporte, día, o precio. Los usuarios dispondrán de un perfil de jugador que podrán actualizar y que se irá completando con los partidos que va disputando.

3.2.2. Suposiciones y/o restricciones

El proyecto estará limitado por las siguientes restricciones:

1. Restricciones de presupuesto:

- Se utilizarán servicios y/o bibliotecas de libre distribución o servicios que permitan su uso de forma gratuita hasta cierta cuota.

2. Restricciones de recursos:

- El equipo de trabajo está compuesto únicamente por dos integrantes: tutor y alumno.
- El plazo de entrega mínimo dependerá de la terminación del resto de asignaturas, y el máximo de las convocatorias de trabajo de Fin de Grado. Ordinaria: 26/06/2019 y Extraordinaria: 10/07/2019

3. Restricciones de aplicación:

- Las aplicaciones serán desarrolladas para el sistema operativo Android con una versión mínima de API 16. Esto permite una compatibilidad con más del 95 % de los dispositivos móviles actuales[13].
- Las aplicaciones deben actualizar datos en tiempo real para la correcta sincronización entre distintos tipos de usuarios.

3.2.3. Características del proyecto

A continuación se muestran las características generales de este proyecto:

- Proyecto de una duración estimada de 300 horas.
- Desarrollado por el alumno únicamente, y supervisado por el tutor del proyecto, que guiará y propondrá cambios y mejoras en el proyecto.
- Entrega de documentación detallada y completa de todo el proceso, incluyendo diagramas, diseños, planificación y manual de usuario.

- Requisitos susceptibles a cambios, ampliaciones o reducciones dependiendo de prioridad, tiempo etc.
- Se requieren actualizaciones en tiempo real para el correcto funcionamiento de la aplicación.
- La gestión de riesgos se antoja importante debido a que no se tiene familiaridad con métodos de autenticación y servicios de actualización en tiempo real.

3.2.4. Elección de la metodología de desarrollo de software

En la actualidad se utilizan diferentes metodologías para el desarrollo de software desde el desarrollo en cascada hasta las metodologías ágiles, incluyendo proceso unificado, o programación extrema.

Para este proyecto se seleccionó el modelo de Scrum para realizar el desarrollo, principalmente por los siguientes factores:

- **Simplicidad:** el proyecto cuenta con un equipo limitado a una persona, el alumno, y a un tiempo limitado de 300 horas.
- **Rapidez:** focalizar el proyecto a disponer de un producto mínimo viable y añadir valor y funcionalidad en cada iteración se antojaba como la mejor opción considerando la limitación del tiempo.
- **Flexibilidad:** el producto puede evolucionar y sufrir cambios en los requisitos o funcionalidad. Es por ello, que se eligió Scrum para la realización del proyecto teniendo en cuenta que se puede adaptar en cada iteración.
- **Retroalimentación:** al crear un producto en varias iteraciones, se obtiene retroalimentación del tutor en cada una de las etapas.

Scrum es un método de desarrollo ágil, y como tal hay que *valorar más el software que funciona a la documentación exhaustiva*. Pero dadas las características del proyecto y las condiciones de entrega de un Trabajo de Fin de Grado, es necesaria la entrega de una documentación completa. Scrum permite esto añadiendo en cada *sprint* la realización de la documentación correspondiente. De esta forma, en la última iteración se dispondrá de una documentación completa.

3.2.5. Entregables del proyecto

Como se ha mencionado en el punto anterior, los entregables estarán sujetos a cambios a lo largo del desarrollo para obtener así una versión final al término de la última iteración.

Los artefactos entregables serán los siguientes:

- Plan de Desarrollo de Software.
- Seguimiento del proyecto.
- Plan de Gestión de Riesgos.
- Especificación de requisitos.
- Modelo de análisis.
- Modelo de diseño y arquitectura.
- Pruebas.
- Versión final del producto.
- Manual de usuario.
- Manual de instalación

3.3. Organización del proyecto

3.3.1. Roles del proyecto

Los roles del proyecto, atendiendo al modelo de Scrum [14] son, principalmente, los siguientes:

- **Product Owner:** es la persona que se encarga de definir los objetivos del proyecto. Participa activamente en el desarrollo para planificar y priorizar las historias de usuario.
- **Scrum Master:** es el principal valedor para que se cumpla la metodología ágil y se encarga de reducir los posibles impedimentos y no afecten al equipo de desarrollo.
- **Equipo de desarrollo:** grupo de personas con los conocimientos técnicos necesarios para desarrollar el proyecto. Se encargarán de llevar a cabo las historias en cada *sprint* y de la calidad del software a desarrollar.

Para este proyecto, la asignación de roles se ha realizado de la siguiente manera:

Nombre	Rol
Joaquín Adiego Rodríguez	Product Owner
Abel García de Prada	Scrum Master y Equipo de desarrollo

Tabla 3.1: Asignación de roles del proyecto

3.3.2. Reuniones, eventos y planes operativos

Para el seguimiento de Scrum se realizaron los eventos[14] mostrados en la Tabla 3.2.

Eventos	Descripción
Reunión inicial	Reunión inicial en la que se plantea la idea y la planificación del proyecto.
Sprint Planning	Reunión durante la cual se determinan las historias a completar en ese sprint. Pueden ser nuevas, o del anterior sprint que no se completaron o se requieren cambios.
Reunión diaria	El equipo de desarrollo (el alumno) identifica las tareas realizadas el día anterior y las tareas a desarrollar en el día de trabajo.
Demo	Reunión al final de cada <i>sprint</i> para mostrar el trabajo realizado al Product Owner (el tutor).
Retrospectiva	Se analiza lo que se ha realizado correctamente en el último <i>sprint</i> y lo que se puede mejorar de cara al siguiente <i>sprint</i> .

Tabla 3.2: Eventos de Scrum

3.4. Estimaciones del proyecto

3.4.1. Estimación temporal

Para el desarrollo del proyecto, se establecieron unas fechas de inicio y final aproximadas y se dividió en diferentes *sprints*.

La duración de los *sprints* se definió de 3 semanas debido a que la realización del mismo solo cuenta con una persona en el equipo de desarrollo y se consideró que esta duración sería suficiente para aportar valor en cada iteración. Para finalizar, se establecería un *sprint* de dos semanas para terminar los últimos detalles de la aplicación y documentación.

A continuación, en la Tabla 3.3 se muestran las fechas estimadas de cada iteración.

Sprint	Fecha de inicio	Fecha de fin
1	04/02/2019	24/02/2019
2	25/02/2019	17/03/2019
3	18/03/2019	07/04/2019
4	08/04/2019	28/04/2019
5	29/04/2019	19/05/2019
6	20/05/2019	09/06/2019
7	10/06/2019	23/06/2019

Tabla 3.3: Iteraciones del proyecto

Durante estas iteraciones, considerando como días de trabajo los días laborales (de lunes a viernes) y estimando un trabajo diario de 3h persona/día, se puede estimar el trabajo total como vemos a continuación:

- 3 horas persona/día x 5 días/semana x 3 semanas/sprint x 6 sprints de tres semanas = 270 horas/persona
- 3 horas persona/día x 5 días/semana x 2 semanas/sprint x 1 sprint de dos semanas = 30 horas/persona

Esto hace un total de 300 horas/persona que se ciñe al tiempo estimado para la realización del Trabajo de Fin de Grado.

3.4.2. Estimación de costes

La estimación de costes no correspondería a un proyecto real, puesto que del trabajo de desarrollo se encargará un alumno universitario y es un Proyecto de Fin de Grado, por lo que no se percibirá ninguna remuneración. Aún así, según la estimación de 300 horas/hombre y estimando el coste de un Ingeniero Informático en 15 euros/hora, el coste del desarrollo sería de 4500 euros.

A esta estimación habría que añadir los costes derivados por el uso de herramientas y/o tecnologías. En este caso, el coste de usar Firebase es nulo debido a que el plan gratuito Spark [15] será más que suficiente para el desarrollo del proyecto. Además, las otras herramientas utilizadas disponen de un plan gratuito o son de código abierto.

3.4.3. Desviación real del proyecto

El proyecto sufrió alguna desviación de lo estimado en la sección 3.4.1, debido principalmente a la inexperiencia del equipo en el ámbito de las herramientas de Firebase. Esta inexperiencia propició la necesidad de realizar algunas pruebas adicionales para poder trabajar con soltura con la plataforma, lo que conllevó un mayor tiempo para el desarrollo del proyecto.

Además, los sprints 6 y 7 se alargaron unos días más de lo estimado debido a la necesidad de preparar otras asignaturas para poder presentar el TFG en el actual curso académico. Teniendo en cuenta estos retrasos, en general, las estimaciones han sido correctas, y se ha completado el proyecto dentro de las fechas límites establecidas.

Capítulo 4

Seguimiento del proyecto

4.1. Introducción

En este capítulo se comentarán los aspectos fundamentales del seguimiento del proyecto, desde las fases iniciales, hasta la finalización completa de la aplicación.

Como se ha comentado anteriormente, para el desarrollo de la aplicación se ha seguido el modelo de Scrum. En la figura 4.1 (Fuente: [16]) se muestra gráficamente el ciclo de esta metodología presentada previamente en la sección 3.3.2.

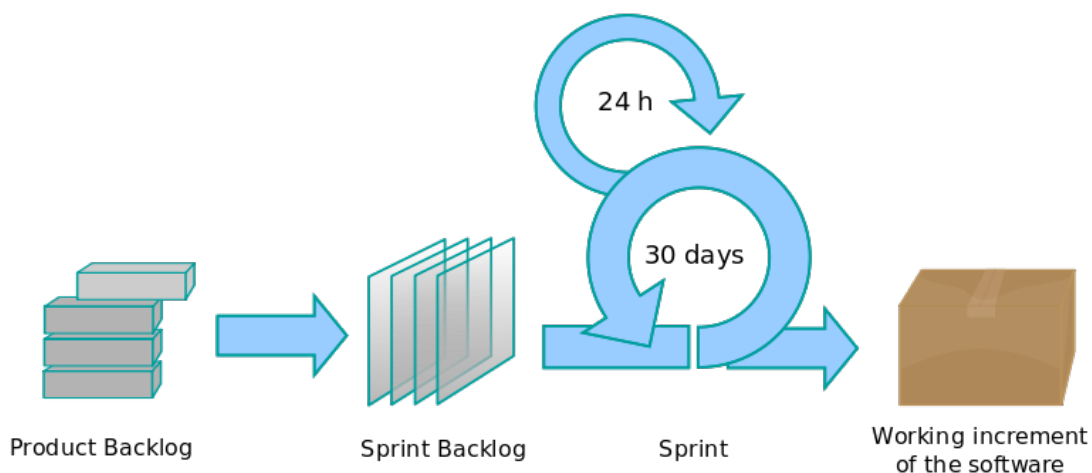


Figura 4.1: Flujo de trabajo de Scrum

En la imagen anterior se pueden diferenciar las diferentes etapas en este desarrollo. A continuación se describirán brevemente cada una de ellas:

1. Inicialmente se forma el Product Backlog o Pila de Producto. Se trata básicamente de la pila de requisitos desde la visión del usuario final. Se conforma con la funcionalidad o historias de usuario de todo el producto ordenadas por prioridad por el cliente.
2. En el sprint planning se seleccionan las historias de usuario con mayor prioridad, de tal forma que la carga de trabajo sea asumible por el equipo de desarrollo.
3. Cuando ya se tienen las historias de usuario seleccionadas, el equipo de desarrollo, se desglosan y dividen las historias en tareas a realizar. Esta lista de tareas formará el Sprint Backlog.
4. Durante la duración del Sprint, en este caso se estableció en tres semanas, se trabaja sobre esta pila de sprint.

5. Al comienzo de cada jornada de trabajo, se realiza una breve reunión para comprobar lo realizado durante el día anterior y lo que se va a realizar en el actual.
6. Al final del sprint, se obtiene una parte funcional del producto, y se examina lo realizado durante el sprint, detectando problemas y proponiendo soluciones para que no se repitan en los próximos sprints.

4.2. Sprint 1: 4 de Febrero - 24 de Febrero

En este Sprint se comienza con la documentación y el desarrollo de las aplicaciones tanto de Jugadores como de Clubes. Además se esboza el primer modelo de dominio de la aplicación. Este Sprint lo forman las historias de usuario de la Tabla 4.1.

ID	Descripción
US01	Como Equipo quiero tener configurado el entorno de trabajo completo para poder empezar a trabajar en el proyecto
US23	Como Product Owner quiero tener una documentación inicial del proyecto para tener una visión global del mismo
US02	Como usuario Jugador quiero poder registrarme en la aplicación mediante email y contraseña.
US03	Como usuario Jugador quiero poder identificarme de forma única en la aplicación de Jugadores con mis credenciales de usuario.
US04	Como usuario Club quiero poder identificarme de forma única en la aplicación de Clubes con mis credenciales de usuario.
US24	Como Product Owner quiero tener un modelo de dominio inicial de ambas aplicaciones a desarrollar.

Tabla 4.1: Historias de usuario del Sprint 1

4.2.1. US01 - Como Equipo quiero tener configurado el entorno de trabajo completo para poder empezar a trabajar en el proyecto

Descripción

En esta historia de usuario se configurará el entorno de trabajo completo que permita empezar el desarrollo del mismo. Es lo básico para empezar a trabajar sin ningún tipo de inconveniente que podría conllevar el retraso en la entrega del proyecto.

Estimación

Esta historia de usuario se ha valorado con 2 puntos, debido a la necesidad de configurar las herramientas que se emplearán para desarrollar el proyecto.

Tareas

- Crear el proyecto en Overleaf para comenzar la documentación.
- Configurar el proyecto en Overleaf y la estructura de la documentación.
- Instalación y configuración de Android Studio en su última versión.
- Creación del repositorio en Github que permita la gestión del control de versiones.
- Creación del proyecto Android en Android Studio.
- Enlazar Android Studio con Github.

- Crear Proyecto en la consola de Firebase.
- Enlazar Firebase con la aplicación Android.

4.2.2. US23 - Como Product Owner quiero tener una documentación inicial del proyecto para tener una visión global del mismo

Descripción

En esta historia de usuario se desarrollará la documentación inicial básica para iniciar el proyecto. Esto incluirá la primera versión de los requisitos, de la gestión de riesgos y del plan de desarrollo de software.

Estimación

Esta historia de usuario se ha valorado con una estimación de 3 puntos debido a la necesidad de comenzar la documentación de la gestión de riesgos, el plan de desarrollo de software y los requisitos iniciales.

Tareas

- Comienzo del Plan de Desarrollo de Software.
- Iniciación del Plan de Gestión de Riesgos.
- Documentación inicial de requisitos funcionales, no funcionales y de información.

4.2.3. US02 - Como usuario Jugador quiero poder registrarme en la aplicación mediante email y contraseña.

Descripción

En esta historia de usuario se desarrolla una pantalla de registro y se permite la creación de nuevos usuarios en la aplicación.

Estimación

Debido a la experiencia limitada del servicio de Firebase Authentication para la creación de nuevos usuarios, se ha estimado con 5 puntos esta historia.

Tareas

- Creación de una interfaz para el registro de usuarios.
- Gestionar la comunicación con Firebase Authentication.
- Gestionar los posibles errores que puedan surgir de esta comunicación.

4.2.4. US03 - Como usuario Jugador quiero poder identificarme de forma única en la aplicación de jugadores con mis credenciales.

Descripción

En esta historia de usuario se permite la identificación de los usuarios. Una funcionalidad básica y fundamental que permita a los jugadores acceder a la aplicación con las credenciales identificativas.

Estimación

En esta historia se requiere la creación de una interfaz de Login, y gestionar los usuarios a través de Firebase Authentication, así como los errores que puedan surgir. Por ello, se ha estimado con 5 puntos esta historia.

Tareas

- Creación de una interfaz de usuario para la identificación de usuarios.
- Gestionar la comunicación con Firebase Authentication.
- Gestionar los posibles errores que puedan surgir de esta comunicación.

4.2.5. US04 - Como usuario Club quiero poder identificarme de forma única en la aplicación de Clubes con mis credenciales.

Descripción

Análoga a la historia de usuario US03, se requiere la identificación única en la aplicación para clubes. De esta forma, solo los usuarios con credenciales de club podrán acceder a esta aplicación.

Estimación

Puesto que esta historia de usuario comparte muchas de las tareas con la historia de usuario US03 y se podría reutilizar la mayor parte de ella, la estimación asignada es de 3 puntos.

Tareas

- Creación de una interfaz de usuario para la identificación de usuarios.
- Gestionar la comunicación con Firebase Authentication.
- Gestionar los posibles errores que puedan surgir de esta comunicación.

4.2.6. US24 - Como Product Owner quiero tener un modelo de dominio inicial de ambas aplicaciones a desarrollar.

Descripción

En esta historia se desarrollará el modelo de dominio inicial de la aplicación que permita tener una visión global que se irá adaptando y evolucionando a lo largo del proyecto.

Estimación

Debido a que se trata de un modelo de dominio inicial, y nos encontramos en fases iniciales del proyecto, esta historia de usuario no se antoja muy complicada. Se ha estimado con 2 puntos.

Tareas

- Estudio para identificar una primera versión del modelo de dominio de la aplicación.
- Desarrollo de la documentación pertinente.

4.3. Sprint 2: 25 de Febrero - 17 de Marzo

En este Sprint se continúa con el desarrollo de las aplicaciones tanto de Jugadores como de Clubes, y se comienzan a desarrollar los perfiles de usuario y de club. Este Sprint lo forman las historias de usuario de la Tabla 4.2.

ID	Descripción
US05	Como usuario Jugador quiero poder crear y editar mi perfil.
US06	Como usuario Jugador quiero poder añadir una imagen de perfil.
US07	Como usuario Club quiero poder crear y editar mi perfil.
US08	Como usuario Club quiero poder añadir el logo de mi club a mi perfil.

Tabla 4.2: Historias de usuario del Sprint 2

4.3.1. US05 - Como usuario Jugador quiero poder crear y editar mi perfil.

Descripción

El perfil de usuario permitirá mantener un perfil de jugador actualizado que será visible desde otras funcionalidades. Además, se establecerá la conexión con Firestore para almacenar esta información.

Estimación

Se ha estimado con 5 puntos debido a la necesidad de crear un layout e iniciar una comunicación con Firebase Firestore para almacenar dicho perfil de usuario.

Tareas

- Creación de una interfaz de usuario para recoger los datos del usuario.
- Gestionar la comunicación con Firebase Firestore.
- Gestionar los posibles errores.

4.3.2. US06 - Como usuario Jugador quiero poder añadir una imagen de perfil.

Descripción

A través de esta funcionalidad se permitirá acceder al almacenamiento del dispositivo y seleccionar una imagen de perfil que permitirá una mejor identificación del jugador.

Estimación

Debido a la inexperiencia con Firebase Storage y los permisos de usuario, se ha valorado esta historia con 5 puntos.

Tareas

- Implementación de un selector de imágenes para poder acceder a nuestra galería.
- Mostrar solicitud de permisos de acceso a almacenamiento.
- Gestionar la comunicación con Firebase Storage.
- Gestionar los posibles errores.

4.3.3. US07 - Como usuario Club quiero poder crear y editar mi perfil.

Descripción

Análoga a la historia de usuario US06, el club también mantendrá un perfil actualizado que permita localizar y mostrar detalles de contacto.

Estimación

Puesto que las tareas son similares a las de la US06, y se puede reutilizar y adaptar esa funcionalidad, se ha estimado esta historia en 3 puntos.

Tareas

- Creación de una interfaz de usuario para recoger los datos del club.
- Gestionar la comunicación con Firebase Firestore.
- Gestionar los posibles errores.

4.3.4. US08 - Como usuario Club quiero poder añadir el logo de mi club a mi perfil.

Descripción

Análoga a la historia de usuario US07, el club también podrá seleccionar una imagen de su dispositivo y colocarla en el perfil para una mejor identificación del club.

Estimación

Puesto que las tareas son similares a las de la US07, y se puede reutilizar y adaptar esa funcionalidad, se ha estimado esta historia en 3 puntos.

Tareas

- Implementación de un selector de imágenes para poder acceder a nuestra galería.
- Mostrar solicitud de permisos de acceso a almacenamiento.
- Gestionar la comunicación con Firebase Storage.
- Gestionar los posibles errores.

4.4. Sprint 3: 18 de Marzo - 7 de Abril

En este Sprint se continúa con el desarrollo de las aplicaciones tanto de Jugadores como de Clubes, se crean los menús de navegación y se crean las visualizaciones de los perfiles. Este Sprint lo forman las historias de usuario de la Tabla 4.3.

ID	Descripción
US09	Como usuario Jugador quiero poder visualizar mi perfil de usuario.
US10	Como usuario Jugador quiero poder cerrar mi sesión.
US11	Como usuario Club quiero poder visualizar mi perfil de club.
US12	Como usuario Club quiero poder cerrar mi sesión.
US25	Como Product Owner quiero tener una primera versión de la arquitectura y diseño del proyecto.

Tabla 4.3: Historias de usuario del Sprint 3

4.4.1. US09 - Como usuario Jugador quiero poder visualizar mi perfil de usuario.

Descripción

Con esta funcionalidad se permitirá al jugador tener un acceso rápido a su perfil y visualizar la información relacionada.

Estimación

Esta historia de usuario se ha estimado con 5 puntos debido a la necesidad de crear una interfaz y poblarla con los datos recuperados de Firestore.

Tareas

- Creación de una interfaz de usuario para mostrar el perfil.
- Gestionar la comunicación con Firebase Firestore.
- Gestionar los posibles errores.

4.4.2. US10 - Como usuario Jugador quiero poder cerrar mi sesión.

Descripción

Como en cualquier aplicación con sesiones de usuario, es importante disponer de la opción de cerrar sesión. De esta forma, el usuario aparecerá como no logeado y tendrá que volver a iniciar sesión para disponer de la funcionalidad completa de la aplicación.

Estimación

Se ha estimado esta historia con 2 puntos puesto que se trata de gestionar la cuenta en Firebase Authentication y su implementación se antoja simple a nivel de interfaz.

Tareas

- Añadir la opción de Cerrar Sesión en la interfaz de usuario.
- Gestionar la comunicación con Firebase Authentication.
- Gestionar los posibles errores.

4.4.3. US11 - Como usuario Club quiero poder visualizar mi perfil de club.

Descripción

De forma análoga a la historia de usuario US09, el club también podrá ver su perfil.

Estimación

Puesto que las tareas son similares a las de la US09, y ya se tiene cierta experiencia, se ha estimado esta historia en 3 puntos.

Tareas

- Creación de una interfaz de usuario para mostrar el perfil.
- Gestionar la comunicación con Firebase Firestore.
- Gestionar los posibles errores.

4.4.4. US12 - Como usuario Club quiero poder cerrar mi sesión.

Descripción

De forma análoga a la historia US10, el club también podrá cerrar su sesión.

Estimación

Se estima con 2 puntos, puesto que cuenta con las mismas tareas que la US10.

Tareas

- Añadir la opción de Cerrar Sesión en la interfaz de usuario.
- Gestionar la comunicación con Firebase Authentication.
- Gestionar los posibles errores.

4.4.5. US25 - Como Product Owner quiero tener una primera versión de la arquitectura y diseño del proyecto.

Descripción

Esta historia de usuario permitirá disponer de una primera versión de la documentación con la arquitectura elegida, los motivos, las características y el cómo se ha implementado en el proyecto.

Estimación

Esta historia de usuario se ha estimado con 3 puntos debido a la necesidad de documentar el por qué se ha elegido esta arquitectura y sus características.

Tareas

- Documentar la arquitectura elegida.
- Documentar cómo se ha implementado esta arquitectura.

4.5. Sprint 4: 8 de Abril - 28 de Abril

En este Sprint se continúa con el desarrollo de las aplicaciones tanto de Jugadores como de Clubes, y se gestiona la creación de nuevos partidos y los listados de los mismos. Este Sprint lo forman las historias de usuario de la Tabla 4.4.

ID	Descripción
US13	Como usuario Club quiero poder crear nuevos partidos para que los jugadores puedan visualizar y acceder a ellos.
US14	Como usuario Jugador quiero poder ver un listado de los partidos disponibles para unirme.
US15	Como usuario Club quiero poder ver un listado completo de todos los partidos que he creado para tener un seguimiento en tiempo real.

Tabla 4.4: Historias de usuario del Sprint 4

4.5.1. US13 - Como usuario Club quiero poder crear nuevos partidos para que los jugadores puedan visualizar y acceder a ellos.

Descripción

Esta funcionalidad es una de las imprescindibles y más importantes de la aplicación porque es la que permitirá a los clubes crear los partidos a los que los jugadores podrán unirse posteriormente.

Estimación

Se ha estimado esta historia con 5 puntos puesto que hay que crear una interfaz de usuario y gestionar las comunicaciones con Firestore.

Tareas

- Creación de una interfaz de usuario que permita recoger los datos del partido.
- Gestionar la comunicación con Firebase Firestore.
- Gestionar los posibles errores.

