Resumen del proyecto – App Web para Profesores

Este documento resume todo lo trabajado durante el setup inicial del proyecto, incluyendo decisiones tecnológicas, instalación local y estado actual del sistema.

1. Objetivo del trabajo realizado

El objetivo fue dejar un entorno de desarrollo local completamente funcional para una app web profesional orientada a docentes e instituciones, con backend real, base de datos persistente y una arquitectura preparada para escalar.

2. Stack tecnológico elegido

- **Node.js**: runtime principal del backend.
- **NestJS (TypeScript)**: framework backend estructurado, con módulos, servicios y controladores.
- **PostgreSQL 16**: base de datos relacional, ejecutándose en Docker.
- **Docker + Docker Compose**: para levantar y mantener la base de datos en segundo plano.
- **Prisma ORM (v6.x)**: ORM clásico y estable para modelar la base, migraciones y acceso tipado.
- **pnpm**: gestor de paquetes rápido y eficiente.
- **Arquitectura monorepo**: separación clara entre API, frontend y futuros workers.

3. Estructura del proyecto

La estructura actual del repositorio es la siguiente:

profesores-app/ ■■ apps/ ■ ■■ api/ (NestJS + Prisma) ■ ■■ web/ (frontend – pendiente) ■ ■■ worker/ (futuro: PDFs, jobs) ■■ packages/ ■ ■■ shared/ (tipos compartidos – futuro) ■■ docker-compose.yml ■■ package.json ■■ pnpm-workspace.yaml

4. Instalación y configuración realizadas

- Instalación y verificación de Node.js y pnpm.
- Instalación y configuración de Docker Desktop.
- Creación de contenedor PostgreSQL con Docker Compose.
- Inicialización de monorepo con pnpm.
- Creación de la API con NestJS.
- Instalación y configuración de Prisma ORM.
- Conexión de Prisma con PostgreSQL.
- Ejecución de la primera migración.
- Creación de un endpoint real que escribe y lee desde la base.

5. Estado actual del sistema

- PostgreSQL corriendo en Docker (puerto 5432).
- Base de datos 'profesores' creada.
- Tablas creadas y controladas por Prisma.
- API NestJS corriendo en http://localhost:3000.
- Endpoint /health/db funcionando correctamente.
- Inserción y lectura de datos confirmada.

6. Cómo levantar el proyecto localmente

Levantar base de datos (una vez por sesión):
cd ~/Documents/profesores-app
docker compose up -d

Levantar API:
cd ~/Documents/profesores-app/apps/api
pnpm run start:dev

7. Próximos pasos previstos

- Definir el modelo real del sistema (usuarios, instituciones, roles).
- Implementar autenticación y autorización.
- Construir el banco de preguntas y exámenes.
- Agregar frontend en React.
- Preparar el sistema para deploy en VPS.