Memoria

My Trainer: Aplicación para el seguimiento y monitorización de partidos en interiores

Trabajo de Fin de Grado

Ingeniería Informática

Julio 2022

VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Autor

Paula Terleira Fernández

Tutores

André Filipe Sales Mendes

Juan Francisco de Paz Santana

Gabriel Villarrubia González

CERTIFICADO DE LOS TUTORES

D. André Filipe Sales Mendes, D. Juan Francisco de Paz Santana y D. Gabriel Villarrubia González, profesores del departamento de Informática y Automática de la Universidad de Salamanca.

HACEN CONSTAR:

Que el trabajo de fin de grado titulado “My Trainer: Aplicación para el seguimiento y monitorización de partidos en interiores” ha sido realizado por Paula Terleira Fernández, con DNI 70908664D, bajo su dirección.

SALES MENDES ANDRE FILIPE - Y5601546W

Firmado digitalmente por SALES MENDES ANDRE FILIPE - Y5601546W
Fecha: 2022.07.03 07:22:24 +02'00'

D. André Filipe Sales Mendes

DE PAZ SANTANA JUAN FRANCISCO - 76125754D

Firmado digitalmente por DE PAZ SANTANA JUAN FRANCISCO - 76125754D
Nombre de reconocimiento (DN): c=ES, serialNumber=IDCES-76125754D, givenName=JUAN FRANCISCO, sn=DE PAZ SANTANA, cn=DE PAZ SANTANA JUAN FRANCISCO - 76125754D
Fecha: 2022.07.04 10:19:49 +02'00'

D. Juan Francisco de Paz Santana

VILLARRUBIA GONZALEZ GABRIEL - 71932845T

Firmado digitalmente por VILLARRUBIA GONZALEZ GABRIEL - 71932845T
Fecha: 2022.07.04 10:17:54 +02'00'

D. Gabriel Villarrubia González

RESUMEN

Este documento recoge la memoria del proyecto de fin de grado “My Trainer: Aplicación para el seguimiento y monitorización de partidos en interiores”, este proyecto consiste en la creación de una página web, para facilitar el control de equipos deportivos y sus actividades.

Actualmente, en el mundo del deporte no sirve con ir a una competición y participar, el deporte se encuentra en un momento de auge del análisis de datos o scouting deportivo, análisis de los datos del juego para optimizar el planteamiento de las actividades, es una parte muy importante en la preparación de los entrenamientos, partidos, competiciones…

La aplicación creada busca facilitar el análisis de los datos de los jugadores de un equipo, de esta forma el cuerpo técnico puede consultar los puntos fuertes y débiles de cada jugador, permitiéndoles adaptar los entrenamientos a las necesidades de su equipo, logrando de esta forma explotar sus puntos fuertes y mejorar sus puntos débiles.

La página web permitirá al usuario dar de alta su propio equipo, añadiendo a los jugadores, las balizas del equipo y los distintos eventos programados.

Todo esto se realizará con la ayuda de las balizas localizadoras, estas les mostrarán las posiciones en tiempo real de los jugadores, permitiendo al cuerpo técnico ver los distintos movimientos y poder rectificarles las mismas. Además, pueden comprobar las zonas más transitadas por cada uno de ellos en el mapa de calor de cada jugador.

También se les mostrarán distintas tablas y gráficas que les permitirá consultar los distintos datos y estadísticas, de forma visual. Estas estadísticas podrán consultarlas según sus necesidades: las estadísticas de cada jugador en los distintos eventos, las estadísticas de un evento con la comparación de todos los jugadores o las estadísticas generales del equipo con los datos de todos los integrantes del equipo.

Para la realización del proyecto se han empleado numerosas herramientas y bibliotecas, las principales son:

- Para el desarrollo web se ha empleado el framework de JavaScript, vue js.
- Para el tratamiento de datos se han empleado dos bases de datos, MongoDB: encargada de los datos de los distintos usuarios, equipos, jugadores… y InfluxDB, encargada de recopilar y tratar los datos obtenidos de las balizas localizadoras.

En relación con la documentación del proyecto, esta estará dividida en el documento actual, la memoria, y seis anexos, en estos anexos se explicarán las técnicas empleadas para lograr la conclusión del proyecto, empleando distintas técnicas como las propias del proceso unificado. Además, se dotará de una documentación técnica para comprender el código realizado y un manual para el correcto uso de la aplicación por parte del usuario.

Palabras clave: análisis de datos, deporte, rendimiento, monitorización partidos, estadísticas deporte, scouting.

SUMMARY

This document contains the report of the final degree project "My Trainer: Application for the tracking and monitoring of indoor matches", this project consists of the creation of a website to facilitate the control of sports teams and their activities.

Currently, in the world of sport it is not enough to go to a competition and participate, sport is at a time of boom in data analysis or sports scouting, analysis of game data to optimize the approach of activities, is a very important part in the preparation of training, matches, competitions...

The application created seeks to facilitate the analysis of the data of the players of a team, in this way the coaching staff can consult the strengths and weaknesses of each player, allowing them to adapt the training sessions to the needs of their team, thus managing to exploit their strengths and improve their weaknesses.

The website will allow the user to register their own team, adding players, team markers and scheduled events.

All this will be done with the help of the locator beacons, which will show the players' positions in real time, allowing the coaching staff to see the different movements and to be able to rectify them. They can also check the most frequented areas for each of them on the heat map of each player.

They will also be shown different tables and graphs that will allow them to consult the different data and statistics, in a visual way. These statistics can be consulted according to their needs: the statistics of each player in the different events, the statistics of an event with the comparison of all the players or the general statistics of the team with the data of all the members of the team.

Numerous tools and libraries have been used for the realization of the project, the main ones being:

- For web development, the JavaScript framework, vue js, was used.
- For data processing, two databases have been used, MongoDB: in charge of the data of the different users, teams, players... and InfluxDB, in charge of collecting and processing the data obtained from the locator beacons.

In relation to the documentation of the project, this will be divided into the current document, the report, and six annexes, in these annexes will be explained the techniques used to achieve the conclusion of the project, using different techniques such as those of the unified process. In addition, technical documentation will be provided to understand the code and a manual for the correct use of the application by the user.

Keywords: data analysis, sport, performance, match monitoring, sport statistics, scouting.

Índice

Índice de tablas

Índice de figuras

M.1. INTRODUCCIÓN.

En la actualidad, las nuevas tecnologías se encuentran en todos los ámbitos de la vida, el deporte, por supuesto, es uno más de estos, desde el nivel amateur al profesional, se han ido desarrollando nuevas tecnologías, abriendo un mundo nuevo de posibilidades.

El scouting deportivo, consiste en el análisis de los datos del juego para recopilar datos y obtener feedback que permita mejorar el planteamiento de las distintas actividades a realizar, y mejorar así el rendimiento de los jugadores.

La aplicación creada busca facilitar el análisis de los datos de los jugadores de un equipo, de esta forma el cuerpo técnico puede consultar los puntos fuertes y débiles de cada jugador, permitiéndoles adaptar los entrenamientos a las necesidades de su equipo, logrando de esta forma explotar sus puntos fuertes y mejorar sus puntos débiles.

Se mostrará al usuario la información relacionada con su equipo de la forma más visual y sencilla para poder consultar los datos deseados de la mejor forma posible, mediante el uso de tablas, gráficas y un mapa de visualización de posiciones en tiempo real y un mapa de calor, en el que se podrán consultar las zonas más frecuentadas por cada jugador.

Estos datos serán obtenidos mediante un sistema de localización, que facilitará las posiciones de los jugadores en tiempo real, permitiendo que esta información posteriormente sea tratada para permitir mostrar los datos de una forma visual como se ha comentado anteriormente.

Para la realización del proyecto se llevará a cabo la realización de una página web con la que interactuará el usuario, para la realización de esta se empleará el framework de JavaScript VueJS.

Además se dispondrá de una API, encargada del tratamiento de datos y conexión con las distintas librerías y bases de datos:

- MongoDb: encargada del tratamiento de datos de los usuarios, equipos, jugadores…
- InfluxDB: encargada de la recolección y tratamiento de los datos obtenidos a través de los sistemas de recolección de datos.

A lo largo de este documento se explicarán los aspectos más relevantes del desarrollo del proyecto, estos son:

- Comparativa con otros softwares similares
- Objetivos del proyecto
- Conceptos teóricos necesarios para la correcta comprensión
- Técnicas y herramientas empleadas
- Aspectos relevantes del proyecto
- Conclusiones del mismo
- Posibles líneas futuras.

Esta memoria se ve complementada con los seis anexos, donde se puede consultar la información explicada con mayor detalle.

Los anexos son:

Anexo I. Plan de proyecto
Anexo II. Especificación de requisitos.
Anexo III. Análisis de requisitos.
Anexo IV. Diseño del sistema software
Anexo V. Documentación Técnica
Anexo VI. Manual de usuario.

M.2. ESTADO DEL ARTE

A continuación, se mostrarán algunos softwares de tratamiento de datos en el deporte que existen en el mercado.

M.2.1. FutApp

FutApp [1], es un software que facilita el tratamiento de los datos de jugador y de los eventos del equipo mediante el uso de un calendario.

Esta aplicación está dirigida a los equipos modestos, únicamente trata los datos introducidos por el usuario, no cuenta con ningún sistema de localización ni recolección de datos, su objetivo es facilitar la gestión de un equipo, permitiendo mantener un control de forma digital en lugar de mediante el uso de papel.

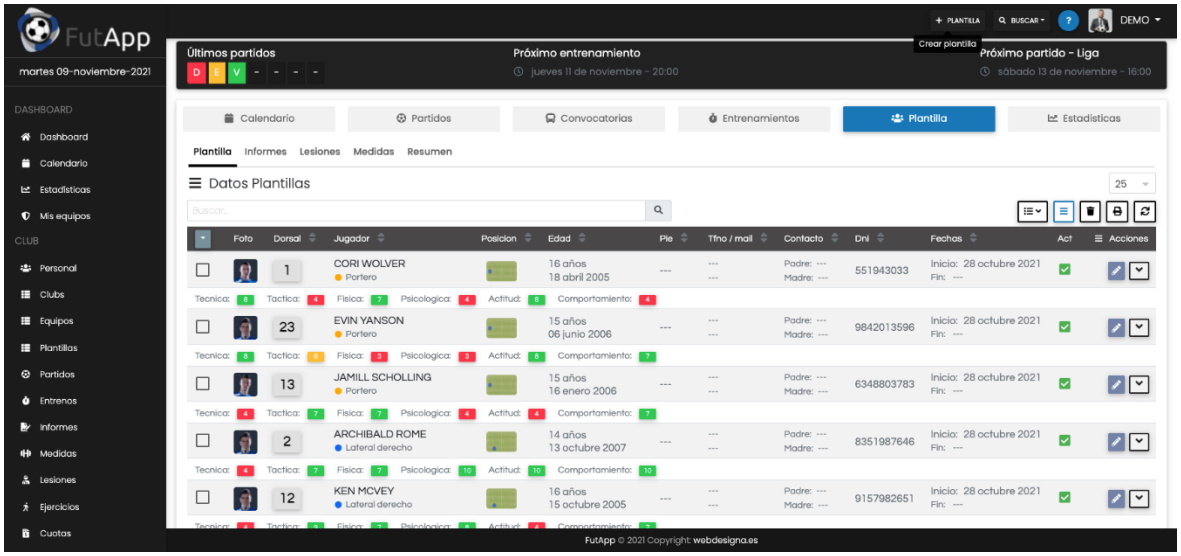

Figura 1: FutApp

M.2.2. Catapult Sports

Catapult sports [2], es un software con mayor sofisticación que el anterior. Permite al usuario controlar los datos recogidos durante los partidos o entrenamientos, mediante el uso de distintas tecnologías de localización según el deporte, algunos de los sistemas de localización empleados son:

- LPS (Sistema Posicionamiento Local) empleado tanto en interiores como exteriores.
- GNSS y LPS, permitiendo el rastreo de alta precisión.

Muestra al usuario los datos recopilados mediante los sistemas mencionados, empleando diferentes gráficas, tablas y un mapa de calor con las zonas transitadas por los jugadores.

Además permite mantener el control de jugadores y eventos mencionados en la aplicación anterior.

Figura 2: Catapult Sports

M.3. OBJETIVOS.

En este apartado se van a explicar los distintos objetivos que debe cumplir el proyecto, tanto los objetivos del sistema como los objetivos a nivel personal del desarrollador.

M.3.1. Objetivos del sistema

Los objetivos principales de la aplicación son los siguientes:

- **Gestión usuarios:** el sistema debe ser capaz de gestionar la autenticación de los usuarios, es decir, permitir el registro de nuevos usuarios y en caso de disponer de una cuenta registrada, permitirle el acceso a la aplicación. Además, deberá permitir al usuario modificar su contraseña cuando lo desee y en caso de haberla olvidado.
- **Gestión de equipos:** el sistema debe permitir la creación y gestión de los distintos equipos creados por los usuarios. Permitiendo al usuario modificar los datos del equipo.
- **Gestión de jugadores:** el sistema debe permitir al usuario dar de alta nuevos jugadores con sus datos personales, además permitirá modificar a los distintos jugadores asociados a un equipo, así como consultar los datos de cada uno de ellos.

Por último se permitirá al usuario controlar la disposición de sus jugadores en tiempo real mediante el uso de un mapa, así como las zonas por las que se ha desplazado el mismo empleando un mapa de calor.

- **Gestión de balizas:** el sistema debe permitir al usuario dar de alta en su equipo las balizas localizadoras del mismo, modificar los datos de estas mismas balizas, así como la asociación con los jugadores del mismo equipo. También se permitirá desasociar los jugadores de las balizas, al igual que se permitirá eliminar las balizas deseadas del equipo.
- **Gestión de eventos:** el sistema debe permitir al usuario la visualización de los distintos eventos asociados a su equipo. Además, se le permitirá dar de alta nuevos eventos, modificar los mismos y eliminarlos.
- **Gestión de estadísticas:** el sistema permitirá al usuario consultar las estadísticas de su equipo, estas se podrán consultar en varios formatos, además se podrán consultar las de todo el equipo o las estadísticas de cada jugador individualmente.

Además de los objetivos funcionales del usuario, el sistema deberá cumplir ciertos objetivos no funcionales que permitan el correcto funcionamiento del sistema, algunos de ellos serán:

- **Seguridad**: el sistema deberá garantizar la seguridad de los datos almacenados por el usuario, tanto para la autenticación como los relativos a su perfil y equipo.
- **Funcionamiento en tiempo real**: como se ha comentado se permitirá al usuario el seguimiento de sus jugadores en tiempo real, por lo que se tiene que dar el soporte necesario para el correcto funcionamiento.

M.3.2. Objetivos personales

Uno de los objetivos más importantes del proyecto es poner en práctica lo aprendido durante la realización del Grado en Ingeniería Informática, creando un proyecto completo en el que se han empleado conocimientos adquiridos en numerosas asignaturas, teniendo que utilizarlos de forma transversal, según las necesidades en cada momento.

Con relación al tema elegido, es un tema que me motivaba especialmente ya que siempre me ha gustado mucho el deporte y lo he practicado, por lo que unir dos de mis pasiones era un plus añadido, a la motivación que ya me generaba el poder crear una aplicación que mostrará todos esos datos que tantas veces había visto en televisión y me preguntaba cómo se recopilarían y quién podría dedicarse a ello.

Además, el uso de un sistema de localización en interiores, el cual nunca había empleado me generaba especial ilusión y ganas de trabajar con él, incluyendo todas las herramientas y librerías nuevas que conocería y tendría que poner en práctica para lograr mi objetivo.

M.4. CONCEPTOS TEÓRICOS.

En este apartado se explicarán distintos conceptos que deben ser detallados para comprender totalmente el proyecto.

M.4.1. RTLS

Real-Time Location Systems (RTLS) [3] o Sistema de ubicación en tiempo real, este sistema es empleado principalmente en interiores, estos sistemas proporcionan información en tiempo real sobre la ubicación de objetos, animales, personas… Esta ubicación permitirá mostrar a los usuarios las posiciones sobre un mapa.

La localización de los usuarios, se realizará mediante una infraestructura fija, se establecerán una serie de anclajes fijos en posiciones conocidas, estos serán los "Anchor", que se dispondrán en las esquinas del terreno de juego.

Los usuarios se localizarán mediante el uso de "Tags", estos se localizarán empleando las posiciones de los comentados anteriormente anchors, y esas posiciones serán a su vez recogidas por el listener.

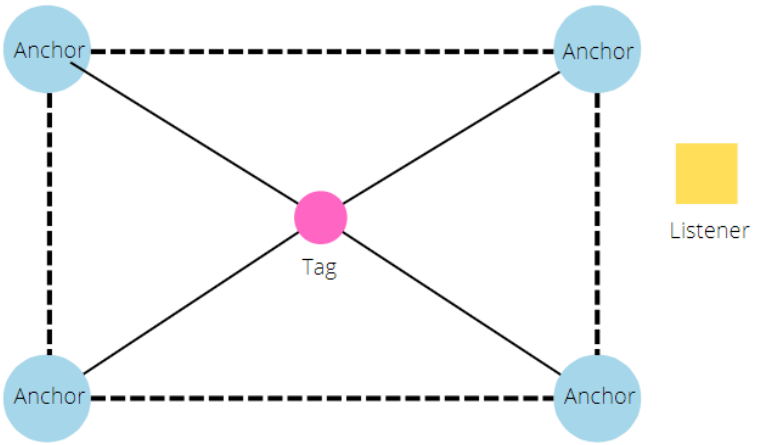

Figura 3. Esquema RTLS

M.4.2. Protocolo MQTT

MQTT [5] es un protocolo de mensajería estándar para Internet de las cosas (IoT). Es un transporte de mensajería extremadamente ligero ideal para conectar dispositivos remotos con un espacio de código pequeño y un ancho de banda de red mínimo.

Es un protocolo de comunicación m2m (Machine to machine) de tipo queue, está basado en la pila TCP manteniendo abierta la comunicación, permitiendo que esta sea reutilizada.

El funcionamiento se basa en un servicio de mensajería push con un patrón publicador-suscriptor, empleando un servidor central (Broker).

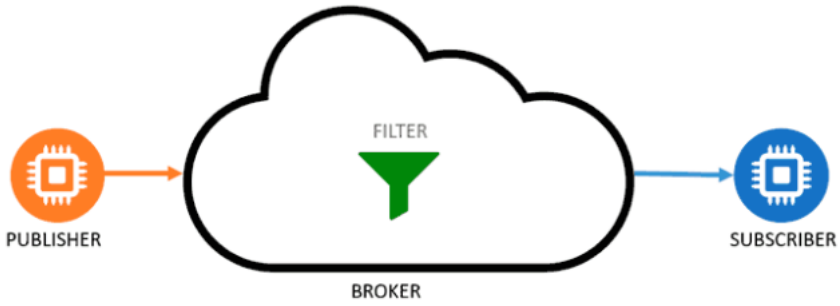

Figura 4: Esquema MQTT

El cliente puede enviar un mensaje en un topic, los cuáles organizan los mensajes de forma jerárquica. El resto de clientes pueden suscribirse a dicho topic, y el bróker les redirigirá los mensajes pertenecientes a ese topic.

