Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN DE INGENIERÍA DE SOFTWARE

WEB APP PARA LA GESTIÓN CAMPEONATOS DE PADEL

Alumno: Daniel Moñivas Pérez

Tutor: Joaquín Nicolás Adiego Rodríguez

A mi familia, que me ha apoyado en este camino.

Agradecimientos

A mi familia en primer lugar, que me ha apoyado en los momentos más complicados y gracias a ellos he podido seguir adelante durante estos años académicos.

A mis amigos y amigas, que en ellos he tenido siempre un apoyo y me han ayudado tanto en lo académico como en lo personal.

A mis compañeros de universidad, que han sido un gran apoyo en muchos momentos complicados durante estos años.

Gracias a todos.

Resumen

El objetivo de este proyecto es crear una página web para la creación y seguimiento de campeonatos y amistosos de pádel.

En esta página web se pretende que los usuarios puedan crear tanto torneos y amistosos, como la visualización de los torneo creados por otras personas, a los cuales pueden tanto apuntarse y realizar el seguimiento de estos torneos.

El desarrollo se ha llevado a cabo con el framework React js y CSS para la parte de front, y para la base de datos se ha utilizados Firebase.

Abstract

The objective of this project is to create a web page for the creation and monitoring of paddle tennis championships and friendlies.

On this web page it is intended that users can create both tournaments and friendly tournaments, as well as the visualization of tournaments created by other people, to which they can both sign up and monitor these tournaments.

The development has been carried out with the React js framework and CSS for the front part, and Firebase has been used for the database.

Índice general

Agradecimientos **III**

Resumen **V**

Abstract **VII**

Lista de figuras **XIII**

Lista de tablas **XV**

1. Introducción **1**

1.1. Introducción 1

1.2. Motivación 1

1.3. Objetivos de proyecto 2

1.3.1. Objetivos de desarrollo 2

1.3.2. Objetivos personales 2

1.4. Estructura de la memoria 3

2. Desarrollo y Planificación del Proyecto **5**

2.1. Scrum 5

2.1.1. Planificación con Scrum y definición 5

2.1.2. Equipo Scrum 6

2.1.3. Eventos Scrum 6
2.1.4. Artefactos Scrum 7
2.2. Proyecto adaptado a Scrum. 7
2.3. Historias de usuario 8
2.3.1. Sprint 0 10
2.3.2. Sprint 1 11
2.3.3. Sprint 2 12
2.3.4. Sprint 3 13
2.3.5. Sprint 4 14
2.3.6. Sprint 5 15
2.3.7. Sprint 6 16
2.3.8. Sprint 7 17
2.3.9. Sprint 8 18

3. Tecnologías utilizadas **21**

3.1. React.js 21
3.1.1. Ciclo de vida 23
3.1.2. Hooks 24
3.2. NPM 24
3.2.1. Router 24
3.2.2. React Bootstrap 24
3.3. Firebase 25
3.4. Git 25
3.4.1. Github 25
3.5. Trello 25
3.6. LaTeX 26
3.6.1. Overleaf 26
3.7. Astah 26

4. Análisis del proyecto **27**

4.1. Requisitos funcionales (RF) 27

4.2. Requisitos no funcionales (RNF) 27

4.3. Casos de uso 28

4.4. Modelo de dominio 29

4.5. Diagramas de secuencia 30

5. Diseño **37**

5.1. SaaS 37

5.1.1. Ventajas 37

5.1.2. Desventajas 38

5.2. BaaS 38

5.2.1. Ventajas 39

5.2.2. Desventajas 39

5.3. El patrón Model-View-ViewModel 39

5.3.1. Ventajas 40

5.3.2. Desventajas 40

5.4. Diseño guiado por componentes 40

5.4.1. Ventajas 41

5.4.2. Desventajas 41

5.4.3. Componentes creados en este proyecto 41

5.5. Algoritmo 44

5.5.1. Algoritmo Fuerza bruta 44

5.6. Interfaz 44

5.7. Base de Datos 52

5.7.1. Authentication 52

5.7.2. Firestore Database 53

6. Plan de riesgos y estimación de costes **55**

6.1. Plan de riesgos 55

6.1.1. Riesgos encontrados en este proyecto 56

6.2. Presupuesto 58

7. Implementación **59**

7.1. Estructura del código del proyecto 59

7.2. Decisiones a lo largo del proyecto 61

7.3. Cambios realizados a lo largo del proyecto. 61

8. Conclusiones y futuro **63**

8.1. Conclusiones 63

8.2. Trabajo futuro 63

A. Manual de usuario **65**

Bibliografía **67**

Índice de figuras

3.1. React DOM 22
3.2. Props 22
3.3. JSX 23
3.4. Componente React 23

4.1. CU Apuntarse a un fanatic 29
4.2. CU Eliminar un amistoso 30
4.3. Diagrama de casos de uso 34
4.4. Modelo de dominio 35
4.5. Secuencia Inicio Sesión 35
4.6. Secuencia Registro Amistosos 36
4.7. Secuencia Ver Clasificación 36

5.1. BaaS 38
5.2. MVVM 40
5.3. Diagrama de componentes 43
5.4. Emparejamientos 44
5.5. Captura Home 45
5.6. Captura Amistosos 46
5.7. Captura Fanatics 46

5.8. Captura Inicio Sesión 47
5.9. Captura Registrarse 47
5.10. Captura Crear Amistoso 48
5.11. Captura Crear Fanatic 48
5.12. Captura Mi Perfil 49
5.13. Captura Credenciales 49
5.14. Captura Mis Partidos 50
5.15. Captura Mis Amistosos 50
5.16. Captura Mis Fanatics 51
5.17. Captura Siguiente Partido 51
5.18. Captura Clasificación 52
5.19. Captura Firebase Authentication 53
5.20. Captura Firestore Database 53

Índice de cuadros

2.1. Planificación inicial de sprints 8
2.2. Historias de usuario 9
2.3. Tareas del sprint 0 11
2.4. Tareas del sprint 1 12
2.5. Tareas del sprint 2 13
2.6. Tareas del sprint 3 14
2.7. Tareas del sprint 4 15
2.8. Tareas del sprint 5 16
2.9. Tareas del sprint 6 17
2.10. Tareas del sprint 7 18
2.11. Tareas del sprint 8 19

4.1. Requisitos funcionales 32
4.2. Requisitos no funcionales 33

6.1. Riesgo de falta de formación 56
6.2. Riesgo de enfermedad 56
6.3. Riesgo de falta de tiempo 57
6.4. Riesgo de cambio de requisitos 57
6.5. Riesgo de limitaciones tecnológicas 57

6.6. Riesgo de averías 57

Capítulo 1

Introducción

1.1. Introducción

La organización de competiciones de muchos deportes puede ser una tarea bastante complicada y laboriosa, sobre todo cuando se trata de organizar estos campeonatos con gente que no se conoce entre sí, ya que muchas veces hay que tener a personas responsables para poder organizar todas estas tareas. Aunque ya existen algunas páginas webs en las que puedes apuntarte a partidos, suelen pertenecer a clubes concretos y no están muy extendidas, de ahí que haya surgido esta idea de crear una web para la organización de amistosos y de campeonatos Fanatic entre los usuarios registrados.

1.2. Motivación

La motivación que me ha llevado a desarrollar esta página web es aplicar los conocimientos que he adquirido durante estos años en el grado junto con un hobbie que practico en mi tiempo libre como es el pádel, y gracias a esto aprender nuevas tecnologías mientras aprendo como organizar torneos de un deporte que me gusta.

La idea de la creación de este proyecto, se establece porque después de echar un ver varias páginas webs cuyo propósito es la creación de torneos de pádel, me di cuenta de que cada una tenía sus contras, ya que habia alguna que tienes que registrarte antes de poder ver nada sobre la página [5], y otras que solo te dejan crear tú los torneos [4].

Por este motivo, la idea de este proyecto es crear una aplicación en la que el usuario sea capaz de ver partidos, tanto amistosos como en torneos, ya creados en los que poder apuntarse con gente que no tiene por qué ser conocida, y también poder crear y organizar estos partidos entre los diferentes usuarios que estén registrados en ella. De esta manera poder tener todos los pros de las diferentes páginas web que podemos encontrar, eliminando la parte negativa.

1.3. Objetivos de proyecto

1.3.1. Objetivos de desarrollo

El objetivo de este proyecto es crear una página web que sirva para organizar y apuntarse a partidos amistosos y torneos de pádel, creando estos partidos o apuntandote a partidos ya creados.

Los webparts principales de la página web serán:

- Barra Menú: en este webpart podemos volver al inicio, entrar en amistosos y fanatics, y entrar en el perfil del usuario o el inicio de sesión y el registro.
- Amistosos: aquí podremos ver los amistosos creados, y si hemos inicado sesión podremos crearlos y apuntarnos.
- Fanatics: componente es similar al de amistosos.
- Mi perfil: en este componente podemos iniciar sesión y registrarnos, y cuando inicias sesión puedes ver tu perfil y los partidos a los que te has apuntado.

Para conocer las características de cada partido guardamos datos como son:

- Localidad.
- Dirección.
- Fecha y hora.
- Duración.
- Participantes apuntados-Partidos del amistoso/torneo.
- Nivel (0-5).

1.3.2. Objetivos personales

El objetivo que tenía con la creación de este proyecto, es ampliar mis conocimientos en una materia que me gustó especialmente durante los años académicos como es el desarrollo web. Entre las habilidades que he aprendido y mejorado se encuentran:

- El framework React js.
- La mejora de CSS para los estilos usados con React.
- Conocer la utilidad de Firebase como base de datos y para guardar usuarios.
- Obtener mayor experiencia en cuanto a la licitación de requisitos y el diseño de las soluciones a los diferentes casos de uso.

1.4. Estructura de la memoria

La memoria va a tener las siguientes partes:

- En el **Capítulo 2** veremos la planificación inicial del proyecto y su desarrollo utilizando la metodología Scrum.
- En el **Capítulo 3** veremos en detalle las tecnologías y herramientas usadas para el desarrollo del proyecto.
- En el **Capítulo 4** realizaremos una estimación de costes y un plan de riesgos que pudiera suceder durante el desarrollo de la página web.
- En el **Capítulo 5** veremos como ha ido la planificación inicial del proyecto siguiendo el método Scrum.
- En los **Capítulos 6, 7 y 8** veremos el análisis, el diseño y la implementación de este proyecto.
- En el **Capítulo 9** veremos tras realizar el proyecto, las conclusiones y las futuras mejoras de la página.
- Finalmente veremos el **"Manual de usuario"** y el **"Manual de despliegue"**.

Capítulo 2

Desarrollo y Planificación del Proyecto

2.1. Scrum

2.1.1. Planificación con Scrum y definición

Scrum [22] es un marco que promueve y simplifica el trabajo colaborativo entre equipos. Su objetivo principal es reducir la complejidad para abordar un proyecto. Se divide en 3 pilares principales:

- **Transparencia:** es fundamental para que todos los miembros del equipo y los interesados de la organización tengan un entendimiento compartido y vean lo mismo con respecto a la entrega de valor y el progreso. Entender y ver lo mismo es importante para el trabajo en equipo y el alineamiento hacia objetivos compartidos.
- **Inspección:** es evaluar la información del avance de producto, de la adopción de producto y los "Outcomes" o impacto en las clientes producidas con el lanzamiento al mercado, de la calidad de producto, de la capacidad de trabajo en equipo, de la motivación del equipo, del logro de objetivos de negocio entre otros. La inspección se puede dar en el equipo de desarrollo, equipo de Scrum, interesados y la organización misma.
- **Adaptación:** es la reflexión a partir de la inspección para cambiar la dirección del plan. Esta adaptación puede ocurrir a nivel de equipo de Scrum, Product Owner, interesados y la organización misma. Esta adaptación puede implicar por ejemplo adaptar el Product Backlog, Sprint Backlog, la forma de trabajo, la forma de realizar las pruebas, las prácticas y herramientas, etc. La inspección y adaptación son continuas, así como la mejora de la transparencia.

2.1.2. Equipo Scrum

En cada equipo de Scrum, hay diferentes puestos y responsabilidades que cada persona de ese equipo debe cubrir, esos roles son:

- **Product Owner:** es el responsable de la cartera de productos, conocida como pila de producto o Product Backlog. Por esta razón, comprende las necesidades de los usuarios dentro del negocio. También se encarga de obtener el máximo valor al mínimo coste.
- **Scrum Master:** es la figura que lidera los equipos en la gestión ágil de proyectos. Su misión es que los equipos de trabajo alcancen sus objetivos, eliminando cualquier dificultad que puedan encontrar en el camino.
- **Equipo de desarrollo:** se encargan de desarrollar el producto, auto-organizándose y auto-gestionándose para conseguir entregar un incremento de software al final del ciclo de desarrollo. En este rol es en el que se encuentran los perfiles más técnicos, que se encargan únicamente del desarrollo.

2.1.3. Eventos Scrum

Scrum cuenta con cinco eventos para mantener los mínimos necesarios para facilitar el control empírico de procesos funciona y por tanto no dinamitar ninguno de los pilares fundamentales de Scrum.

- **Organización del backlog:** es una lista de trabajo ordenado por prioridades para el equipo de desarrollo que se obtiene de la hoja de ruta y sus requisitos. La organización y actualización de esta lista es responsabilidad del Product Owner.
- **Planificación Sprint:** en esta reunión, todo el equipo de desarrollo planifica el trabajo que se va a realizar (alcance) durante el sprint actual. Esta reunión la dirige el experto en scrum y, en ella, el equipo decide el objetivo del sprint. Tras esta reunión se añaden historias de usuario desde el backlog al sprint.
- **Sprint:** es el periodo en el que un equipo scrum trabaja para finalizar un incremento. La duración del sprint idealmente es de 2 semanas, pero depende del equipo y el proyecto. Todos los eventos (desde la planificación hasta la retrospectiva) tienen lugar durante el sprint.
- **Scrum diario o daily:** es una reunión diaria de corta duración(unos 15 minutos, aunque es variable). El objetivo del scrum diario es que todos los miembros del equipo estén en sintonía, se adecuen al objetivo del sprint y dispongan de un plan para las próximas 24 horas. Lo que cada miembro del equipo comenta en esta daily es:
  - tareas que se hicieron ayer tras la daily.
  - que tareas tengo para hoy.
  - problema u obstáculo que me impidan avanzar.

- **Revisión de sprint:** una vez finalizado el sprint, el equipo se reúne en una sesión informal para ver el evolutivo desarrollado durante el sprint. Se revisan todas las tareas del backlog que entraban en el sprint para comprobar que están finalizadas. Por último y tras revisar el backlog, el product owner es el que decide si se lanza el evolutivo.
- **Retrospectiva:** es una reunión que se realiza tras un sprint para analizar lo que ha funcionado y lo que no tras la finalización del sprint. La idea de esta reunión es que el equipo detecte los errores que se dieron en el anterior sprint para intentar mejorarlo, y ver que cosas se han hecho bien para continuar.

2.1.4. Artefactos Scrum

Los artefactos son aquellos elementos físicos que se producen como resultado de la aplicación de Scrum. Son los siguientes:

- **Backlog del producto (Product Backlog):** es la lista principal del trabajo que debe realizar el propietario del producto o el gestor de productos.Es una lista dinámica con requisitos, mejoras y correcciones que el Product Owner debe ir revisando continuamente. En resumen, se trata de una lista con las 'cosas por hacer' para la finalización del proyecto.
- **Backlog del Sprint (Sprint Backlog):** se trata de la lista de elementos, historias de usuario o correcciones de errores, seleccionadas por el equipo de desarrollo, para su implementación en el ciclo actual de sprint. En la reunión de planificación del sprint, se decidirá que tareas se sacan del backlog y se introducen en el sprint. Se puede realizar alguna modificación de las tareas iniciales durante el sprint, pero no se debe comprometer el objetivo inicial del sprint.
- **Incremento:** es el producto final que se ha conseguido tras finalizar un sprint. Esto incluye todas las treas, historias de usuario y casos de uso, que posteriormente se pondrá a disposición del usuario final. Esta metodología ágil se basa en realizar estos incrementos o evolutivos de manera iterativa e incremental, de ahí el nombre. Gracias a las iteraciones, nos aseguramos que el ciclo de vida del software(planificación, diseño, desarrollo, testeo y entrega) se hace en un tiempo de 4 semanas o menos.

