



UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA



ERICA CRIZOLGO JACINTO

TRABALHO PRÁTICO DA DISCIPLINA DE
PROGRAMAÇÃO E DESENVOLVIMENTO DE
SOFTWARE I

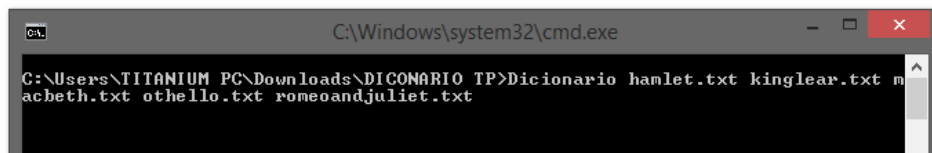
Belo Horizonte

2017

Introdução

O programa desenvolvido cria um índice invertido a partir de entradas passadas pelo usuário de arquivos de extensão txt. Após o usuário entrar com a relação e arquivos que deverão ser utilizados para criar o Dicionário, o programa busca os arquivos no local e cria um objeto da class Dicionario, Existe um menu com as opções: consultar se uma palavra existe ou sair do programa.

A compilação do programa pode ser realizada por linha de comando, no sistema operacional Windows através do Prompt de Comando (cmd.exe), executando o aplicativo Dicionario seguido pelos nomes dos arquivos que serão lidos com a extensão .txt, conforme imagem abaixo:



```
C:\Windows\system32\cmd.exe
C:\Users\TITANIUM PC\Downloads\DICONARIO TP>Dicionario hanlet.txt kinglear.txt m
acheth.txt othello.txt romeoandjuliet.txt
```

Caso algum dos arquivos informados pelo usuário não exista ou seja inválido, o programa irá abrir os arquivos válidos, exibirá para o usuário a mensagem: “FALHA EM ABRIR AQUIVO: ARQUIVO.txt”, e continuará a execução do programa, conforme exemplo abaixo:



```
C:\Windows\system32\cmd.exe - Dicionario kinglear.txt othello.txt AQUIVO.txt ...
C:\Users\TITANIUM PC\Downloads\DICONARIO TP>Dicionario kinglear.txt othello.txt
AQUIVO.txt macheth.txt
FALHA EM ABRIR ARQUIVO: AQUIVO.txt
INDICE INVERTIDO
OPÇÕES:
1. CONSULTA
2. SAIR
```

O tratamento de exceção é realizado na abertura dos arquivos a serem lidos pelo programa, o método *void Dicionario::Abrir(char *arquivo)* contém um throw, utilizado para esse tratamento no método *void Dicionario::Gerar_Dicionario(int quantidade_arquivos, char *argv[])*.

Desenvolvimento

Os arquivos passados pelo usuário são abertos, e cada linha é lida pelo programa, e enquanto não chegar ao fim da leitura de todas as linhas, ocorre a quebra da linha em palavras, onde cada palavra é normalizada, ou seja, são desconsiderados os caracteres que não letras ou números, e as letras maiúsculas são substituídas por minúsculas.

A *class Dicionario* possui um vector de Struct onde são salvas as palavras e a respectiva lista de arquivos onde a palavra foi encontrada. Após a palavra ser normalizada, o método *void Dicionario::Insere(Palavra nova)* e busca se a palavra já existe no Dicionário, se não existir, a palavra é adicionada e o nome do arquivo onde a palavra está é adicionado no vector *lista_arquivos* na Struct.

Caso já exista a palavra, é realizado um *sort* na lista de arquivos que contém a palavra, e verifica-se através da *binary_search* se o nome do arquivo já está na lista, se não estiver, ele é incluído.

O método que busca a palavra realiza uma busca linear em todo o vector de *Struct* de palavras, retornando o índice da palavra caso ela exista, se não, retorna -1.

A busca que é realizada quando o usuário consulta se uma palavra existe no Dicionário é feita pelo método *void Dicionario::Consulta(string entrada)*, utilizando o método *int Dicionario::Busca(Palavra nova)*, que retorna -1 caso não encontre a palavra, juntamente com a mensagem "PALAVRA NAO ESTA PRESENTE NO DICIONARIO.", que é impressa na tela para o usuário.

