



\*\*

# PROYECTO: RAYOSTRENGTH APP

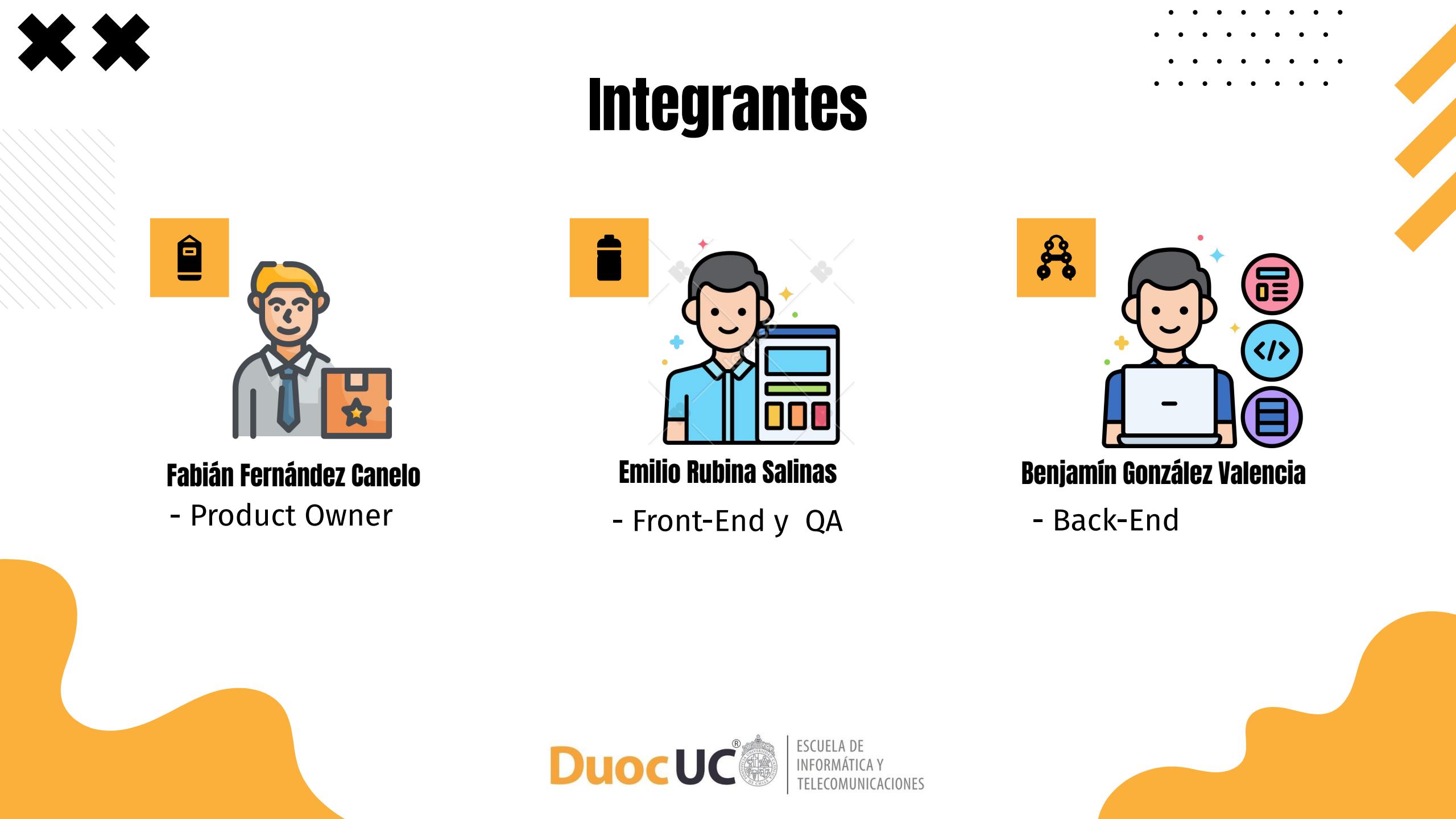
PRESENTACIÓN FINAL CAPSTONE

\*\*



DuocUC®

ESCUELA DE  
INFORMÁTICA Y  
TELECOMUNICACIONES



# Integrantes



**Fabián Fernández Canelo**  
- Product Owner



**Emilio Rubina Salinas**  
- Front-End y QA



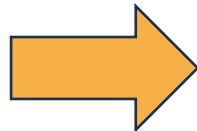
**Benjamín González Valencia**  
- Back-End

# Descripción del proyecto



## Problema o dolor

- Las personas que entran no cuentan con una plataforma centralizada para gestionar rutinas personalizadas.
- Muchos usuarios pierden seguimiento del progreso, no saben si están mejorando y no tienen claridad sobre qué entrenar.
- Los entrenadores no tienen un sistema simple para crear, editar y asignar rutinas a múltiples usuarios.
- Todo se hace de manera desordenada: en notas, mensajes, PDFs, o planillas que no se actualizan en tiempo real.



