



Escuela  
Politécnica  
Superior

# Plataforma de gestión de resultados de aprendizaje para fútbol base



Grado en Ingeniería Informática

## Trabajo Fin de Grado

Autor:

Jorge Moreno Ozores

Tutores:

Virgilio Gilart Iglesias

Diego Marcos Jorquera



Universitat d'Alacant  
Universidad de Alicante



# Plataforma de gestión de resultados de aprendizaje para fútbol base

---

Trabajo Fin de Grado

## Autor

Jorge Moreno Ozores

## Tutores

Virgilio Gilart Iglesias

*Tecnología Informática Y Computación*

Diego Marcos Jorquera

*Tecnología Informática Y Computación*



Grado en Ingeniería Informática



Escuela  
Politécnica  
Superior



Universitat d'Alacant  
Universidad de Alicante

ALICANTE, Junio 2024



# Índice general

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>Marco Teórico</b>	<b>3</b>
2.1	Estudio del mercado . . . . .	3
2.1.1	Bcoach . . . . .	3
2.1.1.1	Manejo de estadísticas e indicadores de rendimiento . . . . .	4
2.1.2	Football tactical Board . . . . .	5
2.1.2.1	Manejo de estadísticas e indicadores de rendimiento . . . . .	5
2.1.3	Sportlyzer Players App . . . . .	6
2.1.3.1	Entrenadores . . . . .	6
2.1.3.2	Club es . . . . .	7
2.1.3.3	Precios . . . . .	8
2.1.3.4	Manejo de estadísticas e indicadores de rendimiento . . . . .	8
2.1.4	LongoMatch . . . . .	8
2.1.4.1	Manejo de estadísticas e indicadores de rendimiento . . . . .	9
2.1.5	Picco: Estadísticas y rendimiento . . . . .	9
2.1.5.1	Manejo de estadísticas e indicadores de rendimiento . . . . .	10
2.1.6	Teamtag . . . . .	10
2.1.6.1	Manejo de estadísticas e indicadores de rendimiento . . . . .	11
2.1.7	Conclusiones . . . . .	11
2.1.7.1	Manejo de estadísticas e indicadores de rendimiento . . . . .	12
2.2	Tabla comparativa . . . . .	14

<b>3</b>	<b>Objetivos</b>	<b>15</b>
3.1	Objetivos generales . . . . .	15
3.2	Objetivos específicos . . . . .	16
<b>4</b>	<b>Metodología</b>	<b>17</b>
4.1	Metodología de realización . . . . .	17
4.2	Elección de las herramientas . . . . .	17
4.2.1	Github . . . . .	18
4.2.2	Visual Studio Code . . . . .	19
4.2.3	Draw.io . . . . .	19
4.2.4	Moqups . . . . .	20
4.2.5	Vue.js . . . . .	20
4.2.5.1	Dependencias . . . . .	21
4.2.6	MySQL . . . . .	25
4.2.7	Framework DAO . . . . .	25
<b>5</b>	<b>Desarrollo</b>	<b>27</b>
5.1	Requisitos . . . . .	27
5.1.1	Funcionales . . . . .	27
5.1.1.1	Generales . . . . .	27
5.1.1.2	Entrenamientos . . . . .	28
5.1.1.3	Seguimiento y Evaluación del Rendimiento . . . . .	28
5.1.1.4	Equipo . . . . .	28
5.1.1.5	Intervalos de tiempo . . . . .	28
5.1.1.6	Actores . . . . .	29
5.1.2	No Funcionales . . . . .	29
5.1.2.1	Rendimiento . . . . .	29
5.1.2.2	Seguridad . . . . .	29
5.1.2.3	Usabilidad . . . . .	30
5.1.2.4	Escalabilidad . . . . .	30
5.1.2.5	Compatibilidad . . . . .	30

---

---

5.1.2.6	Fiabilidad . . . . .	30
5.2	Análisis funcional del sistema . . . . .	30
5.2.1	Historias de usuario . . . . .	30
5.3	Implementación . . . . .	33
5.3.1	Matriz DAFO . . . . .	36
5.3.1.1	Promedio GPA . . . . .	36
5.3.1.2	Desviación estándar . . . . .	36
5.3.1.3	Pendiente de tendencia . . . . .	36
5.3.1.4	Cambios positivos y negativos entre los intervalos de tiempo	36
<b>6</b>	<b>Conclusiones</b>	<b>37</b>
	<b>Bibliografía</b>	<b>39</b>

---





## Índice de figuras

2.1	Bcoach - Estadísticas de Jugadores . . . . .	4
2.2	Bcoach - Crear categoría de tarea . . . . .	5
2.3	Football Tactical Board - Pantalla del entrenador . . . . .	6
2.4	Sportlyzer Players App - Gráfico de los tiempos de finalización del entrenamiento . . . . .	8
2.5	Longomatch - Creación de eventos en el vídeo . . . . .	9
2.6	Picco - Interfaz durante el partido . . . . .	10
2.7	Teamtag - Interfaz durante el partido . . . . .	11
4.1	Logo de Github . . . . .	18
4.2	Logo de Visual Studio Code . . . . .	19
4.3	Logo de Draw.io . . . . .	19
4.4	Logo de Moqups . . . . .	20
4.5	Logo de Vue.js . . . . .	20
4.6	Icono de gráfico importado desde la biblioteca de Iconify . . . . .	21
4.7	Gráfico generado con ChartJS . . . . .	24
4.8	Logo de MySQL . . . . .	25



# Índice de tablas

2.1	Comparativa de funcionalidades de aplicaciones . . . . .	14
-----	--	----



## Índice de Códigos

4.1	Instalación del paquete iconify . . . . .	21
4.2	Importar el componente Icon en Vue . . . . .	21
4.3	Código para agregar un icono a la página . . . . .	21
4.4	Instalación del paquete iconify . . . . .	22
4.5	Código para agregar un icono a la página . . . . .	22
4.6	Código para agregar un icono a la página . . . . .	24



# **1 Introducción**





## 2 Marco Teórico

### 2.1 Estudio del mercado

Antes de empezar con el desarrollo del proyecto es importante un estudio de las alternativas en el mercado y entender el objetivo de este trabajo.

#### 2.1.1 Bcoach

Esta aplicación ofrece la opción de crear equipos personalizados y con cada uno realizar una tarea planificada animada con elementos gráficos para que sea fácilmente visualizable. Ofrece 3 planes con más o menos funcionalidades para el entrenador:

**Bcoach - Licencia Tablet/Ipad (2,99€/mes)** Este es el plan más sencillo, incluye gestión de entrenamientos; pizarra táctica; estadísticas de partidos y la app estará disponible en cualquier dispositivo Tablet/iPad. Toda la información será almacenada de manera local en un solo dispositivo.

**Bcoach Cloud - Licencia Multiplataforma (5,99€/mes)** Este plan incluye todo lo anterior, además de que la información estará sincronizada en la nube y ofrece acceso a Bcoach web.