La comunicación seguirá los siguientes pasos:

1. Conexión: el cliente envía un mensaje "Connect" y el bróker responde con "Connack".

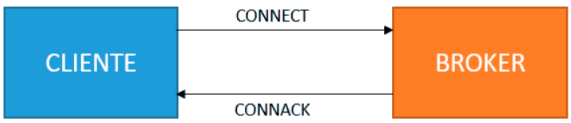

Figura 5: Esquema conexión MQTT

2. Envío de mensajes: el cliente emplea mensajes "Publish", el cual contiene el topic y payload. El bróker responde con "Suback"
3. Suscripción: Para realizar la suscripción o desuscripción, se emplearan "Subscribe" y "Unsubscribe", el bróker responderá con "Suback" y "Unsuback".

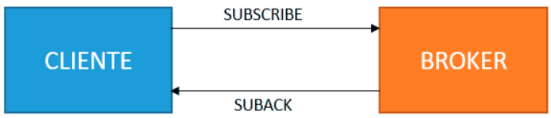

Figura 6: Esquema suscripción MQTT

MQTT dispone de un servicio de calidad (QoS) con tres niveles:

1. QoS 0 unacknowledged: el mensaje se envía una única vez. Conlleva la probabilidad de que alguno no se entregue.
2. QoS 1 acknowledged: el mensaje se envía hasta que se asegura la entrega. Conlleva la probabilidad de recibir mensajes duplicados.
3. QoS 2 assured: Se asegura que el mensaje se envía una única vez al suscriptor.

M.4.3. Protocolo WebSocket

WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y full-dúplex a través de un único socket TCP.

Está diseñado para implementarse en navegadores y servidores web.

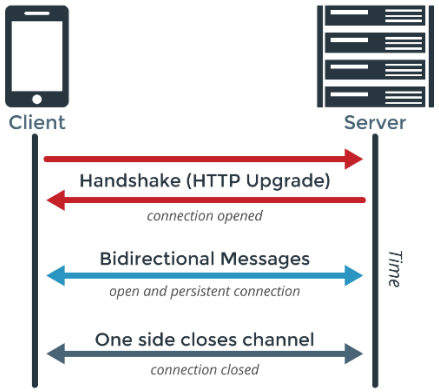

Figura 7: Esquema protocolo webSocket

M.4.4. Raspberry pi 3.

La Raspberry pi 3 B+ [7], es un microordenador de placa reducida, es la tercera generación de la Raspberry pi.

Algunas de sus características son:

- Procesador quad-core de 64 bits a 1,4 GHz.
- LAN inalámbrica de doble banda.
- Bluetooth 4.2/BLE
- Ethernet más rápido.
- Puerto HDMI y 4 USB 2.0

Figura 8: Raspberry Pi 3 B+

M.4.5. Módulo DWM1001C

El módulo DWM1001C [8] de Qorvo, anteriormente Decawave, combina el módulo DW1001, un MCU nRF52832 (sistema que soporta bluetooth y NFC) y un acelerómetro de 3 ejes.

Las características de este módulo son:

- Crea sistemas RTLS escalables, empleando el mismo diseño para las anclas, etiquetas y puertas de enlace.
- Bluetooth integrado.
- Sensor de movimiento a borde.
- Canal 5(6.5 GHz)/ tasa de datos 6,8 Mbps

Este módulo se puede utilizar para diversas aplicaciones, en nuestro caso el seguimiento de activos.

M.4.6. Red de localización

La red de localización de activos empleada estará formada por los siguientes elementos, algunos de ellos ya mencionados, esto permitirá comprender la estructura final de esta red.

- Anchors: nodos cuyas posiciones serán estáticas, en nuestro caso se ubicarán en las esquinas del terreno de juego. Estos nodos están formados por módulos DWM1001, los cuales se comunicaran con los tags, explicados posteriormente, para calcular sus posiciones.
- Tags: nodos dinámicos, formados por módulos DWM1001, cuyas posiciones serán calculadas a partir de las conexiones con los anchors.

Figura 9: Estructura Tags y Anchors DWM1001C

- Gateways: son las puertas de enlace, encargadas de recibir los datos de los dispositivos anteriores, para posteriormente comunicarse con la api de la aplicación y comunicarle los datos. Esta estará formada por uno de los elementos explicados anteriormente *Raspberry pi 3.* y el *Módulo DWM1001C.* Esta unión se muestra en la *Figura 10: Estructura gateway.*

Figura 10: Estructura gateway

M.5. TÉCNICAS Y HERRAMIENTAS

A continuación, se van a describir las diferentes técnicas, herramientas, bibliotecas y frameworks, empleadas en el desarrollo del proyecto.

Se van a dividir en tres secciones, las empleadas para la creación del frontend, para la API y para la documentación.

M.5.1. Frontend

Para la realización del frontend, es decir el sistema web con el cual interacciona el usuario se empleará el framework vue.js, y las siguientes herramientas:

M.5.1.1. Visual Studio Code

Visual Studio Code [9] es un editor de código fuente ejecutado en el escritorio, soporta Javascript, Typescript y nodeJs. Es un entorno de desarrollo integrado (IDE), desarrollado por Windows.

Se empleará en el proyecto debido a que incluye soporte para la depuración y finalización inteligente de código, y ya había sido utilizado previamente durante el grado.

M.5.1.2. HTML

HyperText Markup Language, lenguaje de marcas de hipertexto, para elaborar páginas web, define una estructura básica y un código para la definición de una página, como el texto, imágenes…

HTML [10] utiliza marcas para etiquetar texto, imágenes y otros contenidos, además incluyen elementos especiales como por ejemplo <head>, <title>, <body> entre otros.

M.5.1.3. SCSS

SCSS, emplea la sintaxis CSS pero con todas las ventajas SASS [11], como la declaración de variables o el anidamiento.

CSS [12], es el lenguaje de estilo empleado para el diseño web, permite darle forma, color y posición a una página web.
Los archivos .scss, son procesados por el servidor encargado de ejecutar la aplicación web, de tal forma que genera un fichero .css entendible por el navegador.

M.5.1.4. Javascript

JavaScript [13] es un lenguaje de programación basado en ECMAScript, definido como orientado a objetos. Se emplea en las páginas webs para mejorar la interfaz de usuario y las páginas web dinámicas.

Adopta nombres y convenciones del lenguaje de programación Java, sin embargo tiene diferentes semánticas y propósitos.

M.5.1.5. Vue JS

VueJS [14] es un marco de trabajo de JavaScript para la realización de interfaces de usuario, es un modelo de programación declarativo, y basado en componentes que facilita la realización de manera eficiente el desarrollo de interfaces de usuario. Emplea una arquitectura Modelo-Vista-VistaModelo.

Además se emplearán distintas librerías ofertadas por vue:

- Vue-router [15]: librería oficial de enrutamiento de vue-js, permitiendo redirigir a las distintas vistas.
- Vuex [16]: es un patrón de gestión de estado y biblioteca para aplicaciones vue.js. Permite mantener un almacenamiento centralizado para los componentes de una aplicación.

Vue es la principal herramienta empleada, por lo que para facilitar la comprensión del sistema se explicará brevemente algunos de los directorios más relevantes de la estructura de un proyecto vue.

- **NodeModules**: es un directorio que tienen todas las aplicaciones generadas con npm, contiene todas las dependencias instaladas mediante npm, estas se instalan a partir del fichero package.json empleando el comando *npm i*.
- **Src**: directorio principal que contiene todos los ficheros necesarios para el funcionamiento de la página web.

- **App.vue**: es el elemento principal sobre el que se sostiene el resto de la página.
- **Main.js**: es el encargado de crear la instancia de la aplicación y cargar los plugins y librerías que se van a emplear.
- **Router**: directorio encargado de almacenar los ficheros necesarios para gestionar y traducir las distintas vistas y componentes que se van a cargar durante el uso de la página.
- **Assets**: directorio en el que se encuentran las imágenes y elementos multimedia.

En este proyecto además se encontrará el directorio Store, encargado de gestionar el ya mencionado almacenamiento centralizado Vuex.

M.5.1.6. Vue-Material

Vue-Material [17], es un framework simple, construido con las especificaciones de Google material design, permite crear aplicaciones bien diseñadas, las cuales se ajustan a todos los navegadores web modernos, con una API fácil de usar.

M.5.1.7. Vuetify

Vuetify [18], es un framework completo de interfaz de usuario, construido sobre VueJS, su objetivo es facilitar al desarrollador las herramientas necesarias para crear aplicaciones atractivas para el usuario. Vuetify está desarrollado para ser fácil de aprender y con muchos componentes elaborados a partir de la especificación Material Design.

M.5.1.8. Axios

Axios [19] es un cliente HTTP, basado en promesas para node.js y el navegador. En el lado del servidor usa el modo nativo HTML, y en el lado del cliente usa XMLHttpRequests.

M.5.1.9. Heatmap.js

HeatmapJS [20] es una biblioteca de JavaScript que permite visualizar datos tridimensionales. Es una de las bibliotecas de visualización de mapas de calor más avanzada.

M.5.1.10. Three.js

Three.js [21] es una biblioteca de JavaScript que permite crear y mostrar gráficos animados en un navegador web, se emplea junto con el elemento canvas de HTML.

M.5.1.11. Plotly

Plotly [22] es una biblioteca de código libre de gráficos declarativa de alto nivel, que permite realizar gráficas interactivas. Cuenta con más de 40 tipos de gráficos, incluidos gráficos en 3D.

M.5.2. API

Para la realización de la API, se emplearán algunas de las herramientas mencionadas en el frontend, como son: *Visual Studio Code*, *Javascript*, *Axios*

M.5.2.1. NodeJS

NodeJS [23], es un entorno en tiempo de ejecución multiplataforma basado en Javascript, orientado a eventos asíncronos.

Uno de los motivos por el que se ha decidido el uso de este entorno, es su velocidad, debido a que es un modelo de evaluación de un único hilo de ejecución,

M.5.2.2. Npm

NPM [24] o "Node Package Manager", es el administrador de paquetes predeterminado para NodeJs, entorno de ejecución de Javascript.

M.5.2.3. MongoDB

MongoDB [25] es un sistema de base de datos, de código abierto orientado a documentos. Almacena los datos en estructuras BSON, similar a JSON, con un esquema dinámico permitiendo la fácil integración de los datos en distintas aplicaciones.

A continuación se muestra en la *Figura 11: Interfaz gráfica MongoDB* la interfaz gráfica de mongoDB, y la forma de almacenamiento empleada en forma de documentos.

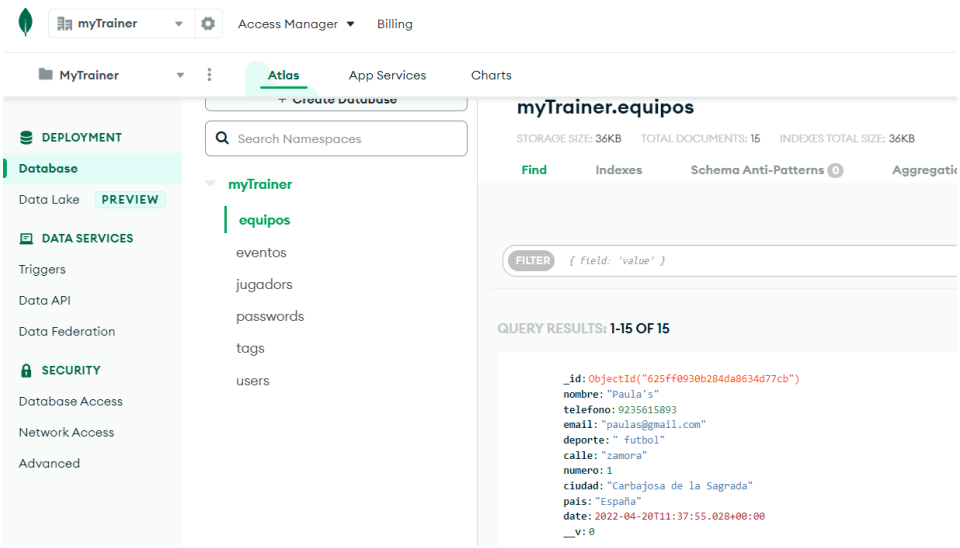

Figura 11: Interfaz gráfica MongoDB

M.5.2.4. Mongoose

Mongoose [26] es una librería para nodeJS, que permite escribir consultas a mongoDB con características como validaciones, construcciones de queries, middlewares, conversión de tipos y otras que permiten enriquecer la base de datos.

Crea una conexión entre

MongoDB y el marco de la aplicación web *Express.*

M.5.2.5. Express

Express [27] es un framework web back-end más popular para NodeJS, es un software gratuito de código abierto. Está diseñado para construir aplicaciones web y APIs.

Incluye miles de métodos de programa de utilidad HTTP y middleware, para la creación de una API, sólida, rápida y sencilla.

M.5.2.6. InfluxDB

InfluxDB [28] es una base de datos de series temporales (Time Series Database) de código abierto. Emplea el lenguaje de programación Go para el almacenamiento y recuperación de datos de datos de series de tiempo como por ejemplo en la monitorización de actividades.

La interfaz gráfica de InfluxDB, permite consultar los datos almacenados en la base de datos, así como realizar distintas queries, obteniendo los datos de la forma solicitada. En la *Figura 12: Interfaz gráfica InfluxDB*, se puede observar cómo se realiza una query, y se solicita la realización de distintas operaciones con los datos de la base de datos, almacenados en Buckets, a su vez se solicitan las columnas deseadas en la respuesta.

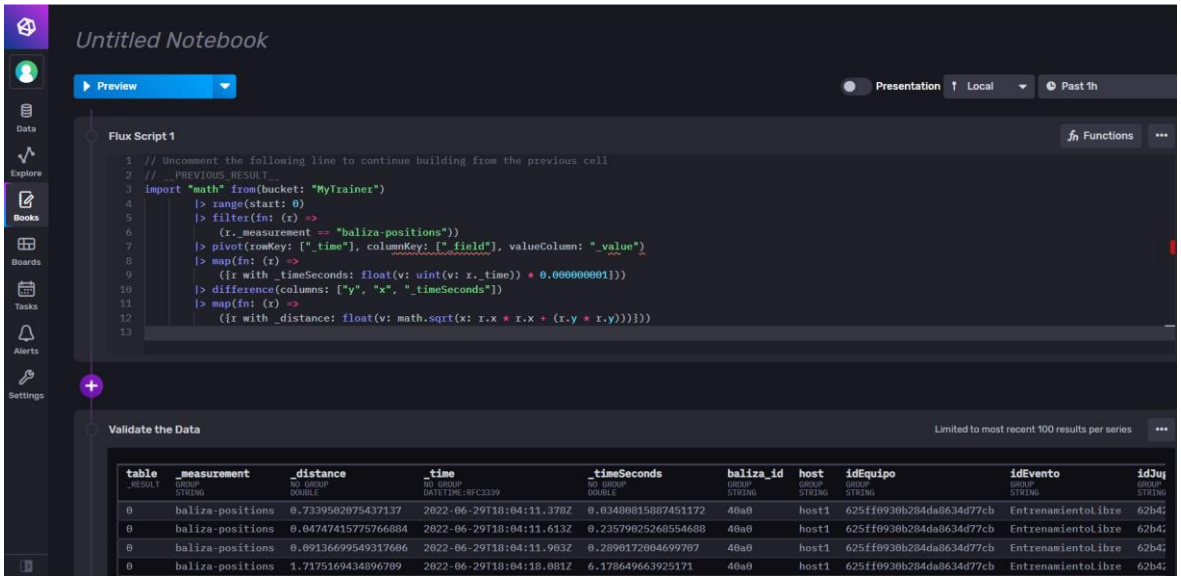

Figura 12: Interfaz gráfica InfluxDB

M.5.2.7. Nodemailer

Nodemailer [29] es un módulo para aplicaciones nodeJS, que permite enviar correos electrónicos de forma sencilla. Algunas de sus características son: tiene un único módulo sin dependencias, una gran seguridad, compatibilidad con Unicode y soporte de Windows.

M.5.2.8. SocketIo

SocketIO [30] es una biblioteca que permite la comunicación de baja latencia, bidireccional y basada en eventos entre un cliente y un servidor. Existen varias implementaciones de socketIO, la empleada en el proyecto es la implementación para JavaScript.

Permite la construcción de aplicaciones web en tiempo real, y es una biblioteca basada en el protocolo WebSocket, el cual proporciona un respaldo al sondeo largo HTTP o a la reconexión automática.

M.5.2.9. JWT

JSON web token (JWT) [31], es un token de seguridad encargado de definir una forma de transmisión de información segura mediante un objeto JSON. Este es creado al registrar un usuario. Este token se devuelve al cliente y este lo enviará en cada solicitud realizada al servidor para que se realice la comprobación en este.

M.5.3. Documentación

Por último se van a comentar las herramientas empleadas para la realización de los distintos anexos así como el tratamiento de las versiones del código durante la creación del mismo.

M.5.3.1. GitHub

GitHub [32] es una plataforma de desarrollo colaborativo, permite alojar proyectos empleando el sistema de control de versiones "Git".

Durante el proyecto se ha empleado para el control de versiones y como respaldo ante una posible pérdida de datos o fallo en el código actual.

M.5.3.2. Visual Paradigm

Visual Paradigm [33] es una herramienta UML CASE, que permite realizar el modelado UML y sus distintos diagramas (diagramas de clase, diagramas de secuencia…).

M.5.3.3. EZEstimate

Permite realizar la estimación del coste y esfuerzo de un proyecto software. Emplea la metodología de puntos de casos de uso para obtener la estimación.

M.5.3.4. JSDoc

```
/**
 * JSDoc
 */
```

JSDoc [34] es una sintaxis que permite agregar documentación al código fuente de JavaScript. Genera documentación de código de forma automática, generando una página HTML con la documentación.

M.6. ASPECTOS RELEVANTES

En este punto se describirán los aspectos y partes más importantes durante la realización del proyecto.

M.6.1. Marco de trabajo

Para el desarrollo del proyecto se ha empleado el proceso unificado, este es un marco de trabajo genérico que permite especializarse en diferentes sistemas software, de diferentes áreas, tamaños y niveles de aptitud. Sus características principales son:

- **Iterativo e incremental**: es un marco de trabajo iterativo e incremental ya que está compuesto de 4 fases: Inicio, elaboración, construcción y transición. Además, cada una de estas fases estará dividida en una o varias iteraciones. Al final de cada iteración se obtiene un producto incrementado, es decir que se ha mejorado con respecto a las funcionalidades de la iteración anterior.
- **Dirigido por casos de uso**: los casos de uso tienen una gran importancia ya que son los encargados de capturar los requisitos para definir el contenido de cada iteración.
- **Centrado en la arquitectura**: no existe un único modelo para todos los aspectos del sistema, por esto existen numerosos modelos y vistas que definen la arquitectura del software del proyecto.

