

FitTrack

Autor: Sergio Javier Lorente Guerra

Ciclo Formativo: CFGS Desarrollo de Aplicaciones Multiplataforma

Centro Educativo: IES Monte Naranco

Índice

1. Introducción
 - 1.1 Contexto
 - 1.2 Presentación de la aplicación
 - 1.3 Objetivos
2. Análisis y Diseño de la Aplicación
 - 2.1 Análisis de requisitos funcionales
 - 2.2 Diagramas de Casos de Uso
 - 2.3 Diseño de Interfaces
 - 2.4 Diseño de Datos
3. Implementación
 - 3.1 Tecnologías utilizadas
4. Aplicación en el Ámbito Empresarial
 - 4.1 Idea de negocio
 - 4.2 Planificación de RRHH
 - 4.3 Planificación de la producción
 - 4.4 Prevención de Riesgos Laborales
 - 4.5 Previsiones de inversión y financiación
5. Conclusión
 - 5.1 Valoración personal
 - 5.2 Posibles ampliaciones

6. Bibliografía y Webgraffía

1. Introducción

1.1 Contexto

En la actualidad, el mercado de aplicaciones de fitness ofrece múltiples soluciones generalistas para el seguimiento de la actividad física y la dieta, como MyFitnessPal, Nike Training Club o Freeletics. Estas plataformas combinan bibliotecas de ejercicios, seguimiento nutricional y rutinas adaptativas, pero suelen enfocarse en objetivos generales de salud o entrenamiento cardiovascular. Nuestra propuesta se centra en los entusiastas y profesionales del entrenamiento de fuerza, ofreciendo métricas más especializadas y un control detallado del progreso de levantamiento de peso.

1.2 Presentación de la aplicación

FitTrack es una aplicación multiplataforma diseñada para entrenadores y usuarios que realizan entrenamientos de fuerza. Permite a los entrenadores crear bloques de ejercicios preconfigurados —organizados en semanas, días y series— y asignarlos a sus clientes. Los usuarios registran datos de cada serie (peso, repeticiones, percepción de esfuerzo), generando estadísticas y estimaciones de repetición máxima (RM) a lo largo del tiempo.

1.3 Objetivos

- Desarrollar una herramienta que facilite la planificación y seguimiento de entrenamientos de fuerza.
- Proporcionar a los entrenadores un panel de control para asignar y supervisar bloques de ejercicio.
- Ofrecer a los usuarios una agenda digital que documente de forma estructurada su progreso.
- Generar gráficos y estadísticas dinámicas que ayuden en el análisis de rendimiento.

