**Portada**

**Trabajo de Fin de Ciclo (DAW)** **Alejandro Santos Cabrera** **Plataforma Web Fitness**

**ÍNDICE**

[**1. Introducción (Idea del proyecto) 3**](#_wwqc5apk5n0l)

[**2. Debilidades del proyecto 3**](#_2bmxxqrekr17)

[**3. Fortalezas del proyecto 4**](#_z67t0bij99bq)

[**4. Objetivos 4**](#_prisuyh8omsa)

[**5. Fases y subfases del proyecto 4**](#_fmqb8zoea5or)

[**6. Temporalización 5**](#_xjvymgprnbfv)

[**7. Medios a emplear 6**](#_r8ngyywzcjbq)

[**8. Parte desarrollada (Entrega 3) 6**](#_1h507876bovz)

[**9. Anexos 7**](#_ov6fh7iih1tm)

## 

## 

## 

## 1. Introducción (Idea del proyecto)

Este Trabajo de Fin de Ciclo del Grado Superior en Desarrollo de Aplicaciones Web (DAW) persigue el diseño y desarrollo de una plataforma web completa para usuarios que buscan gestionar sus entrenamientos y nutrición. La aplicación incluirá:

* Listado de ejercicios organizados por grupo muscular y tipo de equipo (barbell, dumbbell, bodyweight, etc.).
* Imágenes y vídeos demostrativos para mejorar la técnica.
* Creación de rutinas personalizadas con orden, repeticiones, series y descanso.
* Posibilidad de iniciar una rutina como sesión real, registrando peso real, repes efectivas y comentarios.
* Registro de progreso físico con fotos, peso y comentarios.
* Visualización de evolución en gráficos.
* Consulta nutricional de alimentos vía API FatSecret.
* Recálculo de macros según porción personalizada (ej. 150 g).
* Creación de dietas divididas en comidas (desayuno, comida, cena...) con alimentos importados desde FatSecret.
* Estructura modular escalable y diseño responsive.

## 2. Debilidades del proyecto

* Gestión de multimedia (imágenes y vídeos) que puede consumir muchos recursos en alojamiento gratuito.
* Riesgo de falta de planificación detallada, lo que podría retrasar entregas.
* Experiencia limitada en despliegue completo de proyectos individuales.
* Dependencia del ritmo de trabajo personal.

## 3. Fortalezas del proyecto

* Alto valor práctico para usuarios interesados en fitness y gimnasio.
* Tecnologías modernas, con gran documentación y comunidad (React, FastAPI, PostgreSQL).
* Arquitectura escalable: fácil incorporación futura de comunidades, IA o wearables.
* Aplicación de conocimientos del ciclo DAW: frontend, backend, bases de datos y despliegue.

## 4. Objetivos

Crear una web responsive que funcione en escritorio y móvil.

Diseñar y conectar una base de datos relacional bien estructurada.

Desarrollar el frontend en React.js con Tailwind CSS.

Implementar una API RESTful en FastAPI con Uvicorn.

Gestionar versiones con Git y GitHub.

Desplegar el frontend en Vercel y el backend en Railway o contenedor Docker.

Entregar la documentación y prototipos en Figma para la defensa del TFC.

## 5. Fases y subfases del proyecto

**Fase 1 – Planificación y Diseño (marzo – abril):**

* Análisis de requisitos, definición de funcionalidades
* Wireframes en Figma
* Estructura inicial del proyecto y repositorio

**Fase 2 – Preparación de entornos:**

* Instalación de entornos virtuales (Python, Node.js)
* Configuración de Docker + PostgreSQL
* Repositorio Git y GitHub

**Fase 3 – Desarrollo Backend:**

* FastAPI + SQLModel
* Modelado de base de datos
* Endpoints CRUD
* Relación Rutina–Sesión y copia dinámica de ejercicios

**Fase 4 – Desarrollo Frontend:**

* React + Tailwind CSS
* Componentes y navegación
* Conexión a API
* Validaciones y estado

