



Entrenador Personal Virtual

**Aplicación móvil de gestión inteligente de entrenamientos
personales**

Trabajo de Fin de Grado – DAW

Autor: David Cuevas Gil

Ciclo Formativo: Técnico Superior en Desarrollo de Aplicaciones Web

Fecha: 06/12/2025

INDICE

1. Introducción
 - 1.1. Arquitectura del proyecto
 - 1.2. Objetivo general
2. Matriz DAFO
 - 2.1. Análisis de la Matriz DAFO
3. Análisis del mercado
4. Análisis y diseño del software
 - 4.1. Diagrama de casos de uso.
 - 4.2. Diagrama de clases.
 - 4.3. Diagrama Entidad-Relación.
 - 4.4. Diagrama de componentes.
 - 4.5. Diagrama de actividades.
 - 4.6. Diagrama de secuencia.
 - 4.7. Diagrama de despliegue.
5. Casos de prueba

1. INTRODUCCIÓN

Entrenador Personal Virtual es una aplicación móvil diseñada para facilitar la creación y gestión de rutinas de entrenamiento personalizadas. Permite a los usuarios organizar sus sesiones, consultar ejercicios con vídeos e imágenes, usar utilidades deportivas como Tabata o EMOM y realizar un seguimiento de su progreso desde una interfaz clara e intuitiva.

El objetivo del proyecto es ofrecer una herramienta moderna y accesible que mejore la organización, constancia y motivación del usuario en su actividad deportiva diaria.

1.1. Arquitectura del Proyecto

Frontend — Aplicación móvil

- Desarrollada con React Native + Expo + JavaScript.
- Navegación fluida basada en componentes reutilizables.
- Comunicación con el backend mediante peticiones HTTP (REST + JSON).
- Manejo de autenticación mediante almacenamiento del token JWT.

Backend — API REST con Laravel

- Implementado con PHP 8 y Laravel 12 siguiendo arquitectura de controladores REST.
- Autenticación mediante:
 - JWT (para la app móvil).
 - Sesiones (compatibilidad y pruebas).
- Gestión de archivos multimedia a través de Cloudinary.
- Validación centralizada y respuestas JSON estandarizadas.
- Separación en capas: Controladores → Servicios → Repositorios → Modelos.

1.2. Objetivo General

El objetivo principal de este proyecto es desarrollar una aplicación móvil que permita gestionar entrenamientos personalizados, mostrando ejercicios con imágenes y vídeos explicativos, y ofreciendo herramientas como temporizadores deportivos y organización por rutinas.

La aplicación busca que cualquier usuario pueda entrenar de forma guiada, estructurada y accesible desde una sola plataforma.

2. MATRIZ DAFO

Debilidades	Amenazas
<ul style="list-style-type: none"> • Falta de notificaciones push. • Desarrollo individual. • Algunas funcionalidades en estado inicial. • Falta un sistema avanzado de edición de rutinas. • Dependencia de la conexión a Internet. • Falta integración con IA. 	<ul style="list-style-type: none"> • Competidores de gran potencial y contenidos premium. • Rápida evolución de la integración con IA. • Dependencias de librerías React Native. • Los usuarios buscan soluciones consolidadas.
Fortalezas	Oportunidades
<ul style="list-style-type: none"> • App completa que integra rutinas, ejercicios, publicaciones, calendario y utilidades. • Arquitectura sólida: React Native + Laravel + MySQL. • Contenido multimedia con imágenes en Cloudinary y vídeos cargados mediante URLs desde la base de datos. • Rutinas estructuradas por fases. • Sistema de autenticación seguro. • Diseño propio en la interfaz y para la experiencia del usuario. 	<ul style="list-style-type: none"> • Creciente interés en apps de entrenamiento desde casa. • Apps que combinen rutinas, utilidades deportivas y contenido multimedia. • Posibilidad de añadir funciones futuras. • Apps simples y claras, ideal para usuarios principiantes o intermedios que no quieren complejidad extrema. • Posibilidad futura de integrar IA en recomendaciones o rutinas.

2.1. Análisis de la Matriz DAFO

La matriz DAFO del proyecto *Entrenador Personal Virtual* permite identificar los puntos clave que influyen en su desarrollo y en su posible evolución futura.

Entre las fortalezas, destacan la amplitud funcional de la aplicación, su arquitectura sólida y la integración de contenido multimedia que mejora la experiencia del usuario.

El diseño propio y la estructura de rutinas por fases aportan un valor diferenciador frente a soluciones amateurs.

Las debilidades se centran en aspectos propios de un proyecto desarrollado por una sola persona, como la ausencia actual de notificaciones push, ciertas funcionalidades aún poco maduras y la falta de herramientas avanzadas de edición de rutinas.

También se identifica la dependencia de conexión a Internet como un factor limitante para algunos usuarios.

En cuanto a las oportunidades, el crecimiento del interés por las aplicaciones de entrenamiento, la tendencia hacia soluciones simples y accesibles y la posibilidad de incorporar futuras funciones inteligentes colocan al proyecto en un espacio con potencial de evolución.

Por otro lado, las amenazas principales provienen de la fuerte competencia en el sector fitness, la rápida evolución tecnológica y las expectativas de usuarios que buscan plataformas consolidadas o integradas con IA.

Este análisis permite establecer una visión clara y realista del estado actual del proyecto y de las áreas sobre las que se puede seguir trabajando para potenciar su crecimiento.