En algunos proyecto podemos encontrar más artefactos como 'Definition of Done(DoD)', 'Definition of Ready(DoR)', o Burndown Chart, pero estos son menos importantes y no son necesarios en muchos proyectos.

2.2. Proyecto adaptado a Scrum.

Debido a que en este proyecto no puede haber todos los roles ya que al ser un trabajo de fin de grado tiene que ser individual, se ha tenido que adaptar los roles a las personas involucradas en dicho proyecto.

La organización de este equipo ante estas circunstancias será con el alumno formando el 'Equipo de Desarrollo' y también siendo el 'Product Owner', y el tutor será el encargado de ser el 'Scrum Master', que tendrá la tarea de asesorar en el cumplimiento de esta metodología.

Para el proyecto, hemos decidido que cada sprint sea de 2 semanas, finalizando cada sprint el jueves. Este día también vamos a realizar la 'Revisión del Sprint', la 'Retrospectiva' y la 'Planificación del Sprint' de las siguientes 2 semanas.

Para la gestión tanto del 'Sprint Backlog' como del 'Product Backlog', hemos utilizado la herramienta de 'Trello'. Dicha herramienta nos facilita el seguimiento de las tareas que realizamos en los sprints.

Este proyecto comenzará en Marzo de 2021 con el primer sprint. En este primer sprint vamos a ver la planificación, análisis y la estructura del proyecto, ya que tras tener claro los requisitos e historias de usuarios que vamos a tener que implementar, tenemos que ver como podemos llevar a cabo esas tareas con las tecnologías que vamos a utilizar, y en caso de algún problema volver a tratar de solucionarlo.

En el desarrollo de este proyecto, debido a la dificultad de compaginar el trabajo laboral con el trabajo de fin de grado, finalmente la presentación del TFG se realiza un año después de los previsto, por lo que veremos diferencias entre los sprints planeados y los sprints que hemos hecho finalmente.

En la siguiente tabla vemos la planificación inicial de los sprints del proyecto:

Sprint	**Inicio**	**Fin**	**Comentarios**
Sprint 0	25/02/2021	11/03/2021	
Sprint 1	11/03/2021	25/03/2021	
Sprint 2	25/03/2021	08/04/2021	
Sprint 3	08/04/2021	22/04/2021	
Sprint 4	22/04/2021	06/05/2021	
Sprint 5	06/05/2021	20/05/2021	
Sprint 6	20/05/2021	03/06/2021	
Sprint 7	03/06/2021	17/06/2021	
Sprint 8	17/06/2021	01/07/2021	
Sprint 9	05/07/2021	19/07/2021	Sprint de refuerzo. Opcional

Cuadro 2.1: Planificación inicial de sprints

2.3. Historias de usuario

En esta tabla, vamos a enumerar todas las historias de usuarios que vamos a implementar en el proyecto.

ID	Título	Descripción
1	Crear Amistosos	Como usuario quiero crear partidos amistosos.
2	Crear Fanatics	Como usuario quiero crear torneos fanatic.
3	Ver Amistosos	Como usuario quiero ver todos los amistosos creados.
4	Ver Fanatics	Como usuario quiero ver todos los fanatics creados.
5	Apuntarme Amistosos	Como usuario quiero poder apuntarme a los amistosos creados.
6	Apuntarme Fanatic	Como usuario quiero poder apuntarme a los Fanatics creados.
7	Registrarme	Como usuario quiero poder registrarme como usuario en la página.
8	Iniciar sesión	Como usuario poder iniciar sesión con mi usuario.
9	Ver mis datos	Como usuario quiero poder ver mis datos.
10	Editar mi perfil	Como usuario quiero poder actualizar mis datos como usuario.
11	Eliminar perfil	Como usuario quiero poder eliminar mi cuenta.
12	Ver mis amistosos	Como usuario quiero poder ver los amistosos a los que estoy apuntado.
13	Ver mis fanatic	Como usuario quiero ver los fanatics a los que estoy apuntado.
14	Ver clasificación	Como usuario poder ver como va la clasificación de los fanatic.
15	Sugerir partidos	Como usuario quiero que la aplicación me sugiera contra quién jugar el siguiente partido.
16	Eliminar amistosos	Como administrador quiero poder eliminar amistosos.
17	Eliminar fanatics	Como admininistrador quiero poder eliminar fanatics.

Cuadro 2.2: Historias de usuario

Para la realización de todos los sprints, vamos a utilizar unos códigos para indicar de que tipo es cada tarea que vamos a realizar en el sprint:

- **DOC**: son las tareas que tiene que ver con toda la parte de documentación.
- **DES**: son las tareas tienen que ver con el desarrollo de la aplicación.
- **ERR**: son las tareas que tienen que ver con la búsqueda y la resolución de algún tipo de error que tenga la página.
- **EVOL**: son las tareas tienen que ver con la mejora o la evolución de alguna de las funcionalidades de la aplicación.

2.3.1. Sprint 0

Este sprint inicial, le vamos a dedicar para poder tener preparada la documentación inicial de la aplicación, y también el diseño lógico.

En este sprint vamos a ver tareas como definir todas las historias de usuario, el documento inicial, que incluye toda la parte de planificación y también hay una parte de investigar sobre la tecnología que vamos a utilizar para desarrollar el proyecto, ya que necesitamos una base para ver si podemos cumplir todos los objetivos iniciales.

En este sprint hemos invertido un total de 33 horas, y hemos dejado como incompletas las tareas de Redactar la introducción, que acabaremos en el siguiente sprint, y también la formación de React js y de firebase, ya que no me dió tiempo a acabarla en este sprint y se pasa al siguiente.

Debido a que va a haber tareas que no vamos a poder completar, las que no podamos las pasamos al siguiente sprint, dejandolas en este sprint como tareas incompletas. Esto lo haremos en todos los sprints en los que no lleguemos a tiempo para entregar la tarea.

Tarea	Tipo	Descripción	Tiempo	Estado
T - 001	DOC	Búsqueda de páginas web similares	1h	Completada
T - 002	DOC	Investigación funcionamiento Fanatics	1h	Completada
T - 003	DOC	Definición inicial de historias de usuario	3h	Completada
T - 004	DOC	Redactar la Introducción	3h	Incompleta
T - 005	DOC	Diseño inicial del proyecto	5h	Completada
T - 006	DOC	Creación repositorio GitLab	1h	Completada
T - 007	DES	Investigación firebase	7h	Incompleta
T - 008	DES	Aprendizaje React	12h	Incompleta
Total			**33h**	

Cuadro 2.3: Tareas del sprint 0

2.3.2. Sprint 1

Este sprint comienza el día 11 de Marzo. Empezaremos por finalizar las tareas pendientes del anterior sprint, y después nos pondremos con las tareas del nuevo sprint.

En este sprint lo que haremos será definir la metodología que vamos a utilizar, que va a ser Scrum, y vamos a empezar a documentar este capítulo, explicando en que consiste el método Scrum y como tiene que ser el proyecto para adecuarse a esta metodología.

También vamos empezar con la parte de desarrollo, implementando partes iniciales de la página web, y los componentes más sencillos para ir aprendiendo de manera práctica como trabajar con React, aún sin guardar nada en firebase. Además, vamos a empezar a hacer bocetos de algunos de los componentes de la página.

Estos componentes van a ser la vista de la home, con un componentes para hacer el fondo y otro componente que va a ser la barra del menú.

En este sprint hemos dedicado un total de 31h, sobre todo empleando el esfuerzo en los primeros componentes de la página.

Tarea	Tipo	Descripción	Tiempo	Estado
T - 009	DOC	Redactar la Introducción	2h	Completada
T - 010	DES	Realizar bocetos	4h	Incompleta
T - 011	DOC	Aprendizaje React	6h	Incompleta
T - 012	DES	Creación Componente Barra Menú	6h	Incompleta
T - 013	DES	Creación vista Home	7h	Completada
T - 014	DOC	Añadir tecnologías usadas a la documentación	1h	Completada
T - 015	DES	Aprendizaje estilos React-Bootstrap	5h	Incompleta
Total			**31h**	

Cuadro 2.4: Tareas del sprint 1

2.3.3. Sprint 2

En este sprint, vamos a intentar resolver todas las tareas que hemos dejado como incompletas en los anteriores sprints, para que no vayamos acumulando tareas pendientes y no nos de problemas alguna tarea incompleta a la hora de empezar otra.

Las tareas nuevas que vamos a llevar a cabo están relacionadas con dejar el componente de la Barra Menú completo.
Esto requiere que tengamos en la parte izquierda del componente, 3 botones, uno que nos devuelva a la Home, otro que nos lleve a los Amistosos, y el último que nos llevaría a los fanatics.
En la parte de la derecha, tendríamos una modal, ya que hay que diferencia si has iniciado sesión. En caso de que no hayas iniciado sesión, debe haber 2 botones, uno que lleve a iniciar sesión y otro que lleve a una página para registrarse.

También vamos a aplicar formato a un texto y un gif de la página home, para dejar esta visto como completada.

En este sprint, también desarrollamos la posibilidad de iniciar sesión con firebase, aunque aún no hayamos desarrollado la parte de Registro y tengamos que crear el usuario desde Firebase para las pruebas.

Este sprint nos ha llevado 32 horas de trabajo hasta el día del comienzo del tercer sprint.

Tarea	Tipo	Descripción	Tiempo	Estado
T - 016	DOC	Redactar la Introducción	1h	Completada
T - 017	DES	Investigar firebase	3h	Completada
T - 018	DES	Aprendizaje React	6h	Completada
T - 019	DES	Creación componente Barra Menú	6h	Completada
T - 020	DOC	Aprendizaje estilos React-Bootstrap	3h	Incompleta
T - 021	DOC	Creación bocetos	5h	Incompleta
T - 022	DOC	Componente inicio sesión	8h	Incompleta
Total			**32h**	

Cuadro 2.5: Tareas del sprint 2

2.3.4. Sprint 3

En este sprint, vamos a acabar el componente de iniciar sesión, ya que lo creamos para que fuera funcional y poder probar así el registro con firebase, pero ahora tenemos que darle estilos para que el componente creado sea coherente con el resto de la página.

Ahora que ya hemos probado a iniciar sesión con firebase tras crear una cuenta desde firebase, ahora vamos a desarrollar el componente de Registrarse, para que un usuario pueda crearse una cuenta y así probar tanto el modal del componente de Barra Menú como el Inicio Sesión, y también ver como guardar los datos en Firestore Database.

En este apartado, también vamos a redactar los sprints anteriores, y vamos a acabar los bocetos, que era lo que nos quedaba pendiente del anterior sprint.

Tarea	Tipo	Descripción	Tiempo	Estado
T - 023	DES	Componente inicio sesión	3h	Completada
T - 024	DES	Aprendizaje estilos React-Bootstrap	2h	Completada
T - 025	ERR	Pruebas para Registro e Inicio Sesión	4h	Completada
T - 026	DES	Desarrollo componente Registrarse	12h	Completada
T - 027	EVOL	Modificación Dropdown Barra Menú	5h	Incompleta
T - 028	ERR	Resolución errores inicio sesión	5h	Incompleta
T - 029	DOC	Redactar sprint anteriores	2,5	Completada
Total			**33'5h**	

Cuadro 2.6: Tareas del sprint 3

2.3.5. Sprint 4

En este sprint, vamos a completar las tareas que nos habíamos podido completar en el anterior sprint, y vamos a redactar la parte de la memoria en la que vemos Scrum y su adecuación al proyecto.

También vamos a añadir una parte de control de errores, que servirá para que no podamos registrar un usuario que ya existe, que no podamos iniciar sesión con un usuario incorrecto y controlar esos escenarios incorrectos. Esta tarea la marcaremos como completada, pero la vamos a tener que implementar mas veces para otros componentes.

Además, vamos a añadir en el dropdown que aparece tras iniciar sesión, para que podamos ver el perfil, cambiar credenciales y ver Mis Partidos, aunque quede solo planteado.

En este sprint, sobre todo nos vamos a centrar en la creación y desarrollo del componente de amistosos, que aunque no nos de tiempo a acabarle debido a su complejidad, queremos tener por lo menos la parte de la interfaz y crear el amistos en firebase de manera correcta. Si podemos, plantearemos la parte de apuntarse a estos amistosos para usuarios que estén registrados, y el botón para poder crear un amistoso, aunque no lo implementemos.

Tras la finalización de este sprint, dejé de realizar el TFG ya que encontré trabajo y me centré en eso, por eso volví a ponerme a realizar el TFG el **2 de Mayo**, y a partir de ahí seguí el mismo proceso en el desarrollo, con sprints de 2 semanas, pero en esta ocasión empezando los lunes.

Este sprint nos ha llevado 31 horas desarrollarle.

Tarea	Tipo	Descripción	Tiempo	Estado
T - 030	ERR	Resolución errores inicio sesión	3h	Completada
T - 031	EVOL	Modificación Dropdown Barra Menú	3h	Completada
T - 032	DOC	Sprint 4 y sección de scrum	3h	Completada
T - 033	EVOL	Control Errores	3h	Completada
T - 034	DES	Componente Amistosos	16h	Incompleta
T - 035	ERR	Creación Amistoso en Firebase	3h	Incompleta
Total			**31h**	

Cuadro 2.7: Tareas del sprint 4

2.3.6. Sprint 5

En este sprint, vamos a finalizar las tareas del componente de amistosos, en el que el planteamiento es crear un Carousel en el que podamos ver todos los partidos amistosos creados que hay, y en caso de haber iniciado sesión, también nos de la opción de apuntarnos al partido y de crear un amistoso.

También vamos a implementar todas las prueebas para algunass excepciones que serían, por ejemplo, apuntarse 2 veces al mismo partido.

Además, en este sprint vamos a empezar con la parte de la interfaaz de Mi Cuenta, que sería el Dropdown que nos muestra tras iniciar sesión, y que tendría 3 opciones, la de Mi Perfil en la que veremos nuestros datos, la de Mis Credenciales para modificarlas, y la parte de mis Partidos en la que podremos ver los amistosos y los fanatics.

También la idea es poder empezar con la funcionalidad de Editar el Perfil y cambiar las credenciales, aunque no nos de tiempo y lo acabemos en el siguiente sprint.

En este sprint, el tiempo invertido ha sido de 36 horas.

Tarea	Tipo	Descripción	Tiempo	Estado
T - 036	DES	Componente Amistosos	3h	Completada
T - 037	ERR	Creación Amistoso en Firebase	5h	Completada
T - 038	DES	Desapuntarse amistoso	12h	Completada
T - 039	DES	Apuntarse amistoso	8h	Completada
T - 040	DES	Vista Mi Perfil	4h	Incompleta
T - 041	DES	Vista Mis Credenciales	3	Incompleta
T - 042	DES	Vista Mis Partidos	1h	Incompleta
Total			**36h**	

Cuadro 2.8: Tareas del sprint 5

2.3.7. Sprint 6

En este sprint, vamos a intentar hacer las tareas que quedaron como pendientes en el anterior sprint.

Vamos a implementar la funcionalidad de Editar Mis Perfil, ya que solo habiamos creado la vista. Esa funcionalidad va a ser la implementación de 2 botones, uno para borrar el usuario, y otro para actualizar los datos. Antes de aceptar, nos saltará un pop-up pidiendo la confirmación antes de borrar o actualizar.