Como se ha comentado previamente, el proceso unificado está formado por varias fases, a su vez formadas por distintas iteraciones. Las fases del proceso unificado son:

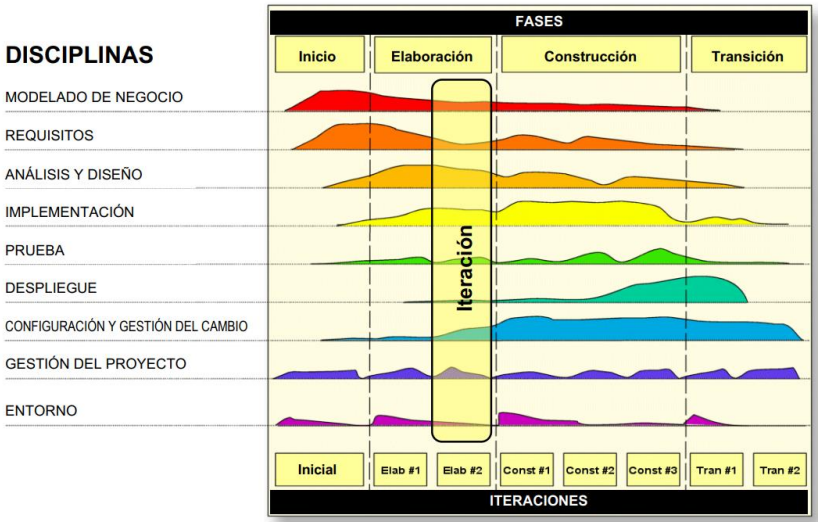

Figura 13: Diagrama de fases proceso unificado

- **Inicio:** Se define el alcance del proyecto y se desarrollan los distintos casos de negocio.
- **Elaboración:** Se obtiene una visión refinada del proyecto a realizar, especificando en detalle los casos de uso y el diseño de la arquitectura del sistema.
- **Construcción:** Comprende la evolución hasta llegar a ser un producto listo con unos requisitos mínimos, además se refinan los detalles menores.

- **Transición:** el producto se convierte en una versión beta, para ser probado. Se corrigen problemas y se incorporan mejoras sugeridas durante las pruebas.

A continuación, se muestran las fases e iteraciones llevadas a cabo para la realización de este proyecto.

Inicio	**8,31 días**	**lun 17/01/22**	**mié 26/01/22**
Iteración 1	**8,31 días**	**lun 17/01/22**	**mié 26/01/22**
Hito iteración 1	0 días	mié 26/01/22	mié 26/01/22
Hito Inicio	0 días	mié 26/01/22	mié 26/01/22
Elaboración	**68,31 días**	**mié 26/01/22**	**vie 22/04/22**
Iteración 1	**3,75 días**	**mié 26/01/22**	**lun 31/01/22**
Hito iteración 1	0 días	lun 31/01/22	lun 31/01/22
Iteración 2	**40,5 días**	**lun 31/01/22**	**lun 21/03/22**
Hito iteración 2	0 días	mar 22/03/22	mar 22/03/22
Iteración 3	**24 días**	**mar 22/03/22**	**vie 22/04/22**
Hito iteración 3	0 días	vie 22/04/22	vie 22/04/22
Hito elaboración	0 días	vie 22/04/22	vie 22/04/22
Construcción	**41,69 días**	**vie 22/04/22**	**mié 15/06/22**
Iteración 1	**18 días**	**vie 22/04/22**	**lun 16/05/22**
Hito iteración 1	0 días	lun 16/05/22	lun 16/05/22
Iteración 2	**17,13 días**	**lun 16/05/22**	**lun 06/06/22**
Hito iteración 2	0 días	lun 06/06/22	lun 06/06/22
Iteración 3	**6,5 días**	**lun 06/06/22**	**mié 15/06/22**
Hito iteración 3	0 días	mié 15/06/22	mié 15/06/22
Hito Construcción	0 días	mié 15/06/22	mié 15/06/22
Transición	**13 días**	**mié 15/06/22**	**vie 01/07/22**
Iteración 1	**13 días**	**mié 15/06/22**	**jue 30/06/22**
Hito iteración 1	0 días	vie 01/07/22	vie 01/07/22
Hito transición	0 días	vie 01/07/22	vie 01/07/22

Figura 14: Fases e iteraciones del proyecto

M.6.2. Estimación de coste y planificación temporal

Las primeras acciones llevadas a cabo son la de estimación del coste y planificación temporal, ya que permiten identificar las tareas a realizar y obtener una aproximación del tiempo necesario para la realización de cada una.

Para consultar la información más detallada sobre la *Estimación de coste* y la *Planificación temporal*, consultar el **Anexo I: Plan de proyecto.**

M.6.2.1. Estimación de coste

La estimación del coste y esfuerzo permite obtener una relación entre el personal y el tiempo empleado en la realización del proyecto, estimando la duración del proyecto.

Para ello se empleará el método de Punto de Casos de Uso (UCP) y el programa EZEstimate, con el que obtenemos los siguientes resultados.

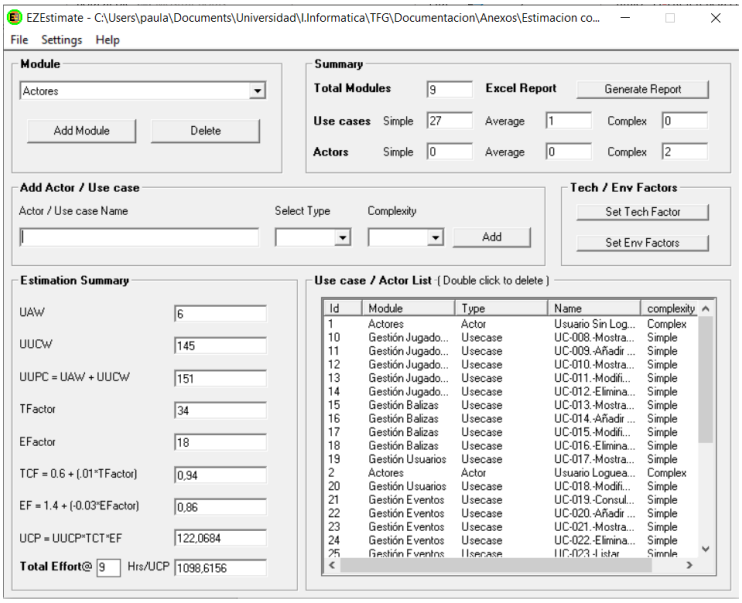

Figura 15: Resultados estimación coste EZEstimate

M.6.2.2. Planificación temporal

La realización de la planificación temporal nos permitirá analizar el proyecto dividiéndolo en partes más manejables e identificar las tareas realizadas y obtener una aproximación de la duración del proyecto.

Esta planificación se realiza al inicio del proyecto para permitir generar una idea de las tareas a realizar en cada momento para cumplir los plazos establecidos.

Para la realización de la planificación temporal se empleará el programa Microsoft Project, en el que se creará un calendario de trabajo acorde a las necesidades personales del desarrollador.

Como se ha mostrado previamente en la *Figura 14: Fases e iteraciones del proyecto*, obtendremos una tabla con las fechas de inicio y fin de cada fase e iteración del proyecto, además estas se representarán mediante un diagrama de Gantt, como en la *Figura 16: Diagrama de Gantt en* la que se pueden observar el inicio y fin de cada una de las tareas a realizar, así como las tareas críticas, aquellas que no se pueden retrasar sin retrasar la fecha de entrega.

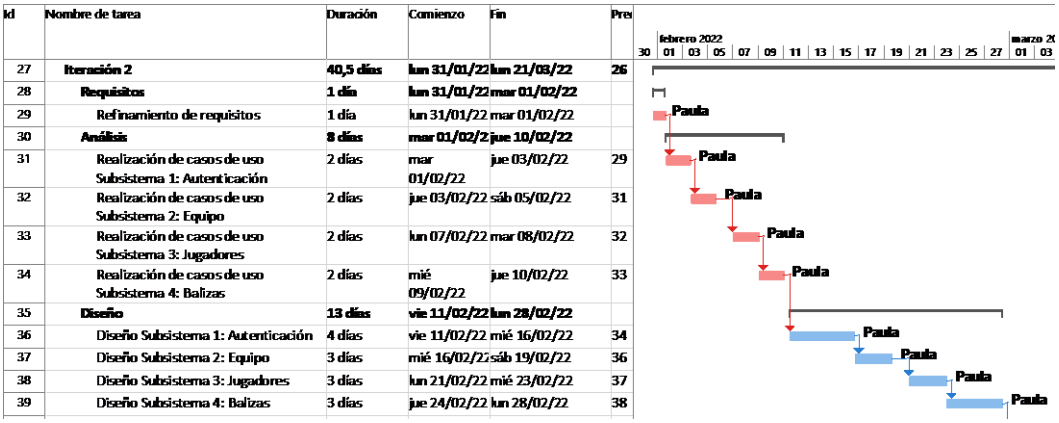

Figura 16: Diagrama de Gantt

Por último, al igual que en la *Estimación de coste*, obtendremos las estadísticas del proyecto.

Se puede observar como en ambas estimaciones, *Figura 15: Resultados estimación coste EZEstimate* y *Figura 17: Estadísticas planificación temporal*, se obtienen unas horas de trabajo bastante similares, cerca de las 1000 horas de trabajo, unos 130 días.

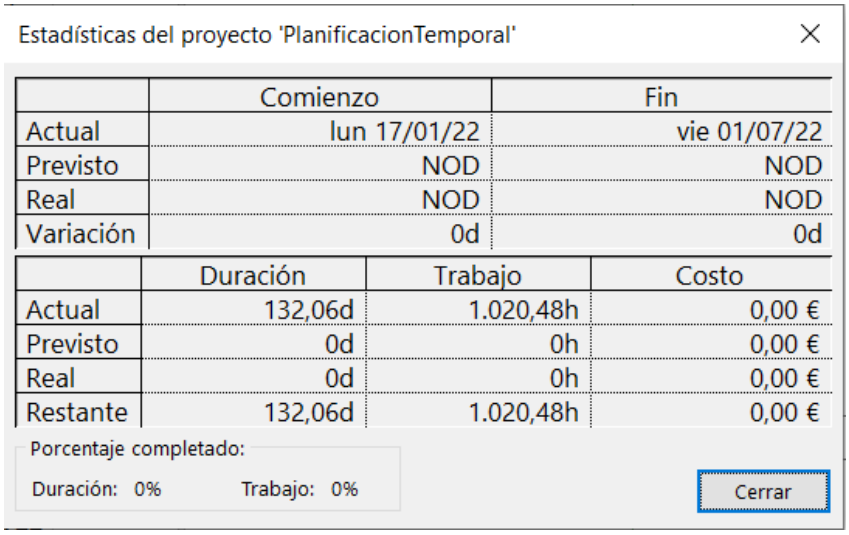

Figura 17: Estadísticas planificación temporal

M.6.3. Especificación de requisitos

A la hora de realizar la especificación de requisitos, se empleará la metodología de Durán y Bernárdez, y el programa Visual Paradigm para la realización de los diagramas.

Para consultar la información más detallada sobre la especificación de requisitos consultar el **Anexo II-Especificación de requisitos**.

A continuación se va a explicar brevemente las fases principales realizadas en la especificación de requisitos, mostrando un ejemplo de sus representaciones.

M.6.3.1. Objetivos del sistema

Se describen los principales objetivos buscados por el sistema para la correcta realización del proyecto.

Los objetivos del proyecto definidos son:

- Gestión de autenticación.
- Gestión de usuarios.
- Gestión de equipo.
- Gestión de jugadores.
- Gestión de balizas.
- Gestión de eventos.
- Visualización mapa tiempo real.
- Gestión de estadísticas.

Se representan empleando una tabla como la siguiente:

OBJ-001	**Gestión de autenticación**
Versión	1.0
Autor	Paula Terleira Fernández.
Descripción	El sistema debe permitir el registro de usuarios, así como el inicio de sesión en este si dispone de cuenta. También proporcionará al usuario la opción de modificar su contraseña en caso de olvido.
Subobjetivos	Ninguno
Importancia	Vital
Urgencia	Alta
Estado	Verificado
Estabilidad	Alta
Comentarios	-

Tabla 1: Ejemplo Objetivos del sistema

M.6.3.2. Requisitos de información

Con el fin de alcanzar los objetivos del sistema, se establecerán unos requisitos sobre la información que almacenará el sistema.

Los requisitos de información definidos para el sistema son:

- Información usuario.
- Información equipo.
- Información jugadores.
- Información balizas.
- Información eventos.
- Información estadísticas.

Se representan con una tabla como la siguiente:

IRQ-001	**Información usuario**
Versión	1.0
Autor	Paula Terleira Fernández.
Objetivos asociados	*OBJ-001.- Gestión de autenticación*
Descripción	El sistema deberá almacenar la información de los usuarios registrados en la aplicación. En concreto:
Datos Específicos	· Nombre. · Equipo. · Email. · Contraseña.
Importancia	Vital
Urgencia	Alta
Estado	Verificado
Estabilidad	Alta
Comentarios	-

Tabla 2: Ejemplo Requisitos de información

M.6.3.3. Requisitos no funcionales

Son aquellos requisitos relacionados con el funcionamiento del sistema.

Los requisitos no funcionales definidos para el sistema son:

- Multiplataforma.
- Usabilidad.
- Concurrencia.
- Funcionamiento en tiempo real.
- Seguridad de datos.

M.6.3.4. Requisitos funcionales

Son los encargados de definir las funciones del sistema.

M.6.3.4.1. Diagrama de paquetes.

Lo primero que se realiza en esta sección es un diagrama de paquetes para obtener una visión general del sistema.

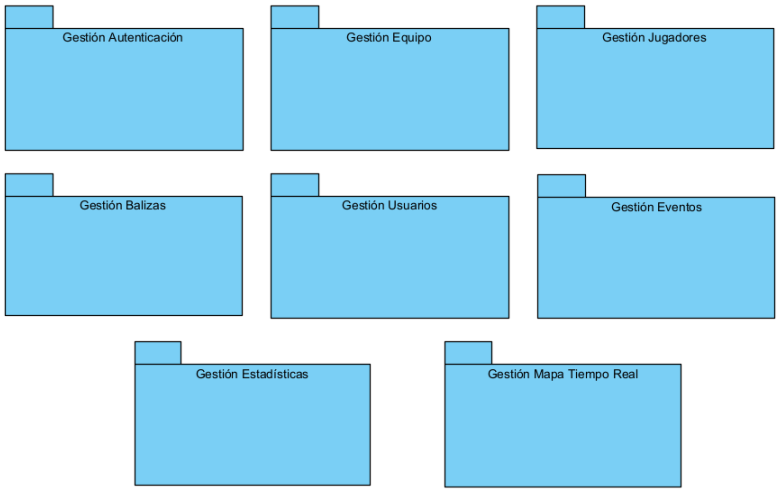

Figura 18: Diagrama de paquetes

M.6.3.4.2. Definición de actores

Se realiza la definición de los actores involucrados en el sistema.

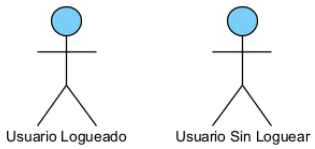

Figura 19: Diagrama de actores

M.6.3.4.3. Casos de uso

Los casos de uso son las descripciones de una acción o actividad llevada a cabo por un actor, se describirán mediante el uso de un diagrama y una tabla como las siguientes.

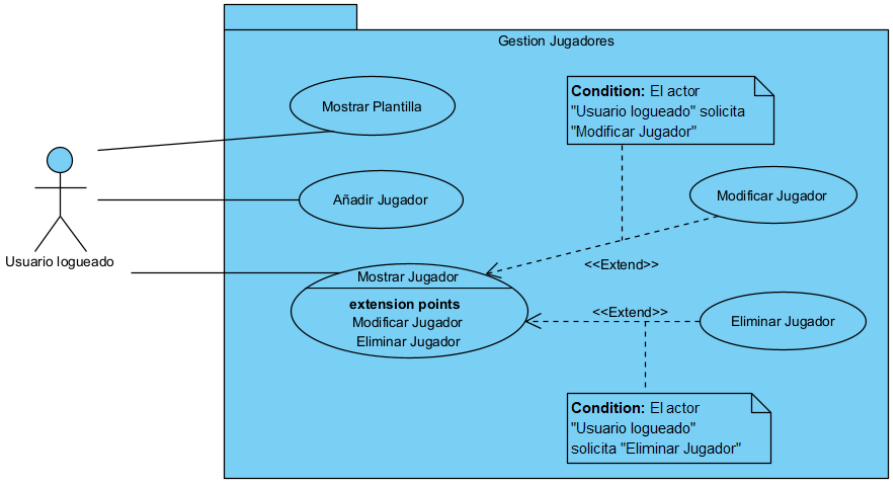

Figura 20: Ejemplo diagrama casos de uso

UC-008	**Mostrar Plantilla**	
Versión	1.0	
Autor	Paula Terleira Fernández.	
Dependencias	· OBJ-003.- Gestión de jugadores · IRQ-003.- Información jugadores	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el actor *UC-008.- Mostrar Plantilla* solicite la visualización de la plantilla.	
Precondiciones	Usuario logueado, el usuario debe haberse registrado previamente.	
Secuencia	**Paso**	**Acción**
	1	El actor *ACT-002.- Usuario logueado* solicita mostrar la plantilla.
	2	El sistema mostrará al usuario los datos de la plantilla almacenados.
Postcondición		
Excepciones	**Paso**	**Acción**
	-	-
Frecuencia esperada	Alta	
Importancia	Vital	
Urgencia	Alta	
Estado	Verificado	
Estabilidad	Alta	
Comentarios	-	

Tabla 3: Ejemplo casos de uso

M.6.4. Análisis de requisitos

En el análisis de requisitos se lleva a cabo un análisis, refinamiento y estructuración de los requisitos del sistema obtenidos en el **Anexo II: Especificación de requisitos**.

A continuación, se explicarán brevemente las distintas fases de dicho análisis, para consultar la información más detallada sobre el análisis de requisitos consultar el **Anexo III-Análisis de requisitos**.

M.6.4.1. Modelo de dominio

Representa las clases conceptuales del sistema, modelando las distintas clases del sistema.

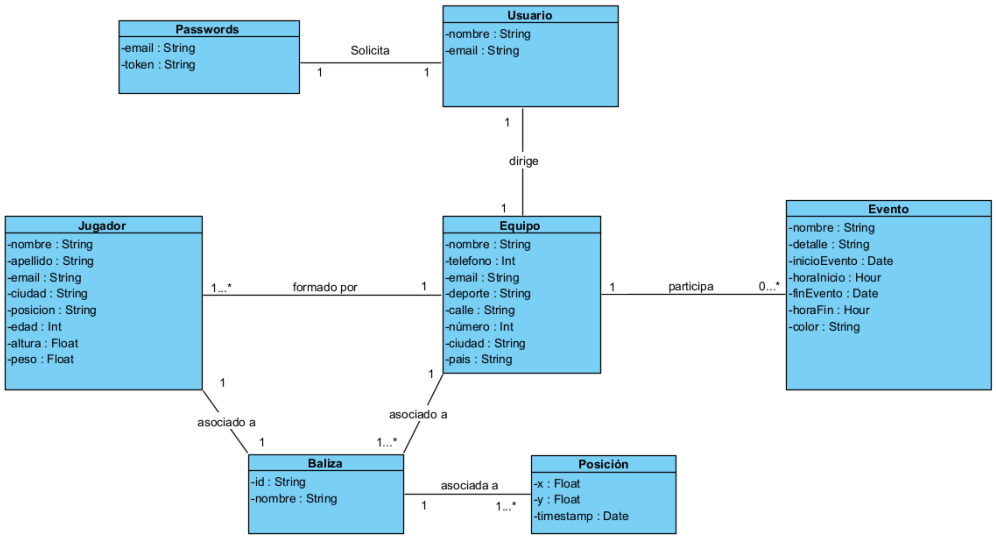

Figura 21: Modelo de dominio

M.6.4.2. Paquete de análisis

Representación del sistema mediante la descomposición en paquetes de análisis.

