



Departamento de Bases de Datos
Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑERÍA INFORMÁTICA
MENCIÓN EN SISTEMAS DE INFORMACIÓN

PLAY2GETHER

**Plataforma colaborativa para la realización
de eventos deportivos informales.**

Estudiante: Sandra Dopico Cantarero
Director/a/es/as: Alejandro Cortiñas Álvarez
Miguel Ángel Rodríguez Luaces

A Coruña, 5 de junio de 2019.

A mis padres Arturo y Mar, espero que estéis orgullosos de mi

Agradecimientos

En primer lugar quería agradecer a mis tutores, Alejandro y Miguel, por la paciencia , la ayuda y la atención que me prestaron. Ellos siempre sabían como y cual era la mejor manera de solucionar los problemas.

A mi familia, por intentar ponerse en mi lugar cuando estaba de mal humor por que las cosas no salían como tenían que salir.

Y finalmente a mis compañeros de Universidad, Cristina, Laura, David y José, gracias por hacer que esta estapa de mi vida haya sido mucho más amena, las penas son menos cuando se comparten y a vuestro lado hasta se hacían divertidas.

Resumen

En cualquier comunidad, tanto jóvenes como adultos practican deportes en grupo de manera habitual. Sin embargo, en algunos de ellos se requiere un número importante de personas, por lo que es frecuente tener problemas para encontrar jugadores. Por ello surgió Play2Gether, una aplicación que permitirá a sus usuarios formar parte de una comunidad virtual centrada en organizar y jugar partidos de fútbol, tennis, padel, baloncesto...

Para alcanzar estos objetivos se gestionarán dos tipos de usuarios: administrador y usuarios normales. Los administradores serán los encargados de gestionar los usuarios, deportes y las localizaciones. Por otro lado, los usuarios podrán principalmente planificar eventos deportivos y unirse a partidos organizados.

En el desarrollo de la aplicación se utilizará Java, Hibernate y Spring-Boot para la parte del servicio, para la parte del cliente Vue.js, englobando HTML, CSS y JavaScript en un único archivo y finalmente PostgreSQL en cuanto a la parte de almacenamiento de la información. Este proyecto se gestionó siguiendo una metodología iterativa e incremental para el desarrollo de software.

Abstract

In any community, both young people and adults practice group sports on a regular basis. However, some of them require a significant number of people, so it is common to have problems finding players. This is why Play2Gether was born, an application that will allow its users to be part of a virtual community focused on organizing and playing soccer, tennis, padel, basketball games ...

To achieve these objectives, two types of users will be managed: administrator and normal users. Administrators will be responsible for managing users, sports and locations. On the other hand, users can mainly plan sporting events and join organized matches.

In the development of the application, Java, Hibernate and Spring-Boot will be used for the service part, for the Vue.js client part, including HTML, CSS and JavaScript in a single file and finally PostgreSQL in terms of the storage part of the information. This project was managed following an iterative and incremental methodology for software development.

Palabras clave:

Aplicación Web

Java

Vue.js

Spring Boot

Hibernate

PostgreSQL

Leaflet

Git

Deportes

Partidos

Jugadores

Red Social

Keywords:

Web Application

Java

Vue.js

Spring Boot

Hibernate

PostgreSQL

Leaflet

Git

Sports

Games

Players

Social Network

Índice general

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	1
2	Fundamentos tecnológicos	3
2.1	Estado del arte	3
2.1.1	Timpik	3
2.1.2	Sport-Meeting	4
2.1.3	Fubles	5
2.1.4	Conclusiones	6
2.2	Tecnologías utilizadas	7
2.2.1	Vue.js	7
2.2.2	Node.js	7
2.2.3	Cascading StyleSheets	8
2.2.4	HyperText Markup Language	8
2.2.5	JavaScript	8
2.2.6	Leaflet	9
2.2.7	Bootstrap	9
2.2.8	Java	9
2.2.9	Hibernate	9
2.2.10	Spring Boot	10
2.2.11	Git Lab	10
2.2.12	PostgreSQL	10
3	Metodología y planificación	11
3.1	Metodología de desarrollo	11
3.1.1	Fase preliminar	11
3.1.2	Iteraciones	12

3.1.3	Herramientas Utilizadas	12
3.2	Planificación y seguimiento	13
3.2.1	Tareas	13
3.2.2	Recursos	14
3.2.3	Planificación	14
3.2.4	Costes	14
3.2.5	Seguimiento	15
4	Análisis	17
4.1	Actores	17
4.2	Requisitos por Módulos	18
4.3	Requisitos por Actores	20
4.3.1	Usuario Anónimo	20
4.3.2	Usuario Administrador	20
4.3.3	Usuario Básico, Creador y Jugador	22
4.4	Arquitectura del sistema	26
4.5	Interfaz de usuario	26
4.5.1	Componentes y Navegación	28
4.6	Modelo conceptual de datos	28
4.6.1	User	33
4.6.2	UserAuthority	33
4.6.3	NormalUser	34
4.6.4	AdminUser	34
4.6.5	SocialRelationship	34
4.6.6	SocialFriendship	34
4.6.7	SocialBlock	34
4.6.8	Team	35
4.6.9	Sport	35
4.6.10	Location	35
4.6.11	Game	35
4.6.12	Player	36
4.6.13	Player Valoration	36
4.6.14	Comment	36
4.6.15	UserMessage	36
4.6.16	GameMessage	36

ÍNDICE GENERAL

5 Diseño	37
5.1 Arquitectura tecnológica del sistema	37
5.2 Diseño de la aplicación	38
5.3 Back-end	38
5.3.1 DAOs	39
5.3.2 Servicios	40
5.3.3 Controladores	43
5.3.4 Patrones	51
5.4 Front-end	53
5.4.1 Componentes	54
5.4.2 Rutas	54
5.4.3 Comunicación con el servidor	55
5.4.4 Patrones	61
6 Implementación y pruebas	63
6.1 Implementación	63
6.1.1 Recomendación de Partidos	63
6.1.2 Motor de Renderizado de pantallas	67
6.2 Pruebas	69
6.2.1 Pruebas Unitarias	69
6.2.2 Pruebas REST	69
6.2.3 Pruebas de integración y aceptación	70
7 Solución desarrollada	71
7.1 Acceso a la Aplicación	71
7.2 Administración	71
7.3 Usuarios Básicos	75
8 Conclusiones y trabajo futuro	79
8.1 Conclusiones	79
8.2 Trabajo Futuro	79
A Glosario de acrónimos	83
B Manual de Instalación	85
C Contenido del CD	87
D Mockups	89

Índice de figuras

2.1	Logo de TimPik	3
2.2	Pantalla de búsqueda de eventos deportivos	4
2.3	Pantalla de detalle de un partido	4
2.4	Logo de Sport Meeting	4
2.5	Pantalla de registro en la aplicación	5
2.6	Pantalla de búsqueda de eventos deportivos	5
2.7	Logo de Fubles	6
2.8	Pantalla de búsqueda de eventos deportivos	6
2.9	Pantalla de detalle de un partido	7
2.10	Logo Vue.js	7
2.11	Logo Node.js	8
2.12	Logo css	8
2.13	Logo HTML	8
2.14	Logo JavaScript	8
2.15	Logo Leaflet	9
2.16	Logo Bootstrap	9
2.17	Logo Java	9
2.18	Logo Hibernate	10
2.19	Logo Spring Boot	10
2.20	Logo Git Lab	10
2.21	Logo PostgreSQL	10
3.1	Metodología Incremental	11
3.2	Planificación inicial	15
3.3	Planificación final	16
3.4	Diagrama de Gantt planificación	16
4.1	Relaciones de herencia entre actores	18

4.2 Caso de Uso: Usuario Anónimo	21
4.3 Caso de Uso: Usuario Administrador	21
4.4 Caso de Uso Crear Partido: Usuario Básico	22
4.5 Caso de Uso Gestión Partidos: Usuario Básico/Creador/Jugador	23
4.6 Caso de Uso Gestión Valoraciones: Usuario Jugador	24
4.7 Caso de Uso Gestión Perfiles: Usuario Básico	25
4.8 Caso de Uso Gestión Social: Usuario Básico	25
4.9 Arquitectura del sistema	26
4.10 Patrón Model-View-ViewModel	27
4.11 Estructura Componente vue	28
4.12 Diagrama de Clases	33
5.1 Arquitectura tecnológica del sistema	37
5.2 Diagrama PlayerDAO	39
5.3 User Service	40
5.4 Sport Service	41
5.5 Team Service	41
5.6 SocialRelationShip Service	41
5.7 PlayerValorationService Service	42
5.8 Game Service	42
5.9 Comment Service	42
5.10 Location Service	43
5.11 Player Service	43
5.12 User Resource	44
5.13 Sport Resource	45
5.14 Team Resource	45
5.15 Social Resource	46
5.16 PlayerValoration Resource	47
5.17 Game Resource	47
5.18 Comment Resource	48
5.19 Location Resource	49
5.20 Player Resource	49
5.21 Account Resource	50
5.22 Diagrama del Patrón DAO	51
5.23 Diagrama MVC Spring	52
5.24 Diagrama MVC Spring	53
5.25 Diagrama de componentes del Cliente	54
5.26 Comunicación de los componentes de Administración del Cliente al Servidor . .	56

ÍNDICE DE FIGURAS

5.27 Comunicación de los componentes del Usuario del Cliente al Servidor	57
5.28 Comunicación de los componentes del Usuario del Cliente al Servidor	58
5.29 Comunicación de los componentes del Usuario del Cliente al Servidor	59
5.30 Comunicación de los componentes comunes del Cliente al Servidor	60
5.31 Patrón Promise	61
5.32 Patrón Callback	62
6.1 Algoritmo de Recomendación: Jugadores	63
6.2 Algoritmo de Recomendación: Jugadores	64
6.3 Algoritmo de Recomendación: Deportes	65
6.4 Algoritmo de Recomendación: Localizaciones	66
6.5 RecomendacionDTO	66
6.6 Componentes de visualización de resultado según deporte	67
6.7 Cambio de plantilla según valor del componente de entrada	67
6.8 Resultado partido de Tennis (JSON)	68
6.9 Resultado partido de Fútbol (JSON)	68
6.10 Funcionamiento Motor Plantillas	69
7.1 Página de Login	72
7.2 Página de Registro	72
7.3 Página para dar de alta un deporte	73
7.4 Página para editar un deporte	73
7.5 Página para visualizar una localización	74
7.6 Página para dar de alta una localización	74
7.7 Página para dar de baja usuarios	75
7.8 Página de búsqueda de partidos: Mapa	76
7.9 Página de búsqueda de partidos: Calendario	76
7.10 Página del Perfil Público	77
7.11 Página del Tablón de Actividades	77

Índice de cuadros

3.1	Costes de los recursos humanos	15
4.1	Componentes de las pantallas del Administrador y Navegación	29
4.2	Componentes de las pantallas del Usuario y navegación	29
4.3	Componentes de las pantallas de la aplicación y navegación	30
4.4	Componentes de las pantallas del Usuario y navegación	31
4.5	Componentes de las pantallas del Usuario y navegación	32
4.6	Componentes Comunes a todos los usuarios	32
5.1	Acceso a UserResource	44
5.2	Acceso a SportResource	45
5.3	Acceso a TeamResource	46
5.4	Acceso a SocialResource	46
5.5	Acceso a PlayerValorationResource	47
5.6	Acceso a GameResource	48
5.7	Acceso a CommentResource	48
5.8	Acceso a LocationResource	49
5.9	Acceso a PlayerResource	50
5.10	Acceso a AccountResource	50
5.11	Rutas de los componentes del cliente	55

Capítulo 1

Introducción

1.1 Motivación

La motivación que llevo a la realización de este proyecto fue la desesperación cada fin de semana de los amigos de la autora de no encontrar personas suficientes para poder realizar un partido de fútbol. Además observaba como todos ellos realizaban una búsqueda exhaustiva de jugadores a través de las redes sociales y en su mayoría todos esos esfuerzos no obtenían resultados.

El objetivo principal es facilitar la planificación de estas actividades deportivas proporcionando una comunidad virtual donde convivan usuarios con las mismas necesidades deportivas.

1.2 Objetivos

Este Trabajo de Fin de Grado consiste en el desarrollo de una aplicación web para todos aquellos usuarios interesados en participar en eventos deportivos en grupo. Los objetivos principales que debe cumplir son los siguientes:

- **Facilitar la planificación de los partidos.** En el momento de creación de un partido, se tendrá acceso al pronóstico metereológico por si se ha escogido una localización al aire libre. Una vez creado, otros usuarios de la aplicación podrán apuntarse hasta alcanzar un número máximo de participantes. Del mismo modo podrá cancelarse si no se llega al mínimo de jugadores.
- **Conseguir participantes de una manera más sencilla.** Uno de los problemas de los deportes en grupo es la dificultad de encontrar jugadores que tengan disponibilidad. Al

publicar el partido en nuestra plataforma todas aquellas personas interesadas en él se apuntarán y pasarán a formar parte de él sin necesidad de ponernos en contacto con ellos, por lo que el esfuerzo de búsqueda de participantes acaba desapareciendo.

- **Conseguir que los usuarios encuentren lo que buscan.** En primer lugar la plataforma ofrece 2 formas diferentes de visualizar los partidos vigentes: en mapa o en calendario. En mapa, el usuario elige los partidos en función de la localización de estos, a parte puede filtrar por creador del partido, tipo de deporte, por la media de edad de los jugadores e incluso por dificultad, la experiencia que poseen los jugadores apuntados. En calendario, por si se prioriza la disponibilidad horaria. También nos recomendará partidos a los usuarios en función de su histórico, ya sea por deporte, localización o jugadores.
- **Adaptar la aplicación para que la adición de nuevos deportes sea lo más sencillo posible.** La aplicación actualmente ofrece la posibilidad de crear partidos de fútbol, baloncesto, tennis y padel, pero en un futuro se podrá pensar en añadir otro nuevo deporte por lo que esta transición ha de ser lo más sencilla posible. Para ello se asociará a cada deporte sus plantillas de "Relleno de Resultados" y "Visualización de Resultados" y se guardará el resultado del partido en un JSON. De esta forma cuando creemos un nuevo deporte, solo habría que crear las plantillas y asociar ese deporte a esas plantillas y cada plantilla dependiendo del deporte que sea tratará ese JSON de una manera u otra.
- **Servir de canal de comunicación entre jugadores y creadores.** Tenemos a nuestra disposición un espacio único para el partido creado, es decir, cualquier duda referida al partido podrá ser realizada en el propio foro del partido, comunicándonos por mensaje directo con otros participantes o el propio creador, y en caso de cancelación se avisará por correo a todos los jugadores, evitando así la necesidad de tener que comunicarnos por aplicaciones externas.

Capítulo 2

Fundamentos tecnológicos

2.1 Estado del arte

En este apartado se estudiarán las aplicaciones web del sector, centrándose en los objetivos citados anteriormente (sección 1.2).

2.1.1 Timpik

Timpik es una herramienta para poder realizar quedadas. Su mayor ventaja es la gran variedad de deportes que ofrece. El organizador al crear un deporte puede detallar las sanciones o el precio por jugador. Este coste se abonará a través del monedero virtual.



Figura 2.1: Logo de TimPik

En su página web da la posibilidad de organizar un partido, de buscar eventos, buscar jugadores a través de un correo electrónico o Facebook y una funcionalidad a destacar, crear un club para organizar partidos con tu grupo de amigos de una manera más rápida y directa. La búsqueda de eventos se realiza a través de un calendario [1]. Te permite filtrar por tus deportes favoritos o por provincia. En la pantalla del evento aparece información básica del partido junto con los jugadores y un apartado de comentarios.

En el muro público de un jugador se observar el número de partidos jugados, organizados, deportes que practica, clubs a los que pertenece, medallas conseguidas...

2.1. Estado del arte

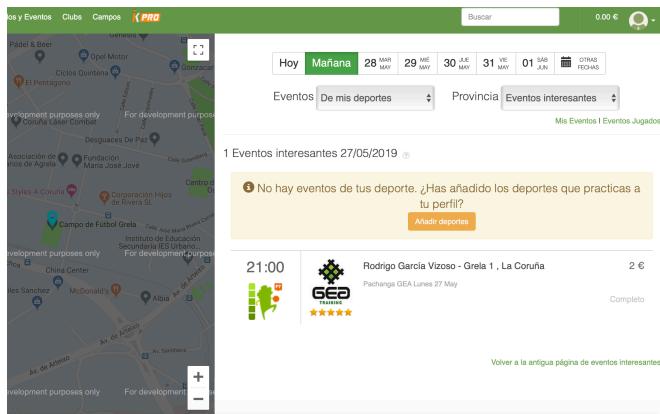


Figura 2.2: Pantalla de búsqueda de eventos deportivos

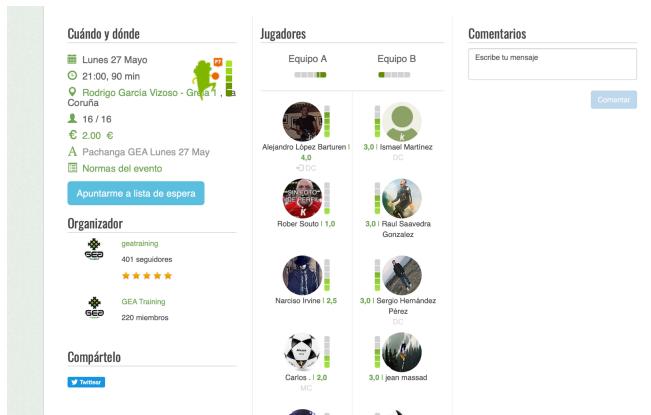


Figura 2.3: Pantalla de detalle de un partido

2.1.2 Sport-Meeting

Esta aplicación Web es ofrecida por Decathlon. Cuenta con perfiles específicos para que asociaciones y clubes deportivos puedan utilizar esta red social como medio de difusión de los eventos que organicen.



Figura 2.4: Logo de Sport Meeting

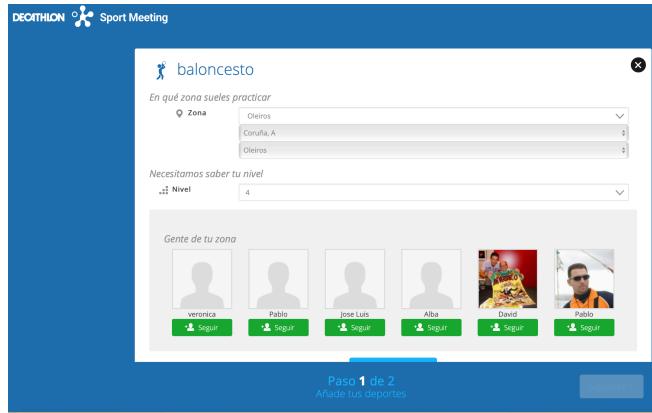


Figura 2.5: Pantalla de registro en la aplicación

Al iniciar sesión en su página web nos permite configurar nuestros deportes favoritos y en que zona queremos practicarlos. En función de la zona recomendará usuarios y eventos. En cuanto a la búsqueda de eventos deportivos se disponen en una lista ordenados por fecha de creación, pudiéndose filtrar por zona, nivel o fecha

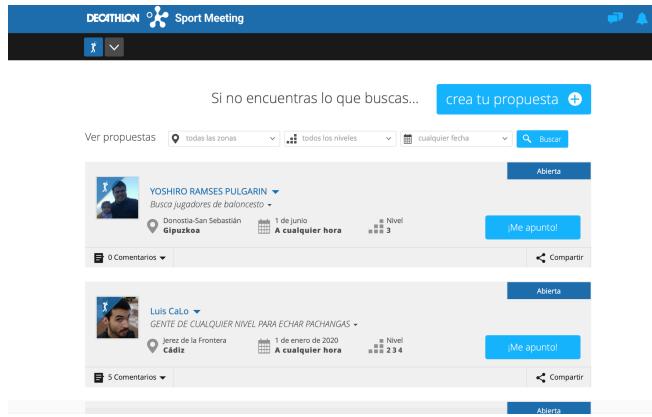


Figura 2.6: Pantalla de búsqueda de eventos deportivos

En el perfil del usuario se pueden ver las propuestas creadas y realizadas. Aparte como comentamos anteriormente esta plataforma esta creada al estilo red social y es por ello que desde la propia app se anima a los usuarios a que suban sus fotos practicando deporte.

2.1.3 Fubles

El deporte principal es el fútbol. El registro en su página web y el funcionamiento es parecido al de Timpik, pero con la diferencia de que los pagos suelen abonarse en efectivo.

Hay que destacar que permite a los usuarios nombrar “procurador” a alguno de sus contactos, dándole permisos para gestionar su inscripción o cancelación en las quedadas.



Figura 2.7: Logo de Fubles

La búsqueda de eventos es a través de un mapa. El filtrado es más avanzado que en el resto de aplicaciones, nos permite filtrar por pabellones, espacios con tragamonedas, eventos deportivos que tengan lugar en ubicaciones de interior.

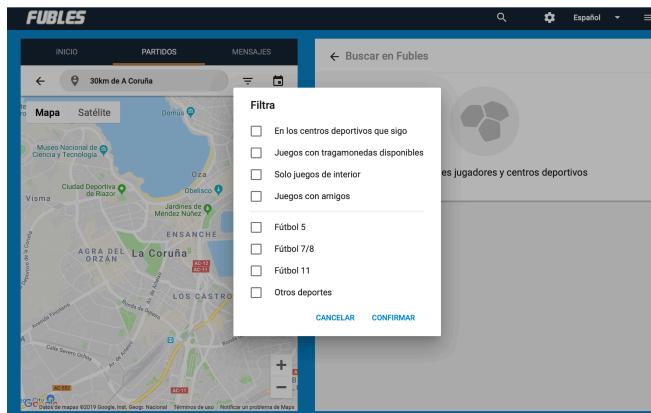


Figura 2.8: Pantalla de búsqueda de eventos deportivos

En el perfil del usuario se pueden ver los datos básicos de cada jugador más los partidos jugados, creados, las veces que ha sido votado mejor jugador en el partido, los goles, las veces que no ha aparecido a un partido finalmente a pesar de estar apuntado...

En la pantalla del evento concreto aparece la información básica del partido junto con los jugadores apuntados, las valoraciones, los resultados finales y un apartado donde la gente puede dejar comentarios acerca del partido.

2.1.4 Conclusiones

Se intentará englobar características y añadir funcionalidades nuevas para alcanzar los objetivos citados en el apartado 1.2. Para lograr ser **un canal de comunicación entre usuarios**.

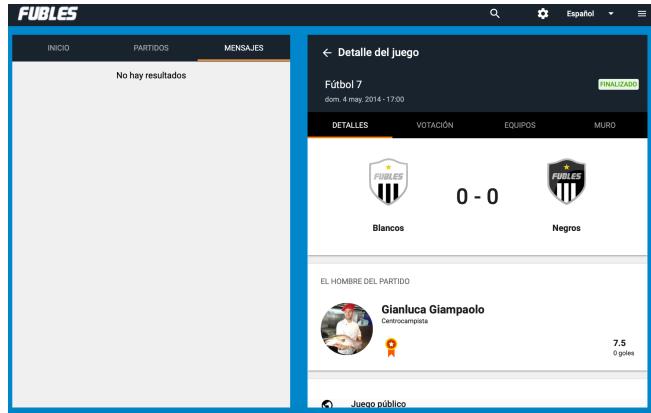


Figura 2.9: Pantalla de detalle de un partido

rios se tomará como referencia las 3 aplicaciones en cuanto a mensajes directos, peticiones de amistad... y se añadirán funcionalidades nuevas. La notificación de las actividades que realiza un usuario, y el acceso a un tablón de actividades de la gente a la que seguimos. Para **conseguir que los usuarios encuentren lo que buscan**, la mejor app es Fubles [2.1.3], ya que posee filtros más avanzados. Añadiremos nuevas funcionalidades como la búsqueda por media de edad y la recomendación de partidos. Sport Meeting [2.1.2] ofrece esta funcionalidad durante el registro aunque en función de la zona escogida, mientras que en Play2gether se basa en el histórico.

2.2 Tecnologías utilizadas

2.2.1 Vue.js

Vue [2],[3],[4],[5],[6],[5] es un framework progresivo para construir interfaces de usuario. Su núcleo es bastante pequeño y se escala a través de plugins. Engloba en un mismo archivo HTML, CSS y JavaScript.



Figura 2.10: Logo Vue.js

2.2.2 Node.js

Node.js[7] es una plataforma de desarrollo para la creación de aplicaciones destinadas a la Web, orientada a redes y centrada en la velocidad y la escalabilidad.



Figura 2.11: Logo Node.js

2.2.3 Cascading StyleSheets

CSS (Cascading StyleSheets) es un lenguaje de estilo para describir el diseño de una página web (colores, disposición de elementos y fuentes). Aunque es independiente de HTML (HyperText Markup Language) suele emplearse conjuntamente.



Figura 2.12: Logo css

2.2.4 HyperText Markup Language

HTML (HyperText Markup Language) es el lenguaje de marcado o lenguaje de marcas para describir la estructura de páginas web.



Figura 2.13: Logo HTML

2.2.5 JavaScript

JavaScript es el lenguaje utilizado en páginas Web para incrementar la funcionalidad de las mismas y la interacción con el usuario.



Figura 2.14: Logo JavaScript

2.2.6 Leaflet

Leaflet.js [8] es una librería JavaScript open source para creación de mapas y visualización de datos geográficos.



Figura 2.15: Logo Leaflet

2.2.7 Bootstrap

Bootstrap[9] es un conjunto de herramientas para desarrollo con HTML, CSS y JavaScript que ofrece variables Sass con las que personalizar la interfaz.



Bootstrap

Figura 2.16: Logo Bootstrap

2.2.8 Java

Se trata de un lenguaje de programación interpretado (comunica a través de un intérprete). Posee varias cualidades: Orientado a objetos, multiplataforma, multihilo, orientación a la red...



Figura 2.17: Logo Java

2.2.9 Hibernate

Hibernate [10],[11],[12] es un framework ORM (Object-Relational Mapping), es decir, una herramienta para realizar una correspondencia entre objetos y las tablas de una base de datos relacional.



Figura 2.18: Logo Hibernate

2.2.10 Spring Boot

Spring Boot [13],[14] es un framework que facilita la configuración de un proyecto Spring, convirtiéndola en automática.



Figura 2.19: Logo Spring Boot

2.2.11 Git Lab

GitLab es un software en forma de interfaz web que permite el trabajo de varios colaboradores en un repositorio y una fácil gestión de equipos de desarrolladores.



Figura 2.20: Logo Git Lab

2.2.12 PostgreSQL

PostgreSQL [15] es un sistema de gestión de base de datos relacional orientado a objetos. Destacar el control de concurrencias multiversión (MVCC) y el uso de bloqueos de lectura.



