



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Alcoy

FutApp, una app para gestionar información sobre partidos
de fútbol

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Lopez Terrades, Hector

Tutor/a: Esparza Peidro, Javier

CURSO ACADÉMICO: 2023/2024

Resumen

El proyecto consiste en una aplicación móvil para la gestión de partidos de fútbol en Android, permitiendo a los usuarios registrar, visualizar y gestionar información detallada sobre partidos, jugadores y equipos. Utilizando Firebase Realtime Database, ofrece una interfaz intuitiva con Fragments y RecyclerViews, además de un cronómetro en tiempo real para registrar eventos durante los partidos.

Palabras clave: Aplicación, Android, Fútbol, Balompié, Estadísticas, Deporte, Datos.

Abstract

The project consists of a mobile application for football match management on Android, allowing users to record, view and manage detailed information about matches, players and teams. Using Firebase Realtime Database, it offers an intuitive interface with Fragments and RecyclerViews, as well as a real-time timer to record events during matches.

Keywords: Application, Android, Football, Soccer, Football, Statistics, Sport, Data.

Resum

El projecte consisteix en una aplicació mòbil per a la gestió de partits de futbol en Android, permetint als usuaris registrar, visualitzar i gestionar informació detallada sobre partits, jugadors i equips. Utilitzant Firebase Realtime Database, oferix una interfície intuïtiva amb Fragments i RecyclerViews, a més amb un cronòmetre en temps real per a registrar events durant els partits.

Paraules clau: Aplicació, Android, Futbol, Estadístiques, Esport, Dades.

Índice general

Resumen	I
Abstract	I
Resum	I
1 Introducción	1
1.1 Motivación	1
1.2 Estudio del mercado	1
1.3 Objetivos del proyecto	2
1.4 Estructura del resto de la memoria	2
2 Análisis o definición del problema	3
2.1 Requisitos funcionales	3
2.1.1 Gestión de equipos	3
2.2 Gestión de jugadores	3
2.2.1 Gestión de partidos	3
2.2.2 Seguimiento en tiempo real	3
2.2.3 Estadísticas y clasificación	4
2.3 Requisitos no funcionales	4
2.4 Diagramas UML	5
3 Diseño o solución del problema	6
3.1 Arquitectura de la aplicación	6
3.1.1 Componentes Principales de la Arquitectura:	6
3.1.2 Comunicación Cliente-Servidor	7
3.2 Tecnologías utilizadas	7
3.2.1 Android SDK	7
3.2.2 Java	8
3.2.3 Google Firebase	8
3.3 Componentes principales	8
3.3.1 Capa del cliente: Aplicación Android	9
3.3.1.1 MainActivity	9
3.3.1.2 Fragment	9
3.3.1.2.1 PartidosFragment	10
3.3.1.2.2 JugadoresFragment	10
3.3.1.2.3 EquiposFragment	11
3.3.1.2.4 EstadísticasFragment	11
3.3.1.2.5 ClasificaciónFragment	12
3.3.1.3 Adapters	12
3.3.1.3.1 PartidosAdapter	13
3.3.1.3.2 JugadoresAdapter	13
3.3.1.3.3 EquiposAdapter	13
3.3.1.3.4 EstadísticasAdapter	14
3.3.1.3.5 ClasificaciónAdapter	14
3.3.2 Capa del cliente: Tipos de datos utilizados para almacenamiento	15
3.3.2.1 Clase Equipo	15

3.3.2.2	Clase Evento	16
3.3.2.3	Clase Jugador	16
3.3.2.4	Clase Partido	17
3.3.3	Capa del servidor: Firebase	18
3.3.3.1	Firebase Realtime Database	18
3.3.3.2	Firebase Authentication	18
3.3.3.3	Firebase Storage	19
4	Resultado	20
4.1	Activity Login	20
4.2	Fragment Partidos	21
4.3	Fragment Jugadores	24
4.4	Fragment Equipos	26
4.5	Fragment Estadísticas	28
4.6	Fragment Clasificación	30
5	Conclusión	31
6	Anexos	32
6.1	Estructura base de datos	32
7	Bibliografía	34

Índice de figuras

1	Diagrama de casos de uso de la aplicación.	5
2	Diagrama de la arquitectura de la aplicación.	6
3	Diagrama de los fragments de la aplicación.	9
4	Diagrama de la clase Equipo.	15
5	Diagrama de la clase Evento.	16
6	Diagrama de la clase Jugador.	16
7	Diagrama de la clase Partit.	17
8	Captura de pantalla de la activity de Login.	20
9	Captura de pantalla del fragment de Partidos.	21
10	Captura de pantalla de los eventos de un partido.	22
11	Captura de pantalla de la selección de jugadores que van a disputar el encuentro.	22
12	Captura de pantalla de la selección de jugadores que van a disputar el encuentro.	23
13	Captura de pantalla del fragment de Jugadores.	24
14	Captura de pantalla para añadir a un jugador.	25
15	Captura de pantalla del dialogo de Jugadores.	25
16	Captura de pantalla del fragment de Equipos.	26
17	Captura de pantalla de añadir un equipo.	27
18	Captura de pantalla de los jugadores del equipo.	27
19	Captura de pantalla de las estadísticas.	28
20	Captura del fragment de Clasificación.	30
21	Captura de la base de datos almacenada en Google Firebase.	33

1 Introducción

1.1 Motivación

La motivación detrás de la creación de FutApp surge de la creciente necesidad de herramientas tecnológicas en el ámbito deportivo, especialmente en el fútbol. En los últimos años, la digitalización ha transformado muchos aspectos de la vida cotidiana, y el deporte no es una excepción.

Sin embargo, la gestión de partidos de fútbol, especialmente a nivel amateur y semi-profesional, aún depende en gran medida de métodos manuales y poco eficientes. Este proyecto se propone llenar ese vacío proporcionando una solución moderna y accesible para la gestión de partidos en tiempo real.

La novedad de FutApp reside en su capacidad para registrar y actualizar eventos del partido en tiempo real, algo que pocas aplicaciones en el mercado ofrecen de manera integral. Además, la aplicación no solo se centra en la gestión de eventos, sino que también proporciona un análisis detallado de las estadísticas del partido y una clasificación dinámica de los equipos. Esto facilita a entrenadores, jugadores y aficionados el seguimiento del rendimiento de sus equipos y jugadores favoritos.

1.2 Estudio del mercado

Para entender mejor la necesidad y posicionamiento de FutApp, se realizó un estudio de aplicaciones similares disponibles en el mercado. Entre ellas se encuentran:

- TeamSnap: Es una aplicación muy popular para la gestión de equipos deportivos, pero su enfoque principal es la organización de equipos y la programación de eventos, sin un énfasis en el seguimiento en tiempo real de los partidos.
- My Football Manager: Esta aplicación se centra en la gestión de equipos y jugadores, permitiendo a los usuarios asumir el rol de manager de un equipo de fútbol. Aunque ofrece una amplia gama de funcionalidades, carece de una herramienta robusta para la actualización en tiempo real de eventos durante el partido.
- FútbolEmotion: Es una aplicación dirigida principalmente a aficionados, ofreciendo noticias, resultados y estadísticas. Sin embargo, no proporciona herramientas para la gestión activa de partidos y eventos en tiempo real.

La principal carencia de estas aplicaciones es la falta de una solución completa y en tiempo real para la gestión de partidos, lo cual es el objetivo principal de FutApp. Nuestra aplicación está diseñada para llenar este vacío y proporcionar una herramienta integral que cubra todas las necesidades de gestión y seguimiento de partidos de fútbol.

1.3 Objetivos del proyecto

El proyecto FutApp tiene como objetivo principal desarrollar una aplicación móvil que facilite la gestión integral de partidos de fútbol en tiempo real. Los objetivos específicos incluyen:

- **Gestión de equipos:** Permitir a los usuarios crear, editar y administrar equipos de fútbol, incluyendo la asignación y gestión de jugadores.
- **Gestión de partidos:** Facilitar la creación, edición y administración de partidos de fútbol, incluyendo la programación y el seguimiento de eventos en tiempo real.
- **Seguimiento en tiempo real:** Proporcionar una herramienta para registrar y actualizar eventos durante el partido (goles, tarjetas), con un cronómetro en vivo.
- **Estadísticas y clasificación:** Ofrecer una visualización detallada de las estadísticas de partidos y jugadores, así como una clasificación dinámica de los equipos basada en los resultados de los partidos.
- **Interfaz intuitiva:** Diseñar una interfaz de usuario intuitiva y accesible que permita una experiencia de usuario fluida y eficiente.

1.4 Estructura del resto de la memoria

La estructura de la memoria del proyecto está organizada para proporcionar una visión clara y detallada del desarrollo y funcionamiento de FutApp. A continuación, se describe brevemente el contenido de cada sección:

- **Análisis o definición del problema:** En esta sección se detallan los requisitos y funcionalidades de la aplicación, utilizando texto plano y diagramas UML como el diagrama de casos de uso. La idea es describir con todo lujo de detalles el problema que se pretende resolver, es decir, la aplicación que se pretende construir, sin incluir detalles de implementación.
- **Diseño o solución del problema:** Aquí se describe cómo se ha construido la aplicación a un nivel alto, utilizando diagramas de bloques, dibujos y UML para describir la arquitectura de la aplicación. Se incluye una descripción de las distintas capas de la aplicación y las tecnologías utilizadas.
 - **Arquitectura de la aplicación:** Se describen las capas de la aplicación, su función y cómo se conectan entre sí (por ejemplo, GUI, lógica de negocio, capa de datos).
 - **Tecnologías utilizadas:** Se presentan las tecnologías empleadas y se explica brevemente por qué se eligieron.
 - **Descripción de capas o componentes:** Se detalla cada capa o componente de la aplicación, describiendo su función y cómo se integra en la arquitectura general.
- **Resultado o tour por la aplicación:** Esta sección ofrece una demostración de la aplicación, utilizando capturas de pantalla y descripciones para mostrar las principales funcionalidades y la interfaz de usuario.
- **Conclusiones y trabajos futuros:** Se resumen los logros del proyecto, se presentan impresiones personales sobre el desarrollo del proyecto y se sugieren posibles mejoras y expansiones futuras.

