

FIGHT-zone

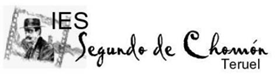
Proyecto 2 DAM



16 de junio de 2025

I.E.S Segundo de Chomón

Mario Pérez Gracia



Contenido

[1. Descripción del proyecto 3](#_Toc200365729)

[1.1 Contexto del proyecto 3](#_Toc200365730)

[1.1.1 Ámbito y entorno 3](#_Toc200365731)

[1.1.2 Análisis de la realidad. 3](#_Toc200365732)

[1.1.3 Solución y justificación de la solución propuesta 3](#_Toc200365733)

[1.1.4 Destinatarios. 3](#_Toc200365734)

[1.2 Objetivo del proyecto 3](#_Toc200365735)

[1.3 Objetivo del proyecto en lengua extranjera 3](#_Toc200365736)

[2 Documento de Acuerdo del proyecto. 3](#_Toc200365737)

[2.1 Requisitos funcionales y no funcionales. 3](#_Toc200365738)

[2.2 Tareas 3](#_Toc200365739)

[2.3 Metodología a seguir para la realización del proyecto. 3](#_Toc200365740)

[2.4 Planificación temporal de tareas. 3](#_Toc200365741)

[2.5 Presupuesto (gastos, ingresos, beneficio) 3](#_Toc200365742)

[2.6 Contrato/Pliego de condiciones 3](#_Toc200365743)

[2.7 Análisis de riesgos 3](#_Toc200365744)

[3 Documento de análisis y diseño 3](#_Toc200365745)

[3.1 Modelado de datos. 3](#_Toc200365746)

[3.2 Análisis y diseño del sistema funcional 3](#_Toc200365747)

[3.3 Análisis y diseño de la interfaz de usuario. 3](#_Toc200365748)

[3.4 Diseño de la arquitectura de la aplicación 3](#_Toc200365749)

[3.4.1 Tecnologías/Herramientas usadas y descripción de las mismas 3](#_Toc200365750)

[3.4.2 Arquitectura de componentes de la aplicación 4](#_Toc200365751)

[4 Documento de implementación e implantación del sistema 4](#_Toc200365752)

[4.1 Implementación 4](#_Toc200365753)

[4.2 Pruebas 4](#_Toc200365754)

[5 Documento de cierre 4](#_Toc200365755)

[5.1 Documento de instalación y configuración. 4](#_Toc200365756)

[5.2 Manual de usuario 4](#_Toc200365757)

[5.3 Resultados obtenidos y conclusiones. 4](#_Toc200365758)

[5.4 Diario de bitácora 4](#_Toc200365759)

[6 Bibliografía. 4](#_Toc200365760)

[7 Anexos 4](#_Toc200365761)

# **1. Descripción del proyecto**

## 1.1 Contexto del proyecto

### 1.1.1 Ámbito y entorno

Este proyecto se basa en las artes marciales y la tecnología. La idea nace al ver que muchos torneos todavía se organizan de forma manual o con herramientas que no están pensadas para eso. El objetivo es modernizar y facilitar todo lo relacionado con la organización y participación en estos eventos, usando una app que funcione como red social y una parte web para los organizadores

### 1.1.2 Análisis de la realidad.

͏Hoy en día, muchos torneos de artes marciales no tienen una forma fija para controlar todo: las inscripciones, los combates, los participantes, los ͏resultados, etc. Se usa a menudo WhatsApp, hojas de Excel o papeles sueltos; algo que hace difícil la organización. También los peleadores no tienen͏ un lugar donde ͏ver p͏róximos even͏tos o͏ quien͏es m͏ás van a participar

### 1.1.3 Solución y justificación de la solución propuesta

Para solucionar esto, propongo una aplicación móvil llamada **FIGHT-ZONE**, que es parecida a una red social, pero centrada en torneos de artes marciales. Desde la app se pueden ver eventos, inscribirse, mirar los perfiles de otros peleadores y seguir las peleas. También hay una parte web pensada para los organizadores, donde pueden gestionar los participantes, hacer los cuadros de combate y llevar el control del torneo. Con esta herramienta, todo es más fácil, rápido y organizado

### 1.1.4 Destinatarios.

* **Peleadores**, que quieren participar en torneos y ver su progreso
* **Organizadores**, que buscan una forma más práctica de gestionar sus eventos
* **Entrenadores y academias**, que quieren seguir a sus alumnos
* **Aficionados**, que simplemente quieren ver torneos y estar al tanto de todo

