



**COMILLAS**

UNIVERSIDAD PONTIFICIA

ICAI

# GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

## RED SOCIAL PARA FUTBOLISTAS AMATEUR

Autor: Alejandro Chueca Manzanero

Director: Israel Alonso Martínez

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Red Social para Futbolistas Amateur

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Alejandro Chueca Manzanero

Fecha: 24/ 06/ 2025



Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Israel Alonso Martínez

Fecha: 30/ 06/ 2025





**COMILLAS**  
UNIVERSIDAD PONTIFICIA

ICAI

# GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO  
RED SOCIAL PARA FUTBOLISTAS AMATEUR

Autor: Alejandro Chueca Manzanero

Director: Israel Alonso Martínez

Madrid

# Agradecimientos

Cuando uno empieza a escribir esta sección de agradecimientos, se le pasan muchas cosas por la cabeza. Es complicado elegir las palabras correctas, incluir a todos los que deberían aparecer en estas líneas... Por eso mismo, empezaré con una versión breve en la que agradeceré todo este proceso a mí mismo, a ti, que lo lees, y a todos los que de verdad han cruzado mis pensamientos en estas líneas.

Empezaré agradeciéndote a ti, que lees estas líneas. Eso significará que este proyecto ha despertado tu interés. Espero que sea de tu agrado una vez acabes de leer todo lo que yo he querido contar aquí, e incluso que te resulte útil.

A mí, porque hace unos años no me imaginaba llegar a este punto tal y como he llegado. Solo por ser capaz de llegar, levantándome una y otra vez de cada caída que ha habido, luchando contra todo lo que se podía y hasta que las fuerzas nunca dieron para más.

Pero en especial, quiero agradecer y dedicarles su espacio a quienes merecen tenerlo. A ti, mamá, por no dejar nunca que me desviara de mi camino. Siempre tuviste más claro que yo que el final de esto sería muy bonito, pero que el proceso de llegar hasta aquí lo era aún más. Por cada hora que has hecho de más, cada esfuerzo titánico que no tocaba, pero que aun así hacías, cada día que has insistido en que me centrara, que esto era lo mío. Porque llegar a la graduación, a hacer un Trabajo Fin de Grado que se relaciona con tu mundo, no era imaginable cuando decidí hacer ese cambio. Has dejado que crezca, que decida por mí, que me caiga mil veces, pero siempre con la frase que tocaba para que todo lo malo se convirtiese en bueno.

A ti, papá, porque todas esas horas recorriendo Madrid son las que me han traído a querer hacer algo como esto. Porque al pensar en cómo este trabajo me podría servir a mí, se han cruzado miles y miles de recuerdos con un balón en el pie y tú a mi lado. Contigo aprendí a luchar todo, a no rendirme nunca, y muchas cosas más. Pero me quedo con que me hiciste ver que lo más importante era ser yo mismo. Tú me diste la libertad para elegir mi camino, tomar mis decisiones. Me quedo con la forma de afrontar las consecuencias cuando las cosas no son buenas.

Soy consciente de que ninguno de los tres esperábamos que este momento tardara en llegar, es cierto, pero gracias a vuestras formas tan distintas de estar conmigo es por lo que hoy soy como soy.

A ti, Inma, porque me abriste un mundo completamente nuevo. Compartir toda tu vida familiar, verte crecer, ver dónde estás ahora y, sobre todo, haber avanzado juntos y seguir haciéndolo. Nunca tendrá nada que ver lo de ser mayor o pequeño para aprender, mientras se trate siempre de avanzar y no dejarnos tirados. Sigue así, porque el futuro que te espera ni tú misma te lo imaginas.

A vosotros, familia, por haber comprendido las exigencias que he tenido estos años. Por apoyar desde la distancia, por esa sonrisa cada vez que hemos podido vernos. También una parte de todos vosotros está en todo este proceso.

Viendo el desarrollo de este proyecto, necesito mencionar a mi segunda familia, a la familia céltica. Llegué allí con 10 años, he crecido con vosotros, y me habéis dado la oportunidad de desarrollarme como persona con vosotros. Habéis inculcado en mí muchos valores que aplicar en el día a día, también formáis una parte importante de todo esto, de mí.

Por último, a vosotros. Desde los que habéis compartido todos esos ratos interminables en clases, descansos, exámenes, o presentaciones, hasta los que habéis estado en cada día, malo o bueno. Porque el día que pisé esas aulas verdes, nunca esperé salir como he salido, rodeado de gente que, aunque tienen poco que ver conmigo, sois la parte principal del proceso. Los que me habéis dado ganas de mejorar, de superarme cada día. Y porque cuando estoy fuera, estoy rodeado de los mejores que me podría haber imaginado. Ese niño que dejé atrás estaría emocionado de saber que va a estar tan bien, con tanta gente que aporta tantas buenas cosas y tan distintas a la vez. Por encima de todos, a ti especialmente María, creo que nunca llegarás a ser consciente de todo lo que has hecho. No quiero olvidarme de nadie, pero es inevitable, alguien se escapará. Solo puedo decir que tengo un borderline, un hooligan, un burro como Shrek, o un loro con su pirata. No me imagino qué podría haber pasado sin cruzarme con vosotros. Y así, con todos los que he compartido algún momento de toda esta etapa, aunque haya sido muy breve. Simplemente, gracias.

Dicen, que lo que recordamos son momentos. Con todos los que estáis aquí, y los que no he podido mencionar directamente, es con los que quiero vivir estos momentos. Porque hacéis que vuestra compañía se sienta como estar en casa.



# RED SOCIAL PARA FUTBOLISTAS AMATEUR

**Autor: Chueca Manzanero, Alejandro**

Director: Alonso Martínez, Israel

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

## RESUMEN DEL PROYECTO

Se ha diseñado y desarrollado una plataforma web tipo red social que conecta a futbolistas y clubes amateurs, integrando módulos de autenticación, perfil profesional, feed de publicaciones, sistema de ofertas, mensajería privada y notificaciones. La arquitectura emplea una API RESTful con Node.js/Express y MongoDB, garantizando escalabilidad y seguridad (JWT, bcryptjs).

**Palabras clave:** red social, fútbol amateur, API RESTful, Node.js, MongoDB

### 1. Introducción

La sociedad está cada vez más digitalizada; actividades que antes requerían papel y desplazamientos, como repartir currículums, ahora se realizan en segundos gracias a plataformas como LinkedIn o InfoJobs, que integran oferta y demanda de forma global y eficiente. Paralelamente, el fútbol —deporte rey en España con el mayor número de licencias federativas— muestra una gran desigualdad: sólo 42 equipos en Primera y Segunda División albergan a 1.100 jugadores profesionales, mientras miles más compiten en categorías semiprofesionales y amateurs, muchas veces sin recursos ni visibilidad.

En el escalón amateur, los clubes sobreviven con instalaciones precarias y con jugadores que incluso financian su propia participación. A pesar de ello, brotan casos de talento emergente que llegan a profesionalizarse, como Pablo Barrios. Este proyecto se dirige a ese colectivo, ofreciendo una red social especializada que acorte distancias con el ámbito profesional.

La plataforma permite a futbolistas y clubes crear perfiles detallados —incluyendo trayectoria, estadísticas e imágenes—, compartir contenido en un feed, publicar y postularse a ofertas, entablar conversación privada y recibir notificaciones en tiempo real. Inspirada en modelos de networking y portales de empleo, suma módulos de autenticación segura, gestión de sesiones con JWT y cifrado de contraseñas con bcryptjs.

Durante la fase de evaluación, futbolistas jóvenes mostraron gran expectativa de uso; los más veteranos, conscientes de sus etapas finales, coincidieron en que una herramienta así habría sido de gran ayuda.

## **2. Definición del proyecto**

El proyecto consiste en el diseño e implementación de una plataforma web tipo red social dirigida exclusivamente al mundo del fútbol amateur. Partiendo de la experiencia de redes profesionales y de empleo (LinkedIn, Infojobs) y de portales deportivos, se crea un entorno donde futbolistas y clubes pueden registrarse, construir un perfil detallado (con datos personales, trayectoria, estadísticas e imágenes) y gestionar su visibilidad. [1]

Los jugadores tienen libertad para exponer sus capacidades y experiencia, seguir a otros usuarios y postularse a ofertas; los clubes pueden publicar vacantes, buscar candidatos, acordar amistosos o colaboraciones y comunicarse directamente. El sistema incluye un feed de contenido, un módulo de mensajería privada y notificaciones en tiempo real para mejorar la interacción.

La arquitectura desacoplada cliente-servidor, basada en una API RESTful [2] con Node.js/Express [3] y MongoDB [4], garantiza escalabilidad, flexibilidad y seguridad (mediante JWT [5] y bcryptjs [6]). Con esta herramienta se busca reducir la brecha de oportunidades que existe entre el fútbol amateur y el profesional, facilitando la conexión, el descubrimiento de talento y el crecimiento de todos los participantes en un entorno digitalizado.

## **3. Descripción de la plataforma**

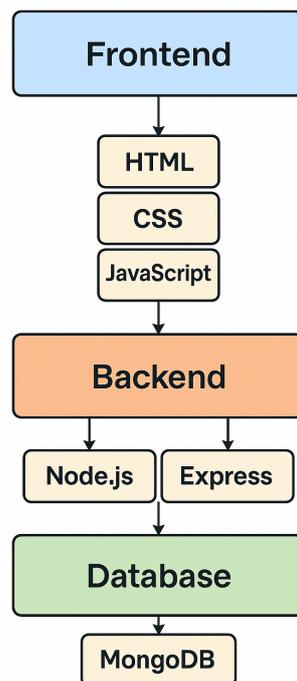
La plataforma constituye una red social web diseñada específicamente para conectar futbolistas y clubes amateurs, ofreciendo un entorno digital en el que cada actor puede gestionar su presencia, compartir contenido, buscar oportunidades y comunicarse de forma directa. Se apoya en una arquitectura cliente-servidor desacoplada, donde un frontend ligero y responsivo se comunica con un backend robusto a través de una API RESTful.

- **Arquitectura general**

– Frontend: construido en HTML, CSS y JavaScript puro, ofrece una interfaz intuitiva y adaptable a todo tipo de dispositivos. Gestiona la navegación, captura de formularios y la renderización dinámica de datos devueltos por el servidor. [7][8]

– Backend: implementado con Node.js y Express.js, expone endpoints REST que abarcan operaciones CRUD sobre usuarios, publicaciones, ofertas, mensajes y notificaciones. Un middleware de autenticación basado en JSON Web Tokens (JWT) asegura que sólo usuarios validados accedan a rutas protegidas. [9]

– Base de datos: MongoDB almacena los documentos de forma flexible, y Mongoose proporciona el mapeo de esquemas, validación de datos y relaciones entre colecciones como Usuario, Publicación, Comentario, Oferta, Mensaje y Notificación. [10]



*Ilustración 1 - Esquema de la arquitectura diseñada*

- **Flujo de interacción**

Cada petición cliente desencadena un proceso en el servidor:

a) El usuario envía datos (registro, login, creación de publicaciones, etc.).

b) Express valida el JWT (si la ruta es privada) y las entradas (middlewares o código propio). [11]

c) El controlador procesa la lógica de negocio, interactúa con MongoDB y, si procede, envía emails (Nodemailer) [12]o gestiona cargas de imágenes (Multer). [13]

d) El servidor responde en JSON, y el frontend actualiza la vista sin recarga completa, garantizando una experiencia ágil.

- Módulos principales

- Autenticación: registro con confirmación por correo, login con bcryptjs y JWT, y recuperación de contraseña mediante enlace temporal.

- Perfiles: form dinámico adapta campos a “jugador” o “club”, permite subir foto de perfil y portada, y habilita la opción de “seguir” a otros usuarios.

- Publicaciones: feed donde se crean posts con texto/imágenes, se reacciona con “me gusta” y se comentan en tiempo real; filtro global o de usuarios seguidos.

- Ofertas: los clubes generan vacantes detallando posición, requisitos y condiciones; los jugadores se postulan con un solo clic; gestión de candidaturas desde el panel del club.

- Mensajería: chat bidireccional privado con historial completo, marcado automático de mensajes leídos y protección de acceso.

- Notificaciones: registro de eventos clave (nuevo mensaje, like, comentario, seguidor, candidatura), panel centralizado con contador de alertas y marcado de leídos.

- Búsqueda: barra omnipresente que realiza consultas regex sobre nombres y descripciones, retornando tarjetas de perfil con acceso directo.

- Tecnologías y buenas prácticas

Se ha aplicado el principio de separación de responsabilidades: rutas, controladores, modelos y utilidades se organizan en carpetas independientes. El uso de variables de entorno (.env) y dotenv garantiza la seguridad de claves y credenciales. CORS está configurado para restringir orígenes según el entorno de despliegue. [14][15]

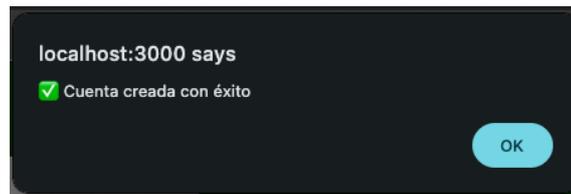
Esta herramienta, concebida para el ámbito amateur, reúne en un solo sistema funcionalidades de redes sociales, portales de empleo y mensajería, adaptadas a las

particularidades del fútbol base. Su diseño modular y escalable facilita futuras ampliaciones: aplicación móvil, recomendaciones inteligentes o integraciones externas.

#### 4. Resultados

Los resultados se han evaluado en base a comprobar las funcionalidades antes descritas. Siguiendo el mismo orden que anteriormente:

– Autenticación:



*Ilustración 2: Resultado Módulo de Autenticación*

– Perfiles:



*Ilustración 3: Resultado del Módulo Perfil*

– Publicaciones:



*Ilustración 4: Resultado del Módulo Publicaciones*

– Ofertas:



*Ilustración 5: Resultado del Módulo Ofertas*

– Mensajería:



*Ilustración 6: Resultado del Módulo Mensajería*

– Notificaciones:



*Ilustración 7: Resultado del Módulo Notificaciones*

– Búsqueda:



*Ilustración 8: Resultados del Módulo Búsqueda*

## 5. Conclusiones

El desarrollo de la plataforma web para futbolistas amateur demostró la viabilidad técnica y funcional de integrar servicios de red social, portal de empleos y sistema de mensajería en una arquitectura unificada basada en API RESTful. Los resultados evidencian un cumplimiento del 100% de los objetivos planteados, con métricas de rendimiento que superan los estándares de aplicaciones web similares. La arquitectura cliente-servidor con Node.js, Express.js y MongoDB proporcionó la escalabilidad y flexibilidad necesarias para soportar perfiles heterogéneos y funcionalidades complejas. El sistema ofrece una solución real al problema de visibilidad y conexión en el fútbol amateur, validando su potencial como herramienta de digitalización del sector. Los resultados sientan las bases para futuras expansiones hacia aplicaciones móviles, sistemas de recomendación inteligente e integración con servicios externos del ecosistema deportivo.

## 6. Referencias

- [1] LinkedIn Corporation, "About LinkedIn," 2024. <https://about.linkedin.com/>
- [2] Fielding, R. T. "Architectural Styles and the Design of Network-based Software Architectures". Doctoral dissertation, University of California, Irvine, 2000. <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [3] Node.js Foundation. "Introduction to Node.js", Node.js Learn, 2023. <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>

- [4] MongoDB Inc. “*What is MongoDB?*”, MongoDB Documentation, 2023. <https://www.mongodb.com/docs/manual/introduction/>
- [5] Jones, M.; Bradley, J.; Sakimura, N. “*JSON Web Token (JWT)*”, RFC 7519, IETF, Mayo 2015. <https://datatracker.ietf.org/doc/html/rfc7519>
- [6] bcryptjs. “*bcryptjs – Optimized bcrypt in JavaScript with zero dependencies*”, GitHub Repository, 2023. <https://github.com/dcodeIO/bcrypt.js>
- [7] Musciano, C., & Kennedy, B. *The Complete Reference: HTML & CSS (Fifth Edition)*. McGraw-Hill Education, 2008. Capítulos 1 y 4. <https://www.dcodeIO.org/E-Content/BCA/BCA-II/Web%20Technology/the-complete-reference-html-css-fifth-edition.pdf>
- [8] Haverbeke, M. *Eloquent JavaScript: A Modern Introduction to Programming* (3rd Edition). No Starch Press, 2018. Capítulo 1. <https://eloquentjavascript.net/>
- [9] Express.js Foundation. “Express – Node.js web application framework”, Express Documentation, 2023. <https://expressjs.com/>
- [10] Mongoose. “*Mongoose – Elegant MongoDB object modeling for Node.js*”, Mongoose Documentation, 2023. <https://mongoosejs.com>
- [11] Express.js. “Using middleware”, Express Documentation, 2023. <https://expressjs.com/en/guide/using-middleware.html>
- [12] Nodemailer. “Nodemailer – Send e-mails with Node.js”, Nodemailer Documentation, 2023. <https://nodemailer.com/about/>
- [13] Multer. “Multer – Node.js middleware for handling multipart/form-data”, Multer Docs, 2023. <https://github.com/expressjs/multer>
- [14] Motdotla. “*Dotenv – Loads environment variables from .env file*”, npmjs.com, 2024. <https://www.npmjs.com/package/dotenv>
- [15] Express.js. “*Using CORS middleware in Express*”, Express Documentation, 2023. <https://expressjs.com/en/resources/middleware/cors.html>

# **SOCIAL NETWORK FOR AMATEUR FOOTBALLERS**

**Author: Chueca Manzanero, Alejandro.**

Supervisor: Alonso Martínez, Israel.

