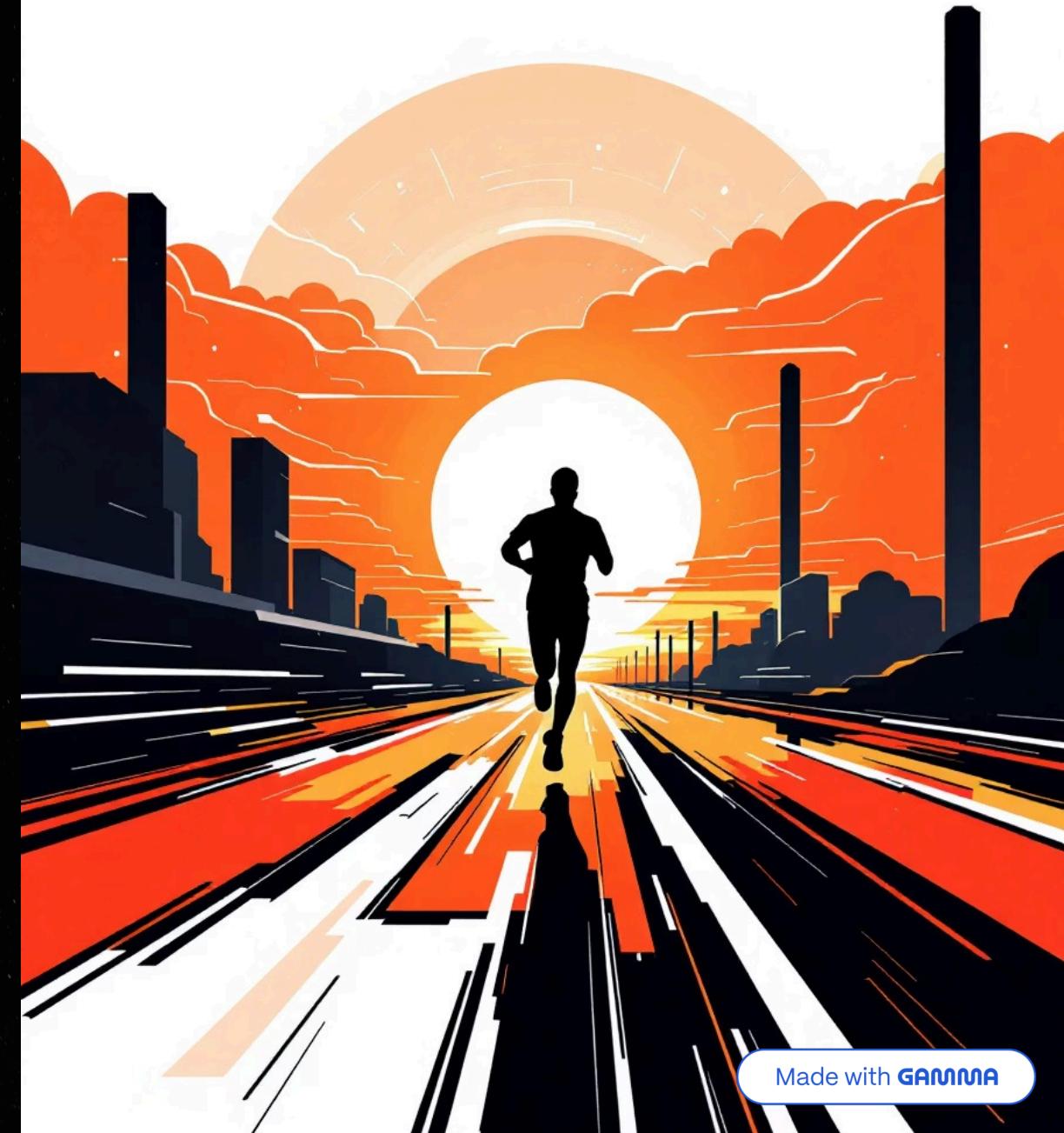


# Ghost Running

## Equipo: UNemployed

- Ángel Gómez
- Jauri Cortés
- Diego Cabrera
- Pablo Luna



Made with GAMMA

# Recordando el proyecto

## El Problema

Muchos runners pierden motivación al entrenar solos, sin un referente claro contra el cual medir su progreso y desempeño diario.



### Graba tu entrenamiento

Registra cada sesión con precisión



### Compete contra tu Ghost

Reta a tu versión anterior



### Mejora continua

Supera tus propias marcas

## Nuestra Solución

Ghost Running permite competir contra versiones previas de ti mismo, grabando entrenamientos para usarlos como "fantasmas" en futuras sesiones.

# Arquitectura del Proyecto

Implementamos una arquitectura en tres capas claramente definidas para garantizar escalabilidad, mantenibilidad y separación de responsabilidades en el desarrollo de Ghost Running.



