

# Propuesta proyecto FINAL

3	Título y descripción
4	Motivación
5	Stack
8	Historias de usuario
11	Base de datos
14	Repositorio

## TÍTULO DE LA APLICACIÓN



**FitConnet**  
LET'S THRIVE

## DESCRIPCIÓN

Esta app esta diseñada para funcionar como una aplicación dual, es decir, funciona como una red social y una de salud y bien estar.

En ella podrás, mantener un historial de tu actividades física pero además podrás compartirla con tus amigos que tengan la app.

Esta pensada para crear comunidad y hacer planes de entreno conjuntos, además de compartir los entrenos individuales en el feed.

## MOTIVACIÓN

Esta idea nace reciclada de un proyecto que se nos mandó hacer en la dual, en el cual se nos daban 3 palabras clave de las cuales saliera una idea para una aplicación, las mías fueron: Humanidad – Naturaleza – Gráfica.

Lo primero que se me ocurrió fue hacer una aplicación simple de fitness, donde registraras tu actividad y progreso, sin embargo pensé ya había demasiadas aplicaciones y quería hacer algo más novedoso, diferente.

Fue ahí cuando se me ocurrió hacer la aplicación como una red social, esto dá significado al eslogan, ya que este significa prosperemos, vamos a prosperar o a prosperar. Tiene un mensaje de comunidad que se ayuda a mejorar en el área de la salud, tema que me es especialmente relevante porque antes de programador soy nutricionista.

Además es una idea poco explotada ya que hay menos de 5 apps en el mercado, de las cuáles las más conocidas son: Fitbit, MapMyFitness y Nike Training Club.

Sin embargo, todas tienen partes de pago para funciones premium y están más enfocadas al entrenamiento con planes y programas, mientras que mi app esta más enfocada en la parte social así como instagram.

---

01 Springboot: jpa, security, web, devtools, test

---

02 Java 21

---

03 MYSQL

---

04 Apache Commons Lang

---

05 Project Lombok

---

06 JSON Web Token

---

07 JavaFaker

---

08 Hibernate Validator

---

09 Jakarta Expression Language

---

10 SpringDoc OpenAPI

---

11 Swagger

---

12 MapStruct

---

## STACK TECNOLÓGICO

### FRONTED

---

01 React (Next.js)

---

02 html

---

03 SCSS

---

04 Bootstrap

---

Las tecnologías seleccionadas para el desarrollo del backend de la aplicación, como Spring Boot, Hibernate Validator y JSON Web Tokens (JWT), ofrecen una combinación robusta y altamente segura para una red social de compartición de entrenamientos. Spring Boot proporciona un entorno de desarrollo ágil y fácil de configurar, permitiendo una rápida implementación de servicios RESTful para la gestión de usuarios, publicaciones y relaciones sociales. Hibernate Validator, junto con Jakarta Expression Language (EL), asegura la validez y consistencia de los datos ingresados por los usuarios, garantizando la integridad de la información en la plataforma. La utilización de JSON Web Tokens (JWT) para la autenticación y autorización proporciona una capa adicional de seguridad, permitiendo un manejo seguro de sesiones de usuario y acceso controlado a recursos sensibles.

Por otro lado, la elección de React con el framework Next.js y Bootstrap para el desarrollo del frontend brinda una experiencia de usuario moderna, intuitiva y altamente adaptable. Next.js, con su estructura modular y capacidades de enrutamiento, facilita la creación de interfaces dinámicas y responsivas, esenciales para una aplicación interactiva como una red social. Bootstrap complementa esta experiencia al proporcionar un conjunto de componentes y estilos predefinidos que aseguran la coherencia visual y la compatibilidad entre dispositivos, permitiendo un desarrollo frontend eficiente y estéticamente atractivo. Juntos, estas tecnologías respaldan la visión de una aplicación robusta, segura y centrada en el usuario para la compartición de entrenamientos, brindando una experiencia integral y satisfactoria a los usuarios.

---

### <sup>01</sup> RF01 - Sistema de inicio de sesión:

Como usuario registrado, quiero poder iniciar sesión en la aplicación utilizando mi correo electrónico o cuenta de redes sociales, para acceder a todas las funcionalidades de la plataforma.

---

### <sup>02</sup> RF02 - Registro de usuario:

Como usuario nuevo, deseo poder registrarme en la aplicación utilizando mi correo electrónico o cuenta de redes sociales, para poder acceder a todas las funcionalidades de la plataforma.

---

### <sup>03</sup> RF03 - Gestión de perfil:

Como usuario, quiero tener un perfil personalizado donde pueda ver y editar mi información personal, incluyendo mi nombre, edad, género, y foto de perfil.

---

### <sup>04</sup> RF04 - Agregar amigos:

Como usuario, quiero poder ver una lista de mis amigos en la aplicación y tener la opción de agregar nuevos amigos utilizando su nombre de usuario.

---

### <sup>05</sup> RF05 - Visualización de entrenamientos de amigos:

Como usuario, deseo poder ver los últimos entrenamientos realizados por mis amigos en la página principal de la aplicación.

---



---

<sup>06</sup> **RF06 - Compartir entrenamientos:**

Como usuario, quiero poder compartir mis entrenamientos con mis amigos en la plataforma,.

---

<sup>07</sup> **RF07 - Seguir a otros usuarios:**

Como usuario, deseo poder seguir a otros usuarios en la aplicación para poder ver sus entrenamientos y mantenerme al tanto de su progreso fitness.

