UNIVERSIDADE FEDERAL DE MINAS GERAIS – UFMG

Trabalho Prático de Programação e Desenvolvimento de Software 2
Máquina de Busca

Alunos:

Bruno Rafael Martins Alves - 2018106753

João Vítor David Prates - 2018099352

Matheus Cascalho dos Santos – 2018014697

Turma: TF1

Professores: Thiago Ferreira de Noronha, Lucas Victor Silva Pereira.

Belo Horizonte

2019/1

Introdução

O nosso software, denominado Máquina de Buscas, possui um funcionamento que se baseia em uma interação indireta entre um usuário e um determinado banco de dados no qual estão inseridos documentos diversos que podem ser acessados pelo usuário através de uma busca realizada por ele.

O intuito de Máquina de Busca é entregar ao usuário o documento que mais se aproxima da pesquisa realizada pelo mesmo. Cada palavra nos documentos do banco de dados é lida e depois guardada na forma de um vetor, de modo que, as suas coordenadas dependem de fatores como a quantidade de repetições em cada documento e a quantidade de documentos nos quais essa aparece. Esses vetores são determinantes para o funcionamento do software já que, esse realiza diversos cálculos com base nos vetores, e criam uma ordem de importância de cada palavra em cada arquivo para determinar quais palavras representam melhor cada documento.

Implementação

Máquina de Busca tem como característica fundamental sua grande modularização. Essa possui um sistema de hierarquia direta, partindo de princípios básicos de leitura de arquivos até uma estrutura de armazenamento de dados complexa.

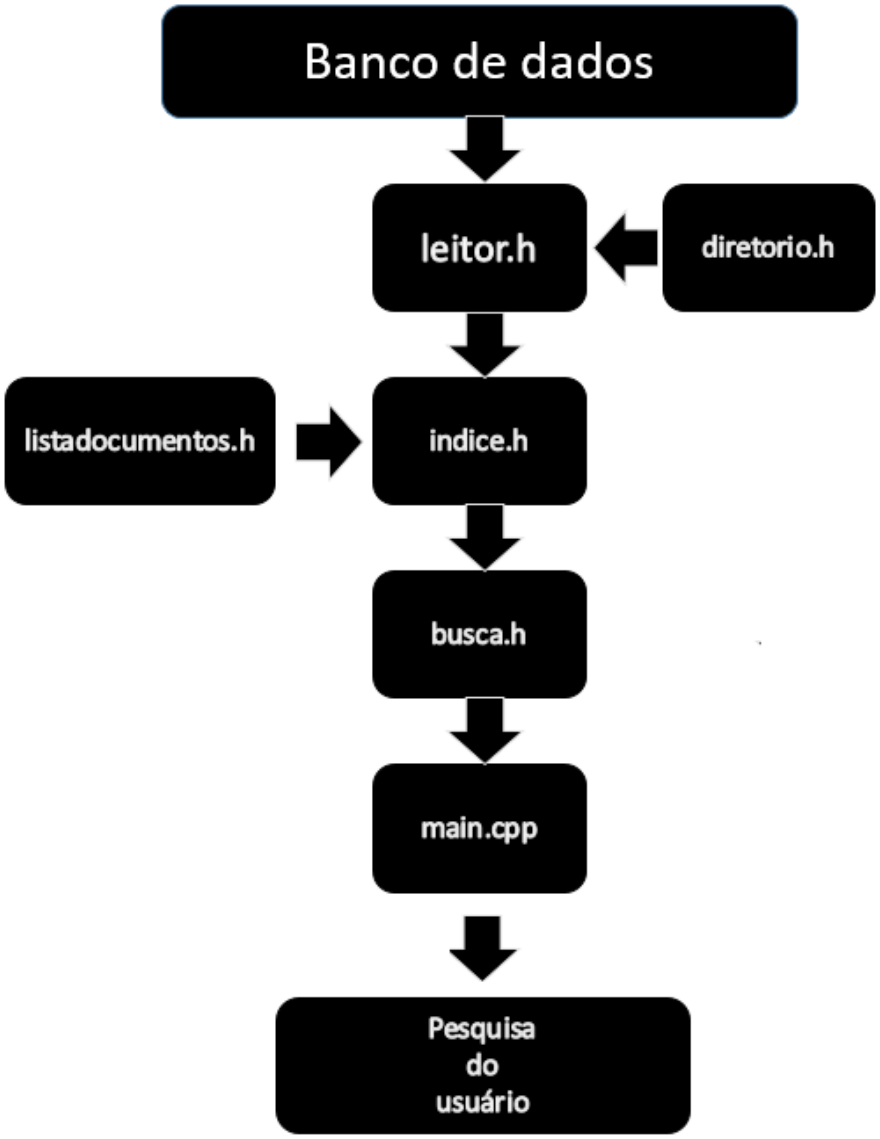
O software inicia com o módulo leitor.h e diretório.h. A função destes, de modo geral, é basicamente, ler os arquivos presentes no banco de dados e listar o conteúdo deles.

