**Nome:** Poliane Haito de Oliveira

**Disciplina:** Programação BackEnd

**Atividade:**

Realize a construção de uma API restful, contendo rotas de requisição GET para listagem de clientes, POST para o cadastro de clientes, PUT para alteração de clientes, DELETE para a remoção de clientes.

A implementação pode ser realizada via código por programação utilizando algum tipo de linguagem como NODEJS, PHP, Java, C# ou qualquer outra linguagem, como também pode ser utilizando algum serviço de APIS como Mongodb API ou firebase do Google.

**Links de referência**

[**https://www.mongodb.com/features/mongodb-rest-api**](https://www.mongodb.com/features/mongodb-rest-api)

[**https://firebase.google.com/docs/reference/rest/database**](https://firebase.google.com/docs/reference/rest/database)

O que deve ser entregue como resposta é um endereço de um servidor como <https://servidor.com.br/clientes>  capaz de executar as operações básicas de CRUD (create, retrieve, update e delete)

Ao requisitamos as rotas, as mesmas devem entregar as seguintes informações:

**POST**

Na requisição http na rota de cadastro  deve permitir enviar dados do registro de uma cliente

| **{ "nome": "João", "sobreNome": "da Silva" }** |
| --- |

Na resposta http não exibe nada apenas status 200

| **status 200** |
| --- |

**PUT**

Na requisição http na rota de alteração, um ID deve ser informado para realizar a atualização nos campos..

Ex: neste caso estamos mudando o sobrenome do cliente para  registro de \_id igual "123456789"

| **{ \_id: "123456789",  "sobreNome": "de Souza" }** |
| --- |

A resposta http deve exibir os dados do registro alterados.

| **{ \_id: "123456789",   "nome": "João", "sobreNome": "de Souza" }** |
| --- |

**DELETE**

Na requisição http deve permitir passar o ID do registro a ser removido

| **{ \_id: "123456789" }** |
| --- |

Na resposta http, nada precisa ser carregado no corpo da resposta apenas o status 200

|  |
| --- |

**GET**

Na requisição na rota de consulta, a saída deve ser uma lista de todos os registros cadastrados. Exemplo:

| **[ { \_id: "123456789",   "nome": "João", "sobreNome": "de Souza" } , { \_id: "123456710",   "nome": "Zé", "sobreNome": "de Souza" } , { \_id: "123456711",   "nome": "Maria", "sobreNome": "de Souza" } ]** |
| --- |

**Resposta:**

Para implementar a API utilizei a linguagem de programação **NODEJS** e o serviço de API’s **Mongodb**.

O *link* público da **api-clientes** no Render é: **https://api-clientes-70xi.onrender.com**