



IES COMERCIO LOGROÑO

Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Web (DAW)

LeagueNow TFG



Alumno: Sergio Ortiz de Zarate Garrido

Tutor: Sara Berezo Loza

TFG - 2º DAW IES Comercio 2024-2025

Introducción	4
Contexto del problema o necesidad	4
Justificación del proyecto	5
Metodología utilizada	6
Objetivos generales.....	7
Objetivos específicos	7
Fases del proyecto	8
Requisitos funcionales	8
Gestión de usuarios	8
Gestión de equipos.....	8
Gestión de jugadores	8
Gestión de partidos y resultados.....	8
Clasificación	8
Búsqueda interactiva	9
Calendario	9
Diseño	10
Mockups y vistas creadas	10
Usuario registrado	11
🛠️ Usuario administrador	11
Descripción de componentes hardware y software	13
Componentes hardware.....	13
Componentes software.....	14
Diagrama de Casos de Uso	14
Base de datos	16
Elección del Sistema Gestor de Base de Datos	17
Diseño de la Base de Datos	17
Relación entre las Tablas.....	17
Seguridad y Gestión de Datos	17
Implementación y Configuración	18
Tecnologías y Herramientas Utilizadas	18
Estructura del Proyecto.....	18

Conección a la Base de Datos.....	19
Helpers y Librerías	20
Flujo General de la Aplicación.....	20
Pruebas unitarias.....	23
Pruebas de integración.....	24
Pruebas de usuario	24
Pruebas de rendimiento (básicas)	24
Ampliación y posibles mejoras.....	25
1. Gestión completa de partidos	25
2. Ampliación de privilegios de usuario	25
3. Calendario interactivo más completo	25
4. Compatibilidad y diseño responsive	26
5. Mejoras de rendimiento	26
6. Sistema de notificaciones.....	26
7. Estadísticas avanzadas	26
Conclusión.....	26
Principales logros y aprendizajes	27
Dificultades encontradas	27
Aplicabilidad del proyecto	27
Bibliografía	27
Documentación técnica y frameworks	28
Recursos visuales y datos de ejemplo	28
Fuentes deportivas para datos y estadísticas.....	28

Introducción

Contexto del problema o necesidad

En las ligas deportivas de nivel amateur o local es muy común que la gestión de la competición se realice de forma manual o poco profesional. Muchos torneos se organizan utilizando herramientas como hojas de cálculo, documentos compartidos, grupos de mensajería instantánea o incluso con anotaciones en papel. Aunque estas soluciones pueden ser suficientes para ligas pequeñas, con el tiempo acaban generando problemas como desorganización, errores en los resultados, dificultad para seguir la clasificación o pérdida de información importante.

Además, quienes se encargan de organizar estas competiciones suelen ser personas que lo hacen por afición o voluntariado, sin contar necesariamente con conocimientos técnicos avanzados. Por eso, existe una necesidad real de contar con una herramienta digital sencilla, práctica y accesible que permita gestionar correctamente todos los aspectos básicos de una liga deportiva: desde el registro de equipos y jugadores hasta el seguimiento de los resultados y la clasificación general.

Este proyecto nace con ese objetivo. Se trata de desarrollar una aplicación web que facilite la gestión de una liga de manera ordenada, permitiendo centralizar toda la información y mejorar la experiencia tanto para los organizadores como para los participantes. A diferencia de plataformas profesionales como *Flashscore*, esta solución se enfoca en cubrir únicamente las funcionalidades esenciales, con una interfaz simple y adaptada al contexto amateur.

La idea es ofrecer una herramienta funcional que se pueda utilizar fácilmente desde cualquier navegador, sin necesidad de instalaciones ni configuraciones complicadas, contribuyendo así a mejorar la organización y el seguimiento de competiciones deportivas de pequeño o mediano tamaño.

Justificación del proyecto

La importancia de este proyecto radica en su utilidad práctica y en la aplicabilidad directa que puede tener en situaciones reales. No se trata simplemente de un ejercicio académico, sino de una herramienta que podría ser utilizada por cualquier persona o grupo que organice competiciones deportivas a nivel local o amateur. Al contrario que muchas plataformas profesionales que resultan complejas, con exceso de funciones innecesarias o requieren conocimientos técnicos para su uso, esta aplicación está pensada para ser sencilla, funcional y directa.

Actualmente, no es raro encontrar torneos o ligas donde la gestión se basa en apuntes a mano, hojas de Excel compartidas o incluso en grupos de WhatsApp donde se suben fotos de resultados. Estos métodos improvisados terminan generando confusión, errores, pérdida de datos y mucha dependencia de la persona que organiza. Este tipo de problemas se solucionan con una aplicación web como la propuesta en este proyecto, que centraliza la información y la presenta de forma clara, accesible y estructurada.

Además, desde el punto de vista académico y personal, este proyecto es muy relevante porque me ha permitido aplicar de forma práctica los conocimientos adquiridos a lo largo del grado, especialmente en áreas como el desarrollo web, la gestión de bases de datos, el diseño de software con patrón MVC y la seguridad de usuarios. Todo lo aprendido se ve reflejado en el desarrollo de esta aplicación, que va mucho más allá de un simple trabajo de clase.