- **Bibliografía:** Se enumeran todas las fuentes y referencias utilizadas a lo largo del proyecto.

2 Análisis o definición del problema

En esta sección, se detallarán los requisitos y funcionalidades de la aplicación FutApp. Utilizaremos tanto texto plano como diagramas UML para describir el problema a resolver y la solución propuesta. El objetivo es proporcionar una visión exhaustiva y clara de lo que se pretende construir, sin entrar en detalles de implementación.

2.1 Requisitos funcionales

Los requisitos funcionales son aquellos que describen las funcionalidades específicas que debe tener la aplicación para cumplir con los objetivos planteados.

2.1.1 Gestión de equipos

- **Crear equipo:** El usuario debe poder crear un equipo nuevo, especificando nombre, ciudad, estadio y otros detalles relevantes.
- **Visualizar equipo:** El usuario debe poder visualizar los datos del equipo.

2.2 Gestión de jugadores

- **Añadir jugadores:** El usuario debe poder añadir jugadores a un equipo, especificando nombre, apellidos y número.
- **Visualizar jugadores:** El usuario debe poder visualizar los datos del jugador, tanto como sus estadísticas

2.2.1 Gestión de partidos

- **Crear partido:** El usuario debe poder crear un partido nuevo, especificando los equipos participantes, fecha y hora.
- **Editar partido:** El usuario debe poder editar la información de un partido existente.
- **Visualizar partidos:** El usuario debe poder ver una lista de partidos jugados.

2.2.2 Seguimiento en tiempo real

- **Iniciar cronómetro:** El usuario debe poder iniciar el cronómetro al comienzo del partido.
- **Detener cronómetro:** El usuario debe poder detener el cronómetro en cualquier momento.
- **Registrar eventos:** El usuario debe poder registrar eventos durante el partido (goles, tarjetas).

- **Pausa para descanso:** El usuario debe poder pausar el cronómetro para el descanso y reiniciarlo desde el minuto 45 en la segunda mitad.

2.2.3 Estadísticas y clasificación

- **Registrar resultados:** Al finalizar un partido, el usuario debe poder registrar los resultados.
- **Actualizar estadísticas:** La aplicación debe actualizar automáticamente las estadísticas de equipos y jugadores basadas en los resultados.
- **Visualizar clasificación:** El usuario debe poder ver la clasificación de equipos basada en puntos, victorias, empates, derrotas, y diferencia de goles.

2.3 Requisitos no funcionales

Los requisitos no funcionales describen las cualidades y restricciones del sistema que afectan a su operación y desarrollo.

- **Usabilidad:** La interfaz de usuario debe ser intuitiva y fácil de usar.
- **Rendimiento:** La aplicación debe responder rápidamente a las interacciones del usuario.
- **Seguridad:** Los datos del usuario deben estar protegidos contra accesos no autorizados.
- **Compatibilidad:** La aplicación debe ser compatible con la mayoría de los dispositivos Android.
- **Mantenibilidad:** El código debe estar bien documentado y estructurado para facilitar el mantenimiento y la expansión futura.

2.4 Diagramas UML

El diagrama de casos de uso muestra las interacciones entre los actores (usuarios) y la aplicación, destacando las funcionalidades principales.

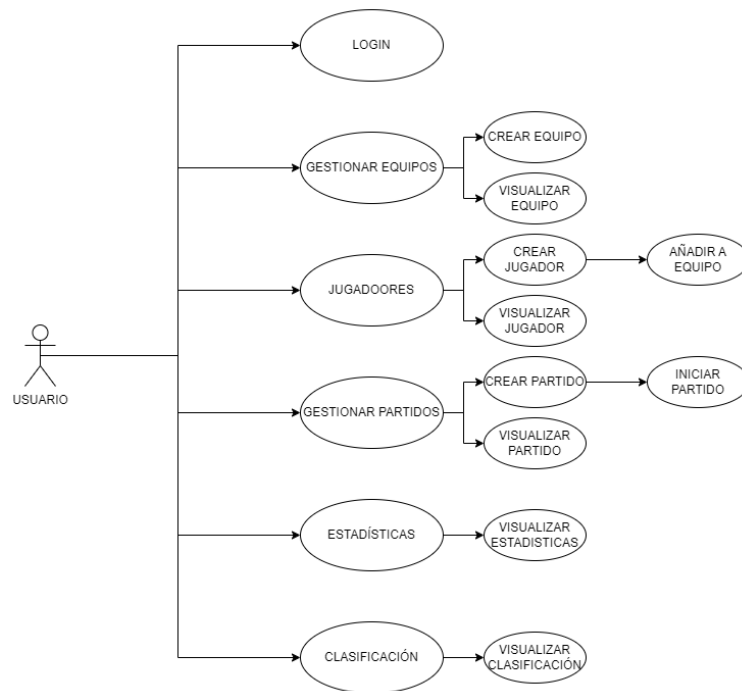


Figura 1: Diagrama de casos de uso de la aplicación.

Como hemos podido observar en la figura superior se muestra el diagrama de casos de uso de la aplicación FutApp donde podemos observar las 6 funcionalidades básicas de la aplicación donde en próximas líneas entraremos en mayor profundidad.

3 Diseño o solución del problema

3.1 Arquitectura de la aplicación

La arquitectura de la aplicación está basada en un modelo cliente-servidor, donde la aplicación Android actúa como cliente y Firebase Realtime Database proporciona el backend como servidor. Este enfoque permite una comunicación eficiente y segura entre la interfaz de usuario y el almacenamiento de datos, facilitando la gestión de partidos, jugadores y equipos en tiempo real.

3.1.1 Componentes Principales de la Arquitectura:

1) **Cliente** (Aplicación Android)

- **Fragments:** La aplicación está estructurada en varios fragments, cada uno de los cuales gestiona una funcionalidad específica (partidos, jugadores, equipos, estadísticas y clasificación). Estos fragments proporcionan una interfaz de usuario intuitiva y responden a las acciones del usuario.
- **Adapters:** Los adapters se encargan de la interacción con Firebase Realtime Database, recuperando y enviando datos según sea necesario. Cada fragment utiliza adapters para gestionar la visualización de los datos en RecyclerViews.
- **Actividades:** Además de los fragments, la aplicación incluye actividades específicas como la del cronómetro, que permite registrar eventos en tiempo real durante los partidos.

2) **Servidor**(Firebase Realtime Database)

- **Base de datos en tiempo real:** Firebase actúa como el backend de la aplicación, almacenando toda la información relacionada con partidos, jugadores, equipos y eventos. La base de datos en tiempo real permite la sincronización instantánea de los datos entre todos los clientes conectados.
- **Autenticación y seguridad:** Firebase también proporciona servicios de autenticación y seguridad, asegurando que solo los usuarios autorizados puedan acceder y modificar los datos.



Figura 2: Diagrama de la arquitectura de la aplicación.

3.1.2 Comunicación Cliente-Servidor

La comunicación entre la aplicación Android (cliente) y Firebase Realtime Database (servidor) se realiza mediante llamadas a la API de Firebase. Estas llamadas permiten:

- **Lectura de datos:** Los adapters consultan Firebase para recuperar información sobre partidos, jugadores y equipos, que luego se muestran en los RecyclerViews de los fragments correspondientes.
- **Escritura de datos:** Las acciones del usuario, como añadir un nuevo partido, actualizar el resultado de un partido o registrar un evento durante un partido, se envían a Firebase para su almacenamiento.
- **Actualización en tiempo real:** Gracias a la funcionalidad de base de datos en tiempo real de Firebase, cualquier cambio en los datos se refleja instantáneamente en la aplicación, proporcionando una experiencia de usuario fluida y actualizada.

3.2 Tecnologías utilizadas

Las tecnologías empleadas para el desarrollo de la aplicación incluyen:

- **Android SDK y Java:** Para el desarrollo de la aplicación móvil.
- **Firebase Realtime Database:** Para la gestión y almacenamiento de datos en tiempo real.
- **UML (Unified Modeling Language):** Para la modelización del sistema y la creación de diagramas de casos de uso, secuencia y clases.
- **Herramientas de diseño:** Draw.io para la creación de diagramas UML para la creación de prototipos de UI.

Estas tecnologías fueron seleccionadas por su robustez, popularidad y amplia documentación, lo que facilita el desarrollo y el mantenimiento de la aplicación.

3.2.1 Android SDK

El Android SDK (Software Development Kit) es un conjunto integral de herramientas proporcionado por Google para el desarrollo de aplicaciones en el sistema operativo Android. Este kit incluye Android Studio, el entorno de desarrollo integrado (IDE) oficial, que ofrece un editor de código avanzado, herramientas de depuración y un emulador para probar aplicaciones en diversas configuraciones de hardware y versiones de Android. Además, el SDK proporciona bibliotecas y APIs que permiten a los desarrolladores interactuar con las funciones del sistema operativo, así como herramientas de compilación como Gradle, que facilitan la gestión de dependencias y automatización del proceso de construcción.

