Imagen que contiene texto, dibujo, señal

Descripción generada automáticamente

FitBuddy es una aplicación móvil multiplataforma que conecta a usuarios y entrenadores para mejorar la experiencia del entrenamiento. La app permite registrarse, configurar un perfil, buscar compañeros de entrenamiento mediante filtros de ubicación y objetivos, crear y compartir rutinas o rutas, comunicarse a través de un chat interno y llevar un historial de entrenamientos. A diferencia de otras aplicaciones deportivas, FitBuddy se centra en generar vínculos reales y motivación compartida, combinando tecnología con el aspecto social del entrenamiento.

FITBUDDY

Práctica Profesional III

2025

Alumnos: Fernandez, Mauricio  
Rojas, Maximiliano

Ledesma, Emmanuel

Sanchez, Agustina

Contenido

[1. Estado del arte 2](#_Toc163835330)

[2. Descripción del proyecto 2](#_Toc163835331)

[3. Objetivo 2](#_Toc163835332)

[4. Alcance 2](#_Toc163835333)

[5. Casos de uso 2](#_Toc163835334)

[6. Especificación de requerimientos 2](#_Toc163835335)

[7. Recursos disponibles 2](#_Toc163835336)

[a. Personal 2](#_Toc163835337)

[b. Material 2](#_Toc163835338)

[c. Tiempo 2](#_Toc163835339)

[8. Plan de trabajo 2](#_Toc163835340)

[9. Gestión de riesgos 2](#_Toc163835341)

[10. Diseño del sistema 2](#_Toc163835342)

[a. Arquitectura 3](#_Toc163835343)

[b. Diagrama de componentes 3](#_Toc163835344)

[c. DER 3](#_Toc163835345)

[d. Diagrama de clases 3](#_Toc163835346)

[e. Diagrama de secuencia 3](#_Toc163835347)

[f. Diagrama de actividad 3](#_Toc163835348)

[g. Interfaces de usuarios 3](#_Toc163835349)

[h. Interfaces con otros sistemas 3](#_Toc163835350)

[11. Desarrollo 3](#_Toc163835351)

[12. Implementación y prueba 3](#_Toc163835352)

[13. Conclusiones finales 3](#_Toc163835353)

# Estado del arte

El desarrollo de FitBuddy se basa en tecnologías modernas para construir una aplicación móvil eficiente y escalable:

* React Native: Framework de código abierto creado por Meta que permite el desarrollo multiplataforma (iOS y Android) desde una sola base de código. Desde 2015, ha evolucionado para soportar funcionalidades nativas avanzadas con excelente rendimiento.
* Node.js: Entorno de ejecución para JavaScript que permite crear servidores eficientes y APIs escalables. Es ampliamente usado en aplicaciones en tiempo real.
* SQLite3 + Backend-as-a-Service (BaaS): Para gestión de datos local y en la nube. Ofrece flexibilidad en el desarrollo inicial con proyección a escalabilidad futura.
* Figma: Herramienta de diseño colaborativo ampliamente utilizada para prototipado rápido y pruebas de interfaz.

Sistemas similares existentes:

* Strava: App social para atletas con foco en seguimiento de rutas y estadísticas.
* Nike Run Club: Aplicación orientada a corredores, ofrece rutinas y seguimiento de objetivos.

FitBuddy se diferencia por su función de emparejamiento inteligente entre usuarios con objetivos y niveles similares, y su enfoque en la conexión social y colaborativa del entrenamiento.

# Descripción del proyecto

FitBuddy es una aplicación móvil que busca mejorar la experiencia de entrenamiento deportivo al conectar a personas con objetivos similares, facilitando la planificación de rutinas y rutas de manera colaborativa. La app permitirá a los usuarios emparejarse con entrenadores o compañeros de entrenamiento, compartir experiencias, rutinas, rutas, y fomentar hábitos saludables mediante la interacción social.  
  
La problemática que aborda es la dificultad para mantener constancia en la actividad física por falta de compañía, motivación o conocimiento técnico para planificar entrenamientos efectivos.

# Objetivo

El objetivo de la aplicación es conectar a personas con intereses y objetivos deportivos similares para fomentar hábitos de entrenamiento más saludables y sostenidos. A diferencia de otras plataformas, esta aplicación se enfoca en crear un espacio donde los usuarios puedan conocerse entre sí, establecer vínculos reales y compartir experiencias, progresos y rutinas de forma colaborativa. La conexión entre usuarios se realiza considerando tanto sus preferencias deportivas como su cercanía geográfica, facilitando así encuentros presenciales y entrenamientos compartidos. Además, la aplicación cuenta con entrenadores disponibles que brindan orientación técnica y acompañamiento personalizado, fortaleciendo una comunidad activa y comprometida con el bienestar físico.