También hay que destacar que, aunque se ha desarrollado con un enfoque concreto (una liga deportiva), la base del proyecto es totalmente ampliable. En el futuro, se podrían añadir nuevas funcionalidades, adaptar la plataforma a otros tipos de competiciones, mejorar el diseño o incluso convertirla en una app móvil. Por lo tanto, no solo se justifica por la necesidad inmediata que cubre, sino también por su potencial de evolución y su valor como base para desarrollos más complejos.

En resumen, este proyecto es importante porque responde a una necesidad real y concreta, porque es funcional, accesible y útil, y porque representa una oportunidad de aplicar de forma real todos los conocimientos adquiridos durante la carrera.

Metodología utilizada

El desarrollo del proyecto se ha realizado de forma individual, siguiendo una metodología basada en fases ordenadas y progresivas. Desde el principio tuve claro que era importante organizar bien el trabajo para no atascarme y poder ir avanzando poco a poco, así que dividí el proceso en etapas claras: análisis de necesidades, diseño de la base de datos y de la estructura del sistema, implementación funcional, pruebas y ajustes finales.

Para empezar, analicé qué funcionalidades básicas debía tener una herramienta de gestión de liga deportiva: registro de equipos, jugadores, resultados, clasificaciones, sistema de usuarios, control de accesos, etc. A partir de ahí, elaboré un esquema general de cómo debería funcionar la aplicación, qué vistas necesitaría y cómo se conectaría todo internamente. Esto me sirvió para tener una idea clara de por dónde empezar y cómo estructurar el proyecto desde el principio.

Una vez definido el funcionamiento general, pasé al diseño de la base de datos, que es una parte fundamental del sistema. Utilicé **phpMyAdmin** para crear las tablas necesarias, como la de usuarios, equipos, jugadores y partidos, asegurándome de que las relaciones entre ellas fueran coherentes y funcionales. También fui añadiendo campos pensando en futuras ampliaciones, como por ejemplo el control de roles para distinguir entre administradores y usuarios normales.

En cuanto al desarrollo de la aplicación web, utilicé principalmente **PHP** para la lógica del servidor, junto con **HTML, CSS y JavaScript** para el diseño de la interfaz. Seguí el patrón de diseño **MVC (Modelo-Vista-Controlador)** para mantener el código organizado y separado por responsabilidades. Esta estructura me ha facilitado mucho el mantenimiento y la ampliación del código, ya que cada parte está bien diferenciada: los modelos se encargan de la base de datos, los controladores gestionan la lógica, y las vistas presentan los datos al usuario.

Para el entorno de desarrollo utilicé **AMPPS**, un paquete que incluye servidor Apache, MySQL y phpMyAdmin, y que permite simular un entorno de servidor local directamente en el ordenador. Gracias a esta herramienta pude programar y probar la aplicación de forma rápida y cómoda, sin necesidad de subirla a un servidor externo durante el desarrollo. Esto me permitió trabajar de forma más ágil, detectar errores y hacer correcciones en tiempo real.

En resumen, el proyecto se ha desarrollado siguiendo una metodología práctica y estructurada, priorizando primero la funcionalidad básica, y después la estabilidad del sistema. A medida que completaba cada parte, realizaba pruebas para asegurarme de que todo funcionaba correctamente antes de continuar. Esta forma de trabajar me ha permitido avanzar con seguridad y mantener un buen control del progreso en todo momento.

todo funcionaba correctamente antes de continuar. Esta forma de trabajar me ha permitido avanzar con seguridad y mantener un buen control del progreso en todo momento.

Objetivos generales

- Desarrollar una aplicación web para la gestión de una liga deportiva, con funciones básicas pero bien implementadas, priorizando la facilidad de uso y la funcionalidad real.

Objetivos específicos

- Diseñar una interfaz clara, sencilla y ordenada, donde el usuario pueda moverse sin complicaciones.
- Implementar un sistema de registro e inicio de sesión con control de roles (usuario normal y administrador).
- Permitir la gestión completa de los equipos, jugadores y partidos de la liga.
- Mostrar automáticamente la clasificación según los resultados introducidos.
- Organizar todo el proyecto siguiendo el patrón de diseño MVC para mantener el código estructurado.
- Proteger los datos de los usuarios con prácticas básicas de seguridad como el hash de contraseñas.
- Diseñar una base de datos funcional y bien organizada, con relaciones lógicas entre tablas.
- Usar AMPPS como entorno local de desarrollo, facilitando el testeo y la puesta en marcha de la aplicación.
- Hacer pruebas de funcionamiento durante todo el desarrollo, corrigiendo errores a medida que aparecían.

Fases del proyecto

Requisitos funcionales

Para que la aplicación cumpla su objetivo principal —gestionar de forma sencilla una liga deportiva— se han definido e implementado los siguientes requisitos funcionales:

Gestión de usuarios

- El sistema permite a los usuarios registrarse mediante un formulario.
- Los usuarios pueden iniciar sesión con su nombre de usuario y contraseña.
- El sistema distingue entre dos tipos de usuarios: administrador y usuario normal.
- El administrador puede ver, modificar o eliminar cualquier usuario, así como cambiar su rol.

Gestión de equipos

- Se puede visualizar una lista de todos los equipos registrados en la liga.
- Desde cada equipo se puede acceder a su plantilla de jugadores.
- El administrador puede añadir, editar o eliminar equipos.