## Propuesta de solución

- RayoStrength app centraliza todo el proceso de entrenamiento: rutinas, usuarios y progreso.
- Permite a los administradores y coaches crear, modificar y asignar rutinas de manera rápida y organizada.
- Ofrece a los usuarios una forma clara de ver su entrenamiento y registrar avances.
- Asegura orden, trazabilidad y una experiencia más profesional tanto para usuarios como para entrenadores.

XX

# Objetivo General

Desarrollar una aplicación móvil que permita a entrenadores personales gestionar rutinas, pagos, vídeos y videollamadas de forma simple y eficiente

## Objetivos Específicos

- Implementar un sistema para crear, editar, eliminar y asignar rutinas desde la app.
- Diseñar una interfaz móvil intuitiva y fácil de usar para la gestión de usuarios y rutinas.
- Incorporar funcionalidad de videollamadas.
- Construir un backend robusto que asegure seguridad, integridad y disponibilidad de la información.
- Permitir que los usuarios visualicen claramente sus rutinas y registren o sigan su progreso.
- Garantizar una experiencia estable mediante pruebas, validaciones y control de calidad.
- Integrar enlaces a vídeos de ejercicios (Youtube).

# Alcances y limitaciones del proyecto

## Alcance

- Desarrollo de una aplicación móvil para gestión de rutinas de entrenamiento.
- Módulos de administración de usuarios, incluyendo: creación, edición y eliminación.
- Creación, modificación, eliminación y asignación de rutinas personalizadas.
- Conexión con un backend para manejo de datos y autenticación.
- Interfaz intuitiva y orientada a una experiencia simple y funcional.

## Limitaciones

- La aplicación no incluye seguimiento avanzado del progreso físico (gráficas, estadísticas complejas).
- No posee integración con sensores o dispositivos externos (EJ: smartwatch, banda cardíaca).
- No ofrece funciones de comunidad, chat o interacción social entre usuarios.
- El sistema de rutinas se centra en CRUD básico, sin recomendaciones automáticas o IA.
- La versión actual está enfocada en funcionalidades base, sin módulos de pagos ni monetización.

# Metodología de trabajo para el desarrollo del proyecto

Se trabajó con **Scrum**, organizando el proyecto en **9 épicas** y **6 sprints**. Cada sprint incluyó planificación, desarrollo, pruebas y revisión.