4.5.2. US14 - Como usuario Jugador quiero poder ver un listado de los partidos disponibles para unirme.

Descripción

Esta historia de usuario permitirá a los jugadores ver un listado completo de los partidos disponibles creados por los clubes correspondientes.

Estimación

Esta historia de usuario se ha estimado con 8 puntos debido a la necesidad de observar en tiempo real los posibles cambios en los partidos. Además, hay que gestionar la paginación para no cargar todos los partidos a la vez, lo que podría consumir demasiados recursos si hay muchos partidos en la aplicación. Y por supuesto hay que crear un layout para esta información.

Tareas

- Creación de un layout para los partidos.
- Gestionar la comunicación con Firebase Firestore.
- Gestionar la actualización en tiempo real del listado.
- Gestionar la paginación.
- Gestionar los posibles errores.

4.5.3. US15 - Como usuario Club quiero poder ver un listado completo de todos los partidos que he creado para tener un seguimiento en tiempo real.

Descripción

Al igual que los jugadores necesitan ver los partidos disponibles, los clubes necesitan mantener un listado de todos los partidos que han creado para gestionar esos partidos de forma eficiente y rápida.

Estimación

Se ha estimado esta historia con 3 puntos debido a la similitud con la US12 y la posibilidad de reutilizar el código implementado.

Tareas

- Creación de un layout para los partidos.
- Gestionar la comunicación con Firebase Firestore.
- Gestionar la actualización en tiempo real del listado.
- Gestionar la paginación.
- Gestionar los posibles errores.

4.6. Sprint 5: 29 de Abril - 19 de Mayo

En este Sprint se continúa con el desarrollo de las aplicaciones tanto de Jugadores como de Clubes, y se gestiona la información completa de cada partido. Este Sprint lo forman las historias de usuario de la Tabla 4.5.

ID	Descripción
US16	Como usuario quiero consultar la información completa de un partido.
US17	Como usuario quiero consultar el listado de jugadores que se han inscrito a un partido.
US18	Como usuario Jugador quiero consultar el perfil del club que ha organizado el partido.

Tabla 4.5: Historias de usuario del Sprint 5

4.6.1. US16 - Como usuario quiero consultar la información completa de un partido.

Descripción

Esta funcionalidad trata de mostrar toda la información relacionada con un partido, de tal forma que se tenga un visión más completa que el mostrado en los listados.

Estimación

Se ha estimado con 5 puntos puesto que se necesita la creación de un layout, y las consultas a la base de datos.

Tareas

- Creación de layout con la información completa del partido.
- Gestionar la comunicación con Firestore.
- Gestionar los posibles errores.

4.6.2. US17 - Como usuario quiero consultar el listado de jugadores que se han inscrito a un partido.

Descripción

Esta funcionalidad permitirá visualizar un listado de todos los jugadores que se han apuntado a un partido con su imagen y nombre.

Estimación

Se ha estimado con 5 puntos puesto que hay que crear un layout para los jugadores y gestionar el listado de los mismos.

Tareas

- Creación de un layout para los jugadores.
- Gestionar la comunicación con Firebase Firestore.
- Gestionar los posibles errores.

4.6.3. US18 - Como usuario Jugador quiero consultar el perfil del club que ha organizado el partido.

Descripción

Como jugador se necesita poder consultar el perfil del club que ha creado el partido. De esta forma, el jugador podrá acceder a la información de contacto si así lo necesitase, y a la localización de las instalaciones donde se va a disputar el partido.

Estimación

Se ha estimado con 3 puntos esta historia de usuario debido a que se puede reutilizar parte de la implementación de la US12.

Tareas

- Creación de una interfaz de usuario para mostrar el perfil.
- Gestionar la comunicación con Firebase Firestore.
- Gestionar los posibles errores.

4.7. Sprint 6: 20 de Mayo - 9 de Junio

En este Sprint se continúa con el desarrollo de las aplicaciones tanto de Jugadores como de Clubes, se implementan varios filtros en los listados y se añaden las uniones y abandonos a los partidos. En principio sería el final del desarrollo de las aplicaciones. Este Sprint lo forman las historias de usuario de la Tabla 4.6.

ID	Descripción
US19	Como usuario Jugador quiero filtrar la lista de partidos según diferentes parámetros.
US20	Como usuario Club quiero filtrar la lista de mis partidos según el estado de los mismos.
US21	Como usuario Jugador quiero poder unirme y abandonar un partido.
US22	Como usuario Club quiero poder actualizar el estado de un partido.

Tabla 4.6: Historias de usuario del Sprint 6

4.7.1. US19 - Como usuario Jugador quiero filtrar la lista de partidos según diferentes parámetros.

Descripción

A nivel de experiencia de usuario, poder filtrar los partidos dependiendo de ciertos parámetros es fundamental.

Los filtros permitidos serán en función de partidos a los que nos hemos unido previamente, por deporte, fecha o precio. Se podrán marcar uno o varios al mismo tiempo y se actualizará el listado en función.

Estimación

Esta historia de usuario tendrá una estimación de 3 puntos debido a que el filtrado no tiene demasiada complejidad.

Tareas

- Creación de una interfaz de usuario que permita seleccionar los filtros.
- Gestionar la comunicación con Firebase Firestore.
- Gestionar los posibles errores.

4.7.2. US20 - Como usuario Club quiero filtrar la lista de mis partidos según el estado de los mismos.

Descripción

Al igual que en la aplicación de jugadores, se antoja importante filtrar los partidos de la aplicación de Clubes. De esta manera, la gestión de los partidos será más rápida porque estarán filtrados dependiendo del estado de los mismos.

Estimación

Se ha estimado esta historia de usuario con 3 puntos.

Tareas

- Creación de una interfaz de usuario que permita seleccionar los filtros.
- Gestionar la comunicación con Firebase Firestore.
- Gestionar los posibles errores.

4.7.3. US21 - Como usuario Jugador quiero poder unirme y abandonar un partido.

Descripción

Esta funcionalidad es una de las mas importantes, puesto que el ciclo de un partido pasa por estas acciones. Esta unión deberá ser controlada, de tal forma que si el partido admite un máximo de 10 personas, no se permitirá la unión cuando el partido esté completo.

Estimación

Debido a la inexperiencia en el uso de transacciones en Firebase Firestore, a la necesidad de crear dos transacciones, una para unirse y otra para abandonar, se estimado esta historia en 5 puntos.

Tareas

- Creación de un layout para la unión y abandono a partidos.
- Creación de transacciones en Firebase Firestore.
- Gestionar los posibles errores.

4.7.4. US22 - Como usuario Club quiero poder actualizar el estado de un partido.

Descripción

Esta funcionalidad permitirá al club gestionar el ciclo de vida de un partido. Podrá cancelar el partido en cualquier momento, así como establecer y confirmar la convocatoria del mismo. De esta forma el club tendrá el control total de sus partidos.

Estimación

Se ha estimado con 3 puntos esta historia puesto que su complejidad no es muy alta.

Tareas

- Creación de un layout para la gestión de los partidos.
- Gestionar las comunicaciones con Firebase Firestore.
- Gestionar los posibles errores.

4.8. Sprint 7: 10 de Junio - 23 de Junio

En este Sprint se continúa con la documentación del proyecto, se terminaría la memoria y con ello se completaría el Trabajo de Fin de Grado. Este Sprint lo forman las historias de usuario de la Tabla 4.7.

ID	Descripción
US26	Como Product Owner quiero disponer del Plan de Desarrollo de Software final y su seguimiento.
US27	Como Product Owner quiero disponer de la memoria completa del proyecto.

Tabla 4.7: Historias de usuario del Sprint 7

4.8.1. US26 - Como Product Owner quiero disponer del Plan de Desarrollo de Software final y su seguimiento.

Descripción

En esta historia de usuario se deberá aportar la versión final del Plan de Desarrollo de Software y el Seguimiento realizado durante todo el desarrollo.

Estimación

Se ha estimado esta historia de usuario con 5 puntos.

Tareas

- Aportar la versión final del Desarrollo de Software.
- Aportar todo el seguimiento del proyecto.

4.8.2. US27 - Como Product Owner quiero disponer de la memoria completa del proyecto.

Descripción

En esta historia de usuario se deberá aportar una memoria completa del trabajo realizado.

Estimación

Esta historia de usuario se ha estimado con 5 puntos.

Tareas

- Realizar el manual de usuario.
- Revisar la memoria completa, aportar las versiones finales de Modelo de Dominio, el diseño final de la aplicación y las valoraciones personales obtenidas al desarrollar este trabajo.
- Corregir los posibles errores y las mejoras reportadas por el tutor.

Capítulo 5

Plan de Gestión de Riesgos

5.1. Introducción

La Gestión de Riesgos permite la resolución de cualquier situación adversa que pueda surgir a lo largo del desarrollo del proyecto. Por eso es importante definirlos en etapas iniciales del desarrollo con el fin de tener planes de acción en caso de que ocurran.

5.1.1. Gestión del riesgo

A continuación se identifican los riesgos y sus planes de acción:

R01	Falta de experiencia del alumno
Descripción	El alumno ha realizado prácticas y proyectos simples en Android, pero puede encontrar dificultades en el desarrollo de un proyecto complejo.
Consecuencia	Ralentización del desarrollo que puede afectar al resto del proyecto
Probabilidad	Media
Impacto	Bajo
Estrategia	Reservar el riesgo
Plan de acción	Consulta de documentación, tutoriales o consultar con un experto en la materia.
Plan de contingencia	Comprobación de conocimientos adquiridos antes de comenzar el desarrollo.
<hr/>	
R02	Mala planificación
Descripción	Debido a errores en la estimación, la planificación no se ciñe a la realidad y pueden producirse retrasos en las fechas de entrega.
Consecuencia	Retraso en la entrega de proyecto
Probabilidad	Alta
Impacto	Crítico
Estrategia	Reservar el riesgo
Plan de acción	Realizar una planificación que se ajuste a la realidad.
Plan de contingencia	Mantener un seguimiento exhaustivo durante el ciclo de desarrollo.

R03	Pérdida de artefactos
Descripción	Por cualquier motivo, se pierden parcial o completamente los artefactos de entrega.
Consecuencia	Ralentización alta del proyecto, conllevaría la repetición de los artefactos perdidos.
Probabilidad	Baja
Impacto	Crítico
Estrategia	Evitación del riesgo
Plan de acción	-
Plan de contingencia	Creación de copias de seguridad en documentación y uso de control de versiones para el desarrollo.
R04	Diseño pobre o incorrecto
Descripción	El diseño de la solución no es correcto o es pobre debido a la inexperiencia del equipo en la captación de requisitos.
Consecuencia	Ralentización del proyecto y actualizar el diseño en consecuencia
Probabilidad	Medio-Bajo
Impacto	Crítico
Estrategia	Reducción del riesgo
Plan de acción	Rehacer o actualizar el diseño de la solución.
Plan de contingencia	Consulta de cada diseño con el cliente para verificar la calidad del diseño.
R05	Disponibilidad de miembros del equipo
Descripción	Baja disponibilidad de algún miembro del equipo.
Consecuencia	Si el tiempo de disponibilidad es demasiado bajo puede acarrear la no consecución de los objetivos y entrega del proyecto.
Probabilidad	Baja
Impacto	Crítico
Estrategia	Reservar el riesgo
Plan de acción	Actualizar la planificación del proyecto para adecuarse a la disponibilidad de los miembros.
Plan de contingencia	Consulta de disponibilidad del equipo
R06	Mala estimación en los puntos de una historia de usuario
Descripción	Debido a errores en la estimación, la planificación no se ciñe a las fechas límite.
Consecuencia	Puede ocasionar retrasos en la planificación.
Probabilidad	Media-Alta
Impacto	Medio
Estrategia	Reducción del riesgo
Plan de acción	Replanificación del sprint.
Plan de contingencia	Realizar una pequeña investigación sobre las necesidades de cada historia antes de realizar la estimación de los puntos.

R07	Seguimiento de tareas limitado
Descripción	Debido a diversas causas, el seguimiento del proyecto no se realiza adecuadamente y se pierde la noción de qué tareas se han realizado, o que funcionalidad todavía queda por desarrollar.
Consecuencia	Puede ocasionar retrasos en la planificación.
Probabilidad	Baja
Impacto	Medio
Estrategia	Evitación del riesgo
Plan de acción	Revisión completa del proyecto para comprobar lo desarrollado y lo que no.
Plan de contingencia	Realizar un seguimiento de lo desarrollado y lo que falta por realizar apoyándose en aplicaciones externas y en la reunión diaria de Scrum.

5.1.2. Control de riesgos

Debido a la realización de un proyecto complejo, que comprende la realización de dos aplicaciones que deben sincronizarse en tiempo real, se han disparado algunos riesgos que comentaremos a continuación.

El riesgo R01 se ha disparado debido a la inexperiencia del alumno en algunas tecnologías empleadas, lo que ha conllevado emplear más tiempo del estimado inicialmente para poder realizar la entrega del proyecto en los plazos estimados.

El riesgo R05 se ha disparado debido a la falta de disponibilidad del alumno en periodos de exámenes, lo que ha acarreado el retraso en la entrega del proyecto, y la ampliación del tiempo estimado para la realización completa del proyecto.

El riesgo R06 se ha disparado debido a que algunas estimaciones eran inferiores al tiempo empleado finalmente, debido a la inexperiencia en algunas facetas del desarrollo, lo que ha implicado aumentar el tiempo dedicado a resolver este defecto.

Capítulo 6

Requisitos

6.1. Los requisitos en el desarrollo ágil

La especificación de requisitos en el desarrollo ágil difiere en varios puntos respecto al desarrollo tradicional. Es por ello, que se comentarán las principales diferencias a fin de clarificar los objetivos.

Al igual que en un desarrollo tradicional, los requisitos y funcionalidad deseada es definida por el cliente. Las principales diferencias se encuentran en quién, cómo y dónde se especifican los requisitos.

En un proyecto tradicional, los requisitos son recogidos por una persona o equipo especializado en captación de requisitos y los especifica en un documento invariable al comienzo del proyecto.

En cambio, en un desarrollo ágil, las necesidades del cliente son conocidas por todo el equipo, y se intenta guiar al cliente para adaptar y orientar el producto a realizar. Los requisitos en una metodología ágil se separan en historias de usuario que se agrupan en un *Backlog* ordenado por prioridad que irá evolucionando a lo largo del desarrollo.

Teniendo esto en cuenta, se obtienen los siguientes requisitos que han ido evolucionando a lo largo del desarrollo.

6.2. Requisitos funcionales

Debido a la creación de dos aplicaciones, se definirán los requisitos de cada una de ellas de manera separada. En la tabla 6.1 se mostrarán los requisitos funcionales acordes a la aplicación de los jugadores.

ID	Nombre	Descripción
RF01	Registrar usuarios	El sistema permitirá el registro de nuevos usuarios.
RF02	Iniciar sesión	El sistema permitirá iniciar sesión.
RF03	Cerrar sesión	El sistema permitirá cerrar sesión.
RF04	Visualizar perfil usuario	El sistema permitirá visualizar el perfil actual del usuario.
RF05	Editar perfil	El sistema permitirá editar el perfil del usuario.
RF06	Imagen de perfil	El sistema permitirá añadir una imagen de perfil al usuario.
FR07	Visualizar lista de partidos	El sistema permitirá ver la lista de próximos partidos.
FR08	Filtrar lista de partidos	El sistema permitirá aplicar diferentes filtros al listado de partidos.
FR09	Visualizar información de partido	El sistema permitirá ver la información completa del partido.
FR10	Inscribirse a un partido	El sistema permitirá a los usuarios inscribirse a los partidos.
FR11	Abandonar partido	El sistema permitirá a los usuarios abandonar partidos.
FR12	Visualizar jugadores	El sistema permitirá visualizar a los jugadores inscritos en un partido.
FR13	Visualizar organizador	El sistema permitirá ver el perfil del organizador del partido.

Tabla 6.1: Requisitos funcionales Cribi

En la Tabla 6.2 se mostrarán los requisitos funcionales de la aplicación para organizadores.

ID	Nombre	Descripción
RF14	Iniciar sesión II	El sistema permitirá iniciar sesión.
RF15	Cerrar sesión II	El sistema permitirá cerrar sesión.
RF16	Visualizar perfil organizador	El sistema permitirá visualizar el perfil actual del organizador.
RF17	Editar perfil II	El sistema permitirá editar el perfil del organizador.
RF18	Imagen de perfil II	El sistema permitirá añadir una imagen de perfil (logo) al organizador.
RF19	Creación de Partidos	El sistema permitirá al organizador crear nuevos partidos.
RF20	Visualizar lista de partidos II	El sistema permitirá visualizar el listado de partidos organizados por el propio organizador.
RF21	Visualizar jugadores II	El sistema permitirá visualizar el listado de jugadores inscritos a un partido.
RF22	Cambiar estado de un partido	El sistema permitirá cambiar el estado de un partido al organizador.

Tabla 6.2: Requisitos funcionales Cribi Clubs

6.3. Requisitos no funcionales

A continuación, en la Tabla 6.3 se presentan los requisitos no funcionales comunes a ambas aplicaciones.

ID	Nombre	Descripción
RNF01	Facilidad de instalación	La aplicación debe poder ser instalada por cualquier usuario a partir de un apk.
RNF02	SDK mínimo	La aplicación debe ser compatible con cualquier dispositivo con versión superior a la API 16 (98 % dispositivos).
RNF03	Solicitud de permisos	Para dispositivos con versiones superiores a la API 23.0, se solicitarán los permisos necesarios en tiempo de ejecución.
RNF04	Sistema de base de datos	El sistema utilizará Firebase Firestore como base de datos no relacional.
RNF05	Gestión de sesiones	El sistema utilizará Firebase Authentication como gestor de sesiones de usuario.
RNF06	Identificación mediante correo electrónico	El acceso será mediante correo electrónico y contraseña
RNF07	Unicidad de correo electrónico	El correo electrónico utilizado para la identificación ha de ser único.
RNF08	Identificador único de usuarios	Se identificará a los usuarios por un identificador único generado por Firebase.
RNF09	Almacenamiento de imágenes de perfil	El sistema utilizará Firebase Storage como almacenamiento de imágenes de perfil.
RNF010	Tiempo de respuesta	La aplicación debe garantizar unos tiempos de respuesta no superiores a 10 segundos.

Tabla 6.3: Requisitos no funcionales

6.4. Requisitos de información

Los requisitos de información del proyecto se presentan en la Tabla 6.4.

ID	Nombre	Descripción
RIN01	Contenido de un usuario	id, nombre, descripción, ciudad, número de partidos jugados, imagen de perfil
RIN02	Contenido de un organizador	id, nombre, imagen de perfil, ciudad, dirección, teléfono, dirección web, email
RIN03	Contenido de un partido	id, deporte, estado del partido, fecha del partido, fecha límite para reservar la pista, duración, número de jugadores inscritos, precio de la pista, id del organizador
RIN04	Contenido de un jugador	id del jugador, nombre, imagen de perfil, fecha de unión a partido

Tabla 6.4: Requisitos no funcionales de información

Capítulo 7

Análisis

En este capítulo se identifican los actores del sistema, los casos de uso y se describen cada uno de ellos. Además se añade el modelo de dominio, su descripción y los diagramas de actividad pertinentes. Todo esto permitirá esclarecer y presentar una descripción completa de la funcionalidad desarrollada a lo largo de este proyecto.

Como ya se ha comentado a lo largo de la memoria, el proyecto ha sido desarrollado con Scrum. Por lo tanto, al igual que los requisitos, los casos de uso han ido evolucionando a lo largo de todo el proceso. Además, al usar esta metodología, los requisitos y los casos de uso serán escritos como historias de usuario en el Plan de Seguimiento.

7.1. Modelo de Casos de Uso

7.1.1. Actores

Puesto que se trata de una aplicación para jugadores y otra para organizadores, tendremos dos actores diferentes que se describen a continuación.

- **Jugador:** representa al usuario que busca partidos para disputar.
- **Organizador:** representa al usuario organizador que dispone de alguna pista para ofrecer.

7.1.2. Diagrama de casos de uso

Actor Jugador

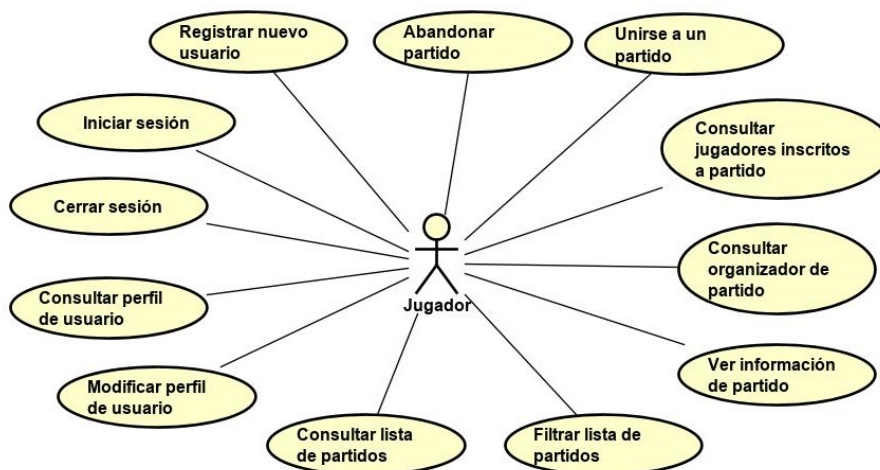


Figura 7.1: Diagrama de casos de uso actor Jugador

Actor Organizador



Figura 7.2: Diagrama de casos de uso actor Organizador

7.1.3. Especificación de casos de uso

Aplicación Cribi para jugadores

CU01	Iniciar sesión
Actor	Jugador
Descripción	El usuario desea identificarse en la aplicación
Precondición	El usuario no está identificado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el usuario selecciona 'Iniciar sesión'. 2. El sistema muestra la pantalla de identificación. 3. El usuario introduce las credenciales. 4. El sistema comprueba que las credenciales son correctas. 5. El sistema identifica al usuario y muestra su pantalla de inicio.
Postcondición	El usuario aparece identificado en el sistema.
Flujo alternativo	<ol style="list-style-type: none"> 2.a 3.a El usuario cancela y el caso de uso queda sin efecto. 4.a El sistema no dispone de conexión a internet para realizar la identificación. Se notifica y el caso de uso queda sin efecto. 4.b Las credenciales del usuario no son correctas. Se notifica y el caso de uso continúa en el paso 2.

Tabla 7.1: Descripción de CU01

CU02	Registrar nuevo usuario
Actor	Jugador
Descripción	El usuario desea registrarse en la aplicación.
Precondición	El usuario no está identificado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el usuario selecciona 'Registrarse'. 2. El sistema muestra la pantalla de registro. 3. El usuario introduce los datos para su registro. 4. El sistema comprueba los datos y crea un nuevo usuario.
Postcondición	Se crea una nueva cuenta de usuario.
Flujo alternativo	<ol style="list-style-type: none"> 3.a El usuario cancela y el caso de uso queda sin efecto. 4.a El sistema no dispone de conexión a internet para realizar el registro. Se notifica y el caso de uso continúa en el paso 2. 4.b Los datos introducidos por el usuario no son válidos para el registro. Se notifica y el caso de uso continúa en el paso 2.

Tabla 7.2: Descripción de CU02

CU03	Consultar perfil
Actor	Jugador
Descripción	El usuario desea consultar su perfil.
Precondición	El usuario está identificado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el usuario selecciona 'Consultar perfil'. 2. El sistema muestra el perfil del usuario.
Postcondición	Se muestra el perfil completo del usuario.
Flujo alternativo	<ol style="list-style-type: none"> 2.a El sistema no dispone de conexión a internet para realizar la consulta de perfil. Se notifica y el caso de uso queda sin efecto.

Tabla 7.3: Descripción de CU03

CU04	Modificar perfil de usuario
Actor	Jugador
Descripción	El usuario desea modificar su perfil.
Precondición	El usuario está identificado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el usuario selecciona 'Modificar perfil'. 2. El sistema muestra un formulario con los datos actualizables y sus valores actuales. 3. El usuario modifica los datos a su elección. 4. El sistema comprueba los datos y actualiza el perfil.
Postcondición	El perfil ha sido actualizado.
Flujo alternativo	<ol style="list-style-type: none"> 3.a El usuario cancela y el caso de uso queda sin efecto. 4.a El sistema no dispone de conexión a internet para actualizar los datos. Se notifica y el caso de uso queda sin efecto. 4.b Los datos introducidos por el usuario no son válidos. Se notifica y el caso de uso continúa en el paso 2.

Tabla 7.4: Descripción de CU04

CU05	Consultar lista de partidos
Actor	Jugador
Descripción	El usuario desea ver la lista de partidos disponibles.
Precondición	El usuario está identificado en el sistema.
Flujo normal	1. El caso de uso comienza cuando el usuario selecciona 'Mostrar partidos'. 2. El sistema muestra un listado con los partidos disponibles.
Postcondición	Se muestra un listado con los partidos disponibles.
Flujo alternativo	2.a El sistema no dispone de conexión a internet para actualizar los datos. Se notifica y el caso de uso queda sin efecto.

