

Desafio Técnico – Desenvolvedor(a) Full-Stack

Objetivo

Desenvolver um **Mini-CRM** com funcionalidades básicas de cadastro, visualização e acompanhamento de clientes e interações comerciais.

Stack obrigatória

- **Frontend:** React.js (typescript)
 - **Backend:** Node.js + Express (typescript)
 - **Banco de Dados:** MySQL
 - **Containerização:** Docker + docker-compose
-

Funcionalidades:

1. Autenticação

- Login com usuário e senha
- Proteção de rotas com autenticação JWT

2. Gestão de Clientes

- Criar, listar, editar e deletar oportunidades / clientes
- Campos obrigatórios: nome, email, telefone, empresa, cargo
- Acompanhamento do funil (coluna)

3. Gestão de Interações

- Criar, listar, editar e deletar interações associadas a clientes
- Campos obrigatórios: tipo (reunião, ligação, e-mail, outro), data, observações

Exemplo de interface:

The screenshot shows the Pipedrive CRM interface for the 'Sales' module. At the top, there's a search bar with the placeholder 'Search Pipedrive'. To the right of the search bar are navigation buttons for 'Pipeline', 'Everyone', and a user profile icon. On the far left is a vertical sidebar with icons for Sales, Opportunities, Contacts, and Tasks, along with a '+' button to add new items.

The main area displays a grid of deals categorized by stage:

- Lead In**: \$56,750 - 7 deals
- Contact Made**: \$72,500 - 5 deals
- Needs Defined**: \$111,500 - 5 deals
- Proposal Made**: \$71,000 - 4 deals
- Negotiations Started**: \$54,500 - 2 deals

Each deal card includes the deal name, company name, amount, and a small circular icon indicating its status or priority level (e.g., red, green, yellow).

Entregáveis:

- Vídeo da utilização da interface funcional
- Código fonte (.zip ou git)
- Print de todas as telas do sistema
- Documentação do banco de dados (schemas, tabelas e colunas)
- READ-ME para executar e testar

Estrutura de pastas sugerida

- mini-crm/
- └── backend/
- | └── src/
- | | └── controllers/
- | | └── routes/
- | | └── services/
- | | └── middlewares/
- | | └── index.js
- | └── Dockerfile
- └── package.json
-
- └── frontend/
- | └── public/
- | └── src/
- | | └── components/
- | | └── pages/
- | | └── App.jsx
- | └── Dockerfile
- └── package.json
-
- └── docker-compose.yml
- └── README.md

Exemplo de `docker-compose.yml`

```
version: "3.9"

services:
  mysql:
    image: mysql:8.0
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: crm
    ports:
      - "3306:3306"
    volumes:
      - mysql_data:/var/lib/mysql

  backend:
    build: ./backend
    environment:
      DB_HOST: mysql
      DB_USER: root
      DB_PASSWORD: root
      DB_NAME: crm
    ports:
      - "3001:3001"
    depends_on:
      - mysql

  frontend:
    build: ./frontend
    ports:
      - "3000:3000"
    depends_on:
      - backend

volumes:
  mysql_data:
```