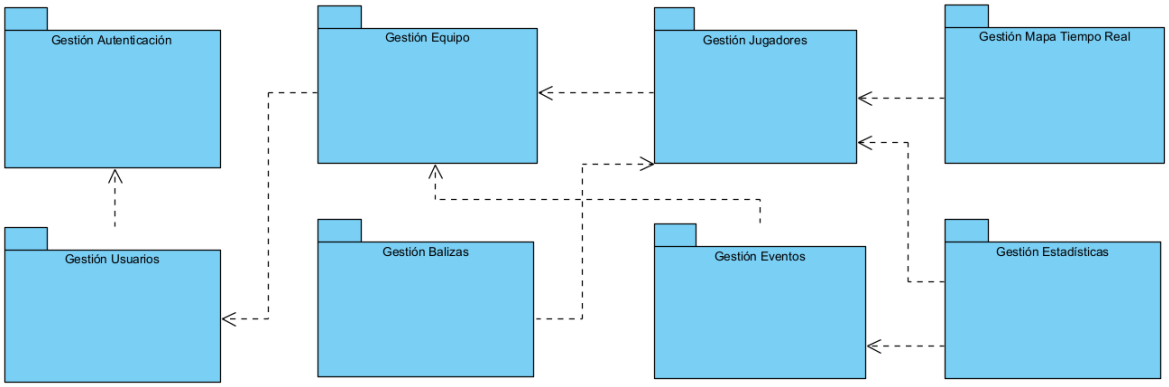

Figura 22: Diagrama paquete de análisis

M.6.4.2.1. Clases de análisis

Una vez definidos los distintos paquetes de análisis se representarán las distintas clases de análisis de tipo entidad, control e interfaz, agrupadas en los paquetes mostrados en *Figura 22: Diagrama paquete de análisis.*

Se muestra un ejemplo de los diagramas especificados en el **Anexo III: Análisis de requisitos.**

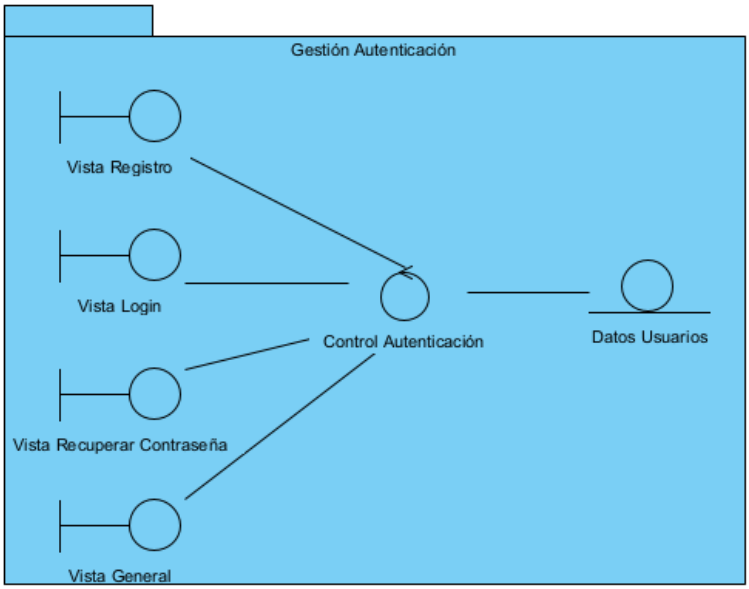

Figura 23: Ejemplo diagrama clases de análisis

M.6.4.3. Realización de casos de uso análisis

Durante la realización de los casos de uso de análisis se han explicado de forma general los pasos seguidos mediante el uso de diagramas de secuencia. A continuación se muestra un ejemplo de los diagramas empleados para los distintos casos de uso.

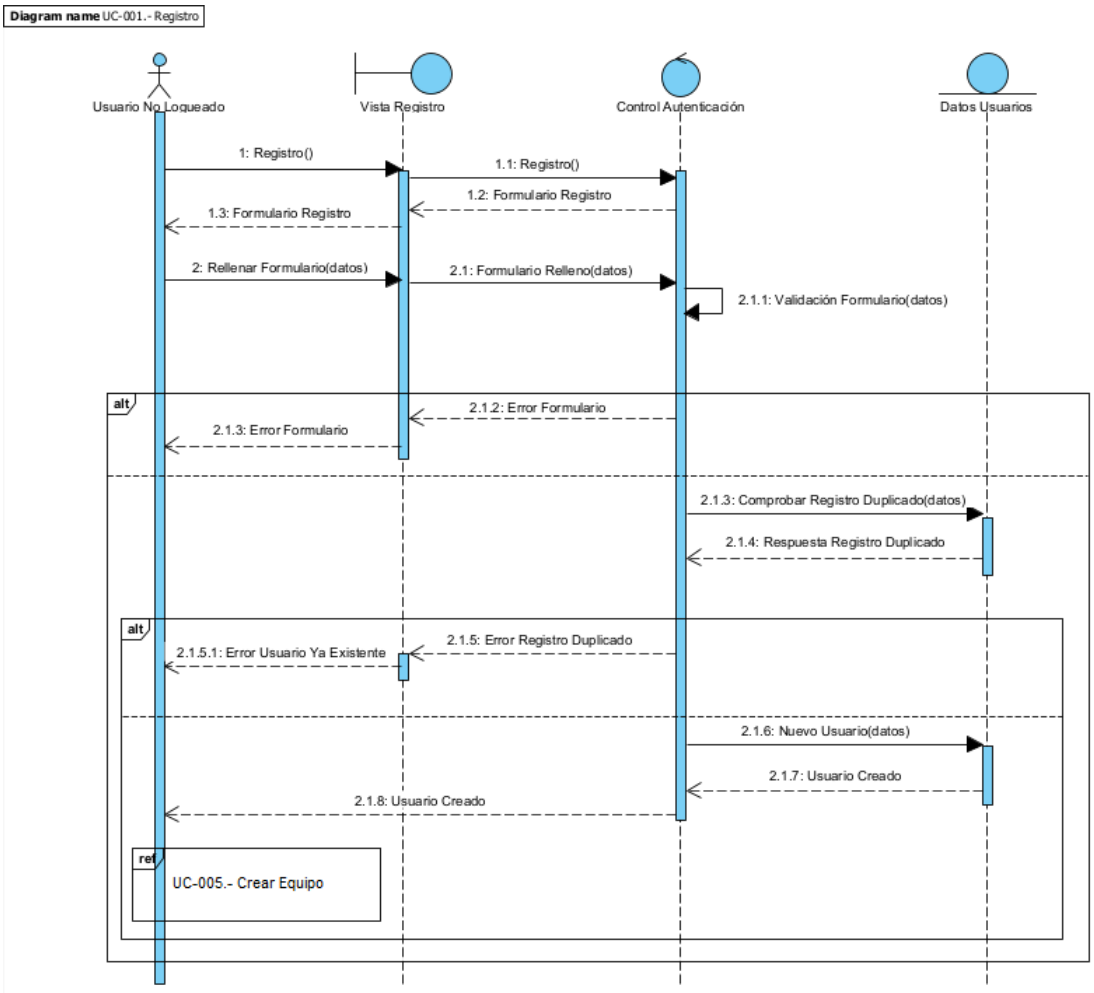

Figura 24: Ejemplo diagrama de secuencia casos de uso análisis

M.6.5. Diseño del sistema

El diseño del sistema está centrado en el modelo de soluciones, ya que nos permitirá contemplar las distintas soluciones y elegir la más adecuada.

Para consultar la información más detallada sobre el diseño del sistema consultar el **Anexo IV: Diseño del sistema Software**.

M.6.5.1. Patrones arquitectónicos

A continuación se mostrarán los patrones empleados en el proyecto:

M.6.5.1.1. Patrón MVVM

Se empleará el patrón Modelo-Vista-ModeloVista, facilita la separación del desarrollo de la interfaz gráfica del desarrollo del backend, permitiendo que la vista no dependa de ninguna plataforma. Además los cambios de la vista se muestran automáticamente en el modelo vista y al contrario.

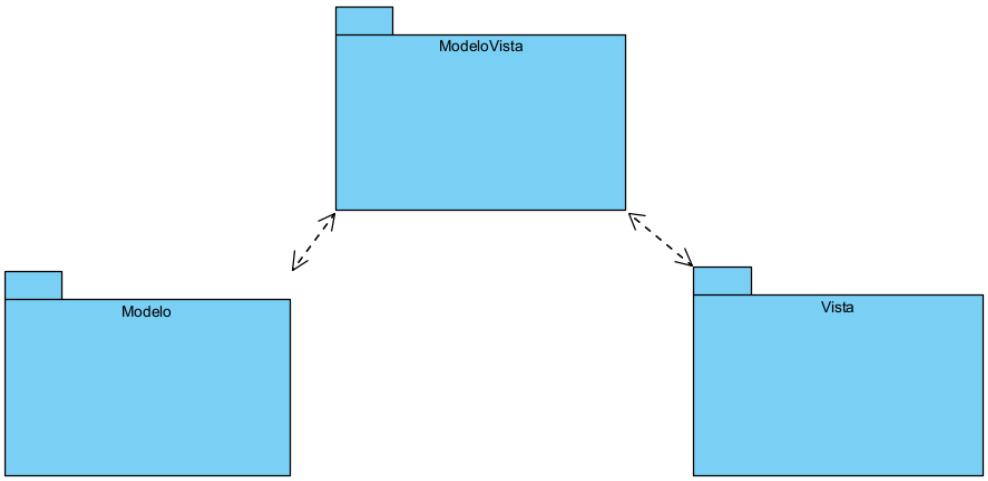

Figura 25: Diagrama patrón MVVM

M.6.5.1.2. Patrón State Management

Se empleará Vuex para VueJS, este es un patrón de gestión de estado, el cuál servirá como un almacén centralizado para los componentes de la aplicación. Permitiendo el acceso a datos a través de stores, permitiendo un acceso ordenado y más veloz a los datos.

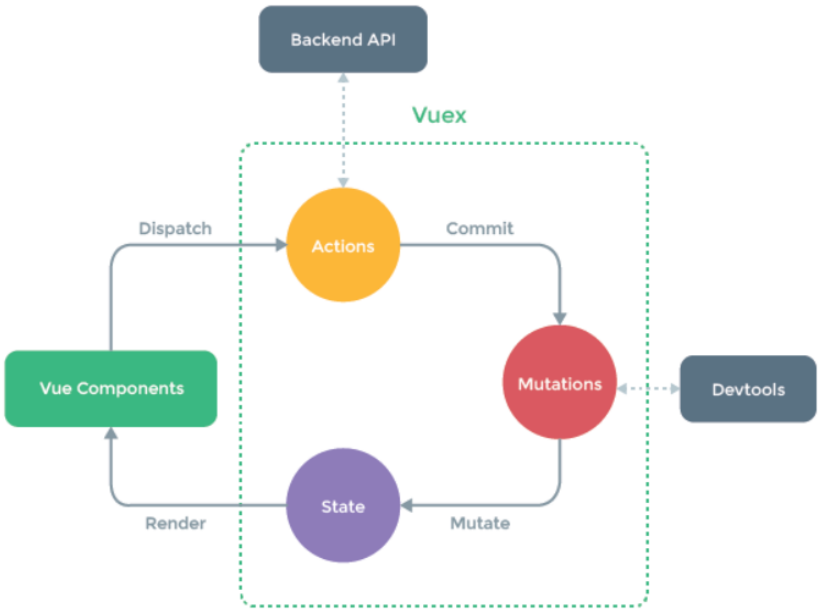

Figura 26: Diagrama patrón State Management, Vuex

M.6.5.1.3. Patrón Publisher-subscriber

Este patrón permite la comunicación entre los editores, remitentes del mensaje, y los suscriptores, receptores. A través del envío de mensajes a través de un canal, este es el intermediario entre ambos ya que ni el editor ni el receptor conocen la existencia del otro, ambos se comunican con el intermediario que filtra y transmite los mensajes.

Este patrón nos permite mejorar la escalabilidad de la red y hacerla más dinámica.

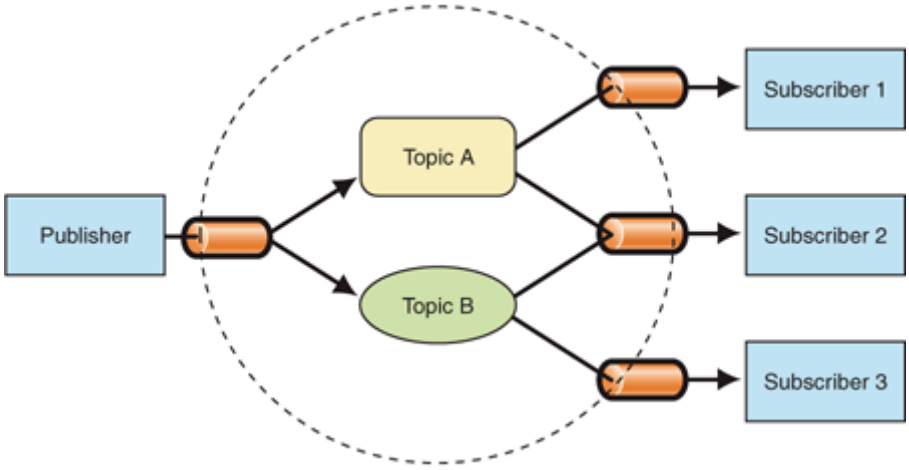

Figura 27: Diagrama patrón publisher-subscriber

En nuestro sistema se empleará este patrón en el tratamiento de la información en tiempo real obtenida de las balizas localizadoras, momento en el cuál se emplean distintas tecnologías como MQTT, o webSockets, siguiendo el protocolo de este patrón.

M.6.5.2. Subsistemas de diseño

Se realizará una descomposición de los subsistemas según su función, de forma general se dividirán en Frontend y Backend.

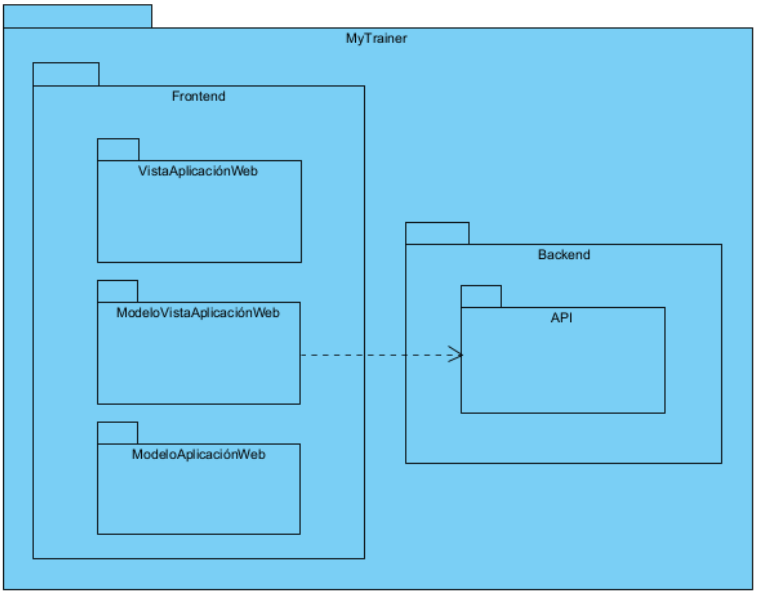

Figura 28: Diagrama subsistemas de diseño

M.6.5.3. Clases de diseño

En esta sección se describen los contenidos de la clase encargada del frontend y del backend. En el caso del frontend donde se aplica el patrón MVVM, se muestra el contenido de cada capa perteneciente al patrón, a continuación se mostrará la capa ModeloVista, el resto se pueden consultar en el **Anexo IV: Diseño del sistema software.**

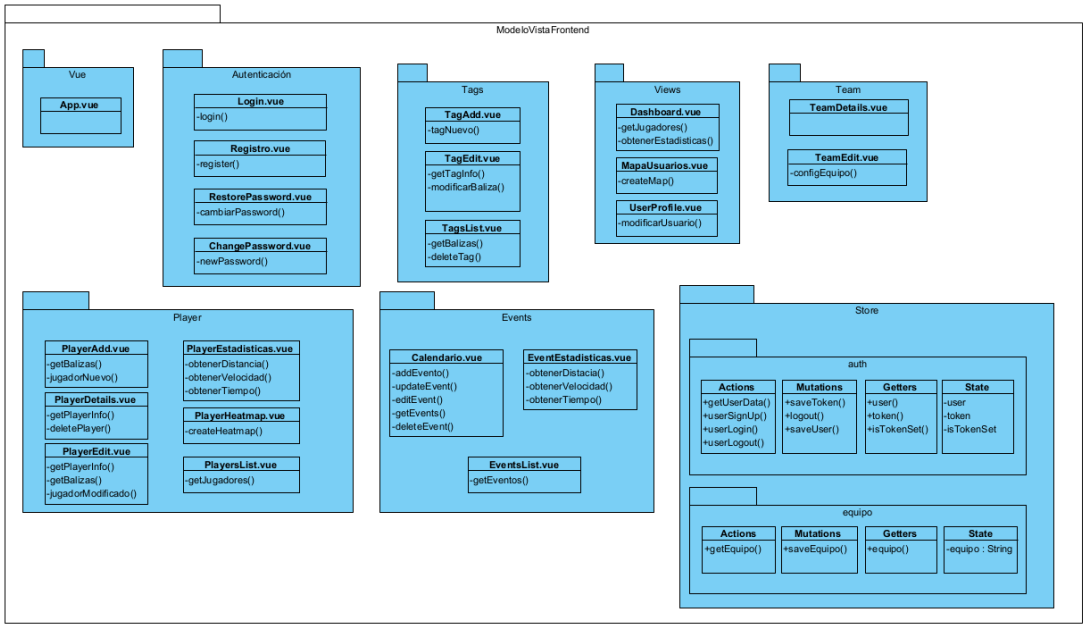

Figura 29: Diagrama clases de diseño ModeloVista

M.6.5.4. Realización de casos de uso

Para la realización de los casos de uso del sistema software, se parte de los realizados en el **Anexo III: Análisis de requisitos**, se refinarán de modo que sean la mayor aproximación a los métodos implementados en el proyecto.

A continuación se muestra uno de ellos como ejemplo:

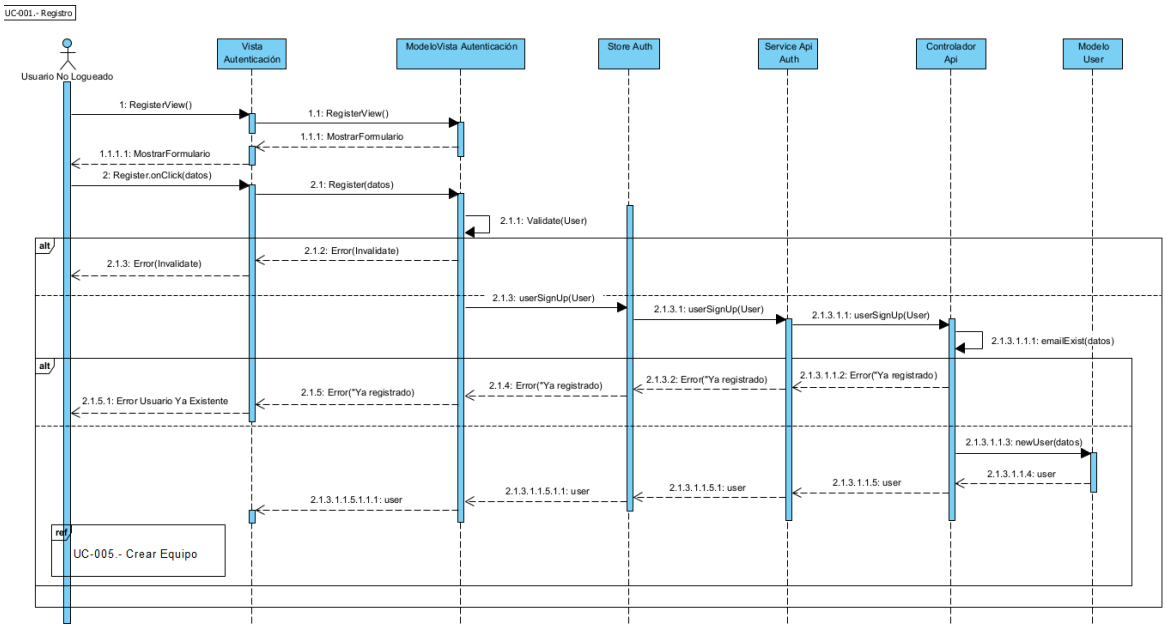

Figura 30: Realización casos de uso

M.6.5.5. Diseño de la base de datos

El almacenamiento de los datos necesarios para el sistema, se va a realizar a través de dos bases de datos distintas:

M.6.5.5.1. MongoDB

Se encarga del tratamiento de usuarios, jugadores, equipos… En la *Figura 31: Diseño de MongoDB* se muestra la estructura de dicha base de datos.