El Android SDK también incluye documentación extensa, ejemplos de código y herramientas como el Android Debug Bridge (ADB), que permite gestionar dispositivos Android desde la línea de comandos. Con actualizaciones frecuentes y un amplio soporte comunitario, el Android SDK es una herramienta esencial que facilita la creación de aplicaciones robustas, eficientes y compatibles con una amplia gama de dispositivos y versiones de Android, asegurando que los desarrolladores puedan maximizar el potencial de la plataforma Android. Google, 2024-b

3.2.2 Java

Java es un lenguaje de programación de alto nivel, orientado a objetos, y ampliamente utilizado en el desarrollo de aplicaciones para diversas plataformas, incluyendo móviles y empresariales. Creado por Sun Microsystems y ahora propiedad de Oracle Corporation, Java se destaca por su portabilidad, gracias a su capacidad de ejecutarse en cualquier dispositivo con una Máquina Virtual de Java (JVM) compatible. Este principio "write once, run anywhere" (escribir una vez, ejecutar en cualquier lugar) permite a los desarrolladores crear aplicaciones que funcionen en múltiples sistemas operativos sin necesidad de modificar el código fuente.

Java proporciona una rica API (Application Programming Interface) con numerosas bibliotecas para manejar tareas comunes como manipulación de datos, redes, entrada/salida (I/O), y interfaces gráficas de usuario (GUI). Su sistema de manejo de excepciones y recolección automática de basura contribuyen a la creación de código eficiente y confiable. Con un fuerte soporte comunitario y herramientas de desarrollo avanzadas, Java sigue siendo una opción popular para desarrolladores que buscan crear aplicaciones robustas, seguras y escalables. Formate, 2021

3.2.3 Google Firebase

Google Firebase es una plataforma de desarrollo de aplicaciones que proporciona una variedad de servicios backend, como bases de datos en tiempo real, autenticación, almacenamiento en la nube y hosting, entre otros. Firebase Realtime Database, en particular, es una base de datos NoSQL que permite a los desarrolladores almacenar y sincronizar datos entre sus usuarios en tiempo real. La elección de Firebase para este proyecto se basa en su integración directa con el ecosistema de Google, su facilidad de uso y la rapidez con la que permite implementar y escalar aplicaciones. Jose Luis Bustos, 2024

A diferencia de otras tecnologías de bases de datos como MongoDB, Firebase ofrece una solución más completa y orientada a las aplicaciones móviles. Firebase no solo proporciona una base de datos, sino también un conjunto de herramientas integradas que facilitan el desarrollo y la gestión de aplicaciones móviles. Por ejemplo, Firebase Authentication simplifica la gestión de usuarios, mientras que Firebase Cloud Messaging facilita el envío de notificaciones push. Además, la documentación extensa y el soporte comunitario de Firebase hacen que sea una opción más accesible y eficiente para desarrolladores individuales o pequeños equipos. En comparación, MongoDB requiere una mayor configuración y gestión de infraestructura, lo cual puede ser una barrera significativa para proyectos que buscan una solución rápida y escalable. Google, 2024-a

3.3 Componentes principales

La arquitectura de la aplicación está basada en un modelo cliente-servidor. La aplicación móvil actúa como el cliente, que interactúa con el usuario y realiza solicitudes a un servidor remoto, que en este caso es Firebase Realtime Database. Esta estructura permite que los datos se almacenen y gestionen de manera centralizada, proporcionando una sincronización en tiempo real entre los usuarios. A continuación, se detallan las dos capas principales: la capa del cliente y la capa del servidor.

3.3.1 Capa del cliente: Aplicación Android

La capa del cliente se encarga de la interacción del usuario con la aplicación. Está compuesta por una serie de actividades y fragmentos que proporcionan una interfaz intuitiva y funcional, utilizando tecnologías como RecyclerView y Adapters para manejar y mostrar los datos. StackOverFlow, 2024

3.3.1.1 MainActivity

La **MainActivity** es la actividad principal que actúa como contenedor para los diferentes fragmentos de la aplicación. Utiliza un NavigationDrawer para permitir la navegación entre los distintos fragmentos, proporcionando una experiencia de usuario fluida y organizada. Esta actividad también maneja la configuración inicial y los elementos de interfaz comunes que se comparten entre los fragmentos.

3.3.1.2 Fragment

En el contexto de nuestra aplicación, un fragment es un elemento crucial que permite representar y gestionar diferentes partes de la interfaz de usuario de manera independiente dentro de una actividad en Android. Cada fragmento de nuestra aplicación cumple una función específica, como la gestión de partidos, jugadores, equipos, estadísticas y clasificación. Estos fragmentos se combinan para formar una interfaz intuitiva y dinámica que ofrece a los usuarios una experiencia fluida y eficiente al interactuar con la aplicación.

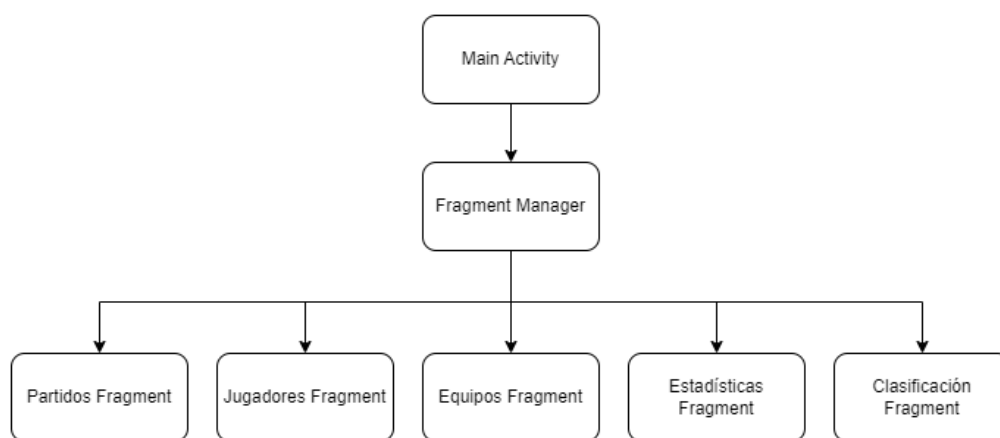


Figura 3: Diagrama de los fragments de la aplicación.

3.3.1.2.1 PartidosFragment

El **PartidosFragment** desempeña un papel fundamental en la gestión de los partidos de fútbol dentro de la aplicación. Este fragmento permite a los usuarios acceder y gestionar fácilmente los partidos programados y disputados, ofreciendo una interfaz intuitiva y funcionalidades avanzadas para el seguimiento detallado de cada encuentro.

Una de las características destacadas del PartidosFragment es su capacidad para mostrar una lista completa de partidos, permitiendo a los usuarios visualizar rápidamente los detalles de cada encuentro, como la fecha, los equipos participantes y el resultado final. Además, este fragmento ofrece la posibilidad de añadir nuevos partidos mediante un botón flotante (FAB), lo que facilita la programación y organización de futuros encuentros.

Al añadir un nuevo partido desde el PartidosFragment, los usuarios tienen la oportunidad de seleccionar los jugadores participantes para cada equipo. Esta funcionalidad garantiza que solo los jugadores inscritos en los equipos puedan ser incluidos en los partidos, lo que contribuye a una gestión precisa y ordenada de los encuentros.

Una vez iniciado un partido, el PartidosFragment ofrece una vista del cronómetro en tiempo real, que permite a los usuarios registrar eventos importantes, como goles y tarjetas, mientras el partido está en curso. Esta característica proporciona una herramienta invaluable para el seguimiento y análisis detallado de cada encuentro, permitiendo a los usuarios registrar y revisar eventos específicos en tiempo real.

En resumen, el PartidosFragment es una parte integral de la aplicación de gestión de equipos y partidos de fútbol, ofreciendo una amplia gama de funcionalidades para la gestión eficiente y el seguimiento detallado de los partidos. Desde la visualización de la lista de partidos hasta la gestión de eventos en tiempo real, este fragmento proporciona una experiencia completa y satisfactoria para los usuarios que desean administrar y seguir de cerca los encuentros de fútbol.

3.3.1.2.2 JugadoresFragment

El **JugadoresFragment** desempeña un papel crucial en la gestión de los jugadores dentro de la aplicación de fútbol. Este fragmento proporciona a los usuarios una vista completa y detallada de todos los jugadores inscritos en la liga, permitiéndoles visualizar y actualizar sus estadísticas de manera eficiente.

Una de las principales características del JugadoresFragment es su capacidad para mostrar una lista exhaustiva de jugadores, incluyendo detalles como sus nombres, números de camiseta y estadísticas individuales, como goles y tarjetas recibidas. Esta funcionalidad permite a los usuarios realizar un seguimiento preciso del rendimiento de cada jugador a lo largo de la temporada.

Además de la visualización de jugadores existentes, el JugadoresFragment también ofrece la posibilidad de añadir nuevos jugadores a la base de datos. Al pulsar el botón flotante para añadir jugadores, los usuarios pueden completar fácilmente los detalles del nuevo jugador, como su nombre, número de camiseta y equipo al que pertenece. Esta funcionalidad simplifica el proceso de incorporación de nuevos jugadores a la liga, garantizando que todos los datos estén debidamente registrados y actualizados en la aplicación.

En resumen, el JugadoresFragment es una parte esencial de la aplicación de gestión de equipos y partidos de fútbol, proporcionando a los usuarios una plataforma centralizada para la gestión eficiente de los jugadores. Ya sea para visualizar estadísticas existentes o añadir nuevos jugadores a la base de datos, este fragmento ofrece una experiencia completa y satisfactoria para los usuarios que desean administrar y seguir de cerca el desempeño de los jugadores en la liga.

3.3.1.2.3 EquiposFragment

El **EquiposFragment** desempeña un papel fundamental en la visualización y gestión de los equipos de fútbol dentro de la aplicación. Este fragmento proporciona a los usuarios una vista organizada y completa de todos los equipos registrados en la base de datos, facilitando así la navegación y la interacción con la información del equipo.