Figura 2.21: Logo PostgreSQL

Capítulo 3

Metodología y planificación

3.1 Metodología de desarrollo

En este proyecto se ha optado por utilizar una metodología iterativa e incremental, además de estar dirigida por las funcionalidades del sistema. Cada iteración ha supuesto un incremento en la funcionalidad hasta alcanzar un producto entregable.

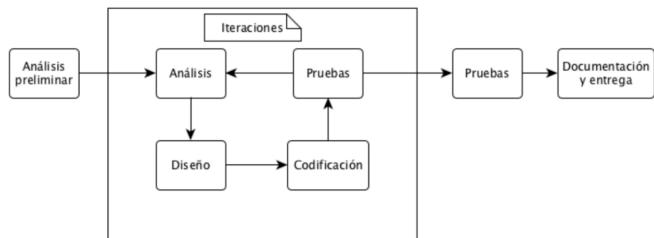


Figura 3.1: Metodología Incremental

Aunque esta metodología tenga ciertas similitudes con Scrum, no es la misma ya que el equipo de desarrollo solo cuenta con una persona, la autora, no existe el rol de Scrum Máster y únicamente se realiza un tipo de reuniones que podría considerarse una combinación entre Planificación del Sprint y Revisión del Sprint.

3.1.1 Fase preliminar

Lo primero que se consideró realizar fué el estudio de los objetivos a cumplir por la aplicación. Una vez decidido a donde queríamos llegar, comenzó el estudio de qué tecnologías eran las más adecuadas para utilizar en su desarrollo para alcanzar estos objetivos. Así, se realizó

una búsqueda y análisis de diferentes herramientas, de las cuales se escogieron las citadas en el apartado de Tecnologías Utilizadas[2.2]

Finalmente después de haber estudiado el objetivo global del proyecto era necesario empezar a pensar en los objetivos individuales que deberá alcanzar cada iteración. Para ello previamente se construyó un punto de partida que consistía en:

- **Modelo de casos de uso:** Sus funcionalidades serán implementadas en sus respectivas iteraciones. Gracias a él onseguiremos definir el comportamiento del sistema
- **Mockups:** Se crearán maquetas de como serían las interfaces de la aplicación. Servirán, junto con el modelo de casos de uso, para planificar las iteraciones.
- **Modelo de datos:**Este modelo definirá los diferentes datos que debe manejar el sistema y las distintas relaciones que existirán entre ellos.

3.1.2 Iteraciones

Cada una de las iteraciones que se han llevado a cabo incluye las siguientes fases:

- **Análisis:** Se analizará la iteración teniendo claro los nuevos requisitos, y analizando las diferentes formas de alcanzar los objetivos.
- **Diseño:** Se definirán los distinto módulos de manera cuidadosa ya que estos podrán ser reutilizados en iteraciones posteriores.
- **Desarrollo:** Se implementará lo definido en la fase anterior, definiéndose las entidades y métodos para poder llevar a cabo los casos de uso definidos.
- **Pruebas:** Se realizarán primero pruebas individuales y luego globales de las funcionalidades creadas, las cuales comprobarán que lo realizado cumple con las expectativas marcadas y posee un comportamiento acertado.

3.1.3 Herramientas Utilizadas

- **Latex:** [16] Es un sistema de composición de textos.Se ha utilizado para la realización de esta memoria.
- **Postman:** [17] Herramienta que permite realizar peticiones HTTP a cualquier API para comprobar que todas nuestras conexiones REST funcionaban correctamente.
- **Eclipse:** [18] No es más que un entorno de desarrollo integrado (IDE). Se utilizó para la codificación de la capa de servicio.

- **Sublime:** [19] Es un editor de código multiplataforma y ligero. Fue utilizado para la codificación de la capa cliente.
- **MagicDraw:** [20] Herramienta CASE compatible con el estándar UML. Se ha utilizado en este proyecto para la elaboración de los diferentes diagramas de clases y casos de uso.
- **Balsamiq Mockups:** [21] Es una herramienta que permite diseñar maquetas de interfaz. Se ha utilizado para desarrollar una primera idea de como serían las pantallas de la aplicación.
- **Draw.io:** [22] Es una aplicación web que nos permite crear diagramas de todo tipo desde nuestro navegador.

3.2 Planificación y seguimiento

En este apartado se mostrará una planificación inicial del proyecto y se hará una estimación de este. Por último se tendrán en cuenta las desviaciones producidas y el coste.

3.2.1 Tareas

El trabajo se divide en 4 iteraciones de desarrollo con una duración de entre 3 y 4 semanas, dependiendo de la carga de trabajo y de la disponibilidad de la autora, y una iteración final de pruebas y elaboración de memoria.

- **Primera Iteración:** En la primera iteración se realizaría todo lo referente a la autenticación de los usuarios (Inicio de Sesión, Registro...) y lo que sería el Módulo de Administración (Gestión de Deportes, Localizaciones y Usuarios)
- **Segunda Iteración:** Se centraría en los partidos, es decir, es su creación, las diferentes formas de mostrarle a los jugadores los partidos disponibles, la gestión de jugadores...
- **Tercera Iteración:** Aquí se trabajarían los temas de valoraciones, comentarios, relleno de resultados, renderizado de pantallas según el deporte, la recomendación de partidos según su histórico...
- **Cuarta Iteración:** Nos centraríamos en la parte social del proyecto, es decir, mensajes directos, foros, seguidores, notificaciones, tablón de actividades...
- **Quinta Iteración:** Por último, se pasó a realizar las diferentes pruebas finales para comprobar que la aplicación ofrecía las funcionalidades deseadas de una manera correcta y a redactar la memoria del TFG.

3.2.2 Recursos

En cuanto a los recursos humanos para este proyecto podemos diferenciar 2 roles diferentes: Directores y Analista/Programador.

- **Directores:** En este caso se encargaban de la planificación y supervisión tanto del proyecto global, como de las iteraciones. En cuanto a la comunicación entre los directores y la autora se han realizado reuniones de seguimiento cada 4 semanas, dependiendo de la duración de la iteración y de la disponibilidad de los asistentes, en las que se planifica qué realizar en la siguiente iteración y se revisa lo anterior.
- **Analista/Programador:** Este sería el rol de la autora, se encargaría del análisis, diseño, desarrollo y pruebas de cada una de las iteraciones a realizar.

3.2.3 Planificación

Se ha realizado un Diagrama de Gantt exponiendo como tareas las iteraciones mencionadas en el apartado [3.2.1]. Las fechas de fin e inicio del proyecto correspondería con el 28 de Noviembre de 2018 y el 21 de junio de 2019.

Se debe mencionar que durante el análisis preliminar se interrumpió el proyecto en 2 ocasiones, aunque estas paradas estaban previstas desde un principio, por lo que no afectó a la planificación del proyecto:

- **Del 8 al 21 de Enero :** La causa fue la preparación de los exámenes del primer cuatrimestre.
- **Del 1 al 3 de Febrero:** Por motivos personales de la autora.

3.2.4 Costes

La autora ha trabajado **5h de media los lunes, martes y jueves y 6h de media los sábados y domingos**, lo que suponen **27 horas semanales**. Teniendo en cuenta que la duración del proyecto asciende a **205 días totales**, el total de horas ascenderían a **790 horas**. Por otra parte, los directores del proyecto dedicaron aproximadamente **40 horas** entre reuniones y la revisión de la memoria.

- **Hora de trabajo del Director:** 26,69€/h
- **Hora de trabajo del Analista/Programador:** 19,06€/h

CAPÍTULO 3. METODOLOGÍA Y PLANIFICACIÓN

Etapas proyecto	Fecha Inicio	Fecha Fin
Estudio de la Tecnología	28 Noviembre	8 Febrero
Análisis Preliminar	28 Noviembre	8 Febrero
Funcionalidad		
Prototipos de Pantalla		
Diagrama conceptual BD		
Servicio REST		
Planificación	7 Febrero	8 Febrero
Dividir en Iteraciones		
Desarrollo 1 Iteración	8 Febrero	8 Marzo
Registro de Usuarios		
Inicio Sesión		
Perfiles de Usuario		
Gestión de Deportes		
Gestión de Localizaciones		
Gestión de Usuarios		
Desarrollo 2 Iteración	8 Marzo	5 Abril
Gestión de Partidos		
Gestión de Jugadores		
Meteorología		
Mapas y Calendarios		
Filtrado de Partidos		
Desarrollo 3 Iteración	5 Abril	3 Mayo
Notificaciones de partidos		
Relleno de Resultados		
Visualización de Resultados		
Valoraciones jugadores		
Valoraciones partidos		
Partidos Recomendados		
Desarrollo 4 Iteración	3 Mayo	31 Mayo
Foro del partido		
Mensajería		
Solicitudes de amistad		
Notificación de usuarios		
Tablón de Actividades		
Memoria	31 Mayo	21 Junio

Figura 3.2: Planificación inicial

Actor	Horas/persona
Sandra Dopico Cantarero	790
Alejandro Cortiñas Álvarez	40
Miguel Ángel Rodríguez Luaces	40

Cuadro 3.1: Costes de los recursos humanos

Estos costes se han obtenido de la siguiente manera:

- **Director del Proyecto:** Suponemos que el salario bruto sería de unos 35.000€/año, pero si tenemos en cuenta lo que la empresa paga a la seguridad social por ese trabajador(35%) nos quedan 26.500€/año, dividiendo entre 1700 horas trabajadas nos quedan 15.588€/mes, dividiendo entre 160 horas trabajadas nos quedan 97.43€/hora.
- **Analista/Programador:** Se parte de un salario bruto de 25.000€ y se calcularía de la misma forma, quedaría en uno 33.750€, y haciendo el cálculo obtendríamos esos 19,06€/h.

3.2.5 Seguimiento

Durante la realización del proyecto se produjeron una serie de cambios en la duración de algunas tareas casi imperceptibles.

3.2. Planificación y seguimiento

Pero como se puede observar los retrasos producidos durante el proyecto fueron compensados en iteraciones posteriores, por lo que el esfuerzo final para la realización del proyecto no ha variado. Hay que objetar que únicamente se incluyó el Diagrama de Gantt de la planificación final debido a la escasa diferencia respetto a la planificación inicial.

Etapas proyecto	Fecha Inicio	Fecha Fin
Estudio de la Tecnología	28 Noviembre	8 Febrero
Análisis Preliminar	28 Noviembre	8 Febrero
Funcionalidad		
Prototipos de Pantalla		
Diagrama conceptual BD		
Servicio REST		
Planificación	7 Febrero	8 Febrero
Dividir en Iteraciones		
Desarrollo 1 Iteración	8 Febrero	10 Marzo
Registro de Usuarios		
Inicio Sesión		
Perfiles de Usuario		
Gestión de Deportes		
Gestión de Localizaciones		
Gestión de Usuarios		
Desarrollo 2 Iteración	10 Marzo	3 Abril
Gestión de Partidos		
Gestión de Jugadores		
Meteorología		
Mapas y Calendarios		
Filtrado de Partidos		
Desarrollo 3 Iteración	5 Abril	4 Mayo
Notificaciones de partidos		
Relleno de Resultados		
Visualización de Resultados		
Valoraciones jugadores		
Valoraciones partidos		
Partidos Recomendados		
Desarrollo 4 Iteración	6 Mayo	25 Mayo
Foro del partido		
Mensajería		
Solicitudes de amistad		
Notificación de usuarios		
Tablón de Actividades		
Memoria	26 Mayo	21 Junio

Figura 3.3: Planificación final

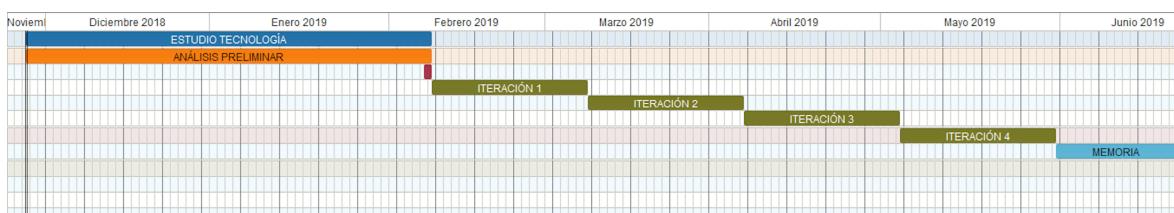


Figura 3.4: Diagrama de Gantt planificación

Capítulo 4

Análisis

En este apartado de la memoria nos centraremos en el análisis de la aplicación, describiendo los actores y requisitos, la arquitectura del sistema y los datos mediante un Diagrama de Clases.

4.1 Actores

Un actor es una entidad que participa en un caso de uso, y que puede tener relaciones de herencia con otros actores. En este apartado se mostrán los 5 actores presentes en la aplicación.

- **Usuario Anónimo:** Sólo podrá iniciar sesión o registrarse en la app.
- **Administrador:** Se encarga de la gestión de Deportes, Localizaciones y Usuarios.
- **Usuario Básico:** Tiene acceso a prácticamente todas las funcionalidades de la aplicación, excepto a las de gestión del Administrador y las funcionalidades añadidas que tienen los 2 siguientes actores.
- **Usuario Creador:** Es la persona responsable de la creación de un partido. Engloba todas las funcionalidades del Usuario Básico solo que tendrá funcionalidades añadidas sobre el partido que ha creado.
- **Usuario Jugador:** Es la persona que ha jugado un determinado partido. Engloba todas las funcionalidades del Usuario Básico y además este tendrá funcionalidades añadidas sobre el partido que ha jugado y sobre los demás jugadores del partido.

En resumen se pueden diferenciar 3 usuarios principales, que son el Usuario Anónimo, Administrador y Usuario Básico. Luego un Usuario Básico a mayores puede ser Usuario Creador y Usuario Jugador, con funcionalidades añadidas.

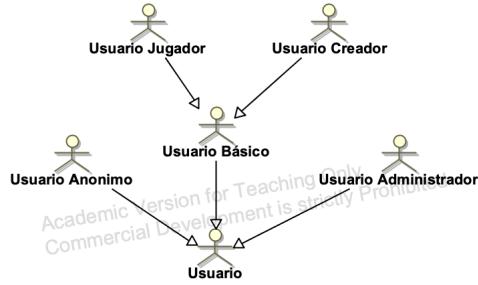


Figura 4.1: Relaciones de herencia entre actores

4.2 Requisitos por Módulos

Uno de los pasos que debemos realizar antes del desarrollo para describir el comportamiento del sistema es la Especificación de Requisitos (ERS). A continuación se detallan los distintos requisitos finales divididos por los módulos principales

Autentificación

- Es obligatorio que todos los usuarios se autentiquen para poder llevar a cabo cualquiera de las funcionalidades que ofrece la aplicación.
- Los Usuarios Anónimos podrán registrarse o loguearse, mientras que el Usuario Básico y el Administrador podrán hacer logout.

Deportes

- El Administrador podrá dar de alta deportes.
- Se podrán visualizar los detalles del deporte.
- Se podrán editar las localizaciones donde se juega ese deporte, el ícono del deporte y los componentes de entrada y visualización del deporte necesarios para el renderizado de pantallas en la parte del cliente del formulario de los resultados del deporte y sus detalles finales.

Localizaciones

- El Administrador podrá dar de alta localizaciones a través de un mapa.
- Se podrán visualizar los detalles de la localización.

- Se podrá editar el nombre de la localización y las coordenadas de esta a través del mapa.

Equipos

- Un Usuario Básico podrá dar de alta en el sistema nuevos equipos externos.

Usuarios

- El Administrador podrá eliminar usuarios de la aplicación.
- Cada usuario podrá ver sus datos personales.
- Cada usuario podrá editar cualquier información personal suya, la foto de perfil, la contraseña y los equipos a los que pertenece externamente fuera de la aplicación.
- Podrán buscar a otros usuarios gracias a su nombre y apellidos.
- En el perfil de cada uno se podrá observar tanto los próximos partidos que va a jugar, como los organizados, jugados, comentarios de otros jugadores sobre él y los partidos que la aplicación te recomendará según los partidos que jugaste en un pasado. Estas recomendaciones serán en base a los deportes, ubicaciones o usuarios con los que jugastes y poseen una buena valoración.
- Para cada usuario se calculará la experiencia media que posee en función de las valoraciones que ha recibido en anteriores partidos.
- Tendrán la posibilidad de empezara seguir a otros usuarios, por lo que cada usuario de la aplicación tendrá una lista de seguidores y seguidos.
- Podrá enviar mensajes directos.
- Observar las actividades de sus seguidos en un Tablón, donde se notificará cuando alguien que sigue crea un partido, se apunta, se desapunta...

Partidos

- Se podrán crear partidos
- Se podrán visualizar los partidos tanto en un mapa como en un calendario [1].
- Se podrán filtrar partidos.
- Se podrá ver el detalle del partido, es decir, su información básica, los jugadores apuntados a él y hasta su información metereológica [23].

- Se podrá cancelar un partido.
- Marcar un partido para recibir notificaciones sobre él, por ejemplo quién se apunta al partido, quién se desapunta...
- Compeltar los resultados finales del partido por parte del creador.
- Todos los jugadores de un partido podrán valorar el partido una vez finalizado, dándole una nota del 1 al 5.
- Cualquier usuario Básico podrá comentar en el foro de un partido.

Jugadores

- Se crearán jugadores una vez que un usuario se apunte a un partido.
- Se podrán valorar jugadores. La valoración consistirá en una nota del 1 al 5 y un comentario.

4.3 Requisitos por Actores

En este apartado utilizaremos los casos de uso, también llamados requisitos funcionales para describir la secuencia de interacciones entre un actor y el sistema. En nuestro caso disponemos de los actores citados en el apartado 4.1

4.3.1 Usuario Anónimo

Es aquel usuario que accede a la página Web y solo tiene acceso a la página para **loguearse** o **registrarse**.

4.3.2 Usuario Administrador

Este actor es el encargado de gestionar la aplicación, este por defecto lo creamos nosotros al dar de alta el sistema. Este administrador cuenta con actividades de gestión y no tiene acceso a ninguna de las funcionalidades de un Usuario Básico. Para acceder debemos ingresar para con un login y password de un Usuario Administrador.

- **Gestión de Deportes:** El administrador podrá dar de alta nuevos deportes en el sistema como darlos de baja y editar los existentes, tanto las localizaciones donde se puede jugar a ese deporte, como los nombres de los componentes de visualización de los deportes en la parte cliente, como el icono de este.

CAPÍTULO 4. ANÁLISIS

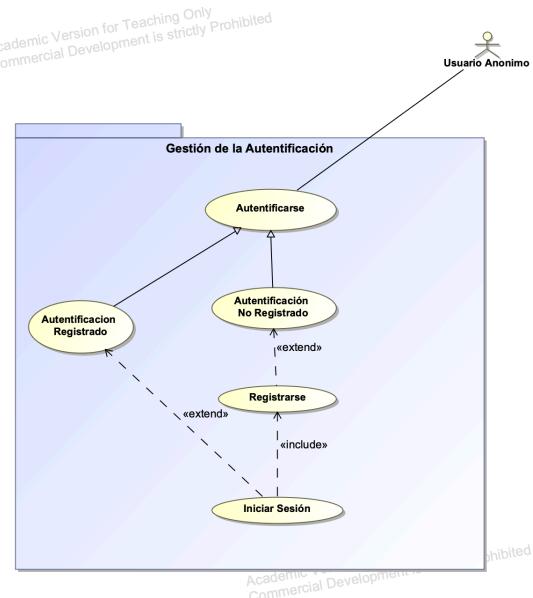


Figura 4.2: Caso de Uso: Usuario Anónimo

- **Gestión de Localizaciones:** El administrador podrá dar de alta localizaciones apoyándose en un mapa y darlas de baja en el sistema. También podrá editarlas, únicamente su nombre o sus coordenadas geográficas.
- **Gestión de Usuarios:** Finalmente el administrador tendrá la posibilidad de observar los usuarios vigentes en la aplicación y dar de baja usuarios.

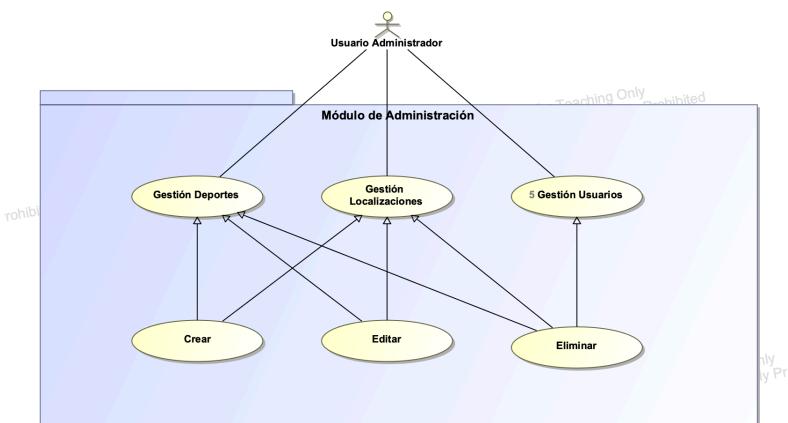


Figura 4.3: Caso de Uso: Usuario Administrador

4.3.3 Usuario Básico, Creador y Jugador

Estos 3 usuarios pueden hacer lo mismo, pero el Usuario Creador y el Usuario Jugador tienen actividades adicionales que pueden llevar a cabo, mientras que el Usuario Básico no. Como son muchas funcionalidades lo vamos a dividir en 5 módulos bien diferenciados con sus respectivos casos de uso.

1. Crear Partido

En este caso de uso el usuario ha de seleccionar, para poder **dar de alta en el sistema un evento deportivo**, un tipo de deporte, una localización, la fecha y hora de inicio y fin, y por último un máximo y mínimo de jugadores. Una vez seleccionada una localización, el **calendario** [1] que se dispone a modo de ayuda nos va a ir mostrando los partidos que se van a celebrar en esa localización a lo largo de los días. También disponemos de **información metereológica** [23] de la localización elegida.

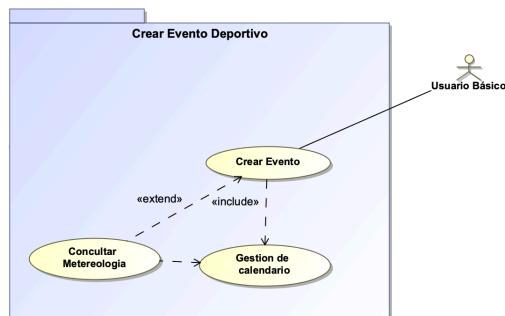


Figura 4.4: Caso de Uso Crear Partido: Usuario Básico

2. Gestión Partidos

Este es un módulo de gestión bastante completo, hay que tener en cuenta que muchos de los casos de uso presentes dependen mucho del 'estado' del partido. Además, todos los casos de uso podrán ser llevados a cabo por cualquiera de los 3 actores, excepto los marcados directamente con su actor.

2.1. Búsqueda de Partidos

Primero, la aplicación nos facilita la **búsqueda de partidos**, nos ofrece 2 vistas distintas, en modo **mapa o en calendario** [1]. Además hay que destacar que nos permite aplicar **filtros**

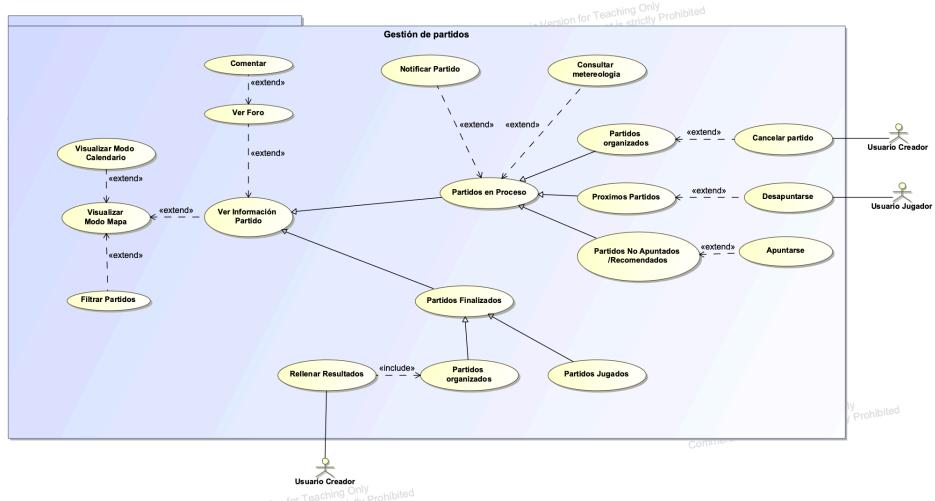


Figura 4.5: Caso de Uso Gestión Partidos: Usuario Básico/Creador/Jugador

por si queremos una búsqueda más concreta, ya sea por creador del partido, por tipos de deportes, por dificultad o por media de edad.

2.2 Detalles del Partido

Cuando accedemos a un partido, este en el estado en el que este, podemos observar la **información del partido**, que esto incluye tanto la información básica como los jugadores apuntados al partido.

2.2.1 Partido en Proceso

Si el partido está aun en proceso, podrémos **consultar la metereología** [23], **marcar el partido para que nos notifiquen** de cualquier cambio en él. En el caso de ser el Usuario Creador podrá decidir **cancelar el partido**, como consecuencia se enviará un correo avisando a todos los jugadores. En el caso de ser un Usuario Básico, se nos da la posibilidad de **apuntarnos al partido**, en cambio, si somos un Usuario Jugador, tendremos la opción de **desapuntarnos**.

2.2.2 Partido Finalizado

Si el partido está finalizado si fuimos el creador (Usuario Creador), debemos **rellenar los resultados**, por lo tanto este partido se incorporará a partidos pendientes de llenar resultado. Si fuimos jugadores (Usuario Jugador) de ese partido, posteriormente debemos **valorar tanto a los jugadores como al partido**.

3. Gestión de Valoraciones

Será realizada únicamente por el Usuario Jugador, es decir que cuando este usuario haya finalizado un partido, este pasa al apartado de partidos pendientes de valorar. **La valoración se hace de los partidos y jugadores** y consiste en darle un valor numérico entre 1 al 5 tanto al partido como a los demás jugadores, además, junto a la nota numérica, podemos dejarle a los jugadores un **comentario**.

