

DESAFIO SELEÇÃO JAVA v2

1. DESAFIO

O desafio consiste em criar um microservice que consulte 2 APIs externas, gerar um CSV e um JSON e fazer o download.

- a. Deve ser usado a API para consultar os estados do Brasil:
<https://servicodados.ibge.gov.br/api/v1/localidades/estados>
- b. Deve ser usado a API para consultar as cidades:
<https://servicodados.ibge.gov.br/api/v1/localidades/estados/{UF}/municipios>
- c. A documentação completa das APIs está no site:
<https://servicodados.ibge.gov.br/api/docs/localidades>
- d. Os campos do CSV/JSON deverá ser:
 - i. idEstado
 - ii. siglaEstado
 - iii. regioaoNome
 - iv. nomeCidade
 - v. nomeMesorregiao
 - vi. nomeFormatado {cidade/UF}

2. REQUISITOS

Abaixo seguem os requisitos da biblioteca a ser desenvolvida.

- a. NO CSV, a primeira linha (cabeçalho) deve conter o nome de cada campo e a(s) linha(s) subsequente(s) deve(m) conter os valores resultante da consulta a API.
- b. Deverá ter um endpoint que retorna um json com todos os dados.
- c. Deverá ter um endpoint que retorna um CSV com todos os dados.
- d. Deverá ter um endpoint que envia um parâmetro, **nomeCidade**, e retorna somente o ID da cidade.
- e. Usar um cache no item (d), para que quando o nome de uma cidade for enviado mais de uma vez, evite a chamada do serviço externo.
- f. No endpoint do CSV deverá retornar um objeto do tipo **java.io.OutputStream** como saída da transformação.

3. ENTREGA

O código da biblioteca, diagrama de classes e classe de testes devem ser enviados para o GitHub ou BitBucket do candidato.

4. SOBRE A AVALIAÇÃO

Iremos avaliar o teste com os critérios abaixo e você será enquadrado como Pleno ou Sênior.

a. Para Pleno:

- i. Deverá usar somente o Spring boot e suas bibliotecas;
- ii. Os procedimentos da biblioteca devem ser logados utilizando o mecanismo de Log do Java;
- iii. Nível de cumprimento dos requisitos;
- iv. Abrangência dos testes unitários;
- v. A estrutura da biblioteca deverá ser flexível a ponto de permitir o fácil desenvolvimento de futuros formatos de exportação, como XML;
- vi. Uso adequado de padrões de projetos;
- vii. Flexibilidade do código para futuras evoluções;
- viii. Clean code;

b. Para Sênior:

- i. Todos os critérios do Desenvolvedor II;
- ii. Utilização de princípios SOLID;
- iii. Utilização de design patterns;
- iv. Otimizações em relação ao uso de memória;
- v. Utilização de bibliotecas corretas do Spring Boot;
- vi. Implementação de Circuit Breaker no acesso aos serviços externos;
- vii. Implementação do Swagger nas API;
- viii. Implementação correta do Cache.