

## **Desenvolvedor Java BackEnd**

### **Sobre o Hopi Hari**

Hopi Hari é um parque temático brasileiro localizado no Km 72 da Rodovia dos Bandeirantes, no município de Vinhedo - SP. O parque está a 30 km de Campinas e 72 km de São Paulo. Com 760 mil metros quadrados, é considerado um dos maiores parques de diversão da América Latina e foi projetado como um país fictício.

### **Teste para BackEnd**

Montamos este teste para conhecer seus conhecimentos e habilidades em linguagem Java, programação orientada a objetos e boas práticas de programação.

O teste consiste em montar uma aplicação Java Web capaz de persistir um cadastro em um banco de dados em memória ou em arquivo, listar os cadastros em uma interface simples.

### **Proposta**

O 'novo' sistema de cadastro de visitantes que adquiriram o passaporte para acesso anual 'Anuali' ao parque precisa de uma nova cara! Isso porque a área responsável definiu critérios específicos para inclusão e manutenção dessas informações.

Seu desafio é elaborar um sistema em Java capaz de:

1. Permitir o cadastro dos visitantes.
2. Apresentar um cadastro contendo rg, nome, e-mail, telefone, cidade do visitante e período de validade do passaporte 'Anuali' que deve ser de 1 ano da data de inclusão, (sendo que rg, nome e e-mail são obrigatórios).
3. Persistir a informação cadastrada em um banco de dados em memória, como HSQLDB ou arquivo.
4. Obter, a qualquer momento, a lista de todos os visitantes cadastrados.
5. Impedir a utilização do mesmo documento para mais de um visitante, caso já exista, será uma renovação onde deverá ser atualizado os dados e revalidado o período de validade do passaporte.
6. Obrigatório o cadastro conter foto, pode ser feito o upload de uma foto do visitante armazenada localmente ou de uma rede social.

### **Casos de uso**

#### **+ Cadastro com sucesso:**

1. É informado o RG do visitante e verificado se já existe cadastro.

2. Se tratando de um novo cadastro será cadastrado os dados do visitante 'Rodrigo' e feito o upload de sua foto.
3. O sistema recebe o cadastro e verifica ser um novo cadastro.
4. O sistema persiste os dados em um banco de dados em memória ou em um arquivo
5. O sistema informa que o visitante foi cadastrado corretamente e mostra uma imagem de sucesso

#### **+ Cadastro já existente:**

1. É informado o RG do visitante 'João' e verificado se já existe cadastro.
2. Cadastro encontrado, será carregado o cadastro completo e permitindo atualizar os dados do visitante.
3. O sistema recebe a atualização e exibe uma mensagem para confirmação da atualização dos dados.
4. O sistema informa que o passaporte do visitante foi renovado corretamente e mostra uma imagem de sucesso

#### **+ Relatório de visitantes cadastrados:**

1. É clicado em 'Visualizar relatório dos visitantes'
2. O sistema consulta o banco de dados em memória ou em arquivo
3. O sistema apresenta todos os visitantes cadastrados. Cada linha tem as informações: rg, nome, e-mail, telefone, cidade do visitante e período de validade do passaporte.

### **Instruções**

Não há certo ou errado. Queremos apenas saber mais sobre seus conhecimentos na linguagem Java, como uso de bibliotecas públicas, e seu cuidado com o código fonte, levando em consideração clareza de ideias, organização de código, documentação e testes.

Crie um novo projeto no seu próprio GitHub e siga os seguintes passos:

1. Crie um novo projeto dentro do seu GitHub (<https://github.com>).
2. Desenvolva um sistema que atenda os casos de uso apresentados.
3. Para montar seu sistema, leve em consideração a arquitetura de referência, disponível no anexo ao final do documento
4. Criar uma interface em HTML que contenha um formulário para cadastro.
5. Criar uma interface em HTML que liste os visitantes cadastrados.

6. Criar uma ou mais classes que contenha(m) as regras para persistir e recuperar os cadastros de visitantes.
7. Documente como o projeto deve ser iniciado para que possamos rodar sua aplicação
8. Suba a sua proposta para o projeto que você criou no GitHub. Exemplo: <https://github.com/seuNome>
9. Envie-nos o link do GitHub do seu projeto para podermos realizar o download. Exemplo: <https://github.com/seuNome/test-Java.git>
10. Aguarde o RH entrar em contato.

### **Regras**

1. Você poderá utilizar o Java em qualquer versão.
2. Você poderá utilizar quaisquer frameworks da linguagem Java.
3. Para persistir as informações, você poderá utilizar um banco de dados em memória gerenciado por você ou utilizar um banco, como HSQLDB.
4. Você também pode optar por gravar em arquivo.
5. Detalhes como criação de testes unitários, ordenação da lista de cadastrados ou filtro da lista são opcionais, mas se você fizer iremos apreciar!

### **O que apreciamos**

- \* Organização;
- \* Simplicidade;
- \* Objetividade;
- \* Reuso de código;
- \* Testes unitários;
- \* Padronização de código;
- \* Padrões de projeto;

### **Quem buscamos**

Queremos uma pessoa que goste do que faz, trabalhe em equipe e tenha vontade de inovar, buscando sempre atualização e conhecimento.

Se você se identificou, venha fazer parte do nosso time!

## Anexo – Arquitetura de Referência

