**Projeto Prático**

Esse projeto prático possui como finalidade consolidar os conhecimentos adquiridos com o curso até este momento.

No final deste projeto você terá uma api de listagem e inclusão de ***clientes.***

**1** . Criar uma nova api com o nome de **clientes-api**, baseada na estrutura vista até então: controllers, model e routes. Essa api deverá ser capaz de se comunicar com o banco de dados Mongo.

* Após criar a estrutura do seu projeto, o ideal seria criar a conexão com o MongoDB através do uso do framework **mongoose (npm install mongoose).**
* Será importante criar a string de conexão com o banco de dados, cujo nome será **clientes.**

Após se certificar de que existe conexão com o mongo, será necessário estruturar o seu objeto que será armazenado no banco, o **clientes (ClientesSchema)**. O seu schema de cliente deverá conter as seguintes propriedades:

*nome (String)*

*email: (String)*

*cpf: (Number)*

*dataNascimento: (Date)*

*estadoCivil: (String)*

*telefone: (Number)*

*comprou: (Boolean)*

**2**. Você deverá criar um repositório no github e como sugestão deixamos o nome: reprograma-projeto-integracao-mongo-1.

**3**. Recomendamos criar uma coleção do postman para armazenar as suas requisições.

**4**. O Seu projeto deverá conter as seguintes rotas/endpoints com os respectivos retornos:

1. POST/clientes

Incluir um novo cliente conforme as característisticas informadas no contrato do model. (HTTP 201 OK)

Resposta:

{

    "status": true,

    "mensagem": "Aluna incluida com sucesso"

}

1. GET/clientes

Retornar a lista com todos os clientes (HTTP 200 OK)

Resposta:

[

    {

        "\_id": "5dcd81f42e2aab923cbd653d",

        "email": "marilia.oliveira@outlook.com",

        "nome": "Marilia Messias",

        "cpf": 1234567890,

        "dataNascimento": "1992-04-03T03:00:00.000Z",

        "estadoCivil": "Solteira",

        "telefone": 123456789,

        "comprou": true

    }

]

1. GET/clientes/compradores

Retorna a lista com os clientes que já realizaram alguma compra (HTTP 200 OK)

Resposta:

[

    {

        "nome": "Marilia Messias",

        "email": "marilia.oliveira@outlook.com"

    },

    {

        "nome": "Joaquina Messias",

        "email": "marilia.oliveira@outlook.com"

    },

    {

        "nome": "Ana Messias",

        "email": "marilia.oliveira@outlook.com"

    },

    {

        "nome": "Camila Messias",

        "email": "marilia.oliveira@outlook.com"

    }

]

1. GET/clientes/{cpf}

Retornar a lista de clientes de acordo com o elemento informado como parâmetro (HTTP 200 OK)

Resposta:

[

    {

        "\_id": "5dcd82ad5777b7b184ae38e3",

        "email": "marilia.oliveira@outlook.com",

        "nome": "Joaquina Messias",

        "cpf": 123456789,

        "dataNascimento": "1992-04-03T03:00:00.000Z",

        "estadoCivil": "Solteira",

        "telefone": 123456789,

        "comprou": true

    }

]