Universidade Federal de Minas Gerais
Prog. e Desen. de Software II - Prof^{o} Thiago F. de Noronha

Trabalho Prático:

Máquina de Busca

Vinícius F. Campos

24 de novembro de 2018

1. Introdução

Índices invertidos são a base para o desenvolvimento de mecanismos de buscas tais como o Google e o Yahoo!. São estruturas contendo uma entrada para cada termo e a lista de arquivos que contém tal palavra. O objetivo deste trabalho é criar uma estrutura de índice invertido, indexar arquivos e habilitar a busca pelos termos contidos neles. A implementação foi feita em C++ e para os testes unitários foi utilizada a biblioteca *Boost.Test*.

2. Especificação do Problema

O mecanismo de busca desenvolvido neste trabalho permite a indexação de arquivos e a consulta pelos termos destes arquivos. Para tanto, uma lista de arquivos é carregada no sistema. Cada palavra do arquivo é tratada e gera um token correspondente de acordo com as seguintes regras:

- transformar todas as letras maiúsculas em minúsculas.
- apagar todos os caracteres que não são letras ou números

É gerado uma entrada no índice invertido para cada token e, associado a ele, a lista de arquivos em que há ocorrência do termo. Após indexação dos arquivos, o usuário pode fazer consultas por palavras. É gerado um token para o termo pesquisado e o índice invertido é acessado para recuperar em quais arquivos há uma ou mais ocorrências do termo.

3. Algoritmos e Linguagem utilizada

O trabalho foi desenvolvido em C++ no Windows. Esta escolha foi feita para melhor abstrair e modularizar o problema, baseando-se nos princípios da orientação à objetos.

Foi criada a seguinte classe:

1. **MecanismoDeBusca**: abstração do indice invertido, possui os métodos de indexação, tokenização e pesquisa. Possui uma propriedade privada do tipo *map<string, set<string>>* para gravar a associação do token e da lista de arquivos que o contém.
    - *IndexarArquivo*: recebe o caminho do arquivo a ser indexado, faz a leitura de seu conteúdo e insere cada palavra no índice invertido. Caso o arquivo não seja aberto com sucesso, uma exceção é lançada.
    - *TokenizarPalavra*: recebe uma string e gera o token contendo penas letras minúsculas e números.
    - *InserirEntrada*: recebe o token e o caminho do arquivo, gera uma entrada para o token caso não exista e associa ao arquivo caso já não tenha sido associado.
    - *Pesquisar*: recebe uma string, converte em token, realiza a busca no índice invertido e retorna o resultado.

4. Testes Unitários

Para a criação dos testes unitários foi utilizada a biblioteca *Boost.Test*. Entretanto, não foi possível compilar o projeto de teste pois a instalação no Windows não foi finalizada com sucesso. Os seguintes casos de teste foram implementados:

- *testa_se_indexacao_de_arquivos_nao_existentes_lanca_excecao*: testa lançamento de exceções caso o arquivo não seja aberto com sucesso.
- *testa_regras_para_tokenizacao*: testa as regras de transformar palavras em tokens contendo apenas letras minúsculas e números.
- *testa_se_pesquisa_pesquisa_sem_ter_indexado_nenhum_nao_lanca_excecao*: testa se não lança exceção se não tiver indexado nenhum arquivo.
- *testa_se_pesquisa_pesquisa_sem_ter_indexado_nenhum_arquivo_retorna_vazio*: testa se retorna vazio para termos não inseridos
- *testa_se_nao_grava_quantidade_duplicada_de_arquivos*: testa se não grava quantidade duplicada de arquivos.
- *testa_insercao_e_pesquisa_por_termos_contidos_em_mais_de_um_arquivo*: testa se grava mais de um arquivo corretamente.
- *testa_se_retorno_ja_esta_em_ordem_crescente*: testa se retorno da pesquisa é em ordem crescente.