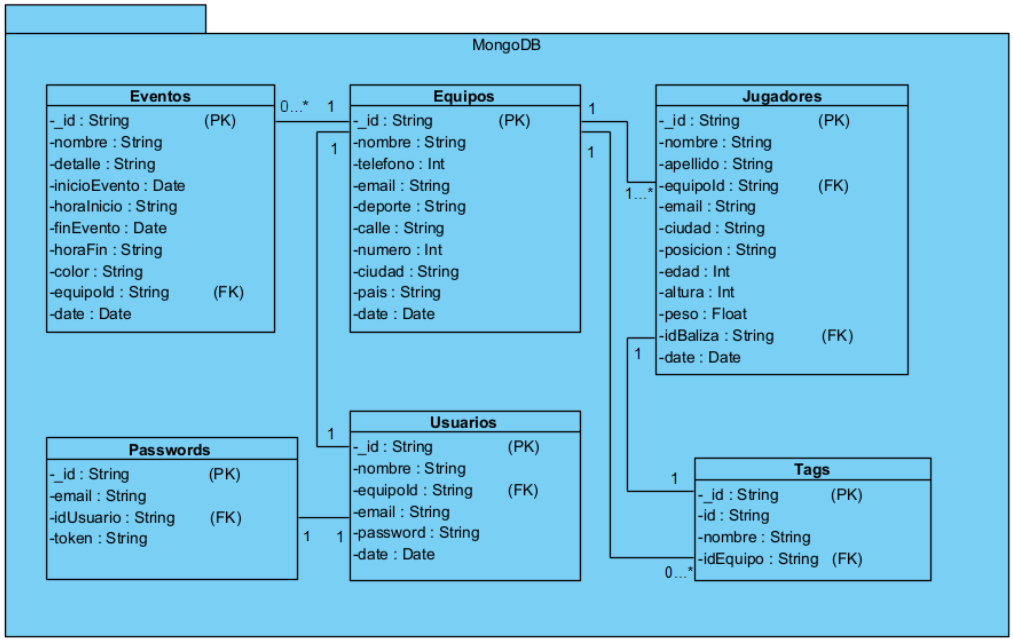

Figura 31: Diseño de MongoDB

M.6.5.5.2. InfluxDB

Esta base de datos se encargará de almacenar las posiciones de las balizas localizadoras asociadas a su id. En la *Figura 32: Diseño InfluxDB* se muestra la estructura de la base de datos InfluxDB.

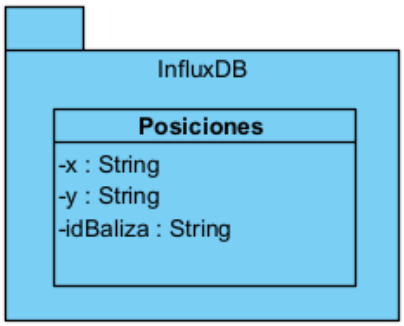

Figura 32: Diseño InfluxDB

M.6.5.6. Modelo de despliegue

El modelo de despliegue representa la arquitectura de un sistema en tiempo de ejecución. Los nodos que lo conforman son los siguientes:

- **ClienteWeb:** dispositivo que cuenta con un navegador web y conexión a la red que le permite acceder a la aplicación web y visualizar la información del sistema.
- **ServidorWeb:** proporciona la aplicación web al cliente.
- **Servidor Api:** encargado de la API del sistema. Procesa las peticiones realizadas por el cliente y se encarga de obtener los datos del sistema y modificarlos teniendo en cuenta las peticiones de los clientes.
- **Broker MQTT:** es el servidor MQTT, encargado de recibir y difundir los mensajes entre sus clientes.
- **Collector MQTT:** encargado de recopilar los datos de los sensores inalámbricos.
- **Servidor MongoDB:** base de datos empleada para almacenar la mayor parte de la información tratada por el sistema. Permite el tratamiento de los datos en tiempo real, permitiendo añadir, eliminar o modificar los datos deseados.
- **Servidor InfluxDB:** base de datos empleada para almacenar las posiciones de las balizas localizadoras, estos datos se almacenarán pero no se permite su modificación. Se emplea para almacenar un gran número de posiciones por segundo, y nos permite realizar operaciones de forma sencilla.

En la *Figura 33: Diagrama modelo de despliegue* se muestran los distintos nodos y sus relaciones.

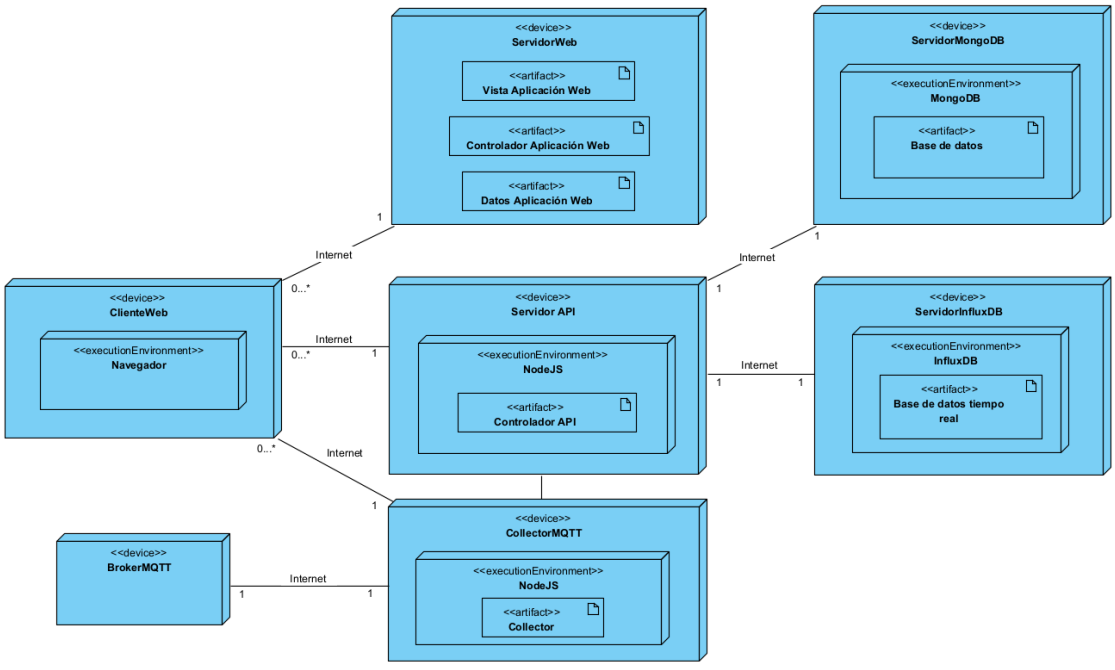

Figura 33: Diagrama modelo de despliegue

M.6.6. Implementación

La fase de implementación es la etapa de mayor duración del proyecto, ya que en esta se lleva a cabo la codificación del sistema, empleando las *TÉCNICAS Y HERRAMIENTAS*, descritas previamente.

El desarrollo se ha llevado a cabo partiendo de los resultados obtenidos en la fase anterior Diseño del sistema, descrita en el **Anexo IV: Diseño del sistema software**.

Para consultar la información más detallada sobre la implementación técnica del proyecto consultar el **Anexo V: Documentación Técnica**, en el cual se incluye la referencia a la documentación generada del código implementado.

A continuación se describirá el proceso seguido para la realización del proyecto, dividiéndolo en dos grandes bloques: frontend y API.

M.6.6.1. Frontend

El proceso de implementación del frontend comenzó con la planificación de las distintas vistas de usuario y su implementación empleando el framework vue.

Según avanzaba el proceso de las vistas de usuario se realizó de forma paralela a la implementación de la API, la planificación e implementación de las llamadas necesarias a esta para poder obtener y presentar al usuario la información solicitada. Así como la implementación del almacenamiento centralizado de Vuex.

Por último se implementaron los dos mapas de la aplicación, tanto el mapa en tiempo real como el mapa de calor de los jugadores, ya que para la realización de ambos se necesitaba tener desarrollada previamente toda la lógica de la API.

M.6.6.2. API

Como se ha comentado anteriormente, se realizó la implementación de la API, de forma paralela a la implementación del frontend.

En esta se ha llevado a cabo la implementación de las conexiones con ambas bases de datos, MongoDB e influxDB.

También se ha implementado toda la lógica necesaria, para realizar la conexión de la API, con la Red de localización, permitiendo de esta forma obtener las posiciones de las balizas localizadoras y almacenarlas en la base de datos influxDB, a la que se solicitarían a través de la conexión implementada y de las distintas queries, los datos necesarios de cada jugador.

M.6.7. Pruebas

Las pruebas son una parte muy importante en el desarrollo de un sistema software, ya que permite comprobar el buen funcionamiento del mismo y encontrar posibles errores, que podían haber pasado inadvertidos sin la realización de pruebas.

Durante la realización del proyecto sean ido realizando pruebas regularmente, en el proceso de implementación de cada componente y al finalizar la misma. El realizar estas pruebas permiten encontrar con mayor facilidad el error y solucionarlo antes de propagarlo.

Al finalizar la implementación de cada componente se realizaba una prueba no solo del componente, si no del proyecto entero por si las modificaciones podían haber tenido consecuencias en otros componentes.

Así mismo, al finalizar la implementación del sistema completo, se han realizado numerosas pruebas del sistema final, con el fin de comprobar el correcto funcionamiento del sistema previamente a la entrega del mismo.

Estas pruebas han sido muy útiles en la realización del proyecto, ya que han permitido encontrar pequeños defectos que se han podido resolver con facilidad.

M.6.8. Documentación

Como se ha mencionado anteriormente se ha realizado la documentación del código para permitir consultar los ficheros empleados y ver la utilidad de ellos, para ello se ha empleado la herramienta *JSDoc*, descrita previamente.

Se podrá consultar la información más detalla sobre la documentación técnica en el **Anexo V: Documentación Técnica.**

En la *Figura 34: Interfaz gráfica documentación técnica* se muestra la interfaz de la página HTML generada, esta permite acceder a los ficheros y consultar su código.

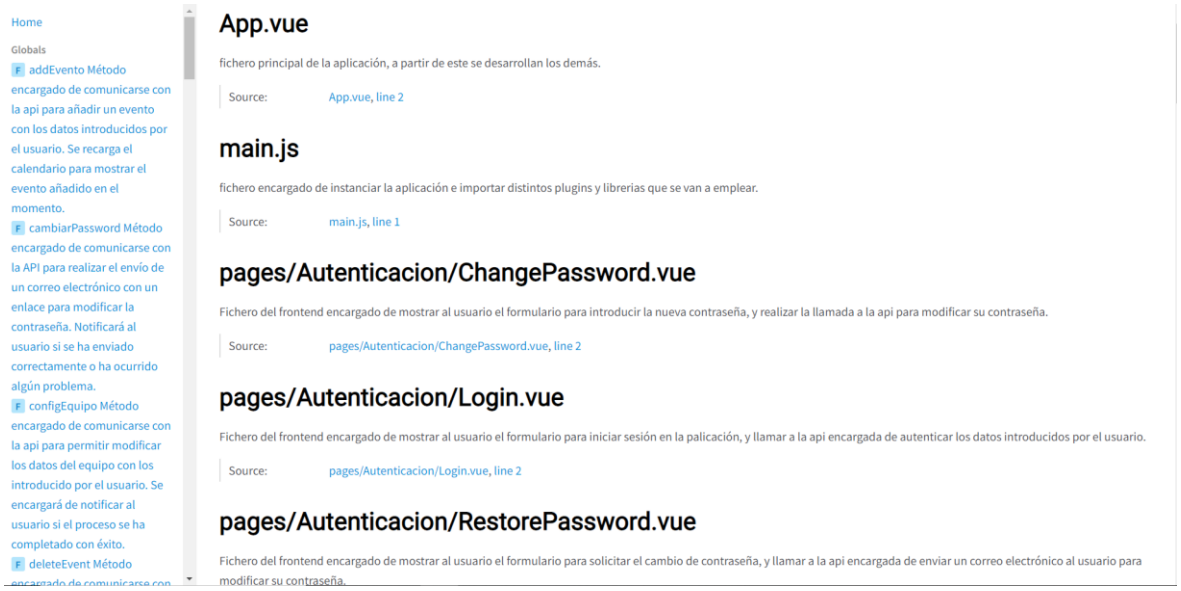

Figura 34: Interfaz gráfica documentación técnica

M.6.9. Funcionalidad del sistema

La funcionalidad del sistema reside en la página web, ya que las balizas localizadoras no permiten la interacción del usuario, únicamente son los dispositivos encargados de enviar a la API, las posiciones de los jugadores.

Se va a mostrar una selección las pantallas principales de la página web con las que se pretenden mostrar el diseño empleado en la totalidad de la aplicación y algunas de sus funcionalidades, divididas por grupos de gestión.

Para consultar la funcionalidad completa y detallada del sistema consultar el **Anexo VI: Manual de usuario.**

M.6.9.1. Autenticación

A continuación se muestra el diseño de la vista empleada para las acciones de autenticación realizadas por el usuario no logueado, estas engloban: Registro, Inicio de sesión y las pantallas encargadas de Recuperar contraseña.

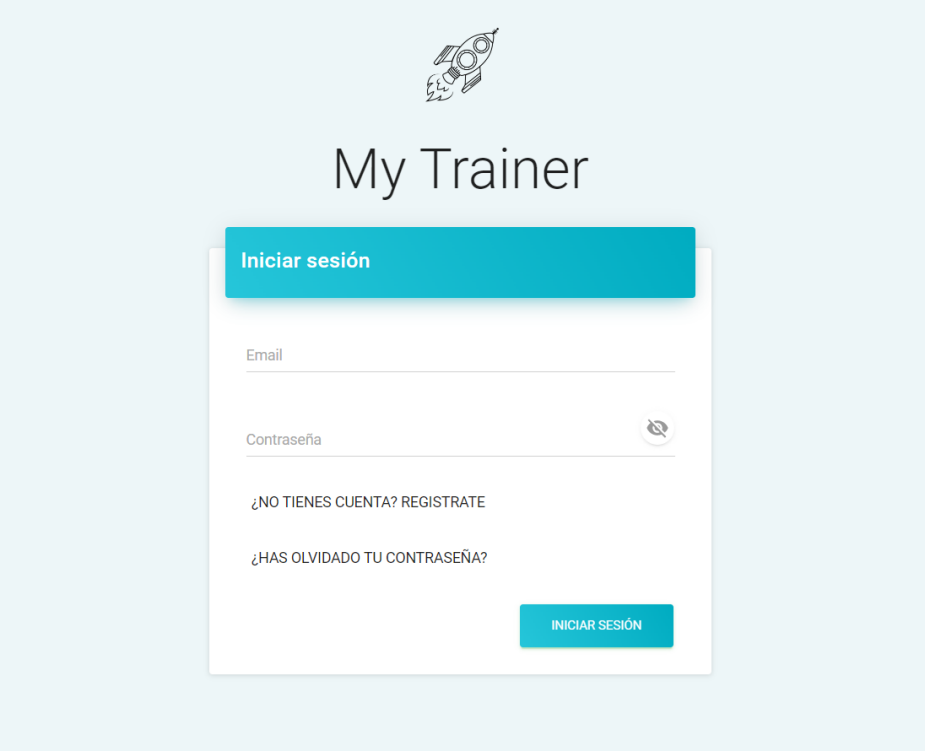

Figura 35: Pantalla inicio de sesión

Se va a destacar le opción de recuperar la contraseña, primeramente se mostrará el siguiente formulario, donde el usuario solicitará el envio de un email para restablecer su contraseña.

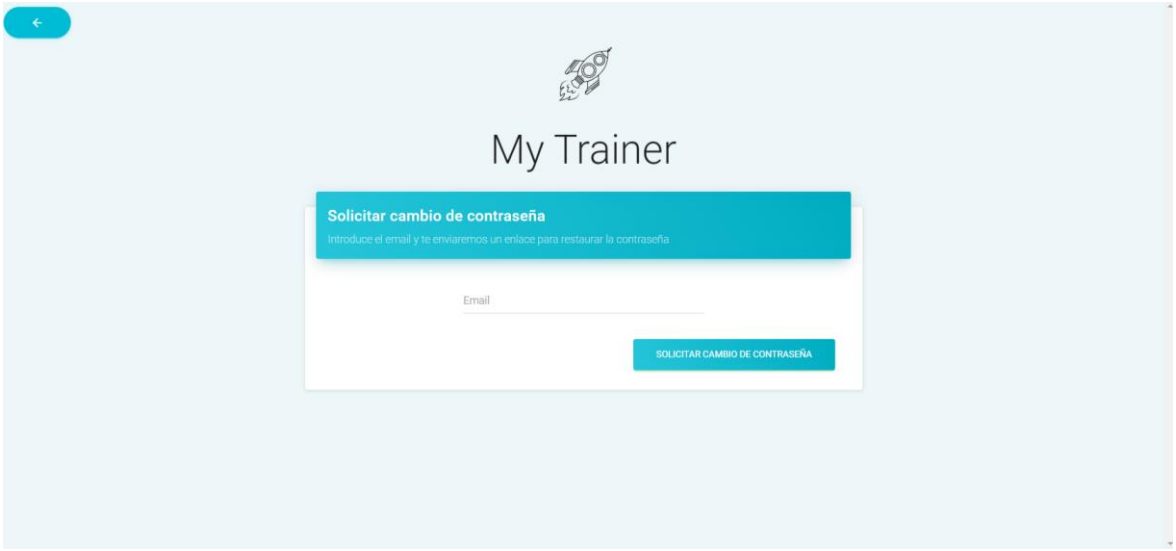

Figura 36: Pantalla restablecimiento contraseña

Se le enviará al usuario un email como el siguiente para permitirle modificar su contraseña.

