



Universidad Politécnica
de Madrid



**Escuela Técnica Superior de
Ingenieros Informáticos**

Grado en Ingeniería Informática

Trabajo Fin de Grado

**Desarrollo de una Aplicación Social para
la Organización de Partidos y Torneos
Deportivos Amateurs**

Autor: Nerea Tanarro Ferrero

Tutor(a): Fernando Pérez Costoya

Madrid, Enero 2026

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Ingeniería Informática

Título: Desarrollo de una Aplicación Social para la Organización de Partidos
y Torneos Deportivos Amateurs

Enero 2026

Autor: Nerea Tanarro Ferrero

Tutor:

Fernando Pérez Costoya

Departamento de Arquitectura y Tecnología de Sistemas Informáticos
(DATSI)

ETSI Informáticos

Universidad Politécnica de Madrid

Agradecimientos

Antes de poner fin a esta etapa de mi vida, donde dejo mucho esfuerzo y sacrificio, así como risas y anécdotas, me gustaría agradecer a todas aquellas personas que me han acompañado durante este camino.

Quiero agradecer a mi familia y a mis amigos por acompañarme y apoyarme durante esta etapa de mi vida. Sin su comprensión, paciencia y ánimo constante, no hubiese sido posible alcanzar este objetivo.

A todos mis amigos y a todas las personas que he tenido la oportunidad de conocer a lo largo de estos últimos años, quienes, de una forma u otra, han contribuido a mi crecimiento académico y, sobre todo a mi crecimiento personal, brindándome apoyo y momentos inolvidables.

A la universidad, por ofrecerme la oportunidad de adquirir los conocimientos y competencias necesarias para una futura vida profesional. En especial, agradecer a todos aquellos profesores y profesoras que, gracias a su dedicación, disponibilidad y constante preocupación por nuestro aprendizaje, han logrado, con su esfuerzo, formarnos y prepararnos como los mejores ingenieros e ingenieras, y, sobre todo, como mejores personas.

Y, por último, de manera muy especial, quiero agradecer a mi madre todo el esfuerzo que ha realizado para que su hija pudiera lograr todo lo que se propone. Gracias por ser un pilar imprescindible en este camino, por acompañarme en cada paso, por creer en mí incluso cuando yo misma dudaba, y por ofrecerme siempre tu apoyo incondicional. Todo esto no habría sido posible sin ti.

Resumen

El presente trabajo tiene como objetivo el diseño e implementación de una aplicación móvil y web denominada “SportCrew”, orientada a deportistas amateurs, que permita organizar de forma sencilla partidos, torneos y actividades deportivas con otras personas.

Pretende ser una herramienta que combine la gestión deportiva (reservas, organización de equipos y torneos) con una fuerte apuesta por la interacción social entre personas con los mismos gustos deportivos, que buscan desconectar de sus rutinas practicando lo que más les gusta: el deporte.

Esta propuesta parte de la idea de ofrecer una plataforma unificada y versátil que abarque varios de los deportes más practicados actualmente, entre ellos el fútbol, el baloncesto, el tenis y el pádel; y que solvete las limitaciones y carencias observadas en aplicaciones actuales, generalmente enfocadas a un único deporte o carentes de funcionalidades colaborativas.

El proyecto se desarrolla bajo un enfoque ágil e incremental, priorizando la construcción de un MVP – “Minimum Viable Product”, que permita demostrar las funcionalidades principales en un entorno real. Para ello, se ha llevado a cabo un análisis previo de requisitos, definiendo las funcionalidades principales y la estructura de los datos necesarios para su correcto funcionamiento. A partir de este análisis, se ha diseñado la arquitectura de la aplicación, prestando especial atención a la escalabilidad, la organización de información y la experiencia de usuario. Para cumplir con estos objetivos, se apuesta por el uso de tecnologías multiplataforma como React Native y Firebase, que facilitan el desarrollo, el mantenimiento y la escalabilidad de la aplicación.

Con respecto al desarrollo de SportCrew, se ha incluido la implementación de un sistema de autenticación de usuarios, la gestión de perfiles, un sistema de mensajería, la creación y búsqueda de partidos deportivos, así como la organización de torneos en distintos deportes y modalidades. Además, se ha diseñado una estructura que permite el seguimiento del estado de los partidos y torneos, la gestión de inscripciones y la visualización de resultados y clasificaciones.

Para la persistencia de los datos se ha utilizado una base de datos en la nube, lo que permite el acceso en tiempo real a la información, garantizando la sincronización entre los distintos usuarios de la plataforma. A su vez, se ha puesto especial énfasis en el desarrollo del frontend, buscando una interfaz clara, accesible, intuitiva y fácil de usar, que mejore la interacción del usuario con la aplicación.

Este trabajo finaliza con la verificación del correcto y esperado funcionamiento de la aplicación mediante pruebas funcionales, comprobando que SportCrew presenta una solución viable y completa para la organización de actividades deportivas.

Abstract

The objective of this project is to design and implement a mobile and web application called “SportCrew,” aimed at amateur athletes, which allows them to easily organize games, tournaments, and sports activities with other people.

It aims to be a tool that combines sports management (bookings, team and tournament organization) with a strong focus on social interaction between people with the same sporting interests, who are looking to disconnect from their routines by practicing what they love most: sports.

This proposal is based on the idea of offering a unified and versatile platform that covers several of the most popular sports today, including soccer/football, basketball, tennis, and paddle tennis, and that addresses the limitations and shortcomings observed in current applications, which are generally focused on a single sport or lack collaborative features.

The project is being developed using an agile and incremental approach, prioritizing the construction of an MVP (Minimum Viable Product) that will demonstrate the main functionalities in a real environment. To this end, a preliminary requirements analysis has been carried out, defining the main functionalities and the structure of the data necessary for its correct operation. Based on this analysis, the architecture of the application has been designed, paying special attention to scalability, information organization, and user experience. To meet these objectives, we are committed to the use of multi-platform technologies such as React Native and Firebase, which facilitate the development, maintenance, and scalability of the application.

Regarding the development of SportCrew, a user authentication system, profile management, a messaging system, the creation and search for sports matches, and the organization of tournaments in different sports and modalities have been included. In addition, a structure has been designed that allows the status of matches and tournaments to be tracked, registrations to be managed, and results and rankings to be viewed.

A cloud database has been used for data persistence, allowing real-time access to information and ensuring synchronization between the different users of the platform. At the same time, special emphasis has been placed on frontend development, seeking a clear, accessible, intuitive, and easy-to-use interface that improves user interaction with the application.

This project concludes with verification of the correct and expected functioning of the application through functional testing, confirming that SportCrew presents a viable and comprehensive solution for the organization of sporting activities.

Tabla de contenidos

1	Introducción	1
1.1	Objetivos	1
2	Planificación del TFG	3
3	Trabajos previos y análisis de aplicaciones existentes	4
3.1	Análisis de las aplicaciones existentes.....	4
3.1.1	CeleBreak	4
3.1.2	Playtomic.....	5
3.1.3	PickApp	5
3.2	Análisis de alternativas tecnológicas	6
4	Stack de desarrollo. Tecnologías utilizadas	7
4.1	Entorno de desarrollo.....	7
4.2	Frontend	7
4.2.1	Navegación y estructura de pantallas	8
4.2.2	Interfaz de usuario y estilos	9
4.3	Backend.....	9
4.3.1	Autenticación de usuarios	10
4.3.2	Base de datos en la nube	10
4.4	Control de versiones.....	11
4.5	Herramientas para pruebas.....	11
4.6	Herramientas de apoyo al desarrollo (extensiones IDE).....	12
4.7	Herramienta de apoyo al diseño: Figma.....	12
5	Fase de Análisis	13
5.1	Requisitos Funcionales	13
5.2	Requisitos No Funcionales	14
6	Fase de Diseño	16
6.1	Tipo de arquitectura.....	16
6.1.1	Model.....	16
6.1.2	View.....	16
6.1.3	ViewModel	17
6.1.4	Esquema del flujo de datos	17
6.2	Enfoque ágil e incremental	17
6.3	Diseño Backend	17
6.4	Diseño Base de Datos.....	18
6.4.1	Users – Usuarios.....	18
6.4.2	Facilities – Instalaciones	18
6.4.3	Teams – Equipos.....	18
6.4.4	Matches – Partidos.....	19
6.4.5	Tournaments – Torneos	20

6.4.6	Notifications – Notificaciones	20
6.4.7	Chats.....	20
6.4.7.1	Messages – Mensajes.	21
6.5	Diseño Frontend – Prototipo Figma	21
6.5.1	Inicio de sesión de un usuario	22
6.5.2	Registro y verificación de un nuevo usuario	23
6.5.3	Creación de un partido	24
6.5.4	Búsqueda de un torneo.....	25
6.5.5	Creación de un chat grupal.....	26
6.5.6	Envío de mensajes por un chat individual	27
6.5.7	Visualización de las 4 pantallas de la sección “Actividad”	28
6.5.8	Visualización del perfil del usuario con el correspondiente cierre de sesión 29	
7	Fase de Desarrollo	30
7.1	Principales Funcionalidades de SportCrew	30
7.2	Logo de SportCrew	30
7.3	Pantallas Desarrolladas.....	31
7.3.1	Pantalla de inicio de sesión.....	31
7.3.2	Pantalla ¿Has olvidado tu contraseña?	32
7.3.3	Pantalla de Registro	32
7.3.4	Pantalla principal	34
7.3.4.1	Pantalla barra de búsquedas	35
7.3.5	Pantalla de cada botón de los deportes	37
7.3.6	Creación de Partidos.....	38
7.3.7	Creación de Torneos	38
7.3.8	Búsqueda de Partidos.....	39
7.3.9	Búsqueda de Torneos	39
7.3.10	Pantalla Actividad	40
7.3.11	Chats	41
7.3.12	Perfil de usuario.....	43
8	Pruebas y validaciones.....	45
9	Resultados y conclusiones	46
9.1	Resultados	46
9.2	Conclusiones.....	46
9.3	Trabajos Futuros.....	46
10	Análisis de Impacto	47
10.1	Impacto personal.....	47
10.2	Impacto empresarial.....	47
10.3	Impacto social	47

10.4	Impacto económico.....	47
10.5	Impacto medioambiental	47
10.6	Impacto cultural.....	47
10.7	Objetivos de Desarrollo Sostenible.....	47
11	Bibliografía	48
12	Anexos	50

Índice de ilustraciones

Ilustración 1: Diagrama de Gantt.....	3
Ilustración 2: CeleBreak.....	4
Ilustración 3: Playtomic	5
Ilustración 4: PickApp.....	5
Ilustración 5: React Native	7
Ilustración 6: Expo Go	8
Ilustración 7: TypeScript.....	8
Ilustración 8: Estructura de SportCrew utilizando Expo Router	9
Ilustración 9: Ejemplo de un componente nativo proporcionado por React Native	9
Ilustración 10: Firebase Authentication	10
Ilustración 11: Cloud Firestore.....	10
Ilustración 12: Git y GitHub.....	11
Ilustración 13: Android Studio	11
Ilustración 14: Figma	12
Ilustración 15: Visión General del Prototipo en Figma	21
Ilustración 16: Inicio de sesión en Figma (1).....	22
Ilustración 17: Inicio de sesión en Figma (2).....	22
Ilustración 18: Registro y verificación de un nuevo usuario en Figma	23
Ilustración 19: Creación de un partido en Figma (1).....	24
Ilustración 20: Creación de un partido en Figma (2).....	24
Ilustración 21: Creación de un partido en Figma (3).....	25
Ilustración 22: Búsqueda de un torneo en Figma (1)	25
Ilustración 23: Búsqueda de un torneo en Figma (2)	25
Ilustración 24: Creación de un chat grupal en Figma (1)	26
Ilustración 25: Creación de un chat grupal en Figma (2)	26
Ilustración 26: Envío de mensajes por un chat individual en Figma (1)	27
Ilustración 27: Envío de mensajes por un chat individual en Figma (2)	27
Ilustración 28: Envío de mensajes por un chat individual en Figma (3)	28
Ilustración 29: Visualización de las 4 pantallas de la sección "Actividad" en Figma.....	28
Ilustración 30: Visualización del perfil del usuario y cierre de sesión en Figma	29
Ilustración 31: Logo de SportCrew	30
Ilustración 32: Pantalla de inicio de sesión	31
Ilustración 33: Recuperar contraseña	32
Ilustración 34: Correo para resetear la contraseña.....	32
Ilustración 35: Pantalla para crear una cuenta	33
Ilustración 36: Pantalla informativa, verificación de la cuenta.....	33
Ilustración 37: Correo recibido para verificar la cuenta	34
Ilustración 38: Pantalla de inicio	35
Ilustración 39: Barra de búsqueda (1)	35
Ilustración 40: Barra de búsqueda (2)	36
Ilustración 41: Perfil público de un usuario	36
Ilustración 42: Información de una instalación	36
Ilustración 43: Pantalla de los botones de cada deporte	37
Ilustración 44: Pantalla de los partidos de un deporte.....	37
Ilustración 45: Pantalla crear partidos	38
Ilustración 46: Pantalla crear torneos	39
Ilustración 47: Búsqueda de Partidos.....	39
Ilustración 48: Búsqueda de Torneos	40
Ilustración 49: Inscripción sin equipo creado previamente	40

Ilustración 50: Apartado de Actividad	41
Ilustración 51: Colección de chats individuales	42
Ilustración 52: Colección de chats grupales	42
Ilustración 53: Formulario para crear chat grupal.....	42
Ilustración 54: Chat grupal con mensajes de sistema.....	43
Ilustración 55: Perfil del usuario registrado.....	44

1 Introducción

En los últimos años, la práctica del deporte amateur ha sufrido un gran crecimiento, impulsado por una mayor concienciación sobre la importancia de la actividad física para mantener una buena salud física y mental, así como el bienestar personal. Al mismo tiempo, la digitalización ha impactado en la forma en la que las personas se organizan y se relacionan, dando lugar a múltiples aplicaciones orientadas a la gestión de actividades deportivas.

