

DOCUMENTACIÓN TEMBLEQUE FS LIVE

ÍNDICE

I. Introducción

- a. Presentación del proyecto
- b. Objetivos del proyecto
- c. Justificación del proyecto

II. Análisis de requerimientos

- a. Identificación de necesidades y requerimientos
- b. Identificación de público
- c. Estudio de mercado y competencia

III. Diseño y planificación

- a. Definición de la arquitectura del proyecto
- b. Diseño de la interfaz de usuario
- c. Planificación de las tareas y los recursos necesarios

IV. Implementación y pruebas

- a. Desarrollo de las funcionalidades del proyecto
- b. Pruebas unitarias y de integración
- c. Corrección de errores y optimización del rendimiento

V. Documentación

- a. Documentación técnica
- b. Documentación de usuario
- c. Manual de instalación y configuración

VI. Mantenimiento y evolución

- a. Plan de mantenimiento y soporte
- b. Identificación de posibles mejoras y evolución del proyecto
- c. Actualizaciones y mejoras futuras

VII. Conclusiones

- a. Evaluación del proyecto
- b. Cumplimiento de objetivos y requisitos
- c. Lecciones aprendidas y recomendaciones para futuros proyectos

VIII. Bibliografía y referencias

- a. Fuentes utilizadas en el proyecto
- b. Referencias y enlaces de interés

I. Introducción

a. Presentación del proyecto

El presente proyecto consiste en el desarrollo de una aplicación web orientada al equipo de fútbol sala del pueblo, con el objetivo de proporcionar una herramienta digital que combine funcionalidad informativa y social. La plataforma está diseñada para permitir la publicación de noticias relacionadas con el equipo, así como la interacción entre usuarios mediante comentarios y reacciones (me gustas) en dichas publicaciones.

La aplicación cuenta con un sistema de registro y autenticación de usuarios, diferenciando entre usuarios estándar y administradores. Estos últimos tienen acceso a funciones adicionales de gestión de contenido y usuarios. La navegación por las noticias está optimizada con filtros temáticos (como fichajes, partidos o resultados), permitiendo al usuario acceder fácilmente al contenido que más le interesa. Además, se muestran estadísticas visuales, como el número total de noticias disponibles.

El desarrollo de este proyecto integra todos los conocimientos adquiridos a lo largo del ciclo formativo, aplicando tecnologías de backend, frontend y bases de datos para crear una solución web funcional, segura y escalable.

b. Objetivos del proyecto

Los principales objetivos del proyecto son:

- **Digitalizar la presencia del equipo de fútbol sala** del pueblo mediante una plataforma web moderna y funcional.
- **Ofrecer un espacio de comunicación e interacción** donde los aficionados puedan mantenerse informados y participar activamente con comentarios y reacciones.

- **Permitir la gestión de contenido informativo**, como noticias, a través de un sistema de administración exclusivo para usuarios con rol de administrador.
- **Implementar una arquitectura escalable y modular**, que permita extender las funcionalidades en el futuro (por ejemplo, control de estadísticas de jugadores, calendario de partidos, etc.).
- **Poner en práctica los conocimientos adquiridos** en el ciclo formativo mediante la creación de un sistema completo que abarque frontend, backend y base de datos.
- **Demostrar capacidad técnica y organizativa**, generando una aplicación que pueda formar parte del portfolio profesional del alumno.

c. Justificación del proyecto

La necesidad de herramientas digitales en entornos deportivos locales es cada vez más evidente. El equipo de fútbol sala del pueblo carecía hasta ahora de una plataforma que centralizara información relevante y permitiera la interacción directa con los seguidores. La comunicación se realizaba de forma dispersa, sin una estructura clara ni un medio que fomentara la participación activa de la comunidad.

Este proyecto responde a esa carencia, ofreciendo una solución real, útil y fácilmente ampliable. Además, la diferenciación de roles (usuario y administrador) permite una gestión segura y controlada del contenido, asegurando que la información publicada esté supervisada por responsables del equipo. Esta funcionalidad, unida a la experiencia social que brinda la interacción entre usuarios, convierte la aplicación en un recurso valioso tanto para la afición como para la organización del club.

