

Teste Técnico:

Objetivo: O objetivo deste teste é avaliar as habilidades do candidato em desenvolvimento de APIs RESTful utilizando Node.js e boas práticas de desenvolvimento. O sistema deverá ser construído de forma simples e eficiente, com foco na qualidade do código e na organização.

Requisitos do Sistema:

1. Tecnologia:

- Utilizar a última versão estável do **Node.js**.
- O sistema deve ser uma **API RESTful**.
- Encorajamos a utilizar um **framework** de desenvolvimento (por exemplo, **Express.js**, **NestJS** ou similar).

2. Funcionalidades Principais:

- **Cadastro de usuários:** Implementar um endpoint para o cadastro de novos usuários com validação de dados.
- **Autenticação de usuários:** Implementar um sistema de autenticação utilizando **JWT (JSON Web Token)** ou outra solução adequada para garantir a segurança na comunicação.
- **Consumo de API externa:** O sistema deve consumir uma API externa (rickandmortyapi - > <https://rickandmortyapi.com/documentation/>) e exibir ou armazenar as informações de forma relevante.
- Usuário autenticado pode fazer 10 consultas de informações e não autenticados 3 consultas .
- Usuário autenticados podem
 - Favoritar até 3 personagens;
 - Listar os personagens favoritos
 - Atualizar personagem favorito
 - Remover favorito
 - Saber quantos episódios cada personagem favorito aparece
 - Saber quantos episódios todos os favoritos aparecem. Obs se 2 favoritos aparecem no episódio 2, conta apenas 1 episódio.
- **Tratamento de erros:** Implementar um sistema de tratamento de erros, retornando códigos de erro HTTP apropriados.

3. Qualidade do Código:

- O código deve ser modular e organizado, com separação clara entre camadas de serviço, controle de rotas e dados.

4. README:

- O candidato deve fornecer um **README.md** explicando como rodar o sistema em ambiente de desenvolvimento.
- Instruções para **iniciar o projeto**, incluindo dependências, variáveis de ambiente necessárias e comandos para rodar o sistema.

Critérios de Avaliação:

- **Funcionalidade:** A API funciona como esperado e atende aos requisitos.

- **Código:** O código é limpo, modular e bem estruturado.
- **Boas Práticas:** Uso de boas práticas no desenvolvimento da API, tratamento de erros e segurança.
- **Documentação:** O README está claro e completo, com instruções claras para rodar o sistema.

Fique a vontade para adicionar mais funcionalidades.