



Desafio Spring

▶ Veterinária

// Projeto em grupo

Objetivo

O objetivo é verificar os conceitos de Java e Spring.

Você está pronto? Vamos! 😊👍



Veterinária

A clínica veterinária “Zoo World” quer implementar um sistema desenvolvido em Java para a administração de cada um de seus pacientes. Os veterinários têm dois tipos de especialidades: Animais Domésticos e Animais de Fazenda, com base no fato de que atendem cães, gatos, porquinhos-da-índia, coelhos, tartarugas, cavalos, vacas e ovelhas.

- De cada um de seus pacientes, eles precisam obter os seguintes dados:
 - número do paciente (gerado automaticamente),
 - espécie,
 - raça,
 - cor / s,
 - data de nascimento e
 - nome.
- Ao mesmo tempo, cada paciente tem um proprietário atribuído que deve ser conhecido:
 - cpf (formatado com a máscara ###.###.###-##),
 - nome,
 - sobrenome,
 - data_de_nascimento,
 - endereço e
 - telefone_contato.
- Cada consulta que você fizer a um paciente deve ser registrada, levando em consideração:
 - data e hora,
 - motivo,
 - diagnóstico possível e
 - tratamento a ser seguido.
- Por outro lado, cada consulta deve ter também um médico veterinário responsável do qual ficam armazenados:
 - cpf (formatado com a máscara ###.###.###-##),



- nome,
- sobrenome,
- número de registro e
- especialidade.

Solicita-se a realização da modelagem de cada uma dessas classes, levando em consideração suas relações e a interação entre cada uma delas. Implementar (nas devidas classes) os métodos necessários para:

- Cadastrar/editar médicos;
 - Não é permitido o cadastro de médicos em duplicidade;
 - Todos os dados do médico são obrigatórios para realização do cadastro/edição;
 - Não é permitido a exclusão de médicos que já realizaram consultas.
- Cadastrar/editar proprietários;
 - Não é permitido o cadastro de proprietários em duplicidade;
 - Todos os dados do proprietários são obrigatórios para realização do cadastro/edição;
 - Não é permitido a exclusão de proprietários de animais que já realizaram consultas.
- Cadastrar/editar pacientes;
 - Todos os dados do paciente são obrigatórios para realização do cadastro/edição;
 - Não é permitido a exclusão de animais que já realizaram consultas.
- Criar/editar consultas;
 - Uma consulta deve exigir o médico e o paciente para ser registrada, bem como a data e hora e o motivo.
- Ver a lista de todos os pacientes registrados junto com seus proprietários. A lista deve estar ordenada em ordem crescente de nome do proprietário do animal;
- Ver a lista de todas as consultas atribuídas a um determinado paciente. A lista deve estar ordenada pela data da consulta, em ordem decrescente.
- Ver o total de consultas atribuídas a cada um dos médicos.



- Ver a lista de todas as consultas marcadas para um mesmo dia. É importante mostrar dados relevantes do animal (paciente), do proprietário e do médico. A lista deve estar ordenada em ordem crescente de data e hora.

Os registros devem ser mantidos em arquivos (txt ou json), portanto os registros relacionados aos médicos, animais, proprietários e consultas devem ser armazenados em estruturas de arquivos que permitam a recuperação dos dados sempre que solicitados. Lembrem-se do seguinte requisito: uma mesma pessoa pode ter vários animais.

Construa endpoints no **padrão REST** capazes de executar a persistência e obtenção de dados a fim de atender aos requisitos existentes. Cada grupo é responsável por definir a estrutura do payload de cada endpoint, bem como os parâmetros necessários a cada um deles. **Busque utilizar diferentes formas de passagem de parâmetros.**