

Teste técnico sênior que pode ser realizado em até 2-3 dias:

Desafio Técnico Sênior – Aplicação Fullstack de Gestão de Pokémons

Objetivo

Desenvolver uma aplicação fullstack simples que permita gerenciar um conjunto de Pokémons, integrando dados oficiais da API POKEDEX (<https://www.pokemon.com/br/pokedex>). O candidato deverá construir tanto o backend quanto um frontend básico, entregando um produto mínimo viável que demonstre as principais habilidades e boas práticas.

Observação: O desafio foi pensado para ser concluído em até 2-3 dias. Foque em entregar um MVP (Produto Mínimo Viável) com uma arquitetura bem definida e código limpo. Funcionalidades extras (testes, containerização, etc.) são diferenciais, mas não obrigatórios.

<https://pokeapi.co/docs/v2>

<https://www.pokemon.com/br/pokedex>

Parte 1: Backend

Escopo e Funcionalidades

- **CRUD de Pokémons:**
 - **Criar:** Endpoint **POST** para cadastrar um novo Pokémon.
 - **Listar:** Endpoint **GET** para retornar todos os Pokémons cadastrados.
 - **Consultar:** Endpoint **GET** para buscar um Pokémon específico por ID.
 - **Atualizar:** Endpoint **PUT** ou **PATCH** para atualizar informações de um Pokémon.
 - **Deletar:** Endpoint **DELETE** para remover um Pokémon.
- **Integração com a API POKEDEX:**
 - Durante o cadastro (ou atualização), busque informações da POKEDEX para enriquecer ou validar os dados (ex.: nome, tipo, habilidades).

- Caso a integração seja mais complexa, é aceitável implementar uma versão simplificada (ex.: apenas validar o nome ou buscar um campo extra).

Requisitos Técnicos

- **Linguagem:** TypeScript.
 - **Framework:** Express, NestJS ou outro de sua preferência.
 - **Persistência:** Utilize um banco de dados simples (pode ser SQLite, MongoDB ou até armazenamento em memória, se necessário).
 - **Estrutura do Código:** Organize o código em camadas (rotas, controllers, serviços, modelos).
 - **Tratamento de Erros:** Retorne mensagens e códigos HTTP apropriados para cada situação.
 - **Documentação:** Um README com instruções claras de instalação e execução.
 - **Diferenciais:**
 - Implementação de testes unitários ou de integração.
 - Uso de Swagger/OpenAPI para documentar a API.
 - Containerização (Docker) para facilitar a execução.
-

Parte 2: Frontend

Escopo e Funcionalidades

- **Interface Básica:** Desenvolva uma aplicação web simples para:
 - **Listar:** Exibir os Pokémons cadastrados.
 - **Detalhar:** Visualizar detalhes de um Pokémon selecionado.
 - **Operações CRUD:** Permitir a criação, edição e remoção de Pokémons através de formulários simples.
- **Integração:** Conecte a interface com os endpoints da API REST criada.
- **Experiência de Usuário:**
 - Interface limpa e responsiva.
 - Feedback visual para operações de sucesso e erro.

Requisitos Técnicos

- **Framework/Linguagem:** React, Angular ou Vue (à escolha do candidato).
 - **Organização:** Estructure a aplicação em componentes/modularmente.
 - **Diferenciais:**
 - Testes unitários para os componentes.
 - Uso de um gerenciador de estado (como Redux, Context API ou similar).
-

Critérios de Avaliação

- **Arquitetura e Design:**
 - Organização modular do código (separação de responsabilidades).
 - Adoção de boas práticas e padrões de projeto.
 - **Integração e Funcionalidades:**
 - Funcionamento correto do CRUD e integração com a API POKEDEX.
 - Consistência na comunicação entre backend e frontend.
 - **Qualidade do Código:**
 - Legibilidade, manutenção e clareza do código.
 - Implementação de tratamento de erros e validações.
 - **Documentação:**
 - README claro com instruções para rodar a aplicação.
 - Documentação dos endpoints (ex.: Swagger) e, se aplicável, dos componentes.
 - **Entrega:**
 - Controle de versão adequado (commits descritivos, repositório organizado).
-

Entrega

- **Repositório Git:** Disponibilize o link do repositório (GitHub, GitLab, etc.) contendo o código completo.
 - **Documentação:** Inclua um README com:
 - Instruções para instalação e execução do backend e frontend.
 - Explicação das principais decisões de arquitetura e possíveis melhorias futuras.
-

Boa sorte! Este desafio visa avaliar suas habilidades de desenvolvimento fullstack, capacidade de integração com APIs externas e aplicação de boas práticas de desenvolvimento.