

## Trabalho Final – Verificador Ortográfico

INF 01203 – Estruturas de Dados

## 1. Introdução

Este trabalho tem por objetivo proporcionar aos alunos a oportunidade de aplicar os conhecimentos adquiridos e as estruturas de dados desenvolvidas em aula na solução de um problema que utilize várias dessas estruturas.

É esperado que o aluno desenvolva uma biblioteca contendo todas as funções de manipulação das estruturas e algumas funções de aplicação.

O trabalho será realizado em **duplas**. O trabalho deverá ser apresentado pelos dois integrantes da dupla.

## 2. Cenário

A aplicação modelada corresponde à identificação dos erros ortográficos mais frequentes em um dado texto. São fornecidos:

- arquivo texto referente ao dicionário;
- arquivo texto referente ao texto a ser analisado;
- uma constante  $k$ , que indicará os erros mais frequentes.

A aplicação deverá listar os erros mais frequentes encontrados no texto.

Principais definições:

- **palavra** - uma sequência de letras. Todos os outros caracteres deverão ser considerados como espaços separadores de palavras. Deve-se desprezar diferenças entre letras maiúsculas e minúsculas;
- **erro ortográfico** - uma palavra que não esteja contemplada no dicionário de palavras fornecido.
- **ordem lexicográfica** - é análoga à ordem das palavras em um dicionário. Ela se baseia na ordenação dos caracteres estabelecida na tabela ASCII, da mesma forma que a ordem das palavras em um dicionário se baseia na ordenação das letras no alfabeto.

## 3. Funcionamento da Aplicação

Implementar uma aplicação que liste os  $k$  erros ortográficos mais frequentes de um dado texto.

## Dados de Entrada:

- arquivo texto referente ao dicionário (pegue alguns dicionários para teste: [inglês](#), [português](#));
- arquivo texto;
- constante  $k$  (zero indica que todos os erros devem ser exibidos).

## Dados de Saída:

- listagem dos  $k$  erros mais frequentes, por ordem decrescente de frequência e, dentro desta, por ordem lexicográfica. A listagem deverá ter o seguinte formato (exemplo com  $k=4$ )

| Número (K) | Erro         | Frequência | Ocorre nas linhas           |
|------------|--------------|------------|-----------------------------|
| 1          | thhe         | 10         | 10 30 50 55 78 92* 100* 125 |
| 2          | uiversity    | 7          | 9 10* 88* 200               |
| 3          | andd         | 5          | 2 30* 95 160                |
| 4          | techhnologye | 5          | 10 55 80 100 120            |

O símbolo \*, após um número de linha, indica que existe mais do que uma ocorrência do erro nessa linha.

O programa também deve mostrar na tela os **tempos gastos** nas seguintes operações:

- carga do dicionário para a estrutura de dados;
- busca por erros.

## 4. Datas Importantes

- **13/11/09**: cada dupla deve submeter dois textos para testes (um em inglês e outro em português).
- **09 e 10/12/09** apresentação (horário da aula) e entrega dos fontes documentados.

## 5. Critérios de Avaliação

O trabalho deve ser realizado em duplas e deverá ser apresentado e defendido na data prevista.

Para a avaliação serão adotados diversos critérios:

- tempo gasto na carga da estrutura;
- escolha das estruturas de dados envolvidas;
- organização e documentação da biblioteca;
- organização, documentação e interface do programa de aplicação.

O programa deve indicar os tempos gastos na criação da estrutura e o tempo gasto em cada busca.

A escolha das estruturas de dados deve demonstrar conhecimento teórico e prático buscando a melhor combinação que atinja os resultados satisfatoriamente. Esse trabalho não avalia apenas o desempenho, mas a capacidade do aluno de criar estruturas elegantes e fáceis de serem mantidas. Para avaliar esse critério, é muito importante que o aluno **DESCREVA COM RIQUEZA DE DETALHES** as estruturas utilizadas no programa.

Por fim, serão avaliados a organização da biblioteca e do programa. A interface do programa também será avaliada. Uma interface atraente e fácil de usar é muito importante nesse tipo de software.

## Observações:

- O plágio é terminantemente proibido e a sua detecção incorrerá na divisão da nota obtida pelo número de alunos envolvidos. Para detectar o plágio, usaremos o software MOSS (<http://www.cs.berkeley.edu/~aiken/moss.html>).