---

<sup>08</sup> **RF08 - Creación de entrenamientos personalizados:**

Como usuario, quiero poder crear entrenamientos personalizados y compartirlos con mis amigos, para que puedan unirse a mí en sesiones de entrenamiento colaborativas.

---

<sup>09</sup> **RF09 - Actualización de información de entrenamientos:**

Como usuario, deseo tener la capacidad de actualizar la información de mis entrenamientos.

---

---

<sup>10</sup> **RF10 - Administración de usuarios:**

Como administrador, quiero poder ver una lista de todos los usuarios registrados en la aplicación, para gestionar sus cuentas y resolver cualquier problema que puedan tener

---

<sup>11</sup> **RF11 - Moderación de contenidos:**

Como administrador, quiero tener la capacidad de revisar y moderar el contenido generado por los usuarios, como publicaciones, comentarios y entrenamientos compartidos, para garantizar que cumplan con las normas de la comunidad y los estándares de calidad.

---

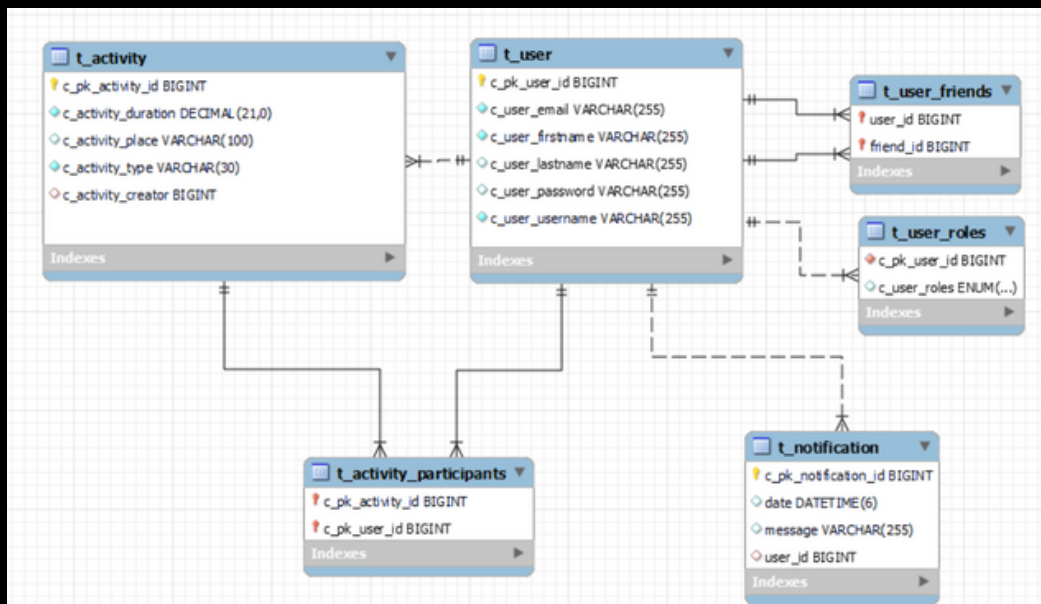
<sup>12</sup> **RF12 - Gestión de notificaciones y comunicaciones:**

Como administrador, quiero poder enviar notificaciones y comunicaciones a los usuarios de la aplicación, como anuncios importantes, actualizaciones de funciones o recordatorios de eventos, para mantenerlos informados y comprometidos con la plataforma.

---

## BASE DE DATOS

### DIAGRAMA EER



### Entidades:

- User: Esta entidad representa la entidad de usuarios, en ella se guardará información como: el identificador, nombre y apellidos, nombre de usuario, email, la contraseña de registro, el rol que ocupa como usuario o administrador, también se guarda las actividades que el usuario crea, tanto como en las que participa. Por último en esta entidad se guarda los amigos y las notificaciones.
- Activity: esta entidad representa las actividades que el usuario crea, en ella se guardará información como: el identificador, el tipo de actividad (caminar, correr, nadar, ciclismo...), duración de la actividad, lugar en el que se realiza, el usuario que la crea y los amigos del usuario que participan en ella.
- Notifications: esta entidad representa las notificaciones que recibirá el usuario cuando alguien le siga, le comente en alguna publicación, etc. En esta entidad se guardará información como: la identificación, el mensaje de la notificación, quien la recibe y la fecha cuando se crean.

### Relaciones:

- User-Activity: esta entidad se relaciona con actividad en una relación  $n:m$  en la que un usuario puede tener muchas actividades, de esta relación nace una tabla donde se guardan los datos de las actividades en las que el usuario ha participado.
- User-Rol: esta entidad se relaciona con rol en una relación  $1:n$ , en la que un usuario puede tener un rol, pero un rol puede ser usado por muchos usuarios.
- User-Friends: en esta relación la entidad User relaciona consigo misma en una relación  $1:n$ , en la que un usuario puede tener muchos amigos.
- User-Notifications: en esta relación la entidad User relaciona con la entidad notificaciones en una relación  $1:n$ , donde un usuario puede tener muchas notificaciones y mientras que una notificación solo puede estar asociada a un único usuario.

ENLACE AL REPOSITORIO

Link: [https://github.com/Dani-Ps/proyecto\\_final\\_iesalixar](https://github.com/Dani-Ps/proyecto_final_iesalixar)



# FIN