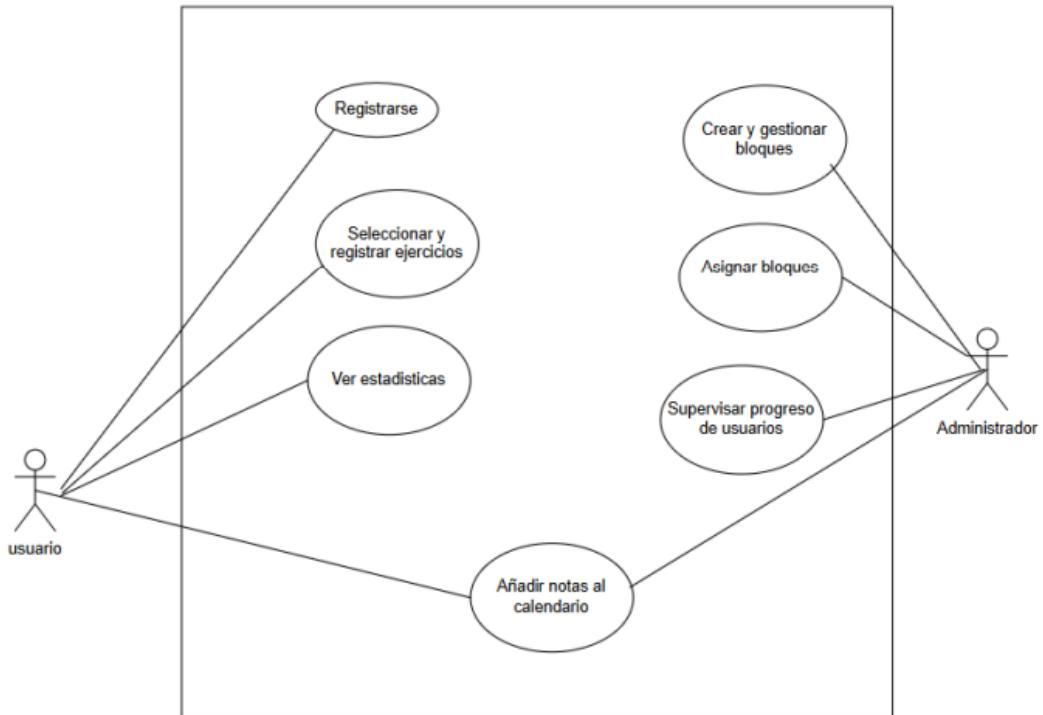
2. Análisis y Diseño de la Aplicación

2.1 Análisis de requisitos funcionales

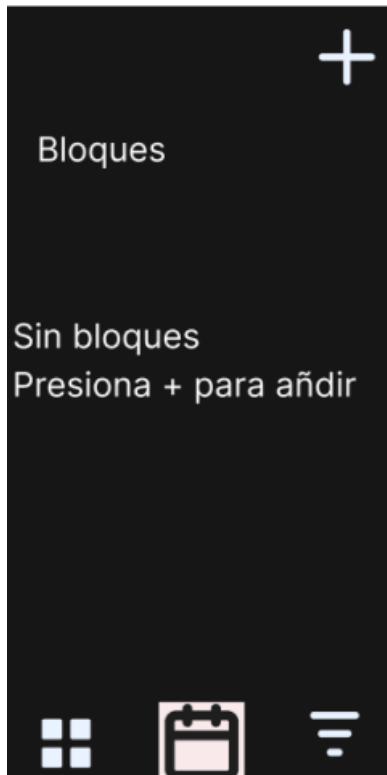
Nº	Nombre	Descripción	Prioridad
RF1	Inicio de sesión	Pantalla de autenticación con usuario y contraseña para acceder a la plataforma.	Alta

RF2	Bloque de entrenamientos	Estructura de semanas y días con ejercicios donde el usuario registra peso, repeticiones y series.	Alta
RF3	Gestión de bloques de entrenamiento	El entrenador puede crear, asignar y modificar bloques preconfigurados para sus clientes.	Alta
RF4	Calendario	Posibilidad de añadir notas personalizadas en fechas específicas (objetivos, recordatorios).	Baja
RF5	Estadísticas	Generación de gráficos dinámicos con Chart.js y ng2-charts para mostrar progreso en peso, repeticiones y estimación de RM.	Alta
RF6	Supervisión de usuarios	Panel de entrenador para supervisar registros y estadísticas de cada usuario.	Media

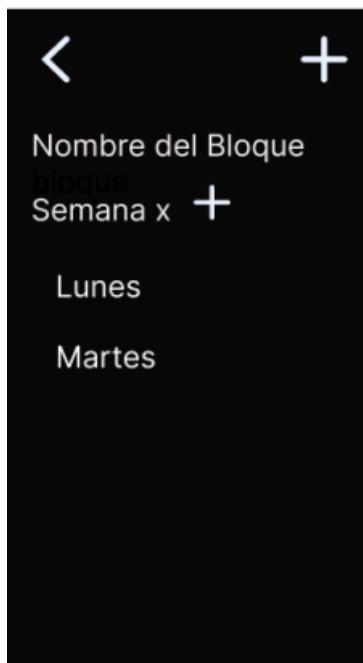
2.2 Diagramas de Casos de Uso



2.3 Diseño de Interfaces

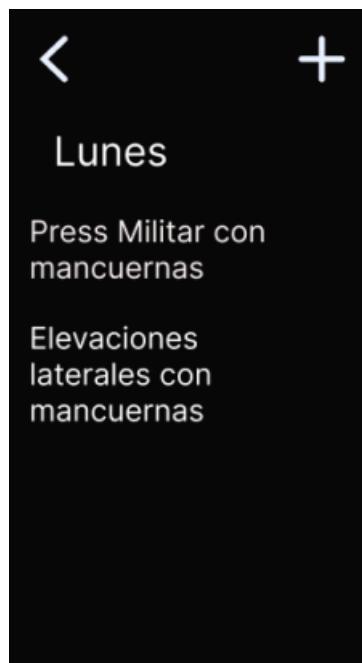


Sección de bloques/ vista administrador o entrenador En el bloque + se añaden los bloques. Cada bloque que cree el entrenador puede ser asignado a uno o varios usuarios.