Desde el punto de vista académico, este proyecto representa la culminación del ciclo formativo, ya que abarca todos los aspectos fundamentales del desarrollo web profesional: diseño de interfaces, gestión de bases de datos, seguridad, control de acceso, desarrollo backend y experiencia de usuario. A su vez, demuestra la capacidad del alumno para llevar a cabo un proyecto real con aplicación práctica en su entorno inmediato.

II. Análisis de requerimientos

a. Identificación de necesidades y requerimientos

La necesidad principal que cubre esta aplicación web es la **falta de digitalización e información centralizada** sobre el equipo de fútbol sala del pueblo. Hasta ahora, la afición y miembros del equipo dependían de métodos poco estructurados (redes sociales o el boca a boca) para enterarse de noticias, partidos o resultados.

A través de esta aplicación, se busca:

- **Ofrecer noticias actualizadas** del equipo de forma estructurada y accesible.
- **Facilitar la interacción social** mediante comentarios y reacciones (me gustas) en las publicaciones.
- **Filtrar contenidos** según intereses (fichajes, partidos, resultados).
- **Gestionar usuarios y permisos**, estableciendo roles diferenciados entre usuarios estándar y administradores.
- **Ofrecer un entorno seguro**, donde solo usuarios registrados puedan interactuar y donde la gestión de contenidos esté controlada por administradores.

Requisitos funcionales principales:

- Registro y autenticación de usuarios.
- Roles diferenciados: USER y ADMIN.
- Publicación, lectura y filtrado de noticias.
- Interacción con publicaciones: comentarios y me gustas.
- Gestión de usuarios por parte de administradores: eliminación y cambio de rol.
- Perfil de usuario editable con posibilidad de foto.

b. Identificación del público objetivo

Aunque la aplicación es accesible para cualquier persona interesada, está especialmente pensada para la **afición del equipo**, quienes son los principales consumidores de información deportiva relacionada con el club. También podrá ser utilizada por jugadores, cuerpo técnico y personal administrativo del equipo, pero su enfoque principal es informativo y de interacción con la comunidad seguidora.

El público objetivo abarca:

- Aficionados locales del equipo.
- Familiares y amigos de los jugadores.
- Miembros del club (jugadores, entrenadores, directiva).
- Personas interesadas en seguir al equipo, aunque no vivan en el pueblo.

c. Estudio de mercado y competencia

Durante la fase de planteamiento del proyecto, se han tomado como referencia diferentes plataformas digitales ya implementadas por otros clubes deportivos, tanto a nivel local como profesional. Estas han servido de inspiración para determinar funcionalidades útiles y buenas prácticas de experiencia de usuario.

Una referencia concreta ha sido la web del **Club Deportivo Toledo** (<https://cdtoledo.es/>), que ofrece noticias, resultados, calendario y un enfoque visual claro hacia la afición. Asimismo, se han revisado páginas similares de equipos de la misma competición para identificar carencias y oportunidades de mejora en la digitalización del equipo de fútbol sala del pueblo.

Lo que diferencia a esta aplicación es su enfoque comunitario y escalabilidad: no se limita a ser un tablón informativo, sino que promueve la participación directa de los usuarios, con la posibilidad de escalar el sistema a nuevas funcionalidades como una tienda online, sistemas de socios, estadísticas y más.

III. Diseño y planificación

a. Definición de la arquitectura del proyecto

La aplicación ha sido diseñada siguiendo el patrón **Modelo–Vista–Controlador (MVC)**, un enfoque ampliamente utilizado en el desarrollo web para estructurar proyectos de forma modular, clara y mantenible.

- **Modelo:** Representado por las entidades JPA, como Usuario, Noticia, Foto, etc., y gestionado a través de los repositorios (JpaRepository). Se comunica con una base de datos MySQL y emplea Hibernate como proveedor ORM.
- **Vista:** La interfaz de usuario ha sido construida con **Thymeleaf**, que permite generar páginas dinámicas desde el backend. Se hace uso de HTML semántico, y se está desarrollando con enfoque **responsive**, utilizando elementos como <header>, <nav>, <main> y <footer> para mantener una estructura clara y accesible.

- **Controlador:** Se encarga de recibir las peticiones del usuario, interactuar con los servicios y devolver las vistas correspondientes. Los controladores están organizados por dominio, y aplican anotaciones de seguridad (`@PreAuthorize`) cuando es necesario limitar el acceso según el rol del usuario.