Gestión de jugadores

- Cada equipo tiene su propia lista de jugadores.
- Se pueden registrar nuevos jugadores con su nombre, dorsal, posición y equipo.
- El administrador puede modificar o eliminar jugadores ya existentes.

Gestión de partidos y resultados

- Se pueden introducir los resultados de los partidos jugados entre equipos.
- El sistema calcula y actualiza la clasificación automáticamente en base a los resultados.

Clasificación

- El sistema muestra la clasificación general de la liga en base a los puntos obtenidos.

- También se ofrece una vista reducida de la clasificación en la ficha de cada equipo.

Búsqueda interactiva

- El sistema incluye un buscador interactivo para encontrar jugadores o equipos desde la página principal.
- Ofrece autocompletado para facilitar la búsqueda.

Calendario

- Cada equipo tiene un calendario donde se muestran sus próximos partidos y los ya jugados.
- Se visualiza con FullCalendar y se muestra el escudo del equipo rival en cada evento.

Diseño

Mockups y vistas creadas

Durante el desarrollo de la aplicación, se han creado distintas interfaces visuales que permiten a los usuarios interactuar con el sistema de forma sencilla e intuitiva. A continuación, se muestran algunas de las vistas más representativas del proyecto:

Diagramas de casos de uso

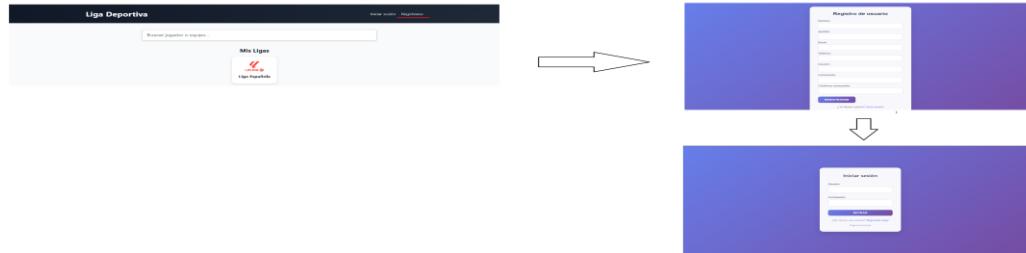
Si un usuario no tiene una cuenta registrada tendrá la posibilidad de acceder

El usuario no logueado representa a cualquier visitante que accede a la aplicación sin haberse identificado previamente. Este tipo de usuario tiene acceso limitado, pero suficiente como para conocer el funcionamiento básico de la plataforma.

Cuando accede a la aplicación, el usuario tiene dos opciones:

- Puede **navegar libremente** por la mayoría de secciones públicas, como la clasificación de la liga, el listado de equipos o las fichas de equipo con sus jugadores y calendario. Sin embargo, hay ciertas funciones que estarán restringidas, como la gestión de usuarios o jugadores, y el acceso a secciones personalizadas como “Mis Ligas”.
- Si lo desea, el usuario puede **registrarse mediante el formulario** correspondiente. Tras completar el registro y validar sus credenciales, podrá iniciar sesión y acceder a funcionalidades adicionales disponibles para usuarios registrados.

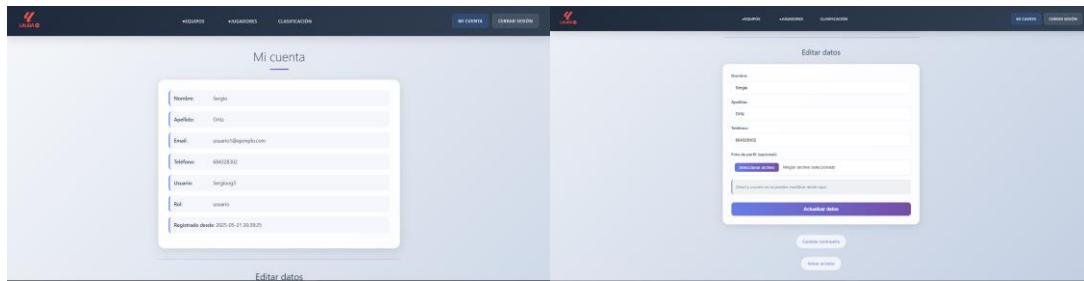
De este modo, la aplicación permite una navegación abierta pero controlada, incentivando el registro sin obligarlo, lo cual facilita una mejor experiencia de usuario.



Usuario registrado

El usuario registrado es aquel que ha creado una cuenta e iniciado sesión correctamente. Este tipo de usuario tiene acceso a funcionalidades extra dentro de la aplicación, como por ejemplo:

- Acceso a opciones de configuración o personalización básica de su experiencia.



Este rol está pensado para ofrecer una experiencia más completa a los usuarios sin necesidad de tener privilegios administrativos.

Usuario administrador

El administrador es un tipo de usuario con permisos especiales dentro del sistema. Además de contar con todas las funciones de un usuario registrado, tiene acceso exclusivo a un **panel de control** desde el cual puede realizar tareas de gestión. Entre las funciones que puede llevar a cabo se encuentran:

- **Gestión de usuarios:** puede ver la lista de todos los usuarios registrados, cambiar sus roles o eliminarlos si es necesario

Liga Deportiva

Hola, sergio [Panel de Control](#) Cerrar sesión

Panel de Control - Administrador

Bienvenido, sergio (admin)

[Cerrar sesión](#)

Listado de Usuarios

ID	USUARIO	NOMBRE	APELLIDO	EMAIL	TELÉFONO	ROL	CAMBIAR ROL
1	sergio	Sergio	Admin	sergio@ejemplo.com	123456789	admin	No se puede cambiar el rol del admin
3	Sergioog5	Sergio	Ortiz	usuario1@ejemplo.com	684328302	usuario	Usuario Cambiar
4	miguifer	Miguel	Fernandez	miguifer@gmail.com	684328301	usuario	Usuario Cambiar

- **Gestión de equipos:** puede crear, modificar o eliminar equipos de la liga.

