

Desafio técnico – nível pleno

Sobre o desafio

Tecnologias

O desafio deverá ser realizado utilizando a linguagem node ou python.

Nesse desafio, você deverá criar uma aplicação para nos mostrar suas skills e se elas estão de acordo com o dia-a-dia do nosso time de desenvolvimento!

Essa será uma aplicação para gerenciar projetos. Será permitida a criação de um usuário com name, password e username, bem como fazer o CRUD de *projects*:

Criar um novo *project*;

- Listar todos os *projects*;
- Alterar o title e deadline de um *projeto* existente;
- Marcar um *projeto* como feito;
- Excluir um *projeto*;

Tudo isso para cada usuário em específico (o username será passado pelo header). A seguir veremos com mais detalhes o que e como precisa ser feito.

Backend – Banco de dados e Rotas da aplicação

O banco de dados deve ser modelado em PostgreSQL, conforme você notar os campos abaixo descritos nas rotas da aplicação.

Seguem as rotas desejadas para o desafio proposto.

POST /users

A rota deve receber name, e username dentro do corpo da requisição. Ao cadastrar um novo usuário, ele deve ser armazenado dentro de um objeto no seguinte formato:

```
{
  id: 'uuid',
  name: 'João Silva',
  password: '*****',
  username: 'joao.silva',
}
```

Certifique-se de que a rota não permita usuários duplicados.

POST /project

A rota deve receber `title`, `zip_code`, `deadline` e `cost` dentro do corpo da requisição e, uma propriedade `username` contendo o `username` do usuário dentro do header da requisição. Ao criar um novo *project*, ele deve ser armazenado dentro do banco referenciado ao usuário que está criando esse projeto. Cada projeto deverá estar no seguinte formato:

```
{
  id: 'uuid'
  title: 'Nome do projeto',
  zip_code: 88010400,
  cost: 9500
  done: false,
  deadline: '2022-09-31T00:00:00.000Z',
  username: 'joao.silva',
  created_at: '2022-09-26T00:00:00.000Z'
  updated_at: '2022-09-26T00:00:00.000Z'
}
```

GET /projects

A rota deve receber, pelo header da requisição, uma propriedade `username` contendo o `username` do usuário e retornar uma lista com todos os projetos desse usuário.

GET /project

A rota deve receber o `id` do projeto, e retornar as informações do mesmo, e ao invés de mostrar o número do CEP (`zip_code`) deverá exibir a localização (Cidade/UF) onde o projeto será executado, sendo assim deverá ser realizada uma chamada API de onde essa informação possa ser obtida. Pode ser usada a seguinte documentação (ou alguma outra de sua preferência):

PUT /projects/:id

A rota deve receber, pelo header da requisição, uma propriedade `username` contendo o `username` do usuário e receber as propriedades `title`, `zip_code`, `cost` e `deadline`, dentro do corpo. É preciso validar que cada usuário só consiga obviamente editar seus próprios projetos.

PATCH /projects/:id/done

A rota deve receber, pelo header da requisição, uma propriedade username contendo o username do usuário e alterar a propriedade done para true no *project* marcando assim o projeto como concluído.

DELETE /projects/:id

A rota deve receber, pelo header da requisição, uma propriedade username contendo o username do usuário e excluir o *projeto*, observando que cada usuário só pode excluir obviamente seus próprios projetos.

Frontend

Para o frontend deve-se criar uma tela onde seja possível criar usuários e outra que possa ser feito a gestão dos projetos (CRUD) e marca-los com concluído. Todas as funcionalidades só deverão estar acessíveis a usuários logados.



Entrega

Esse desafio deve ser entregue a partir do Github. Você deve criar um repositório e marcá-lo como público para que possamos clonar o mesmo, e testarmos sua aplicação.