



Escuela
Politécnica
Superior

Plataforma de análisis 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 análisis 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, Julio 2024

Índice general

1	Introducción	1
2	Marco Teórico	3
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	4
2.1.2.1	Manejo de estadísticas e indicadores de rendimiento	5
2.1.3	Sportlyzer Players App	5
2.1.3.1	Entrenadores	5
2.1.3.2	Clubes	6
2.1.3.3	Precios	7
2.1.3.4	Manejo de estadísticas e indicadores de rendimiento	7
2.1.4	LongoMatch	7
2.1.4.1	Manejo de estadísticas e indicadores de rendimiento	8
2.1.5	Picco: Estadísticas y rendimiento	9
2.1.5.1	Manejo de estadísticas e indicadores de rendimiento	9
2.1.6	Teamtag	9
2.1.6.1	Manejo de estadísticas e indicadores de rendimiento	10
2.1.7	Conclusiones	10
2.1.7.1	Manejo de estadísticas e indicadores de rendimiento	11
2.2	Tabla comparativa	13

3	Objetivos	15
3.1	Objetivos generales	15
3.2	Objetivos específicos	16
4	Metodología	17
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
5	Desarrollo	27
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
5.2.2	Análisis de entidades de negocio	35
5.2.3	Análisis de interfaces	36
5.2.3.1	Inicio de sesión	36
5.2.3.2	Selección de organización	36
5.2.3.3	Pantalla principal de la organización	37
5.2.3.4	Detalles del Actor	37
5.2.3.5	Estadísticas	37
5.2.3.6	Calendario	37
5.2.3.7	Menú	38
5.2.4	Análisis de servicios	38
5.2.4.1	Autenticación	38
5.2.4.2	Organizaciones	38
5.2.4.3	Actores	38
5.2.4.4	Estadísticas	39
5.2.4.5	Calendario	39
5.3	Diseño del sistema	39
5.3.1	Arquitectura general	39
5.3.2	Diseño de interfaces de usuario	40
5.3.2.1	Inicio de sesión	40
5.3.2.2	Selección de organización	41
5.3.2.3	Página principal	41
5.3.2.4	Equipo	42
5.3.2.5	Jugador	42
5.3.2.6	Estadísticas	43
5.3.2.7	Calendario	44
5.3.3	Diseño lógico de la base de datos	44
5.3.3.1	User	45

5.3.3.2	Organization	45
5.3.3.3	User has organization	45
5.3.3.4	Actor	46
5.3.3.5	Session	46
5.3.3.6	Actor has actortype	46
5.3.3.7	Kpi	47
5.3.3.8	Exercise	47
5.3.3.9	Exercise has session has actor has kpi	47
5.3.4	Diseño de servicios	48
5.4	Implementación	51
5.4.1	Arquitectura técnica	51
5.4.2	Implementación de interfaces de usuario	52
5.4.2.1	Inicio de sesión	52
5.4.2.2	Selección de organización	53
5.4.2.3	Página principal	53
5.4.2.4	Equipo	54
5.4.2.5	Jugador	54
5.4.2.6	Estadísticas	55
5.4.2.7	Calendario	55
5.4.3	Esquema físico de la base de datos	55
5.4.4	Implementación de servicios	55
5.4.4.1	Matriz DAFO	55
5.4.4.2	Promedio GPA	55
5.4.4.3	Desviación estándar	55
5.4.4.4	Pendiente de tendencia	55
5.4.4.5	Cambios positivos y negativos entre los intervalos de tiempo	55
5.5	Pruebas y validación	55
6	Conclusiones	57
	Bibliografía	59

Índice de figuras

2.1	Bcoach - Estadísticas de Jugadores	4
2.2	Bcoach - Crear categoría de tarea	4
2.3	Football Tactical Board - Pantalla del entrenador	5
2.4	Sportlyzer Players App - Gráfico de los tiempos de finalización del entrenamiento	8
2.5	Longomatch - Creación de eventos en el vídeo	8
2.6	Picco - Interfaz durante el partido	9
2.7	Teamtag - Interfaz durante el partido	10
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
5.1	Mapa conceptual de Sergio Yáñez Cánovas y Daniel González Luque	35
5.2	Modelo Entidad-Relación de Sergio Yáñez Cánovas y Daniel González Luque	36
5.3	Arquitectura general del sistema de Sergio Yáñez Cánovas	39
5.4	Mockup de inicio de sesión	40
5.5	Mockup de selección de equipo	41
5.6	Mockup de la página principal	41
5.7	Mockup de la página de equipo	42
5.8	Mockup de la página de jugador	42

5.9	Mockup de la página de estadísticas	43
5.10	Mockup de la página de calendario	44
5.11	Modelo lógico de Sergio Yáñez Cánovas y Daniel González Luque	45
5.12	Relación entre las tablas user, organization y user has organization	46
5.13	Relación entre las tablas user, actor, session, actor has actortype, kpi, exercise, exercise has session has actor has kpi	48
5.14	Arquitectura técnica del sistema	51
5.15	Implentación de la interfaz de inicio de sesión	52
5.16	Implentación de la interfaz de selección de organización	53
5.17	Implentación de la interfaz de selección de organización	53
5.18	Implentación de la interfaz de equipo	54
5.19	Implentación de la interfaz de equipo	54
5.20	Implentación de la interfaz de equipo	54

Índice de tablas

2.1	Comparativa de funcionalidades de aplicaciones	13
-----	--	----

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
5.1	tipo dato user	48
5.2	tipo dato organización	49
5.3	tipo dato actor	49
5.4	tipo dato sesión	49
5.5	tipo dato kpi	50
5.6	tipo dato exercise	50

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.

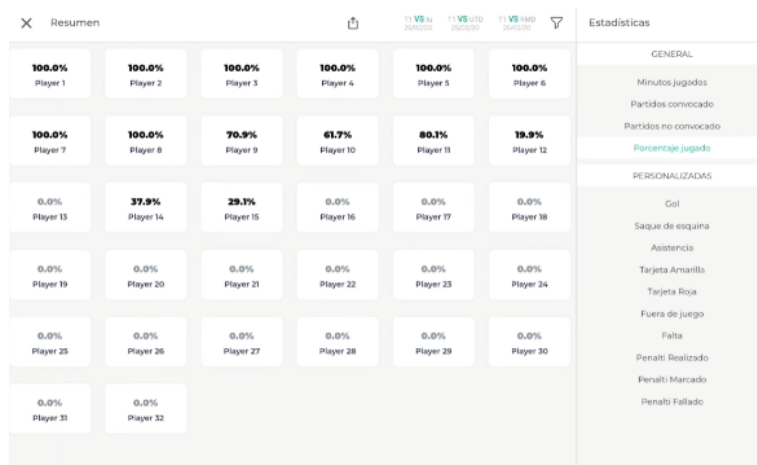


Figura 2.1: Bcoach - Estadísticas de Jugadores

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.