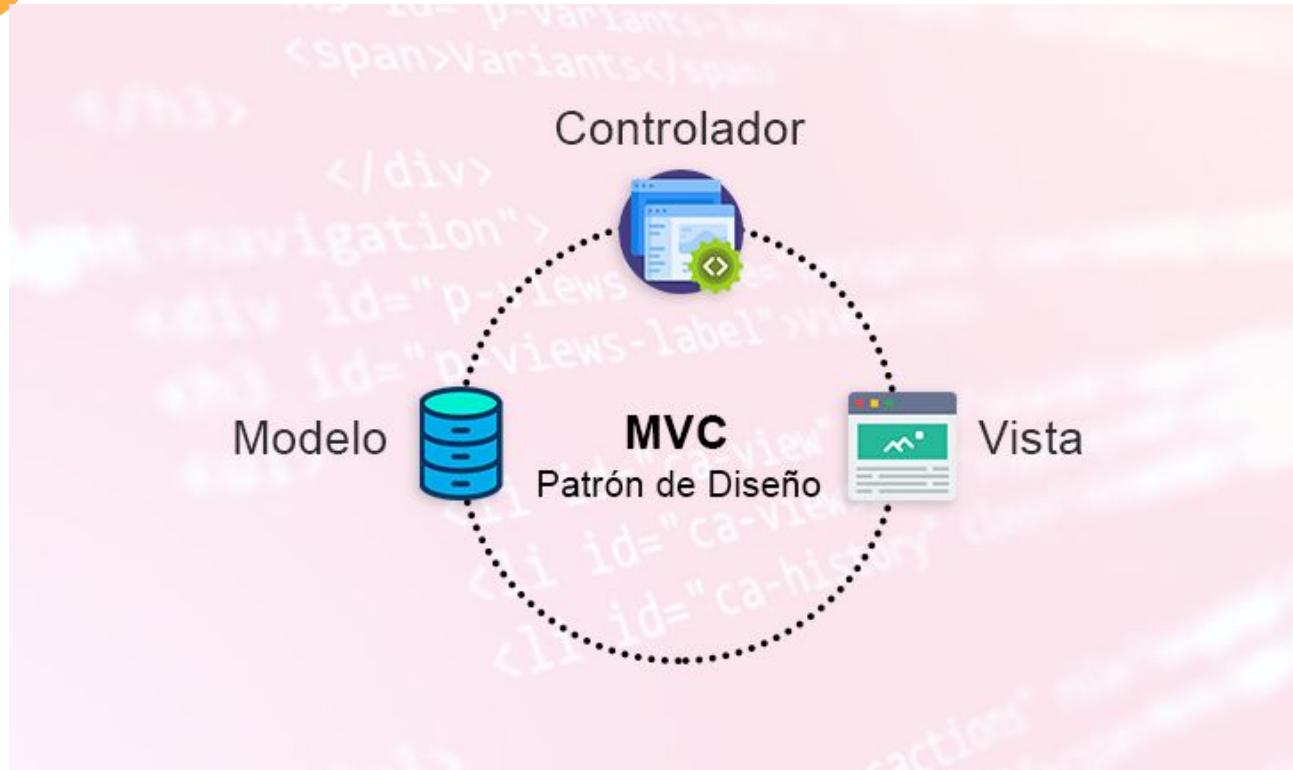
El avance se gestionó en **Jira**, asegurando un proceso iterativo, ordenado y con entregas funcionales en cada ciclo.



# Roadmap para el desarrollo del proyecto

Actividad	August	September	October	November	December	
Sprints	Sprint 1 - Base	Sprint 2 - Gestión	Sprint 3 - Gestión	Sprint 4 - Dashbo	Sprint 5 - Infra y	Sprint 6 - Revisión
> RC-1 Gestión de Usuarios						
> RC-19 Gestión de Coaches						
> RC-29 Gestión de Rutin...						
> RC-42 Dashboard (visualización de progreso y d...						
> RC-50 Integración con Google Shee...						
> RC-59 Infraestructura y despliegue						
> RC-67 Seguridad y control de acce...						
> RC-72 Gestión de errores (bugs)						