## 1.2 Objetivo del proyecto

El objetivo del proyecto es crear una app móvil y una web para ayudar a organizar y seguir torneos de artes marciales de forma más sencilla. La idea es que sea útil tanto para los que participan como para los que organizan, y así crear una comunidad más conectada y moderna dentro de estos deportes

## 1.3 Objetivo del proyecto en lengua extranjera

Tenemos como objetivo traducir nuestro proyecto a inglés, si es bien recibida en España, para que todos los demás países puedan tener acceso a mi aplicación

# **2 Documento de Acuerdo del proyecto.**

## 2.1 Requisitos funcionales y no funcionales.

**Requisitos funcionales**

* Registro e inicio de sesión de usuarios
* Ver torneos disponibles y detalles de cada uno
* Inscribirse a un torneo desde la app
* Ver perfiles de otros peleadores
* Visualizar cuadros de combate y resultados
* Sistema de roles para controlar permisos
* Panel de administración para organizadores (solo en la parte web)
* Gestión de torneos, emparejamientos y resultados desde la web

**Requisitos no funcionales**

* Interfaz fácil de usar y moderna
* Compatible con Android
* Base de datos SQLServer
* Seguridad en los datos personales (autenticación, validación)
* Escalable, para poder crecer si la usa más gente

## 2.2 Tareas

* **Análisis y diseño del proyecto**
* **Conexión con la base de datos y lógica de negocio**
* **Configuración de la base de datos**
* **Desarrollo de la app (front-end móvil)**
* **Diseño de interfaces (app y web)**
* **Desarrollo del panel web para organizadores**
* **Pruebas funcionales y de usuario**
* **Documentación del proyecto**
* **Despliegue y presentación**

## 2.3 Metodología a seguir para la realización del proyecto.

Se seguirá una **metodología ágil**, tipo **SCRUM**, dividiendo el trabajo en **sprints semanales**. Al final de cada sprint se revisa lo desarrollado, se hacen pruebas y se ajustarán tareas para la siguiente semana. Esto ayuda a ir avanzando poco a poco, corrigiendo errores y mejorando el diseño según se desarrolla

## 2.4 Planificación temporal de tareas.

| **Semana** | **Tareas** |
| --- | --- |
| 1 | Análisis del proyecto, definición de requisitos |
| 2 | Diseño de la base de datos y primeras pantallas |
| 3 | Desarrollo de la API |
| 4 | Funcionalidad de inscripción y perfiles |
| 5 | Desarrollo de la app móvil (registro, login, ver torneos) |
| 6 | Desarrollo del panel web para organizadores |
| 7 | Gestión de torneos y cuadros de combate |
| 8 | Pruebas, ajustes y mejoras |
| 9 | Documentación y preparación de la entrega |

## 2.5 Presupuesto (gastos, ingresos, beneficio)

| **Concepto** | **Detalle** | **Coste aproximado** |
| --- | --- | --- |
| Dominio web | Para el acceso a la parte web del sistema | 10 €/año |
| Hosting | Alojamiento para la web y la base de datos | 30–50 €/año (según proveedor) |
| SQL Server | Uso de una instancia en la nube (Azure o VPS propio) | 0–20 €/mes (uso básico en Azure o local con versión gratuita) |
| Herramientas de desarrollo | Android Studio, Visual Studio Code, SQL Server Management Studio | 0 € (software gratuito) |

**Ingresos:**  
Ninguno, al tratarse de un proyecto académico sin uso comercial real por ahora

**Beneficio:**  
El objetivo no es económico, sino práctico. El beneficio principal es la adquisición de experiencia profesional y el aprendizaje de herramientas reales

## 2.6 Contrato/Pliego de condiciones

* El proyecto se realiza como parte del módulo de Proyecto de DAM
* El desarrollo es individual, sin empresa cliente real
* El producto debe cumplir con los requisitos definidos
* La app debe ser funcional y demostrable

## 2.7 Análisis de riesgos

| **Riesgo** | **Posible impacto** | **Solución** |
| --- | --- | --- |
| Problemas con la base de datos | Pérdida de datos | Copias de seguridad y pruebas continuas |
| Errores en la app o web | Fallos en la entrega | Fase de pruebas y revisión en cada sprint |
| Falta de tiempo | No completar el proyecto | Buena planificación y priorizar tareas clave |
| Dificultad técnica (por ejemplo, conexión app-web) | Retrasos | Buscar ayuda, documentación oficial |

# **3 Documento de análisis y diseño**

## 3.1 Modelado de datos.

El sistema utiliza una base de datos relacional implementada en SQL Server, con varias tablas principales que modelan la información necesaria para gestionar los torneos de artes marciales, los peleadores y sus combates. Las entidades y sus relaciones principales son:

* **Users**: almacena los datos de los usuarios registrados, incluyendo su rol (usuario normal, peleador, organizador o administrador)
* **Fighters**: tabla heredada de usuario que es peleador, con atributos específicos como categoría de peso, altura, alcance, y estadísticas de combates (victorias, derrotas, empates)
* **Tournaments**: almacena los torneos, con datos como nombre, fechas, tipo de deporte, organizador y lugar
* **Participantes**: relaciona usuarios inscritos a torneos, garantizando que un usuario no se inscriba varias veces al mismo torneo (restricción UNIQUE)
* **Fights**: representa cada pelea dentro de un torneo, con referencia a los dos peleadores, el estado de la pelea y el ganador si ya se conoce
* **FightResults**: guarda el resultado de cada pelea, incluyendo el método de victoria y duración

## 3.2 Análisis y diseño del sistema funcional

El sistema tiene dos módulos funcionales: la app móvil para usuarios, peleadores y organizadores, y la aplicación web exclusiva para organizadores

* **App móvil:** permite a los usuarios registrarse, ver torneos disponibles, inscribirse en ellos, consultar perfiles de peleadores y seguir el progreso de las peleas
* **Web para organizadores:** facilita la creación y gestión de torneos, asignación de peleas, seguimiento de resultados y actualización del estado de cada combate

La lógica del sistema está dividida entre el cliente (app móvil y web) y el servidor (backend con SQL Server). La app y la web consumen servicios para consultar y actualizar la base de datos, gestionando inscripciones, resultados y datos de usuarios

## 3.3 Análisis y diseño de la interfaz de usuario.

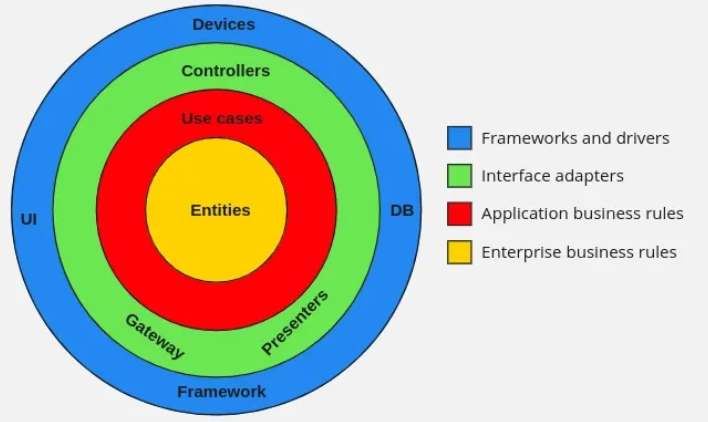
La interfaz de usuario está diseñada para ser intuitiva y accesible:

* En la app, las pantallas principales incluyen: listado de torneos, calendario, lista de peleadores y perfil
* La navegación es sencilla, basada en menús y pestañas para que el usuario pueda acceder rápidamente
* En la web, el organizador tiene un panel de control con acceso a la gestión de torneos, creación de combates, asignación de peleadores y actualización de resultados

## 3.4 Diseño de la arquitectura de la aplicación

El backend ha sido desarrollado usando la arquitectura de desarrollo Clean Architecture

**Clean Architecture** es un patrón de diseño de software que busca organizar el código de forma que sea fácil de mantener, probar y escalar. Su principal objetivo es separar las responsabilidades en capas, de modo que el núcleo de la aplicación (la lógica de negocio) esté aislado del exterior como de la interfaz de usuario o la base de datos.



### 3.4.1 Tecnologías/Herramientas usadas y descripción de las mismas

* **App móvil:** desarrollada en **Dart** usando **Flutter**
* **Web:** implementada con **Razor**
* **Base de datos:** **SQL** **Server**, alojada en un servidor propio o en la nube
* **Backend:** API REST que comunica la app y la web con la base de datos, gestionando la lógica de negocio y seguridad hecha en **.NET 8** con **C#**
* **Herramientas adicionales:** **Visual Studio** para desarrollo web y la API, **SQL Server Management Studio** para la gestión de base de datos y **Visual** **Studio** **Code** para la app

### 3.4.2 Arquitectura de componentes de la aplicación

La arquitectura sigue un modelo cliente-servidor:

* **Clientes:** app móvil (Android) y aplicación web, que consumen la API REST
* **Servidor:** API REST que procesa peticiones, aplica lógica de negocio y accede a la base de datos SQL Server
* **Base de datos:** almacena toda la información estructurada y garantiza integridad y seguridad

# **4 Documento de implementación e implantación del sistema**

## 4.1 Implementación

En esta fase se desarrolla el código fuente de la aplicación, implementando todas las funcionalidades definidas en el análisis y diseño.