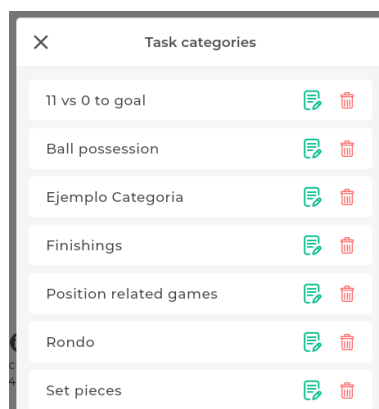


Figura 2.2: Bcoach - Crear categoría de tarea

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.

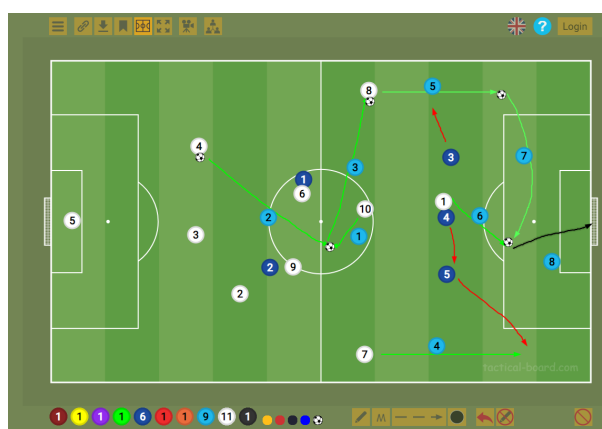


Figura 2.3: Football Tactical Board - Pantalla del entrenador

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.

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:

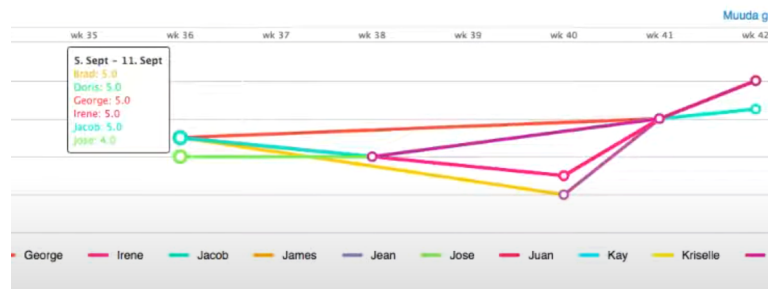


Figura 2.4: Sportlyzer Players App - Gráfico de los tiempos de finalización del entrenamiento

- 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.

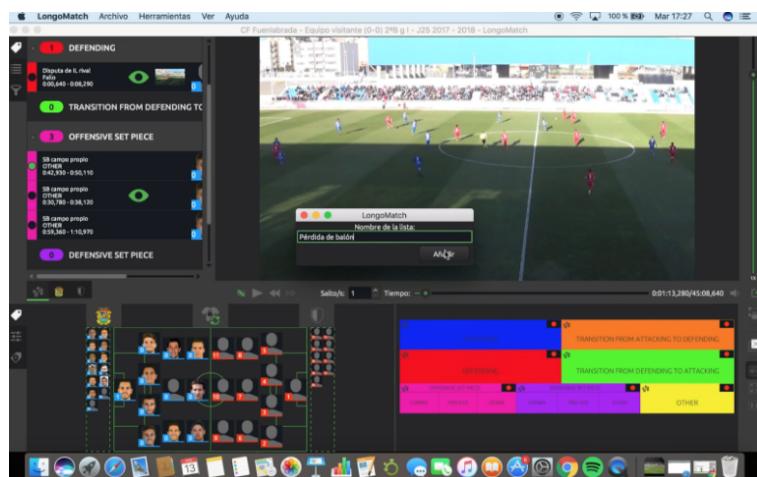


Figura 2.5: Longomatch - Creación de eventos en el vídeo

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. Per-

mite 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.

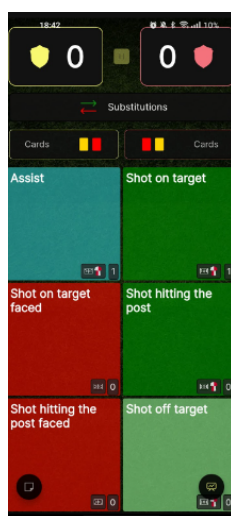


Figura 2.7: Teamtag - Interfaz durante el partido

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

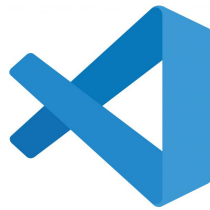


Figura 4.2: Logo de Visual Studio Code

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       }
71     }
72   }
73 }
```

```
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:

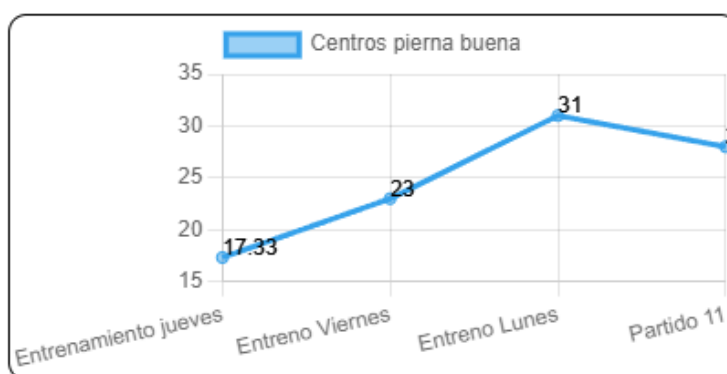


Figura 4.7: Gráfico generado con ChartJS

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

Esta herramienta será proporcionada por el tutor Diego Marcos Jorquera. Se trata de un framework que gestiona los accesos a la base de datos, haciendo las peticiones desde la aplicación de Vue.js mucho mas sencillas. También ha sido implementado en los trabajos de [Cánovas (2020)] y [Luque (2020)].

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.

Inicio de sesión**Como** *Usuario*,**Quiero** poder iniciar sesión en la aplicación,**Para que** pueda acceder a la gestión del equipo y sus estadísticas.

Descripción La aplicación debe proporcionar como pantalla inicial un inicio de sesión, donde solicite el correo y la contraseña almacenados en la base de datos. Después serán validados y cargará la siguiente vista.

