

## **Documentación TFG**

### **Motivación**

Pues la motivación de para realizar este proyecto viene desde tiempo atrás ya que soy un apasionado del mundo fitness y me encanta aprender de todo sobre el porque no hacer que el fitness sea más cómodo y que llame más la atención a la gente menos apasionada del fitness pues BuidYourBD se encarga de hacer precisamente eso .Tras un largo tiempo usando apps de fitness vi que no existía ninguna que tuviera un todo en 1 con lo cual motivación extra para hacerlo y conseguir que la comunidad fitness crezca.

### **Objetivos**

Los objetivos del proyecto son bastante sencillos: Conseguir una comunidad sana, estable y con ganas de mejorar tanto física como mentalmente.

Que la gente pueda subir sus entrenamientos o receta por ejemplo entrenamientos hit, recetas con pocas calorías, etc.

Que sobre todo para la gente nueva en este mundo que la app tenga acceso a tu ubicación y puedas meterte en un chat dependiendo del gimnasio que elijas cercano a tu ubicación para que puedas hacer amigos en esos gimnasios tanto desde fuera como desde dentro de ellos.

Que puedas trackear tus alimentos para poder llevar un conteo de tus calorías, grasas, hidratos y proteínas diarios. Serian los objetivos de este proyecto.

### **Metodología**

Se ha decidido utilizar la metodología SCRUM en la que dividiremos el trabajo en “sprints”. En cada sprint se hace una planificación detallada de lo que se hace en cada uno. Que tendrán una duración aproximada de entre 1 semana y 2 semanas.

## **Backlog**

En este caso nuestro Backlog esta dividido en varias partes:

### **1. Funcionalidades Básicas**

- **Registro y Gestión de Perfil de Usuario**
  - Crear formulario de registro (nombre, correo, contraseña).
  - Página de inicio de sesión para usuarios existentes.
  - Recuperación de contraseña.
  - Edición de perfil de usuario (foto, datos personales).
  - Visualización del historial de entrenamientos.
  
- **Modo Oscuro**
  - Implementar un modo oscuro para la aplicación.
  - Permitir al usuario alternar entre modo claro y oscuro según preferencia.
  
- **Seguimiento de Progreso**
  - Registro de peso y calorías quemadas.
  - Visualización de estadísticas de progreso en gráficos (peso, calorías, etc.).
  - Establecer metas de progreso personalizadas.
  
- **Recordatorios**
  - Implementar recordatorios para beber agua.
  - Recordatorios para comer.
  - Recordatorios para entrenar.

### **2. Rutinas y Recetas**

- **Rutinas Personalizadas**
  - Permitir que el usuario cree sus propias rutinas de entrenamiento.
  - Crear plantillas de rutinas prediseñadas.
  - Recomendación de rutinas según objetivos del usuario.

- **Recetario de Cocina**

- Crear base de datos de recetas con información nutricional.
- Permitir que el usuario filtre recetas por tipo.
- Función que permite que el usuario guarde sus recetas favoritas.

### **3. Funcionalidades Avanzadas**

- **Obtención de Gimnasios y Comunidad**

- Integración de APIs para obtener gimnasios cercanos a la ubicación del usuario.
- Crear una sección para mostrar la lista de gimnasios y sus detalles (ubicación, horarios, etc.)
- Implementación de una funcionalidad de chats en los gimnasios para interactuar con otros usuarios.

- **Comunidad Fitness y Contenido Compartido**

- Crear un recetario compartido donde los usuarios puedan subir y compartir recetas.
- Permitir que los usuarios compartan sus rutinas de entrenamiento en una sección comunitaria.
- Implementar un sistema de comentarios y valoraciones para recetas y rutinas de entrenamiento.

### **4. Optimización y Mejoras**

- **Interfaz y Experiencia de Usuario (UX/UI)**

- Diseño responsivo.
- Mejora de la navegación y accesibilidad.

### **5. Seguridad**

- **Protección de Datos**

- Encriptación de contraseñas y datos sensibles.
- Implementación de autenticación de dos factores(2FA).

## 6. Corrección de Errores

- **Mantenimiento y Soporte.**
  - Solución de errores en la carga de datos de entrenamiento.
  - Actualización de la base de datos para incluir nuevas recetas y rutinas.

### Tecnologías y herramientas

**Android Studio:** Es el framework para poder hacer aplicaciones en Android.

**MySQL:** Se usará para hacer la base de datos de la aplicación.

**Kotlin:** Será el lenguaje usado para poder hacer la aplicación en Android.

**Google Maps API:** Permitirá integrar mapas interactivos en mi aplicación.

**FireBase Authentication:** Es una herramienta muy útil para gestionar la autenticación de usuarios de manera rápida y segura.

**Retrofit:** Es una biblioteca de Android que facilita la conexión con APIs RESTful para enviar y recibir datos.

**Firestore:** Es una base de datos en tiempo real de Firebase.

**WorkManager:** es una herramienta para ejecutar tareas en segundo plano, incluso si la app se cierra o el dispositivo se reinicia.

**Google Fit API:** La Google Fit API proporciona acceso a datos relacionados con la actividad física del usuario. En "BuildYourBD".

Hemos decidido usar estas herramientas por estas decisiones. Se decidió hacer un programa en kotlin para poder hacer un app para lo cual necesitamos Android studio, MySQL y firestore serán las bases de datos que se intentaran implementar porque MySQL es una base de datos normal y firestore es una base de datos en tiempo real, Google maps API es la mejor api para poder realizar la cuestión de obtener los gimnasios cerca de tu ubicación y además en parte es gratis , firebase Authentication es una de las herramientas que hemos visto más fiables para la autenticación de usuarios de manera rápida y segura, retrofit es una de las herramientas que vamos a usar para enviar y recibir datos porque al parecer está bastante optimizada, es muy completa y es gratis, WorkManager hemos visto que es una herramienta que no da mucho conflicto para implementarla en proyectos de Android studio , Y Google fit API porque era de las pocas API fitness que nos permitían usar la API de forma gratuita.