Tambiém vamos a implementar lo mismo en la pestaña de Mis Credenciales, en la que solo estará el botón de actualizar credenciales y cambiará el registro de Firebase Auth.

La parte más compleja de este sprint, va a ser que en la vista Patidos, nos aparezca un Carousel con todos los Amistosos(añadiremos los fanatics) a los que estamos apuntados.

Por último y si conseguimos acabar a tiempo, comenzaremos a ver como plantear la parte de los Fanatics, ya que necesitaremos ver como implementar el algoritmo para sugerir los partidos.

Este sprint tendrá una duración de 28 horas.

Tarea	Tipo	Descripción	Tiempo	Estado
T - 043	DOC	Sprint 5 y 6	1,5h	Completada
T - 044	DES	Vista Mi Perfil	1h	Completada
T - 045	DES	Vista Mis Credenciales	0,5h	Completada
T - 046	DES	Vista Mis Partidos	4h	Completada
T - 047	DES	Editar Mi Perfil	6h	Completada
T - 048	DES	Borrar Perfil	2h	Completada
T - 049	DES	Actualizar Credenciales	3h	Completada
T - 050	DES	Vista Fanatics	7h	Incompleta
T - 051	DES	Búsqueda del Algoritmo para la creación	3h	Incompleta
Total			**28h**	

Cuadro 2.9: Tareas del sprint 6

2.3.8. Sprint 7

En este sprint, que es uno de los finales, nos vamos a centrar en finalizar realizar el algoritmo para la búsqueda de rivales, en crear la pestaña de Fanatics de manera similar a como la creamos en Amistosos.

También vamos a crear en cada Fanatics (que podemos verlos en la pestaña de Mis Partidos del Dropdown), una clasificación que nos muestre los puntos de cada usuario en cada Fanatic.
En cada fanatic al que estemos apuntado, también podremos ir al siguiente partido, en el que tendremos 4 dropdown para seleccionar los oponentes si queremos hacerlo a mano, o tendremos un botón que nos va a sugerir los rivales que deberiamos seleccionar.

También vamos a implementar un rol de Administrador, el cual nos va a servir para que esos usuarios, puedan eliminar desde la interfaz los partidos que estén finalizados o que hayan sido anulados.

Este sprint nos va a servir para finalizar la fase de desarrollo e implementación, ahora nos queda la comprobación de errores y redactar lo que nos falta de la memoria.

Este sprint nos ha llevamos 37 horas para implementarlo.

Tarea	Tipo	Descripción	Tiempo	Estado
T - 052	DOC	Redactar Sprint 7	1h	Completada
T - 053	DES	Búsqueda del Algoritmo para la creación	12h	Completada
T - 054	DES	Vista Fanatics	4h	Completada
T - 055	EVOL	Rol Administrador	3h	Completada
T - 056	DES	Implementar Mis Fanatics	6h	Completada
T - 057	EVOL	Modificar componente Mis Partidos	7h	Completada
T - 058	DOC	Redactar las Tecnologías utilizadas	4h	Incompleta
Total			**37h**	

Cuadro 2.10: Tareas del sprint 7

2.3.9. Sprint 8

En este último sprint, nos vamos a encargar de realizar pruebas sobre la aplicación para intentar encontrar algún fallo de la misma. En caso de encontrar algún bug lo añadiremos como tarea al sprint y lo arreglaremos.

En lo que más nos vamos a centrar en este último sprint es en realizar lo que nos falta de memoria, ya que aún nos falta por implementar varias secciones como las Tecnologías utilizadas, que es la que dejamos pendiente del anterior sprint, y otras secciones que no hemos redactado como el plan de riesgos y costes, y la parte de análisis, que aunque ya habíamos realizado los diagramas anteriormente y lo había estado analizando, no lo hemos añadido en la memoria.

Por último también añadiremos el manual de usuario, para indicar En esta interacción se completó primeramente algunas de las tareas del anterior *Sprint*. Estas tareas se basaban en la formalización de las regex y la creación de una de sus etapas.

Después se analizó marzo, el cuál contenía muchos documentos (163). Con ello, las mejoras de cada expresión regular perteneciente a cada etapa se realizaron correctamente. No obstante, hubo error al mejorar la expresión regular de la etapa abierto, ya que ésta era la más complicada.

Al analizar el mes de abril, las expresiones regulares se mejoraron correctamente a excepción de la etapa abierto. En dicho mes, las expresiones regulares empezaron a dar sus frutos y cada vez eran menos documentos los que se tenía que analizar.

Por último, a partir de este *Sprint* no se pudo dedicar todo el tiempo que necesitaba el proyecto, ya que el alumno tenía que realizar otras tareas relacionadas con la universidad (prácticas de empresa). Por lo que se propuso al tutor en una reunión la nueva fecha de entrega.

En esta iteracción se han consumido 44 horas como se puede comprobar en la tabla **??**

Tarea	Tipo	Descripción	Tiempo	Estado
T - 059	DOC	Redactar Tecnologías utilizadas	3h	Completada
T - 060	ERR	Búsqueda de bugs en la aplicación	8h	Completada
T - 061	ERR	Correción de literales	1h	Completada
T - 062	DOC	Redactar Plan de riesgos	5h	Completada
T - 063	DOC	Redactar Análisis del proyecto	5h	Completa
T - 064	DOC	Redactar Implementación	3h	Completada
T - 065	DOC	Redactar Conclusiones	1h	Completada
T - 066	DOC	Redactar Manual de usuario	3h	Completada
Total			**29h**	

Cuadro 2.11: Tareas del sprint 8

Capítulo 3

Tecnologías utilizadas

3.1. React.js

React.js [20] es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Ahora mismo, esta biblioteca es mantenida por Facebook y tiene más de mil desarrolladores libres. Se creó en 2010, pero no fue hasta 2014 que se popularizó su uso.
Actualmente el está en un proceso de mejora continua, en el que se van sacando nuevas versiones periódicamente.

React tiene funcionalidades y características que vamos a ver a continuación:

- **Modularidad:** al igual que otros framework similares, el uso de los componentes en react supone una gran ventaja, ya que permite desarrollar una aplicación grande creando muchos componentes independientes, lo que permite encontrar fácilmente errores y desarrollar varios componentes en paralelo sin problema.
- **Estados y reactividad:** los estados de react nos permiten realizar muchas acciones, como por ejemplo volver a renderizar un componente tras el cambio de un estado sin necesidad de ninguna acción del usuario .
- **Componentes:** React nos permite crear una aplicación juntando componentes qee realizan una función en concreto y son totalmente independientes, en vistas, desde las cuales llamamos a los componentes, obteniendo así el diseño de una página.
- **Virtual Dom:** El Virtual DOM es una representación en memoria del DOM real que actúa de intermediario entre el estado de la aplicación y el DOM de la interfaz gráfica que está viendo el usuario.[8] Cuando un elemento de abajo se cambiar, provoca que se vuelvan a renderizar los componentes padres, pero no se renderizan los hermanos, lo que permite una mejor eficiencia. [9]

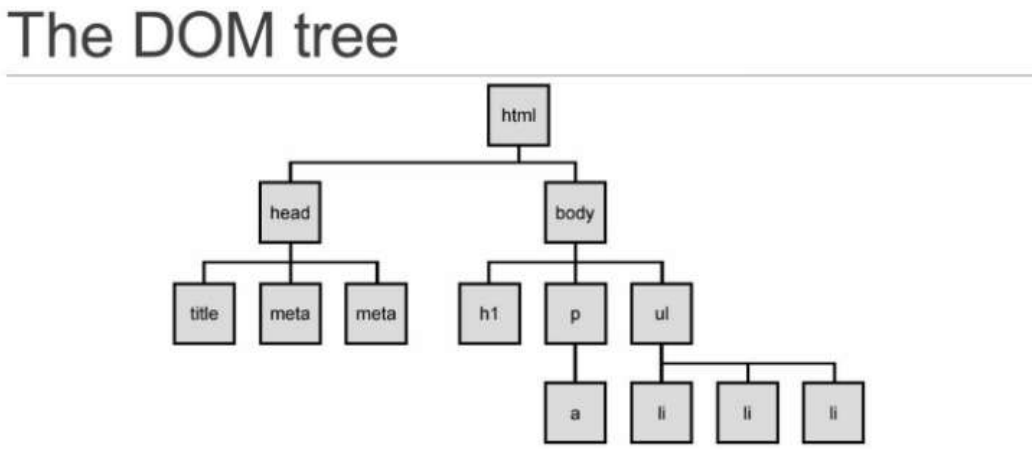

Figura 3.1: React DOM

- **Render Props:** Es un técnica para compartir código entre componentes en React utilizando una propiedad cuyo valor es una función. [18]

```
<DataProvider render={data => (
  <h1>Hello {data.target}</h1>
)}/>
```

Figura 3.2: Props

- **Eventos:** para los eventos y las transiciones, la biblioteca que más vamos a utilizar en este proyecto ofrece una gran variedad de posibilidades. [19]

- **JSX:** es una extensión de React para estructurar los componentes.Tiene una sintaxis parecida a HTML.

```
export default class Company extends React.Component<Props, {}>{
    public render() {
        return (
            <h1>
                {this.props.name}
            </h1>
        );
    }
}
```

Figura 3.3: JSX

3.1.1. Ciclo de vida

Las aplicaciones desarrolladas con React, están formadas por componentes. Estos componentes tienen distintos estados desde que se crean, hasta que el componente se destruye. En esta imagen vamos a ver los estados por los que pasa y vamos a explicar los más importantes.

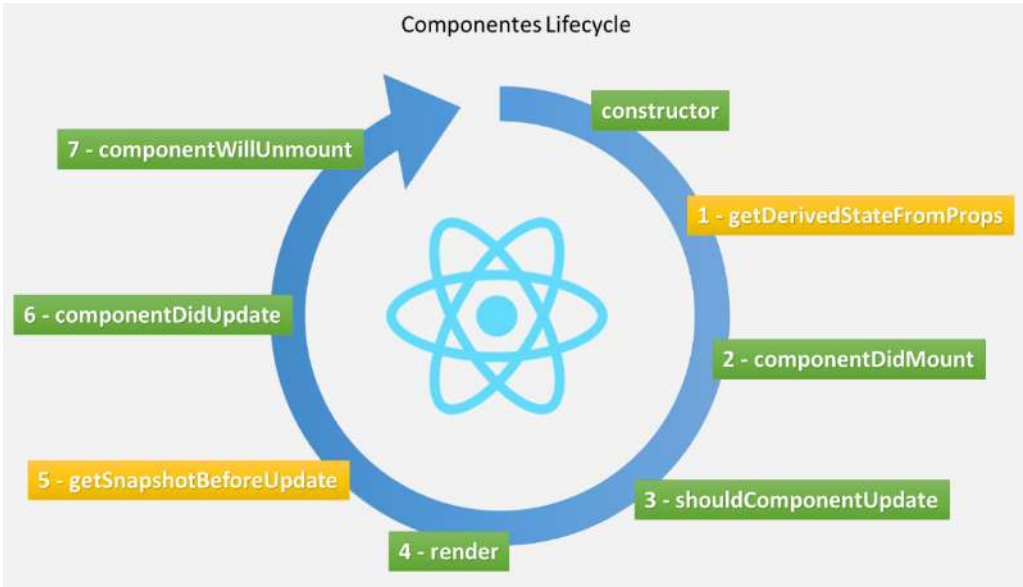

Figura 3.4: Componente React

- **shouldComponentUpdate:** permite al desarrollador prevenir el re-renderizado innecesario de un componente, devolviendo falso si no es necesario.
- **componentDidMount:** es llamado una vez que el componente se monta (el componente se crea en la interfaz). Se utiliza cuando se va a cargar datos desde una API.

- **componentWillUnmount:** es llamado inmediatamente antes de que el componente es "desmontado". Se utiliza para eliminar alguna dependencia.

- **render:** este método es llamado cada vez que se necesita actualizar el componente. Es el único requerido por todos los componentes.

3.1.2. Hooks

Los hooks son los nuevos estado de React incorporados en la versión 16.8. Permiten usar el estado sin tener que escribir un componente. Los 2 hooks más usados son:

- **useState:** contiene un par de valores: el valor del estado, y una función para cambiar su valor. Puedes llamar a esta función desde un evento. Es similar a this.setState en una clase.

- **useEffect:** se ejecuta cada vez que se renderiza el componente. Tiene un comportamiento similar a componentDidMount y componentDidUpdate.

3.2. NPM

Node Packager Manager (NPM) es un gestor de paquetes desarrollado en su totalidad bajo el lenguaje JavaScript, a través del cual podemos obtener cualquier librería con tan solo una sencilla línea de código, lo cual nos permitirá agregar dependencias de forma simple, distribuir paquetes y administrar eficazmente tanto los módulos como el proyecto a desarrollar en general. [17]

3.2.1. Router

Es un plugin que nos permite cambiar la navegación de la página entre las distintas vistas y componentes. Tenemos también varios componentes que podemos utilizar dependiendo de las características de la página. [21]

3.2.2. React Bootstrap

Vuetify [19] es un framework de componentes. Nos permite utilizar componentes ya creados, lo que facilita el desarrollo de aplicaciones complejas, ofreciendo una gran variedad de componentes y posibilidades. Es una de las librerías de componentes máas utilizadas.

3.3. Firebase

Firebase básicamente es una plataforma móvil diseñada y creada por Google, teniendo como principal función desarrollar y facilitar la creación de aplicaciones para dispositivos móviles que cuenten con una alta calidad a pesar de su rápida elaboración. Firebase tiene muchas funcionalidades.

A continuación vamos a ver las funcionalidades que hemos utilizado:

- **Firestore Database:** base de datos de documentos NoSQL que permite almacenar, sincronizar y consultar fácilmente datos en tus apps.
- **Authentication:** proporciona servicios de backend, SDK fáciles de usar y bibliotecas de IU ya elaboradas para autenticar a los usuarios en tu app.

que es git

3.4. Git

Git es un sistema de control de versiones, un software que sirve básicamente para gestionar las versiones por las que va pasando el código de los proyectos. Actualmente, es el más popular de los sistemas de control de versiones en la actualidad y una de las herramientas más indispensables para el desarrollo de proyectos.

A pesar de que un sistema de control de versiones sirve para controlar los estados por los quee va pasando un proyecto, la herramienta de git facilita sobre todo el desarrollo en equipo, razón por la que se utiliza ampliamente en el mundo laboral.

En este proyecto se han utilizado diferentes herramientas procedentes de Git para mantener, organizar y actualizar el código fuente del proyecto. [14]

3.4.1. Github

Se trata de una herramienta que utiliza el control de versiones de Git, y se utiliza como repositorio para proyectos. No requiere de pago, por lo que se utiliza para muchos proyectos personales.

3.5. Trello

Trello es una herramienta visual que permite a los equipos gestionar cualquier tipo de proyecto y flujo de trabajo, así como supervisar tareas.

En este proyecto, lo hemos utilizado para llevar un seguimiento de las tareas e historioaas de usuario que teniamos pendientes. [25]

3.6. LaTeX

LaTeX es un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica. Este sistema es el utilizado para redactar la memoria junto con Overleaf [15]

3.6.1. Overleaf

Overleaf es un editor colaborativo de LaTeX basado en la nube que se utiliza para escribir, editar y publicar documentos científicos. Se asocia con una amplia gama de editoriales científicas para proporcionar plantillas oficiales de LaTeX. Es la herramienta utilizada para desarrollar la memoria con LaTeX.

3.7. Astah

Astah es una herramienta de diseño de sistemas que soporta UML y que hemos utilizado en nuestro proyecto para hacer el diagrama de modelo de dominio, el diagrama de casos de uso, el diagrama de secuencia y el de componentes. [2]

Capítulo 4

Análisis del proyecto

En este capítulo de análisis, vamos a mostrar los requisitos funcionales y requisitos no funcionales que debe satisfacer la aplicación.
También vamos a ver las acciones que puede realizar cada tipo de usuario con la aplicación, nombrándolas en el diagrama de casos de uso.
Además, vamos a mostrar en el modelo de dominio todas las entidades que participan, con los atributos y relaciones que tiene con otras entidades.