Criterios de Aceptación

- El usuario debe ver campos para ingresar el correo y la contraseña.
- Debe haber una lógica de acceso a la base de datos después de pulsar el botón "Iniciar Sesión".
- Al ingresar credenciales **válidas**, el usuario debe ser dirigido a la pantalla de selección de equipo.
- Al ingresar credenciales **inválidas**, se mostrará el correspondiente cuadro de error, pidiendo volver a introducir las claves.
- Debe de haber un botón para cerrar sesión que redirigirá a esta vista de nuevo.

Seleccionar el Equipo**Como** *Usuario*,**Quiero** seleccionar un equipo al sobre el cual tenga permisos o esté relacionado en la base de datos,**Para que** pueda acceder a la gestión del equipo y sus estadísticas.

Descripción Después de iniciar sesión, la aplicación consultará a la base de datos los equipos disponibles para el usuario de la sesión, mostrándole una lista de estos equipos. En el caso de que la base de datos no admita que un entrenador entrene dos equipos distintos, se hará la lógica igualmente para cuando se implementen los permisos.

Criterios de Aceptación

- El usuario debe ver una lista de equipos disponibles.
- Al seleccionar un equipo, el usuario debe ser dirigido a la pantalla principal del equipo.
- Si el usuario no tiene permisos sobre ningún equipo, se mostrará el mensaje correspondiente y será redirigido al inicio de sesión.

Vista general del equipo *Home*

Como *Entrenador/a*,

Quiero tener una vista sobre la información general del equipo,

Para que tenga una visión rápida del estado de mi equipo, sus estadísticas y un calendario rápido.

Descripción Una vez seleccionado el equipo, la vista principal *Home* debe proporcionar un resumen general y rápido del estado del equipo, mostrando la lista de jugadores, estadísticas, gráficos de rendimiento y un calendario de eventos.

Criterios de Aceptación

- El entrenador debe ver una lista de los jugadores del equipo.
- Debe ver una lista de los KPIs **de todo el equipo** con su correspondiente puntuación de rendimiento.
- Basado en la lista de KPIs, debe mostrarse una columna de gráficos de rendimiento.
- Debe haber un calendario mensual con fechas de sesiones.

Detalles del jugador

Como *Entrenador/a*,

Quiero ver de una forma detallada la evolución individual de cada jugador,

Para que pueda conocer sus datos personales y analizar detenidamente su rendimiento.

Descripción La pantalla de detalles del jugador servirá para analizar más detalladamente el rendimiento individual de un jugador. Debe mostrar información personal del jugador, sus estadísticas de rendimiento acompañadas de gráficos y una matriz DAFO sobre dichas estadísticas.

Criterios de Aceptación

- El entrenador debe ver la información personal del jugador (fecha de nacimiento, DNI, número de teléfono).
- Debe ver una lista de los KPIs **individuales** del jugador con su correspondiente puntuación de rendimiento.
- Basado en la lista de KPIs, debe mostrarse una columna de gráficos de rendimiento.
- Debe haber una matriz DAFO con fortalezas, debilidades, oportunidades y amenazas.

Visualización de Estadísticas

Como *Entrenador/a*,

Quiero ver de una manera gráfica las estadísticas de mi equipo,

Para que pueda analizar el rendimiento del equipo y tomar decisiones basadas en datos.

Descripción La pantalla de estadísticas debe mostrar gráficos detallados de todos los KPIs entrenados a lo largo de todas las sesiones de entrenamiento, con el fin de que el entrenador pueda tomar decisiones basadas en los gráficos que está viendo.

Criterios de Aceptación

- El entrenador debe ver todos los gráficos sobre todos los indicadores de rendimiento **en equipo** que se hayan entrenado alguna vez.
- Debe haber botones para generar informes de todos los KPIs o de un KPI específico en formato .pdf o .csv.

- Los gráficos deben mostrar la evolución del rendimiento en diferentes periodos de tiempo.

Visualización de Estadísticas

Como *Entrenador/a*,

Quiero ver el calendario de sesiones registradas

Para que pueda planificar las actividades del equipo de manera efectiva.

Descripción La pantalla de calendario debe mostrar una vista, por defecto mensual pero puede ser cambiada, donde aparecerán todas las sesiones registradas en la base de datos, mostrando su nombre, hora de inicio y hora de fin.

Criterios de Aceptación

- El entrenador debe ver un calendario con fechas de sesiones.
- Las fechas deben estar claramente marcadas y diferenciadas.
- El calendario se debe poder modificar para navegar entre los años, meses y semanas más rápidamente.

Cambiar intervalo de visualización

Como *Entrenador/a*,

Quiero poder cambiar el intervalo de visualización de las estadísticas,

Para que pueda analizar el rendimiento del equipo durante diferentes periodos de tiempo.

Descripción La aplicación debe permitir al entrenador seleccionar el intervalo de visualización de las estadísticas y gráficos de rendimiento. Una vez cambiado el intervalo, se realizarán los cálculos pertinentes para mostrar correctamente los gráficos e indicadores durante el periodo seleccionado.

Criterios de Aceptación

- La aplicación debe tener un menú superior con opciones para seleccionar el intervalo de tiempo deseado.
- Al seleccionar un intervalo, las estadísticas y gráficos deben actualizarse para reflejar el periodo de tiempo seleccionado.
- La opción seleccionada debe estar marcada para que el entrenador sepa qué intervalo esta viendo.

5.2.2 Análisis de entidades de negocio

A continuación, determinaremos las relaciones que van a tener cada una de las entidades de la aplicación. Dado que este trabajo está basado para seguir los trabajos de fin de grado de [Luque (2020)] y de [Cánovas (2020)], quienes se centraron en el backend de una sistema de gestión para el fútbol base.