**Bcoach Pro - Licencia + Asesoramiento(299,99€/mes)** Esta licencia es la más completa de todas, ofreciendo adicionalmente asesoramiento deportivo.

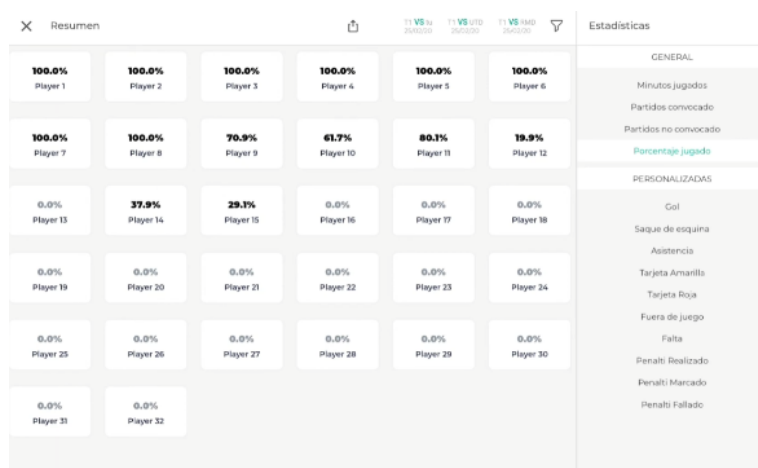
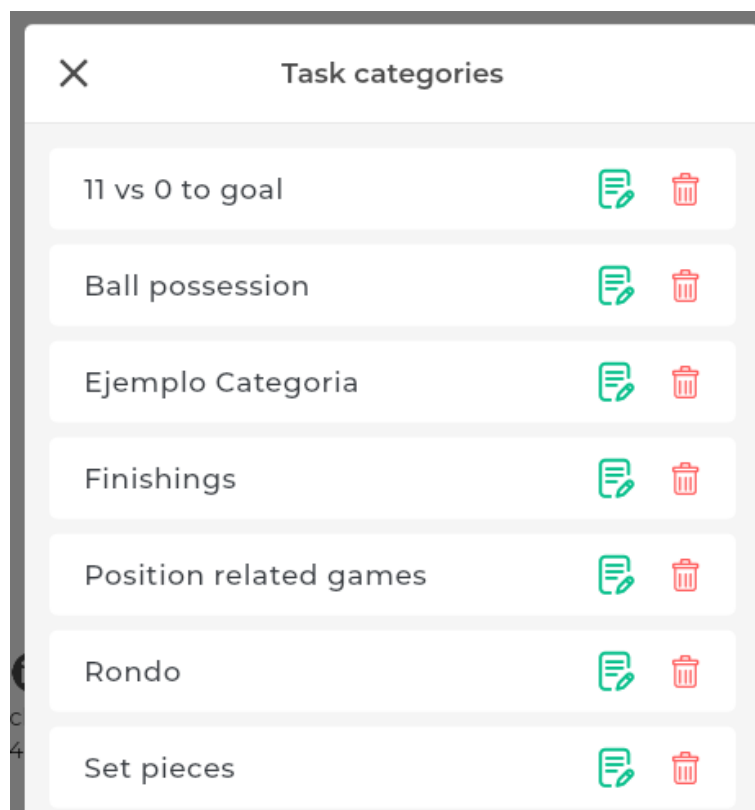


Figura 2.1: Bcoach - Estadísticas de Jugadores

### 2.1.1.1 Manejo de estadísticas e indicadores de rendimiento

Bcoach nos ofrece categorías de tarea, que se relacionarían directamente con los indicadores de rendimiento que estamos buscando. Además, ofrece estadísticas que irán siendo modificadas durante los partidos, que serán de uso para analizar esas categorías de tarea.



**Figura 2.2:** Bcoach - Crear categoría de tarea

### 2.1.2 Football tactical Board

Aplicación web gratuita con una gran variedad de deportes. Para fútbol, ofrece tres tipos de vista (Horizontal, Vertical y Medio campo). Ofrece únicamente una pizarra con figuras gráficas con la posibilidad de crear una ‘conferencia’ dónde es posible alojar a muchos usuarios que verán al mismo tiempo lo que está sucediendo en la pizarra. Dispone de una versión Pro que elimina los anuncios y permite descargar y grabar las acciones con la pizarra.

#### 2.1.2.1 Manejo de estadísticas e indicadores de rendimiento

Football Tactical Board es una herramienta principalmente enfocada en la creación de diagramas tácticos y la visualización de estrategias de juego. No está diseñada específicamente para la gestión y análisis de estadísticas avanzadas como los KPIs. Las funcionalidades ya nombradas permiten hacerse durante una conferencia en tiempo real, sin embargo, no aparece la opción sobre la creación de métricas.

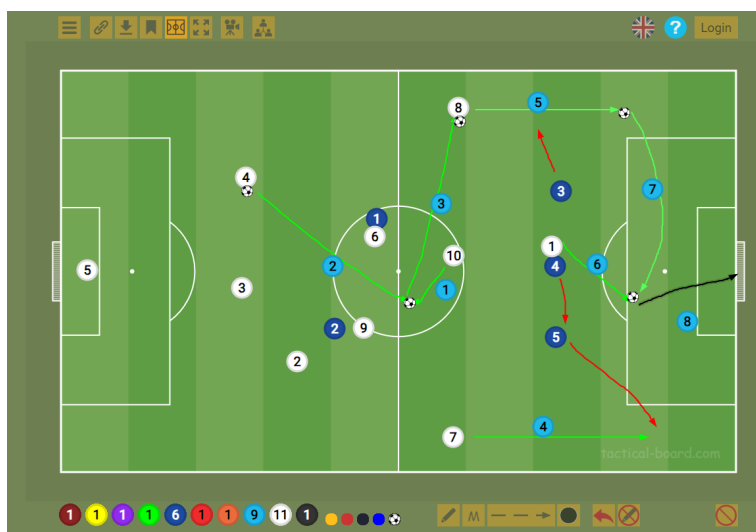


Figura 2.3: Football Tactical Board - Pantalla del entrenador

### 2.1.3 Sportlyzer Players App

Aplicación disponible tanto para Android como para IOS que ofrece opciones para entrenadores, clubes y padres

#### 2.1.3.1 Entrenadores

**Horarios y Tareas** Dispone de un calendario compartido entre todos los usuarios, cuándo el entrenador actualiza el calendario con un nuevo evento, todo el mundo podrá ver los cambios. El calendario ofrece las siguientes acciones:

- Importar partidos y competiciones desde Excel
- Configurar entrenamientos regulares
- Compartir eventos por email
- Widget para mostrar el calendario en la web del club
- Anotar asistencia de los deportistas
- Asignación y seguimiento de deberes

**Disponibilidad** Esta opción se usa para planificar actividades que necesitan mínimo de personas. Recoge información sobre qué deportistas asistirán o no a cada evento del calendario. Los enlaces son únicos para cada deportista, por lo que no necesitarán registrarse.