LALIGA

EQUIPOS JUGADORES CLASIFICACIÓN MI CUENTA CERRAR SESIÓN

Listado de Equipos

+ Agregar

NOMBRE	CIUDAD	ESCUDO	ACCIONES
Real Madrid	Madrid		Editar Borrar
FC Barcelona	Barcelona		Editar Borrar
Atlético de Madrid	Madrid		Editar Borrar
Girona FC	Girona		Editar Borrar
Athletic Club	Bilbao		Editar Borrar

- **Gestión de jugadores:** puede añadir jugadores a los equipos, editar sus datos o eliminarlos.

NOMBRE	DORSAL	POSICIÓN	EQUIPO	ACCIONES
David Alaba	4	Defensa	Real Madrid	<button>Editar</button> <button>Borrar</button>
Ferland Mendy	23	Defensa	Real Madrid	<button>Editar</button> <button>Borrar</button>
Aurélien Tchouaméni	18	Centrocampista	Real Madrid	<button>Editar</button> <button>Borrar</button>
Toni Kroos	8	Centrocampista	Real Madrid	<button>Editar</button> <button>Borrar</button>
Luka Modrić	10	Centrocampista	Real Madrid	<button>Editar</button> <button>Borrar</button>
Federico Valverde	15	Centrocampista	Real Madrid	<button>Editar</button> <button>Borrar</button> Este

Este tipo de usuario es clave para el correcto funcionamiento y mantenimiento del sistema.

Descripción de componentes hardware y software

Para el desarrollo y ejecución de la aplicación web de gestión de liga deportiva, se han utilizado diversos componentes tanto de hardware como de software que permiten el correcto funcionamiento del sistema.

Componentes hardware

El proyecto se ha desarrollado y probado en un entorno local con las siguientes características de hardware:

- **Procesador:** AMD Ryzen 7 3700X, un procesador de alto rendimiento de 8 núcleos y 16 hilos, ideal para multitarea y desarrollo eficiente.
- **Memoria RAM:** 32 GB, que facilita la ejecución fluida de múltiples aplicaciones y procesos durante el desarrollo.
- **Disco duro:** Unidad M.2 NVMe Samsung 970 de 1 TB, que proporciona alta velocidad de lectura y escritura, reduciendo tiempos de carga y guardado.
- **Tarjeta gráfica:** NVIDIA RTX 3060 Ti, utilizada principalmente para tareas gráficas y apoyo en visualizaciones.
- **Conexión a Internet:** necesaria para la descarga de herramientas, documentación y para pruebas que requieran acceso externo.

Este entorno de hardware asegura una experiencia de desarrollo ágil y sin limitaciones de recursos, aunque la aplicación está diseñada para ser accesible desde cualquier dispositivo con un navegador web.

Componentes software

- **Servidor local AMPPS:** Una plataforma que integra Apache, MySQL y PHP, facilitando la configuración y gestión del entorno de desarrollo local. AMPPS ha permitido crear un servidor web y una base de datos para probar la aplicación antes de su posible despliegue en un servidor externo.
- **Lenguaje de programación PHP:** Utilizado para desarrollar la lógica del servidor y manejar las interacciones con la base de datos.
- **MySQL:** Sistema gestor de bases de datos relacional empleado para almacenar la información relativa a usuarios, equipos, jugadores, partidos y demás datos de la liga.
- **HTML, CSS y JavaScript:** Tecnologías usadas para la creación de la interfaz gráfica y mejorar la experiencia de usuario. JavaScript también ha sido empleado para funcionalidades interactivas, como el buscador con autocompletado y el calendario de partidos (integrado mediante la librería FullCalendar).
- **Modelo Vista Controlador (MVC):** Patrón de diseño empleado para organizar el código, separar la lógica de negocio, la presentación y el control de flujo, facilitando el mantenimiento y escalabilidad de la aplicación.
- **Herramientas adicionales:** Se han usado editores de código como Visual Studio Code para el desarrollo, y phpMyAdmin para la gestión visual de la base de datos.

En resumen, el sistema está basado en un entorno estándar y accesible, que no requiere componentes hardware especiales y que utiliza tecnologías de programación y gestión de bases de datos ampliamente conocidas y documentadas.

Diagrama de Casos de Uso

El esquema funcional representa de forma gráfica la estructura básica de la aplicación y cómo se relacionan entre sí sus componentes principales. En este diagrama se muestran los elementos clave del sistema y la interacción entre ellos, tanto a nivel de software como de flujo de datos.