Os próximos módulos a serem utilizados são listadedocumentos.h e indice.h. O objetivo da união deles é criar um índice invertido, esse que tem como objetivo catalogar, em cada palavra do banco de dados, em quais documentos ela se fez presente, e quantas vezes se fez presente.

Feito o índice invertido, o próximo passo será entrar no módulo busca.h, cuja função é transformar cada palavra em um vetor o qual o módulo e a direção dependeriam, além de outros fatores, dos elementos do índice invertido. Com esses vetores, e a busca feita pelo usuário, podemos ranquear quais documentos são melhores representados pela palavra buscada.

O usuário do programa não tem acesso a nenhum desses passos, mas sim ao que será executável no módulo main.cpp, onde ele poderá digitar sua busca e terá como retorno a ordem dos documentos representados por essa palavra. Outra opção que o executável dá ao usuário é alterar o diretório do banco de dados.

Para uma melhor explanação da posição de cada módulo, podemos conferir o organograma abaixo, em que as setas representam a relação de dependência entre os elementos que compõe Máquina de Busca.



Com o intuito de gerar um melhor entendimento a respeito da forma como foi implementada a Máquina de Busca, os módulos serão explicados separadamente em forma de tópicos, seguindo a ordem de funcionamento do software.

- **diretorio.h**

Quando o software é inicializado, um dos primeiros módulos a serem utilizados é o diretorio.h cujo principal objetivo é concatenar nomes de arquivos e seus respectivos diretórios para facilitar uma posterior leitura do banco de dados que acontecerá em um módulo diferente desse. O principal conteúdo de diretorio.h é a classe denominada Diretorio cujas variáveis representam os nomes dos documentos, seus respectivos diretórios e a união de ambas as coisas, sendo todas essas armazenadas em forma de strings. Além disso, a classe conta com métodos que permitem e facilitam a manipulação dessas variáveis.

O software possui um diretório padrão, que pode ser alterado, em que entrega o endereço de um arquivo texto denominado sumário o qual apresenta, como conteúdo, os nomes de todos os outros arquivos presentes no banco de dados. Esse sumário, para ser considerado padrão deve estar em uma pasta com o nome Diretorio no mesmo diretório do software.

- **leitor.h**

O único módulo diferente do main.cpp que não apresenta uma classe específica é leitor.h. O objetivo deste é, intuitivamente, ler arquivos do banco de dados.

Em geral, a leitura dos documentos presentes no banco de dados altera listas fornecidas comumente como parâmetro das funções, cujo conteúdo são, ora palavras lidas de um arquivo para uma posterior utilização em outros módulos, ora o nome dos arquivos do banco de dados lidos do sumário.

Outra funcionalidade de leitor.h é fazer o processo de padronização das palavras que são lidas dos arquivos. Para que esse papel seja realizado da maneira correta, foi implementada no leitor a função transformaString, que é capaz de formatar strings fornecidas por meio do uso da técnica de passagem por referência. E é graças a presença dessa função que as interações entre o leitor e a classe Indice podem ocorrer sem grandes complicações.