Vista del usuario y administrador. Dentro de cada bloque se mostrará esta interfaz. Semana X indicará el número de la semana en la que te encuentras. El símbolo "+" en la parte superior derecha permitirá crear nuevos subbloques dentro de la semana, mientras que el símbolo "+" al lado derecho de la semana permitirá añadir una nueva semana dentro del bloque de entrenamiento. Esta estructura permite que los

entrenamientos se organicen de manera flexible y personalizada según la planificación establecida.



En cada entrenamiento podrás añadir los ejercicios que quieras hacer ese día.

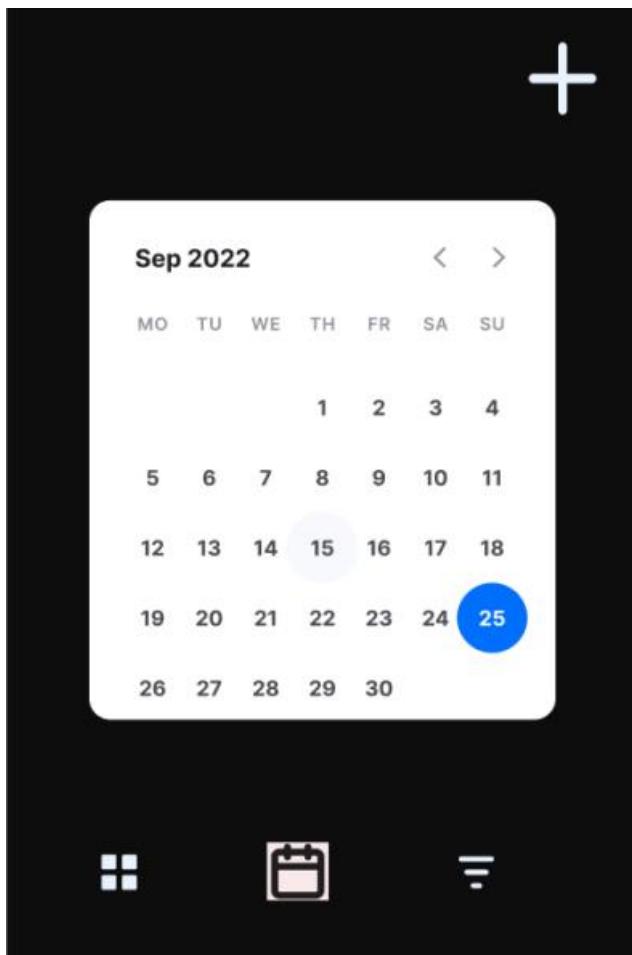


Aquí anotarías tus marcas personales por ejercicio.



Esta sección corresponde a la serie, y como solo hay una, se identifica como la serie número 1. En ella anotarías los datos relacionados con el peso utilizado, el número de

repeticiones y la percepción del esfuerzo realizado. En "Objetivo" registrarías lo que planeas hacer en esa serie, mientras que en "Real" anotarías lo que realmente lograste. El símbolo "+" que aparece debajo permite añadir otra serie. En "Estadísticas" se generará, a partir de los datos registrados, una posible estimación del RM (Repetición máxima), es el peso máximo que una persona puede levantar a una repetición.



Seleccionando el día puedes añadir una nota dándole al + que está en la esquina superior derecha.

Estadísticas

Usuarios

Juan

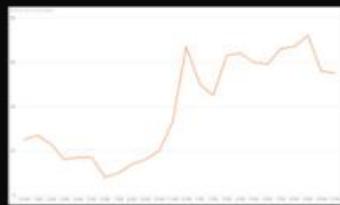
Manolo

José



Esta es la vista que tendría el entrenador o administrador en la parte de gráfica (ícono de abajo a la derecha) donde tendrá la una lista con todos los usuarios asociados a él. El entrenador podrá elegir uno de ellos y ver una gráfica y la opción de elegir entre los ejercicios que tiene dicho usuario.