Se crea tanto la aplicación móvil/web para usuarios y organizadores, como la base de datos en SQL Server.

## 4.2 Pruebas

Se realizan pruebas funcionales para verificar que todas las funcionalidades cumplen con los requisitos

También se realizan pruebas de usabilidad para garantizar que la interfaz sea intuitiva y fácil de usar

Se realizan pruebas de rendimiento para evaluar la capacidad de respuesta. Finalmente, se corrigen errores detectados y se repiten las pruebas hasta obtener un sistema estable y fiable para su salida

# **5 Documento de cierre**

## 5.1 Documento de instalación y configuración.

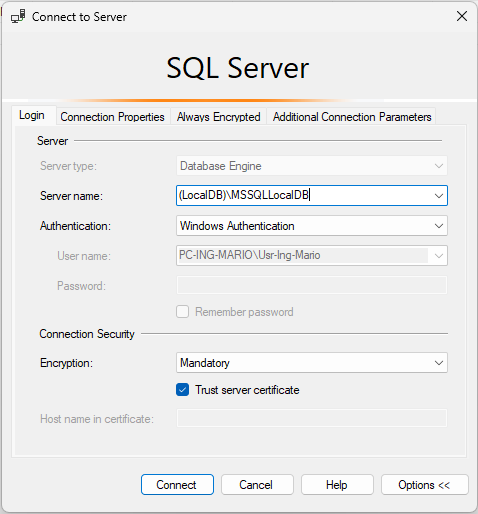
Paso1:

Bajarse el repositorio <https://github.com/MarioPerez125/TFC.git>  
Instalar las herramientas de estos enlaces:

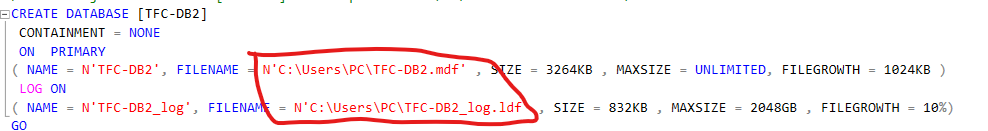
<https://go.microsoft.com/fwlink/?linkid=2313753>

<https://visualstudio.microsoft.com/es/thank-you-downloading-visual-studio/?sku=Community&channel=Release&version=VS2022&source=VSLandingPage&cid=2030&passive=false>

Paso2:

Ejecutar el sql server y configurarlo de esta manera para comunicarse con la local:

Abres el archivo TFC-Database.sql y tendrás que cambiar esta parte con la ruta de tu pc:

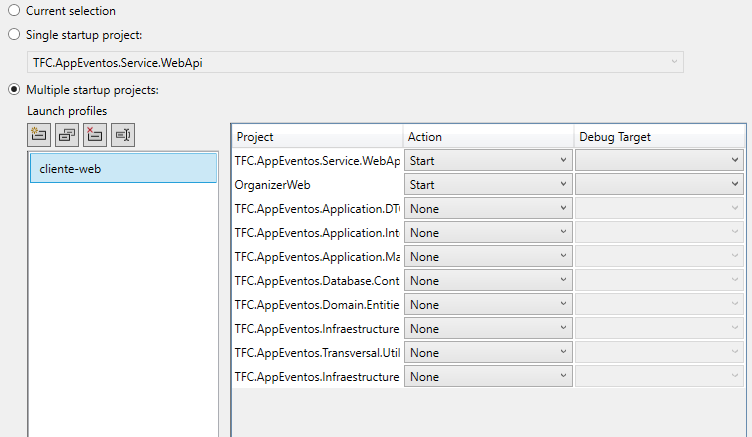
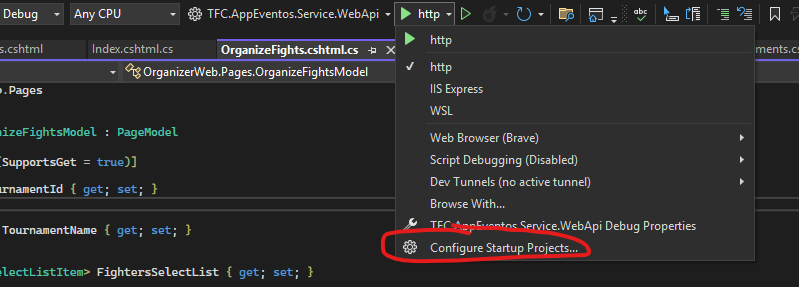


