

Memoria del proyecto

MartialMind: Aplicación para guiar el aprendizaje en distintas artes marciales y seguir los resultados de las competiciones.

MartialMind: Application to guide learning in different martial arts and follow the results of the competitions.

Trabajo de Fin de Grado
INGENIERÍA INFORMÁTICA



**VNiVERSiDAD
D SALAMANCA**

Julio de 2023

Autor

Francisco José Cruz Pérez

Tutores

André Felipe Sales Mendes

Héctor Sánchez San Blas

Resumen

En la actualidad las artes marciales mixtas son el deporte con el mayor crecimiento a nivel mundial, lo que ha hecho que cada vez más personas se interesen por el aprendizaje de las distintas artes marciales que existen.

Las artes marciales poseen un contexto milenario y un trasfondo muy rico, ya que estas disciplinas no están basadas únicamente en apartado físico del ser humano, sino además en el respeto hacia el rival, la convicción y disciplina personal, y sobre todo en el crecimiento personal.

Con todo esto, el desarrollo de aplicaciones móviles que orientan el aprendizaje de artes marciales se encuentra en auge, permitiendo la visualización de vídeos instructivos acerca de los movimientos que componen el arte marcial, mostrar un progreso en estos ejercicios mediante la inclusión de niveles de habilidad, la recomendación de rutinas de ejercicio físico a realizar o el seguimiento de los resultados de los eventos.

Como objetivo principal de este proyecto se tiene el desarrollo de una aplicación móvil multiplataforma mediante el kit de desarrollo de Flutter. La aplicación permitirá realizar todas las tareas, que realizan otras aplicaciones, mencionadas anteriormente reunidas en un solo lugar.

Cabe destacar que otro de los objetivos de la aplicación es conseguir la optimización de una aplicación que permita seguir un aprendizaje, desarrollar una presentación dinámica y simple estéticamente de los datos, además de una navegación fluida dentro de la aplicación, para permitir al usuario un uso diario liviano con el que perseguir su meta de aprender las distintas artes marciales.

Palabras clave: artes marciales, multiplataforma, Flutter, resultados.

Summary

Currently, mixed martial arts is the sport with the highest growth worldwide, which has made more and more people interested in learning the different martial arts that exist.

Martial arts have an ancient context and a very rich background, since these disciplines are not based solely on the physical aspect of the human being, but also on respect for the rival, conviction and personal discipline, and above all on personal growth.

With all this, the development of mobile applications that guide the learning of martial arts is booming, allowing the viewing of instructional videos about the movements that make up the martial art, showing progress in these exercises by including skill levels, the recommendation of physical exercise routines to be carried out or the monitoring of the results of the events.

The main objective of this project is the development of a multiplatform mobile application using the Flutter development kit. The application will allow you to perform all the tasks, which are performed by other applications, mentioned above gathered in one place.

It should be noted that another of the objectives of the application is to achieve the optimization of an application that allows learning to be followed, to develop a dynamic and aesthetically simple presentation of the data, as well as fluid navigation within the application, to allow the user to use light journal with which to pursue your goal of learning the different martial arts.

Keywords: martial arts, multiplatform, Flutter, results.

Índice de contenido

| | | |
|-------|---|----|
| 1. | Introducción | 1 |
| 2. | Objetivos del proyecto..... | 3 |
| 2.1 | Objetivos del sistema | 3 |
| 2.2 | Objetivos personales | 4 |
| 3. | Conceptos teóricos | 5 |
| 3.1 | Estructura jerárquica | 5 |
| 3.2 | Bcrypt..... | 6 |
| 3.3 | API-REST | 7 |
| 4. | Técnicas y herramientas | 8 |
| 4.1 | Herramientas de diseño | 8 |
| 4.1.1 | Adalo..... | 8 |
| 4.1.3 | WhatFont..... | 10 |
| 4.1.4 | Google Fonts | 10 |
| 4.1.5 | Draw.io..... | 11 |
| 4.2 | Herramientas de codificación..... | 12 |
| 4.2.1 | Dart | 12 |
| 4.2.2 | DartPad | 13 |
| 4.2.3 | Librerías Flutter..... | 14 |
| 4.3 | Herramientas de desarrollo..... | 15 |
| 4.3.1 | Visual Studio Code..... | 15 |
| 4.3.2 | Android Studio | 15 |
| 4.4 | Herramientas de gestión del proyecto..... | 16 |
| 4.4.1 | Microsoft Office Word | 16 |
| 4.4.2 | Google Drive..... | 16 |
| 4.5 | Herramientas adicionales | 17 |

| | | |
|-------|------------------------------------|----|
| 4.5.1 | Firestore | 17 |
| 4.5.2 | SportsData.io | 18 |
| 5. | Aspectos relevantes del desarrollo | 19 |
| 5.1 | Marco de trabajo | 19 |
| 5.2 | Planificación temporal | 20 |
| 5.3 | Especificación de requisitos | 23 |
| 5.3.1 | Objetivos del sistema | 23 |
| 5.3.2 | Requisitos de información | 25 |
| 5.3.3 | Requisitos funcionales | 26 |
| 5.3.4 | Requisitos no funcionales | 28 |
| 5.4 | Análisis de requisitos | 29 |
| 5.4.1 | Diagrama de clases | 29 |
| 5.4.2 | Diagramas de secuencia | 30 |
| 5.4.3 | Vista de arquitectura | 31 |
| 5.5 | Implementación | 32 |
| 6. | Conclusiones y líneas de trabajo | 35 |
| 6.1 | Conclusiones | 35 |
| 6.2 | Líneas de desarrollo futuras | 36 |
| 7. | Bibliografía | 37 |

Índice de ilustraciones

| | |
|---|----|
| Ilustración 1: Estructura jerárquica | 5 |
| Ilustración 2: Bcrypt hashing | 6 |
| Ilustración 3: API-REST | 7 |
| Ilustración 4 Adalo | 9 |
| Ilustración 5: WhatFont..... | 10 |
| Ilustración 6: Google Fonts | 10 |
| Ilustración 7: Draw.io | 11 |
| Ilustración 8: Dart | 12 |
| Ilustración 9: DartPad | 13 |
| Ilustración 10: Android Studio | 15 |
| Ilustración 11: Cloud Firestore | 17 |
| Ilustración 12: Firebase Storage | 18 |
| Ilustración 13: SportsData.io | 18 |
| Ilustración 14: Fases del proceso unificado | 19 |
| Ilustración 15: Diagrama de paquetes Gestión de usuarios..... | 28 |
| Ilustración 16: Modelo de dominio del sistema..... | 29 |
| Ilustración 17: Diagrama de secuencia de Inicio de sesión..... | 30 |
| Ilustración 18: JSON recuperado de SportsData.io | 32 |
| Ilustración 19: Estructura de los datos de usuario Firebase | 33 |

Índice de tablas

| | |
|--|----|
| Tabla 1: Objetivo del sistema gestión de resultados..... | 24 |
| Tabla 2: Requisito del sistema Arte Marcial | 25 |
| Tabla 3 Requisito funcional del sistema Usuario loggeado | 26 |
| Tabla 4 Requisito funcional caso de uso Inicio de sesión..... | 27 |
| Tabla 5: Definición requisito no funcional Disponibilidad..... | 28 |

1. Introducción

El bienestar personal y la salud mental están directamente relacionados con la seguridad en uno mismo y la capacidad de gestionar las emociones, gracias a las artes marciales se puede obtener una forma efectiva de mejorar en ambos aspectos. En la actualidad el crecimiento exponencial que han alcanzado los deportes de contacto en los que se ponen en práctica estas disciplinas ha hecho que cada vez más personas se sumerjan en ellas, lo que provoca la necesidad de herramientas que faciliten su aprendizaje y seguimiento.

En este sentido, este proyecto propone el diseño e implementación de una aplicación móvil que brinde al usuario una plataforma interactiva y accesible para aprender artes marciales, a la vez que seguir de cerca los resultados y eventos relacionados, además de la recomendación de rutinas de ejercicios de cardio y fuerza. Todas estas funcionalidades se pueden encontrar en otras aplicaciones de este sector, debido a esto el objetivo primordial es el acoplar todas esas acciones en una misma aplicación.

En este documento se recogen los que son los aspectos más relevantes del proceso de desarrollo del proyecto siguiendo la guía de la plataforma Diaweb:

- **Objetivos del proyecto:** Especificación detallada de los objetivos que se busca satisfacer con la realización del proyecto.
- **Conceptos teóricos:** Explicación concisa de algunos conceptos destacados.
- **Técnicas y herramientas:** Presentación de las técnicas y herramientas utilizadas en el proyecto.
- **Aspectos relevantes del desarrollo del proyecto:** Se destacan los aspectos más relevantes del proyecto.
- **Conclusiones y líneas de trabajo futuro:** Se recogen las conclusiones obtenidas durante el desarrollo del proyecto y se proponen las maneras de continuarlo.

Además, con motivo de completar la memoria, se adjuntas los anexos técnicos de cada uno de los apartados:

- **Anexo I. Planificación temporal:** se lleva a cabo el estudio de los aspectos relevantes a la hora de llevar a cabo la planificación temporal de la elaboración y el desarrollo del proyecto software.
- **Anexo II. Especificación de requisitos del software:** se tiene como objetivo plasmar la especificación de requisitos software del proyecto a desarrollar.
- **Anexo III. Especificación de diseño:** se recoge toda la fase de análisis de requisitos del proyecto.
- **Anexo IV. Documentación técnica:** presenta al programador el código de la aplicación de una manera legible y sencilla.
- **Anexo V. Manual de usuario:** se muestra un tutorial para conseguir una utilización correcta y efectiva de la aplicación.

2. Objetivos del proyecto

En este apartado de la memoria se documentan los objetivos que se han seguido para la el desarrollo del proyecto:

2.1 Objetivos del sistema

Primero se describen los objetivos que el sistema debe cumplir, estos son los correspondientes al satisfacer un correcto funcionamiento del sistema mediante la satisfacción de sus requisitos.

- **Gestión de usuarios:** El sistema deberá gestionar el posible registro en el sistema por parte del usuario, así como el inicio de sesión en el mismo si dispone de cuenta. Además, deberá permitir al usuario registrado la modificación de los datos de su perfil.
- **Gestión de artes marciales:** El sistema deberá gestionar las distintas artes marciales que se presentarán a los usuarios loggeados, además de gestionar el nivel de experiencia que el usuario tiene en cada una en función de los ejercicios completados.
- **Gestión de competiciones:** El sistema deberá gestionar las distintas competiciones que se realizarán en las fechas indicadas y su estado de finalización.
- **Gestión de resultados:** El sistema deberá gestionar los resultados de las distintas competiciones, así como los peleadores de cada pelea, sus fotos y el ganador de la misma.
- **Gestión de ejercicios:** El sistema deberá gestionar los distintos ejercicios que componen cada arte marcial, tanto los vídeos como las descripciones, como si han sido completados o no.
- **Gestión de rutinas:** El sistema deberá gestionar las distintas rutinas de ejercicio físico asignadas para cada arte marcial, además de presentarlas en función de si el usuario ha seleccionado ese arte marcial o no.

2.2 Objetivos personales

Entre los objetivos personales destaca la programación y realización de las distintas tareas que componen el diseño, desarrollo e implementación de un proyecto software planteado por el propio desarrollador, tarea que contribuye al crecimiento personal y busca poner a prueba aquellos conocimientos obtenidos durante la realización del grado, además de la adquisición de nuevos conocimientos.

Se busca un crecimiento profesional mediante el uso de herramientas y técnicas no utilizadas previamente, contribuyendo a la capacidad de asumir responsabilidades y de poder abarcar proyectos futuros de una mayor complejidad que necesiten de estas herramientas.

El proyecto atiende a la necesidad de la sociedad actual de adoptar un estilo de vida disciplinado y activo gracias al aprendizaje de artes marciales, contribuyendo a la adición de calma a un mundo que cada vez es más frenético y estresante.

Para finalizar, se busca la capacidad de poder realizar una exposición oral del proyecto con la plena seguridad en el trabajo realizado.

