

Uma agência de banco de sangue forneceu uma lista de candidatos a doadores e precisa de um sistema que processe esses dados para extrair algumas informações.

Implemente um sistema web que receba um JSON com os dados e mostre os seguintes resultados:

- Quantos candidatos temos nessa lista em cada estado do Brasil?
- IMC médio em cada faixa de idade de dez em dez anos: 0 a 10; 11 a 20; 21 a 30, etc. ( $IMC = \text{peso} / \text{altura}^2$ )
- Qual o percentual de obesos entre os homens e entre as mulheres? (É obeso quem tem  $IMC > 30$ )
- Qual a média de idade para cada tipo sanguíneo?
- A quantidade de possíveis doadores para cada tipo sanguíneo receptor.

ATENÇÃO: Somente pessoas com idade de 16 a 69 anos e com peso acima de 50 Kg podem doar sangue.

Tipo sanguíneo	Pode doar para	Pode receber de
A+	AB+ e A+	A+, A-, O+ e O-
A-	A+, A-, AB+ e AB-	A- e O-
B+	B+ e AB+	B+, B-, O+ e O-
B-	B+, B-, AB+ e AB-	B- e O-
AB+	AB+	A+, B+, O+, AB+, A-, B-, O- e AB-
AB-	AB+ e AB-	A-, B-, O- e AB-
O+	A+, B+, O+ e AB+	O+ e O-
O-	A+, B+, O+, AB+, A-, B-, O- e AB-	O-

O sistema deverá ser implementado em java com Spring boot. Para o front-end, utilize preferencialmente Angular e para o banco de dados, MySQL. Caso não tenha condições ou prefira utilizar outras tecnologias para o front-end e para o banco de dados, fique à vontade, mas lembre-se que para avaliarmos, precisaremos executar o programa, portanto, forneça tudo o que for necessário para que possamos executar e testar o sistema em nosso próprio ambiente.

Poderão ser utilizadas bibliotecas de terceiros, desde que possam ser baixadas automaticamente no nosso ambiente de testes (pelo Maven ou pelo npm).

O arquivo JSON com os dados é fornecido junto com o este documento.

Prazo para o teste: 4 dias corridos.

Conhecimentos e habilidades que serão avaliadas:

### Back-end

- Linguagem Java
- POO
- Spring boot
- JPA
- Imutabilidade
- Segurança
- Protocolo HTTP
- Rest
- JSON
- Maven

### Front-end

- Usabilidade
- Bom gosto
- Responsividade
- Html
- CSS
- Javascript
- Ajax
- Angular
- Type Script
- npm

### Banco de dados

- SQL
- DDL
- Normalização
- Desempenho
- Integridade

### Geral

- Lógica
- Arquitetura
- Padronização do código
- Boas práticas
- Recursos avançados