**UNIVERSIDADE FEDERAL DE MINAS GERAIS**

**Trabalho prático Máquina de Busca**

**Alunos:**

Geam

Júlia Miranda (2016026370)

Kelly

**Sumário**

Introdução

O código

Implementação

Testes de Unidades

Bibliografia

**Introdução**

Máquina de busca (também chamada de motor de busca, ferramenta de busca ou buscador) é um programa que objetiva encontrar palavras-chaves fornecidas pelo usuário em documentos ou base de dados. É uma importante e indispensável ferramenta no meio de acesso à informação.

O presente trabalho tem como proposta o desenvolvimento de uma máquina de busca que fará a leitura de diversos arquivos previamente fornecidos e, então receberá do usuário uma palavra-chave. O programa fará a busca pela palavra fornecida pelo usuário e apresentará os arquivos mais relevantes, conforme um ranking que será apresentado posteriormente.

Conforme orientações fornecidas para a realização deste trabalho prático, algumas preocupações tiveram que ser tomadas, tais como o tratamento dos textos lidos, a forma que o programa deve *rankear* os resultados obtidos para o usuário. Essas preocupações serão mais bem detalhadas posteriormente.

**O código**

O trabalho realizado pode ser entendido por três partes bem definidas. A primeira consiste na leitura dos arquivos e na preparação dos seus dados, de modo que é convertido numa string ideal na qual é realizada a busca. Essa etapa foi desenvolvida por meio de uma estrutura de dados. A segunda parte consiste na busca propriamente dita e na classificação dos arquivos associados à busca. A terceira parte consiste na realização de testes de unidades, em que os métodos desenvolvidos são testados, cujo intuito é maximizar a assertividade do código.

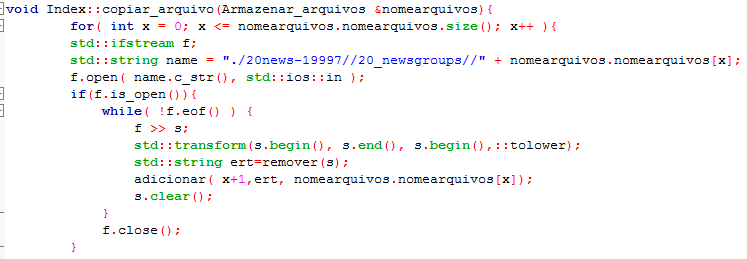
**Implementação**

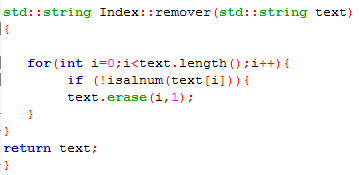
A princípio, o sistema recebe como entrada um conjunto de arquivos de texto. Primeiramente esses arquivos são abertos e lidos, palavra por palavra e então é construído o índice invertido.

Ao realizar a leitura dos arquivos, os dados lidos são devidamente tratados, o que otimiza o desenvolvimento das funções associadas à busca. Abaixo estão listados os tratamentos realizados:

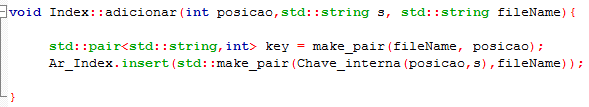
* são considerados apenas caracteres que são letras, números e pontuação;
* são desconsiderados acentuações e “cê-cedilha” (ç);
* as letras maiúsculas são transformadas em minúsculas;
* são apagados todos os caracteres que não são letras ou números.

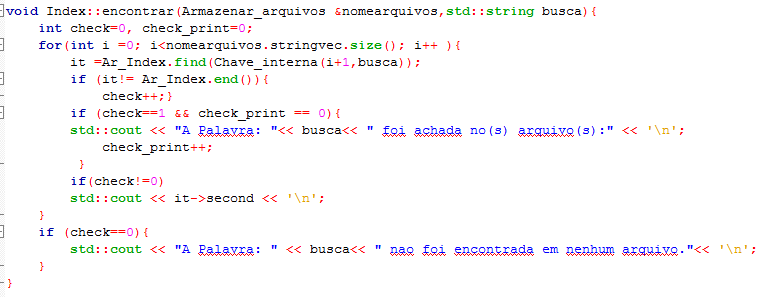
Os três primeiros tratamentos citados ocorrem na função “copiar arquivo”, por meio de funções simples da linguagem. O quarto ocorre na função “remover”

Função copiar arquivo

Função remover

Na sequência, todo o arquivo lido e tratado é armazenado em uma string e então, é construído o índice invertido. Este é implementado por meio da estrutura de dados do tipo dicionário “map”. Dicionários são estruturas associativas que armazenam elementos combinando uma chave com um valor. As chaves são todas as palavras contidas nos documentos e o valor associado a cada chave é o conjunto com o nome dos documentos.