**Asistencia** Anota la asistencia de los deportistas, pudiendo ser evaluada en un registro con el total de ausencias de cada uno, la anotación se puede hacer de forma offline.

**Retroalimentación** Forma en la que los jugadores pueden comentar y valorar cada sesión, pueden dar una nota al entrenamiento, valorar cuán duro ha sido o simplemente dejar un comentario.

**Planificación y análisis** Para ir desarrollando a los jugadores, la plataforma permite crear ciclos con planes de entrenamiento, pudiendo calibrar su intensidad. Después de cada uno se podrá recoger los comentarios y un análisis de los resultados.

**Evaluación y prueba de jugadores** Seguimiento de cada jugador, evaluación de cada aptitud (física, táctica o técnica), personalidad y habilidades. Se puede enviar un informe de la evaluación a cada jugador.

**Mensajería** Permite enviar SMS o correos electrónicos a cada jugador y mantener conversaciones

### 2.1.3.2 Clubes

**Base de datos** Es la manera para gestionar todos los miembros del club, en la base de datos se incluyen perfiles para deportistas, que rellenando un formulario de inscripción se crearán sus perfiles.

**Horarios y deberes de equipo** Los clubes también pueden ver el calendario compartido por el entrenador y los jugadores.

**Facturas** Enviar facturas a todo el club y hacer un seguimiento de los pagos, pudiendo enviar recordatorios a los impagados.

---

**Padres y Jugadores** Este grupo de usuarios dispondrá de una app móvil donde tendrán acceso al horario de entrenamiento; mensajería con el entrenador; historial de asistencia; organizar desplazamientos a otros pares/jugadores y actualizar los detalles de contacto del jugador.

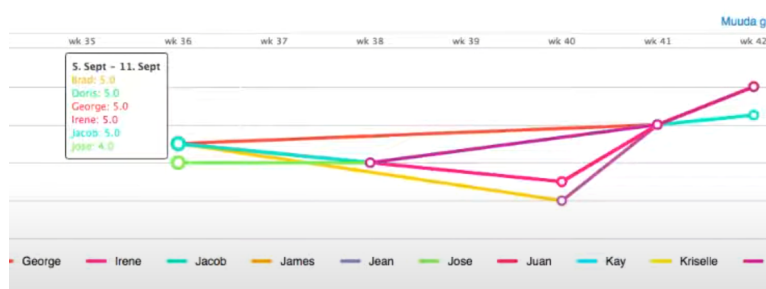
### 2.1.3.3 Precios

La aplicación ofrece precios mensuales o anuales (con un 30% de descuento sobre el mensual) dependiendo del tamaño del club. La diferencia de precios escala con una diferencia de 7€ entre los tramos que van en grupos de 25 jugadores.

### 2.1.3.4 Manejo de estadísticas e indicadores de rendimiento

Aunque esta aplicación se enfoca en la gestión y mejora del rendimiento deportivo de los jugadores, ofrece algunas funcionalidades que pueden ser útiles para el análisis de indicadores de rendimiento. Pudiendo servir para su análisis, no se menciona en su documentación ni aparece de forma clara dentro de las interfaces ninguna opción para crear estos indicadores.

La aplicación ofrece precios mensuales o anuales (con un 30% de descuento sobre el mensual) dependiendo del tamaño del club. La diferencia de precios escala con una diferencia de 7€ entre los tramos que van en grupos de 25 jugadores.



**Figura 2.4:** Sportlyzer Players App - Gráfico de los tiempos de finalización del entrenamiento

### 2.1.4 LongoMatch

Esta aplicación ofrece la retransmisión de partidos repetidos, con la posibilidad de dibujar sobre las imágenes de vídeo y colocar momentos importantes para ser analizados. Ofrece dos

planes diferentes:

- 15€/mes o 150€/año: Proyectos ilimitados, equipos, paneles y eventos. Análisis en directo con múltiples cámaras y administrador de base de datos.
- 55€/mes o 550€/año: Ofrece las posibilidades anteriores, junto con zoom durante el video, importar o exportar archivos XML y compatibilidad con otras herramientas.

#### 2.1.4.1 Manejo de estadísticas e indicadores de rendimiento

Aunque LongoMatch no está específicamente diseñado para la creación de KPIs personalizados, se pueden crear plantillas de etiquetado que correspondan a los KPIs específicos que deseemos rastrear. Por ejemplo, puedes tener etiquetas para "Controles de Balón Exitosos", "Intercepciones", etc. Además, los informes generados por LongoMatch pueden incluir gráficos y estadísticas basadas en los eventos etiquetados, permitiendo una evaluación detallada de los KPIs.

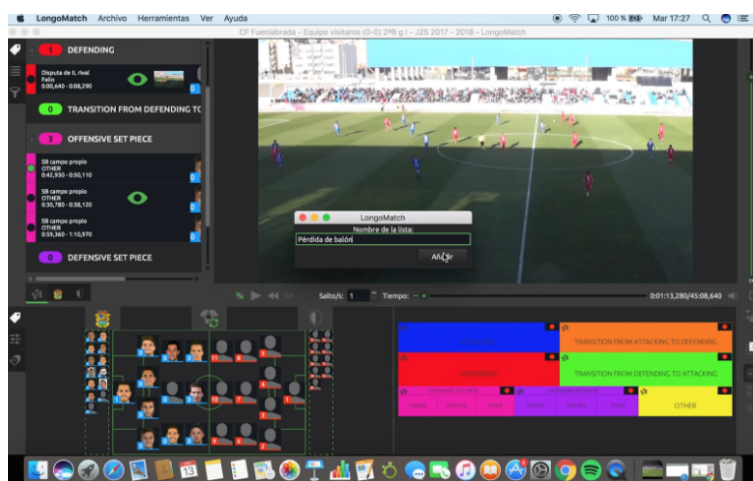


Figura 2.5: Longomatch - Creación de eventos en el vídeo

#### 2.1.5 Picco: Estadísticas y rendimiento

Esta app es completamente gratis (con opción a pago para quitar anuncios) contiene una multitud de opciones para la recogida y el análisis de los datos. Durante el partido, el analista puede apuntar cada dato estadístico que él considere. Esta pantalla de datos puede ser la más

útil a la hora de recoger datos durante el partido, pues solo hay que marcar cada registro a la vez que se va desarrollando el partido.

### 2.1.5.1 Manejo de estadísticas e indicadores de rendimiento

Picco es especialmente útil para la creación y seguimiento de KPIs debido a su capacidad de personalización y análisis detallado. Tiene la posibilidad de definir KPIs específicos, por ejemplo, puedes crear indicadores como "Tasa de éxito en pases", "Distancia recorrida por jugador", entre otros. Todos estos siendo recogidos y actualizados en tiempo real asegurando datos precisos y disponibilidad inmediata.