3. ANÁLISIS DEL MERCADO

En la actualidad, el mercado de aplicaciones orientadas al entrenamiento y a la actividad física presenta una amplia variedad de soluciones, aunque muchas de ellas se centran únicamente en aspectos concretos como rutinas personalizadas, vídeos de ejercicios, estadísticas de progreso o sistemas de suscripción para acceder a contenido premium.

Sin embargo, pocas aplicaciones logran integrar en un solo entorno todas las funcionalidades que un usuario principiante o intermedio necesita para entrenar de forma guiada y organizada.

La mayoría de apps consolidadas del sector —como FitCoach, Freeletics o Adidas Training— ofrecen planes de entrenamiento avanzados, sistemas de recomendación y seguimiento del progreso, aunque suelen requerir suscripción y están orientadas a usuarios con cierta experiencia o constancia en el entrenamiento. Además, estas aplicaciones priorizan análisis complejos o rutinas muy intensivas, lo que puede alejar a usuarios que buscan soluciones más simples y accesibles.

Por otro lado, existen apps enfocadas únicamente en utilidades deportivas (como temporizadores Tabata, EMOM o cronómetros), y otras centradas en la creación de rutinas, pero sin ofrecer contenido multimedia claro o sin permitir organizar entrenamientos por fases. Tampoco es común encontrar apps que combinen rutinas, ejercicios, temporizadores y publicaciones en una misma plataforma.

Entrenador Personal Virtual se posiciona en un punto intermedio entre ambas realidades: ofrece una herramienta completa, fácil de utilizar, que incluye rutinas estructuradas, un módulo de ejercicios con imágenes y vídeos, utilidades deportivas integradas, un sistema básico de publicaciones y un calendario interactivo. Su enfoque se dirige a usuarios que desean mejorar su organización y constancia sin depender de suscripciones ni configuraciones complejas.

Este espacio del mercado, todavía poco cubierto por aplicaciones integrales y accesibles, representa una oportunidad para el crecimiento del proyecto, especialmente si se incorporan en el futuro funcionalidades como notificaciones push, estadísticas avanzadas o recomendaciones inteligentes basadas en IA.

4. ANALISIS Y DISEÑO DEL SOFTWARE

4.1 - Diagrama de Casos de Uso

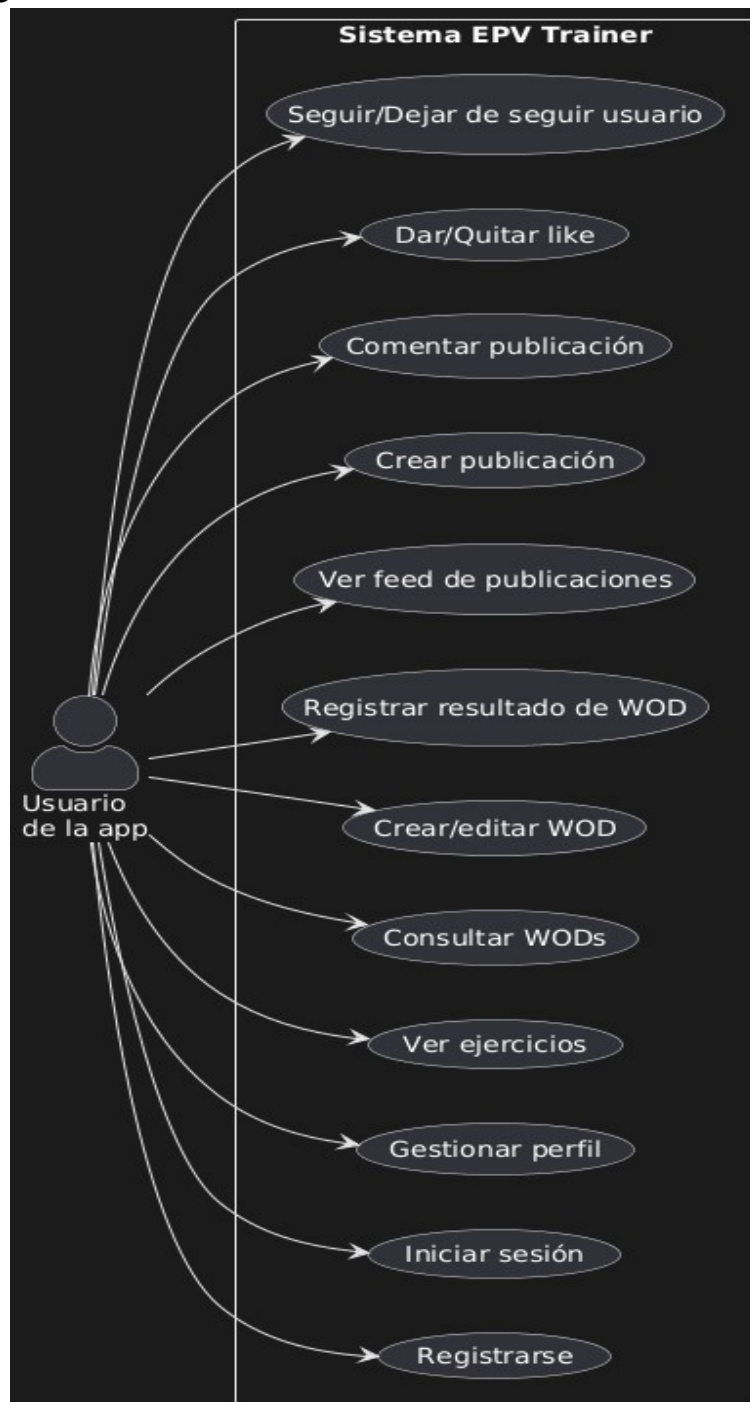


Figura 1 – Diagrama de casos de uso.