4.1. Requisitos funcionales (RF)

Los requisitos funcionales son las declaraciones de los servicios que debe ofrecer la aplicación.
En nuestro proyecto, al tratarse de un desarrollo Scrum, estos requisitos funcionales, estos requisitos tienen una relación con las historias de usuario que hemos visto anteriormente. En esta tabla enumeramos los requisitos funcionales: 4.1:

4.2. Requisitos no funcionales (RNF)

Los requisitos no funcionales se refieren a las cualidades, características y restricciones de la aplicación. Al contrario que los requisitos funcionales, no especifican ninguna funcionalidad del sistema. Podemos verlos en la tabla: 4.2.

4.3. Casos de uso

Un diagrama de caso de uso es una descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso.
En esta sección veremos los casos de uso que pueden realizar los distintos usuarios que pueden usar la aplicación, los usuarios sin registrados, los usuarios registrados y los administradores.

Por una parte, los usuarios no registrados o que no han iniciado sesión, las acciones que pueden realizar una vez están en la página principal, es ver los amistosos que hay ya creados, los fanatics que hay creados, registrarse como un nuevo usuario y iniciar sesión. Tras iniciar sesión pasaríamos al siguiente actor, el usuario registrado.

El usuario registrado puede realizar las siguientes tareas:

- Crear un amistoso tras entrar en la pantalla de amistosos.
- Crear un fanatic tras entrar en la pantalla de fanatics.
- Apuntarse a un fanatic tras pedir la confirmación.
- Ver la clasificación de un fanatic tras haberte apuntado a alguno e ir a la pantalla de mis fanatics.
- Sugerir el siguiente partido en el fanatic tras haberse apuntado a un fanatic e ir a la pantalla de mis fanatics.
- Apuntarse y desapuntarse a amistosos, se apunta desde la pantalla Amistosos, y para desapuntarse hay que ir a Mis Amistosos.
- Ver mi Cuenta entrando desde el Dropdown de Mi Cuenta.
- Editar mi perfil desde el Dropdown de Mi Cuenta.
- Eliminar mi perfil desde el Dropdown de Mi Cuenta.
- Ver mis amistosos desde el Dropdown de Mis Partidos.
- Ver mis fanatics desde el Dropdown de Mis Partidos.

Cada uno de estos casos de uso, tiene un flujo para llevarse a cabo. Debido a que son muchos casos de uso, vamos a realiza el flujo de 2 casos de uso para detallar como se comportaría el programa ante la interacción de un usuario. Vamos a explicar los casos de uso de Eliminar un amistoso 4.2 y de Apuntarse a un fanatic 4.1.

Por último, nos falta por ver el usuario administrador, que tiene las mismas posibilidades que el usuario registrado, pero incorpora 2 botones, uno en la pantalla de Amistosos para borrar los Amistosos, y otro la pantalla de Fanatics para eliminar los Fanatics.

Podemos verlo en el siguiente diagrama: 4.3

UseCase	Apuntarse Fanatics
Summary	El usuario debe poder apuntarse a un fanatic.
Actor	:Usuario
Precondition	1-El usuario debe haber iniciado sesión.
Postcondition	1-El usuario debe estar apuntado al fanatic.
Base Sequence	1-El usuario entra en la pestaña de fanatics. 2-El sistema devuelve la interfaz de los fanatics. 3-El usuario selecciona un fanatics del Carousel y pulsa Apuntarme. 4-El sistema pide una confirmación con un pop-up. 5-El usuario selecciona Sí. 6-El sistema solicita el partido a firebase. 7-El sistema introduce al usuario en el fanatic. 8-El sistema devuelve una confirmación de la operación.
Branch Sequence	5.1-El usuario selecciona No, se vuelve al paso 2.
Exception Sequence	6.1-El usuario ya está apuntado al partido, vuelta al punto 2. 6.2-El partido está completo. 8.1-El sistema devuelve un error al introducir al usuario, vuelve al paso 2.
Sub UseCase	Iniciar sesión

Figura 4.1: CU Apuntarse a un fanatic

4.4. Modelo de dominio

El modelo de dominio es una representación de conceptos o entidades significativas del mundo real pertinentes al dominio que deben modelarse en software. [16]

En nuestro proyecto, en el modelo de dominio podemos ver varias tablas.

- **Usuario:** los usuarios se correponden con los usuarios registrados en la aplicación, cuyos datos guardamos en Firebase.
- **Club:** sería un tipo de dato que corresponde al club al que pertenece cada usuario. Sería un atributo del usuario.
- **Administrador:** sería un tipo de usuario que añadimos nosotros desde la base de datos a mano para que puedan borrar partidos.
- **Amistoso:** sería la entidad que guarda los datos de los amistosos creados.
- **Fanatic:** sería la entidad que guarda los datos de los fanatics creados.

UseCase	Eliminar amistoso
Summary	El usuario Administrador debe poder eliminar un partido amistoso.
Actor	:Administrador
Precondition	1-El usuario debe haber iniciado sesión.
Postcondition	1-El amistoso no debe estar en la base de datos.
Base Sequence	1-El usuario va a la pestaña de amistosos. 2-El sistema le devuelve la interfaz de amistosos. 3-El usuario pulsa el botón eliminar partido. 4-El sistema pide una confirmación. 5-El usuario selecciona Sí. 6-El sistema elimina el elemento de firebase. 7-El sistema devuelve una confirmación de la operación.
Branch Sequence	5.1-El usuario selecciona No, se vuelve al paso 2. 5.2-No hay amistosos, se acaba el caso de uso.
Exception Sequence	6.1-El sistema no puede eliminar el partido.
Sub UseCase	Iniciar sesión

Figura 4.2: CU Eliminar un amistoso

En este modelo de dominio vamos a ver como están relacionadas las distintas entidades, con sus atributos, entre sí. 4.4

4.5. Diagramas de secuencia

Los diagramas de secuencia son una solución de modelado dinámico que centran específicamente en líneas de vida o en los procesos y objetos que coexisten simultáneamente, y los mensajes intercambiados entre ellos para ejecutar una función antes de que la línea de vida termine. [6]

A pesar de que hay muchos diagramas posibles, vamos a realizar 3 de ellos para ver como opera la aplicación.

En el primer diagrama, vamos a representar las acciones que se crean cuando el usuario inicia sesión con usuario y contraseña. Podemos ver como el usuario llama a la interfaz(Web) pasandole el usuario y contraseña, y la página a su vez, llama a la base de datos para crear traernos el usuario a la interfaz y de esta manera vemos el inicio de sesión. 4.5

En el segundo diagrama, podemos ver como el usuario le da al botón de Crear Amistoso. Entonces la interfaz recibe el mensaje y le lleva al formulario de creación del amistoso. Ahi

el usuario introduce todos los datos que son necesarios para la creación del partido, y tras confirmar los datos y guardarlos en la base de datos, devuelve el mensaje de ok a la interfaz y lleva al usuario a la pantalla de Amistosos. 4.7.

En este tercer diagrama de secuencia, vemos como el usuario que ha iniciado sesión, le da al botón de Mis Partidos del droprdown de mi cuenta. Entonces le aparece una pantalla con todos los amistosos y fanatics de ese usuario tras haber traído los datos de firebase.
El usuario pulsa a Mis Fanatics para ver los fanatics a los que está apuntado, entonces tras pulsar el botón de clasificación, se realiza una llamada a la base de datos con ese fanatic, que nos devuelve la clasificación del usuario en una nueva pantalla.

Número	Descripción
RF-1	El sistema deberá permitir iniciar sesión a un usuario existente.
RF-2	El sistema deberá permitir registrar un nuevo usuario.
RF-3	El sistema deberá permitir que todos los usuarios vean los partidos amistosos.
RF-4	El sistema deberá permitir que todos los usuarios vean los fanatics.
RF-5	El sistema deberá permitir al usuario registrado editar sus datos.
RF-6	El sistema deberá permitir al usuario registrado borrar su perfil.
RF-7	El sistema deberá permitir al usuario registrado crear un fanatic.
RF-8	El sistema deberá permitir al usuario registrado crear un amistoso.
RF-9	El sistema deberá permitir al usuario registrado borrar su perfil.
RF-10	El sistema deberá permitir al usuario registrado ver los amistosos y fanatics a los que está apuntado.
RF-11	El sistema deberá permitir al usuario registrado apuntarse a un fanatic.
RF-12	El sistema deberá permitir al usuario registrado apuntarse a un amistoso.
RF-13	El sistema deberá permitir al usuario registrado desapuntarse a un amistoso.
RF-14	El sistema deberá permitir al usuario registrado desapuntarse a un fanatic.
RF-15	El sistema deberá permitir al usuario registrado ver la clasificación de sus fanatics.
RF-16	El sistema deberá permitir al usuario registrado crear el siguiente partido del fanatics.
RF-17	El sistema deberá proporcionar una sugerencia del siguiente partido de fanatic.
RF-18	El sistema deberá permitir al administrador borrar amistoso y fanatics.

Cuadro 4.1: Requisitos funcionales

ID	Descripción
RNF-1	El sistema deberá funcionar con React.js
RNF-2	El sistema deberá disponer de conexión a internet.
RNF-3	El sistema deberá acceder a Firebase para obtener los usuarios y los partidos.
RNF-4	El sistema deberá permitir el acceso desde cualquier ordenador.