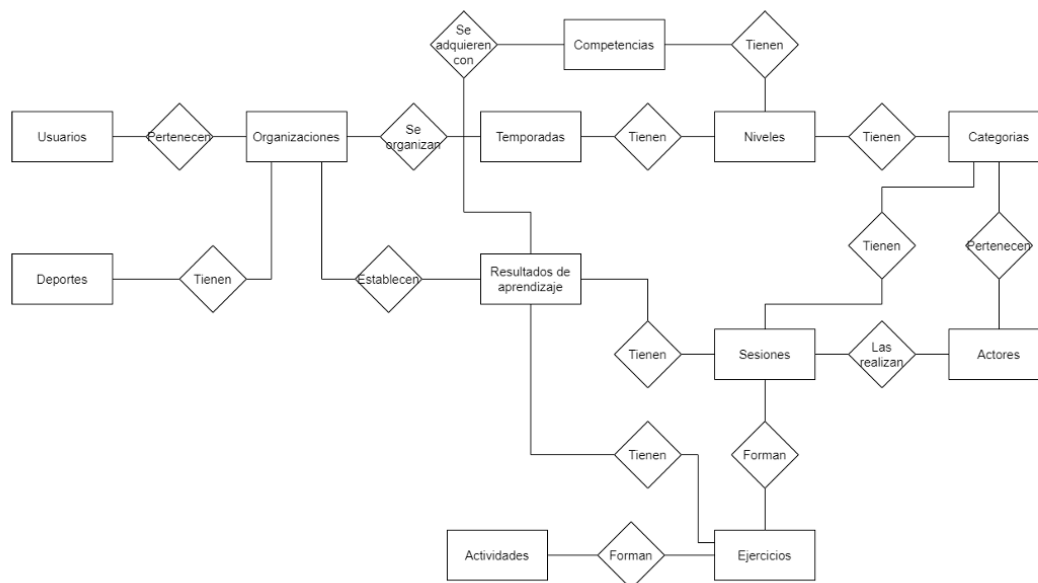


Figura 5.1: Mapa conceptual de Sergio Yáñez Cánovas y Daniel González Luque

Después de analizar el mapa conceptual, con la base de datos que se nos ha proporcionado podemos analizar el modelo Entidad-Relación, donde estarán expresadas las relaciones y atributos entre las entidades de ésta.

y pasar a la vista principal.

5.2.3.3 Pantalla principal de la organización

Muestra una visión general de la organización, dentro de esa visión general se incluirá una versión resumida de las diferentes interfaces de la aplicación (Organización, Estadísticas y Calendario). Sobre la organización mostrará una lista con los nombres de los jugadores (Actores en la base de datos de tipo 2). Sobre las estadísticas mostrará dos cuadros, el primero será un listado de los KPIs calculados de las sesiones individuales de cada jugador y mostrados de manera grupal. El segundo será una pequeña columna de gráficos sobre dichos KPIs. Por último, en la parte inferior se mostrará un pequeño calendario que mostrará las sesiones extraídas de la base de datos.

5.2.3.4 Detalles del Actor

Presenta información detalla sobre un jugador específico, mostrando la información dividida en cinco cuadros. El primer cuadro mostrará los datos personales del jugador (foto de perfil, nombre y apellidos, fecha de nacimiento, DNI y teléfono). El segundo cuadro mostrará los KPIs que ese jugador ha entrenado por lo menos una vez, junto con la evaluación de rendimiento de cada ejercicio. El tercero contendrá una matriz DAFO (debilidades, amenazas, fortalezas y oportunidades) calculada a partir de un análisis estadístico a partir de las diferentes evaluaciones a lo largo del tiempo. El cuarto cuadro contendrá los gráficos de la evolución de los KPIs. Por último, el último cuadro contendrá el historial de partidos del jugador con los minutos jugados.

5.2.3.5 Estadísticas

Proporciona una vista detallada de las estadísticas de la organización a través de gráficos de varios KPIs. Incluirá botones para generar informes de todos los KPIs.

5.2.3.6 Calendario

Permite una navegación sencilla entre las fechas del año para mostrar todas las sesiones que tienen que ver con la organización seleccionada por el usuario.

5.2.3.7 Menú

Será implantado en todas las interfaces, dispondrá de un menú superior y uno lateral. El menú superior mostrará el nombre de la organización seleccionada, el nombre de la interfaz actual, el nombre y foto del usuario de la sesión junto con un botón de cerrar sesión. Este menú superior contendrá un menú inferior donde el usuario podrá elegir el intervalo de tiempo de la recogida de KPIs, dicho intervalo se aplicará a todas las interfaces. El menú lateral será un menú desplegable que contendrá las diferentes vistas de la aplicación, desplegado se verá el nombre junto al icono mientras que si no lo está solo se verá el icono.

5.2.4 Análisis de servicios

Este trabajo seguirá una arquitectura basada en un patrón de diseño MVC (Modelo-Vista-Controlador) implementado con Vue.js. Describiremos los servicios que nuestra aplicación proporcionará, enfocados en la gestión y el análisis de la organización y sus actores.

5.2.4.1 Autenticación

- Validar las credenciales del usuario con la base de datos.
- Obtener un token de sesión para el framework DAO.
- Manejar el cierre de sesión eliminando el token de autenticación.

5.2.4.2 Organizaciones

- Consultar y devolver las organizaciones asociadas al usuario autenticado.
- Establecer la organización seleccionada durante la sesión.

5.2.4.3 Actores

- Consultar y devolver la lista de actores que pertenecen a la organización seleccionada.
 - Consultar y devolver la información detallada de un jugador específico, incluyendo datos personales y registros de su rendimiento.
-

5.2.4.4 Estadísticas

- Consultar y devolver los KPIs de forma grupal para toda la organización.
- Consultar y devolver los KPIs de forma individual de un actor específico de la organización.
- Generar informes en formatos .pdf o .csv de todos los KPIs.
- Actualizar las consultas seleccionando el intervalo de tiempo para la visualización de estadísticas y gráficos.

5.2.4.5 Calendario

- Consultar y devolver las sesiones de la base de datos sobre la organización seleccionada.
- Permitir la visualización del calendario en diferentes periodos.

5.3 Diseño del sistema

5.3.1 Arquitectura general

Siguiendo los trabajos de [Luque (2020)] y [Cánovas (2020)], ellos definen que "la estructura de su sistema estará formada por 4 fases: planificación, ejecución, análisis y evaluación, que forman un ciclo completo y cíclico para el desarrollo de los deportistas."

