

Dicas para o Teste - Desenvolvimento Fullstack com Next.js e NestJS

Organizacao do Projeto

- Crie dois projetos separados:
 - /backend -> NestJS
 - /frontend -> Next.js
- Utilize monorepo (com Nx ou Turborepo) somente se estiver confortável, senão é melhor manter separado para não perder tempo.

Planejamento Inicial

- Liste as tarefas antes de começar:
 - Configuração dos projetos (NestJS, Next.js)
 - Modelagem do banco (User)
 - Implementação dos endpoints no backend
 - Integração do Redis
 - Desenvolvimento das telas no frontend
 - Integração frontend-backend com React Query
 - Implementação dos testes
 - Montagem do README com instruções

Backend (NestJS)

- Escolha Prisma (mais produtivo e moderno) para trabalhar com PostgreSQL ou MySQL.
- Crie um módulo dedicado para Redis (CacheModule do NestJS pode ser usado de forma nativa).
- Utilize DTOs e validações com class-validator e class-transformer.
- Estruture com separação clara:
 - Modulos -> /users, /database, /cache
 - Services, Controllers, DTOs e Entities
- Comece com testes unitários simples (principalmente no UserService) usando Jest.

Frontend (Next.js)

- Crie páginas básicas:
 - /users -> Lista de usuários (com opções de editar e deletar)
 - /users/create e /users/edit/[id] -> Cadastro e edição
- Utilize React Query para:
 - Fetch de listagem e detalhes

- Mutations para create, update e delete
- Invalidação de cache local após mutations
- Não se preocupe com UI/UX avançado. Pode usar componentes simples do próprio HTML ou bibliotecas como Chakra UI, ShadCN ou MUI se quiser rapidez, mas não é obrigatório.

Redis e Cache

- Foque em cache para consultas pesadas, especialmente:
 - GET /users -> Liste todos os usuários
 - GET /users/:id -> Buscar usuário específico
- Na criação, atualização ou deleção, lembre-se de invalidar/atualizar o cache para manter os dados consistentes.

Documentacao (README)

- Inclua:
 - Passo a passo para rodar backend e frontend
 - Comandos de npm install, npx prisma migrate dev, etc.
 - Informações sobre variáveis de ambiente (.env.example)
 - Como rodar os testes (npm test)

Bonus (Se Sobrar Tempo)

- Configuração de variáveis com dotenv
- Logs básicos com console.log ou, melhor, com Winston (se desejar)
- Autenticação com JWT é opcional, mas se tiver domínio é um bom diferencial

Dicas Finais

- Foque primeiro no funcional mínimo (CRUD funcionando + cache + React Query).
- Só depois pense nos bônus (auth, logs, etc.).
- Priorize código limpo, commit organizado e documentação clara.
- Se estiver apertado, implemente testes no backend, que é onde mais contam.