

## Desenvolvimento de Aplicativo CRUD em React.js

**i** Desenvolver um sistema CRUD de produtos em React.js, e o GitHub ou Bitbucket para hospedar o código.

### Setup inicial

1. Crie um novo projeto React.js.
2. Configure rotas usando **react-router-dom**.
3. Use **axios** ou **fetch** para chamar as respectivas apis.
  - a. Link para documentação na api <https://documenter.getpostman.com/view/12499698/2sA3BkbsqD>
  - b. Usar seu login e senha enviado junto com o teste.

### **1** Criação de Produto

1. Crie uma página/formulário para criar novos produtos.
2. Implemente campos para inserir o nome, descrição e preço do produto.
3. Valide os campos da solicitação, por exemplo:
  - a. **name** é obrigatório.
  - b. **price** deve ser numérico.
  - c. **status** é um enumerado de:
    - i. em estoque;
    - ii. em reposição;
    - iii. em falta.
  - d. **stock\_quantity** deve ser numérico.
4. Envie uma solicitação para a API para criar um novo produto quando o formulário for enviado.

```
1 Método API
2 POST http://34.71.240.100/api/product/create
```

### **2** Listagem de Produtos

1. Crie uma página para exibir todos os produtos cadastrados.
2. Faça uma solicitação para a API de listagem de produtos.
3. Exiba os produtos em uma lista, mostrando nome, descrição, preço, status e quantidade estoque.

```
1 Método API
```

```
2 POST http://34.71.240.100/api/product/list
```

### 3 Atualização de Produto

1. Permita a edição de um produto existente.
2. Implemente uma página/formulário para editar os campos do produto.
3. Pré-preencha os campos do formulário com as informações do produto existente.
4. Valide os campos editados da mesma forma que na criação.
5. Envie uma solicitação para a API para atualizar o produto quando o formulário for enviado.

```
1 Método API
2 PUT http://34.71.240.100/api/product/update
```

### 4 Exclusão de Produto

1. Permita a exclusão de um produto existente.
2. Implemente uma funcionalidade para excluir o produto quando solicitado pelo usuário.
3. Confirme com o usuário antes de excluir o produto.

```
1 Método API
2 DELETE http://34.71.240.100/api/product/delete
```

### 5 Testes

1. **Implemente testes** para os componentes principais usando uma biblioteca como **Jest** ou **React Testing Library**.

### 6 Estilo e Design

1. Use CSS na biblioteca como **Bootstrap** para estilizar o aplicativo e torná-lo visualmente **atraente e responsivo**.
2. Mantenha uma **boa organização** no código CSS e evite estilos inline sempre que possível.

### 7 Publicação no GitHub


1. Publique o código-fonte do projeto no GitHub ou Bitbucket.
2. Inclua um **README.md** com instruções claras sobre como configurar e executar o projeto.
3. Adicione uma breve descrição do projeto e dos recursos utilizados.

### 8 Documentação

1. **Documente** os componentes principais, explicando suas responsabilidades e propriedades.
2. Adicione comentários **relevantes** no código para explicar partes complexas ou importantes.

## Critérios de Avaliação

- Funcionalidade **correta** do aplicativo CRUD.
- **Boa prática** de desenvolvimento React.js.
- **Qualidade do código** (legibilidade, **organização**, comentários quando necessário).
- Utilização eficaz de **componentes e estados** do React.
- Interface do usuário **atraente e responsiva**.
- **Documentação** clara e completa.
- Será um diferencial utilizar **SCSS**.

 O **prazo** para o envio do projeto é até dia **16/05/2024 às 23:59**. Enviar o teste para [adriel.oliveira@bevioficial.com.br](mailto:adriel.oliveira@bevioficial.com.br) ou para **18 99793-1278 (whatsapp)**