**Fase 5 – Integración y despliegue:**

* Pruebas con Postman y pytest
* Despliegue en Railway y Vercel
* Documentación final y defensa

## 6. Temporalización

| **Fase** | **Duración** | **Periodo** |
| --- | --- | --- |
| Planificación y Diseño | 3 semanas | Marzo – 1ª quincena Abril |
| Preparación de entornos | 2 semanas | 1ª – 2ª quincena Abril |
| Desarrollo Backend | 4 semanas | Finales Abril – Mayo |
| Desarrollo Frontend | 3 semanas | Mayo |
| Integración, Pruebas y Despliegue | 4 semanas | Finales Mayo – 2ª quincena Junio |

## 7. Medios a emplear

**Hardware**

* Ordenador con al menos 16 GB RAM y SSD.
* Conexión a Internet estable.

**Software**

**Hardware:** PC personal

**Software:**

* Visual Studio Code
* Docker Desktop
* PostgreSQL
* FastAPI, React, Tailwind
* Figma (diseño)
* Git + GitHub

**Servicios externos:**

* Railway, Vercel
* FatSecret API

## 8. Parte desarrollada

**Fase desarrollada: Fase 3 – Desarrollo Backend**

Durante esta fase se ha llevado a cabo:

* Configuración del entorno virtual Python 3.11
* Instalación de FastAPI, Uvicorn, SQLModel, psycopg2
* Docker Compose para PostgreSQL
* Generación de modelos (Usuario, Ejercicio, Rutina, Sesión, Progreso, Dieta...)
* Relaciones dinámicas: iniciar rutina como sesión
* Estructura de proyecto y conexión a base de datos