3. Conceptos teóricos

Para comprender el desarrollo del proyecto en su totalidad es necesario incidir en determinados conceptos teóricos sobre las herramientas y métodos utilizados en la elaboración del trabajo.

3.1 Estructura jerárquica

Es un principio de organización usado en el diseño de sistemas que implica la disposición de los distintos elementos en niveles o capas. Los principios de esta estructura se basan en la idea de que cada nivel contiene subelementos relacionados más específicos.

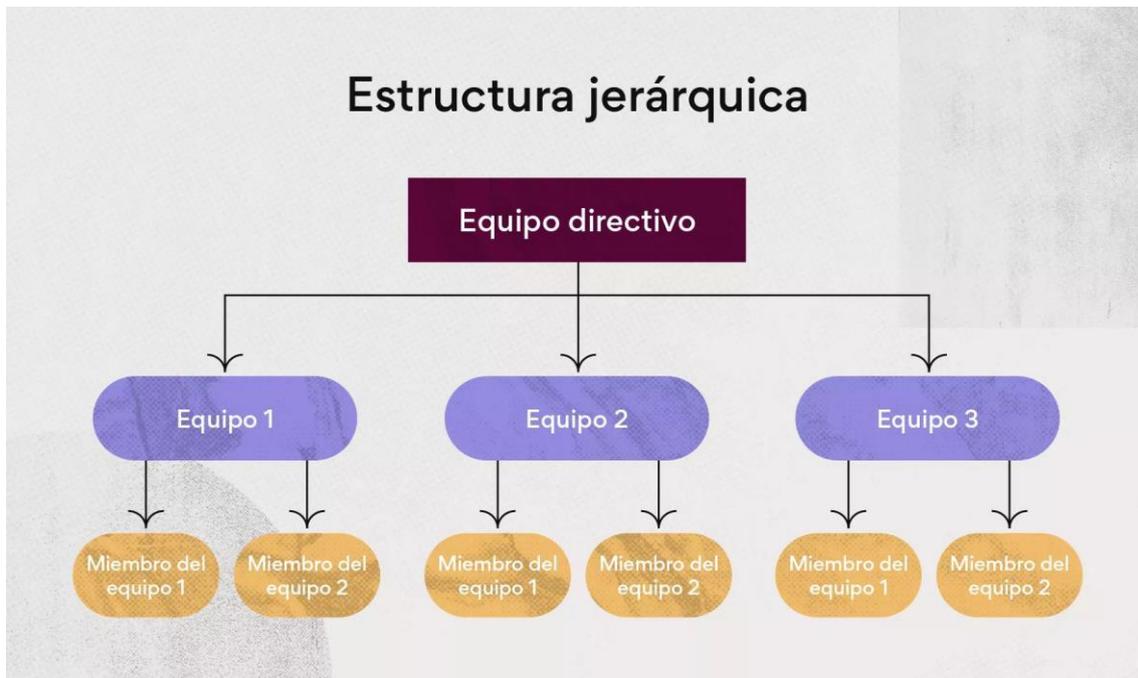


Ilustración 1: Estructura jerárquica

En la *Ilustración 1* se muestra un modelo de estructura jerárquica aplicada a un equipo directivo formado por varios equipos, que a su vez se encuentran compuestos por los miembros de cada equipo.

En el proyecto, esta estructura jerárquica ha sido aplicada a la organización, tanto local como en la base de datos de Firebase, de las artes marciales. En las que cada arte marcial está compuesta por una serie de ejercicios, que a su vez disponen de un vídeo y una descripción. Además, para cada evento recuperado, se recupera la lista de peleas que lo forman, que a su vez está compuesta por los peleadores que forman cada pelea.

3.2 Bcrypt

Es un algoritmo de hash seguro diseñado para el almacenamiento de contraseñas de forma segura en Firebase. Está diseñado deliberadamente para ser lento y resistente frente a los ataques de fuerza bruta y de recuperación de contraseñas originales a partir de los hashes almacenados. Esta característica de lentitud es fundamental para dificultar la tarea de adivinar el hash almacenado.

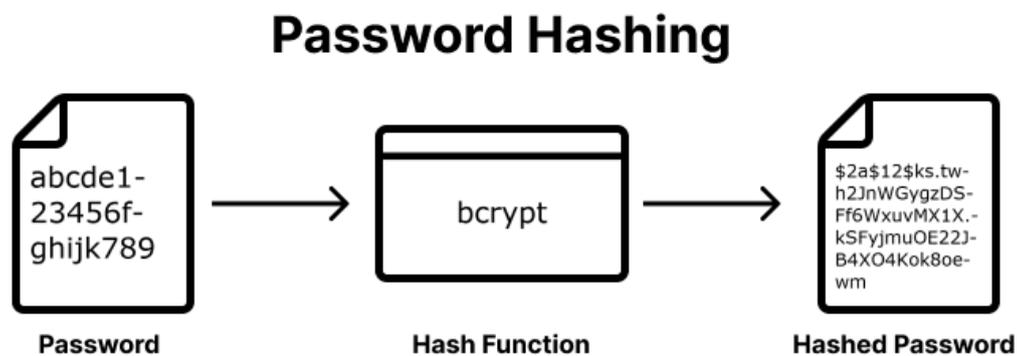


Ilustración 2: Bcrypt hashing

Al utilizar Bcrypt en el proyecto, se asegura que incluso si el atacante obtiene el acceso a los hashes de la base de datos, resultaría sumamente difícil y lento el lograr descifrar las contraseñas originales gracias a este factor de lentitud deliberada.

3.3 API-REST

O Transferencia de Estado Representacional consiste en un estilo arquitectónico utilizado para la creación de servicios web mediante las peticiones HTTP GET, POST, PUT y DELETE, para realizar operaciones en los recursos, representados en formato XML o JSON, además de una correcta separación entre cliente y servidor.

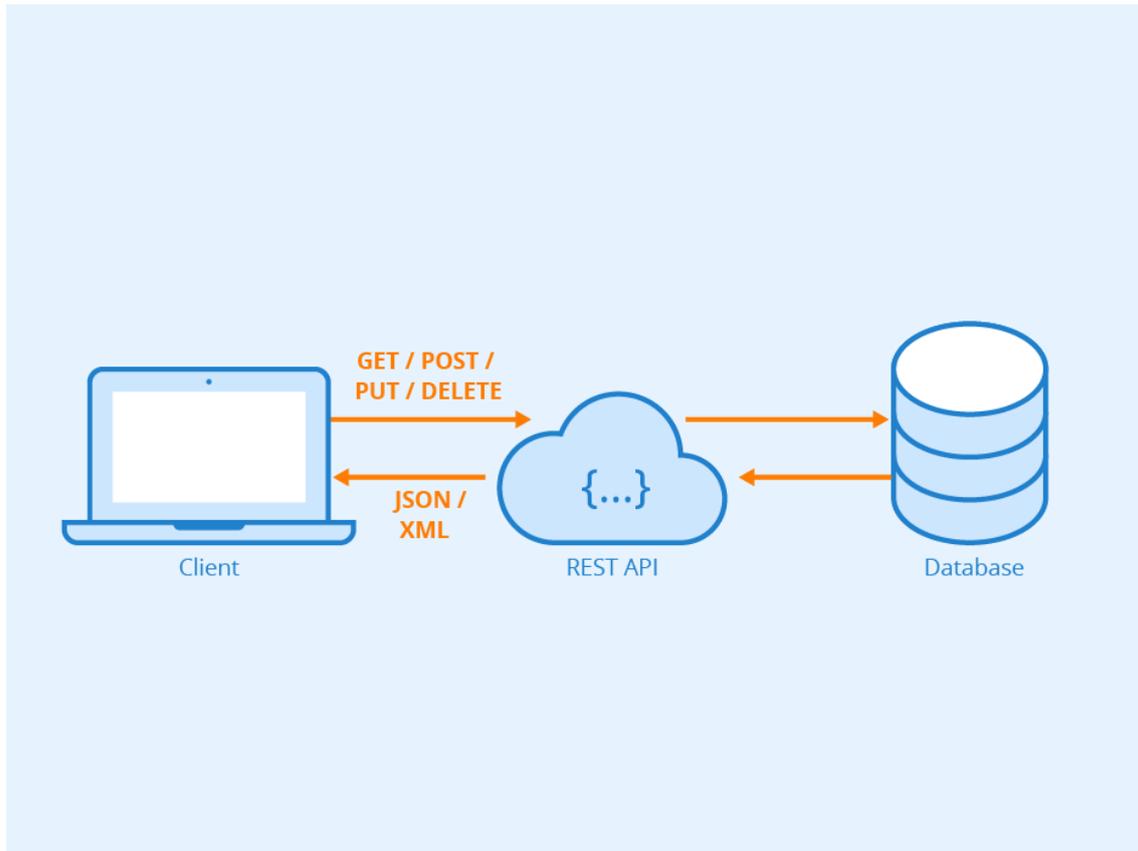


Ilustración 3: API-REST

En el proyecto se ha hecho uso únicamente de la petición GET para obtener los distintos resultados de los eventos a partir de los endpoints correspondientes.

4. Técnicas y herramientas

En este apartado se presentan y detallan las técnicas y herramientas utilizadas durante la elaboración del proyecto.

4.1 Herramientas de diseño

Se presentarán a continuación las distintas herramientas involucradas en la fase de diseño de la elaboración del proyecto software.

4.1.1 Adalo

Es una plataforma de desarrollo de aplicaciones sin usar código que permite crear diferentes aplicaciones móviles y web sin necesidad de conocimientos de programación. El uso de esta herramienta a la hora de realizar el diseño de los distintos prototipos de la fase de diseño es debido a la interfaz visual intuitiva basada en arrastrar y soltar los distintos componentes, además de la posibilidad de implementar lógica en las aplicaciones con la que establecer las reglas que determinen el funcionamiento de la aplicación. Se presenta a continuación un primer prototipado de la aplicación.

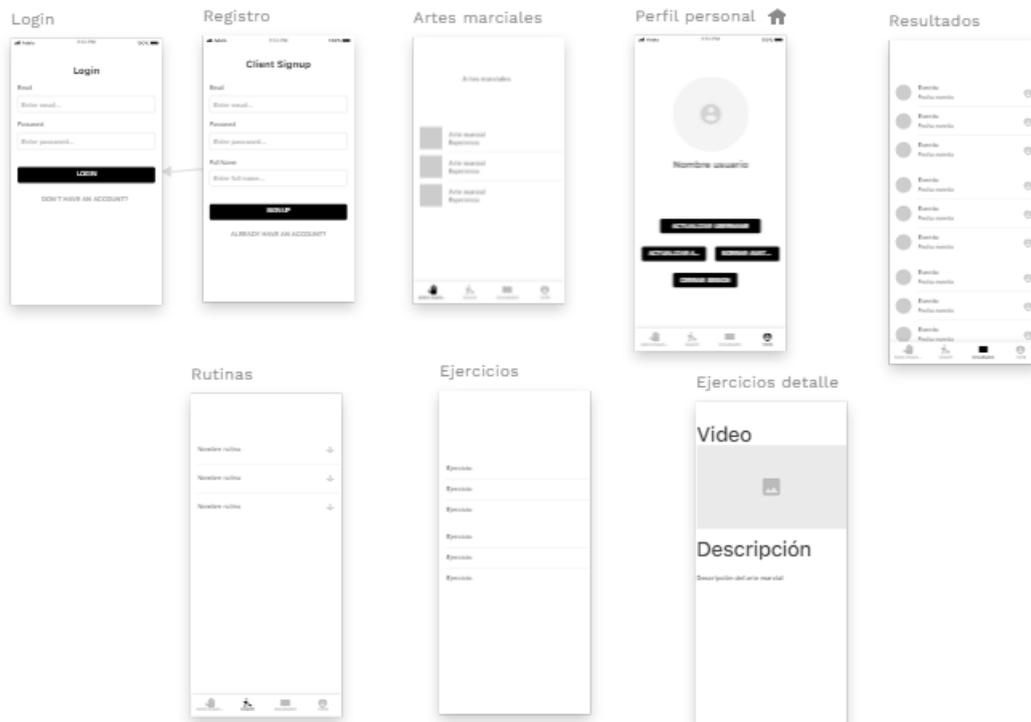


Ilustración 4 Adalo

Además, Adalo permite la visualización en tiempo real de la aplicación y la posibilidad de probarla en distintos dispositivos para asegurar el funcionamiento correcto de la aplicación y facilitando las pruebas con los usuarios.

