UNIVERSIDADE DE VASSOURAS – CAMPUS MARICÁ

ENGENHARIA DE SOFTWARE

HUGO LELLY DE LIMA MARINHO

ESTRUTURA DE DADOS

P2 - ENUNCIADO E REGRAS

*MARICÁ, RJ*

*2023*

INFORMAÇÕES:

CURSO:

**ENGENHARIA DE SOFTWARE**

DISCIPLINA:

**ESTRUTURA DE DADOS**

PROFESSOR:

**MÁRCIO GARRIDO**

TÍTULO:

**P2 - ENUNCIADO E REGRAS**

TURMA:

**2022.1 – TURMA A**

MATRÍCULA:

**202211182**

ALUNO:

**HUGO LELY DE LIMA MARINHO**

“A educação tem raízes amargas, mas os seus frutos são doces.”  
Aristóteles, filósofo grego.

SUMÁRIO

[INFORMAÇÕES: 1](#_Toc136282005)

[Sumário 2](#_Toc136282006)

[ENUNCIADO: 3](#_Toc136282007)

[REGRAS DE AVALIAÇÃO: 4](#_Toc136282008)

[FLUXOGRAMA: 5](#_Toc136282009)

[CODIGO PYTHON: 6](#_Toc136282010)

ENUNCIADO:

A Universidade de Vassouras do Campus 1 foi convidada pela Prefeitura de Maricá para promover uma solução tecnológica em um dos problemas sociais da cidade, o abandono de animais. Mesmo considerado crime (O abandono de animais é crime, previsto na Lei de Crimes Ambientais - Lei Federal n° 9.605 de 1998), e notório que o índice de abandono vem crescendo a cada ano.

Os alunos do curso de Engenharia de Software foram convocados para a reunião com a secretaria da cidade para entender a demanda solicitada e alguns pontos foram levantados.

A prefeitura precisa de um sistema que possa cadastrar todos os animais por tipo (canino, felino etc.) e para tanto, é uma premissa que seja possível inserir novos tipos dinamicamente.

Precisa ainda, que sejam classificados por idade aproximada, cor, porte e se possui alguma particularidade.

No mesmo sistema, deverá ter também um cadastro de pessoas interessadas na adoção, contendo os dados principais de contato e qual espécie teria o interesse de adotar.

Ao escolher a espécie, deve também informar se possui alguma preferência do animal.

Por fim, no final do mês a prefeitura emitirá um relatório de cruzamento de espécies disponíveis x possíveis candidatos, ou quando um candidato a adoção ligar, que o atendente possa pesquisar se há algum animal com as características informadas.

Os alunos anotaram atentamente a todas as observações, criaram o fluxograma do estudo de caso, e posteriormente o primeiro protótipo em Python, ainda que em modo texto, e sem requisitos gráficos.

A ideia foi apenas validar a proposta do programa junto ao solicitante.

REGRAS DE AVALIAÇÃO:

* **FLUXOGRAMA EM PDF NO REPOSITÓRIO (1 PONTO)**
  + Organização clara **(0,5 Pontos)**
  + Funcional de acordo com o enunciado **(0,5 Pontos)**
* **ESTRUTURA DE DADOS DO ALGORITMO (4 PONTOS)**
  + Uso de ao menos 4 métodos de fila ou pilha **(0,5 Ponto)**
  + Uso de ao menos 4 métodos recursivos **(0,5 Pontos)**
  + Uso de pesquisa binária, lista encadeada **(0,5 Pontos)**
  + O programa rodar com tratamento de erros - "entradas inválidas do usuário" **(0,5 Pontos)**
  + O programa atender ao enunciado proposto **(2 Pontos)**
* **ORGANIZAÇÃO DO PROJETO (3 PONTOS)**
  + Hierarquia dos arquivos e organização das pastas - diretórios, nome de arquivos, classes, etc. **(1 Ponto)**
  + Relação commit/dia com no mínimo 50 commits no final do projeto e no máximo 5 commits dia **(1 Ponto)**
  + Organização do README do projeto contendo título do projeto, enunciado, participantes, nome do professor (linkando para meu GITHUB), disciplina e ao menos 3 imagens do fluxograma, código e o programa rodando. **(1 Ponto)**

FLUXOGRAMA:

CODIGO PYTHON:

O Sistema de Adoção foi dividido em 9 arquivos, para que cada função possa passar por manutenções futuras.

* menu.py

Interage com todos os códigos, chamando as funções de cada código de acordo com a necessidade.

* verificararquivos.py

Verifica a existência dos arquivos em “animais.txt” e “candidatos.txt”, caso o arquivo não exista, ele gera um novo arquivo.

* limparterminal.py

Chama a função para limpar o terminal, fazendo com que o código fique limpo no terminal.

* cadastraranimal.py

Realiza o cadastro dos animais seguindo os critérios solicitados e salva as informações no arquivo “animais.txt”.

* cadastrarcandidato.py

Realiza o cadastro dos candidatos, seguindo os critérios solicitados e salva as informações no arquivo “candidatos.txt”.

* consultaranimais.py

Realiza a consulta dos animais cadastrados, trazendo a opção preenchida ou todos os animais, com a informação do número de animais que atende a consulta.

* consultarcandidatos.py

Realiza a consulta dos candidatos cadastrados, trazendo a opção preenchida ou todos os candidatos, com a informação do número de candidatos que atende a consulta.

* combinarinteresses.py

Exibe os animais e candidatos que possuem compatibilidade.

* exibirsobre.py

Na função sobre, e printado no terminal os dados do trabalho acadêmico.