Además, se ha implementado seguridad mediante **Spring Security**, configurando un `SecurityFilterChain` que gestiona el acceso a las rutas según el rol (USER o ADMIN). También se emplea un sistema de autenticación basado en formularios personalizados y codificación de contraseñas con `BCryptPasswordEncoder`.

El proyecto utiliza las siguientes tecnologías y herramientas:

- **Spring Boot:** Framework principal para construir la aplicación.
- **Spring Data JPA:** Acceso a datos y persistencia.
- **Spring Security:** Gestión de autenticación y autorización.
- **Thymeleaf:** Motor de plantillas para el frontend.
- **MySQL:** Sistema de gestión de base de datos relacional.
- **Lombok:** Reducción de código repetitivo en las entidades.
- **Maven:** Gestión de dependencias y ciclo de vida del proyecto.

b. Diseño de la interfaz de usuario

El diseño de la interfaz ha sido concebido con un enfoque **responsive**, adaptándose a diferentes tamaños de pantalla y dispositivos. La navegación es clara y estructurada, con una cabecera fija, un menú de navegación principal y una sección de noticias como contenido destacado.

Se han implementado mejoras progresivas en la experiencia de usuario:

- Incorporación de filtros por **categoría de noticias** (fichajes, partidos, resultados), que permiten a los usuarios segmentar el contenido según sus intereses.
- **Paginación** en el listado de noticias para evitar una sobrecarga visual y mejorar la carga de la aplicación.
- Elementos visuales como indicadores de número total de noticias o feedback del usuario al realizar interacciones (comentarios, me gustas).
- Uso de formularios validados, incluyendo validación en el cliente y restricciones en el servidor.

El diseño general está pensado para ser intuitivo, accesible y acorde con el propósito informativo y social del proyecto.

c. Planificación de las tareas y los recursos necesarios

La planificación del proyecto se ha abordado en fases lógicas, priorizando la funcionalidad principal antes de acometer mejoras visuales y estructurales. Aunque no se ha utilizado una herramienta de gestión de tareas formal como Trello, el desarrollo ha seguido una metodología progresiva:

1. Fase 1 – Funcionalidades básicas:

- CRUDs para entidades principales: usuarios, noticias, comentarios.
- Sistema de registro y login.
- Gestión de roles y autenticación.

2. Fase 2 – Relaciones entre entidades:

- Asociación de noticias con múltiples imágenes.
- Foto de perfil para los usuarios.

3. Fase 3 – Mejora de usabilidad e interfaz:

- Implementación de filtros de noticias por categoría.
- Integración de paginación.
- Mejora visual de navegación, formularios y secciones clave.

4. Fase 4 – Seguridad y control de acceso:

- Configuración avanzada con SecurityConfig.
- Protección de rutas según rol.
- Lógica de negocio para evitar que los administradores se autodegraden o eliminen.

5. Fase 5 – Últimos ajustes:

- Preparación de la presentación.
- Ajustes de diseño responsive.
- Correcciones finales y documentación.

A pesar de la limitación temporal, se ha priorizado la calidad y funcionalidad completa sobre la cantidad de características, dejando espacio para futuras mejoras como un sistema de socios, tienda de merchandising, calendario de partidos o estadísticas del equipo.

IV. Implementación y pruebas

a. Desarrollo de las funcionalidades del proyecto

El desarrollo de la aplicación se ha abordado de manera incremental, aplicando una lógica modular. Se comenzó implementando las funcionalidades principales que garantizan el funcionamiento básico del sistema:

- **Registro y autenticación de usuarios** mediante formulario, con cifrado de contraseñas (BCrypt) y control de acceso por roles.
- **Gestión CRUD de noticias**, comentarios y me gustas.
- **Relación entre entidades**, como:
 - Noticias con múltiples imágenes.
 - Usuarios con foto de perfil.
- **Roles diferenciados** (USER y ADMIN) con restricciones específicas para cada uno: los administradores pueden editar noticias, gestionar usuarios y asignar o degradar roles.

- **Navegación avanzada** con filtros de noticias por categoría (fichajes, partidos, resultados).
- **Paginación** del listado de noticias para mejorar la carga y organización visual.