# Alcance

Desde:  
- Desarrollo de la aplicación móvil con funcionalidades de registro, emparejamiento, rutas compartidas, rutinas personalizadas y chat integrado.  
  
Hasta:  
- Prototipo funcional disponible para pruebas en dispositivos Android e iOS.  
- Validación básica de funcionalidades principales con usuarios reales.  
  
Funcionalidades fuera de alcance (para versiones futuras):  
- Integración con dispositivos externos o wearables.  
- Implementación de gamificación.  
- Sistema de notificaciones avanzadas o seguimiento por voz.

# Casos de uso

Detalle de los casos de usos (diagrama de casos de usos de UML)

**Caso de Uso para el registro e inicio de sesión**

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama, Esquemático

El contenido generado por IA puede ser incorrecto.

# Especificación de requerimientos

Lista de requerimientos funcionales y no funcionales del proyecto.

**Requerimientos Funcionales**

**RF1**. Permitir registro, inicio de sesión y edición de perfil:  
 **Actor:** Usuario / Entrenador  
Ambos pueden crear una cuenta, acceder al sistema y modificar su información personal (foto, descripción, objetivos, etc.).

**RF2**. Mostrar usuarios disponibles para emparejamiento según filtros:  
**Actor:** Usuario  
El usuario puede buscar compañeros de entrenamiento o entrenadores utilizando filtros como ubicación, disciplina, horarios, nivel, entre otros.

**RF3**. Permitir creación, edición y consulta de rutinas de entrenamiento  
**Actor:** Usuario / Entrenador  
El usuario puede generar rutinas propias o seguir rutinas sugeridas. El entrenador puede diseñar y ofrecer rutinas a otros usuarios.

**RF4**. Facilitar la planificación y visualización de rutas  
**Actor:** Usuario  
Puede seleccionar rutas sugeridas o crear y compartir rutas personalizadas adaptadas a su entrenamiento.

**RF5**. Incluir sistema de mensajería interna entre usuarios  
**Actor:** Usuario  
Permite el envío y recepción de mensajes entre usuarios y entrenadores para coordinar encuentros, compartir rutinas o resolver dudas.

**RF6**. Registrar historial de entrenamientos y rutinas  
**Actor:** Usuario  
El usuario accede a un historial detallado de sus rutinas completadas y entrenamientos realizados, permitiéndole evaluar su progreso.

**Requerimientos No Funcionales**

**RNF1**. Interfaz clara, accesible y responsive  
El diseño debe ser simple, comprensible y adaptado a distintos dispositivos. La organización visual debe facilitar la navegación, con textos y botones claros y coherentes.

**RNF2**. Alta disponibilidad del sistema y sincronización con la nube  
El sistema debe funcionar de manera estable, con acceso a los datos desde diferentes dispositivos gracias a la sincronización en la nube.

**RNF3**. Seguridad en el manejo de datos personales y conversaciones  
Se deben proteger los datos personales y los mensajes mediante cifrado y buenas prácticas de desarrollo seguro.

**Requerimientos de Interfaz**

**UI1**. Diseño intuitivo, con colores cálidos y amigables  
La aplicación debe transmitir cercanía y comodidad visual mediante una paleta de colores cálidos y un diseño visual armonioso.

**UI2**. Navegación clara por pestañas o menú inferior  
El acceso a las secciones principales debe ser rápido e intuitivo, utilizando pestañas o un menú inferior que permita una navegación fluida.

**UI3**. Pantallas principales del sistema  
Las vistas esenciales de la app deben incluir:

* Inicio
* Emparejamientos
* Rutinas
* Rutas
* Chat
* Perfil

# Recursos disponibles

Detalle de los recursos necesarios para la ejecución del proyecto

## Personal

El equipo de desarrollo está conformado por 4 integrantes, con funciones distribuidas para cubrir todas las etapas del proyecto:

* **Análisis de requerimientos**: recopilación de necesidades del usuario y definición de funcionalidades.
* **Diseño de interfaz y experiencia de usuario (UI/UX)**: desarrollo de maquetas, flujo de pantallas y componentes visuales.
* **Desarrollo backend y base de datos**: implementación de la lógica del servidor, endpoints, y persistencia de datos.
* **Desarrollo frontend**: construcción de la aplicación móvil, integración de interfaces y comunicación con la API.
* **Documentación y presentación del proyecto**: elaboración de entregables formales, informes y material para exposición.

