

# Relatório Final – Sistema de Cadastro de Zoológico em Java

## 1. Introdução

O presente relatório tem como objetivo apresentar o desenvolvimento do projeto final da disciplina de Programação Orientada a Objetos (POO). O projeto consistiu na criação de um **sistema de cadastro de animais de um zoológico**, aplicado em Java, utilizando os principais pilares da orientação a objetos: **abstração, encapsulamento, herança e polimorfismo**.

A escolha do tema “zoológico” surgiu por se tratar de uma proposta equilibrada: nem tão simples a ponto de limitar o uso de conceitos aprendidos, nem tão complexa a ponto de comprometer o prazo e o escopo do trabalho. Assim, ele se mostrou adequado para aplicar e demonstrar todo o conteúdo visto na disciplina.

---

## 2. Etapa 1 – Concepção da Ideia

Na primeira fase, a equipe discutiu possíveis temas e rapidamente percebeu que um sistema de zoológico atendia perfeitamente às necessidades do projeto.

A ideia permitiria trabalhar com:

- **Hierarquia de classes**
- **Classes abstratas**
- **Métodos sobrescritos**
- **Várias espécies de animais**
- **Interações variadas (alimentar, cadastrar, emitir som, etc.)**

Dessa maneira, o grupo concluiu que o zoológico era o tema ideal para representar uma aplicação prática e completa dos conceitos de POO.

---

## 3. Etapa 2 – Planejamento das Classes

Com o tema definido, iniciou-se o planejamento da estrutura do código. A equipe decidiu que os animais seriam organizados por **espécies**, facilitando a criação de hierarquias e o uso de herança.

As primeiras decisões estruturais foram:

- Criar uma classe **Animal** como **classe abstrata**, servindo de base para todas as outras.
- Separar os animais por categorias, começando pelos **mamíferos**.
- Criar uma classe **Zoologico**, responsável pelo armazenamento e manipulação dos animais cadastrados.

Esse planejamento serviu de guia para a fase de implementação.

---

## 4. Etapa 3 – Implementação Inicial

A implementação começou pela estrutura principal do sistema. As primeiras classes desenvolvidas foram:

### 4.1 animal.java (classe abstrata)

Responsável por definir os atributos e métodos básicos comuns a todos os animais, como:

- Nome
- Idade
- Espécie
- Método abstrato **emitirSom()**

### 4.2 mamifero.java

Subclasse que herda de **Animal** e representa o primeiro grupo de espécies.

### 4.3 zoologico.java

Classe responsável por armazenar, gerenciar e manipular a lista de animais cadastrados.

### 4.4 Criação dos primeiros animais

Foram desenvolvidas três classes específicas:

- **Leão**
- **Macaco**
- **Elefante**

Cada uma sobrescrevia métodos conforme suas características, especialmente **emitirSom()**.

## **4.5 Desenvolvimento do main.java**

O arquivo principal (main) foi criado para permitir a interação com o usuário. Nele surgiram as principais funções do sistema:

### **Cadastrar animal**

- **Listar animais**
- **Emitir som dos animais**
- **Alimentar todos os animais**
- **Remover animal**
- **Encerrar o programa**

Assim, o sistema já estava funcional e permitia realizar operações básicas de POO de maneira dinâmica.

---

## **5. Etapa 4 – Aperfeiçoamento e Correções**

Com o protótipo inicial funcionando, o grupo passou para a fase de melhorias. Nessa etapa foram realizados os seguintes avanços:

### **5.1 Correção de bugs**

A equipe revisou entradas incorretas, falhas de cadastro e problemas de organização das classes.

### **5.2 Aperfeiçoamento de encapsulamento, herança e polimorfismo**

Foram refinados:

- Métodos getters e setters
- Uso correto de atributos privados e protegidos
- Melhora na sobrescrita dos métodos dos animais
- Melhor padronização da herança entre classes

### 5.3 Salvamento dos dados

Foi criada uma função capaz de **salvar todos os animais cadastrados e removidos**, de forma que:

- Mesmo após fechar o programa, os dados permanecem salvos.
- Ao reiniciar o sistema, o zoológico é automaticamente reconstruído a partir do arquivo.

Isso trouxe grande avanço na usabilidade e robustez do sistema.

---

## 6. Etapa 5 – Expansão Final e Funcionalidades Avançadas

Na última fase do projeto, o sistema foi expandido, refinado e ganhou funções mais sofisticadas.

### 6.1 Novas espécies

Além dos mamíferos, foram criadas classes para novos grupos de animais:

- **Aves**
- **Répteis**
- **Peixes**
- **Invertebrados**

Essa ampliação permitiu testar ainda mais a herança e o polimorfismo.

### 6.2 Melhoria do sistema de cadastro

O processo de cadastro foi reajustado para se tornar:

- Mais intuitivo
- Menos repetitivo
- Mais simples para o usuário identificar a espécie e inserir dados corretamente

### 6.3 Implementação de Thread de Auto-Save

Para aumentar a segurança dos dados, foi adicionada uma **Thread de auto-save**, que salva automaticamente todas as alterações feitas pelo usuário, evitando perda de dados em caso de falhas ou fechamento inesperado.

---

## 7. Conclusão

O projeto do sistema de zoológico cumpriu todos os objetivos propostos pela disciplina de Programação Orientada a Objetos.

Ao longo do desenvolvimento, foi possível aplicar de forma prática e eficiente:

- Abstração
- Encapsulamento
- Herança
- Polimorfismo
- Manipulação de arquivos
- Uso de threads
- Estruturação de um sistema completo em Java

O trabalho demonstrou a capacidade da equipe em idealizar, planejar, implementar e aprimorar um sistema real, envolvendo desde a concepção da ideia até o refinamento final.

Com isso, o grupo finaliza o projeto com um sistema funcional, organizado e alinhado aos princípios da POO.