A lo largo del desarrollo, las funcionalidades eran probadas de forma iterativa e integradas al repositorio GitHub solo después de comprobar su correcto funcionamiento, respetando la coherencia general del sistema.

b. Pruebas unitarias y de integración

Aunque no se han implementado pruebas unitarias automatizadas, se ha seguido un enfoque práctico basado en pruebas manuales constantes. Cada funcionalidad nueva se validaba en el navegador, verificando que:

- Las rutas funcionasen correctamente.
- El flujo de autenticación y autorización respondiera según el rol del usuario.
- Las interacciones entre entidades fueran consistentes.
- La navegación entre páginas y filtros funcionase sin errores.
- Las validaciones del lado servidor y cliente actuaran correctamente.

En especial, se comprobó que usuarios no autenticados no pudieran interactuar con noticias, y que los administradores tuvieran acceso solo a las funciones de gestión correspondientes.

c. Corrección de errores y optimización del rendimiento

Durante el desarrollo se presentaron varios retos técnicos, principalmente relacionados con las **relaciones entre entidades**. Un ejemplo concreto fue la eliminación de una noticia, que al estar relacionada con imágenes, comentarios y me gustas, requería una gestión cuidadosa en cascada

para evitar errores y pérdida de integridad en la base de datos.

Otro problema frecuente fue el manejo de imágenes: desde la correcta carga de archivos y asignación de rutas, hasta la gestión de archivos obsoletos al actualizar una foto o eliminar una entidad relacionada.

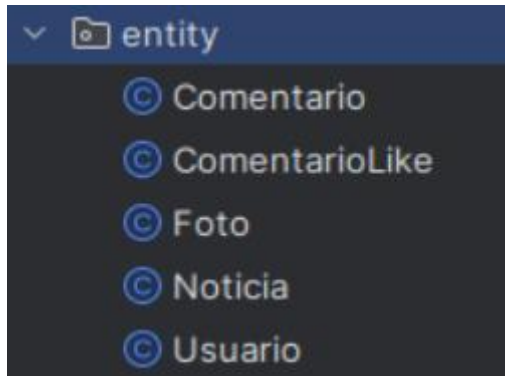
Estos problemas se abordaron con lógica específica en los controladores y servicios, y con el uso de anotaciones JPA como `cascade = CascadeType.ALL` y `orphanRemoval = true`, además de ajustes en el almacenamiento local de archivos (uploads). A través del ensayo-error y la observación detallada de excepciones, se depuraron estos comportamientos hasta alcanzar una integración estable.

En conjunto, este proceso ha permitido desarrollar una aplicación funcional, estable y con un control efectivo sobre las relaciones entre datos y la experiencia del usuario.

V. Documentación

a. Documentación técnica

La aplicación está construida sobre una arquitectura Modelo–Vista–Controlador (MVC), con una separación clara de responsabilidades. Las entidades principales son:



- Usuario: con atributos como nombre, apellidos, email, contraseña cifrada, rol y foto de perfil.
- Noticia: contiene título, contenido, fecha de publicación, categoría (como "fichajes", "partidos" o "resultados") y lista de imágenes asociadas.
- Comentario: enlazado a usuarios y noticias, permite la interacción social.
- ComentarioLike: entidad que gestiona reacciones a los comentarios.
- Foto: entidad que permite que entidades como noticias y usuarios puedan tener imágenes.

Las relaciones entre estas entidades están mapeadas mediante JPA con anotaciones como `@OneToMany`, `@ManyToOne` y `@OneToOne`, utilizando `CascadeType.ALL` y `orphanRemoval = true` para mantener la integridad referencial.

El backend está desarrollado en Java con Spring Boot, utilizando módulos como:

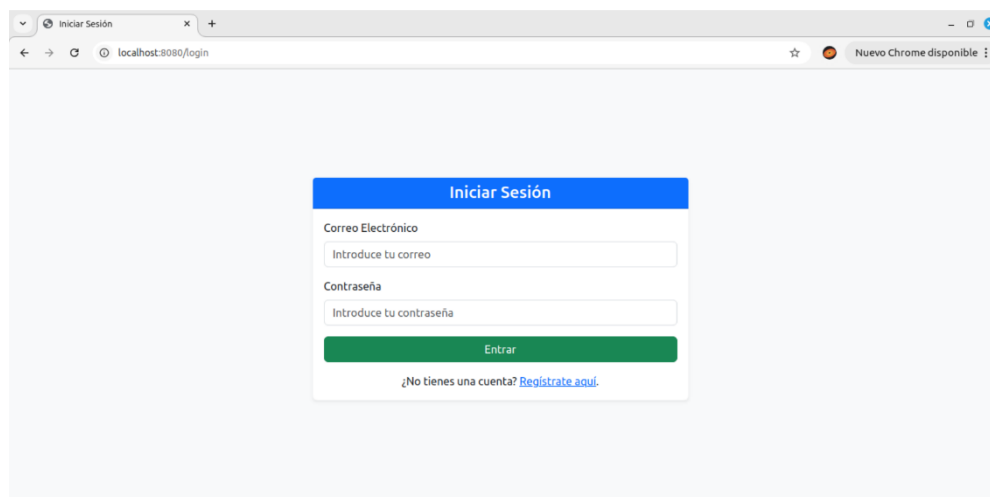
- spring-boot-starter-web (gestión de peticiones HTTP).
- spring-boot-starter-data-jpa (persistencia de datos con Hibernate).
- spring-boot-starter-security (autenticación y autorización).
- thymeleaf y thymeleaf-extras-springsecurity6 (motor de plantillas y soporte de seguridad en vistas).
- lombok (reducción de código repetitivo en entidades).

La base de datos es MySQL y las operaciones se realizan a través de `JpaRepository`. La autenticación se gestiona con Spring Security, usando `BCryptPasswordEncoder` para proteger las contraseñas. Las rutas se protegen mediante filtros definidos en la clase `SecurityConfig`.

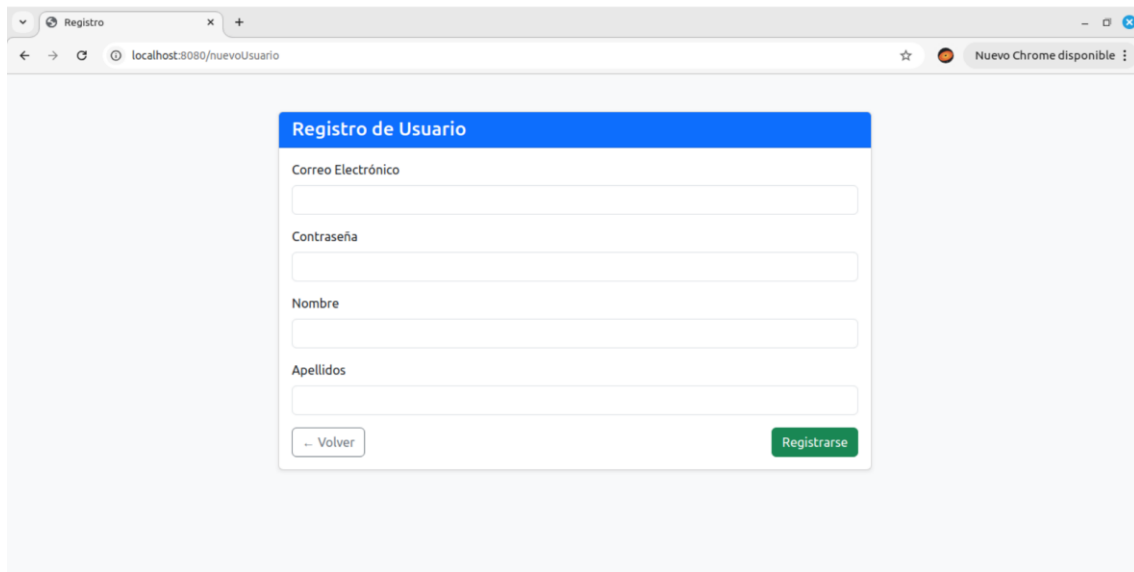
b. Documentación de usuario

Los usuarios pueden registrarse mediante un formulario accesible desde la página principal. Una vez registrados, deben iniciar sesión para poder interactuar con el contenido.

Vista Login:



Vista Registro:



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/nuevoUsuario'. The page contains a registration form titled 'Registro de Usuario' with a blue header. The form has four input fields: 'Correo Electrónico', 'Contraseña', 'Nombre', and 'Apellidos'. Below these fields are two buttons: a light blue button labeled '... Volver' and a green button labeled 'Registrarse'.

Las acciones disponibles para cada tipo de usuario son:

Usuarios estándar:

- Consultar noticias publicadas, filtrándolas por categorías.
- Navegar a través de un sistema de paginación entre las noticias listadas.



- Ver el número total de noticias publicadas (como métrica visual).



- Comentar en las noticias y dar "me gusta" a comentarios de otros usuarios.