Para esta aplicación de gestión de liga deportiva, el esquema funcional incluye:

- Interfaz de usuario (Frontend): las páginas web y componentes visuales con los que interactúan los usuarios, desarrollados con HTML, CSS, JavaScript y la librería Bootstrap para facilitar un diseño responsive y moderno.
- Controlador (Control): la capa que recibe las peticiones del usuario, procesa la lógica de negocio y decide qué datos mostrar o qué acción realizar.
- Modelo: componente encargado de gestionar la lógica de acceso y manipulación de datos, conectándose con la base de datos.
- Base de datos: donde se almacenan todos los datos persistentes, como usuarios, equipos, jugadores, resultados y calendario.
- Servidor web: en este caso Apache integrado en AMPPS, que recibe las solicitudes HTTP y ejecuta el código PHP.

Este esquema permite entender cómo se organizan las diferentes partes de la aplicación y facilita su mantenimiento y evolución futura.

Esquema funcional en texto tipo diagrama

Usuario

↓

Interfaz Web (HTML, CSS, Bootstrap, JS)

↓

Controlador (PHP, lógica MVC)

↓

Modelo (acceso y manipulación de datos)

↓

Base de datos (MySQL)

↑

|-----|

Servidor Web (Apache/AMPPS)

Cuadro resumen de Casos de Uso:

+-----+	
Sistema Liga	+-----+
[Usuario no registrado]	
- Navegar la aplicación	
- Registrarse	
[Usuario registrado]	
- Iniciar sesión	
- Acceder a "Mis Ligas"	
- Visualizar calendarios y resultados	
- Editar perfil (opcional)	
- Cerrar sesión	
[Administrador]	
- Acceder al panel de administración	
- Gestionar usuarios	
- Gestionar equipos	
- Gestionar jugadores	
- Gestionar resultados y calendario	
+-----+	

Base de datos

Para el desarrollo de la aplicación web de gestión de la liga deportiva, se ha diseñado y utilizado una base de datos relacional que almacena toda la información necesaria para el correcto funcionamiento del sistema.

Elección del Sistema Gestor de Base de Datos

Se ha optado por MySQL como sistema gestor de base de datos (SGBD) debido a su popularidad, estabilidad y buena integración con PHP, el lenguaje principal utilizado en el proyecto. Además, la base de datos se administra a través de la herramienta phpMyAdmin, que facilita la creación, modificación y mantenimiento de las tablas y datos.

Diseño de la Base de Datos

El diseño se ha realizado bajo un esquema relacional, estructurando la información en distintas tablas que permiten la gestión modular y eficiente de los datos. Las principales entidades identificadas en el sistema son:

- **Usuarios:** Contiene la información personal y de acceso de los usuarios registrados, incluyendo campos como nombre, apellido, email, teléfono, rol, nombre de usuario y contraseña cifrada.
- **Equipos:** Almacena los datos de los equipos que participan en la liga, con campos como nombre, escudo, localidad, y cualquier otra información relevante.
- **Jugadores:** Guarda los datos de los jugadores asociados a cada equipo, con atributos como nombre, dorsal, posición y el identificador del equipo al que pertenecen.
- **Partidos:** Registra los enfrentamientos entre equipos, con datos de fecha, hora, localización, resultado y equipo local y visitante.
- **Ligas y Clasificaciones:** Contiene la información sobre las distintas ligas gestionadas y la clasificación actualizada de los equipos según los resultados de los partidos.

Relación entre las Tablas

Las tablas se relacionan mediante claves primarias y foráneas para garantizar la integridad referencial. Por ejemplo, la tabla Jugadores incluye un campo `id_equipo` que hace referencia a la tabla Equipos. De esta manera, se puede consultar fácilmente qué jugadores pertenecen a cada equipo.

Seguridad y Gestión de Datos

Para proteger la información sensible, como las contraseñas de los usuarios, se emplea un algoritmo de hash (bcrypt) para almacenar las contraseñas de forma

segura. Asimismo, se implementan controles para evitar la duplicidad de registros en campos únicos como el email o el nombre de usuario.

Implementación y Configuración

Tecnologías y Herramientas Utilizadas

La aplicación web se ha desarrollado utilizando **PHP** para la lógica backend, junto con **HTML, CSS y JavaScript** para la interfaz y la interactividad. La base de datos es **MySQL**, gestionada mediante **phpMyAdmin**, y se ha empleado el patrón de arquitectura **Modelo-Vista-Controlador (MVC)** para organizar el código y facilitar su mantenimiento y escalabilidad.

Para el diseño y maquetación del frontend, se ha utilizado el framework CSS **Bootstrap** en su versión 5, importado mediante CDN. Aunque no se haya instalado localmente, Bootstrap aporta un conjunto de estilos y componentes predefinidos que permiten desarrollar una interfaz responsive y coherente de forma rápida y eficiente. De este modo, se facilita la adaptación de la aplicación a diferentes dispositivos (ordenadores, tablets, móviles) sin necesidad de crear todo el CSS desde cero.

El entorno de desarrollo se basa en un servidor local con **XAMPP**, que incluye Apache y MySQL, y se utiliza un editor avanzado como **Visual Studio Code** para la programación.

Estructura del Proyecto

La estructura principal del proyecto es la siguiente:

- /public/css/: Contiene los archivos de estilos CSS que definen la apariencia visual de la aplicación.
- /public/js/: Aquí se encuentran los archivos JavaScript responsables de la interactividad y funcionalidades dinámicas en el frontend.
- /views/index/: Carpeta que almacena las vistas principales, con archivos PHP que generan el contenido HTML que el usuario ve.
- /config/: Carpeta destinada a la configuración de la aplicación, incluyendo la conexión a la base de datos y otras constantes globales.
- /helpers/: Scripts auxiliares que contienen funciones específicas, por ejemplo, para la generación automática de partidos, cálculos de resultados, o cualquier otra funcionalidad de apoyo al core de la aplicación.

