1. **Desafio BackEnd**

Atualmente não é possível que o cliente realize a alteração de dados cadastrais através dos canais online. Exemplo: Endereço.

Apenas através da central de atendimento é possível solicitar a alteração deste dado cadastral. Com o objetivo de reduzir custos de call center e proporcionar comodidade e agilidade para os clientes, precisamos possibilitar a alteração dos dados cadastrais através dos nossos canais digitais.

1. **API**:

**User Stories:**

Sendo um cliente que consulto os meus dados cadastrais nos canais digitais

Posso realizar a alteração dos meus dados de endereço

Para que possa ter maior controle das minhas informações e autonomia para gerenciá-las, por meio do auto-atendimento.

**Cenário 1 - Consultar Cliente**

Dado que o cliente deseja consultar o seu cadastro.

Criar endpoint para consulta dos dados cadastrais utlizando CPF como parâmetro.

**Cenário 2 - Consultar CEP**

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

Criar endpoint para consulta de endereço utlizando CEP como parâmetro.

**Cenário 3 – Consultar Estados**

Dado que o cliente deseja escolher o seu estado.

Criar endpoint para consulta de estados.

**Bônus:**

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 4 – Consultar Municípios**

Dado que o cliente deseja escolher o seu município.

Criar endpoint para consulta de municípios com base no estado.

**Bônus**:

**Cenário 5 – Alterar endereço**

Dado que o cliente deseja alterar o seu endereço.

Criar endpoint para alterar o endereço do cliente.

**Observação**

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 em memória

**Bônus**:

* Padrão de arquitetura: CLEAN
* LOG estruturado
* Git Flow