En este diagrama se representan las principales acciones que el usuario puede realizar dentro de la plataforma: registrarse, iniciar sesión, gestionar su perfil, consultar ejercicios, crear y publicar WODs, así como interactuar en el módulo social mediante publicaciones, comentarios, likes y seguimiento de otros usuarios.

4.2 - Diagrama de Clases

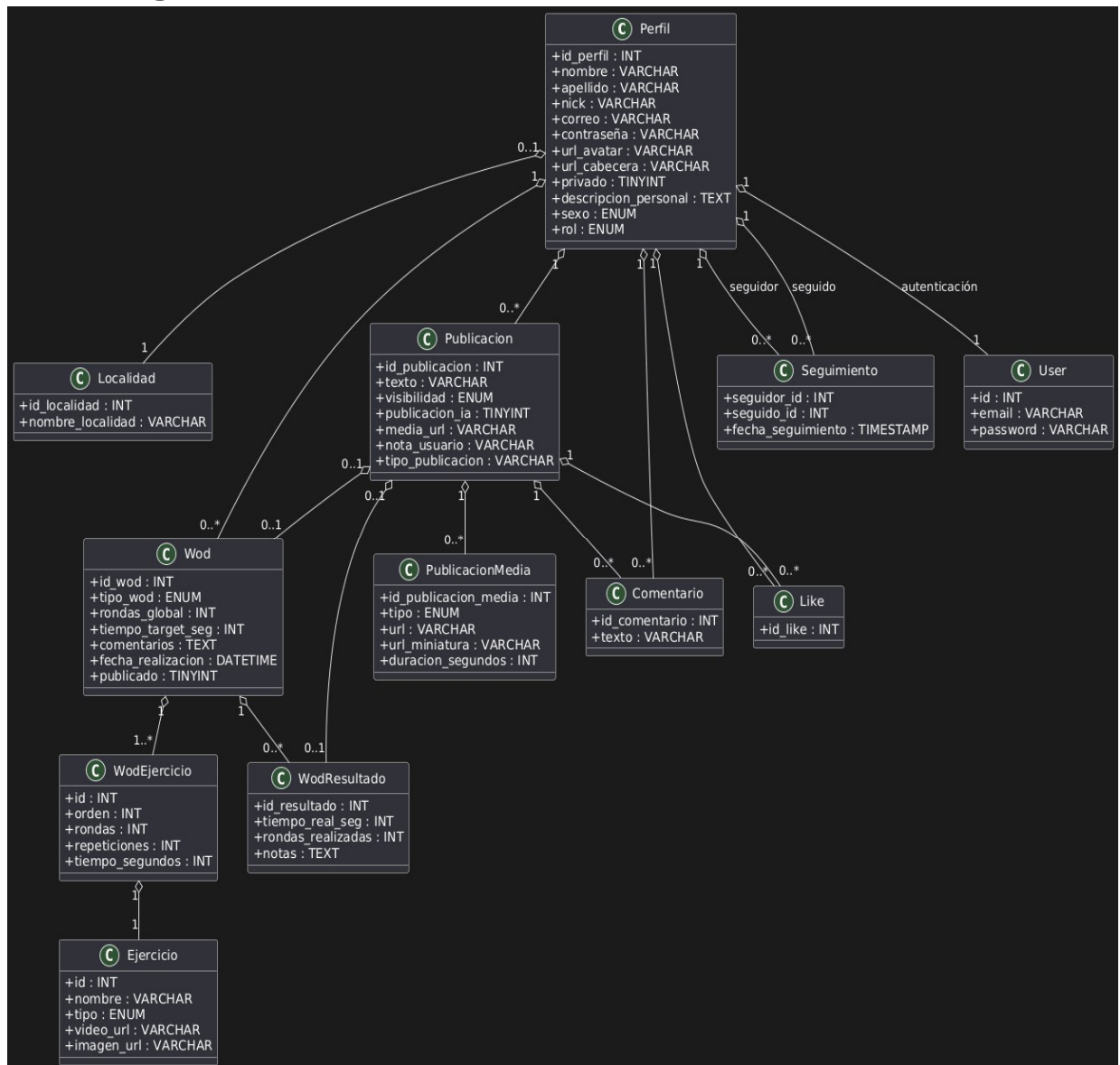


Figura 2 – Diagrama de clases del sistema.

Este diagrama muestra la estructura lógica del backend, incluyendo las entidades principales (Perfil, Ejercicio, WOD, Publicación, Comentario, Like, Seguimiento) y sus relaciones. Se representan atributos clave, relaciones 1-N, N-M y asociaciones necesarias para el funcionamiento del sistema, así como dependencias entre controladores, servicios y repositorios.