4.1.3 WhatFont

Esta extensión para el navegador permite determinar cuales son las fuentes que son usadas en una determinada página web. Esto permite el uso en el proyecto de la letra usada por la compañía de la UFC en su página web.

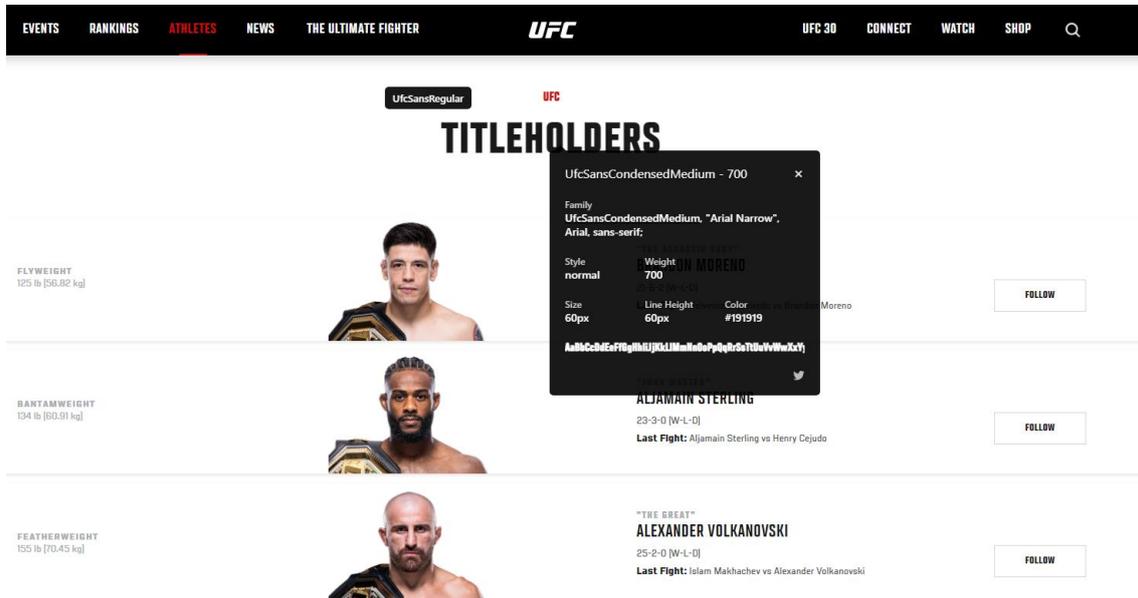


Ilustración 5: WhatFont

4.1.4 Google Fonts

En relación con el uso de WhatFont, se hace uso de Google Fonts para descargar las fuentes que van a ser utilizadas en la app.

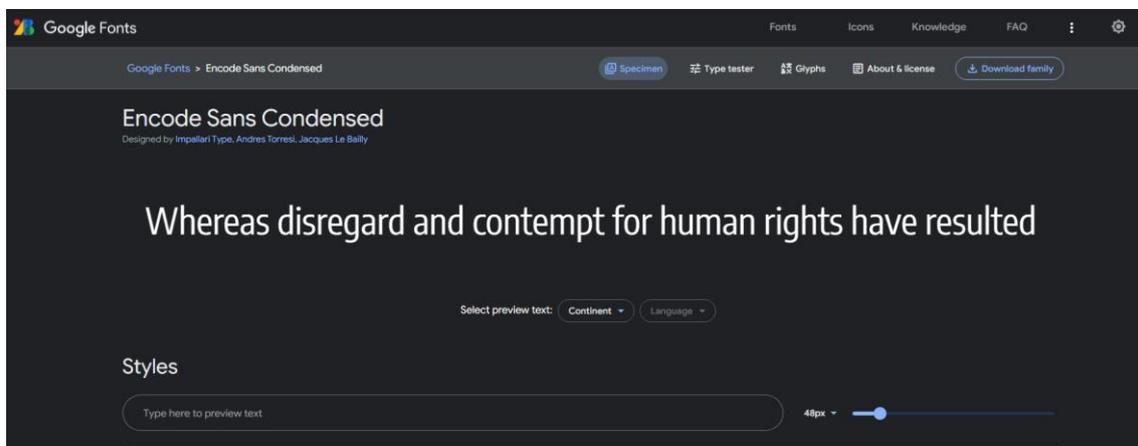


Ilustración 6: Google Fonts

4.1.5 Draw.io

Esta aplicación en línea de Diagrams.net permite la realización de distintos tipos de diagramas, como los diagramas de casos de uso, diagramas de secuencia o diagramas de paquetes.

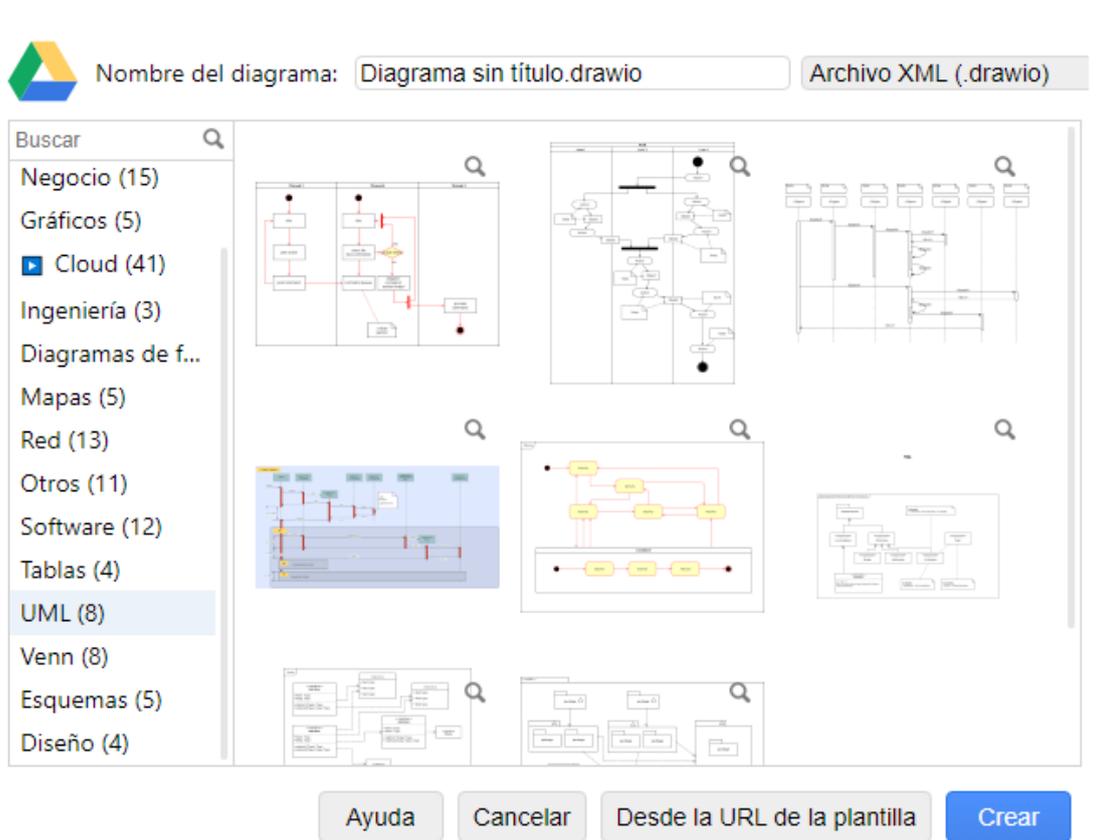


Ilustración 7: Draw.io

4.2 Herramientas de codificación

En este apartado se muestran las herramientas utilizadas en la parte de codificación del proyecto software.

4.2.1 Dart

Es el lenguaje predeterminado de Flutter, es de código abierto y hace uso de una máquina virtual para compilar el código de la aplicación a código nativo, tiene una sintaxis muy sencilla y es multiparadigma.

Dart ofrece la posibilidad de trabajo frontend como backend, es decir, permite el desarrollo de aplicaciones full stack.

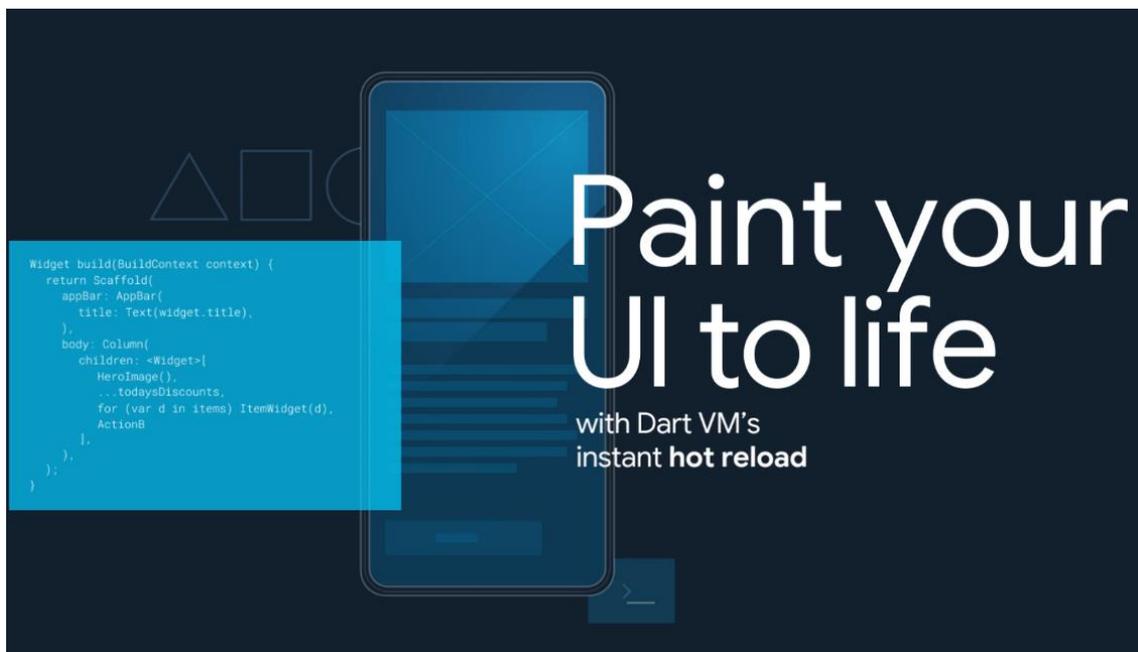
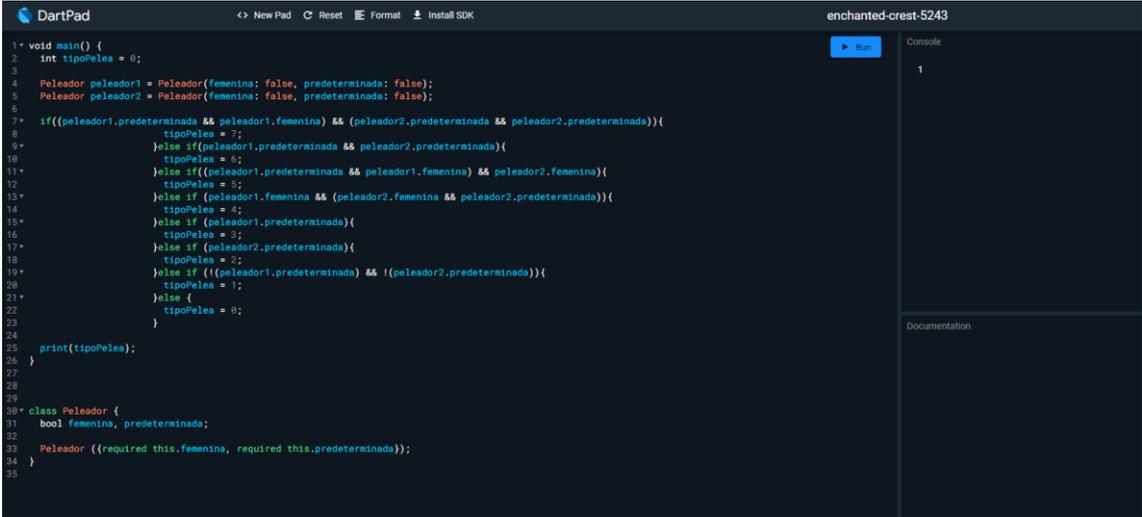