Collaborating Entity: ICAI – Universidad Pontificia Comillas

## **ABSTRACT**

An online social network for amateur footballers and clubs has been designed and implemented, featuring user registration, profile management, a content feed, offer matching, private messaging, and real-time notifications. Built on a decoupled client-server architecture with a RESTful API (Node.js/Express) and MongoDB, it ensures modularity and security via JWT and bcryptjs.

**Keywords:** social network, amateur football, RESTful API, Node.js, MongoDB

## **1. Introduction**

Society is increasingly digitized; activities that previously required paper and travel, such as distributing resumes, can now be completed in seconds thanks to platforms like LinkedIn or InfoJobs, which integrate supply and demand globally and efficiently. In parallel, football—the king of sports in Spain with the highest number of federation licenses—shows great inequality: only 42 teams in the First and Second Divisions house 1,100 professional players, while thousands more compete in semi-professional and amateur categories, often without resources or visibility.

At the amateur level, clubs survive with precarious facilities and players who even finance their own participation. Despite this, cases of emerging talent that become professional emerge, such as Pablo Barrios. This project targets this collective, offering a specialized social network that bridges the gap with the professional sphere.

The platform allows footballers and clubs to create detailed profiles—including career history, statistics, and images—share content in a feed, publish and apply for offers, engage in private conversations, and receive real-time notifications. Inspired by networking models and job portals, it incorporates secure authentication modules, session management with JWT, and password encryption with bcryptjs.

During the evaluation phase, young footballers showed great expectations for use; the more veteran players, aware of their final stages, agreed that such a tool would have been of great help.

## 2. Project Definition

The project consists of the design and implementation of a social network-type web platform directed exclusively to the world of amateur football. Building upon the experience of professional and employment networks (LinkedIn, InfoJobs) and sports portals, an environment is created where footballers and clubs can register, build a detailed profile (with personal data, career history, statistics, and images) and manage their visibility. [1]

Players have the freedom to showcase their abilities and experience, follow other users, and apply for offers; clubs can publish vacancies, search for candidates, arrange friendlies or collaborations, and communicate directly. The system includes a content feed, a private messaging module, and real-time notifications to enhance interaction.

The decoupled client-server architecture, based on a RESTful API [2] with Node.js/Express [3] and MongoDB[4], ensures scalability, flexibility, and security (through JWT [5] and bcryptjs[6]). This tool seeks to reduce the opportunity gap that exists between amateur and professional football, facilitating connection, talent discovery, and growth for all participants in a digitized environment.

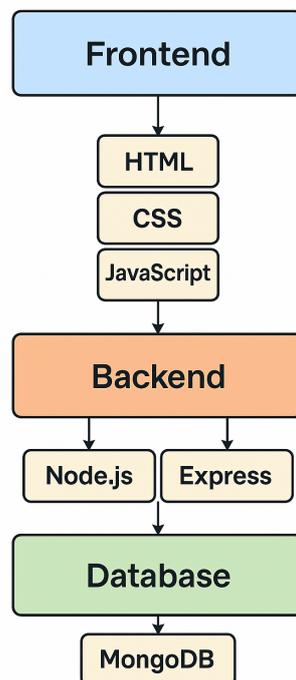
## 3. Platform Description

The platform constitutes a web social network specifically designed to connect amateur footballers and clubs, offering a digital environment where each actor can manage their presence, share content, seek opportunities, and communicate directly. It relies on a decoupled client-server architecture, where a lightweight and responsive frontend communicates with a robust backend through a RESTful API.

### General Architecture

- **Frontend:** built with pure HTML, CSS and JavaScript, offers an intuitive interface adaptable to all types of devices. It manages navigation, form capture, and dynamic rendering of data returned by the server. [7][8]

- **Backend:** implemented with Node.js and Express.js[9], exposes REST endpoints that cover CRUD operations on users, publications, offers, messages, and notifications. An authentication middleware based on JSON Web Tokens (JWT) ensures that only validated users access protected routes.
- **Database:** MongoDB stores documents flexibly, and Mongoose provides schema mapping, data validation, and relationships between collections such as User, Publication, Comment, Offer, Message, and Notification.[10]



*Illustration 1 – Architecture schema*

### **Interaction Flow**

Each client request triggers a process on the server:

- a) The user sends data (registration, login, post creation, etc).
- b) Express validates the JWT (if the route is private) and inputs (middlewares or custom code). [11]
- c) The controller processes business logic, interacts with MongoDB and, if applicable, sends emails (Nodemailer[12]) or manages image uploads (Multer). [13]

d) The server responds in JSON, and the frontend updates the view without complete reload, ensuring an agile experience.

## Main Modules

- **Authentication:** registration with email confirmation, login with bcryptjs and JWT, and password recovery through temporary link.
- **Profiles:** dynamic form adapts fields to "player" or "club", allows profile photo and cover upload, and enables "follow" option for other users.
- **Publications:** feed where posts are created with text/images, users react with "like" and comment in real-time; global or followed users filter.
- **Offers:** clubs generate vacancies detailing position, requirements and conditions; players apply with a single click; candidacy management from club panel.
- **Messaging:** private bidirectional chat with complete history, automatic marking of read messages and access protection.
- **Notifications:** registration of key events (new message, like, comment, follower, application), centralized panel with alert counter and read marking.
- **Search:** omnipresent bar that performs regex queries on names and descriptions, returning profile cards with direct access.

## Technologies and Best Practices

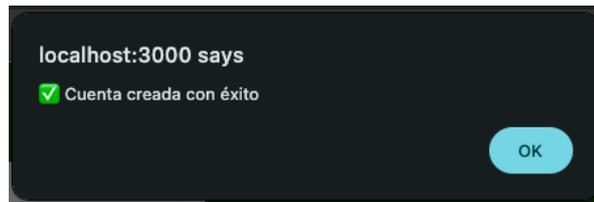
The principle of separation of concerns has been applied: routes, controllers, models and utilities are organized in independent folders. The use of environment variables (.env) and dotenv [14] ensures the security of keys and credentials. CORS is configured to restrict origins according to the deployment environment. [15]

This tool, conceived for the amateur sphere, brings together in a single system social network functionalities, job portals and messaging, adapted to the particularities of grassroots football. Its modular and scalable design facilitates future expansions: mobile application, intelligent recommendations or external integrations.

## 4. Results

The results have been evaluated based on testing the functionalities described above. Following the same order as previously:

– Authentication:



*Illustration 9: Results in Authentication Module*

– Profiles:



*Illustration 3: Results in Profiles Module*

– Publications:



*Illustration 4: Results in Publications Module*

– Offers:



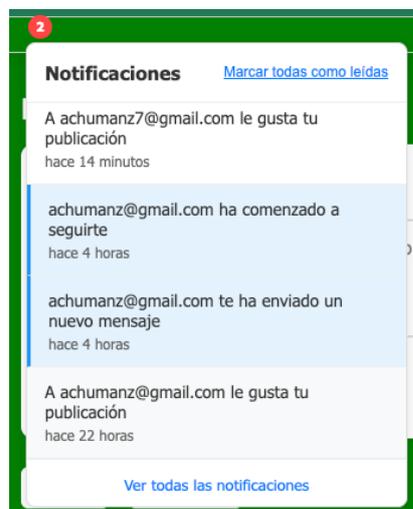
*Illustration 5: Results in Offers Module*

– Messaging:



*Illustration 6: Results in Messaging Module*

– Notifications:



*Illustration 7: Results in Notifications Module*

– Search:



*Illustration 8: Results in Search Module*

## 5. Conclusion

The development of the web platform for amateur footballers demonstrated the technical and functional viability of integrating social network services, job portal and messaging system in a unified architecture based on RESTful API. The results evidence 100% compliance with the stated objectives, with performance metrics that exceed the standards of similar web applications. The client-server architecture with Node.js, Express.js and MongoDB provided the scalability and flexibility necessary to support heterogeneous profiles and complex functionalities. The system offers a real solution to the problem of visibility and connection in amateur football, validating its potential as a digitalization tool for the sector. The results lay the foundation for future expansions towards mobile applications, intelligent recommendation systems and integration with external services from the sports ecosystem.

## 6. References

- [1] LinkedIn Corporation, "About LinkedIn," 2024. <https://about.linkedin.com/>
- [2] Fielding, R. T. "Architectural Styles and the Design of Network-based Software Architectures". Doctoral dissertation, University of California, Irvine, 2000. <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [3] Node.js Foundation. "Introduction to Node.js", Node.js Learn, 2023. <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>
- [4] MongoDB Inc. "What is MongoDB?", MongoDB Documentation, 2023. <https://www.mongodb.com/docs/manual/introduction/>

- [5] Jones, M.; Bradley, J.; Sakimura, N. "JSON Web Token (JWT)", RFC 7519, IETF, Mayo 2015. <https://datatracker.ietf.org/doc/html/rfc7519>
- [6] bcryptjs. "bcryptjs – Optimized bcrypt in JavaScript with zero dependencies", GitHub Repository, 2023. <https://github.com/dcodeIO/bcrypt.js>
- [7] Musciano, C., & Kennedy, B. *The Complete Reference: HTML & CSS (Fifth Edition)*. McGraw-Hill Education, 2008. Capítulos 1 y 4. <https://www.dcpvhpm.org/E-Content/BCA/BCA-II/Web%20Technology/the-complete-reference-html-css-fifth-edition.pdf>
- [8] Haverbeke, M. *Eloquent JavaScript: A Modern Introduction to Programming* (3rd Edition). No Starch Press, 2018. Capítulo 1. <https://eloquentjavascript.net/>
- [9] Express.js Foundation. "Express – Node.js web application framework", Express Documentation, 2023. <https://expressjs.com/>
- [10] Mongoose. "Mongoose – Elegant MongoDB object modeling for Node.js", Mongoose Documentation, 2023. <https://mongoosejs.com>
- [11] Express.js. "Using middleware", Express Documentation, 2023. <https://expressjs.com/en/guide/using-middleware.html>
- [12] Nodemailer. "Nodemailer – Send e-mails with Node.js", Nodemailer Documentation, 2023. <https://nodemailer.com/about/>
- [13] Multer. "Multer – Node.js middleware for handling multipart/form-data", Multer Docs, 2023. <https://github.com/expressjs/multer>
- [14] Motdotla. "Dotenv – Loads environment variables from .env file", npmjs.com, 2024. <https://www.npmjs.com/package/dotenv>
- [15] Express.js. "Using CORS middleware in Express", Express Documentation, 2023. <https://expressjs.com/en/resources/middleware/cors.html>

## *Índice de la memoria*

<b>Capítulo 1.</b>	<b>Introducción .....</b>	<b>7</b>
<b>Capítulo 2.</b>	<b>Descripción de las Tecnologías.....</b>	<b>9</b>
2.1	Tecnologías de Frontend .....	9
2.1.1	HTML.....	9
2.1.2	CSS.....	9
2.1.3	JavaScript .....	10
2.2	Tecnologías de Backend .....	10
2.2.1	Node.js .....	10
2.2.2	Express.js.....	10
2.2.3	APIs RESTful.....	11
2.2.4	JSON Web Token (JWT).....	12
2.2.5	Nodemailer .....	12
2.2.6	Multer .....	13
2.2.7	Express.static.....	13
2.2.8	Middleware de autenticación .....	13
2.2.9	Bcryptjs.....	14
2.2.10	Dotenv.....	14
2.2.11	Cors .....	14
2.3	Tecnología de Base de Datos .....	15
2.3.1	MongoDB .....	15
<b>Capítulo 3.</b>	<b>Estado de la Cuestión .....</b>	<b>16</b>
<b>Capítulo 4.</b>	<b>Definición del Trabajo .....</b>	<b>20</b>
4.1	Justificación .....	20
4.2	Objetivos .....	21
4.3	Metodología .....	24
4.4	Planificación y Estimación Económica .....	27

---

<b>Capítulo 5. Estructura Base .....</b>	<b>29</b>
<b>5.1 Arquitectura del Sistema.....</b>	<b>29</b>
5.1.1 Estructura Cliente-Servidor Desacoplada.....	29
5.1.2 Comunicación mediante API RESTful .....	30
5.1.3 Estructura de Carpetas del Backend .....	30
5.1.4 Persistencia de Datos con MongoDB .....	31
<b>5.2 Diseño de la Base de Datos .....</b>	<b>31</b>
<b>Capítulo 6. Sistema Desarrollado .....</b>	<b>33</b>
<b>6.1 Registro y Autenticación .....</b>	<b>33</b>
6.1.1 Archivo .env .....	35
<b>6.2 Creación y Edición del Perfil .....</b>	<b>36</b>
<b>6.3 Publicaciones .....</b>	<b>38</b>
<b>6.4 Ofertas.....</b>	<b>40</b>
<b>6.5 Mensajes .....</b>	<b>42</b>
<b>6.6 Notificaciones .....</b>	<b>43</b>
<b>6.7 Búsqueda.....</b>	<b>44</b>
<b>Capítulo 7. Análisis de Resultados .....</b>	<b>46</b>
<b>7.1 Módulo de Autenticación.....</b>	<b>46</b>
<b>7.2 Módulo de Creación de Perfil .....</b>	<b>48</b>
<b>7.3 Módulo de Publicaciones .....</b>	<b>50</b>
<b>7.4 Módulo de Ofertas.....</b>	<b>51</b>
<b>7.5 Módulo de Mensajes.....</b>	<b>55</b>
<b>7.6 Módulo de Notificaciones .....</b>	<b>56</b>
<b>7.7 Módulo de Búsqueda .....</b>	<b>58</b>
<b>Capítulo 8. Conclusiones y Trabajos Futuros .....</b>	<b>60</b>

<b>Capítulo 9. Bibliografía .....</b>	<b>63</b>
<b>ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS.....</b>	<b>65</b>
<b>ANEXO II. Modelos de la Base de Datos.....</b>	<b>67</b>
9.1.1 Usuario .....	67
9.1.2 Publicación .....	68
9.1.3 Comentario .....	68
9.1.4 Oferta .....	69
9.1.5 Mensaje.....	69
9.1.6 Notificación .....	72

## *Índice de figuras*

Figura 1: Vista de la página de inicio de sesión .....	33
Figura 2: Formulario de registro.....	34
Figura 3: Correo de verificación de cuenta .....	35
Figura 4: Vista de recuperación de contraseña .....	35
Figura 5: Formulario de reseteo de contraseña.....	35
Figura 6: Formulario de Edición de Perfil.....	37
Figura 7: Contador de seguidores y seguidos.....	38
Figura 8: Formulario de Publicaciones.....	39
Figura 9: Caja de reacciones (Likes y Comentarios).....	39
Figura 10: Vista del Panel de Ofertas para Jugadores .....	40
Figura 11: Vista del panel de Ofertas para Clubes .....	41
Figura 12: Vista del Sistema de Mensajes.....	42
Figura 13: Vista del Panel de Notificaciones .....	44
Figura 14: Vista del Menú de Búsqueda .....	45
Figura 15: Ejemplo de Formulario de Registro .....	47
Figura 16: Confirmación Visual de Registro.....	47
Figura 17: Confirmación de Verificación.....	47
Figura 18: Mensaje Visual para Recuperación de Contraseña .....	48
Figura 19: Mensaje de Confirmación de Contraseña Actualizada .....	48
Figura 20: Vista genérica del Perfil de Jugador .....	49
Figura 21: Vista Genérica del Perfil del Club .....	49
Figura 22: Cabecera del Perfil de Jugador.....	50
Figura 23: Feed de Publicaciones General .....	51
Figura 24: Feed de Publicaciones de Seguidos .....	51
Figura 25: Vista de Oferta Creada.....	52
Figura 26: Formulario de Edición de Oferta .....	52

---

Figura 27: Versión Actualizada de la Oferta .....	53
Figura 28: Vista de Oferta para Jugadores .....	53
Figura 29: Vista de Lista de Candidatos a una Oferta .....	54
Figura 30: Vista Actualizada de la Oferta .....	54
Figura 31: Vista de Chat (Emisor).....	55
Figura 32: Bandeja de Entrada de Mensajes .....	55
Figura 33: Vista de Chat (Receptor).....	56
Figura 34: Vista Actualizada de Panel de Notificaciones .....	57
Figura 35: Vista de Resultados de Búsqueda .....	58
Figura 36: Vista de Perfil Ajeno en Resultados de Búsqueda.....	59

## *Índice de tablas*

Tabla 1. Planificación Temporal .....	27
Tabla 2: Estimación Económica .....	28

## Capítulo 1. INTRODUCCIÓN

Actualmente, nos encontramos en una sociedad que cada vez está más digitalizada. En aquellos parques llenos de vida gracias a los niños, ya cada vez quedan menos, pues esos niños ahora prefieren la vida digital, conocer las cosas a través de una pantalla. No debemos caer en el error, puesto que todo el mundo se ha ido adaptando poco a poco a las nuevas tecnologías. En lo que a este proyecto respecta, ya no se ven a jóvenes recorriéndose las calles con una pila de currículums para encontrar un trabajo. Ahora, se crean cuentas en aplicaciones como LinkedIn o Infojobs, y a través de ellas llegan a su objetivo. No se puede negar la utilidad que tienen, o al menos la cantidad de usuarios distintos en dichas plataformas. Realmente, son herramientas que han llegado para formar parte de la sociedad y mejorar la calidad de vida de la gente.