Tabla 7.5: Descripción de CU05

CU06	Filtrar listado de partidos
Actor	Jugador
Descripción	El usuario desea filtrar los partidos para seleccionar el que corresponda a sus intereses.
Precondición	El usuario está identificado en el sistema.
Flujo normal	1. El caso de uso comienza cuando el usuario selecciona 'Filtrar partidos'. 2. El sistema muestra diferentes filtros al usuario. 3. El usuario selecciona uno o varios filtros. 4. El sistema filtra la lista de partidos. 5. El sistema muestra un listado filtrado de los partidos
Postcondición	Se muestra un listado de partidos filtrados.
Flujo alternativo	3.a El usuario cancela. El caso de uso queda sin efecto.

Tabla 7.6: Descripción de CU06

CU07	Consultar información de partido
Actor	Jugador
Descripción	El usuario desea ver la información completa de un partido.
Precondición	El usuario está identificado en el sistema.
Flujo normal	1. El caso de uso comienza cuando el usuario selecciona un partido del listado ofrecido por 'Mostrar partidos'. 2. El sistema muestra la información completa del partido.
Postcondición	Se muestra la información completa del partido.
Flujo alternativo	Ninguno.

Tabla 7.7: Descripción de CU07

CU08	Consultar organizador de un partido
Actor	Jugador
Descripción	El usuario desea ver la información completa del organizador de un partido.
Precondición	El usuario está identificado en el sistema.
Flujo normal	1. El caso de uso comienza cuando el usuario selecciona 'Mostrar organizador' en la información de un partido. 2. El sistema muestra la información completa del organizador.
Postcondición	Se muestra la información completa del organizador del partido.
Flujo alternativo	2.a El sistema no dispone de conexión a internet. Se notifica, y el caso de uso queda sin efecto.

Tabla 7.8: Descripción de CU08

CU09	Consultar jugadores inscritos a un partido
Actor	Jugador y Organizador
Descripción	El usuario desea ver todos los jugadores inscritos a un partido.
Precondición	El usuario está identificado en el sistema.
Flujo normal	1. El caso de uso comienza cuando el usuario selecciona la opción de 'Mostrar jugadores inscritos' en la información de un partido. 2. El sistema muestra un listado completo de los jugadores inscritos.
Postcondición	Se muestra un listado con los jugadores inscritos.
Flujo alternativo	2.a El sistema no dispone de conexión a internet. Se notifica, y el caso de uso queda sin efecto.

Tabla 7.9: Descripción de CU09

CU10	Unirse a partido
Actor	Jugador
Descripción	El usuario desea unirse a un partido.
Precondición	El usuario está identificado en el sistema y no está inscrito al partido.
Flujo normal	1. El caso de uso comienza cuando el usuario selecciona 'Unirse a partido' en la información de un partido. 2. El sistema muestra información relativa al partido. 3. El usuario acepta dicha información. 4. El sistema comprueba si el jugador puede unirse al partido. 5. El sistema inscribe al jugador en el partido
Postcondición	El usuario aparece inscrito al partido.
Flujo alternativo	3.a El usuario rechaza la información. El caso de uso queda sin efecto. 4.a El sistema determina que el usuario no puede unirse al partido. Se notifica y el caso de uso queda sin efecto.

Tabla 7.10: Descripción de CU10

CU11	Abandonar partido
Actor	Jugador
Descripción	El usuario desea abandonar un partido.
Precondición	El usuario está identificado en el sistema y aparece como inscrito al partido.
Flujo normal	1. El caso de uso comienza cuando el usuario selecciona 'Abandonar partido' en la información de un partido. 2. El sistema muestra información relativa al partido. 3. El acepta dicha información. 4. El sistema elimina al jugador del partido
Postcondición	El usuario aparece como no inscrito al partido.
Flujo alternativo	3.a El usuario cancela. El caso de uso queda sin efecto.

Tabla 7.11: Descripción de CU11

CU12	Cerrar sesión
Actor	Jugador
Descripción	El usuario desea cerrar sesión.
Precondición	El usuario está identificado en el sistema.
Flujo normal	1. El caso de uso comienza cuando el usuario selecciona 'Cerrar sesión'. 2. El sistema cierra la sesión del usuario. 3. El sistema muestra la pantalla de identificación.
Postcondición	El usuario aparece como no identificado en el sistema.
Flujo alternativo	Ninguno.

Tabla 7.12: Descripción de CU11

Aplicación Cribi Clubs para organizadores

CU13	Iniciar sesión como organizador
Actor	Organizador
Descripción	El usuario desea identificarse en la aplicación
Precondición	El usuario no está identificado en el sistema.
Flujo normal	1. El caso de uso comienza cuando el usuario selecciona 'Iniciar sesión'. 2. El sistema muestra la pantalla de identificación. 3. El usuario introduce las credenciales. 4. El sistema comprueba que las credenciales son correctas. 5. El sistema identifica al usuario y muestra su pantalla de inicio.
Postcondición	El usuario aparece identificado en el sistema.
Flujo alternativo	2.a 3.a El usuario cancela y el caso de uso queda sin efecto. 4.a El sistema no dispone de conexión a internet para realizar la identificación. Se notifica y el caso de uso queda sin efecto. 4.b Las credenciales del usuario no son correctas. Se notifica y el caso de uso continúa en el paso 2.

Tabla 7.13: Descripción de CU13

CU14	Consultar perfil
Actor	Organizador
Descripción	El usuario desea consultar su perfil.
Precondición	El usuario está identificado en el sistema.
Flujo normal	1. El caso de uso comienza cuando el usuario selecciona 'Consultar perfil'. 2. El sistema muestra el perfil del usuario.
Postcondición	Se muestra el perfil completo del usuario.
Flujo alternativo	2.a El sistema no dispone de conexión a internet para realizar la consulta de perfil. Se notifica y el caso de uso queda sin efecto.

Tabla 7.14: Descripción de CU14

CU15	Modificar perfil de organizador
Actor	Organizador
Descripción	El usuario desea modificar su perfil.
Precondición	El usuario está identificado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el usuario selecciona 'Modificar perfil'. 2. El sistema muestra un formulario con los datos actualizables y sus valores actuales. 3. El usuario modifica los datos a su elección. 4. El sistema comprueba los datos y actualiza el perfil.
Postcondición	El perfil ha sido actualizado.
Flujo alternativo	<ol style="list-style-type: none"> 3.a El usuario cancela y el caso de uso queda sin efecto. 4.a El sistema no dispone de conexión a internet para actualizar los datos. Se notifica y el caso de uso queda sin efecto. 4.b Los datos introducidos por el usuario no son válidos. Se notifica y el caso de uso continúa en el paso 2.

Tabla 7.15: Descripción de CU15

CU16	Consultar lista de partidos organizados
Actor	Organizador
Descripción	El usuario desea ver la lista de partidos que ha organizado.
Precondición	El usuario está identificado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el usuario selecciona 'Mostrar partidos organizados'. 2. El sistema muestra un listado con los partidos que ha creado el propio organizador.
Postcondición	Se muestra un listado con los partidos organizados.
Flujo alternativo	2.a El sistema no dispone de conexión a internet para actualizar los datos. Se notifica y el caso de uso queda sin efecto.

Tabla 7.16: Descripción de CU16

CU17	Filtrar listado de partidos organizados
Actor	Organizador
Descripción	El usuario Organizador desea filtrar los partidos que ha organizado por estado de cada uno de ellos para acceder rápidamente a aquellos que tiene pendientes.
Precondición	El usuario está identificado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el usuario selecciona 'Filtrar partidos'. 2. El sistema muestra diferentes estados al usuario. 3. El usuario selecciona un estado. 4. El sistema filtra la lista de partidos. 5. El sistema muestra un listado con los partidos que se encuentran en ese estado.
Postcondición	Se muestra un listado de partidos con el estado elegido.
Flujo alternativo	3.a El usuario cancela. El caso de uso queda sin efecto.

Tabla 7.17: Descripción de CU17

CU18	Consultar información de partido
Actor	Organizador
Descripción	El usuario desea ver la información completa de un partido.
Precondición	El usuario está identificado en el sistema.
Flujo normal	1. El caso de uso comienza cuando el usuario selecciona un partido del listado ofrecido por 'Mostrar partidos'. 2. El sistema muestra la información completa del partido.
Postcondición	Se muestra la información completa del partido.
Flujo alternativo	Ninguno.

Tabla 7.18: Descripción de CU18

CU19	Consultar jugadores inscritos a un partido
Actor	Organizador
Descripción	El usuario desea ver todos los jugadores inscritos a un partido.
Precondición	El usuario está identificado en el sistema.
Flujo normal	1. El caso de uso comienza cuando el usuario selecciona la opción de 'Mostrar jugadores inscritos' en la información de un partido. 2. El sistema muestra un listado completo de los jugadores inscritos.
Postcondición	Se muestra un listado con los jugadores inscritos.
Flujo alternativo	2.a El sistema no dispone de conexión a internet. Se notifica, y el caso de uso queda sin efecto.

Tabla 7.19: Descripción de CU19

CU20	Cambiar estado de partido
Actor	Organizador
Descripción	El usuario desea cambiar el estado de un partido.
Precondición	El usuario está identificado en el sistema.
Flujo normal	1. El caso de uso comienza cuando el usuario selecciona 'Cambiar estado de partido' en la información de un partido. 2. El sistema muestra un listado de los posibles estados siguientes. 3. El usuario selecciona un estado. 4. El sistema cambia el estado del partido.
Postcondición	El estado del partido cambia al elegido por el organizador.
Flujo alternativo	3.a El usuario cancela y el caso de uso queda sin efecto 4.a El sistema no dispone de conexión a internet. Se notifica, y el caso de uso queda sin efecto.

Tabla 7.20: Descripción de CU20

CU21	Crear nuevo partido
Actor	Organizador
Descripción	El usuario desea crear un nuevo partido.
Precondición	El usuario está identificado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el usuario selecciona 'Crear nuevo partido'. 2. El sistema muestra un formulario con los datos requeridos para crear el partido. 3. El usuario introduce los datos. 4. El sistema comprueba los datos y crea el partido.
Postcondición	Se crea un nuevo partido.
Flujo alternativo	<ol style="list-style-type: none"> 3.a El usuario cancela y el caso de uso queda sin efecto. 4.a El sistema no dispone de conexión a internet. Se notifica, y el caso de uso queda sin efecto. 4.b Los datos introducidos no son válidos. Se notifica, y el caso de uso queda sin efecto.

Tabla 7.21: Descripción de CU21

CU22	Cerrar sesión de organizador
Actor	Organizador
Descripción	El usuario desea cerrar sesión.
Precondición	El usuario está identificado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el usuario selecciona 'Cerrar sesión'. 2. El sistema cierra la sesión del usuario. 3. El sistema muestra la pantalla de identificación.
Postcondición	El usuario aparece como no identificado en el sistema.
Flujo alternativo	Ninguno.

Tabla 7.22: Descripción de CU22

7.1.4. Modelo de Dominio

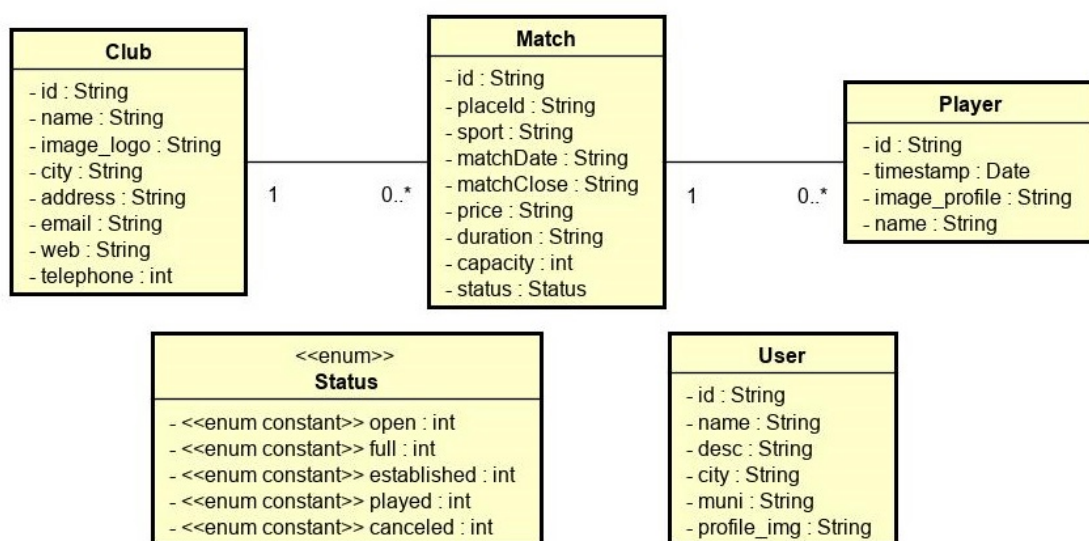


Figura 7.3: Modelo de dominio de la aplicación.

7.1.5. Descripción de las clases del modelo de dominio

User

- **Descripción:** clase que modela un usuario jugador.
- **Atributos:**
 - *id*: identificador único para cada usuario.
 - *name*: nombre del usuario.
 - *descr*: breve descripción del usuario.
 - *city*: ciudad del usuario.
 - *muni*: localidad del usuario.
 - *profile_img*: url de acceso a la imagen de perfil del usuario.

Player

- **Descripción:** clase que modela a un jugador inscrito en un partido.
- **Atributos:**
 - *id*: id del usuario.
 - *timestamp*: Fecha de inscripción en el partido.
 - *profile_img*: url de acceso a la imagen de perfil del usuario.
 - *name*: nombre del usuario.

Club

- **Descripción:** clase que modela a un organizador o club.
- **Atributos:**
 - *id*: identificador único de organizador.
 - *name*: nombre del club organizador.
 - *image_logo*: url de acceso a la imagen de perfil del organizador
 - *city*: ciudad en la que el organizador ofrece sus instalaciones.
 - *address*: dirección completa de las instalaciones.
 - *email*: correo electrónico de contacto.
 - *web*: dirección web de contacto.
 - *telephone*: teléfono de contacto.

Match

- **Descripción:** clase que modela un partido.
- **Atributos:**
 - *id*: identificador único de partido.
 - *sport*: deporte del partido a disputar.
 - *matchDate*: fecha de comienzo del partido.
 - *matchClose*: fecha límite para unirse a un partido.
 - *placeId*: identificador del club organizador donde que ofrece las instalaciones.
 - *price*: precio del partido.
 - *duration*: duración del partido.
 - *capacity*: número de jugadores necesarios para disputar el partido.
 - *status*: estado en el que se encuentra el partido.

7.1.6. Diagramas de actividad Cribi

CU01: Iniciar sesión

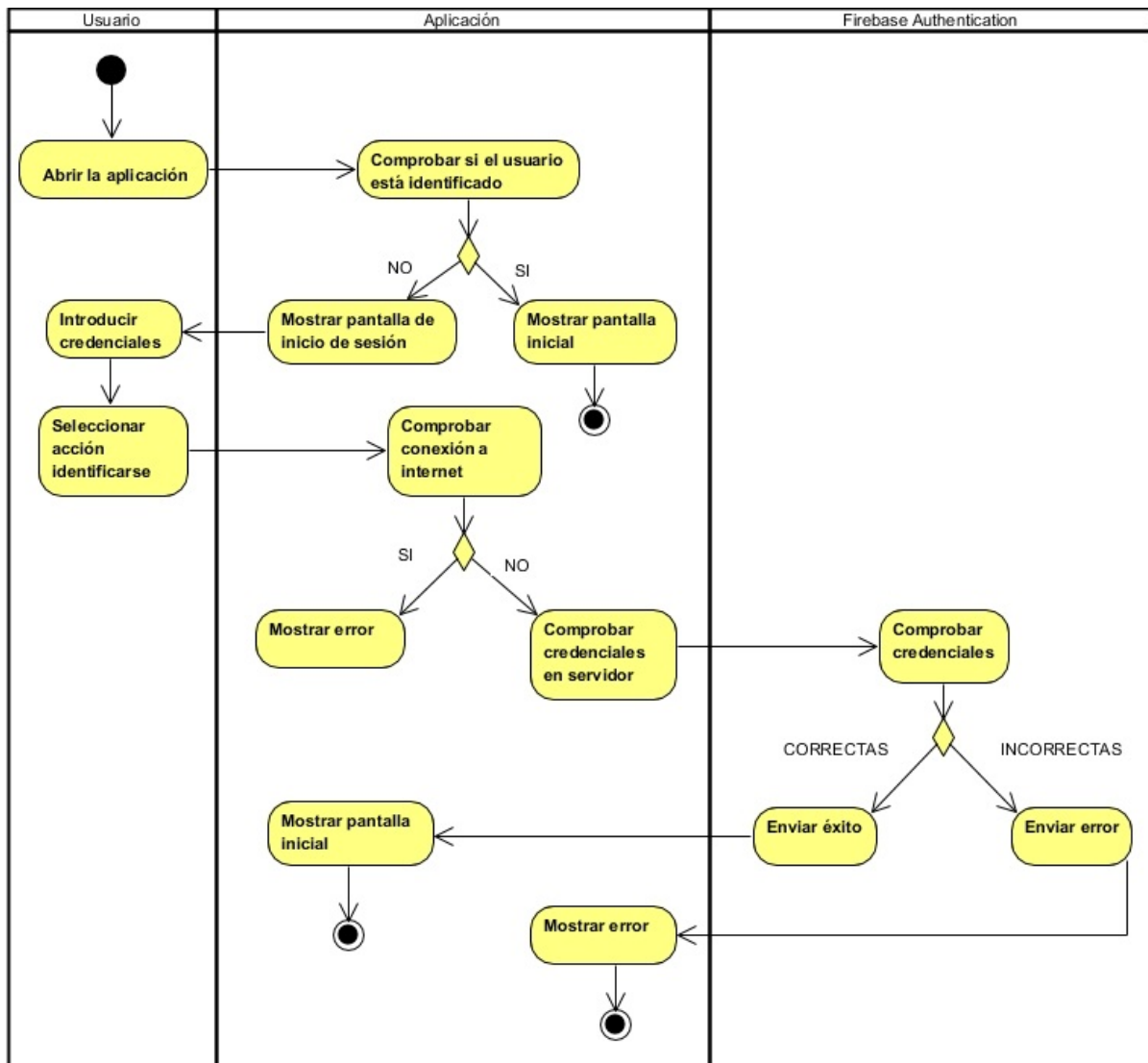


Figura 7.4: Diagrama de actividad CU01.

CU02: Registrarse

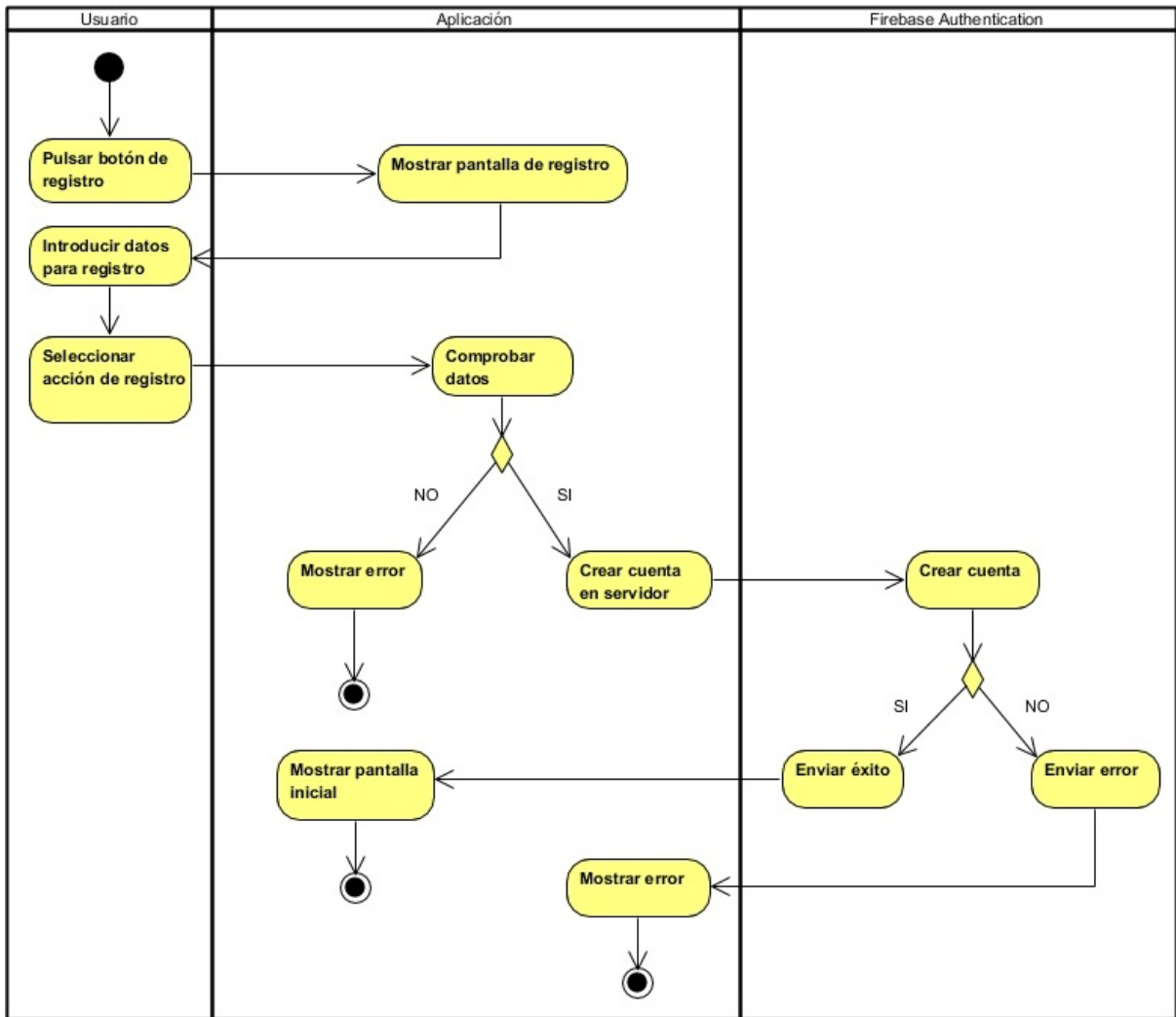


Figura 7.5: Diagrama de actividad CU02.

CU03: Consultar perfil

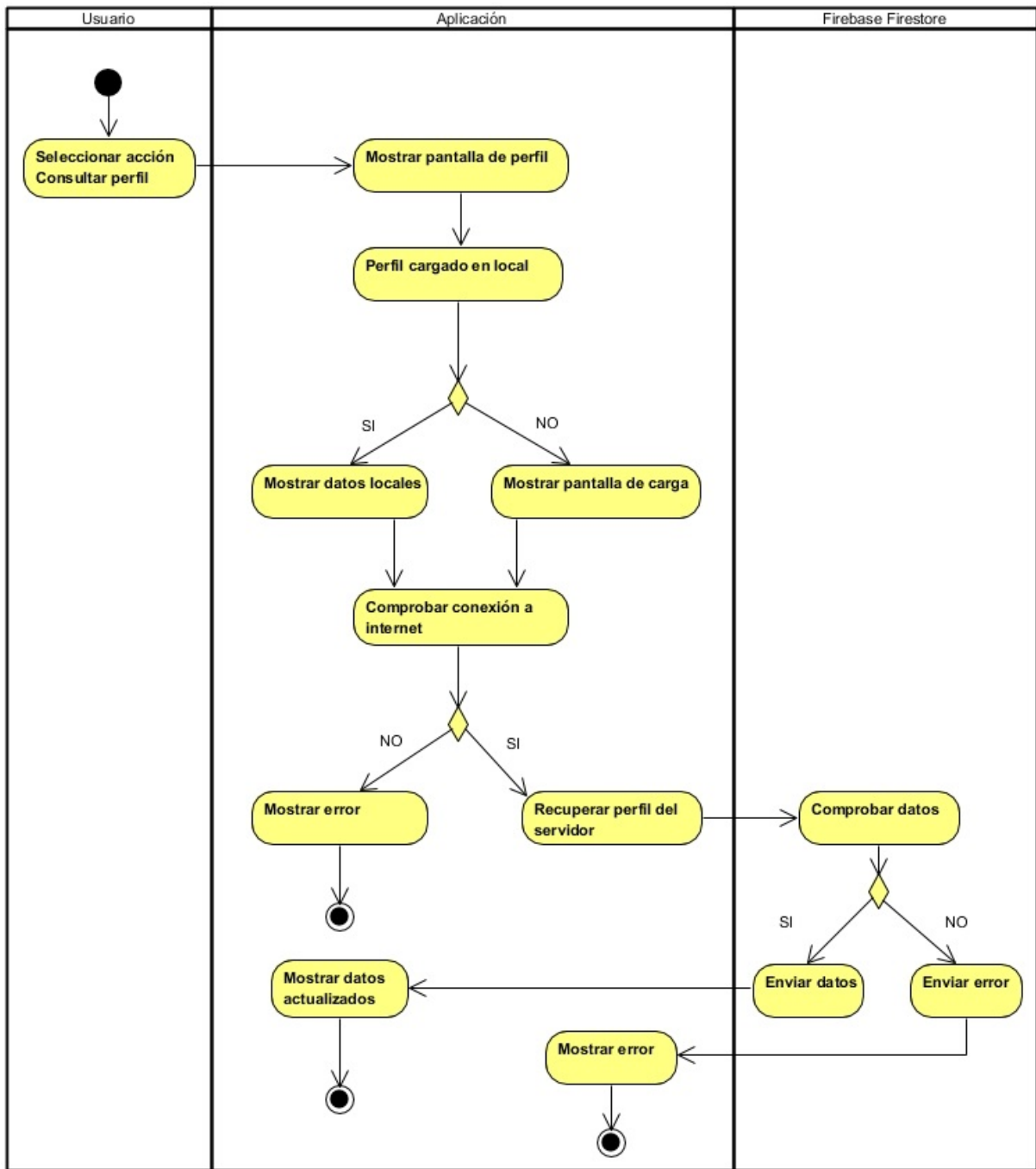


Figura 7.6: Diagrama de actividad CU03.