Hay que cambiar PC por tu usuario y ejecutar el comando

Paso3:

Abrir el tfc\TFC\TFC-AppEventos\TFC-AppEventos.sln con Visual Studio

Paso4:

Asignar el WebApi y la Web como proyectos de inicio:

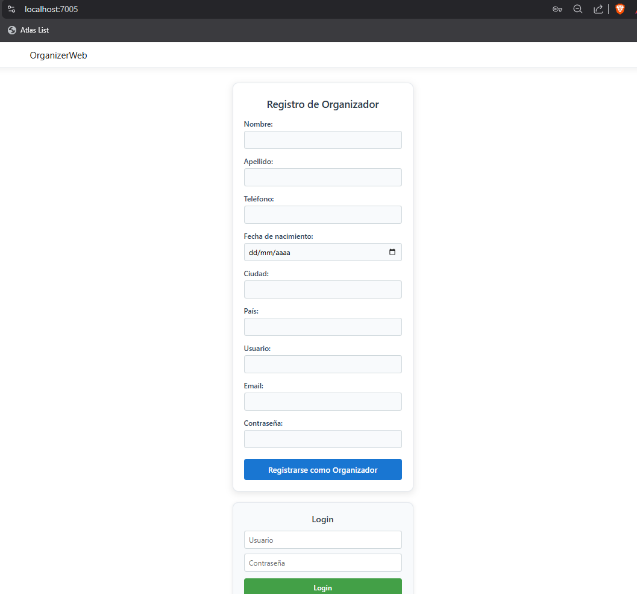
La api y la web ya estarán listas para usarse

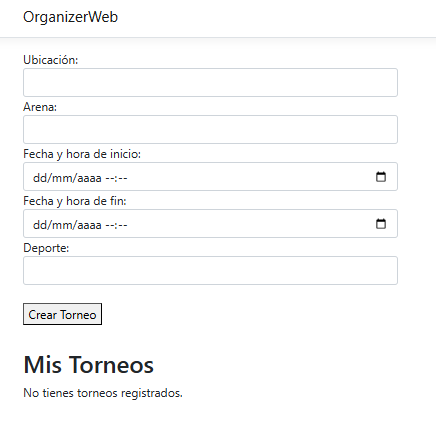
Paso5:

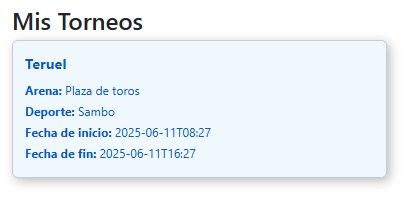
Ejecutar la app desde Flutter

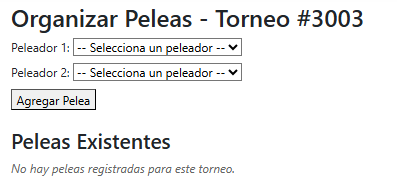
## 5.2 Manual de usuario

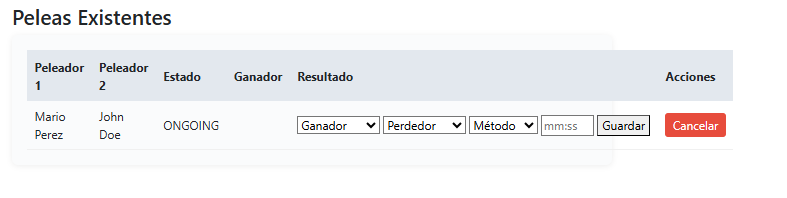
Web:

Ejecutarás el servicio y se te abrirá esta ventana:

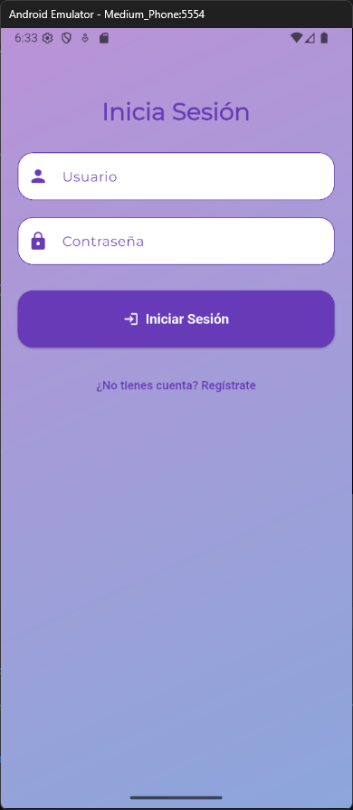
Aquí, te registrarás

En esta, tendrás que rellenar un formulario para crear un torneo, que mas adelante administrarás.