La segunda pata sobre la que se cimienta este proyecto es el fútbol, considerado deporte rey, y que además tiene la capacidad de unir a un gran número de personas. Sin ir más lejos, en España, es el deporte que cuenta con mayor número de licencias federativas. Sin embargo, no todos los aficionados a este deporte son realmente conscientes de lo que se esconde detrás de esa fachada idílica que es el fútbol profesional. Los cálculos son realmente sencillos. En el escalón profesional, que comprende Primera División y Segunda División, hay un total de 42 equipos, 20 en Primera y 22 en Segunda. Cada equipo cuenta con un máximo de 25 fichas o licencias profesionales, por lo que el resultado es que tan solo 1100 personas en toda España son jugadores profesionales. Por debajo de ellos, tenemos dos escalones más, uno Semi-Profesional, y otro completamente Amateur, en el que se dan muchos casos en los que los propios jugadores son los que pagan a sus clubes para jugar. Ya solo en ese escalón Semi-Profesional, el número de equipos, y, por consiguiente, de jugadores, aumenta en una gran cantidad con respecto al número de jugadores profesionales, por lo que la diferencia con el escalón Amateur ya es abrumadora.

A este último escalón es al que este proyecto se dirige. Entre todos los equipos y jugadores que aparecen aquí, es muy probable encontrarnos con muchas diferencias entre unos y otros.

El resumen que hacen siempre de estos equipos consiste en unas instalaciones en condiciones muy mejorables, sobreviviendo a base de jugadores del propio pueblo o de la zona, sin recursos que les ayuden a salir adelante, más allá de toda su base formativa. De todos los que se encuentran en esta situación, siempre salen casos de jugadores que llegan a profesionales. Se puede hablar de Pablo Barrios, jugador del Atlético de Madrid en Primera División, que empieza en un club en el barrio de Moratalaz, pero no de muchos más casos. Es por eso, por lo que este proyecto quiere brindar tanto a clubes como a jugadores una herramienta que, sin ser tampoco una completa revolución del ámbito del fútbol amateur, les ayude a todos a tener esa visibilidad y de alguna forma, una oportunidad de que alrededor se reconozca tanto a clubes como a jugadores.

El proyecto en sí nace en esa idea principal de ser una herramienta de ayuda, y toma forma a través de la existencia previa de otras aplicaciones que, sin compartir el ámbito en el que funcionan, realizan una labor similar. Teniendo en cuenta el auge de estas aplicaciones, se desarrollará de tal forma que el propio usuario tendrá toda la libertad para exponerse a sí mismo y sus capacidades, o los clubes tendrán también la misma libertad para tratar de llegar a todos los jugadores que les sea posible, entablar relación con otros clubes para posibles acuerdos o partidos, o incluso ofrecer puestos en sus equipos.

Antes de comenzar con este proyecto, era importante conocer la opinión de aquellos a quienes se dirige. El resultado ha sido bastante dispar, principalmente por la gran diferencia de contextos que surgen entre los clubes o entre los jugadores. Aun así, la expectativa de uso ha resultado favorable, principalmente en aquellos jugadores jóvenes a los que les gustaría llegar más lejos de lo que están. La opinión de los jugadores más veteranos en este sentido fue distinta, dado que son conscientes de que sus mejores años han pasado y ya solo quieren disfrutar, pero comparten que una herramienta así muchas veces les habría resultado de gran ayuda.

## Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

En este capítulo, se describirán las distintas tecnologías implementadas en el desarrollo de este proyecto. Dichas tecnologías estarán divididas en distintos bloques, de tal forma que se aporte claridad a la hora de explicar las distintas partes que componen la aplicación y cómo se integra cada una en las distintas funcionalidades.

### ***2.1 TECNOLOGÍAS DE FRONTEND***

Para el desarrollo de frontend, se han utilizado tres lenguajes distintos, HTML, CSS y JavaScript. Es la combinación de estos tres lenguajes la que consigue una interfaz visual de cara a un navegador web.

#### **2.1.1 HTML**

El lenguaje HTML es la base fundamental del desarrollo web, puesto que es el lenguaje que interpreta el propio navegador. No se trata de un lenguaje de programación al uso, sino que se trata de un lenguaje de descripción basado en etiquetas, que el navegador interpreta para mostrar el contenido en la pantalla. El nombre de cada etiqueta define la colocación de los bloques de contenido. [1]

#### **2.1.2 CSS**

CSS es el lenguaje que permite controlar la apariencia visual del contenido definido en el HTML. De esta forma, con una misma base estructural de HTML, se pueden obtener distintos diseños gracias a cambios en la hoja de estilos que se define con CSS. En sus versiones más modernas, CSS ya puede hacer diseños denominados *responsive*, los cuales responden a cambios en la pantalla, en función del dispositivo. [1]

### **2.1.3 JAVASCRIPT**

El lenguaje JavaScript, enfocado en Frontend, responde a las acciones del usuario. Se trata de un lenguaje interpretado, es decir, se ejecuta directamente en el navegador, y basado en eventos como un movimiento de ratón, un click o una tecla pulsada. De esta forma, aporta a la página mayor dinamismo e interactividad con el usuario.

Por otra parte, se ha convertido también en un lenguaje de propósito general de backend, gracias a entornos como Node.js.

Es un lenguaje que permite modificar dinámicamente la estructura HTML y su contenido, crear efectos visuales, realizar peticiones o integrar APIs externas. [2]

## **2.2 TECNOLOGÍAS DE BACKEND**

### **2.2.1 NODE.JS**

Node.js es un entorno de ejecución de JavaScript que está del lado del servidor que permite ejecutar código JavaScript fuera del navegador, convirtiéndose en una herramienta eficaz en el desarrollo de Backend de forma escalable y eficaz. Se adapta a un entorno de operaciones de entrada y salida que no bloquea el programa mientras se realizan estas operaciones, permitiendo así que se manejen distintas peticiones simultáneamente. De esta forma, se reduce el uso de recursos y se aumenta la eficiencia.

Además, incluye un gestor de paquetes (Node Package Manager, npm) que facilita la gestión de las librerías que se quieran implementar en el desarrollo web. [3]

### **2.2.2 EXPRESS.JS**

Express.js es un framework diseñado para Node.js que proporciona un conjunto de funcionalidades para el desarrollo de aplicaciones web y APIs RESTful. Facilita la gestión de rutas, middleware y peticiones HTTP. Se trata de un estándar en el desarrollo del backend con Javascript.

Express permite la definición de rutas que responden a peticiones como GET, PUT, POST o DELETE de forma sencilla. También gestiona middlewares, funciones que se ejecutan entre la petición y la respuesta al cliente, como puede ser la autenticación. [4]

### **2.2.3 APIs RESTFUL**

Una API RESTful es un estilo de arquitectura para el diseño de servicios web, permitiendo la comunicación entre cliente y servidor a través del protocolo HTTP. Se organiza en torno a recursos identificados mediante URLs y utiliza los métodos estándar de HTTP. Son independientes del cliente, lo que promueve la interoperabilidad y permite el desarrollo de aplicaciones web. [5]

#### **2.2.3.1 Protocolo HTTP**

El protocolo HTTP es un protocolo de comunicación basado en texto. Permite la transmisión de información entre clientes y servidores, siendo este protocolo la base sobre la que se construyen servicios de Internet, además de ser el núcleo de tecnologías como páginas web o APIs RESTful.

Funciona bajo un modelo cliente-servidor, en el cual el cliente realiza peticiones al servidor y este responde. Los mensajes intercambiados incluyen una línea de estado, cabeceras y un cuerpo de contenido que puede estar vacío.

Este protocolo define varios métodos estándar para describir la intención de la solicitud. Los implementados en la mayoría de los casos son:

- GET: para obtener recursos del servidor
- POST: para enviar o crear recursos
- PUT: para reemplazar un recurso existente (actualizar la información)
- PATCH: para modificar un recurso parcialmente
- DELETE: para eliminar un recurso

Además, este protocolo incluye códigos de estado para indicar el resultado de cada operación. Estos códigos se dividen según la clase de la respuesta, definiéndose según el primer dígito. [6]

- 1xx: Informativo. La petición se ha recibido y se está procesando
- 2xx: Éxito. La petición se ha recibido, procesado y aceptado
- 3xx: Redirección. Hay que hacer más acciones para completar la petición
- 4xx: Error del cliente. La petición no es correcta o no puede realizarse
- 5xx: Error del servidor. El servidor no ha podido responder a una petición que es válida

#### **2.2.4 JSON WEB TOKEN (JWT)**

Es un estándar abierto que permite la transmisión segura y compacta de información en formato JSON. Utilizado para autenticación y autorización en arquitecturas sin estado, como APIs RESTful.

Un token JWT se compone de:

- Header: especifica el tipo de token y el algoritmo de firma
- Payload: contiene la información
- Signature: resultado de firmar el header y el payload con una clave secreta o un certificado

Gracias a JWT, el servidor no necesita mantener sesiones en memoria y, por el sistema de firma, se evita que el token se manipule, aunque no está cifrado. [7]

#### **2.2.5 NODEMAILER**

Se trata de una librería de Node.js diseñada para mandar correos electrónicos de manera automática desde aplicaciones backend. Es una de las soluciones más populares para implementar funcionalidades como verificación de cuenta, recuperación de contraseña o notificaciones por email.

Nodemailer permite conectar fácilmente con servicios SMTP (Simple Mail Transfer Protocol), y es compatible con autenticación segura, cifrado TLS/SSL y personalización de mensajes. [8] [9]

### **2.2.6 MULTER**

Multer es un middleware para Node.js diseñado para gestionar peticiones del estilo multipart/form-data, el tipo de formulario empleado para subir archivos desde el cliente al servidor. En combinación con Express.js, se implementa para permitir la carga de imágenes desde formularios HTML.

Multer interpreta las peticiones que contienen archivos y los almacena en disco o memoria. Una vez procesado el archivo, queda disponible para manipularlo desde el backend.

Para poder garantizar su seguridad, se suele proteger la ruta de subida de archivos con JWT. [10]

### **2.2.7 EXPRESS.STATIC**

Es un middleware integrado en Express.js que permite servir automáticamente archivos estáticos desde el servidor, como imágenes. Se han implementado controles adicionales, como el control de subidas con Multer o acceso restringido a determinadas rutas. [11]

### **2.2.8 MIDDLEWARE DE AUTENTICACIÓN**

Es un middleware personalizado de Express.js, es decir, una función intermedia que se ejecuta antes de que una petición alcance el controlador final. En concreto, este middleware intercepta las peticiones entrantes a rutas protegidas, extrae el token de la cabecera *Authorization* y lo verifica con la clave secreta del servidor. Si el token es válido, permite acceso a la ruta que protegía. En caso de no ser válido, la petición se rechaza con un 401 – Unauthorized. De esta forma, solo los usuarios autenticados pueden acceder a determinadas rutas. [12]

### **2.2.9 BCRIPTJS**

Es una biblioteca de JavaScript que implementa el algoritmo bcrypt para su uso en Node.js. Está diseñada para encriptar contraseñas antes de almacenarlas en la base de datos, haciendo muy complicada su recuperación.

Este algoritmo es resistente a ataques de fuerza bruta y ataques por diccionario gracias al Salting, que añade una cadena aleatoria antes del cifrado, y el Costo configurable, que determina el número de rondas de procesamiento del cifrado. [13]

### **2.2.10 DOTENV**

Dotenv es una biblioteca de Node.js que permite cargar variables de entorno desde un archivo .env, facilitando su configuración segura y flexible de parámetros sensibles y dependientes del entorno.

El archivo generado para su configuración nunca se incluye en el repositorio. Si se hace en un servidor local, es un archivo que no puede verse ya que es exclusivo para el sistema. Su acceso debe realizarse por consola. [14]

### **2.2.11 CORS**

Cross-Origin Resource Sharing, CORS, es un mecanismo de seguridad implementado por los navegadores modernos que permite o restringe que una aplicación web en un origen acceda a recursos de otro origen distinto. Por defecto, las peticiones HTTP están bloqueadas por el navegador como medida de protección.

Para permitir estas peticiones entre frontend HTML y servidor Node.js es necesario configurar cabeceras CORS adecuadas en el servidor. [15]

## **2.3 TECNOLOGÍA DE BASE DE DATOS**

### **2.3.1 MONGODB**

MongoDB es un sistema de gestión de bases de datos NoSQL, orientado a documentos. Está diseñado para almacenamiento y procesamiento de grandes volúmenes de datos. A diferencia de las bases de datos relacionales, MongoDB no utiliza tablas, sino colecciones de documentos en formato BSON (la forma binaria de JSON), lo que sugiere el uso de datos en este formato dentro de la aplicación.

Dentro de las características que se ofrecen dentro del entorno MongoDB, se ha empleado MongoDB Atlas, la cual ofrece soporte multi-cloud y simplifica los procesos de mantenimiento y escalado de la base de datos. [16]

#### **2.3.1.1 Mongoose**

Es la biblioteca de Node.js que actúa como Object Data Modeling para MongoDB. Aporta una capa de abstracción estructurada sobre la base de datos, permitiendo trabajar con documentos MongoDB utilizando modelos definidos por esquemas. Estos esquemas definen la estructura de cada documento. Además, gracias a Mongoose se pueden relacionar documentos, validar los datos y reutilizar los modelos. [17]

## **Capítulo 3. ESTADO DE LA CUESTIÓN**

Para el desarrollo de este proyecto, se ha separado en tres grandes bloques que ayudan a asentar tanto el proceso como el contexto en el que se empleará.

En primer lugar, tenemos el entorno digital, que cada vez engloba a más y más gente. En este sentido, las redes sociales han cobrado una gran importancia. [18] No es de extrañar que la población joven tenga cuenta en algunas de las redes sociales más importantes, como Instagram o TikTok. Estas redes sociales aportan un punto de vista muy concreto para desarrollar ciertas funcionalidades, principalmente en el tema de publicaciones con imágenes y las posibles reacciones a estas publicaciones. [19]

Sin salir de las redes sociales, nos encontramos con LinkedIn, que ya cierra un poco el foco y se centra en el ámbito profesional. Es una plataforma que permite al usuario exponerse como posible candidato a un puesto de trabajo. Tener un perfil creado en esta red social permite extender la información profesional que un candidato puede ofrecer a una empresa, dejando al currículum como una forma resumida de lo que se expone en los distintos perfiles. De cara a la empresa, les permite realizar búsquedas independientes de candidatos, o exponer mucha más información y más específica sobre la empresa. Dado el aumento en la variabilidad en los empleos que se ha dado en los últimos años, es una herramienta que ayuda tanto a los empleados a encontrar una vía de cambio, como a las empresas de cara a buscar futuros candidatos, ampliando sus propios límites en cuanto al rango de acción que alcanzaban, gracias a poder publicar sus ofertas y recibir candidaturas. [20]

Por último, todas estas redes sociales tienen algo muy importante en común, que es la capacidad que dan a los usuarios de relacionarse gracias a la funcionalidad de mensajería. Esta característica, aunque muchas veces ya se da por hecho que existe en las aplicaciones, es sobre la que se fundamenta la idea que tiene este proyecto de conectar personas y clubes de una manera más rápida y eficaz, ampliando también la red de contactos que se pueda tener.

Pasando a la segunda parte de la base de este proyecto, tenemos las propias aplicaciones web. El avance hacia una era digitalizada no se limita solo a las redes sociales, se extiende mucho más allá. Cada vez son menos las empresas que no cuentan con un sitio web en el que ofrecer sus servicios, aunque solo puedan atraer al cliente y no vender a través de la web. [21]

Desarrollarlo en un entorno web y no como aplicación móvil también se basa en la facilidad que supondría a la hora de acceder desde un ordenador, dado que también se podría acceder a la versión web desde un dispositivo móvil.

Dentro de todo este desarrollo web es donde encontramos el modelo API REST, que permite una comunicación sencilla entre frontend y backend, además de ser el modelo que mejor se adapta a una plataforma modular, escalable y orientada a prestar un servicio como este. [22][23]

La tercera, y última parte de la base de este proyecto, es el ámbito donde se busca ayudar, el fútbol. Este deporte se encuentra dividido en federaciones, cada una con una serie de equipos y competiciones. El escalón más alto actualmente es la FIFA (Fédération Internationale de Football Association), la cual engloba a todos los países del mundo. Tiene a cargo competiciones como el Mundial de Clubes, o el Mundial, la máxima competición de selecciones nacionales, y de la que España fue campeón en 2010. [24] Un escalón por debajo estarían las federaciones continentales, en el caso de Europa, sería la UEFA (Union des associations européennes de football), que recoge a todos los equipos de los países asociados a Europa futbolísticamente (incluye países como Azerbaiyán o Kazajistán, que no son europeos geográficamente). Están a cargo de competiciones como la Eurocopa, de la cual España es el máximo ganador con 4 trofeos, o las competiciones de clubes como la Champions League, la Europa League o la Conference League. Tanto en la Champions League como en la Europa League, los equipos más laureados son españoles, siendo estos el Real Madrid, con 15 Champions League, y el Sevilla, con 7 Europa League. [25] El siguiente escalón serían las federaciones nacionales, como la FA (Football Association) en Inglaterra, la más antigua de todas, o la RFEF (Real Federación Española de Fútbol). Su

poder de decisión ya se queda en las competiciones nacionales, que suelen ser las ligas, como la Liga EASports (Primera División) y la Liga Hypermotion (Segunda División) en España, o las copas nacionales, en el caso de España, la Copa del Rey. En este caso, las divisiones dependientes de la RFEF son las cinco primeras categorías, de las cuales dos son profesionales, una está en proceso de profesionalizarse, y dos son semi-profesionales.[26] El último escalón que tenemos son las federaciones territoriales, en el caso de Madrid, es la RFFM (Real Federación de Fútbol de Madrid). Aquí es donde se encuentran la mayoría de los equipos de la Comunidad de Madrid, y donde de verdad se evidencia la diferencia entre unos equipos con recursos y otros que no tienen tantas posibilidades.