No obstante, en la práctica diaria, la organización de partidos y torneos entre deportistas amateurs continúa presentando importantes dificultades. En muchas ocasiones, y especialmente en deportes de equipo como el fútbol donde implica ser un amplio número de personas, resulta complejo reunir al número necesario de participantes. Esto obliga a contactar con muchas personas, buscar conocidos o depender de terceros para encontrar jugadores disponibles. Además, muchas veces, esta tarea se ve dificultada por la falta de contactos o por la incompatibilidad de horarios entre los posibles participantes, llegando en ocasiones a impedir la realización de la actividad deportiva.

Esta problemática, observada de manera recurrente en mi entorno cercano, ha sido una de las principales motivaciones para realizar el desarrollo de este proyecto. En ese momento, surge la idea de crear una aplicación que facilite, en la medida de lo posible, la creación y organización de partidos y torneos deportivos, reduciendo en la medida de lo posible, las barreras existentes y fomentando la participación en actividades deportivas de forma más accesible y eficiente.

1.1 Objetivos

El objetivo principal de este Trabajo de Fin de Grado es el diseño e implementación de una aplicación funcional que permita organizar y gestionar actividades deportivas de forma sencilla, eficiente y accesible para deportista amateurs, poniendo especial énfasis en la importancia de las interacciones sociales en el contexto de la digitalización, creando una buena capa social para facilitar la conexión entre personas con los mismos gustos deportivos.

De este objetivo principal surgen los siguientes objetivos más específicos:

- Analizar las necesidades de los usuarios y definir los requisitos funcionales y no funcionales de la aplicación.
- Diseñar un prototipo mediante la herramienta Figma, con el correspondiente comportamiento del flujo de pantallas, alineado con las necesidades analizadas y los requisitos definidos, que sirva como base para el desarrollo de la interfaz y la experiencia de usuario.
- Diseñar la arquitectura de la aplicación, teniendo en cuenta los criterios de escalabilidad, mantenibilidad y experiencia de usuario.
- Implementar un sistema de autenticación y gestión de perfiles de usuario.
- Desarrollar funcionalidades para la creación y búsqueda de partidos deportivos.
- Implementar un sistema de organización y gestión de torneos en distintas modalidades y deportes.
- Integrar funcionalidades sociales como la mensajería entre usuarios, un sistema de seguidores y seguidos, y la creación de equipos deportivos, con el objetivo de conectar a usuarios con los mismos gustos deportivos.
- Desarrollar una interfaz intuitiva, coherente y funcional que facilite la interacción con la aplicación y que, además, permita la integración de

nuevas funcionalidades sin comprometer la coherencia visual ni la usabilidad.

- Garantizar la persistencia y la sincronización de los datos mediante el uso de una base de datos en la nube, más concretamente mediante Firebase.
- Validar el correcto funcionamiento de la aplicación mediante pruebas funcionales.

2 Planificación del TFG

Para alcanzar el éxito esperado, es imprescindible una adecuada planificación de las tareas a realizar durante el proyecto. A continuación, se muestra la planificación temporal del desarrollo de este Trabajo de Fin de Grado, representada mediante un diagrama de Gantt. Gracias a este diagrama, se puede visualizar de forma sencilla y clara la distribución de las distintas tareas que han conformado el proyecto.

El eje horizontal del diagrama se muestra el número de semanas que ha durado la realización de este proyecto, un periodo total de 16 semanas. Estas semanas se encuentran comprendidas entre el 25 de septiembre de 2025 y el 15 de enero de 2026.

En el eje vertical se puede observar las principales tareas y las fases en las que se ha dividido el desarrollo del proyecto.

Las tareas realizadas han sido las siguientes:

- Análisis inicial y documentación, donde se ha definido el alcance del proyecto y se ha elaborado un estudio de aplicaciones existentes del mercado, así como de las alternativas tecnológicas para el desarrollo de la aplicación.
- Diseño de la arquitectura y el modelo de datos, donde se han definido los requisitos funcionales y no funcionales de la aplicación, junto con un boceto de las colecciones y campos que se van a almacenar en la base de datos.
- Diseño del prototipo y el flujo de pantallas mediante la herramienta Figma.
- Desarrollo del frontend y del backend, seguido de la integración de la capa social, una de las funcionalidades clave de la aplicación.
- Pruebas y correcciones, una fase específica dedicada a validar el correcto funcionamiento del sistema una vez se ha implementado las principales funcionalidades.
- Redacción de la memoria y preparación de la defensa, con la que concluye la planificación del proyecto.

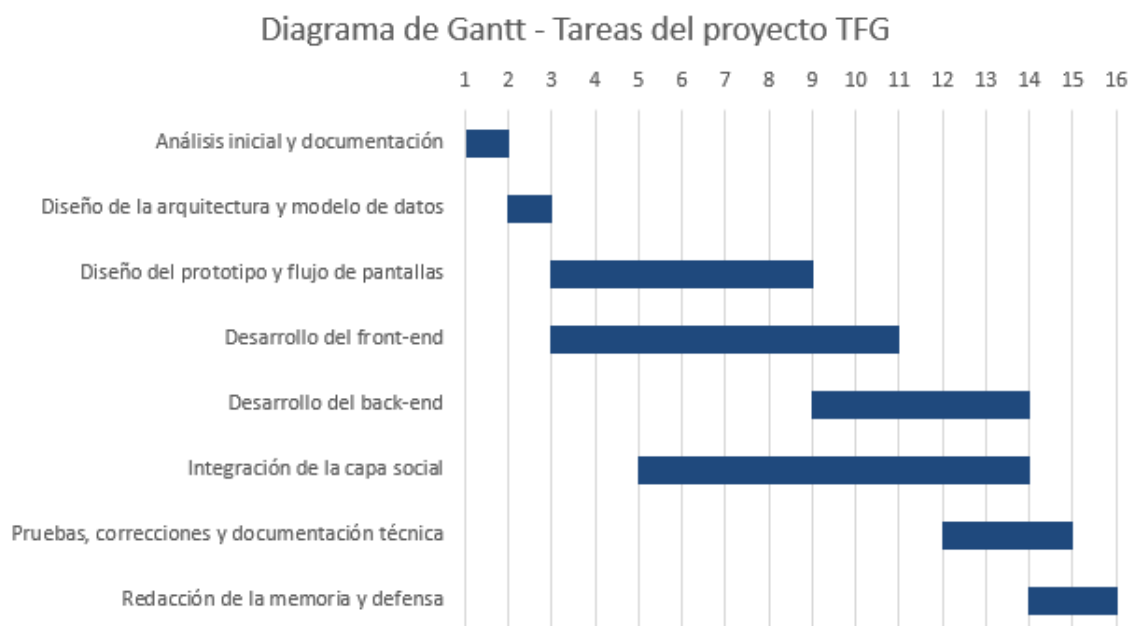


Ilustración 1: Diagrama de Gantt

3 Trabajos previos y análisis de aplicaciones existentes

El uso de aplicaciones en la práctica deportiva se ha convertido en algo habitual en los últimos años, favoreciendo la aparición de nuevas formas de organización, gestión, registro y participación en actividades deportivas. Esta evolución ha beneficiado el desarrollo de numerosas aplicaciones orientadas a facilitar la práctica deportiva, especialmente en los deportes de equipo.

Estas soluciones buscan cubrir necesidades como la conexión entre deportistas con intereses comunes. Sin embargo, la variedad de aplicaciones existentes presenta diversos enfoques y funcionalidades, lo que hace necesario un análisis previo que permita identificar y comparar las características que ofrece cada una de ellas.

El estudio de las aplicaciones existentes resulta fundamental para contextualizar el desarrollo de SportCrew, ya que permite identificar tanto las funcionalidades más relevantes como las limitaciones y carencias presentes en las soluciones actuales, sirviendo así de base para la definición de la propuesta desarrollada en este trabajo.

3.1 Análisis de las aplicaciones existentes

En el ámbito de las aplicaciones orientadas a la organización de actividades deportivas, se han desarrollado diversas soluciones que abordan problemas concretos relacionados con la gestión de partidos, la búsqueda de jugadores o la reserva de instalaciones. Sin embargo, estas aplicaciones tienen una característica en común, presentan arquitecturas diseñadas para un único deporte o un conjunto muy limitado de disciplinas, lo que condiciona su escalabilidad y su reutilización para otros contextos deportivos.

3.1.1 CeleBreak

CeleBreak es una aplicación centrada exclusivamente a un único deporte, el fútbol. Tiene como objetivo facilitar la organización de partidos, torneos, ligas y entrenamientos conectando a jugadores individuales para la formación de equipos. Desde un punto de vista funcional, la aplicación resuelve de forma específica la problemática en los deportes de equipo, la dificultad para reunir un número significativo de jugadores para partidos informales. Sin embargo, su diseño está claramente orientado a un único deporte, lo que limita su extensión hacia otras disciplinas deportivas sin una reestructuración significativa. [1]



Ilustración 2: CeleBreak

3.1.2 Playtomic

Con respecto a Playtomic, se trata de una plataforma especializada en deportes de raqueta, enfocada principalmente a la gestión de reservas de instalaciones deportivas y la organización de partidos y torneos. Su arquitectura funcional está orientada a la interacción de los usuarios y los clubes deportivos, ya que prioriza la gestión de los recursos físicos, como las pistas, y la planificación de partidos. Para la mayoría de las instalaciones deportivas, la plataforma actúa como una herramienta de apoyo en la gestión los horarios, evitando solapamientos en las reservas. Aunque ofrece un conjunto de funcionalidades sólido, su utilidad se limita a la práctica deportes de raqueta, especialmente el pádel, lo que restringe su aplicación a otros tipos de disciplinas deportivas. [2]



Ilustración 3: Playtomic

3.1.3 PickApp

PickApp es una aplicación orientada al baloncesto. Combina funcionalidades de red social con herramientas como la organización de partidos informales. Además, la plataforma permite a los usuarios localizar pistas cercanas, descubrir otros jugadores en su ciudad y crear o unirse a partidos organizados por la comunidad. Desde un punto de vista funcional, esta solución aborda la gran problemática en los deportes de equipo practicados de forma no federada: la dificultad para reunir un número suficiente de jugadores y encontrar espacios para la práctica deportiva. Sin embargo, su diseño está claramente orientado a un único deporte lo que limita su extensibilidad a otros deportes sin una adaptación significativa de su modelo funcional.

Además, aunque incorpora elementos sociales para la conexión de jugadores, estos están sujetos a la organización puntual de partidos, sin ofrecer una capa social generalizada que facilite la interacción entre usuarios con los mismos gustos deportivos. [3]



Ilustración 4: PickApp

3.2 Análisis de alternativas tecnológicas

Tras el análisis de distintas tecnologías de desarrollo multiplataforma y de diversas opciones para la implementación del backend, se decidió emplear React Native junto con Expo Cli y Expo Router para el desarrollo del frontend, y Firebase como plataforma Backend as a Service (BaaS) para el backend de la aplicación.

Antes de tomar esta decisión, se evaluaron varias alternativas tecnológicas que, finalmente, fueron descartadas por diferentes motivos, principalmente relacionados con la curva de aprendizaje, la complejidad de la implementación y la viabilidad temporal del proyecto.

En cuanto al desarrollo del frontend, se consideró el uso de Flutter [4], una herramienta potente para el desarrollo multiplataforma. Sin embargo, esta opción fue descartada principalmente por su mayor curva de aprendizaje. El hecho de que utilice Dart suponía una barrera adicional, mientras que React Native se basa en JavaScript/TypeScript, lenguajes con los que ya contaba con experiencia previa en el desarrollo de interfaces.

Respecto al backend, se valoró la implementación de una solución basada en Java junto con Spring Boot [5], debido a la familiaridad con el lenguaje y el ecosistema Maven. Finalmente, fue descartada por motivos de viabilidad temporal. Dado el alcance del proyecto y las horas disponibles, el hecho de implementar y mantener un servidor propio habría supuesto una carga de trabajo elevada.

Del mismo modo, se consideró el uso de Python junto con FastAPI, una opción ligera y eficiente para el desarrollo de servicios backend. Aun así, esta alternativa fue descartada por el mismo motivo que la anterior. Al implicar la gestión de una infraestructura propia resultaba poco viable desde el punto de vista temporal.

La elección de Firebase como plataforma BaaS permitió simplificar el desarrollo del backend, ofreciendo servicios integrados como autenticación, base de datos en tiempo real y almacenamiento. Estas características facilitaron el desarrollo, la escalabilidad y la sincronización de los datos entre los distintos usuarios de la aplicación.

4 Stack de desarrollo. Tecnologías utilizadas

La elección del stack de desarrollo se ha realizado teniendo en cuenta factores como la experiencia previa, la viabilidad temporal del proyecto, la escalabilidad del sistema y la facilidad de mantenimiento. A continuación, se describen las tecnologías y las herramientas utilizadas durante la implementación de la aplicación SportCrew.

4.1 Entorno de desarrollo

El desarrollo de la aplicación SportCrew se ha llevado a cabo en un entorno orientado a simplificar el control del código y la integración con las tecnologías seleccionadas. Para ello, se ha empleado Visual Studio Code como entorno de desarrollo integrado (IDE), por su ligereza, versatilidad y extensa compatibilidad con proyectos que utilizan React Native, TypeScript y Firebase.

Se ha empleado Node.js como gestor del entorno de ejecución. Se trata de una herramienta esencial para manejar las dependencias del proyecto a través de npm y para ejecutar Expo Cli. Este entorno ha hecho posible que las librerías empleadas en el desarrollo se instalen y configuren de manera adecuada.

Con el objetivo de mejorar la calidad del código y optimizar el proceso de desarrollo, se ha incorporado diversas extensiones de Visual Studio Code, entre las que destacan herramientas para el formateo automático del código, la detección temprana de errores y el autocompletado de estructuras habituales en React Native y TypeScript.

El entorno de desarrollo configurado ha permitido trabajar de forma eficiente y organizada, facilitando la implementación progresiva de las distintas funcionalidades de la aplicación y su posterior validación.

4.2 Frontend

En la implementación del frontend se ha utilizado React Native [6] como framework principal. Me ha permitido desarrollar una aplicación multiplataforma a partir de una única base de código y ha permitido garantizar una experiencia de usuario coherente en las distintas plataformas.