**Figura 2.6:** Picco - Interfaz durante el partido

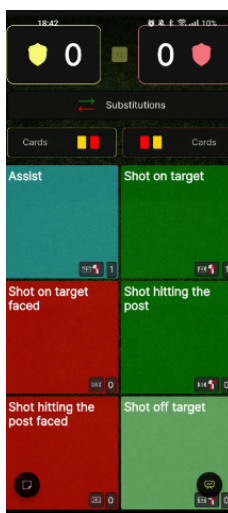
### 2.1.6 Teamtag

Esta aplicación es parecida a la Picco, teniendo funcionalidades más allá de los pocos paneles que contiene Picco, esta aplicación se enfoca en la gestión de un club, pudiendo crear eventos de partidos para ir apuntando las estadísticas a medida que transcurre el evento. Permite crear tus propios equipos y alineaciones para cuando un evento estadístico sea registrado, indicar que jugador/a lo ocasionó y en qué zona del campo ocurrió.



### 2.1.6.1 Manejo de estadísticas e indicadores de rendimiento

Al igual que la aplicación anterior, permite la definición de indicadores personalizados que se irán recogiendo en tiempo real.



**Figura 2.7:** Teamtag - Interfaz durante el partido

### 2.1.7 Conclusiones

Una vez analizadas las aplicaciones del mercado, podemos concluir la función de cada una. Bcoach:

- Planificación de entrenamientos con animaciones.
- Pizarra táctica.
- Estadísticas de partidos.
- Planes desde 2,99€/mes.

Sportlyzer Players App:

- Para entrenadores, clubes y jugadores.
- Horarios y tareas con calendario compartido.
- Asignación de deberes y seguimiento de asistencia.

- Evaluación y análisis de jugadores.
- Mensajería con jugadores.
- Planes desde 7€/mes por jugador.

LongoMatch:

- Retransmisión de partidos repetidos.
- Dibujar sobre video y análisis de momentos importantes.
- Planes desde 15€/mes.

Picco y Teamtag:

- Recogida y análisis de datos estadísticos.
- Gratis (con opción de pago para quitar anuncios).

Dependiendo de las necesidades, Bcoach o Sportlyzer podrían ser las más completas para la gestión de los equipos. Y si el problema es el presupuesto, Football Tactical Board o Picco podrían ser también buenas opciones

#### **2.1.7.1 Manejo de estadísticas e indicadores de rendimiento**

El manejo de estadísticas e indicadores de rendimiento es crucial para el análisis y la mejora continua del desempeño de los jugadores y equipos. Cada una de las aplicaciones analizadas ofrece diferentes niveles de soporte para estas funcionalidades:

##### **Bcoach**

- Ofrece categorías de tarea que se relacionan directamente con los indicadores de rendimiento.
  - Estadísticas que se modifican durante los partidos para analizar esas categorías.
-

**Football Tactical Board**

- Enfocada en la creación de diagramas tácticos y estrategias de juego.
- No diseñada específicamente para la gestión y análisis de estadísticas avanzadas como los KPIs.

**Sportlyzer Players App**

- Enfocada en la gestión y mejora del rendimiento deportivo.
- Funcionalidades útiles para el análisis de indicadores de rendimiento, aunque no ofrece una opción clara para la creación de KPIs.

**LongoMatch**

- Permite etiquetar eventos específicos durante los partidos y analizarlos posteriormente.
- Informes generados pueden incluir gráficos y estadísticas basadas en los eventos etiquetados.

**Picco**

- Especialmente útil para la creación y seguimiento de KPIs debido a su capacidad de personalización.
- Recogida y actualización de datos en tiempo real.

**Teamtag**

- Permite la definición de indicadores personalizados que se irán recogiendo en tiempo real.
- Gestión de equipos y alineaciones, registro de eventos y análisis de datos.

En resumen, todas las aplicaciones analizadas ofrecen herramientas útiles para el manejo de estadísticas e indicadores de rendimiento, aunque no todas ofrecen la opción de crear indicadores personalizados.

---

## 2.2 Tabla comparativa

Aplicación	Estadísticas predefinidas	Creación de nuevas estadísticas	En tiempo real	Individuales	Grupales	Informe generado	Seguimiento continuo	Toma de decisión	KPIs Personalizables
Bcoach	✓	✓	✓	✓	✓	✓	✓	✗	✓
Sportlyzer Players App	✓	✗	✗	✓	✓	✓	✓	✗	✗
LongoMatch	✗	✓	✓	✗	✓	✗	✗	✗	✗
Picco	✓	✓	✓	✗	✓	✗	✗	✗	✓
Teamtag	✓	✗	✓	✓	✓	✓	✓	✗	✓

**Tabla 2.1:** Comparativa de funcionalidades de aplicaciones

## 3 Objetivos

### 3.1 Objetivos generales

La plataforma va dirigida a todos los entrenadores/as de divisiones inferiores con el fin de agilizar y conseguir de la manera más eficaz y eficiente posible el correcto desarrollo de los jugadores.

Siguiendo las definiciones de la doctora Deborah Hoare en [Hoare (1999)], después de identificar y seleccionar talentos, es imperativo poner a disposición del deportista una infraestructura adecuada que les permita desarrollar su potencial al máximo. Esto implica no solo un buen programa de entrenamiento y competición bien estructurado, sino también el acceso al equipo necesario para su competencia.

La plataforma facilita este acceso proporcionando un sistema integrado que ayuda a los entrenadores a planificar y ejecutar programas de entrenamiento adaptados a las necesidades individuales o grupales de cada jugador. También permitirá un seguimiento detallado del progreso del equipo, garantizando maximizar sus resultados y que se identifiquen áreas de mejora comprendidas en una matriz DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades).

Se busca la optimización del proceso de evaluación, implementando herramientas automatizadas que permitan a los entrenadores realizar análisis y valoraciones en tiempo real durante los entrenamientos y partidos. Los datos podrán ser capturados mediante dispositivos que el usuario lleve consigo o a través de aplicaciones móviles. Los datos recogidos se utilizarán para elaborar análisis históricos de los jugadores para predecir tendencias. Esto ayudará a los entrenadores a tomar decisiones correctas sobre la carga de entrenamiento y la rotación del equipo para optimizar el rendimiento.

En resumen, la plataforma busca ser una herramienta esencial en el arsenal de cualquier entrenador que trabaje con jóvenes jugadores, proporcionándoles las capacidades necesarias

para fomentar el desarrollo de sus capacidades en el fútbol y otros deportes. Con esta herramienta, los entrenadores pueden asegurarse de que cada jugador no solo alcanza su potencial técnico y táctico, sino que también recibe el soporte físico y psicológico necesario para su desarrollo integral.

## **3.2 Objetivos específicos**

### **1. Identificación y desarrollo del talento:**

- Descripción de la infraestructura adecuada y programas de entrenamiento.

### **2. Acceso a herramientas y equipo necesario:**

- Planificación y ejecución de programas de entrenamiento adaptados a las necesidades individuales y grupales.

### **3. Seguimiento y evaluación del progreso:**

- Uso de herramientas automatizadas para el análisis en tiempo real durante entrenamientos y partidos.

### **4. Análisis histórico y predicción de tendencias:**

- Utilización de datos históricos para optimizar la carga de entrenamiento y la rotación del equipo.
-

## 4 Metodología

Una metodología de desarrollo de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información [(Maida, 2015)].

