

Desafio Agenda de contatos

Requisitos

- Crie uma API REST de Agenda de Contato.
- A agenda deve permitir armazenar múltiplos contatos, contendo pelo menos as informações de nome e sobrenome.
- Um contato pode possuir múltiplos telefones. O armazenamento de telefones deve permitir incluir separadamente as informações de DDD e número. Não deve aceitar valores que não sejam numéricos.
- Um contato pode possuir múltiplos endereços. O armazenamento de endereços deve permitir incluir separadamente as informações de CEP, logradouro, número, cidade e estado.
- Os dados devem ser armazenados em um banco de dados de sua preferência.

Requisitos básicos

O sistema deve permitir a interação com o usuário com as seguintes ações:

- Adicionar um contato e seus dados.
- Listar todos os contatos da agenda.
- Buscar um contato de acordo com uma palavra-chave (Utilize os dados de nome, sobrenome para realizar a busca).
- Remover um contato da agenda.
- Remover todos os contatos da agenda.
- Atualizar os dados de um contato.
- Adicionar um telefone a um contato.
- Adicionar um endereço a um contato.
- Remover um telefone de um contato da agenda.
- Atualizar os dados de um telefone.
- Remover um endereço de um contato da agenda.
- Atualizar os dados de um endereço.
- Exibir todas as informações de um contato da agenda.
- Listar todos os telefones de um contato da agenda.
- Listar todos os endereços de um contato da agenda.
- Exibir todas as informações de um telefone de um contato da agenda.
- Exibir todas as informações de um endereço de um contato da agenda.

Requisitos bônus

Ações adicionais que podem ser incluídas na Agenda como um bônus:

- Exibir a lista de contatos com paginação. Incluindo a opção para navegar para a página seguinte ou página anterior.
- Exibir a lista de telefones com paginação. Incluindo a opção para navegar para a página seguinte ou página anterior.
- Exibir a lista de endereços com paginação. Incluindo a opção para navegar para a página seguinte ou página anterior.
- Exibir a documentação da API através do Swagger ou outra ferramenta.
- Criar validações para não permitir a inserção de dados vazios ou nulos, utilizar Spring Validation.
- Criar testes unitários para os serviços.

Regras de Negócio:

RN1: O sistema não deve permitir adicionar um contato com os mesmos dados de outro contato já existente;

RN2: O sistema não deve permitir adicionar um telefone com os mesmos dados de outro telefone já existente no mesmo contato;

RN3: O sistema não deve permitir adicionar um endereço com os mesmos dados de outro endereço já existente no mesmo contato.