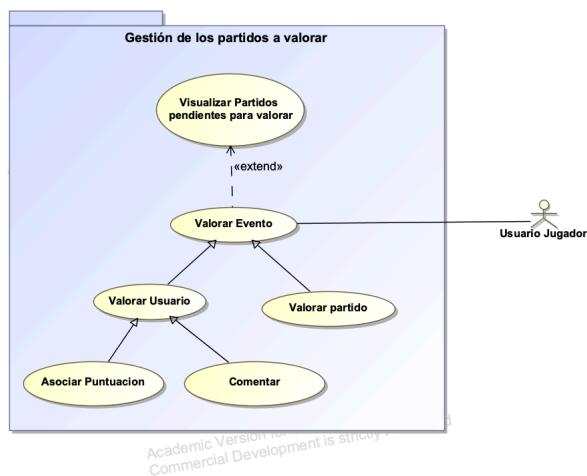


Figura 4.6: Caso de Uso Gestión Valoraciones: Usuario Jugador

4. Gestión de Perfiles

En este módulo reflejamos la parte del perfil del usuario. En la parte privada del usuario podrá tanto ver sus **datos personales**, como **editarlos**, no solo la información básica, si no que podrá modificar la contraseña, cambiar la foto de perfil e incluso añadir, eliminar y dar de alta en el sistema equipos externos de los que forma parte. Por otra lado, tenemos el perfil público dónde se muestra nombre, ciudad a la que pertenece y experiencia. El usuario podrá observar los **comentarios anónimos** que le deja la gente, los **próximos partidos** que tiene por jugar, **los organizados**, **los jugados** y **partidos recomendados** por parte de la aplicación en función de los partidos jugados. Hay que destacar que el resto de usuarios van a poder observar, aparte del nombre, ciudad y experiencia, los comentarios, próximos partidos, y los organizados por esa persona.

CAPÍTULO 4. ANÁLISIS

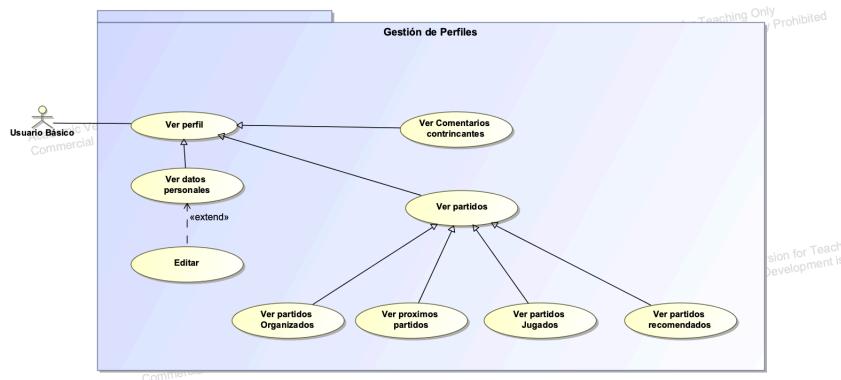


Figura 4.7: Caso de Uso Gestión Perfiles: Usuario Básico

5. Gestión social

Se consideró representar así los casos de uso ya no se pueden acceder a través de ningún otro, si no que muchas son accesibles únicamente mediante el menú. Cada usuario tendrá una **lista de seguidos y seguidores**. Desde esta lista podrá acceder al perfil de cada uno de ellos. Tiene la opción de **seguir o dejar de seguir** a un usuario, además si está siguiendo a una persona, puede **marcar a ese usuario para que se le notifique** de las actividades que realiza. Por otro lado tenemos la **mensajería**, nos permite entablar conversaciones con cualquier usuario. Finalmente tenemos el **Tablón de Actividades**, donde se mostrarán las actividades de todas las personas que el usuario sigue.

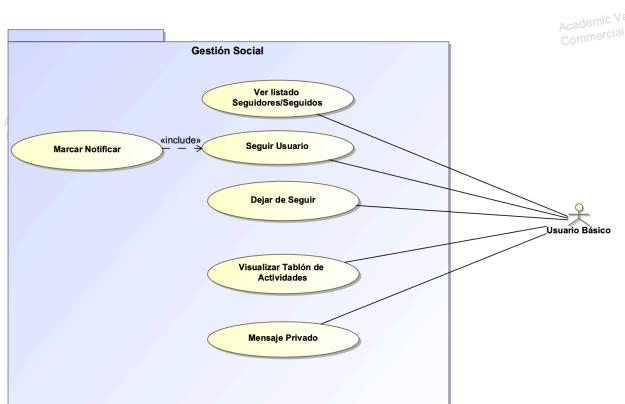


Figura 4.8: Caso de Uso Gestión Social: Usuario Básico

4.4 Arquitectura del sistema

Tenemos una arquitectura en 3 capas (modelo, controlador, vista) orientada a servicios, ya que utilizamos un servicio externo que sería el DatabaseLoader que carga los datos iniciales. Tenemos un Cliente Web que realizará peticiones REST al Servidor Web. Solo se conoce la capa inmediatamente inferior consiguiendo abstracción, aislamiento de cambios, incremento de escalabilidad, tolerancia a fallos y rendimiento.

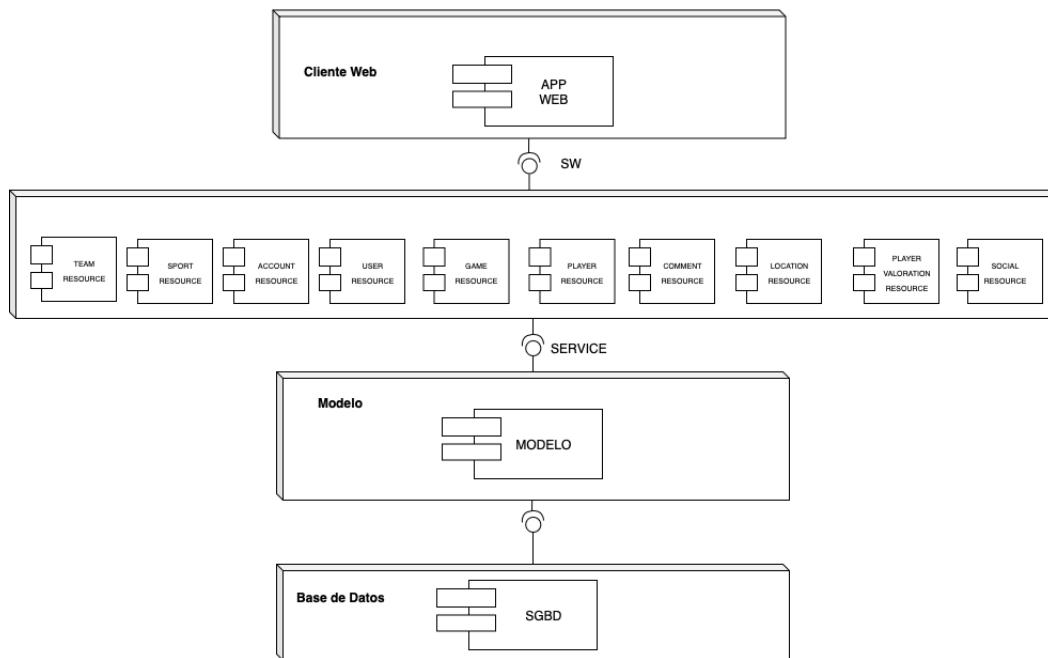


Figura 4.9: Arquitectura del sistema

4.5 Interfaz de usuario

Para la parte del cliente vamos a utilizar Vue.js, que está basado en el patrón **MVVM (Model–View–Viewmodel)** de esta manera separamos la interfaz de usuario de la lógica de negocio enlazando las propiedades y métodos públicos con la vista.

- **Vista:** Es el HTML y el CSS, es la plantilla o estructura de la interfaz de usuario.
- **Vista-Modelo:** Es el JS asociado a la vista, es una abstracción de la vista, en la que se puede acceder a propiedades y métodos públicos. Representa el estado de los datos del dominio en un momento concreto.
- **Modelo:** El cliente rest.

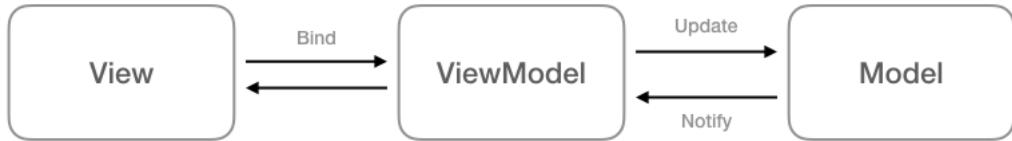


Figura 4.10: Patrón Model-View-ViewModel

Los componentes son elementos html reusables y configurables. Básicamente es una instancia Vue que será utilizada dentro de otra, y que será referenciada mediante una etiqueta específica. Los componentes son componibles, de tal modo que se pueden añadir componentes dentro de componentes, y tienen un orden jerárquico: los padres pueden modificar los datos de los hijos, pero no al revés. Los hijos únicamente pueden notificar eventos a los padres, pero no modificar su estado directamente.

Un componente en Vue está formado por 3 partes diferenciadas:

- **Template:** Se localiza dentro del html, Vue lo interpretará y renderizará de acuerdo a un modelo de datos. De este modo, podemos utilizar Vue en el front-end de manera similar a cualquier motor de templates de back-end. En el caso de Vue, todo lo que se encuentra entre corchetes en determinados tags html se interpreta como javascript y su valor es modificado en función del valor de nuestros objetos.
- **Script:** Nos permite añadir a aplicaciones web ya existentes y aprovecharlo sin mayores problemas de instalación, simplemente cargándolo en esta parte del componente.
- **Style:** En este apartado es donde añadiremos los estilos que se aplicarán a ese componente

En el componente principal, App.vue, el de la siguiente figura, tenemos el "MenuBar.vue" y el Footer que serán los elementos comunes a todas las pantallas y realmente lo que va a variar va a ser el router-view, ya que va a injectar un componente u otro dependiendo de las rutas a las que accedamos. Estas rutas se van a definir en el componente AppRoute.vue a través de VueRoute, este tema lo trataremos en el apartado 5.4.2.

```
1 <template>
2   <div id="app">
3     <notifications
4       :max="3"
5       :width="400"
6       position="top center">
7       <!-- -->
8     </notifications>
9     <MenuBar>
10    <!-- -->
11  </MenuBar>
12
13  <router-view class="content">
14    <!-- -->
15  </router-view>
16  <footer class="footer-distributed">
17    <div class="footer-left">
18      <h3>PLAY2GETHER</h3>
19      <p class="footer-company-name">PLAY2GETHER &copy; 2019</p>
20    </div>
21
22  </div>
23 </template>
24
25 <script>
26 import auth from './common/auth'
27 import MenuBar from './components/MenuBar'
28 import { HTTP } from './common/http-common'
29
30 export default {
31   name: 'App',
32   components: { MenuBar },
33   data() {
34     return {
35       // enlazamos el objeto donde vamos a guardar los datos de au
36       // de esta forma se activa el data-binding y los podemos usa
37       // propiedades públicas en los componentes
38       storeAuth: auth,
39     }
40   },
41   data() {
42     return {
43       auth,
44     }
45   },
46 </script>
47 <style scoped lang="scss">
```

Figura 4.11: Estructura Componente vue

4.5.1 Componentes y Navegación

En la tabla de "Componentes Comunes" no mostraremos los elementos de "LoadingPage.vue" ni "NotFound.vue" ya que son componentes sin ninguna funcionalidad importante.

4.6 Modelo conceptual de datos

En este apartado se va a hablar de la capa de modelo, la encargada de implementar los casos de uso y la lógica de negocio. Para ello vamos a mostrar el diseño de la BD, con sus entidades y atributos que forman la base de esta aplicación.

Componentes Administración	Descripción
Deportes	En este componente se visualizará la lista de deportes. Al pulsar en "Nuevo Deporte" se cargará a la derecha el componente "DeportesForm"
DeportesDetail	En este componente se visualizarán los detalles del deporte y se podrán editar estos datos dandole a "Editar", cambiándose el componente por el "DeportesForm".
DeportesForm	Es el componente utilizado para la creación de nuevos deportes o la edición de los existentes
LocalizacionesDetail	En este componente se visualizarán los detalles de la Localización y se podrán editar estos datos dandole a "Editar" emitiéndose una señal al componente padre (Mapa.vue), recargándose una parte del componente del parent solo apta para la edición.
Mapa	En este componente se visualiza un mapa con todas las localizaciones vigentes (GIS), en el momento de darle al botón de "Nueva Localización" se recargará una parte de este componente solo apta para cuando se quiera dar de alta localizaciones
UserList	Es un componente que muestra la lista de todos los usuarios existentes en la aplicación y da la posibilidad de darlos de baja

Cuadro 4.1: Componentes de las pantallas del Administrador y Navegación

Componentes Usuario	Descripción
CommentGame	Es un componente para visualizar el foro de un juego. Se observará en los datos básicos del partidos, los comentarios y se podrá crear nuevos comentarios desde este componente.
Recomendations	Es un componente que se encarga de filtrar las recomendaciones según Deporte, Localización y Jugadores. Depende de lo que elijamos el componente "GameList.vue" mostrará en pantalla unos partidos recomendados u otros.

Cuadro 4.2: Componentes de las pantallas del Usuario y navegación

Componentes Usuario	Descripción
ActivitiesFollowed	En este componente se visualiza el Tablón de Actividades pre-cargando dentro de este componente el componente "GameList.vue", es decir, se van a mostrar en una lista las diferentes actividades que realizaron las personas a las que seguidos. Dependiendo del tipo de actividad, si seleccionamos en ella nos llevará a GameDetail.vue, si el usuario ha creado o se ha apuntado a un partido, o a CommentGame.vue si ha comentado en un partido.
Calendar	Es un componente para visualizar el calendario [1], se utiliza tanto en GameCreate.vue para visualizar los partidos creados y ver la disponibilidad que tenemos como en Game.vue para buscar partidos por fechas concretas. En ambos casos, cuando pinches en un evento te lleva a su detalle, al componente "GameDetail.vue"
FutbolForm	Es el componente que se carga al clickear sobre algún partido de Fútbol o de Baloncesto que salga en el modal del icono de la exclamación en el "MenuBar.vue". Se utiliza para llenar los resultados finales con pequeños cambios dependiendo de cada deporte, una vez enviado el formulario nos llevará al componente "FutbolResult.vue".
FutbolResult	Es el componente que visualiza los resultados finales de un partido de Fútbol o de Baloncesto con pequeños cambios dependiendo de cada deporte. Cabe comentar que cuando pulsamos en un jugador, nos enviará al perfil público de esa persona, al componente "GameUser.vue"
TennisForm	Es el componente que se carga al clickear sobre algún partido de Tennis o de Padel que salga en el modal del icono de la exclamación en el "MenuBar.vue". Se utiliza para llenar los resultados finales con pequeños cambios dependiendo de cada deporte, una vez enviado el formulario nos llevará al componente "TennisResult.vue".
TennisResult	Es el componente que visualiza los resultados finales de un partido de Tennis o de Padel con pequeños cambios dependiendo de cada deporte. Cabe comentar que cuando pulsamos en un jugador, nos enviará al perfil público de esa persona, al componente "GameUser.vue"
Game	En este componente se visualizarán todos los partidos, ya sea en mapa o en calendario [1]. Cuando pulsamos en un marcador del mapa se nos abre una lista con todos los partidos vigentes en esa ubicación. Si pulsamos en un partido de la lista o en un evento del calendario [1] nos lleva a "GameDetail.vue" para ver su detalle. En este componente cabe destacar que se puede realizar un filtrado de los partidos sobre el mapa.

Cuadro 4.3: Componentes de las pantallas de la aplicación y navegación

Componentes Usuario	Descripción
GameCreate	Es el componente que nos permite crear partidos. A la derecha de todo se carga a la vez el componente "Calendar.vue". Cuando elegimos una localización nos da la opción de ver la previsión del tiempo para esa localización cargando el componente "Weather.vue". Finalmente cuando le damos a crear nos enviará al componente "GameDetail.vue"
GameDetail	Es el componente que nos permite ver los detalles del partido, tanto la información básica como sus jugadores. Si pulsamos sobre el botón de la nube nos permitirá observar la previsión meteorológica [23] cargando el componente "Weather.vue", podemos pulsar en la campanita para activar las notificaciones del partido, luego tenemos el botón de comentarios que nos enviará al componente "CommentGame.vue" y finalmente, cuando pulsamos en un jugador, nos enviará al perfil público de esa persona, al componente "GameUser.vue".
GameList	Es el componente que va a utilizar tanto "GameUser.vue" para mostrar los partidos próximos, organizados, jugados y recomendados, como los comentarios anónimos que dejan otros jugadores sobre ese usuario. Cuando se pulse en cualquier partido se pasará al componente "GameDetail.vue" en caso de ser un partido aun vigente o un partido terminado pero al que aún su creador no completó los resultados", a "TennisResult.vue" o "FutbolResult.vue" en caso de partidos finalizados.
GameUser	Es el componente que va a visualizar el perfil público de la persona, tanto sus datos personales como los partidos próximos, jugados, organizados y recomendados, junto con los comentarios de otras personas hacia él, gracias al componente "GameList.vue". Y se podrá acceder a los detalles de los partidos pulsando en cada uno, la navegación de componentes en este caso está descrita en el anterior componente.
ValorationGame	Es el componente que se carga cuando pulsamos algún partido que salga en el modal que se abre al pulsar la campanita del "MenuBar.vue". Nos permite tanto valorar el partido como a los jugadores, una vez valorado nos redirige a "GameDetail.vue" en caso de ser un partido sin resultados aún rellenados por el creador, o a "TennisResult.vue" o "FutbolResult.vue" dependiendo del deporte, si es Tennis o Padel o Fútbol o baloncesto.

Cuadro 4.4: Componentes de las pantallas del Usuario y navegación

Componentes Usuario	Descripción
Weather	Este componente se encarga de realizar una llamada a la API de DarkSky a través de una clave. De esta forma podremos obtener la previsión metereológica [23] de cualquier localización a lo largo de una semana únicamente gracias a la latitud y longitud. Se utiliza este componente tanto en "GameCreate.vue" o "GameDetail.vue"
UserDetail	Es un componente que se encarga de visualizar toda la información personal del usuario. Cuando pulsas editar te recarga el componente "Registro.vue" para poder llevar a cabo esa edición.

Cuadro 4.5: Componentes de las pantallas del Usuario y navegación

Componentes Comunes	Descripción
Home	Es la pantalla de inicio, no ejerce ninguna funcionalidad en concreto, simplemente visualiza unas imágenes con un pequeño resumen de que es la aplicación. Se puede acceder a él clickeando en el logo de la aplicación arriba a la izq.
Login	Es el componente utilizado para el inicio de sesión. Se accede cuando se clickean en "login" arriba a la derecha en el "MenuBar.vue". Desde este componente también podemos dirigirnos al componente de Registro clickeando en su respectivo botón.
MenuBar	Visualiza lo que sería la barra de navegación de la aplicación. Muestra tanto las opciones de la barra, las del menú del usuario. En el caso de estar logueado como usuario, a mayores muestra las notificaciones de partidos pendientes de llenar resultados, partidos pendientes de valorar y la sección de mensajería. Este componente cambia dependiendo de los permisos que se tenga.
Registro	Este componente permite a los usuarios crearse una cuenta en la aplicación para poder disfrutar de todas las funcionalidades que ofrece esta. Este componente también se reutiliza para la edición de los datos personales del usuario. Se puede acceder desde el componente "Login.vue" pulsando en "registro", como desde el componente "UserDetail.vue" pulsando en "editar".

Cuadro 4.6: Componentes Comunes a todos los usuarios

CAPÍTULO 4. ANÁLISIS

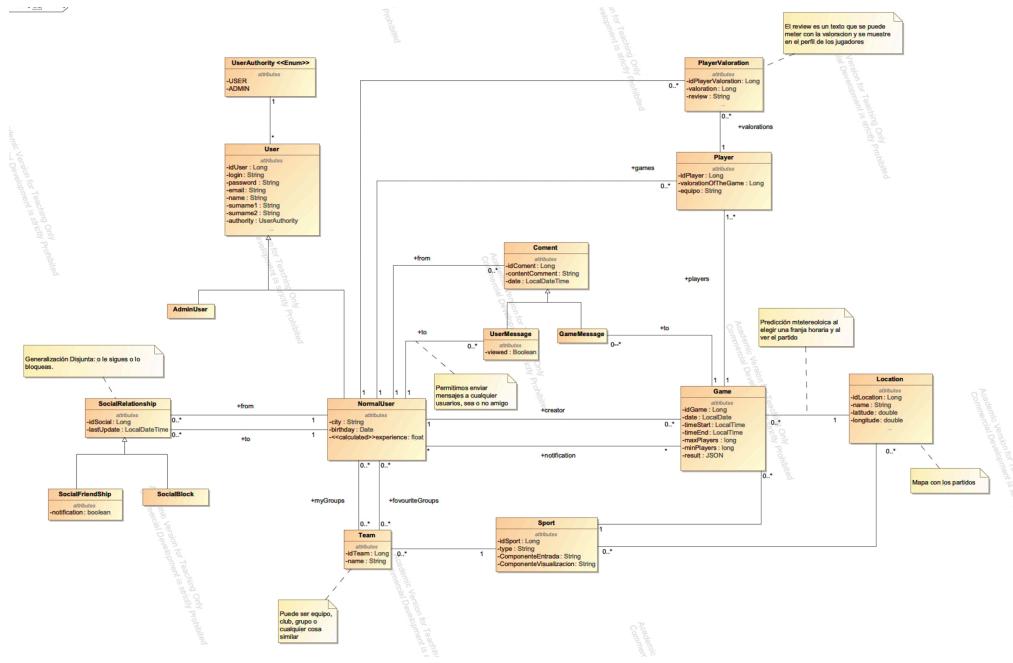


Figura 4.12: Diagrama de Clases

4.6.1 User

Representa los elementos comunes de todos los usuarios de la aplicación, tanto Administradores como Usuarios Básicos.

- **idUser**: Identificador del usuario.
- **login**: Login del usuario.
- **password**: Contraseña del usuario.
- **email**: Correo electrónico del usuario.
- **name**: Nombre del usuario.
- **surname1**: Primer apellido del usuario.
- **surname2**: Segundo apellido del usuario.
- **authority**: Permisos del usuario.

4.6.2 UserAuthority

Es un enumerado que representa los permisos de los usuarios.

- **USER:** Para los Usuarios Básicos. Tiene acceso a su información de usuario.
- **ADMIN:** Para los Usuarios Administradores. Se encarga de la gestión de los datos del sistema.

4.6.3 NormalUser

Usuario no Administrador

- **city:** Ciudad de residencia del usuario
- **birthday:** Fecha de nacimiento del usuario
- **experience:** Media de las valoraciones recibidas en los partidos jugados por los el resto de jugadores.

4.6.4 AdminUser

Usuario Administrador

4.6.5 SocialRelationship

Representa las relaciones entre usuarios, ya sean de amistad o de bloqueo

- **idSocial:** Identificador de la relación entre 2 usuarios
- **lastUpdate:** Fecha de alta de la relación

4.6.6 SocialFriendship

Representa una relación de amistad entre 2 usuarios.

- **notification:** Indica si queremos recibir notificaciones sobre las actividades que realiza esta amistad.

4.6.7 SocialBlock

Representa una relación de bloqueo entre 2 usuarios. Esta entidad se representa en el diagrama, pero realmente no se ha desarrollado ninguna funcionalidad respecto a esta clase finalmente, pero como en un futuro puede desarrollarse decidimos mostrarla para tener en cuenta su posibilidad.

4.6.8 Team

Representa los equipos externos a la aplicación de los que forman parte los usuarios.

- **idTeam:** Identificador del equipo.
- **name:** Nombre del equipo.

4.6.9 Sport

Representa los deportes que hay dados de alta en la aplicación.

- **idSport:** Identificador del deporte.
- **type:** Indica el tipo de deporte que es, sea Fútbol, Padel, Tennis, Baloncesto...
- **componente de Entrada:** Indica el nombre de la plantilla para el relleno de resultados.
- **componente de Visualización:** Indica el nombre de la plantilla para visualizar los resultados del partido.

4.6.10 Location

Representa las localizaciones donde se van a jugar los eventos deportivos.

- **idLocation:** Identificador de la localización.
- **name:** Nombre de la localización.
- **latitud:** Coordenada que mide el ángulo entre cualquier punto al ecuador.
- **longitud:** Coordenada que mide el ángulo a lo largo del ecuador desde cualquier punto.

4.6.11 Game

Representa el evento deportivo que se va a realizar.

- **idGame:** Identificador del partido.
- **date:** Fecha en la que tendrá lugar el partido.
- **timeStart:** Hora de inicio del partido.
- **timeEnd:** Hora de finalización del partido.
- **maxPlayers:** Número máximo de jugadores permitidos.
- **minPlayers:** Número mínimo de jugadores necesarios para llevarse a cabo el partido.

- **result:** Resultado del partido en formato JSON para que sea utilizable para todo tipo de deportes.

4.6.12 Player

Representa a los jugadores que han decidido apuntarse a un evento deportivo

- **idPlayer:** Identificador del jugador.
- **valorationOfTheGame:** Valoración del juego por parte de ese jugador.
- **equipo:** Equipo del que forma parte dentro del partido (A o B)

4.6.13 Player Valoration

Representa las valoraciones de cada uno de los jugadores después de terminar un partido.

- **idPlayerValoration:** Identificador de la valoración de un jugador
- **valoration:** Valoración numérica entre 1 y 5.
- **review:** Comentario sobre el jugador.

4.6.14 Comment

Representa los elementos comunes tanto de los comentarios realizados en el foro del partido, como los mensajes directos entre los usuarios.

- **idComment:** Identificador del comentario.
- **contentComment:** Contenido del comentario.
- **date:** Fecha de creación del comentario.

4.6.15 UserMessage

Representa los mensajes directos entre los usuarios. No hace falta que ambos usuarios que establezcan una conversación tengan una relación de amistad

- **viewed:** Estado del mensaje, si fue visto o no.

4.6.16 GameMessage

Representa los comentarios realizados en el foro de un partido.

Capítulo 5

Diseño

En este capítulo se expondrán los resultados en cuanto a diseño. Para ello, se explicará la arquitectura que seguirá el proyecto y el diseño de la aplicación, tanto la parte front-end como la back-end.

