

# ■ Trabalho Final – Sistema de Agenda do Veterinário

## ■ Objetivo

O objetivo deste trabalho é desenvolver um sistema orientado a objetos em Java que simule a agenda de um veterinário, permitindo o cadastro de animais e o agendamento de consultas. O projeto deve demonstrar a aplicação prática dos principais conceitos de Programação Orientada a Objetos (POO) estudados durante a disciplina.

A nota deste trabalho vale **5,0 pontos** e será somada à **nota de participação (5,0 pontos)**, totalizando 10,0 pontos na média final da disciplina.

## ■ Descrição do Sistema

O sistema deve representar uma agenda diária de uma clínica veterinária. O veterinário pode cadastrar até 10 consultas por dia. Caso a agenda esteja cheia, o sistema deve exibir uma mensagem informando que não há mais horários disponíveis.

Cada consulta deve estar associada a um animal, e o sistema deve permitir:

Cadastrar os dados do animal: nome, idade e raça. Escolher o tipo de consulta: Banho, Banho e Tosa, Tosa ou Consulta.

Se o tipo escolhido for Consulta, o sistema deve perguntar se o animal está doente (s/n). Caso esteja, o usuário deve informar qual é a doença, sendo que Cachorros, Gatos e Coelhos possuem doenças diferentes definidas em suas respectivas classes.

O programa pode ser feito com menu interativo no console ou usando JOptionPane para entrada de dados.

## ■■ Requisitos Técnicos

Conceito de OO	Aplicação Esperada
Classes e Objetos	Criar classes como Animal, Cachorro, Gato, Consulta etc.
Encapsulamento	Atributos privados e acesso controlado por métodos get e set.
Construtores	Criar construtores para inicializar objetos com seus dados obrigatórios.
Composição	A classe Consulta deve conter um objeto Animal.
Herança	Animal é uma classe pai e Cachorro, Gato, Coelho são subclasses.
Polimorfismo	Tratamento diferenciado de doenças de acordo com o tipo do animal.
Tratamento de Exceções	Validar entradas e lançar exceções personalizadas (ValidacaoException).

## ■ Validações Obrigatórias

Nome do animal deve conter apenas letras e espaços (mínimo 2 caracteres). Idade deve ser um número inteiro positivo (não pode ser 0 ou negativo). Raça deve ser informada. Agenda não pode ultrapassar 10 consultas.

## ■ Fluxo de Perguntas do Sistema

1. O que o usuário quer fazer: listar as consultas ou cadastrar uma consulta.
  - Se for listar, mostrar as consultas já cadastradas.
  - Se não houver consultas, mostrar 'Nenhuma cadastrada ainda'.
  - Se houver, mostrar a lista e pedir para o usuário apertar um botão para voltar.
2. Se for cadastrar:
  - Perguntar o tipo de animal (Cachorro, Gato ou Coelho).
  - Perguntar nome, idade e raça.
  - Perguntar o tipo de atendimento (Banho, Tosa, Banho e Tosa, Consulta).
  - Se for Consulta, pedir a doença.
  - A doença deve ser buscada nas possíveis doenças definidas em cada classe do animal.

## ■■■ Doenças Comuns por Animal

**Cachorro:** Cinomose, Parvovirose, Leptospirose, Dermatite, Otite.

**Gato:** PIF (Peritonite Infecciosa Felina), Rinotraqueíte, Calicivirose, Panleucopenia, Otite.

**Coelho:** Mixomatose, Pasteurelose, Sarna Auricular, Pododermatite, Coccidiose.

## ■ Entrega

O trabalho deve ser entregue com o código-fonte completo e funcional (em pastas organizadas), via GitHub.

## ■ Critérios de Avaliação

Nota máxima: 5,0 pontos + 0,5 ponto extra (bônus) para quem implementar testes unitários com JUnit.

Critério	Peso
Aplicação correta dos conceitos de OO	1,5
Organização e clareza do código	1,0
Funcionamento geral do sistema	1,5
Validações e tratamento de exceções	1,0
Total	5,0
Bônus: Testes Unitários (JUnit)	+0,5