Se a palavra consultada existir, o método *void Dicionario::Imprimir(int indice)* recebe o índice da palavra, e imprime a lista de arquivos que contém a palavra.

Por fim, após o método *void Dicionario::Gerar_Dicionario(int quantidade_arquivos, char *argv[])* abrir e ler todas as linhas do arquivo aberto, fecha-se o arquivo e um *sort* irá ordenar o vector de Struct *dicionario_*.

O *sort* é utilizado para ordenar as listas para que a *binary_search* seja executada de forma eficiente.

Testes realizados

Para os testes foram utilizados os arquivos hamlet.txt, kinglear.txt, macbeth.txt, othello.txt e romeoandjuliet.txt, que contém as obras de William Shakespeare. Após gerar o Dicionário, foram consultadas algumas palavras, conforme imagens:

```
C:\Windows\system32\cmd.exe - Dicionario hamlet.txt kinglear.txt macbeth.txt...
C:\Users\TITANIUM PC\Downloads\DICONARIO TP>Dicionario hamlet.txt kinglear.txt m
acheth.txt othello.txt romeoandjuliet.txt
                                INDICE INVERTIDO
OPCOES:
1.CONSULTA
2.SAIR
1
INSIRA PALAURA: heart
hamlet.txt
kinglear.txt
macbeth.txt
othello.txt
romeoandjuliet.txt

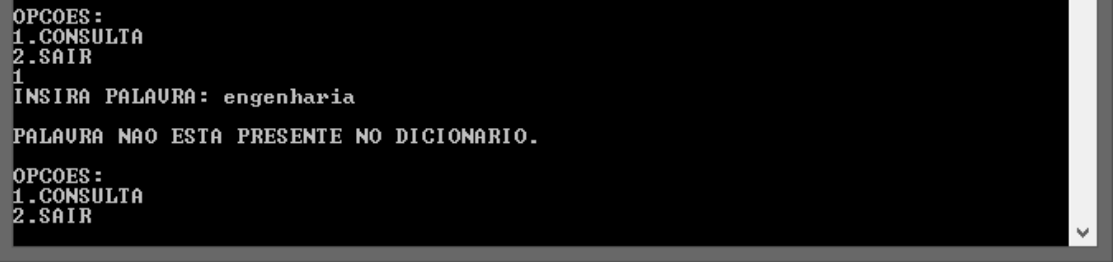
OPCOES:
1.CONSULTA
2.SAIR
-
```

```
OPCOES:
1.CONSULTA
2.SAIR
1
INSIRA PALAURA: HEART
hamlet.txt
kinglear.txt
macbeth.txt
othello.txt
romeoandjuliet.txt
```

```
OPCOES:
1.CONSULTA
2.SAIR
1
INSIRA PALAURA: HE*art
hamlet.txt
kinglear.txt
macbeth.txt
othello.txt
romeoandjuliet.txt
```

Mesmo que a palavra pesquisada possua algum caractere especial entre as letras, a palavra será normalizada para ser consultada no dicionário, considerado que quando é executada a leitura do arquivo de entrada, a palavra é normalizada, deixando todas as letras minúsculas, e os caracteres entre as letras são removidos, como exemplo a palavra guarda-chuva, é inserida no dicionário como guardachuva, logo, quando o usuário realizar a consulta da palavra com o

hífen, só será possível encontra-la no dicionário se o hífen for desconsiderado, sendo assim, é necessário a normalização da palavra a ser buscada. Ocorrendo a busca de uma palavra que não está presente no dicionário, uma mensagem é informada ao usuário.



```
OPCOES:  
1.CONSULTA  
2.SAIR  
1  
INSIRA PALAURA: engenharia  
PALAURA NAO ESTA PRESENTE NO DICCIONARIO.  
OPCOES:  
1.CONSULTA  
2.SAIR
```

Bibliografia

<http://www.cplusplus.com/reference/algorithm/sort/>

http://www.cplusplus.com/reference/algorithm/binary_search/

<http://www.cplusplus.com/reference/string/string/>

http://www.cplusplus.com/reference/vector/vector/push_back/

<http://www.cplusplus.com/reference/string/string/compare/>

<http://www.cplusplus.com/reference/fstream/ifstream/>

<http://www.cplusplus.com/doc/tutorial/files/>

<http://www.cplusplus.com/doc/tutorial/exceptions/>