**Trivia PWA:**

Esta es la documentación paso a paso de todo lo que realizamos hoy para construir una **PWA de trivia** con funcionalidades offline, login/registro, backend en Node.js y frontend en React.

**1. Estructura del Proyecto**

trivia-pwa/

client/ # Frontend en React (PWA)

server/ # Backend en Node.js (API login, registro, preguntas)

**2. Backend en Node.js**

**Archivos principales:**

* server/index.js: punto de entrada con Express
* server/routes/authRoutes.js: rutas de login y registro
* server/controllers/authController.js: lógica para crear usuarios y loguearlos
* server/package.json: configuración con dependencias express y cors

**Comandos:**

cd server

npm install

npm start

El backend corre en http://localhost:3001

**3. Frontend en React**

**Archivos clave:**

* client/src/components/Auth/Login.js: formulario de login
* client/src/components/Auth/Register.js: formulario de registro
* client/src/services/api.js: funciones loginUser y registerUser usando fetch
* client/src/App.js: controla la sesión del usuario
* client/src/App.css: estilos centrados y modernos para login/registro

**Comandos:**

cd client

npm install

npm start

**4. IndexedDB para modo offline**

* Se creó client/src/services/indexedDB.js con funciones:
  + initDB(): inicializa base de datos TriviaDB
  + saveQuestions(): guarda preguntas
  + getQuestions(): obtiene preguntas locales
* Quiz.js carga preguntas desde IndexedDB y sincroniza si hay conexión

**5. Progressive Web App (PWA)**

**Archivos requeridos:**

* public/index.html: estructura base HTML
* public/manifest.json: metadatos de instalación
* src/serviceWorkerRegistration.js: registro del service worker
* src/index.js: activación con serviceWorkerRegistration.register()