Ilustración 8: Dart

4.2.2 DartPad

DartPad es una herramienta de código abierto que permite interactuar con el lenguaje Dart desde cualquier navegador moderno. En el proyecto es usada para probar el funcionamiento de determinadas partes de la lógica de la aplicación sin tener que compilar todo el proyecto.



```
1* void main() {
2   int tipoPelea = 0;
3
4   Peleador peleador1 = Peleador(femenina: false, predeterminada: false);
5   Peleador peleador2 = Peleador(femenina: false, predeterminada: false);
6
7*  if((peleador1.predeterminada && peleador1.femenina) && (peleador2.predeterminada && peleador2.predeterminada)){
8     tipoPelea = 7;
9*   }else if(peleador1.predeterminada && peleador2.predeterminada){
10    tipoPelea = 6;
11*   }else if((peleador1.predeterminada && peleador1.femenina) && peleador2.femenina){
12    tipoPelea = 5;
13*   }else if (peleador1.femenina && (peleador2.femenina && peleador2.predeterminada)){
14    tipoPelea = 4;
15*   }else if (peleador1.predeterminada){
16    tipoPelea = 3;
17*   }else if (peleador2.predeterminada){
18    tipoPelea = 2;
19*   }else if (!(peleador1.predeterminada) && !(peleador2.predeterminada)){
20    tipoPelea = 1;
21*   }else {
22    tipoPelea = 0;
23   }
24
25   print(tipoPelea);
26 }
27
28
29
30* class Peleador {
31   bool femenina, predeterminada;
32
33   Peleador({required this.femenina, required this.predeterminada});
34 }
35
```

Ilustración 9: DartPad

4.2.3 Librerías Flutter

Las librerías son esenciales para la realización de ciertas funciones, las usadas en el proyecto son:

- **cupertino_icons:** proporciona iconos con diseño Apple.
- **Firestore_core:** configuración e inicialización básica para el uso de Firestore en la aplicación.
- **Firestore_auth:** autenticación de usuarios en Firestore.
- **Shared_preferences:** utilizada para guardar preferencias o datos locales en la aplicación.
- **Dio:** librería HTTP para realizar peticiones de red.
- **Intl:** mostrar texto y formato específicos en función del idioma y región del dispositivo.
- **http:** utilizada para realizar web scrapping.
- **Cloud_firestore:** proporciona una interfaz para acceder y manipular datos a tiempo real desde Firestore.
- **Flutter_exit_app:** implementa el cierre de la aplicación en función de si es un dispositivo Android o iOS.
- **Youtube_player_flutter:** permite reproducir vídeos de YouTube dentro de la aplicación mediante un reproductor personalizable.
- **Image_picker:** proporciona la capacidad de seleccionar imágenes desde la galería del dispositivo.
- **Firestore_storage:** permite subir, descargar y eliminar archivos de la base de datos.
- **Timezone:** permite obtener información sobre zonas horarias y realizar conversiones.
- **Font_awesome_flutter:** proporciona una amplia gama de iconos.

4.3 Herramientas de desarrollo

4.3.1 Visual Studio Code

Se utiliza Visual Studio Code tanto para el desarrollo de código backend como para el conocimiento de los distintos atajos de teclado que van a ser usados en Android Studio por una mayor comodidad.

4.3.2 Android Studio

Se utiliza Android Studio tanto para desarrollo de código backend como para el desarrollo de todo el código frontend de la aplicación. Constituye la principal herramienta de desarrollo utilizada en la elaboración del proyecto debido a la mayor comodidad que brinda a la hora de crear y utilizar las máquinas virtuales en las que se compilará el código.

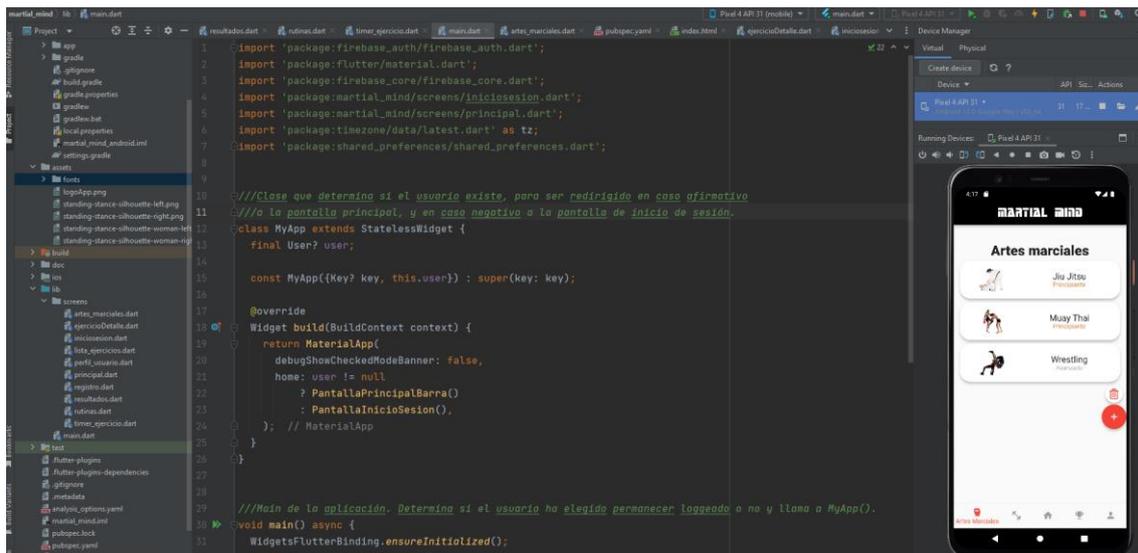


Ilustración 10: Android Studio

4.4 Herramientas de gestión del proyecto

4.4.1 Microsoft Office Word

Se utiliza esta herramienta para la documentación de toda actividad relacionada con el proyecto, desde la toma de anotaciones para la realización de una determinada funcionalidad del código hasta la realización de los documentos referentes a la memoria del proyecto.

4.4.2 Google Drive

Se utiliza esta herramienta para almacenar cualquier avance referente al código o a la documentación del proyecto, manteniendo así en todo momento una copia de seguridad en la nube del progreso realizado.

4.5 Herramientas adicionales

4.5.1 Firebase

Se utiliza Firebase con el objetivo de almacenar la información del usuario y de las artes marciales haciendo uso de Cloud Firestore. Además del uso de Firebase Storage para el almacenamiento de las imágenes correspondientes a las distintas artes marciales y los avatares de usuario.

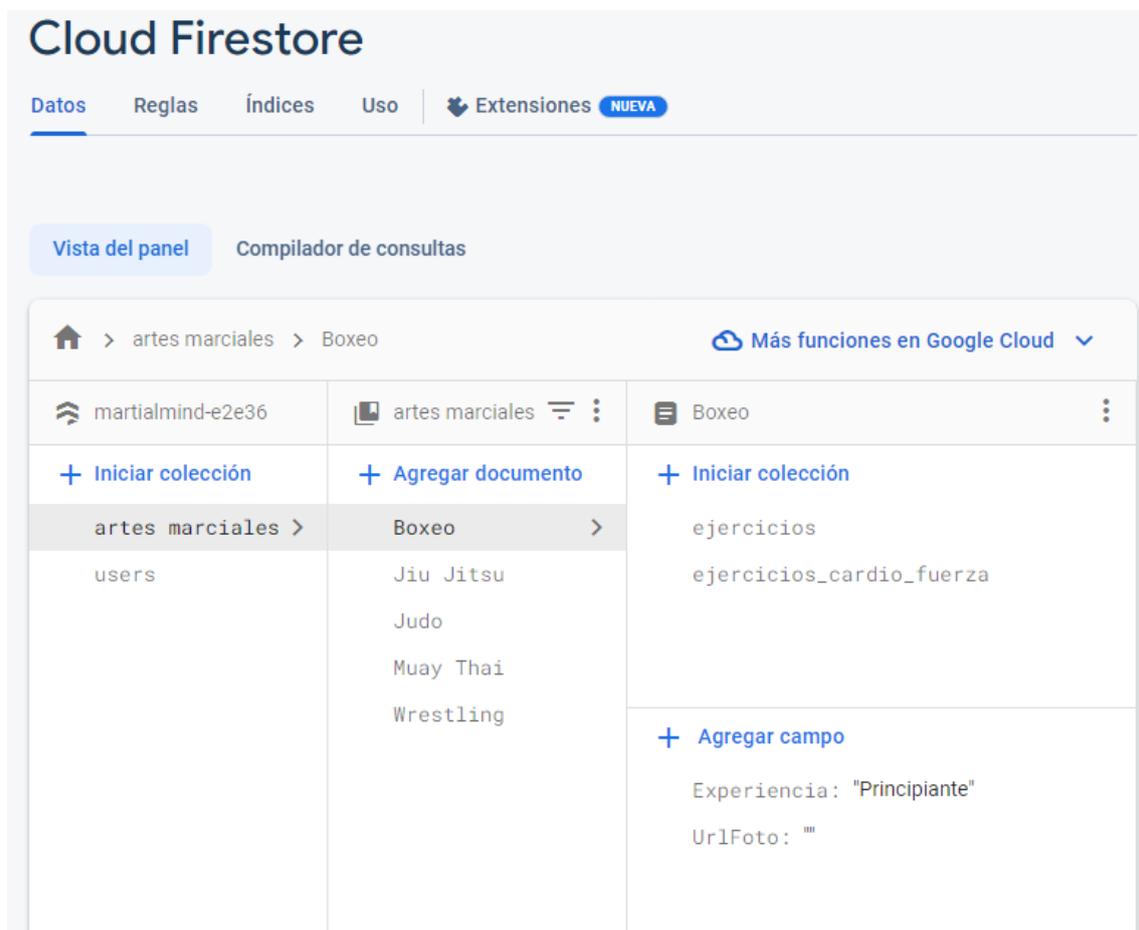


Ilustración 11: Cloud Firestore

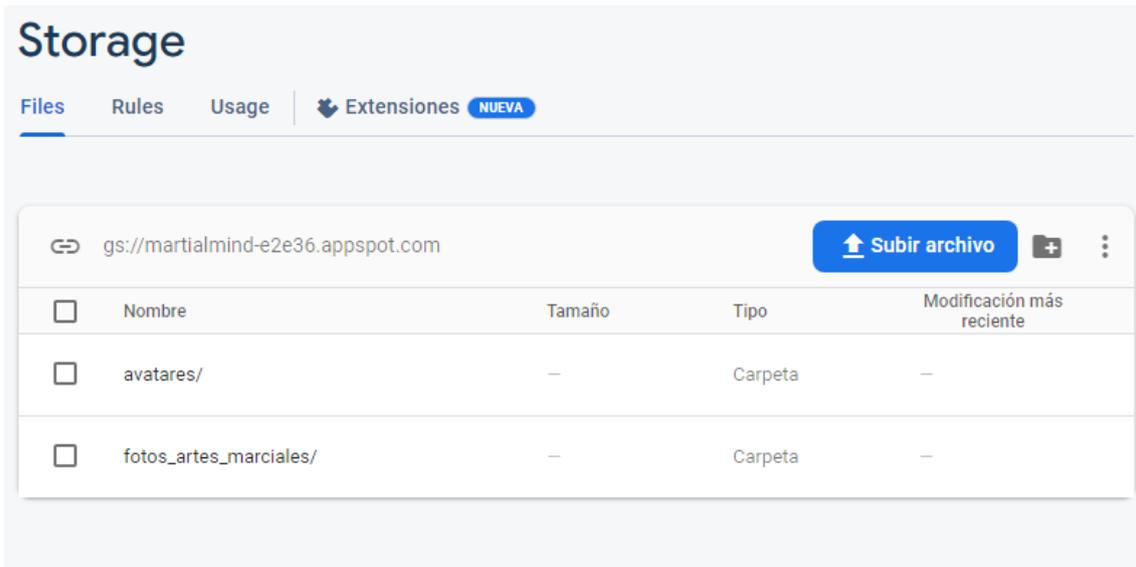


Ilustración 12: Firebase Storage

4.5.2 SportsData.io

SportsData.io es la API responsable de toda la información referente a los eventos y resultados. Ofrece datos *Scrambled* en su prueba gratuita, hay determinados campos específicos de los que no se puede obtener la información a menos de que se pague una suscripción. A pesar de ello es posible recuperar la mayoría de datos relevantes en relación a los eventos, como el nombre, la fecha o el estado de finalización.