Cuadro 4.2: Requisitos no funcionales

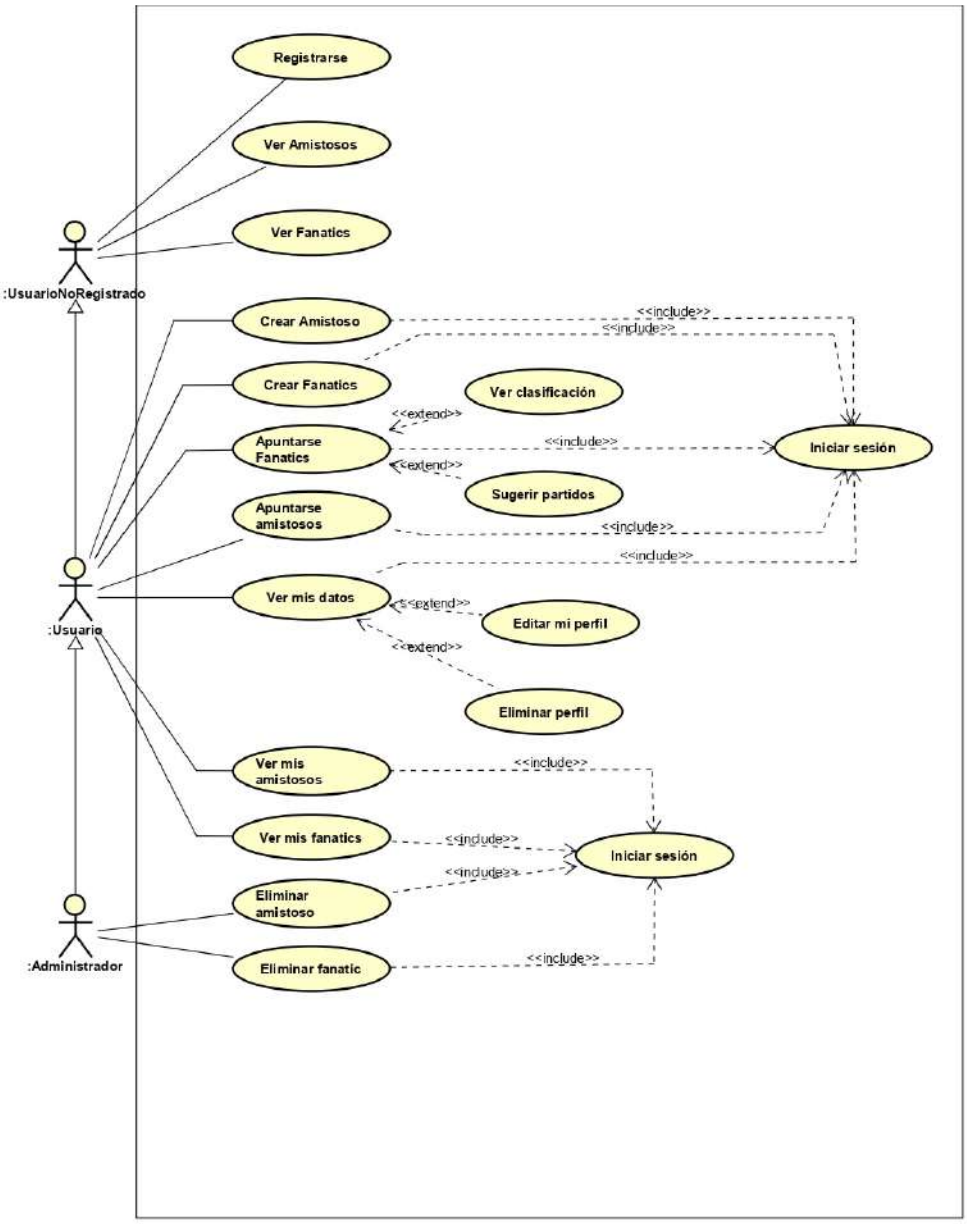

Figura 4.3: Diagrama de casos de uso

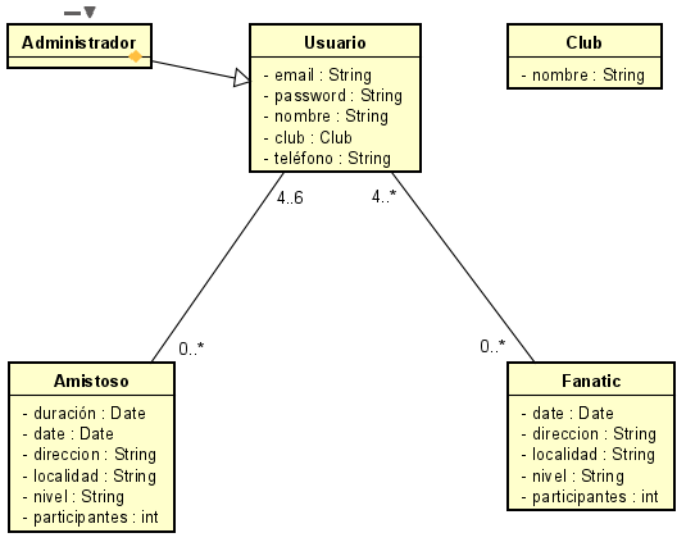

Figura 4.4: Modelo de dominio

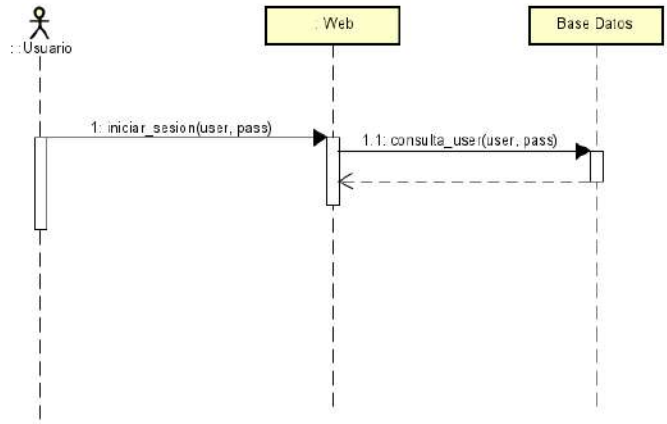

Figura 4.5: Secuencia Inicio Sesión

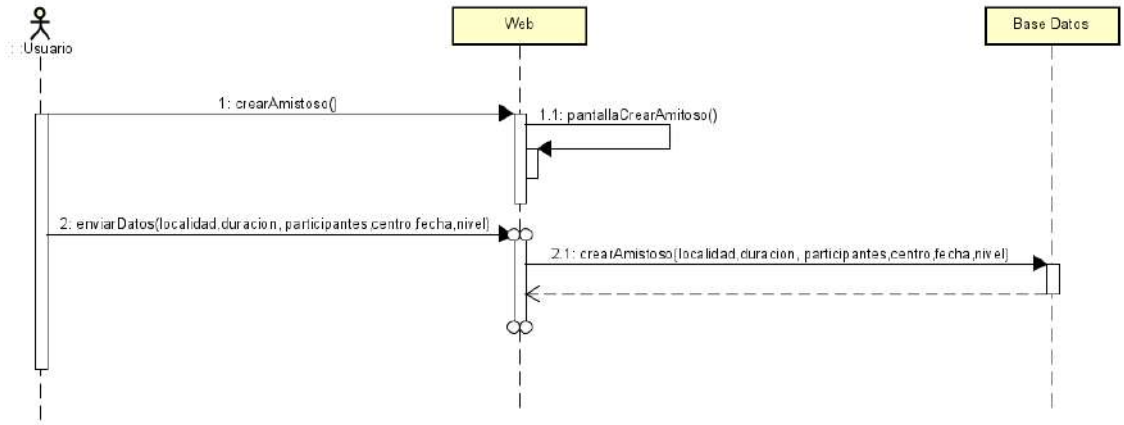

Figura 4.6: Secuencia Registro Amistosos

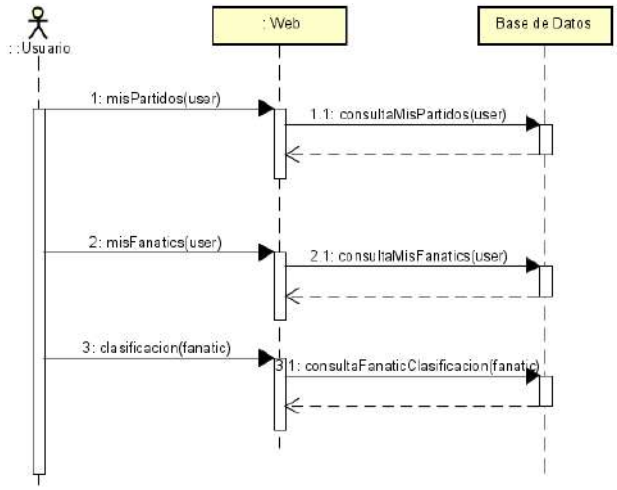

Figura 4.7: Secuencia Ver Clasificación

Capítulo 5

Diseño

Para el diseño de este proyecto, los dos modelos que hemos aplicado son BaaS y SaaS. En cuanto al diseño arquitectónico, nos hemos basado fundamentalmente en el patrón MVVM, el cual hemos desarrollado con componentes. En este capítulo vamos a explicar estos patrones y modelos y vamos a ver también los bocetos de la interfaz que hemos realizado en la página.

5.1. SaaS

El software como servicio (SaaS) [23] es un modelo de distribución y de licencias usado para entregar aplicaciones de software a través de Internet, es decir, como un servicio.

Los usuarios suelen tener acceso a aplicaciones basándose en un modelo de suscripción, lo que hace que SaaS sea la plataforma ideal para aplicaciones de software empresarial tales como el correo electrónico, la mensajería instantánea y la gestión de relaciones con los clientes (CRM).

En este apartado vamos a ver las ventajas y desventajas del modelo SaaS. [27]

5.1.1. Ventajas

- **Ahorro en costos de utilización:** no es necesario que compres el software, solo necesitas conexión a Internet.
- **Acceso en cualquier sitio:** se alojan directamente en la nube, sin que se necesite de un ordenador físico que lo mantenga. Esto brinda agilidad, practicidad y usabilidad.
- **Adaptabilidad para cubrir las necesidades del cliente:** se adapta perfectamente a lo que necesites en tu empresa, pudiendo ampliar o reducir los paquetes, aumentando su eficiencia y logrando una personalización adecuada.

- **Actualizaciones automáticas:** el proveedor del servicio gestiona todas las actualizaciones, eliminando la necesidad de descargar e instalar la aplicación.
- **Fácil integración con otros sistemas:** el software está diseñado para permitir desde ya esta integración de manera rápida y muy sencilla.
- **Garantiza la continuidad del negocio:** en caso de un fallo del proveedor de energía o algún otro problema, SaaS te garantiza una solución y que tu empresa va a poder continuar con la operación y mantener la información bien protegida.

5.1.2. Desventajas

- Este Servicio depende de tener conexión a internet.
- No se puede modificar el sistema, ya que es el proveedor el que tiene la potestad de modificar las funcionalidades.
- Al ser centralizado, un fallo deja sin servicio a todos los usuarios.

5.2. BaaS

BaaS [3] es una plataforma que automatiza el desarrollo del lado del backend y se encarga de la infraestructura en la nube. Con un BaaS, subcontratará las responsabilidades de ejecución y mantenimiento de servidores a un tercero y se centrará en el desarrollo del lado del cliente o de la interfaz.

En la figura5.1 resumimos como funciona dicha arquitectura:

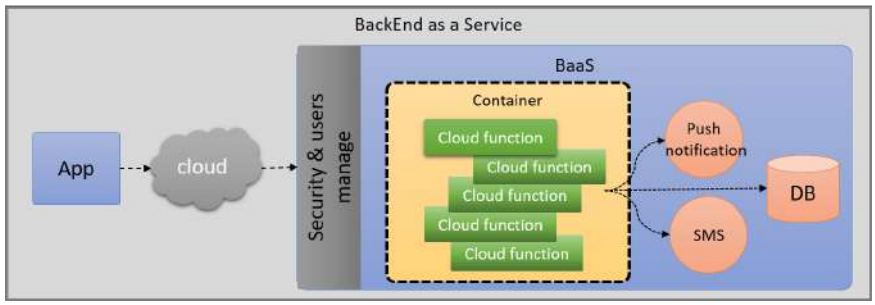

Figura 5.1: BaaS

Esta tecnología permite que el desarrollador se quite de encima tareas que se traducen en mucho trabajo, como son la seguridad, sistemas de autenticación,etc. Ahora veremos en esta lista las ventajas y desventajas:

5.2.1. Ventajas

- **Gestión sencilla:** las gestiones como rotación de dispositivos de almacenamiento, traslado de datos, verificacion de integridad, etc, las realiza el proveedor.
- **Optimización de costes:** el proveedor lleva a cabo todas estas tareas por un precio fijo mensual.
- **Seguridad:** la información que guardamos, se encuentra respaldada por una copia de seguridad en un sistema BaaS, que no está expuesto a amenazas de hackers, desastres naturales y errores humanos. Además, los datos almacenados en BaaS están encriptados.
- **Escalabilidad:** pueden crecer en función de la demanda del producto.
- **Optimización de los recursos IT:** las empresas que utilizan este modelo pueden utilizar al personal informático en otras tareas al ser un servicio externo.

5.2.2. Desventajas

- Ofrece menos opciones de personalización al ser un servicio externo.
- Este Servicio depende de tener conexión a internet.

En este modelo, la tecnología usada ha sido la base de datos de Firebase.

5.3. El patrón Model-View-ViewModel

En este proyecto hemos utilizado el patrón MVVM (Model-View-ViewModel) [10]. Es un método de programación que trata de separar al máximo la interfaaz de usuario con la lógica de la aplicación. Está formado por 3 partes:

- **View:** es responsable de definir la estructura, el diseño y la apariencia de lo que ve el usuario en la pantalla. En nuestro proyecto, está formado por JSX y por CSS.
- **Model:** son entidades no visuales que encapsulan los datos de la aplicación. Representa el modelo de dominio.
- **ViewModel:** implementa propiedades y comandos a los que la vista puede enlazar datos y notifica a la vista los cambios de estado a través de eventos de notificación de cambios. Las propiedades y comandos que proporciona el modelo de vista definen la funcionalidad que ofrece la interfaz de usuario, pero la vista determina cómo se va a mostrar esa funcionalidad.

En la descripción de este patrón, podemos ver que es muy similar al patrón MVC(Modelo-Vista-Controlador), pero se diferencia de él en que el patrón MVVM permite que cuando el usuario interaccione con la interfaz, los datos del modelo se actualicen automáticamente.

Figura 5.2: MVVM

Este modelo se adapta muy bien a la tecnología que usamos para nuestro proyecto, React.js. Ahora vamos a hablar de las ventajas y desventajas del patrón MVVM [26]:

5.3.1. Ventajas

- La lógica de negocio está desacoplada de la interfaz de usuario.
- Los componentes pueden ser reutilizados
- El mantenimiento de los sistemas es simplificado

5.3.2. Desventajas

- La curva de aprendizaje para nuevos desarrolladores es un poco superior a los otros modelos que son más simples.
- La distribución de componentes nos obliga a la creación y mantenimiento de un mayor número de ficheros.
- Debes de adaptarte a una estructura predefinida y eso incrementa la complejidad del sistema.

5.4. Diseño guiado por componentes

El diseño guiado por componentes [7] es un patrón de diseño que permite desgranar un sistema software en partes más pequeñas o componentes. Hoy en día es un diseño muy utilizado para el desarrollo de páginas web. Vamos a ver las ventajas y desventajas de utilizar este tipo de diseño :

5.4.1. Ventajas

- **Facilidad de Instalación:** cuando una nueva versión esté disponible, usted podrá reemplazar la versión existente sin impacto en otros componentes.
- **Costos reducidos:** el uso de componentes de terceros permite distribuir el costo del desarrollo y del mantenimiento.
- **Facilidad de desarrollo:** los componentes implementan un interface bien definida para proveer la funcionalidad definida permitiendo el desarrollo sin impactar otras partes del sistema.
- **Reusable:** el uso de componentes reutilizables significa que ellos pueden ser usados para distribuir el desarrollo y el mantenimiento entre múltiples aplicaciones y sistemas.
- **Mitigación de complejidad:** cuando un problema requiere demasiado esfuerzo, se divide en partes más pqueñas y simples.

5.4.2. Desventajas

- Aumenta la complejidad para los diseñadores que tienen que desgranar la solución.
- Requiere un mayor esfuerzo de testeo, ya que hay que comprobar la funcionalidad y compatibilidad de cada componente.

5.4.3. Componentes creados en este proyecto

En esta sección vamos a explicar brevemente la funcionalidad de cada uno de los componentes creados en este proyecto. Los componentes en React tienen 2 partes, la parte de la lógica, en la podemos crear funciones que modifiquen, creen o eliminen datos del modelo, y la parte de la interfaz, que estaría formado por el JSX.

Como se puede ver en la figura 5.3 solo han sido necesario crear 5 componentes:

- **App:** componente inicial, contiene toda la aplicación
- **BarraMenú:** es el componente que aparece en la barra superior. Cambia en función de si el usuario ha iniciado sesión. En caso de no haber iniciado sesión, da la posibilidad de registrarse e iniciar sesión, y en caso contrario aparece un dropdown quee permite ir a mis credenciales, mi perfil, mis partidos y cerrar sesión.
- **CardAmistosos:** muestra todos los amistosos creados, y en caso de haber iniciado sesión, nos deja crear un amistoso
- **FormCrearAmistoso:** formulario para crear un amistoso.
- **FormRegistroUsuario:** formulario para crear un usuario.

- **CardFanatics:** muestra todos los fanatics creados, y en caso de haber iniciado sesión, nos permite.
- **FormCardFanatics:** formulario para crear un fanatic.
- **CardInicioSesion:** permite iniciar sesión.
- **CardMisPartidos:** muestra todos los amistosos y fanatics a los que estoy apuntado.
- **CardMisFanatics:** muestra los fanatics a los que estoy apuntado.
- **CardSiguienteFanatic:** muestra al usuario el siguiente partido, sugeriendo uno.
- **CardClasificaciónFanatic:** muestra al usuario la clasificación de ese fanatic.
- **CardMisAmistosos:** muestra los amistosos a los que estoy apuntado.
- **CardCredencial:** permite a un usuario cambiar sus credenciales.
- **MiPerfil:** muestra al usuario sus datos y le permite actualizarlos.

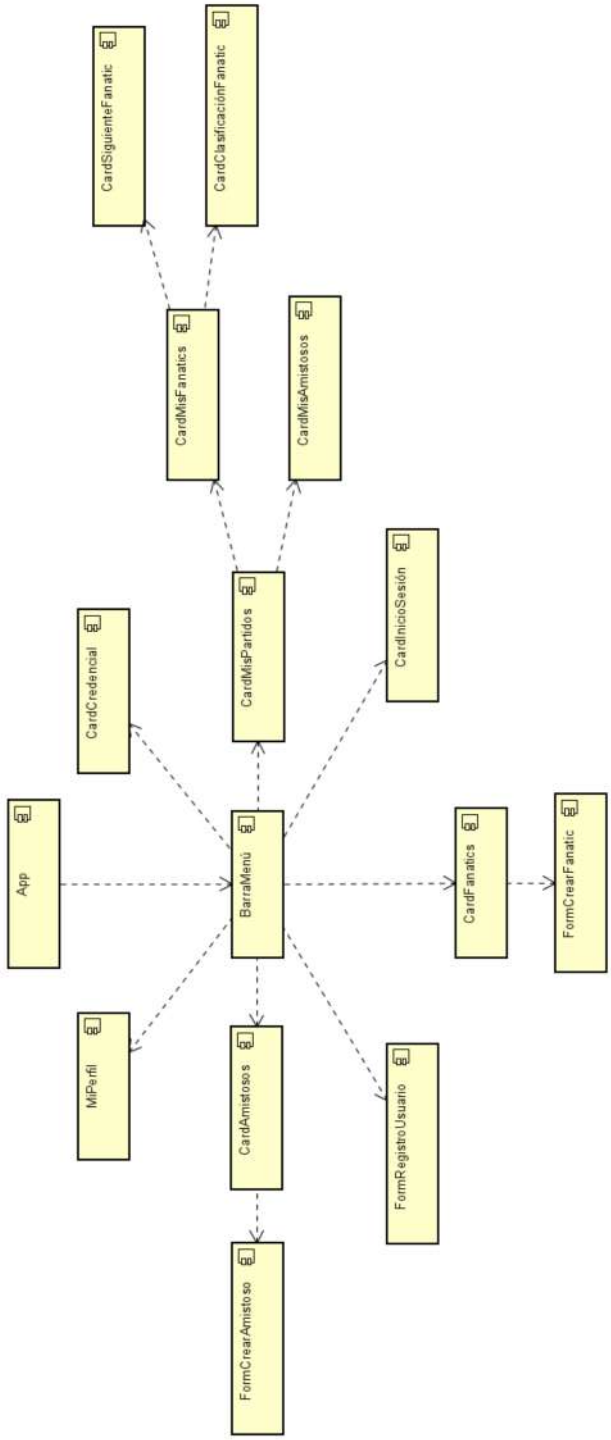

Figura 5.3: Diagrama de componentes

5.5. Algoritmo

En esta sección vamos a explicar brevemente el algoritmo utlizado para emparejar a los usuarios en los torneos de fanatics.

5.5.1. Algoritmo Fuerza bruta

Los algoritmos de fuerza bruta son capaces de encontrar la solución a cualquier problema por complicado que sea. Su fundamento es muy simple, probar todas las posibles combinaciones, recorrer todos los caminos hasta dar con la situación que es igual que la solución. [1]

En la base de datos que explicaremos posteriormente, tenemos una tabla de fanatics, y dentro de esta tabla, tenemos 2 arrays, en uno guardamos los usuarios que hay apuntados en total, y en el otro guardamos una cadena que une los usuarios que ya han jugado de esta manera üser1;user2". 5.4

Entonces lo que hace el algoritmo es elegir un usuario, y probar combinaciones con otro usuario con las siguientes condiciones:

- que no hayan jugado juntos (que no estén en el array "parejasUsuarios").
- que no jueguen como pareja el mismo usuario.
- que no juegue el mismo usuario en los dos equipo.