## Capa de Presentación (Frontend)

Interfaz móvil creada con **Expo + React Native + TypeScript**. Captura GPS, muestra mapas con **react-native-maps**, genera snapshots y envía los datos al backend. Maneja navegación con **React Navigation** y estado con contextos.

## Capa de API / Controladores (Backend)

Capa construida con **Node.js + Express + TypeScript**. Expone endpoints REST, recibe solicitudes del cliente y coordina validaciones básicas. Define la comunicación entre el frontend y el backend.



## Capa de Lógica de Negocio (Backend)

Implementada en **TypeScript** dentro del backend. Contiene la lógica central: cálculos de distancia/velocidad, validaciones y reglas del dominio. Reutilizable y separada de los controladores para mantener claridad y escalabilidad.

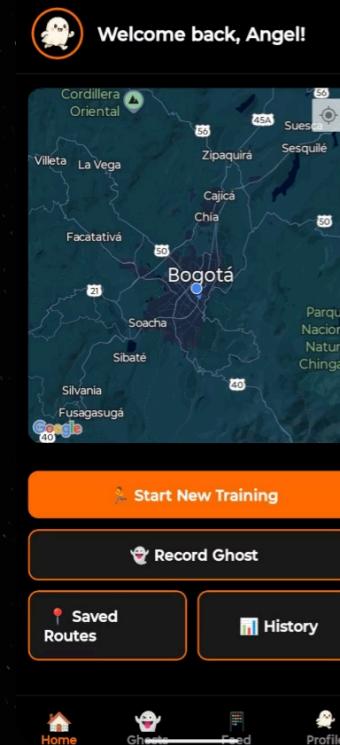
## Capa de Persistencia (Back + Database)

Manejada con **TypeORM** y **PostgreSQL**. Administra el guardado de datos y el almacenamiento de imágenes en disco usando el sistema de archivos del servidor.

# La aplicación en funcionamiento

Video Demostrativo  
Enlace al video:

[Insertar URL del video aquí]



# Componentes Técnicos Clave

1

## Sistema de Tracking GPS

Utiliza **Expo Location** para capturar coordenadas cada 3 segundos con precisión adecuada para actividad física. Los datos incluyen latitud, longitud, altitud y timestamps, que luego se envían al backend para construir el track completo del entrenamiento.

2

## Competencia contra Fantasma

Implementación backend que permite **reemplazar o comparar** una carrera previa del usuario en la misma ruta. El sistema identifica un entrenamiento marcado como *ghost* y puede sustituirse por uno nuevo mediante validaciones en el endpoint

3

## Módulo de Análisis de Rendimiento

Realiza cálculos de **distancia, duración, velocidad promedio, ritmo y calorías estimadas**.

Estos valores se derivan de los datos de coordenadas y se almacenan junto al entrenamiento.

El frontend muestra estos resultados en pantallas de detalle.

4

## Sistema de Autenticación

Autenticación estándar basada en el backend, gestionada desde el frontend mediante AuthContext.

El usuario se identifica por email o con OAuth de Google y el sistema maneja permisos necesarios para ubicación.



# ¿Que logramos?

01

---

Aplicacion desplegable para IOS y Android.

02

---

Login y Signup con credenciales y con Google.

03

---

Grabado de entrenamiento y calculo de estadisticas.

04

---

Sincronizacion de funcionalidad Ghost.

05

---

Guardado y visualizacion de entrenamientos ya completados.