### 4.1 Metodología de realización

Para la realización de este trabajo, se ha decidido utilizar la metodología SCRUM, basada en las siguientes aspectos [Agilest (2024)] :

- **Adaptabilidad:** Los cambios pueden ser implementados en un proyecto en curso. Con Scrum puede variar lo que se debe hacer en un proyecto, pero el tiempo y coste son constantes.
- **Priorización de Tareas:** Las tareas se priorizan por orden de importancia, lo que generalmente significa que las tareas que se completan primero probablemente afectarán más al trabajo.
- **Sprints:** Los sprints son períodos delimitados durante el cual el equipo trabaja para completar una cantidad de trabajo [Rrehkof (2024)]. En nuestro caso, la planificación de los sprints se realizaba semanalmente, realizando un seguimiento a lo largo de la semana y presentando el incremento del trabajo al principio de la nueva reunión.

### 4.2 Elección de las herramientas

A continuación, detallamos las razones específicas para la elección de las siguientes herramientas como componentes clave del desarrollo.

### 4.2.1 Github



**Figura 4.1:** Logo de Github

Es una plataforma de online de desarrollo de software que nos facilitará el seguimiento de los cambios, pudiendo retroceder entre las nuevas versiones gracias a la tecnología en la que se basa Github, Git [Walsh (2024)].

Git es un software de control de versiones gratis y de código abierto, funciona como un rastreador de contenido permitiendo a los desarrolladores revertir y regresar a una versión anterior de su código. Con Git, cada desarrollador tiene una copia completa del repositorio, lo que permite trabajar sin conexión y realizar operaciones localmente. Además, Git soporta la creación de ramas para trabajar en nuevas características o correcciones de errores de manera independiente, que luego pueden fusionarse en la rama principal. Su uso de algoritmos de hashing garantiza la integridad de los datos, y su rendimiento superior permite manejar grandes cantidades de datos y cambios de manera eficiente [Montiel (2024)].

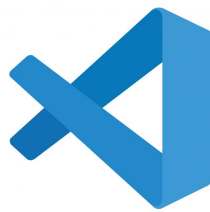
Por otro lado, Github ofrece un servicio para la gestión de proyectos, pudiendo definir listas de tareas dentro de un proyecto. Las listas de tareas que hemos considerado son las siguientes.

- **Pendientes:** En esta lista se encuentran las tareas que se que no se han empezado a trabajar.
  - **Proceso:** En esta lista se encuentran las tareas en las que se está trabajando en ese momento.
  - **Bloqueadas:** En esta lista se encuentran las tareas que no pueden ser completadas debido a que dependen de otra que no se ha completado todavía.
  - **Finalizadas:** En esta lista se encuentran las tareas que ya han sido completadas.
-



Los documentos compartidos y la planificación de reuniones se realiza mediante una aplicación de mensajería instantánea.

#### 4.2.2 Visual Studio Code



**Figura 4.2:** Logo de Visual Studio Code

Un editor de código fuente desarrollado por Microsoft. Es un software libre y multiplataforma disponible para Windows, Linux y macOS. Tiene una buena integración con Git, incluyendo los comandos principales para las acciones en un repositorio remoto. Además, dispone de un gran número de extensiones para escribir y ejecutar código en cualquier lenguaje de programación [Flores (2024)]. Por todo esto hemos decidido que este entorno de programación es el más adecuado para llevar a cabo el trabajo.

#### 4.2.3 Draw.io



**Figura 4.3:** Logo de Draw.io

Es una herramienta gratuita de diagramas en línea que permite a los usuarios crear una variedad de diagramas, gráficos y diagramas de flujo. Gracias a su interfaz intuitiva y su amplia gama de plantillas y figuras podremos dibujar cualquier diagrama para explicar cualquier estructura.

---

#### 4.2.4 Moqups



**Figura 4.4:** Logo de Moqups

Esta herramienta online nos permitirá el diseño de prototipos para crear las interfaces de nuestra aplicación. Está diseñada para facilitar la colaboración en equipo y la visualización rápida de ideas de diseño. Algunas características clave para que podamos usarla incluyen su interfaz intuitiva y su biblioteca de componentes y plantillas, con componentes prediseñados, iconos y plantillas que podemos utilizar para acelerar el proceso de diseño.

#### 4.2.5 Vue.js



**Figura 4.5:** Logo de Vue.js

Es un framework progresivo para construir interfaces de usuario. Está diseñado desde cero para ser utilizado incrementalmente. La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes [VueJS (2024b)]. Comparándolo con otros frameworks, podemos concluir que Vue.js es el más intuitivo para un usuario que empieza desde cero con los frameworks de desarrollo de interfaces, como Angular o React [VueJS (2024a)].

---

#### 4.2.5.1 Dependencias

**Iconify** Iconify es un proyecto de código abierto que proporciona una forma sencilla de utilizar iconos dentro de las aplicaciones Vue. El paquete que ofrece, **@iconify/vue**, permite agregar y personalizar rápidamente iconos en nuestro proyecto. Para instalarlo, ejecutaremos el comando de instalación:

Código 4.1: Instalación del paquete iconify

```
1 npm install --save-dev @iconify/vue
```

Para agregar los iconos que ofrece Iconify debemos importar el componente Icon del paquete en nuestros componentes de Vue. Lo haremos con el siguiente fragmento de código:

Código 4.2: Importar el componente Icon en Vue

```
1 <script setup>
2   import { Icon } from '@iconify/vue'
3 </script>
```

Para agregar iconos debemos buscar los que queramos en la página web de Iconify en [(Iconify, 2024)]. Una vez hayamos elegido el estilo del icono, podemos copiar el código del componente en la página web, por ejemplo:

Código 4.3: Código para agregar un icono a la página

```
1 <template>
2   <Icon icon="streamline:graph-dot-solid" />
3 </template>
```

Dando como resultado:



**Figura 4.6:** Icono de gráfico importado desde la biblioteca de Iconify

**ViteJS** ViteJS es una herramienta de construcción de frontend que ofrece una experiencia de desarrollo moderna, rápida y eficiente para proyectos web. Gracias a su recarga instantánea

podemos mejorar considerablemente la productividad a medida que desarrollamos el proyecto. Nuestro proyecto principal de Vue lo crearemos con esta herramienta:

Código 4.4: Instalación del paquete iconify

```
1  npm init @vitejs/app
```

**ChartJS** Una biblioteca de JavaScript que permite crear gráficos interactivos y visualizaciones de datos en aplicaciones web. Es fácil de usar y compatible con múltiples tipos de gráficos, como barras, líneas, etc. Gracias a su sencillez para integrarla en nuestro proyecto de Vue podremos usarla con total normalidad.