Figura 5.4: Emparejamientos

5.6. Interfaz

En esta sección, vamos a ver una a una las interfaces y vamos a esplicar su funcionamiento y las posibilidades que tiene el usuario para interactuar con la aplicación. Este proyecto se ha desarrollado con el objetivo de crear una página web sencilla y simple, con el objetivo de que al usuario le cueste poco esfuerzo aprender a utilizar la página web.

En esta primera vista 5.5 podemos ver la página Home, en la que tenemos 2 partes: una es la BarraMenú, en la que las funciones que podemos realizar son iniciar sesión, registrarnos, en el caso de que no hayamos iniciado sesión, y ver nuestra cuenta, en caso de que hayamos iniciado sesión, en la parte derecha, y en la parte izquierda tenemos el botón de vuelta a la Home, que es el icono de Padel Smash, y además podemos ver la página de amistosos y fanatics.

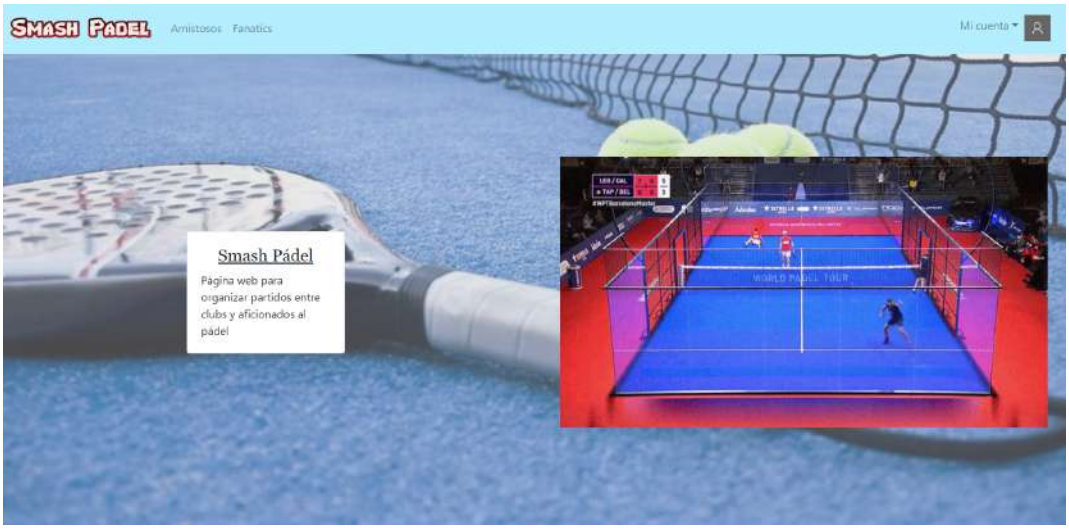

Figura 5.5: Captura Home

En esta vista 5.6 podemos ver los amistoso que hay creados. Podemos pasar de uno a otro con la flechas de un Carousel, y en caso de haber iniciado sesión, nos aparece debajo un botón para crear un amistoso. En cada amistoso, nos dejará apuntarnos en caso de haber iniciado sesión. Tras pulsar el botón nos pedirá una confirmación, y tras pulsar que sí nos apuntará al partido en la base de datos tras comproobar que el usuario no estaba apuntado anteriormente y que el partido no está cómpleto. También dispondremos en la parte superior de la barra menú, que nos permite las mismas operaciones que en la vista anterior, y que estará visible en toda la aplicación.

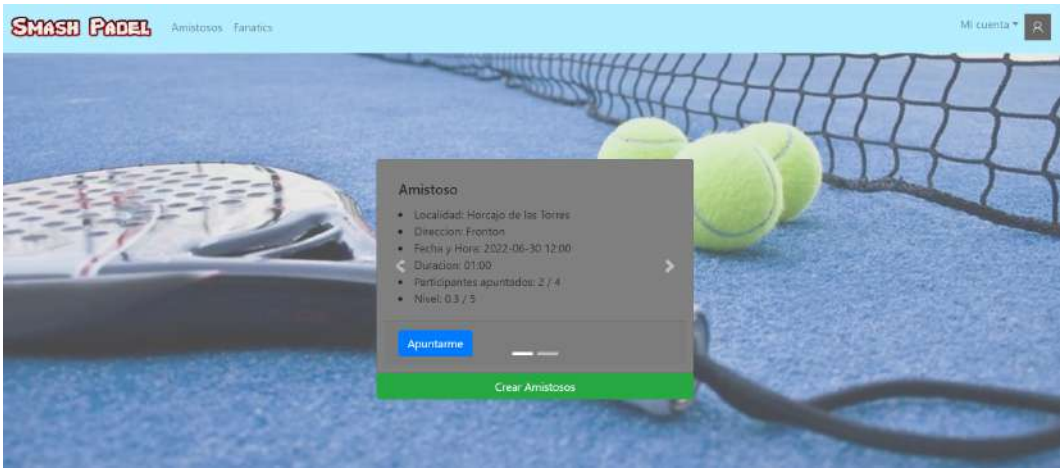

Figura 5.6: Captura Amistosos

En esta vista 5.7 podemos ver los fanatics que hay creados. Es similar a la página de amistosos. En ella podremos apuntarnos al fanatic, y se realizarán las mismas comprobaciones para ver que el usuario no está metido en el torneo y que el límite de usuario aún permite la inscripción de otro usuario.

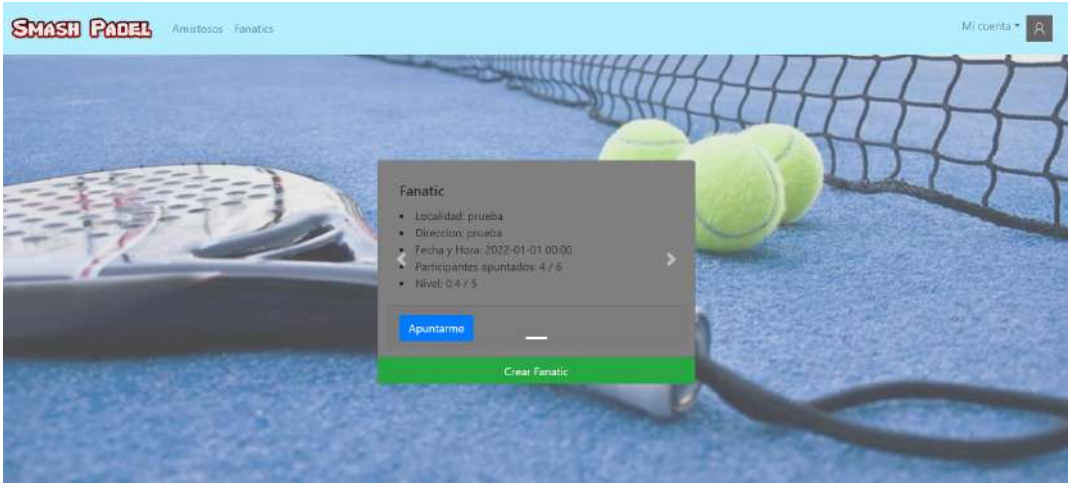

Figura 5.7: Captura Fanatics

En esta vista 5.8 podemos iniciar sesión si estamos registrados en Firebase. Hace falta usuario y contraseña. Tenemos que introducir el correo que corresponde al campo email con el que registramos el usuario, y la contraseña sería el campo que no pide confirmar en el registro. Tras introfucir estos datos pulsamos el botón iniciar sesión.

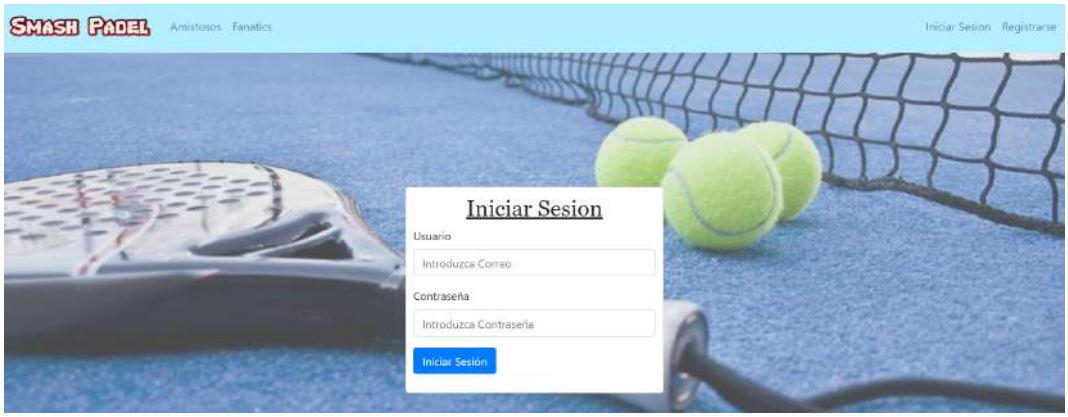

Figura 5.8: Captura Inicio Sesión

En esta vista 5.9 vemos el formulario para registrarnos como usuario. En esta pantalla, debemos rellenar todos los datos, que constan del Nombre, Apellidos, Telefono, Club(hay 3 clubes guardados), Género, Contraseña y la confirmación de la contraseña, que se ha añadido una comprobación para que muestre error en caso de que no coincidan o no sean válidas(menos de 6 caracteres). Tras esto se crea un registro en la base de datos de Usuarios.

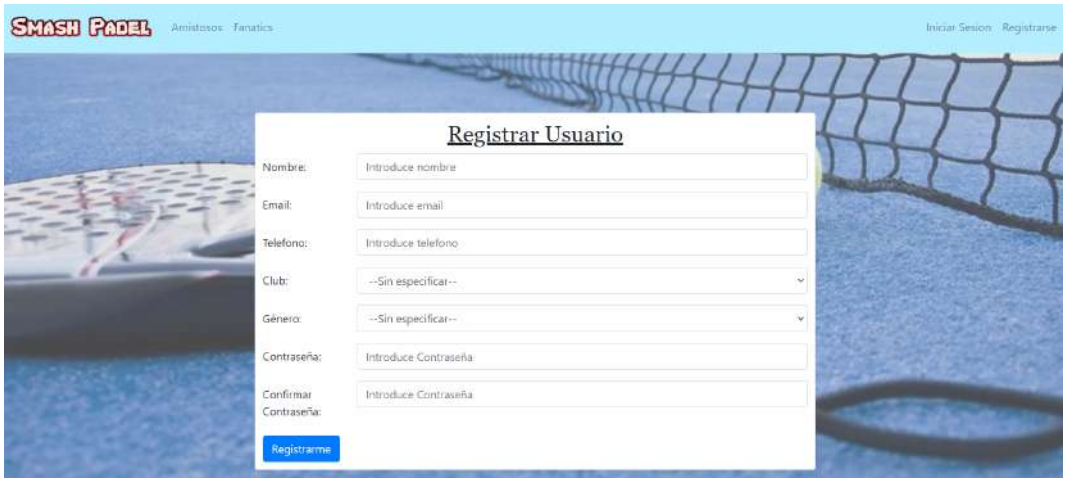

Figura 5.9: Captura Registrarse

En esta vista 5.10 vemos un formulario para crear un Amistoso. En este formulario tenemos que rellenar los campos de Localidad, Centro, Fecha y hora(es un Calendar), Duración, Polideportivo y Nivel, que sería un valor decimal entre 0 y 5, para equilibrar los niveles de los partidos. Tras esto se crea un registro en la base de datos de Amistosos.

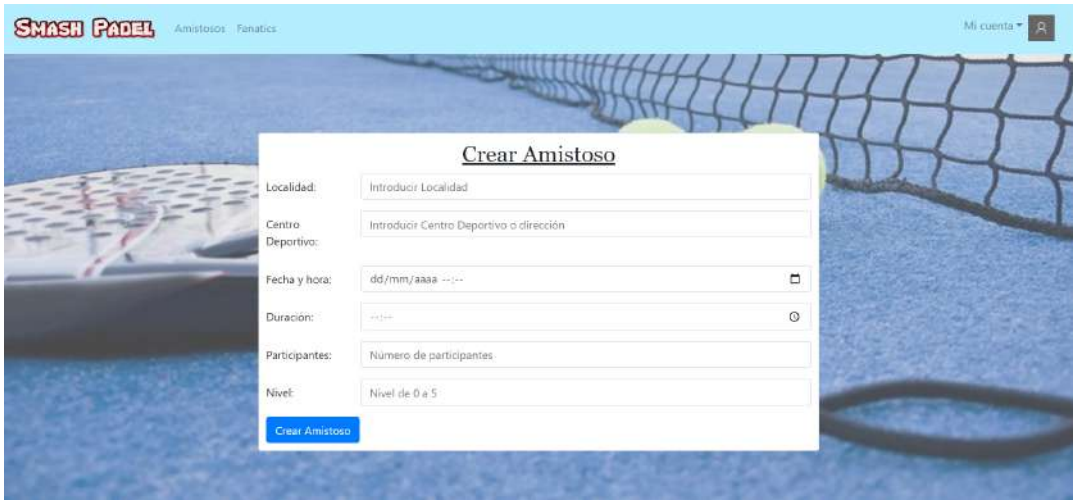

Figura 5.10: Captura Crear Amistoso

En esta vista 5.11 vemos un formulario para crear un Fanatic. Es similar al de crear un amistoso, sólo que no aparece el campo de duración, ya que el Fanatic depende de la disponibilidad de los usuarios para poder avanzar el torneo.

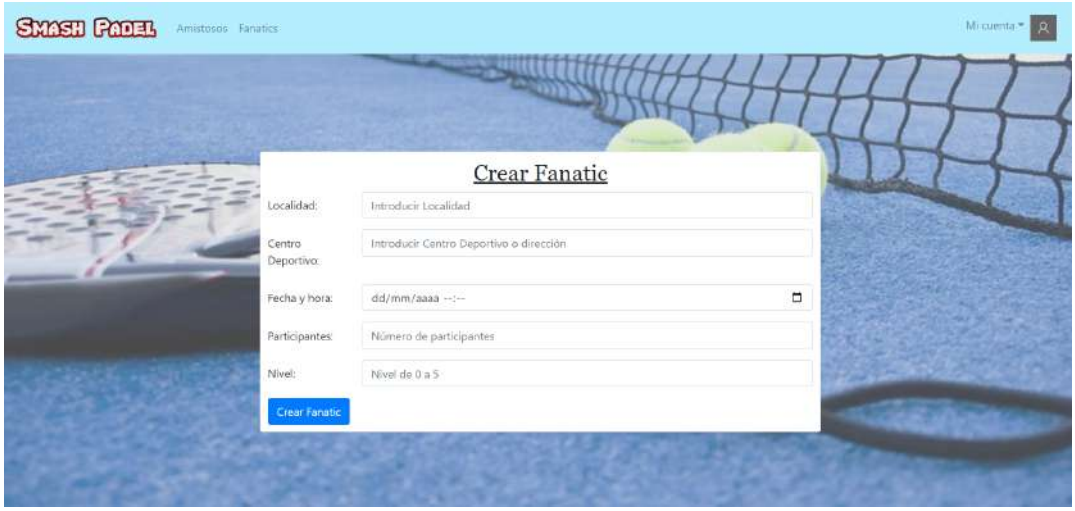

Figura 5.11: Captura Crear Fanatic

En esta vista 5.12 vemos nuestro perfil, en el que aparecen los datos de Nombre, Email, Telefono, Club y Género. Estos valores podemos cambiarlos y, tras pulsar el botón de actualizar, se realiza una llamada a la base de datos dee firebase para modificar el registro. También podemos pulsar el botón de Borrar Usuario, lo que elimina el usuario de la base de

datos.

