

# **DESAFIO SELEÇÃO JAVA**

## **v4**

## 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:
  - a. idEstado
  - b. siglaEstado
  - c. regioaoNome
  - d. nomeCidade
  - e. nomeMesorregiao
  - f. 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 Júnior:

- a. Os procedimentos da biblioteca devem ser logados utilizando o mecanismo de Log do Java;
- b. Nível de cumprimento dos requisitos;
- c. Abrangência dos testes unitários de 30% de cobertura;

### B. Para Pleno:

- a. Todos os critérios do Desenvolvedor I;
- b. Deverá usar somente o Spring boot e suas bibliotecas;
- c. Abrangência dos testes unitários de 70% de cobertura;
- d. A estrutura da biblioteca deverá ser flexível a ponto de permitir o fácil desenvolvimento de futuros formatos de exportação, como XML;
- e. Uso adequado de padrões de projetos;
- f. Flexibilidade do código para futuras evoluções;
- g. Clean code;

### C. Para Sênior:

- a. Todos os critérios do Desenvolvedor II;
- b. Utilização de princípios SOLID;
- c. Utilização de design patterns;
- d. Otimizações em relação ao uso de memória;
- e. Utilização de bibliotecas corretas do Spring Boot;
- f. Implementação de Circuit Breaker no acesso aos serviços externos;
- g. Implementação do Swagger nas API;
- h. Implementação correta do Cache.