5. Execuções e Testes

Para executar o projeto, utilize o comando abaixo:

g++ -g main.cpp mecanismo-de-busca.cpp mecanismo-de-busca.h

No exemplo da figura 1, verifica-se:

1. A lista de arquivos que foi indexada (livros de Shakespeare). Uma exceção é lançada ao abrir o último arquivo pois ele não existe. Entretanto, ela é tratada e não bloqueia o programa, o usuário apenas é informado sobre o erro.
2. Pesquisa por um termo que existe em apena um arquivo.
3. O termo pesquisado é tratado e retorna os mesmos valores para a busca sem caracteres inválidos.
4. Pesquisa por um termo que existe em mais de um arquivo. O resultado é em ordem alfabética apesar de a indexação não ter sido.
5. Pesquisa por números também é possível.
6. Para pesquisas por termos que não estão contidos nos arquivos indexados, a mensagem "Nenhum resultado foi encontrado para sua busca"é exibida.

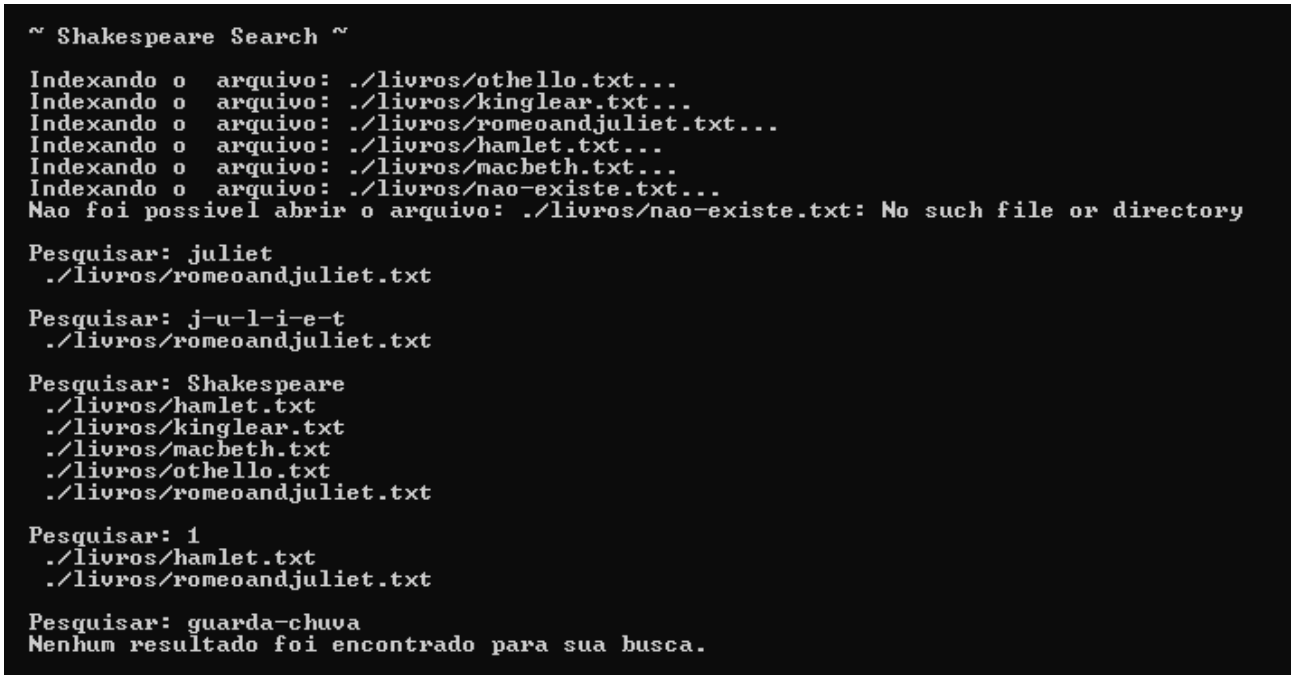

Figura 1: Exemplo em execução.