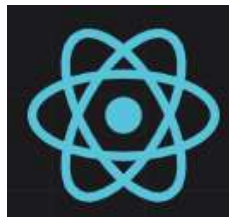


Ilustración 5: React Native

Con respecto a la gestión del proyecto y la ejecución de la aplicación se ha empleado Expo Cli [7]. Se trata de una herramienta de línea de comandos que simplifica el desarrollo de aplicaciones móviles al abstraer la compleja configuración nativa de iOS y Android. Gracias a Expo Cli ha sido posible crear,

ejecutar aplicaciones rápidamente, incluso en dispositivos físicos vía QR. Para esto último ha sido necesario también utilizar la aplicación Expo Go [8].



Ilustración 6: Expo Go

El proyecto se ha desarrollado usando TypeScript [9], tal y como se puede observar en los archivos (.ts) y (.tsx). Se trata de un lenguaje de programación que es un superconjunto de JavaScript, es decir, añade funcionalidades a JavaScript.



Ilustración 7: TypeScript

4.2.1 Navegación y estructura de pantallas

La navegación entre pantallas se ha implementado mediante Expo Router [10]. Se trata de un sistema de enrutamiento basado en archivos para aplicaciones de React Native y web, que simplifica la navegación creando automáticamente rutas a partir de la estructura de carpetas. La aplicación se organiza a partir del directorio /app, en el que se definen las distintas pantallas y secciones de la aplicación, incluyendo una estructura específica para la navegación mediante pestañas /(tabs).

También se han utilizado archivos de tipo _layout.tsx para definir layouts comunes y gestionar la navegación compartida entre distintas pantallas. Este archivo ha permitido una organización clara y modular del frontend, facilitando la escalabilidad de la aplicación y la incorporación de nuevas pantallas sin comprometer nada del código.

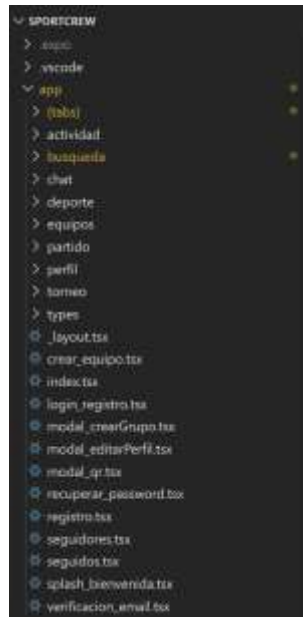


Ilustración 8: Estructura de SportCrew utilizando Expo Router

4.2.2 Interfaz de usuario y estilos

La mayoría de la interfaz de usuario se ha implementado utilizando los componentes nativos que proporciona React Native, como View, TouchableOpacity, ScrollView, etc. Asimismo, para la definición de estilos se ha utilizado StyleSheet, lo que ha permitido mantener una separación clara entre la lógica y el formato visual de la aplicación.

```

/**BANNER DE NOTIFICACIONES*/
(hayNotificacionesNoLeidas && (
  <TouchableOpacity style={styles.notificationBanner}
    onPress={() => router.push("/actividad/notificaciones")}
    activeOpacity={0.85}
  >
    <Ionicons name="notifications-outline" size={20} color="■ #1B4332" />
    <Text style={styles.notificationText}>Tienes notificaciones sin leer!</Text>
  </TouchableOpacity>
))

```

Ilustración 9: Ejemplo de un componente nativo proporcionado por React Native

Este enfoque ha facilitado la creación de una interfaz coherente, intuitiva y funcional, adaptable a distintos tamaños de pantalla y orientada a mejorar la experiencia de usuario.

4.3 Backend

Como ya se comentó en el capítulo anterior, para la implementación del backend se ha optado por el uso de Firebase, una plataforma Backend as a Service (BaaS) que proporciona servicios en la nube sin necesidad de gestionar una infraestructura propia. Esta elección ha permitido simplificar el desarrollo del backend y, así poder centrarnos en la implementación de la lógica de la aplicación.

En concreto, se han utilizado los siguientes servicios que proporciona Firebase: Firebase Authentication y Firebase Firestore.

4.3.1 Autenticación de usuarios

Para la parte del acceso de los usuarios a la aplicación y de su correspondiente autenticación, se ha utilizado Firebase Authentication [11]. Este servicio que incluye Firebase, permite gestionar de forma segura el registro, inicio de sesión y control de acceso de los usuarios, integrándose de manera directa con el frontend desarrollado.

El uso de Firebase Authentication ha facilitado la implementación de un sistema de autenticación fiable, reduciendo la complejidad asociada a la gestión manual de credenciales y proporcionando una base segura para el acceso a las funcionalidades de la aplicación.



Ilustración 10: Firebase Authentication

4.3.2 Base de datos en la nube

La persistencia de los datos de la aplicación se ha desarrollado mediante Cloud Firestore [12]. Se trata de una base de datos NoSQL que facilita el manejo de grandes volúmenes de datos no estructurados. Está orientada a documentos y permite el almacenamiento y sincronización de la información en tiempo real. Firestore se ha utilizado para gestionar los datos relacionados con usuarios, instalaciones, partidos, chats, torneos, equipos y notificaciones.

El uso de Cloud Firestore permite un acceso eficiente a los datos y facilita la actualización automática de la interfaz de usuario, garantizando la consistencia de los datos entre los distintos usuarios de la plataforma.



Ilustración 11: Cloud Firestore

La combinación de Firebase Authentication y Cloud Firestore ha permitido implementar un backend escalable y adecuado para una aplicación orientada a la interacción entre usuarios, manteniendo una arquitectura sencilla y alineada con los requisitos del proyecto.

4.4 Control de versiones

Para el control de versiones del código fuente se ha utilizado Git [13], en combinación con la plataforma GitHub [14] como repositorio remoto. Esta combinación ha permitido mantener un historial de cambios organizado, facilitar la gestión de versiones y asegurar la trazabilidad de la evolución del proyecto a lo largo del tiempo.



Ilustración 12: Git y GitHub

4.5 Herramientas para pruebas

Para las pruebas y la ejecución de SportCrew se ha utilizado distintas herramientas, lo que ha permitido validar su funcionamiento en diferentes entornos y plataformas a lo largo del desarrollo.

En la primera fase de la implementación, se hicieron pruebas mediante un emulador Android configurado con Android Studio [15] que simulaba un dispositivo virtual Pixel 8a. Este entorno permitió realizar las primeras comprobaciones del comportamiento de la aplicación en un dispositivo móvil.

Sin embargo, tras diversos problemas técnicos relacionados con el uso del emulador de Android Studio, se tomó la decisión de utilizar Expo Cli como la principal herramienta de ejecución. En particular, se ha hecho uso de la ejecución en entorno web, lo que ha facilitado la validación continua de la interfaz y de la lógica de la aplicación de manera ágil.

De forma complementaria, la aplicación ha sido probada en dispositivos reales mediante Expo Go, garantizando que se trata de una implementación con un diseño responsive.



Ilustración 13: Android Studio

4.6 Herramientas de apoyo al desarrollo (extensiones IDE)

Como se ha comentado anteriormente en el subcapítulo 4.1, con el objetivo de mejorar la productividad y la calidad del código, se han utilizado, como ayuda a la programación, las siguientes extensiones de Visual Studio Code.

- **ES7 + React/Redux/React-Native snippets:** Añade atajos de código para React y React Native.
- **Prettier – Code Formatter:** Formatea el código automáticamente (espacios, comillas, salto de línea...) para ver el código limpio y ordenado.
- **ESLint:** Detecta errores y malas prácticas en tiempo real mientras escribes. Evita algunos bugs antes de ejecutar la app.
- **Error Lens:** Muestra los errores de ESLint directamente en la línea sin tener que pasar el ratón.

4.7 Herramienta de apoyo al diseño: Figma

Para el diseño del prototipo de la aplicación y la definición del flujo de pantallas se ha utilizado la herramienta Figma [16]. Mediante esta herramienta se ha podido crear un prototipo visual previo al desarrollo, facilitando la planificación de la interfaz de usuario y sirviendo como referencia durante la implementación del frontend. El uso de esta herramienta ha contribuido a mantener la coherencia visual de la aplicación y a mejorar la experiencia de usuario desde las primeras fases del proyecto.



Ilustración 14: Figma

5 Fase de Análisis

Como ya se ha comentado, actualmente, muchas personas interesadas en la práctica deportiva amateur se enfrentan a bastantes dificultades para organizar partidos o torneos de forma regular. Entre los principales problemas se encuentra la falta de contactos con gustos similares, la complejidad para reunir un número significativo de participantes y la ausencia de una herramienta común que permita gestionar de manera eficiente la organización de encuentros e inscripciones de distintos deportes y las comunicaciones entre jugadores.

Esta variedad de aplicaciones obliga a los usuarios a utilizar múltiples plataformas o canales externos para organizar su actividad deportiva, generando una experiencia poco eficiente y desestructurada.

Después de analizar esta situación, se ha identificado la necesidad de crear una plataforma unificada que permita a los usuarios amateur crear, gestionar y participar en partidos y torneos de distintos deportes, facilitando la comunicación, el seguimiento entre jugadores y la gestión de eventos deportivos orientado a un ocio saludable.

5.1 Requisitos Funcionales

- **RF1: Gestión de usuarios.**
El sistema debe permitir el registro de nuevos usuarios.
- **RF2: Autenticación de usuarios.**
El sistema debe permitir a los usuarios autenticarse mediante credenciales válidas
- **RF3: Validación de cuenta.**
El sistema debe requerir la validación de la cuenta a través de un correo electrónico antes de permitir el acceso completo a la aplicación.
- **RF4: Recuperación de la contraseña.**
El sistema debe permitir al usuario poder recuperar la contraseña mediante el envío de un correo electrónico.
- **RF5: Gestión del perfil de usuario.**
El sistema debe permitir a los usuarios modificar su nombre, su nombre de usuario y la foto de perfil. Además, debe permitir la visualización de las estadísticas personales de juego y consultar los últimos partidos disputados.
- **RF6: Exclusividad en el nombre de usuario.**
El sistema debe garantizar que no existan dos o usuarios con el mismo nombre de usuario.
- **RF7: Seguimiento entre usuarios.**
El sistema debe permitir a los usuarios seguir y dejar de seguir a otros usuarios.
- **RF8: Seguimiento mediante código QR.**
El sistema debe permitir el seguimiento entre usuarios mediante el uso de códigos QR.
- **RF9: Gestión de equipos.**
El sistema debe permitir crear equipos y parejas por deporte y modalidad, consultar los equipos a los que pertenece un usuario y utilizar equipos previamente creados para la inscripción en partidos y torneos.

- **RF10: Gestión de partidos.**
El sistema debe permitir a los usuarios:
 - Crear partidos indicando la instalación, la modalidad, el nivel, la fecha y la hora.
 - Inscribirse en partidos de forma individual o mediante equipos.
 - Abrir o cerrar un partido a nuevas inscripciones.
 - Añadir notas informativas a un partido.
 - Modificar o eliminar partidos a su propio creador.
- **RF11: Gestión de torneos.**
El sistema debe permitir:
 - Crear torneos deportivos definiendo nombre, instalación, modalidad, formato, número de equipos, fecha y notas.
 - Inscribirse en torneos de acuerdo con la modalidad de estos.
 - Generar automáticamente los enfrentamientos una vez completado el número de participantes.
 - Calcular automáticamente el ganador del torneo al finalizar todos los enfrentamientos.
- **RF12: Clasificación y estados de los encuentros.**
El sistema debe gestionar el estado de los partidos y torneos, diferenciando entre: próximos, pendientes y finalizados.
- **RF13: Introducción de resultados.**
El sistema debe permitir que cualquier participante introduzca los resultados de un encuentro finalizado.
- **RF14: Consulta de información de eventos.**
El sistema debe permitir consultar la información detallada de partidos, torneos y equipos.
- **RF15: Búsqueda y filtrado de eventos.**
El sistema debe permitir la búsqueda de partidos y torneos mediante filtros como: instalación, modalidad, nivel, formato, fecha y hora.
- **RF16: Sistema de notificaciones.**
El sistema debe guardar notificaciones automáticas ante los siguientes eventos:
 - Creación de equipos y partidos.
 - Inscripción de usuarios o equipos a un partido o torneo.
 - Edición de un partido o torneo. Es decir, cambios relevantes en eventos en los que el usuario participa.
- **RF17: Mensajes**
El sistema debe permitir:
 - Envío y recepción de mensajes individuales.
 - Creación de chats de grupo.
 - Notificación de mensajes no leídos.
 - Generación de mensajes del sistema.

5.2 Requisitos No Funcionales

- **RNF1: Usabilidad.**
La aplicación debe ser intuitiva y fácil de usar para usuarios sin conocimientos técnicos.
- **RNF2: Diseño Responsive.**
La aplicación debe poder ser utilizada en distintos dispositivos móviles y en entorno web.

- **RNF3: Rendimiento.**
El sistema debe ofrecer tiempos de respuesta adecuados en operaciones habituales como cargar partidos, mensajes, torneos y notificaciones.
- **RNF4: Escalabilidad.**
El sistema debe permitir la incorporación de un número creciente de usuarios y eventos sin ralentizar su funcionamiento.
- **RNF5: Seguridad.**
El sistema debe:
 - Proteger los datos personales de los usuarios.
 - Restringir el acceso a la información según el usuario autenticado.
 - Garantizar la integridad de las inscripciones y resultados.
- **RNF6: Disponibilidad.**
El sistema debe estar disponible de forma continua, minimizando interrupciones del servicio.
- **RNF7: Mantenibilidad.**
La arquitectura del sistema debe permitir la incorporación de nuevas funcionalidades y el mantenimiento del código de forma sencilla.
- **RNF8: Persistencia de los datos.**
El sistema debe garantizar la correcta persistencia y consistencia de los datos almacenados.
- **RNF9: Fiabilidad.**
El sistema debe gestionar correctamente situaciones excepcionales, como cancelaciones de eventos.

6 Fase de Diseño

El diseño del sistema tiene como punto de partida los requisitos identificados previamente durante la fase de análisis. El objetivo principal es establecer una estructura que permita implementar la aplicación de forma coherente, escalable y flexible para adaptarse a los posibles cambios durante el desarrollo.

El diseño se ha planteado teniendo en cuenta el carácter práctico del proyecto y el enfoque ágil adoptado, priorizando el desarrollo de un MVP (“Minimum Viable Product”) que permita validar las funcionalidades principales de la aplicación desde las primeras iteraciones.

