MÁQUINA DE BUSCA

Bruno Rafael Martins Alves

João Vítor David Prates

Matheus Cascalho dos Santos

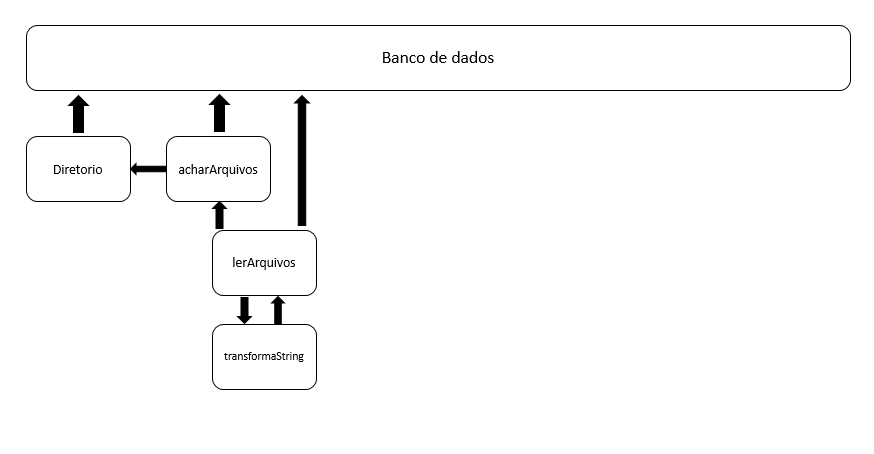
Kethellen da Silva Pereira

**Introdução**

Máquina de Busca é um software cujo funcionamento se baseia em uma interação indireta entre o usuário e um determinado banco de dados. O software armazena palavras lidas de um banco de dados formado por documentos diversos, na forma de vetor, e seu módulo é maior quanto menos vezes a palavra foi repetida em diferentes documentos. Através de cálculos, mede-se a importância de cada uma delas como representante de determinado documento. A interação entre o usuário e a máquina acontece a partir do momento em que o mesmo realiza a busca e uma ou mais palavras são interpretadas pela máquina de busca, como as palavras-chave. Assim, retorna-se ao usuário a lista de documentos ordenados, com base na representatividade dessas palavras.

**Implementação**

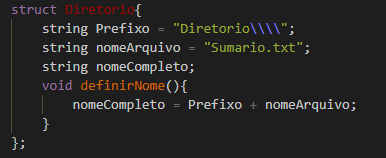
Para um melhor entendimento de como funciona a Máquina de Busca, é importante observarmos o organograma abaixo:



A partir da forma como cada função se relaciona é possível explicitar melhor como cada uma dessas foi implementada, bem como seus objetivos e pré-condições.

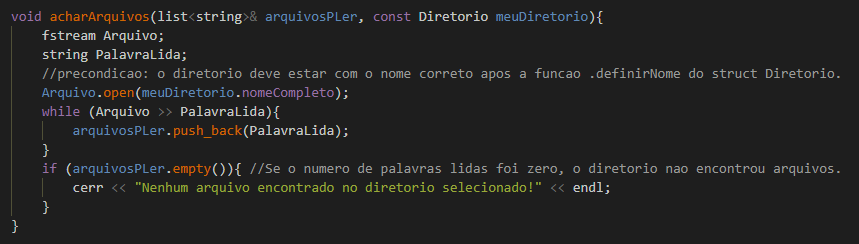
**Struct Diretorio**

A primeira ação realizada pela Máquina de Busca após sua inicialização é definir um diretório pelo qual os documentos serão lidos. Esse diretório é definido dentro da struct Diretório, como se observa abaixo:



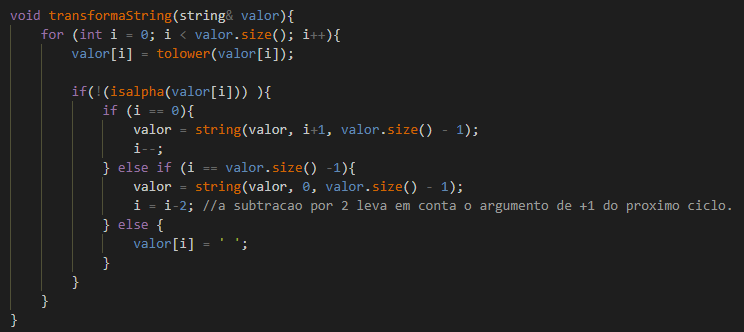
O objetivo é associar um Diretório pré-definido a um arquivo denominado Sumario.txt e transformar isso em uma string, que será utilizada por outras funções como auxiliar para a leitura dos arquivos. O Sumário.txt possui uma listagem com o nome de todos os outros arquivos do banco de dados da Máquina de Busca.

**Void acharArquivos**

****

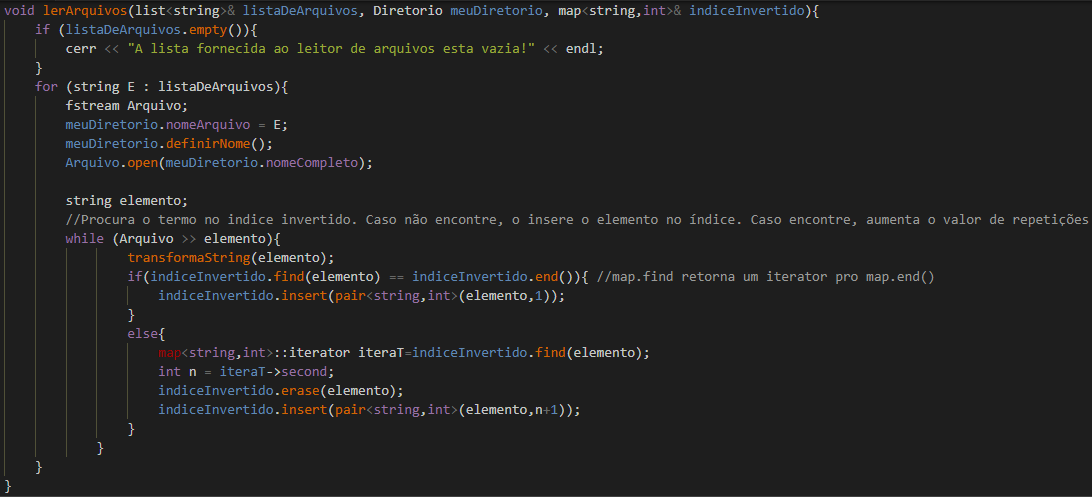
O objetivo dessa função é armazenar o nome dos arquivos do banco de dados em um list que será utilizado posteriormente pelo programa, para a leitura dos mesmos. A função possui a pré-condição de garantir que o diretório do Sumário.txt esteja correto. Por esse motivo, é importante que essa função seja chamada depois de definir o diretório através da struct Diretório. Caso o Sumário.txt esteja vazio, o programa retornará que o banco de dados está vazio.

**void transformaString**



O objetivo dessa função é padronizar determinada palavra, transformando-as em minúsculas e retirando ocasionais hifens.

**void lerArquivos**

****

A função ler Arquivos possui dois objetivos: ler os arquivos e criar um índice invertido. A função lê os arquivos listados anteriormente em um list, através da função acharArquivos e, enquanto lê, cada palavra vai sendo inserida em um índice invertido. Esse índice invertido é do tipo map e possui dois dados, sendo eles o nome da palavra e a quantidade de arquivos em que elas foram mencionadas. Antes de inserir cada palavra no índice invertido ou aumentar a quantidade de repetições, a função transformaString é chamada para padronizar as palavras.