Dando click tendras el menú de organización de los torneos

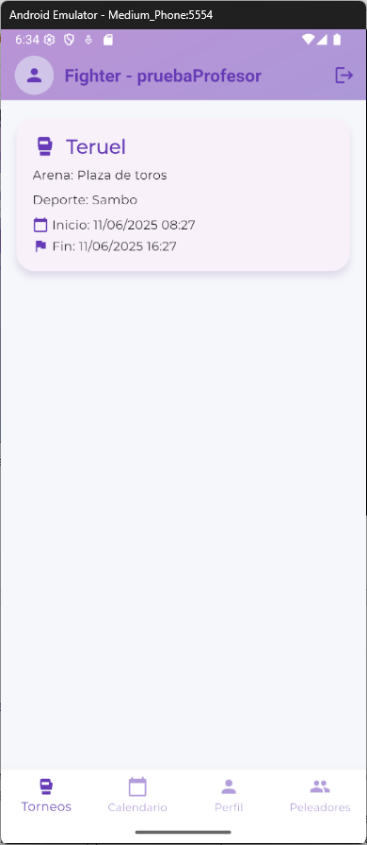
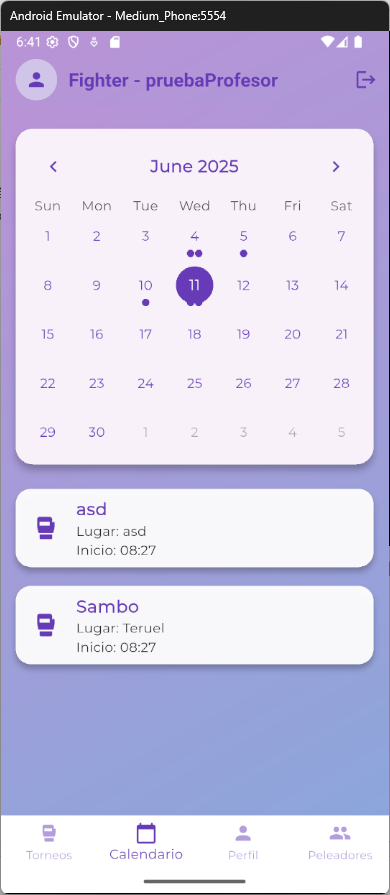
Los peleadores tienen que registrarse desde la app para que te aparezcan como peleadores.

Finalmente aquí puedes darle un resultado a la pelea

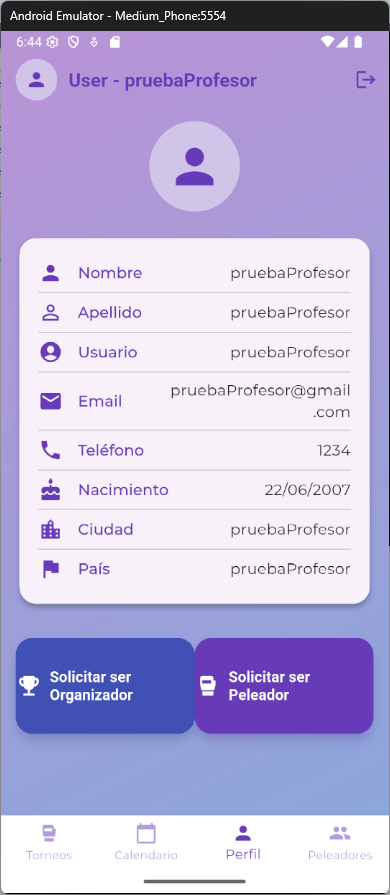
App:

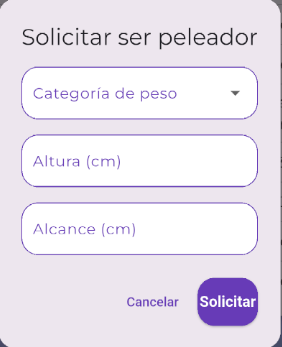
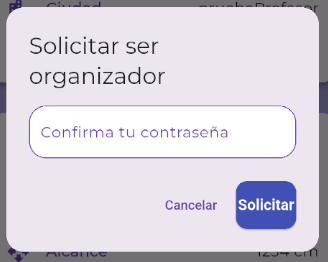
Al abrir la app te saldrá este menú, si no tienes cuenta te registras.