Para usarla, crearemos un componente para la creación de una gráfica que podremos usar en todas las vistas que queramos:

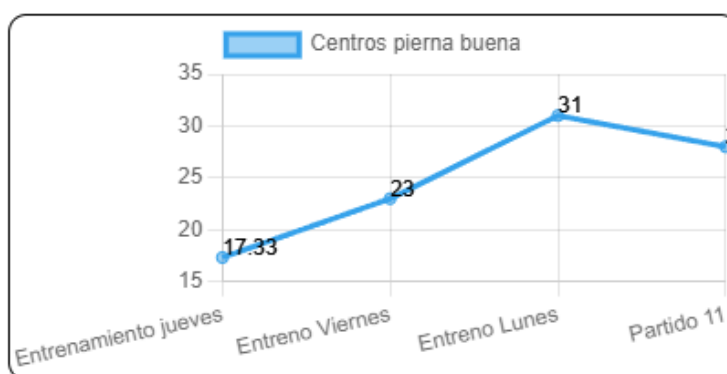
Código 4.5: Código para agregar un icono a la página

```
1 <template>
2   <div class="chart-container">
3     <canvas :id="chartId"></canvas>
4   </div>
5 </template>
6
7
8 <script>
9 import Chart from 'chart.js/auto';
10
11 export default {
12   name: 'KpiChart',
13   props: {
14     chartData: {
15       type: Array,
16       required: true
17     },
18     chartId: {
19       type: String,
20       required: true
21     },
22     chartName: {
23       type: String,
24       required: true
25     }
26   }
27 }
```

```
26  },
27  mounted() {
28
29    try{
30      const ctx = document.getElementById(this.chartId);
31      if (this.chartData && this.chartData.length > 0) {
32        new Chart(ctx, {
33          type: 'line',
34          data: {
35            labels: this.chartData.map(row => row.time),
36            datasets: [
37              {
38                label: this.chartName,
39                data: this.chartData.map(row => row.score)
40              }
41            ]
42          },
43          options: {
44            animation: {
45              duration: 1500, // milisegundos
46              easing: 'easeOutCubic', // Tipo de animación
47              onProgress(animation) {
48                const chartInstance = animation.chart;
49                const ctx = chartInstance.ctx;
50                const dataset = chartInstance.data.datasets[0];
51                const meta = chartInstance.getDatasetMeta(0);
52                meta.data.forEach((bar, index) => {
53                  const data = dataset.data[index];
54                  ctx.fillText(data, bar.x, bar.y - 5);
55                });
56              },
57              onComplete(animation) {
58                const chartInstance = animation.chart;
59                const ctx = chartInstance.ctx;
60                const dataset = chartInstance.data.datasets[0];
61                const meta = chartInstance.getDatasetMeta(0);
62                meta.data.forEach((bar, index) => {
63                  const data = dataset.data[index];
64                  ctx.fillText(data, bar.x, bar.y - 5);
65                });
66              }
67            },
68          }
```

```
69     });  
70   } else {  
71     console.error('No hay datos para la grafica');  
72   }  
73 }  
74 catch(error){  
75   console.error("Error:",error)  
76 }  
77 }  
78 };  
79 </script>
```

Como resultado de una de nuestras gráficas:



**Figura 4.7:** Gráfico generado con ChartJS

**Vue-router** La biblioteca oficial de enrutamiento para Vue.js, diseñada para crear aplicaciones de una sola página (SPA). Proporciona una forma estructurada y eficiente de manejar las rutas y la navegación en una aplicación Vue.js [Imagina (2024)]. Crear rutas para mostrar diferentes interfaces resultará muy sencillo añadiendo las rutas como la siguiente:

Código 4.6: Código para agregar un icono a la página

```
1 routes: [  
2   { path: '/user/:id', component: User }  
3 ]
```

**Vuex** Sirve como un almacén centralizado para todos los componentes de una aplicación, con reglas que garantizan que el estado solo pueda ser mutado de manera predecible. Gracias

a esto podremos crear variables globales que compartirán todos los componentes del proyecto de Vue.

### 4.2.6 MySQL



**Figura 4.8:** Logo de MySQL

Es el sistema de gestión de bases de datos relacional más extendido en la actualidad de código abierto que utiliza el lenguaje SQL (Structured Query Language) para acceder y gestionar datos almacenados en tablas. Siendo la elección popular para gestionar datos en entornos tanto pequeños como grandes, usaremos esta herramienta para la gestión de nuestra base de datos.

### 4.2.7 Framework DAO

Cambiar en `database.js` `require("mysql")` por `mysql2`, compatible con el método `auth_caching_sha2_password` el que viene en MySQL Server 8.4.





# 5 Desarrollo

## 5.1 Requisitos

En esta sección realizaremos una toma de los requisitos generales del sistema, seguidos de requisitos específicos.

### 5.1.1 Funcionales

Los requisitos funcionales son especificaciones que definen las funciones o características que un sistema debe poseer. Estos describen el comportamiento del sistema bajo ciertas condiciones y definen lo que el sistema debe hacer.

#### 5.1.1.1 Generales

- **R1:** Cada uno de los usuarios tendrá acceso a una información específica, pudiendo acceder solamente a los datos y funciones a los que tenga permiso ver.
- **R2:** Cada uno de los usuarios tendrá acceso a una información específica, pudiendo acceder solamente a los datos y funciones a los que tenga permiso ver.
- **R3:** Se podrán generar informes de rendimiento para los jugadores y el equipo técnico.
- **R4:** Cada entrenador de una organización podrá acceder a la información de todos los actores que forman parte de la organización.
- **R5:** Los entrenadores tendrán acceso a todas las sesiones de entrenamiento y a estadísticas individuales y grupales de cada jugador.

#### 5.1.1.2 Entrenamientos

- **R6:** Los usuarios tendrán un apartado para acceder a un calendario, que mostrará cada sesión.
- **R7:** Para cada sesión, se mostrará específicamente el nombre de ésta, el día y la hora de inicio y fin de la sesión.

#### 5.1.1.3 Seguimiento y Evaluación del Rendimiento

- **R8:** Evaluar el rendimiento de los jugadores mediante la recolección de estadísticas durante los entrenamientos y partidos.
- **R9:** Mostrar el historial de puntuaciones dadas por el entrenador por cada intervalo de tiempo definido.
- **R9:** Elaborar gráficos de las puntuaciones de los KPIs para todo el equipo para que se pueda apreciar la evolución del equipo en un conjunto.
- **R10:** Elaborar gráficos de las puntuaciones de los KPIs para cada jugador, pudiendo apreciarse la evolución individual.
- **R11:** Desarrollar un algoritmo estadístico para generar una matriz DAFO (debilidades, Amenazas, Fortalezas y Debilidades) que permitan al entrenador analizar de una manera más precisa las necesidades de sus jugadores.

#### 5.1.1.4 Equipo

- **R12:** El entrenador podrá ver toda la información detallada del equipo
- **R13:** El entrenador podrá ver toda la información detallada de cada jugador individualmente

#### 5.1.1.5 Intervalos de tiempo

- **R14:** Los usuarios podrán acceder a un histórico de los datos en un intervalo de tiempo correspondiente a una sesión.
-

- **R15:** Los usuarios podrán acceder a un histórico de los datos en un intervalo de tiempo correspondiente a un mes.
- **R16:** Los usuarios podrán acceder a un histórico de los datos en un intervalo de tiempo correspondiente a un trimestre.

