1. **Desafio BackEnd**

**User Stories:**Sendo uma pessoa que deseja ser cliente, o mesmo deve efetuar o seu cadastro. Depois de realizado o cadastro o sistema deve enviar um e-mail para o cliente para que o cadastro possa ser ativado.  
Ao clicar no link de ativação o sistema deve registrar a ativação do usuário.  
Depois de ativado o cadastro o sistema deve permitir que o cliente efetue o login.

Sendo um cliente que consulto os meus dados cadastrais e posso realizar a alteração dos meus dados de endereço.

**Cenário 1 – Cadastro Cliente**

Dado que o cliente deseja realizar o seu cadastro.

Criar endpoint para cadastro do cliente (e-mail, nome, senha, lista de endereços) **Bonus:**

* E-mail não pode ser repetido, retornar mensagem apropriada caso já exista e-mail;

**Cenário 2 – Produtor de E-mail Queue**

Dado que o cliente realizou o cadastro criar um produtor de mensageria que envia uma mensagem para o MOM.

**Bonus:**

* Mensagem deve conter token, nome, e-mail, e-mail body;

**Cenário 3 – Consumidor de E-mail Queue**

Dado que MOM contém mensagens na fila criar um consumidor que lê a mensagem e faz o envio de e-mail.

**Bonus:**

* Caso erro de envio de e-mail, enviar a mensagem para uma fila de e-mail com erro;

**Cenário 4 – Ativação cadastro**

Criar endpoint validar a link de ativação e ativar o cadastro do cliente

**Bonus:**

* Validar token, retornar mensagem associada;

**Cenário 5 – Login cliente**

Criar endpoint para o cliente realizar o login com e-mail e senha

**Bonus:**

* retornar mensagem associada em caso de erro;
* Sucesso retornar objeto cliente

**Cenário 6 - Consultar CEP**

Dado que o cliente deseja consultar o seu endereço pelo CEP.

Criar endpoint para consulta de endereço utilizando CEP como parâmetro.

**Cenário 7 – Consultar Estados**

Dado que o cliente deseja escolher o seu estado.

Criar endpoint para consulta de estados.

**Bonus:**

Ordenar os estados da seguinte maneira:

* O primeiro e o segundo estado devem ser São Paulo e Rio de Janeiro, respectivamente;
* Os demais estados devem estar ordenados em ordem alfabética.

**Cenário 8 – Consultar Municípios**

Dado que o cliente deseja escolher o seu município.

Criar endpoint para consulta de municípios com base no estado.

**Cenário 9 – Alterar endereço**

Dado que o cliente deseja alterar o seu endereço.

Criar endpoint para alterar o endereço do cliente.

**Observação**

Responder o código de status HTTP apropriado.

Em caso de erro, todas as mensagens devem retornar o formato:    {"mensagem": "mensagem de erro"}

Para facilitar o desenvolvimento da aplicação, você poderá usar os serviços abaixo:

https://viacep.com.br/ws/{cep}/json

https://servicodados.ibge.gov.br/api/v1/localidades/estados/

<https://servicodados.ibge.gov.br/api/v1/localidades/estados/>{id}/municipios

1. **Requisitos Essenciais:**

* Linguagem: JAVA
* Framework: Spring Boot
* Testes unitários: Junit
* Mock: Mockito
* RESTFull
* Database Mysql
* Menssageria: ActiveMQ
* Documetnação: Suwgger
* Ambiente de dev e prod
* Docker com loadbalance para dev e prod

**Bônus**:

* JWT com token
* Criptografia não reversível para senha
* Padrão de arquitetura: CLEAN
* LOG estruturado
* Git Flow