**Guía de Instalación y Ejecución del Proyecto RedSocial\_PT**

**Introducción**

Este documento describe cómo instalar, configurar y ejecutar el proyecto RedSocial\_PT, compuesto por:

* Base de datos PostgreSQL
* Backend en Node.js (Express + Prisma)
* Frontend en React (Vite + TailwindCSS)

Todo el proyecto está dockerizado para facilitar su despliegue.

**Requisitos Previos**

* **Docker** y **Docker Compose** instalados.
* **PNPM** instalado globalmente: npm install -g pnpm
* Versiones recomendadas:
  + Node.js: v22.15.1
  + npm: 11.3.0

**Estructura del Proyecto**

RedSocial\_PT/

├── backend/

│ ├── Dockerfile

│ ├── wait-for-it.sh

│ ├── .env

│ ├── .env.test

│ ├── package.json

│ ├── pnpm-lock.yaml

├── frontend/

│ ├── Dockerfile

│ ├── .env

│ ├── package.json

├── docker-compose.yml

**Instalación Manual (para desarrollo)**

**Backend**

1. Eliminar node\_modules y pnpm-lock.yaml si existen.

pnpm install

pnpm install -D ts-node

1. Generar cliente Prisma:

pnpm prisma generate

1. Migrar Base de Datos:

pnpm prisma migrate dev --name init\_schema

1. Resetear Base de Datos (opcional para tests):

pnpm prisma migrate reset

1. Insertar datos de prueba:

pnpm prisma:seed

1. Crear Base de Datos de testing:

docker exec -it nombre\_del\_contenedor\_postgres psql -U postgres

CREATE DATABASE red\_social\_test\_db;

1. Ejecutar servidor:

pnpm dev

1. Inicializar contraseñas de usuarios de prueba (solo en desarrollo):

curl -X POST http://localhost:3000/api/dev/init-passwords

**Frontend**

1. Instalar dependencias:

pnpm install

1. Crear archivo .env en /frontend/:

VITE\_API\_URL=http://localhost:3000/api

1. Ejecutar frontend:

pnpm run dev

**Instalación Completa con Docker**

**Paso 1: Preparar Archivos**

* Verificar que existen los archivos:
  + /backend/wait-for-it.sh (script de espera)
  + /frontend/.env con VITE\_API\_URL=http://backend:3000/api

**Paso 2: Construir y levantar contenedores**

docker compose build --no-cache

docker compose up -d

* **Base de datos**: disponible en database:5432 (internamente) y localhost:5433 (externamente).
* **Backend**: disponible en http://localhost:3000/api
* **Swagger UI**: disponible en http://localhost:3000/api-docs
* **Frontend**: disponible en http://localhost:5173

**Paso 3: Comandos adicionales**

* Apagar contenedores:

docker compose down

* Ver logs en vivo:

docker compose logs -f

* Reconstruir solamente frontend:

docker compose build frontend

**Observaciones Importantes**

* La primera vez, asegúrate de inicializar usuarios de prueba usando:

curl -X POST http://localhost:3000/api/dev/init-passwords

* No exponer el servicio init-passwords en producción.
* Asegurar en entorno real el uso de HTTPS y JWT.
* Variables sensibles deben manejarse con .env y docker secrets si es necesario.

**Autor**

* Proyecto: RedSocial\_PT
* Tecnologías: Node.js, Prisma, PostgreSQL, React, TailwindCSS, Docker.
* Contacto: David Gomez - bojacasa@gmail.com