6.1 Tipo de arquitectura

La aplicación se ha diseñado siguiendo la arquitectura MVVM (Model-View-ViewModel) [17], con el objetivo de lograr una adecuada separación de responsabilidades, facilitar el mantenimiento del código y permitir la evolución progresiva del sistema conforme a nuestro enfoque.

Esta arquitectura resulta adecuada para aplicaciones con una fuerte interacción con el usuario y un uso intensivo de datos remotos, como es el caso de SportCrew.

6.1.1 Model

Es el encargado de representar los datos y la lógica de negocio de la aplicación. Incluye las principales entidades del dominio, tales como usuarios, partidos, torneos, equipos, chats y notificaciones, así como las operaciones necesarias para su gestión.

Entre sus responsabilidades podemos encontrar:

- La definición de la estructura de los datos del sistema.
- La comunicación con el backend para el almacenamiento y recuperación de la información.
- El procesamiento de datos relacionados con estadísticas de juego, resultados y estados de los encuentros.

6.1.2 View

La vista representa la interfaz de usuario de la aplicación y se encarga exclusivamente de mostrar información y recoger las acciones del usuario. Su diseño se basa en el prototipo previamente elaborado en Figma, garantizando una experiencia de usuario coherente e intuitiva.

La vista incluye, entre otras, las siguientes pantallas:

- Pantalla principal y búsqueda general.
- Pantallas de creación de partidos y torneos.
- Pantalla de búsqueda e inscripción de partidos y torneos.
- Pantalla con los formularios para crear equipos y crear chats de grupo.
- Pantallas de actividades y notificaciones.
- Pantalla de los chats.
- Perfil de usuario.

6.1.3 ViewModel

El ViewModel actúa como intermediario entre la vista y el modelo, gestionando la lógica de presentación y evitando la comunicación directa entre ambos componentes.

Sus principales funciones son:

- Solicitar y actualizar datos a través del modelo.
- Gestionar el estado de la aplicación.
- Procesar la información antes de ser mostrada a la interfaz.
- Coordinar la actualización de la vista ante cambios en los datos, como nuevas inscripciones, resultados, mensajes o notificaciones.

6.1.4 Esquema del flujo de datos

1. La vista captura una acción del usuario (por ejemplo, crear un partido o inscribirse en un torneo).
2. El ViewModel procesa la acción y solicita las operaciones necesarias al modelo.
3. El modelo gestiona la lógica correspondiente y actualiza los datos en el backend.
4. El ViewModel recibe la respuesta y actualiza el estado de la aplicación.
5. La vista se actualiza automáticamente reflejando los nuevos datos.

6.2 Enfoque ágil e incremental

El enfoque ágil e incremental ha permitido dividir el desarrollo en pequeñas iteraciones, priorizando en cada una de ellas la implementación de funcionalidades con mayor valor para el usuario.

Como punto de partida, se ha definido un MVP (“Minimum Viable Product”) que incluye las funcionalidades esenciales de la aplicación, como la gestión de usuarios, la creación e inscripción en partidos y la consulta de eventos. A partir de ahí, el sistema se ha ido ampliando progresivamente con funcionalidades adicionales, como la gestión de torneos, el sistema de mensajes y las notificaciones.

Podemos afirmar que este enfoque ha permitido introducir mejoras y ajustes de manera continua a lo largo del desarrollo.

6.3 Diseño Backend

En este proyecto no se dispone de un backend tradicional basado en servicios REST, sino que se utiliza Firebase como Backend as a Service (BaaS). La autenticación de usuarios y la persistencia de datos se realizan mediante Firebase Authentication y Cloud Firestore, accediendo a la información directamente desde el cliente.

En este contexto, la arquitectura MVVM se adapta perfectamente al comportamiento de React, donde el papel de ViewModel se realiza mediante el uso de hooks como useState y useEffect, responsables de gestionar el estado de la aplicación, realizar consultas a la base de datos y procesar la información antes de mostrarla en la interfaz.

6.4 Diseño Base de Datos

La base de datos utilizada en el proyecto se apoya en Cloud Firestore, el sistema de almacenamiento NoSQL proporcionado por Firebase. A diferencia de las bases de datos relacionales, Firestore no utiliza un esquema fijo, sino que organiza la información en colecciones y documentos, permitiendo una estructura flexible y fácilmente adaptable a la evolución de la aplicación. Por esta razón, resulta adecuada para nuestro desarrollo ágil e incremental.

A continuación, se muestran las colecciones que han sido necesarias para el desarrollo de SportCrew. Consta de 7 colecciones principales en las que podemos encontrar: usuarios, torneos, equipos, notificaciones, partidos, instalaciones y chats. Además, dentro de la colección de chats, podemos encontrar una colección donde se almacenan todos los mensajes.

6.4.1 Users – Usuarios

Todos los documentos almacenados en esta colección tienen obligatoriamente los siguientes campos, los usados para crear una cuenta en el sistema.

- **uid:** Identificador del usuario.
- **nombre:** Guarda el nombre completo de usuario.
- **username:** Se trata del nombre de usuario.
- **email:** Correo electrónico del usuario.
- **createdAt:** Fecha de creación de este documento.

A medida que el usuario va utilizando el sistema, se le generará los siguientes campos.

- **followers:** Array con los uid de los seguidores de un usuario.
- **following:** Array con los uid de los usuarios seguidos.

6.4.2 Facilities – Instalaciones

Las instalaciones tienen los datos guardados.

- **id:** Identificador de la instalación.
- **nombre:** Nombre de la instalación.
- **imagen:** Enlace de la imagen asociada a esa instalación.
- **direccion:** Lugar donde está ubicada la instalación.
- **ciudad:** Ciudad donde se encuentra la instalación.
- **descripcion:** Descripción para informar si son outdoor o indoor.
- **deportes:** Array donde aparecen los deportes que se pueden practicar en dicha instalación.

6.4.3 Teams – Equipos

Los documentos de esta colección tienen los siguientes 6 campos.

- **id:** Identificador del equipo.
- **name:** Nombre asignado al equipo.
- **sport:** Deporte al que pertenece.
- **modality:** Modalidad del deporte.
- **members:** Array donde se recogen los uid de los usuarios que pertenecen a este equipo.
- **createAt:** Fecha de creación del equipo.

6.4.4 Matches – Partidos

En esta colección se guardan dos tipos de documentos. Un tipo pertenece a los partidos sueltos y el otro corresponde a los partidos pertenecientes a un torneo, es decir, los que se generan automáticamente.

Un partido simple contiene:

- **id:** Identificador del partido.
- **creadoEn:** Fecha de creación.
- **creadorId:** Identificador del usuario que ha creado el partido.
- **creadorNombre:** Nombre del creador del partido.
- **deporte:** Nombre del deporte.
- **equipoLocal:** Map compuesto por el “**equipoId**” y un array “**jugadores**” con los uid de los jugadores.
- **equipoVisitante:** Map compuesto por el “**equipoId**” y un array “**jugadores**” con los uid de los jugadores.
- **estado:** Guarda si el partido está “**abierto**”, “**cerrado**”, “**caducado**” o “**finalizado**”.
 - Si el partido está “**abierto**” quiere decir que todavía hay plazas libres.
 - Aparece “**cerrado**” cuando ya no hay plazas libres.
 - “**Caducado**” cuando ha llegado la fecha del partido y no se han inscrito el mínimo de jugadores.
 - “**Finalizado**” cuando ha llegado la fecha del partido.
- **fecha:** Fecha del partido.
- **instalacionId:** Identificador de la instalación.
- **instalacionNombre:** Nombre de la instalación.
- **maxJugadoresPorEquipo:** Número máximo de jugadores por equipo.
- **minJugadoresPorEquipo:** Número mínimo de jugadores por equipo.
- **modalidad:** Modalidad del deporte.
- **nivel:** “**Principiante**”, “**Medio**” o “**Avanzado**”.
- **notas:** Se guarda información adicional.

Cuando el partido se ha jugado y el estado es “finalizado”, se insertan nuevos campos al documento.

- **ganador:** uid del usuario ganador.
- **resultado:** Map donde se registra el resultado. Existen dos estructuras:
 - Si el **tipo:** “**puntos**”, guarda un número “**local**”, otro “**visitante**”.
 - Si el **tipo:** “**sets**”, guarda dos arrays de números (“**local**” y “**visitante**”) con los resultados de cada set.
- **updatedAt:** Fecha en la que se introduce el resultado.

Los partidos que corresponden a un torneo tienen los mismos campos que los anteriores, pero añadiendo algunos más para poder identificar tanto la ronda como el torneo al que pertenecen.

- **jornada** o **ronda:** Dependiendo del formato del torneo.
- **torneoId:** Identificador del torneo.

6.4.5 Tournaments – Torneos

En los documentos de esta colección se guarda tanto los campos principales como todos los enfrentamientos que componen la estructura completa del torneo. Esta estructura puede variar dependiendo del formato del torneo.

- **creadoEn:** Fecha de creación.
- **creadorId:** Identificador del usuario creador del torneo.
- **deporte:** Nombre del deporte.
- **equipos:** Array que contiene los identificadores de los participantes. Si se trata de una modalidad individual, estos identificadores corresponderán a uid de usuarios, si no serán id de equipos.
- **estructura:** Map con todos los enfrentamientos del torneo. Sus campos dependen del formato guardado.
- **estado:** Puede ser “abierto”, “cerrado”, “en_curso”, “caducado”, “finalizado”.
- **fecha:** Fecha de inicio del torneo.
- **formato:** Tipo de torneo. De este campo depende la estructura.
- **instalacion:** Nombre de la instalación.
- **instalacionId:** Identificador de la instalación.
- **maxTeams:** Número máximo de equipo.
- **modalidad:** Modalidad del torneo.
- **nivel:** “Principiante”, “Medio” o “Avanzado”.
- **nombre:** Nombre del torneo.
- **notas:** Información adicional.
- **partidos:** Array con los identificadores de cada partido.

6.4.6 Notifications – Notificaciones

Para las notificaciones solo ha sido necesario guardar los siguientes datos.

- **id:** Identificador de la notificación.
- **createdAt:** Fecha de creación del documento.
- **message:** Mensaje que se muestra en la notificación.
- **read:** booleano para saber si el usuario ha leído el aviso.
- **targetId:** Identificador de la estructura que se ha modificado. Puede ser un partido, un torneo o un equipo.
- **targetType:** Tipo de estructura, es decir, si es un partido, un equipo o un torneo.
- **title:** Título de la notificación.
- **userId:** Identificador del usuario que recibe la notificación.

6.4.7 Chats.

Esta colección almacena otra colección con los mensajes de cada uno de los chats.

- **id:** Identificador del chat.
- **createAt:** Fecha de creación.
- **lastMessage:** Último mensaje de la conversación.
- **lastMessageAt:** Fecha y hora del último mensaje.
- **members:** Array con los uid de los usuarios pertenecientes al chat.
- **name:** Solo para chat grupales, sirve para guardar el nombre del grupo.
- **type:** Guarda el tipo de chat, si es “individual” o “group”.

- **unread:** Array de contadores por usuario para calcular los mensajes leídos y generar el badge de mensajes no leídos.

6.4.7.1 Messages – Mensajes.

Se trata de una colección de documentos donde se almacenan todos los mensajes de un chat. Además, existen dos tipos de mensaje, el creado por el sistema y el enviado por un usuario.

Para los mensajes del sistema:

- **id:** Identificador del mensaje.
- **createdAt:** Fecha de creación.
- **text:** Contenido del mensaje.
- **type:** Tipo de mensaje. En este caso es “**system**”.

Para los mensajes de un usuario:

- **id:** Identificador del mensaje.
- **createdAt:** Fecha de creación.
- **senderId:** Identificador del usuario emisor.
- **senderName:** Nombre del usuario emisor.
- **text:** Contenido del mensaje.
- **type:** Tipo de mensaje. En este caso es “**person**”.

6.5 Diseño Frontend – Prototipo Figma

Para diseñar el frontend se ha implementado el siguiente prototipo utilizando la herramienta Figma. Consta de 60 pantallas, donde se puede ver un flujo completo de las siguientes acciones.



Ilustración 15: Visión General del Prototipo en Figma

- Inicio de sesión de un usuario.
- Registro y verificación de un nuevo usuario.
- Creación de un partido.
- Búsqueda de un torneo.
- Creación de un chat grupal.
- Envío de mensajes por un chat individual.
- Visualización de las 4 pantallas de la sección “Actividad”.
- Visualización del perfil del usuario con el correspondiente cierre de sesión.

