

# Trabalho Prático - Algoritmos e Estrutura de Dados II

June 1, 2023

## 1 O que deve ser implementado

- Um programa que leia um arquivo texto (no formato .txt) e colete todas as palavras presentes no texto.
  - A leitura do arquivo deverá desprezar espaços em branco e sinais de pontuação, que serão considerados separadores de palavras. Além disso, a leitura deverá converter todas as letras maiúsculas em minúsculas.
- Uma árvore binária AVL (implementada pelo próprio aluno na linguagem JAVA).
  - Além dos atributos elementares da árvore AVL, os nós deverão conter um Objeto “Palavra” com:
    - \* Nome (tipo String);
    - \* Contador (Integer).
- Uma lista dupla encadeada formada por objetos do tipo “Palavra”.

Em seguida, utilizando a estrutura desenvolvida, crie uma árvore AVL que armazene todos os nomes contidos no arquivo texto. Na árvore, as palavras devem ser ordenadas por ordem lexicográfica. Durante a inserção, uma vez identificado uma determinada palavra que já existe na estrutura, o atributo contador da mesma deverá ser incrementado. Assim, ao final de uma execução, sua estrutura deverá conter todas as palavras do texto e a quantidade de vezes que cada uma delas aparece.

Por fim, utilizando alguma estratégia de caminamento na árvore, percorra todos os nós e adicione as “Palavras” encontradas na sua lista dupla encadeada. Você deverá inserir as palavras na lista de modo que aquelas que mais aparecem no texto estejam no início e as que menos aparecem fiquem no fim.

## 2 Testando o seu programa

Crie uma classe Principal que leia um arquivo de texto seguindo as especificações apresentadas na seção anterior. Monte a sua árvore AVL e imprima todas as palavras contidas no texto em ordem lexicográfica. Monte também a sua lista dupla (também seguindo as especificações da seção anterior) e, em seguida, imprima os nomes em função do atributo contador em ordem decrescente.

- **Exemplo de arquivo de texto de entrada:**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

- **Exemplo de impressão das palavras em ordem lexicográfica (Impressão da lista):**

ad: 1  
aliqua: 2  
aliquip: 1  
aute: 1  
commodo: 1  
consectetur: 1  
consequat: 1  
dolore: 4  
dolor: 3

- **Exemplo de impressão das palavras em ordem decrescente pela quantidade de vezes que cada uma delas aparece no texto:**

dolore - 4 vezes  
ipsum - 4 vezes  
ut - 3 vezes  
in - 2 vezes  
eiusmod - 2 vezes

### 3 O que deverá ser entregue?

- O projeto JAVA contendo toda a implementação com cada código fonte bem organizado e comentado.
- Um arquivo .txt único contendo TODO o código fonte (Copie cada classe implementada e cole neste arquivo).
  - **OBS: Este arquivo (arquivo .txt) será utilizado para testar o seu programa e o programa de outras duplas.**
- Documentação do trabalho com:
  - **Descrição sobre a implementação do programa.** Não faça “print screens” de telas. Ao contrário, procure resumir ao máximo a documentação, fazendo referência ao que julgar mais relevante. É importante, no entanto, que seja descrito o funcionamento das principais funções e procedimentos utilizados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado.
  - **A documentação deverá ser entregue no formato PDF.**

## 4 Informações sobre a avaliação

- O trabalho vale **15 pontos**;
- O trabalho poderá ser realizado em dupla;
- Cada aluno da dupla será avaliado individualmente;
- A dupla terá um tempo de 10 minutos para apresentar em laboratório o código e a funcionalidade do programa;
- A pontuação será calculada em função dos documentos entregues e da apresentação.