- **/librerias/**: Carpeta donde se alojan las librerías propias del proyecto que manejan el núcleo de la aplicación, como la gestión de la base de datos, la autenticación de usuarios, y otros servicios comunes reutilizables.

El archivo principal `index.php` funciona como punto de entrada y gestor de rutas, redirigiendo las peticiones a los controladores adecuados.

Conexión a la Base de Datos

La conexión con la base de datos se realiza mediante una clase llamada `DataBase`, ubicada en `/librerias/`. Esta clase utiliza PDO para establecer una conexión segura y permitir la ejecución de consultas preparadas.

Ejemplo simplificado de conexión:

```
class DataBase { private $host = 'localhost'; private $db_name = 'liga_deportiva'; private  
$user = 'root'; private $pass = ""; private $conn;  
  
public function connect() {  
    $this->conn = null;  
    try {  
        $this->conn = new PDO("mysql:host={$this->host};dbname={$this->db_name}", $this->user, $this->pass);  
        $this->conn->setAttribute(PDO::ATTR_ERRMODE,  
PDO::ERRMODE_EXCEPTION);  
    } catch (PDOException $e) {  
        die("Error de conexión: " . $e->getMessage());  
    }  
    return $this->conn;  
}  
  
}
```

Esta clase es utilizada por los modelos para realizar las consultas necesarias de forma segura y eficiente.

Helpers y Librerías

En la carpeta `/helpers/` se encuentran scripts que implementan funcionalidades específicas que no corresponden directamente al modelo, vista o controlador, pero que apoyan la lógica de la aplicación. Por ejemplo, se han desarrollado funciones para la generación automática del calendario de partidos, cálculos de estadísticas, y otras operaciones recurrentes.

La carpeta `/librerias/` contiene el núcleo de la aplicación, incluyendo la gestión de la conexión a la base de datos, funciones comunes para la autenticación de usuarios, gestión de sesiones, y manejo general del sistema.

Flujo General de la Aplicación

Al recibir una solicitud, `index.php` actúa como enrutador y delega la petición al controlador correspondiente. El controlador llama a los modelos para obtener los datos de la base de datos mediante la clase `DataBase`, procesa la información y finalmente carga la vista correspondiente desde `/views/index/`. En estas vistas, se utilizan los archivos CSS y JavaScript alojados en `/public/css/` y `/public/js/` para ofrecer una experiencia visual atractiva y dinámica.

`config/configurar.php`

Archivo con la configuración principal, como datos de conexión a la base de datos y constantes globales.

`controllers/Buscar.php`

Controlador que gestiona la búsqueda de jugadores y equipos.

`controllers/Equipos.php`

Controlador encargado de las acciones relacionadas con los equipos (mostrar, editar, eliminar, etc.).

`controllers/Jugador.php`

Controlador que maneja las operaciones sobre los jugadores.

`controllers/Principal.php`

Controlador principal que gestiona la página de inicio y otras vistas generales.

`controllers/Usuario.php`

Controlador que gestiona el registro, login y administración de usuarios.

`helpers/generar_partidos.php`

Script con funciones para crear automáticamente el calendario de partidos.

`helpers/mockear_resultados.php`

Funciones para simular o generar resultados de partidos de forma automática.

`helpers/url_helper.php`

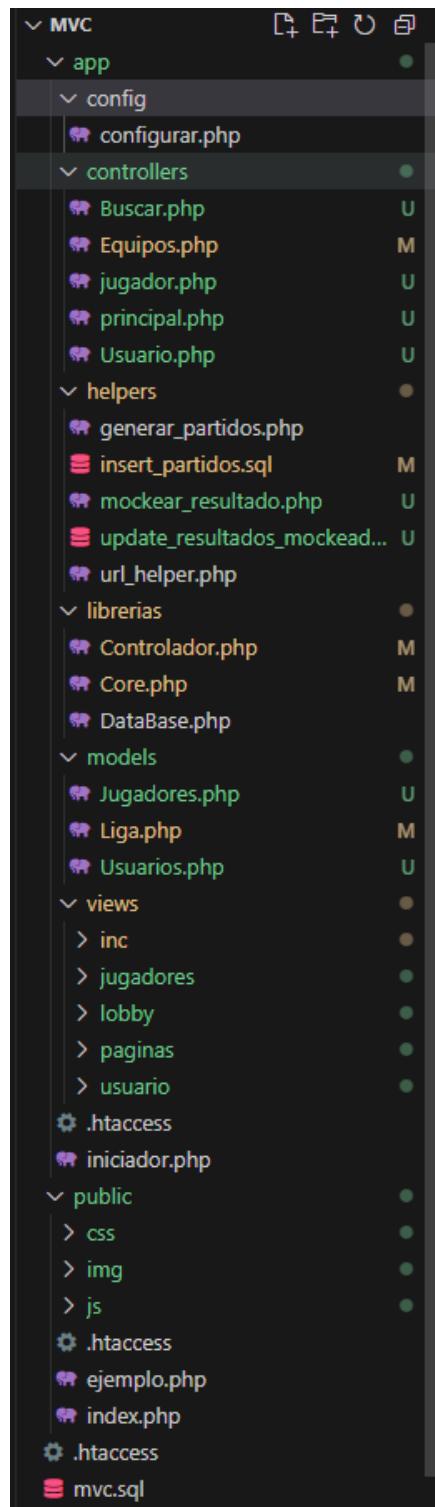
Funciones auxiliares para manejar URLs y redirectiones en la aplicación.