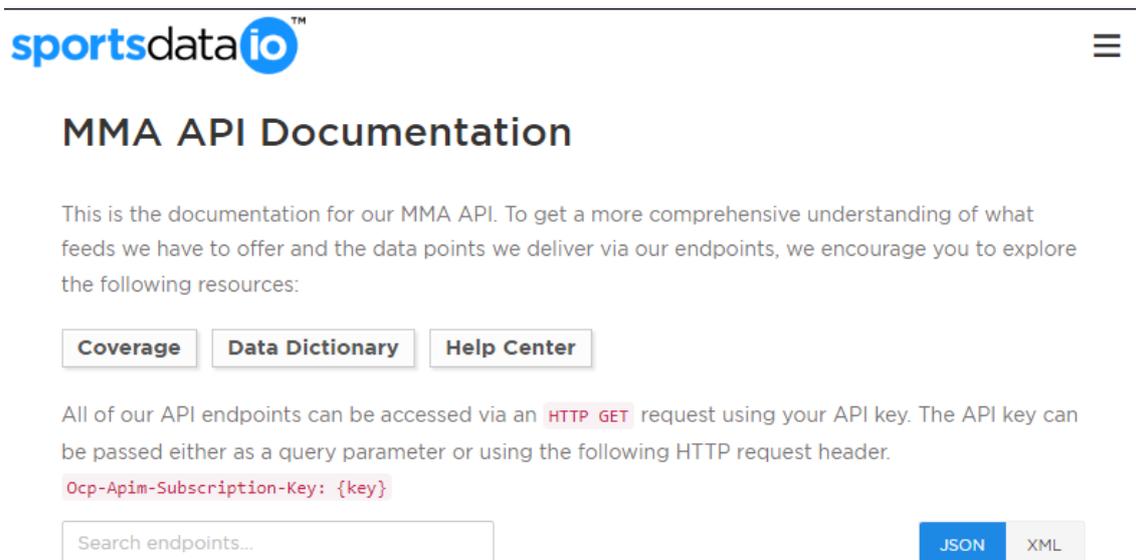


Ilustración 13: SportsData.io

5. Aspectos relevantes del desarrollo

En esta sección del documento se presentan y describen los elementos clave identificados durante el desarrollo del proyecto, se detallan aquellos pasos que se han seguido en cada fase permitiendo su correcta elaboración.

5.1 Marco de trabajo

Durante el desarrollo del proyecto se adoptó el enfoque del Proceso Unificado como marco de trabajo, caracterizado por su orientación a casos de uso y su carácter iterativo e incremental.

El Proceso Unificado se basa en cuatro fases principales detalladas en el *Anexo I. Planificación temporal*: Inicio, Elaboración, Construcción y Transición, cada una dividida en iteraciones, que aportan un incremento al producto en desarrollo, añadiendo o mejorando sus funcionalidades.

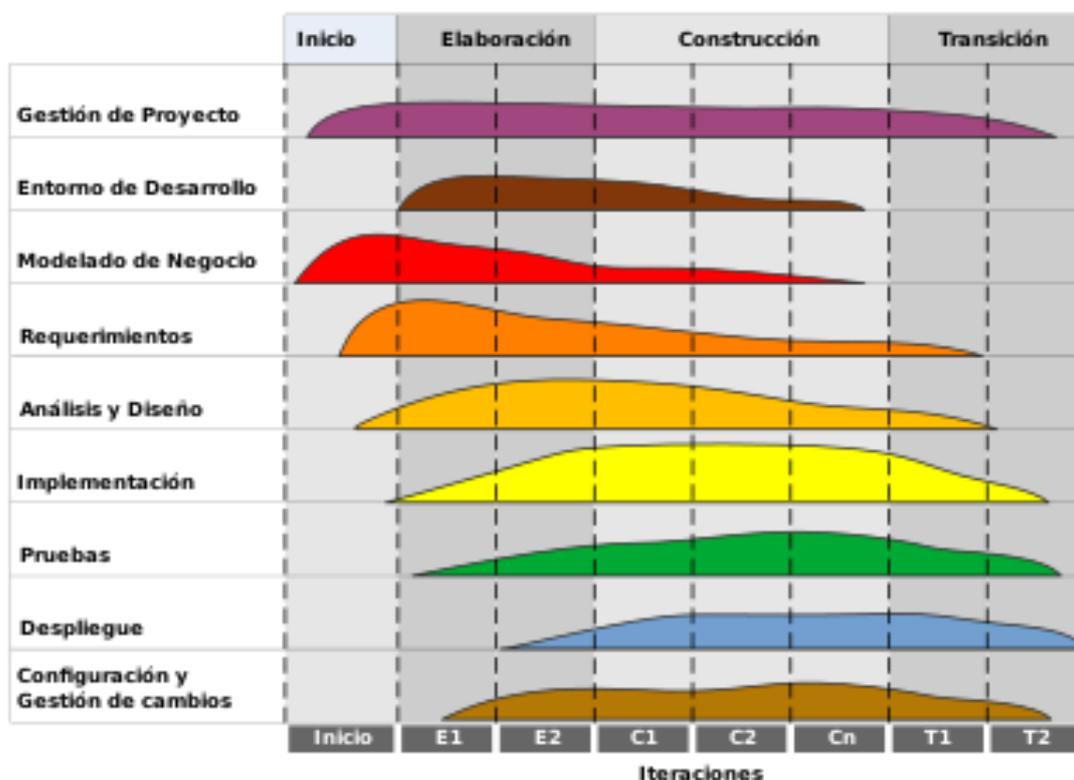


Ilustración 14: Fases del proceso unificado

Al estar dirigido por casos de uso, permite capturar los requisitos funcionales y guiar el desarrollo a través del diseño, implementación y pruebas. Además, no existe un modelo único, se parte de una arquitectura inicial que irá siendo refinada y mejorada obteniendo el producto final.

5.2 Planificación temporal

Uno de los primeros pasos realizados fue la planificación temporal del proyecto, que permitió identificar todas las tareas necesarias y estimar el tiempo requerido para completarlas.

Mediante esta planificación es posible establecer una fecha estimada de finalización del proyecto basada en los tiempos estimados de cada tarea individual. Al usar el marco de trabajo del Proceso Unificado, es necesario establecer la relación entre las tareas propias de cada marco y las tareas definidas en la planificación temporal.

A continuación, se presentan las tareas necesarias para llevar a cabo el proyecto, junto con su respectivo diagrama de Gantt que proporciona una visualización clara de la secuencia y duración de las tareas a lo largo del tiempo.

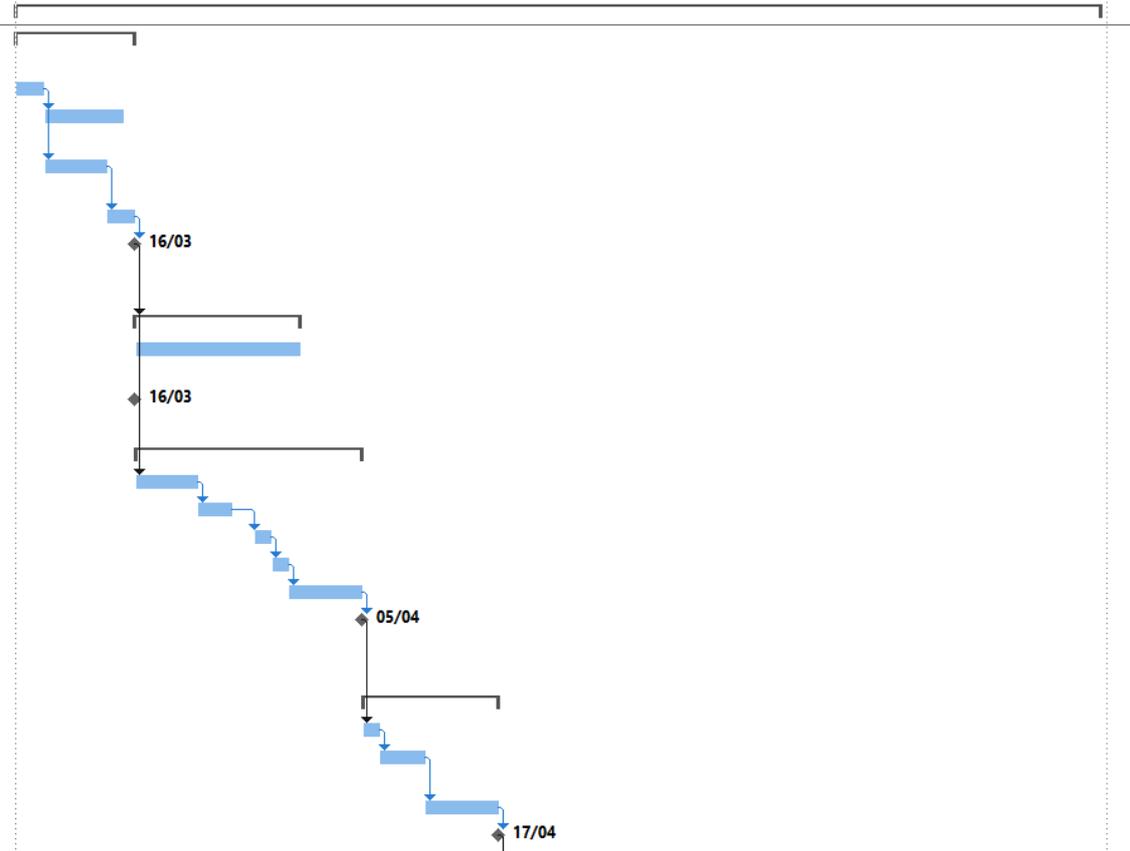


Ilustración 17: Diagrama de Gantt

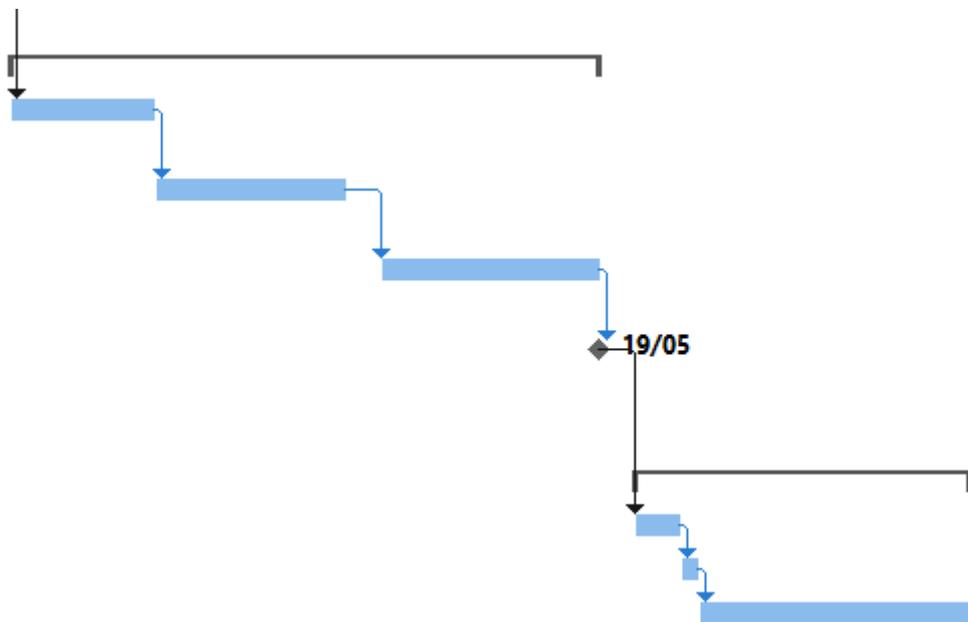


Ilustración 18: Diagrama de Gantt 2

5.3 Especificación de requisitos

Durante esta fase, se han identificado y recopilado los objetivos que se pretenden lograr con el proyecto para satisfacer los requisitos del sistema. Además, se ha elaborado un catálogo de requisitos, que pretende capturar las necesidades que el sistema debe cumplir y ejerce de base para las siguientes etapas del proyecto. Incluye los requisitos de información, funcionales, actores y casos de uso.