6.5.1 Inicio de sesión de un usuario

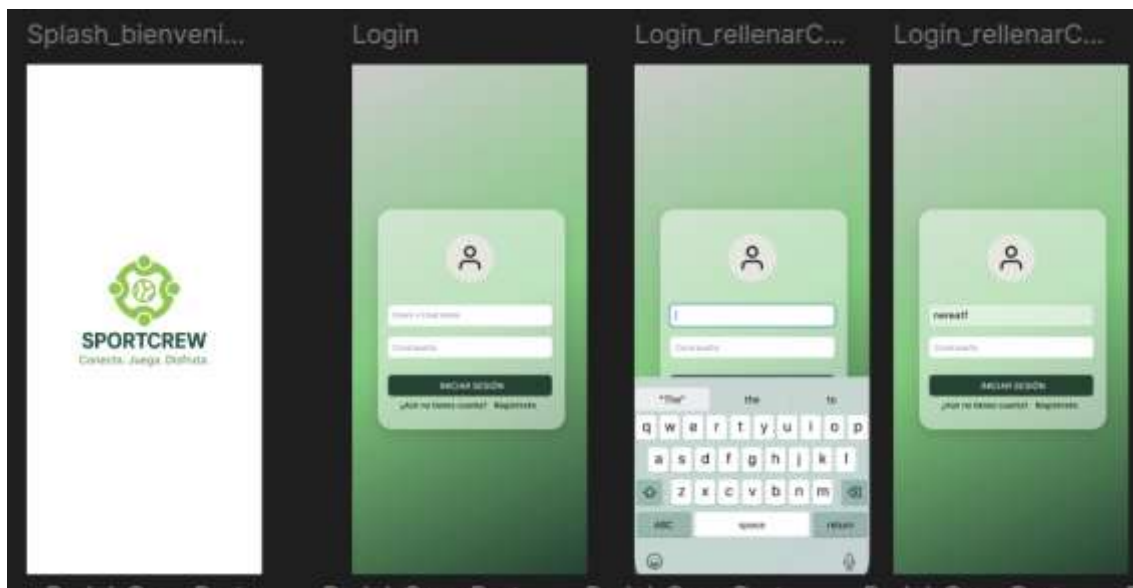


Ilustración 16: Inicio de sesión en Figma (1)

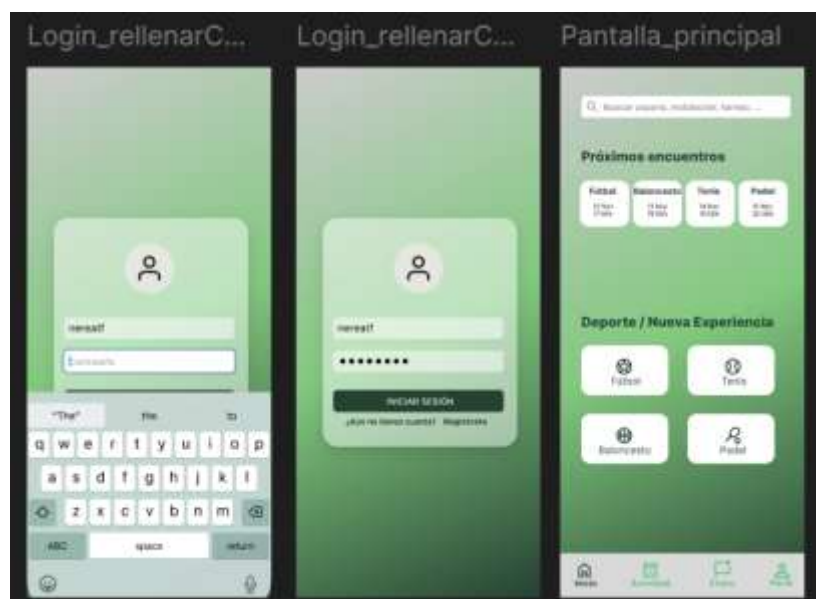


Ilustración 17: Inicio de sesión en Figma (2)

6.5.2 Registro y verificación de un nuevo usuario

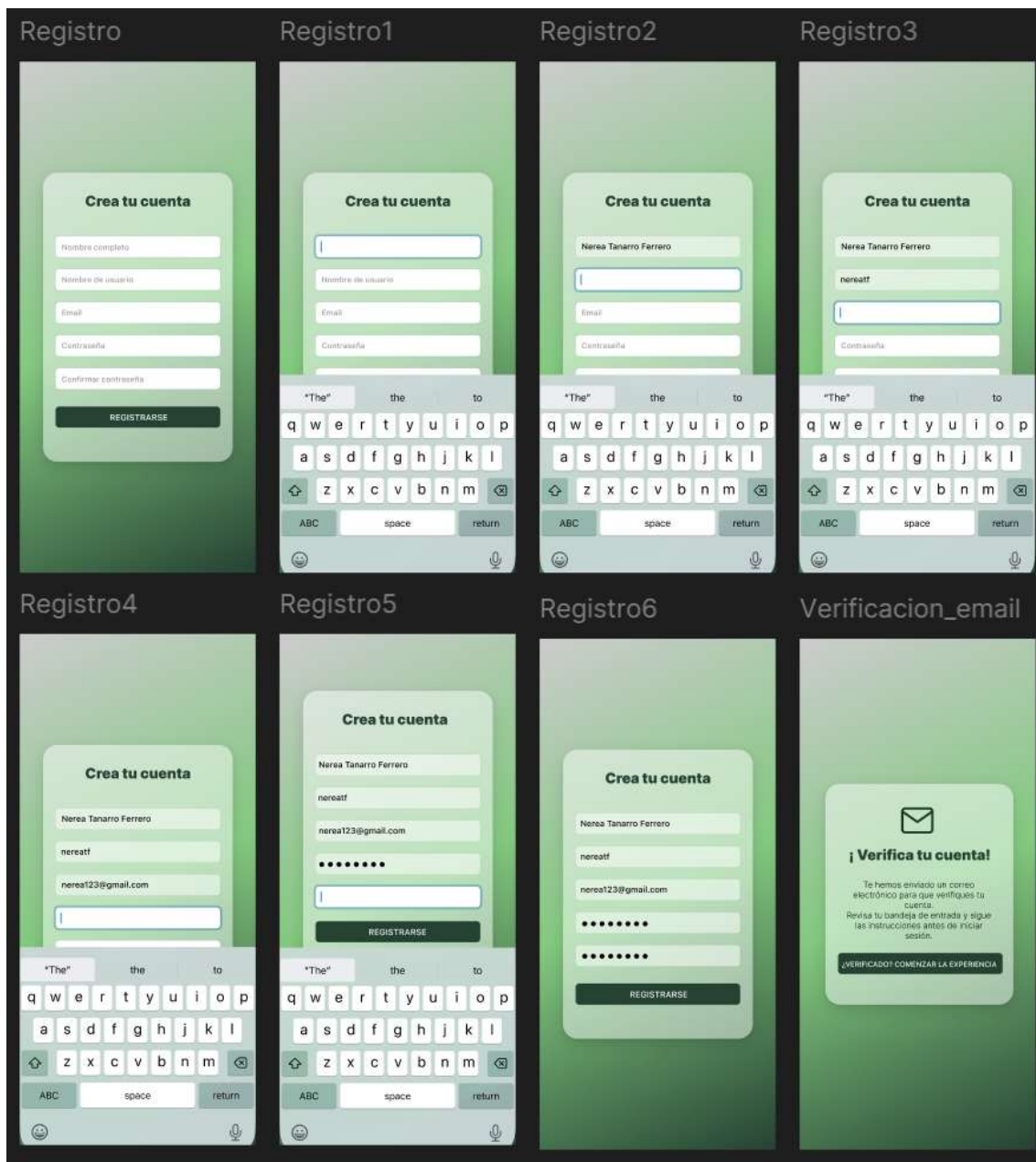


Ilustración 18: Registro y verificación de un nuevo usuario en Figma

6.5.3 Creación de un partido

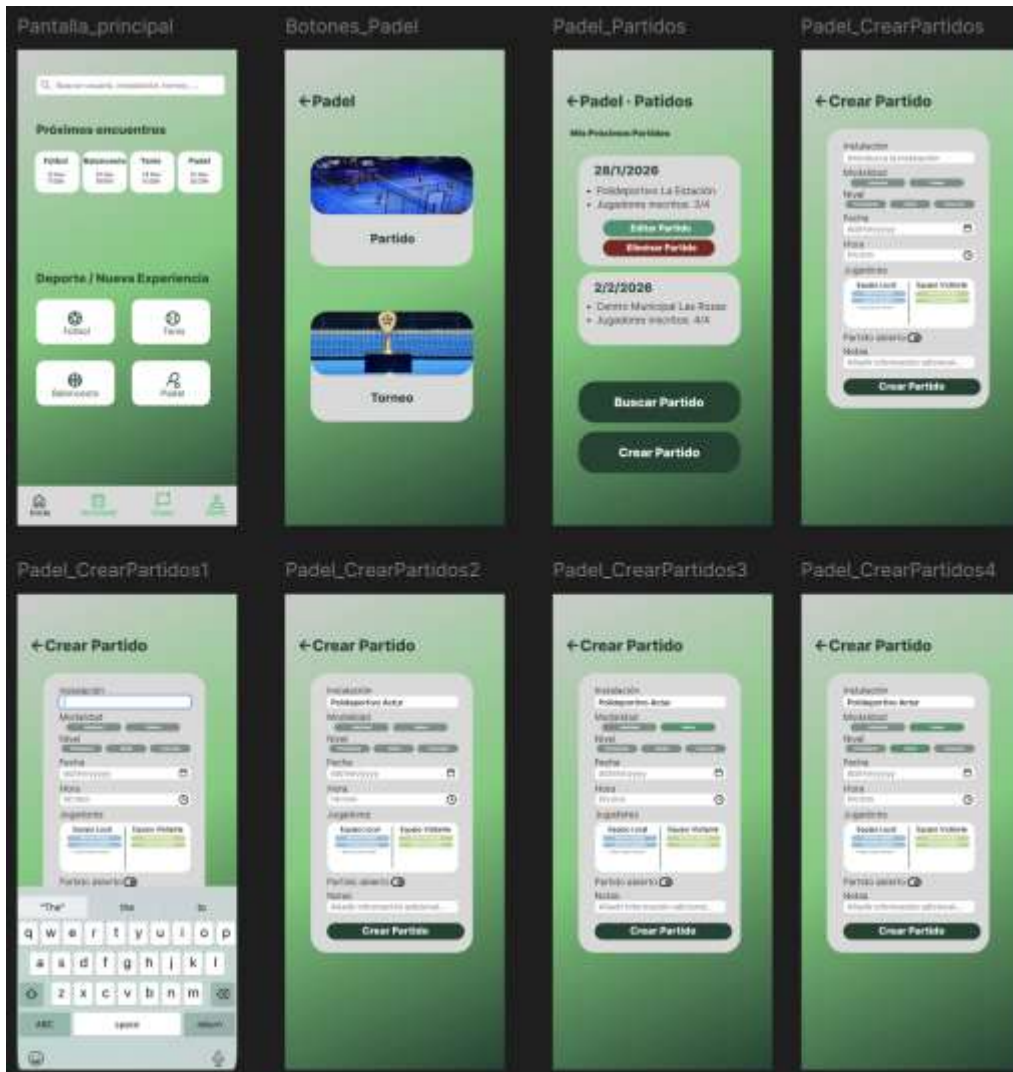


Ilustración 19: Creación de un partido en Figma (1)



Ilustración 20: Creación de un partido en Figma (2)



Ilustración 21: Creación de un partido en Figma (3)

6.5.4 Búsqueda de un torneo



Ilustración 22: Búsqueda de un torneo en Figma (1)



Ilustración 23: Búsqueda de un torneo en Figma (2)

6.5.5 Creación de un chat grupal

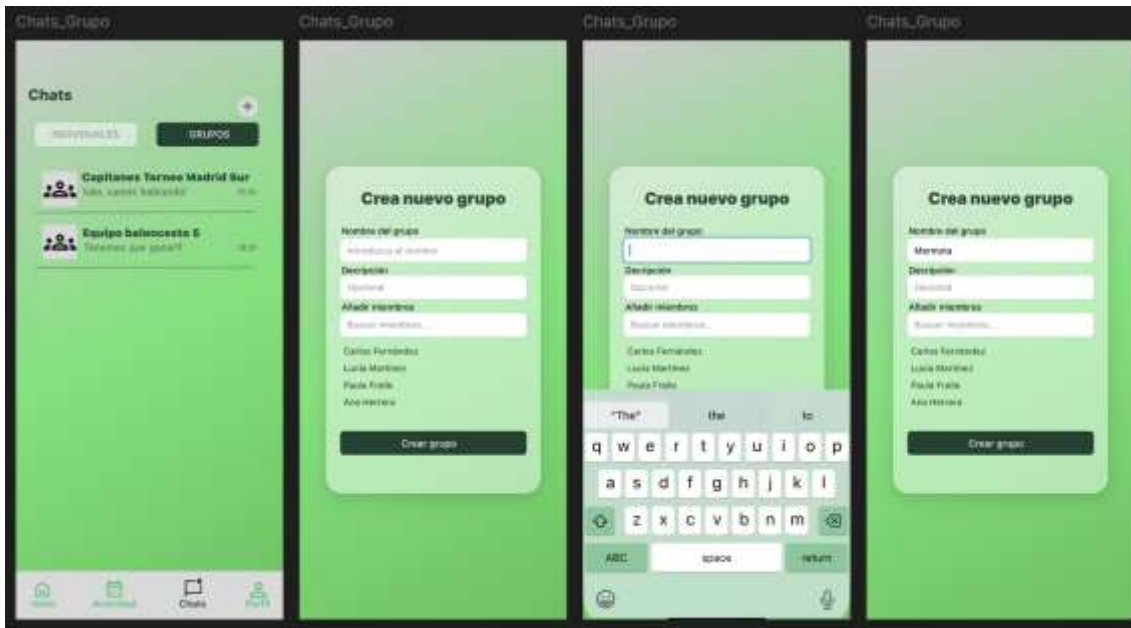


Ilustración 24: Creación de un chat grupal en Figma (1)

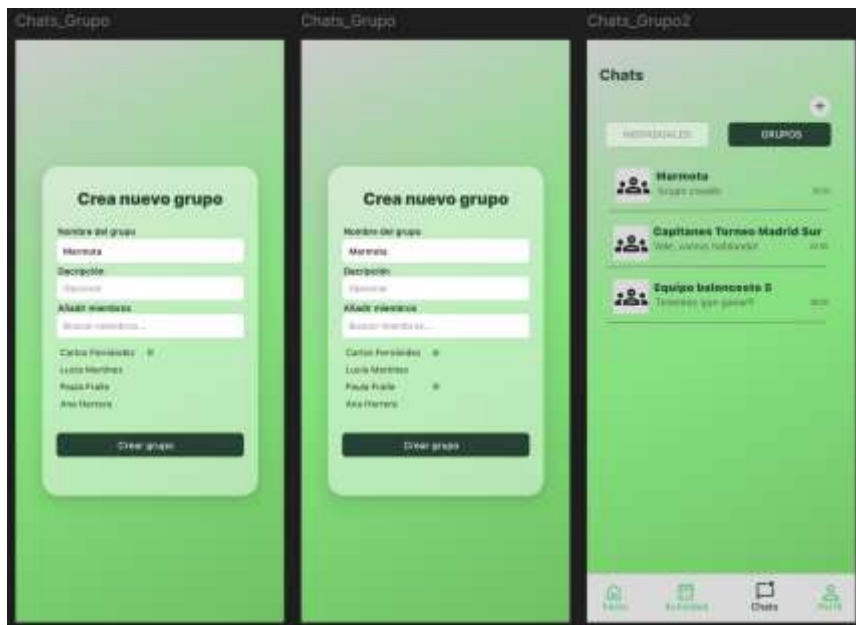


Ilustración 25: Creación de un chat grupal en Figma (2)

6.5.6 Envío de mensajes por un chat individual



Ilustración 26: Envío de mensajes por un chat individual en Figma (1)

