

Teste de programação

1- API Rest

Utilize o framework Spring Boot com a linguagem Java na versão 11 ou 17.

Crie um CRUD para um cadastro de Pessoa com os seguintes campos/atributos:

- ID
- Nome
- CPF
- Data de Nascimento

A Pessoa deve ter uma lista de Contato com os seguintes campos/atributos:

- ID
- E-mail
- Telefone

OBS: Os contatos devem ser da própria pessoa. Exemplo:

João da Silva ...

Telefone: 44 9999-8888, E-mail: joaodasilva@gmail.com

Telefone: 44 7777-5555, E-mail: joao.silva@elotech.com.br

Requisitos:

- Possuir os endpoints:
 - GET (buscar uma única Pessoa)
 - GET (busca paginada)
 - POST
 - PUT
 - DELETE
- Utilizar algum banco de dados relacional (preferencialmente H2 ou Postgres)

Validações:

- Todos os campos são obrigatórios, tanto da Pessoa como do Contato
- A Pessoa deve possuir ao menos um Contato
- Validar sintaxe do CPF
- Validar sintaxe do e-mail
- A Data de Nascimento não poderá ser data futura

Requisitos Técnicos:

- Testes unitários

- Publicar o código em repositório público (github ou similar)

Opcional:

- Implementar o front-end em Angular (versão 2.XXXX) para consumir a API.

2- Teste lógico

Problema:

Uma escola possui a seguinte política de notas:

- Cada aluno recebe uma nota de 0 até 100 por disciplina/matéria.
- Qualquer nota menor que 40 é uma nota de reprovação.

O professor Adalberto costuma adaptar um pouco as regras para ajudar os alunos:

- Se a diferença entre a nota e o próximo múltiplo de 5 for menor que 3, a nota é arredondada para o próximo múltiplo de 5.
- Se o valor da nota é menor do que 38, não ocorrerá nenhum arredondamento, pois o resultado ainda será uma nota de reprovação.

Exemplo:

- Nota 84. Arredonda para 85 (85 - 84 é menor que 3)
- Nota 29. Não arredonda (resultado é menor que 40)
- Nota 57. Não arredonda (60 - 57 é igual a 3)

Resolva este Problema utilizando Java (versão 11 ou 17) e escreva testes unitários para comprovar o funcionamento do método.

Publicar o código em repositório público (github ou similar).