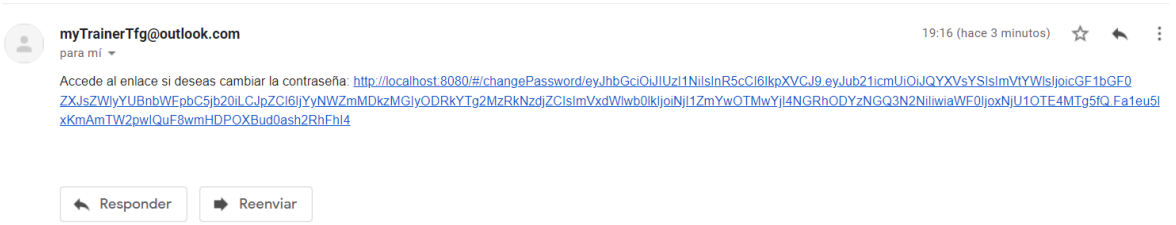

Figura 37: Email modificación contraseña

M.6.9.2. Inicio

Una vez ha iniciado sesión el usuario se accede a la pantalla de inicio, donde se puede observar la estructura principal de la aplicación, con un menú lateral de acceso a las pantallas principales y una barra de navegación superior.

Figura 38: Pantalla de inicio

M.6.9.3. Gestión de equipo

El sistema permite al usuario gestionar los datos del equipo, mostrando los datos del mismo y permitiendo modificarlos.

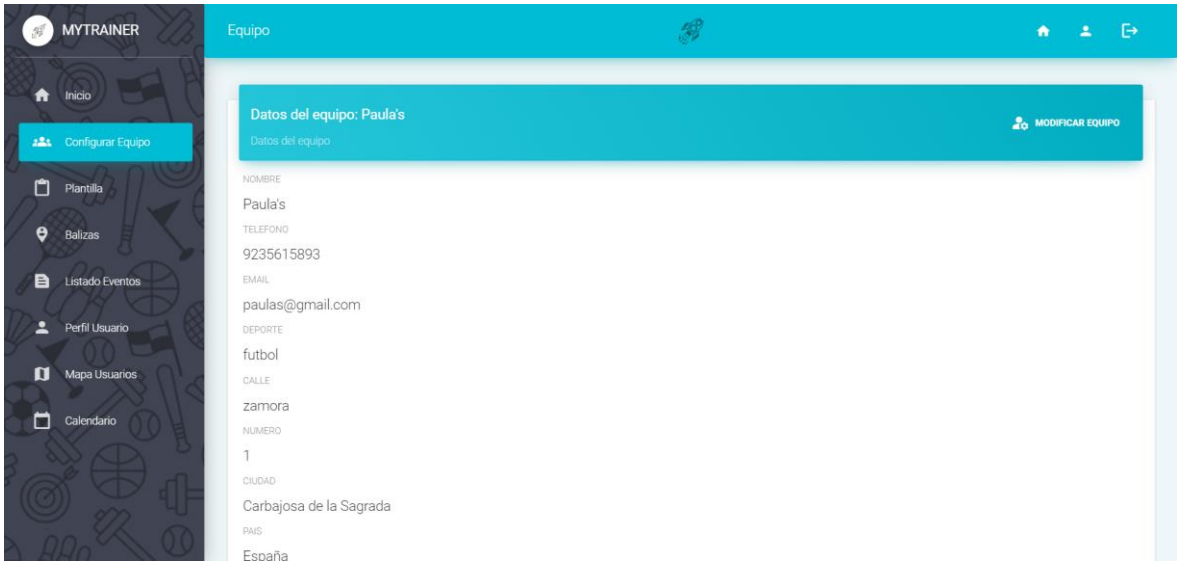

Figura 39: Pantalla datos del equipo

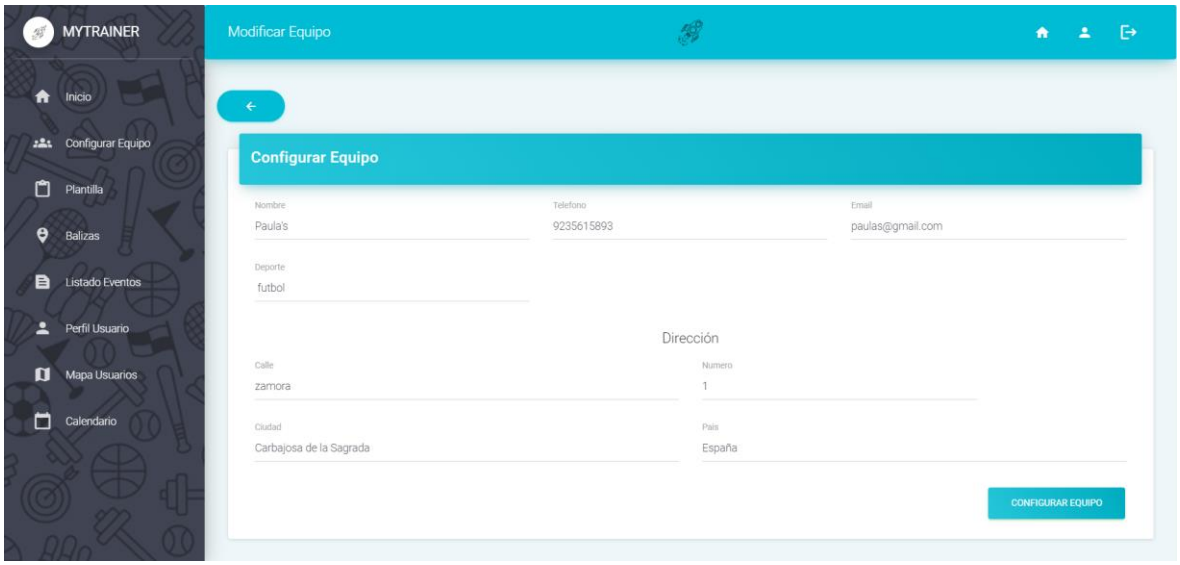

Figura 40: Pantalla modificar equipo

M.6.9.4. Gestión jugadores

El usuario podrá consultar los jugadores pertenecientes a su equipo, a partir de la opción "Plantilla", una vez ha accedido a esta el usuario podrá añadir nuevos jugadores, además podrá acceder a los datos del mismo y a sus estadísticas, pantalla en la que se hará mayor hincapié.

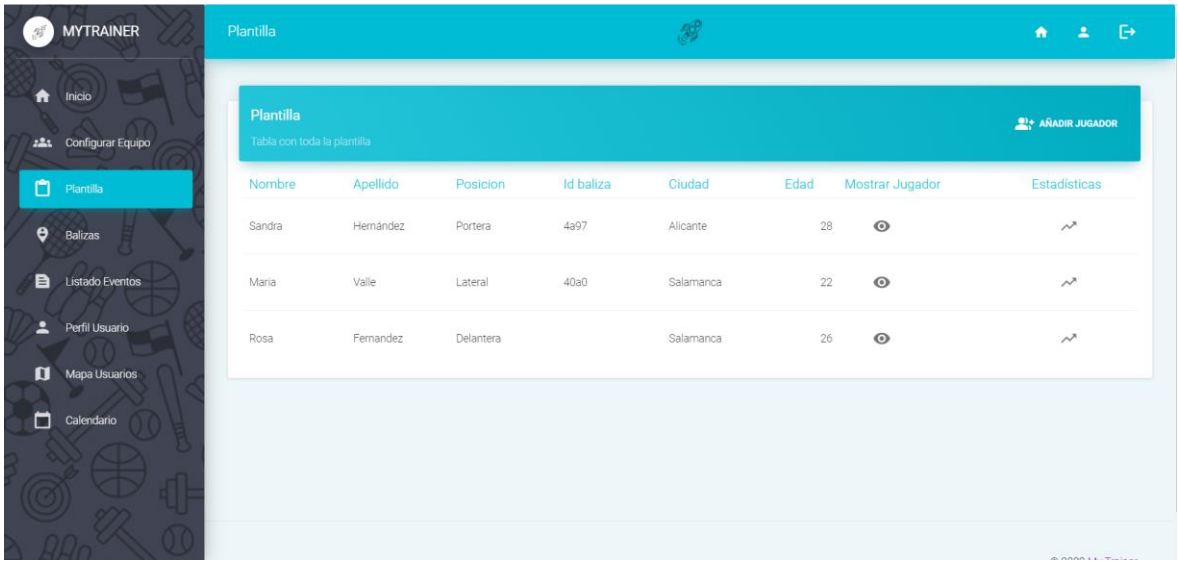

Figura 41: Pantalla Plantilla

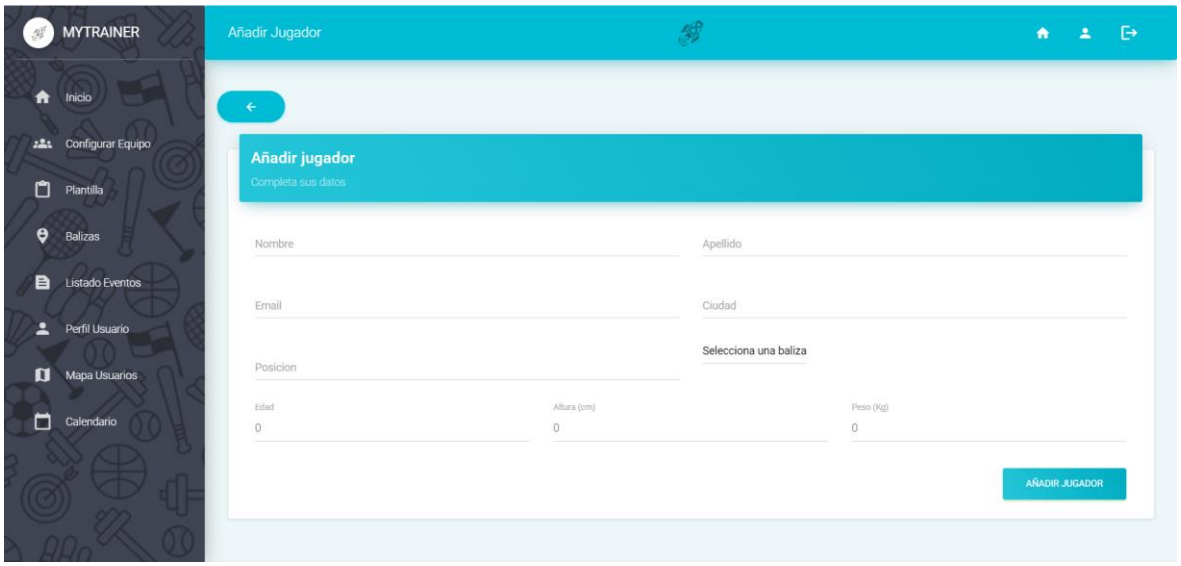

Figura 42: Pantalla añadir jugador

Al acceder a través del icono 👁 , se le mostrarán los datos del jugador seleccionado, además desde esta pantalla podrá acceder a las pantallas que le permitirán modificar y eliminar el jugador.

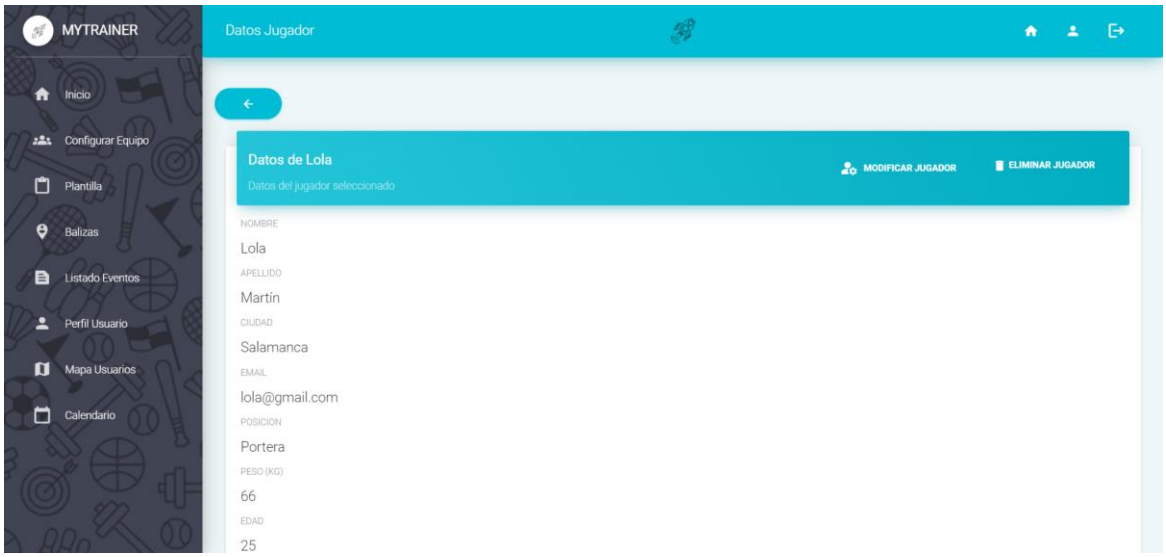

Figura 43: Pantalla datos jugador

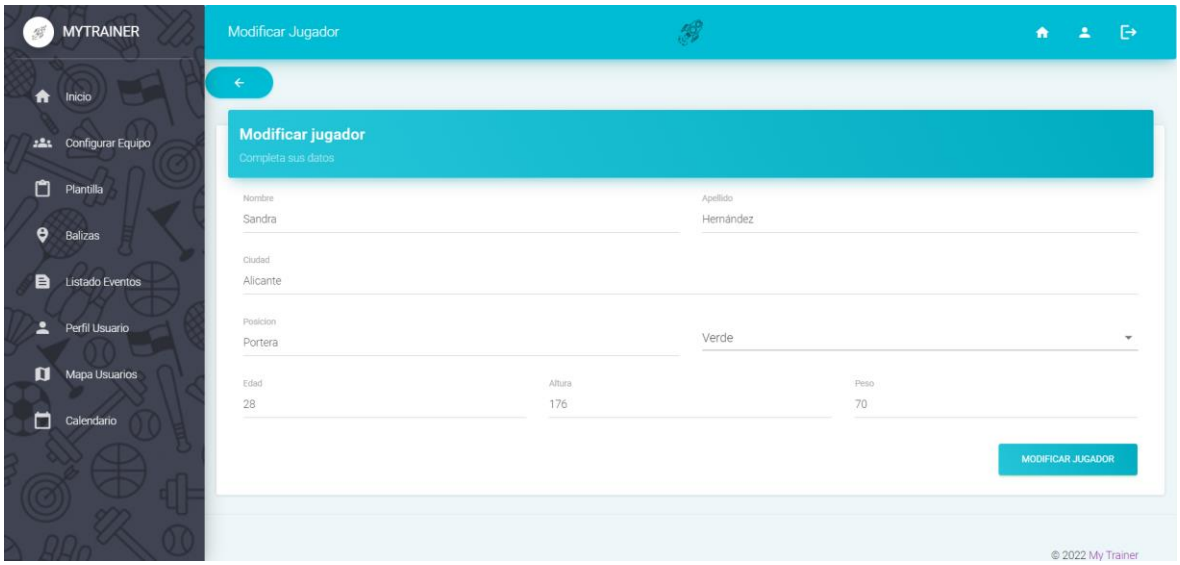

Figura 44: Pantalla modificar jugador

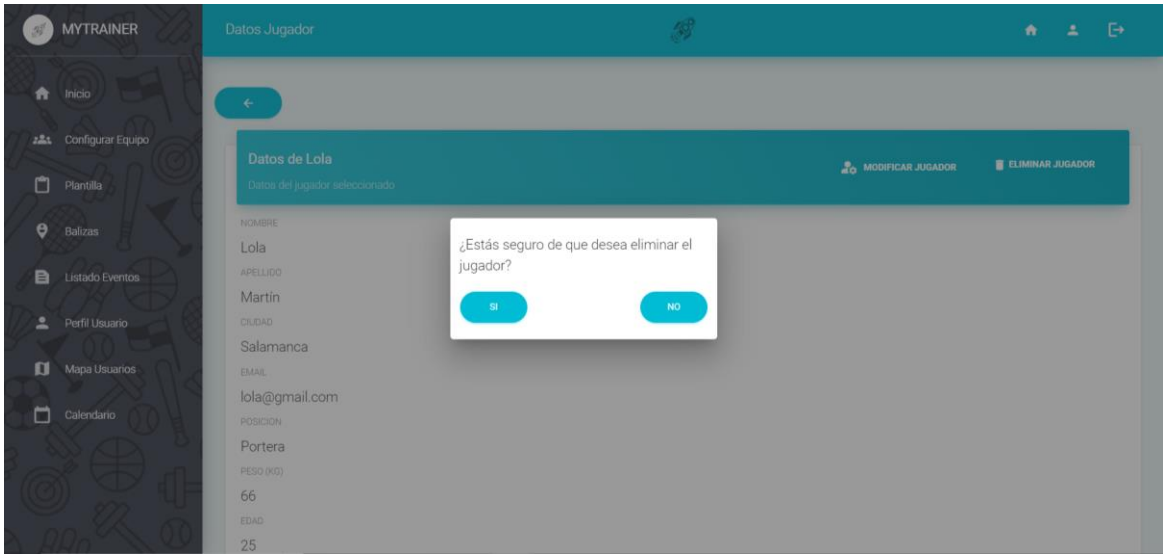

Figura 45: Pantalla eliminar jugador

Por último el usuario puede acceder a las estadísticas del jugador, mediante el icono , en esta pantalla el sistema le mostrará las estadísticas del jugador en cada evento, en un formato gráfica permitiendo comparar los datos obtenidos en cada uno de ellos.

Además se le mostrará un botón para acceder al mapa de calor del jugador.

Figura 46: Pantalla estadísticas del jugador

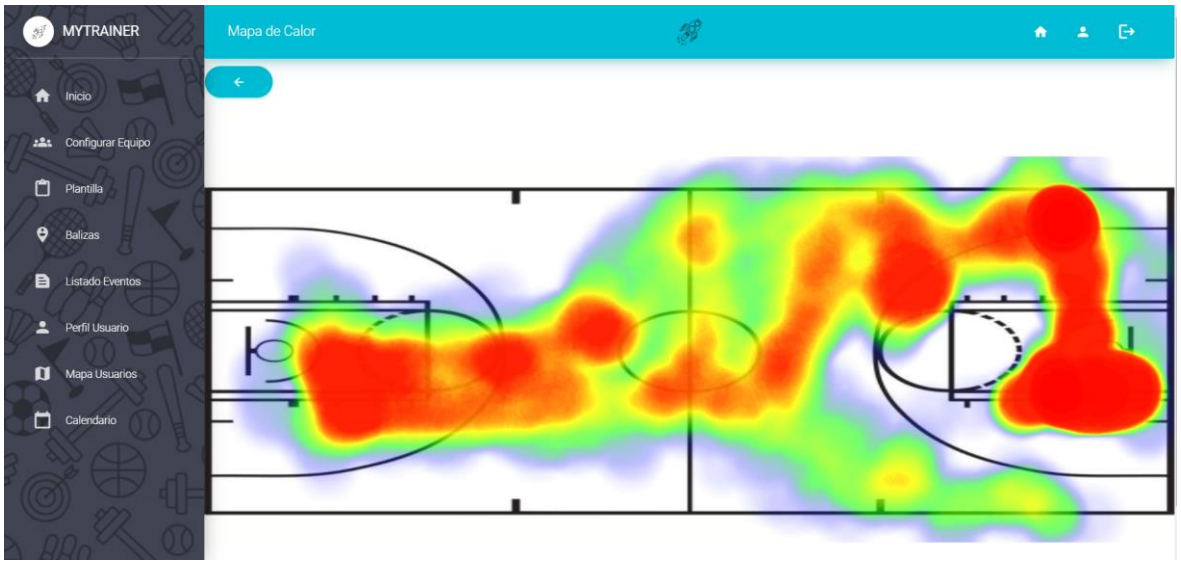

Figura 47: Pantalla mapa de calor

M.6.9.5. Gestión de balizas

El usuario accederá a la pantalla balizas, permitiendo al usuario consultar el listado de balizas, además permitirá al usuario añadir una baliza, editarla o eliminarla.