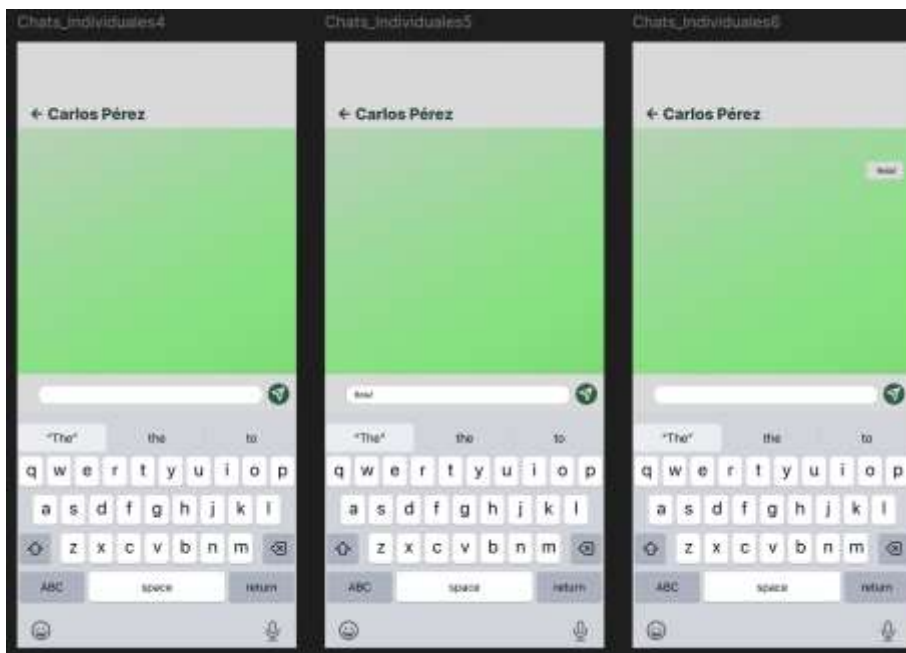


Ilustración 27: Envío de mensajes por un chat individual en Figma (2)

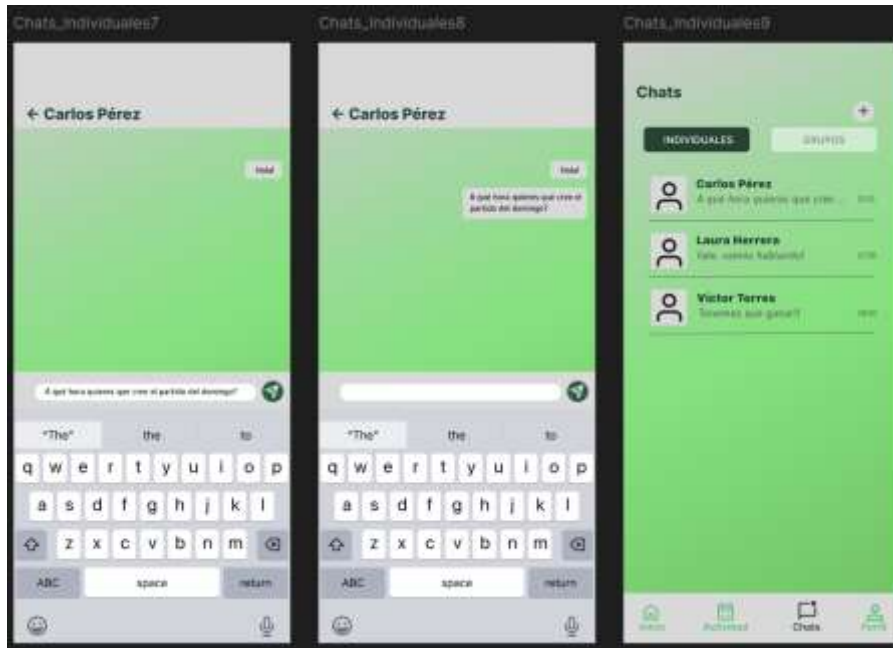


Ilustración 28: Envío de mensajes por un chat individual en Figma (3)

6.5.7 Visualización de las 4 pantallas de la sección “Actividad”



Ilustración 29: Visualización de las 4 pantallas de la sección "Actividad" en Figma

6.5.8 Visualización del perfil del usuario con el correspondiente cierre de sesión

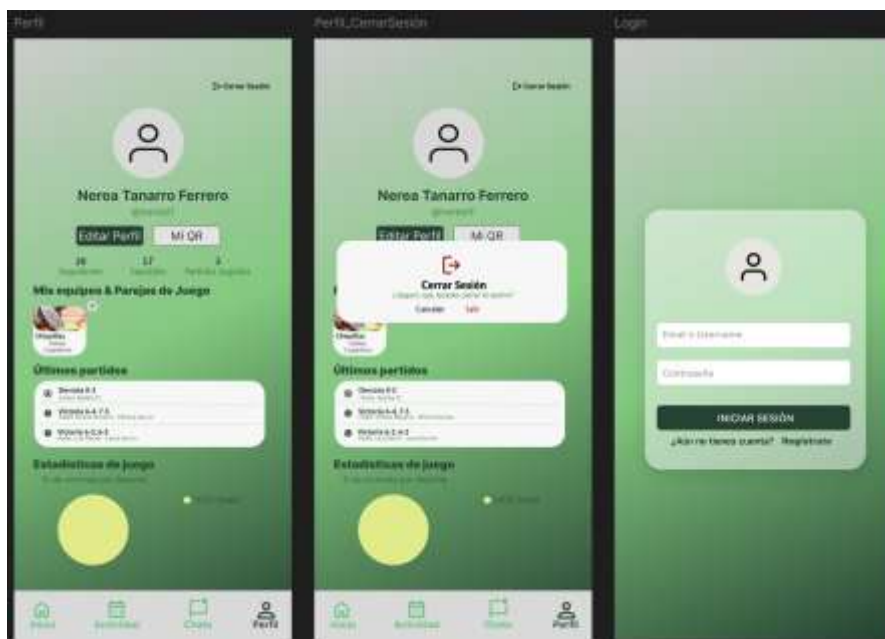


Ilustración 30: Visualización del perfil del usuario y cierre de sesión en Figma

7 Fase de Desarrollo

Como se ha comentado anteriormente, esta aplicación intenta solventar la problemática de reunir a un número significativo de jugadores para poder practicar algunos de los deportes más comunes. Además, busca fomentar el deporte como una vía de desconexión, ocio saludable y socialización. Para ello, se han integrado cuatro de los deportes más practicados en España, de forma que el usuario pueda gestionar todas sus actividades deportivas desde una única aplicación.

Más allá de lo deportivo, la aplicación incorpora una fuerte componente social, ofreciendo funcionalidades como chats individuales y grupales, creación de equipos, seguimiento entre usuarios y sus perfiles públicos donde se refleja la actividad y el progreso de cada persona. Con respecto a esta última funcionalidad, se ha implementado un sistema de códigos QR que facilita el seguimiento mutuo entre usuarios tras disputar partido, agilizando la creación de nuevas relaciones para futuros encuentros deportivos.

7.1 Principales Funcionalidades de SportCrew

La plataforma ofrece las siguientes funcionalidades:

- Pantalla de bienvenida o splash con el logotipo y el eslogan.
- Registro e inicio de sesión de usuarios.
- Validación de la cuenta mediante correo electrónico.
- Recuperación de contraseña a través del correo electrónico.
- Cuatro pantallas principales: inicio, actividad, chats y perfil de usuario.
- Creación de partidos, torneos y equipos.
- Sistema de seguimiento entre usuarios (seguir, dejar de seguir y QR) con acceso a perfiles de otros usuarios mediante buscador o lista de seguidores.
- Sistema de notificaciones integrado en la propia app.

7.2 Logo de SportCrew

Con este logotipo he querido representar la esencia del proyecto, combinando el deporte con la relación social. Está formado por varias figuras humanas dispuesta en círculo simbolizando la unión, la cooperación y la interacción entre personas. En el centro se observa una pelota, representando el deporte como punto de encuentro.

Tanto el logotipo como el slogan: “Conecta. Juega. Disfruta” aparecen en la primera pantalla de nuestra aplicación: “splash_bienvenida.tsx”



Ilustración 31: Logo de SportCrew

7.3 Pantallas Desarrolladas

7.3.1 Pantalla de inicio de sesión

El inicio de sesión de los usuarios se gestiona mediante Firebase Authentication, apoyado por distintas validaciones implementadas en el frontend.

El acceso a la aplicación se permite únicamente cuando las credenciales introducidas son válidas, pudiendo realizarse el inicio de sesión utilizando tanto el correo electrónico como el nombre de usuario y su respectiva contraseña.

Como medida de seguridad, no se permite el acceso a aquellos usuarios que no hayan verificado previamente su cuenta a través del correo electrónico.

Una vez autenticado, el usuario mantiene la sesión activa hasta que decida cerrarla manualmente, lo que evita la necesidad de introducir las credenciales en cada acceso a la aplicación.



Ilustración 32: Pantalla de inicio de sesión

7.3.2 Pantalla ¿Has olvidado tu contraseña?

Existe una opción en la pantalla de inicio de sesión donde el usuario puede cambiar la contraseña. Para la implementación de este apartado se ha usado una de las funcionalidades que presenta Firebase Authentication.



Ilustración 33: Recuperar contraseña

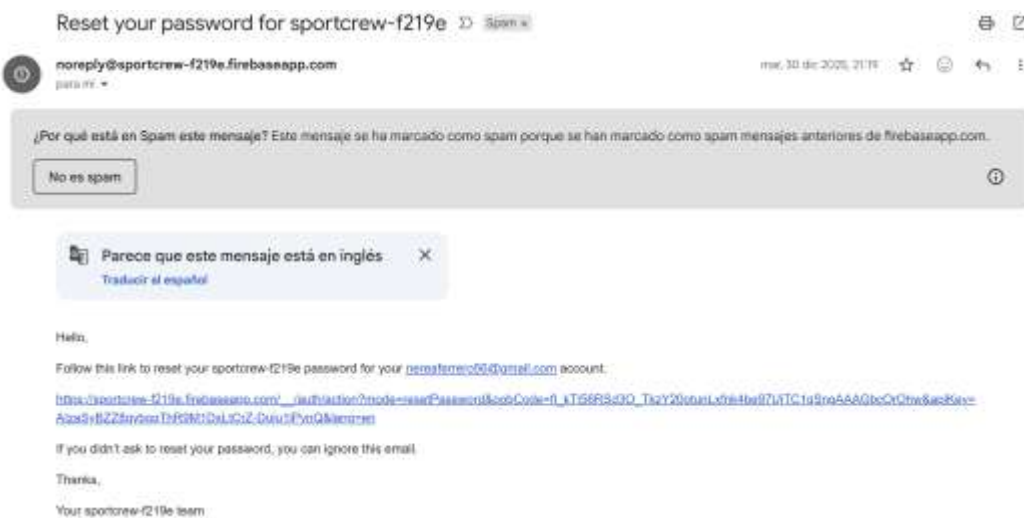


Ilustración 34: Correo para resetear la contraseña

7.3.3 Pantalla de Registro

Al igual que el inicio de sesión, el registro de nuevos usuarios también funciona mediante Firebase Authentication.

Con el objetivo de garantizar la integridad y validez de la información almacenada, se ha implementado diversas medidas de validación en el frontend. Estas validaciones impiden el registro de usuarios con campos incompletos, formatos de correo electrónico incorrectos o contraseñas que no coincidan entre sí. De este modo, se evita almacenar en la base de datos información innecesaria o incorrecta.

Una vez el usuario ha introducido todos los datos solicitados, el sistema envía automáticamente un correo electrónico de verificación. Durante este proceso, la aplicación muestra una pantalla informativa indicando la necesidad de confirmar la cuenta para poder completar el registro y así, acceder a todas las funcionalidades de la aplicación.



Ilustración 35: Pantalla para crear una cuenta



Ilustración 36: Pantalla informativa, verificación de la cuenta

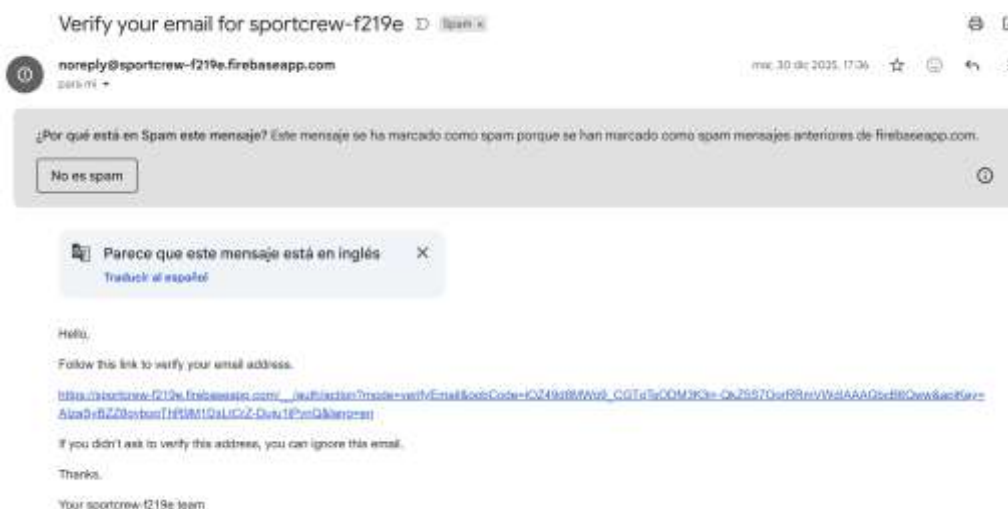


Ilustración 37: Correo recibido para verificar la cuenta

7.3.4 Pantalla principal

Se trata del punto de acceso a las funcionalidades más relevantes de la aplicación. En ella se incorpora una barra de búsqueda que permite localizar tanto a otros usuarios como a instalaciones deportivas, facilitando la consulta de información detallada de cada una de ellas.

Además, presenta cuatro botones principales, correspondientes a los deportes disponibles. A través de estos accesos, el usuario puede crear nuevos partidos o torneos, así como buscar partidos y torneos existentes.

Se ha implementado un banner informativo que solo se muestra cuando existen notificaciones sin leer, permitiendo al usuario saber que tiene nueva información pendiente de consultar.