En cuanto a estadísticas se refiere, los clubes y jugadores que se encuentran en federaciones nacionales están presentes en plataformas como Transfermarkt o BeSoccer, aplicaciones centradas en las estadísticas completas y detalladas del fútbol, tanto de partidos como de clubes, jugadores y entrenadores. Dado que se centran en los escalones más altos, donde los clubes más pobres tienen la suficiencia económica como para tener recursos distintos, sus usuarios principalmente son los aficionados. [27][28]

Más enfocado al ámbito social, existe Footvisual, una plataforma que permite el networking entre sus usuarios, compartir experiencias y promocionarse. Por otra parte, existe FutbolJobs, que resulta ser similar a Infojobs o LinkedIn, pero aplicado al fútbol. Son dos plataformas que no comparten sus funcionalidades y objetivos principales. Mientras que FutbolJobs cuenta con el sistema de ofertas y solicitudes para jugadores o entrenadores, Footvisual se limita a ser muy similar a cualquier otro tipo de red social, enfocándose en el contenido del usuario y no entrando en ese sistema de ofertas. [29][30]

Este proyecto busca la combinación de ambas características, permitiendo así la completa interacción entre los distintos usuarios para ampliar la red de contactos, y ofreciendo la posibilidad de generar ofertas para atraer jugadores nuevos, o los propios jugadores buscar otro club en el que seguir jugando. Además, gracias al nivel de detalle que se puede ofrecer en el perfil, se puede llegar a ofrecer un listado de estadísticas de cada usuario, de cara a ser meramente informativas, permitir comparaciones cuantitativas entre perfiles similares, o

incluso llegar a realizar predicciones en base al resto de jugadores que componen el equipo. En el caso de los clubes, estas predicciones también llegarían a ser entre equipos, sobre todo a la hora de organizar partidos amistosos, ya que este tipo de partidos corren a cargo de los dos equipos involucrados para organizarse y disputarse.

De esta forma, se quiere dotar a los equipos de una mayor capacidad de alcance que sus alrededores, pudiendo así conocer más clubes y jugadores con los que entablar relaciones que pueden llegar a ser fructíferas y beneficiosas para ambas partes. En cuanto a los jugadores, permitirán entablar relación entre ellos y los clubes, dándose a conocer más ampliamente y expandiendo así sus posibilidades de crecimiento y mejora en otros equipos distintos. También permitiría compartir experiencias de cara al aprendizaje, ya que, como en muchos otros ámbitos de la vida, la experiencia es un factor muy importante en el deporte que muchas veces marca diferencias entre un jugador y otro, o entre dos equipos.

## **Capítulo 4. DEFINICIÓN DEL TRABAJO**

En este capítulo se expondrá en qué consiste este proyecto, comentando sus diferentes fases hasta llegar a completarse. A su vez, estarán acompañados de los objetivos planteados en cada una de las fases y que definirán el sistema una vez finalizados.

### **4.1 JUSTIFICACIÓN**

A la luz del análisis del estado de la cuestión, se puede concluir que, aunque existen soluciones parciales en este ámbito, no hay una herramienta que por sí sola agrupe todas las características que ofrecen cada una por separado. Por un lado, estaría la parte estadística, por otro la parte social, y por un tercero la parte más profesional en cuanto a sistema de ofertas. Es por este motivo que se pretende desarrollar una plataforma que combine todas estas características. Así, se cubrirían las distintas necesidades que pudieran surgir, todo bajo un mismo entorno. Dentro de este proyecto, el usuario podrá construir y compartir un perfil más cerca de la profesionalidad, interactuar con otros usuarios, recibir ofertas de traspaso de un club a otro y mantener una comunicación directa con los distintos usuarios de la plataforma.

Frente a las plataformas existentes, este proyecto integrará funcionalidades como un sistema de perfiles adaptado al fútbol basado en el sistema de LinkedIn, un feed para publicación de contenido, interacción social mediante reacciones (Me gusta/Likes) o comentarios, mensajería privada o notificaciones en tiempo real. Además, se ofrece control completo de las sesiones de los usuarios y se podrán conseguir perfiles mucho más completos gracias a la opción de subida de imágenes.

Para lograr estas funcionalidades, se han integrado una serie de tecnologías para el desarrollo web bajo una arquitectura API RESTful escalable y modular, que podrá adaptarse a futuras integraciones móviles o servicios externos.

Desde el punto de vista técnico, el proyecto se apoya en las tecnologías de desarrollo web modernas para poder construir sobre ellas toda una plataforma profesional. Si cambiamos el punto de vista hacia el mercado, la ausencia de un espacio o plataforma que integre todas las funciones que pueden ser claves en el desarrollo tanto de clubes como de jugadores, hace que se encuentre una oportunidad para hacer de este entorno en crecimiento un entorno aún más competitivo y digitalizado.

En definitiva, este proyecto responde a una necesidad real que no está completamente resuelta, posicionándose como una solución que resulta viable, escalable y de valor añadido, tanto para su uso directo como de cara a una evolución comercial o una integración en el ecosistema deportivo, ya sea actual o en el que surja con el paso del tiempo.

## **4.2 OBJETIVOS**

El proyecto tiene como objetivo el diseño y desarrollo de una plataforma web, al estilo de una red social, orientada al ámbito deportivo, enfocada en el mundo amateur, donde los recursos no son tan amplios y potentes como en el mundo profesional. Tendrá un enfoque muy específico en el fútbol, pero es una base extensible a otros deportes. El sistema busca ofrecer un espacio seguro donde jugadores y clubes puedan crear perfiles profesionales, interactuar entre sí, compartir contenido y experiencias, acceder a nuevas y mejores oportunidades profesionales y deportivas. A diferencia de una red social al uso, o un portal de empleo más clásico, esta plataforma tiene la ambición de combinar ambas dimensiones en un mismo entorno.

Desde un punto de vista general, el proyecto tiene como objetivo construir una aplicación web moderna, segura y escalable, estructurada bajo una arquitectura cliente-servidor, que permita a los diferentes tipos de usuarios, en este caso, personas o clubes, desarrollar su perfil público, relacionarse con otros miembros de la red, compartir contenido audiovisual, gestionar ofertas y comunicarse de forma directa a través de un sistema de mensajería.

Para lograr alcanzar este objetivo global, se han definido una serie de objetivos más específicos, con la intención de orientar y estructurar el desarrollo del sistema a nivel funcional y técnico.

En primer lugar, se propone diseñar e implementar un sistema completo de autenticación de usuarios, que incluya funcionalidades de registro, inicio de sesión, recuperación de contraseña y verificación por correo electrónico. Este sistema debe estar protegido por cifrado de contraseñas, para lo que se empleará *bcryptjs*, y control de sesión mediante tokens JWT (*JSON Web Tokens*), garantizando así seguridad y privacidad de los datos.

Una vez que se ha procedido con el registro, se plantea la creación de un módulo de perfil dinámico y editable, adaptado al tipo de usuario registrado. Para los jugadores, el perfil incluirá datos personales, trayectoria deportiva, estadísticas y presentación profesional. Para los clubes, el perfil permitirá detallar su localización, metodología, categorías, calendarios de partidos y estructura interna. En ambos casos, existirá la opción de elegir imágenes de perfil (en el caso de clubes, será el escudo) y de fondo, para mejorar la representación visual del usuario.

Uno de los componentes principales de este proyecto es el sistema de publicaciones, tipo “feed”, que permite a los usuarios compartir contenido textual y/o multimedia, así como interactuar con otras publicaciones mediante reacciones (likes) y comentarios. Este módulo busca fomentar la interacción social y la visibilidad de los usuarios en la comunidad. Por este motivo, se ofrecerá la opción a los usuarios de visualizar en su feed personal tanto publicaciones de otros usuarios en general, como de usuarios a los que siguen.

La segunda parte principal del proyecto es el módulo de gestión de ofertas deportivas, en el que los clubes pueden publicar vacantes o necesidades específicas, para jugadores, entrenadores o staff, y estos podrán postularse a dichas ofertas de forma estructurada. Cada oferta incluirá detalles como la posición solicitada, la ubicación, el nivel competitivo o los objetivos y condiciones que vayan asociados a ese puesto.

Para facilitar la comunicación entre usuarios, se ha planteado el desarrollo de un sistema de mensajería privada, con conversaciones bidireccionales protegidas por autenticación. Además, llevará consigo un sistema de notificaciones de tiempo real para informar a los usuarios de interacciones relevantes con respecto a su perfil.

Como último de los objetivos funcionales de la plataforma, se integrará un buscador avanzado que permita encontrar perfiles en base a palabras clave, con filtrado contextual por posición, edad, o ubicación. De esta forma, se optimizará la usabilidad y navegabilidad de la plataforma.

Pasando a un punto de vista más técnico, el objetivo principal es la persistencia de datos mediante el uso de MongoDB, que ofrece flexibilidad y rendimiento, con una gestión estructurada a través del ODM (*Object Data Modeling*) Mongoose, una herramienta que permite mapear los objetos de una aplicación como esta, a documentos de la base de datos de MongoDB.

El backend se desarrollará utilizando Node.js y Express.js, organizando las rutas en un modelo RESTful que facilita el mantenimiento y la escalabilidad del sistema. También se hará uso de herramientas adicionales como Multer, para la subida de imágenes, Nodemailer para envío de correos electrónicos, y dotenv para la gestión de configuraciones sensibles.

El conjunto de estos objetivos busca responder a una necesidad real detectada en el ámbito deportivo digital, y aspiran a aportar una solución tecnológica completa que ayude a que el mundo del fútbol amateur avance y se acerque hacia el mundo profesional, consiguiendo mayor visibilidad para todos los actores involucrados, jugadores, entrenadores y cuerpos técnicos, y clubes.

### **4.3 METODOLOGÍA**

Para el desarrollo de este proyecto se ha seguido una metodología de trabajo estructurada, iterativa y orientada a resultados, basándose en principios de desarrollo ágil y buenas prácticas de la ingeniería del software, como la modularidad o la potencial reutilización de código. Además, se han aplicado los principios básicos de seguridad y ha existido una buena consistencia en el desarrollo. Esta metodología ha permitido avanzar progresiva y organizadamente en la implementación del sistema.

En primer lugar, se realizó un análisis de requisitos para poder definir con precisión las funcionalidades que debía ofrecer la plataforma. Se identificaron los usuarios, entre los que destacaron jugadores y clubes, así como sus distintas necesidades dentro del sistema. Este análisis permitió establecer las bases del proyecto, estructurándolo en torno a distintos módulos diferenciados entre sí: autenticación, perfil, publicaciones, ofertas, mensajería, notificaciones y buscador.

Una vez se establecen los requisitos del sistema, se procede al diseño de la arquitectura general del sistema, optando por una arquitectura cliente-servidor desacoplada, es decir, frontend y backend se comunican a través de una API RESTful. Esto permite mejor escalabilidad y facilita el mantenimiento para el medio y largo plazo. Los modelos principales se definieron mediante Mongoose, permitiendo mapear las entidades del sistema en esquemas bien estructurados y validados.

El desarrollo del sistema se dividió en los módulos expuestos anteriormente, de tal forma que cada funcionalidad se implementó de forma independiente, sin perder la visión global del producto final.

#### **4.3.1 AUTENTICACIÓN**

El primer módulo implementado fue el de autenticación, que comprende todo el proceso desde el registro en la plataforma, pasando por la verificación de la cuenta, el inicio de sesión

y mantenimiento con token, hasta la recuperación y restablecimiento de la contraseña si fuese necesario.

#### **4.3.2 PERFIL**

Una vez autenticado el usuario, se accede a la creación del perfil. Este perfil es editable en todo momento, incluyendo la gestión de imágenes. Estará dividido en secciones de tal forma que la información quede clara tanto para el propio usuario como para la visualización de otros usuarios.

#### **4.3.3 PUBLICACIONES**

El módulo de publicaciones se corresponde con la página principal del perfil, a la que el usuario accede cada vez que inicia sesión. Se corresponde con la parte más enfocada a lo social, y comprende el proceso de crear publicaciones, compartir la publicación y ver las publicaciones que otros usuarios comparten, y las interacciones mediante likes o comentarios en las publicaciones. También se incluye aquí un filtro para ver todas las publicaciones existentes, por orden de publicación de más nuevo a más antiguo, o solamente las que han publicado usuarios a los que el propio usuario sigue.

#### **4.3.4 OFERTAS**

Una vez que el módulo más orientado a lo social estuvo activo, el siguiente paso fue el módulo profesional, es decir, todo el sistema de ofertas. En este caso, es el club el que crea la oferta, con la posibilidad de editarla, mientras que el jugador se postula para la oferta. Como en el propio ámbito profesional, que la candidatura se acepte no implicará nunca que el traspaso sea definitivo, solo que el club tiene interés en ver al jugador. El club tendrá la posibilidad de ver todo el listado de jugadores que se inscriben en la oferta. Una vez que el club considera que la oferta queda satisfecha con algún jugador, podrá eliminarla.

#### **4.3.5 MENSAJERÍA**

El sistema de mensajería privada busca que los usuarios puedan comunicarse también dentro de la plataforma. Así, se pueden resolver dudas, plantear acuerdos, o simplemente aumentar

la red de contactos y reforzar relaciones. Los mensajes que se envían entre usuarios serán privados en todo momento y estarán protegidos.

#### **4.3.6 NOTIFICACIONES**

El sistema de notificaciones internas aportará al usuario conocimiento constantemente de las interacciones que se produzcan con su perfil o módulos asociados. Las principales notificaciones serán de interacciones con publicaciones, nuevos seguidores, nuevas ofertas de clubes a los que el usuario siga, actualizaciones en el estado de candidaturas que haya presentado el usuario, o nuevos mensajes recibidos. En el caso específico de los clubes, podrán recibir notificaciones periódicas sobre el número de candidatos que se han postulado para una de sus ofertas.

#### **4.3.7 BUSCADOR**

Adicionalmente, se implementará un buscador para que el usuario pueda buscar a otros usuarios, clubes o jugadores, en base a una serie de palabras clave que se buscarán en los perfiles.

Durante el proceso, se fueron realizando pruebas continuas dentro de cada módulo, para comprobar que su funcionamiento era el esperado. Se prestó especial atención a la validación de tokens JWT, al cifrado de contraseñas y al acceso a las distintas rutas del sistema, buscando garantizar la máxima seguridad del sistema.

Todo este proceso se desarrolló en el entorno de desarrollo VisualStudio Code, guardando siempre una copia del módulo antes de realizar avances, para poder regresar en caso de error o cambio de planteamiento.

En conjunto, esta metodología ha resultado eficaz para alcanzar los objetivos propuestos, permitiendo que la plataforma tenga las características que se buscaban (modularidad, seguridad, escalabilidad y extensibilidad), y ha quedado preparada para su uso inmediato como para una futura evolución hacia nuevas funcionalidades o nuevas plataformas.

#### **4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA**

Dado el planteamiento de llevar a cabo una estrategia iterativa y modular, se ha seguido la siguiente planificación:

Desarrollo de Frontend	Semanas 3-4 de noviembre  Semanas 3-4 de enero  Semana 1 de febrero
Diseño de Backend	Semanas 2-4 de febrero
Diseño BBDD	Semanas 1-3 de marzo
Desarrollo de funciones	Semana 4 de marzo  Semanas 1-3 de abril  Semana 4 de mayo  Semana 1 de junio
Acoplamiento con Frontend	Semana 4 de marzo  Semanas 1-3 de abril  Semana 4 de mayo  Semana 1 de junio
Testeo y corrección de errores	Semanas 2-3 de junio

*Tabla 1. Planificación Temporal*

Dentro del desarrollo de funciones, junto con el acoplamiento de estas funciones con el Frontend, fue donde se siguió esa estrategia modular, implementando los módulos descritos uno por uno.

La estimación de coste sería la que sigue:

<b>CONCEPTO</b>	<i>Básico</i>	<i>Completo</i>	<i>Económico</i>
Desarrollo	20.750€	38.850€	15.950€
Infraestructura (1 año)	1.200€	2.400€	600€
Herramientas y servicios	800€	1.200€	400€
Diseño UX/UI	2.500€	3.800€	1.500€
Testing y QA	2.000€	3.000€	1.000€
Gestión del proyecto	3.000€	5.000€	-
Legal y Admin	1.00€	1.500€	-
<b>TOTAL PROYECTO</b>	31.250€	55.750€	15.950€

*Tabla 2: Estimación Económica*

En este punto, nos encontraríamos en la situación entre los planes Económico y Básico. No obstante, el desarrollo completo ofrecería la mejor cobertura a nivel económico para poder solventar cualquier problema que pudiera surgir, ya sea en términos de desarrollo o de almacenamiento de la base de datos.