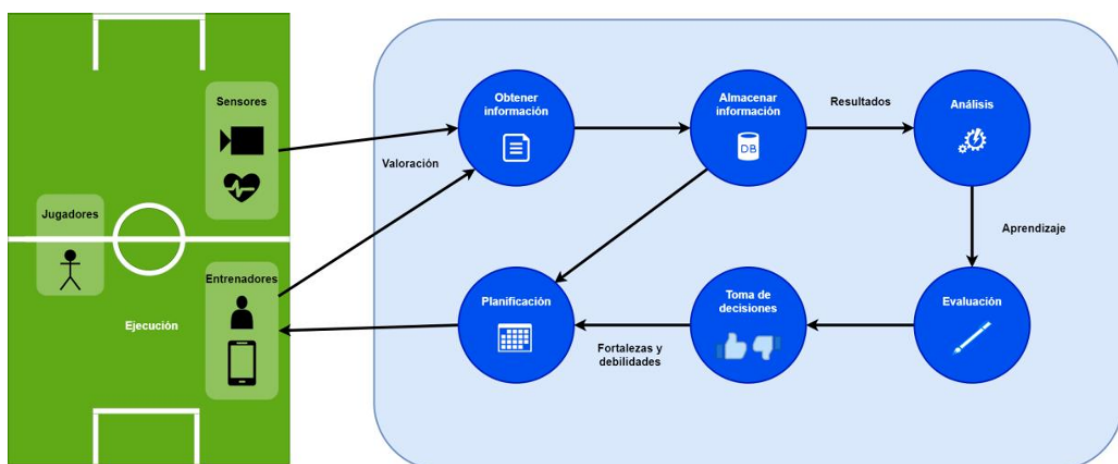


Figura 5.3: Arquitectura general del sistema de Sergio Yáñez Cánovas

Nuestro sistema se centrará en la fase de **análisis y la de evaluación** de los datos recopilados. Una vez acabe la fase de recopilación y almacenamiento de datos, la información será totalmente visible a través de la aplicación web. Esta información conllevará los nuevos resultados de la sesión dados por el entrenador y la actualización de los gráficos y las matrices DAFO de cada jugador. Para la actualización de los nuevos resultados de la matriz DAFO, se volverá a hacer el cálculo estadístico siguiendo cada resultado de nuevo teniendo en cuenta que se evalúan las pendientes y la consistencia del jugador. Una vez la aplicación obtenga los resultados serán totalmente visibles para el usuario.

5.3.2 Diseño de interfaces de usuario

A continuación, se mostrarán los mockups elaborados para todas las interfaces definidas en Análisis de interfaces. Estas interfaces pueden ser modificadas a lo largo del trabajo.

5.3.2.1 Inicio de sesión

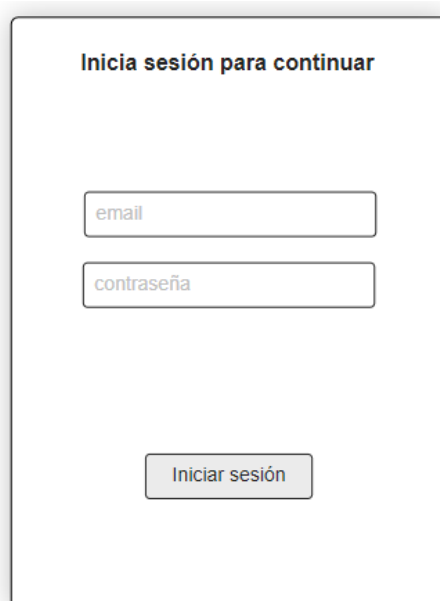
The image shows a login form mockup. It is a vertical rectangle with a light gray border. At the top, the text "Inicia sesión para continuar" is centered in a bold, black font. Below this, there are two input fields: the first is labeled "email" and the second is labeled "contraseña", both in a light gray font. At the bottom of the form, there is a button labeled "Iniciar sesión" in a light gray font. The entire form is set against a white background.

Figura 5.4: Mockup de inicio de sesión

Este *mockup* es lo que verá el usuario nada más entre en la aplicación, solo tendrá que introducir el correo electrónico y la contraseña correspondientes para iniciar su sesión.

5.3.2.2 Selección de organización



Figura 5.5: Mockup de selección de equipo

Una vez iniciada la sesión, el usuario verá las organizaciones que están relacionadas con él, pudiendo seleccionar cualquiera de ellas.

5.3.2.3 Página principal

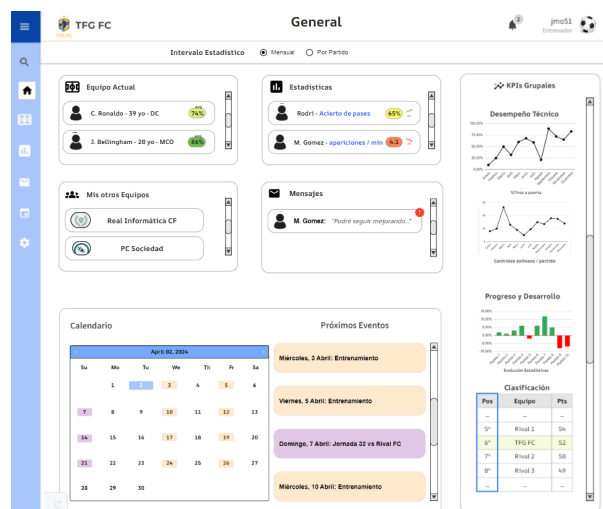


Figura 5.6: Mockup de la página principal

Una vez seleccionada la organización, se mostrará una vista principal con las funciones resumidas de la aplicación.

5.3.2.4 Equipo

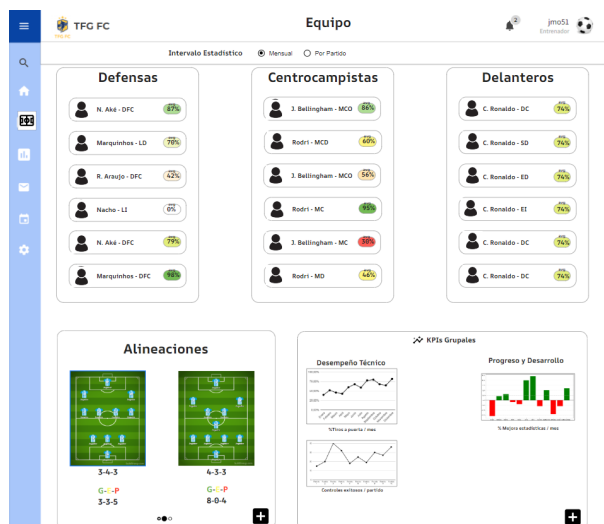


Figura 5.7: Mockup de la página de equipo

Pudiendo seleccionar esta vista desde la vista principal o desde el menú lateral, mostrará un listado de todos los jugadores que componen la organización, junto con alineaciones que se han dado y sus estadísticas.

5.3.2.5 Jugador

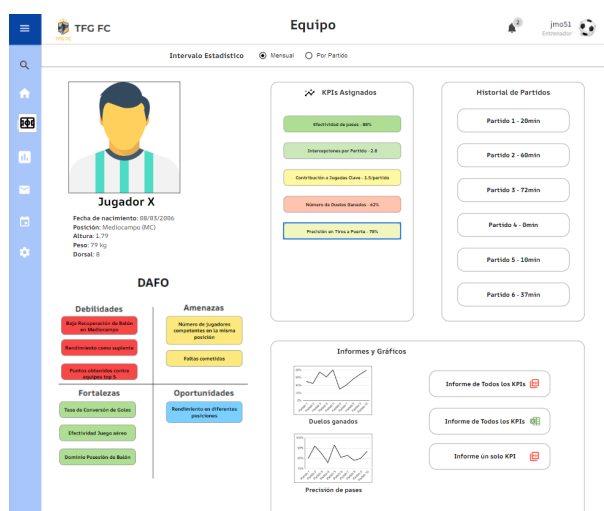


Figura 5.8: Mockup de la página de jugador

A esta vista se accederá haciendo click sobre un jugador, ya sea en la vista principal o en al vista del equipo. Esta vista mostrará su información personal, su seguimiento de estadísticas y una matriz DAFO para guiar al usuario sobre su rendimiento de una forma más precisa.

