



## Trabalho 1

### **PROBLEMA:**

Implementar um sistema orientado a objetos em Python para gerenciamento de adoções de animais por uma ONG<sup>1</sup> especializada na adoção de animais.

### **ESCOPO DO DESENVOLVIMENTO:**

A adoção de um animal (gato ou cachorro) segue um processo para garantir o bem-estar do animal e a adoção responsável.

Os animais são doados por pessoas para a ONG. O cadastro dos animais possui: número do chip de identificação, nome e raça. No cadastro dos cães, além desses dados, também deve constar o tamanho (pequeno, médio ou grande).

Para cada animal, também é registrado o histórico de vacinação, contendo, em cada registro: data, o animal e a vacina aplicada. Enquanto o animal permanecer na ONG de adoção, sempre que uma nova vacina é aplicada, o histórico de vacinas do animal é atualizado.

O cadastro do doador do animal possui: cpf, nome, data de nascimento e endereço. Os animais doados são relacionados aos seus doadores por meio do registro da doação, que deve conter: a data da doação, o animal doado, o doador e o motivo da doação.

Para adotar um dos animais que estejam hospedados na ONG, primeiramente a pessoa interessada precisa ser cadastrada, incluindo: cpf, nome, data de nascimento, endereço, tipo da habitação (casa ou apartamento; pequeno, médio ou grande) e se já possui outros animais em casa.

Após o cadastro, a adoção do animal inicia com a avaliação do perfil do interessado e a escolha do animal a ser adotado. Se o animal escolhido puder ser adotado, o adotante deve assinar um termo de responsabilidade, comprometendo-se com o cuidado do animal adotado. Na adoção são registrados: a data, o animal adotado, o adotante e se foi assinado o termo de responsabilidade.

O sistema deve permitir emitir um relatório das adoções realizadas em determinado período. Também deve ser possível emitir um relatório das doações recebidas em determinado período.

Deve permitir também listar todos os animais que estão disponíveis para adoção, ou seja, aqueles que foram doados, já possuem as vacinas básicas e ainda não foram adotados.

---

<sup>1</sup> Organização Não Governamental



## UNIVERSIDADE FEDERAL DE SANTA CATARINA

**Disciplina:** INE5605 – Desenvolvimento de Sistemas Orientados a Objetos I

**Professores:** Thaís Idalino, Jean Hauck e Eduardo Camilo Inacio

---

Considere algumas regras:

1. Somente podem adotar animais as pessoas com mais de 18 anos completos.
2. Pessoas que doaram um animal não podem adotar um animal.
3. Somente podem ser adotados os animais que já receberam as vacinas: raiva, leptospirose e hepatite infecciosa.
4. Cães de porte grande não podem ser adotados por pessoas que moram em apartamento pequeno.

### RESTRIÇÕES DE ESCOPO:

Para simplificar este trabalho, o sistema contempla somente algumas das funcionalidades de um processo de adoção animal, não abordando todas as questões legais e de segurança, por exemplo.

Para este tema padrão, serão considerados:

- **cadastros:** doador, adotante e animal.
- **registros:** doação, vacina e adoção.
- **relatórios:** doações por período, adoções por período e animais disponíveis para adoção.

### ENTREGAS:

**Parte 1** (Parcial): Deve ser apresentado para o(a) professor(a) em sala de aula no dia **14/10/2024**:

- Um **documento** em formato PDF descrevendo como será a divisão das atividades do trabalho entre os membros da equipe;
- Uma figura com o diagrama de classes **somente das entidades** (Model), seguindo a notação UML 2;
- Código-fonte orientado a objetos em Python somente com a **implementação das entidades** (Model).

**Parte 2** (Parcial): Deve ser apresentado para o(a) professor(a) em sala de aula no dia **14/10/2024**:

- Uma imagem do diagrama de classes, seguindo a notação UML 2. Este diagrama de classes deve conter: o controlador principal, além de uma modelagem completa (controlador-entidade-tela) para apenas UMA das entidades do trabalho.



**Parte 3** (Completo): Deve ser postado um arquivo ZIP por equipe no Moodle até o dia **04/11/2024** às 23:59, contendo:

- **Código fonte** completo do sistema orientado a objetos em Python;
- Imagem contendo o **diagrama de classes**, seguindo a notação UML 2. Este diagrama de classes deve conter: o controlador principal, um dos controladores e também suas telas e entidades envolvidas;
- Link do **vídeo com a apresentação do trabalho**, máximo 15 minutos (obrigatório). Cada membro da equipe deve apresentar a sua parte do trabalho, mostrando o sistema funcionando e também apresentando o código-fonte que foi desenvolvido;
- **Link do git** com commits do trabalho (obrigatório, notas individuais virão daqui).

### **VÍDEO DE APRESENTAÇÃO DO TRABALHO:**

A apresentação do trabalho será realizada através da **gravação de um vídeo**. Na apresentação deve ser **demonstrado o sistema desenvolvido em execução**. Cada um dos membros do grupo deverá apresentar no vídeo gravado **a parte do código-fonte que desenvolveu**, explicando o código-fonte implementado. A nota de cada membro da equipe será **individual**, dependendo da sua participação individual no desenvolvimento do trabalho e na apresentação:

- A participação individual do aluno no trabalho deve ser comprovada por meio da listagem dos commits na ferramenta de controle de versões do código-fonte (GitHub, codigos.ufsc ou similar), ou outra forma que comprove de maneira inequívoca que os códigos foram elaborados pelo aluno.
- Os alunos devem iniciar o vídeo com a demonstração do funcionamento completo de cadastros e registros, e relatórios.
- Na sequência, devem apresentar o código fonte ou execução que indiquem a presença de cada item dos critérios de avaliação (associação, agregação, composição, herança, classes abstratas, uso de MVC, tratamento de exceções, etc.)
- A correção será feita com base no vídeo, auxiliada pela análise dos arquivos enviados e dos commits.

O aluno que não participar do vídeo de apresentação (salvo por motivo regimentalmente justificável) receberá nota zero no trabalho.

### **CRITÉRIOS DE AVALIAÇÃO:**

- Cadastro dos doadores, dos adotantes e dos animais, contemplando para cada um: inclusão, exclusão, alteração e listagem (2,0 pontos).
- Registro das doações, das vacinas e das adoções, contemplando para cada um: inclusão, exclusão, alteração e listagem (1,5 pontos).



**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**Disciplina:** INE5605 – Desenvolvimento de Sistemas Orientados a Objetos I

**Professores:** Thaís Idalino, Jean Hauck e Eduardo Camilo Inacio

---

- Emissão dos relatórios (1,5 pontos).

Além desses critérios funcionais, também serão avaliados:

- Entrega da Parte 1 e da Parte 2 em sala de aula (0,5 ponto)
- Qualidade, uso da notação e consistência do diagrama de classes (0,5 ponto)
- Utilização correta de: associação, agregação e composição (1,0 ponto)
- Utilização correta do MVC (1,0 ponto)
- Utilização correta de: herança e classes abstratas (1,0 ponto)
- Tratamento de todas as exceções (1,0 ponto)