Estadísticas

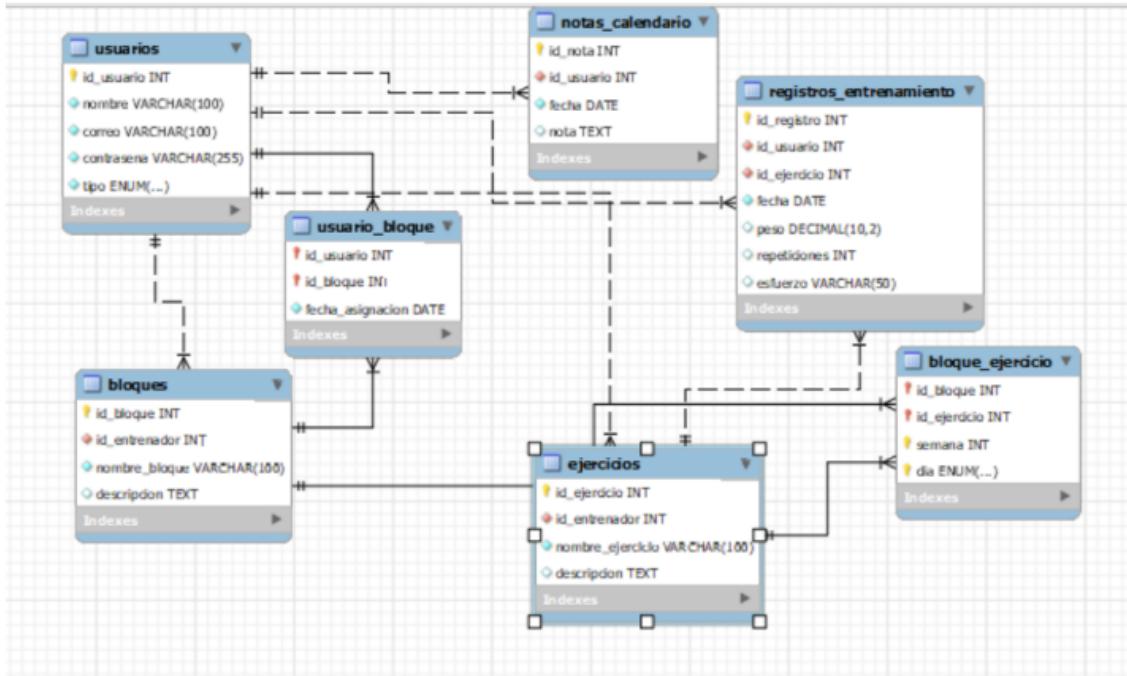


Escoge un ejercicio



Esta pantalla estará disponible tanto en la vista de usuario como de entrenador para que ambos sean conscientes de los avances.

2.4 Diseño de Datos



3. Implementación

3.1 Tecnologías utilizadas

El desarrollo de la aplicación se llevará a cabo utilizando IntelliJ IDEA como entorno de desarrollo para crear la API del backend, que será construida con Spring Boot utilizando Java como lenguaje de programación. Para el desarrollo del frontend, se utilizará Visual Studio Code, donde se implementará la interfaz de usuario utilizando Ionic con el framework Angular y TypeScript como lenguaje de programación. La base de datos utilizada será MySQL, gestionada a través de MySQL Workbench, lo que permitirá almacenar y administrar la información de los usuarios y entrenamientos de forma eficiente. Finalmente, el despliegue de la aplicación se realizará en un servidor proporcionado por el centro, que servirá como plataforma para alojar y gestionar la aplicación.

3.2 Arquitectura del backend.

5. Aplicación en el Ámbito Empresarial

5.1 Idea de negocio

FitTrack se comercializa como una plataforma SaaS dirigida a gimnasios y entrenadores personales que desean digitalizar y optimizar la gestión de sus clientes y rutinas. Por un lado, los gimnasios podrán centralizar reservas de sala, historiales de entrenamiento y pagos en un único panel, detectando automáticamente usuarios inactivos y lanzando

retos o promociones para mejorar la retención. Por otro, los entrenadores obtienen una herramienta para crear y asignar bloques de ejercicios (semanas, días y series), seguir el progreso de varias personas al mismo tiempo y ofrecer soporte remoto gracias a gráficos de evolución de peso y repeticiones. Esta doble vertiente (B2B y B2C) permite incrementar el ingreso medio por cliente, reducir un 15 % el churn y ahorrar hasta un 20 % en costes administrativos.

5.2 Planificación de RRHH

Para poner en marcha y mantener FitTrack durante el primer año se recomienda incorporar, de forma escalonada según avance el proyecto:

- **Product Manager:** responsable del roadmap, coordinación con gimnasios piloto y validación de hipótesis de negocio.
- **Dos desarrolladores backend:** programadores Java con Spring Boot, encargados de la API REST, la lógica de negocio y la integración con la base de datos MySQL.
- **Dos desarrolladores frontend:** expertos en Ionic + Angular y TypeScript, que implementen la interfaz en móvil y web.
- **Ingeniero de DevOps:** especializado en la administración del servidor del centro educativo, despliegue, backups y monitorización.
- **Diseñador UX/UI:** que defina flujos, prototipe pantallas y realice pruebas de usabilidad.
- **Ingeniero de QA:** para pruebas manuales y automatizadas (regresión), informe de cobertura y gestión de incidencias.
- **Dos perfiles de Customer Success:** formadores y soporte al cliente, recabando feedback para las siguientes iteraciones.