## **Capítulo 5. ESTRUCTURA BASE**

Este capítulo presenta de forma detallada la base del sistema desarrollado, es decir, su arquitectura y sus módulos. Dado que se integran múltiples funcionalidades interconectadas, se estructura el contenido en distintos apartados que abordan tanto la configuración técnica inicial como el desarrollo de cada componente.

### ***5.1 ARQUITECTURA DEL SISTEMA***

El sistema desarrollado en este proyecto se basa en una arquitectura cliente-servidor, basada en el intercambio de datos mediante API RESTful. Esta elección responde a las necesidades que surgen al construir una plataforma de este estilo, la cual también debe ser capaz de diferenciar las responsabilidades del frontend (interfaz de usuario) y del backend (lógica del servidor y gestión de datos)

La arquitectura se ha diseñado con el objetivo de permitir una separación clara de capas, facilitar el mantenimiento futuro del sistema y sentar las bases para una posible expansión a aplicaciones móviles u otros clientes externos.

A continuación, se describen los principales componentes de la arquitectura:

#### **5.1.1 ESTRUCTURA CLIENTE-SERVIDOR DESACOPLADA**

La plataforma se divide en dos componentes principales:

##### ***5.1.1.1 Frontend (cliente)***

Desarrollado en HTML, CSS y JavaScript puro. Es el encargado de la interfaz gráfica y la interacción con el usuario. Toda la lógica de visualización se gestiona en esta parte, que a su vez consume los servicios proporcionados por la API del backend

### **5.1.1.2 Backend (servidor)**

Desarrollado en Node.js con Express.js como framework. Gestiona las rutas, los controladores, la lógica de negocio y la interacción con la base de datos. Ofrece una API RESTful que permite realizar operaciones CRUD (crear, leer, actualizar, eliminar) sobre los distintos recursos del sistema

### **5.1.2 COMUNICACIÓN MEDIANTE API RESTFUL**

El frontend y el backend se comunican a través de peticiones HTTP. Para ello, el backend expone una serie de endpoints estructurados por recursos, como puede ser /api/usuarios o /api/ofertas. Cada uno de ellos permite al cliente interactuar con la base de datos a través de operaciones GET, para obtener información, POST, para crear nuevos elementos, PUT, para actualizar la información existente, o DELETE para eliminar registros. Cada una de estas peticiones necesita ir acompañada de cabeceras, y, en caso de necesidad adicional, de un token JWT que autentique al usuario. La respuesta se obtiene en formato JSON, permitiendo una integración eficiente y flexible.

### **5.1.3 ESTRUCTURA DE CARPETAS DEL BACKEND**

El backend sigue una organización modular basada en responsabilidades. Las principales carpetas que componen el proyecto son:

- Routes/: define las rutas disponibles para cada recurso.
- Models/: contiene los distintos esquemas de Mongoose que representan las entidades del sistema
- Middlewares/: contiene funciones auxiliares como la verificación de tokens.
- Utils/: contiene utilidades como el envío de correo con Nodemailer o la carga de imágenes con Multer
- Uploads/: almacenamiento local de las imágenes subidas (perfil, fondo y publicaciones)
- .env: es el archivo de configuración, que contiene variables sensibles como claves de sesión, puertos y contraseñas SMTP

#### **5.1.4 PERSISTENCIA DE DATOS CON MONGODB**

La base de datos utilizada es MongoDB, gracias a que permite guardar la información en formato JSON, de forma que no necesitamos alterar la información en ningún momento para su almacenamiento. Además, ofrece una gran flexibilidad en el almacenamiento de información, lo que resulta conveniente para modelos dinámicos como el perfil de un usuario o una publicación. De la gestión de la base de datos se encarga Mongoose.

### **5.2 DISEÑO DE LA BASE DE DATOS**

El diseño de la base de datos constituye uno de los pilares fundamentales del sistema, ya que define la estructura sobre la cual se almacena, consulta y manipula toda la información que generan los usuarios, ya sea de creación propia o mediante interacciones.

Dada la naturaleza de MongoDB, se pueden representar cómodamente entidades más complejas, adaptándose mejor a las necesidades dinámicas de una red social. Cada entidad se representa mediante un modelo definido, en el que se especifican campos, tipo de campo, restricciones y referencias a otras colecciones, si las hubiera.

Los modelos implementados en el proyecto son:

#### **5.2.1 USUARIO**

Contiene la información principal de los usuarios registrados, tanto para personas como para clubes. Incluye campos como nombre, email, el tipo de usuario o la contraseña. Además, los campos específicos de cada tipo de usuario, las imágenes y los campos de seguridad.

#### **5.2.2 PUBLICACIÓN**

Representa el contenido generado por los usuarios. Sus campos son autor, en referencia al usuario, texto, imagen, fecha y los campos de interacción.

### **5.2.3 COMENTARIO**

Contiene los comentarios asociados a publicaciones. Se compone del autor, la publicación a la que se asocia, el contenido y la fecha

### **5.2.4 OFERTA**

Solo pueden crearla los clubes. Incluye título, descripción, requisitos, fecha y autor. Además, contiene una lista de candidatos (usuarios) que se postulan

### **5.2.5 MENSAJE**

Dentro del sistema de chat privado, cada mensaje contendrá emisor y receptor referenciados, contenido, fecha y una confirmación de si el destinatario del mensaje lo ha leído.

### **5.2.6 NOTIFICACIÓN**

Es el sistema de notificaciones interno. Contiene destinatario, tipo, mensaje, fecha y la comprobación de que se ha leído.

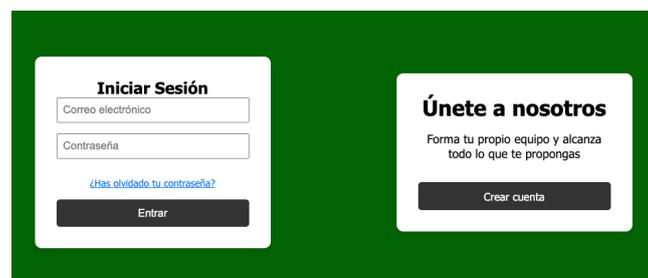
## Capítulo 6. SISTEMA DESARROLLADO

Este capítulo se centrará en la implementación de los distintos módulos que forman parte del sistema, teniendo en cuenta la arquitectura y los modelos que se han expuesto anteriormente. Cada parte del sistema responde a una necesidad específica que pudieran tener los usuarios. Por este motivo, el propio capítulo se dividirá en base a las distintas funcionalidades. También se tendrá en cuenta el siguiente flujo de uso, sobre todo en los primeros pasos que se dan en la plataforma, desde el registro hasta la creación del perfil completo. A partir de ese momento, el usuario tendrá mayor libertad para moverse por la plataforma.

### 6.1 REGISTRO Y AUTENTICACIÓN

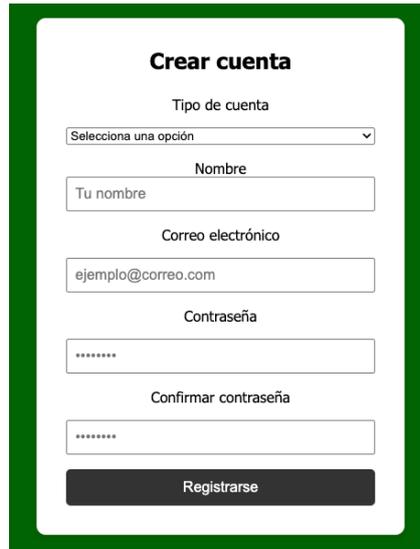
El primer módulo implementado fue el de autenticación, que comprende todo el proceso desde el registro en la plataforma, pasando por la verificación de la cuenta, el inicio de sesión y mantenimiento con token, hasta la recuperación y restablecimiento de la contraseña si fuese necesario.

El primer paso dentro del sistema debe ser registrarse, o iniciar sesión en caso de ya estar registrado. Para ello, se ha desarrollado en primer lugar una interfaz sencilla y práctica. De un solo vistazo, el usuario podrá identificar claramente dos bloques diferentes, un formulario para iniciar sesión mediante correo electrónico y contraseña, y un acceso a una visual distinta para rellenar el formulario de registro.



*Figura 1: Vista de la página de inicio de sesión*

Comenzando por este segundo caso, el formulario de registro, el usuario podrá elegir el tipo de perfil que tendrá, ya sea club o persona, y proporcionará su nombre, un correo electrónico y una contraseña.



*Figura 2: Formulario de registro*

Esta información se enviará al backend, que validará el email para ver que no exista ya una cuenta con dicho email, y generará un token provisional, el token de verificación. Adicionalmente, y en caso de que sea un email válido, se cifrará la contraseña y entonces se enviará el token de verificación, que le llegará al usuario al correo que haya facilitado en el formulario de registro, a la vez que crea el registro de usuario en la base de datos, con el campo confirmado en false.

El siguiente paso para el usuario debe ser verificar la cuenta, a través del enlace recibido en su cuenta de correo. Dicho enlace está acompañado del token que se generó previamente, y será este token el que se compruebe para actualizar el campo confirmado a true y permitir el acceso al sistema. Una vez que el usuario valide su cuenta, volverá a la página inicial, pero esta vez podrá iniciar sesión.

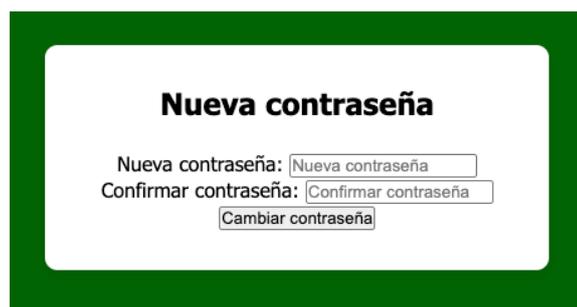


*Figura 3: Correo de verificación de cuenta*

El otro proceso que se abre en este módulo es el de recuperación de la contraseña. Para ello, el usuario accederá al enlace que se facilita en el bloque de inicio de sesión. A través de él, podrá introducir el correo electrónico que tenga asociado al perfil y recibirá un enlace en dicho correo que le redirigirá a un formulario en el que podrá poner una contraseña nueva. Al igual que en el caso de registro, una vez que se confirma el cambio de contraseña, se vuelve a la página inicial para poder iniciar sesión con la contraseña nueva.



*Figura 4: Vista de recuperación de contraseña*



*Figura 5: Formulario de reseteo de contraseña*

### 6.1.1 ARCHIVO .ENV

Este archivo recoge todas las variables de entorno críticas, de tal forma que se protege la información sensible y se permite una configuración flexible. Este archivo no se comparte ni se sube en repositorios.

Las variables incluidas en este archivo son:

- La conexión a MongoDB
- El token JWT
- La configuración del servicio de correos automáticos
- El puerto de escucha del servidor

El acceso se realiza mediante la librería dotenv.

```
# === Conexión a MongoDB Atlas ===
MONGO_URI=mongodb+srv://user:id@nombrecluster.16vmmjt.mongodb.net/tfg?retryWrites
=true&w=majority&appName=nombrecluster

# === Puerto del servidor ===
PORT=5050

# === JWT ===
JWT_SECRET=clave secreta
JWT_EXPIRES_IN=tiempo

# === Email para envío de confirmaciones y recuperación ===
EMAIL_USER=email de envío
EMAIL_PASS=contraseña de aplicacion
```

Este fragmento debe rellenarse con los valores que se elijan en el diseño

## **6.2 CREACIÓN Y EDICIÓN DEL PERFIL**

Independientemente del rol que tenga el usuario, una vez registrados es conveniente acceder a la creación del perfil. En este proceso, el usuario será el que decida la información que quiere compartir con el resto de los usuarios de la plataforma.

Como ya se ha explicado anteriormente, el perfil se define en base al modelo Usuario. No obstante, no es necesario completar todos los campos del formulario para tener un perfil activo, con rellenar la información básica para identificar un club o una persona será suficiente.

El resto de información que rellena cada usuario dependerá de su rol en la plataforma. Un jugador tendrá la opción de rellena información acerca de su trayectoria, sus capacidades futbolísticas, sus características personales o estadísticas de juego. Por el contrario, los clubes aportarán información sobre su metodología, sus equipos o sus logros.

Todos los campos serán llenados por los usuarios a conveniencia, salvo aquellos que el propio sistema requerirá a la hora de crear los perfiles. Se debe tener en cuenta que cuanto más información se tenga en el perfil (información precisa y clara), más fácil será la red de contactos. Por eso mismo, resulta algo inconveniente dejar la creación y edición del perfil para más adelante.



Editar perfil

Nombre completo

Edad

Posición

Equipo actual

Descripción

*Figura 6: Formulario de Edición de Perfil*

A su vez, el usuario tendrá la opción de subir imágenes tanto de perfil como de fondo. De esta forma, el usuario tendrá un perfil más profesional y resultará más atractivo de cara al resto de usuarios.

A nivel de backend, cada vez que un usuario quiera entrar en esta página, se hará una petición GET para obtener toda la información requerida y que exista, mientras que, al editar el perfil, se realizarán peticiones PUT con la información nueva. Todas las rutas de edición están protegidas con Middleware, de tal forma que un usuario no autenticado no puede acceder a los formularios para editar el perfil. Además, se aporta una lógica interna que impide cargar tipos de datos incorrectos, como algo distinto a un número entero en número de goles que el usuario persona ha marcado.

Como última característica dentro del módulo de perfiles, se dará al usuario la oportunidad de seguir a otros usuarios, ya sea por afinidad, interés, o gusto en su contenido. Los seguidores cobrarán importancia sobre todo en el módulo de publicaciones, ya que se implementará un filtro que dependerá de si el usuario sigue a quien hace la publicación o no.



*Figura 7: Contador de seguidores y seguidos*

### **6.3 PUBLICACIONES**

El módulo de publicaciones se corresponde con la página principal del perfil, a la que el usuario accede cada vez que inicia sesión. Se corresponde con la parte más enfocada a lo social, y comprende el proceso de crear publicaciones, compartir la publicación y ver las publicaciones que otros usuarios comparten, y las interacciones mediante likes o comentarios en las publicaciones. También se incluye aquí un filtro para ver todas las publicaciones existentes, por orden de publicación de más nuevo a más antiguo, o solamente las que han publicado usuarios a los que el propio usuario sigue.

El feed de publicaciones será la parte principal de la plataforma. Este sistema permitirá a los usuarios compartir contenido mediante texto e imágenes, interactuar mediante comentarios y reaccionar mediante likes. De esta forma, se contribuye a dinamizar la actividad de la red social y se fomenta la participación en ella de jugadores y clubes.

En primer lugar, todo usuario autenticado podrá crear publicaciones desde su página principal. Cada publicación contiene texto, una imagen opcional y una fecha automática de creación. La interfaz envía esta información al backend, que valida el token JWT, procesa la imagen subida con multer y guarda una publicación, asociada al usuario. Una vez todo este proceso se realiza correctamente, es posible ver la publicación en el feed.

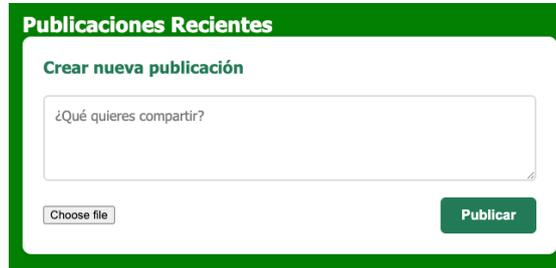


Figura 8: Formulario de Publicaciones

A la hora de visualizar las distintas publicaciones, el feed tendrá dos opciones, mostrar todas las existentes o solo las creadas por los usuarios a los que sigue el usuario principal. Se muestran foto y nombre del autor, texto de la publicación, la imagen asociada y el contador de likes y comentarios.

Para reaccionar a las publicaciones, los usuarios no tendrán más que pulsar el icono correspondiente. Esta funcionalidad envía una petición PUT y se añade o elimina el ID del usuario del array (lista) de likes de la publicación. El contador se actualiza dinámicamente.

Por otra parte, cada publicación puede recibir comentarios de otros usuarios. Estos comentarios se gestionan con un modelo independiente, Comentario, que se enlaza con la publicación original. Primero, el usuario escribe su comentario en el espacio correspondiente. Después, se envía una petición POST para publicar el comentario, que el backend guarda y devuelve en tiempo real. Adicionalmente, se implementa una petición GET para obtener los comentarios existentes.



Figura 9: Caja de reacciones (Likes y Comentarios)

El conjunto de funcionalidades que están presentes en este módulo permite a los usuarios expresar su identidad, compartir información relevante y mantener una interacción fluida, alimentando así la dinámica social dentro de la aplicación.

## 6.4 OFERTAS

Una vez que el módulo más orientado a lo social estuvo activo, el siguiente paso fue el módulo profesional, es decir, todo el sistema de ofertas. En este caso, es el club el que crea la oferta, con la posibilidad de editarla, mientras que el jugador se postula para la oferta. Como en el propio ámbito profesional, que la candidatura se acepte no implicará nunca que el traspaso sea definitivo, solo que el club tiene interés en ver al jugador. El club tendrá la posibilidad de ver todo el listado de jugadores que se inscriben en la oferta. Una vez que el club considera que la oferta queda satisfecha con algún jugador, podrá eliminarla.