5.1 Arquitectura tecnológica del sistema

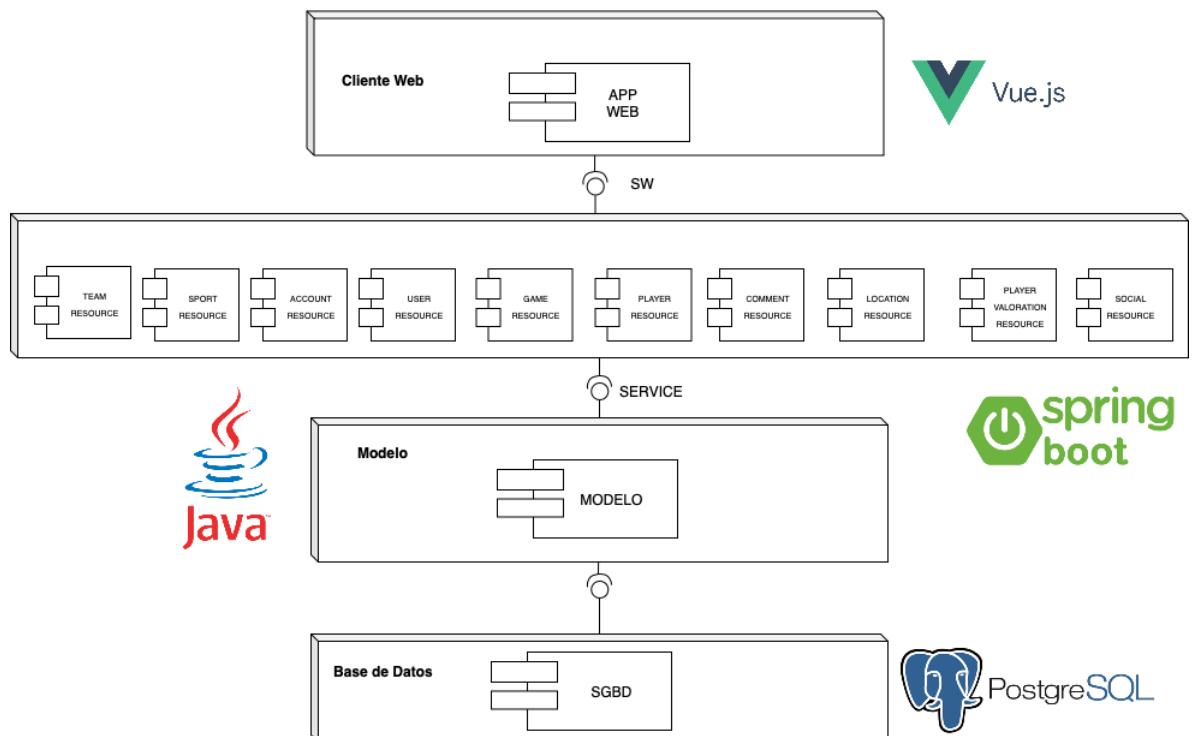


Figura 5.1: Arquitectura tecnológica del sistema

En este apartado se va a realizar una descripción tecnológica del sistema, para ello se utilizará la figura 4.9 de la parte de Análisis señalando en ella las tecnologías utilizadas en cada componente.

Primero, en la capa inferior de la arquitectura, en la Base de Datos, se utilizará PostgreSQL, un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto. Para la conexión de la BD con la capa superior se utilizará JDBC (Java Database Connectivity), es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java utilizando el dialecto SQL del modelo de base de datos que se utilice.

Para la conexión entre la capa del modelo y el servidor se utilizarán los DAOs (Data Access Object) como objeto de acceso a datos, suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos. Estos DAOs se comunican con los servicios implementados en Spring-Boot.

Para conectar los servidores Web con el cliente Web, que es un navegador, se hará uso de REST. Este cliente fué implementado con Vue.js, es un Framework progresivo, es decir, es un Framework que sirve para consumir la interfaz del usuario.

5.2 Diseño de la aplicación

Esta aplicación se dividirá en 2 partes. La parte front-end, el cliente, es la parte con la que interactúan los usuarios, la capa más superficial de la app que recogerá los datos de entrada del usuario. Por otro lado está la parte back-end, el servicio, la parte que el usuario final no puede ver. Su función es acceder a la información que se solicita, a través de la aplicación, para luego combinarla y devolverla al usuario final. Los datos de entrada del usuario como posteriormente los que se expondrán al cliente serán devueltos en DTOs. La finalidad de los DTO es congregar información de múltiples fuentes o tablas y concentrarlas en una única clase simple.

5.3 Back-end

Esta parte de la aplicación es el servicio REST. Se divide en capa de persistencia, de acceso a los datos, los servicios y finalmente los controladores a través de los cuales el Cliente Web se comunica a través de peticiones con el Servidor Web.

5.3.1 DAOs

Data Access Object (DAO), permite separar la lógica de acceso a datos de los Business Objects, de tal forma que el DAO encapsula toda la lógica de acceso de datos al resto de la aplicación. De esta forma, el DAO proporcionará los métodos necesarios para insertar, actualizar, borrar y consultar la información (métodos CRUD), mientras que por otro lado la capa de negocio solo se preocupará por la lógica de negocio y utilizará los DAOs para interactuar con la fuente de datos.

En el proyecto, si vamos a la carpeta `es.udc.lbd.asi.restexample.model.repository` se encontrará el DAO `es.udc.lbd.asi.restexample.model.repository.PlayerDAO` y su implementación `es.udc.lbd.asi.restexample.model.repository.PlayerDAOHibernate`. A continuación se expondrá un ejemplo de DAO, ya que el resto seguirán la misma estructura.

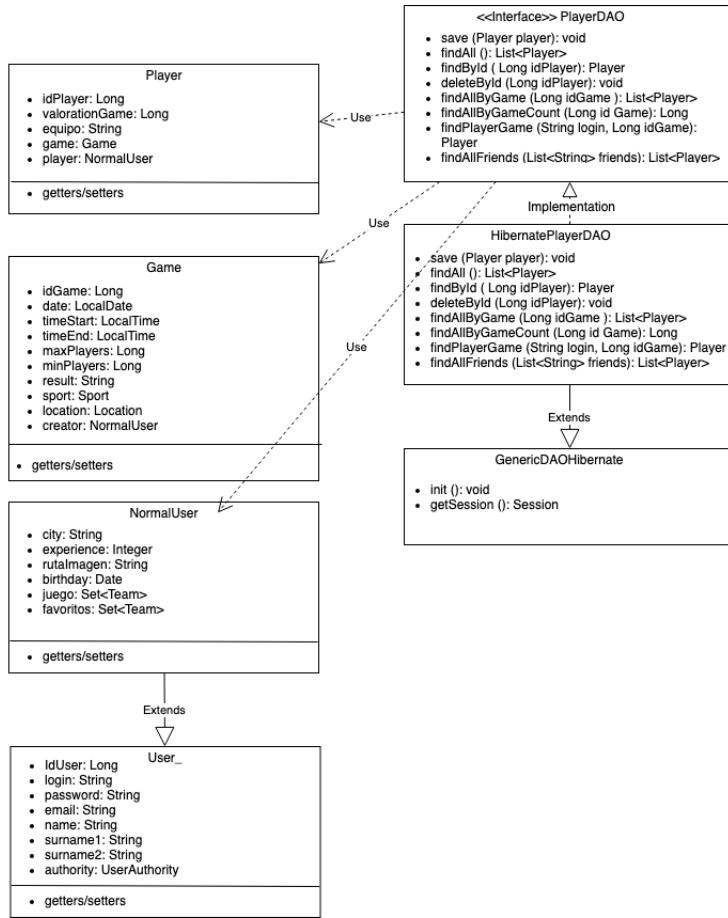


Figura 5.2: Diagrama PlayerDAO

Se puede observar que el DAO posee alguno de los métodos CRUD: crear, eliminar y buscar

jugadores concretos. A mayores tenemos otros métodos bastante interesantes como buscar todos los jugadores de un partido, buscar el jugador concreto de un usuario para un partido, buscar el número de jugadores apuntados a un partido o buscar todos los jugadores de todas las personas a las que sigo.

5.3.2 Servicios

La capa de servicios consiste en la lógica que realiza las funciones principales de la aplicación, las operaciones de alto nivel. Para ello las operaciones llamarán a los DAOs que se encargarán del acceso a datos y luego en el servicio se utilizarán estos datos para ofrecer funcionalidades mucho más complejas y avanzadas.

En el proyecto, si vamos a la carpeta `es.udc.lbd.asi.restexample.model.service` se encontrará la fachada del Servicio `es.udc.lbd.asi.restexample.model.service.PlayerServiceInterface` y su implementación `es.udc.lbd.asi.restexample.model.service.playerService`.

En nuestro caso se encontrarán varios servicios con sus respectivas operaciones a las que se podrá acceder desde el cliente a través de peticiones a los controladores. Los distintos servicios que se encontrarán son:

UserService



Figura 5.3: User Service

CAPÍTULO 5. DISEÑO

SportService



Figura 5.4: Sport Service

TeamService



Figura 5.5: Team Service

SocialRelationshipService



Figura 5.6: SocialRelationship Service

PlayerValorationService



Figura 5.7: PlayerValorationService Service

GameService



Figura 5.8: Game Service

CommentService



Figura 5.9: Comment Service

LocationService

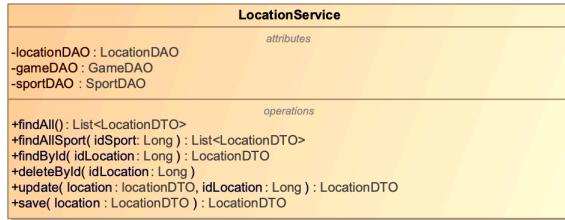


Figura 5.10: Location Service

PlayerService



Figura 5.11: Player Service

5.3.3 Controladores

Los controladores se sitúan en el paquete `es.udc.lbd.asi.restexample.web`. En este apartado se mostrarán los diagramas de cada controlador con sus respectivas operaciones REST y las URL necesarias para acceder a estas desde el cliente a través de peticiones REST.

UserResource

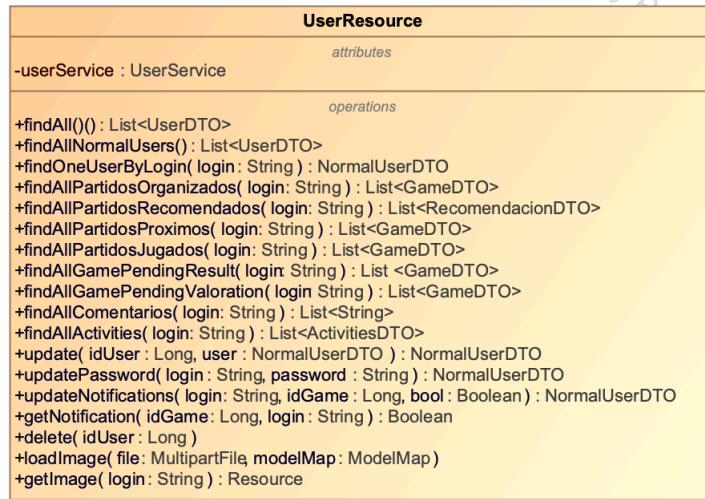


Figura 5.12: User Resource

URL	OPERACIÓN	ACCIÓN
/api/users/	GET	findAll
/api/users/normal	GET	findAllNormalUsers
/api/users/login	GET	findOneUserByLogin
/api/users/login/organizados	GET	findAllPartidosOrganizados
/api/users/login/recomendados	GET	findAllPartidosRecomendados
/api/users/login/proximos	GET	findAllPartidosProximos
/api/users/idUser	PUT	update
/api/users/login/password	PUT	updatePassword
/api/users/notifications/login/idGame/bool	PUT	updateNotifications
/api/users/activities/login	GET	findAllActividades
/api/users/notifications/login/idGame	PUT	getNotification
/api/users/idUser	DELETE	delete
/api/users/uploadFile	POST	loadImage
/api/users/imagenes/login	GET	getImage

Cuadro 5.1: Acceso a UserResource

CAPÍTULO 5. DISEÑO

SportResource

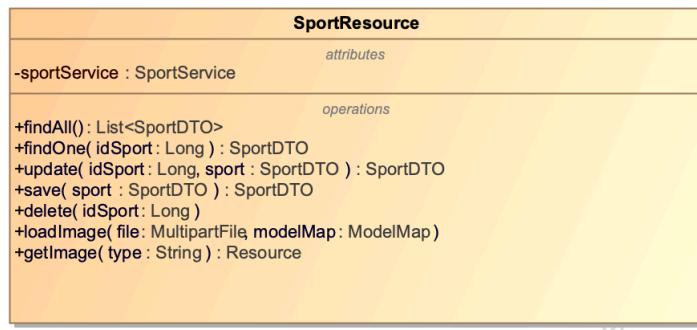


Figura 5.13: Sport Resource

URL	OPERACIÓN	ACCIÓN
/api/sports/	GET	findAll
/api/sports/idSport	GET	findOne
/api/sports/idSport	PUT	update
/api/sports/organizados	POST	save
/api/sports/idSport	DELETE	delete
/api/sports/uploadFile	POST	loadImage
/api/sports/imagenes/type	GET	getImage

Cuadro 5.2: Acceso a SportResource

TeamResource



Figura 5.14: Team Resource

URL	OPERACIÓN	ACCIÓN
/api/teams	GET	findAll
/api/teams	POST	save

Cuadro 5.3: Acceso a TeamResource

SocialResource

Figura 5.15: Social Resource

URL	OPERACIÓN	ACCIÓN
/api/social/loginFrom/loginTo	GET	findARelation
/api/social/follow/login/type	GET	findRelations
/api/social/friendShip	POST	save
/api/social/block	POST	save
/api/social/loginFrom/loginTo/notification	PUT	update
/api/social/friendShip/loginFrom/loginTo/typeRelation	DELETE	deleteRelationShip

Cuadro 5.4: Acceso a SocialResource

CAPÍTULO 5. DISEÑO

PlayerValorationResource



Figura 5.16: PlayerValoration Resource

URL	OPERACIÓN	ACCIÓN
/api/playersValoration	GET	findAll
/api/playersValoration	POST	save

Cuadro 5.5: Acceso a PlayerValorationResource

GameResource

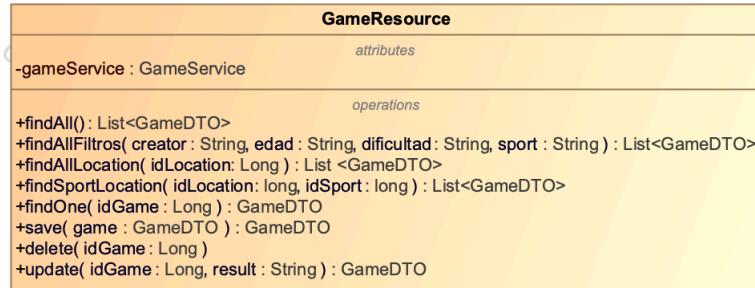


Figura 5.17: Game Resource

URL	OPERACIÓN	ACCIÓN
/api/games	GET	findAll
/api/games/idGame	GET	save
/api/games/filtro	GET	findAllFiltros
/api/games/locations/idLocation	GET	findAllLocation
/api/games/sportsLocation/idSport/idLocation	GET	findAllSportLocation
/api/games/idGame	PUT	update
/api/games/	POST	save
/api/games/idGame	DELETE	delete

Cuadro 5.6: Acceso a GameResource

CommentResource



Figura 5.18: Comment Resource

URL	OPERACIÓN	ACCIÓN
/api/comments/user/UserFrom/UserTo	GET	findAllUserFromUser
/api/comments/user/login	GET	findAllUserMessage
/api/comments/game/idGame	GET	findAllByGame
/api/comments/user/countMessages/login	GET	findAllMessagesNotViewed
/api/comments/user/UserFrom/UserTo	PUT	updateState
/api/comments/game	POST	save
/api/comments/user	POST	save

Cuadro 5.7: Acceso a CommentResource

LocationResource



Figura 5.19: Location Resource

URL	OPERACIÓN	ACCIÓN
/api/locations	GET	findAll
/api/locations/filter/idSport	GET	findAllSport
/api/locations/idLocation	GET	findOne
/api/locations/idLocation	DELETE	delete
/api/locations/idLocation	PUT	update
/api/locations	POST	save

Cuadro 5.8: Acceso a LocationResource

PlayerResource

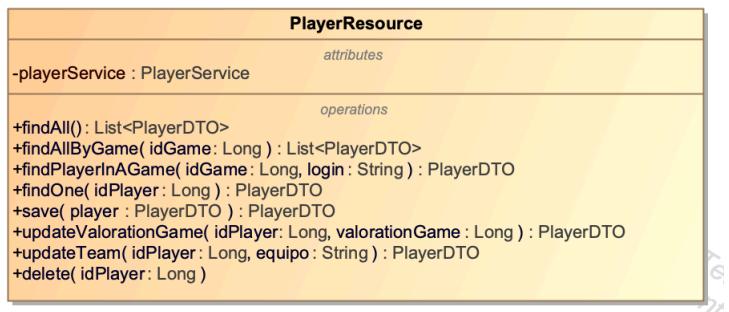


Figura 5.20: Player Resource

URL	OPERACIÓN	ACCIÓN
/api/players	GET	findAll
/api/players/idGame	GET	findAllByGame
/api/players/idGame/login	GET	findPlayerInAGame
/api/players/findPlayer/idPlayer	GET	findOne
/api/players/idPartido/team/equipo	PUT	updateTeam
/api/players/idPartido/valorationGame	PUT	updateValorationGame
/api/players	POST	save
/api/players/idPlayer	DELETE	delete

Cuadro 5.9: Acceso a PlayerResource

AccountResource

Figura 5.21: Account Resource

URL	OPERACIÓN	ACCIÓN
/api/authenticate	POST	authenticate
/api/register	POST	registerAccount
/api/account	GET	getAccount

Cuadro 5.10: Acceso a AccountResource

5.3.4 Patrones

DAO

Utilizaremos los DAOs para encapsular todo el acceso a una fuente de datos. La utilidad e importancia de éste reside en la necesidad de las aplicaciones de acceder a almacenes de datos, que pueden ser reemplazados por otros. Para aumentar la mantenibilidad de la aplicación, es preciso, por tanto, separar la lógica de negocio del origen de los datos.

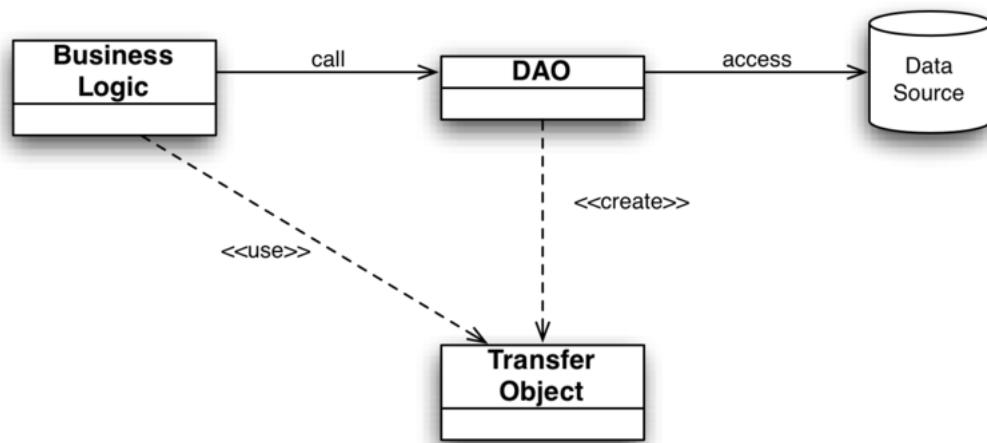


Figura 5.22: Diagrama del Patrón DAO

DTO

Se utilizarán los DTOs para transportar los datos desde la base de datos hacia la capa de lógica de negocio y viceversa. La primera razón de su utilización es para ocultar información al cliente, como puede ser en el caso de userDTO, de esta forma la password esta guardada en la BD pero no se pasa al lado cliente ya que es un dato crítico. Otra ventaja de este patrón es que nos permite reducir el número de llamadas remotas trayendo todos los datos necesarios en un único objeto en una única llamada.

Singleton

Garantizará que una clase solo tenga una instancia, y proporcionará un punto de acceso global a ella. Con Spring, el patrón singleton es el ámbito por defecto de un Bean. El contador de Spring creara una única instancia compartida de la clase asignada a ese Bean, por lo que siempre que se solicite ese Bean, se estará inyectando el mismo objeto. Es decir, nuestras clases con las anotaciones `@Repository` y `@Service` son Singleton (DAOs y services).

Front Controller y MVC

Spring es una implementación del patrón de diseño “Front controller”, sería concretamente el DispatcherServlet. También podemos hablar del patrón mvc (modelo-vista-controlador). El funcionamiento de estos 2 patrones en Spring es el siguiente:

- Las peticiones HTTP se canalizan a través del frontController, que no es más que un servlet, y a través de la URL sabe a qué controler enviarlo.
- Se llama al controller que ejecuta la lógica de negocio. Llama al servicio que llama a los DAOs que hacen las peticiones a las BD para obtener los datos. Cuando los tiene, lo que hace es devolver los datos al servlet encapsulados en un DTO.
- Finalmente, el frontController redirige la petición hacia la vista. Spring ofrece un servicio Rest al que a través de él se establecerá la conexión con la aplicación cliente (Vue).

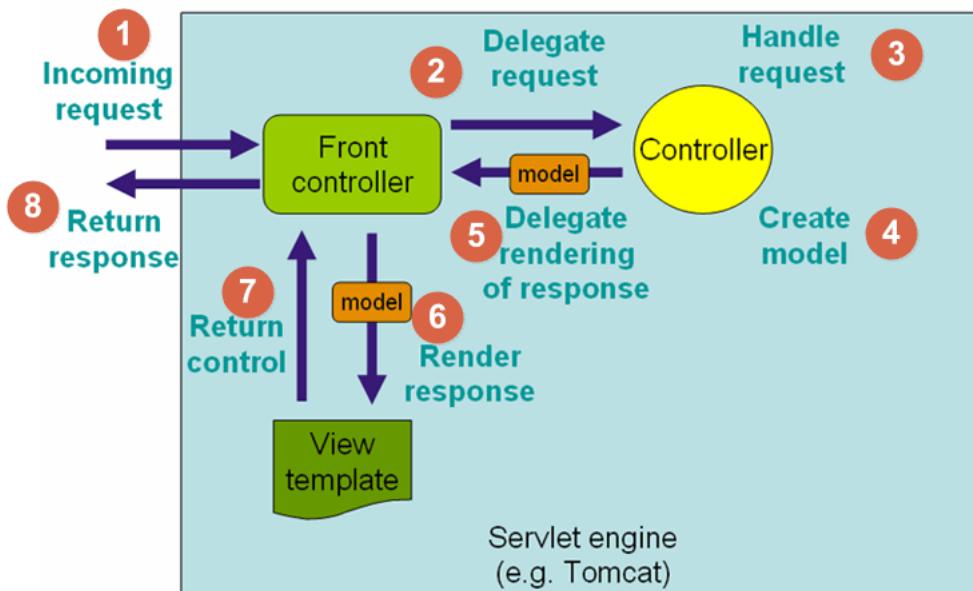


Figura 5.23: Diagrama MVC Spring

Fachada

El patrón Facade tiene la característica de ocultar la complejidad proporcionando una interface de alto nivel, la cual se encarga de realizar la comunicación con todos los subsistemas necesarios. La fachada es una buena estrategia cuando requerimos interactuar con varios subsistemas para realizar una operación concreta ya que se necesita tener el conocimiento técnico y funcional para saber qué operaciones de cada subsistema tenemos que ejecutar y en qué orden, lo que puede resultar muy complicado cuando los sistemas empiezan a crecer demasiado.

Es decir, en resumen, el patrón fachada proporciona una interfaz unificada de comunicación entre subsistemas y cliente.

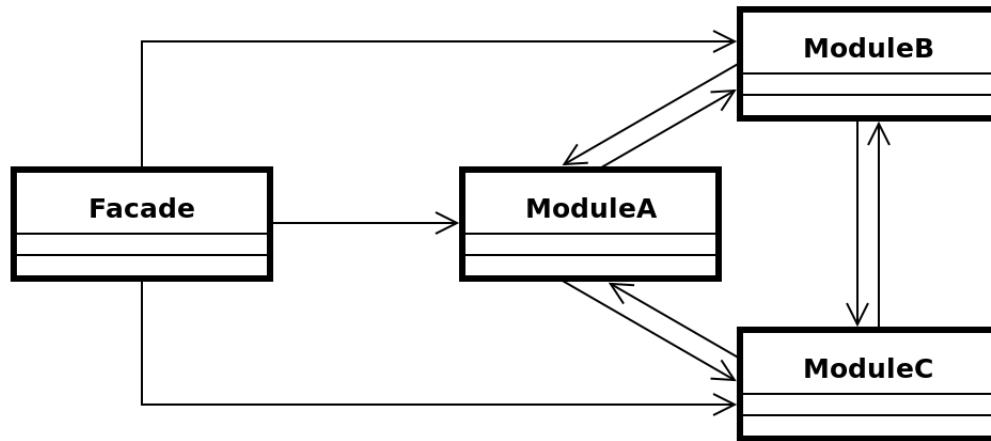


Figura 5.24: Diagrama MVC Spring

Inyección de Dependencias

El patrón de inyección de dependencias (DI, Dependency Injection) es un patrón que implementa el principio de inversión de control (IoC), donde el control se invierte estableciendo dependencias entre objetos. Esto se consigue inyectando objetos en otros objetos mediante un ensamblador, en vez de a través de los objetos mismos. La inyección de dependencias resuelve el problema de hacer que una clase sea independiente de cómo los objetos que requiere son creados.

5.4 Front-end

Para la parte Front-end se ha decidido utilizar Vue.js. Vue se basa en una implementación muy ligera del patrón view-model como se explicaba en el apartado 4.5 que ayudará a relacionar nuestra capa de presentación con nuestra capa de negocio de forma sencilla y eficiente. Es un framework progresivo, es decir, que es muy fácil añadir a cualquier proyecto ya existente, en cambio frameworks como Angular están más orientados a comenzar proyectos desde 0.

5.4.1 Componentes

Se hará una clara distinción entre los componentes citados en el apartado 4.5.1 según los permisos. Por un lado están los componentes de Administración, los componentes de los Usuarios Básicos y finalmente los componentes Comunes, que serán accesibles tanto para Usuarios Anónimos, como para el Administrador como para los Usuarios Básicos. En este diagrama no se mostrarán los componentes "LoadingPage.vue" ni "NotFound.vue" ya que son componentes sin ninguna funcionalidad importante.