`librerias/controlador.php`

Clase base para todos los controladores del sistema.

`librerias/core.php`

Sistema de enrutamiento: interpreta la URL y carga el controlador y método correspondiente.



librerias/database.php

Clase que gestiona la conexión a la base de datos usando PDO.

models/Jugadores.php

Modelo que gestiona las consultas relacionadas con los jugadores.

models/Liga.php

Modelo principal para obtener datos de equipos, partidos, resultados y clasificación.

models/Usuarios.php

Modelo encargado de las operaciones sobre los usuarios (registro, login, gestión, etc.).

views/inc/footer.php

Fragmento común que contiene el pie de página de la aplicación.

views/inc/header.php

Fragmento común con el encabezado y menú de navegación.

views/jugadores/

Vistas relacionadas con la gestión y visualización de jugadores.

views/lobby/

Vista principal (inicio) con buscador, botones de acceso y secciones destacadas.

views/paginas/equipos.php

Vista que muestra todos los equipos disponibles.

views/usuario/

Vistas para el login, registro y panel de administración de usuarios.

.htaccess

Archivo de configuración de Apache que gestiona las URLs amigables y redirecciones del proyecto.

iniciador.php

Carga los archivos esenciales del sistema (configuración, librerías y helpers) al iniciar la aplicación.

public/css/

Carpeta que contiene los archivos de estilos CSS de la aplicación.

public/img/

Almacena imágenes utilizadas en la web, como escudos de equipos o iconos.

public/js/

Archivos JavaScript que añaden interactividad y funcionalidades dinámicas.

Pruebas

Para asegurar el correcto funcionamiento de la aplicación, se realizaron distintos tipos de pruebas a lo largo del desarrollo: pruebas unitarias, de integración y de usuario. Aunque no se utilizaron herramientas automatizadas de testing, sí se llevaron a cabo múltiples comprobaciones manuales y controladas desde el navegador y el entorno local (XAMPP).

Pruebas unitarias

Se probaron individualmente las funciones principales de los modelos y helpers, como por ejemplo:

- Consultas a la base de datos (obtener jugadores, equipos, resultados...).
- Generación de partidos con el script `generar_partidos.php`.
- Cálculos de resultados aleatorios con `mockear_resultados.php`.
- Funciones de enrutamiento y carga de controladores desde `core.php`.

Estas pruebas se hacían verificando los resultados esperados con datos conocidos en la base de datos y mostrando los outputs por consola o mediante mensajes temporales en la vista.

Pruebas de integración

Se comprobó que el flujo completo de la aplicación funcionaba correctamente, por ejemplo:

- Que desde la vista de equipos se pudiera acceder al detalle de un equipo y desde ahí ver sus jugadores y partidos.
- Que al registrar un usuario, los datos se guardasen correctamente y se pudiese iniciar sesión con ellos.
- Que al cambiar el rol de un usuario desde el panel de administración, el cambio se reflejara correctamente en la base de datos y en el acceso posterior.

Pruebas de usuario

Se realizaron pruebas desde la perspectiva del usuario final para comprobar:

- Navegación fluida y correcta entre secciones (inicio, equipos, jugadores, clasificación...).
- Formularios de login y registro funcionando correctamente, tanto con datos válidos como con errores.
- Comprobación de mensajes de error personalizados en caso de usuario duplicado, campos vacíos, contraseñas incorrectas, etc.

Ejemplos concretos:

- Si se intenta iniciar sesión con un usuario correcto, se redirige al área correspondiente y se establece la sesión.
- Si se introduce una contraseña incorrecta, se muestra un mensaje de error sin acceder al sistema.
- Si un usuario intenta acceder al panel de administración sin ser administrador, se le bloquea o redirige.

Pruebas de rendimiento (básicas)

Aunque no se realizaron pruebas avanzadas de rendimiento, se comprobaron tiempos de carga generales en el entorno local. Se validó que las consultas no produjeran demoras significativas y que las páginas se cargaran en un tiempo aceptable incluso con datos de prueba de múltiples jugadores y partidos.

Ampliación y posibles mejoras

Aunque el sistema desarrollado cumple con los requisitos funcionales iniciales, existen varias líneas de mejora y ampliación que podrían implementarse en futuras versiones para aumentar su robustez, funcionalidad y escalabilidad. Algunas de las más relevantes son:

1. Gestión completa de partidos

Actualmente, los partidos se generan automáticamente y se muestran en listados y calendarios, pero no hay una gestión detallada partido a partido. Una mejora importante sería implementar una **gestión individual de cada partido**, permitiendo:

- Añadir o modificar manualmente el resultado de un partido.
- Registrar alineaciones, goleadores, tarjetas, estadísticas del partido, etc.
- Validar o confirmar resultados desde un panel de administración.