Esta es otra de las funcionalidades de la plataforma, pero más orientada a los clubes, dado que tendrán muchas más acciones posibles que los jugadores, que solamente podrán ver la oferta y postularse en caso de estar interesados. Los clubes publicarán la oferta, podrán editarla si lo ven necesario, podrán ver la lista de candidatos que se han apuntado a la oferta y decidir sobre estos perfiles, y podrán eliminar la oferta en el caso de que ya no les interese tenerla activa. Este módulo permite dinamizar el contacto más profesional que personal dentro del ámbito del fútbol.

El proceso que sigue este módulo comienza desde los clubes, que publicarán una oferta. Esta oferta contendrá un título, una descripción detallada, unos requisitos para el perfil buscado y una fecha de publicación. Si el backend autoriza la petición POST, la oferta pasa a estar pública y activa.

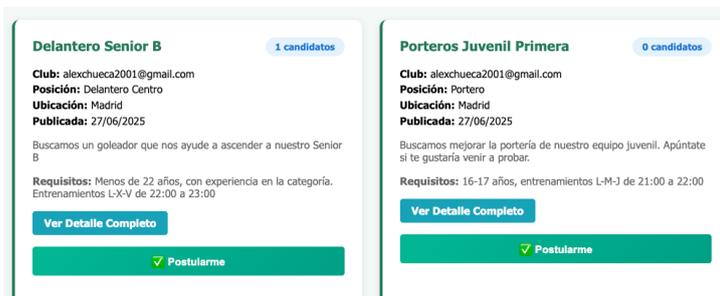


Figura 10: Vista del Panel de Ofertas para Jugadores

Una vez activada, el destinatario principal es el jugador, aunque también los clubes podrán ver lo que otros clubes publican en cuanto a ofertas. La vista de la oferta se construye con el título, un resumen, el club que la publica, y una opción de acceder al detalle y otra de postularse, esta última solo en el caso de jugadores.

El jugador que pulsa el botón de “Postularme” lanza una petición para verificar que dicho usuario no se ha postulado previamente a esa oferta, y añade el ID al array de candidatos. Si este paso es correcto, el botón se desactiva.

Por último, los clubes podrán gestionar la lista de posibles candidatos a través de un panel de gestión que estará vinculado al perfil del club. Esta gestión se realiza mediante una petición que verifica la identidad del usuario y recupera todos los usuarios que se han postulado.

Adicionalmente, se puede tanto editar como eliminar las ofertas, si los propios clubes así lo desean.



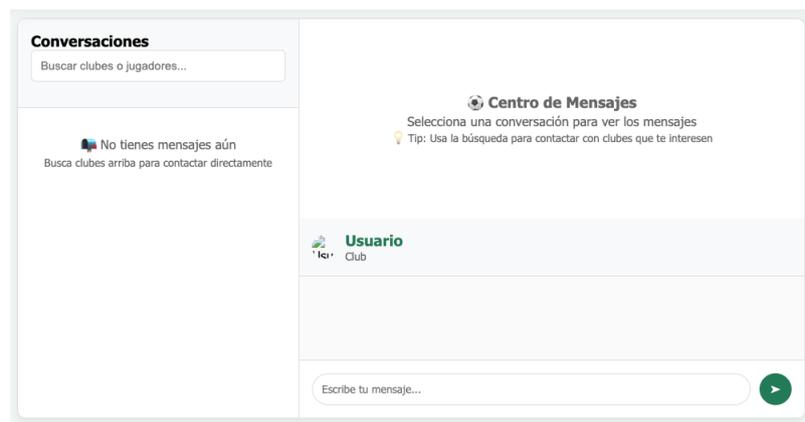
*Figura 11: Vista del panel de Ofertas para Clubes*

Esta funcionalidad aporta gran valor al sistema, gracias a que conecta oferta y demanda dentro del ecosistema deportivo. Está diseñado para ser ampliado con un mayor número de filtros de búsqueda, ya que ahora mismo solo permite filtrado por posición o ciudad.

## 6.5 MENSAJES

El sistema de mensajes es la tercera funcionalidad clave de este proyecto. Este sistema se implementa con el objetivo de facilitar la comunicación directa entre usuarios, permitiendo el acceso solo a aquellos usuarios que estén autenticados. Esta funcionalidad permite enviar y recibir mensajes en tiempo real, dejando un espacio para futuras mejoras.

Cada conversación se ha almacenado como una colección de mensajes individuales que contienen emisor, receptor, mensaje, fecha y hora del envío y estado de lectura. Cada uno de los mensajes es un documento dentro de la colección Mensaje, diseñado para consultas eficientes mediante filtros por usuarios.



*Figura 12: Vista del Sistema de Mensajes*

El envío de mensajes se realiza desde una pestaña independiente, en la que se podrán ver los distintos chats en un lado, y el chat con un usuario concreto en el otro. Al redactar y enviar un mensaje, se realiza una petición post con ID del mensaje y contenido. El backend verifica que existen emisor y receptor, y guarda el mensaje como no leído. Por último, se devuelve el mensaje formateado para mostrarse en la conversación de manera inmediata.

Dentro del bloque de conversaciones, podremos ver un listado de conversaciones recientes, mientras que la conversación completa se mostrará a la derecha. Ambas funcionan en base a una consulta GET, ya sea a conversaciones o a un usuario concreto, y se muestran ordenados cronológicamente.

Cada vez que el usuario accede a la conversación, se marcan como leídos todos los mensajes recibidos.

Por seguridad, el acceso a los mensajes solo se puede realizar en caso de ser emisor o receptor y se evitan envíos vacíos o a usuarios inexistentes.

Este sistema refuerza el componente social y profesional, dado que no hay un contexto específico en la conversación que mantengan los usuarios.

## **6.6 NOTIFICACIONES**

El sistema de notificaciones internas aportará al usuario conocimiento constantemente de las interacciones que se produzcan con su perfil o módulos asociados. Las principales notificaciones serán de interacciones con publicaciones, nuevos seguidores, nuevas ofertas de clubes a los que el usuario siga, actualizaciones en el estado de candidaturas que haya presentado el usuario, o nuevos mensajes recibidos. En el caso específico de los clubes, podrán recibir notificaciones periódicas sobre el número de candidatos que se han postulado para una de sus ofertas. De esta forma, el usuario se mantiene conectado con los eventos que ocurran en su red y mejora su experiencia de uso

En primer lugar, se definen los eventos que generarán notificaciones automáticas:

- Nuevo mensaje recibido
- Like o comentario en una publicación propia
- Nuevo candidato a una oferta (solo para clubes)
- Nuevo seguidor

Cada una de ellas se registra independientemente, asociada al receptor y con una referencia al tipo de evento.

La generación se implementa en los controladores del backend que corresponden, ya sea al guardar nuevo mensaje, registrar un comentario, dar «me gusta» o postularse a una oferta.

El acceso al listado de notificaciones se hará desde el header principal, de tal forma que al entrar en esta sección se realiza una petición GET que devuelve todas las notificaciones, ordenadas cronológicamente de más reciente a más antigua

Cada una muestra un mensaje resumen del evento, un icono, la fecha y el estado de lectura. Se marcarán como leídas aquellas notificaciones que el usuario abra.



*Figura 13: Vista del Panel de Notificaciones*

De cara al futuro, se ofrece la posibilidad de incorporar alertas por correo electrónico o notificaciones móviles, en caso de crear una aplicación móvil asociada.

Esta funcionalidad ofrece una mejora en la comunicación del sistema con el usuario final, permitiendo que esté informado en todo momento, pero sin llegar a ser intrusivo. Contribuye también a reforzar el carácter dinámico e interactivo de la plataforma.

## **6.7 BÚSQUEDA**

Por último, se ha implementado un sistema de búsqueda que permitirá localizar tanto personas como clubes a través de palabras clave. De esta forma, se fomenta la interacción entre los usuarios y se puede ampliar la red de contactos tanto a nivel profesional como a nivel personal.

El sistema está diseñado para permitir buscar perfiles mediante un campo de texto, mostrando resultados dinámicos basados en coincidencias parciales y redirigir al usuario a la página del perfil correspondiente al hacer clic en un resultado.

El frontend dispone de una barra de búsqueda en el propio header común, que redirige a la página de resultados al hacer “Enter” o al pulsar el icono de búsqueda. La respuesta contiene un conjunto de perfiles coincidentes representados en tarjetas con nombre, cabecera e imagen del perfil, junto con un enlace al perfil completo.



*Figura 14: Vista del Menú de Búsqueda*

El funcionamiento del backend es algo más complejo. En primer lugar, extrae el parámetro “q” de la URL, es decir, el parámetro de búsqueda, y realiza una consulta sobre la colección Usuario. Se le añade \$regex para habilitar las coincidencias parciales y no sensibles a mayúsculas o minúsculas. Devuelve un listado de usuarios que coinciden con los parámetros de búsqueda, llegando a limitarlo en caso de ser necesario.

Se puede implementar en un futuro un sistema de filtros, ordenación o sugerencias automáticas.

## **Capítulo 7. ANÁLISIS DE RESULTADOS**

El apartado de resultados tiene como finalidad recoger y analizar los logros funcionales alcanzados a lo largo del desarrollo del proyecto, contrastándolos con los objetivos previamente definidos. A través de este análisis se expone el grado de completitud de las distintas funcionalidades implementadas, su coherencia con las necesidades del usuario objetivo y su aportación práctica dentro del contexto del fútbol amateur.

Esta sección también permite identificar de forma crítica qué aspectos del sistema han funcionado según lo esperado y cuáles podrían beneficiarse de mejoras o ampliaciones. Todo ello contribuye a evaluar la madurez del producto, su aplicabilidad en entornos reales y su potencial para futuras evoluciones tecnológicas o comerciales. En definitiva, el análisis de resultados funcionales no solo evidencia el trabajo técnico realizado, sino también la capacidad de la herramienta para generar valor y responder a un problema real no resuelto.

### ***7.1 MÓDULO DE AUTENTICACIÓN***

Desde un punto de vista funcional, el módulo de autenticación ha respondido tal y como se esperaba. A lo largo de las pruebas realizadas con distintos tipos de usuarios —jugadores y clubes—, se ha comprobado que el sistema permite crear cuentas nuevas sin errores, valida correctamente el correo electrónico y evita accesos no autorizados. Cuando un usuario intenta iniciar sesión sin haber confirmado su cuenta o con credenciales incorrectas, el sistema reacciona con mensajes claros que guían al usuario sin frustrarlo, algo que era prioritario en la experiencia de uso.

El flujo de registro y activación resultó fluido, sin bloqueos ni incidencias relevantes. Además, durante las pruebas se confirmó que todas las rutas protegidas permanecen inaccesibles si no se presenta un token JWT válido. Esta protección ha sido esencial en funcionalidades como la edición del perfil, el envío de mensajes o la postulación a ofertas,

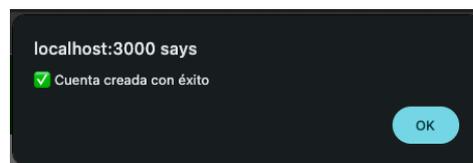
asegurando que únicamente los usuarios correctamente autenticados puedan operar con normalidad.



The screenshot shows a registration form with the following fields and elements:

- Título:** Crear cuenta
- Tipo de cuenta:** Dropdown menu with 'Persona' selected.
- Nombre completo:** Text input field containing 'Demo TFG'.
- Correo electrónico:** Text input field containing 'ejemplo@correo.com'.
- Contraseña:** Password input field with masked characters '\*\*\*\*\*'.
- Confirmar contraseña:** Password input field with masked characters '\*\*\*\*\*'.
- Botón:** 'Registrarse' button.

*Figura 15: Ejemplo de Formulario de Registro*



*Figura 16: Confirmación Visual de Registro*

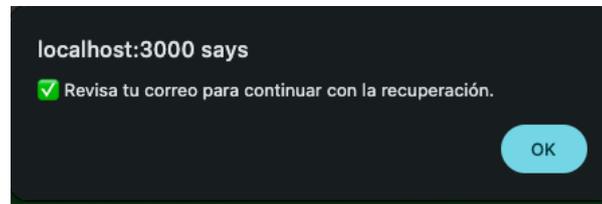
### **Cuenta verificada**

Tu cuenta ha sido verificada correctamente.

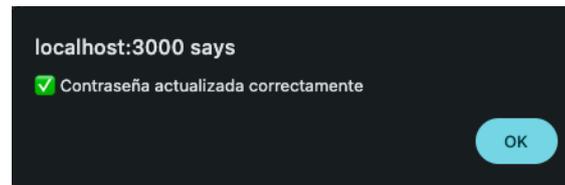
[Ir al login](#)

*Figura 17: Confirmación de Verificación*

Otro resultado positivo ha sido la correcta recuperación de contraseñas. Los usuarios que olvidaron sus credenciales pudieron iniciar el proceso de forma autónoma, recibir un enlace de restablecimiento y cambiar su contraseña desde un formulario seguro. En todos los casos evaluados, se observó un comportamiento intuitivo y sin incidencias.



*Figura 18: Mensaje Visual para Recuperación de Contraseña*



*Figura 19: Mensaje de Confirmación de Contraseña Actualizada*

En términos de experiencia de usuario, el diseño del proceso ha resultado sencillo y claro, tanto en el registro como en el login y recuperación. No se detectaron puntos de fricción importantes en la interacción, lo que refuerza la eficacia del sistema en contextos reales.

En conjunto, los resultados indican que el módulo de autenticación cumple de forma satisfactoria todos los requisitos funcionales definidos, ofreciendo una experiencia segura, intuitiva y alineada con los estándares actuales de plataformas profesionales.

## **7.2 MÓDULO DE CREACIÓN DE PERFIL**

El módulo de perfil ha funcionado justo como se esperaba, y eso es algo que vale la pena destacar. Durante las pruebas, tanto jugadores como clubes pudieron crear y personalizar su perfil sin complicaciones, eligiendo la información que querían mostrar y adaptándola a su rol. El sistema se comportó de forma inteligente: si el usuario era un club, aparecían los campos adecuados para ellos; si era un jugador, lo mismo. Esa diferenciación automática hizo que la experiencia fuera mucho más clara desde el principio.

<p><b>Perfil</b></p> <p>Perfil de demostración para la entrega del Trabajo de Fin de Grado</p> <p><b>Trayectoria</b></p> <ul style="list-style-type: none"> <li>ICAI (2021-2025)</li> <li>Defensa TFG 2025</li> </ul> <p><b>Capacidades del jugador</b></p> <ul style="list-style-type: none"> <li>Velocidad</li> <li>juego aéreo</li> <li>salida de balón</li> </ul> <p><b>Capacidades personales</b></p> <ul style="list-style-type: none"> <li>Liderazgo</li> <li>trabajo en equipo</li> </ul>	<p><b>Estadísticas individuales</b></p> <p>Goles: 15</p> <p>Asistencias: 10</p> <p>Partidos jugados: 20</p> <p>Minutos jugados: 2000</p> <p>Valoración media: 6.8</p> <p><b>Clasificación del equipo</b></p> <p>Equipo: Defensa TFG</p> <p>Posición en liga: 3º</p> <p>Puntos: 62</p> <p>PJ: 30   PG: 18   PE: 8   PP: 4</p>
---	--

Figura 20: Vista genérica del Perfil de Jugador

 <p><b>Celtic Castilla</b> Madrid</p> <p><a href="#">Editar perfil</a></p>	<p><b>Descripción del club</b></p> <p>El mejor club de Chamberí</p> <p><b>Metodología</b></p> <p>Club familiar</p> <p><b>Campo de juego</b></p> <p>Canal Isabel II</p> <p><b>Equipos del club</b></p> <ul style="list-style-type: none"> <li>Senior A</li> </ul>	<p><b>Última jornada</b></p> <p>Senior A 6 - 4 Robledo</p> <p><b>Logros temporada 2024/25</b></p> <p>2 ligas</p> <p><b>Contacto</b></p> <p>alexhueca2001@gmail.com</p> <p>Avenida Pablo Iglesias 15</p>
--	--	---

Figura 21: Vista Genérica del Perfil del Club

Uno de los aspectos más positivos fue lo intuitivo que resultó todo. Editar la información, cambiar la foto de perfil o subir una imagen de fondo se hacía de forma sencilla, sin pasos innecesarios ni errores inesperados. Además, la plataforma respondía bien: los cambios se guardaban correctamente, y el perfil quedaba visible con todo el contenido actualizado. A nivel técnico, se comprobó que nadie podía editar lo que no le correspondía, y eso refuerza la sensación de seguridad desde el punto de vista del usuario.

Otro punto fuerte fue la sensación de personalización. Cada perfil, aunque seguía una estructura común, transmitía algo único según lo que el usuario decidiera mostrar. Y eso en una red como esta —donde la identidad deportiva y profesional es tan importante— aporta un valor enorme. Muchos usuarios, de hecho, completaron su perfil sin necesidad de ayuda, lo que demuestra que la interfaz guía bien y no se vuelve confusa.