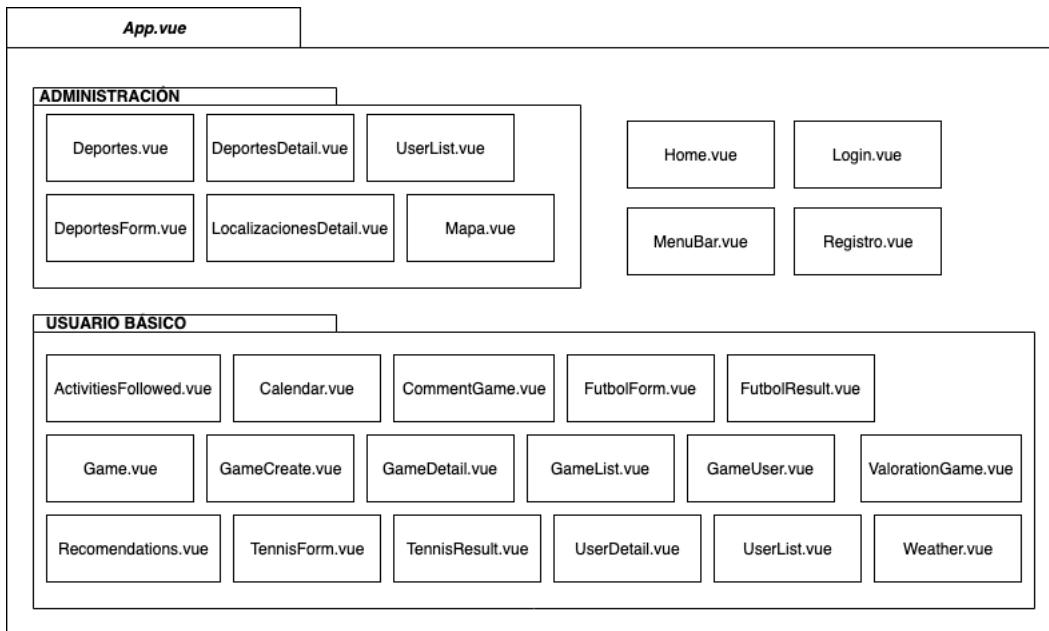


Figura 5.25: Diagrama de componentes del Cliente

Dentro de un mismo componente tenemos 3 apartados, como bien se decía anteriormente, la parte del html que define la vista, el css con los estilos aplicados a la vista del componente, y la parte javascript que define el comportamiento del componente.

Cabe destacar que estos paquetes, pueden incluir otros paquetes. Por ejemplo, los componentes "Game.vue" y "GameCreate.vue" hacen uso del componente "Calendar.vue"

5.4.2 Rutas

Para el enrutamiento, es decir, para asociar a cada componente una ruta, Vue.js utiliza VueRoute. Se van a definir las rutas en el AppRoute.vue. Luego en nuestro componente principal (App.vue) en el router-view se va a inyectar un componente u otro dependiendo de las rutas a las que vayamos acceder. Las rutas de los componentes del cliente son las siguientes.

Nombre	Ruta	Componente
Home	/	Home.vue
Registro	/login/createAccount	Registro.vue
Login	/login	Login.vue
UserDetail	/users/:id	UserDetail.vue
UserUpdate	/users/:id/edit	Registro.vue
UserList	/users	UserList.vue
LocalizacionesDetail	/localizaciones/details	LocalizacionesDetail.vue
Mapa	/localizaciones/mapaue	Mapa.vue
Deportes	/deportes	/ Deportes.vue
DeportesDetail	/deportes/:id	DeportesDetail.vue
DeportesCreate	/deportes/new	DeportesForm.vue
DeportesUpdate	/deportes/:id/edit	DeportesForm.vue
GameCreate	/games/new	GameCreate.vue
GameDetail	/games/:id	GameDetail.vue
CommentGame	/games/:id/comment	CommentGame.vue
ValorationGame	/games/:id/valoraciones	ValorationGame.vue
FutbolForm	/games/:id/futbol/completeResultado	FutbolForm.vue
TennisForm	/games/:id/tennis/completeResultado	Tennisform.vue
FutbolResult	/games/:id/futbol/resultado	FutbolResult.vue
TennisResult	/games/:id/tennis/resultado	TennisResult.vue
GameUser	/users/:id/perfilPublico	GameUser.vue
GameList	/games/organizados/user/:id	GameList.vue
Recomendations	/games/recomendados	Recomendations.vue
Game	/games	Game.vue
Weather	/games/weather	Weather.vue
Calendar	/games/calendar	Calendar.vue
ActivitiesFollowed	/activities/followed	ActivitiesFollowed.vue

Cuadro 5.11: Rutas de los componentes del cliente

5.4.3 Comunicación con el servidor

En este apartado de la memoria se verá como los componentes del cliente se comunican con el servidor a través de peticiones HTTP. A continuación se mostrarán las relaciones componente/Cliente-controlador/Servidor , es decir, cada componente del cliente que peticiones realiza a que controlador.

Componentes de Administración

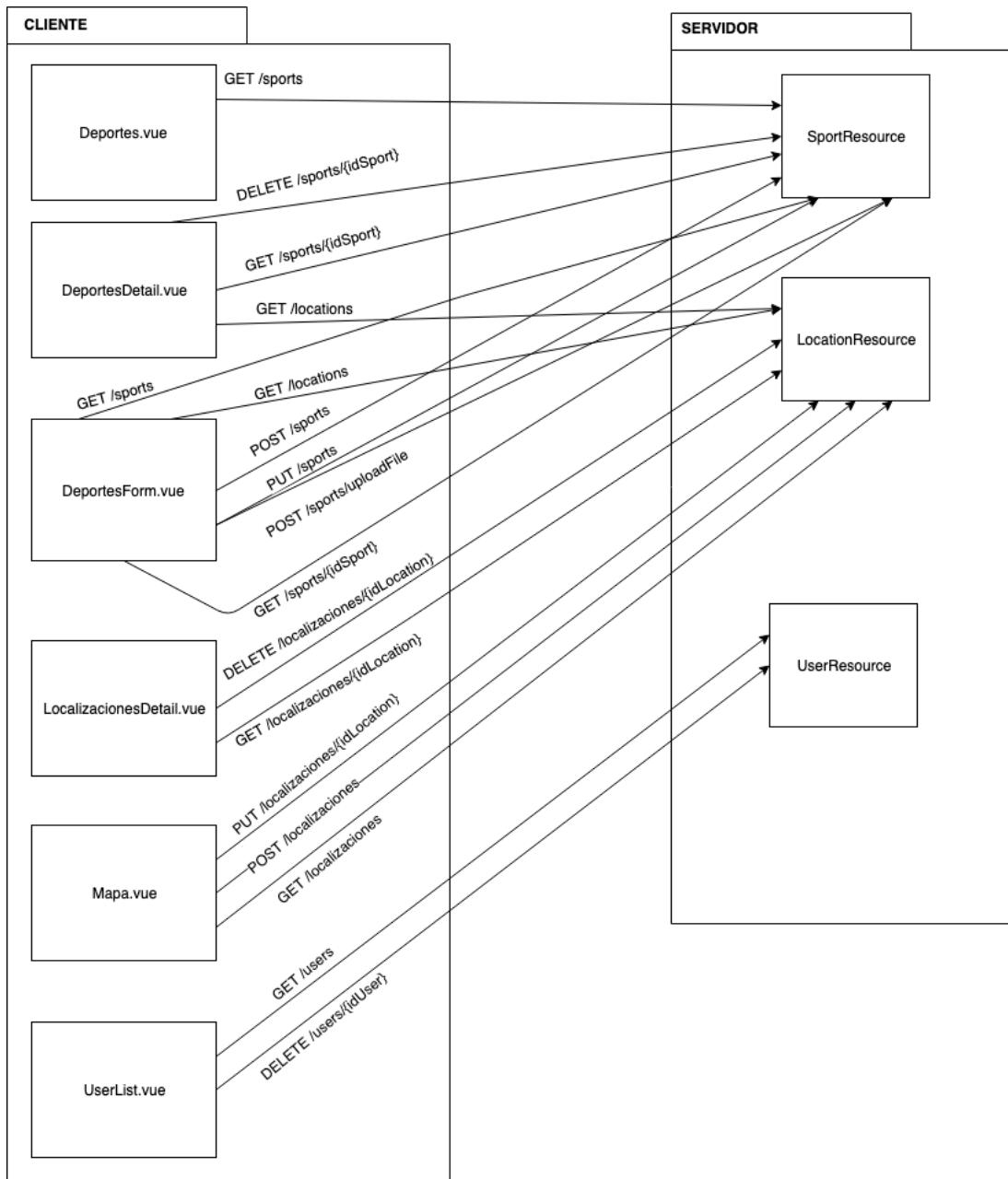


Figura 5.26: Comunicación de los componentes de Administración del Cliente al Servidor

Componentes de Usuarios Básicos

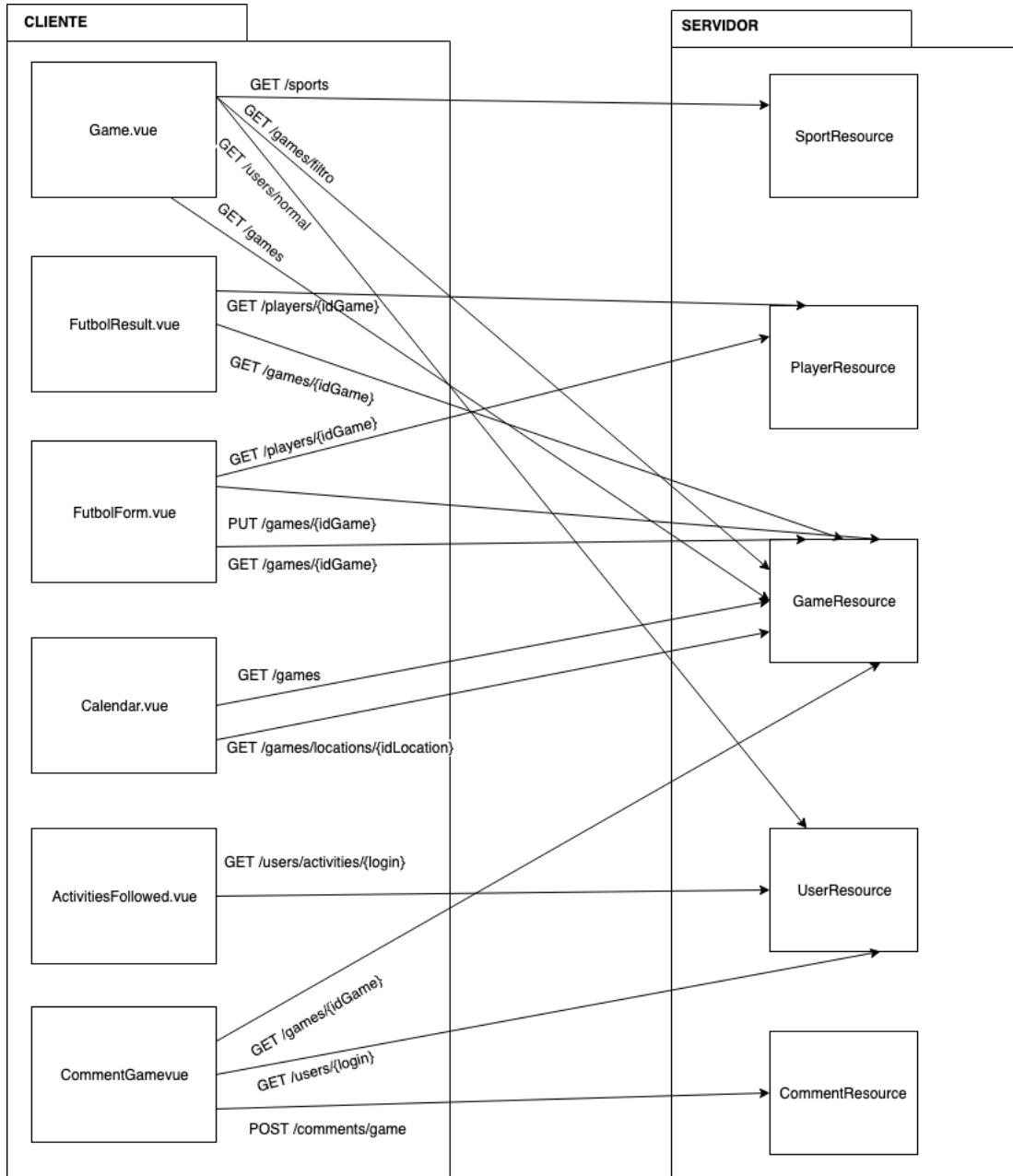


Figura 5.27: Comunicación de los componentes del Usuario del Cliente al Servidor

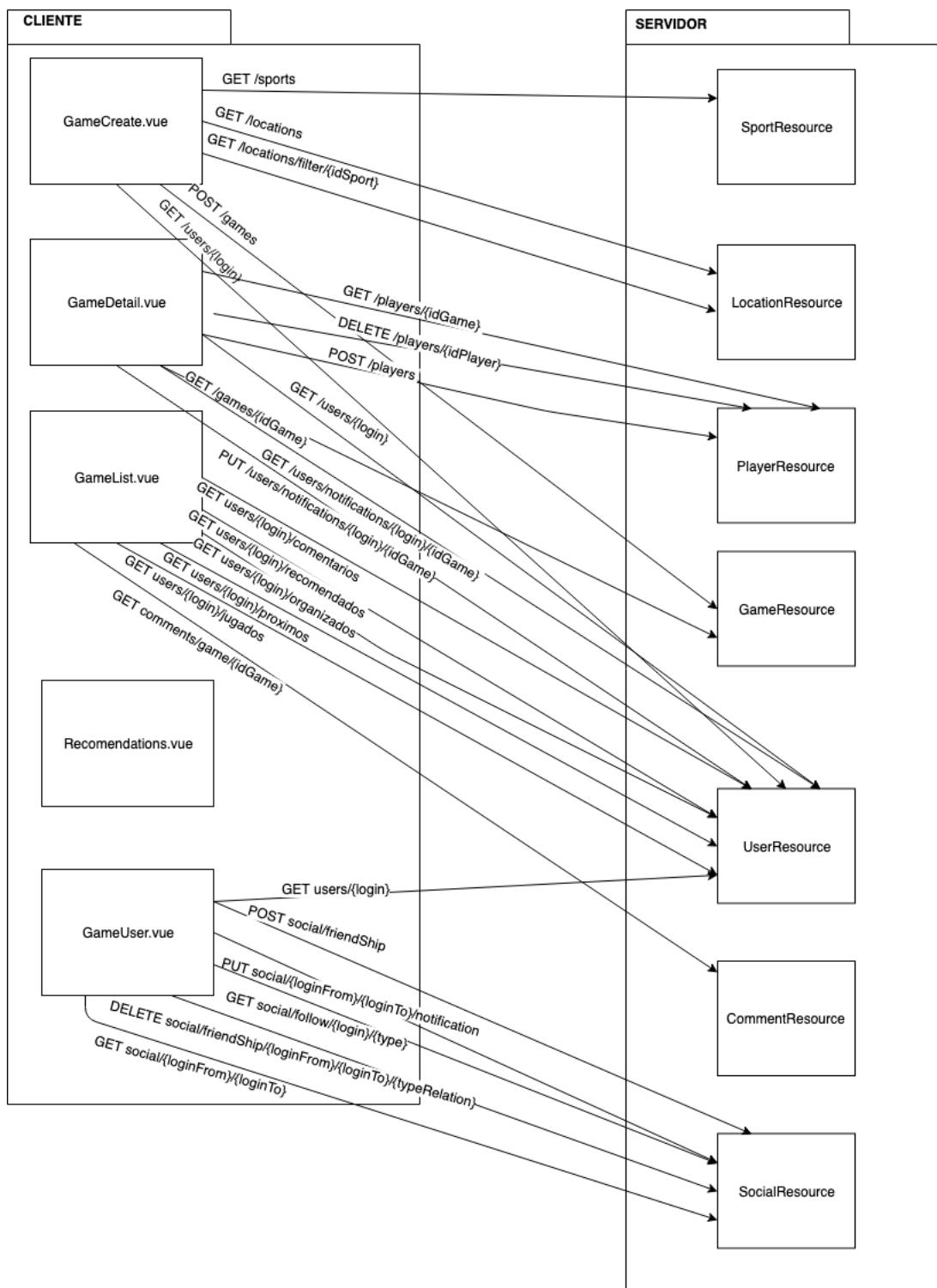


Figura 5.28: Comunicación de los componentes del Usuario del Cliente al Servidor

CAPÍTULO 5. DISEÑO

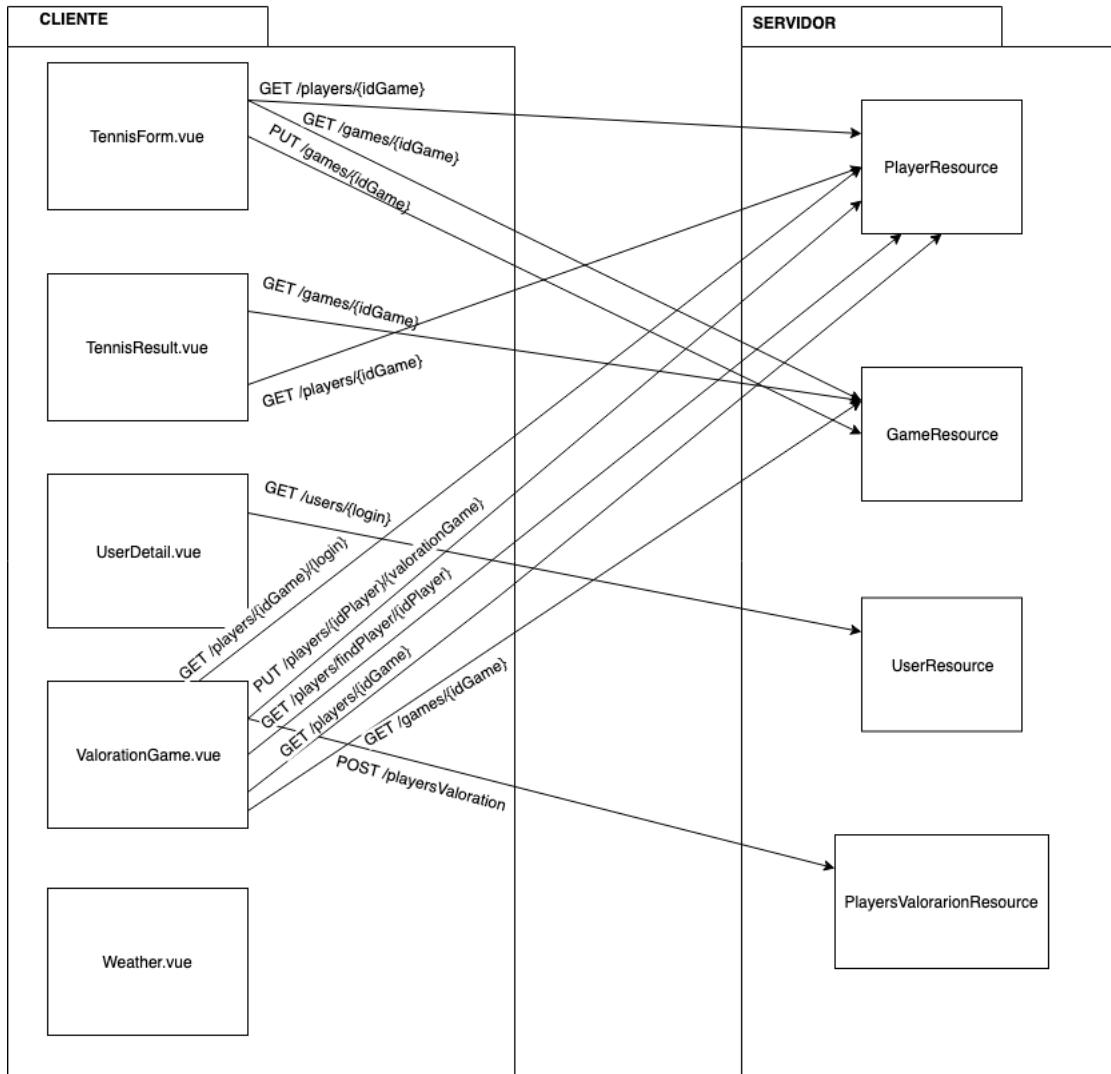


Figura 5.29: Comunicación de los componentes del Usuario del Cliente al Servidor

Componentes Comunes

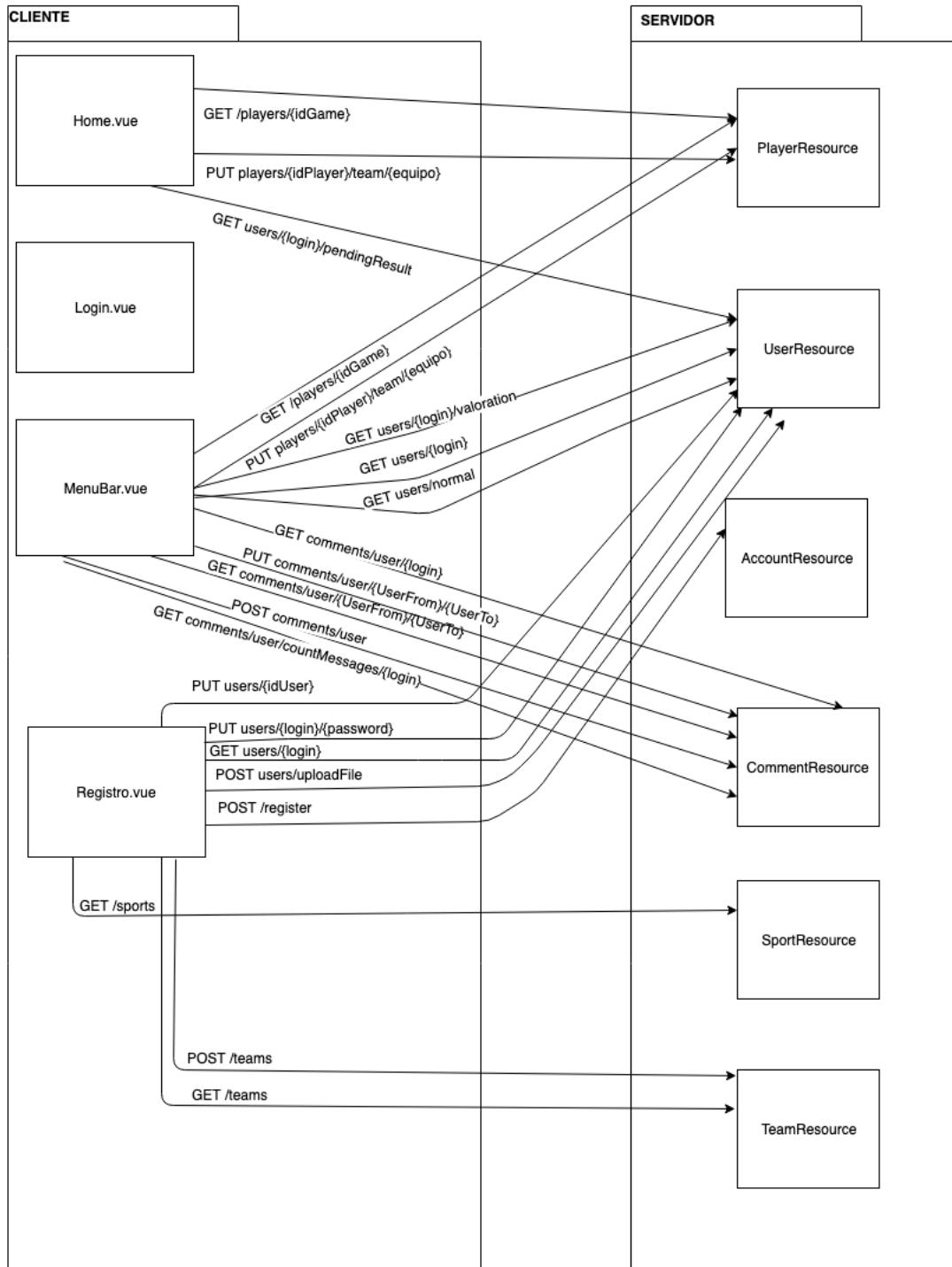


Figura 5.30: Comunicación de los componentes comunes del Cliente al Servidor

5.4.4 Patrones

Como vimos en el apartado 4.5, Vue.js está basado en el patrón MVVM (**Model-View-Viewmodel**), pero el MVVM no es el único patrón que influye en las Interfaces de Usuario.

Callback y Promise

Vue.js hace uso de los patrones **Callback** y **Promise**. Al incorporar estos patrones, se logra que el cliente pueda seguir trabajando hasta que la información esté disponible, ejecuta una funcionalidad de forma asíncrona. Es decir, se asiste a una función que no puede devolver inmediatamente su resultado y devuelve una promesa de que tendrá el resultado en un futuro, es decir, devuelve un objeto promise que gestionará un callback. En la parte del cliente habrá muchas partes del código donde se utilizarán estos patrones, por ejemplo, en todos los casos en los que en la parte cliente se haga una petición HTTP. El cliente hará la petición y esa petición no devuelve inmediatamente la respuesta, así que este sigue trabajando, le promete al cliente que tendrá la respuesta, en el momento que llegue el resultado, se ejecutara el callback.

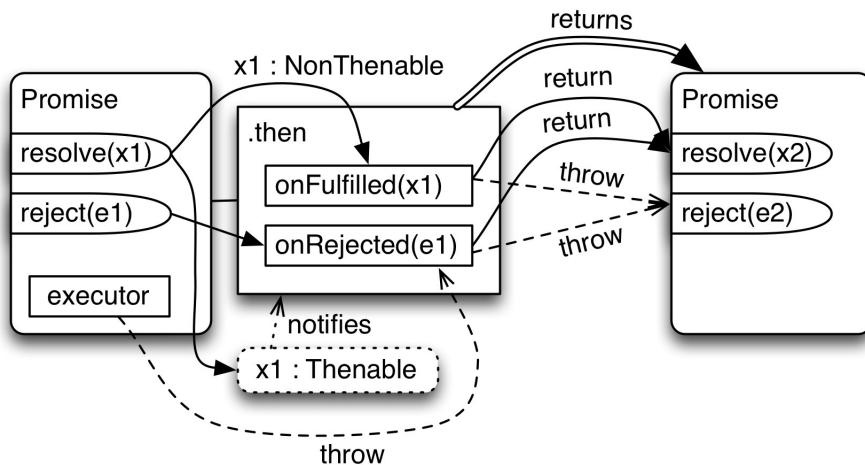


Figura 5.31: Patrón Promise

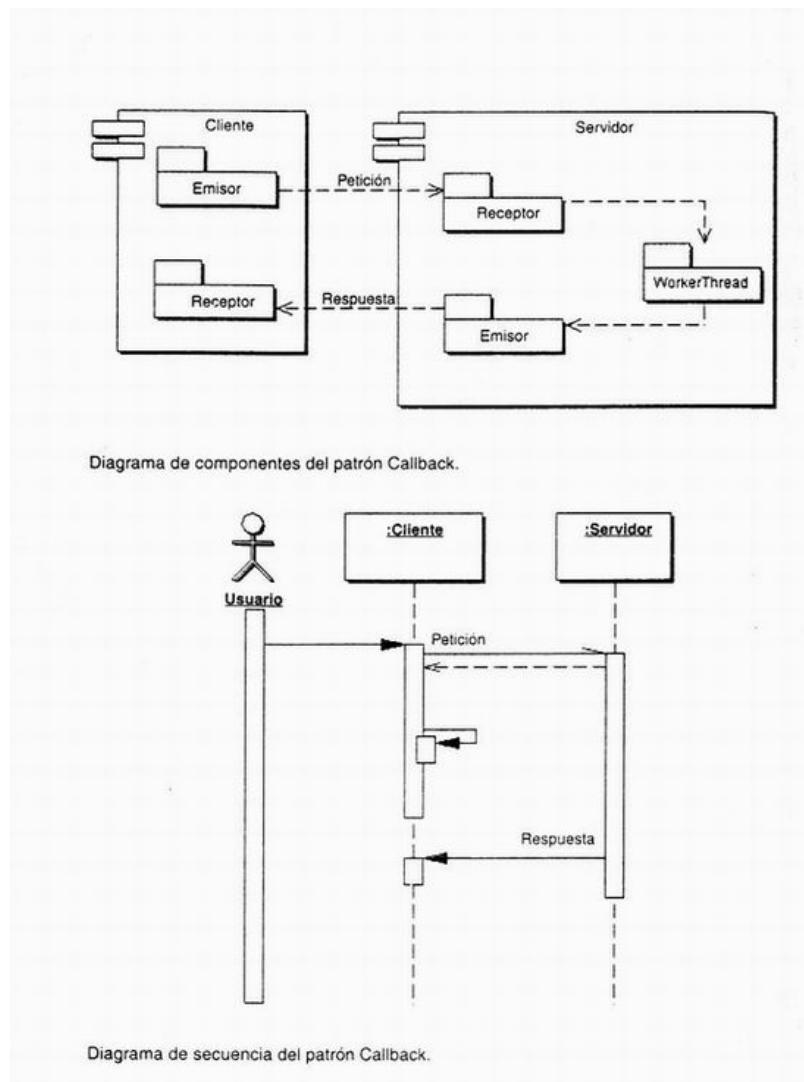


Figura 5.32: Patrón Callback

Capítulo 6

Implementación y pruebas

6.1 Implementación

En este apartado se describirán aquellos algoritmos complejos de los que disponemos a lo largo de la en la aplicación. Aparte se mostrarán las distintas pruebas que se han llevado a cabo.