Una de las principales características del EquiposFragment es su capacidad para mostrar una lista exhaustiva de equipos, donde cada elemento muestra el nombre del equipo junto con su escudo, si está disponible en la base de datos. Esta vista intuitiva permite a los usuarios identificar rápidamente los equipos deseados y acceder a información adicional sobre ellos con solo un toque.

Además de la visualización de equipos, el EquiposFragment también ofrece la funcionalidad de explorar los jugadores asociados a cada equipo. Al seleccionar un equipo de la lista, el fragmento despliega una lista detallada de jugadores que pertenecen a ese equipo en particular. Esta característica permite a los usuarios obtener una visión completa de la plantilla de jugadores de cada equipo y realizar un seguimiento de sus actuaciones individuales a lo largo de la temporada.

En resumen, el EquiposFragment sirve como una herramienta valiosa para la gestión y exploración de equipos de fútbol en la aplicación. Ya sea para visualizar información básica sobre equipos o explorar la lista de jugadores asociados a cada equipo, este fragmento ofrece una experiencia intuitiva y eficiente para los usuarios que desean obtener una visión completa del panorama futbolístico en la liga.

3.3.1.2.4 EstadísticasFragment

El **EstadísticasFragment** desempeña un papel crucial en el análisis del rendimiento de jugadores y equipos a lo largo del torneo. Este fragmento permite a los usuarios seleccionar y visualizar estadísticas específicas, como goles, tarjetas rojas y tarjetas amarillas, utilizando un spinner intuitivo.

Una de las características destacadas del EstadísticasFragment es su capacidad para ofrecer una amplia gama de estadísticas que pueden ser seleccionadas y exploradas por los usuarios según sus necesidades específicas. Al utilizar el spinner, los usuarios pueden elegir el tipo de estadística que desean visualizar, lo que proporciona una flexibilidad significativa en el análisis del rendimiento.

Además de la visualización de estadísticas, el EstadísticasFragment también ofrece herramientas para el análisis y la interpretación de los datos presentados. Los usuarios pueden examinar tendencias a lo largo del tiempo, identificar patrones de juego y evaluar el impacto de diferentes estrategias en el rendimiento general de los jugadores y equipos.

En resumen, el EstadísticasFragment es una herramienta poderosa para el análisis y la visualización de estadísticas en la aplicación. Con su capacidad para seleccionar y explorar diversas estadísticas de manera intuitiva, este fragmento proporciona a los usuarios una visión completa del rendimiento de jugadores y equipos, facilitando así la toma de decisiones informadas y la mejora continua en el campo de juego.

3.3.1.2.5 ClasificaciónFragment

El **ClasificaciónFragment** desempeña un papel fundamental en la presentación de la clasificación de los equipos en la aplicación. Utilizando un RecyclerView, este fragmento muestra la clasificación actualizada de los equipos basada en los puntos acumulados a lo largo del torneo.

Una de las características destacadas del ClasificaciónFragment es su capacidad para presentar la información de manera ordenada y clara, permitiendo a los usuarios visualizar fácilmente la posición de cada equipo en la clasificación. Los equipos se clasifican según su rendimiento en los partidos, reflejando su éxito en términos de victorias, empates y derrotas.

Además de mostrar la clasificación actual, el ClasificaciónFragment también ofrece la funcionalidad de actualización en tiempo real, lo que significa que la clasificación se ajusta automáticamente en función de los resultados de los partidos. Esto garantiza que los usuarios siempre tengan acceso a la información más reciente sobre el rendimiento de los equipos en el torneo.

En resumen, el ClasificaciónFragment es una herramienta esencial para los usuarios que desean seguir de cerca la evolución de la clasificación en la aplicación. Con su capacidad para presentar la información de manera clara y actualizada, este fragmento proporciona una visión completa y precisa del rendimiento de los equipos a lo largo del torneo.

3.3.1.3 Adapters

Los adaptadores desempeñan un papel crucial en la gestión de la interacción entre la aplicación y Firebase Realtime Database, así como en la actualización dinámica de la interfaz de usuario. Cada adaptador está diseñado para manejar un conjunto específico de datos y facilitar su representación en los RecyclerView correspondientes.

Cada vez que se produce un cambio en los datos almacenados en Firebase, los adaptadores son responsables de actualizar automáticamente la interfaz de usuario para reflejar estos cambios en tiempo real. Esto garantiza una experiencia fluida y actualizada para los usuarios, ya que cualquier modificación en los datos se refleja instantáneamente en la pantalla.

Además de su función principal de actualizar la interfaz de usuario, los adaptadores también proporcionan una capa de abstracción entre los datos almacenados en Firebase y la representación visual en la aplicación. Esto simplifica el proceso de gestión de datos y permite una integración más eficiente con Firebase Realtime Database.

En resumen, los adaptadores son componentes esenciales en la arquitectura de la aplicación, ya que facilitan la interacción con Firebase y garantizan una representación precisa y actualizada de los datos en la interfaz de usuario.

3.3.1.3.1 PartidosAdapter

El **PartidosAdapter** desempeña un papel fundamental en la gestión de los datos de los partidos y los eventos asociados en la aplicación. Este adaptador se encarga de recuperar y actualizar la información almacenada en Firebase Realtime Database para garantizar que los datos presentados en el **PartidosFragment** estén siempre sincronizados y sean consistentes con la base de datos en la nube.

Al recuperar los datos de los partidos y los eventos asociados desde Firebase, el **PartidosAdapter** se asegura de que la información mostrada en la interfaz de usuario sea precisa y actualizada. Además, este adaptador también facilita la interacción con los datos, permitiendo a los usuarios ver los detalles de los partidos y acceder a la información detallada de los eventos registrados durante cada partido.

La sincronización bidireccional entre el **PartidosAdapter** y Firebase Realtime Database garantiza que cualquier cambio realizado en los datos, ya sea desde la aplicación o desde la base de datos en la nube, se refleje instantáneamente en la interfaz de usuario. Esto proporciona una experiencia fluida y sin problemas para los usuarios, ya que siempre tienen acceso a la información más reciente y actualizada sobre los partidos y eventos de fútbol.

3.3.1.3.2 JugadoresAdapter

El **JugadoresAdapter** desempeña un papel crucial en la gestión de los datos de los jugadores y sus estadísticas en la aplicación. Este adaptador se encarga de recuperar la información de los jugadores almacenada en Firebase Realtime Database y presentarla de manera precisa en el **JugadoresFragment**, donde los usuarios pueden visualizar las estadísticas de los jugadores de fútbol.

Al interactuar con Firebase Realtime Database, el **JugadoresAdapter** garantiza que la información mostrada en el **JugadoresFragment** esté siempre sincronizada y sea consistente con la base de datos en la nube. Esto permite a los usuarios ver los detalles de cada jugador, incluidas sus estadísticas de goles y tarjetas.

La funcionalidad del **JugadoresAdapter** permite una visualización clara y organizada de los datos de los jugadores, lo que facilita a los usuarios gestionar el equipo y hacer un seguimiento del rendimiento de cada jugador a lo largo de la temporada. Gracias a la sincronización en tiempo real con Firebase, los cambios realizados en las estadísticas de los jugadores se reflejan de inmediato en la aplicación, brindando una experiencia fluida y actualizada para los usuarios.

3.3.1.3.3 EquiposAdapter

El **EquiposAdapter** desempeña un papel esencial en la gestión de los datos de los equipos en la aplicación. Este adaptador se encarga de interactuar con Firebase Realtime Database para recuperar y actualizar la información de los equipos, garantizando que la información mostrada en el **EquiposFragment** esté siempre actualizada y sea coherente con la base de datos en la nube.

Al utilizar el **EquiposAdapter**, los usuarios pueden visualizar una lista de todos los equipos almacenados en la base de datos y acceder a detalles específicos de cada equipo, como su nombre, estadio y ciudad. Además, el adaptador facilita la gestión de la lista de equipos, permitiendo a los usuarios añadir nuevos equipos o realizar cualquier actualización necesaria en los datos existentes.

La sincronización en tiempo real con Firebase garantiza que cualquier cambio realizado en la base de datos, ya sea añadir un nuevo equipo o actualizar la información de un equipo existente, se refleje de inmediato en la interfaz de usuario del **EquiposFragment**. Esto proporciona una experiencia de usuario fluida y actualizada, permitiendo a los usuarios gestionar eficazmente la información de los equipos y mantenerse al día con los detalles de la liga de fútbol.

3.3.1.3.4 EstadísticasAdapter

El **EstadísticasAdapter** desempeña un papel fundamental en la presentación de las estadísticas en el **EstadísticasFragment** de la aplicación. Este adaptador se encarga de gestionar la visualización de las estadísticas específicas, como los goles, tarjetas amarillas y tarjetas rojas, permitiendo a los usuarios seleccionar y ver las estadísticas de su interés.

Al utilizar el **EstadísticasAdapter**, los usuarios pueden interactuar con un spinner para seleccionar el tipo de estadística que desean visualizar. Una vez seleccionada la estadística deseada, el adaptador recupera los datos correspondientes de Firebase Realtime Database y los presenta de manera clara y organizada en el **EstadísticasFragment**.

Este adaptador garantiza que los usuarios puedan analizar el rendimiento de los jugadores y equipos en función de diferentes métricas, lo que les permite tomar decisiones informadas sobre la estrategia del equipo y evaluar el desempeño de los jugadores en cada partido. La presentación ordenada y precisa de las estadísticas facilita la comprensión y el análisis de los datos, mejorando así la experiencia de usuario en la aplicación.

3.3.1.3.5 ClasificaciónAdapter

El **ClasificaciónAdapter** desempeña un papel crucial en la presentación de la clasificación de los equipos en el **ClasificaciónFragment** de la aplicación. Este adaptador se encarga de gestionar la actualización de los puntos y posiciones de los equipos en función de los resultados de los partidos.

