



Teste prático para desenvolvedores Back-end.

Este teste busca identificar diferentes níveis de habilidade, medindo conhecimento, organização e esforço. Para isso ele é dividido em 3 níveis:

- O marcador **(*)**: é obrigatório para júnior e acima;
- O marcador **(*)**: é obrigatório para pleno e acima;
- O marcador **(*)**: é obrigatório para sênior;

1. **(*)**: Elabore um documento *Readme.md* no projeto que tenha os seguintes detalhes:
 - a. Descreva a stack utilizada, como versão da linguagem utilizada, framework e o porquê de ter escolhido, pacotes adicionados, banco de dados, etc....
 - b. Descreva como subir o sistema para executar os testes.
2. **(*)**: A estrutura base do sistema vai contar com 3 entidades:

User:

id: id primary_key
name: string:100
email: string:191

Post:

id: id primary_key
user_id: id foreign:users
title: string:100
description: text

Comment:

id: id primary_key
user_id: id foreign:users
post_id: id foreign:posts
description: text

essas são as exigências mínimas para essa atividade.

3. **(*)**: Crie um sistema de autenticação por token para chamadas de REST API.
4. **(*)**: Com o usuário você pode fazer dois processos de CRUD:
 - a. CRUD simples;
 - i. Colocar checagem de permissão se possuir autenticação;
 - b. Ou se possuir autenticação;
 - i. Registro na fase de autenticação com edição de perfil do usuário logado.
5. **(*)**: Com as postagens é preciso fazer um CRUD simples com algumas exigências:
 - a. Apenas o próprio usuário pode editar ou excluir as postagens;

- b. (*): a postagem tenha a possibilidade de adicionar uma imagem em uma api dedicada a isso;
 - c. (*): as edições sejam salvas como um histórico;
 - d. (*): a postagem tenha um contador de visualizações;
 - e. (*): a postagem tenha um contador de curtidas e não curtidas;
6. (*): Com os comentários é preciso fazer um CRUD simples com algumas exigências:
- a. Apenas o próprio usuário pode editar os comentários;
 - b. Usuário do comentário pode remover o comentário;
 - c. (*): Usuário da postagem também pode remover o comentário;
 - d. (*): adicionar marcador que foi removida pelo usuário ou dono da postagem;
 - e. (*): Mandar um e-mail para o usuário da postagem que ele possui um novo comentário em seu post;
7. (*): Crie uma rota que gere um relatório que traga os posts com os seguintes campos:
- a. Título;
 - b. Quantos comentários eles possuem;
 - c. (*): Quantas visualizações;
 - d. (*): Quantas curtidas;
 - e. (*): Quantas não curtidas;
8. (*): Documentar as chamadas da API com uma ferramenta de requisição, exemplo: Postman e enviar o arquivo de configuração;