### 

### 

## 9. Anexos

Figura 1. Estructura del backend

Figura 1. Estructura del backend


Figura 2. Servidor API funcionando

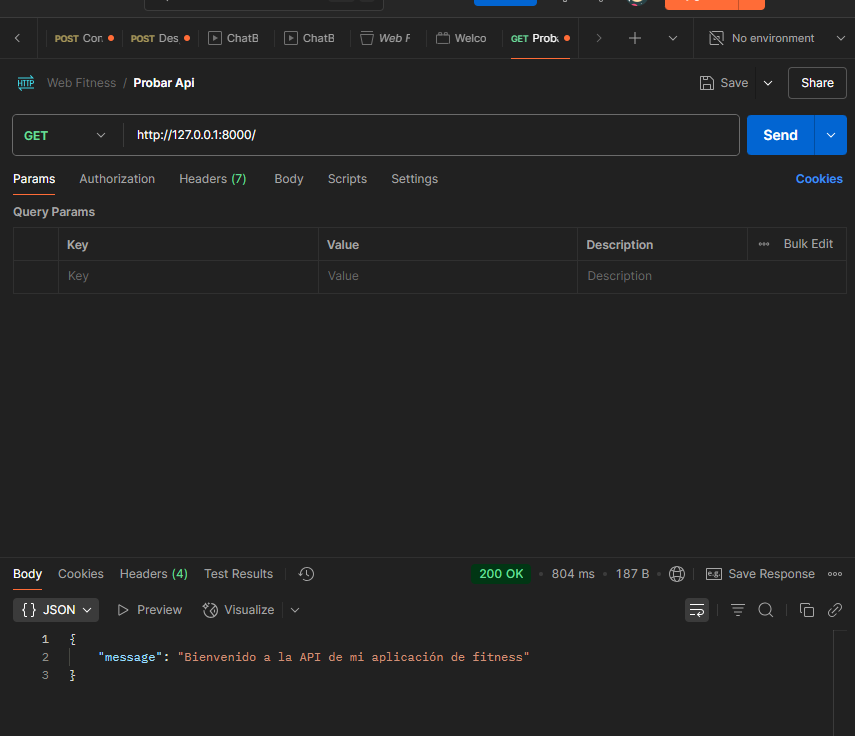


Figura 3. Diagrama Entidad-Relación

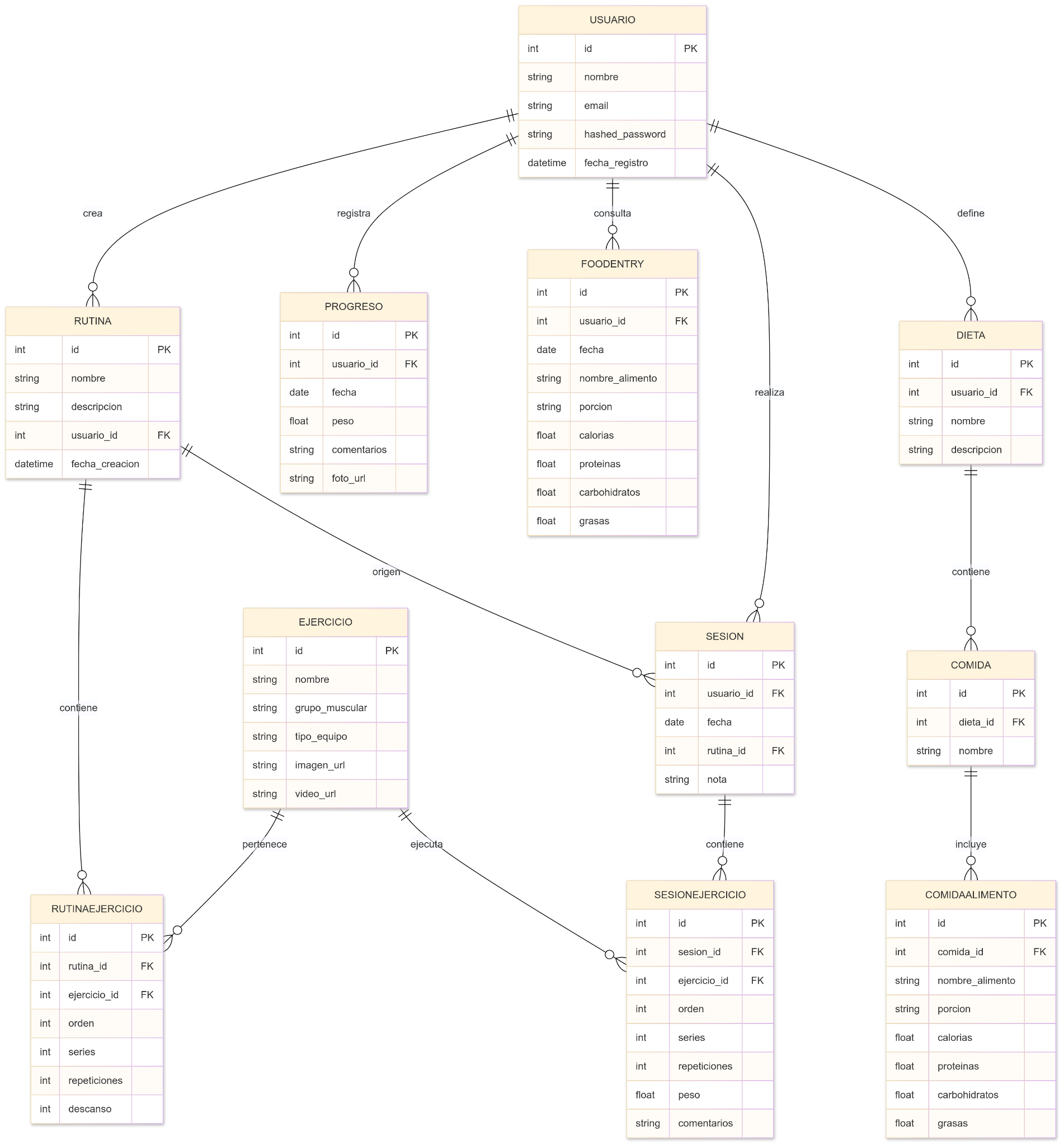


Figura 4. Wireframe de la pestaña “Ejercicios”