#### 5.1.1.6 Actores

- **R17:** Los entrenadores podrán acceder a la información de los actores que componen la organización y sean jugadores.
- **R18:** Cada ventana del actor tendrá su información correspondiente al seguimiento de dicho actor.

### 5.1.2 No Funcionales

Los requisitos no funcionales describen cómo el sistema debe comportarse. Estos incluyen aspectos como el rendimiento, la seguridad, la usabilidad, la escalabilidad y la disponibilidad del sistema.

#### 5.1.2.1 Rendimiento

- **Capacidad de carga:** La plataforma debe ser capaz de manejar una cantidad considerable de usuarios simultáneos sin afectar al rendimiento.
- **Tiempo de Respuesta:** La carga de los datos debe completarse en un tiempo rápido.
- **Optimización:** Las consultas a la base de datos deben de estar optimizadas para minimizar el tiempo de espera.

#### 5.1.2.2 Seguridad

- **Autenticación:** Las operaciones de carga de datos no se realizarán hasta que el usuario no se haya autenticado correctamente.
  - **Protección de datos:** Los datos personales de los usuarios deben ser almacenados y procesados conforme a las normativas de protección de datos.
-

### 5.1.2.3 Usabilidad

- **Interfaz intuitiva:** La interfaz de usuario debe ser intuitiva y fácil de usar, permitiendo a los usuarios utilizar la plataforma desde cero sin tener que formarse para ello.
- **Documentación:** Los usuarios deben poder acceder a una documentación clara para usar la plataforma.

### 5.1.2.4 Escalabilidad

- **Escalabilidad horizontal:** La plataforma debe poder escalar horizontalmente (añadiendo mas servidores) para reducir la carga de éstos y mejorar la experiencia del usuario.
- **Escalabilidad vertical:** La plataforma debe poder escalar verticalmente (mejorando los servidores) para manejar el aumento de usuarios y la carga de trabajo.

### 5.1.2.5 Compatibilidad

- **Multiplataforma:** La plataforma ser accesible en distintos sistemas operativos, así como desde diferentes tipos de dispositivo (móvil u ordenadores).

### 5.1.2.6 Fiabilidad

- **Tolerancia a fallos:** El sistema debe manejar los errores de una manera adecuada, proporcionando mensajes de error claros para su rápida resolución.
- **Consistencia de datos:** Los datos deben mantenerse consistentes y libres de errores.

## 5.2 Análisis funcional del sistema

### 5.2.1 Historias de usuario

Para la gestión ágil de este proyecto, comenzaremos a describir historias de usuario. Siguiendo una estructura sistemática para reflejar las necesidades de los usuarios que se traducen en requisitos funcionales para el sistema que influirán en el diseño y desarrollo de la plataforma.

---

**Añadir KPIs a la pantalla de estadísticas**

**Como** *Entrenador/a*,

**Quiero** Poder añadir y quitar KPIs de la pantalla de estadísticas según sea necesario,  
**Para que** Enfocarme en medir y visualizar solo los KPIs específicos que necesito, facilitando la toma de decisiones rápidas y efectivas para mejorar el rendimiento del equipo.

**Descripción** La pantalla para evaluar los indicadores de rendimiento debe mostrar a la vez tantos como el entrenador necesite al mismo tiempo, así como eliminar de la vista aquellos que ya no son relevantes en ese momento.

**Criterios de Aceptación**

- El entrenador debe poder acceder a una interfaz de usuario para ver los KPIs.
- La aplicación debe recomendar algunos KPIs predefinidos en caso de que haya menos de 5 en la pantalla.
- El entrenador debe poder borrar todos los indicadores de una sola vez y limpiar la vista.

**Creación de nuevos KPIs**

**Como** *Entrenador/a*,

**Quiero** Poder crear y eliminar nuevos indicadores de rendimiento con los datos almacenados en la base de datos,  
**Para que** Explotar todos los datos recopilados de una manera eficiente para ver las fortalezas y debilidades del equipo.

**Descripción** Después de cada partido o entrenamiento, los datos comunicados por el entrenador serán almacenados en una base de datos. A partir de esos datos y con una fórmula general, el entrenador debe poder elegir dos datos y crear un nuevo indicador de rendimiento para ser evaluado.

### Criterios de Aceptación

- El entrenador debe poder acceder a una interfaz de usuario para crear los KPIs.
- En la interfaz se debe poder definir el nuevo nombre del indicador, descripción y los datos que involucra.
- La aplicación no debe permitir nombres repetidos de indicadores.
- La aplicación no debe permitir que dos indicadores involucren los mismos datos de la misma forma.
- El indicador debe poder asignarse a jugadores o equipos.

#### Edición de KPIs

**Como** *Entrenador/a*,

**Quiero** Poder editar los indicadores de rendimiento ya sea por su nombre o por sus parámetros,

**Para que** Cambiar los indicadores según el progreso que necesiten mis entrenamientos.

**Descripción** Una vez haya definido al menos un KPI, el entrenador debe ser capaz de editar el nombre, atributos y descripción del indicador. Este cambio puede ser debido ya sea por el cambio de opinión del entrenador del entrenador, o por una necesidad de utilizar ese nombre o parámetros para otra perspectiva en las sesiones de entrenamiento.

### Criterios de Aceptación

- El entrenador debe poder seleccionar un KPI existente en la interfaz para editarlo.
- La interfaz de edición debe poder permitir editar el nombre, parámetros del KPI o jugadores y equipo a los cuales esté asignado este indicador.
- Los cambios deben de ser en cascada, reflejando la modificación en todas las vistas en las que el KPI esté implicado.
- Se debe permitir cancelar la edición si el usuario cambia de idea.

**Ver estadísticas**

**Como** *Jugador/a*,

**Quiero** Poder ver un listado de todas mis estadísticas y mis indicadores de rendimiento,

**Para que** Poder fijarme en mis puntos débiles e intentar mejorarlos según me dicte el entrenador.

**Descripción** Cada jugador debe poder acceder a una interfaz donde se muestren todos los indicadores que el entrenador ha asociado a ese jugador en concreto, también debería ver una gráfica sobre el progreso y la evolución de sus indicadores. Así se podría fomentar la auto-crítica por parte del jugador.

**Criterios de Aceptación**

- Los jugadores deben poder acceder a un menú propio donde se muestren sus datos.
- El menú debe incluir gráficas con marcas temporales.
- Se debe poder filtrar por rango de fechas, seleccionar estadísticas para que solo se muestren esas y por tipos de estadísticas.

**Ver progreso de los jugadores**

**Como** *Entrenador/a*,

**Quiero** Poder generar un informe gráfico que muestre la evolución de los KPIs asociados a jugadores,

**Para que** Poder evaluar a mis jugadores y darles las indicaciones adecuadas para maximizar el margen de mejora de mi equipo.