En la parte inferior, podemos encontrar el acceso a los otros apartados de la aplicación: “Actividad”, “Chats” y “Perfil”.

Por último, la pantalla principal incluye un apartado resumen en el que se muestran los datos más relevantes de los próximos encuentros del usuario autenticado, ofreciendo una vista rápida de su actividad deportiva más próxima.



Ilustración 38: Pantalla de inicio

7.3.4.1 Pantalla barra de búsquedas

A través de esta pantalla, el usuario podrá visualizar el perfil de otros usuarios y seguir o dejar de seguir a una persona.



Ilustración 39: Barra de búsqueda (1)



Ilustración 40: Barra de búsqueda (2)

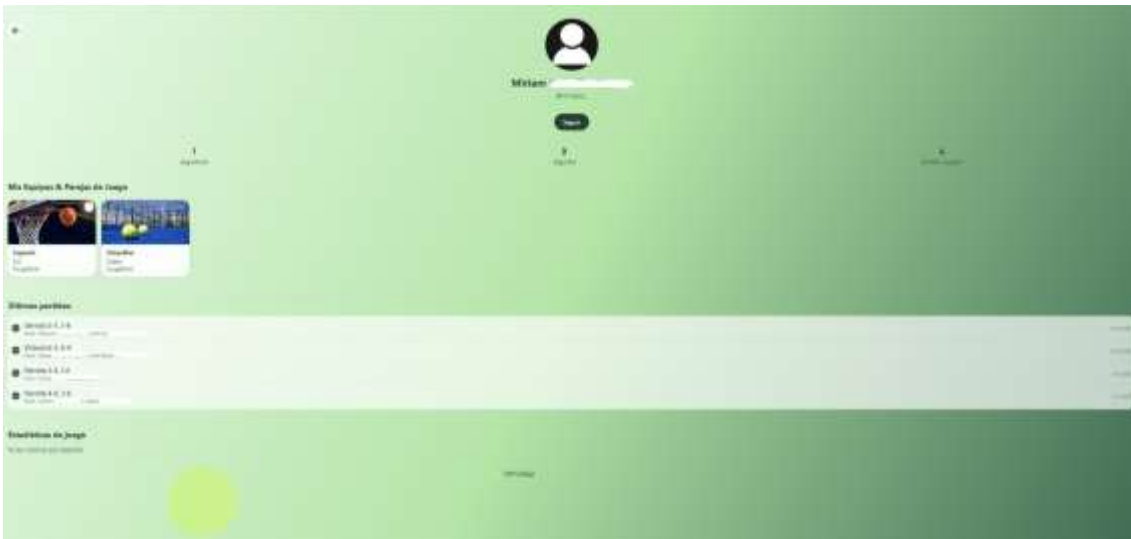


Ilustración 41: Perfil público de un usuario



Ilustración 42: Información de una instalación

7.3.5 Pantalla de cada botón de los deportes

Desde la pantalla inicial cada botón correspondiente a un deporte da acceso a una pantalla específica en la que se presentan dos opciones: partidos y torneos.

Esta estructura permite al usuario gestionar de forma clara y organizada todas las actividades asociadas a cada deporte.

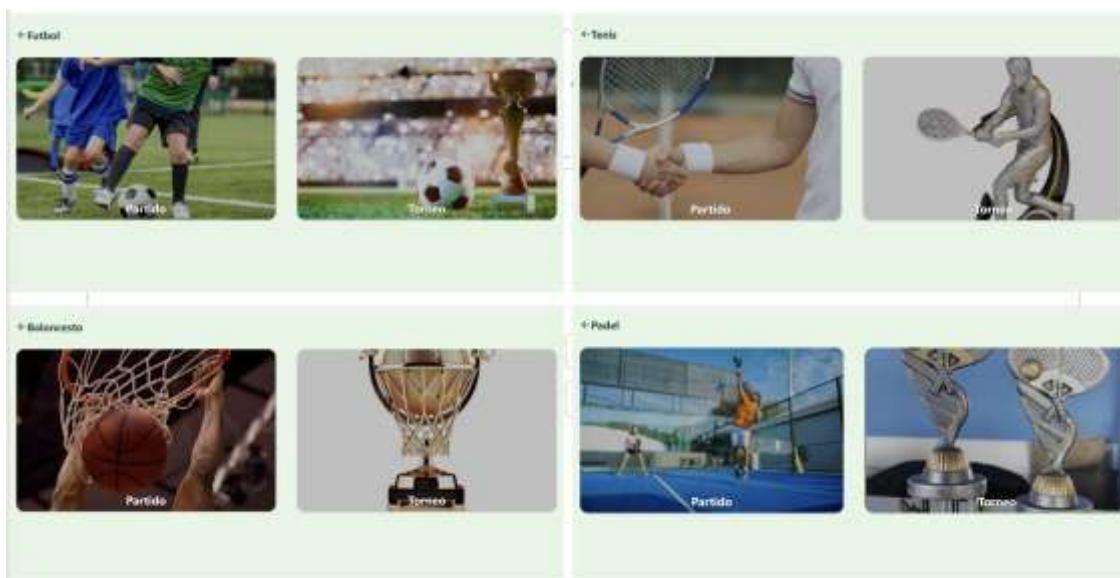


Ilustración 43: Pantalla de los botones de cada deporte

En ambas secciones se muestra, en primer lugar, una lista con los próximos partidos o torneos disponibles. En el caso de que el usuario sea el creador del evento, se habilitan dos opciones adicionales que permiten modificar o eliminar el partido o torneo correspondiente. Además, en estas pantallas observamos los accesos directos para crear nuevos eventos o buscar existentes.



Ilustración 44: Pantalla de los partidos de un deporte

7.3.6 Creación de Partidos

Para crear un partido, el usuario debe introducir una serie de datos obligatorios, entre los que se incluyen la instalación, la modalidad de juego (individual, dobles, 5x5, 3x3, fútbol 7 o fútbol 11), el nivel de los participantes (principiante, medio o avanzado) y la fecha y hora del encuentro.

El usuario que crea el partido queda inscrito automáticamente. Además, el sistema permite que la inscripción se pueda hacer de manera individual, colectiva o mediante un equipo previamente creado.

El creador puede decidir si el partido permanece abierto o cerrado a nuevas inscripciones y dispone de un apartado opcional donde puede añadir notas con información adicional relevante.

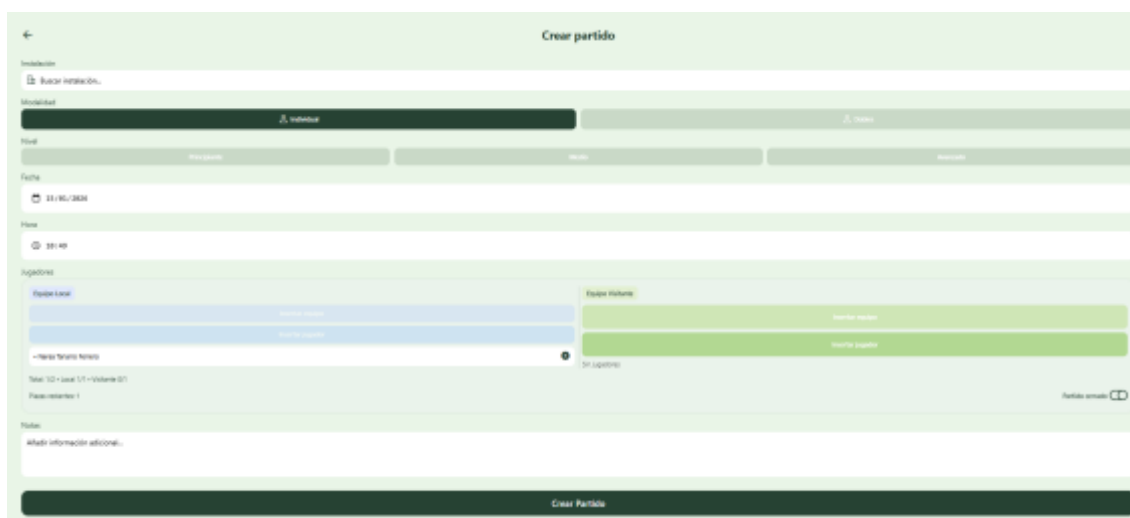


Ilustración 45: Pantalla crear partidos

7.3.7 Creación de Torneos

En este caso, el usuario que los crea puede actuar únicamente de organizador, por lo que no queda inscrito automáticamente. Para su creación se solicitan datos como el nombre del torneo, la instalación, la modalidad, el formato de competición, el nivel, el número de equipos participantes, la fecha de inicio y, de forma opcional, un apartado de notas.

Una vez alcanzado el número necesario de equipo o jugadores, el sistema genera automáticamente los enfrentamientos correspondientes. Tras la finalización de todos los partidos, el ganador del torneo se calcula y asigna de manera automática.

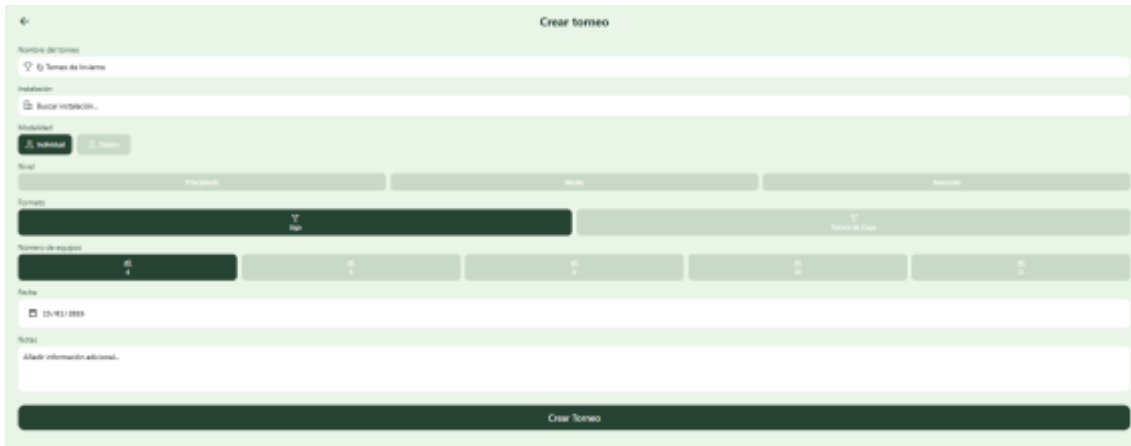


Ilustración 46: Pantalla crear torneos

7.3.8 Búsqueda de Partidos

La aplicación ofrece una funcionalidad de búsqueda avanzada de partidos, permitiendo al usuario aplicar distintos filtros para reducir el número de resultados. Entre los filtros disponibles se encuentran la instalación, la modalidad, el nivel, la fecha y la hora.

Si el usuario ya está inscrito, el sistema impide que pueda volver a inscribirse. En el momento de la inscripción, se solicita el tipo de participación (individual o mediante equipo) y, siempre que sea posible, se permite seleccionar la preferencia de equipo local o visitante.



Ilustración 47: Búsqueda de Partidos

7.3.9 Búsqueda de Torneos

De forma similar, la búsqueda de torneos permite aplicar filtros como instalación, modalidad, formato, nivel y fecha. Al igual que en los partidos, no se permite la inscripción en torneos en los que el usuario ya participa.

El tipo de inscripción depende de la modalidad seleccionada. En modalidades individuales, el usuario se inscribe directamente, mientras que en modalidades por equipos es necesario disponer de un equipo creado previamente para poder completar la inscripción.



Ilustración 48: Búsqueda de Torneos



Ilustración 49: Inscripción sin equipo creado previamente

7.3.10 Pantalla Actividad

La pantalla de actividad permite al usuario consultar y gestionar todos los partidos, torneos y notificaciones asociadas a su cuenta. Esta pantalla se organiza en cuatro pestañas, tres de ellas dedicadas a la gestión de encuentros deportivos y una destinada a las notificaciones.

Las tres primeras pestañas clasifican los partidos y torneos según su estado:

- **Próximos:** incluye los partidos y torneos que han alcanzado el número máximo de participantes, garantizando que el encuentro se celebrará según lo previsto.
- **Pendientes:** recoge los encuentros que aún disponen de plazas libres. Cuando se aproxima la fecha de inicio, el sistema evalúa automáticamente el número de participantes. Aquellos partidos o torneos que no alcanzan el mínimo de jugadores requerido se eliminan, mientras que los que sí cumplen esta condición se mantienen activos y se disputan con normalidad.
- **Finalizados:** muestra todos los partidos y torneos que ya se han celebrado. En el caso de los partidos, algunos muestran el resultado ya registrado, mientras que otros quedan a la espera de que cualquier participante introduzca el resultado. La inserción del resultado se realiza desde esta misma pestaña mediante un formulario emergente (modal), adaptado al tipo de deporte. Para baloncesto y fútbol se introducen los puntos, mientras que para tenis y pádel se registran los sets disputados.

En estas tres pestañas, cada evento dispone de un botón de acceso a los detalles de cada uno, permitiendo al usuario consultar toda la información relacionada con el mismo.

La cuarta pestaña corresponde a las **notificaciones**, donde se muestran tanto las leídas como las no leídas. Cada notificación, al igual que con los eventos, también incluye un botón de acceso a los detalles del partido, torneo o equipo afectado. Entre los tipos de notificaciones implementadas se encuentran:

- Creación e inscripción de jugadores.
- Creación de un equipo.
- Edición de partidos o torneos. En este caso notifica a todos los participantes excepto al creador del evento.



Ilustración 50: Apartado de Actividad

7.3.11 Chats

Este desarrollo incorpora una pantalla de mensajería que permite la comunicación entre usuarios mediante chats individuales y grupales, con un funcionamiento similar al de las aplicaciones de mensajería existentes.

Desde esta pantalla el usuario puede continuar conversaciones previamente iniciadas o crear nuevas. Se ha implementado un sistema de avisos que muestra un banner con el número de mensajes no leídos, facilitando la identificación de nuevas interacciones.

También, el sistema permite la creación de grupos de conversación, la gestión de sus miembros y la posibilidad de abandonar un grupo. Además, se ha implementado mensajes automáticos del sistema que informan de eventos relevantes, como la incorporación o salida de usuarios de un grupo.

Todo el sistema de mensajería funciona en tiempo real, garantizando una comunicación fluida entre los participantes.