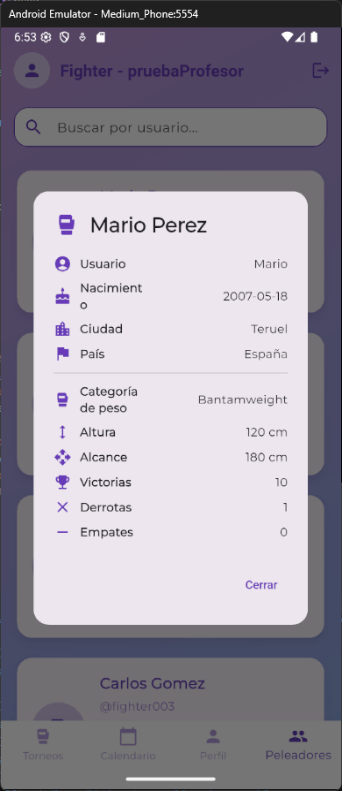
Después aparecerá esta ventana que es el scroll panel de los torneos disponibles en este momento, quiero decir, Los que la fecha no haya terminado aún, si los clicamos podemos ver las peleas que se están realizando o nos podemos apuntar si queremos:

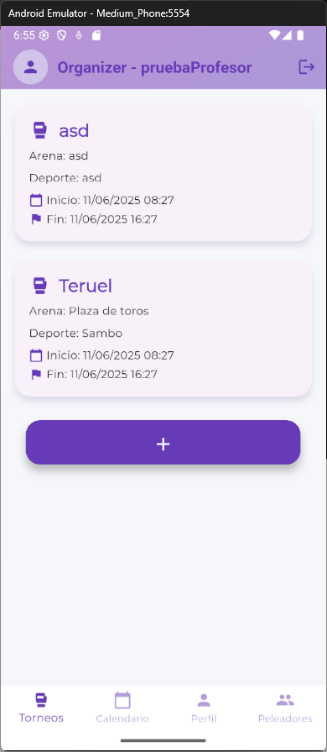
La segunda pestaña es un calendario con los torneos, en el que si clicamos en el torneo nos sale el mismo menú de antes:

La tercera pestaña es nuestro perfil, cuando nos creemos la cuenta arriba a la izquierda pondrá que somos user, lo que vendría siendo un aficionado, pero en la pestaña de perfil nos dejara registrarnos como organizador o como peleador

Si nos registramos como peleador, tendremos otra card debajo de los detalles de usuario, que tendrá los detalles de peleador, y podremos participar en peleas.

Para registrarse como peleador tendrás que rellenar un pequeño formulario con datos o también puedes elegir ser un organizador en el que te aparecerá simplemente para poner la contraseña

Luego Tenemos la ventana de Peleadores que nos muestra una lista de todos los peleadores registrado en la app y sus estadísticas

Por último, siendo organizador tienes acceso a crear torneos solamente, pero en un futuro probablemente a organizar las peleas también



## 5.3 Resultados obtenidos y conclusiones.

Tenemos una app funcional, rápida y estética que cumple todas las funciones que tiene que cumplir para poder organizar eventos de peleas.

Esta aplicación es muy escalable, ya que ahora mismo está preparada para poder poner cualquier deporte, lo cual a priori suena bien, pero complica demasiado las cosas, lo cual a futuro sería mejor dividir la app según el deporte que practiques.

Cada arte marcial tiene sus reglas, sus puntajes, su duración, sus categorías, su ranking Etc. lo cual hace imposible poder englobar todas en una sola aplicación, a futuro me gustaría añadir, a parte de dividirla por deportes, cosas como los asaltos, que repito no en todos los deportes de contacto existen, las tarjetas de los jueces, que son las puntuaciones que ha dado cada juez a la pelea, para luego sacar una decisión Etc.

Como conclusión quiero decir que es una app que cumple con su cometido pero que tiene una escalabilidad que prácticamente no tiene fin y siempre se le puede añadir alguna funcionalidad.

## 5.4 Diario de bitácora

Los primeros días estuve diseñando la estructura y las entidades básicas de la aplicación, a medida que avanzaba me iba topando con problemas principalmente de la base de datos, porque cuando modificas elementos en SQL Server, a veces el Visual Studio no los detecta, y estuve varios días que no funcionaba una relación.

Traté de implementar JWT ya que no era muy complicado, pero a la larga traía más problemas que beneficios, al final lo dejé sin usar, pero guardando el código por si posterior mente lo implementaba.