5.3.2.6 Estadísticas



Figura 5.9: Mockup de la página de estadísticas

Pudiendo seleccionar esta vista desde la vista principal o desde el menú lateral, mostrará un listado de gráficos sobre la evolución de las puntuaciones dadas por el entrenador durante las sesiones.

5.3.2.7 Calendario

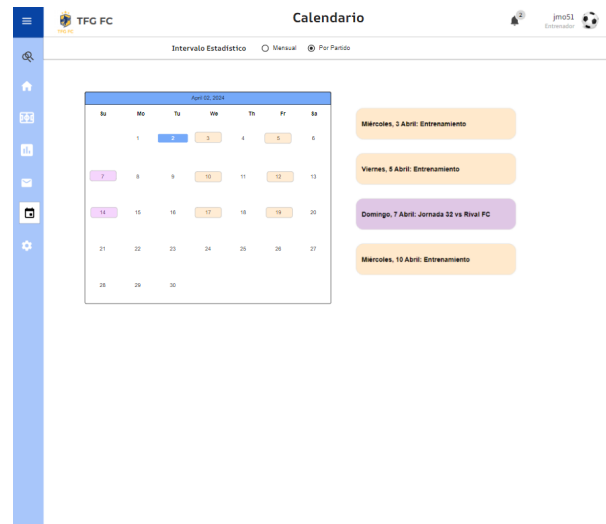


Figura 5.10: Mockup de la página de calendario

esta vista también será accesible por el menú lateral y por la vista principal. Mostrará un calendario junto con las sesiones más próximas.

5.3.3 Diseño lógico de la base de datos

Dado el apartado de Análisis de entidades de negocio, el diseño del modelo lógico de la base de datos que vamos a utilizar vendrá dado gracias a los trabajos de [Luque (2020) y [Cánovas (2020)]:

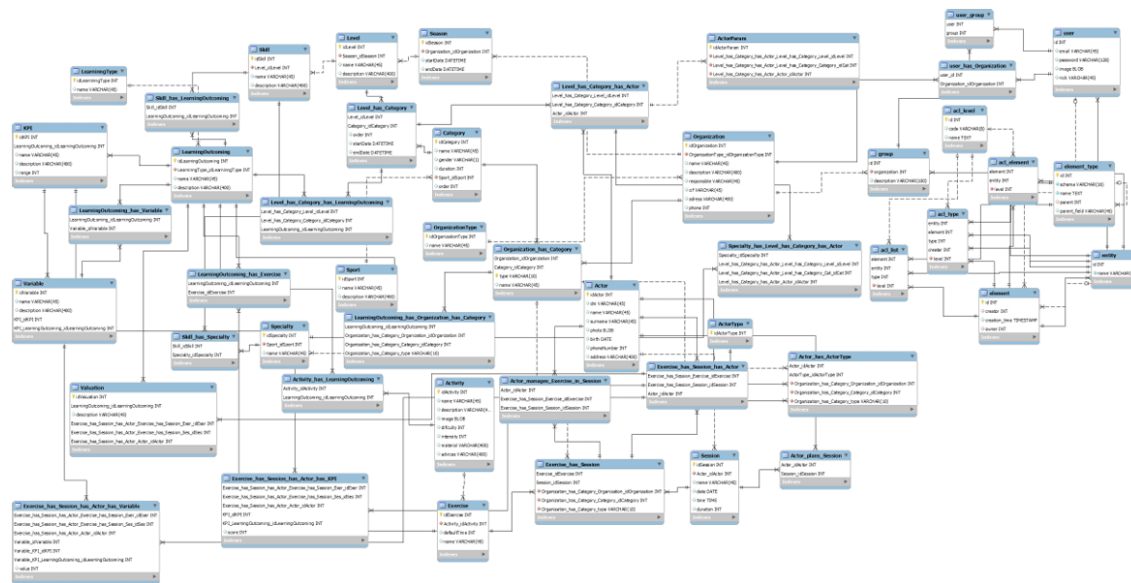


Figura 5.11: Modelo lógico de Sergio Yáñez Cánovas y Daniel González Luque

Dado este esquema lógico, procederemos a analizar las principales tablas a las que accedemos durante la implementación del proyecto.

5.3.3.1 User

Esta tabla contiene los atributos: email, nick, password e image. Los atributos email y password se utilizarán para validar el inicio de sesión del usuario, mientras que los atributos nick e image serán utilizados en la cabecera de la aplicación para mostrar al usuario que ha iniciado la sesión.

5.3.3.2 Organization

En esta tabla se almacenan los datos de las organizaciones, de la que solamente accederemos a su ID y su nombre para ser mostrado por pantalla.

5.3.3.3 User has organization

Contiene la información necesaria para saber si el usuario que ha iniciado sesión pertenece a una organización de la base de datos.

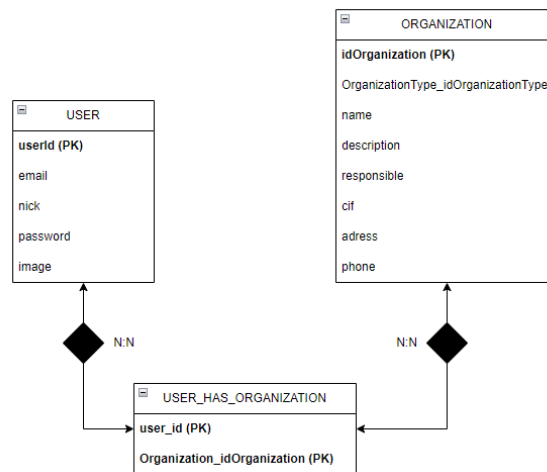


Figura 5.12: Relación entre las tablas user, organization y user has organization

5.3.3.4 Actor

Contiene la información de los actores que componen la organización, como se define en los trabajos de Sergio y Daniel: "Un actor es cualquier participante o miembro de la organización. Jugadores, Entrenadores, Preparadores físicos, Entrenadores de porteros, Coordinadores, ..." [Luque (2020), [Cánovas (2020)]. Esta tabla será utilizada para extraer la información personal de cada jugador, así como verificar las sesiones asociadas con dicha organización.

