**Desafio Serasa Experian – Nível 3**

**Objetivo: Criar um serviço do tipo API RESTful, para cadastro de pessoas com score e suas regiões de afinidades e criar interfaces integradas com o API de Pessoas.**

1. ***Para a construção desse serviço, algumas premissas e sugestões***

* Usar linguagem Java (preferência Java 11);
* Usar maven no build do projeto;
* Pode criar archetype usando o start.spring.io (adicione as dependências que achar relevantes);
* Usar framework Spring (incluindo Spring Boot, para iniciar o serviço);
* Montar um banco de dados em memória (pode usar o H2 ou HSQLDB), usando Hibernate na persistência de dados;
* Necessário pelo menos um teste unitário para cada método da camada Service, usando JUnit e Mockito;
* Documentar contratos REST usando Swagger;

1. ***Endpoints do serviço***

* POST /pessoa
  + Informar a seguinte estrutura de dados na inclusão:

|  |
| --- |
| {  “nome”: “Fulano de Tal”,  “telefone”: “99 99999-9999”,  “idade”: 99,  “cidade”: “Cidade de Fulano”,  “estado”: “XX”,  “score”: 1000, // Entre 0 e 1000  “regiao”: “sudeste”  } |

* + Adicionar um atributo id automático e data de inclusão, além dos dados do POST, durante inclusão dos dados no banco;
  + Retornar 201 no sucesso da inclusão;
* POST /afinidade
  + Informar a seguinte estrutura de dados na inclusão:

|  |
| --- |
| {  “regiao”: “sudeste”,  “estados”: [  “SP”,  “RJ”,  “MG”,  “ES”  ]  } |

* + Retornar 201 no sucesso da inclusão;
* POST /score
  + Informar a seguinte estrutura de dados na inclusão:

|  |
| --- |
| {  “scoreDescricao”: “Insuficiente”,  “inicial”: 0,  “final”: 200  } |

* + Retornar 201 no sucesso da inclusão;
* GET /pessoa/{id}
  + Se id encontrado no banco, retornar a seguinte estrutura de dados:

|  |
| --- |
| {  “nome”: “Fulano de Tal”,  “telefone”: “99 99999-9999”,  “idade”: 99,  “scoreDescricao”: “Recomendável”,  “estados”: [  “SP”,  “RJ”,  “MG”,  “ES”  ]  } |

* + Se id encontrado no banco, retornar 200, com a estrutura de dados;
  + Se id não encontrado no banco, retornar 204 (no content);
* GET /pessoa
  + Retornar uma lista de todo o cadastro, sendo cada item da lista com a seguinte estrutura de dados:

|  |
| --- |
| [  {  “nome”: “Fulano de Tal”,  “cidade”: “Cidade de Fulano”,  “estado”: “XX”,  “scoreDescricao”: “Recomendável”,  “estados”: [  “SP”,  “RJ”,  “MG”,  “ES”  ]  },  {  “nome”: “Sicrano de Tal”,  “cidade”: “Cidade de Sicrano”,  “estado”: “XX”,  “scoreDescricao”: “Insuficiente”,  “estados”: [  “RS”,  “PR”,  “SC”  ]  }  ] |

* + Se algum cadastro encontrado no banco, retornar 200, com a estrutura JSON;
  + Se nenhum item encontrado no banco, retornar 204 (no content);

1. ***Lógica do serviço***

* Montar lógica na camada Service, para associar a regiao da afinidade, com a regiao da pessoa, e retornar a lista de estados correspondentes à regiao.
* Montar lógica na camada Service, para retornar o atributo **scoreDescricao**, correspondente ao score encontrado entre **inicial** e **final**;
* Cadastrar via POST os seguintes dados na estrutura score:

|  |  |  |
| --- | --- | --- |
| **scoreDescricao** | **inicial** | **final** |
| Insuficiente | 0 | 200 |
| Inaceitável | 201 | 500 |
| Aceitável | 501 | 700 |
| Recomendável | 701 | 1000 |

1. ***Estrutura do Banco de dados***

* pessoa
  + id – numérico
  + dataInclusao – data
  + nome – texto
  + telefone – texto
  + idade – numérico
  + cidade – texto
  + estado – texto
  + regiao – texto
* afinidade
  + regiao – texto
  + estados – lista
* score
  + descricao – texto
  + inicial – numérico
  + final – numérico

**Frontend**

**Requisitos Técnicos**

Usar angular framework. Se achar necessário pode-se adicionar qualquer biblioteca, ferramenta ou prática que possa melhor o processo de desenvolvimento/entrega.

1 – Criar tela com lista de pessoas cadastradas exibindo nome, idade e scoreDescricao. Ao clicar no nome deve ir para página de mais detalhes daquela pessoa. Deve-se poder acessar essa tela via rota /pessoas

2 – Criar tela com detalhes com seguintes dados da pessoa: nome, telefone, idade e scoreDescricao e estados. Deve-se poder acessar essa tela via rota /pessoas/:id

3 – Criar tela de cadastro de pessoa com os seguintes campos em um formulário

nome, telefone, idade, cidade, estado, score e regiao. O campo regiao deve ser uma lista das afinidades cadastradas. Todos os campos são obrigatórios. Deve-se poder acessar essa tela via rota /pessoas/cadastro

4 – Criar tela com listagem de scores com os seguintes campos scoreDescricao, inicial, final. Deve-se poder acessar essa tela via rota /scores

5 – Criar tela de cadastro de afinidade com os seguintes campos em um formulário

região e estados. O campo de estados é uma lista de 1 ou mais estados, os campos são obrigatórios. Deve-se poder acessar essa tela via rota /afinidades/cadastro

6 – Na mesma tela de cadastro de afinidades deve existir uma lista com as afinidades cadastradas exibindo os campos: regiao e estados.

Entregar o código fonte em até **5 dias úteis** em um repositório público (como **GitHub**), encaminhando no e-mail que recebeu da recrutadora.