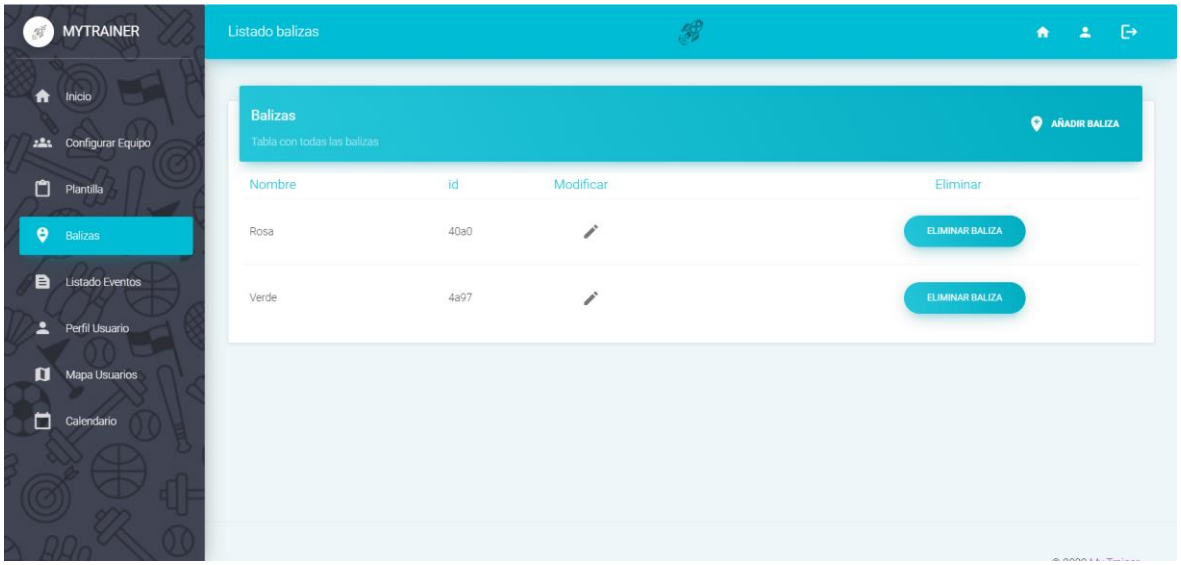

Figura 48: Pantalla balizas

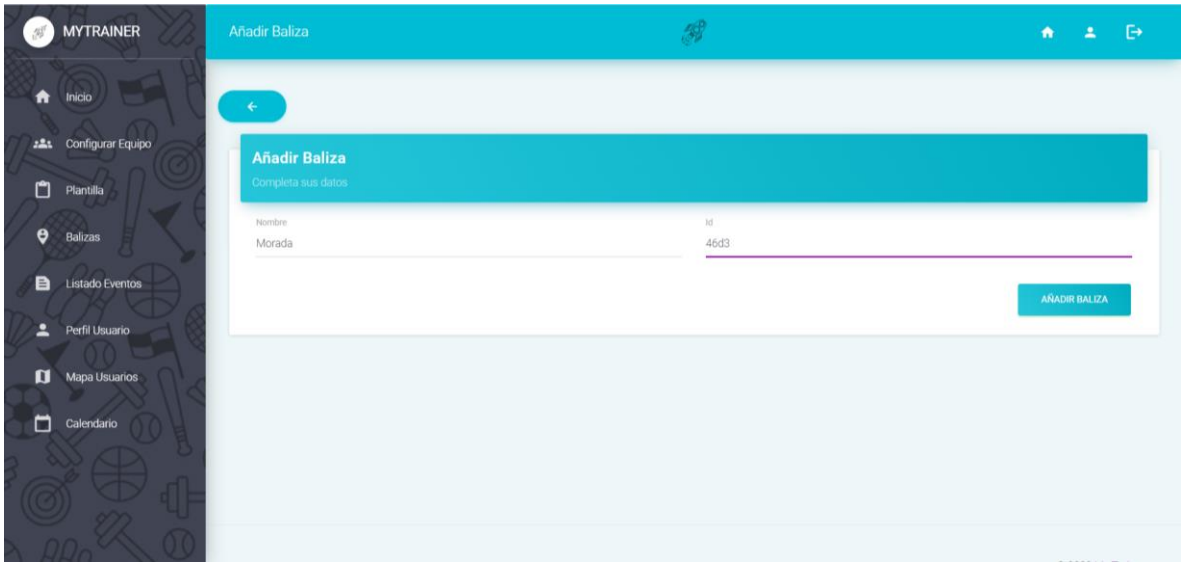

Figura 49: Pantalla añadir baliza

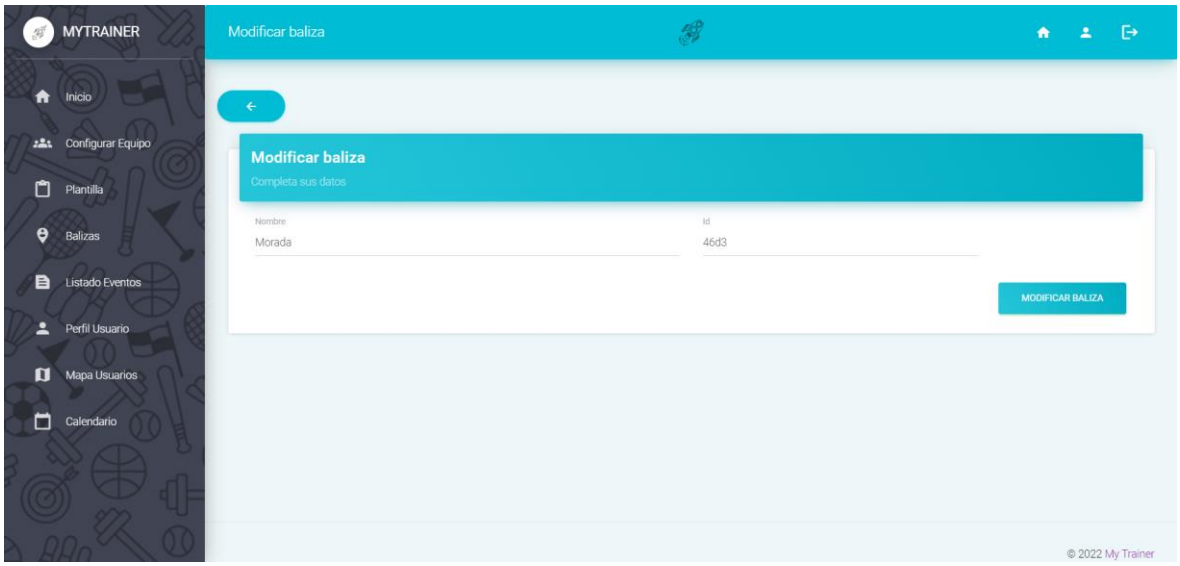

Figura 50: Pantalla modificar baliza

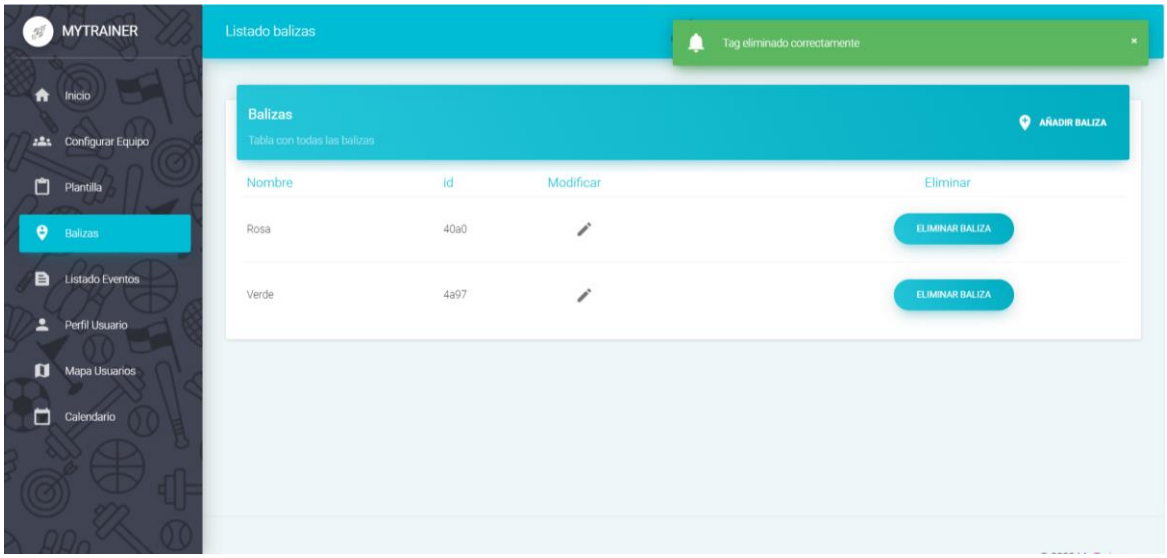

Figura 51: Pantalla baliza eliminada

M.6.9.6. Gestión eventos

El usuario podrá consultar los usuarios mediante un listado de eventos o a través del calendario.

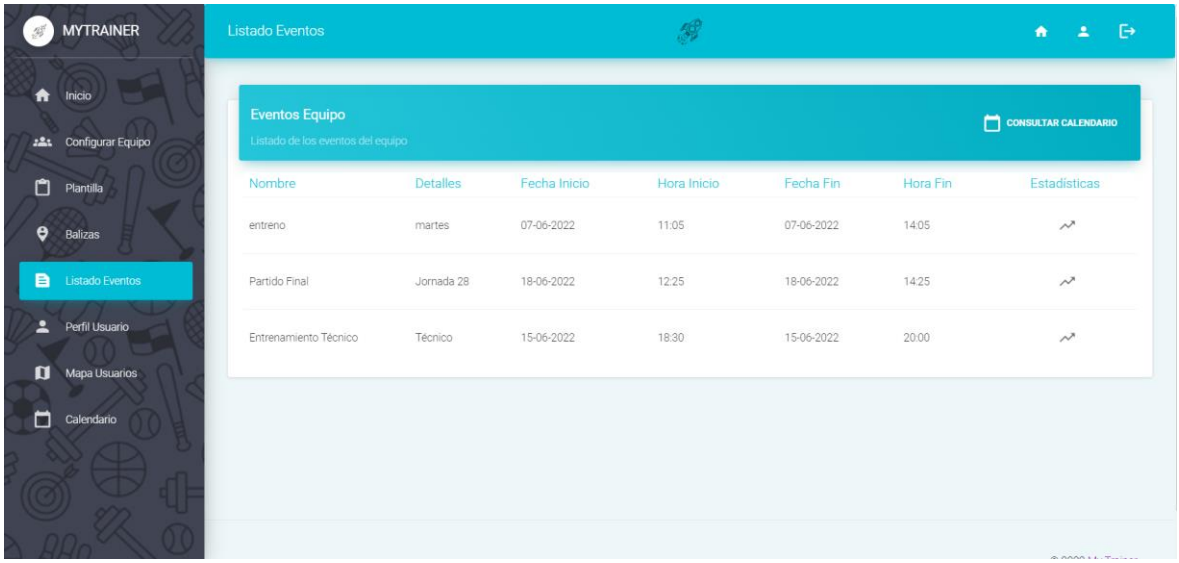

Figura 52: Pantalla listado eventos

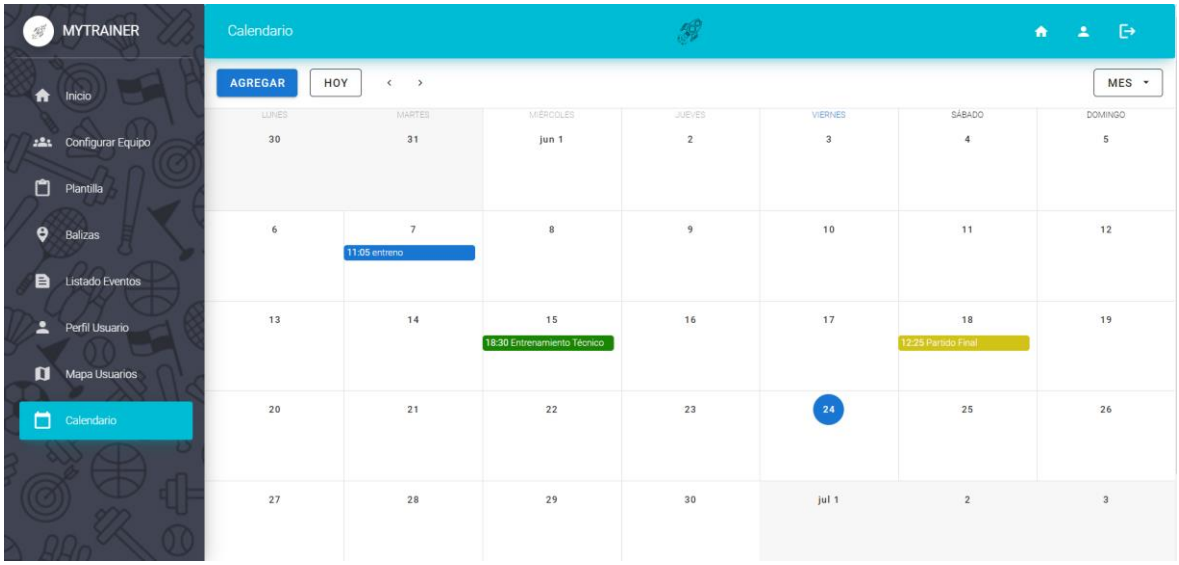

Figura 53: Pantalla calendario

Una vez ha accedido al calendario, al seleccionar un evento, se le permitirá editarlo, eliminarlo o consultar las estadísticas del evento. Además de añadir un nuevo evento.

