

TESTE PRÁTICO: Desenvolvedor Frontend

CRUD COM ANGULAR 17+

Descrição do Projeto:

Você deverá criar uma aplicação Angular para consumir e manipular os dados da API pública escolhida por você, a versão do angular deve ser superior a versão 17 e o design system ou biblioteca de componentes fica a sua escolha. A aplicação deverá permitir a listagem, visualização, criação, edição e exclusão.

Você poderá escolher qualquer uma das API's abaixo para realização do teste:

<https://rickandmortyapi.com/>

<https://thedogapi.com/>

<https://developer.marvel.com/>

Requisitos Técnicos

1. **Versão do Angular:** 17+
2. **State Management:** Uso de **Signals** para gerenciamento de estado.
3. **Testes Unitários:** Implementação de testes unitários utilizando **Jest**.
4. **HTML e CSS:** Desenvolver um template amigável e funcional. Não é necessário criar um design elaborado, mas a estrutura deve ser clara e organizada.
5. **Organização do Projeto:** Divisão clara entre componentes, serviços, e outros módulos.
6. **Recursos Avançados do Angular:** Espera-se que utilizem conceitos como:
 - Lazy loading de módulos ou lazy component com standalone;
 - Componentização bem definida;

- Uso de diretivas ou pipes personalizados, se necessário.
 - Uso de formulários reativos, se necessário.
-

Funcionalidades Obrigatórias

1. Listagem:

- Exibir uma tabela ou grid com informações básicas dos itens (ex.: nome, descrição e thumbnail).
- Paginação ou scroll infinito.

2. Busca:

- Campo de busca para filtrar itens pelo nome ou outra propriedade.

3. Visualização Detalhada:

- Ao clicar em um item, exibir detalhes em um modal ou página dedicada.

4. Criação de Novo Item:

- Formulário para criação.
- Simular a criação localmente se a API não permitir o POST.

5. Edição de Item:

- Permitir a edição das informações de um item da lista.
- Atualizar os dados localmente se a API não permitir o PUT.

6. Exclusão de Item:

- Implementar funcionalidade de exclusão com confirmação.
-

Critérios de Avaliação

1. Organização e Clareza do Código:

- Estrutura clara e bem modularizada (pasta de serviços, modelos, etc.).
- Nomenclatura adequada de variáveis, funções e componentes.

2. Uso de Recursos do Angular:

- Implementação de Signals para gerenciamento de estado.
- Uso de RxJS (se necessário), diretivas, pipes, formulários reativos.

3. Testes Unitários:

- Cobertura mínima: 80%.

- Criar testes para os serviços, componentes principais e lógica de negócio.
 - 4. Componentização:**
 - Reutilização de componentes sempre que possível.
 - 5. Performance e Escalabilidade:**
 - Boa utilização de lazy loading ou lazy component (se for standalone) e otimização da lógica.
 - 6. Interface de Usuário:**
 - Aplicação deve ser funcional e intuitiva, mesmo que o design não seja o foco principal.
 - 7. Padrões de Projeto e Arquitetura:**
 - Organização de pastas, de código, padrões de código adotados, padrões de projeto, separação de estruturas, escalabilidade, performance.
-

Entrega

- Você deverá entregar o código em um repositório no GitHub (ou plataforma similar), com as instruções de instalação e execução no arquivo README.md.
-

Dicas

- Leia a documentação da API escolhida com atenção.
- Utilize boas práticas de programação.
- Priorize a clareza e escalabilidade do código.

Boa sorte! =)

