DESAFIO SELEÇÃO JAVA
v3

NEGÓCIO FICTÍCIO

Minha empresa, que chama ***PrintLocal***, tem no seu dia a dia a necessidade de gerar em PDF uma listagem com os estados e cidades do Brasil para poder enviar por email.
Nós geramos essa listagem para todos os nossos clientes, que hoje são 10 mil.
Gostaria de poder possibilitar que os próprios clientes iniciem a geração do PDF e recebam por email.

DESAFIO

1. PLANEJAMENTO

Extrair do “Negócio Fictício” uma lista de funcionalidades a serem desenvolvidas.
Por exemplo:
Criar API **Ou** Gerar PDF.

2. ARQUITETURA

A arquitetura do teste deve ser a de microservices, utilizando os padrões de mercado e boas práticas. Fazer um desenho da arquitetura a ser implementada.

3. API EXTERNA

Deve ser usado a API para consultar os estados do Brasil:
https://servicodados.ibge.gov.br/api/v1/localidades/estados

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

4. PDF

Pode ser montado uma listagem simples, com colunas siglaEstado, regiaoNome, nomeCidade, nomeMesorregiao, nomeFormatado {cidade/UF};

5. ENTREGA

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

6. SOBRE A AVALIAÇÃO

Iremos avaliar o teste com os critérios abaixo:

a. **Para Sênior:**

i. Deverá usar somente o Spring boot, Spring Cloud e suas bibliotecas;
ii. Pode ser usado banco de dados, mensageria, etc..
iii. Funcionalidades planejadas;
iv. Arquitetura criada;
v. Os procedimentos da biblioteca devem ser logados utilizando o mecanismo de Log do Java;
vi. Nível de cumprimento dos requisitos;
vii. Abrangência dos testes unitários;
viii. Flexibilidade do código para futuras evoluções;
ix. Clean code;
x. Utilização de princípios SOLID;
xi. Utilização de design patterns;
xii. Otimizações em relação ao uso de memória;
xiii. Utilização de bibliotecas corretas do Spring Boot;
xiv. Implementação de Circuit Breaker no acesso aos serviços externos;
xv. Implementação correta do Cache;
xvi. Gerar PDF de forma performática;
xvii. Nomenclatura da API;