2. Ampliación de privilegios de usuario

Actualmente existen roles básicos (como administrador y usuario normal), pero sería interesante **añadir nuevos roles**, como por ejemplo:

- **Moderador:** con permisos para gestionar comentarios o resultados.
- **Editor de equipos:** con acceso limitado solo a los datos de su equipo.
- **Usuario premium:** con acceso a funcionalidades extra como estadísticas avanzadas o alertas personalizadas.

Esto permitiría una gestión más granular y profesional del sistema.

3. Calendario interactivo más completo

Aunque se ha integrado un calendario básico (por ejemplo, usando FullCalendar), podría mejorarse con funciones como:

- **Filtrado por jornadas o competiciones.**
- **Notificaciones de eventos próximos.**
- **Vista semanal o diaria de partidos.**
- **Enlace directo desde el calendario al detalle del partido.**

4. Compatibilidad y diseño responsive

Aunque se ha usado Bootstrap para adaptar el diseño a distintos dispositivos, una mejora futura sería **pulir el diseño responsive** en profundidad, adaptando las vistas para que sean completamente funcionales y cómodas tanto en móviles como en tablets.

5. Mejoras de rendimiento

- **Optimización de consultas a la base de datos**, especialmente en páginas como la clasificación o los listados largos de jugadores y resultados.
- Implementación de **caché** para evitar recargar datos repetidamente.
- Posibilidad de usar AJAX para cargar ciertos contenidos sin recargar toda la página.

6. Sistema de notificaciones

Se podría añadir un sistema de **notificaciones internas** o por correo electrónico, para avisar a los usuarios de:

- Resultados de sus equipos favoritos.
- Cambios en la clasificación.
- Comienzo de nuevas jornadas.

7. Estadísticas avanzadas

Añadir un módulo con estadísticas detalladas, tanto a nivel de jugador como de equipo: goles por partido, rachas, rendimiento local/visitante, etc.

Conclusión

A lo largo del desarrollo de este Trabajo de Fin de Grado se ha conseguido crear una aplicación web funcional para la **gestión de una liga deportiva**, cumpliendo con los objetivos propuestos en las fases iniciales del proyecto. El sistema permite consultar equipos, jugadores, partidos y clasificaciones, así como gestionar usuarios mediante un sistema de login y control de roles. Todo ello siguiendo una arquitectura MVC clara y estructurada.

Principales logros y aprendizajes

Entre los principales logros se encuentran la **implementación completa del CRUD de jugadores**, la **generación automatizada de partidos**, la **visualización dinámica de resultados** y la incorporación de funcionalidades modernas como el **calendario interactivo**. Además, se ha integrado Bootstrap para facilitar un diseño adaptable y limpio, y se ha trabajado con PHP y MySQL para manejar el backend de manera eficiente.

Este proyecto ha permitido consolidar conocimientos tanto técnicos (PHP, bases de datos, MVC, sesiones, controladores...) como organizativos (estructuración del código, trabajo modular, depuración de errores).

Dificultades encontradas

Durante el desarrollo surgieron diversas dificultades, especialmente en las primeras fases del proyecto, relacionadas con la gestión de rutas, la conexión con la base de datos o el mantenimiento del patrón MVC. También fue un reto estructurar correctamente las relaciones entre jugadores, equipos y partidos, así como mantener la coherencia de los datos simulados. Sin embargo, gracias a la práctica continua y la búsqueda de soluciones, estas dificultades se superaron con éxito.

Aplicabilidad del proyecto

El sistema desarrollado tiene una clara aplicabilidad en el mundo real, ya que podría ser utilizado por **ligas deportivas amateur o escolares**, permitiendo una gestión digital sencilla y accesible. Además, el proyecto se ha construido con la posibilidad de escalar e implementar mejoras futuras, lo que lo convierte en una buena base para sistemas más complejos.

En resumen, el desarrollo de esta aplicación ha supuesto un aprendizaje significativo, ha cumplido los objetivos marcados, y representa una solución funcional y extensible en el ámbito de la gestión deportiva.

Bibliografía

Durante el desarrollo del proyecto se han utilizado diversas fuentes y recursos para implementar funcionalidades, diseño, así como para obtener imágenes y datos relacionados con el fútbol.

Documentación técnica y frameworks

- Documentación oficial de PHP: <https://www.php.net/docs.php>
- Documentación de MySQL: <https://dev.mysql.com/doc/>
- Documentación de Bootstrap 5: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
- Documentación de FullCalendar: <https://fullcalendar.io/docs>
- Mozilla Developer Network (MDN): <https://developer.mozilla.org/>
- W3Schools: <https://www.w3schools.com/>

Recursos visuales y datos de ejemplo

- Pixabay (<https://pixabay.com/es/>) – Imágenes libres de derechos utilizadas para fondos y elementos gráficos.
- Flaticon (<https://www.flaticon.com/>) – Iconos empleados en botones y cabeceras.

Fuentes deportivas para datos y estadísticas

- Transfermarkt (<https://www.transfermarkt.es/>) – Información detallada de plantillas, fichajes y estadísticas.
- ESPN Deportes (<https://www.espn.com/>) – Resultados, clasificaciones y noticias deportivas.
- Marca (<https://www.marca.com/>) – Información sobre ligas, equipos y resultados.
- Diario AS (<https://as.com/>) – Noticias y estadísticas deportivas.
- SofaScore (<https://www.sofascore.com/>) – Estadísticas en tiempo real y análisis de partidos.