5.3 Planificación de la producción

El desarrollo se estructura en fases encadenadas, con un horizonte de entrega del MVP en torno a nueve meses:

1. **Diseño y prototipado (1,5 meses)**
 - Definición de requisitos funcionales y wireframes de alta fidelidad.
2. **Desarrollo Alpha (2,5 meses)**
 - Backend en Spring Boot, frontend en Ionic/Angular, modelo de datos en MySQL.
3. **Pruebas internas (1 mes)**

- Tests unitarios, de integración y corrección de errores críticos.

4. Desarrollo Beta (2,5 meses)

- Integración de estadística con Chart.js + ng2-charts, notificaciones y ajustes de rendimiento.

5. Piloto con tres gimnasios (1 mes)

- Recogida de feedback real, ajustes UX y mejoras de estabilidad.

6. Lanzamiento del MVP (0,5 meses)

- Despliegue en el servidor del centro, publicación de la documentación de usuario y campaña inicial de marketing.

7. Mantenimiento continuo

- Ciclos quincenales de actualización para incorporar nuevas funcionalidades y corregir incidencias.

5.4 Prevención de Riesgos Laborales

Se han tomado medidas tanto para la salud del usuario como del equipo:

- Ergonomía de la interfaz: uso de contrastes suaves y tipografías legibles, disposición de controles accesibles con una sola mano en dispositivos móviles.
- Descansos activos: la app puede sugerir pequeñas pausas y estiramientos tras sesiones de uso prolongado.
- Condiciones de puesto: el equipo de desarrollo dispondrá de sillas ergonómicas, reposapiés y un protocolo de pausas de 5 min cada 55 min de trabajo frente a la pantalla.

5.5 Previsiones de inversión y financiación

Para el primer año, se ha estimado una inversión total de aproximadamente 163.500 €.

En lo que respecta a la infraestructura, aprovechamos el servidor propio del centro para evitar costes de nube pública. Sólo contemplamos alrededor de 1.000 € anuales en electricidad y mantenimiento, y 10 € para la renovación del dominio, al usar un certificado SSL gratuito con Let's Encrypt. El resto de las herramientas de desarrollo (editores, repositorios, testing) se basa en versiones gratuitas, por lo que no genera gasto adicional.

Para el apartado de personal hemos optado por un equipo ultra-lean de cinco perfiles. Contemplamos un desarrollador full-stack con Spring Boot e Ionic/Angular a 40.000 € brutos, un DevOps/sysadmin a 35.000 €, un perfil combinado de UX/UI y QA a media jornada por 15.000 €, un product manager y customer success también a media jornada por 15.000 €, y finalmente un soporte/community manager freelance con un presupuesto anual de 10.000 €. Aplicando un 30 % de cargas sociales sobre los salarios brutos, el coste total de personal asciende a 149.500 €.

Por último, en operaciones y marketing destinamos 8.000 € a campañas digitales muy focalizadas (Google Ads y redes sociales) y 5.000 € a material promocional y colaboraciones con gimnasios piloto, lo que suma 13.000 €.

Con este planteamiento minimalista y eficiente, la inversión necesaria para validar el proyecto en su primer año se sitúa en un rango de 160.000 € a 170.000 €, lo que permite probar el modelo de negocio antes de plantear una ampliación del equipo o de las partidas de gasto.

6. Conclusión

6.1 Valoración personal

Cuando empecé a plantear FitTrack en el papel pensaba que tenía todo superclaro, pero montar ese puzzle en código resultó un viaje de hostias y miedo. No era aprender frameworks nuevos (Ionic, Angular y Spring Boot ya me sabían a gloria) sino darle forma a la idea y ver en cada compilación que algo fallaba. Cada “paso adelante” traía su dosis de problemas, caos y ganas de lanzarme por la ventana: un interceptor de Angular que devoraba mis peticiones, un build de Android que funcionaba en el emulador y se suicidaba en el móvil real, un endpoint en Spring que lanzaba un 403 sin avisar o los servidores del Naranco que se ríen de mí.