4.3 - Diagrama Entidad-Relación

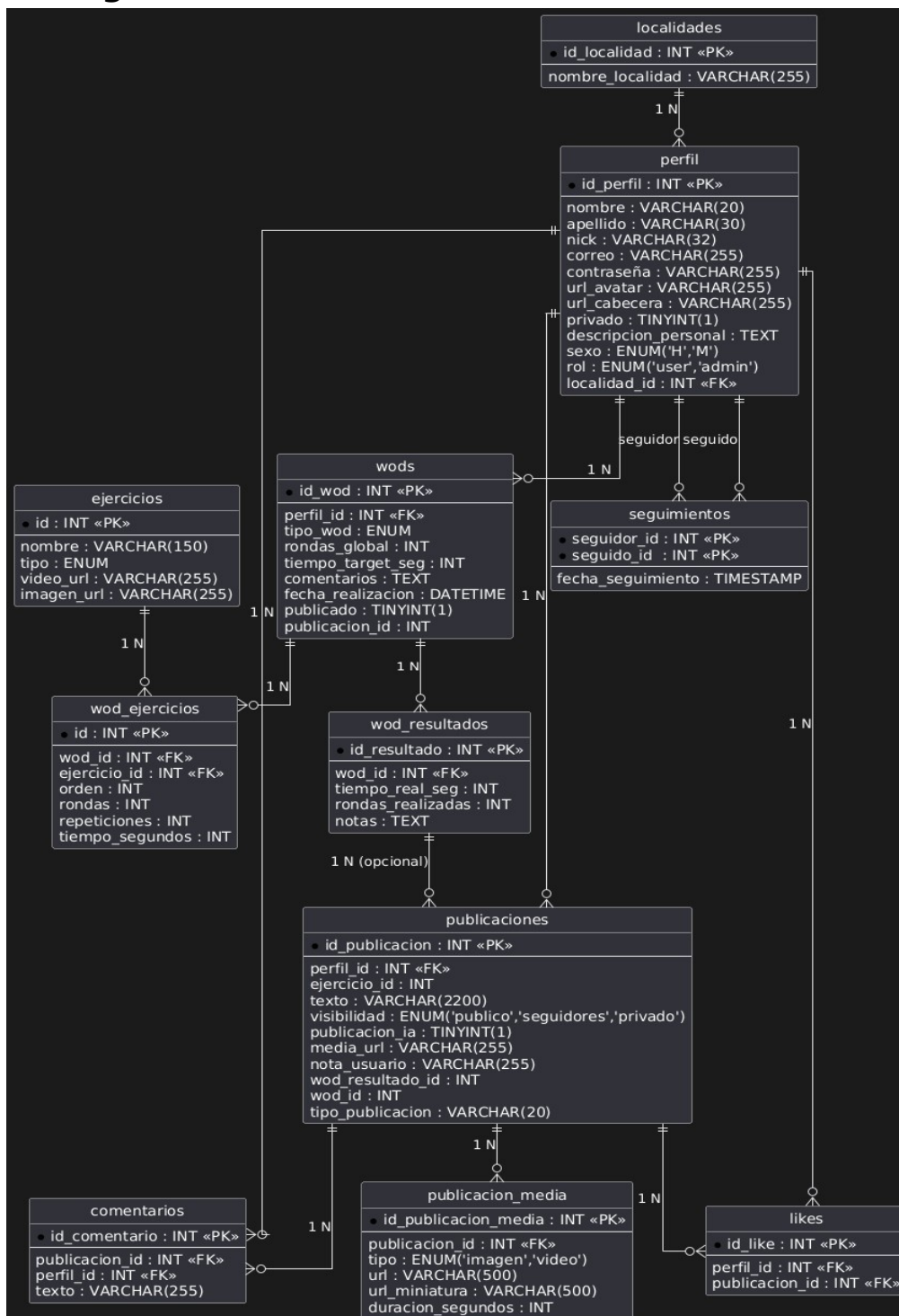


Figura 3 – Diagrama Entidad-Relación de la base de datos.

En él se ilustran las tablas que conforman la base de datos del proyecto, junto con sus claves primarias, claves ajenas y relaciones. Se muestran entidades como perfil, localidades, ejercicios, wods, wod_ejercicios, publicaciones, publicacion_media, comentarios, likes, seguimientos y las interdependencias que permiten la coherencia del modelo.