CU04: Modificar perfil

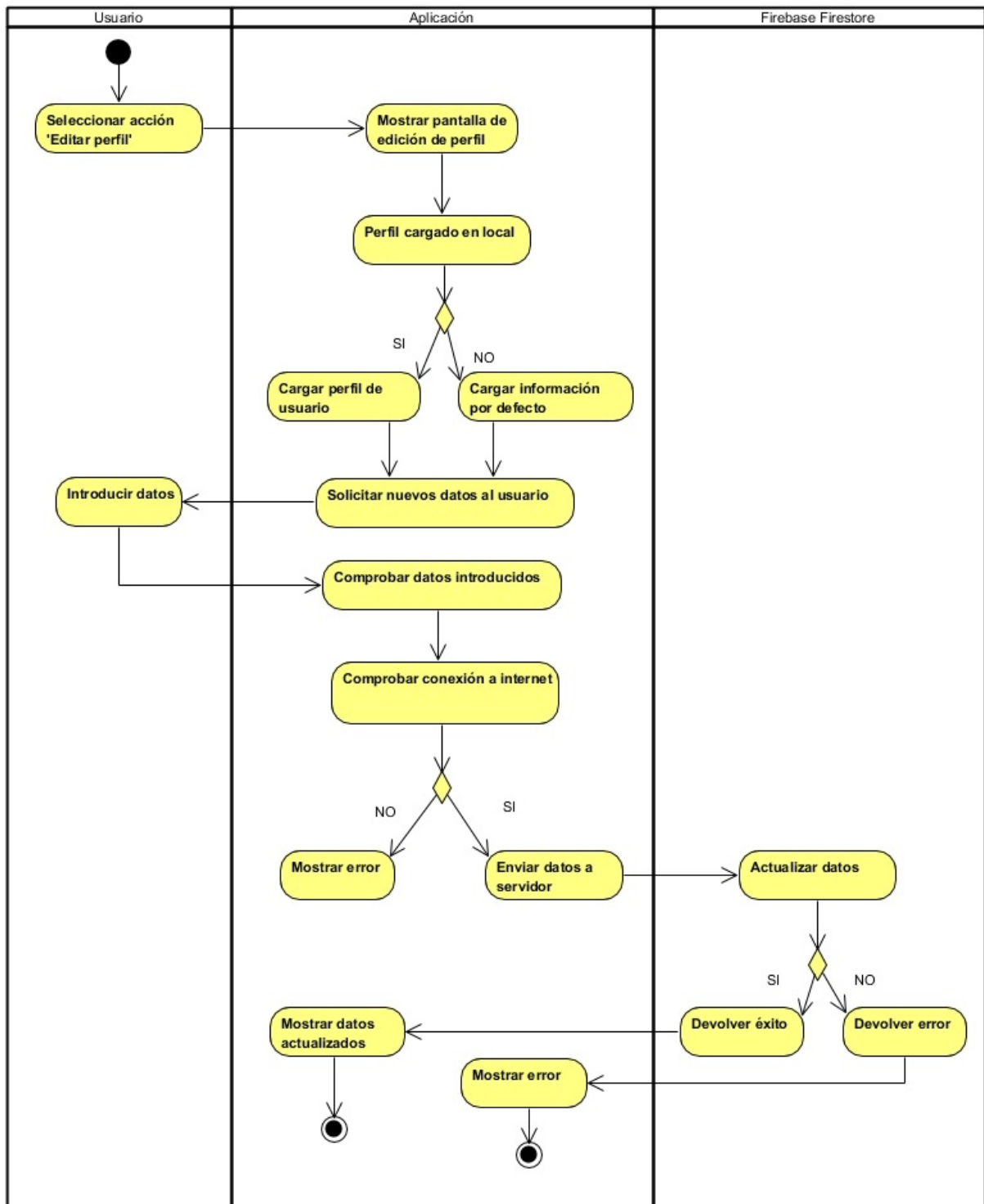


Figura 7.7: Diagrama de actividad CU04.

CU05: Consultar lista de partidos

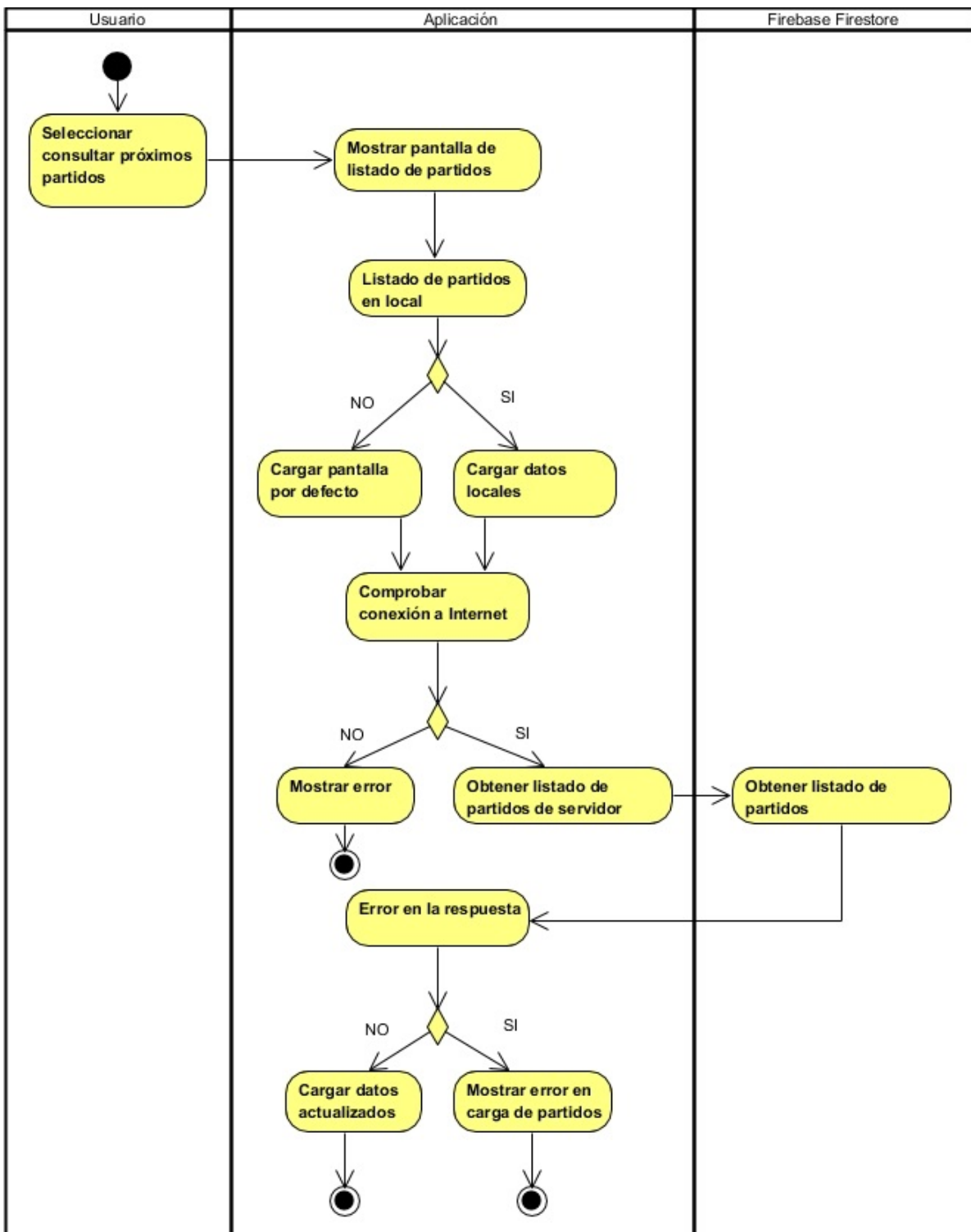


Figura 7.8: Diagrama de actividad CU05.

CU06: Filtrar lista de partidos

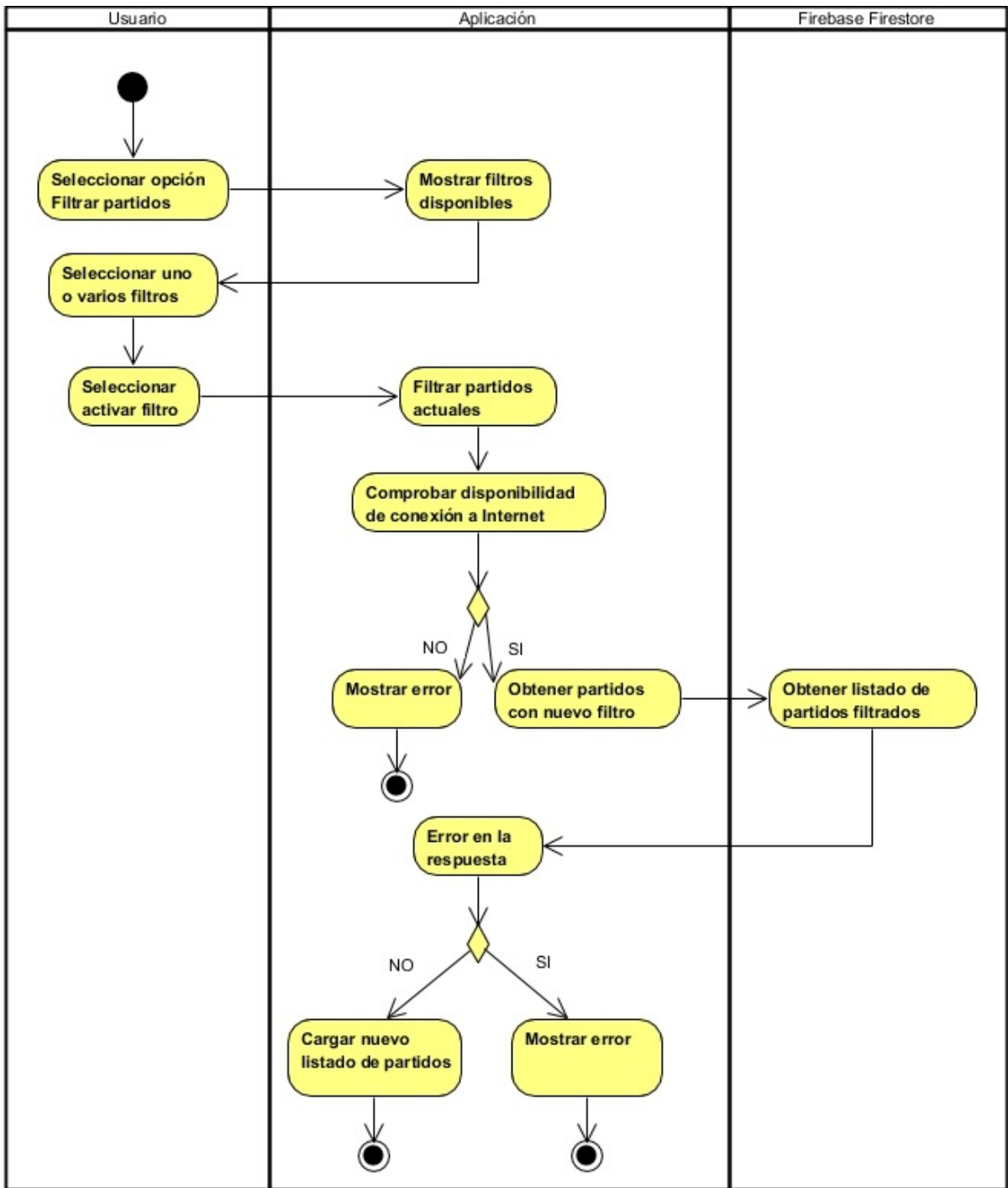


Figura 7.9: Diagrama de actividad CU06.

CU07: Consultar información de partido

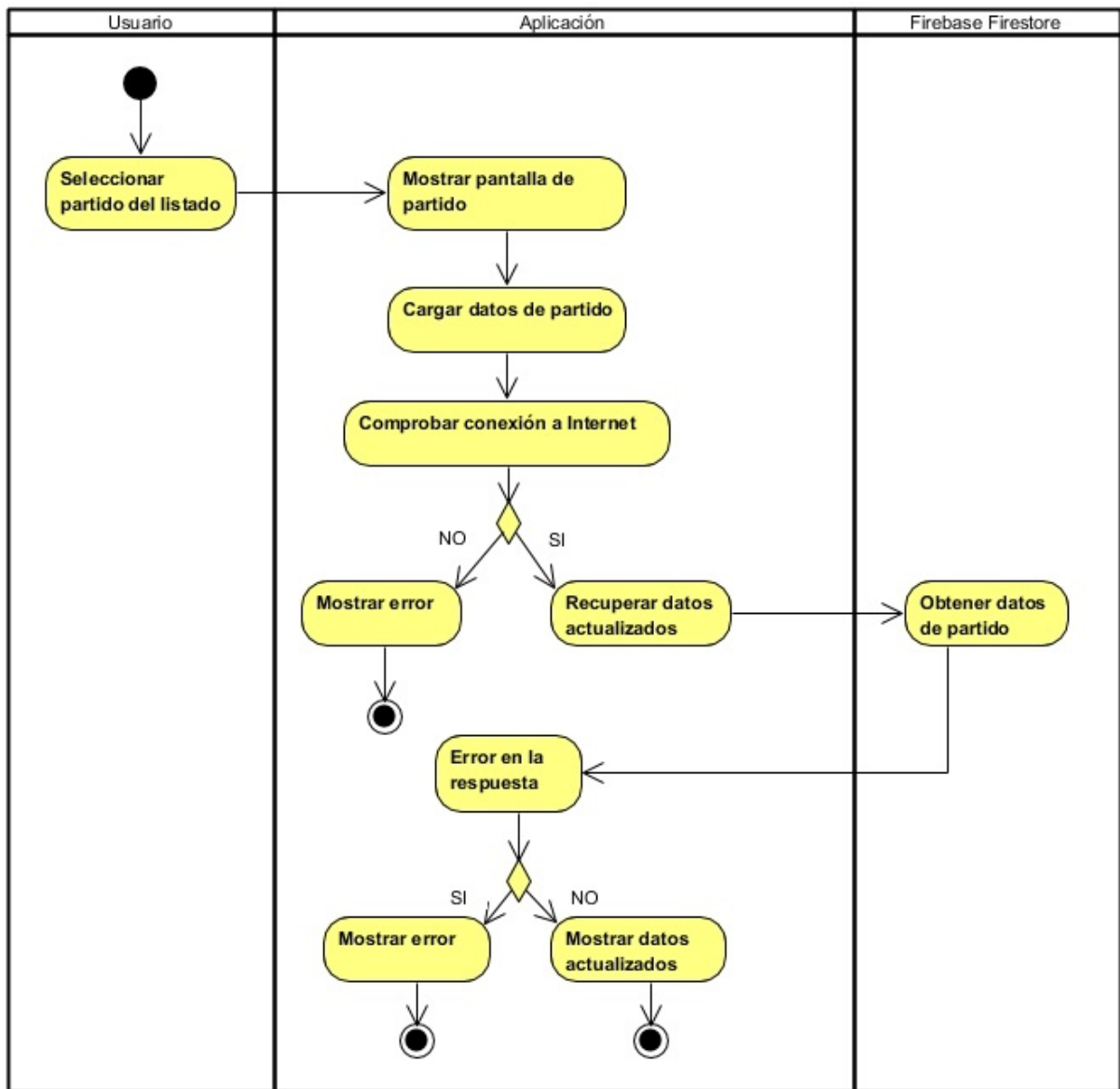


Figura 7.10: Diagrama de actividad CU07.

CU08: Consultar organizador de un partido

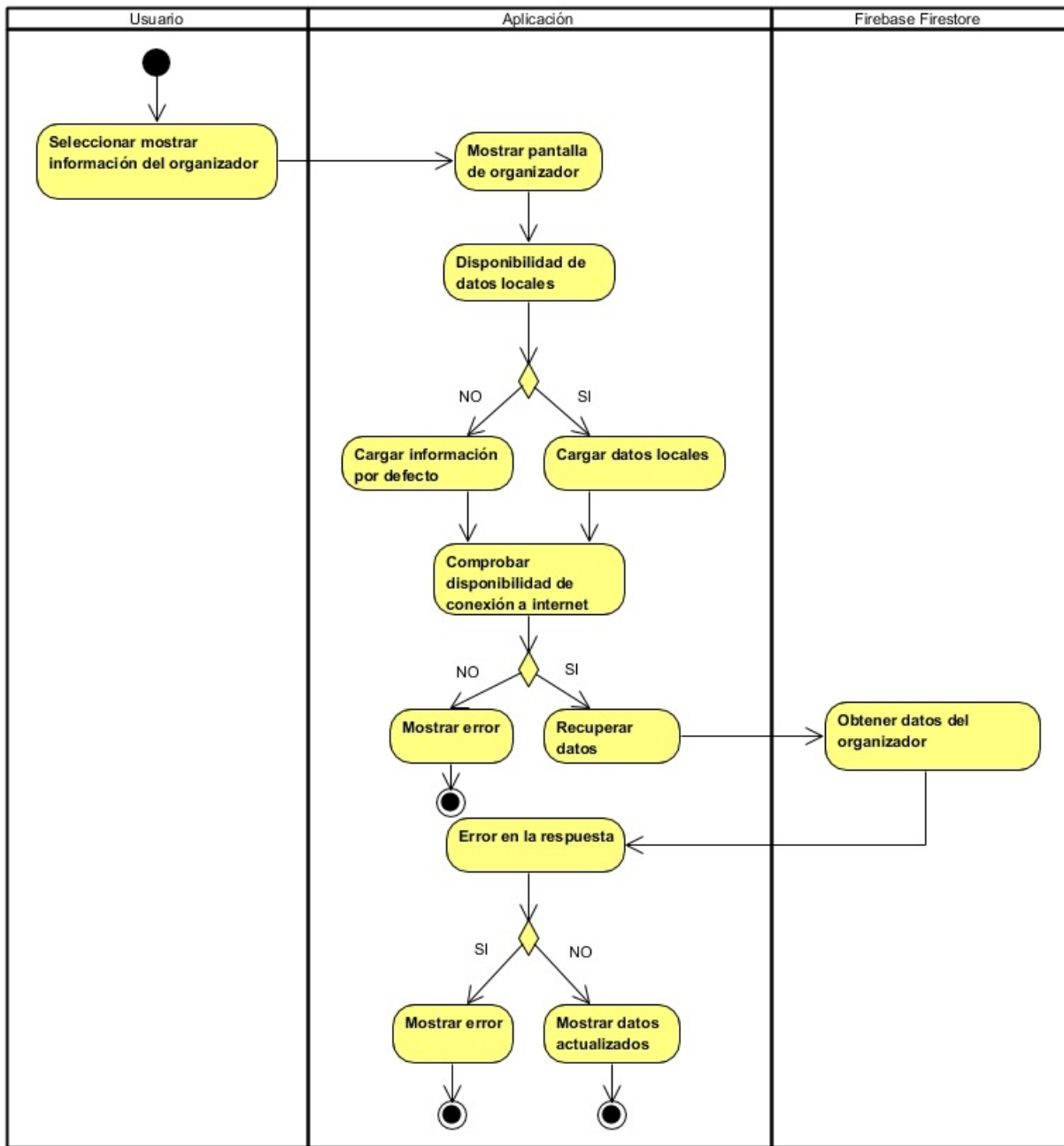


Figura 7.11: Diagrama de actividad CU08.

CU09: Consultar jugadores inscritos

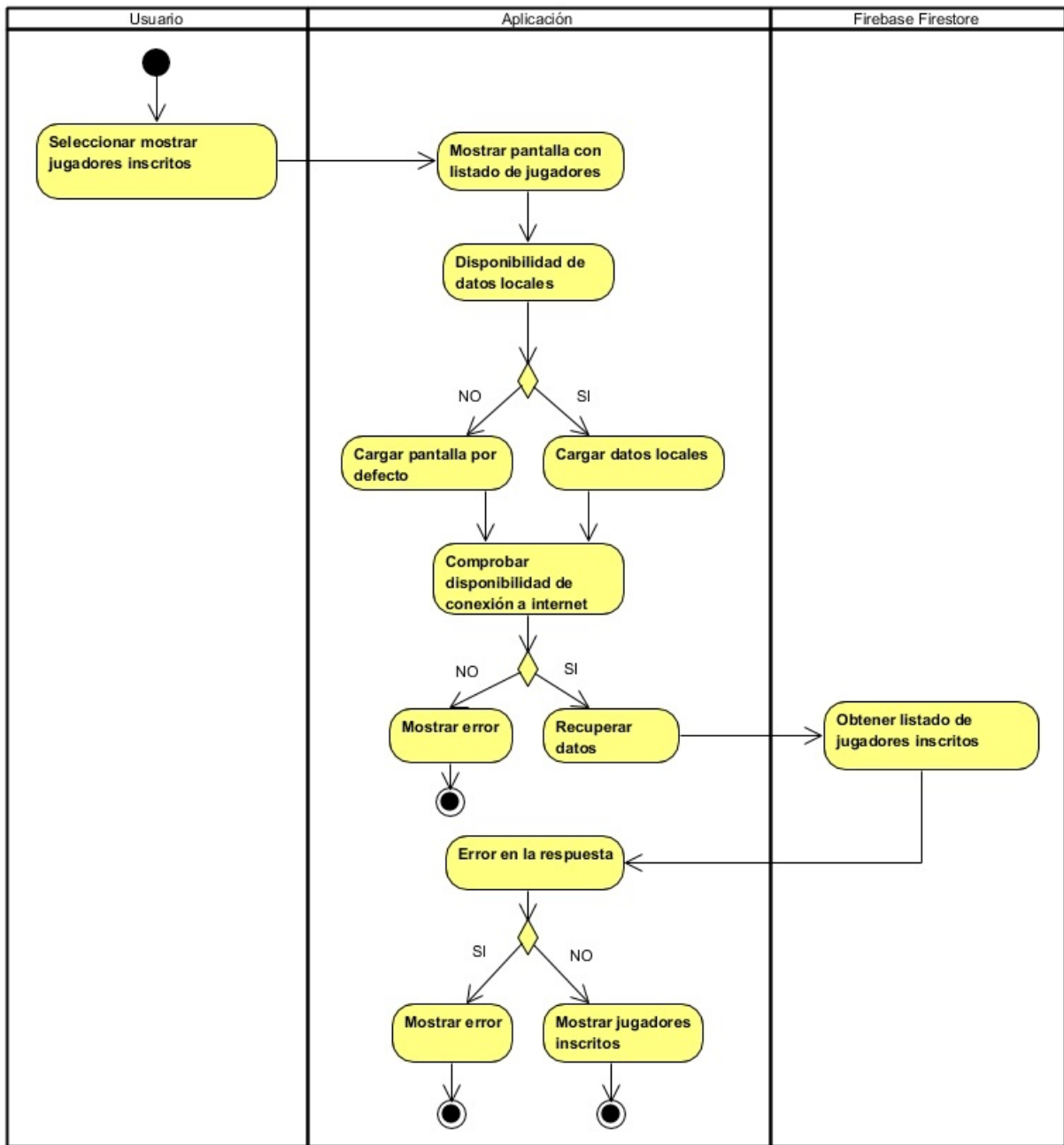


Figura 7.12: Diagrama de actividad CU09.