## 5.3 Implementación

**Descripción** El entrenador es el primer responsable del equipo y de él depende que los jugadores mejoren y exploten sus capacidades lo máximo posible. Es por ello que debe de ser

capaz de generar en la plataforma un informe, ya sea individual de cada jugador o grupal, sobre la evolución de las estadísticas de éstos.

### Criterios de Aceptación

- Los entrenadores deben poder seleccionar uno o varios jugadores junto con uno o varios KPIs para generar el informe.
- El menú debe incluir gráficas con marcas temporales.
- La plataforma debe de ser capaz de detectar una mejora o un empeoramiento de las habilidades del jugador.

#### Historia de Usuario: Análisis y Reportes Detallados

**Como** *Entrenador/a*,

**Quiero** Generar reportes detallados de rendimiento por jugador y por equipo,

**Para que** Tener un seguimiento preciso del progreso a lo largo del tiempo.

**Descripción** Más allá de un visualizado de estadísticas y gráficas, el entrenador debe se capaz de generar informes completos sobre el desarrollo de los jugadores.

### Criterios de Aceptación

- Para generar los informes, se debe poder seleccionar métricas específicas.
- Los informes pueden ser exportados en formatos PDF, Excel o CSV.
- Los informes deben contener gráficas de tendencias y otras comparativas.

#### Historia de Usuario: Acceso a roles de usuario

**Como** *Usuario*,

**Quiero** Poder acceder a los datos a los que tenga acceso,

**Para que** Realizar el seguimiento siendo o el mismo jugador o el tutor legal de éste.



**Descripción** Los padres y tutores también tienen derecho a ver el seguimiento de sus hijos/as, por ello es considerable añadir roles de usuario para que cada persona responsable pueda ver sus propios datos.

#### Criterios de Aceptación

- La plataforma debe tener definidos claramente los roles de "Entrenador", "Jugador", "Padre/Tutor" y "Administrador".
- Cada rol debe poder acceder únicamente a los datos que le corresponden según las definiciones de permisos.
- Debe existir un sistema de autenticación para los usuarios.
- La plataforma debe cumplir con las normativas locales e internacionales de protección de datos

Historia de Usuario: Intervalos Estadísticos para el menú

**Como** *Entrenador/a*,

**Quiero** Poder alternar el intervalo de tiempo en el que se mide la estadística,

**Para que** Tener un mejor control de la evolución de mis equipos durante el periodo de tiempo que estime útil.

**Descripción** Para que el seguimiento de los resultados sea lo mas eficaz y eficiente posible, el entrenador debe poder alternar el intervalo de tiempo en el que se miden los resultados para su propio interés. Si quiere ver resultados a corto plazo, podrá ver la evolución partido a partido, si quiere resultados a largo plazo, oirá verla mensual o trimestralmente.

#### Criterios de Aceptación

- La plataforma tendrá un menú superior para alternar el intervalo.
- Cuando se alterna la opción, todas las vistas serán actualizadas para mostrar la evolución de los jugadores dentro del intervalo seleccionado.

### **5.3.1 Matriz DAFO**

#### **5.3.1.1 Promedio GPA**

GPA es un término que se usa mucho en niveles académicos.

#### **5.3.1.2 Desviación estándar**

#### **5.3.1.3 Pendiente de tendencia**

#### **5.3.1.4 Cambios positivos y negativos entre los intervalos de tiempo**

## **6 Conclusiones**



# Bibliografía

- Agilest. (2024). *Why does scrum work?* <https://www.agilest.org/scrum/why-does-scrum-work/>. (Accessed: 2024-06-25)
- Bcoach. (2020a). *Bcoach: Tutorial categorías y estadísticas*. <https://www.youtube.com/watch?v=gxIszL1BK4>. (Accessed: 2024-03-26)
- Bcoach. (2020b). *Bcoach: Tutorial sobre el resumen de las estadísticas*. <https://www.youtube.com/watch?v=AzIY-7jVVTw>. (Accessed: 2024-03-26)
- Bcoach. (2023). *Apps para entrenadores de fútbol [ios y android]*. <https://bcoach.app/aplicaciones-para-entrenadores-de-futbol/>. (Accessed: 2024-03-24)
- Flores, F. (2024). *Qué es visual studio code y qué ventajas ofrece*. <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>. (Accessed: 2024-06-25)
- Guardian. (2016). *Sir alex ferguson on the importance of communication | guardian live*. <https://www.youtube.com/watch?v=F2NQn4DU4fY>. (Accessed: 2024-05-01)
- Hoare, D. (1999). *Seminario internacional de detección de talentos deportivo*. <https://isidoro-hornillos.webnode.es/products/deteccion-y-seleccion-de-talentos-deportivos-conferencia-/>. (Accessed: 2024-05-15)
- Iconify. (2024). *Biblioteca de iconos, iconify*. <https://icon-sets.iconify.design/>. (Accessed: 2024-06-21)
- Imagina. (2024). *¿qué es vue router y por qué es necesario?* <https://imaginaformacion.com/tutoriales/que-es-vue-router-y-por-que-es-necesario>. (Accessed: 2024-06-25)

- Komarnicki, J. (2023). *Vue 3 crash course / project from scratch*. <https://www.youtube.com/watch?v=KTFH4P8unUQ>. (Accessed: 2024-05-16)
- LongoMatch. (2017a). *Longomatch - start analyzing your games today*. <https://www.youtube.com/watch?v=WCjw7uhEctA>. (Accessed: 2024-03-25)
- LongoMatch. (2017b). *Tutorial longomatch - funcionamiento general*. <https://www.youtube.com/watch?v=WCjw7uhEctA>. (Accessed: 2024-03-25)
- Maida, J., Esteban Gabriel ; Pacienza. (2015). *Metodologías de desarrollo de software*. Cátedra: Seminario de Sistemas.
- Montiel, O. (2024). *La guía para principiantes de git y github*. <https://www.freecodecamp.org/espanol/news/guia-para-principiantes-de-git-y-github/>. (Accessed: 2024-06-25)
- Parmenter, D. (2010). *Key performance indicators: Developing, implementing, and using winning kpis*. Wiley.
- Rrehkof, M. (2024). *Scrum sprints*. <https://www.atlassian.com/agile/scrum/sprints>. (Accessed: 2024-06-25)
- Sportlyzer. (2016). *Manage your sports team with less time: Sportlyzer, player development and team management software*. <https://www.sportlyzer.com/en/features/player-app>. (Accessed: 2024-03-24)
- VueJS. (2024a). *Comparison with other frameworks*. <https://v2.vuejs.org/v2/guide/comparison.html>. (Accessed: 2024-06-25)
- VueJS. (2024b). *¿qué es vue.js?* <https://es.vuejs.org/v2/guide/#!/C2%BFQue-es-Vue-js>. (Accessed: 2024-06-25)
- Walsh, D. (2024). *Qué es github y para qué sirve*. <https://blog.hubspot.es/website/que-es-github#que-es>. (Accessed: 2024-06-25)
-