Utilizando el **ClasificaciónAdapter**, los usuarios pueden visualizar la tabla de clasificación actualizada en tiempo real, lo que refleja el rendimiento de los equipos a lo largo del torneo. Este adaptador recupera los datos relevantes de Firebase Realtime Database y los presenta de manera ordenada y clara en el **ClasificaciónFragment**, asegurando que los usuarios tengan acceso a información precisa y actualizada sobre la posición de sus equipos favoritos.

Al actualizar los puntos y posiciones de los equipos en función de los resultados de los partidos, el **ClasificaciónAdapter** garantiza que la clasificación refleje con precisión el desempeño de los equipos en el torneo. Esto proporciona a los usuarios una visión clara de la situación actual de la liga y les permite seguir de cerca la evolución de sus equipos favoritos a lo largo de la temporada. La presentación ordenada y actualizada de la clasificación mejora la experiencia de usuario y facilita el seguimiento del torneo en la aplicación.

3.3.2 Capa del cliente: Tipos de datos utilizados para almacenamiento

En la capa del cliente de la aplicación, se utilizan varios tipos de datos para almacenar información relevante en la base de datos Firebase Realtime. Estos tipos de datos están representados por las clases principales del modelo de datos de la aplicación:

3.3.2.1 Clase Equipo

La clase “Equipo” representa un equipo de fútbol dentro de la aplicación. Esta clase almacena información relevante sobre el equipo, incluyendo su nombre, estadio, ciudad y varias estadísticas de rendimiento como goles a favor, goles en contra, partidos jugados, victorias, empates y derrotas. Además, cuenta con métodos para actualizar estas estadísticas de manera dinámica, permitiendo reflejar los resultados de los partidos en tiempo real.

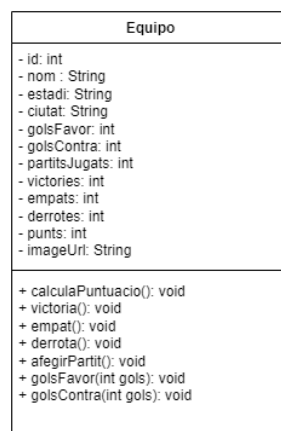


Figura 4: Diagrama de la clase Equipo.

El uso de esta clase en la aplicación es fundamental para la gestión de los equipos. Permite centralizar toda la información relevante y mantener un seguimiento detallado del rendimiento de cada equipo a lo largo de la temporada. La interfaz de usuario accede a los datos encapsulados en esta clase para mostrar detalles y estadísticas de los equipos de manera clara y organizada. La clase “Equipo” facilita la actualización continua de las estadísticas del equipo, asegurando que la información presentada en la aplicación sea siempre precisa y actualizada.

3.3.2.2 Clase Evento

La clase “Evento” centraliza toda la información relevante sobre los eventos que ocurren durante un partido de fútbol. Cada instancia de esta clase contiene detalles específicos sobre un evento particular, como el tipo de evento (gol, tarjeta, etc.), el minuto en el que ocurrió, el jugador involucrado y el equipo correspondiente. Esta estructura permite un registro preciso y detallado de cada acción significativa que tiene lugar durante el partido.

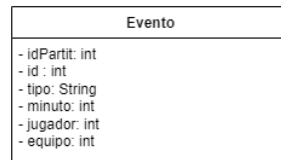


Figura 5: Diagrama de la clase Evento.

En la aplicación, la clase “Evento” es crucial para el registro y la visualización de eventos en tiempo real. Durante un partido, los eventos se crean y almacenan en la base de datos Firebase Realtime, permitiendo un análisis posterior y la generación de estadísticas detalladas. La interfaz de usuario utiliza los datos de esta clase para mostrar un historial de eventos durante el partido, facilitando a los usuarios el seguimiento de todas las incidencias relevantes. Además, estos datos se utilizan para actualizar las estadísticas de jugadores y equipos, reflejando su desempeño de manera precisa y en tiempo real.

3.3.2.3 Clase Jugador

La clase “Jugador” encapsula toda la información relacionada con un jugador de fútbol en la aplicación. Esta clase contiene atributos como el nombre del jugador, apellidos, número de camiseta, goles, tarjetas amarillas y rojas, y el equipo al que pertenece. Además, proporciona métodos para actualizar las estadísticas del jugador durante los partidos, permitiendo incrementar el contador de goles, tarjetas amarillas y tarjetas rojas.

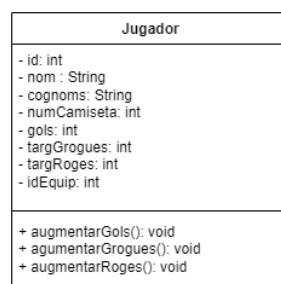


Figura 6: Diagrama de la clase Jugador.

La gestión de jugadores es una parte esencial de la aplicación, y la clase “Jugador” facilita esta tarea al centralizar todos los datos relevantes. La interfaz de usuario accede a los atributos de esta clase para mostrar información detallada sobre los jugadores, incluyendo sus estadísticas de rendimiento. Durante los partidos, los métodos de actualización permiten reflejar en tiempo real el desempeño de los jugadores, asegurando que las estadísticas presentadas sean siempre precisas. Los datos de los jugadores se almacenan en Firebase Realtime Database, permitiendo una gestión eficiente y centralizada de toda la información relacionada con los jugadores.

3.3.2.4 Clase Partido

La clase “Partido” representa un partido de fútbol en la aplicación y centraliza toda la información esencial sobre un partido. Los atributos de esta clase incluyen los equipos que participan, los jugadores de cada equipo, la fecha del partido y el resultado final. Además, esta clase permite registrar eventos específicos que ocurren durante el partido, como goles y tarjetas, actualizando en tiempo real las estadísticas correspondientes.

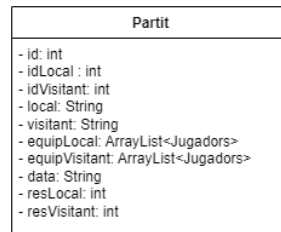


Figura 7: Diagrama de la clase Partit.

En la aplicación, la clase “Partido” es fundamental para la gestión y visualización de los partidos. Al crear un nuevo partido, se instancia un objeto “Partido” con todos los detalles pertinentes, lo que permite una organización clara y estructurada de la información. La interfaz de usuario accede a los datos de esta clase para mostrar información sobre los partidos programados o pasados, incluyendo los resultados y los eventos registrados durante el partido. Los datos del partido se almacenan en Firebase Realtime Database, asegurando una persistencia y accesibilidad eficiente de toda la información relacionada con los partidos.

Estas clases forman la estructura de datos básica utilizada en la capa del cliente de la aplicación para almacenar y gestionar la información relacionada con los equipos, jugadores, partidos y eventos de fútbol.

3.3.3 Capa del servidor: Firebase

Firebase se encarga de proporcionar el backend de la aplicación, incluyendo la gestión de datos, autenticación y almacenamiento de imágenes. Utilizando Firebase Realtime Database, Firebase Authentication y Firebase Storage, se logra una integración completa y eficiente que facilita el desarrollo y mantenimiento de la aplicación.

3.3.3.1 Firebase Realtime Database

Firebase Realtime Database es una base de datos NoSQL alojada en la nube que permite el almacenamiento y la sincronización de datos en tiempo real. Al optar por Firebase en lugar de otras tecnologías como MongoDB, se busca aprovechar su integración nativa con el ecosistema Android, su facilidad de uso y su capacidad para manejar sincronización en tiempo real con una latencia mínima.

Funcionalidad y operaciones

- **Almacenamiento de datos:** Los datos se estructuran en nodos jerárquicos, con colecciones específicas para equipos, jugadores, partidos y eventos.
- **Sincronización en tiempo real:** Firebase garantiza que cualquier cambio en los datos se refleje instantáneamente en todos los dispositivos conectados, lo cual es crucial para una aplicación de gestión de eventos deportivos.
- **Seguridad y autenticación:** Firebase ofrece mecanismos de seguridad y autenticación robustos para proteger los datos y asegurar que solo los usuarios autorizados puedan acceder y modificar la información.

3.3.3.2 Firebase Authentication

Firebase Authentication se utiliza para gestionar la autenticación de los usuarios. Proporciona una forma segura y fácil de autenticar a los usuarios mediante diferentes métodos, como correo electrónico y contraseña, cuentas de Google, Facebook, entre otros. La integración de Firebase Authentication asegura que solo los usuarios registrados y autenticados puedan acceder a las funcionalidades de la aplicación.

3.3.3.3 Firebase Storage

Firebase Storage se utiliza para almacenar las imágenes de los escudos de los equipos. Proporciona una solución escalable y segura para almacenar archivos, permitiendo a la aplicación manejar y presentar imágenes de alta calidad sin preocuparse por la gestión del almacenamiento. Firebase Storage se integra perfectamente con Firebase Realtime Database, permitiendo referenciar las imágenes almacenadas directamente desde la base de datos.

Funcionalidad y operaciones

- **Almacenamiento de imágenes:** Las imágenes de los escudos de los equipos se almacenan en Firebase Storage y se referencian en la base de datos mediante URLs.
- **Acceso seguro:** Firebase Storage asegura que solo los usuarios autorizados puedan acceder y modificar las imágenes, proporcionando un control de acceso granular.
- **Escalabilidad:** La infraestructura de Firebase Storage permite manejar grandes volúmenes de archivos sin comprometer el rendimiento, lo cual es crucial para una aplicación que puede crecer en términos de usuarios y contenido.

4 Resultado

En esta sección, se presenta un recorrido por la aplicación de gestión de equipos y partidos de fútbol, mostrando sus principales funcionalidades y características a través de capturas de pantalla y descripciones detalladas. La aplicación consta de una actividad de Login y cinco fragments principales: Partidos, Jugadores, Equipos, Estadísticas y Clasificación. A continuación, se describe cada elemento y su funcionalidad.