*Figura 22: Cabecera del Perfil de Jugador*

En resumen, se buscaba un módulo que permitiera a los usuarios construirse un perfil claro, completo y personalizado... y eso es exactamente lo que se ha conseguido. Todo funciona como debe, se adapta al tipo de usuario y transmite lo más importante: quién eres, qué haces y cómo quieres presentarte dentro de la comunidad.

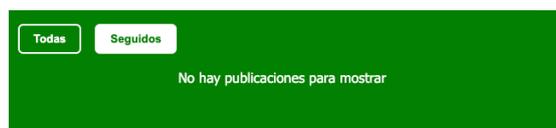
### **7.3 MÓDULO DE PUBLICACIONES**

El módulo de publicaciones ha sido uno de los más vivos y dinámicos de la plataforma, y ha respondido justo como se esperaba. Desde el principio se pensó como el centro social del sistema, el lugar donde jugadores y clubes pudieran expresarse, mostrarse y generar conversación. Y eso es exactamente lo que se ha conseguido.

Durante las pruebas, los usuarios interactuaron con naturalidad: crearon publicaciones con texto, las vieron reflejadas al instante en el feed y comenzaron a generar reacciones y comentarios con fluidez. El sistema respondió con solidez en todos los casos: los likes se actualizaban en tiempo real, los comentarios se mostraban ordenados, y no se detectaron errores ni bloqueos. Incluso el filtro que permite alternar entre publicaciones de toda la red o solo de personas seguidas funcionó sin fallos y aportó un nivel de personalización muy valorado.



*Figura 23: Feed de Publicaciones General*



*Figura 24: Feed de Publicaciones de Seguidos*

Más allá del funcionamiento técnico, lo más destacable fue la sensación de comunidad que generó este módulo. Los perfiles no quedaron como espacios estáticos, sino que se convirtieron en puntos de encuentro, donde los usuarios podían descubrir lo que hacían los demás, animarse entre ellos o simplemente dejar constancia de sus experiencias. Ese tipo de interacción le dio vida al proyecto.

En resumen, se buscaba un espacio participativo, sencillo de usar, que invitara a compartir y conectar. Y eso es lo que el módulo de publicaciones ha ofrecido. No solo ha funcionado bien: ha ayudado a que la red cobre sentido.

## **7.4 MÓDULO DE OFERTAS**

El módulo de ofertas ha sido una de las piezas clave para conectar la parte social con la parte más profesional del sistema, y el resultado ha estado a la altura de lo previsto. Desde el

principio, se diseñó con la idea de ofrecer a los clubes una herramienta útil para publicar vacantes o necesidades dentro de sus equipos, y a los jugadores una vía directa para postularse a ellas. Y en la práctica, esa conexión ha funcionado como se esperaba.

Durante las pruebas, los clubes pudieron crear ofertas con facilidad, incluyendo descripciones claras, requisitos y condiciones. El sistema permitió editar y eliminar las ofertas activas sin errores, lo que ofreció flexibilidad ante cualquier cambio de última hora.



Figura 25: Vista de Oferta Creada



The image shows the "Editar Oferta" (Edit Offer) form. At the top, it says "Oferta: Oferta de TFG", "Creada: 29/6/2025", and "Candidatos: 0". Below this, there are three input fields: "Título de la oferta \*" with the value "Oferta de TFG", "Posición \*" with a dropdown menu showing "Defensa Central", and "Ubicación \*" with the value "ICAI".

Figura 26: Formulario de Edición de Oferta



Figura 27: Versión Actualizada de la Oferta

Por su parte, los jugadores encontraron sencillo el proceso de postulación: con un solo clic podían inscribirse en la oferta que les interesara, y el sistema evitaba que se postularan dos veces por error. Además, una vez inscritos, la oferta mostraba visualmente que ya estaban apuntados, lo cual añadió claridad y orden al proceso.



Figura 28: Vista de Oferta para Jugadores

También se validó que los clubes pudieran acceder al listado de candidatos desde su perfil, revisarlo en detalle y gestionar fácilmente sus decisiones. No hubo fallos en la vinculación entre las ofertas y los jugadores postulados, lo que hizo que la gestión de candidaturas fuera ordenada y sin fricciones.

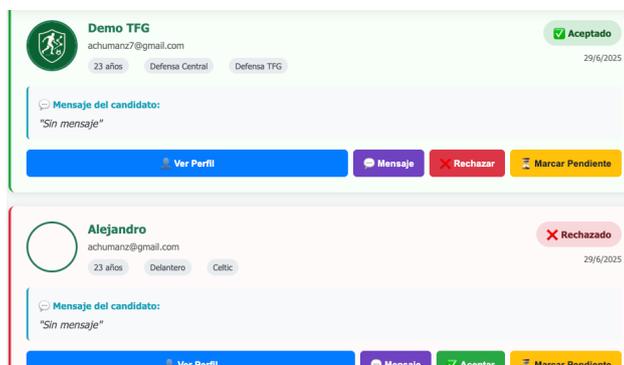


Figura 29: Vista de Lista de Candidatos a una Oferta



Figura 30: Vista Actualizada de la Oferta

Lo más interesante fue ver cómo esta funcionalidad abría nuevas posibilidades dentro del ecosistema amateur: jugadores jóvenes veían en ella una oportunidad para crecer, mientras que los clubes agradecían poder encontrar perfiles que no habrían conocido de otra manera.

En resumen, se esperaba un sistema para crear, postular y gestionar ofertas deportivas con agilidad y sin complicaciones. Y lo que se ha conseguido es justo eso: un módulo estable, útil y con potencial real para generar oportunidades dentro del fútbol amateur.

## 7.5 MÓDULO DE MENSAJES

El módulo de mensajes ha aportado justo lo que se buscaba desde el inicio: una forma natural y directa de que los usuarios puedan hablar entre ellos. Al final, por mucho que haya perfiles bonitos o publicaciones llamativas, la conversación es lo que da vida a cualquier red. En este caso, el sistema ha sabido estar a la altura.

Durante las pruebas, enviar y recibir mensajes fue un proceso fácil y sin trabas. Los chats se creaban al instante, los mensajes aparecían en pantalla sin retraso y todo tenía esa fluidez que se espera de una conversación digital hoy en día. Los usuarios podían mantener varias charlas a la vez, cambiar entre ellas sin perder el hilo y confiar en que todo quedaría guardado tal y como lo dejaron.



Figura 31: Vista de Chat (Emisor)

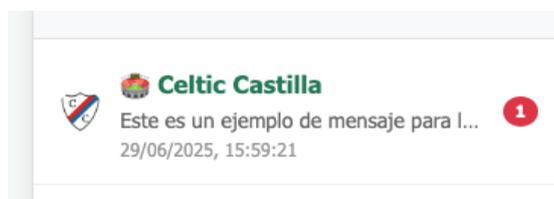


Figura 32: Bandeja de Entrada de Mensajes



*Figura 33: Vista de Chat (Receptor)*

Pero más allá de lo técnico, lo importante es cómo lo percibieron quienes lo usaron. La sensación fue positiva: se sintieron cómodos, seguros y con libertad para conectar. Saber que los mensajes están protegidos, que solo tú y la otra persona podéis verlos, hace que uno se exprese con más confianza. Además, el sistema marcaba automáticamente los mensajes como leídos cuando se abría una conversación, algo que hizo que todo fluyera de forma ordenada, sin duplicidades ni confusiones.

En resumen, este módulo no solo ha funcionado bien, sino que ha aportado calidez a una plataforma pensada para unir personas y oportunidades. La posibilidad de escribirle a un club que te interesa o de hablar con un jugador que te ha llamado la atención es precisamente lo que convierte una red social en una red viva. Y eso, aquí, se ha conseguido.

## **7.6 MÓDULO DE NOTIFICACIONES**

El módulo de notificaciones tenía como objetivo mantener a los usuarios informados en todo momento sobre las interacciones relevantes dentro de la plataforma, sin ser invasivo ni generar distracciones innecesarias. Se esperaba que cada evento importante —como recibir un nuevo mensaje, ser seguido por otro usuario, recibir un like o comentario en una publicación, o postularse a una oferta— se reflejara en una alerta clara y accesible desde el propio sistema. Además, era fundamental que las notificaciones pudieran marcarse como leídas para mantener organizado el panel del usuario.

Durante las pruebas, el comportamiento del módulo cumplió con lo previsto. Los usuarios recibieron notificaciones en tiempo real cuando interactuaban entre sí, y estas se

almacenaban correctamente en el panel correspondiente. El icono de alertas reflejaba el número exacto de eventos pendientes y permitía acceder con rapidez al listado completo. Cada notificación incluía un texto breve, la hora en que ocurrió y el tipo de acción asociada, lo que facilitó al usuario entender de inmediato qué había pasado y si requería alguna respuesta.

Además, se comprobó que al abrir el panel y consultar las notificaciones, estas se marcaban como leídas, permitiendo mantener un seguimiento ordenado sin duplicidades. La navegación entre notificaciones resultó fluida, sin errores visibles ni problemas de sincronización.



*Figura 34: Vista Actualizada de Panel de Notificaciones*

Uno de los puntos más valorados fue que el sistema no saturó al usuario con información innecesaria: únicamente se notificaron acciones que realmente aportaban valor, lo que reforzó la sensación de control y utilidad. En conjunto, este módulo aportó dinamismo a la experiencia, conectando al usuario con lo que sucede a su alrededor sin necesidad de estar pendiente constantemente.

En definitiva, se buscaba un sistema de alertas útil, claro y bien integrado con el resto de funcionalidades, y eso es exactamente lo que se ha conseguido. Las notificaciones no solo informan: ayudan a que el usuario se mantenga conectado al pulso de la red.

## 7.7 MÓDULO DE BÚSQUEDA

El módulo de búsqueda tenía un papel muy claro dentro de la plataforma: facilitar que los usuarios encontraran rápidamente a otros perfiles, ya fueran jugadores o clubes, sin tener que navegar manualmente por listas interminables. La idea era que con tan solo escribir un nombre, una ciudad o una palabra clave, el sistema devolviera resultados relevantes, bien presentados y con acceso directo a los perfiles encontrados.

Y lo cierto es que el comportamiento del sistema ha estado muy en línea con esa expectativa. Durante las pruebas, al introducir cualquier término de búsqueda, el sistema respondió con rapidez y precisión, mostrando una serie de tarjetas visuales que incluían la información básica del usuario (nombre, imagen y tipo de perfil) y que permitían acceder al detalle con un solo clic. No se detectaron errores ni resultados incoherentes; incluso búsquedas con palabras incompletas o mayúsculas mezcladas funcionaron sin problema, lo cual aporta mucha flexibilidad.

Además, la barra de búsqueda siempre estuvo visible y accesible desde cualquier vista, lo que facilitó su uso en distintos momentos. Se comprobó que el backend interpretaba correctamente los parámetros introducidos, utilizaba expresiones regulares para encontrar coincidencias parciales, y que el resultado era ordenado y útil.

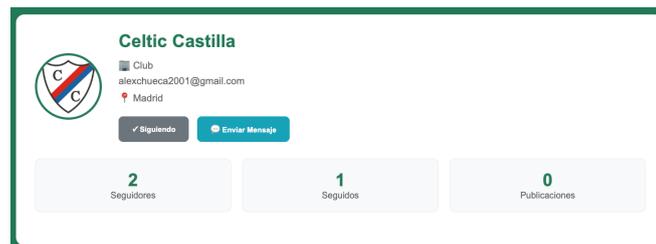


*Figura 35: Vista de Resultados de Búsqueda*

Quizá lo más destacable no fue solo que encontrara lo que se buscaba, sino lo sencillo que resultó usarla. Los usuarios no necesitaron instrucciones ni menús complejos: escribieron lo

que tenían en mente y lo encontraron. Eso, en una red social que pretende conectar personas y oportunidades, es una funcionalidad que marca la diferencia.

Adicionalmente, la vista de resultados de búsqueda ofrecía acceso al perfil de los usuarios encontrados, junto con la opción de seguir ese perfil o enviar un mensaje.



*Figura 36: Vista de Perfil Ajeno en Resultados de Búsqueda*

En resumen, se buscaba un sistema de búsqueda útil, accesible y efectivo. Y eso es exactamente lo que ha ofrecido. Ha cumplido su función sin complicaciones, aportando claridad, rapidez y valor a la experiencia general de la plataforma.

En conjunto, el análisis funcional de los distintos módulos desarrollados permite afirmar que la plataforma cumple con los objetivos fundamentales con los que fue concebida: facilitar la conexión entre futbolistas y clubes amateurs, potenciar su visibilidad y habilitar un entorno de relación activo, accesible y útil. Cada módulo ha aportado su parte —ya sea garantizando un acceso seguro, favoreciendo la creación de una identidad digital, permitiendo la interacción social o facilitando oportunidades reales de crecimiento— y el sistema, como conjunto, ha mostrado una coherencia técnica y una solidez funcional que lo posicionan como una herramienta válida y prometedora en el contexto del fútbol amateur. Estos resultados no solo validan el planteamiento inicial, sino que abren la puerta a su evolución futura.

## **Capítulo 8. CONCLUSIONES Y TRABAJOS FUTUROS**

Este proyecto ha sido, en el fondo, un homenaje al esfuerzo cotidiano que realizan quienes viven el fútbol lejos de los focos: esos futbolistas que entrenan después del trabajo, esos entrenadores que organizan entrenamientos en campos prestados, esos clubes que funcionan con ilusión más que con presupuesto. Cada línea de código y cada decisión de diseño han estado impulsadas por la misma pregunta: “¿Cómo podemos dignificar esa pasión y facilitar que encuentren su sitio?”

Cada módulo, desde la autenticación hasta las notificaciones, se ha pensado con una mirada: ser útil sin abrumar, potenciar el talento sin complejidad. Hemos integrado seguridad avanzada y arquitecturas escalables, pero siempre con la premisa de que bastara con pocos clics para conectar con alguien que comparte tu misma pasión. Y eso se ha logrado: el feed vibra con publicaciones de goles inesperados, el buzón de ofertas brilla con oportunidades reales, el chat guarda conversaciones cargadas de emoción.

Con este proyecto, se ha dado forma a la posibilidad de que un sueño amateur tenga un escaparate digital digno. Sin embargo, no podemos cerrar la puerta a mejorar.

Mirando hacia el futuro, hay varios frentes con los que podemos seguir enriqueciendo la plataforma y convertirla en un recurso todavía más valioso para el fútbol amateur:

Primero de todo, se atenderá a posibles mejoras visuales o funcionales de todo lo implementado, de acuerdo con la respuesta que se obtenga por parte de los usuarios. Antes de avanzar, se debe certificar que el camino recorrido es verdaderamente útil y construir las mejoras sobre una buena base.

Una vez que se obtiene esa buena base, una de las primeras mejoras que puede surgir es la experiencia móvil: hoy muchos usuarios navegan desde el teléfono, así que lanzar una aplicación nativa o evolucionar la web hacia una PWA (Progressive Web App) con

notificaciones push haría que nadie se pierda un mensaje, una oferta o un comentario relevante, incluso cuando no estén pendientes del navegador.

En paralelo, podríamos integrar un motor de recomendaciones inteligente: a partir del historial de navegación, publicaciones ‘likeadas’ y ofertas consultadas, el sistema sugeriría jugadores afines, clubes interesantes o vacantes que encajen mejor con el perfil de cada usuario. Sería como contar con un agente que, de forma proactiva, te acerca oportunidades sin que tengas que buscarlas manualmente.

Otro salto de calidad vendría de la mano de la gamificación: pequeños retos semanales (por ejemplo, “comenta a tres usuarios nuevos” o “participa en dos ofertas esta semana”) podrían premiarse con insignias, visibilidad extra en el feed o cupones de terceros. Esto no solo incentiva la actividad, sino que crea un espíritu de comunidad y competición sana.

En el terreno más técnico, conviene pensar en la analítica avanzada. Un panel para cada club en el que puedan ver métricas —qué tipo de jugadores visita su perfil, de dónde vienen, qué publicaciones generan más interacción— ayudaría a tomar decisiones informadas sobre la estrategia de captación o la gestión de su marca.

No menos importante es la accesibilidad: podemos revisar contrastes de color, etiquetas ARIA (Accesible Rich Internet Applications, etiquetas que mejoran la accesibilidad para personas con discapacidad, proporcionando información adicional a los lectores de pantalla y otras tecnologías de asistencia) [31] y la navegación a través del teclado para asegurarnos de que cualquier persona —con discapacidad visual o motriz— pueda usar la plataforma sin barreras. Del mismo modo, la internacionalización abriría la red a futbolistas de otros países, traduciéndola al inglés, al francés o al portugués.

Por último, la integración con servicios externos enriquecería el ecosistema: un acuerdo con plataformas de streaming podría permitir compartir vídeos de partidos; con proveedores de entrenamiento online se podría ofrecer formación a los usuarios; y, en un futuro, hasta podríamos explorar micropagos para destacar ofertas o perfiles, siempre manteniendo el enfoque amateur y sin encarecer la experiencia.

En conjunto, todas estas mejoras no rompen el espíritu del proyecto, sino que lo potencian: ofrecen más vías de conexión, más información de valor y más herramientas para que cada futbolista o club amateur tenga, literalmente, el mundo del fútbol a un solo clic.

