

Descrição do Case

Contexto: Um escritório de investimentos precisa de uma aplicação para gerenciar clientes e visualizar informações básicas de ativos financeiros. A aplicação será containerizada com Docker, com uma instância de banco de dados para persistir as informações de clientes.

A aplicação deverá incluir:

1. **Cadastro e listagem de clientes** (nome, email e status - ativo/inativo).
2. **Cadastro e exibição de ativos financeiros** (nome do ativo e valor atual) por cliente.
 - a. Deve ser possível visualizar as alocações para cada cliente.
3. Implementação **100% em TypeScript!**

Requisitos Técnicos

1. Backend (Node.js com Fastify + Prisma)

- **CRUD de Clientes:** Criar rotas para cadastrar, listar e editar clientes, usando Fastify e persistindo os dados no banco MySQL via Prisma.
- **Listar Ativos:** Endpoint que retorna uma lista fixa de ativos financeiros (por exemplo, “Ação XYZ”, “Fundo ABC”) com valores estáticos.
- **Ferramentas e bibliotecas:**
 - **Prisma ORM:** Configurar o Prisma para interagir com o MySQL de maneira simples.
 - **Banco de Dados:** Banco MySQL rodando em um contêiner, configurado via Docker Compose.
 - **Zod:** Usar a biblioteca para validação do payload das requisições.

2. Frontend (Next.js)

- **Página de Clientes:** Página simples para listar, adicionar e editar clientes.
- **Página de Ativos:** Exibe a lista fixa de ativos (somente leitura).
- **UI:** Não precisa de design avançado, mas o uso de ShadCN é recomendado para criar uma interface funcional, reutilizável e personalizável.
- **Ferramentas e bibliotecas:**

- **ShadCN:** Usar ferramenta para criar componentes padrão de UI para uniformizar a estilização do site (como Button, Text ...).
- **React Query:** Usar biblioteca para fazer BUSCAS no backend.
- **React Hook Form + Zod:** Usar bibliotecas para ENVIAR E VALIDAR FORMULÁRIOS ao backend.
- **Axios:** Fazer requisições para o backend.

3. Docker Compose

- Configurar o `docker-compose.yml` para rodar:
 - **Serviço db:** Instância de MySQL (ou MariaDB) para armazenar os dados dos clientes.
 - **Serviço backend:** Servidor Fastify configurado para se conectar ao banco de dados e rodar as rotas.

Estrutura do Docker Compose

1. Serviço db usando a imagem Postgres(ou MySQL) com senha segura.
2. Configurar variáveis de ambiente (host, usuário, senha, database) no backend para o Prisma.

Instruções de Implementação

Entregáveis:

1. **Repositórios:** Códigos-fonte do backend e frontend,
2. **Configuração Prisma:** Script de migração inicial para criar a tabela de clientes no banco.

Orientação de Tempo

Prazo: 1 semana (06/06).

Esse prazo cobre a configuração do Docker Compose, o CRUD básico com Prisma e uma interface Next.js simplificada com ShadCN. Não se preocupe em deixar a tela bonita.

Outra dica de ouro: abuse do ChatGPT.