# ¿Que nos faltó?

01

**Login y Signup con Facebook**

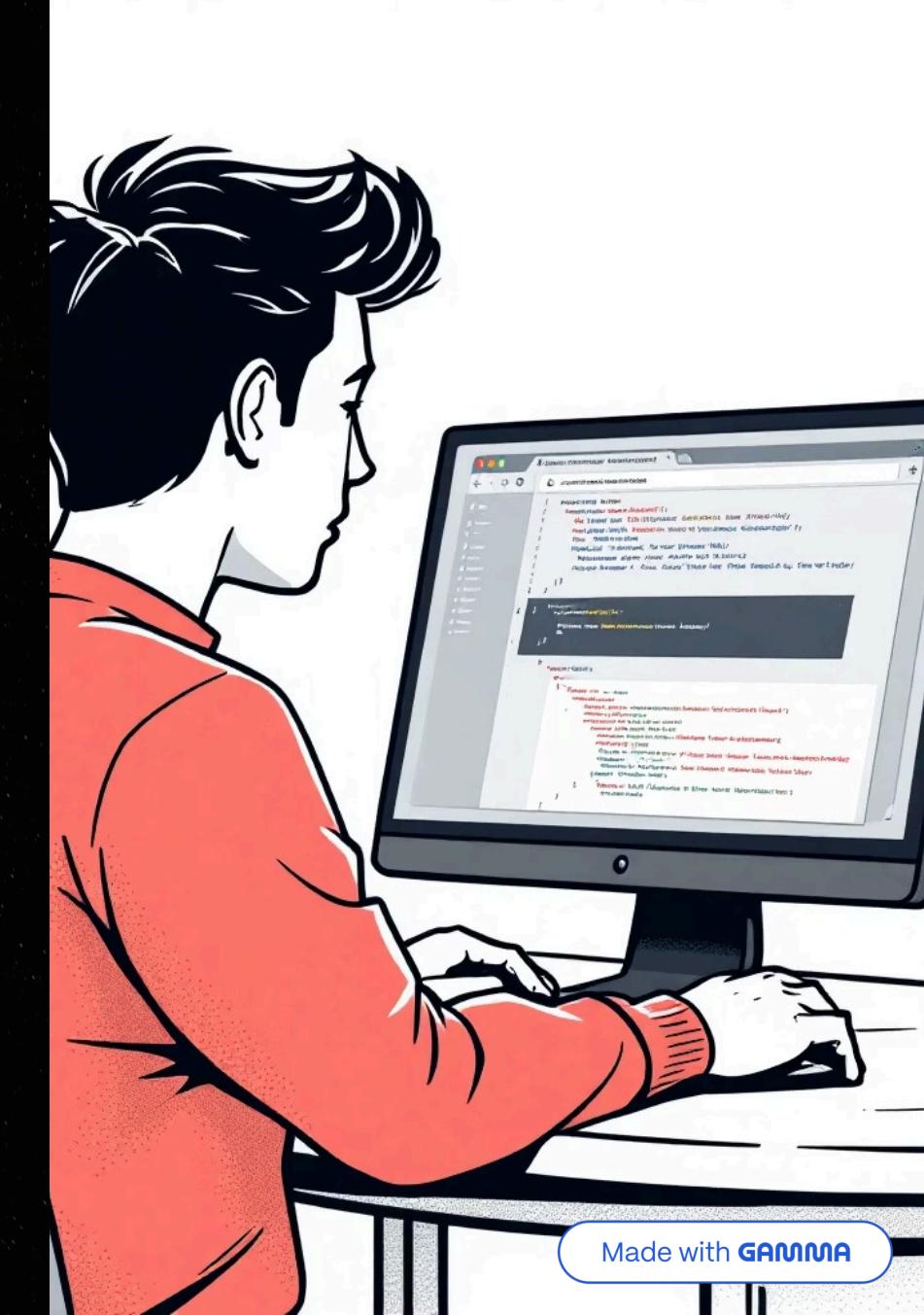
Google OAuth tomo demasiado tiempo.

02

**Parte de desafios semanales.**

03

**Parte del feed e interacciones entre usuarios.  
(No era Must)**



# Desafíos Técnicos Principales

## Iniciar

Principalmente el buscar el hilo de por que tarea iniciar el desarrollo, tal vez fue una de las tareas que mas costo.

## Separacion de Tareas

Una vez encontrada la tarea por la cual iniciar, era dificil la division de tareas sin que el desarrollo fuera tan secuencial.

Una vez encontrada la tarea mas escencial fue mas sencillo ver que tareas podiamos ir haciendo en paralelo.

## Sincronizacion del Ghost y Manejo de mapas.

Dificil manejar la sincronizacion del ghost mientras se ejecutaba el entrenamiento base, ademas el trabajo con mapas no es de lo mas sencillo del mundo.

## Google OAuth

Simplemente Google OAuth.

Nos dio problemas hasta el ultimo momento.

# Proceso de Autoaprendizaje:

## En general

TODO, pues para varias personas de nuestro equipo no habíamos trabajado en una aplicación totalmente hecha, es decir desde la base de datos hasta el frontend. Y mucho menos en TypeScript con React.

Fue una experiencia muy grata ver como poco a poco se iba construyendo el proyecto.

## Division de Tareas y trabajo en equipo

Como funciona el trabajo en equipo y la division de tareas fue una parte esencial del proyecto.

Agradecemos a Git y GitHub.

## Trabajo en Desarrollo Móvil

En nuestro equipo, muy pocas veces se había trabajado en desarrollo móvil, es una lección bastante útil e importante por saber.

# Sugerencias para el curso



## API - REST

- Si pedimos ver API - REST antes de empezar el modulo de desarrollo o cerca del inicio, consideramos que ayuda mucho a entender buena parte del desarrollo.



## Algun tema en Documentación

- Tal vez acerca de como leer, alguna recomendacion desde la experiencia del profe o tips para entender mejor documentacion dificil, puede ser pequeño pero consideramos que es util



## Aprovechar en clase

- Por ejemplo crear un poquito de software en clase, que la parte nueva como conexiones entre capas se visualizara mas en clase.



## Biblioteca de Recursos Curados

- Visualizar algunos ejemplos de software ya creados que nos ayuden a reconocer las capas o los lugares de ciertos archivos, ver como funciona en otros proyectos.