## Capítulo 9. BIBLIOGRAFÍA

- [1] Musciano, C., & Kennedy, B. *The Complete Reference: HTML & CSS (Fifth Edition)*. McGraw-Hill Education, 2008. Capítulos 1 y 4. <https://www.dcehvp.com/E-Content/BCA/BCA-II/Web%20Technology/the-complete-reference-html-css-fifth-edition.pdf>
- [2] Haverbeke, M. *Eloquent JavaScript: A Modern Introduction to Programming* (3rd Edition). No Starch Press, 2018. Capítulo 1. <https://eloquentjavascript.net/>
- [3] Node.js Foundation. “Introduction to Node.js”, Node.js Learn, 2023. <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>
- [4] Express.js Foundation. “Express – Node.js web application framework”, Express Documentation, 2023. <https://expressjs.com/>
- [5] Fielding, R. T. “Architectural Styles and the Design of Network-based Software Architectures”. Doctoral dissertation, University of California, Irvine, 2000. <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [6] Fielding, R., Reschke, J. *"Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content."* RFC 7231, IETF, Junio 2014. <https://datatracker.ietf.org/doc/html/rfc7231>
- [7] Jones, M.; Bradley, J.; Sakimura, N. *"JSON Web Token (JWT)"*, RFC 7519, IETF, Mayo 2015. <https://datatracker.ietf.org/doc/html/rfc7519>
- [8] Nodemailer. “Nodemailer – Send e-mails with Node.js”, Nodemailer Documentation, 2023. <https://nodemailer.com/about/>
- [9] Klensin, J. *"Simple Mail Transfer Protocol"*, RFC 5321, IETF, Octubre 2008. <https://datatracker.ietf.org/doc/html/rfc5321>
- [10] Multer. “Multer – Node.js middleware for handling multipart/form-data”, Multer Docs, 2023. <https://github.com/expressjs/multer>
- [11] Express.js. “Serving static files in Express”, Express Documentation, 2023. <https://expressjs.com/en/starter/static-files.html>
- [12] Express.js. “Using middleware”, Express Documentation, 2023. <https://expressjs.com/en/guide/using-middleware.html>
- [13] bcryptjs. *"bcryptjs – Optimized bcrypt in JavaScript with zero dependencies"*, GitHub Repository, 2023. <https://github.com/dcodeIO/bcrypt.js>

- [14] Motdotla. “Dotenv – Loads environment variables from .env file”, npmjs.com, 2024. <https://www.npmjs.com/package/dotenv>
- [15] Express.js. “Using CORS middleware in Express”, Express Documentation, 2023. <https://expressjs.com/en/resources/middleware/cors.html>
- [16] MongoDB Inc. “What is MongoDB?”, MongoDB Documentation, 2023. <https://www.mongodb.com/docs/manual/introduction/>
- [17] Mongoose. “Mongoose – Elegant MongoDB object modeling for Node.js”, Mongoose Documentation, 2023. <https://mongoosejs.com>
- [18] D. M. Boyd and N. B. Ellison, “Social network sites: Definition, history, and scholarship,” *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007. <https://doi.org/10.1111/j.1083-6101.2007.00393.x>
- [19] A. M. Kaplan and M. Haenlein, “Users of the world, unite! The challenges and opportunities of Social Media,” *Business Horizons*, vol. 53, no. 1, pp. 59–68, 2010. <https://doi.org/10.1016/j.bushor.2009.09.003>
- [20] LinkedIn Corporation, “About LinkedIn,” 2024. <https://about.linkedin.com/>
- [21] T. O’Reilly, “What is Web 2.0: Design patterns and business models for the next generation of software,” *Communications & Strategies*, no. 65, pp. 17–37, 2007. <https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>
- [22] M. Field, *RESTful Web APIs*, O’Reilly Media, 2022. <https://www.oreilly.com/library/view/restful-web-apis/9781449358068/>
- [23] L. Richardson and S. Ruby, *RESTful Web Services*, O’Reilly Media, 2007. <https://www.oreilly.com/library/view/restful-web-services/9780596529260/>
- [24] FIFA, “Who we are,” 2023. <https://www.fifa.com/about-fifa>
- [25] UEFA, “About UEFA,” 2023. <https://www.uefa.com/insideuefa/about-uefa/>
- [26] Real Federación Española de Fútbol (RFEF), “Competiciones y estructura,” 2023. <https://rfef.es/>
- [27] Transfermarkt, “Football Database,” 2024. <https://www.transfermarkt.com/>
- [28] BeSoccer, “All the football in one app,” 2024. <https://www.besoccer.com/>
- [29] FutbolJobs, “Trabajos en el mundo del fútbol,” 2024. <https://futboljobs.com/>
- [30] Footvisual, “Red social para futbolistas,” 2024. <https://footvisual.com/>
- [31] J. Sotoux, “¿Qué son las etiquetas ARIA y para qué sirven?”, *jsotoux.com*, 2023. [Online]. Available: <https://jsotoux.com/que-son-las-etiquetas-aria/#:~:text=ARIA:%20Estados,estado%20actual%20de%20la%20página.>

# **ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS**

ODS 3. Salud y bienestar La plataforma impulsa la práctica deportiva a nivel amateur, facilitando el acceso a entrenamientos, partidos y redes de apoyo entre futbolistas y clubes. Al conectar a sus usuarios, fomenta estilos de vida activos y la prevención de sedentarismo, con el consiguiente impacto positivo en la salud física y mental.

ODS 4. Educación de calidad Aunque no es una herramienta académica, la red social funciona como un espacio de aprendizaje informal: los usuarios pueden compartir contenidos formativos (tácticas, ejercicios, consejos de nutrición) y entablar mentorías entre generaciones de futbolistas, enriqueciendo el conocimiento colectivo del deporte.

ODS 8. Trabajo decente y crecimiento económico Al convertirse en escaparate de talento y ofertas, la aplicación crea vías de inserción laboral dentro del fútbol base: los jugadores amplían sus oportunidades de fichaje o patrocinio, y los clubes acceden a perfiles mejor cualificados. Esto promueve empleos dignos, incluso en entornos con pocos recursos.

ODS 9. Industria, innovación e infraestructuras El proyecto desarrolla una infraestructura digital escalable y segura (API RESTful, JWT, MongoDB), que puede replicarse o extenderse a otras disciplinas deportivas. Su enfoque modular favorece la innovación continua y la integración con sistemas externos (servicios de streaming, analítica deportiva, etc.).

ODS 10. Reducción de las desigualdades Al dirigirse a futbolistas y clubes amateurs—un segmento tradicionalmente infrarepresentado—la plataforma contribuye a equilibrar la visibilidad y las oportunidades frente a ligas profesionales. Rompe barreras geográficas y económicas, facilitando que cualquier participante del fútbol base haga valer su talento.

ODS 17. Alianzas para lograr los objetivos El diseño de la red social promueve alianzas: entre clubes y jugadores, entre entidades formativas (escuelas o academias) y la plataforma misma. Estas colaboraciones pueden extenderse en el futuro a federaciones, patrocinadores o instituciones educativas, generando sinergias que potencien el ecosistema del fútbol amateur.

## ANEXO II. MODELOS DE LA BASE DE DATOS

### 9.1.1 USUARIO

```
const mongoose = require('mongoose');

const usuarioSchema = new mongoose.Schema({
  nombre: { type: String, required: true },
  email: { type: String, required: true, unique: true },
  passwordHash: { type: String, required: true },

  tipoUsuario: { type: String, enum: ['persona', 'club'], required: true },
  publicId: { type: String, unique: true },

  confirmado: { type: Boolean, default: false },
  tokenConfirmacion: { type: String },
  resetToken: { type: String },
  resetTokenExpira: { type: Date },

  // Imágenes
  fotoPerfil: String,
  fondoPerfil: String,
  fotoBanner: String, // para clubes
  bannerPersona: { type: String, default: null },

  // Perfil persona (jugador)
  edad: Number,
  posicion: String,
  equipo: String,
  perfilTexto: String,
  trayectoria: String,
  capacidadesJugador: String,
  capacidadesPersonales: String,

  goles: Number,
  asistencias: Number,
  partidosJugados: Number,
  minutosJugados: Number,
  valoracionMedia: Number,

  // Perfil club
  ciudad: String,
  descripcionClub: String,
  metodologia: String,
  campo: String,
  equiposClub: String,
  jornada: String,
  logros: String,
  emailContacto: String,
```

```
ubicacion: String,  
  
// Seguidores y seguidos  
seguidores: [String],  
seguidos: [String],  
  
// Notificaciones  
notificaciones: [{  
  tipo: String,  
  mensaje: String,  
  leida: { type: Boolean, default: false },  
  fecha: { type: Date, default: Date.now }  
}]  
}, {  
  timestamps: true  
});  
  
module.exports = mongoose.model('Usuario', usuarioSchema);
```

## 9.1.2 PUBLICACIÓN

```
const mongoose = require('mongoose');  
  
const comentarioSchema = new mongoose.Schema({  
  autor: { type: String, required: true },  
  texto: { type: String, required: true },  
  fecha: { type: Date, default: Date.now }  
});  
  
const publicacionSchema = new mongoose.Schema({  
  autor: { type: String, required: true },  
  texto: { type: String, required: true },  
  imagen: { type: String },  
  fecha: { type: Date, default: Date.now },  
  likes: [String],  
  comentarios: [comentarioSchema]  
}, {  
  timestamps: true  
});  
  
module.exports = mongoose.model('Publicacion', publicacionSchema);
```

## 9.1.3 COMENTARIO

```
const mongoose = require('mongoose');  
  
const comentarioSchema = new mongoose.Schema({
```

```
publicacionId: { type: mongoose.Schema.Types.ObjectId, ref: 'Publicacion',
required: true },
autor: { type: String, required: true },
contenido: { type: String, required: true },
creadoEn: { type: Date, default: Date.now }
});

module.exports = mongoose.model('Comentario', comentarioSchema);
```

### 9.1.4 OFERTA

```
const mongoose = require('mongoose');

const ofertaSchema = new mongoose.Schema({
  titulo: { type: String, required: true },
  descripcion: { type: String },
  posicion: { type: String, required: true },
  creador: { type: String, required: true },
  requisitos: { type: String },
  ubicacion: { type: String },
  estado: { type: String, enum: ['activa', 'cerrada'], default: 'activa' },
  candidatos: [{
    email: String,
    estado: { type: String, enum: ['pendiente', 'aceptado', 'rechazado'],
default: 'pendiente' },
    mensaje: String,
    fechaPostulacion: { type: Date, default: Date.now }
  ]},
  creadoEn: { type: Date, default: Date.now }
});

module.exports = mongoose.model('Oferta', ofertaSchema);
```

### 9.1.5 MENSAJE

```
const mongoose = require('mongoose');

const mensajeSchema = new mongoose.Schema({
  emisor: {
    type: String,
    required: true,
    index: true
  },
  },
```

```
receptor: {
  type: String,
  required: true,
  index: true
},
contenido: {
  type: String,
  required: true,
  maxlength: 1000
},
asunto: {
  type: String,
  default: 'Sin asunto',
  maxlength: 100
},
leido: {
  type: Boolean,
  default: false,
  index: true
},
enviadoEn: {
  type: Date,
  default: Date.now,
  index: true
},
// Metadatos adicionales
tipo: {
  type: String,
  enum: ['mensaje', 'notificacion_oferta', 'respuesta_postulacion'],
  default: 'mensaje'
},
referenciaId: {
  type: String, // ID de oferta, publicación, etc.
  default: null
},
eliminadoPor: [{
  usuario: String,
  fecha: { type: Date, default: Date.now }
}]
}, {
  timestamps: true // Añade createdAt y updatedAt automáticamente
});

// Índices compuestos para optimizar consultas frecuentes
mensajeSchema.index({ emisor: 1, receptor: 1, enviadoEn: -1 });
mensajeSchema.index({ receptor: 1, leido: 1 });

// Método para marcar como leído
mensajeSchema.methods.marcarComoLeido = function() {
  this.leido = true;
  return this.save();
};
```

```
// Método estático para obtener conversaciones de un usuario
mensajeSchema.statics.obtenerConversaciones = async function(emailUsuario) {
  const pipeline = [
    {
      $match: {
        $or: [
          { emisor: emailUsuario },
          { receptor: emailUsuario }
        ]
      }
    },
    {
      $sort: { enviadoEn: -1 }
    },
    {
      $group: {
        _id: {
          $cond: [
            { $eq: ['$emisor', emailUsuario] },
            '$receptor',
            '$emisor'
          ]
        },
        ultimoMensaje: { $first: '$$ROOT' }
      }
    },
    {
      $replaceRoot: { newRoot: '$ultimoMensaje' }
    }
  ];

  return this.aggregate(pipeline);
};

// Método estático para contar mensajes no leídos
mensajeSchema.statics.contarNoLeídos = function(emailUsuario) {
  return this.countDocuments({
    receptor: emailUsuario,
    leído: false
  });
};

// Middleware pre-save para validaciones adicionales
mensajeSchema.pre('save', function(next) {
  // Asegurar que emisor y receptor no sean iguales
  if (this.emisor === this.receptor) {
    const error = new Error('Un usuario no puede enviarse mensajes a sí mismo');
    return next(error);
  }
}

// Trimear el contenido
if (this.contenido) {
  this.contenido = this.contenido.trim();
}
```

```
}  
  
  next();  
});  
  
// Método virtual para obtener el "otro usuario" en una conversación  
mensajeSchema.virtual('otroUsuario').get(function() {  
  
  return this.emisor === this.usuarioActual ? this.receptor : this.emisor;  
});  
  
module.exports = mongoose.model('Mensaje', mensajeSchema);
```

### 9.1.6 NOTIFICACIÓN

```
const mongoose = require('mongoose');  
  
const notificacionSchema = new mongoose.Schema({  
  receptor: {  
    type: String,  
    required: true,  
    index: true // Para consultas rápidas por usuario  
  },  
  
  tipo: {  
    type: String,  
    enum: ['like', 'comentario', 'seguidor', 'mensaje', 'oferta', 'sistema'],  
    required: true  
  },  
  
  mensaje: {  
    type: String,  
    required: true,  
    maxlength: 200  
  },  
  
  leida: {  
    type: Boolean,  
    default: false,  
    index: true // Para consultas de no leídas  
  },  
  
  creadaEn: {  
    type: Date,  
    default: Date.now,  
    index: true // Para ordenar por fecha  
  },  
  
  // Referencias opcionales según el tipo  
  publicacionId: {
```

```
type: mongoose.Schema.Types.ObjectId,
ref: 'Publicacion',
sparse: true // Solo algunos documentos tendrán este campo
},

emisor: {
  type: String, // email del usuario que genera la notificación
  sparse: true
},

ofertaId: {
  type: mongoose.Schema.Types.ObjectId,
  ref: 'Oferta',
  sparse: true
},

mensajeId: {
  type: mongoose.Schema.Types.ObjectId,
  ref: 'Mensaje',
  sparse: true
},

// Metadatos adicionales
metadata: {
  type: Map,
  of: mongoose.Schema.Types.Mixed,
  default: new Map()
}
}, {
  timestamps: true // Crea automáticamente createdAt y updatedAt
});

// Índices compuestos para optimizar consultas
notificacionSchema.index({ receptor: 1, creadaEn: -1 });
notificacionSchema.index({ receptor: 1, leida: 1 });
notificacionSchema.index({ receptor: 1, tipo: 1 });

// Middleware para limpiar notificaciones antiguas
notificacionSchema.statics.limpiarAntiguas = function(diasAntiguedad = 30) {
  const fechaLimite = new Date();
  fechaLimite.setDate(fechaLimite.getDate() - diasAntiguedad);

  return this.deleteMany({
    creadaEn: { $lt: fechaLimite },
    leida: true
  });
};

// Método para marcar como leida
notificacionSchema.methods.marcarComoLeida = function() {
  this.leida = true;
  return this.save();
};
```

```
// Método estático para crear notificación
notificacionSchema.statics.crear = async function(datos) {
  try {
    const notificacion = new this(datos);
    await notificacion.save();
    return notificacion;
  } catch (error) {
    console.error('✘ Error al crear notificación:', error);
    throw error;
  }
};

// Método estático para obtener conteo de no leídas
notificacionSchema.statics.contarNoLeidas = function(receptor) {
  return this.countDocuments({ receptor, leida: false });
};

// Middleware pre-save para validaciones adicionales
notificacionSchema.pre('save', function(next) {
  // Validar que el emisor no sea el mismo que el receptor
  if (this.emisor && this.emisor === this.receptor) {
    const error = new Error('El emisor no puede ser el mismo que el receptor');
    return next(error);
  }
  next();
});

// Método virtual para tiempo transcurrido
notificacionSchema.virtual('tiempoTranscurrido').get(function() {
  const ahora = new Date();
  const diff = ahora - this.creadaEn;

  const segundos = Math.floor(diff / 1000);
  const minutos = Math.floor(segundos / 60);
  const horas = Math.floor(minutos / 60);
  const dias = Math.floor(horas / 24);

  if (dias > 0) return `hace ${dias} día${dias > 1 ? 's' : ''}`;
  if (horas > 0) return `hace ${horas} hora${horas > 1 ? 's' : ''}`;
  if (minutos > 0) return `hace ${minutos} minuto${minutos > 1 ? 's' : ''}`;
  return 'ahora mismo';
});

// Configurar virtuals en JSON
notificacionSchema.set('toJSON', { virtuals: true });
notificacionSchema.set('toObject', { virtuals: true });

module.exports = mongoose.model('Notificacion', notificacionSchema);
```