6.1.1 Recomendación de Partidos

Para la recomendación de partidos se desarrolló un algoritmo propio para recomendar a cada jugador partidos en función de los eventos deportivos jugados. Las recomendaciones que se van a realizar van a ser en función de los deportes de los partidos jugados, las ubicaciones y jugadores con los que jugamos en partidos anteriores y tienen buena valoracion de media. A continuación se expondrá y se explicará paso a paso el algoritmo de recomendación realizado.

El método `public List<RecomendacionDTO> findGamesRecomendados(String login)` se encuentra en el `UserService`. Primero se accederá a él en la parte del cliente desde el componente "GameList.vue" haciendo una petición GET al controlador `UserResource`. Después de recibir la respuesta se ejecutará en la parte del cliente el método `filterGamesRecomendados` que mostrarán los partidos recomendados en función de lo que el cliente clickeara previamente, es decir, si elige por jugadores, deporte o localización..

```
HTTP.get(`users/${this.WhatLogin()}/recomendados`)
    .then(response => { this.gamesRecomendados = response.data
        return response.data})
    .then(this.filterGamesRecomendados)
```

Figura 6.1: Algoritmo de Recomendación: Jugadores

Se procederá a explicar paso por paso el algoritmo del lado del Servidor:

Recomendación de partidos según jugadores

Antes de nada se recuperará los partidos jugados por ese usuario, ya que si no ha jugado ningún partido no se le podrá recomendar nada, ya que estas recomendaciones se basan en el histórico.

Después, de todos los partidos jugados se recuperará todos los jugadores de todos esos partidos y finalmente se filtrarán quedándose solo aquellos jugadores que posean una experiencia mayor que 3. Se decide quedarse solo con los jugadores con media mayor que 3 para así únicamente recomendarle al usuario jugadores que sean buenos según las valoraciones del resto de usuarios, ya que esta experiencia es la media de las valoraciones anónimas recibidas.

Después se buscarán los partidos que estén aún vigentes donde estén apuntados esos jugadores que fuimos recolectando. Finalmente se utilizará un HashMap para evitar devolver recomendaciones de partidos repetidos. Finalmente se guardarán los partidos recomendados en un *RecomendacionDTO*, donde se guardará una lista de juegos recomendados y un mensaje que indicará el porqué de esa recomendación, y este objeto se guardará en una lista de objetos, donde en la posición 0 se guardarán las recomendaciones de los jugadores.

```

if(jugados.size()!=0){
    //Recomendar partidos en los que jueguen jugadores buenos con los que ya jugó
    for(Game game:jugados){
        jugadoresJugados.addAll(playerDAO.findAllByGame(game.getIdGame()));
    }
    for(Player j :jugadoresJugados ){
        if((j.getPlayer().getExperience()>new Double(3))&&(j.getPlayer().getLogin()!=getCurrentUserWithoutAuthority().getLogin())){
            jugadoresValoradosBien.add(j);
        }
    }
    for(Player j:jugadoresValoradosBien){
        recomendados.addAll(userDAO.findRecomendadosPlayers(j.getPlayer().getLogin()).stream().map(game -> new GameDTO(game)).collect(Collectors.toList()));
    }

    Map<Long,GameDTO> mapGames=new HashMap<Long, GameDTO>(recomendados.size());
    for(GameDTO g : recomendados) {
        mapGames.put(g.getIdGame(), g);
    }
    for(Entry<Long, GameDTO> g : mapGames.entrySet()) {
        recomendadosLimpia.add(g.getValue());
    }
}

RecomendacionDTO recomendacion=new RecomendacionDTO();
recomendacion.setMensaje("Por que jugaste con ellos");
recomendacion.setGames(recomendadosLimpia);
recomendadosFinal.add(0, recomendacion);

```

Figura 6.2: Algoritmo de Recomendación: Jugadores

Recomendación de partidos según deportes

Después de haber recuperado los partidos recomendados según jugadores con los que el usuario ha jugado y poseen una buena experiencia, ahora se pasará a recomendar partidos de los deportes que el usuario suele jugar.

Primero se recorrerán todos los partidos jugados, se irán guardando en un array de deportes en la posición de cada idSport de cada deporte el número que aparece ese deporte en la lista de partidos jugados. El deporte que aparezca más veces entre los más jugados se utilizará para buscar los partidos recomendados por deportes, buscando los partidos vigentes que sean de ese tipo de Deportes.

Además decidimos que en caso de empate, se recomendarán ambos partidos, así que buscamos si algún otro deportes consiguió el mismo número de eventos jugados y buscamos sus partidos recomendados y los unimos a los anteriores. De esta forma tenemos ya el array de partidos recomendados según los deportes, que se filtrarán a través del HashMap y se almacenará en el objeto *RecomendacionDTO* que finalmente se meterá en la posición 1 de la lista de objetos.

```
for(Game game:jugados){
    sports[game.getSport().getIdSport().intValue()]= (sports[game.getSport().getIdSport().intValue()]+1);
}

Integer iNumeroMayor = sports[1];
Integer iPosicion = 0;

for (int x=1;x<sports.length;x++){
    if (sports[x]>=iNumeroMayor){

        iNumeroMayor = sports[x];
        iPosicion = x;
    }
}
List<GameDTO> gamesDeportes= gameDAO.findAllSport(iPosicion.longValue()).stream().map(game -> new GameDTO(game)).collect(Collectors.toList());

for (int x=1;x<sports.length;x++){

    if (sports[x]==iNumeroMayor){
        iPosicion = x;
        List<GameDTO> gamesDeportesSame= gameDAO.findAllSport(iPosicion.longValue()).stream().map(game -> new GameDTO(game)).collect(Collectors.toList());
        gamesDeportes.addAll(gamesDeportesSame);
    }
}
Map<Long,GameDTO> mapGamesSport=new HashMap<Long, GameDTO>(gamesDeportes.size());
for(GameDTO g : gamesDeportes) {
    mapGamesSport.put(g.getIdGame(), g);
}
for(Entry<Long, GameDTO> g : mapGamesSport.entrySet()) {
    recomendadosLimpi1.addValue(g.getValue());
}

RecomendacionDTO recomendacion1=new RecomendacionDTO();
recomendacion1.setMensaje("Por que te gusta este Deporte");
recomendacion1.setGames(recomendadosLimpi1);
recomendadosFinal.add(1, recomendacion1);
```

Figura 6.3: Algoritmo de Recomendación: Deportes

Recomendación de partidos según localización

Y finalmente haremos lo mismo con las localizaciones. El procedimiento de los partidos recomendados según las localizaciones es el mismo que el de los deportes, solo que estos partidos recomendados se guardarán en la posición 2 de la lista de objetos.

```

for(Game game:jugados){
    locations[game.getLocation().getIdLocation().intValue()]= (locations[game.getLocation().getIdLocation().intValue()]+1);
}

Integer iNumeroMayorL = locations[1];
Integer iPosicionL = 0;

for (int x=1;x<locations.length;x++){
    if (locations[x]>=iNumeroMayorL){
        iNumeroMayorL = locations[x];
        iPosicionL = x;
    }
}

List<GameDTO> gamesLocations= gameDAO.findAllLocation(iPosicionL.longValue()).stream().map(game -> new GameDTO(game)).collect(Collectors.toList());

for (int x=1;x<locations.length;x++){

    if (locations[x]==iNumeroMayorL){
        iPosicion = x;
        List<GameDTO> gamesLocationsSame= gameDAO.findAllLocation(iPosicion.longValue()).stream().map(game -> new GameDTO(game)).collect(Collectors.toList());
        gamesLocations.addAll(gamesLocationsSame);
    }
}

Map<Long,GameDTO> mapGamesLocations=new HashMap<Long, GameDTO>(gamesLocations.size());
for(GameDTO g : gamesLocations) {
    mapGamesLocations.put(g.getIdGame(), g);
}
for(Entry<Long, GameDTO> g : mapGamesLocations.entrySet()) {
    recomendadosLimpia2.addValue(g.getValue());
}

RecomendacionDTO recomendacion2=new RecomendacionDTO();
recomendacion2.setMensaje("Por que te gusta esta ubicación");
recomendacion2.setGames(recomendadosLimpia2);
recomendadosFinal.add(2, recomendacion2);

}

return recomendadosFinal;
}

```

Figura 6.4: Algoritmo de Recomendación: Localizaciones

```

public class RecomendacionDTO {

    @NotEmpty
    private String mensaje;
    @NotNull
    private List<GameDTO> games;

```

Figura 6.5: RecomendacionDTO

6.1.2 Motor de Renderizado de pantallas

Como se vió previamente, uno de los objetivos principales de la aplicación era adaptar la aplicación para que la adición de nuevos deportes fuera de la forma más sencilla y eficiente posible. En la actualidad la plataforma está preparada para gestionar partidos de fútbol, tennis, padel y baloncesto, pero en un futuro pueden aparecer más, por lo que tenemos que hacer el código lo mas mantenible posible.

Para alcanzar este objetivo lo se va a hacer es que cada deporte almacene cual sería el nombre de la plantilla para llenar los resultados y la plantilla para visualizarlos, que realmente serían las 2 únicas interfaces que variarían según el deporte, además de que estas se podrían modificar por parte del administrador directamente desde la app sin modificar ninguna parte del código, ya que cargará una plantilla u otra en función de esos valores.

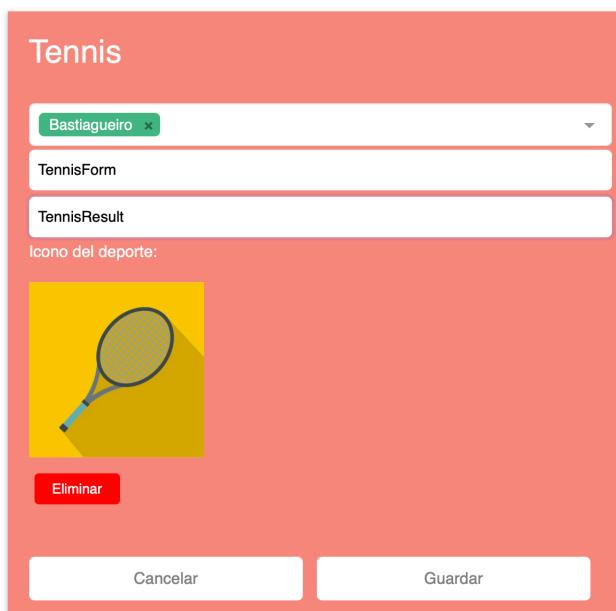


Figura 6.6: Componentes de visualización de resultado según deporte

```
this.$router.replace({ name: this.gameSelect.sport.componenteEntrada,
  params: { id:this.gameSelect.idGame}})
```

Figura 6.7: Cambio de plantilla según valor del componente de entrada

En esta aplicación se decidió únicamente desarrollar 2 plantillas de formulario y otras 2 de visualización. La plantilla *TennisFrom* y *FutbolForm* para llenar los resultados tanto de

Paddel y Tennis como de Fútbol y Baloncesto, y las plantillas *TennisResult* y *FutbolResult* para visualizar los resultados. Se decidió reutilizar las plantillas adaptándolas para que dieran soporte a 2 deportes diferentes ya que ambos poseían grandes similitudes y variaban en muy pocos factores, tanto la plantilla para Tennis y Paddel por un lado como para Fútbol y Baloncesto por otro.

Pero para conseguir esto, el resultado que se guarde en la BD debe ser de un formato que nos permita guardar los resultados de todos los deportes dados de alta en la aplicación, es decir, que nos permita guardar los sets y juegos de un partido de Tennis o Paddel como los goles o puntos de un partido de Fútbol y Baloncesto, por lo que el resultado se decidió guardar en formato JSON, luego cada plantilla dependiendo de a que deportes vaya dirigida hará un tratamiento diferente de este JSON.

```

    ▶ creator: {idUser: 4, login: "sandra", email: "vicfici23@gmail.com", password: null, authority: "USER",...}
      date: "2019-05-30"
      idGame: 2
    ▶ location: {idLocation: 2, name: "Bastiagueiro", latitud: 43.340167, longitud: -8.35426}
      maxPlayers: 4
      minPlayers: 2
    result: "{\"equipoA\":{\"setsGanados\":1,\"sets\":[\"1\",\"6\",\"1\"]},\"equipoB\":{\"setsGanados\":2,\"sets\":[\"6\",\"2\",\"6\"]}}"
  ▶ sport: {idSport: 2, type: "Tennis", componenteEntrada: "TennisForm", componenteVisualizacion: "TennisResult",...}
    componenteEntrada: "TennisForm"
    componenteVisualizacion: "TennisResult"
  
```

Figura 6.8: Resultado partido de Tennis (JSON)

```

    ▶ {idGame: 6, date: "2019-06-04", timeStart: "12:35:00", timeEnd: "12:36:00", maxPlayers: 4,...}
      ▶ creator: {idUser: 4, login: "sandra", email: "vicfici23@gmail.com", password: null, authority: "USER",...}
        date: "2019-06-04"
        idGame: 6
      ▶ location: {idLocation: 1, name: "Parque de Oza", latitud: 43.350538, longitud: -8.401573}
        maxPlayers: 4
        minPlayers: 2
      result: "{\"equipoA\":{\"goles\":\"2\",\"jugadoresA\":[{\"id\":\"sandra\",\"goles\":\"2\"}]},\"equipoB\":{\"goles\":\"1\",\"jugadoresB\":[{\"id\":\"laura\",\"goles\":\"1\"}]}}"
    ▶ sport: {idSport: 1, type: "Futbol", componenteEntrada: "FutbolForm", componenteVisualizacion: "FutbolResult",...}
      componenteEntrada: "FutbolForm"
  
```

Figura 6.9: Resultado partido de Fútbol (JSON)

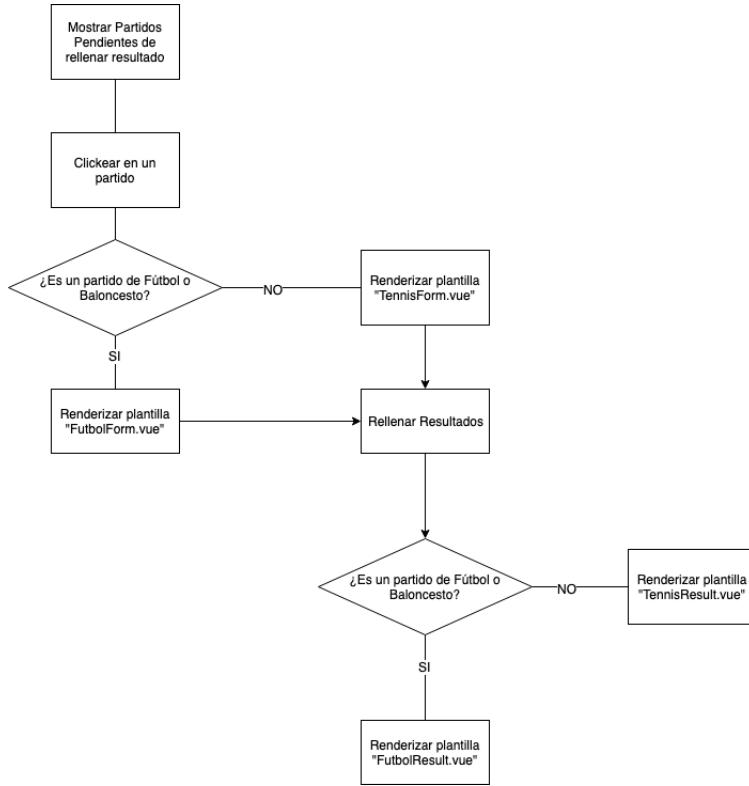


Figura 6.10: Funcionamiento Motor Plantillas

6.2 Pruebas

6.2.1 Pruebas Unitarias

Estas pruebas se han llevado a cabo mediante la ejecución de las diferentes funcionalidades una por una comprobando que todas funcionan correctamente revisando continuamente la BD para observar que no se hayan producido cambios indevidos y que todo sigue su esperado trayecto.

6.2.2 Pruebas REST

Para comprobar que todas las funcionalidades Rest se ha utilizado Post-man, una herramienta que nos permite realizar peticiones HTTP a cualquier API para comprobar el correcto funcionamiento de nuestros desarrollos.

6.2.3 Pruebas de integración y aceptación

Estas pruebas se han llevado a cabo mediante la ejecución de los diferentes casos de uso simulando ser cada uno de los usuarios ofrecidos por el sistema y comprobar que todas las operaciones del sistema funcionan correctamente, comprobando los permisos de cada usuario, creando usuarios, partidos, poniendo en prueba las partes sociales de la aplicación, en resumen, comprobar que toda la aplicación en conjunto funciona correctamente.

Capítulo 7

Solución desarrollada

En este apartado se mostrará una pequeña guía sobre el funcionamiento básico de la aplicación, centrándonos en las operaciones principales. Comenzaremos con lo que son las páginas de acceso a la aplicación, luego con algunas funcionalidades del administrador y finalmente las operaciones esenciales de los usuarios.

7.1 Acceso a la Aplicación

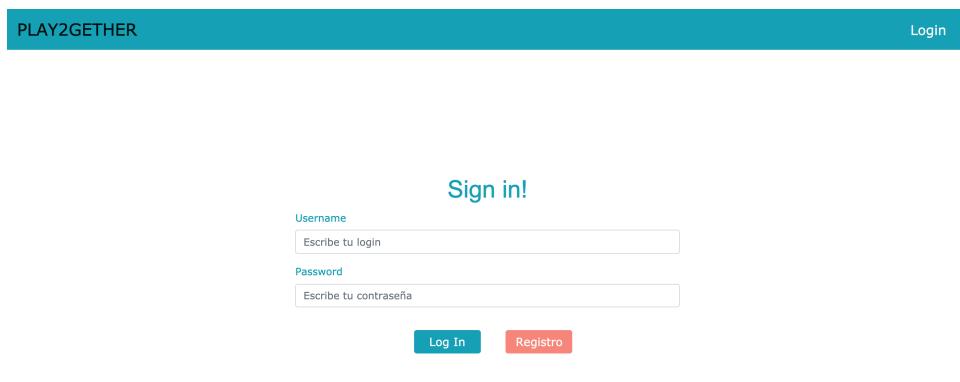
Hay 2 formas de acceder a la aplicación, o logueándote en ella o registrándose rellenando los campos obligatorios. Esto es debido a que no podremos acceder a ninguna funcionalidad a no ser que estemos autenticados. Para loguearse como administrador, el login y contraseña será *pepe* y para loguearse como usuario básico, el login y contraseña será *sandra*, teniendo otros usuarios dados de alta como *laura* o *pepe*

7.2 Administración

Las funcionalidades básicas que trataremos en este apartado serán la gestión de deportes, localizaciones y usuarios.

Tendremos la posibilidad tanto de visualizar, editar y dar de alta deportes en el apartado *Deportes* de la barra de navegación. También se dispondrá de las mismas funcionalidades para las localizaciones, pero con la ayuda de un mapa para poder seleccionar las coordenadas únicamente clickeando en él. Finalmente en la gestión de usuarios se permitirá al administrador dar de baja usuarios.

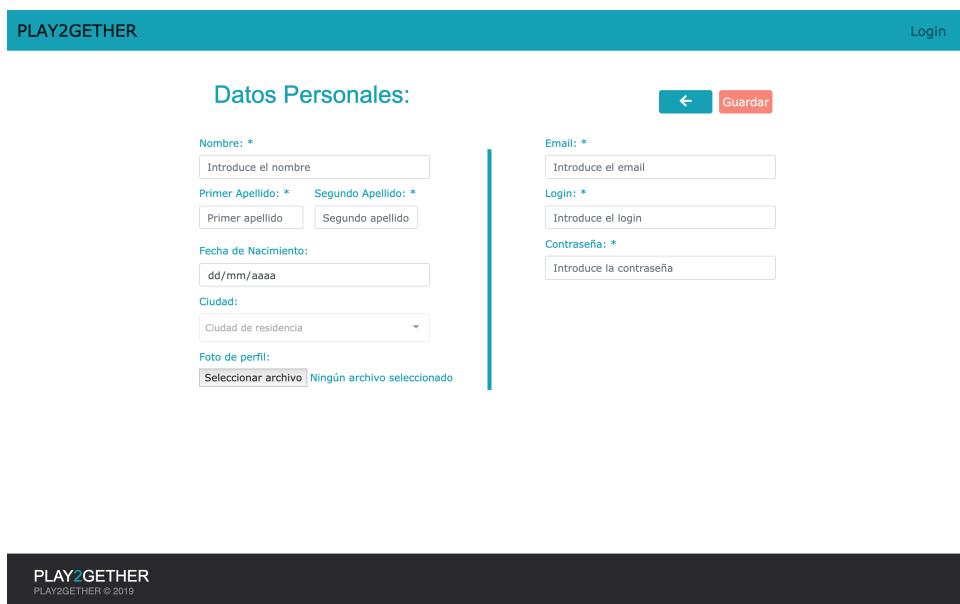
7.2. Administración



The screenshot shows the login page for the PLAY2GETHER application. At the top, there is a teal header bar with the "PLAY2GETHER" logo on the left and a "Login" button on the right. Below the header, the main content area has a light gray background. In the center, the text "Sign in!" is displayed in a bold, blue font. Below this, there are two input fields: "Username" and "Password", each with a placeholder text ("Escribe tu login" and "Escribe tu contraseña" respectively). Underneath the password field is a "Forgot password?" link. At the bottom of the form are two buttons: "Log In" (in green) and "Registro" (in red).



Figura 7.1: Página de Login



The screenshot shows the registration page for the PLAY2GETHER application. At the top, there is a teal header bar with the "PLAY2GETHER" logo on the left and a "Login" button on the right. Below the header, the main content area has a light gray background. The title "Datos Personales:" is centered at the top. To the right of the title are two small buttons: a blue one with a back arrow icon and a red one labeled "Guardar". The form itself is divided into two columns by a vertical teal line. The left column contains fields for "Nombre" (with placeholder "Introduce el nombre"), "Primer Apellido" and "Segundo Apellido" (each with a dropdown menu), "Fecha de Nacimiento" (with placeholder "dd/mm/aaaa"), "Ciudad:" (with a dropdown menu), and "Foto de perfil:" (with a file selection button). The right column contains fields for "Email:" (placeholder "Introduce el email"), "Login:" (placeholder "Introduce el login"), and "Contraseña:" (placeholder "Introduce la contraseña"). At the bottom of the page is a dark horizontal footer bar with the "PLAY2GETHER" logo and the copyright notice "PLAY2GETHER © 2019".

Figura 7.2: Página de Registro

CAPÍTULO 7. SOLUCIÓN DESARROLLADA

The screenshot shows the PLAY2GETHER application interface. At the top, there is a navigation bar with the logo 'PLAY2GETHER' and links for 'Deportes', 'Localizaciones', and 'Usuarios'. On the right side of the bar, it says 'Conectado' with a dropdown arrow. Below the navigation bar, there are two main sections. On the left, a light gray sidebar titled 'Deportes' contains a red button labeled 'NUEVO DEPORTE' and a list of existing sports: 'BALONCESTO', 'FUTBOL', 'PADDLE', and 'TENNIS'. On the right, a red-themed form titled 'Nuevo Deporte' has several input fields: 'Nombre del deporte' (with 'Parque de Oza' typed in), 'Localizaciones' (with a dropdown menu open), 'Nombre del componente de entrada', 'Nombre del componente de visualización', and 'Icono del deporte:' (with a placeholder 'Seleccionar archivo' and a note 'Ningún archivo seleccionado'). At the bottom of the form are 'Cancelar' and 'Guardar' buttons.

Figura 7.3: Página para dar de alta un deporte

The screenshot shows the PLAY2GETHER application interface. At the top, there is a navigation bar with the logo 'PLAY2GETHER' and links for 'Deportes', 'Localizaciones', and 'Usuarios'. On the right side of the bar, it says 'Conectado' with a dropdown arrow. Below the navigation bar, there are two main sections. On the left, a light gray sidebar titled 'Deportes' contains a red button labeled 'NUEVO DEPORTE' and a list of existing sports: 'BALONCESTO', 'FUTBOL', 'PADDLE', and 'TENNIS'. On the right, a red-themed form titled 'Baloncesto' displays the details for basketball: 'Nombre del deporte' (set to 'Parque de Oza'), 'Localizaciones' (with a dropdown menu open), 'Nombre del componente de entrada' (set to 'FutbolForm'), 'Nombre del componente de visualización' (set to 'FutboResult'), and an 'Icono del deporte:' section featuring a basketball icon. Below the icon is a red 'Eliminar' (Delete) button. At the bottom of the form are 'Cancelar' and 'Guardar' buttons.