# Arquitectura del software

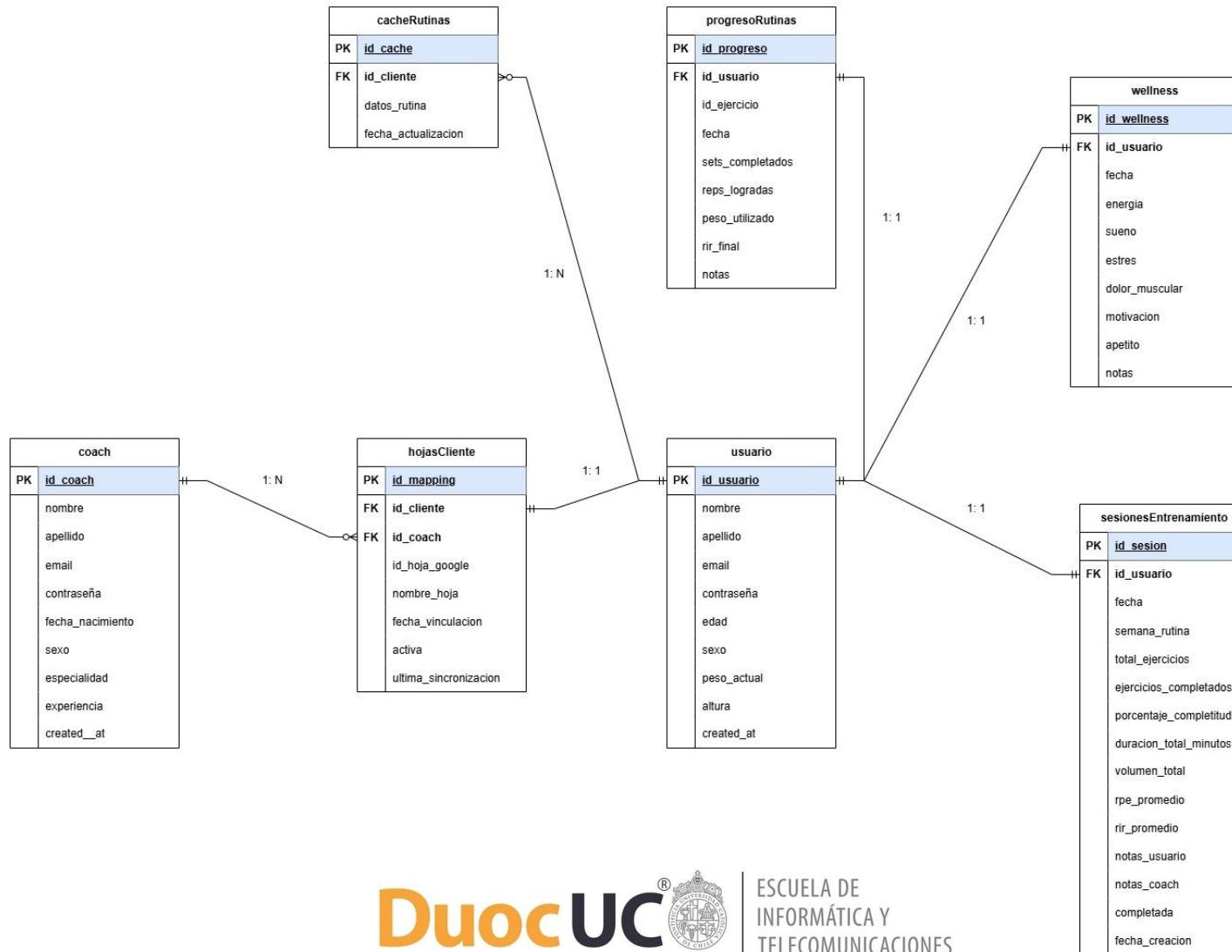


## ¿Por qué MVC?

- Separa claramente las responsabilidades del sistema.
- Facilita el mantenimiento, escalabilidad y pruebas.
- Permite que el front-end y el back-end trabajen de forma más ordenada e independiente.
- Hace más simple integrar nuevas funciones sin afectar el resto del código.

# Modelo de datos

RAYOSTRENGTH



# Tecnologías utilizadas



## Railway

Despliegue backend en la nube



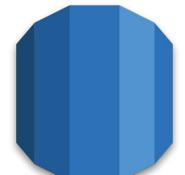
## Github

Utilizada para trabajar código en la nube



## HeidiSQL

Software utilizado para el uso de MySQL y conectarnos a la base de datos



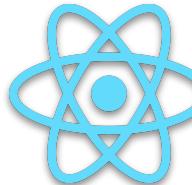
## AWS RDS

Utilizada para despliegue de la base de datos en la nube



## Visual Studio Code

Utilizado como editor de código



## React Native

Biblioteca utilizada para la creación del interfaz de usuario



## JavaScript

Lenguaje de programación de la app



## Node Js

Utilizado como backend para manejar la lógica del servidor, autenticación y comunicación con la base de datos



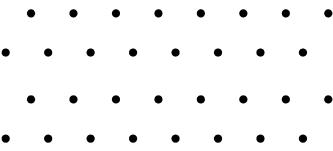
## Expo

Utilizado para probar la app en dispositivos móviles y visualizar los cambios en la app



## Android studio

Utilizado para obtener al apk de la app



# Demostración del resultado del proyecto



# Resultados obtenidos

- Desarrollo de una aplicación móvil funcional para gestión de rutinas.
- Implementación de módulos clave: usuarios, rutinas y asignación de entrenamientos.
- Backend operativo con API y base de datos.
- Integración completa entre frontend, backend y base de datos.
- Validación del sistema mediante pruebas funcionales de QA.
- Producto final estable y alineado con los objetivos del proyecto.

# Obstáculos presentados durante el desarrollo

- Deficiente comunicación
- Retrasos por parte del equipo
- Problemas con GitHub por credenciales
- Complejidad de API de Google Sheets
- Problemas con conexión de back-end con Railway
- Cambios por parte del cliente y agregación de nuevas funcionalidades
- Dificultad al aprender a usar Jira los que provocó una reestructuración tardía del proyecto
- Mala definición de cargas de trabajo

# PREGUNTAS