CU10: Unirse a partido

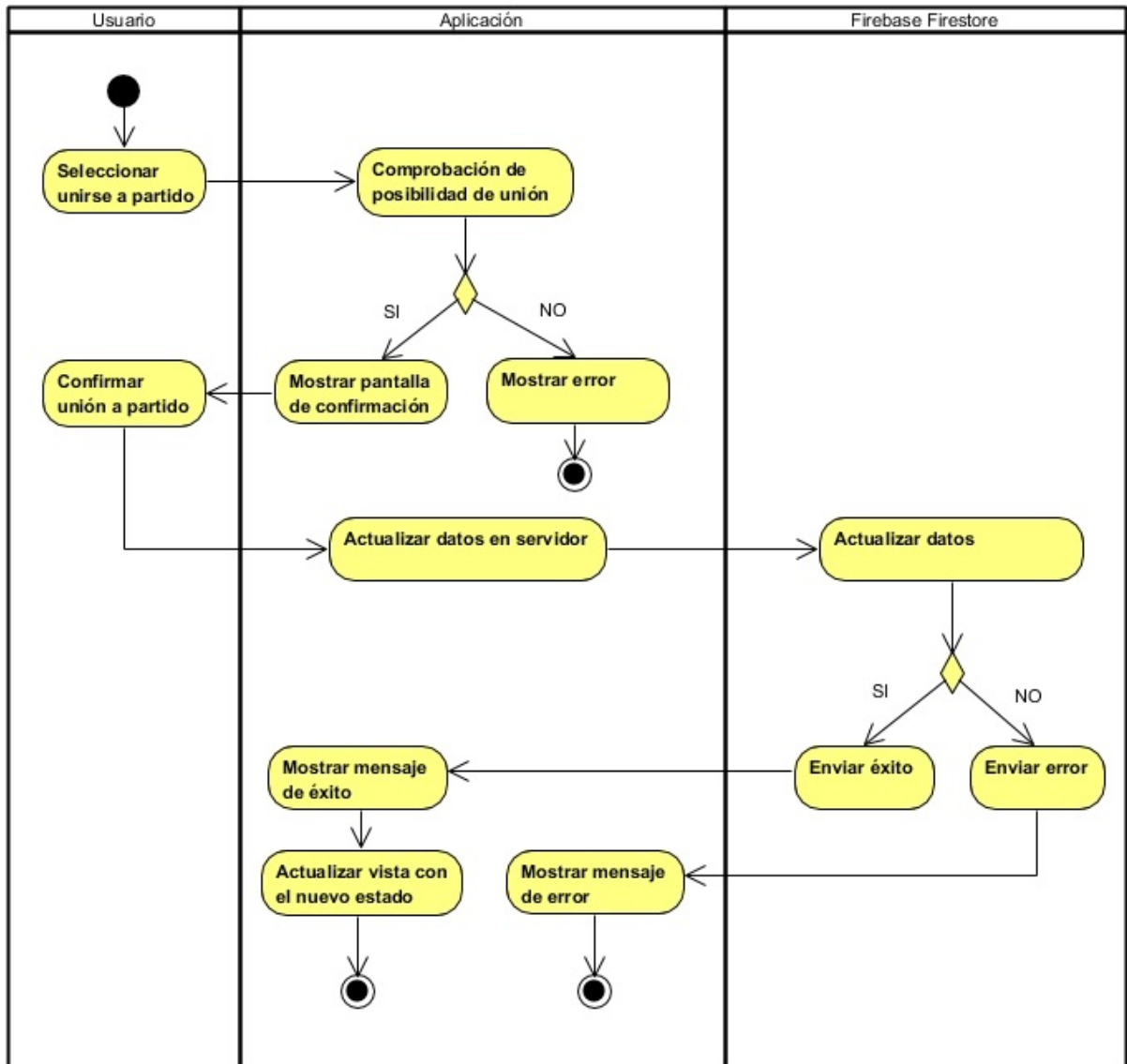


Figura 7.13: Diagrama de actividad CU10.

CU11: Abandonar partido

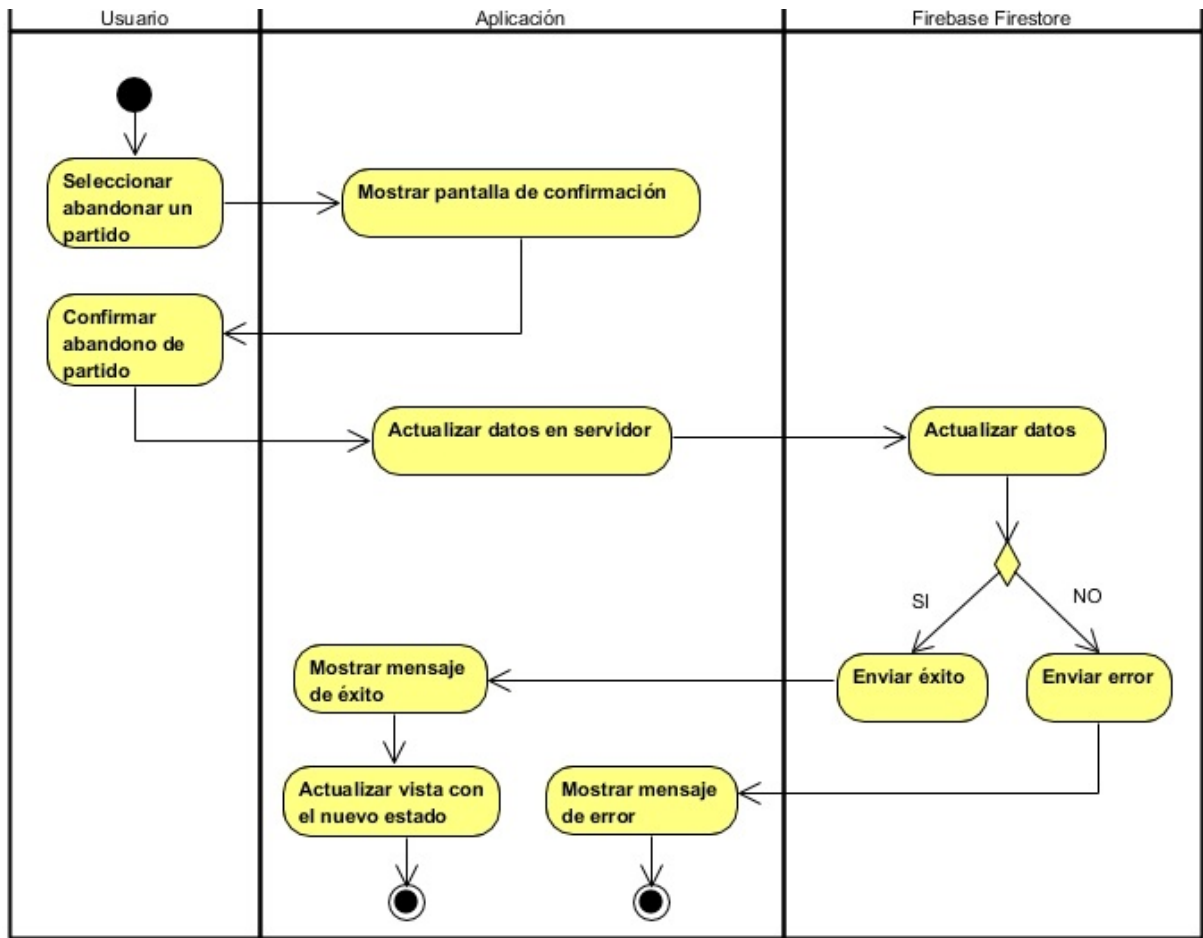


Figura 7.14: Diagrama de actividad CU11.

CU12: Cerrar sesión

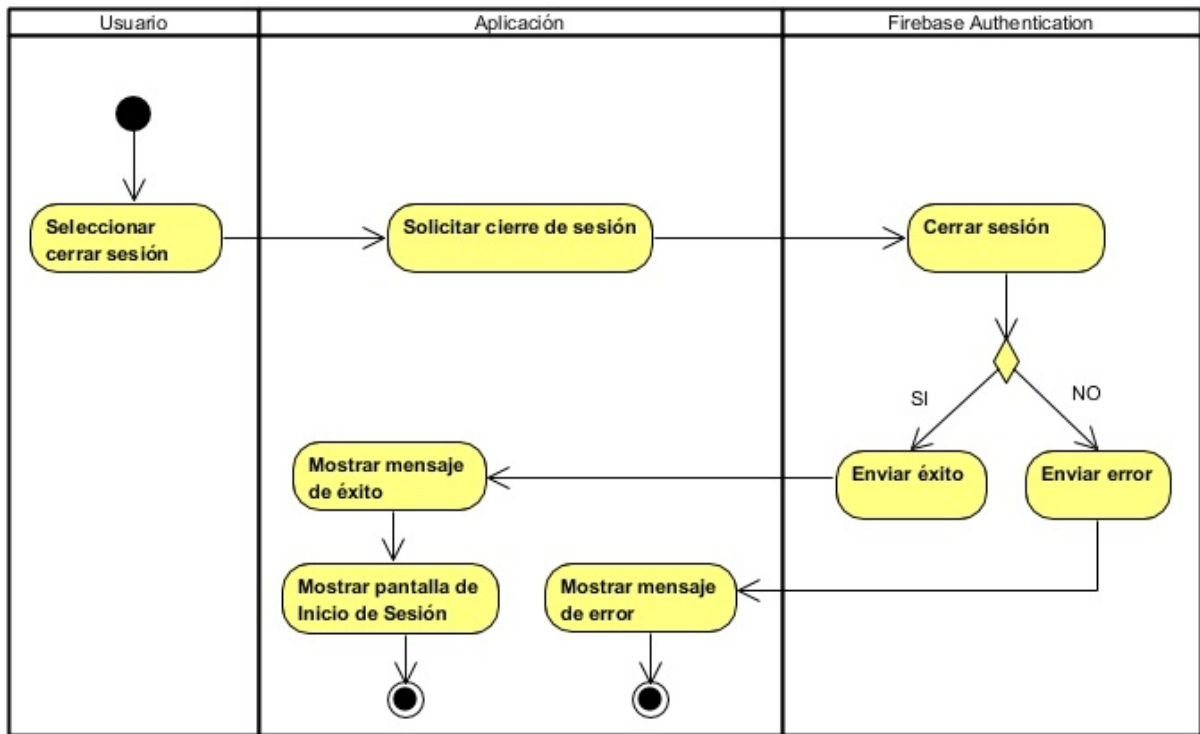


Figura 7.15: Diagrama de actividad CU12.

7.1.7. Diagramas de actividad Cribi Clubs

Debido a que ambas aplicaciones comparten cierta funcionalidad, los diagramas de actividad serán iguales. A continuación se muestran aquellos que son compartidos por ambas aplicaciones:

- **CU13:** Comparte diagrama de actividad con CU01, representado en la Figura 7.4.
- **CU14:** Comparte diagrama de actividad con CU03, representado en la Figura 7.6.
- **CU15:** Comparte diagrama de actividad con CU04, representado en la Figura 7.7.
- **CU18:** Comparte diagrama de actividad con CU07, representado en la Figura 7.10.
- **CU19:** Comparte diagrama de actividad con CU09, representado en la Figura 7.12.
- **CU22:** Comparte diagrama de actividad con CU12, representado en la Figura 7.15.

Se mostrarán a continuación los diagramas de actividad exclusivos de la aplicación Cribi Clubs:

CU16: Consultar lista de partidos organizados

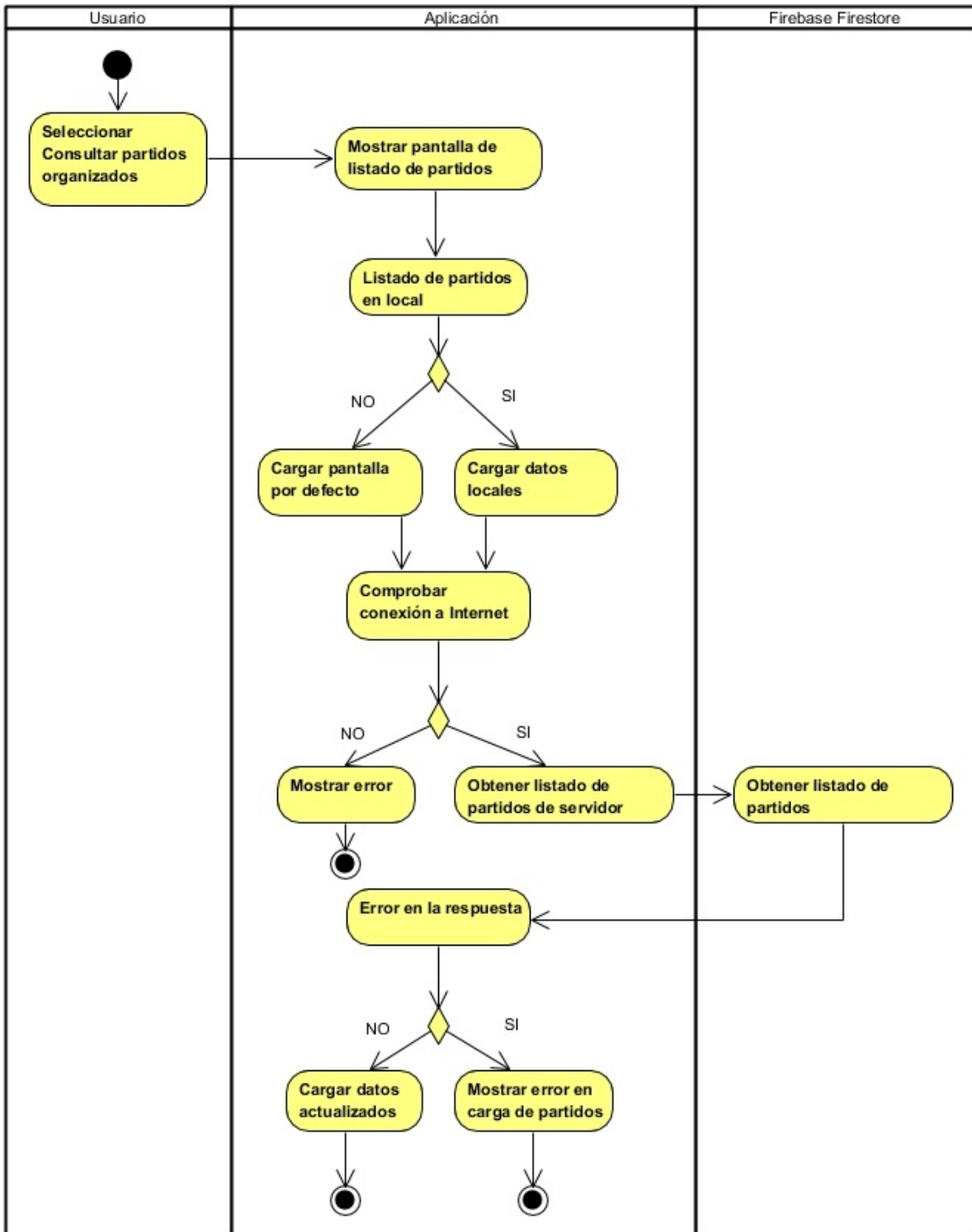


Figura 7.16: Diagrama de actividad CU16.

CU17: Filtrar listado de partidos organizados

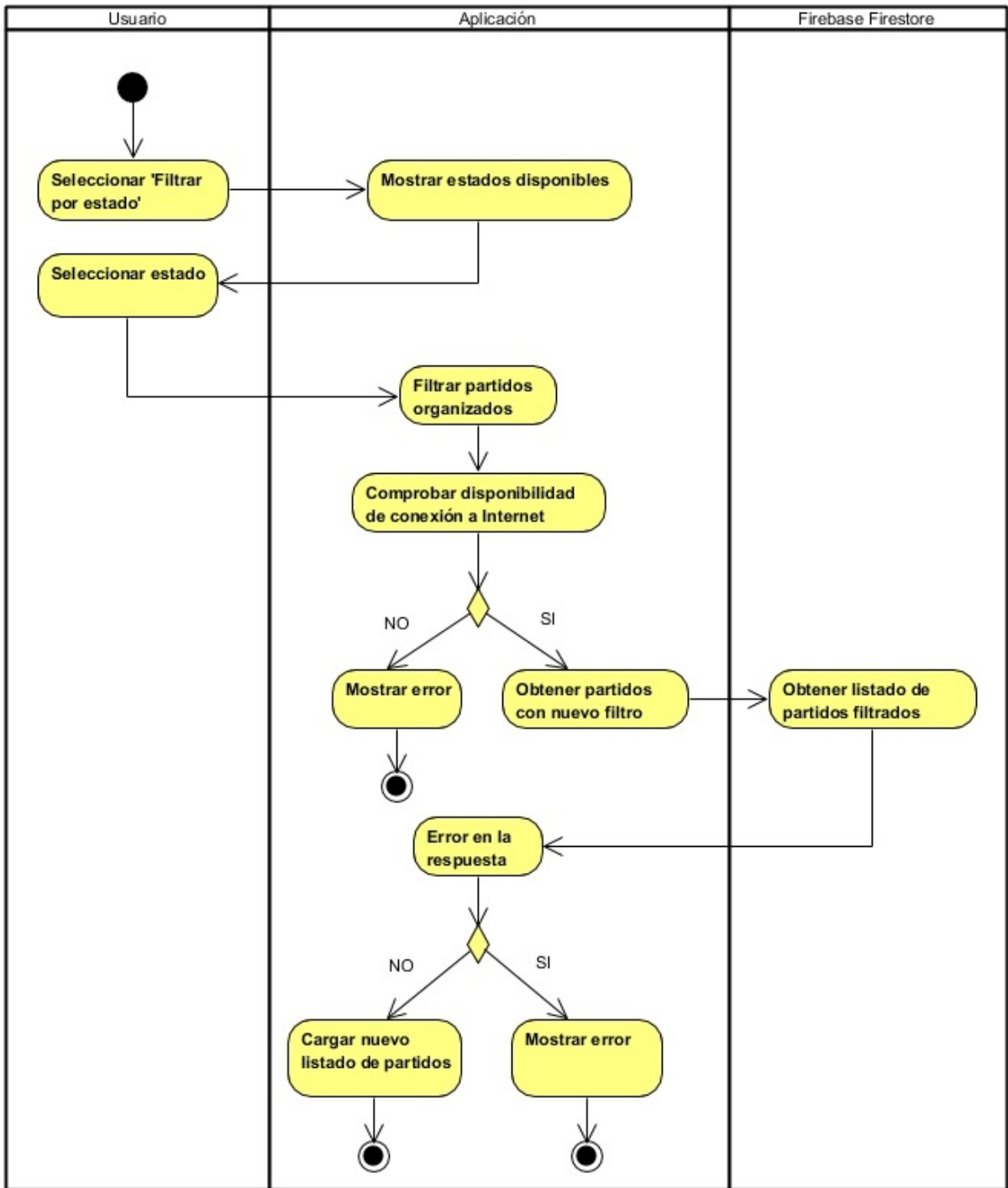


Figura 7.17: Diagrama de actividad CU17.

CU20: Cambiar estado de partido

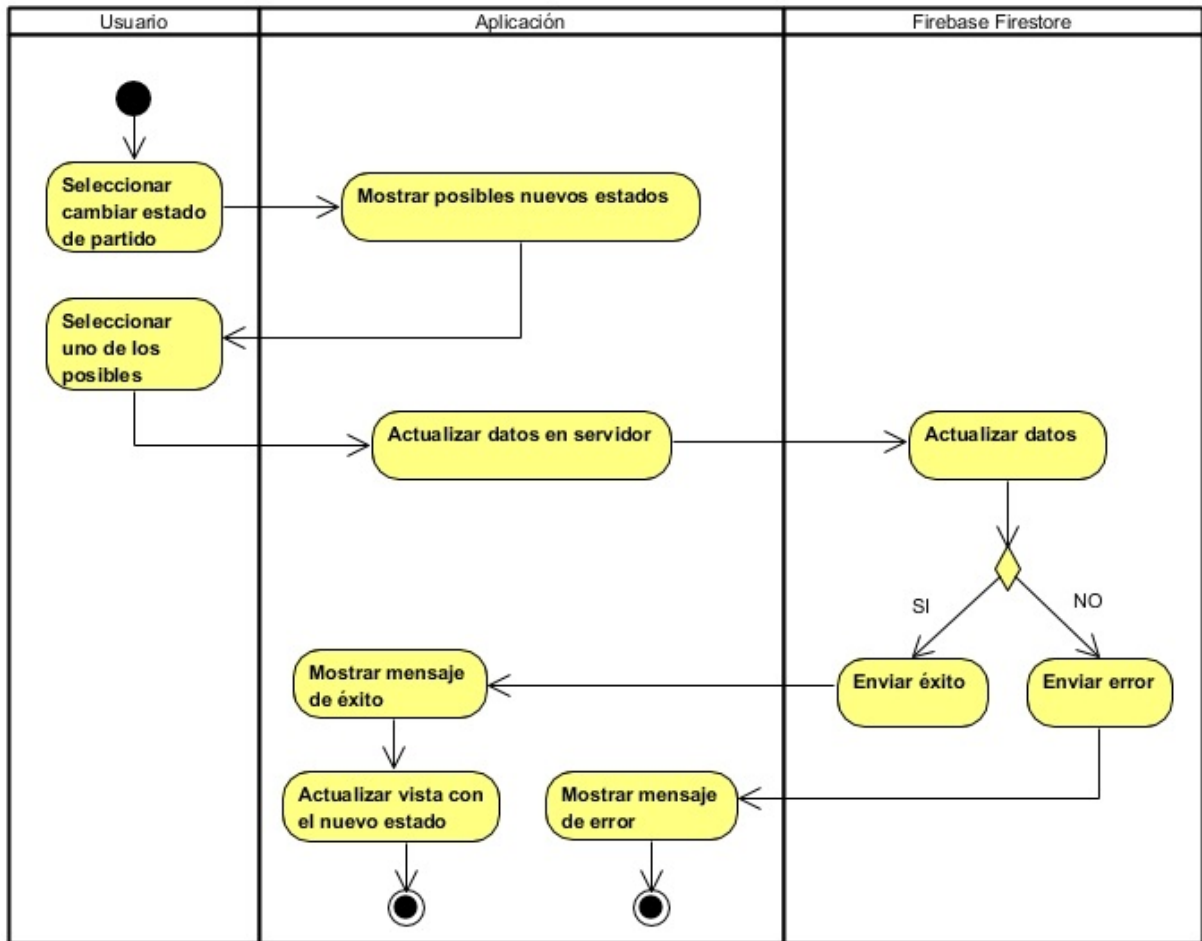


Figura 7.18: Diagrama de actividad CU20.

CU21: Crear nuevo partido

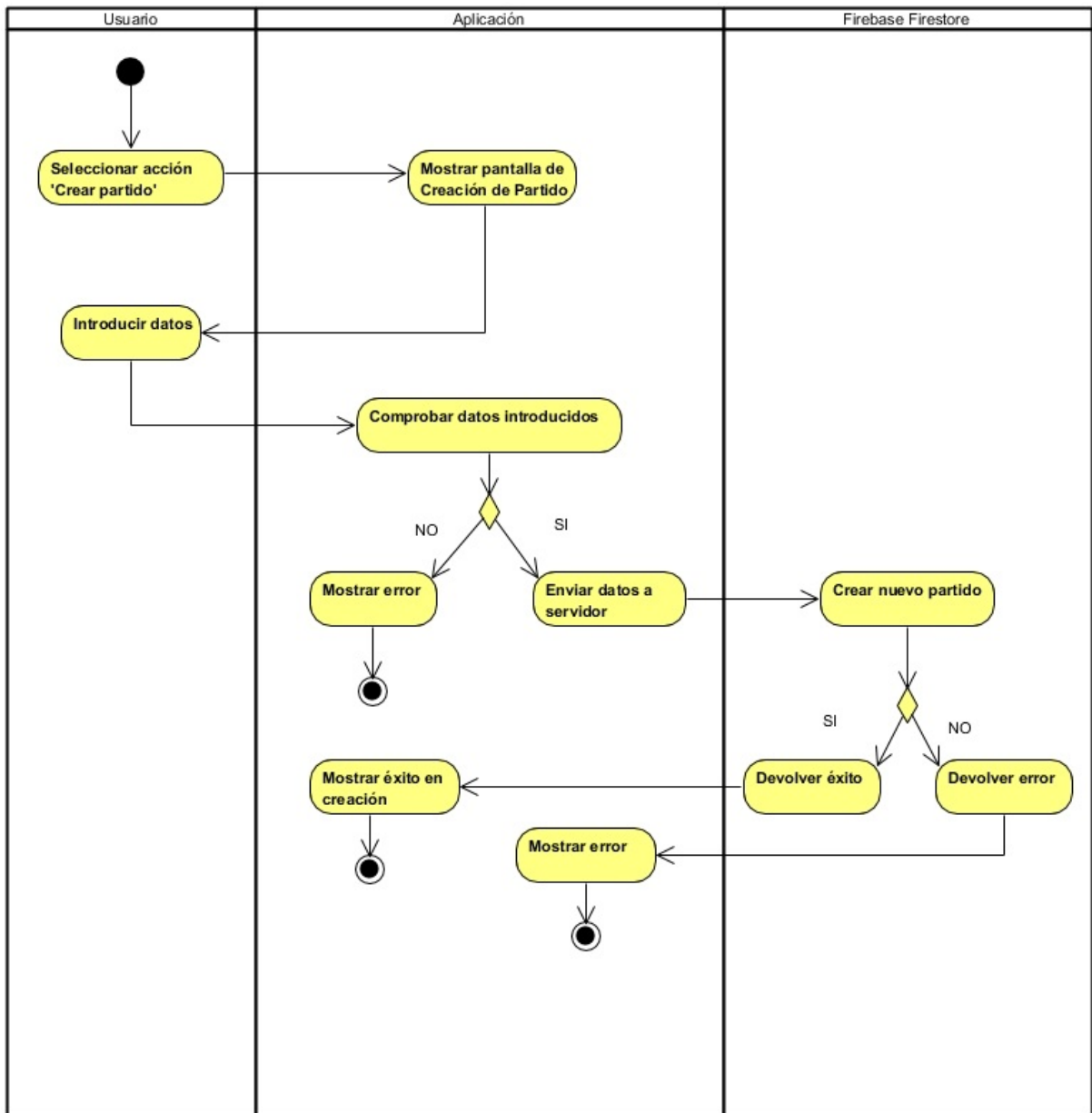


Figura 7.19: Diagrama de actividad CU21.