4.1 Activity Login

La aplicación inicia con una pantalla de **Login**, donde los usuarios deben autenticarse para acceder a las funcionalidades. Esta pantalla es fundamental para asegurar que solo usuarios autorizados puedan gestionar la información de los partidos y equipos. La autenticación se realiza mediante Firebase Authentication, ofreciendo un proceso seguro y eficiente.

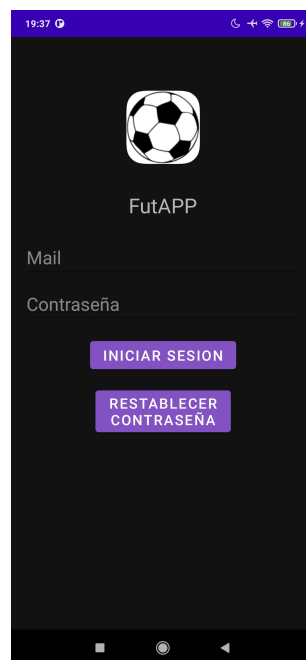


Figura 8: Captura de pantalla de la actividad de Login.

4.2 Fragment Partidos

Este es el fragment que se muestra por defecto una vez se inicia sesión, y es fundamental para la gestión de los partidos en la aplicación. Consta de un *RecyclerView* que presenta una lista de partidos disputados, donde se pueden ver los resultados de cada encuentro. En la parte inferior de la pantalla, hay un botón de acción flotante que permite añadir un nuevo partido que se vaya a disputar.

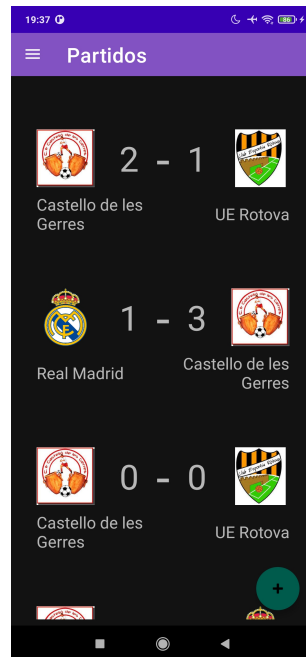


Figura 9: Captura de pantalla del fragment de Partidos.

Al pulsar en uno de los partidos listados, se abrirá un diálogo que muestra los eventos registrados durante ese partido. Estos eventos son indicados con iconos específicos que corresponden al tipo de evento (gol, tarjeta amarilla, tarjeta roja), y se incluye el minuto exacto en que ocurrieron. Esta visualización permite a los usuarios obtener una visión clara y rápida de lo sucedido durante el encuentro.

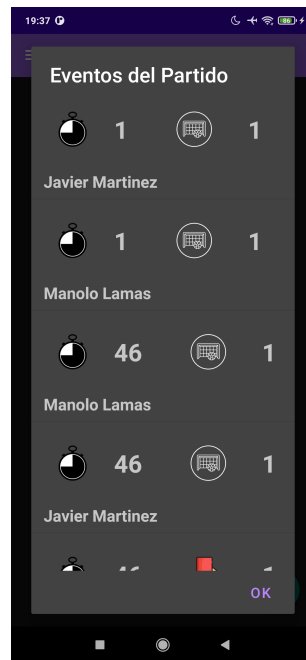


Figura 10: Captura de pantalla de los eventos de un partido.

Si se pulsa el botón para añadir partidos, la aplicación llevará al usuario a una nueva pantalla donde se pueden seleccionar los equipos participantes: tanto el equipo local como el visitante. Además, se puede establecer la fecha y la hora en que se disputará el encuentro, proporcionando una planificación detallada del partido. Una vez seleccionados los equipos y la fecha, la siguiente pantalla permite elegir a los once jugadores titulares que participarán en el encuentro, primero del equipo local y luego del equipo visitante.



Figura 11: Captura de pantalla de la selección de jugadores que van a disputar el encuentro.

Después de seleccionar a los jugadores de ambos equipos, se accede a una pantalla diseñada para registrar los eventos que ocurran durante el partido en tiempo real. En esta pantalla, los jugadores están listados en un *RecyclerView*, y el usuario puede pulsar en el nombre del jugador para registrar un evento específico, como un gol, una tarjeta amarilla o una tarjeta roja. Esta funcionalidad es crucial para mantener un registro preciso y detallado de los eventos del partido. Además, hay un cronómetro que muestra el minuto de juego, el cual se puede pausar si es necesario. El botón de descanso detiene el cronómetro y lo ajusta al minuto 45, indicando el inicio del segundo tiempo del partido.

Una vez finalizado el partido, el usuario puede pulsar el icono de aspa en la esquina superior derecha para guardar todos los datos del partido en la base de datos. Esta acción asegura que toda la información registrada durante el partido se almacene de manera persistente en Firebase Realtime Database, facilitando el análisis posterior y la actualización de las estadísticas de jugadores y equipos.

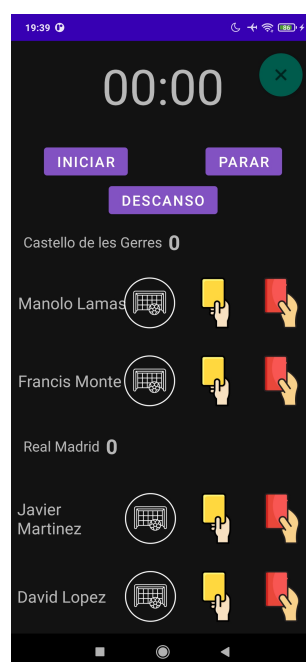
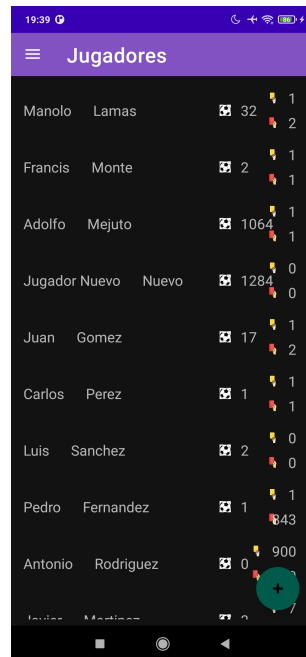


Figura 12: Captura de pantalla de la selección de jugadores que van a disputar el encuentro.

En resumen, el fragment de **Partidos** proporciona una interfaz completa y funcional para gestionar todos los aspectos relacionados con los partidos de fútbol, desde su creación y planificación hasta el registro detallado de los eventos en tiempo real. Esto permite a los usuarios tener un control total sobre los encuentros y mantener una base de datos precisa y actualizada.

4.3 Fragment Jugadores

En esta vista podremos observar la lista completa de jugadores que participan en la liga, cada uno con sus estadísticas individuales. La interfaz presenta los nombres de los jugadores junto con información relevante, como la cantidad de goles, tarjetas amarillas y tarjetas rojas que han recibido. Esta vista es esencial para llevar un seguimiento detallado del desempeño de cada jugador a lo largo de la temporada.



Jugadores	
Manolo Lamas	32 1 2
Francis Monte	2 1 1
Adolfo Mejuto	1064 1 1
Jugador Nuevo Nuevo	1284 0 0
Juan Gomez	17 1 2
Carlos Perez	1 1 1
Luis Sanchez	2 0 0
Pedro Fernandez	1 1 43
Antonio Rodriguez	0 900 7

Figura 13: Captura de pantalla del fragment de Jugadores.

En la parte inferior derecha de la pantalla, hay un botón de acción flotante que permite añadir un nuevo jugador. Al pulsar este botón, se abre una nueva pantalla de formulario donde se deben completar varios campos para registrar al jugador. Estos campos incluyen información básica como el nombre, apellidos y número de camiseta del jugador. Además, hay una lista desplegable que permite seleccionar el equipo al que pertenece el jugador, ya que para añadir un jugador, este debe estar inscrito en un equipo existente en la base de datos.

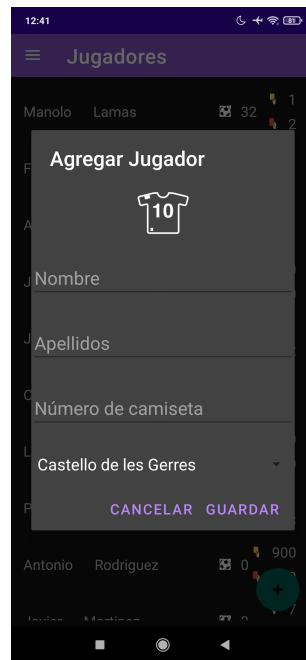


Figura 14: Captura de pantalla para añadir a un jugador.

Una vez se ha completado el formulario y se añade el jugador, la aplicación redirige de nuevo a la lista principal de jugadores. Aquí, el nuevo jugador aparecerá junto con sus estadísticas iniciales. Si se hace una pulsación larga en cualquiera de los jugadores listados, se abrirá una vista más detallada que muestra todas las estadísticas del jugador seleccionado. Esta vista detallada incluye información más específica, como el total de minutos jugados, el rendimiento en los últimos partidos y cualquier otra métrica relevante que se haya registrado.

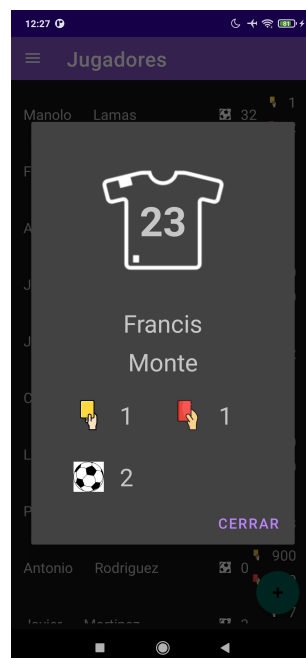


Figura 15: Captura de pantalla del dialogo de Jugadores.

