

---

## Lista de Exercícios 05

### Sistema de Clínica Veterinária 🐶 🐱

Você já possui a classe **Tutor** e **Animal**.

Agora, incremente o projeto considerando:

#### 1. Nova classe:

- a. Crie as classes **Veterinário**, **Atendimento** e **Procedimento**  
**Classe Veterinario**

Atributos:

- nome (String) – Nome completo do veterinário.
- crmv (String) – Registro profissional do veterinário.
- especialidade (String) – Área de atuação (ex.: “Clínico Geral”, “Dermatologia”, “Ortopedia”).
- supervisor (Veterinario) – Associação reflexiva (um veterinário pode ter outro como supervisor).

Métodos:

- Construtor(es) para inicializar nome, CRMV e especialidade.
- Getters e setters para os atributos.
- setSupervisor(Veterinario supervisor) – Define quem é o supervisor.
- retornaDados() – Retorna uma string com as informações básicas do veterinário.

#### **Classe Atendimento**

Representa uma consulta/atendimento feito por um Veterinário a um Animal.

Atributos:

- data (LocalDate ou String)
- descricao (String) → resumo da consulta
- veterinario (Veterinario) → referência ao profissional responsável
- **Lista de Procedimentos** (composição).

Métodos:

- Construtor(es).
- Métodos para adicionar/remover procedimentos.
- exibirDados() para exibir dados do atendimento.

#### **Classe Procedimento**

Parte integrante de um Atendimento.

Atributos:

- nome (String) → Ex.: “Vacina Antirrábica”, “Exame de Sangue”
- custo (double)
- observacoes (String)

2. **Associação**
  - a. Um **Tutor** pode ter **um ou vários animais**.
  - b. Um **Animal** deve estar vinculado a apenas **um tutor**.
3. **Associação com multiplicidade**
  - a. Um **Animal** pode ter **0 ou muitos atendimentos veterinários**
  - b. Um **Atendimento** pertence a apenas **1 animal**.
4. **Composição**
  - a. Um **Atendimento** é composto por **Procedimentos**.
  - b. Se o atendimento for removido, todos os procedimentos associados também desaparecem.
5. **Associação reflexiva**
  - a. Um **Veterinário** pode indicar outro veterinário como **supervisor** (associação reflexiva).
6. **Diagrama UML**
  - a. Atualize o diagrama de classes incluindo Veterinário, Atendimento, Procedimento e todos os novos relacionamentos.
7. **Implementação em Java**
  - a. Crie as classes e métodos necessários.
  - b. Atualize a classe Main para **testar os cenários**:
    - i. Criar tutores, animais e veterinários.
    - ii. Associar animais a seus tutores.
    - iii. Criar atendimentos com procedimentos para os animais.
    - iv. Relacionar veterinários com supervisores.