Capítulo 8

Diseño

8.1. Diseño de la arquitectura

8.1.1. Patrón arquitectónico MVVM

En la actualidad existen diferentes patrones arquitectónicos para el desarrollo de aplicaciones móviles. Entre ellos están los conocidos MVC, MVP o MVVM. Para este proyecto se decidió el uso del patrón MVVM, al que Google le ha dedicado especial atención desde la salida de los componentes de arquitectura en 2017 [17]. Estos componentes de arquitectura forman parte de Android Jetpack[18]. Jetpack es una colección de componentes software de Android que permite desarrollar aplicaciones Android sólidas, mantenibles y testeables.

Jetpack permite acelerar el desarrollo, manteniendo componentes que pueden usarse individualmente y que facilitan tareas tediosas como la administración del ciclo de vida de las aplicaciones. Pero lo más destacado es la escalabilidad, testeabilidad y mantenimiento de las aplicaciones construidas usando estos componentes.

Además, mediante el uso de esta arquitectura, cada componente tendrá una única responsabilidad, tal y como dicta el principio de responsabilidad única. De esta forma se abandonará el error común de introducir toda la lógica de negocio dentro de una Activity, lo que evidentemente la convierte en difícil de leer, testear y mantener.

La arquitectura base de una aplicación utilizando estos componentes se muestra en la Figura 8.1[19].

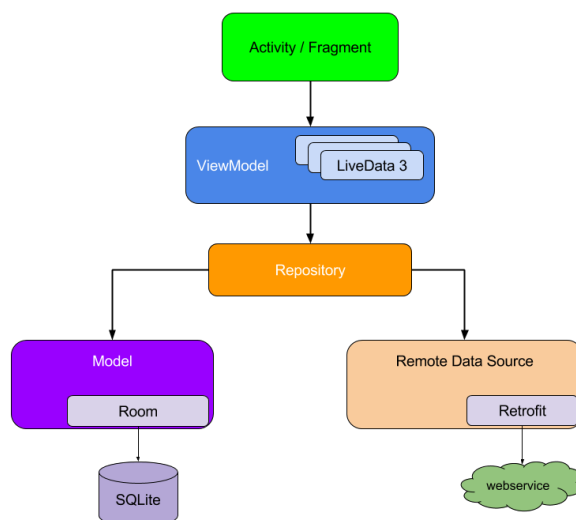


Figura 8.1: Arquitectura MVVM

Como se puede observar, cada componente depende exclusivamente del que se encuentra un nivel más abajo. El repositorio sería el único que dependa de otras clases, puesto que las fuentes de datos podrán ser locales o remotas. Además el ViewModel no tendría ningún conocimiento sobre qué vistas están haciendo uso de él. Así mismo, el repositorio tampoco sabría qué ViewModel estaría haciendo uso de él.

Este diseño permite un desacoplamiento total entre modelo y vista, lo cual permite una testeabilidad completa y simplificada. Además, hace que no se tengan que enviar datos a la propia vista para que se actualice como si que ocurre en otros patrones, ya que estará observando cambios en el modelo a través del LiveData como se comentará posteriormente.

A continuación se detallarán los principales detalles del patrón arquitectónico MVVM (Model-View-ViewModel) [20].

- **Model:** El modelo contendrá la lógica de negocio de la aplicación. El repositorio sería la única fuente de verdad para los ViewModel, de tal forma que cuando el ViewModel necesita de algún dato, lo tomará del repositorio. A su vez, el repositorio será el encargado de obtener esa información desde la fuente de datos local o desde una fuente de datos remota.
- **View:** En el contexto de MVVM, la vista se encargará de mostrar información y de manejar la interacción directa con el usuario. Esta vista no mantendrá lógica de negocio alguna, si no que se encargará de comunicar al ViewModel los eventos producidos al interactuar con el usuario.
- **ViewModel:** El ViewModel establecería un puente entre la vista y el modelo. Se encargaría de solicitar las operaciones necesarias al modelo para proporcionar las necesidades que tiene la vista.

8.1.2. Arquitectura general

Para esta aplicación, la arquitectura MVVM se ha adaptado como en la Figura 8.2 (Fuente: [21]).

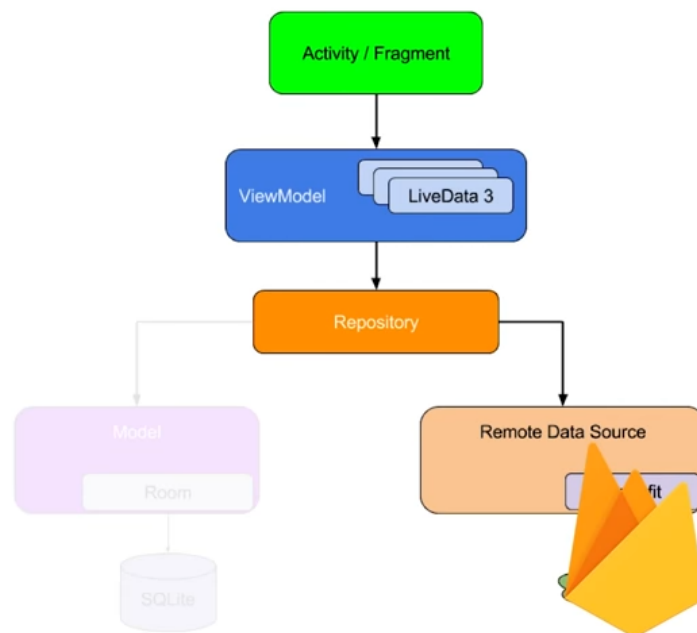


Figura 8.2: Arquitectura MVVM en conjunto con Firestore

A continuación se comentará el uso de cada componente y el por qué se ha utilizado esta solución.

En primer lugar tendremos el repositorio, cuya única fuente de datos será Firebase. Esta elección es debida al aprovechamiento de las cachés que ofrece Firestore de las consultas recientes. Es por ello que no se ha utilizado la capa de persistencia Room para almacenamiento local, aunque podría ser interesante añadirla en el futuro.

Este repositorio gestionará todas las conexiones a la base de datos (Firebase Firestore), al almacenamiento de imágenes (Firebase Storage) y al servicio de autenticación (Firebase Authentication). A través de estas conexiones realizará las operaciones y obtendrá los datos requeridos por otros componentes.

Para mantener estos datos se utilizarán LiveData. LiveData es un objeto observable que respeta los ciclos de vida de los demás componentes. De esta forma, solo notificará de cambios a los observadores que tengan un ciclo de vida activo.

En segundo lugar se encuentra el ViewModel, que como se puede observar en la Figura 8.1, tendrá una conexión con el repositorio, pero el repositorio no tendrá una conexión con el propio ViewModel. Esto permite una estructura completamente testeable y mantenible.

Pero el ViewModel, al ser el nexo entre repositorio y vista, entonces tendrá que mantener una conexión para proporcionar los datos a la vista. Esta conexión se realiza a través de LiveData. De esta forma, el ViewModel recuperará los datos del repositorio y los expondrá a la vista a través de estos observables.

Además, el ViewModel permite administrar el ciclo de vida de una actividad o fragmento puesto que se mantiene en memoria hasta que el ciclo de vida de mismo finaliza. Esto permite que los datos no se pierdan aunque haya cambios de configuración como, por ejemplo, la rotación de pantalla.

Por último, se encuentra la Vista. Esta vista se encargará de recibir los distintos eventos que realice el usuario en la interfaz y mantendrá una conexión con el ViewModel para gestionar estos eventos. Mediante esta conexión, la vista observará el LiveData del ViewModel, de tal manera que actualizará la pantalla en función de los datos del mismo.

8.1.3. Estructura del proyecto

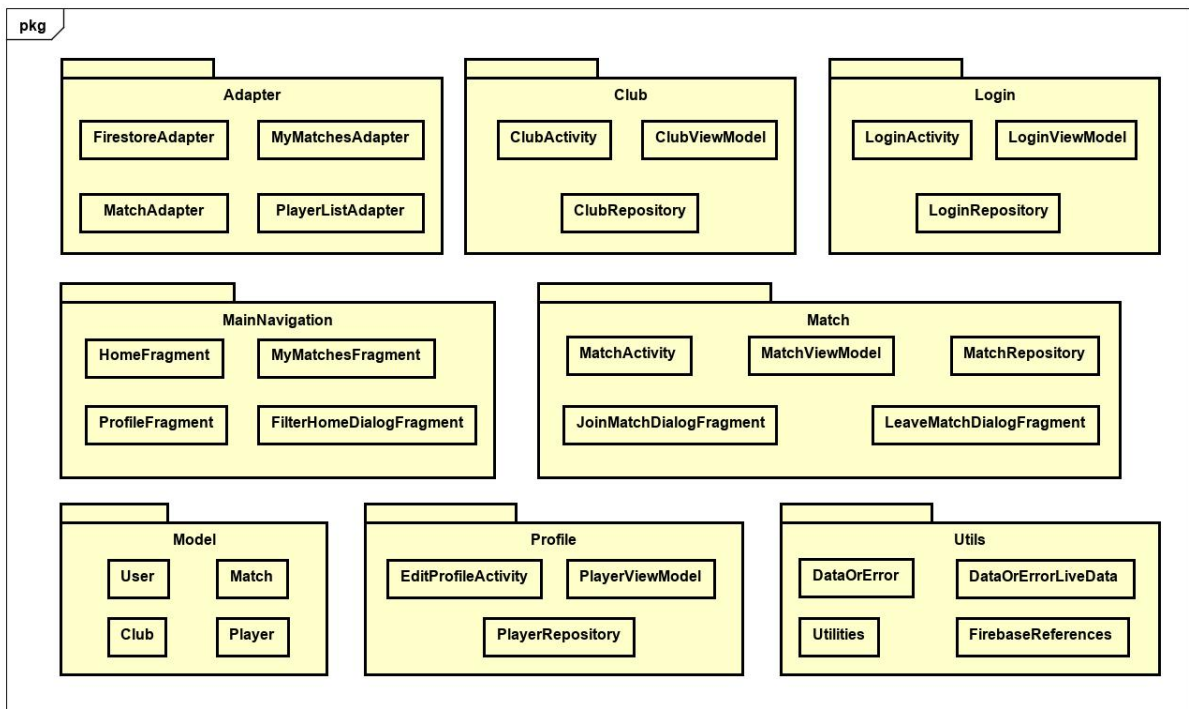


Figura 8.3: Paquetes aplicación Cribi

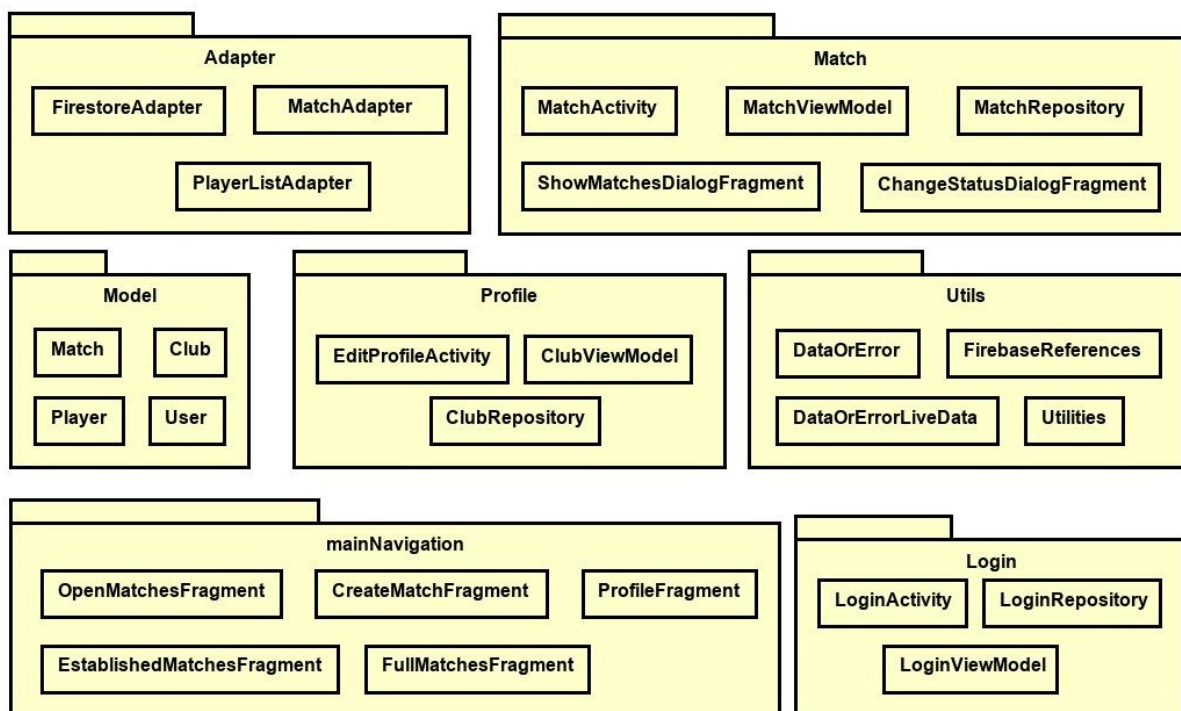


Figura 8.4: Paquetes aplicación Cribi Clubs

8.1.4. Interacción entre componentes

En esta sección se mostrarán algunos de los detalles de la implementación de la arquitectura MVVM en la aplicación Cribi y la interacción entre los diferentes componentes. Esta implementación será común a lo largo de todo el proyecto.

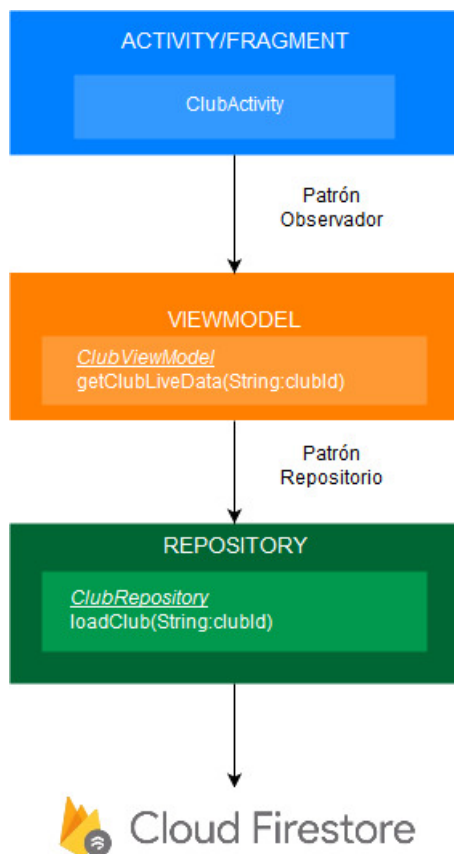


Figura 8.5: Interacción componentes Club.

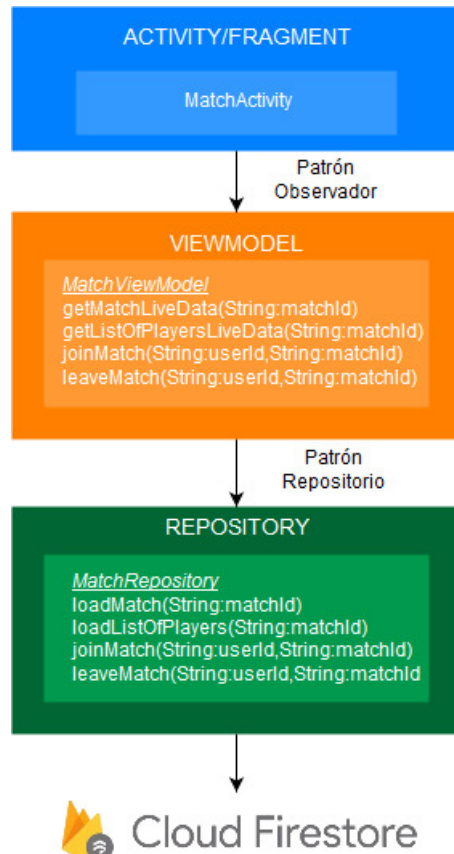


Figura 8.6: Interacción componentes Match.

8.2. Diseño de la base de datos no relacional

Como ya se ha comentado anteriormente, se ha utilizado Firestore como base de datos no relacional y en tiempo real para el desarrollo de este proyecto. Se mostrarán brevemente los diferentes tipos de jerarquía posibles y la elección de la utilizada finalmente. Esta decisión se fundamenta en el artículo [22]:

- **Datos anidados en documentos:** Anidación de objetos complejos dentro de los documentos. Esta solución resultaría ineficiente en este proyecto puesto que necesitamos mantener un índice y realizar consultas continuamente. Además, sería poco escalable puesto que a medida que la base de datos creciese, los documentos y su tiempo de consulta aumentarían.
- **Subcolecciones:** Creación de colecciones dentro de los documentos. Esta solución podría dar resultado teniendo en cuenta que el tamaño del documento no variaría, pero no se dispondría de consultas compuestas que son necesarias para poder filtrar partidos en este aspecto.
- **Colecciones a nivel de raíz:** Creación de colecciones a nivel de raíz de la base de datos. Esta jerarquía permitiría escalar la aplicación y disponer de todas las posibilidades de consultas que ofrece Firestore.

Tras consultar este artículo y la presentación expuesta en Google I/O'19 [23] se decidió que la estructura de la base de datos estaría formada por tres grandes colecciones a nivel de raíz (Matches, Clubs, Users) que permitiesen escalar y aprovechar por completo las consultas que ofrece Firestore.

Además se utilizarían subcolecciones para relacionarlas puesto que no se dispone de sentencias JOIN en este tipo de base de datos. De esta forma, Matches tendría una subcolección con los jugadores que se han inscrito al partido y Users tendría una subcolección con los partidos a los que se ha inscrito el propio usuario.

Capítulo 9

Pruebas

9.1. Introducción

Como en todo proyecto software, la realización de pruebas es fundamental para comprobar el funcionamiento y robustez del código desarrollado.

Mediante estas pruebas, se permite detectar los errores a medida que vamos generando cada funcionalidad, de tal forma que se puedan corregir los posibles errores en etapas tempranas de desarrollo. Además, al final de cada sprint se realizaba una prueba general del sistema.

Las pruebas realizadas en este proyecto son pruebas puramente manuales y de caja negra. Estas pruebas se han llevado a cabo utilizando los dispositivos comentados en la sección 2.2 de esta memoria.

9.2. Casos de prueba

9.2.1. Aplicación para jugadores Cribi

CP01	Registro y configuración de perfil
Descripción	El usuario se registra en la aplicación para poder utilizar los servicios.
Acción	Ir a la vista de registro, introducir los datos de acceso que se quieran para acceder a la aplicación. Pulsar el botón de registrarse, introducir los datos que conformarán el perfil del usuario y guardar los cambios.
Resultado esperado	Se crea una nueva cuenta en el sistema con el perfil configurado.
Resultado obtenido	Correcto

Tabla 9.1: Descripción del CP01

CP02	Identificación
Descripción	Ir a la vista de identificación, introducir las credenciales correctas y pulsar identificarse.
Resultado esperado	El usuario jugador se identifica y se redirecciona a la página de inicio de la aplicación.
Resultado obtenido	Correcto

Tabla 9.2: Descripción del CP02

CP03	Identificación II
Descripción	Ir a la vista de identificación, introducir unas credenciales falsas y pulsar identificarse.
Resultado esperado	Mostrar un toast indicando que las credenciales son erróneas.
Resultado obtenido	Correcto

Tabla 9.3: Descripción del CP03

CP04	Cerrar sesión
Descripción	Ir a la vista de ajustes, y seleccionar Cerrar Sesión.
Resultado esperado	Se cierra la sesión y se redirecciona al usuario a la pantalla de login.
Resultado obtenido	Correcto

Tabla 9.4: Descripción del CP04

CP05	Ver perfil de usuario
Descripción	Ir a la vista de perfil.
Resultado esperado	Se muestra la vista con el perfil del usuario.
Resultado obtenido	Correcto

Tabla 9.5: Descripción del CP05

CP06	Modificar perfil de usuario
Descripción	Desde la vista de perfil, seleccionar editar perfil, introducir los cambios y guardar.
Resultado esperado	Se actualizan los datos de perfil.
Resultado obtenido	Correcto

Tabla 9.6: Descripción del CP06

CP7	Modificar imagen de perfil
Descripción	Desde la vista de perfil, pulsar sobre la imagen, seleccionar una imagen de la galería y guardar.
Resultado esperado	Se actualizan la imagen de perfil.
Resultado obtenido	En la primera ejecución, la lógica de solicitud de permisos para acceder al almacenamiento no era correcta, por lo que no se permitía acceder a la galería aunque se aceptasen los permisos. Solucionado finalmente cambiando la lógica.

Tabla 9.7: Descripción del CP07

CP08	Mostrar listado de partidos
Descripción	Seleccionar la pestaña Home en el menú.
Resultado esperado	Se muestran los próximos partidos.
Resultado obtenido	En la primera ejecución, se obtenía un listado completo de partidos, lo cual era un gasto innecesario de recursos si se disponían de muchos partidos. Solucionado finalmente introduciendo paginación, de tal forma que se carga un número limitado de partidos y queda a disposición del usuario mostrar más, mediante scroll, o no.

Tabla 9.8: Descripción del CP08

CP09	Mostrar listado de partidos inscritos
Descripción	Seleccionar la pestaña Mis partidos en el menú. Se muestra el listado de los partidos a los que el usuario se ha inscrito.
Resultado esperado	Se muestran los próximos partidos que se ha inscrito el jugador.
Resultado obtenido	Correcto.

Tabla 9.9: Descripción del CP09

CP10	Filtrar listado de partidos
Descripción	En la pestaña Home en el menú, pulsar el botón de Filtrar Partidos. Se selecciona uno o varios y se filtra.
Resultado esperado	Se muestran el listado de partidos filtrado según los filtros elegidos.
Resultado obtenido	En la primera ejecución, el filtrado no era correcto debido a que no estaban configurados los índices compuestos en Firestore. Solucionado finalmente creando todos los índices compuestos posibles en Firestore para evitar cualquier problema adicional.

Tabla 9.10: Descripción del CP10

CP11	Ver información de partido
Descripción	En el listado de partidos, seleccionar uno de ellos.
Resultado esperado	Se muestra la información completa del partido.
Resultado obtenido	Correcto.

Tabla 9.11: Descripción del CP11

CP12	Ver jugadores inscritos a un partido
Descripción	En la pantalla de información de un partido seleccionar la opción de Ver jugadores inscritos.
Resultado esperado	Se muestra un listado con los jugadores que se han apuntado al partido.
Resultado obtenido	Correcto.

Tabla 9.12: Descripción del CP12

CP13	Ver organizador de un partido
Descripción	En la pantalla de información de un partido seleccionar la opción de Ver organizador de un partido.
Resultado esperado	Se muestra el perfil completo del organizador del partido.
Resultado obtenido	Correcto.

Tabla 9.13: Descripción del CP13

CP14	Unirse a un partido
Descripción	En la pantalla de información de un partido seleccionar la opción de Unirse a un partido, se muestra una pantalla de confirmación y se confirma.
Resultado esperado	El usuario aparece como inscrito al partido, se notifica al usuario, y se actualiza el botón de unión a partido.
Resultado obtenido	En la primera ejecución no se notificaba al usuario de la unión al partido. Finalmente se ha incluido un Toast para mostrar un mensaje de unión satisfactoria al partido.

Tabla 9.14: Descripción del CP14

CP15	Unirse a un partido II
Descripción	Mismo caso que en CP14 pero tras inscribirse a un partido a la misma hora.
Resultado esperado	Notificación de rechazo al tratar de unirse a un partido que se disputa en la misma fecha que otro al que el usuario ya se había inscrito.
Resultado obtenido	En la primera ejecución se realizaba una comprobación de la fecha exacta, pero se consideró que para mejor experiencia, se comprobaba un rango. Finalmente se ha comprobado un rango de varias horas, antes y después de cada partido, para así evitar problemas de concurrencia.

Tabla 9.15: Descripción del CP15

CP16	Abandonar un partido
Descripción	En la pantalla de información de un partido seleccionar la opción de Abandonar partido.
Resultado esperado	Se abandona el partido, se notifica al usuario y se actualiza el botón de abandonar partido.
Resultado obtenido	En la primera ejecución no se notificaba al usuario del abandono del partido. Finalmente se ha incluido un Toast para mostrar un mensaje de abandono satisfactorio del partido.