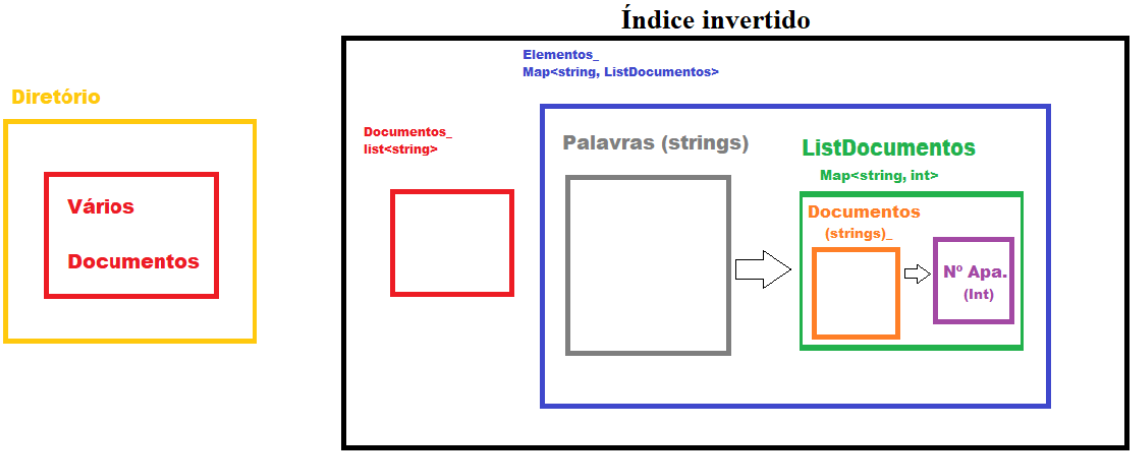
- **listdocumentos.h e indice.h**

Os módulos listdocumentos.h e indice.h são melhor entendidos se colocados juntos, já que listdocumentos.h é um dos complementos de indice.h

O módulo listdocumentos.h possui uma classe chamada ListDocumentos cujo principal objetivo é, para cada documento, guardar em forma de contador a quantidade de vezes que uma determinada palavra apareceu no conteúdo do documento.

O módulo indice.h possui como principal conteúdo a classe Indice. A principal função da classe é criar e guardar o índice invertido, sendo esse um map que recebe a palavra e a lista de documentos na qual essa palavra apareceu. O índice invertido será de suma importância para o software principalmente na parte do cálculo para ranquear as palavras que mais representam determinado documento.

Para um melhor entendimento do modo de operação do índice invertido, é possível atentarmos à figura abaixo.



Os módulos diferentes de indice.h que atuam ativamente neles são listdocumentos.h e leitor.h. ListDocumentos é o responsável por determinar em quais documentos a palavra buscada aparece. Já o leitor.h possui funções que podem alterar a variável todosDocumentos_ presente na classe Indice. Além disso, a função transformaString é elemento fundamental para padronizar as palavras e impedir erros no índice invertido.

- **busca.h**

O módulo busca.h é o módulo final que o software vai utilizar. Ele possui uma classe denominada Busca que possui diversas funções.

A primeira função da classe é receber a busca feita pelo usuário e retorná-la como um vetor de modo que, quando forem feitos os cálculos para ranquear as palavras, todos os vetores já estejam formados.

A segunda função da classe é realizar os cálculos para o ranking cosseno. Para isso, a classe utiliza alguns métodos que geram valores de frequência do termo nos documentos, a importância do termo para cada documento, as coordenadas, seja do documento, seja da expressão de busca e a similaridade entre elas. O cálculo da similaridade segue a seguinte fórmula:

$$sim(d_j, q) = \cos(\theta) = \frac{\sum_{i=1}^{t}(W(d_j, P_i) \times W(q, P_i))}{\sqrt{\sum_i W(d_j, P_i)^2} \times \sqrt{\sum_i W(q, P_i)^2}}$$

A terceira função da classe é retornar, ordenadamente, quais documentos são melhores representados pela busca feita pelo usuário.

- **main.cpp**

O objetivo de main.cpp é gerar um arquivo executável em que o usuário possa fazer a busca de uma palavra-chave e receber o resultado que mais se aproxima do que foi digitado.

A ideia central de main.cpp é ser fácil de utilizar, mas dar ao usuário liberdade de realizar algumas mudanças, como por exemplo, alterar o diretório onde se encontra elementos do banco de dados. A interface do executável é a mais clara e objetiva possível de modo que o usuário não tenha dificuldades para utilizar o software.

- **Funções de teste**

Máquina de Busca possui também módulos que cumprem a tarefa de testar os principais métodos e funções implementados. Esses testes, almejam eliminar os bugs e falhas que possivelmente poderiam aparecer com o uso do software.

Conclusão

Máquina de Busca pode ser caracterizado, como o próprio nome sugere, por um buscador simples. Esse, realiza a função de entregar ao usuário uma lista com os documentos que mais se aproximam das palavras-chave que o mesmo digitou.

O software apresenta um executável de fácil entendimento já que o mesmo tem a característica de ser objetivo, desse modo, o usuário irá enfrentar menos dificuldades para utilizá-lo. Além disso, Máquina de Busca fora testado por módulos amigáveis, com o intuito de apontar possíveis problemas, em sua implementação, que pudessem ser corrigidos. Por fim, Máquina de Busca se mostrou muito sólido no que se refere as possibilidades de não apresentar problemas, seja por mau uso do mesmo, seja por problemas em sua implementação, podendo ser considerado, portanto, como um software confiável.