Además, desde la vista detallada de un jugador, se pueden realizar actualizaciones en sus estadísticas en tiempo real. Por ejemplo, si durante un partido el jugador marca un gol o recibe una tarjeta, estos eventos se pueden registrar y actualizar instantáneamente en su perfil. Esta funcionalidad es crucial para mantener la precisión y actualidad de los datos en la aplicación.

En resumen, el fragment de **Jugadores** proporciona una herramienta completa y accesible para gestionar la información de los jugadores. Permite no solo la visualización de sus estadísticas y logros, sino también la adición de nuevos jugadores y la actualización constante de sus datos, asegurando que los usuarios tengan acceso a información precisa y detallada en todo momento.

4.4 Fragment Equipos

En el fragment de **Equipos**, se presenta una lista de todos los equipos que están almacenados en la base de datos. Cada equipo se muestra con su nombre y, si está disponible, su escudo, proporcionando una representación visual clara y organizada de los equipos participantes en la liga. Esta vista es fundamental para gestionar y visualizar rápidamente la información básica de cada equipo registrado.

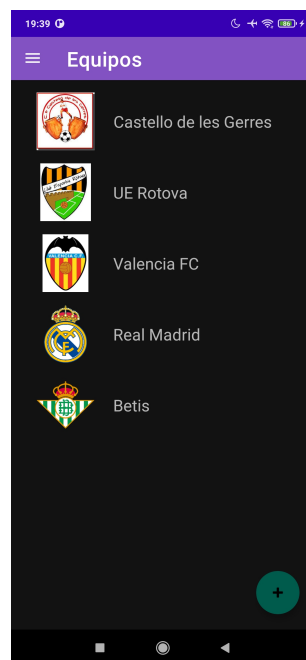


Figura 16: Captura de pantalla del fragment de Equipos.

En la parte inferior derecha de la pantalla, hay un botón de acción flotante que permite añadir un nuevo equipo. Al pulsar este botón, se abre una nueva pantalla de formulario donde se deben completar varios campos para registrar un equipo. Estos campos incluyen el nombre del equipo, el estadio donde juegan sus partidos de local y la ciudad en la que se encuentra. Además, existe la opción de subir una imagen para el escudo del equipo, que se almacenará en Firebase Storage. Este escudo se mostrará posteriormente junto al nombre del equipo en la lista principal.

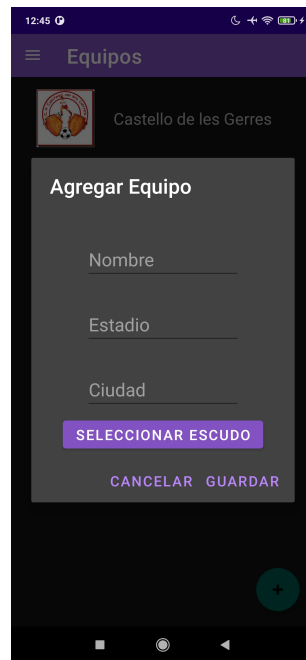


Figura 17: Captura de pantalla de añadir un equipo.

Una vez se ha completado el formulario y se añade el equipo, la aplicación redirige de nuevo a la lista principal de equipos. Aquí, el nuevo equipo aparecerá con su nombre y escudo, si se ha proporcionado uno. Esta funcionalidad de adición es esencial para mantener la base de datos actualizada y permitir la expansión de la liga con nuevos equipos.

Al hacer una pulsación en cualquiera de los equipos listados, se abre una vista detallada que muestra todos los jugadores inscritos en ese equipo. Esta vista detallada incluye no solo los nombres de los jugadores, sino también sus estadísticas generales, como el número de goles, tarjetas amarillas y rojas. Esta funcionalidad es crucial para obtener una visión completa del rendimiento del equipo y sus jugadores.

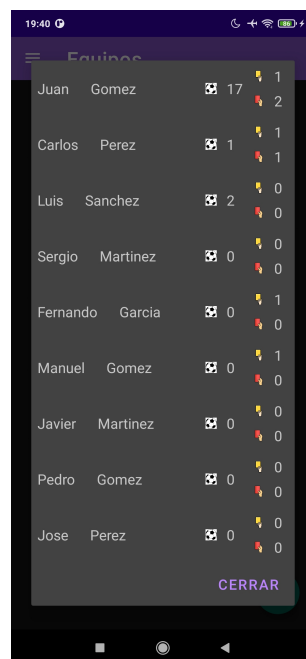


Figura 18: Captura de pantalla de los jugadores del equipo.

En resumen, el fragment de **Equipos** proporciona una herramienta completa y accesible para gestionar la información de los equipos. Permite no solo la visualización de los equipos y sus escudos, sino también la adición de nuevos equipos. Esta funcionalidad asegura que los usuarios tengan acceso a datos precisos y detallados sobre cada equipo y sus jugadores en todo momento.

4.5 Fragment Estadísticas

El fragment de **Estadísticas** es una herramienta clave para analizar el rendimiento de los jugadores en la liga. En esta vista, los usuarios pueden observar y comparar las estadísticas principales de los jugadores, ofreciendo una perspectiva clara y detallada de sus logros y comportamientos en el campo.

Al acceder a este fragment, se presenta una interfaz intuitiva que permite seleccionar el tipo de estadística que se desea visualizar. Las opciones disponibles incluyen goles, tarjetas amarillas y tarjetas rojas. Esta selección se realiza mediante un spinner que facilita la navegación entre los diferentes tipos de estadísticas. Una vez seleccionada la categoría de interés, se muestra una lista descendente de jugadores ordenada según el criterio elegido.

La lista muestra a los jugadores junto con el valor de la estadística seleccionada, por ejemplo, el número de goles anotados, la cantidad de tarjetas amarillas recibidas o las tarjetas rojas acumuladas. Esta funcionalidad es crucial para los entrenadores, jugadores y aficionados que desean tener una visión completa del rendimiento individual de los jugadores, permitiendo identificar a los jugadores más destacados en cada categoría.

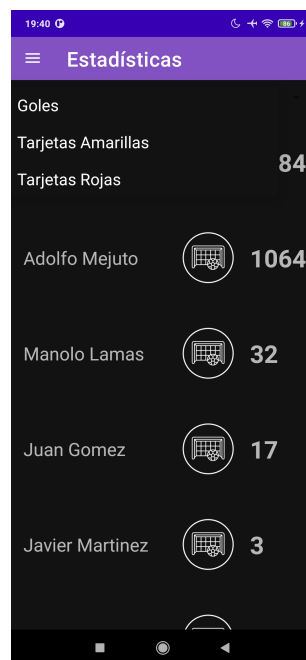


Figura 19: Captura de pantalla de las estadísticas.

Además, la interfaz está diseñada para ser responsiva y fácil de usar, asegurando que los usuarios puedan acceder rápidamente a la información que necesitan sin complicaciones. La presentación visual de las estadísticas es clara y concisa, con un formato que facilita la comparación entre jugadores.

La capacidad de ordenar y filtrar por diferentes tipos de estadísticas proporciona una herramienta poderosa para el análisis detallado del rendimiento de los jugadores. Por ejemplo, los entrenadores pueden utilizar esta información para tomar decisiones estratégicas sobre alineaciones y tácticas, mientras que los aficionados pueden seguir de cerca a sus jugadores favoritos y sus progresos a lo largo de la temporada.

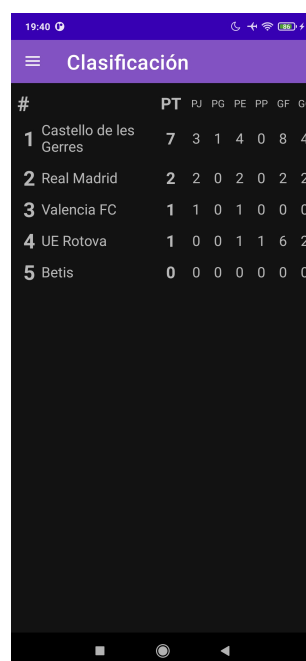
En resumen, el fragment de **Estadísticas** ofrece una visión completa y organizada del rendimiento de los jugadores, permitiendo a los usuarios seleccionar y visualizar las estadísticas que más les interesan. Esta funcionalidad mejora la capacidad de análisis y seguimiento del rendimiento individual, aportando un valor significativo tanto a nivel estratégico como informativo dentro de la aplicación.

4.6 Fragment Clasificación

El fragmento de **Clasificación** proporciona una visión clara y ordenada del rendimiento de los equipos en la liga. En esta vista, los usuarios pueden observar la posición de cada equipo en la clasificación, junto con su puntuación actual basada en el sistema de puntuación establecido por la Federación Internacional de Fútbol Asociación: tres puntos por victoria, un punto por empate y cero puntos por derrota, Javier Mercada (2023).

La clasificación se presenta en forma de una lista ordenada de equipos, con cada equipo mostrando su posición en la tabla, su nombre y el número de puntos acumulados hasta el momento. Esta presentación permite a los usuarios identificar fácilmente los equipos líderes de la liga, así como aquellos que están luchando en la parte inferior de la clasificación.

Además de la posición y los puntos, el fragmento de Clasificación también muestra otros detalles importantes, como el número de partidos jugados, las victorias, los empates y las derrotas de cada equipo. Esta información adicional proporciona una visión más completa del rendimiento de los equipos, permitiendo a los usuarios evaluar su consistencia y éxito en la liga.



#		PT	PJ	PG	PE	PP	GF	GC
1	Castillo de les Gerres	7	3	1	4	0	8	4
2	Real Madrid	2	2	0	2	0	2	2
3	Valencia FC	1	1	0	1	0	0	0
4	UE Rotova	1	0	0	1	1	6	2
5	Betis	0	0	0	0	0	0	0

Figura 20: Captura del fragment de Clasificación.