4.4 - Diagrama de componentes

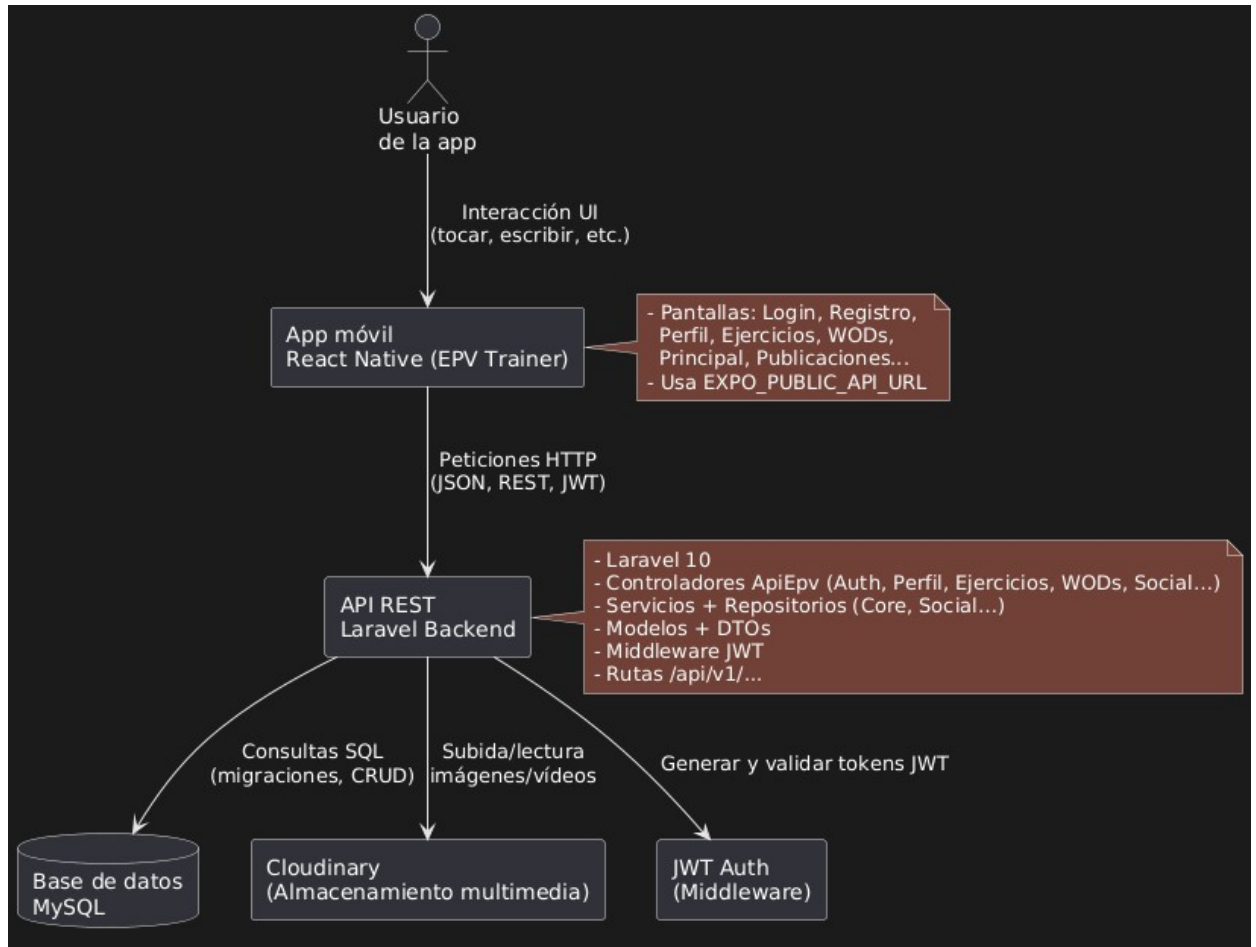


Figura 4 – Diagrama de componentes del sistema.

Este diagrama desglosa la arquitectura principal: la app móvil en React Native, la API REST desarrollada en Laravel, la base de datos MySQL y los servicios externos como Cloudinary o el middleware JWT. Representa cómo cada componente se comunica mediante peticiones HTTP y cómo se gestionan datos, autenticación y contenido multimedia.

4.5 – Diagrama de actividades

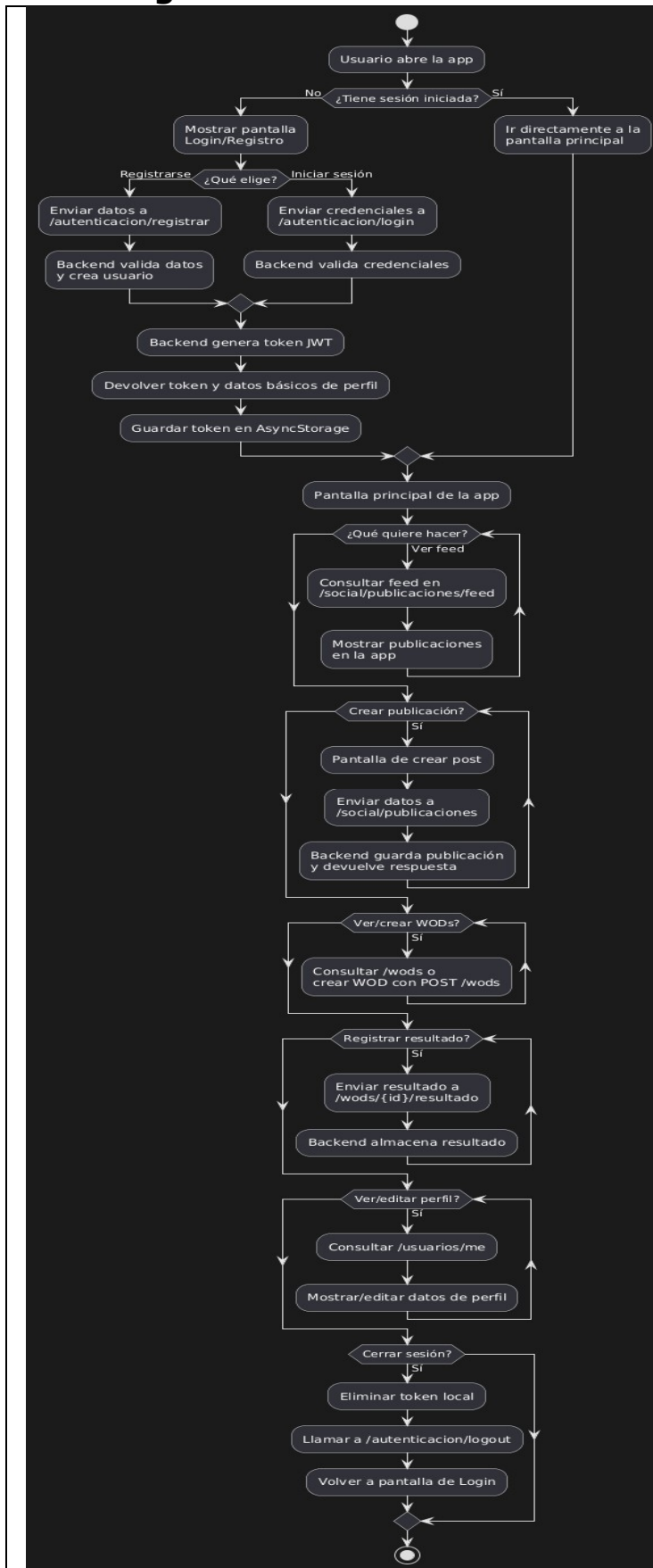


Figura 5 – Diagrama de actividades del flujo principal.