Publicado: 29/05/2025 19:35

Comentarios

Manager: fichajazo!

29/05/2025 19:35

♥ 2

Alfredo: justo lo que necesitabamos

10/06/2025 19:35

Editar

Eliminar

♥ 0

Añadir Comentario

Escribe tu comentario...

Comentar

- Editar su perfil, incluyendo nombre, email y foto de perfil.

Mi Perfil

Nombre:

Alfredo

Apellidos:

Agudo

Email:

nuevo@gmail.com

Foto de perfil:

Seleccionar archivo

Ningún archivo seleccionado

Guardar cambios

Volver

Foto de perfil actual




Administradores (rol ADMIN):

- Publicar, editar y eliminar noticias.



- Gestionar usuarios: eliminar o cambiar roles.

ID	Nombre	Email	Rol	Acciones
1	Alfredo	nuevo@gmail.com	USER	 Hacer ADMIN  Eliminar
3	Manager	admin@gmail.com	ADMIN	 Hacer USER
4	Fernando	fertorres@gmail.com	USER	 Hacer ADMIN  Eliminar
5	Yoel	folat@gmail.com	USER	 Hacer ADMIN  Eliminar
8		nuevo2@gmail.com	USER	 Hacer ADMIN  Eliminar
9		nuevo3@gmail.com	USER	 Hacer ADMIN  Eliminar
10		a@gmail.com	USER	 Hacer ADMIN  Eliminar

[← Volver al Inicio](#)

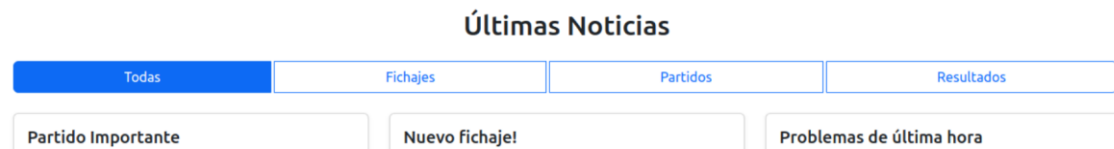
- Acceder a secciones administrativas protegidas por permisos.



Estos roles son manejados en la BD:

		id	password	apellidos	email	nombre	rol
<input type="checkbox"/>	 Editar	1	\$2a\$10\$rp1xzx6w7n01bNumQ3QN5ufjUK9mtk5w9NAz7An/TmE...	Agudo	nuevo@gmail.com	Alfredo	USER
<input type="checkbox"/>	 Editar	3	\$2a\$10\$yHcks.6kJAyg/69RnWBfQOUUnKEUucL6RmRIYWNYSvhD...	Tembleque	admin@gmail.com	Manager	ADMIN

La navegación es intuitiva y responsive. El uso de filtros permite a los usuarios seleccionar noticias por tipo (fichajes, partidos o resultados), mejorando la experiencia de usuario.



c. Manual de instalación y configuración

Para ejecutar la aplicación en un entorno local, se deben seguir los siguientes pasos:

1. Clonar el repositorio desde GitHub:

```
"git clone https://github.com/aagudom12/repositorioAgudo"
```

También se puede descargar el repositorio como archivo ZIP desde GitHub y abrirlo con un IDE como IntelliJ IDEA (recomendado) o Eclipse.

2. Importar el proyecto en IntelliJ IDEA (Community Edition o Ultimate):

- Abrir IntelliJ y seleccionar "Open" o "Import Project".
- Navegar hasta la carpeta del proyecto descargado/clonado.
- Esperar a que se resuelvan las dependencias de Maven automáticamente.

3. Crear una base de datos en MySQL:

- Abrir **phpMyAdmin** o el cliente que prefieras.
- Crear una base de datos con el nombre: **futbolbd**

4. Configurar el archivo application.properties (ya presente en el proyecto):

Asegúrate de que el archivo contiene estos parámetros:

```
spring.application.name=ProyectoDaw
spring.datasource.url=jdbc:mysql://localhost:3306/futbolbd
spring.datasource.username=root
spring.datasource.password=TU_CONTRASEÑA
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

```
spring.jpa.hibernate.ddl-auto=update
spring.jpa.database-platform=org.hibernate.dialect.MySQLDialect
```

```
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
```

```
spring.web.resources.static-locations=file:./uploads/,classpath:./static/
```