Se presentan a continuación ejemplos de cada una de las etapas de la especificación de requisitos, puede consultarse el *Anexo II Especificación de requisitos del software* para más información:

5.3.1 Objetivos del sistema

Los objetivos del sistema son los siguientes:

- Gestión de usuarios
- Gestión de artes marciales
- Gestión de competiciones
- Gestión de resultados
- Gestión de ejercicios
- Gestión de rutinas

En la *tabla 1* se puede observar un ejemplo de especificación de objetivos del sistema:

| | |
|---------------------|--|
| OBJ-0004 | Gestión de resultados |
| Versión | 1.0 |
| Autores | Francisco José Cruz Pérez |
| Fuentes | |
| Descripción | El sistema deberá <i>gestionar los resultados de las distintas competiciones, así como los peleadores de cada pelea, sus fotos y el ganador de la misma.</i> |
| Subobjetivos | Ninguno |
| Importancia | importante |
| Urgencia | inmediata |
| Estado | valido |
| Estabilidad | alta |
| Comentarios | Ninguno |

Tabla 1: Objetivo del sistema gestión de resultados

5.3.2 Requisitos de información

Los requisitos de información del sistema son los siguientes:

- Arte Marcial
- Ejercicio
- Peleador
- Pelea
- Evento
- Rutina
- Usuario registrado
- Ejercicio rutina

En la *tabla 2* se puede observar un ejemplo de especificación de requisitos de información del sistema:

| | | |
|--------------------------------|---|---------------|
| IRQ-0001 | Arte Marcial | |
| Versión | 1.0 | |
| Autores | Francisco José Cruz Pérez | |
| Fuentes | | |
| Dependencias | Ninguno | |
| Descripción | El sistema deberá almacenar la información correspondiente a <i>las distintas artes marciales que los usuarios van a poder aprender</i> . En concreto: | |
| Datos específicos | <ul style="list-style-type: none"> • Nombre • Foto • Lista de ejercicios • Experiencia • Si está seleccionada o no | |
| Tiempo de vida | Medio | Máximo |
| Ocurrencias simultáneas | Medio | Máximo |
| Importancia | importante | |
| Urgencia | inmediata | |
| Estado | valido | |
| Estabilidad | alta | |
| Comentarios | Ninguno | |

Tabla 2: Requisito del sistema Arte Marcial

5.3.3 Requisitos funcionales

Son los requisitos que expresan cómo interacciona el sistema con su entorno además de cuales van a ser su estado y funcionamiento.

Se muestran a continuación ejemplos para cada tipo de requisito funcional del sistema:

- **Actores:** representan a los usuarios que interactúan con el sistema.

| | |
|--------------------|---|
| ACT-0002 | Usuario loggeado |
| Versión | 1.0 |
| Autores | Francisco José Cruz Pérez |
| Fuentes | ? |
| Descripción | Este actor representa <i>a cualquier usuario que dispone de una cuenta y ha realizado la autenticación de credenciales.</i> |
| Comentarios | Ninguno |

Tabla 3 Requisito funcional del sistema Usuario loggeado

- **Casos de uso:** describen el comportamiento dinámico del sistema desde el punto de vista del actor y el modo en el que interactúa con él.

| | | |
|----------------------------|---|---|
| UC-0002 | Inicio de sesión | |
| Versión | 1.0 | |
| Autores | Francisco José Cruz Pérez | |
| Fuentes | | |
| Dependencias | <ul style="list-style-type: none"> • [OBJ-0001] Gestión de usuarios • [NFR-0001] Base de datos Firebase | |
| Descripción | El sistema deberá comportarse tal como se describe en el siguiente caso de uso. | |
| Precondición | | |
| Secuencia normal | Paso | Acción |
| | 1 | El actor Usuario loggeado (ACT-0002) realiza la solicitud de login al sistema. |
| | 2 | El sistema solicita al usuario: <i>correo electrónico</i> y <i>contraseña</i> . |
| | 3 | El actor Usuario sin loggear (ACT-0002) proporciona al sistema los datos y solicita el login. |
| | 4 | El sistema dirige al usuario a la pantalla. |
| Postcondición | | |
| Excepciones | Paso | Acción |
| | 3 | Los datos proporcionados por el usuario son inválidos debido a correo no registrado o contraseña incorrecta, el sistema informa de esto al usuario con el mensaje pertinente y este caso de uso queda sin efecto. |
| Rendimiento | Paso | Tiempo máximo |
| | - | |
| Frecuencia esperada | | |
| Importancia | importante | |
| Urgencia | inmediata | |
| Estado | valido | |
| Estabilidad | alta | |
| Comentarios | Ninguno | |

Tabla 4 Requisito funcional caso de uso Inicio de sesión

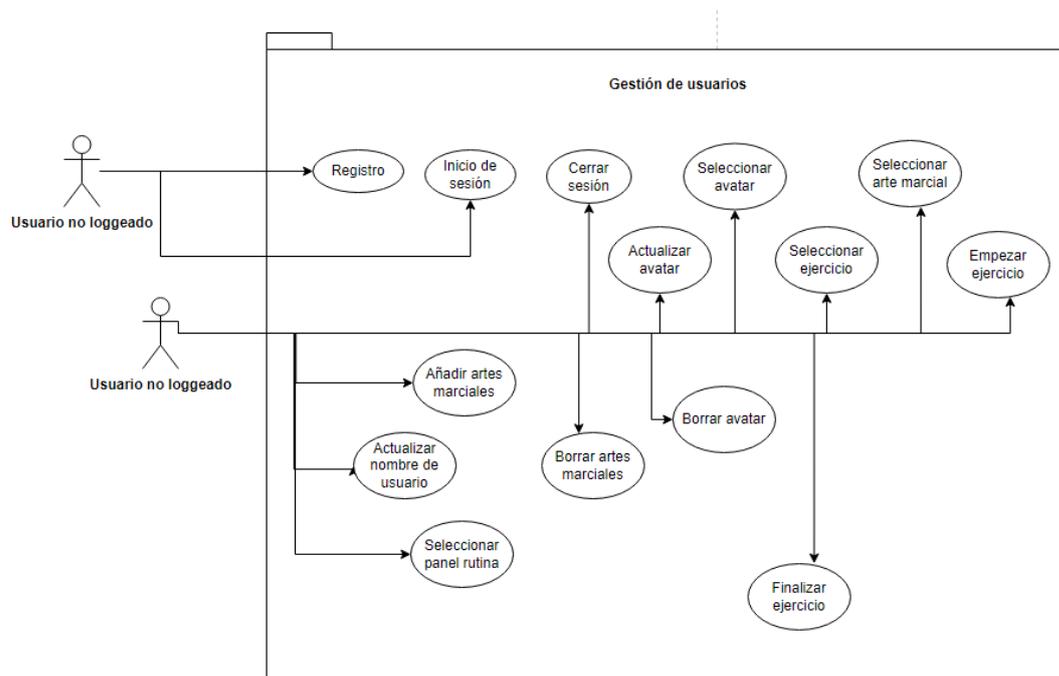


Ilustración 19: Diagrama de paquetes Gestión de usuarios.

5.3.4 Requisitos no funcionales

Los requisitos no funcionales hacen referencia a las restricciones sobre el espacio de posibles soluciones.

| NFR-0003 | Disponibilidad |
|---------------------|--|
| Versión | 1.0 |
| Autores | Francisco José Cruz Pérez |
| Fuentes | |
| Dependencias | Ninguno |
| Descripción | El sistema deberá <i>poder recuperar los datos necesarios para cada usuario en todo momento.</i> |
| Importancia | importante |
| Urgencia | inmediata |
| Estado | valido |
| Estabilidad | alta |
| Comentarios | Ninguno |

Tabla 5: Definición requisito no funcional Disponibilidad

5.4 Análisis de requisitos

En esta fase se lleva a cabo un análisis exhaustivo de todos los requisitos recogidos en la fase anterior, determinando el diagrama de clases, los diagramas de secuencia y una vista de la arquitectura del modelo de análisis.

Para mayor detalle consultar el documento *Anexo III: Especificación de diseño*.

5.4.1 Diagrama de clases

Nos permite representar el modelo de dominio del sistema, recoge las necesidades de almacenamiento y gestión de información del sistema en la base de datos.

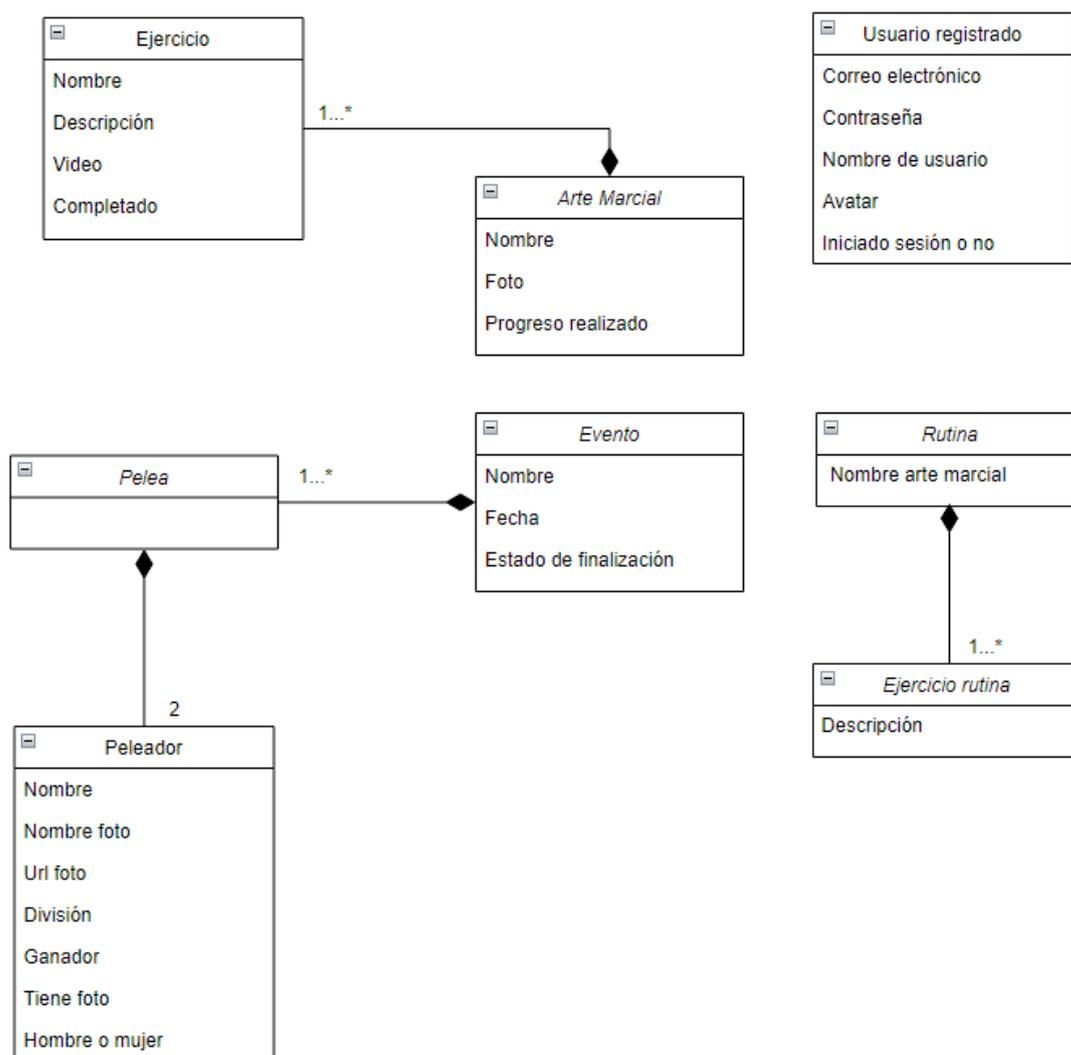


Ilustración 20: Modelo de dominio del sistema

5.4.2 Diagramas de secuencia

Durante la realización de los casos de uso en la fase de análisis se detallan los pasos e interacciones a realizar mediante los diagramas de secuencia.