Figura 7.4: Página para editar un deporte

7.2. Administración

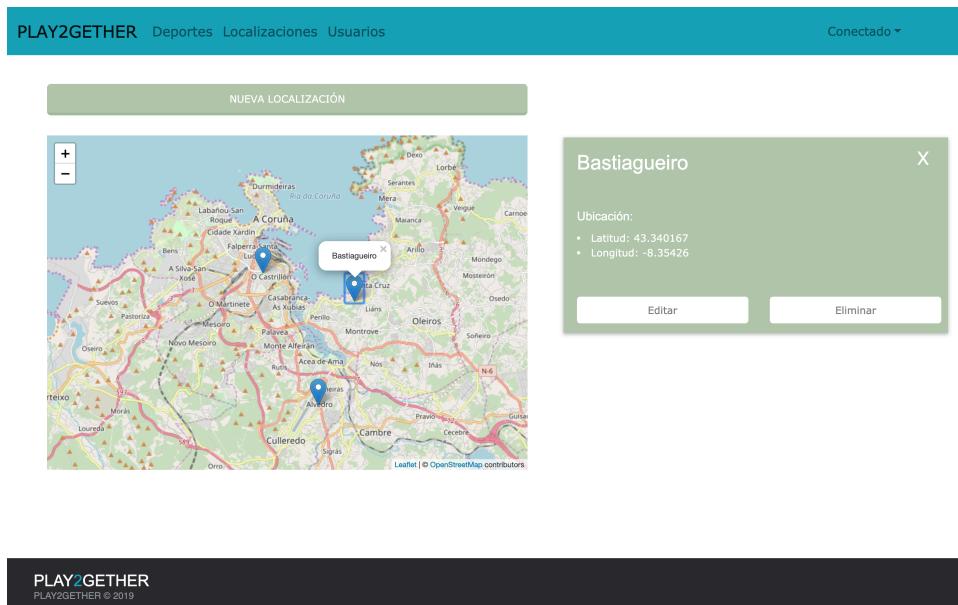


Figura 7.5: Página para visualizar una localización

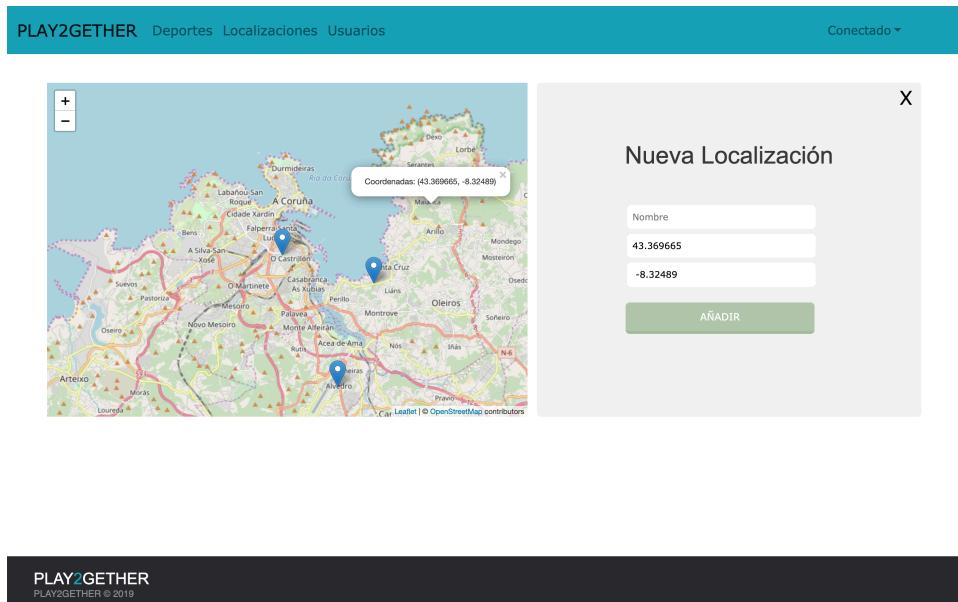


Figura 7.6: Página para dar de alta una localización

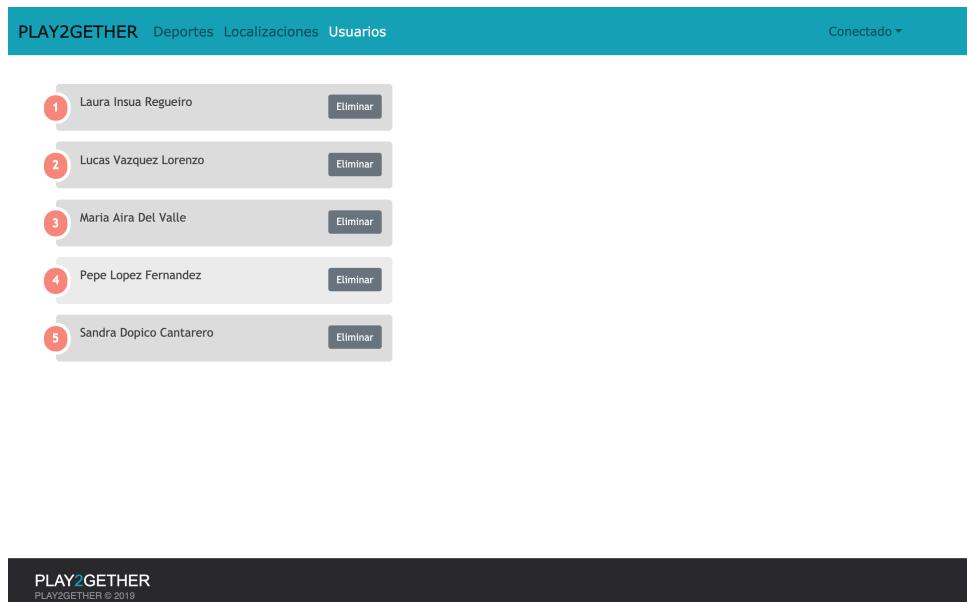


Figura 7.7: Página para dar de baja usuarios

7.3 Usuarios Básicos

En este apartado nos centraremos en las principales funcionalidades, este rol es el que abarca mayores operaciones ya que realmente el administrador se encarga únicamente de la gestión de la aplicación.

Nos permitirá buscar partidos de diferentes formas, tendremos un perfil público, donde podremos ver los diferentes partidos jugados, organizados, a los que me apunté, los partidos que me recomienda la aplicación y los comentarios que me ponen el resto de jugadores después de terminar un partido. Además podremos visualizar el Tablón de Actividades donde podremos observar que hacen las personas a las que seguimos.

7.3. Usuarios Básicos

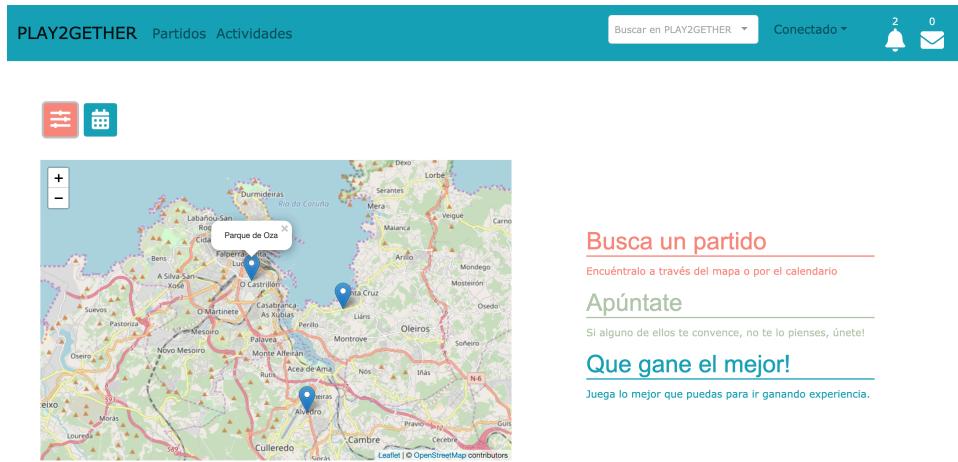


Figura 7.8: Página de búsqueda de partidos: Mapa

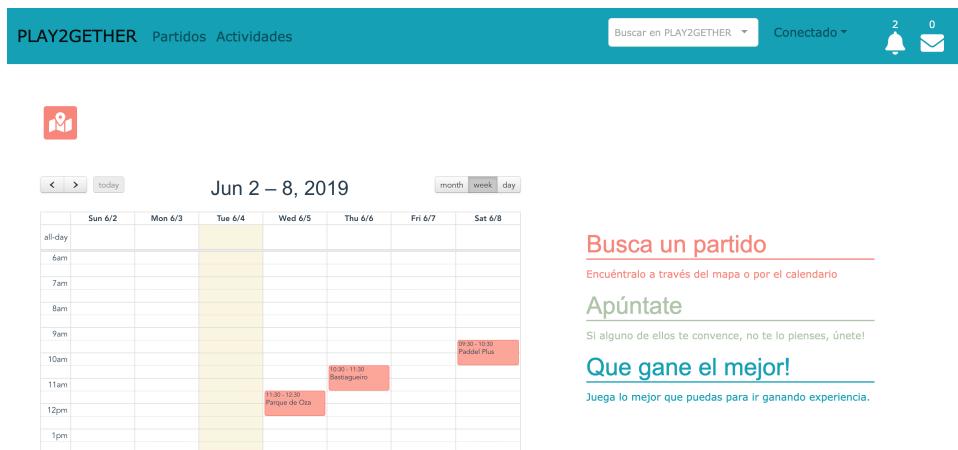


Figura 7.9: Página de búsqueda de partidos: Calendario

CAPÍTULO 7. SOLUCIÓN DESARROLLADA

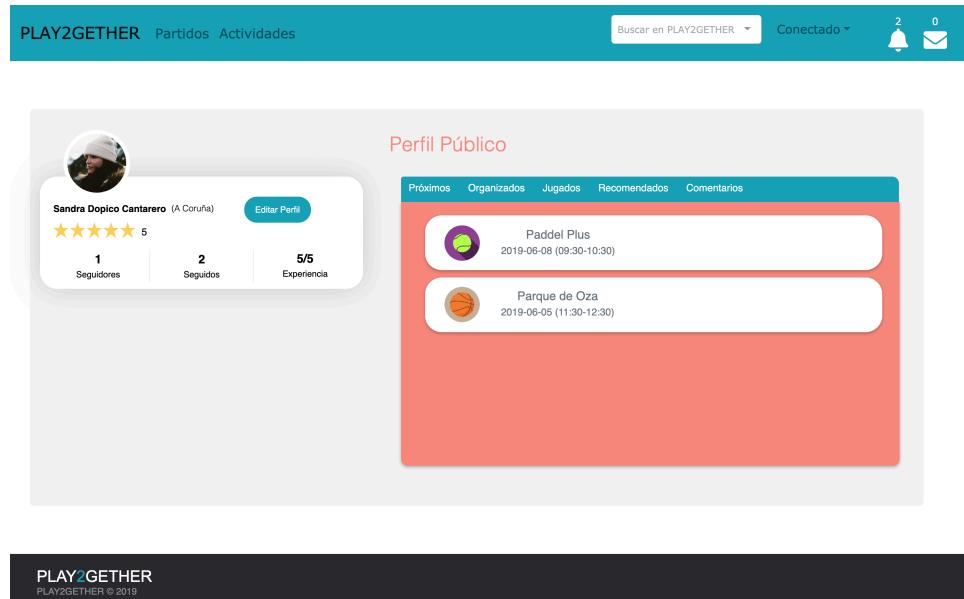


Figura 7.10: Página del Perfil Público

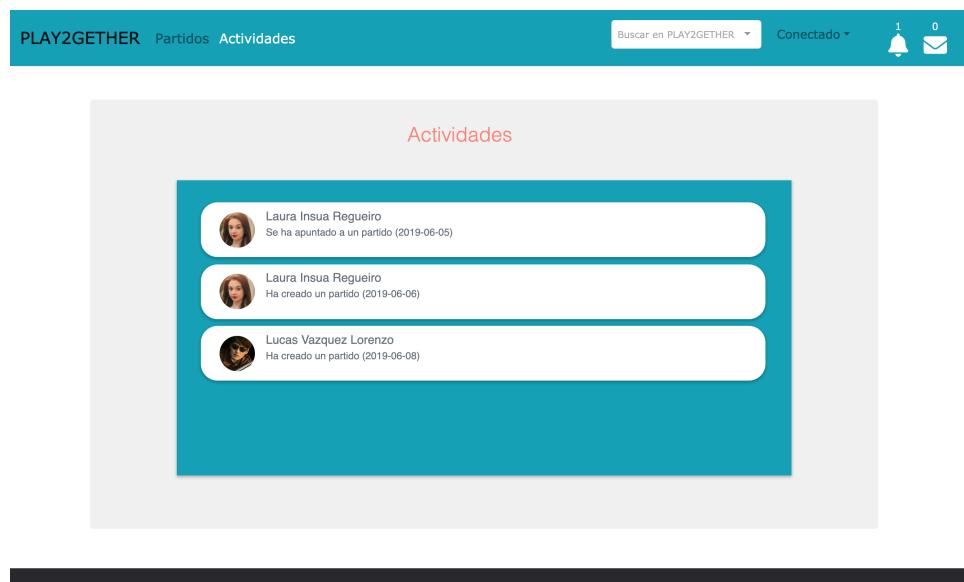


Figura 7.11: Página del Tablón de Actividades

Capítulo 8

Conclusiones y trabajo futuro

8.1 Conclusiones

Mediante la realización del Trabajo de Fin de Grado, se han podido sacar varias conclusiones:

- Los objetivos planteados inicialmente en el apartado 1.2 se han cumplido con éxito: todas las funcionalidades planteadas fueron desarrolladas y comprobadas satisfactoriamente.
- En el plano de la formación se consiguió ampliar los conocimientos en tecnologías muy presentes en el entorno empresarial, pudiendo utilizarlas de manera productiva y eficiente. Adquiriéndose experiencia en el manejo de Spring, Hibernate, servidores REST, a crear un cliente completo a través de Vue.js y a relacionar este cliente web con el servidor web mediante REST.
- Se ha tomado conciencia de la carga de trabajo y todo lo que supone un proyecto de la envergadura del realizado, dando importancia al uso de una metodología y una planificación para que el desarrollo sea ágil y constante.
- Se ha resuelto un número importante de problemas que no se habían afrontado a lo largo de los estudios de grado.
- Se ha estudiado un poco el ámbito de los deportes, más bien la parte referida a la puntuación de estos.

8.2 Trabajo Futuro

Si bien la aplicación desarrollada es completa y funcional, se podría ampliar en varios puntos:

- Aplicación móvil.
- Inclusión de HTTPS.
- Gestión de estadísticas personales a través de GoogleFit. Se podrían almacenar después de un partido los datos estadísticos durante el período del evento deportivo de esa persona, tanto pasos dados, calorías quemadas, la media de velocidad... De esta forma el jugador en sus datos personales podría realizar un seguimiento de sus constantes y ver sus mejorías a través de unos gráficos que ofrecería la aplicación explotando los datos obtenidos a lo largo del tiempo. (Business Intelligence)

Apéndices

Apéndice A

Glosario de acrónimos

REST *Representational State Transfer.*

HTML *Cascading StyleSheets.*

CSS *HyperText Markup Language.*

MVCC *Multi Version Concurrency Control.*

IDE *Integrated Development Environment.*

HTTP *HyperText Transfer Protocol.*

API *Application Programming Interface.*

ERS *Especificación de Requisitos de Software.*

MVVM *Model–View–Viewmodel.*

JDBC *Java Database Connectivity.*

DAO *Data Access Object.*

CRUD *Create, Read, Update and Delete.*

URL *Uniform Resource Locator.*

DTO *Data Transfer Object.*

MVC *Model–View–Controller.*

IoC *Inversion Of Control.*

JSON *JavaScript Object Notation.*

BD *Base de Datos.*

HTTPS *HyperText Transfer Protocol Security.*

GIS *Geographic Information System.*

Apéndice B

Manual de Instalación

Para poder desplegar la aplicación en un ordenador es necesario disponer de:

- Una máquina virtual Java
- Sistema de gestión de base de datos de PostgreSQL.
- Un servidor de aplicaciones, Apache Tomcat
- Node.js versión 8.12
- Apache Maven 3.3.9

Los pasos necesarios para desplegar la aplicación son:

- Crear una bases de datos llamadas tfg creando un usuario llamado asi, con la contraseña asi y que tenga como puerto el 5432. Cabe destacar que la aplicación ya crea automáticamente las tablas como se puede observar en el *application.yml* gracias a la opción *hibernate.ddl-auto: create*
- En *src/main/java/es.udc.lbd.asi.restexample.config/DatabaseLoader.java* se encuentran los datos iniciales que se cargarán en el momento de desplegarse la aplicación.
- Para arrancar el servidor, solo se ha de ir a la carpeta *rest-example-master* y ejecutar el comando *mvn spring-boot:run*
- Para arrancar el cliente, solo se ha de ir a la carpeta *vue-example-master* y ejecutar el comando *npm install* y a continuación *npm run debug* y final mente ir a un navegador y lanzar *http://localhost:1234* para ir a la aplicación. Desde ahí si vamos al apartado de Login podemos loguearnos como administrador: contraseña y usuario *pepe* o como usuario básico, contraseña y usuario *sandra*

Apéndice C

Contenido del CD

La memoria contendría un CD que contendrá:

- **DopicoCantareroSandraTFG2019.pdf:** La memoria del TFG en formato PDF.
- **DopicoCantareroSandraTFG2019resumo.pdf:** El resumen del proyecto en formato PDF.
- **DopicoCantareroSandraTFG2019.zip:** Zip que contiene los archivos necesarios para el despliegue de la aplicación junto con un pequeño manual. Por un lado tenemos la carpeta *rest-example-master*, que contiene el código fuente del servidor, y por otro lado *vue-example-master*, que contiene le código fuente del cliente.

Apéndice D

Mockups

A continuación se incorporarán todos los mockups realizados en la Fase Previa, explicada en el apartado 3.1.1. Cabe comentar que estas pantallas son una idea previa de lo que se quería realizar para poder planificar el trabajo del proyecto y tener una base por la cual empezar, por lo que habrá cambios respecto a las pantallas de la aplicación final.

http://Play2gether.com

PLAY & GETHER

Iniciar Sesión

Filtrar por:

Bienvenido a PLAY2GETHER, selecciona un partido, apúntate y empieza a jugar!

Mapa Calendario

Mapa

Calendario

http://Play2gether.com

PLAY & GETHER

Iniciar Sesión

Filtrar por

Edad:

- <18
- 18<edad<25
- 25<edad<40
- >40

Experiencia: ★★★★★

Creador:

Deporte:

- Fútbol Equipos Q
- Tenis
- Baloncesto
- Paddle

[> Solicitar nuevo Deporte](#)

Aplicar

Mapa Calendario

The screenshot shows a web-based application interface for 'PLAY & GETHER'. On the left, there's a sidebar with filtering options: Age (with checkboxes for <18, 18-25*, 25-40, >40, where 18-25 is checked), Experience (5 stars), Creator (empty input field), and Sport (checkboxes for Fútbol*, Tenis, Baloncesto, Paddle, where Fútbol is checked). Below these are links for 'Solicitar nuevo Deporte' and 'Aplicar'. The main area features a header 'PLAY & GETHER' with an 'Iniciar Sesión' button, a search bar, and navigation icons. Below the header is a map with a green shaded region and a yellow boundary line. At the bottom right is a small double-page icon.

http://Play2gether.com

PLAY & GETHER

Iniciar Sesión

Filtrar por

Mapa Calendario

FÚTBOL

Pabellón Otero Pedrayo

hora: 18:30
fecha: 18/10/2018

[Ver Partido](#)

The screenshot shows a web browser displaying the Play & Gether website. The URL http://Play2gether.com is visible in the address bar. The main header features the text "PLAY & GETHER". On the right side of the header is a "Iniciar Sesión" button. Below the header, there are two navigation buttons: "Filtrar por" on the left and "Mapa" and "Calendario" on the right. The main content area contains a map with several location markers. A green polygon highlights a specific area, and a yellow line outlines a larger region. A callout box on the left provides details about a football match: "FÚTBOL", "Pabellón Otero Pedrayo", "hora: 18:30", "fecha: 18/10/2018", and a link "Ver Partido". The bottom right corner of the map area has a small double-page icon.

http://

 Buscar Partidos  PLAY GETHER

 Compartir  Foro

Información del Partido

 Creador: Jose Luis
UBICACIÓN: Polideportivo María Pita
HORA: 18:30
FECHA: 1/10/2018
DEPORTE: Fútbol
PRECIO/persona: 5€

Jugadores Apuntados

 Jose Manuel
 Rosalia Ares
 Luis Conde
 Roman Junior
 Pedro Munir
 Laura Dopico



 Se espera una probabilidad de lluvias torrenciales del 90%

http:// 

 Buscar Partidos **PLAY GETHER**  Iniciar Sesión

 Compartir  Foro Apuntarse

Información del Partido  Jugadores Anuntados

 Creador: Jose Luis	Alerta Para poder unirse a este partido debe de iniciar sesión <input type="button" value="Aceptar"/>	 Pedro Munir
UBICACIÓN: Polideportivo María		 Laura Dopico
HORA: 18:30		
FECHA: 1/10/2018	Luis Conde	
DEPORTE: Fútbol		Roman Junior
PRECIO/persona: 5€		

 Se espera una probabilidad de lluvias torrenciales del 90%

The screenshot shows a web browser window with the following details:

- Address Bar:** http://
- Title Bar:** PLAY GETHER
- Content Area:**
 - Section Header:** Iniciar Sesión
 - Text:** LOGUIN:
 - Text:** CONTRASEÑA:
 - Button:** Entrar
 - Text:** ¿Aun no tienes una cuenta? [Regístrate»](#) ...

http://

PLAY & GETHER

Registrarse

NOMBRE:

APELLIDOS:

CIUDAD:

FECHA NACIMIENTO:

CORREO ELECTRÓNICO:



LOGUIN:

CONTRASEÑA:

CREAR PERFIL

The screenshot shows a web browser window for <http://Play2gether.com>. The main content area features the **PLAY2GETHER** logo at the top right. On the left, there's a button labeled "Filtrar por" with a left arrow icon. In the center, there are two tabs: "Mapa" (selected) and "Calendario". Below these tabs is a large map area with several location markers (pins). A yellow polygonal line highlights a specific area on the map. In the bottom-left corner of the map area, there are icons of a male and female figure. To the left of the map, a speech bubble contains the text: "Bienvenido a PLAY2GETHER, selecciona un partido, apúntate y empieza a jugar!". Overlaid on the map is a larger white speech bubble containing an "Aviso" (Notice) message: "Espero que te lo hayas pasado bien en el partido que organizaste hoy en 'Otero Pedrayo a las 18:00'. ¿Preparado para llenar los resultados?". A blue rectangular button labeled "Comencemos" (Let's start) is located within this notice bubble. At the top right of the map area, there are three icons: an envelope, a bell with a '2' notification, and a user profile icon for "Sandra Dópico".

HTTP:// 

 Buscar Partidos **PLAY & GATHER**    Sandra Dopico

RESULTADO:

Información del Partido		Jugadores Apuntados	
UBICACIÓN:	Polideportivo María Pita	Jose Manuel	Goles
HORA:	18:30	Rosalia Ares	<input type="text"/>
FECHA:	1/10/2018	Luis Conde	<input type="text"/>
DEPORTE:	Fútbol	Roman Junior	<input type="text"/>
		Pedro Munir	<input type="text"/>
		Laura Dopico	<input type="text"/>
		Sandra Munir	<input type="text"/>
		Pedro Lopez	<input type="text"/>

Subir Resultados

http://Play2gether.com

PLAY & GETHER

Filtrar por:

Bienvenido a PLAY2GETHER, selecciona un partido, apúntate y empieza a jugar!

Mapa Calendario

Sandra Döpico

http://Play2gether.com

PLAY & GETHER

Filtrar por

Edad:

- <18
- 18<edad<25
- 25<edad<40
- >40

Experiencia: ★★★★★

Creador:

Deporte:

- Fútbol Equipos ?
- Tenis
- Baloncesto
- Paddle

[> Solicitar nuevo Deporte](#)

Aplicar

Mapa Calendario

Sandra Dópico

http://Play2gether.com

PLAY & GETHER

Filtrar por

Mapa Calendario

FÚTBOL

Pabellón Otero Pedrayo

hora: 18:30
fecha: 18/10/2018

[Ver Partido](#)

Sandra Döpico

The screenshot shows a web browser window for <http://Play2gether.com>. The main content area displays a map with several location markers. A green shaded area covers a portion of the map, and a yellow line highlights a specific route or boundary. A callout bubble on the right side provides details about a tennis match:

TENIS
Club Tenis "La Raqueta"
hora: 18:30
fecha: 18/10/2018
[Ver Partido](#)

At the top of the page, there are navigation icons (back, forward, search) and a user profile for Sandra Döpico.

The screenshot shows the homepage of the PLAY2GETHER website (<http://Play2gether.com>). The interface includes a navigation bar with back, forward, and search icons, and a user profile for Sandra Dópico.

Header: PLAY2GETHER

User Profile: Sandra Dópico (with a notification icon showing 2)

Left Sidebar: A speech bubble says "Bienvenido a PLAY2GETHER, selecciona un partido, apúntate y empieza a jugar!" (Welcome to PLAY2GETHER, select a game, sign up and start playing!). Below it are icons of a boy and a girl.

Top Navigation: Filtrar por, Mapa, Calendario

Map: A map showing several location markers. A green polygon highlights a specific area, and a yellow line traces a path across the map.

