

## Avaliação técnica – Desenvolvimento | Navita

### Exercício 1)

Dois números inteiros não negativos são chamados de irmãos se eles podem ser obtidos uns dos outros, apenas reorganizando os dígitos de suas representações decimais. Por exemplo, 123 e 213 são irmãos. 535 e 355 também são irmãos.

Um conjunto que consiste em um inteiro não negativo N e todos os seus irmãos é chamado de família de N. Por exemplo, a família de 553 é composta por três números: 355, 535 e 553.

Escreva um algoritmo de contenha uma função com as seguintes entradas e saídas:

- entrada: N (inteiro)
- saída: inteiro

Dado um número inteiro não negativo N, a função deve retornar o maior número na família de N. Ela deve retornar -1 se o resultado exceder 100.000.000.

Exemplo, uma resposta para este exercício pode ter o seguinte formato:

```
solution int(int N);
```

A inserção de N=213 nesta função deve retornar 321. Para N=553, a função deve retornar 553.

### *Critérios de avaliação*

- Organização e qualidade do código;
- Utilização de testes;

## Exercício 2)

Crie uma Web API REST para o gerenciamento de patrimônios de uma empresa.

### *Requisitos*

#### **Patrimônio**

*Campos:*

Nome - obrigatório

MarcaId – obrigatório

Descrição

Nº do tombo

**Regras:** O nº do tombo deve ser gerado automaticamente pelo sistema e não pode ser alterado pelos usuários.

---

#### **Marca**

*Campos:*

Nome – obrigatório

MarcaId - obrigatório

**Regras:** Não deve permitir a existência de duas marcas com o mesmo nome.

---

#### **Usuário**

*Campos:*

Nome – obrigatório

Email – obrigatório

Senha - obrigatório

**Regras:** Não permitir e-mail repetido

---

### *Requisitos técnicos*

- Deve se utilizar C#/Java;
- Os dados devem ser salvos em SQL Server/PostgreSQL;
- Os endpoints devem utilizar o formato JSON;
- A sua aplicação deve conter um arquivo README explicando a solução adotada na sua implementação do desafio;
- Implementar Autenticação com Token (como preferir);

- Os endpoints só poderão ser acessados por usuário autenticados (com exceção do POST de criação de usuário);

### *Observações/Dicas*

- Não se limite às funcionalidades acima. Qualquer outra feature extra é bem-vinda;
- A arquitetura é por sua conta;
- Coloque um script de criação do banco de dados junto ao projeto;
- Não é necessária a criação de telas;
- Comunique qualquer impedimento/dúvida o quanto antes, para que possamos analisar;

### *Critérios de avaliação*

- Organização e qualidade do código;
- Organização da estrutura e arquitetura desenvolvida;
- Documentação do projeto (readme);