

# Universidade Federal do Rio Grande do Norte

*Instituto Metr pole Digital - IMD  
Bacharelado em Tecnologia da Informa  o - BTI  
08 de Dezembro de 2017*

## Relat rio Resumo do PetFera

Componentes: Gleydvan Macedo e Jo o V tor Venceslau Coelho

Professor: Silvio Costa Sampaio

### O que foi feito

Foram criadas as classes para representar um total de 12 poss veis tipos de animais, sendo que cada uma das Classes Anf bio, Mam fero, Reptil e Aves, possuem duas subclasses que representam suas vers es nativas e ex ticas, e todas derivam da classe Animal.

Tamb m foi criada a Classe Funcion rio, que possui como subclasses Veterin rio e Tratador. Al m da Classe Pet\_Fera\_Cadastro, que realiza todas as opera  es solicitadas, al m de um programa auxiliar para filtrar animais de acordo com a classe, veterin rio e/ou tratador associado.

Foram criados dois CSVs para armazenar os Animais e os Funcion rios cadastrados conforme solicitado.

### Dificuldades encontradas

A principal dificuldade encontrada foi durante os processos de cadastro e leitura dos arquivos CSVs, pois n o conseguimos reduzir a quantidade de c digo repetido durante o cadastro por meio de uma fun  o para os campos em comum que os diversos animais compartilham, a solu  o utilizada foi deixar a massiva repeti  o de c digo sem simplifica  o, al m desse problema, tivemos certa dificuldade para utilizar os recurso do C++11, atrasando bastante o progresso na implementa  o dos m todos idealizados, sendo necess rio buscar solu  es alternativas e menos eficientes. Por fim acreditamos ter encontrado o motivo da falta dos recursos do C++11 devido a uma pequena troca nas op  es de compila  o, onde estava sendo utilizada a seguinte sequencia:

```
g++ -Wall -pedantic -std=c++11 -ansi
```

Com o -ansi localizado ap s o -std=c++11, ap s termos compilado cada objeto .o individualmente e nenhum erro ter sido encontrado. Decidimos refazer completamente o makefile, consertando o problema. Por fim as op  es de compila  o ficaram nessa ordem:

```
g++ -Wall -pedantic -ansi -std=c++11
```

## **O que não foi feito**

Devido aos atrasos encontrados, não tivemos tempo de implementar as devidas exceções que podem ocorrer, assim com realizar os devidos testes de situações onde essas exceções podem aparecer, algumas classes faltaram ser revisadas, e a documentação também não foi completada, sendo apenas alguns métodos e classes devidamente documentados.

## **Considerações Finais**

Acreditamos que mesmo o programa estando funcional, ainda está muito “bruto”, podendo ser grandemente melhorado, e muitos trechos simplificados por meio de funções, evitando a repetição do código.

Também deveríamos ter tentado implementar as exceções juntamente com os métodos onde elas poderiam aparecer, mesmo que não conseguíssemos entregar um programa funcional teríamos compreendido melhor o uso das mesmas.

Por fim, gostaríamos de ter organizado melhor o código, porém conciliar os estilos de programação dos dois integrantes foi um processo difícil, assim não foi utilizado um padrão para o código.