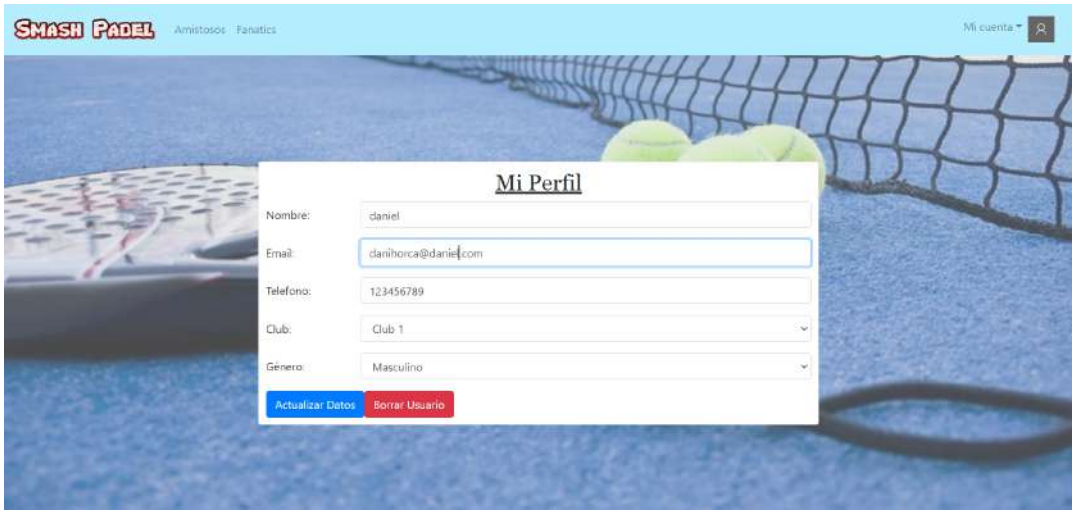

Figura 5.12: Captura Mi Perfil

En esta vista 5.13 podemos cambiar nuestras credenciales. Para ello tenemos que cambiar los dos campos, y para que que se realice la operación correctamente y para eliminar un error humano del usuario, realizamos la comprobación de que sean iguales ambos campos. Tras esto se actualizan las credenciales en firebase.

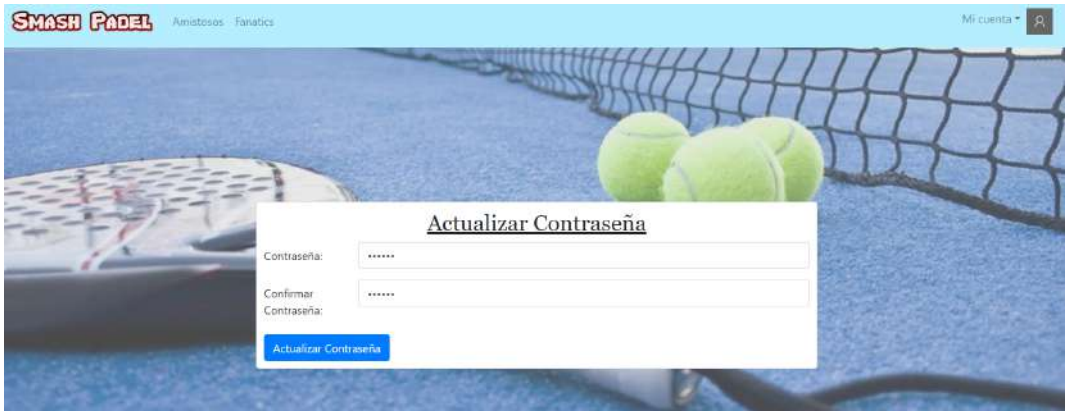

Figura 5.13: Captura Credenciales

En esta vista 5.14 podemos ver los partidos a los que estamos apuntados, tanto los amistosos como los fanatics. Esta vista nos sirve para tener una visión general de los siguientes partidos o torneos que un usuario tiene por delante, pero en esta vista no podría apuntarse o desapuntarse. La función que se puede realizar desde esta pantalla es ir a Mis Amistosos o

Mis Fanatics, los cual nos mostraría los amisotos o fanatics que estamos apuntados, y desde ahí sí podríamos realizar funciones con ellos que veremos en las siguientes vistas.

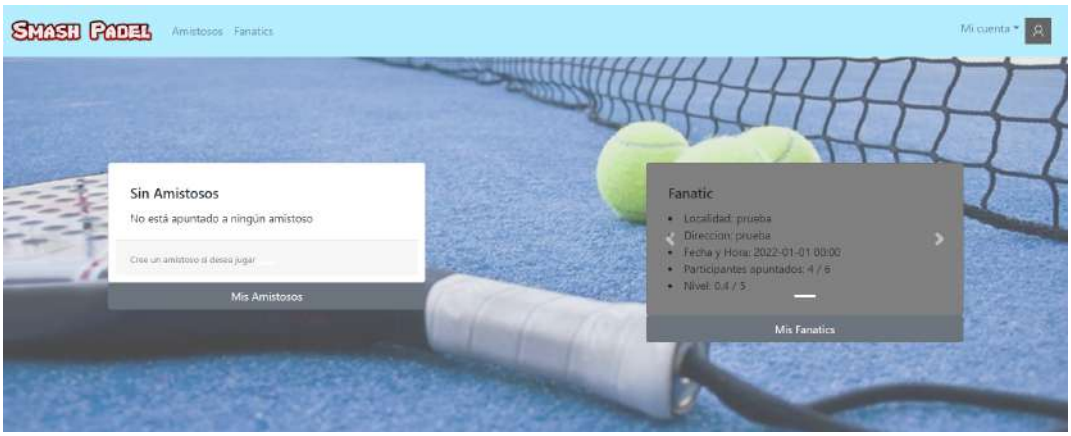

Figura 5.14: Captura Mis Partidos

En esta vista 5.15 podemos ver los amistosos a los que estamos apuntados. En ella, si tenemos amistosos a los que estamos apuntados, nos aparecerá un botón en rojo para desapuntarnos. Tras pulsar nos pedirá confirmación y tras darle a sí eliminaremos al usuario del amistoso.

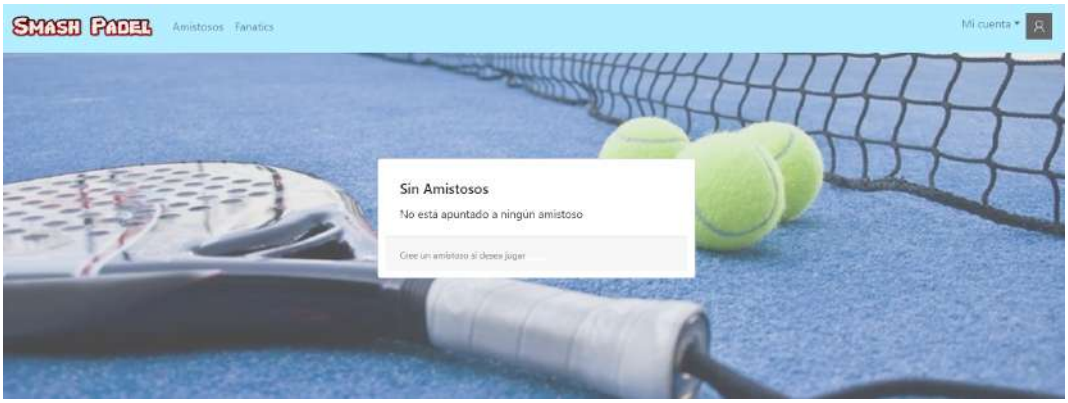

Figura 5.15: Captura Mis Amistosos

En esta vista 5.16 podemos ver los fanatics a los que estamos apuntados. Desde esta vista , podemos ver las acciones que podemos realizar con los fanatics pulsando sus respectivos botones, que son ver la clasificación, y ver el siguiente partido, que los explicaremos en las siguientes pantallas.

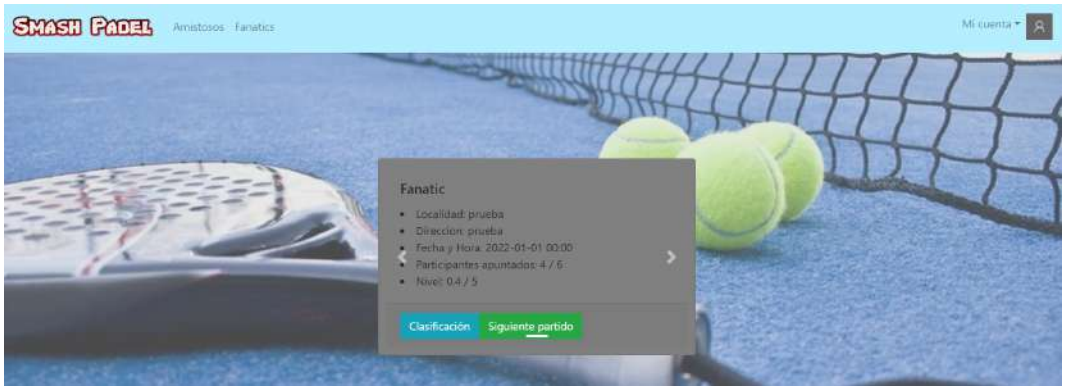

Figura 5.16: Captura Mis Fanatics

En esta vista 5.20 vemos cual va a ser el siguiente enfrentamiento. Desde esta interfaz podemos introducir los usuarios con el desplegable que tenemos, seleccionando a cada participante que queramos en cada una de las parejas. Para ayudar a realizar bien este torneo, hemos introducido la opción de Sugerir partido, que tras pulsar el botón, lo que hace es ofrecernos una posibilidad para jugar un partido con parejas que no hayan jugado antes.

Figura 5.17: Captura Siguiente Partido

En esta vista 5.18 la clasificación del fanatic que hayamos seleccionado. A esta vista llegamos desde el botón de Clasificación de Mis Fanatics, y la función que podemos realizar es que al pulsar el botón de Obtener Clasificación, nos devuelve la clasificación total de ese fanatic.

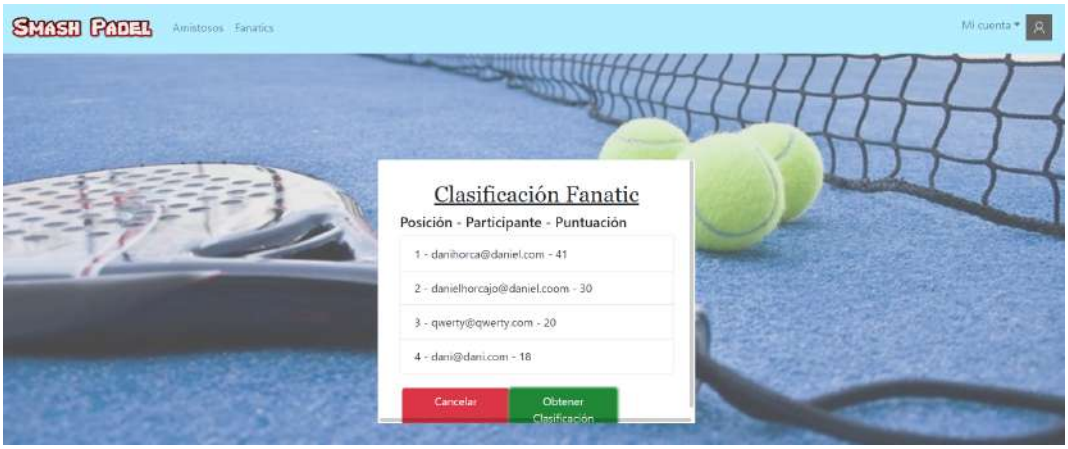

Figura 5.18: Captura Clasificación

5.7. Base de Datos

En este apartado, vamos a hablar de la base de datos que hemos utilizado en este proyecto, que sería Firebase.

Firebase es una plataforma móvil creada por Google, cuya principal función es desarrollar y facilitar la creación de apps de elevada calidad de una forma rápida. [11]
Las funciones que hemos utilizado de firebase en este proyecto son 2:

5.7.1. Authentication

Firebase Authentication proporciona servicios de backend, SDK fáciles de usar y bibliotecas de IU ya elaboradas para autenticar a los usuarios en tu app. Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federada populares, como Google, Facebook y Twitter, etc. [12]

En nuestro proyecto, esta función de firebase ha sido utilizada para guardar los usuarios de la aplicación y su contraseña de forma segura. Además firebase tiene funciones que nos permite tanto la creación, como el borrado de estos registros de manera muy sencilla.

Figura 5.19: Captura Firebase Authentication

5.7.2. Firestore Database

Cloud Firestore es una base de datos de documentos NoSQL que permite almacenar, sincronizar y consultar fácilmente datos en tus apps web y para dispositivos móviles a escala global. [13]

En nuestro proyecto, ha sido la funciona más utilizada por nosotros, ya que nos permite crear una base de datos con tipos de datos complejos como arrays o mapas, los cuales nos permiten introducir y recoger datos.

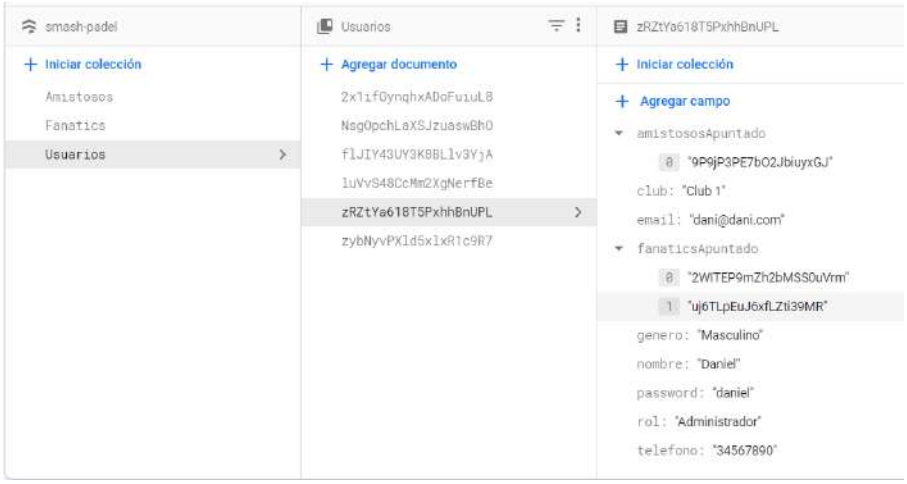

Figura 5.20: Captura Firestore Database

Tenemos registradas las siguientes tablas:

- **Usuarios:** en esta tabla guardamos todos los datos personales del usuario, que luego se le muestran en Mi Perfil.

- **Fanatics:** en esta tabla guardamos los datos de los Fanatics. También guardamos datos como los usuarios apuntados a cada fanatic y los puntos de cada uno.
- **Amistosos:** en esta tabla guardamos los datos de los Amistosos. También guardamos los usuarios que están en cada amistoso.

Capítulo 6

Plan de riesgos y estimación de costes

6.1. Plan de riesgos

El riesgo es un suceso cuya probabilidad de que ocurra es incierta, y en caso de ocurrir, puede poner en peligro los objetivos de un proyecto.

Estos riesgos los podemos dividir en varios tipo:

- **Riesgos del proyecto:** afectan a la planificación, al coste y a la calidad del proyecto. Suelen ser problemas de presupuesto, personal, de tiempo, etc.
- **Riesgos técnicos:** amenazan a la calidad dell producto a desarrollar. Suelen ser problemas de diseño, obsolescencia técnica, implementación, etc.
- **Riesgos de negocio:** son riesgos que suponen una amenaza para el desarrollo del proyecto. Se suele deber a riesgos de mercado, de presupuesto, de ventas. etc..

También se pueden dividir según la probabilidad de predicción sobre esos riesgos:

- **Predecibles y conocidos:** se pueden identificar analizando el proyecto y su entorno. Se puede estimar probabilidad de ocurrencia.
- **Predecibles:** Se pueden intuir a partir de la experiencia en otros proyectos, pero no se puede estimar la probabilidad de ocurrencia.
- **Impredecibles:** Pueden ocurrir, pero son difíciles de identificar con anticipación.