## Material

Material necesario para el diseño, desarrollo, implementación y prueba del sistema.

**Técnicos:**

* **Computadoras personales** con capacidad para desarrollo de software.
* **Herramientas de desarrollo**: Visual Studio Code, Android Studio.
* **Software de diseño y prototipado**: Figma (para UI/UX).
* **Lenguajes y frameworks**: React Native (frontend móvil), Node.js (backend), SQLite3 (base de datos), HTML, CSS y JavaScript (para interfaces web complementarias si se requieren).
* **Sistema de control de versiones**: GitHub, para colaboración, control de cambios y respaldo del código.

**Físicos:**

* **Dispositivos móviles (smartphones)** para realizar pruebas de funcionamiento, compatibilidad y experiencia del usuario.
* **Servicios en la nube**: uso de plataformas como Firebase (autenticación, base de datos en tiempo real) y Vercel o alternativas para el despliegue del backend y hosting.

## Tiempo

Tiempos disponibles para su realización. Termino e hitos a cumplimentar.

El tiempo del proyecto será de una cursada completa, la cual se divide en dos partes principales. Dentro del diagrama de Gantt, se especifica qué tareas se llevarán a cabo a lo largo del tiempo del desarrollo proyecto

# Plan de trabajo

Desarrollo de un diagrama de Gantt con el detalle de las actividades a realizar, los hitos a cumplimentar y el detalle del camino crítico.

Tabla

El contenido generado por IA puede ser incorrecto.

Gráfico, Escala de tiempo, Gráfico en cascada

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

Imagen que contiene biombo, juego, edificio, cuarto

El contenido generado por IA puede ser incorrecto.

# Diseño del sistema

Detalle de diseño de los distintos componentes de sistemas

## Arquitectura

* Capa de Presentación (UI/UX)

Responsable de la interacción directa con el usuario.

Implementada como aplicación móvil multiplataforma utilizando React Native.

Permite registro, inicio de sesión, configuración de perfil, visualización de rutinas, rutas, historial y chat.

Desarrollada por el equipo frontend y de diseño UI/UX, usando herramientas como Figma para prototipado.

* Capa de Negocio (Backend)

Contiene la lógica central de la aplicación y las reglas de negocio.

Implementada con Node.js y Express, gestionando la comunicación entre la app móvil y la base de datos mediante API REST (HTTPS).

Funcionalidades principales:

* Autenticación (registro e inicio de sesión).
* Gestión de Usuarios (perfiles de usuario y entrenador).
* Gestión de Rutinas (crear, editar, compartir).
* Gestión de Rutas (planificar y compartir).
* Mensajería / Chat (comunicación interna).
* Historial de Entrenamientos (registro y consulta de progreso).

Desarrollada por el equipo backend.

* Capa de Datos

Responsable de almacenar y administrar la información persistente del sistema.

Utiliza una combinación de:

* **SQLite3**: para el manejo local y ligero de datos en el dispositivo.
* **Firebase (BaaS)**: para sincronización en la nube, autenticación segura y disponibilidad multiplataforma.

Entidades principales:

* **Usuarios / Entrenadores**
* **Rutinas**
* **Rutas**
* **Mensajes**
* **Historial de Entrenamientos**

Gestionada por el equipo de **bases de datos** y **administración de sistemas**.

## Diagrama de componentes

Diagrama

El contenido generado por IA puede ser incorrecto.

Este diagrama representa la arquitectura lógica del sistema FitBuddy y cómo se relacionan sus componentes principales:

* Aplicación Móvil (React Native): La interfaz que usan usuarios y entrenadores.
* Servidor Backend (Node.js / Express): Gestiona la lógica, reglas de negocio y comunicación con la base de datos.
* Base de Datos (SQLite / Firebase): Almacena y sincroniza datos como usuarios, rutinas, rutas, mensajes e historial.

La comunicación entre la app y el backend se realiza mediante API REST sobre HTTPS, asegurando integridad y seguridad en los datos.

## DER

Diagrama, Esquemático

El contenido generado por IA puede ser incorrecto.

## Diagrama de clases

Diagrama

El contenido generado por IA puede ser incorrecto.

## Diagrama de actividad Diagrama El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

## Interfaces de usuarios

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Una pantalla de un teléfono celular

El contenido generado por IA puede ser incorrecto.

# Desarrollo

Detalle de la organización de las tareas den la ejecución de desarrollo del sistema.

# Implementación y prueba

Descripción de las pruebas a realizar al sistema.

# Conclusiones finales

Resumen de la experiencia de lo actuado en la realización del proyecto, resaltando tanto aspectos positivos como negativos.