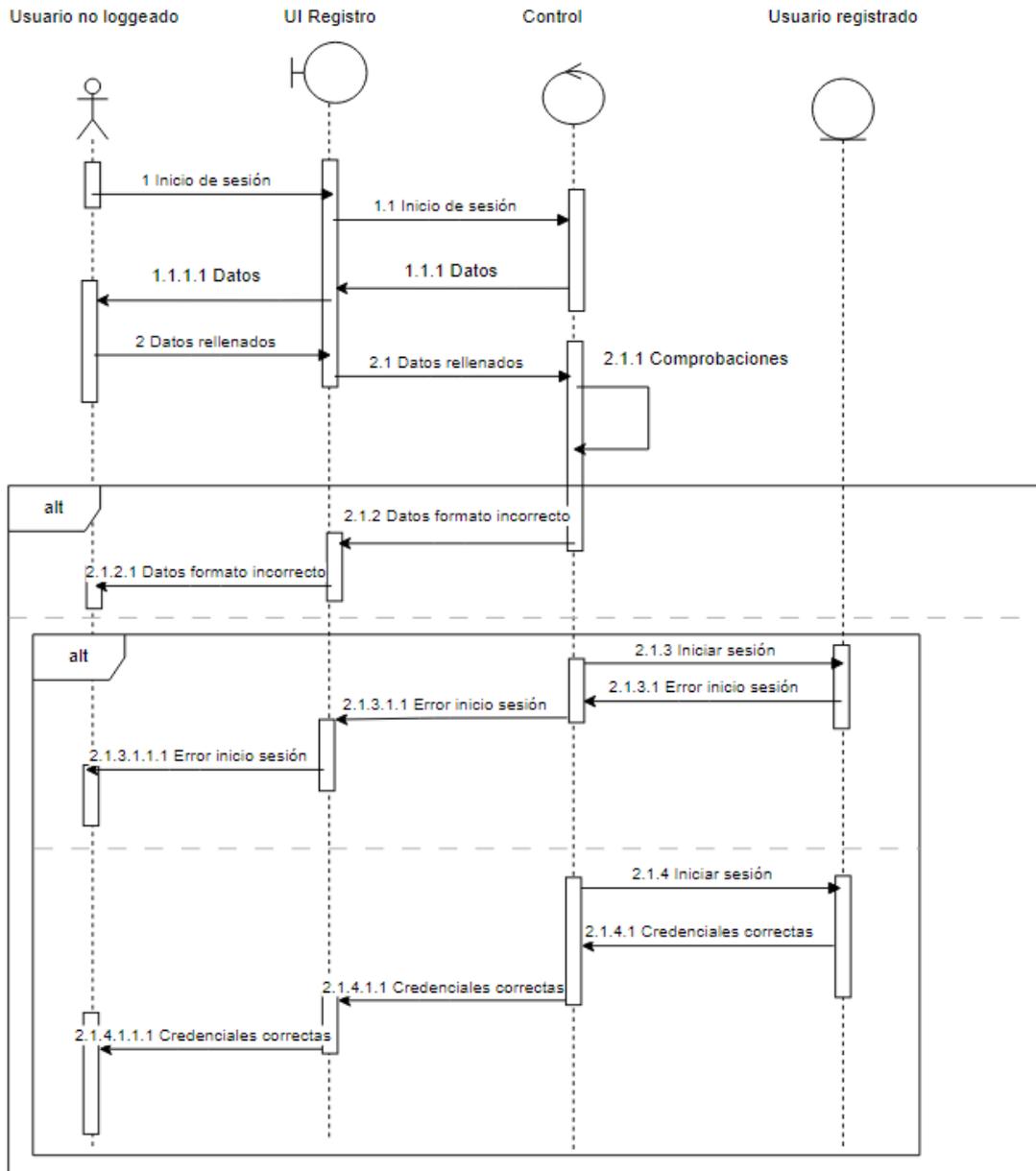


Ilustración 21: Diagrama de secuencia de Inicio de sesión

5.4.3 Vista de arquitectura

En la vista de arquitectura se ordenan las clases obtenidas de los diagramas de secuencia por capas, se presenta a continuación el resultado del diagrama.

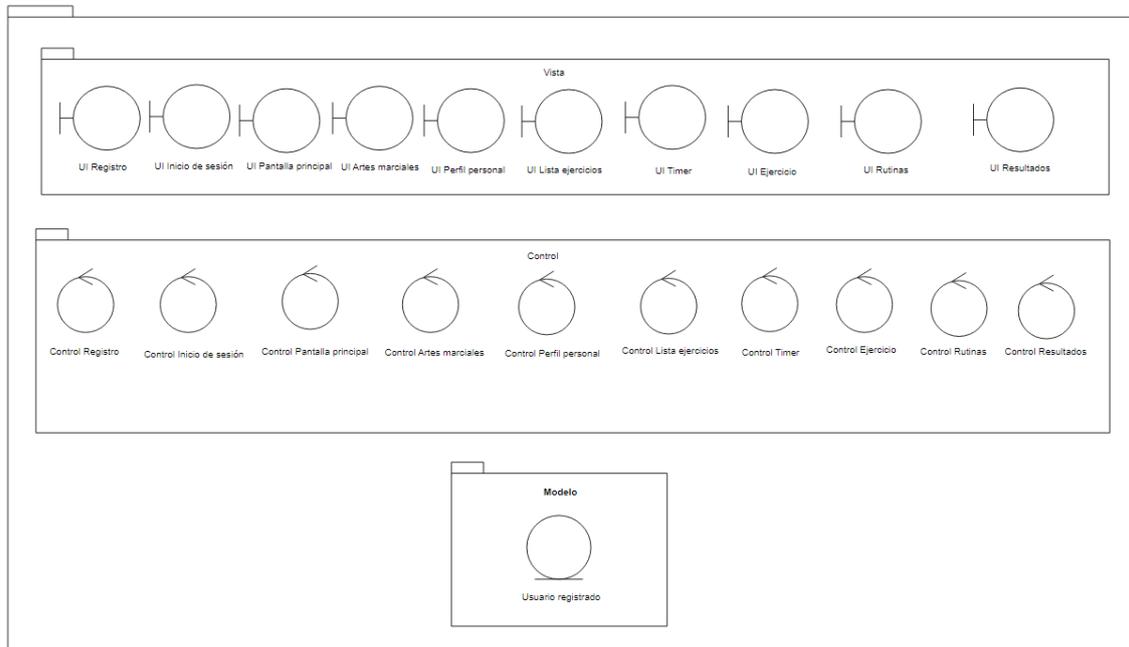


Ilustración 22 Vista de arquitectura

5.5 Implementación

Prosiguiendo con el desarrollo del proyecto, se realiza la implementación de la aplicación con el uso de las herramientas definidas en el apartado de *Técnicas y Herramientas*.

La implementación del proyecto se llevó a cabo prestando especial atención a la correcta recuperación de los datos, tanto de la API como de Firebase, ya que constituyen la principal estructura de datos de la aplicación. Se destacan a continuación los aspectos relevantes relacionados con esta tarea:

- Integración API: la integración de la API de SportsData.io en la aplicación se ha realizado cuidadosamente, siguiendo las mejores prácticas de desarrollo. Se han establecido las conexiones necesarias con los endpoints y se han configurado los métodos adecuados para acceder a los datos proporcionados por la API.

```
[{"eventId":204,"leagueId":1,"Name":"UFC Fight Night: Strickland vs. Isaacs","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-01-14T00:00:00","DateTime":"2023-01-14T15:00:00","Status":"Final","Active":true},{"eventId":205,"leagueId":1,"Name":"UFC 283: Israel vs. Hill","ShortName":"UFC 283","Season":2023,"Day":"2023-01-21T00:00:00","DateTime":"2023-01-21T17:30:00","Status":"Final","Active":true},{"eventId":206,"leagueId":1,"Name":"UFC Fight Night: Leds vs. Spiva","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-02-04T00:00:00","DateTime":"2023-02-04T22:00:00","Status":"Final","Active":true},{"eventId":207,"leagueId":1,"Name":"UFC 284: Makhachev vs. Volkov","ShortName":"UFC 284","Season":2023,"Day":"2023-02-11T00:00:00","DateTime":"2023-02-11T18:00:00","Status":"Final","Active":true},{"eventId":208,"leagueId":1,"Name":"UFC Fight Night: Andrade vs. Blanchfield","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-02-18T00:00:00","DateTime":"2023-02-18T21:00:00","Status":"Final","Active":true}, {"eventId":209,"leagueId":1,"Name":"UFC Fight Night: Hill vs. Allen","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-02-25T00:00:00","DateTime":"2023-02-25T18:00:00","Status":"Final","Active":true}, {"eventId":210,"leagueId":1,"Name":"UFC 285: Jones vs. Ganon","ShortName":"UFC 285","Season":2023,"Day":"2023-03-04T00:00:00","DateTime":"2023-03-04T17:30:00","Status":"Final","Active":true}, {"eventId":211,"leagueId":1,"Name":"UFC Fight Night: Van vs. Dall'Oglio","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-03-11T00:00:00","DateTime":"2023-03-11T15:00:00","Status":"Final","Active":true}, {"eventId":212,"leagueId":1,"Name":"UFC 286: Edwards vs. Usman 2","ShortName":"UFC 286","Season":2023,"Day":"2023-03-18T00:00:00","DateTime":"2023-03-18T13:30:00","Status":"Final","Active":true}, {"eventId":213,"leagueId":1,"Name":"UFC Fight Night: Vera vs. Song","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-03-25T00:00:00","DateTime":"2023-03-25T18:00:00","Status":"Final","Active":true}, {"eventId":214,"leagueId":1,"Name":"UFC 287: Pereira vs. Adesanya 2","ShortName":"UFC 287","Season":2023,"Day":"2023-04-01T00:00:00","DateTime":"2023-04-01T12:00:00","Status":"Final","Active":true}, {"eventId":215,"leagueId":1,"Name":"UFC Fight Night: Holloway vs. Allen","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-04-15T00:00:00","DateTime":"2023-04-15T17:30:00","Status":"Final","Active":true}, {"eventId":216,"leagueId":1,"Name":"UFC Fight Night: Song vs. Sloan","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-04-22T00:00:00","DateTime":"2023-04-22T16:30:00","Status":"Final","Active":true}, {"eventId":217,"leagueId":1,"Name":"UFC 288: Sterling vs. Cejudo","ShortName":"UFC 288","Season":2023,"Day":"2023-04-29T00:00:00","DateTime":"2023-04-29T13:00:00","Status":"Final","Active":true}, {"eventId":218,"leagueId":1,"Name":"UFC Fight Night: Rozenstruk vs. Almeida","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-05-13T00:00:00","DateTime":"2023-05-13T12:00:00","Status":"Final","Active":true}, {"eventId":219,"leagueId":1,"Name":"UFC Fight Night: Vera vs. Hill 2","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-05-20T00:00:00","DateTime":"2023-05-20T15:00:00","Status":"Final","Active":true}, {"eventId":220,"leagueId":1,"Name":"UFC 289: Moes vs. Aldana","ShortName":"UFC 289","Season":2023,"Day":"2023-06-03T00:00:00","DateTime":"2023-06-03T17:30:00","Status":"Final","Active":true}, {"eventId":221,"leagueId":1,"Name":"UFC Fight Night: Vettori vs. Covington","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-06-10T00:00:00","DateTime":"2023-06-10T17:30:00","Status":"Final","Active":true}, {"eventId":222,"leagueId":1,"Name":"UFC Fight Night: Lopez","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-06-24T00:00:00","DateTime":"2023-06-24T11:00:00","Status":"Final","Active":true}, {"eventId":223,"leagueId":1,"Name":"UFC Fight Night: Strickland vs. Magomedov","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-07-01T00:00:00","DateTime":"2023-07-01T16:00:00","Status":"Final","Active":true}, {"eventId":224,"leagueId":1,"Name":"UFC 290: Volkov vs. Rookus","ShortName":"UFC 290","Season":2023,"Day":"2023-07-08T00:00:00","DateTime":"2023-07-08T18:00:00","Status":"Scheduled","Active":true}, {"eventId":225,"leagueId":1,"Name":"UFC Fight Night: Aspinall vs. Silva","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-07-15T00:00:00","DateTime":"2023-07-15T17:30:00","Status":"Scheduled","Active":true}, {"eventId":226,"leagueId":1,"Name":"UFC 291: Holm vs. Guehls 2","ShortName":"UFC 291","Season":2023,"Day":"2023-07-22T00:00:00","DateTime":"2023-07-22T18:00:00","Status":"Scheduled","Active":true}, {"eventId":227,"leagueId":1,"Name":"UFC Fight Night: Sanjoevs vs. Nurmagomedov","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-08-05T00:00:00","DateTime":"2023-08-05T17:30:00","Status":"Scheduled","Active":true}, {"eventId":228,"leagueId":1,"Name":"UFC Fight Night: Ibragimov vs. Don Ajlos","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-08-12T00:00:00","DateTime":"2023-08-12T16:00:00","Status":"Scheduled","Active":true}, {"eventId":229,"leagueId":1,"Name":"UFC Fight Night: Game vs. Spivey","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-08-19T00:00:00","DateTime":"2023-08-19T12:00:00","Status":"Scheduled","Active":true}, {"eventId":230,"leagueId":1,"Name":"UFC 292: Sterling vs. O'Malley","ShortName":"UFC 292","Season":2023,"Day":"2023-08-26T00:00:00","DateTime":"2023-08-26T17:30:00","Status":"Scheduled","Active":true}, {"eventId":231,"leagueId":1,"Name":"UFC Fight Night: Grasso vs. Shevchenko 2","ShortName":"UFC Fight Night","Season":2023,"Day":"2023-09-02T00:00:00","DateTime":"2023-09-02T18:00:00","Status":"Scheduled","Active":true}, {"eventId":232,"leagueId":1,"Name":"UFC 293","ShortName":"UFC 293","Season":2023,"Day":"2023-09-09T00:00:00","DateTime":"2023-09-09T18:00:00","Status":"Scheduled","Active":true}, {"eventId":233,"leagueId":1,"Name":"UFC 294","ShortName":"UFC 294","Season":2023,"Day":"2023-09-16T00:00:00","DateTime":"2023-09-16T18:00:00","Status":"Scheduled","Active":true}],
```