Modela el proceso que sigue un usuario al abrir la app: autenticación, navegación, consulta de información, creación de contenido y acciones en el módulo social. El diagrama representa decisiones, ramificaciones y secuencias operativas del sistema desde una perspectiva funcional.

4.6 - Diagrama de secuencia

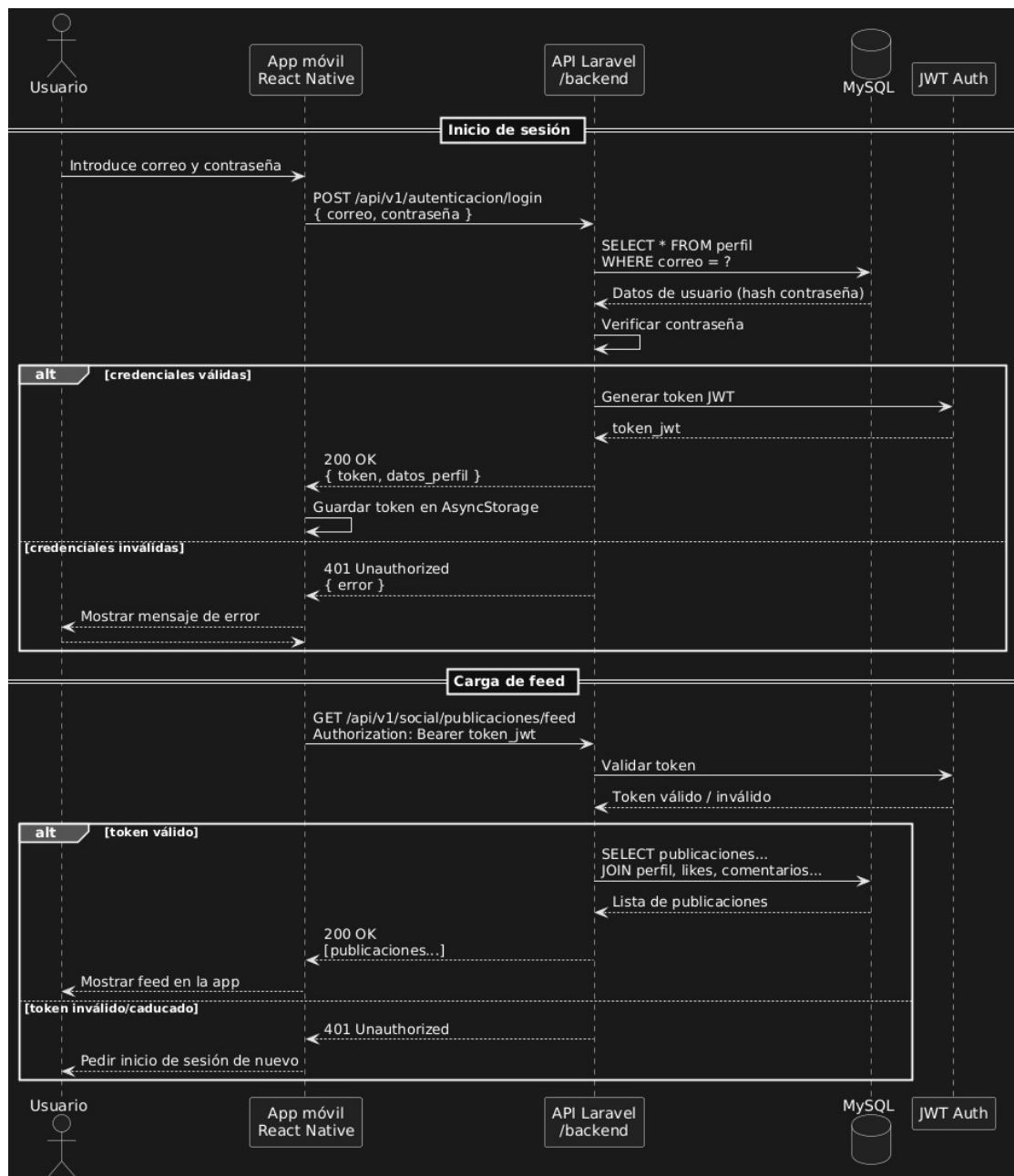


Figura 6 – Diagrama de secuencia del inicio de sesión y carga del feed.

Describe paso a paso la interacción entre el usuario, la app móvil, el backend Laravel, la base de datos y el sistema JWT. Refleja cómo se valida al usuario, cómo se genera el token de acceso y cómo se recuperan las publicaciones del feed de manera ordenada.