Função adicionar



Posteriormente, uma vez que o usuário realiza a busca, é realizada uma classificação dos arquivos que contêm as palavras buscadas, de acordo com a importância do arquivo para a busca.

Essa classificação é realizada por meio do método cosine ranking.

INSERIR IMAGENS QUE ILUSTRAM O MÉTODO COSINE RANKING.

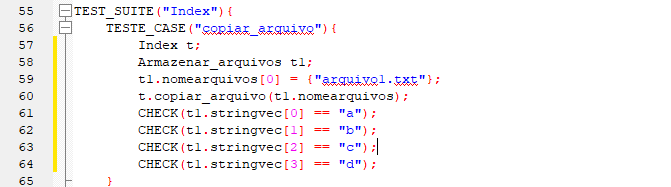
**Testes de Unidades**

Os Testes de unidade têm como objetivo testar cada fase do programa individualmente. Neste TP foram feitos os testes das funções de acordo com sua classe e, para isto, foi incluída a biblioteca chamada “doctest.h”.

**CLASSE INDEX**

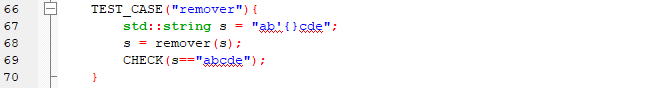
- Teste da função “copiar\_arquivo”

Esta função consiste em abrir os arquivos mapeados e chamar a função “adicionar” para cada palavra lida dos arquivos. O teste consiste na criação de um objeto do tipo “Index” e de um objeto do tipo “Armazenar\_arquivos”. Depois disso é chamada a função “adicionar” em que são passados seus parâmetros e é verificado se o texto que estava no arquivo foi escrito no vetor “stringvec”.



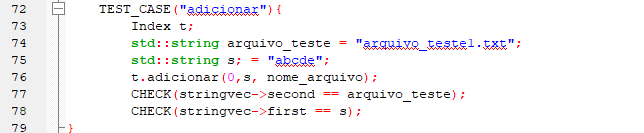
- Teste da função “remover”

Esta função tem como objetivo remover todos caracteres inválidos para a leitura do arquivo. Para testá-la, foi criada uma string com alguns caracteres inválidos. Posteriormente é passada esta string como parâmetro para a função e em seguida, é checado se a string ainda contém os caracteres inválidos.



- Teste da função “adicionar”

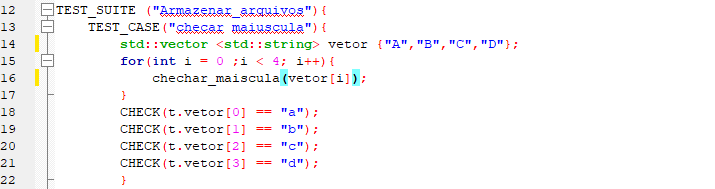
Esta função consiste em adicionar determinada palavra, lida no arquivo, em um vector “stringvec”. O teste consiste na criação de um objeto do tipo “Index” e de uma string. É passado como parâmetro esta string para a função adicionar e depois disso, é checado se a string foi adicionada ao vector “stringvec”.



**CLASSE ARMAZENAR\_ARQUIVOS**

- Teste da função “checar maiúscula”

Esta função tem como objetivo transformar as letras maiúsculas de uma string em letras minúsculas. Para testá-la, foi criado um vector com algumas strings que, por sua vez, é passado como parâmetro e posteriormente checado para averiguar se as strings estão com letras minúsculas.



- Teste da função “nome\_diretórios”

Esta função tem como objetivo ler o nome dos diretórios dos arquivos. Para testá-la, foi criado um vector que contém as strings com todos os nomes dos diretórios. O teste consiste em comparar cada string deste vector com o diretório do arquivo lido, e posteriormente, em conferir se alguma dessas strings correspondem ao diretório do arquivo lido.



**Bibliografia**

<https://pt.wikipedia.org/wiki/Motor_de_busca>

<http://www.linhadecodigo.com.br/artigo/1494/aprendendo-a-documentar-o-seu-codigo.aspx>

<https://homepages.dcc.ufmg.br/~glpappa/aeds2/exemplodoc.pdf>

<https://www.cplusplus.com>