```
spring.servlet.multipart.max-file-size=10MB
spring.servlet.multipart.max-request-size=50MB
```

Estos tengo yo:

```
1  spring.application.name=ProyectoDaw
2
3  spring.datasource.url=jdbc:mysql://localhost:3306/futbolbd
4  spring.datasource.username=root
5  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
6
7  #Configuracion de JPA
8  spring.jpa.hibernate.ddl-auto=update
9  spring.jpa.database-platform=org.hibernate.dialect.MySQLDialect
10
11 #Para ver las SQL que se ejecutan poner las siguientes (opcional)
12 spring.jpa.show-sql=true
13 spring.jpa.properties.hibernate.format_sql=true
14
15 #para que podamos acceder a los archivos de las carpetas, además de hacerlas públicas
16 spring.web.resources.static-locations=file:./uploads/,classpath:./static/
17
18 #Tamaño máximo por archivo de forma global para toda la App, si se excede salta MaxUploadSizeExceededException
19 spring.servlet.multipart.max-file-size=10MB
20 #Tamaño máximo de cada petición (por ejemplo varios archivos en un mismo formulario)
21 spring.servlet.multipart.max-request-size=50MB
```

IMPORTANTE: Cambia TU_CONTRASEÑA por la contraseña real de tu usuario root de MySQL.

5. Ejecutar la aplicación:

- Desde IntelliJ, busca la clase principal con la anotación @SpringBootApplication y ejecuta.
- O desde terminal, en la raíz del proyecto:

```
mvn spring-boot:run
```

6. Acceder a la aplicación:

- Abrir el navegador en <http://localhost:8080>.

7. Notas adicionales:

- La aplicación creará automáticamente todas las tablas necesarias en la base de datos.
- Los recursos estáticos se almacenarán en la carpeta /uploads. Se recomienda crearla manualmente en la raíz del proyecto si vas a trabajar con imágenes u otros ficheros multimedia.

VI. Mantenimiento y evolución

a. Plan de mantenimiento y soporte

El proyecto contempla un enfoque de mantenimiento continuo, centrado en asegurar la estabilidad, funcionalidad y seguridad de la aplicación. Una vez desplegada, se establecerá un **canal de comunicación abierto** para que los usuarios puedan reportar incidencias, errores o sugerencias de mejora.

Las tareas de mantenimiento incluirán:

- Corrección de errores detectados por los usuarios.
- Refactorización de código para mejorar el rendimiento o la organización interna.
- Actualización de dependencias y tecnologías en caso necesario.

- Mejora progresiva de la interfaz y la experiencia de usuario.

El sistema está preparado para permitir modificaciones y nuevas implementaciones sin necesidad de reestructurar completamente la arquitectura.

b. Identificación de posibles mejoras y evolución del proyecto

Aunque la versión actual de la aplicación cubre los requisitos funcionales básicos, existen múltiples áreas en las que se puede evolucionar el sistema a medio y largo plazo. Entre las mejoras identificadas se encuentran:

- **Calendario interactivo de partidos:** permitiría a los usuarios consultar las fechas y horas de los próximos encuentros, así como resultados anteriores.
- **Gestión de categorías:** incorporar secciones específicas para los diferentes equipos del club (como federados, alevines, etc.), permitiendo segmentar noticias y contenido.
- **Clasificación y estadísticas:** implementar un sistema que muestre la posición del equipo en la competición, resultados acumulados, goles, victorias, empates y derrotas.
- **Tienda online:** añadir una sección de ecommerce donde los aficionados puedan comprar merchandising oficial del equipo (camisetas, bufandas, etc.).
- **Sistema de socios:** permitir que los usuarios puedan adquirir su bono desde la aplicación, digitalizarlo, y utilizarlo como credencial de acceso a partidos y eventos.

Estas funcionalidades serían implementadas progresivamente, en función de las prioridades que establezca la directiva del equipo y de los recursos disponibles.

c. Actualizaciones y mejoras futuras

El proyecto se concibe como una aplicación escalable, por lo que se irán aplicando **actualizaciones periódicas** para corregir errores, mejorar el rendimiento y añadir nuevas características. El ritmo de implementación dependerá de:

- La demanda por parte de los usuarios.
- Las decisiones estratégicas del equipo directivo del club.
- La viabilidad técnica y el tiempo de desarrollo estimado para cada mejora.

Se priorizarán aquellas funciones que generen un mayor impacto positivo en la experiencia del usuario y en la gestión interna del equipo.

VII. Conclusiones

a. Evaluación del proyecto

El desarrollo de esta aplicación web ha supuesto un ejercicio completo e integrador de los conocimientos adquiridos a lo largo del ciclo formativo. El proyecto cumple con los requisitos establecidos para un Trabajo Fin de Ciclo, ofreciendo una solución funcional, segura y orientada a un caso real, como es la digitalización del equipo de fútbol sala del pueblo.

Durante su desarrollo, se han abordado aspectos clave del desarrollo web como la autenticación de usuarios, el control de acceso por roles, la gestión de entidades relacionadas, el tratamiento de archivos (imágenes), y la organización estructural bajo el patrón MVC. Además, se han añadido elementos de usabilidad y experiencia de usuario como filtros, paginación y estadísticas visuales, enriqueciendo notablemente la funcionalidad inicial planteada.

b. Cumplimiento de objetivos y requisitos

El proyecto ha cumplido la mayoría de los objetivos planteados inicialmente, destacando especialmente aquellos relacionados con la parte informativa y social de la aplicación. Si bien uno de los objetivos iniciales —permitir que los usuarios se hicieran socios mediante la app— no ha sido implementado, se considera una funcionalidad viable para futuras versiones. La decisión de posponerla responde a una evaluación realista del alcance y la complejidad técnica asociada, especialmente en lo referente a pagos e integración de sistemas de compra.

A cambio, se han desarrollado funcionalidades no contempladas en la planificación inicial, como la posibilidad de comentar y reaccionar a los comentarios con me gustas, aportando un valor añadido significativo al componente social de la aplicación.

En términos generales, el proyecto no solo cumple con los requisitos mínimos exigidos, sino que incorpora elementos que lo sitúan por encima del estándar esperado.

c. Lecciones aprendidas y recomendaciones para futuros proyectos

El desarrollo de este proyecto ha permitido consolidar habilidades técnicas y organizativas. Se ha trabajado de forma autónoma, enfrentando y resolviendo errores de integración entre entidades, problemas con la gestión de archivos y retos comunes en el desarrollo web con Spring Boot.

Una de las lecciones más importantes ha sido el valor de la planificación y la constancia. Aunque el proyecto ha llegado a buen término y ha cumplido con sus objetivos, una mayor regularidad en el trabajo diario habría permitido implementar más funcionalidades y ampliar el alcance del proyecto.

En conclusión, la experiencia ha sido enriquecedora tanto a nivel técnico como personal. La aplicación desarrollada no solo representa una prueba sólida de los conocimientos adquiridos, sino también una herramienta real que puede seguir evolucionando y creciendo en el futuro.

VIII. Bibliografía y referencias

a. Fuentes utilizadas en el proyecto

- **Spring Boot Documentation** – <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
Documentación oficial utilizada como guía para la configuración de seguridad, dependencias y estructura del proyecto.
- **Thymeleaf Documentation** – <https://www.thymeleaf.org/documentation.html>
Referencia utilizada para implementar las vistas dinámicas y trabajar con plantillas HTML.
- **JPA/Hibernate Documentation** – <https://hibernate.org/orm/documentation/>
Base consultada para entender las relaciones entre entidades, anotaciones y comportamiento en cascada.
- **MySQL Documentation** – <https://dev.mysql.com/doc/>
Para la configuración y uso correcto del gestor de base de datos.
- **Club Deportivo Toledo - Web oficial** – <https://cdtoledo.es/>
Página observada como ejemplo de presentación de contenido deportivo digital y estructura general de noticias y comunicación de un club.
- **Baeldung – Spring tutorials** – <https://www.baeldung.com/>
Sitio web con artículos y tutoriales avanzados de Spring Boot y seguridad, útil para resolver problemas puntuales durante el desarrollo.

b. Referencias y enlaces de interés

- **Stack Overflow** – <https://stackoverflow.com/>
Utilizado como fuente de soluciones a errores técnicos específicos durante el desarrollo.
- **Maven Repository** – <https://mvnrepository.com/>
Para la consulta e inclusión de dependencias necesarias en el proyecto.
- **MDN Web Docs** – <https://developer.mozilla.org/>
Referencia utilizada para aspectos de HTML, CSS y JavaScript.