La interfaz del fragmento está diseñada para ser intuitiva y fácil de entender, con una presentación clara y concisa de la información. Los usuarios pueden navegar rápidamente por la clasificación y obtener una visión general del estado actual de la competición, lo que facilita el seguimiento y la comprensión de la posición de sus equipos favoritos.

En resumen, el fragmento de Clasificación ofrece una herramienta poderosa para evaluar el rendimiento de los equipos en la liga, proporcionando una visión completa y organizada de la clasificación actual. Esta funcionalidad es fundamental para los aficionados, entrenadores y jugadores que desean seguir de cerca la competición y entender el posicionamiento de cada equipo en la tabla.

5 Conclusión

La aplicación desarrollada para la gestión de equipos y partidos de fútbol ha demostrado ser una herramienta valiosa y funcional para organizar y seguir una liga de manera eficiente. Se han implementado varias funcionalidades clave que permiten un control detallado de los partidos, el registro de eventos importantes y el mantenimiento de estadísticas precisas tanto para jugadores como para equipos. Las principales características incluyen la gestión de partidos, donde se pueden añadir y consultar partidos, seleccionar jugadores y registrar eventos como goles y tarjetas. Además, se ha implementado la gestión de jugadores, permitiendo añadir nuevos jugadores y consultar sus estadísticas individuales. La gestión de equipos facilita la visualización de equipos y sus jugadores, mientras que la sección de estadísticas y clasificación de equipos ofrece una visión clara del rendimiento general en la liga. La aplicación ha sido desarrollada utilizando Android Studio y Firebase Realtime Database, proporcionando una base sólida y escalable para el almacenamiento y gestión de datos.

A pesar de las funcionalidades implementadas, la aplicación presenta algunas limitaciones y áreas de mejora. Sería beneficioso incorporar la capacidad de registrar substituciones de jugadores durante los partidos, lo que permitiría un seguimiento más detallado del rendimiento de cada jugador. Además, el registro de otros eventos importantes como faltas, asistencias y lesiones podría ofrecer una visión más completa de cada partido. La mejora de la interfaz de usuario también es una prioridad, ya que una UI/UX más intuitiva y atractiva haría la aplicación más accesible y agradable de usar. La implementación de un sistema de notificaciones y alertas para mantener informados a los usuarios sobre los próximos partidos y cambios en la clasificación mejoraría la interacción y el compromiso con la aplicación. También sería útil añadir funcionalidades de administración que permitan a los administradores gestionar equipos, jugadores y partidos de manera más eficiente, incluyendo la edición y eliminación de datos incorrectos o desactualizados.

Mirando hacia el futuro, es importante planificar y ejecutar mejoras y nuevas funcionalidades para asegurar que la aplicación siga siendo útil y relevante. La ampliación de funcionalidades para incluir substituciones, registro de faltas y otras estadísticas proporcionaría una herramienta más completa y detallada. Evaluar y mejorar la escalabilidad de la aplicación garantizará un rendimiento óptimo a medida que crece en volumen de datos y usuarios. Además, la implementación de herramientas de análisis y generación de reportes ayudaría a los usuarios a obtener insights detallados sobre el rendimiento de equipos y jugadores, mejorando la toma de decisiones y estrategias. La integración con redes sociales permitiría a los usuarios compartir estadísticas y resultados, aumentando el alcance y la interacción con la aplicación. Establecer un sistema de feedback para recoger las opiniones de los usuarios y realizar mejoras continuas basadas en sus necesidades y sugerencias asegurará que la aplicación evolucione y se adapte a las futuras necesidades y tecnologías emergentes.

6 Anexos

6.1 Estructura base de datos

```
{
  "equips": [
    null,
    {
      "ciutat": "Nombre Ciudad",
      "derrotes": 0,
      "empats": 2,
      "estadi": "Nombre estadio",
      "golsContra": 4,
      "golsFavor": 8,
      "id": 1,
      "nom": "Nombre equipo",
      "partitsJugats": 1,
      "punts": 5,
      "victories": 1
    }
  ],
  "events": [
    null,
    {
      "equipo": 1,
      "id": 1,
      "idPartit": 0,
      "jugador": 1,
      "minuto": 1,
      "tipo": "gol"
    }
  ],
  "jugadors": [
    null,
    {
      "assistencies": 0,
      "cognoms": "Apellidos",
      "gols": 32,
      "id": 1,
      "idEquip": 1,
      "minuts": 0,
      "nom": "Nombre",
      "numCamiseta": 23,
      "targGrogues": 1,
      "targRoges": 2
    }
  ],
  "partits": [
    {
      "data": "23/05/2024",
      "equipLocal": [
        {
          "assistencies": 0,
          "cognoms": "Apellidos",
          "gols": 17,
          "id": 1,
          "idEquip": 1,
          "minuts": 0,
          "nom": "Nombre",
          "numCamiseta": 23,
          "targGrogues": 1,
          "targRoges": 2
        }
      ]
    }
  ]
}
```

```
    }  
  ],  
  "equipVisitant": [  
    {  
      "assistencies": 0,  
      "cognoms": "Apellidos",  
      "gols": 13,  
      "id": 5,  
      "idEquip": 2,  
      "minuts": 0,  
      "nom": "Nombre",  
      "numCamiseta": 7,  
      "targGrogues": 1,  
      "targRoges": 1  
    }  
  ],  
  "id": 0,  
  "idLocal": 1,  
  "idVisitant": 2,  
  "local": "Nombre equipo Local",  
  "resLocal": 2,  
  "resVisitant": 1,  
  "visitant": "Nombre equipo visitante"  
}  
]  
}
```

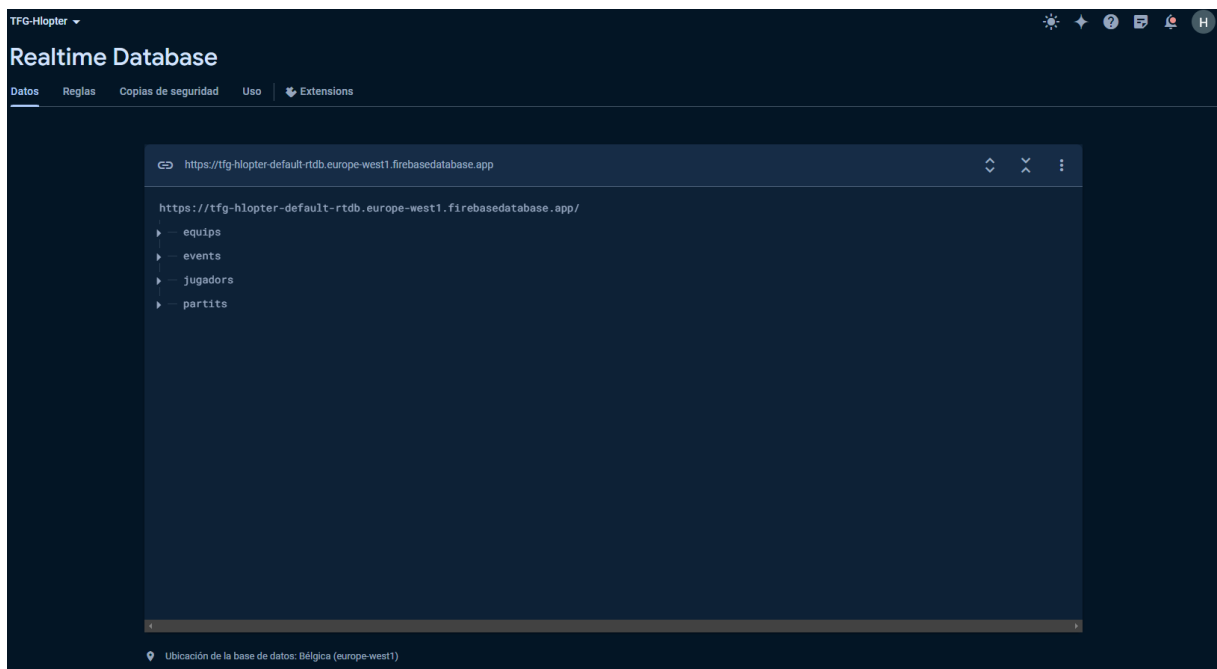


Figura 21: Captura de la base de datos almacenada en Google Firebase.

7 Bibliografía

- Formate. (2021). *Razones para utilizar Java y su popularidad*. Consultado el 1 de junio de 2024, desde <https://www.formate.es/blog/desarrollo-profesional/razones-para-utilizar-java/> (vid. pág. 8).
- Google. (2024-a). *Documentación de Firebase*. Consultado el 1 de junio de 2024, desde <https://firebase.google.com/docs?hl=es-419> (vid. pág. 8).
- Google. (2024-b). *Google developers*. Consultado el 1 de junio de 2024, desde <https://developer.android.com/develop?hl=es-419> (vid. pág. 7).
- Javier Mercada. (2023). *¿CÓMO HAN CAMBIADO LAS REGLAS DEL FÚTBOL A LO LARGO DE LA HISTORIA?* Consultado el 4 de abril de 2023, desde <https://www.relevo.com/futbol/cambiado-reglas-futbol-20230315143611-nt.html> (vid. pág. 30).
- Jose Luis Bustos. (2024). *Por qué usar Firebase*. Consultado el 1 de junio de 2024, desde <https://keepcoding.io/blog/por-que-usar-firebase/#:~:text=Por%20tanto%2C%20la%20principal%20raz%C3%B3n,retrasar%20el%20proceso%20de%20programaci%C3%B3n.> (vid. pág. 8).
- StackOverflow. (2024). *StackOverflow*. Consultado el 1 de junio de 2024, desde <https://stackoverflow.com/> (vid. pág. 9).