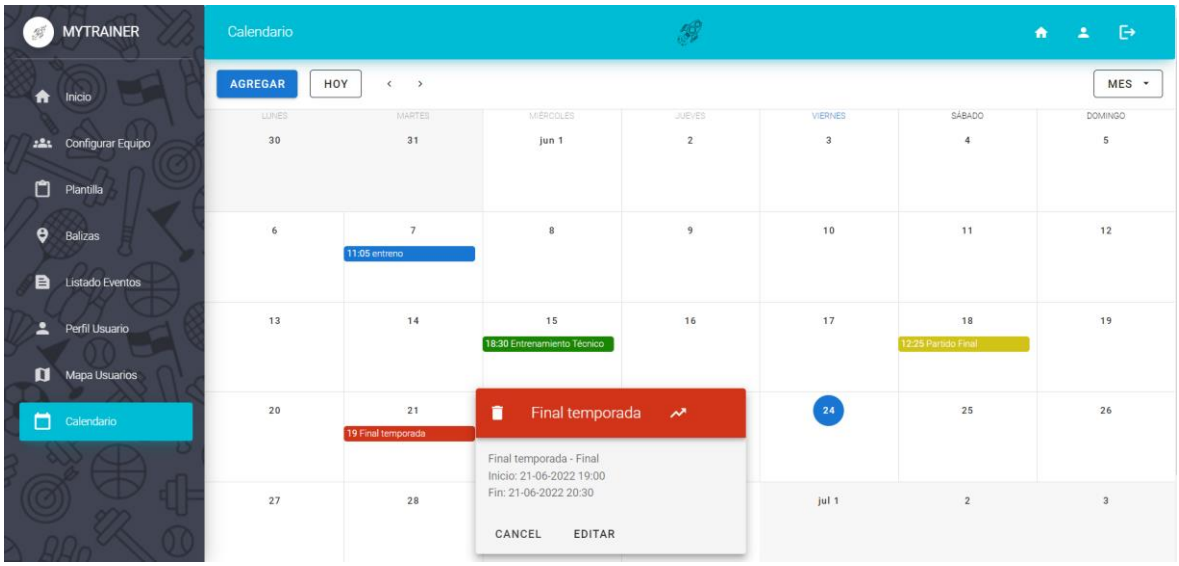

Figura 54: Pantalla mostrar evento

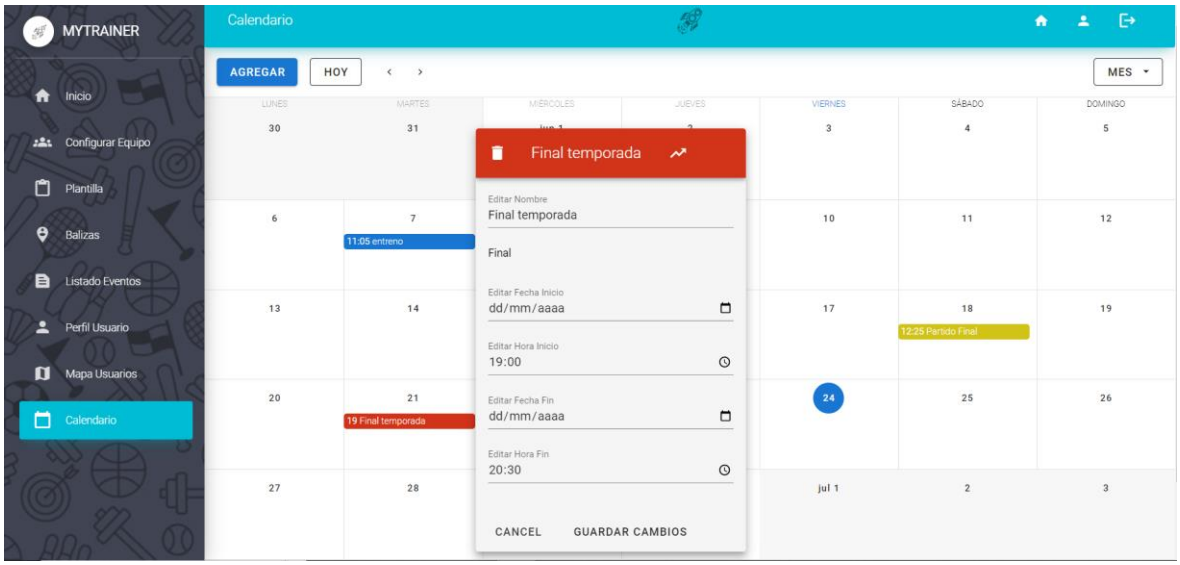

Figura 55: Pantalla editar evento

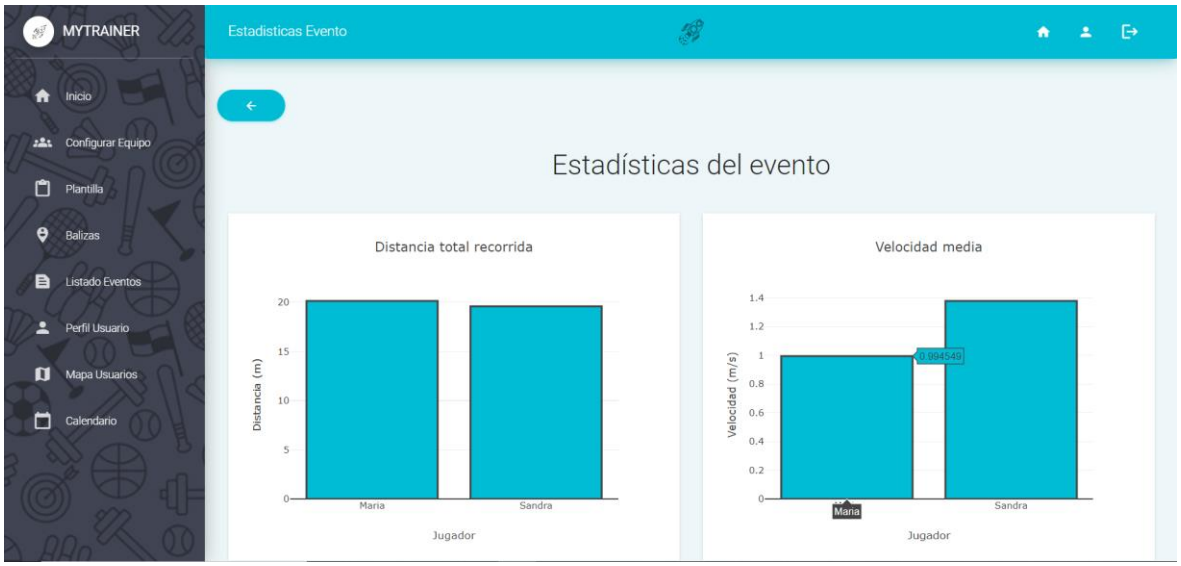

Figura 56: Mostrar estadísticas evento

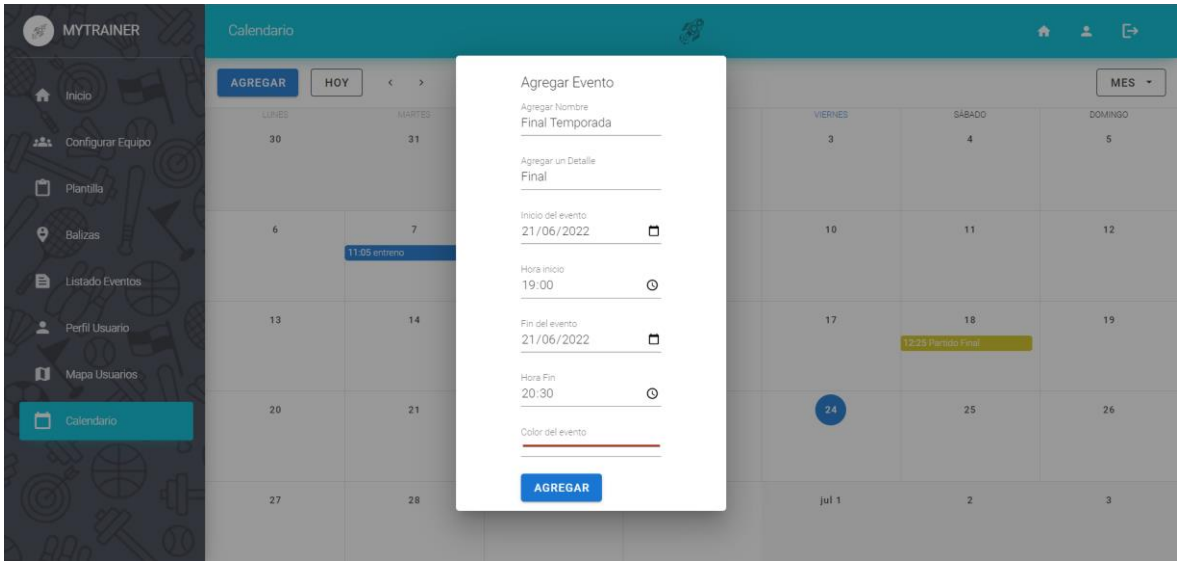

Figura 57: Pantalla agregar evento

M.6.9.7. Perfil usuario

El usuario podrá consultar sus datos accediendo a la pantalla perfil de usuario, donde podrá modificar su contraseña.

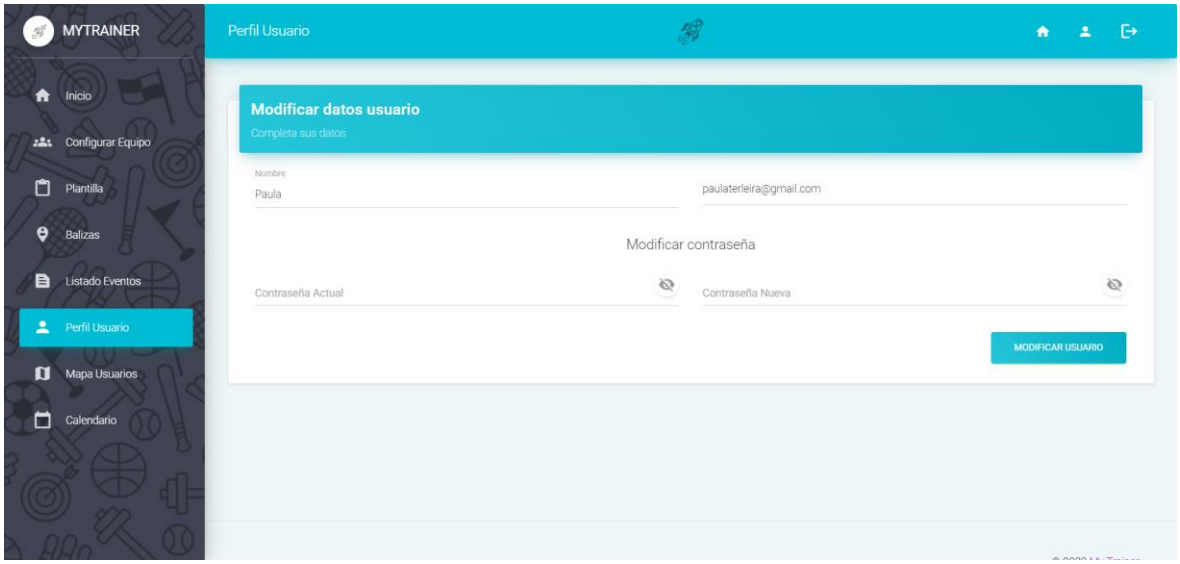

Figura 58: Pantalla perfil usuario

M.6.9.8. Mapa de usuarios tiempo real

El usuario podrá acceder a un mapa en el cual consultará las posiciones en tiempo real de los jugadores del equipo.

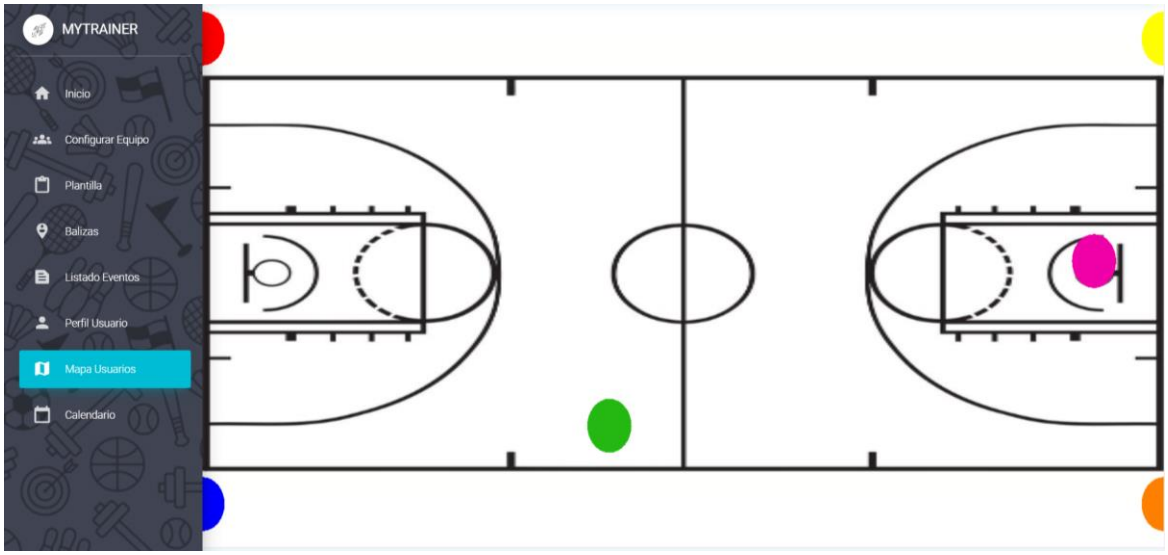

Figura 59: Pantalla mapa tiempo real

M.7. CONCLUSIONES

Una vez finalizado el desarrollo del proyecto es interesante realizar un análisis del proceso seguido y los objetivos conseguidos.

Al finalizar el proyecto, se puede observar cómo se ha conseguido la realización de un sistema funcional que permite al usuario darse de alta, controlar sus datos y los de su equipo, además podrá tratar la información de los jugadores añadidos a su equipo, los tags y los eventos del equipo.

Principalmente se ha conseguido recuperar la información de las posiciones obtenidas mediante una red de sensores inalámbricos, y tratar esta información mostrándola al usuario mediante distintas tablas y gráficas de estadísticas, y sobre todo mediante la representación de la posición de los jugadores en tiempo real sobre un mapa representado por el terreno de juego, y un mapa de calor mostrando las zonas recorridas por los jugadores en mayor o menor medida.

Se puede observar cómo el proyecto cumple con los objetivos preestablecidos, tanto los objetivos funcionales como los objetivos no funcionales.

- El sistema permite la correcta gestión de usuarios, permitiendo el registro, inicio de sesión, modificación de contraseña y recuperación de la misma en caso de olvido mediante el envío de un email.
- El sistema permite agregar el equipo del usuario, consultar sus datos principales y modificar los mismos.
- El sistema permite que un usuario añada jugadores a su equipo, consulte sus datos, modifique los mismos y pueda eliminarlos en caso de que lo desee. Además el usuario podrá consultar las estadísticas y mapa de calor para cada uno de los jugadores del equipo.
- El sistema permite que el usuario añada las balizas localizadoras a su equipo y las asocie al jugador que va a utilizarla en cada momento, así mismo permitirá desasociar las mismas y eliminarlas en caso de no volver a usarlas en dicho equipo.
- El sistema permite al usuario consultar sus eventos de distintas formas, en forma de lista y visualmente mediante el uso de un calendario interactivo, al que se añadirán los eventos nuevos, se eliminaran los deseados y se podrá acceder a consultar los datos de los mismos, así como a modificarlos y consultar las estadísticas de cada evento.
- El sistema mostrará al usuario las distintas estadísticas del equipo, desglosado por jugador, además se permitirá acceder a los datos de cada evento desglosados de nuevo por jugador y a las estadísticas de cada jugador desglosadas por cada evento en el que ha participado

Al igual que los objetivos funcionales, también se han cumplido los objetivos no funcionales, por ejemplo, llevando a cabo la encriptación de las claves de los usuarios y la verificación de los usuarios para poder acceder a cada pantalla de la aplicación, permitiendo de esta forma mantener un nivel de seguridad óptimo o permitiendo la consulta de datos en tiempo real de forma ágil.

Además de los objetivos teóricos anteriormente comentados, a nivel personal, considero que la realización del proyecto me ha supuesto un gran desarrollo, ya que he podido aplicar los conceptos teóricos aprendidos durante la realización del grado, y esto me ha permitido comprenderlos mucho mejor, ya que es distinto el aprendizaje teórico a tener la posibilidad de aplicar lo aprendido de forma práctica.

También se valora positivamente el aprendizaje de numerosas técnicas y herramientas, empleadas en el desarrollo del proyecto, y que no se habían utilizado previamente, como por ejemplo Javascript, Vue, SocketIo o bibliotecas como heatmap.js, three.js o plotly.js, las cuales no conocía y me han parecido muy interesantes y que tienen muchas posibilidades.

Además he aprendido a trabajar con dos bases de datos con las que no había trabajado, y he comprendido mejor como se realiza la integración de las bases de datos en un proyecto.

En definitiva la ejecución de este proyecto ha sido muy positiva para mi, ya que me ha permitido obtener experiencia en la realización de proyectos software, lo que pienso que me ha permitido afianzar mi conocimiento de cara al inicio de mi carrera en el mundo laboral.

M.8. LINEAS FUTURAS

Una vez realizado el proyecto software, se realizará un análisis en busca de posibles mejoras de cara al futuro:

- Añadir nuevos mapas estadísticos, similares al mapa de calor pero dividiendo el terreno de juego en zonas, mostrando el porcentaje de tiempo en esa zona.
- Investigar sobre la aplicación de los datos almacenados como peso, altura y edad, junto con los datos obtenidos de las balizas localizadoras, para obtener datos de las Kcal gastadas o similares.
- Implementarlo para deportes de exterior con otro tipo de localizadores que abarquen distancias mayores.

M.9. REFERENCIAS

[1]. *FutApp*: *Software para Entrenadores Y Equipos de Fútbol*. (2022, 17 junio). *FutApp*. Recuperado 20 de junio de 2022, de https://futapp.es/

[2]. *CatapultSports: Sports technology* ,(s.f), Recuperado 20 de junio 2022, de https://www.catapultsports.com/es/.

[3]. *RTLS: Real Time Location Systems.* (2014, 14 abril). Recuperado 21 junio 2022 de https://www.decawave.com/sites/default/files/resources/aps003_dw1000_rtls_introduction.pdf

[4]. **Llamas L**. (2019, 17 abril). *¿Qué es MQTT? Su importancia como protocolo IoT*. Luis Llamas. Recuperado 21 de junio 2022 de https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/

[5]. *MQTT - The Standard for IoT Messaging*. (s. f.). MQTT. Recuperado 20 de junio de 2022, de https://mqtt.org/

[6]. *WebSockets - Referencia de la API Web | MDN*. (2022, 3 marzo). WebSockets. Recuperado 21 de junio de 2022, de https://developer.mozilla.org/es/docs/Web/API/WebSockets_API

[7]. Ltd, R. P. (s. f.-b). Buy a 3 Model B+ –. Raspberry Pi. Recuperado 22 de junio de 2022, de https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/

[8]. *DWM1001C - Qorvo*. (2017). Qorvo - RF Solutions for Mobile, Infrastructure and Defense. Recuperado 22 de junio de 2022, de https://www.qorvo.com/products/p/DWM1001C

[9]. *Visual Studio Code - Code Editing. Redefined*. (2021, 3 noviembre). Visual Studio Code. Recuperado 22 de junio de 2022, de https://code.visualstudio.com/

[10]. *HTML: Lenguaje de etiquetas de hipertexto | MDN*. (2021, 20 abril). HTML. Recuperado 22 de junio de 2022, de https://developer.mozilla.org/es/docs/Web/HTML

[11]. *Sass: Syntactically Awesome Style Sheets*. (s. f.). Sass. Recuperado 22 de junio de 2022, de https://sass-lang.com/

[12]. *CSS Tutorial*. (s. f.). CSS. Recuperado 22 de junio de 2022, de https://www.w3schools.com/css/

[13]. *Lenguaje Javascript - Javascript en español*. (s. f.). Lenguaje JS. Recuperado 23 de junio de 2022, de https://lenguajejs.com/javascript/

[14]. *Vue.js - The Progressive JavaScript Framework | Vue.js*. (s. f.). VueJS. Recuperado 23 de junio de 2022, de https://vuejs.org/

[15]. *Vue Router*. (s. f.). Vue Router. Recuperado 23 de junio de 2022, de https://router.vuejs.org/

[16]. *What is Vuex? | Vuex*. (s. f.-b). Vuex. Recuperado 23 de junio de 2022, de https://vuex.vuejs.org/

[17]. *npm: vue-material*. (2020, 13 agosto). Vue-Material. Recuperado 22 de junio de 2022, de https://www.npmjs.com/package/vue-material

[18]. *Vuetify -- A Material Design Framework for Vue.js*. (s. f.). Vuetify. Recuperado 23 de junio de 2022, de https://vuetifyjs.com/en/

[19]. *Axios*. (s. f.). Axios. Recuperado 22 de junio de 2022, de https://axios-http.com/

[20]. *Dynamic Heatmaps for the Web*. (s. f.). Heatmap. Recuperado 22 de junio de 2022, de https://www.patrick-wied.at/static/heatmapjs/?utm_source=cdnjs&utm_medium=cdnjs_link&utm_campaign=cdnjs_library

[21]. *Three.js – JavaScript 3D library*. (s. f.). Three Js. Recuperado 23 de junio de 2022, de https://threejs.org/

[22]. *Plotly*. (s. f.). Plotly. Recuperado 23 de junio de 2022, de https://plotly.com/javascript/

[23]. *NodeJS*. (s. f.). Nodejs. Recuperado 23 de junio de 2022, de https://nodejs.org/es/

[24]. *npm*. (s. f.). NPM. Recuperado 24 de junio de 2022, de https://www.npmjs.com/

[25]. MongoDB. (s. f.). *MongoDB: La Plataforma De Datos Para Aplicaciones*. https://www.mongodb.com/es

[26]. *Mongoose ODM v6.4.1*. (s. f.). Mongoose. Recuperado 24 de junio de 2022, de https://mongoosejs.com/

[27]. *Express - Infraestructura de aplicaciones web Node.js*. (s. f.). Express. Recuperado 24 de junio de 2022, de https://expressjs.com/es/

[28]. InfluxData. (2022, 17 junio). *InfluxDB: Open Source Time Series Database*. Recuperado 24 de junio de 2022, de https://www.influxdata.com/

[29]. ***Reinman, A.*** (s. f.). *Nodemailer :: Nodemailer*. Nodemailer. https://nodemailer.com/about/

[30]. *Socket.IO*. (s. f.). SocketIO. Recuperado 24 de junio de 2022, de https://socket.io/

[31]. auth0.com. (s. f.). JWT.IO. JSON Web Tokens - Jwt.Io. Recuperado 24 de junio de 2022, de https://jwt.io/

[32]. *GitHub: Where the world builds software*. (s. f.). GitHub. Recuperado 25 de junio de 2022, de https://github.com/

[33]. *Ideal Modeling & Diagramming Tool for Agile Team Collaboration*. (s. f.). VisualParadigm. Recuperado 25 de junio de 2022, de https://www.visual-paradigm.com/

[34]. *Use JSDoc: Index*. (s. f.). JSDoc. https://jsdoc.app/

[35]. **García Peñalvo, F. J., García Holgado, A., Vázquez Ingelmo, A**. *Tema 5: Introducción al proceso unificado*. Transparencias de Ingeniería del Software I.

Recuperado 11/06/2022 de https://repositorio.grial.eu/bitstream/grial/1144/1/IS_I%20Tema%205%20-%20Proceso%20Unificado.pdf

[36]. **Moreno García, M.N.** *Práctica 1: Estimación del esfuerzo*. Transparencias de Gestión de proyectos.

[37]. **Moreno García, M.N.** *Práctica 2: Planificación temporal*. Transparencias de Gestión de proyectos.

[38]. **García Peñalvo, F. J., García Holgado, A., Vázquez Ingelmo, A.** . *Tema 4: Ingeniería de requisitos*. Transparencias de Ingeniería del Software I. Tema 4. Recuperado 28/05/2022 de https://repositorio.grial.eu/bitstream/grial/2515/1/IS_I%20Tema%204%20-%20Ingenieri%cc%81a%20de%20Requisitos.pdf

[39]. **Durán Toro, A., Bernárdez Jiménez, B.** (2002). *Metodología para la Elicitación de Requisitos de Sistemas Software* (2.3 ed.).

[40]. **Rodríguez, E.** (2020, 30 diciembre). *MVVM – Qué es y cómo funciona. INMEDIATUM - We build and grow business on Internet.* Recuperado 8 de junio de 2022, de *https://inmediatum.com/blog/ingenieria/mvvm-que-es-y-como-funciona/*

[41]. **Moreno García, M.N.** *Tema 1:Diseño de software.* Transparencias de Ingeniería del Software II.