5.3.3.5 Session

Almacena los datos de las sesiones de la organización (nombre, fecha, hora y duración) llevadas por un actor. Estos datos serán extraídos para rellenar el calendario de la aplicación y también para analizar los resultados de sesiones pasadas.

5.3.3.6 Actor has actortype

Indica el tipo de actor dentro de una organización dada, analizando los datos de la base de datos, podemos concluir que el tipo actor '2' corresponde a los jugadores, mientras que el tipo actor '1' a los entrenadores o coordinadores de la organización. Esta información la utilizaremos para extraer a todos los actores de tipo '2' (Jugadores) para hacer un listado en la aplicación.

5.3.3.7 Kpi

Se utiliza para almacenar los indicadores clave de rendimiento (Key Performance Indicators), están relacionados con resultados de aprendizaje 'Learning Outcomes'. Los usaremos para extraer su nombre para relacionarlo con la puntuación de cada ejercicio.

5.3.3.8 Exercise

Servirá para obtener el nombre del ejercicio que se ha realizado para mostrarlo en el análisis. Como se define en los trabajos de [Luque (2020) y [Cánovas (2020)]: "Un ejercicio es una actividad concreta con una duración de tiempo y que deberá ser realizado por un equipo o un conjunto de actores. Un ejercicio es definido para entrenar uno o varios resultados de aprendizajes y debe indicarse si es para todo el equipo, si es un deporte de equipo, o para miembros concretos. Se debe indicar el orden de los ejercicios en la sesión, que lo indicará la posición en la IU. Un ejercicio debe tener una duración concreta."

5.3.3.9 Exercise has session has actor has kpi

De esta tabla obtendremos la puntuación de los ejercicios realizados por un actor dentro de una sesión dada. La puntuación de los ejercicios estará ligada a un KPI de la base de datos, información que será mostrada en el análisis del equipo y de cada jugador.

```
7     userId: string
8     email: string
9     nick: string
10    password: password
11    aclElement: aclElement[]
12    aclType: aclType[]
13    aclList: aclList[]
14    groups: userGroup[]
```

Código 5.2: tipo dato organización

```
1 organization:
2   type: object
3   displayName: Organization
4   description: Organization
5   properties
6     idOrganization: integer
7     OrganizationType_idOrganizationType: int
8     name: string
9     description: string
10    responsible: string
11    cif: string
12    adress: string
13    phone: string
```

Código 5.3: tipo dato actor

```
1 actor:
2   type: object
3   displayName: actor
4   description: Información de un actor
5   properties:
6     idActor: integer
7     dni: string
8     name: string
9     surname: string
10    photo: blob
11    birth: datetime
12    phoneNumber: int
13    address: string
```

Código 5.4: tipo dato sesión

```
1 session:
```

```
2  type: object
3  displayName: session
4  description: Sesiones de una organización
5  properties:
6      idSession: integer
7      Actor_idActor: int
8      name: string
9      date: datetime
10     time: time
11     duration: int
```

Código 5.5: tipo dato kpi

```
1 kpi:
2     type: object
3     displayName: KPI
4     description: KPIs de un resultado de aprendizaje
5     properties
6         idKPI: integer
7         LearningOutcoming_idLearningOutcoming: int
8         name: string
9         description:string
10        range:int
```

Código 5.6: tipo dato exercise

```
1 exercise:
2     type: object
3     displayName: exercise
4     description: Ejercicios de una sesión de entrenamiento
5     properties:
6         idExercise: integer
7         Activity_idActivity: int
8         defaultTime: int
9         name: string
```


5.4 Implementación

5.4.1 Arquitectura técnica

Explicaremos como funciona la arquitectura de la aplicación de Vue.js, que funciona conjuntamente con el lado del servidor que nos ha sido proporcionado.

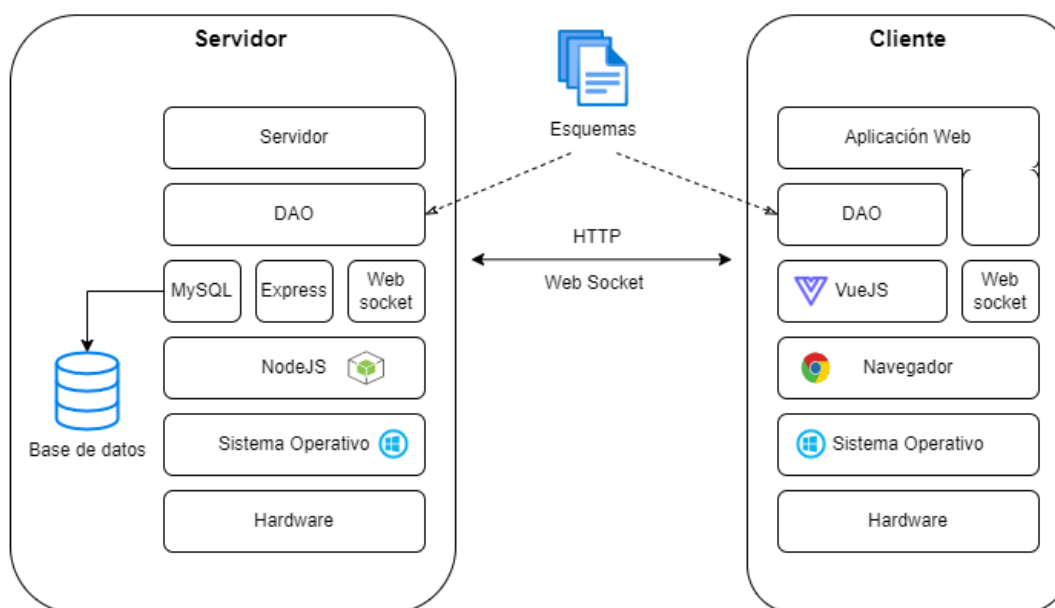


Figura 5.14: Arquitectura técnica del sistema

Tanto en el cliente como en el servidor, las dos capas inferiores corresponderán al hardware y al sistema operativo sobre el que se despliegue la aplicación, en mi caso, la aplicación de Vue ha sido desarrollada en Windows 10.

Siguiendo por el lado del servidor desarrollado en [Luque (2020)] y [Cánovas (2020)], la siguiente capa será conformada por el framework de servidor Node.js, que será el encargado de procesar las peticiones y generar las respuestas hacia el cliente.

La siguiente capa la conforman la aplicación que administra la base de datos de la aplicación, MySQL, Express será utilizado como framework de Node para la creación de los servicios. Y por último WebSocket [IONOS (2020)], que realiza las conexiones cliente-servidor.