4.7 - Diagrama de despliegue

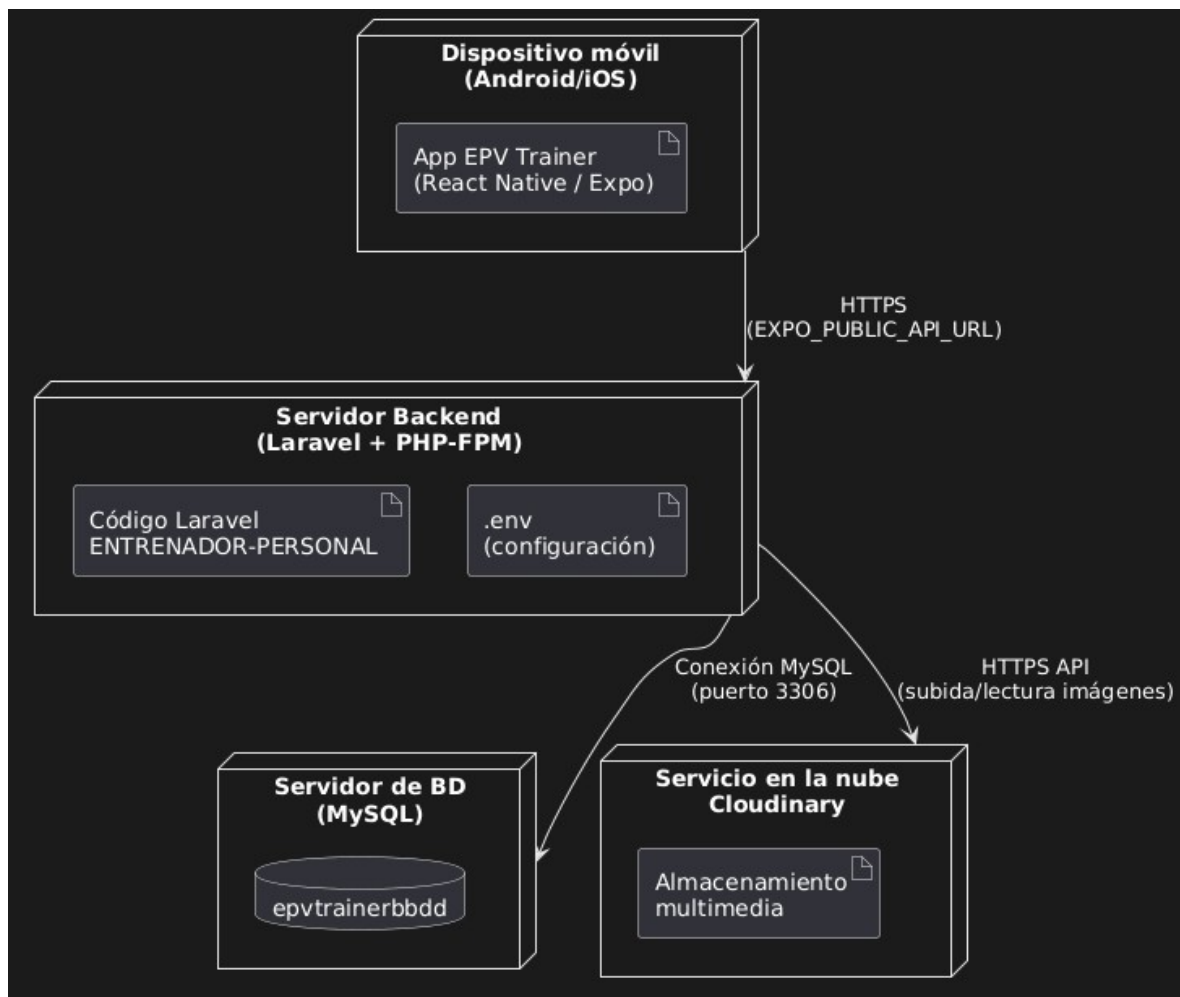


Figura 7 – Diagrama de despliegue del proyecto EPV Trainer.

Muestra la infraestructura donde se ejecuta el backend de Laravel (servidor local o nube), la base de datos MySQL y el dispositivo móvil donde corre la app. El diagrama refleja los nodos, conexiones y tecnologías que intervienen en la puesta en funcionamiento del sistema.

5. CASOS DE PRUEBA

ID	CP-01
Módulo	Autenticación
Caso de prueba	Registro correcto
URL / Método	POST {{base_url}}/autenticacion/regar
Headers	Content-Type: application/json
Body	{ "nombre":"David", "apellido":"Cuevas", "nick":"davidcg", "correo":"david@example.com", "contraseña":"12345678" }
Resultado esperado	"message": "Registro correcto"
Resultado obtenido	"message": "Registro correcto"

ID	CP-02
Módulo	Autenticación
Caso de prueba	Registro correo existente
URL / Método	POST {{base_url}}/autenticacion/regar
Headers	Content-Type: application/json
Body	{ "nombre":"Test", "apellido":"Prueba", "nick":"test1", "correo":"david@example.com", "contraseña":"12345678" }
Resultado esperado	"message": "The correo has already been taken."
Resultado obtenido	"message": "The correo has already been taken."