Me pegué tardes enteras escarbando en logs, poniendo console.log por todos lados y depurando a fuego lento con breakpoints. Hubo momentos en los que cambiaba un par de líneas y creía que estaba salvado, hasta que un nuevo error me recordaba que el proyecto aún estaba en la adolescencia. A pesar del mosquito, cada vez que la app arrancaba sin glitches y la barra de progreso avanzaba suave, la satisfacción era brutal.

El caos de Android me sacó canas antes de tiempo, pero también me obligó a afinar la paciencia y la cabeza de ingeniero. Aprendí a medir cada integración, a diseñar flujos pensando en el peor escenario y a valorar un buen plan de pruebas. Ahora, al mirar FitTrack funcionando, me doy cuenta de que todos esos dolores de cabeza fueron mi mejor formación: no es solo saber programar, sino encajar cada pieza cuando todo está en llamas. Y sí, sacar esto adelante con café de madrugada y algún que otro taco en el

teclado me hace sentir que, de verdad, he exprimido al máximo mi vena de programador.

6.2 Posibles ampliaciones

Para mantener *FitTrack* competitivo y ampliar su mercado, estas son algunas funcionalidades creíbles y alineadas con el stack tecnológico actual:

1. Integración con wearables y IoT

Conectar la app al ecosistema de pulseras y relojes (Fitbit, Garmin, Apple Health, Google Fit) permite importar automáticamente datos de ritmo cardiaco, calorías quemadas y calidad de sueño. Así, entrenadores y gimnasios dispondrán de información en tiempo real, sin necesidad de que el usuario introduzca manualmente sus métricas.

2. Módulo de nutrición y meal-planning

Añadir un apartado donde el entrenador pueda diseñar dietas semanales, generar listas de compra y registrar ingestas diarias. Gráficos de macronutrientes y cálculo automático de calorías potenciarían la propuesta de valor, ofreciendo un servicio 360º de salud y deporte.

3. Marketplace interno de entrenadores y clases

Implementar un portal B2C donde los usuarios puedan buscar y contratar entrenadores independientes o reservar clases colectivas (yoga, HIIT, pilates) directamente desde la aplicación. Con valoraciones, perfiles detallados y pago integrado, se crea un nuevo canal de ingresos.

4. Videollamadas y streaming de sesiones

Sobre la plataforma Ionic/Angular se puede incorporar un servicio de videoconferencia (por ejemplo, usando WebRTC) para clases en directo o entrenamientos one-to-one online. Esto abriría la puerta a clientes remotos y a modelos híbridos presencial-online.

5. Analítica avanzada y cuadros de mando

Más allá de los gráficos básicos con Chart.js, integrar un motor de BI (Power BI Embedded o ElasticSearch + Kibana) para generar informes complejos (cohortes de usuarios, predicción de abandono, rentabilidad por cliente). Sería muy valioso para cadenas de gimnasios y franquicias.

6. Gamificación y comunidad

Desarrollar retos mensuales con recompensas (insignias digitales, descuentos o incluso merchandizing), tableros de clasificación y grupos de apoyo. Mantener enganchados a los usuarios incrementa la retención y genera mayor “boca-oreja”.

7. API pública y ecosistema de partners

Exponer una REST API documentada que permita a terceros (startups de salud, clínicas de fisioterapia, apps de meditación) leer y escribir datos en FitTrack. Con un marketplace de integraciones, la app se convertiría en un hub centralizado de bienestar.

8. Soporte offline y sincronización

Permitir a los usuarios consultar rutinas y registrar ejercicios incluso sin conexión, guardando la información localmente (IndexedDB) y sincronizándola con el backend Spring Boot cuando recuperen cobertura. Muy útil para entrenamientos en exteriores o zonas rurales.

9. Programas corporativos de wellness

Adaptar la plataforma para empresas que quieran ofrecer planes de ejercicio y salud a sus empleados como beneficio social. Módulos de seguimiento de objetivos grupales, retos interdepartamentales y reportes de ROI para RRHH.

10. White-label y multi-marcia

Facilitar la personalización completa de la app (logos, colores, nombre) para gimnasios o cadenas que deseen tener su propia versión bajo sus marcas. Con la arquitectura de microservicios en Spring Boot bastaría un simple parámetro de configuración por cliente.

7. Bibliografía y Webgrafía

Ionic Framework Docs

<https://ionicframework.com/docs>

Angular Official Documentation

<https://angular.io/docs>

Spring Boot Reference Guide

<https://docs.spring.io/spring-boot/docs/current/reference/html/>

ChatGPT (OpenAI)