Ilustración 51: Colección de chats individuales



Ilustración 52: Colección de chats grupales

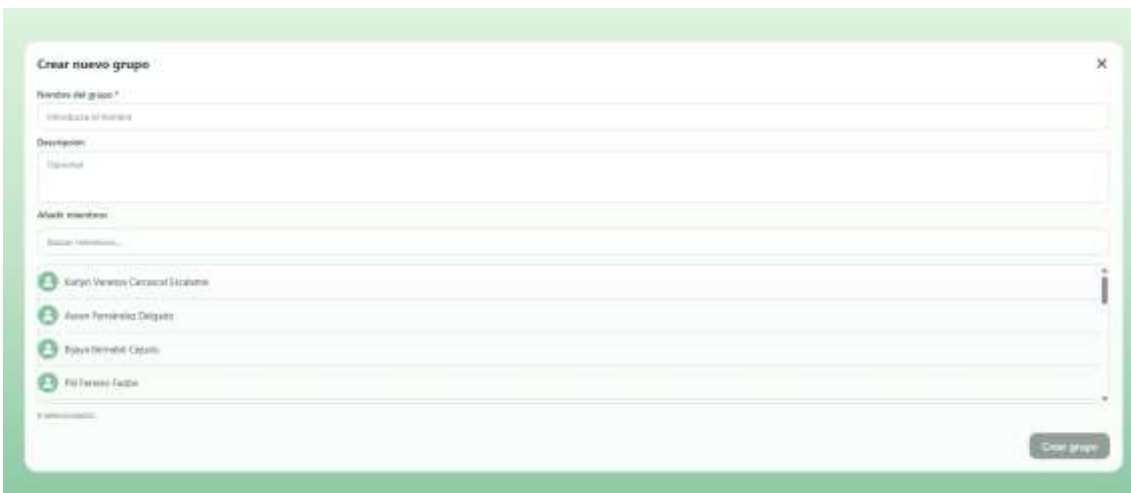


Ilustración 53: Formulario para crear chat grupal



Ilustración 54: Chat grupal con mensajes de sistema

7.3.12 Perfil de usuario

Esta pantalla gestiona la información personal de un usuario. Desde esta sección, el usuario puede editar los datos de su perfil como el nombre, el nombre de usuario y la foto de perfil. El sistema garantiza la exclusividad del nombre de usuario, impidiendo que dos usuarios distintos puedan compartir el mismo identificador.

También se incluye el acceso al código QR que facilita, como ya hemos comentado anteriormente, el seguimiento mutuo entre usuarios de forma rápida y práctica.

El perfil muestra información relevante sobre la actividad del usuario, como el número de seguidores, seguidos y partidos jugados. Además, incorpora un apartado para los equipos y parejas deportivas, desde el cual es posible consultar los equipos a los que pertenece, así como crear nuevos.

Por último, el perfil incluye una sección con el historial de los últimos partidos disputados por el usuario y las estadísticas de juego, destacando el porcentaje de victorias por deporte, lo que permite al usuario visualizar su rendimiento de forma clara y resumida.



Ilustración 55: Perfil del usuario registrado

8 Pruebas y validaciones

Durante todo el periodo de desarrollo se han realizado distintas pruebas con el objetivo de verificar el correcto funcionamiento de las funcionalidades implementadas y así, garantizar una experiencia de usuario adecuada. Dado el carácter práctico del proyecto y el uso de Firebase como Backend as a Service, las pruebas realizadas han sido principalmente de tipo funcional, centrándose en la validación de los flujos principales de la aplicación.

En primer lugar, se han llevado a cabo pruebas relacionadas con el **registro y autenticación de usuarios**, comprobando que:

- El sistema valida correctamente los datos introducidos.
- Impide el registro con campos incompletos o formatos de correos incorrectos.
- Exige la verificación de la cuenta antes de permitir el acceso.

También se ha comprobado el inicio y cierre de sesión, asegurando que la sesión del usuario se mantiene activa hasta que este decide cerrarla manualmente.

Con respecto a la **gestión de partidos y torneos**, se ha realizado pruebas para verificar la creación, edición y eliminación de eventos, así como la inscripción de usuarios tanto de forma individual como por equipos. Además, se ha comprobado que los filtros de búsqueda, la prevención de inscripciones duplicadas y la generación automática de enfrentamientos y ganadores, funciona correctamente.

Respecto a la **gestión de estados de los eventos deportivos**, se ha comprobado los cambios automáticos entre pendientes, próximos, finalizados y la eliminación de aquellos encuentros que no alcanzaban el número mínimo de participantes. Adicionalmente, para completar las pruebas de esta sección, se ha comprobado el funcionamiento de la introducción de **resultados** mediante los formularios específicos y la actualización automática de las estadísticas, últimos partidos y partidos jugados de los usuarios.

La pantalla de actividad y el sistema de notificaciones es preciso y genera correctamente las notificaciones ante los eventos descritos en el capítulo anterior.

En el apartado de **chats**, se han realizado pruebas para verificar el envío y recepción de mensajes en tiempo real, la creación de conversaciones individuales y colectivas, la gestión de miembros, el banner de mensajes no leídos y la correcta visualización de los mensajes del sistema.

Todas las pruebas se han realizado con la participación de distintas personas, con edades comprendidas entre 23 y 60 años, lo que ha permitido detectar y corregir errores a lo largo de todo el desarrollo. Como resultado, se ha conseguido una aplicación funcional estable y coherente con los requisitos definidos.

9 Resultados y conclusiones

9.1 Resultados

En este proyecto se ha logrado el desarrollo completo de una aplicación funcional para la creación y gestión de partidos y torneos dirigida a usuarios amateur. La aplicación integra satisfactoriamente funcionalidades de autenticación y sincronización de datos entre los usuarios, cumpliendo con los objetivos planteados inicialmente. Las pruebas realizadas han permitido verificar su correcto funcionamiento y estabilidad, obteniendo un sistema coherente y utilizable en un entorno real.

Se puede afirmar que el resultado obtenido corresponde con un desarrollo de un MVP (“Minimun Viable Product”).

9.2 Conclusiones

La realización de este Trabajo de fin de Grado ha supuesto un gran reto personal. He podido demostrarme a mí misma que soy capaz de realizar todo lo que me propongo. Ha sido una experiencia muy enriquecedora a nivel personal y académico, ya que me ha permitido desarrollar de manera individual una aplicación desde cero, pasando por todas las fases del proyecto, desde la idea inicial hasta su implementación y validación final.

A lo largo de proceso he tenido la oportunidad de aprender y utilizar nuevos lenguajes y tecnologías con las que no había trabajado con anterioridad. En particular, este proyecto me ha permitido profundizar en el desarrollo frontend, mejorando mis conocimientos acerca del diseño de interfaces y la experiencia de usuario.

Otro punto positivo en mi aprendizaje ha sido el uso de Firebase como plataforma backend. Me ha permitido conocer y utilizar servicios como la autenticación de usuarios y la gestión de bases de datos en tiempo real.

En conjunto, este proyecto ha reforzado mi autonomía como desarrolladora, mi capacidad de organización y resolución de problemas. Me ha permitido consolidar conocimientos y adquirir nuevas competencias útiles para futuros proyectos personales.

9.3 Trabajos Futuros

Aunque la aplicación cumple con los objetivos planteados, existen posibles mejoras y ampliaciones que podrían abordarse en futuros desarrollos.

Una de las líneas de mejora que existe es la incorporación de Firebase Cloud Functions para gestionar procesos automáticos, como el envío de notificaciones push fuera de la aplicación. Esto permitiría informar a los usuarios de eventos importantes incluso cuando no tengan la aplicación abierta, mejorando la experiencia de uso.

Además, también se podría integrar Firebase Storage para la gestión de imágenes, lo que facilitaría el almacenamiento y la organización de las fotos de perfil, de equipo o de las instalaciones. Este cambio enriquecería visualmente la aplicación.

10 Análisis de Impacto

A continuación, se presenta un análisis de impacto de los resultados obtenidos durante la realización del Trabajo Fin de Grado en los siguientes contextos.

10.1 Impacto personal

El impacto personal ha sido muy positivo, ya que el desarrollo del proyecto me ha permitido adquirir nuevos conocimientos técnicos y mejorar competencias relacionadas con el desarrollo de aplicaciones multiplataforma, la resolución de problemas y el trabajo autónomo. Sin embargo, no se han identificado efectos perjudiciales en este ámbito.

10.2 Impacto empresarial

Con respecto al impacto empresarial, esta aplicación puede servir como punto de partida para el desarrollo de un producto digital orientado a la actividad deportiva en el sector del deporte amateur, ya que aporta una solución a una necesidad real.

10.3 Impacto social

La aplicación fomenta la práctica deportiva, la socialización y el ocio saludable, facilitando la comunicación entre personas con intereses comunes. No obstante, un uso excesivo de herramientas digitales podría reducir la interacción presencial.

10.4 Impacto económico

En cuanto al ámbito económico, el impacto es moderado, ya que se trata de una solución digital con costes de desarrollo y mantenimiento relativamente bajos. Como posible efecto adverso, una futura comercialización de la aplicación podría suponer el pago de una cuota anual para acceder a determinadas funcionalidades.

10.5 Impacto medioambiental

El impacto ambiental es reducido, la aplicación no requiere de recursos físicos adicionales y puede contribuir a disminuir el uso de papel en las organizaciones de actividades deportivas. Sin embargo, el uso de infraestructuras en la nube conlleva un consumo energético asociado.

10.6 Impacto cultural.

La aplicación promueve valores como la cooperación, el trabajo en equipo y la participación en actividades deportivas.

10.7 Objetivos de Desarrollo Sostenible

- ODS 3: Salud y bienestar (principal en este proyecto).
- ODS 10: Reducción de las desigualdades (de forma indirecta).
- ODS 11: Ciudades y comunidades sostenibles.

11 Bibliografía

- [1] Celebreak. (2025) “Football Community App” [Online]. Available: <https://celebreak.com/es/>
- [2] PickApp. (2025) “PickApp Basketball” [Online]. Available: <https://pickappbasketball.com/>
- [3] Playtomic. (2025) “Sports Booking Platform” [Online]. Available: <https://playtomic.com/es>
- [4] Flutter. (2025) “Flutter - Documentation” [Online]. Available: <https://flutter.dev/>
- [5] Java y Spring Boot. (2025) “Java and Spring Boot - Documentation” [Online]. Available: <https://www.arquitecturajava.com/spring-boot-que-es/>
<https://www.ibm.com/es-es/think/topics/java-spring-boot>
- [6] React Native. (2025) “React Native - Documentation” [Online]. Available: <https://reactnative.dev/>
- [7] Expo Cli. (2025) “Expo Cli - Documentation” [Online]. Available: <https://docs.expo.dev/more/expo-cli/>
- [8] Expo Go. (2025) “Expo Go - Documentation” [Online]. Available: <https://expo.dev/go>
- [9] TypeScript. (2025) “TypeScript - Documentation” [Online]. Available: <https://www.typescriptlang.org/>
- [10] Expo Router. (2025) “Expo Router - Documentation” [Online]. Available: <https://docs.expo.dev/versions/latest/sdk/router/>
- [11] Firebase Authentication. (2025) “Firebase Authentication - Documentation” [Online]. Available: <https://firebase.google.com/docs/auth?hl=es-419>
- [12] Cloud Firestore. (2025) “Cloud Firestore - Documentation” [Online]. Available: <https://firebase.google.com/docs/firestore?hl=es-419>
- [13] Git. (2025) “Git - Documentation” [Online]. Available: <https://git-scm.com/>
- [14] GitHub. (2025) “GitHub - Documentation” [Online]. Available: <https://github.com/>
- [15] Android Studio. (2025) “Android Developers - Documentation” [Online]. Available: <https://developer.android.com/studio/intro?hl=es-419>
- [16] Figma. (2025) “Figma - Website” [Online]. Available: <https://www.figma.com/es-es/>
- [17] Arquitectura MVVM (Model-View-ViewModel). (2025) “MVVM - Documentation” [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/architecture/maui/mvvm>
- [18] Enlaces al prototipo de Figma. (2025) “Figma- Prototype” [Online]. Available: <https://www.figma.com/design/snLUYdlvke1tY3zkYRgTTY/SPORTCREW?node-id=0-1&t=7uvvxJY00iVO7gMg-1>

<https://www.figma.com/proto/snLUYdlvke1tY3zkYRgTTY/SPORTCREW?node-id=9-15&t=3hFDvUH5AgRyv60J-1&scaling=scale-down&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=9%3A8>

12 Anexos



Recibo digital

Este recibo confirma que su trabajo ha sido recibido por Turnitin. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega: NEREA TANARRO FERRERO
Título del ejercicio: Turnitin Memoria Final
Título de la entrega: TFG_NEREA_TANARRO_FERRERO.pdf
Nombre del archivo: 11178_NEREA_TANARRO_FERRERO_TFG_NEREA_TANARRO_FER...
Tamaño del archivo: 1.03M
Total páginas: 61
Total de palabras: 11,688
Total de caracteres: 73,681
Fecha de entrega: 16-ene-2026 10:25a. m. (UTC+0100)
Identificador de la entrega: 2857779176



Derechos de autor 2026 Turnitin. Todos los derechos reservados.

NEREA TANARRO FERRERO

TFG_NEREA_TANARRO_FERRERO.pdf

-  Turnitin Memoria Final
-  TFG ETSIJNF (Moodle PP)
-  Universidad Politecnica de Madrid

Detalles del documento

Identificador de la entrega

trn:oid:::1:3458603540

Fecha de entrega

16 ene 2026, 10:25 a.m. GMT+1

Fecha de descarga

22 ene 2026, 10:57 a.m. GMT+1

Nombre del archivo

11178_NEREA_TANARRO_FERRERO_TFG_NEREA_TANARRO_FERRERO_530462_1250201993.pdf

Tamaño del archivo

1.0 MB

61 páginas

11.688 palabras

73.681 caracteres




6% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...


Filtrado desde el informe

- Bibliografía
- Coincidencias menores (menos de 10 palabras)

Fuentes principales

- 6%  Fuentes de Internet
- 0%  Publicaciones
- 0%  Trabajos entregados (trabajos del estudiante)

Este documento esta firmado por



Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
Fecha/Hora	Thu Jan 22 22:04:48 CET 2026
Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
Numero de Serie	561
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)