Right Sidebar (dropdown menu):

- Tablón de actividad
- Crear Evento
- Ver Perfil
- Cerrar Sesión

<http://Play2gether.com>

PLAY & GETHER

Filtrar por

Edad:

- <18
- 18<edad<25
- 25<edad<40
- >40

Experiencia: ★★★★★

Creador:

Deporte:

- Fútbol
- Tenis
- Baloncesto
- Paddle

[> Solicitar nuevo Deporte](#)

[Aplicar](#)

Mapa Calendario

DECEMBER 2018						
S	M	T	W	T	F	S
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

✉️ 2 Sandra Döpico

<http://Play2gether.com>

PLAY & GETHER

Filtrar por

Edad:

- <18
- 18<edad<25
- 25<edad<40
- >40

Experiencia: ★★★★★

Creador:

Deporte:

- Fútbol Equipos 
- Tenis
- Baloncesto
- Paddle

[> Solicitar nuevo Deporte](#)

[Aplicar](#)

Mapa Calendario

9/12/2018		
10:00-11:00	 Sandra Dópico	Polideportivo María Pita FÚTBOL
11:00-12:00	 Sandra Dópico	Polideportivo María Pita FÚTBOL
12:00-13:00		
13:00-14:00		

<http://Play2gether.com>

PLAY & GETHER

Filtrar por

Edad:

- <18
- 18<edad<25
- 25<edad<40
- >40

Experiencia: ★★★★☆

Creador:

Deporte:

- Fútbol
- Tenis
- Baloncesto
- Paddle

> Solicitar nuevo Deporte

Aplicar

Mapa Calendario

9/12/2018

10:00-11:00	Sandra Dópico	Polideportivo María Pita FÚTBOL
11:00-12:00	Sandra Dópico	Polideportivo María Pita FÚTBOL
12:00-13:00		+
13:00-14:00		+

Tablón de actividad

- Crear Evento
- Ver Perfil
- Cerrar Sesión

Sandra Dópico

http://Play2gether.com

Buscar Partidos

PLAY & GETHER

Sandra Döpico

Actividad de tus seguidos

 Jose Luis →
Ha creado el siguiente evento:
Polideportivo María Pita
18:30 1/10/2018
FÚTBOL

 Maria López →
Se ha apuntado al siguiente evento:
Polideportivo Montero Pedrayo
16:00 3/10/2018
TENIS

 Jorge Hostal →
Ha comentado en el tablón de este evento:
Polideportivo Ramón de la Sagra
14:45 4/10/2018
FÚTBOL

The screenshot shows a web browser window for the website <http://Play2gether.com>. The page title is "PLAY GETHER". On the left, there's a search bar labeled "Buscar Partidos" and a user profile icon for "Sandra Döpico" with a notification count of 2. The main content area is titled "CREAR PARTIDO" and contains the following fields:

Deporte	ComboBox
Ubicación:	ComboBox
Fecha/Hora	/ /
Nº máximo de jugadores	<input type="text"/>
Nº mínimo de jugadores	<input type="text"/>
Precio Total	<input type="text"/>

http://Play2gether.com

Buscar Partidos

PLAY & GETHER

Sandra Döpico

CREAR PARTIDO

Ubicación:	ComboBox
Fecha/Hora:	/ / 
Nº máximo de jugadores:	<input type="text"/>
Nº mínimo de jugadores:	<input type="text"/>
Precio Total:	<input type="text"/>

S	M	T	W	T	F	S
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

http://Play2gether.com

Buscar Partidos

PLAY & GETHER

Sandra Döpico

CREAR PARTIDO

Ubicación: ComboBox

Día/Hora: / /

Nº máximo de jugadores:

Nº mínimo de jugadores:

Precio Total:

Hora

10:00 -11:00	Atrás
11:00-12:00	
12:00-13:00	

Crear

http://Play2gether.com

Buscar Partidos

PLAY & GETHER

Sandra Döpico

CREAR PARTIDO

Ubicación: Cor

Día/Hora: /

Nº máximo de jugadores:

Nº mínimo de jugadores:

Precio Total:

Horas

Aviso

Durante este periodo de tiempo se espera una probabilidad del 90% de lluvia ¿Desea seleccionar este horario?

No Si

Crear

http://Play2gether.com

Buscar Partidos

PLAY & GETHER

Partidos Organizados Partidos Jugados Próximos Partidos Partidos Recomendados Comentarios de contrincantes (2)

Sandra Dopico



Sandra Dopico Cantarero

Seguidores: 18 Seguidos: 20

Ciudad: Coruña

Equipo Federado: Galicia Gaiteira/Modestos

Experiencia: ★★★★☆

Partidos Organizados

 2-0	Polideportivo María Pita 18:30 1/10/2018 FÚTBOL
 3-1	Polideportivo Montero Pedrayo 16:00 3/10/2018 TENIS
	Polideportivo Ramón de la Sagra 14:45 4/10/2018 FÚTBOL

HTTP://

Buscar Partidos **PLAY GETHER** Sandra Dopico

Compartir Foro **RESULTADO: 2-0**

Información del Partido		Jugadores Apuntados	
UBICACIÓN:	Polideportivo María Pita	Jose Manuel	
HORA:	18:30	Rosalia Ares	
FECHA:	1/10/2018	Luis Conde	
DEPORTE:	Fútbol	Roman Junior	
VALORACIÓN:		Pedro Munir	
		Laura Dopico	
		Sandra Munir	
		Pedro Lopez	

HTTP://

Buscar Partidos **PLAY GETHER** Sandra Dopico

Compartir Foro **RESULTADO: 2-0**

Información del Partido		Jugadores Apuntados	
UBICACIÓN:	Polideportivo María Pita	Jose Manuel	★★★★★
HORA:	18:30	Rosalia Ares	★★★★★
FECHA:	1/10/2018	Luis Conde	★★★★★
DEPORTE:	Fútbol	Roman Junior	★★★★★
VALORACIÓN:	★★★★★	Pedro Munir	★★★★★
		Laura Dopico	★★★★★
		Sandra Munir	★★★★★
		Pedro Lopez	★★★★★

Enviar Valoraciones

http://

Buscar Partidos

PLAY GETHER

Sandra Dopico

Compartir Foro

RESULTADO: 2-0

Información del Partido

Jugadores Anotados

Alerta

¿Está seguro de querer finalizar sus valoraciones?

No Si

Luis Conde

Roman Junior

Pedro Munir ★★★★★

Laura Dopico ★★★★☆

Sandra Munir ★★★★☆

Pedro Lopez ★★★★★

Enviar Valoraciones

UBICACIÓN:	Polideportivo María
HORA:	18:30
FECHA:	1/10/2018
DEPORTE:	Fútbol
VALORACIÓN:	★★★★★

Alerta	
¿Está seguro de querer finalizar sus valoraciones?	
No	Si
Luis Conde	
Roman Junior	★★★★★

Pedro Munir	★★★★★
Laura Dopico	★★★★☆
Sandra Munir	★★★★☆
Pedro Lopez	★★★★★

HTTP://

Buscar Partidos **PLAY & GETHER** Sandra Dopico

Compartir Foro

RESULTADO: 2-0

Información del Partido		Jugadores Apuntados	
UBICACIÓN:	Polideportivo María Pita	Jose Manuel	★★★★★
HORA:	18:30	Rosalia Ares	★★★★★
FECHA:	1/10/2018	Luis Conde	★★★★★
DEPORTE:	Fútbol	Roman Junior	★★★★★
VALORACIÓN:	★★★★★	Pedro Munir	★★★★★
		Laura Dopico	★★★★★
		Sandra Munir	★★★★★
		Pedro Lopez	★★★★★

HTTP://

Buscar Partidos **PLAY & GETHER** Sandra Dopico

Compartir Foro Cancelar partido

Información del Partido

UBICACIÓN: Polideportivo María Pita
HORA: 18:30
FECHA: 1/10/2018
DEPORTE: Fútbol
PRECIO/persona: 5€

Jugadores Apuntados

Jose Manuel	Pedro Munir
Rosalia Ares	Laura Dopico
Luis Conde	Sandra Munir
Roman Junior	Pedro Lopez

Se espera una probabilidad de lluvias torrenciales del 90%

http://

Buscar Partidos

PLAY & GETHER

Sandra Dopico

Compartir Foro Cancelar partido

Información del Partido

Ubicación: Polideportivo María P.

Hora: 18:30

Fecha: 1/10/2018

Deporte: Fútbol

Precio/persone: 5€

Jugadores Anotados

Alerta

¿Está seguro de querer eliminar este evento?

No Si

Luis Conde

Roman Junior

Pedro Munir
Laura Dopico
Sandra Munir
Pedro Lopez

Se espera una probabilidad de lluvias torrenciales del 90%

<http://>

Buscar Partidos **PLAY GETHER** Sandra Dopico

Compartir Foro **RESULTADO: 115-103**

Información del Partido		Jugadores Apuntados	
UBICACIÓN:	Urbanización Os Regos	30	Jose Manuel
HORA:	15:30	30	Pedro Munir
FECHA:	4/10/2018	10	Rosalia Ares
DEPORTE:	Baloncesto	20	Luis Conde
VALORACIÓN:		30	Roman Junior
		10	Pepe Junior
		30	Dolores Fuet
		10	Pedro Lopez

http://Play2gether.com

Buscar Partidos

PLAY & GETHER

Partidos Organizados Partidos Jugados Próximos Partidos Partidos Recomendados Comentarios de contrincantes (2)

Sandra Dopico



Sandra Dopico Cantarero

Seguidores  18	Seguidos  20
Ciudad: Coruña	
Equipo Federado: Galicia Gaiteira/Modestos	
Experiencia: 	

Proximos Partidos

Polideportivo Ramón de la Sagra
14:45 4/10/2018
FÚTBOL

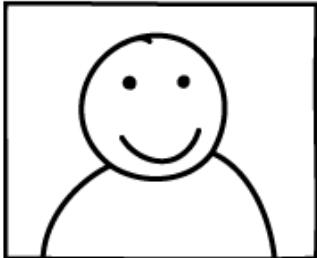
http://Play2gether.com

PLAY & GETHER

Buscar Partidos

Partidos Organizados Partidos Jugados Próximos Partidos Partidos Recomendados Comentarios de contrincantes (1)

Sandra Dopico



Sandra Dopico Cantarero

Seguidores: 18 Seguidos: 20

Ciudad: Coruña

Equipo Federado: Galicia Gaiteira/Modestos

Experiencia: ★★★★☆

Partidos Recomendados

 Jose Ignacio	Polideportivo María Pita 18:30 1/10/2018 FÚTBOL
 Lucrecia Lopez	Polideportivo Montero Pedrayo 16:00 3/10/2018 TENIS
 Fernando Fernandez	Polideportivo Ramón de la Sagra 14:45 4/10/2018 FÚTBOL

HTTP://

Buscar Partidos **PLAY & GETHER** Sandra Dópico

Compartir Notificar Foro Apuntarse

Información del Partido

Creador: Jose Luis	Puntos Jugadores	1	2	3	4	5
UBICACIÓN: Club de Tenis "Laraqueta"	Jose Manuel					
HORA: 18:30	Roman Junior					
FECHA: 1/10/2018	Pedro Munir					
DEPORTE: Tenis						
PRECIO/persona: 5€						

Se espera una probabilidad de lluvias torrenciales del 90%

HTTP://

Buscar Partidos **PLAY & GETHER** Sandra Dópico

Compartir Notificar Foro **Apuntarse**

Información del Partido

Creador: Jose Luis
UBICACIÓN: Polideportivo María Pita
HORA: 18:30
FECHA: 1/10/2018
DEPORTE: Fútbol
PRECIO/persona: 5€

Jugadores Apuntados

Jose Manuel →
Rosalia Ares
Luis Conde
Roman Junior
Pedro Munir
Laura Dópico

Se espera una probabilidad de lluvias torrenciales del 90%

http://

Buscar Partidos

PLAY & GETHER

Sandra Dopico

Compartir Notificar Foro Apuntarse

Información del Partido

Creador: Jose Luis

UBICACIÓN: Polideportivo N

HORA: 18:30

FECHA: 1/10/2018

DEPORTE: Fútbol

PRECIO/persona: 5€

Alerta

¿Está seguro de querer apuntarse a este evento?

No Si

Luis Conde

Roman Junior

Pedro Munir

Laura Dopico

Se espera una probabilidad de lluvias torrenciales del 90%

HTTP:// 

 Buscar Partidos **PLAY & GETHER**    Sandra Dopico

 Compartir  Notificar  Foro  Desapuntarse

Información del Partido

 Creador: Jose Luis
UBICACIÓN: Polideportivo María Pita
HORA: 18:30
FECHA: 1/10/2018
DEPORTE: Fútbol

Jugadores Apuntados

 Jose Manuel
 Rosalia Ares
 Luis Conde
 Roman Junior
 Pedro Munir
 Laura Dopico



 Se espera una probabilidad de lluvias torrenciales del 90%

http://

Buscar Partidos

PLAY & GETHER

Sandra Dopico

Compartir Notificar Foro Desapuntarse

Información del Partido

Creador: Jose Luis

UBICACIÓN: Polideportivo N

HORA: 18:30

FECHA: 1/10/2018

DEPORTE: Fútbol

PRECIO/persona: 5€

Alerta

¿Está seguro de querer cancelar la suscripción a este evento?

No Si

Luis Conde

Roman Junior

Pedro Munir

Laura Dopico

Se espera una probabilidad de lluvias torrenciales del 90%

http://

Buscar Partidos

PLAY & GETHER

Sandra Dópico

Añadir Comentario

Apuntarse

Información del Partido

Creador: Jose Luis

UBICACIÓN: Polideportivo María Pita

HORA: 18:30

FECHA: 1/10/2018

DEPORTE: Fútbol

PRECIO/persona: 5€

Comentarios

 Lucas Porter	***** ***** ***** ***** *****
 María Asunción	***** ***** ***** ***** *****
 Carla Montes	***** ***** ***** ***** *****

The screenshot shows a web browser window with the URL <http://> in the address bar. The page is titled "PLAY GETHER". On the left, there's a search bar labeled "Buscar Partidos" and a user profile for "Sandra Dópico" with a notification count of 2. On the right, there are icons for messaging and a profile picture. A large central form is displayed for creating a new match. It includes a button "Añadir Comentario" with a plus sign, a section for "Información del Partido" with a creator icon and name "Creador: Jose L.", and fields for "UBICACIÓN: Polidepor", "HORA: 18:30", "FECHA: 1/10/2018", "DEPORTE: Fútbol", and "PRECIO/persona: 5€". The main text area for the match description is filled with numerous blacked-out censored text lines. At the bottom of this area are two buttons: "Enviar" (Send) and "Cancelar" (Cancel). To the right of the main form, there are three smaller, partially visible rectangular boxes, each containing several censored lines of text.

http://Play2gether.com

Buscar Partidos

PLAY & GETHER

Sandra Dopico

Partidos Organizados Partidos Jugados Próximos Partidos Partidos Recomendados Comentarios de contrincantes (1)



Sandra Dopico Cantarero

Seguidores  18	Seguidos  20
--	--

Ciudad: Coruña

Equipo Federado: Galicia Gaiteira/Modestos

Experiencia: 

Partidos Jugados

 2-0	Polideportivo María Pita 18:30 1/10/2018 FÚTBOL
 3-1	Club Tenis "La Raqueta" 16:00 3/10/2018 TENIS

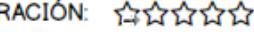
HTTP:// 

 Buscar Partidos **PLAY & GETHER**    Sandra Dópico

 Compartir  Foro **RESULTADO: 3-1**

	1	2	3	4	5
Jose Manuel	5	6	2	6	
Roman Junior	5	6	2	6	
Pedro Munir	5	2	4	6	1
Laura Dópico	5	2	4	6	1

Información del Partido

UBICACIÓN: Club de Tenis "Laraqueta"
HORA: 18:30
FECHA: 1/10/2018
DEPORTE: Tenis
VALORACIÓN: 

Puntos Jugadores

Jugador	1	2	3	4	5
Jose Manuel	5	6	2	6	
Roman Junior	5	6	2	6	
Pedro Munir	5	2	4	6	1
Laura Dópico	5	2	4	6	1

<http://>

Buscar Partidos **PLAY & GETHER** Sandra Dópico

Compartir Foro

RESULTADO: 3-1

	1	2	3	4	5
Puntos Jugadores					
Jose Manuel	★★★☆☆	6	6	2	6
Roman Junior	★★☆☆☆				
Pedro Munir	★★★★☆	2	4	6	1
Laura Dópico	★★☆☆☆				

Información del Partido

UBICACIÓN: Club de Tenis "Laraqueta"
HORA: 18:30
FECHA: 1/10/2018
DEPORTE: Tenis
VALORACIÓN: ★★★★☆

[Enviar Valoraciones](#)

HTTP:// 

 Buscar Partidos **PLAY & GETHER**    Sandra Dópico

 Compartir  Foro

Información del Partido

UBICACIÓN: Club d...
HORA: 18:30
FECHA: 1/10/2...
DEPORTE: Tenis
VALORACIÓN:   

Comentario sobre tu valoración:

RESULTADO: 3-1

2	3	4	5
6	2	6	
4	6	1	

HTTP://

Buscar Partidos **PLAY & GETHER** Sandra Dópico

Compartir Foro

RESULTADO: 3-1

	1	2	3	4	5
Información del Partido					
UBICACIÓN:	Club de Tenis "Lara"				
HORA:	18:30				
FECHA:	1/10/2018				
DEPORTE:	Tenis				
VALORACIÓN:	★★★★★				
Puntos Jugadores	6	6	2	6	
	2	4	6	1	

Alerta
¿Está seguro de querer finalizar sus valoraciones?

No Si

Pedro Munir
 Laura Dópico

Enviar Valoraciones

HTTP:// 

 Buscar Partidos **PLAY & GETHER**    Sandra Dópico

 Compartir  Foro

RESULTADO: 3-1

Información del Partido		Puntos Jugadores					
		1	2	3	4	5	
UBICACIÓN:	Club de Tenis "Laraqueta"	Jose Manuel	★★★☆☆	6	6	2	6
HORA:	18:30	Roman Junior	★★☆☆☆				
FECHA:	1/10/2018	Pedro Munir	★★★★☆	2	4	6	1
DEPORTE:	Tenis	Laura Dópico	★★☆☆☆				
VALORACIÓN:	★★★★☆						

http://

Buscar Partidos

PLAY GETHER

Sandra Dópico

Comentarios de los contrincantes

Comentarios
 Lucas Porter
 María Asunción
 Carla Montes

http://Play2gether.com

PLAY & GETHER

Buscar Partidos

Partidos Organizados Partidos Jugados Próximos Partidos Partidos Recomendados Comentarios

Sandra Dopico

PENDIENTES DE VALORAR

Partidos
Polideportivo Maria Pita Yo Valorar
Club de Tenis "La Raqueta" Marcos Casal Valorar
Polideportivo Montero Pedrayo 16:00 3/10/2018 TENIS
Polideportivo Ramón de la Sagra 14:45 4/10/2018 FÚTBOL

Sandra Dopico Cantarero

Seguidores: 18 Seguidos: 20

Ciudad: Coruña

Equipo Federado: Galicia Gaiteira/Modestos

Experiencia: ★★★★☆

http://Play2gether.com

PLAY & GETHER

Buscar Partidos

Partidos Organizados Partidos Jugados Próximos Partidos Partidos Recomendados Comentarios de contrincantes (2)

Sandra Dopico



Sandra Dopico Cantarero

Seguidores	18
Seguidos	20

Ciudad: Coruña

Equipo Federado: Galicia Gaiteira/Modestos

Experiencia: ★★★★☆

Seguidores

	Rosalia Ramirez Mena	<button>seguir</button>
	Juan Manuel Montes	<button>seguir</button>
	Lucas Ramón Sanjurjo	<button>siguiendo</button>

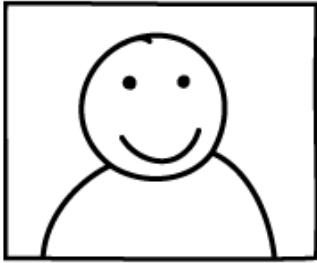
http://Play2gether.com

Buscar Partidos

PLAY & GETHER

Partidos Organizados Partidos Jugados Próximos Partidos Partidos Recomendados Comentarios de contrincantes (1)

Sandra Dopico



Sandra Dopico Cantarero

Seguidores	Seguidos
18	20

Ciudad: Coruña

Equipo Federado: Galicia Gaiteira/Modestos

Experiencia: ★★★★☆

Seguidos

	Rosalia Ramirez Mena	siguiendo
	Juan Pereiro Montes	sigueiendo
	Lucas Ramón Sanjurjo	siguiendo

http://Play2gether.com

PLAY & GETHER

Buscar Partidos

Partidos Organizados Próximos Partidos Comentarios de contrincantes (8)

Sandra Dopico

Jose Manuel López Siguiendo

Seguidores 18 Seguidos 20

Ciudad: Coruña

Equipo Federado: Galicia Gaiteira/Modestos

Experiencia: ★★★★☆

Mensaje Notificar

Partidos Organizados

Polideportivo María Pita 18:30 1/10/2018 FÚTBOL
Polideportivo Montero Pedrayo 16:00 3/10/2018 TENIS
Urbanización Os Regos 15:30 5/10/2018 BALONCESTO
Polideportivo Ramón de la Sagra 14:45 4/10/2018 FÚTBOL

http://Play2.gether.com

PLAY & GATHER

Buscar Partidos

Partidos Organizados Próximos Partidos Comentarios de contrincantes (2)

Sandra Dopico

Bloquear Usuario Compartir Perfil

Jose Manuel López Siguendo

Seguidores 18 Seguidos 20 Ciudad: Coruña Equipo Federado: Galicia Gaiteira/Modestos Experiencia: ★★★★☆

Mensaje Notificar

Partidos Organizados

2-0	Polideportivo María Pita 18:30 1/10/2018 FÚTBOL
3-1	Polideportivo Montero Pedrayo 16:00 3/10/2018 TENIS
	Polideportivo Ramón de la Sagra 14:45 4/10/2018 FÚTBOL

http://

Buscar Partidos

PLAY & GETHER

Sandra Dopico

Compartir Notificar Foro Apuntarse

Información del Partido

Creador: Jose Luis

UBICACIÓN: Polideportivo María Pita

HORA: 18:30

FECHA: 1/10/2018

DEPORTE: Fútbol

PRECIO/persona: 5€

Aviso

¿Seguro que quiere que le notifiquemos si se une algún jugador?

No Si

Roman Junior

Pedro Munir

Laura Dopico

Se espera una probabilidad de lluvias torrenciales del 90%

HTTP://

Buscar Partidos **PLAY & GETHER** Sandra Dopico

Compartir Notificar Foro **Apuntarse**

Información del Partido

Creador: Jose Luis
UBICACIÓN: Polideportivo María Pita
HORA: 18:30
FECHA: 1/10/2018
DEPORTE: Fútbol
PRECIO/persona: 5€

Jugadores Apuntados

Jose Manuel
Rosalia Ares
Luis Conde
Roman Junior
Pedro Munir
Laura Dopico

Se espera una probabilidad de lluvias torrenciales del 90%

HTTP://

PLAY GETHER

Sandra Dopico

Datos Personales

NOMBRE:	██████████	LOGUIN:	██████████
APELLIDOS:	██████████	EXPERIENCIA:	██████████
CIUDAD:	██████████		
FECHA NACIMIENTO:	██████████		
CORREO ELECTRÓNICO:	██████████		

JUEGO EN:

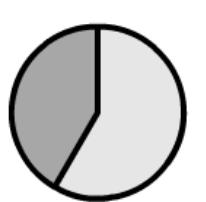
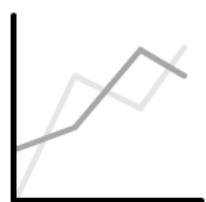
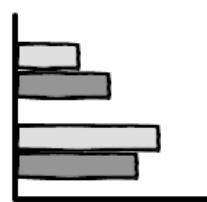
DEPORTE: Q
EQUIPO: Q

FAVORITOS:

DEPORTE: Q
EQUIPO: Q

Estadísticas

PASOS: ██████████
CAL: ██████████
VELOCIDAD: ██████████





Bibliografía

- [1] “Página web de FullCalendar,” (Consultado en Abril 2019). [Online]. Available: <https://fullcalendar.io/docs>
- [2] Evan You, “Guía sobre Vue.js,” 2013, (Consultado en Febrero 2019). [Online]. Available: <https://vuejs.org/v2/guide/>
- [3] ——, “Documentación sobre Vue-Select,” 2013, (Consultado en Febrero 2019). [Online]. Available: <https://vue-select.org>
- [4] ——, “Documentación sobre Vue-MultiSelect,” 2013, (Consultado en Febrero 2019). [Online]. Available: <https://vue-multiselect.js.org>
- [5] ——, “Documentación sobre Vue-Route,” 2013, (Consultado en Febrero 2019). [Online]. Available: <https://router.vuejs.org>
- [6] Michael Thiessen, “Documentación sobre el Renderizado de Vue,” 2018, (Consultado en Febrero 2019). [Online]. Available: <https://michaelnthiessen.com/force-re-render/>
- [7] Ryan Dahl, “Página web de Node.js,” 2009, (Consultado en Febrero 2019). [Online]. Available: <https://nodejs.org/es/docs/guides/getting-started-guide/>
- [8] Vladimir Agafonkin, “Página web de Leaflet,” 2011, (Consultado en Marzo 2019). [Online]. Available: <https://leafletjs.com/>
- [9] Mark Otto, “Página web de Bootstrap,” 2010, (Consultado en Marzo 2019). [Online]. Available: <https://getbootstrap.com/docs/4.3/getting-started/introduction/>
- [10] Christian Bauer,Gavin King, and Gary Gregory, “Página web de Hibernate,” 2001, (Consultado en Febrero 2019). [Online]. Available: <https://hibernate.org>
- [11] Baeldung, “Página web sobre la herencia en Hibernate,” 2018, (Consultado en Febrero 2019). [Online]. Available: <https://www.baeldung.com/hibernate-inheritance>

- [12] CommunityDocumentation, “Página web sobre el mapeo en Hibernate,” 2004, (Consultado en Marzo 2019). [Online]. Available: <https://docs.jboss.org/hibernate/orm/3.6/reference/es-ES/html/collections.html>
- [13] Pivotal Software, “Página web de Spring,” 2002, (Consultado en Marzo 2019). [Online]. Available: <https://spring.io/projects/spring-boot>
- [14] ——, “Página web de Spring,” 2002, (Consultado en Abril 2019). [Online]. Available: <https://www.spring.io/projects/spring-security/>
- [15] Michael Stonebraker, “Página web de PostgreSQL,” 1996, (Consultado en Febrero 2019). [Online]. Available: <https://www.postgresql.org/>
- [16] “Página web de Latex,” (Consultado en Junio 2019). [Online]. Available: <https://www.latex-project.org>
- [17] “Página web de PostMan,” (Consultado en Marzo 2019). [Online]. Available: <https://www.getpostman.com/>
- [18] “Página web de Eclipse,” (Consultado en Febrero 2019). [Online]. Available: <https://www.eclipse.org>
- [19] “Página web de Sublime,” (Consultado en Febrero 2019). [Online]. Available: <https://www.sublimetext.com>
- [20] “Página web de MagicDraw,” (Consultado en Noviembre 2019). [Online]. Available: <https://www.nomagic.com/products/magicdraw>
- [21] “Página web de Balsamiq,” (Consultado en Noviembre 2019). [Online]. Available: <https://balsamiq.com/wireframes/>
- [22] “Página web de Draw.io,” (Consultado en Junio 2019). [Online]. Available: <https://www.draw.io>
- [23] “Página web de Darksky API,” (Consultado en Abril 2019). [Online]. Available: <https://darksky.net/dev/docs>