Arquitetura Completa do Projeto Angular + Backend Node/Express + Banco

# 1. Estrutura de Pastas

project-root/  
├─ frontend/  
│ ├─ src/  
│ │ ├─ app/  
│ │ │ ├─ core/  
│ │ │ │ ├─ services/ (serviços que fazem a comunicação com o backend e outras lógicas)  
│ │ │ │ │ ├─ client.service.ts # Comunicação com backend (API calls)  
│ │ │ │ │ ├─ media.service.ts # Carrega vídeos, imagens, etc.  
│ │ │ │ │ └─ auth.service.ts # Login, registro, autenticação  
│ │ │ │ ├─ guards/ (protege rotas para usuários logados)  
│ │ │ │ │ └─ auth.guard.ts  
│ │ │ │ ├─ interceptors/ (intercepta requisições HTTP para adicionar token)  
│ │ │ │ │ └─ auth.interceptor.ts  
│ │ │ │ └─ core.module.ts (módulo central de serviços)  
│ │ │ ├─ shared/ (componentes, pipes, diretivas reutilizáveis)  
│ │ │ ├─ features/ (módulos com funcionalidades específicas)  
│ │ │ │ ├─ auth/ (login, registro, etc.)  
│ │ │ │ ├─ main-screen/ (tela principal)  
│ │ │ │ ├─ exercises/ (exercícios, pontuação)  
│ │ │ │ └─ feedback/  
│ │ │ ├─ state/ (gerenciamento global, tipo NgRx ou Signals)  
│ │ │ ├─ app-routing.module.ts  
│ │ │ └─ app.module.ts  
│ │ ├─ assets/  
│ │ ├─ environments/  
│ │ └─ main.ts  
│ ├─ angular.json  
│ ├─ package.json  
│ └─ tsconfig.json  
├─ backend/  
│ ├─ src/  
│ │ ├─ controllers/ (lógica das rotas)  
│ │ │ └─ auth.controller.js  
│ │ ├─ models/ (modelos do banco)  
│ │ │ └─ user.model.js  
│ │ ├─ routes/ (definição das rotas da API)  
│ │ │ └─ auth.routes.js  
│ │ ├─ middlewares/ (funções executadas antes das rotas)  
│ │ │ └─ auth.middleware.js  
│ │ ├─ app.js (configurações gerais do Express)  
│ │ └─ server.js (ponto de entrada do backend)  
│ ├─ package.json  
│ └─ .env (variáveis de ambiente, ex: URL do banco)

# 2. Dependências para o Frontend (Angular)

Execute o comando abaixo na pasta frontend para instalar as dependências necessárias:  
  
npm i @angular/core @angular/common @angular/router rxjs zone.js @angular/forms @auth0/angular-jwt bootstrap @ngx-translate/core @ngx-translate/http-loader

# 3. Dependências para o Backend (Node/Express)

Execute o comando abaixo na pasta backend para instalar as dependências necessárias:  
  
npm i express cors dotenv mongoose jsonwebtoken bcryptjs  
npm i nodemon --save-dev (para facilitar no desenvolvimento)

# 4. O que cada camada faz?

- Frontend Angular: é a interface que o usuário vê e interage. Tem telas, formulários, navegação e se comunica com o backend para buscar e enviar dados.  
  
- Backend Node/Express: é o cérebro da aplicação, processando as requisições que o frontend manda, validando, conectando ao banco de dados e devolvendo respostas.  
  
- Banco de Dados: é onde os dados da aplicação ficam guardados, como usuários, sessões, conteúdo, etc.

# 5. Como rodar o projeto

No terminal, execute os comandos abaixo:  
  
Para o frontend:  
cd frontend  
npm install  
ng serve  
  
Para o backend:  
cd backend  
npm install  
nodemon src/server.js

# 6. Dicas finais

- Se não sabe nada de Angular, comece entendendo o básico: componentes, módulos, serviços e rotas.  
  
- No backend, entenda os conceitos de API REST, rotas, middleware e banco de dados.  
  
- A arquitetura ajuda a organizar seu código para facilitar manutenção e escalabilidade.  
  
- Sempre teste as rotas do backend usando ferramentas como Postman.  
  
- No frontend, use os guards e interceptors para segurança, garantindo que só usuários logados acessem certas páginas.