Ilustración 23: JSON recuperado de SportsData.io

- Integración con Firebase: se estableció la conexión y la integración con Firebase para recuperar los datos almacenados en la nube. Las características proporcionadas por Firebase, como Firebase Realtime Database o Firebase Firestore, se utilizan para acceder y recuperar datos de manera eficiente y segura.

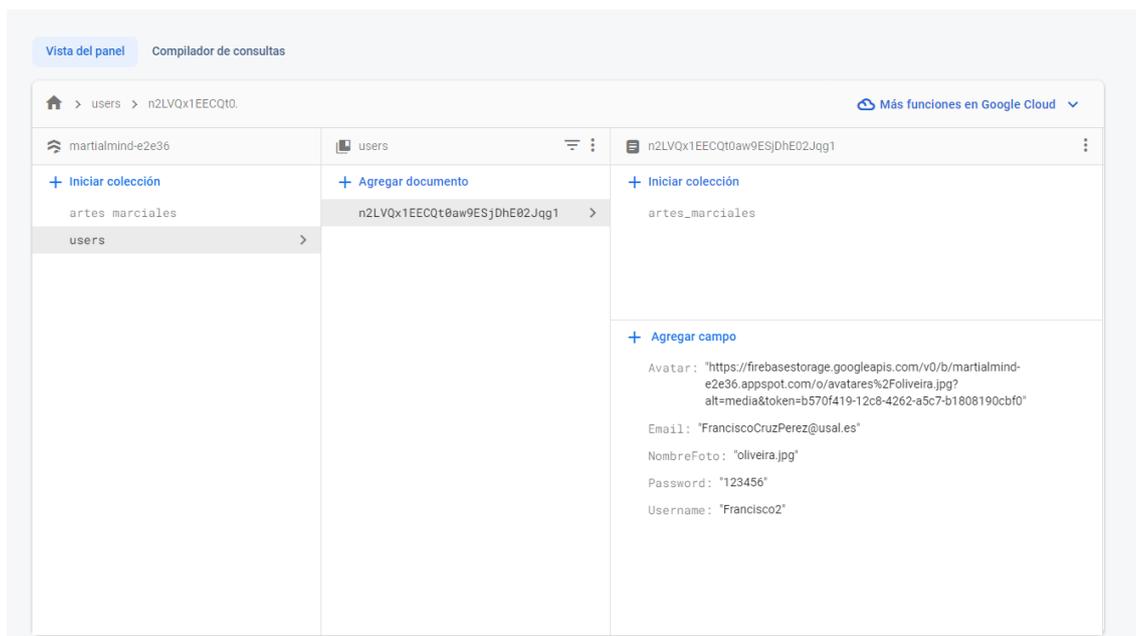


Ilustración 24: Estructura de los datos de usuario Firebase

- Sincronización de datos: se implementó un mecanismo de sincronización adecuado para mantener actualizados los datos entre la API y Firebase. Se han establecido procesos recurrentes o en tiempo real para garantizar que los datos estén siempre actualizados y reflejen los cambios en ambos sistemas.
- Manejo de errores y excepciones: Se han implementado mecanismos para manejar adecuadamente los errores y excepciones que pueden ocurrir al recuperar los datos. Se revisaron posibles problemas de conexión, errores de validación o respuestas inesperadas y se tomaron medidas para notificar a los usuarios.

En resumen, durante la implementación del proyecto, se dedicó especial atención a la correcta recuperación de los datos tanto de la API como de Firebase. Se realizaron las integraciones necesarias y se implementaron mecanismos adecuados para garantizar la obtención precisa y segura de los datos. Se establecieron procesos de sincronización y se manejaron los posibles errores y excepciones que pudieran surgir durante este proceso.

6. Conclusiones y líneas de trabajo

En este apartado final de la memoria del proyecto se presentan las conclusiones que se han obtenido del desarrollo del mismo, además de las líneas de trabajo futuras.

6.1 Conclusiones

La conclusión general, es fundamental comprender el proyecto de desarrollo de software como un conjunto completo de todas sus fases, y no solo enfocarse únicamente en la etapa de la implementación sin haber realizado un estudio y análisis previo. Además, recalcar la importancia del aprendizaje y puesta en práctica de nuevas herramientas de desarrollo como Flutter, que, si bien es emocionante y puede ofrecer beneficios significativos, como se ha dicho anteriormente, constituye tan solo una parte del desarrollo del proyecto software.

Además, destacar que el Proceso Unificado desempeña un papel fundamental en el éxito del desarrollo del proyecto software, ya que, sumado a todas las características expuestas sobre el mismo a lo largo del documento, seguirlo hace que se establezcan prácticas y estándares de desarrollo que promueven la calidad y eficiencia en todas las etapas del proyecto.

Por último, finalizar las conclusiones agradeciendo la posibilidad de abarcar un trabajo de esta magnitud, prestando atención al detalle para ofrecer al usuario el mejor producto posible, adquiriendo durante ello numerosos conceptos teóricos como el uso de API REST o la implementación orientada a objetos de Dart, o la utilización de un IDE como Android Studio.

6.2 Líneas de desarrollo futuras

En este apartado se exponen las posibles líneas futuras de trabajo del proyecto desarrollado:

- **Integración con Apple Vision Pro:** esto permitiría aprovechar las capacidades de reconocimiento de imágenes y visión computarizada de Apple para hacer de la experiencia una más inmersiva, pudiendo detectar los movimientos durante la práctica de un ejercicio, lo que ayudaría al usuario a corregir la ejecución de la técnica.
- **Adición de nuevas artes marciales y documentación:** permitiría a los usuarios explorar una variedad más amplia de disciplinas y técnicas, además de proporcionar información detallada sobre la historia, filosofía o fundamentos de cada arte marcial, logrando un mayor enriquecimiento por parte del usuario.
- **Logros y recompensas:** una funcionalidad que motiva al usuario a progresar en su entrenamiento hasta alcanzar un cierto hito que le brinde un logro especial o una recompensa virtual, como cinturones en la disciplina o medallas. Además de la posibilidad de desbloquear contenidos exclusivos como vídeos de los peleadores personalizados.

7. Bibliografía

- Documentación dart doc no aparece: <https://stackoverflow.com/questions/67651141/flutter-dartdoc-how-to-document-state-objects-in-statefull-widgets/67653598#67653598>
- ¿Qué es Dart?: <https://open-bootcamp.com/aprender-programar/que-es-dart>
- Principios básicos expresiones regulares: <https://www.freecodecamp.org/news/simple-regex-tricks-for-beginners-3acb3fa257cb/>
- Dropdown Button: <https://api.flutter.dev/flutter/material/DropdownButton-class.html>
- Borrar archivos con Cloud Storage: <https://firebase.google.com/docs/storage/flutter/delete-files?hl=es>
- Subir imágenes a Cloud Storage: https://www.youtube.com/watch?v=iAxgbE-a2SI&ab_channel=CódigoCorrecto
- Reproductor de vídeos de YouTube: https://pub.dev/packages/youtube_player_flutter/install
- Salida segura de la aplicación: https://pub.dev/packages/flutter_exit_app
- Hacer que no se actualice automáticamente una página con el initState: https://www.youtube.com/watch?v=aw8OG_zfuxY&ab_channel=EasyLearning
- Fuente de la app: <https://fonts.google.com/specimen/Encode+Sans+Condensed>
- Dar la vuelta a una lista en Flutter: <https://es.code-paper.com/q/15193-reverse-list-in-flutter>
- DartPad: <https://dartpad.dev/>
- Instalar fuentes en el proyecto: https://www.youtube.com/watch?v=11hf7o-kCGY&ab_channel=AbsaCoding
- Tipografía de la UFC: <https://fontmeme.com/fuentes/fuente-sternbach/>
- Multidex support: <https://docs.flutter.dev/deployment/android#enabling-multidex-support>
- Excepciones de firebase: <https://www.appsloveworld.com/flutter/100/6/check-if-an-email-already-exists-in-firebase-auth-in-flutter-app>

- Resolver propiedades de buildgradle a nivel de app:
<https://stackoverflow.com/questions/56938436/first-flutter-app-error-cannot-resolve-symbol-properties>
- Vincular Firebase a Flutter: <https://alejandro-mancilla-umsa.medium.com/conecta-tu-proyecto-flutter-con-firebase-266cb858618f>
- Autenticación de cliente: <https://developers.google.com/android/guides/client-auth?hl=es-419>