Tabla 9.16: Descripción del CP16

9.2.2. Aplicación para organizadores Cribi Clubs

CP17	Identificación organizador
Descripción	Ir a la vista de identificación, introducir las credenciales correctas y pulsar identificarse.
Resultado esperado	El usuario organizador se identifica y se redirecciona a la página de inicio de la aplicación.
Resultado obtenido	Correcto

Tabla 9.17: Descripción del CP17

CP18	Identificación organizador II
Descripción	Ir a la vista de identificación, introducir las credenciales de un jugador y pulsar identificarse.
Resultado esperado	El sistema identifica que el usuario no es un organizador, y muestra un mensaje de error.
Resultado obtenido	Correcto

Tabla 9.18: Descripción del CP18

CP19	Consultar perfil de organizador
Descripción	Ir a la pestaña de ajustes.
Resultado esperado	Se muestra la información del organizador.
Resultado obtenido	Correcto

Tabla 9.19: Descripción del CP19

CP20	Modificar perfil de organizador
Descripción	Ir a pestaña de ajustes, seleccionar modificar perfil y actualizar los datos.
Resultado esperado	El perfil de usuario es actualizado.
Resultado obtenido	Correcto

Tabla 9.20: Descripción del CP20

CP21	Ver listado de partidos abiertos organizados
Descripción	Ir a la pestaña de Home.
Resultado esperado	Se muestra un listado completo de los partidos abiertos.
Resultado obtenido	Correcto

Tabla 9.21: Descripción del CP21

CP22	Ver listado de partidos completos
Descripción	Ir a la pestaña de Partidos Completos.
Resultado esperado	Se muestra un listado completo de los partidos completos, pendientes de confirmación por el organizador.
Resultado obtenido	Correcto

Tabla 9.22: Descripción del CP22

CP23	Ver listado de partidos establecidos
Descripción	Ir a la pestaña de Partidos Establecidos.
Resultado esperado	Se muestra un listado completo de los partidos confirmados por el organizador.
Resultado obtenido	Correcto

Tabla 9.23: Descripción del CP23

CP24	Cambiar el estado de un partido
Descripción	Seleccionar un partido de la lista, seleccionar Cambiar Estado y seleccionar el estado deseado.
Resultado esperado	El estado del partido se actualiza al nuevo.
Resultado obtenido	Correcto

Tabla 9.24: Descripción del CP24

CP25	Crear partidos
Descripción	Ir a la pestaña de Crear partidos, rellenar el formulario con datos correctos y aceptar.
Resultado esperado	Se crea el partido con los datos recibidos.
Resultado obtenido	Correcto

Tabla 9.25: Descripción del CP25

CP26	Crear partidos II
Descripción	Ir a la pestaña de Crear partidos, rellenar el formulario con fecha de cierre posterior a fecha del partido y aceptar.
Resultado esperado	Se muestra un mensaje de error estableciendo que la fecha de cierre ha de ser al menos una hora anterior a la hora del partido.
Resultado obtenido	Correcto

Tabla 9.26: Descripción del CP26

CP27	Cerrar sesión organizador
Descripción	Ir a la pestaña de ajustes y seleccionar cerrar sesión.
Resultado esperado	La sesión de organizador se cierra y se redirecciona a la ventana de identificación.
Resultado obtenido	Correcto

Tabla 9.27: Descripción del CP27

Capítulo 10

Conclusiones

10.1. Valoración personal

A nivel personal este Trabajo de Fin de Grado ha sido muy enriquecedor y me ha permitido afianzar los conocimientos desarrollados a lo largo del grado de Ingeniería Informática y continuar con el aprendizaje en el desarrollo nativo de aplicaciones Android.

Además, me ha permitido planificar y documentar un proyecto software completamente, reforzando los conocimientos en Ingeniería del Software, evolucionando la idea inicial hasta el producto final y empleando LaTeX para el desarrollo de una memoria completa y profesional.

El desarrollo de este proyecto se presentaba como un reto complejo, ya que requería sincronización en tiempo real entre ambas aplicaciones para su correcto funcionamiento. El uso de Firebase ha simplificado en gran parte esta dificultad, permitiendo a las dos aplicaciones el acceso a una base de datos única y en tiempo real.

Destacar también que me ha permitido aprender a utilizar el patrón MVVM en Android, así como muchos de los componentes que ofrece Android Jetpack para el desarrollo de aplicaciones nativas, mantenibles, testeables y robustas.

Para finalizar, cabe resaltar que a nivel personal y educativo ha sido una experiencia muy provechosa, en la que se han conseguido los objetivos propuestos de manera satisfactoria.

10.2. Líneas futuras

Como trabajo futuro se podrían realizar las siguientes mejoras:

- Rediseño de la interfaz con las pautas marcadas por Google.
- Migración de la aplicación a Kotlin.
- Adición de test unitarios para realizar pruebas automáticas, rápidas y eficientes.
- Posible nueva funcionalidad de competiciones, de tal forma que un club pueda crear un torneo y los jugadores se puedan inscribir y seguir los resultados a través de la aplicación.
- Integración de Firebase Cloud Messaging para enviar notificaciones cada vez que se crea un partido y/o cuando se completa.
- Implementación de más filtros en la búsqueda de pistas.
- Implementación de un historial de partidos con resultados que permita a los jugadores tener un historial completo con estadísticas.

Capítulo 11

Bibliografía

- [1] Elena Sanz. *España es uno de los países europeos con más sedentarismo*. URL: <https://www.muyinteresante.es/salud/articulo/espana-es-uno-de-los-paises-europeos-con-mas-sedentarismo-331380271623>. (Último acceso: 15/03/2019).
- [2] Android Studio. *Documentación de Android Studio*. URL: <https://developer.android.com/studio>. (Último acceso: 14/02/2019).
- [3] Github. *Página de inicio de Github*. URL: <https://github.com/>. (Último acceso: 11/03/2019).
- [4] Overleaf. *Documentación de LaTeX*. URL: https://www.overleaf.com/learn/latex/Main_Page. (Último acceso: 14/06/2019).
- [5] Firebase. *¿Qué es firebase?* URL: <https://firebase.google.com/?hl=es-419>. (Último acceso: 14/03/2019).
- [6] Firebase. *Firebase Authentication*. URL: <https://firebase.google.com/docs/auth?hl=es-419>. (Último acceso: 15/03/2019).
- [7] Firebase. *Firebase Realtime Database*. URL: <https://firebase.google.com/docs/database?hl=es-419>. (Último acceso: 15/03/2019).
- [8] Firebase. *Firestore*. URL: <https://firebase.google.com/docs/firestore?hl=es-419>. (Último acceso: 25/05/2019).
- [9] Firebase. *Cómo elegir tu base de datos: Cloud Firestore o Realtime Database*. URL: <https://firebase.google.com/docs/firestore/rtdb-vs-firestore?hl=es-419>. (Último acceso: 15/03/2019).
- [10] Firebase. *Cloud Storage*. URL: <https://firebase.google.com/docs/storage?hl=es-419>. (Último acceso: 07/04/2019).
- [11] ArthurHub. *Android Image Cropper*. URL: <https://github.com/ArthurHub/Android-Image-Cropper>. (Último acceso: 06/04/2019).
- [12] Bump Technologies. *Glide*. URL: <https://bumptech.github.io/glide/>. (Último acceso: 07/04/2019).
- [13] Google. *Versiones de la plataforma*. URL: <https://developer.android.com/about/dashboards>. (Último acceso: 18/02/2019).
- [14] Softeng. *Proceso y roles de Scrum*. URL: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>. (Último acceso: 05/03/2019).
- [15] Firebase. *Firebase pricing*. URL: <https://firebase.google.com/pricing>. (Último acceso: 07/06/2019).
- [16] EPM. *The SCRUM Process*. URL: <https://expertprogrammanagement.com/2010/08/the-scrum-process/>. (Último acceso: 05/03/2019).

- [17] Firebase. *Announcing Architecture Components 1.0 Stable*. URL: <https://android-developers.googleblog.com/2017/11/announcing-architecture-components-10.html>. (Último acceso: 05/02/2019).
- [18] Android Developers. *Introducción a Android Jetpack*. URL: <https://developer.android.com/jetpack?hl=ES>. (Último acceso: 09/06/2019).
- [19] Android Developers. *Guía de arquitectura de apps*. URL: <https://developer.android.com/jetpack/docs/guide>. (Último acceso: 09/06/2019).
- [20] Resocoder. *Introduction to MVVM on android*. URL: <https://resocoder.com/2018/08/31/introduction-to-mvvm-on-android/>. (Último acceso: 03/04/2019).
- [21] Firebase. *Firestore and Android Jetpack: Fit Like a Glove - Doug Stevenson*. URL: <https://www.youtube.com/watch?v=WXc4adLMDqk>. (Último acceso: 05/02/2019).
- [22] Firebase. *Selecciona una estructura de datos*. URL: <https://firebase.google.com/docs/firestore/manage-data/structure-data>. (Último acceso: 13/05/2019).
- [23] Firebase. *Cloud Firestore Data Modeling (Google I/O'19)*. URL: <https://www.youtube.com/watch?v=1W7DWW2jST0>. (Último acceso: 18/05/2019).

Apéndice A

Manual de usuario

A.1. Introducción

En este apéndice se detallarán las funcionalidades desarrolladas para este proyecto mediante descripciones e imágenes ilustrativas.

A.2. Cribi - Aplicación para los jugadores

A.2.1. Identificación de usuarios

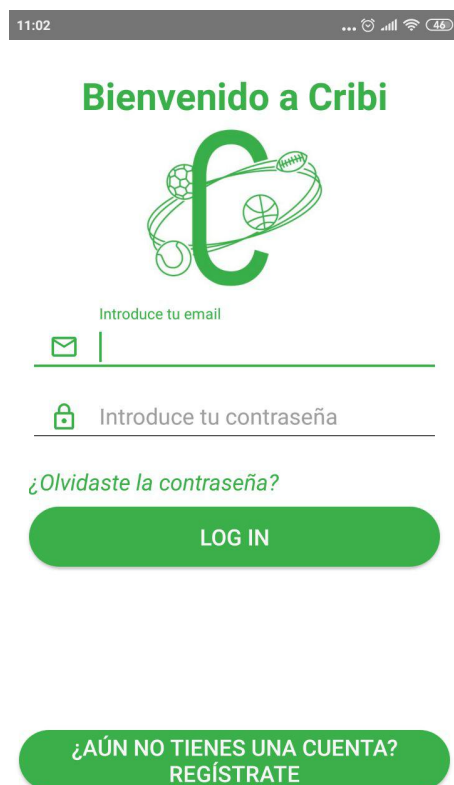


Figura A.1: Página de identificación.



Figura A.2: Fallo en identificación.

A.2.2. Registro de nuevos usuarios



Figura A.3: Página de registro de usuarios.

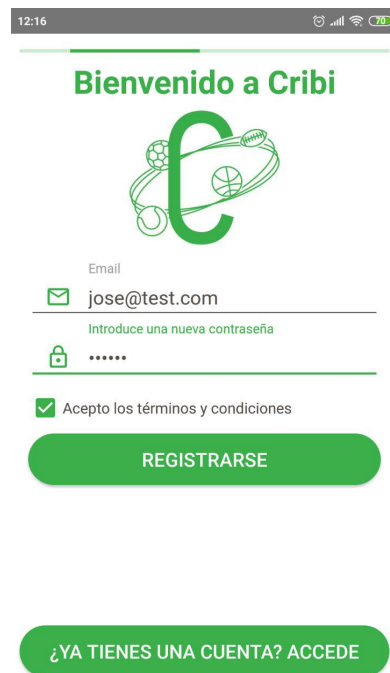


Figura A.4: Registrando nuevo usuario.

Si el registro falla, se mostrará un mensaje de error (Figura A.5). Si los datos son correctos, se redirige a la configuración del perfil (Figura A.6).



Figura A.5: Fallo al registrarse.

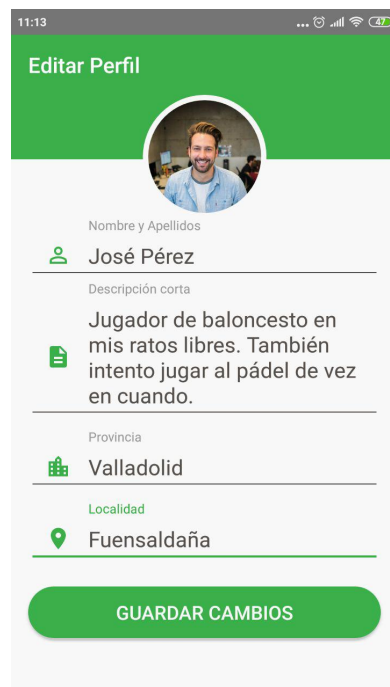


Figura A.6: Configurar perfil de usuario.

Una vez que se configura el perfil de usuario, se muestra la página principal.

A.2.3. Página principal

En esta pantalla podemos distinguir el menú de navegación en la Figura A.8. Como se puede observar, será un menú de navegación con tres niveles: Home, Mi Perfil y Mis Partidos.



Figura A.7: Página principal.



Figura A.8: Menú de navegación.

En esta navegación se disponen las siguientes opciones:

- **Home:** En esta opción, se mostrará un listado completo de todos los partidos disponibles y abiertos que los clubes han publicado. Más información en la subsección A.2.4.
- **Mi Perfil:** En esta opción, se mostrará el perfil del usuario y se dispondrá de ciertas opciones. Para más información, consultar la subsección A.2.11.
- **Mis Partidos:** En esta opción, se mostrará un listado de los partidos a los que el propio usuario se ha inscrito. Para más información, consultar la subsección A.2.14.

A.2.4. Home

En esta pantalla principal 'Home', se dispondrá un listado de los partidos disponibles (Figura A.9). Inicialmente estarán todos mezclados, pero se puede realizar un filtro dependiendo de lo que el usuario esté buscando. Para filtrar los partidos, se pulsaría el botón de la Figura A.10.

Si seleccionamos alguno de los partidos de la lista, se mostraría la información completa de ese partido. Para más información, en la subsección A.2.6.



Figura A.9: Listado de partidos.



Figura A.10: Filtrar partidos.

A.2.5. Filtrar partidos

Al pulsar dicho botón se despliega un modal en el cual se seleccionarán los diferentes filtros a aplicar (Figura A.11). Una vez que se seleccionan los filtros deseados, puede ser uno o varios, se mostrará un listado aplicando los filtros elegidos por el usuario (Figura A.12).

Al igual que en el listado anterior, si seleccionamos un partido de la lista, se mostrará la información completa de dicho partido.

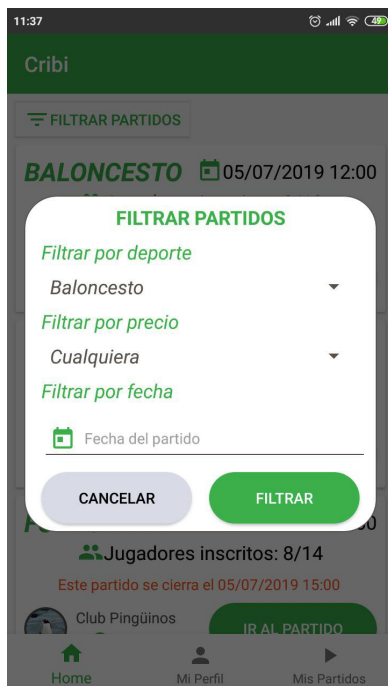


Figura A.11: Modal de selección de filtros.



Figura A.12: Listado de partidos filtrado.

A.2.6. Ver información completa del partido

La pantalla de información del partido, comprende dos ámbitos. Un ámbito informativo (Figura A.13) y un ámbito en el que podemos realizar ciertas acciones (Figura A.14).

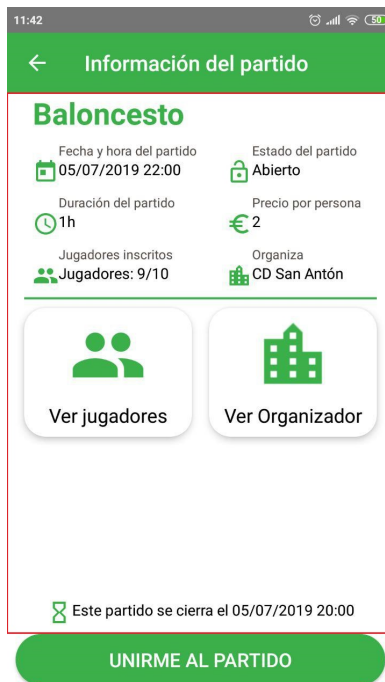


Figura A.13: Información del partido.



Figura A.14: Acciones disponibles.

En la información del partido se muestran todos los datos del mismo y se disponen dos botones. El primero de ellos es el de Ver Jugadores (Véase subsección A.2.7) en el que se muestran los jugadores inscritos al partido. En segundo lugar tenemos el botón de Ver Organizador (Véase subsección A.2.8) en el que se muestra el perfil del club y toda la información acerca del mismo.

Las opciones disponibles dependerán del estado del partido y si el jugador ya se ha inscrito o todavía no. En la Figura A.14 se muestra la opción de unirse al partido, puesto que el partido está abierto y no nos hemos inscrito aún.

Se mostrarán a continuación los demás casos posibles:

- Partido completo a la espera de confirmación del organizador (Figura A.15).
- Partido confirmado por el organizador (Figura A.16).
- Partido confirmado y jugado (Figura A.17).
- Partido cancelado (Figura A.18).

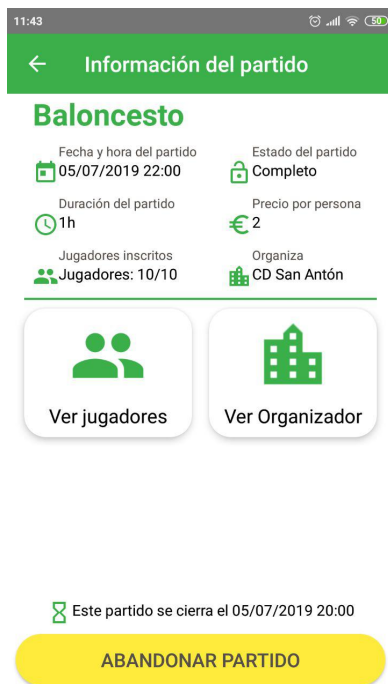


Figura A.15: Partido completo.



Figura A.16: Partido confirmado por el club.

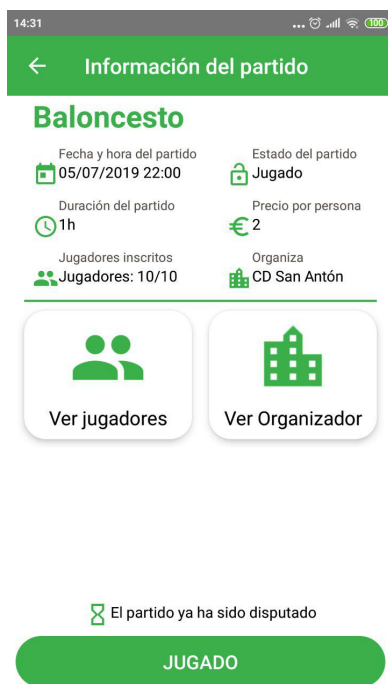


Figura A.17: Partido jugado.



Figura A.18: Partido cancelado.

A.2.7. Ver listado de jugadores inscritos

Si se selecciona ver los jugadores inscritos a un partido, se mostrará un listado con los propios jugadores. Si no hay jugadores inscritos, se mostrará en pantalla (Figura A.19). En cambio, si existen jugadores inscritos, se dispondrá un listado con la imagen de perfil y el nombre (Figura A.20).

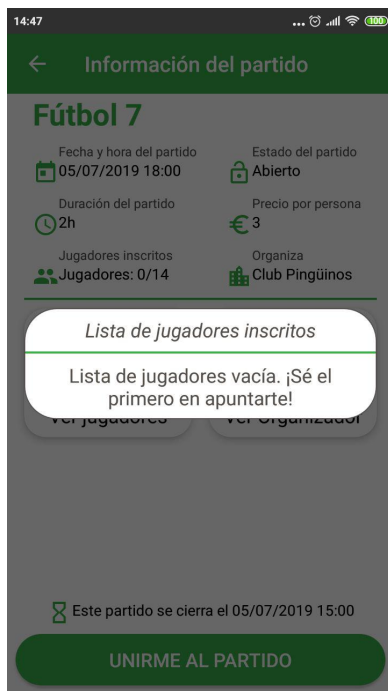


Figura A.19: Lista de jugadores vacía.

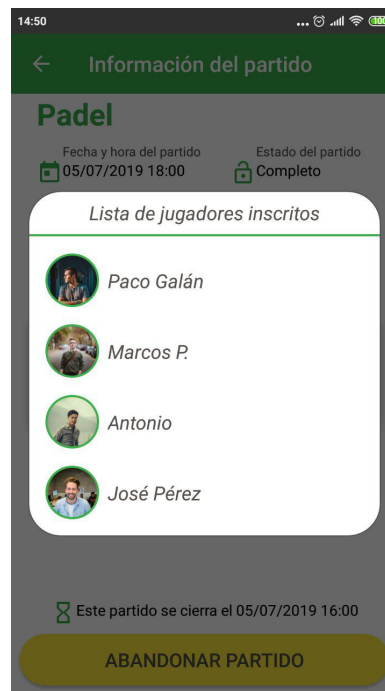


Figura A.20: Lista de jugadores inscritos.

A.2.8. Ver perfil de club organizador

Cuando seleccionamos Ver Organizador, se mostrará el perfil completo del organizador, en el que se distinguirá información de localización e información de contacto.

En la información de localización tendremos la dirección completa de las instalaciones, de tal forma que cualquier jugador que quiera disputar un partido, pueda encontrar la pista.

En la información de contacto, se mostrarán los datos de contacto del organizador, entre ellos el teléfono, el email y la web. De esta forma, cualquier jugador interesado podría contactar con el club organizador para solicitar cualquier cosa que considere oportuna.

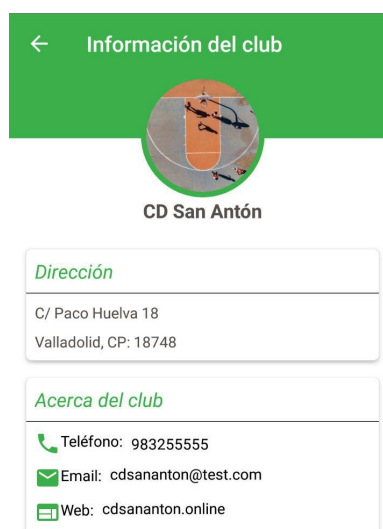


Figura A.21: Perfil del Organizador.

A.2.9. Unirse a partido

Se mostrará a continuación, el ciclo de unión a partido. En primer lugar, si se desea inscribirse en un partido, el partido debe estar abierto. En cualquier otro estado no será posible la unión al mismo. Además, se comprobará si el usuario se ha inscrito a un partido a la misma hora, en este caso, se mostrará un mensaje de error.

Para unirse a un partido, tendremos que acceder a la pantalla de información de un partido y pulsar el botón de 'Unirme al partido' descrito en la Figura A.14. Una vez que se pulsa el botón, se mostrará un diálogo para confirmar dicha unión (Figura A.22).

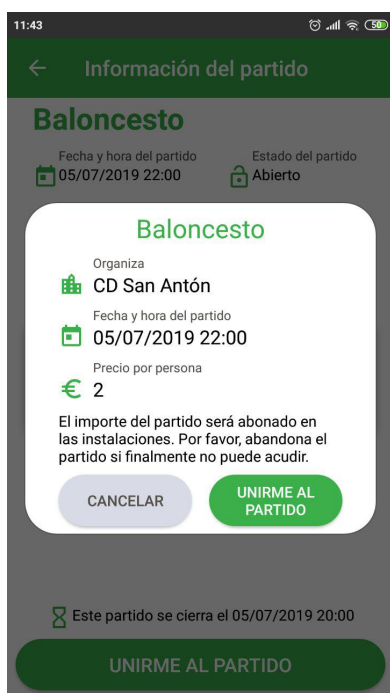


Figura A.22: Diálogo de unión a partido.

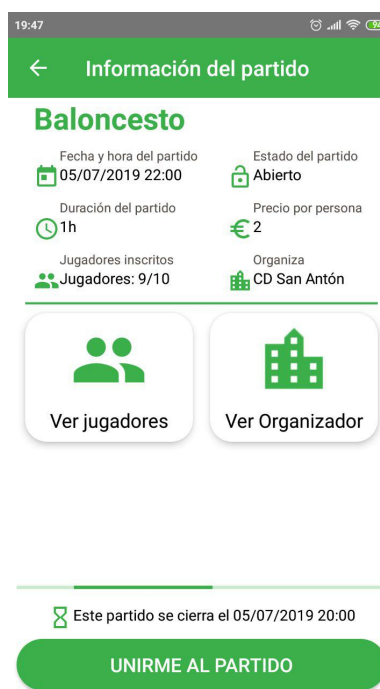


Figura A.23: Uniéndose al partido.