Después se me fueron ocurriendo ideas que hacer, que posteriormente desencadenarían más y más errores, tales como problemas para buscar nombres de usuario para la pagina web, problemas con los botones de la web, con la autentificación, con las entidades en la app, etc...

Haciendo pruebas con la página web, me di cuenta de que estaba llamando a los métodos de mi API directamente, lo cual hacía que no se pudiese separar del código de la API, y tuve que cambiarlo por llamadas a la API y tuve que rehacer varias cosas, poco a poco a medida que iba avanzando tuve muchos menos problemas, ya que principalmente los problemas se derivaban de guardar escasa información en mis clases, ya que solo guardaba lo justo que se guardaba en base de datos, y eso daba muchos problemas a la hora de implementar las interfaces, porque el tener que navegar por 3 clases distintas para sacar el nombre de un usuario es mas complicado que tenerlo simplemente guardado en un dto.

# **6 Bibliografía.**

<https://www.microsoft.com/es-es/sql-server>

<https://learn.microsoft.com/es-es/dotnet/csharp>

<https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-8.0>

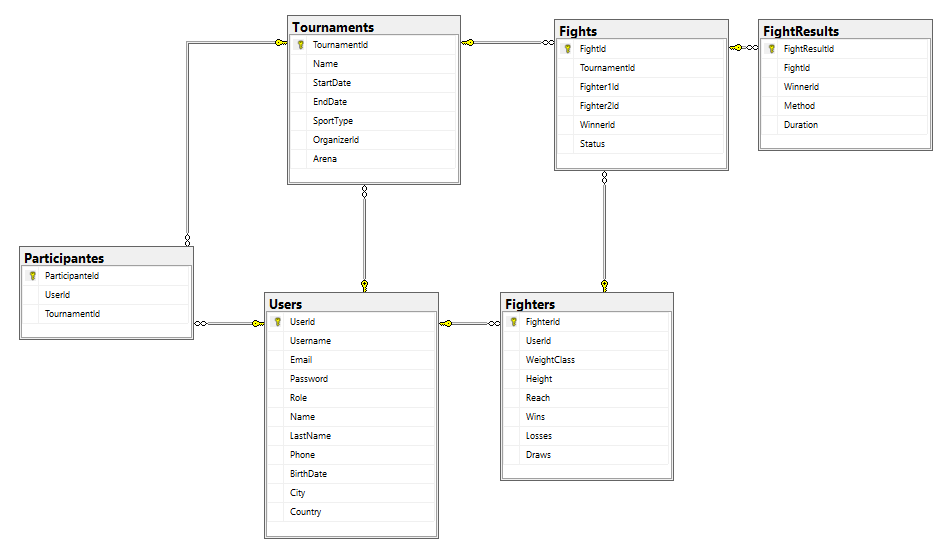
<https://docs.flutter.dev/>

<https://learn.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-9.0&tabs=visual-studio>

# **7 Anexos**

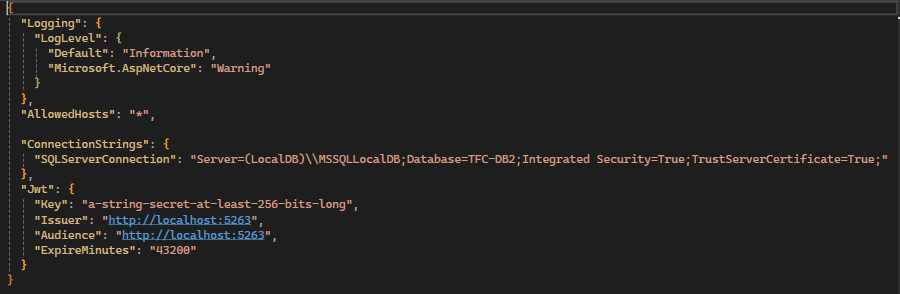
Anexo1:

Diagrama entidad relación



Anexo2:

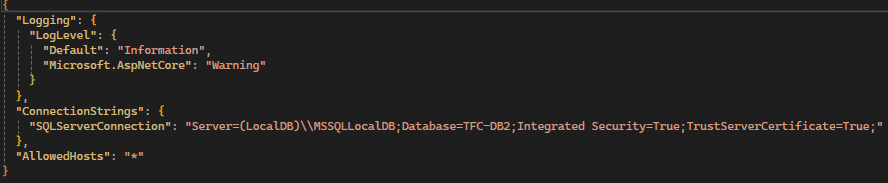
AppSettings y LaunchSettings de la API





Anexo3:

AppSettings y LaunchSettings de la Web





Anexo3:

Configuración Ruta Web para llamar a la API en Program.cs