ID	CP-03
Módulo	Autenticación
Caso de prueba	Login correcto
URL / Método	POST {{base_url}}/autenticacion/login
Headers	Content-Type: application/json, Authorization: Bearer {{token}}
Body	{ "correo":"david@example.com", "contraseña":"12345678" }
Resultado esperado	{ "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...", "token_type": "bearer" }
Resultado obtenido	{ "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...", "token_type": "bearer" }

ID	CP-04
Módulo	Autenticación
Caso de prueba	Login incorrecto
URL / Método	POST {{base_url}}/autenticacion/login
Headers	Content-Type: application/json
Body	{ "correo": "david@example.com", "contraseña": "incorrecta" }
Resultado esperado	{ "error": "Credenciales invalidas" }
Resultado obtenido	{ "error": "Credenciales invalidas" }

ID	CP-05
Módulo	Perfil
Caso de prueba	Ver perfil propio
URL / Método	GET {{base_url}}/usuarios/me
Headers	Authorization: Bearer {{token}}
Body	-
Resultado esperado	{ "id_perfil": 21 }
Resultado obtenido	{ "id_perfil": 21 }

ID	CP-06
Módulo	Perfil
Caso de prueba	Editar perfil
URL / Método	PUT {{base_url}}/usuarios/actualizar-perfil
Headers	Authorization: Bearer {{token}} Content-Type: application/json
Body	{ "descripcion_personal": "Nueva descripción desde Postman" }
Resultado esperado	{ "descripcion_personal": "Nueva descripcion_personal desde Postman" }
Resultado obtenido	{ "descripcion_personal": "Nueva descripcion_personal desde Postman" }

ID	CP-07
Módulo	Ejercicios
Caso de prueba	Listar ejercicios
URL / Método	GET {{base_url}}/ejercicios
Headers	
Body	
Resultado esperado	[{ "id": 58, "nombre": "Abdominales", "tipo": "metabolico", ... }]
Resultado obtenido	[{ "id": 58, "nombre": "Abdominales", "tipo": "metabolico", ... }]

ID	CP-08
Módulo	Ejercicios
Caso de prueba	Detalle ejercicio
URL / Método	GET {{base_url}}/ejercicios/1
Headers	
Body	
Resultado esperado	{ "id": 1, "nombre": "Carga y empuje unilateral con mancuernas", "tipo": "fuerza", "video_url": "https://www.youtube.com/watch?..." }
Resultado obtenido	{ "id": 1, "nombre": "Carga y empuje unilateral con mancuernas", "tipo": "fuerza", "video_url": "https://www.youtube.com/watch?..." }

ID	CP-09
Módulo	WODs
Caso de prueba	Crear WOD
URL / Método	POST {{base_url}}/wods
Headers	Authorization: Bearer {{token}} Content-Type: application/json
Body	{ "tipo_wod": "tiempo", "rondas_global": 5, "tiempo_target_seg": 300, ... }
Resultado esperado	{ "tipo_wod": "tiempo", "rondas_global": 5, "tiempo_target_seg": 300, ... }
Resultado obtenido	{ "tipo_wod": "tiempo", "rondas_global": 5, "tiempo_target_seg": 300, ... }

ID	CP-10
Módulo	WODs
Caso de prueba	Publicar WOD
URL / Método	POST {{base_url}}/wods/145/resultado
Headers	Authorization: Bearer {{token}}
Body	
Resultado esperado	{ "resultado_id": 67, "publicacion_id": 85 }
Resultado obtenido	{ "resultado_id": 67, "publicacion_id": 85 }

ID	CP-11
Módulo	Publicaciones
Caso de prueba	Crear publicación
URL / Método	POST {{base_url}}/social/publicaciones
Headers	Authorization: Bearer {{token}} Content-Type: application/json
Body	{ "texto": "Publicación de prueba desde Postman" }
Resultado esperado	{ "data": { "id_publicacion": 86, "perfil_id": 21, "ejercicio_id": null, ... } }
Resultado obtenido	{ "data": { "id_publicacion": 86, "perfil_id": 21, "ejercicio_id": null, ... } }

ID	CP-12
Módulo	Comentarios
Caso de prueba	Comentar publicación
URL / Método	POST {{base_url}}/social/comentarios
Headers	Authorization: Bearer {{token}} Content-Type: application/json
Body	{ "publicacion_id": 1, "texto": "Buen post!" }
Resultado esperado	{ "data": { "id_comentario": 45, "publicacion_id": 1, "texto": "Buen post!", ... } }
Resultado obtenido	{ "data": { "id_comentario": 45, "publicacion_id": 1, "texto": "Buen post!", ... } }

ID	CP-13
Módulo	Likes
Caso de prueba	Toggle Like
URL / Método	POST {{base_url}}/social/likes/toggle
Headers	Authorization: Bearer {{token}} Content-Type: application/json
Body	{ "publicacion_id": 1 }
Resultado esperado	{ "liked": true, "total_likes": 1, "publicacion_id": 1 }
Resultado obtenido	{ "liked": true, "total_likes": 1, "publicacion_id": 1 }

ID	CP-14
Módulo	Seguimientos
Caso de prueba	Seguir / Dejar de seguir
URL / Método	POST {{base_url}}/social/seguimientos/toggle
Headers	Authorization: Bearer {{token}} Content-Type: application/json
Body	{ "seguido_id": 14 }
Resultado esperado	{ "following": true, "total_seguidores": 4 }
Resultado obtenido	{ "following": true, "total_seguidores": 4 }

ID	CP-15
Módulo	Seguridad
Caso de prueba	Acceso sin token
URL / Método	GET {{base_url}}/usuarios/me
Headers	
Body	
Resultado esperado	{ "message": "Token not provided" }
Resultado obtenido	{ "message": "Token not provided" }

ID	CP-16
Módulo	Seguridad
Caso de prueba	Token inválido
URL / Método	GET {{base_url}}/usuarios/me
Headers	Authorization: Bearer token_invalido
Body	
Resultado esperado	{ "message": "Token inválido." }
Resultado obtenido	{ "message": "Token inválido." }