

Desafio técnico backend - SmartMEI

Construir um serviço para empréstimo de livros. Usuários podem se cadastrar, cadastrar seus livros e emprestar seus livros para outros usuários. De maneira geral são 5 features básicas:

1. cadastrar um usuário (createUser)
2. cadastrar livro para um usuário (addBookToMyCollection)
3. emprestar meu livro para outro usuário (lendBook)
4. devolver o livro (returnBook)
5. pegar detalhes de um usuário (user)

Regras de negócio

- Em alguns lugares você vai encontrar o input `loggedUserId` isso representa o ID do usuário logado, por simplificação do case, para não ter que implementar autenticação, guardar senhas, gerar e parsear tokens nas requisições, optamos por deixar parametrizável. **Dito isso, todas as validações e verificações no serviço que julgar necessária devem ser feitas em relação ao usuário logado**, faz parte da avaliação, nem todas estão explicitamente descritas aqui.
- 1. cadastrar um usuário (createUser)
 - email deve ser único no sistema
- 3. emprestar meu livro para outro usuário (lendBook)
 - Apenas um empréstimo por livro de um usuário por vez deve ser permitido, ou seja, enquanto não for devolvido o livro emprestado este não poderá ser emprestado
- 4. devolver o livro (returnBook)
 - O livro só pode ser devolvido uma vez. Ao tentar devolver um livro já devolvido, um erro deve ser retornado
- 5. pegar detalhes de um usuário (user)
 - Não há necessidade de guardar histórico de empréstimos. Mas dependendo da estrutura de dados escolhida, já pode estar implementada. Ou seja, por simplificação, você pode retornar apenas 1 empréstimo por livro ou o histórico inteiro. Apenas não considerar o histórico de empréstimos uma obrigatoriedade.

Requisitos técnicos obrigatórios

- Deve estar disponível GitHub ou bitbucket. (Você pode usar um repo privado mas não esqueça de dar permissão).
- Deve ser feito com node/typescript ou go.
- Precisa ter testes
- Deve ter alguma base de dados, local ou remota. Apenas deve estar claro na documentação onde está e como fazer o setup inicial

- Deve ter uma documentação simples para execução da aplicação passo-a-passo e como executar os testes implementados no requisito anterior
- Utilizar REST(<https://documenter.getpostman.com/view/7660896/SzzrZaU2>) ou GraphQL (em anexo) seguindo estritamente o arquivo graphql ou swagger em anexo. Estes automáticos vão ser rodados para garantir a corretude da solução.

Requisitos técnicos recomendados

- Mais de um tipo de teste
- Fazer o uso de commits
- documentar o código para ajudar na nossa revisão

Requisitos técnicos opcionais

- Validação mais rigorosa e criteriosa dos inputs
- Uso de docker
- Ter um pipeline de CI/CD configurado
- Se escolher uma estrutura SQL fazer os scripts de migração automáticos.