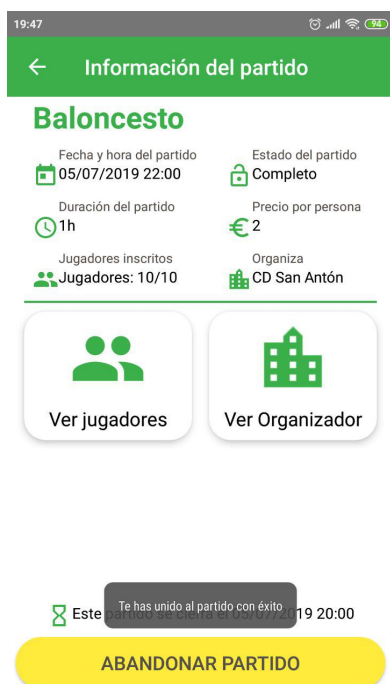


Figura A.24: Éxito al unirse al partido.



Figura A.25: Fallo al unirse al partido.

A.2.10. Abandonar partido

El ciclo para abandonar un partido es similar al de unión. En primer lugar, si se desea abandonar un partido, no se puede sobrepasar el tiempo límite de cierre del mismo.

Para abandonar un partido, tendremos que acceder a la pantalla de información de un partido y pulsar el botón de 'Abandonar partido' mostrado en la Figura A.15. Una vez que se pulsa el botón, se mostrará un diálogo para confirmar dicho abandono (Figura A.26).

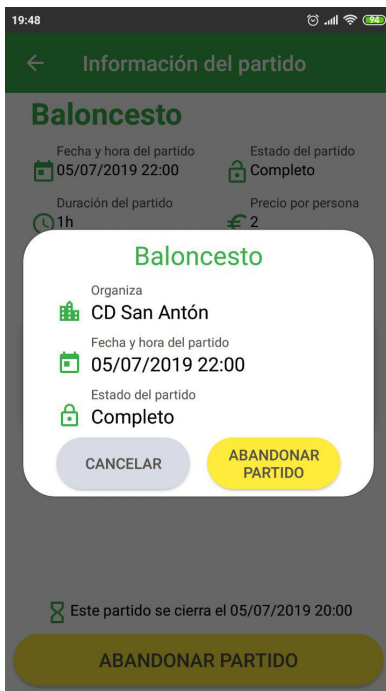


Figura A.26: Diálogo para abandonar un partido.



Figura A.27: Abandonando partido.

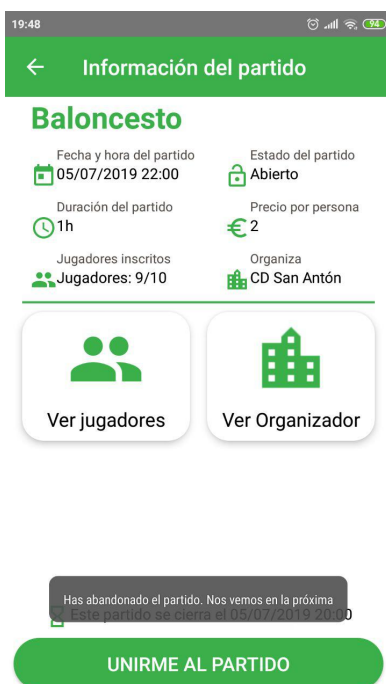


Figura A.28: Éxito al abandonar el partido.

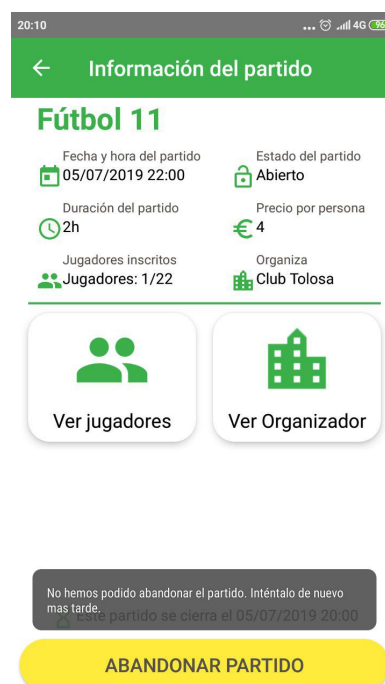


Figura A.29: Fallo al abandonar partido.

A.2.11. Perfil de usuario

En el perfil de usuario se mostrará la información del propio usuario, los partidos que ha jugado en la plataforma, y se mostrarán las opciones para modificar el perfil y cerrar sesión, descritas en las secciones A.2.12 y A.2.13 respectivamente.

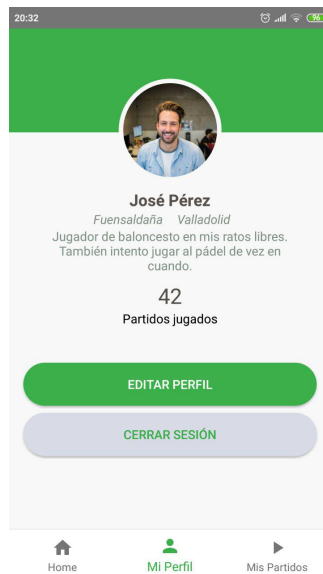


Figura A.30: Perfil de usuario.

A.2.12. Editar perfil de usuario

Si seleccionamos la opción de 'Editar Perfil' se mostrará una ventana con los datos del usuario actualmente, que se podrán modificar y si se pulsa sobre la imagen, se abrirá un selector de imágenes para añadir una nueva (Figura A.32). Esta nueva imagen tendrá un formato cuadrado que se adaptará perfectamente a la interfaz.

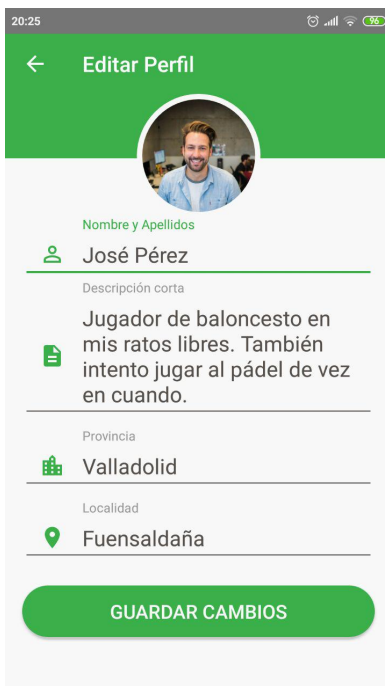


Figura A.31: Editar perfil de usuario.

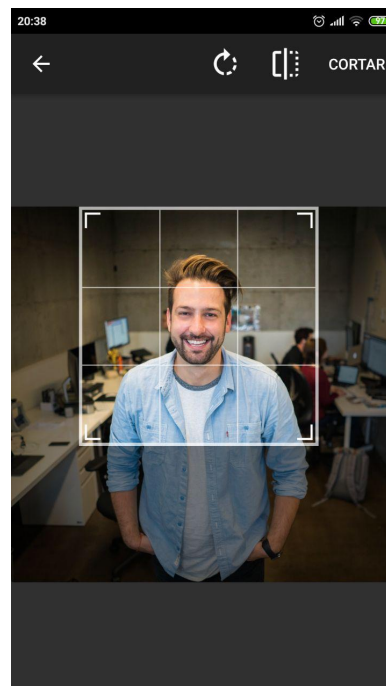


Figura A.32: Añadir imagen de perfil.

A.2.13. Cerrar sesión

La opción de cerrar sesión, como no podía ser de otra manera, cierra la sesión y redirige al usuario a la pantalla de identificación.

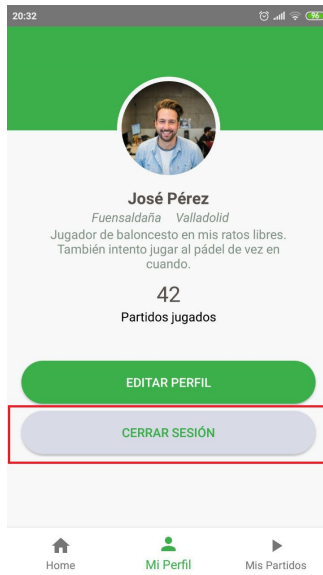


Figura A.33: Cerrar sesión de usuario.

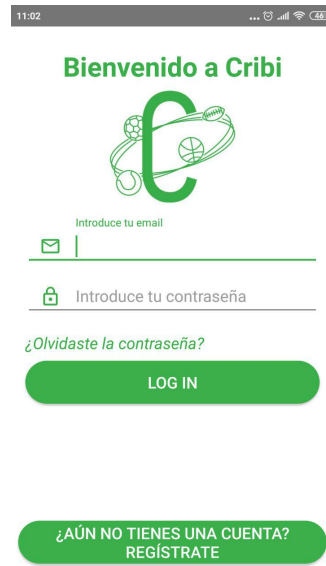


Figura A.34: Cierre de sesión completo.

A.2.14. Listado de partidos inscritos

Esta opción del menú aporta al usuario un listado de los partidos a los que se ha inscrito previamente, de esta forma, el acceso a ellos es más rápido.

Al igual que en el listado de partidos de la pantalla principal, podemos acceder a la información de los partidos y realizar los acciones pertinentes.



Figura A.35: Listado de partidos inscritos.

A.3. Cribi Clubs - Aplicación para Organizadores

A.3.1. Identificación de Clubs

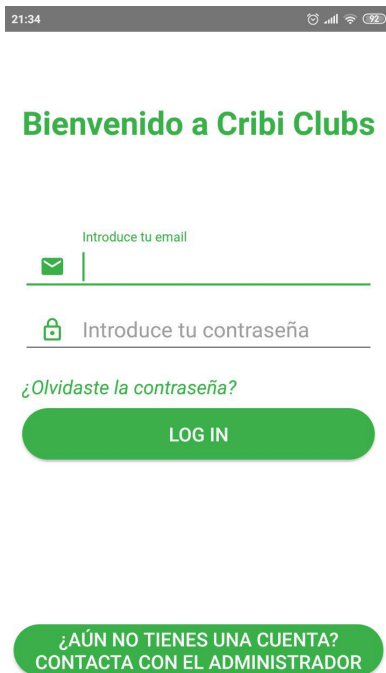


Figura A.36: Página de identificación de clubes.



Figura A.37: Fallo en identificación de club.

A.3.2. Página principal Cribi Clubs

Si el acceso es satisfactorio, se muestra la página de principal de la aplicación. En esta página se muestra un listado de todos los partidos en estado abierto, para que el organizador sepa en todo momento qué partidos están cerca de llegar a la fecha de cierre. En la Figura A.39 se muestra el menú de navegación que se comentará a continuación.



Figura A.38: Página principal Cribi Clubs.

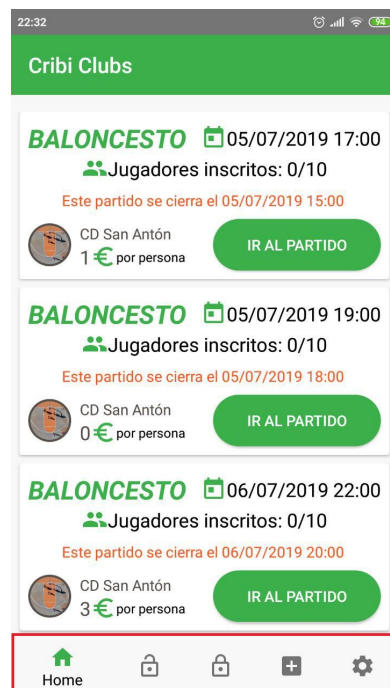


Figura A.39: Menú de navegación Cribi Clubs.

Este menú de navegación cuenta con cinco niveles para facilitar la gestión de las pistas al organizador:

- **Home:** En esta opción se muestra un listado con los partidos creados por el propio organizador y que se encuentran en estado Abierto. Es decir, esperando a que se unan jugadores.
- **Completo:** En esta opción se muestra un listado con los partidos creados por el propio organizador y que se encuentran en estado Completo. Es decir, se han unido suficientes jugadores al partido y se está esperando a que el organizador confirme el partido.
- **Establecido:** En esta opción se muestra un listado con los partidos creados por el propio organizador y que se encuentran en estado Establecido. Es decir, se ha confirmado el partido y se está esperando a que se juegue.
- **Nuevo Partido:** En esta opción se permite al organizador crear un nuevo partido. Más información en la sección A.3.5.
- **Ajustes:** En esta opción el organizador puede ver su perfil, editarlo y cerrar sesión. Más información en la sección A.3.6.

En cualquiera de los listados de partidos anteriores, se permite el acceso a la información completa del partido, donde se permitirá gestionar el estado de los mismos. Para más información, en la sección A.3.3.

A.3.3. Ver información del partido

De forma análoga a la aplicación de los jugadores, se muestra la información de partido. La funcionalidad del listado de jugadores es similar, por lo tanto, se recomienda ver la sección A.2.7 del manual de usuario de Cribi.

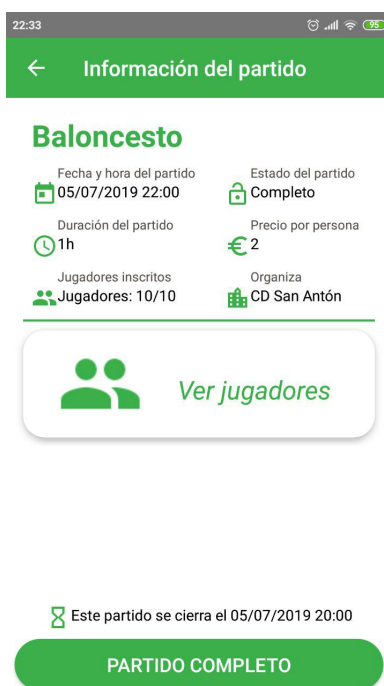


Figura A.40: Información del partido

Una de las diferencias de esta pantalla respecto a la de jugadores, es que no dispondremos de la opción de ver el organizador puesto que es el propio usuario.

La otra diferencia, es en las posibles acciones a realizar. En este caso, se podrá cambiar el estado del partido como se muestra en la siguiente sección.

A.3.4. Cambiar estado del partido

Cambiar el estado de un partido es crítico para un organizador. Es por ello que se detallarán todos y cada uno de los posibles estados y a qué estados pueden cambiarse:

- **Partido abierto:** Estado en el cual se están buscando nuevos jugadores para que se unan al partido. El cambio es automático a partido completo cuando se unen los jugadores necesarios. Aún así, el organizador puede cancelar el partido.
- **Partido completo:** Estado en el cual se ha unido el número de jugadores requerido para disputar el partido. El organizador podrá cambiar el estado a establecido, o bien cancelarlo.
- **Partido establecido:** Estado en el cual el organizador ha confirmado el partido y se está esperando a disputarse. Puede cambiar a estado jugado y a cancelado.
- **Partido jugado:** Estado definitivo y que no puede cambiarse.
- **Partido cancelado:** Estado definitivo y que no puede cambiarse.

El partido se cambiará utilizando el siguiente diálogo, en el cual el organizador seleccionará el nuevo estado de un Spinner.

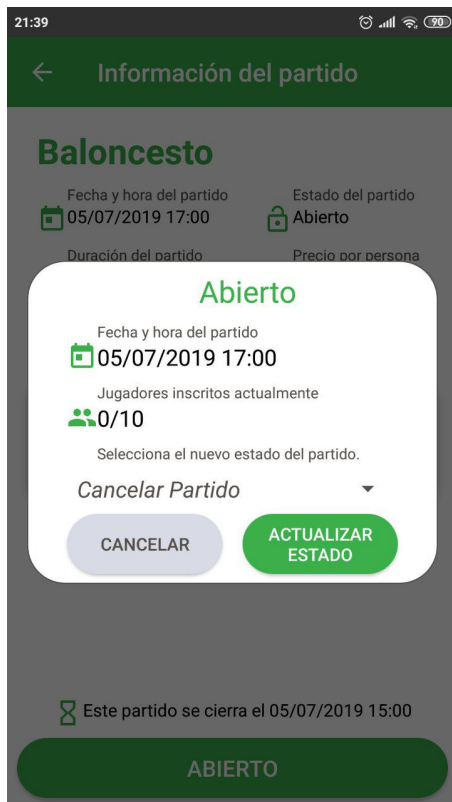


Figura A.41: Cambiar estado de partido.

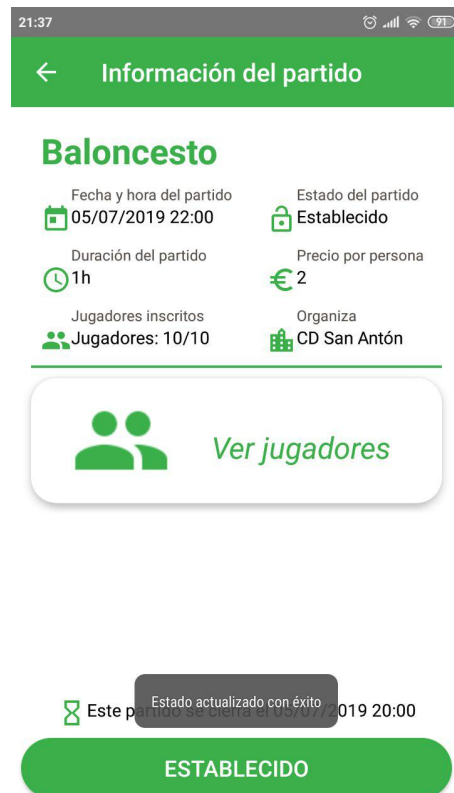


Figura A.42: Cambio de estado exitoso.

A.3.5. Crear nuevo partido

Esta funcionalidad permitirá al organizador crear nuevos partidos seleccionando el deporte, las fechas, la duración y el precio del mismo.

La fecha de cierre no podrá ser posterior a la fecha del partido y tendrá que dejar un margen de una hora entre ambas para que el administrador pueda confirmar el partido en el caso de que se una un jugador a última hora.

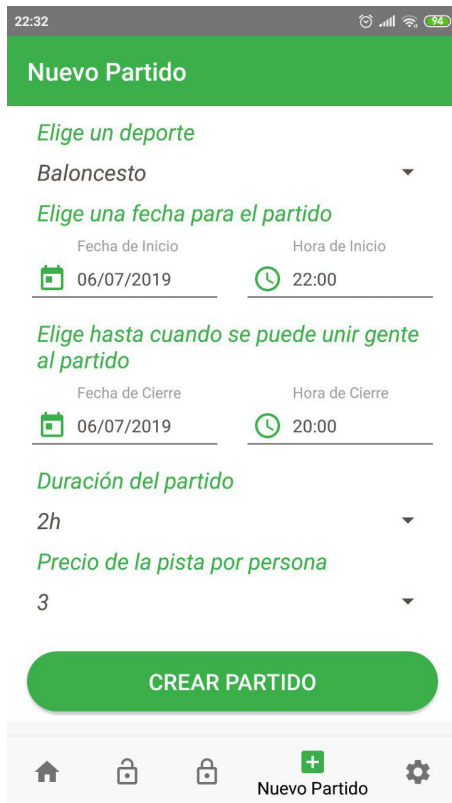


Figura A.43: Crear nuevo partido.

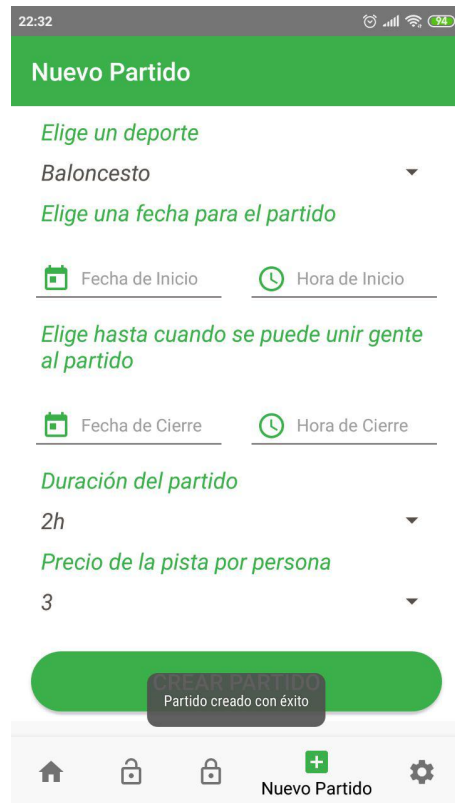


Figura A.44: Creación de partido con éxito.

A.3.6. Ajustes

En los ajustes se mostrará el perfil del club tal y como lo ven los jugadores cuando lo visitan. Además se añaden las opciones de editar perfil (Veáse sección A.3.7) y la opción de cerrar sesión que redirigirá a la pantalla de inicio.



Figura A.45: Ajustes Cribi Clubs.

A.3.7. Editar Perfil

De forma análoga a la edición del perfil de jugadores, los organizadores también dispondrán de esta funcionalidad. Se podrá aportar el nombre del club, la dirección completa, y la información de contacto relevante.

Además, contará con un selector de imágenes en formato cuadrado que permitirá seleccionar el logo del club para una correcta adaptación a las distintas pantallas de Cribi.

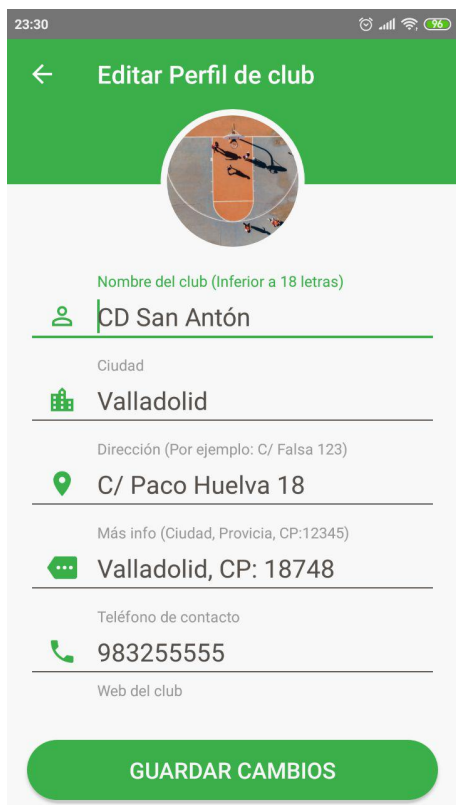


Figura A.46: Editar perfil de club.

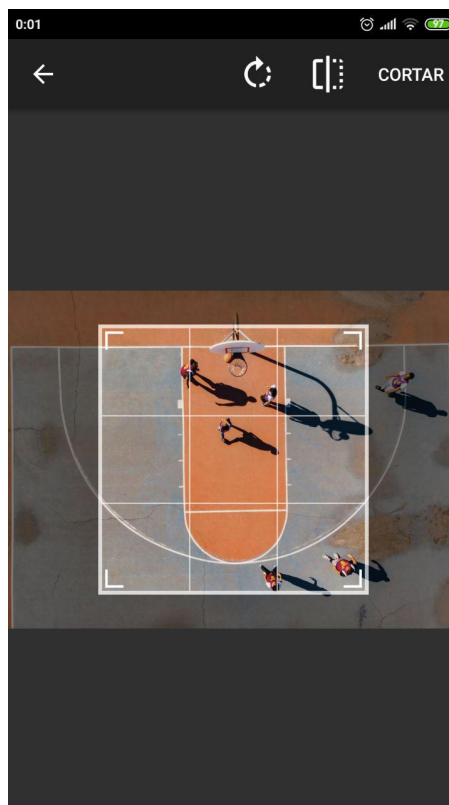


Figura A.47: Selector de logo del club.

Apéndice B

Manual de instalación

Para la instalación de la aplicación, dependerá si se dispone del ejecutable apk, o si se necesita compilar e instalar a través del proyecto de Android Studio.

En el primer caso, si disponemos del ejecutable apk, los pasos a seguir serán los mismos de cualquier aplicación Android, siendo necesarios los siguientes requisitos:

- Dispositivo Android con versión mínima de 4.1.
- Tener activada la opción de «Orígenes desconocidos» en el dispositivo Android accesible desde el menú de Ajustes.
- Disponer de, al menos, 20 Mb de almacenamiento disponibles.

En el segundo caso, el proyecto cuenta con todo lo necesario para la correcta compilación y ejecución del mismo. Simplemente se tendría que abrir Android Studio, importar el proyecto adjunto en el soporte digital, compilarlo y ejecutarlo.

Apéndice C

Contenido del soporte digital

El contenido del soporte digital entregado es el siguiente:

- **cribi:** carpeta con el proyecto de Android Studio de la aplicación Cribi. Contiene todos los ficheros necesarios para importar y compilar la aplicación.
- **cribiclubs:** carpeta con el proyecto de Android Studio de la aplicación Cribi Clubs. Contiene todos los ficheros necesarios para importar y compilar la aplicación.
- **memoria.pdf:** Documento de la memoria del TFG.
- **Cribi.apk:** Contiene el ejecutable de la aplicación Cribi.
- **CribiClubs.apk:** Contiene el ejecutable de la aplicación Cribi Clubs.