En la siguiente capa superior estará el framework DAO (Data Access Object) proporcionado

por nuestro tutor. Se trata de un framework que gestiona los accesos a la base de datos, haciendo las peticiones desde la aplicación de Vue.js mucho mas sencillas.

En la capa superior se encuentra la lógica del servidor, que puede incluye lógica de negocio y controladores que orquestan las diferentes partes de la aplicación.

Siguiendo por el lado del Cliente, posterior a las capas de hardware y sistema operativo, se encuentra la capa del navegador, que permite a los usuarios interactuar con la aplicación web.

La siguiente capa la conforman WebSocket, utilizado para mantener la comunicación en tiempo real con el servidor, y Vue.js, siendo un framework de JavaScript utilizado para construir interfaces de usuario interactivas. [VueJS (2024b)].

En las últimas capas se encontrarán el DAO, contenido dentro de la aplicación web para realizar las peticiones pertinentes. La aplicación web es la interfaz con la que interactúan los usuarios.

5.4.2 Implementación de interfaces de usuario

Mostraremos las interfaces de usuario que hemos desarrollado con Vue.js, los datos que se verán por pantalla son datos inventados de ejemplo para poder realizar la captura de pantalla.

5.4.2.1 Inicio de sesión



Figura 5.15: Implentación de la interfaz de inicio de sesión

La implementación de esta interfaz ha sido muy similar a la del *mockup*, siendo un cuadro en el que se solicitan el email y la contraseña al usuario.

5.4.2.2 Selección de organización

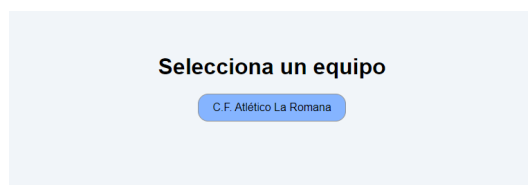


Figura 5.16: Implentación de la interfaz de selección de organización

La implementación de esta interfaz también es similar al *mockup*, mostrando las organizaciones que están relacionadas con el usuario iniciado.

5.4.2.3 Página principal

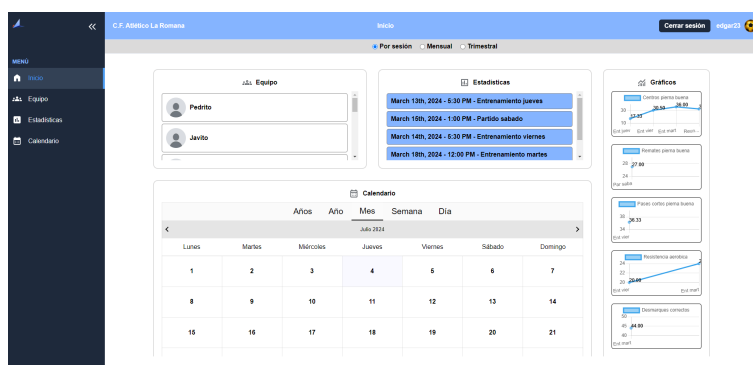


Figura 5.17: Implentación de la interfaz de selección de organización

A diferencia del diseño de la interfaz, se han eliminado los cuadros de mis otros equipos, ya que las tablas de la base de datos no admiten que un actor sea entrenador de dos equipos distintos, por lo tanto aunque se implemente en el futuro dejar un cuadro vacío no tendría sentido.

También se ha eliminado el cuadro de mensajes, considerando que la forma de comunicación idónea puede ser durante las sesiones. En la columna de estadísticas se ha eliminado la clasificación, ya que no tenemos acceso a ella, y finalmente los próximos eventos el cuadro del calendario, para no restarle relevancia a la vista completa.

5.4.2.4 Equipo



Figura 5.18: Implementación de la interfaz de equipo

Esta interfaz puede ser la que más cambios haya recibido. En primer lugar, no se separan jugadores por posiciones. Este dato además de no existir en la base de datos, no es un dato muy recurrente en categorías inferiores de fútbol base.

Tampoco tenemos acceso a las alineaciones que se ha ido utilizando, por lo que descartaremos esa parte. Por último, los gráficos se han eliminado para no restarle importancia al cuadro de las estadísticas.

5.4.2.5 Jugador

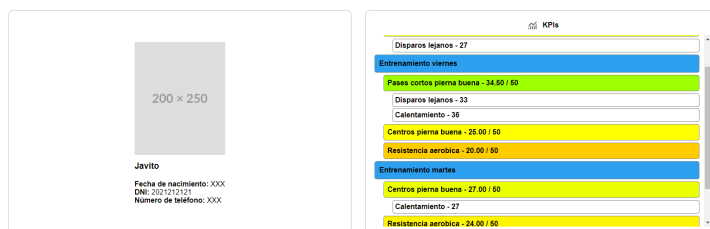


Figura 5.19: Implementación de la interfaz de equipo



Figura 5.20: Implentación de la interfaz de equipo

Este diseño también ha sufrido cambios, ahora solo mostrará gráficos en su cuadro correspondiente, para una mayor visibilidad en caso de muchos datos. Adicionalmente, ya no verá el historial de partidos, dado que es un dato del que no disponemos.

La matriz DAFO estará compuesta de las cuatro tarjetas desplegables para la visualización de los datos de una manera más cómoda. También el cuadro antiguo de KPIs asignados ha sido sustituido por un historial de sesiones, meses o trimestres, según lo que indique el menú, de KPIs con sus puntuaciones y ejercicios.

5.4.2.6 Estadísticas

5.4.2.7 Calendario

5.4.3 Esquema físico de la base de datos

aaa

5.4.4 Implementación de servicios

aaa

5.4.4.1 Matriz DAFO

5.4.4.2 Promedio GPA

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

5.4.4.3 Desviación estándar

aaa

5.4.4.4 Pendiente de tendencia

aaa

5.4.4.5 Cambios positivos y negativos entre los intervalos de tiempo

5.5 Pruebas y validación

aaa

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)
- Cánovas, S. Y. (2020). *Sistema integral para el entrenamiento personalizado de fútbol base*. <https://rua.ua.es/dspace/handle/10045/36931>. (Accessed: 2024-02-20)
- 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 iconoes, 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)
- IONOS. (2020). *¿qué es websocket?* <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-websocket/>. (Accessed: 2024-07-04)
- 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)
- Luque, D. G. (2020). *Sistema de análisis y cuadro de mandos para evaluación de formación personalizada en el fútbol base*. <https://rua.ua.es/dspace/handle/10045/36931>. (Accessed: 2024-02-20)
- 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)
-