Hay una serie de paso a seguir para identificar estos riesgos en los proyecto:

1. Identificación de riesgos.
2. Análisis y establecimiento de prioridades.
3. Planificación de riesgos.
4. Monitorización de riesgos.

6.1.1. Riesgos encontrados en este proyecto

A coontinuación, vamos a hacer un análisis de los riesgos detectados para este proyecto:

Título	Riesgo-01
Descripción	Un miembro del equipo de desarrollo no dispone de los suficientes conocimientos ni formación para realizar el proyecto
Probabilidad	Media
Impacto	Medio
Medidas preventivas	Se habilita una formación relacionada con esta tecnología
Plan de contingencia	Ayudar al miembro del equipo en sus primeras tareas y con su formación

Cuadro 6.1: Riesgo de falta de formación

Título	Riesgo-02
Descripción	Enfermedad de un miembro del equipo
Probabilidad	Baja
Impacto	Alto
Medidas preventivas	Realizar estimaciones con un tiempo de margen para la entrega final del proyecto, para evitar este tipo de contratiempos
Plan de contingencia	Intentar retrasar la entrega del evolutivo

Cuadro 6.2: Riesgo de enfermedad

Título	Riesgo-03
Descripción	Mala planificación del tiempo de un alguna tarea o evolutivo
Probabilidad	Baja
Impacto	Medio
Medidas preventivas	En la estimación del tiempo, poner un tiempo de margen para evitar contratiempos
Plan de contingencia	Intentar retrasar la entrega del evolutivo

Cuadro 6.3: Riesgo de falta de tiempo

Título	Riesgo-04
Descripción	Cambian los requisitos del sistema, añadiendo o quitando funcionalidad
Probabilidad	Alta
Impacto	Medio
Medidas preventivas	Desarrollar de una manera modular, que permitirá facilitar los cambios de última hora
Plan de contingencia	Modificar componentes anteriores

Cuadro 6.4: Riesgo de cambio de requisitos

Título	Riesgo-05
Descripción	Limitaciones de la tecnología para determinados requisitos
Probabilidad	Baja
Impacto	Alto
Medidas preventivas	Realizar un análisis de los requisitos para comprobar que se puede realizar todos.
Plan de contingencia	Cambiar los requisitos problemáticos por otra solución permitida

Cuadro 6.5: Riesgo de limitaciones tecnológicas

Título	Riesgo-06
Descripción	Cortes en el servicio debido a averias en la máquina, cortes de electricidad, etc.
Probabilidad	Baja
Impacto	Medio
Medidas preventivas	Subir a la nube todas las tareas realizadas diariamente
Plan de contingencia	Rehacer todas las tareas o documentos que se hayan perdido tras este problema

Cuadro 6.6: Riesgo de averías

6.2. Presupuesto

En este proyecto, tenemos que sumar los costes de un único desarrollador, que tendría un sueldo de desarrollador junior, que se estima en unos 18.000 euros anuales.
Con este salario, se calcula aproximadamente unos 9 euros la hora, y teniendo en cuenta que este proyecto son 300h, nos da un total de 2700 euros en salarios.

A esto hay que sumar el hardware utilizado por el desarrollador, que tiene una computadora valorada en 600 euros. Consultando la tabla de coeficientes de amortización lineal [24], vemos que los equipos para procesos de información tienen un coeficiente máximo del 25 %, y como este equipo lo hemos usado durante 4 meses, nos da una operación 600*(0.25)*(4/12), que resulta en 50 euros.

El total de la operación sería de 2700 del salario más 50 del equipo informático, lo que nos da un total de 2750 euros como coste final del proyecto.

Capítulo 7

Implementación

En el capítulo de implementación vamos a ver la estructura del código del proyecto.

7.1. Estructura del código del proyecto

Los ficheros del proyecto tienen la siguiente estructura:

```
|   .gitignore
|   package-lock.json
|   package.json
|   yarn.lock
|   README.md
|
+---public
|   index.html
|   logo192.png
|   logo512.png
|   favicon.ico
|   manifest.json
|   robots.txt
|
+---node_modules (se omite su contenido por brevedad)
|
+---src
|   |   App.css
|   |   App.jsx
|   |   firebaseconfig.js
|   |   index.css
|   |   index.jsx
|   |
```

```
|   +---components
|   |    +---BarraMenu.css
|   |    +---BarraMenu.jsx
|   |    +---CardAmistosos.jsx
|   |    +---CardClasificacionFanatic.jsx.jsx
|   |    +---CardCredencial.jsx
|   |    +---CardFanatics.jsx
|   |    +---CardInicioSesion.jsx
|   |    +---CardMisAmistosos.jsx
|   |    +---CardMisPartidos.jsx
|   |    +---CardMisFanatics.jsx
|   |    +---CardSiguienteFanatic.jsx
|   |    +---FormCrearAmistoso.jsx
|   |    +---FormCrearFanatics.jsx
|   |    +---FormRegistroUsuario.jsx
|   |    +---MiPerfil.jsx
|   |
|   |
|   +---imagenes
|   |   |    dejada.gif
|   |   |    imagenFondo.svg
|   |   |    imagenFondoAclarada.svg
|   |   |    juan-lebron.gif
|   |   |    smash_padel.svg
|   |   |    usuario.svg
|   |
|   +---vistas
|   |   +---Amistosos.jsx
|   |   +---ClasificacionFanatic.jsx
|   |   +---CrearAmistoso.jsx
|   |   +---CrearFanatics.jsx
|   |   +---Credencial.jsx
|   |   +---Fanatics.jsx
|   |   +---Home.jsx
|   |   +---IniciarSesion.jsx
|   |   +---MisAmistosos.jsx
|   |   +---MisFanatics.jsx
|   |   +---MisPartidos.jsx
|   |   +---Perfil.jsx
|   |   +---Registrarse.jsx
|   |   +---SiguienteFanatic.jsx
|
```

Ahora vamos a explicar que es cada apartado:

Los primeros archivos tienen que ver con toda la configuración de la página web. Podemos ver las bibliotecas utilizadas, las versiones de las librerías, etc. Además tenemos también el gitIgnore, que su función es ignorar los archivos que se encuentren en él a la hora de subir cambiar a GitHub.

Los archivos que vemos en el apartado de public, son imágenes que vienen a la hora de crear el proyecto, y que hemos utilizado para pruebas.

En node-modules se encuentran todas las dependecias del proyecto, no las especificamos por brevedad.

En la carpeta de src, vemos que tenemos 3 subcarpetas (componentes, imagenes, y vistas), y al principio archivos como App.jsx, que es donde arranca la aplicación, y también firebaseconfig, que nos sirve para traer una instancia de la base de datos de firebase.
En la subcarpeta de imágenes, tenemos guardados los logos y los gifs que utilizamos en la página, desde el logo de padel-smash hasta el fondo de toda la aplicación.
En la subcarpeta de componentes, tenemos todos los componentes que cumplen una funcionalidad concreta.
En la subcarpeta de vistas, tenemos todas las interfaces, que llaman a esos componentes y le dan una estructura y un formato a esa vista.

7.2. Decisiones a lo largo del proyecto

Tanto antes de comenzar el proyecto como durante su desarrollo, se han tomado decisiones imporantes para la implementación de la página web. Comentaremos alguna de ellas.

Primero fue la elección de React.js como framework para el desarrollo en la parte de front. Esta elección se debió a que tras ver cuales eran los frameworks más utilizados de javascript, vi que React era uno de los más utilizados y que ofrecía más posibilidades en el desarrollo, a pesar de que su curva de aprendizaje es más complicada que otros frameworks.

La elección de Firebase la hicimos debido a que la creación de las tabllas era muy sencilla, y Authentication de firebase ofrece muchas facilidades a la hora de guardar usuarios, lo que nos permitía no invertir mucho esfuerzo en esta parte.

7.3. Cambios realizados a lo largo del proyecto.

Entre los cambios más importantes respecto a la planificación inicial del proyecto, se encuentra el despliegue de la página web, ya que debido a problemas con la versión de React y la versión de la herramienta de Firebase para hostear la aplicación, no hemmos podido implementar eso, y quedará como mejora futura, que explicaremos en el apartado de conclusiones.

Capítulo 8

Conclusiones y futuro

8.1. Conclusiones

Una vez finalizado el proyecto, la primera conclusión es que se ha cumplido en su mayoría con los objetivos iniciales para el desarrollo de la página.

El código de la página, lo hemos subido al siguiente repositorio de github: `https://github.com/daniorkjo/SmashPadelTFG.git`. En este repositorio, hemos eliminado el contenido de firebaseconfig.js, que se encuentra en la carpeta de src, ya que contiene las claves para acceder a firebase y al ser el repositorio piúblico, supone un problema de seguridad.

Se ha contruido una página web que nos permite ver y crear tanto amistoso como fanatics, ver nuestro datos y modificarlos, que un algoritmo nos de el siguiente partido de un torneo, y muchas funcionalidades que nos han servido para aprender mucho acerca de como funcionan los frameworks de javascript para la parte de front, en concreto React.js.

También la elección de una metodología que impulsa la creación de componentes que ayuda a la modularización, ha ayudado mucho sobre todo a la hora de eliminar complejidad. Esto es una gran ayuda sobre todo teniendo en cuenta que dejé de realizar el proyecto el año pasado, y a la hora de retomarlo este año no nos ha costado tanto al tener la aplicación modularizada.

A pesar de que el desarrollo de este proyecto es individual y no hemos podido trabajar de verdad con una metodología Scrum real, podemos ver las ventajas que supone esa división de tareas en historias de usuario, simplificando su desarrollo.

8.2. Trabajo futuro

Tenemos algunas tareas pendientes para la mejora de este proyecto:

- La creación de las ligas, que sería otro tipo de torneo junto con amistosos y fanatics, que tendría eliminatorias y clasificación por puntos.
- Crear una página web responsive, ya que finalmente no hemos podido modificar todos los componentes para adecuarlos a su uso desde dispositivo móvil.
- Crear un chat para que los jugadores puedan hablar entre ellos.

Apéndice A

Manual de usuario

En este capítulo vamos a explicar lo que el usuario puede realizar en la página.

Esta es la Pantalla inicial**??**, podemos realizar acciones en la barra del menú superior. En esta barra podemos ir a las siguientes ventanas:

- **Home:** nos lleva a la pantalla Home.
- **Amistosos:** 5.6 nos lleva a todos los amistosos creados.
- **Fanatics:** 5.7 nos lleva a los fanatics creados.
- **Iniciar Sesión:** 5.8 nos pide usuario y contraseña.
- **Registrarse:** 5.9 nos lleva a un formulario para rellenar los datos.
- **Mi cuenta:** en caso de haber iniciado sesión, sale este desplegable, los siguientes puntos salen de este desplegable.
- **Mis Credenciales:** 5.13 permite cambiar las credenciales.
- **Mi perfil:** 5.12 nos permite ver nuestros datos y actualizarlos.
- **Mis partidos:** 5.14 nos lleva una pantalla en la que vemos amistosos y fanatics.

Ahora vamos a ir recorriendo cada pantalla y explicando sus funcionalidades.

En la pantalla de amistosos, podemos realizar 2 acciones. Una es apuntarnos a un partido, en el que pulsaremos el botón de apuntarnos al amistoso que queramos, y nos pedirá una confirmación. En caso de estar ya apuntado o que no haya plazas, nos saltará una alerta para avisarnos.
La otra acción que podemos realizar en crear un amistos, pulsando el botón de Crear Amistos, en el que nos llevará a otra pantalla 5.10, en la que rellenaremos los datos y crearemos el partido.

En la pantalla de Fanatics, podremos realizar las mismas acciones que en el amistoso, tiene la misma funcionalidad.

Ahora vamos a ver la parte la parte de Mi Cuenta, en la que la funcionalidad de Mi perfil 5.12 y Mis credenciales 5.13 son la actualización o borrado del usuario y de sus credenciales. En el apartado de Mis Partidos 5.14, podemos ver los amistosos y los fanatics a los que estamos apuntados. En tendremos disponible un botón para desapuntarnos del amistoso tras pulsar el botón de Mis amistosos que aparece debajo, del fanatic no podemos desapuntarnos debido a que si ha empezado causaría muchos problemas.

Si pulsamos en Mis Fanatics, veremos que en cada Fanatic al que estamos apuntados, podemos ver la clasificación o planificar el siguiente partido pulsando respectivos botones.
En la clasificación, al pulsar el botón de obtener la clasificación 5.18, podemmos ver los puntos que lleva cada usuario. En el siguiente partido 5.20, podemos ver las 2 parejas que juegan, pudiendo elegir los 2 oponentes que juegan entre ellos con el desplegable, o pidiendo a la aplicación que nos sugiera oponentes dando al botón de Sugerir Partido.

Bibliografía

[1] *Algoritmo Emparejamiento*. URL: https://www.monografias.com/docs/Algoritmo-De-Fuerza-Bruta-FKCKDRAZMY.

[2] *Astah*. URL: https://software.com.co/p/astah-professional.

[3] *BaaS*. URL: https://blog.back4app.com/es/que-es-un-baas-backend-como-servicio/.

[4] *Crear Torneos Padel Doleague*. URL: https://www.doleague.com/organizar-torneos/.

[5] *Crear Torneos Padel Xporty*. URL: https://www.xporty.com/landing/padel.

[6] *Diagrama de Secuencia*. URL: https://www.lucidchart.com/pages/es/diagrama-de-secuencia.

[7] *Diseño Guiado por Componentes*. URL: https://adrianalonso.es/arquitectura-del-software/atomic-web-design-o-diseno-guiado-por-componentes/.

[8] *DOM React*. URL: https://es.reactjs.org/docs/faq-internals.html/.

[9] *DOM React*. URL: https://latteandcode.medium.com/y-eso-del-virtual-dom-de-react-qu%C3%A9-es-3feed6366925/.

[10] *El patrón Model-View-ViewModel*. URL: https://docs.microsoft.com/es-es/xamarin/xamarin-forms/enterprise-application-patterns/mvvm.

[11] *Firebase*. URL: https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarroladores-google-seo-sem/.

[12] *Firebase Authentication*. URL: https://firebase.google.com/docs/auth?hl=es-419.

[13] *Firestore Database*. URL: https://firebase.google.com/products/firestore?hl=es-419.

[14] *GIT*. URL: https://desarrolloweb.com/home/git.

[15] *LaTeX*. URL: https://www.latex-project.org/.

[16] *Modelo de Dominio*. URL: https://hmong.es/wiki/Domain_model.

[17] *NPM*. URL: https://openwebinars.net/blog/que-es-node-package-manager/.

[18] *Props React*. URL: https://es.reactjs.org/docs/render-props.html.

[19] *react Bootstrap*. URL: https://react-bootstrap.github.io/components/alerts/.

[20] *React js*. URL: https://es.reactjs.org/.

[21] *Router*. URL: https://reactrouter.com/docs/en/v6.

[22] *Scrum*. URL: https://www.scrum.org/.

[23] *Software como servicio*. URL: https://www.salesforce.com/es/learning-centre/tech/saas/.

[24] *Tabla de coeficientes de amortización lineal*. URL: https://www.agenciatributaria.es/AEAT.internet/Inicio/_Segmentos_/Empresas_y_profesionales/Empresas/Impuesto_sobre_Sociedades/Periodos_impositivos_a_partir_de_1_1_2015/Base_imponible/Amortizacion/Tabla_de_coeficientes_de_amortizacion_lineal_.shtml.

[25] *Trello*. URL: https://trello.com/tour.

[26] *Ventajas del patrón Model-View-ViewModel*. URL: https://docs.microsoft.com/es-es/windows/uwp/data-binding/data-binding-and-mvvm.

[27] *Ventajas SaaS*. URL: https://www.servnet.mx/blog/saas-que-es-y-cuales-son-sus-ventajas.