Sistema de Busca Indexada (SBI)
Relatório Técnico

Gabriel Araújo de Souza[1], Jaine Rannow Budke[2], Mayra Dantas de Azevedo[3]

[1]Instituto Metrópole Digital – Universidade Federal do Rio Grande do Norte (UFRN)
Caixa Postal 1524 – 59.078-970 – Natal – RN – Brazil

gabrieljucurutu@gmail.com, jainebudke@hotmail.com, mayradazevedo@gmail.com

Resumo. *Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.*

1. Introdução

O presente trabalho trata da apresentação de um sistema de busca indexada. Esse tipo de mecanismo funciona como um índice remissivo de um livro, no qual uma palavra é exibida juntamente às páginas em que aparece para facilitar o acesso ao termo no material. No caso deste sistema, ao buscar determinada palavra é apresentado uma lista de ocorrências desta informando o arquivo e as linhas em que ela pode ser encontrada. Tendo em mente as funcionalidades que são necessárias para esse tipo de sistema, foi implementado uma busca indexada em linguagem Java.

O sistema possibilita para o usuário inserir, remover e atualizar arquivos para compor uma base de dados, realizar buscas por uma única palavra ou por várias palavras simultaneamente considerando a lógica do tipo AND ou OR. Na AND, são listados apenas os arquivos que contém todas as palavras solicitadas na busca, na OR são exibidas todas as ocorrências das palavras buscadas.

O projeto foi desenvolvido levando em consideração conceitos aprendidos ao longo da disciplina Linguagem de Programação II e Estrutura de Dados Básicas II. Assim, foi utilizada da compreensão de Orientação a Objetos e Design de Aplicações, para a implementação do sistema, planejamento das funcionalidades e metodologia de desenvolvimento, bem como foi levado em consideração os tipos de árvores, apresentadas na disciplina de Estrutura de Dados, para o armazenamento das palavras, considerando a complexidade das operações que seriam realizadas sobre os dados.

Este relatório tem como objetivo apresentar o planejamento do projeto, as principais decisões tomadas, os resultados obtidos e discutir acerca dos problemas encontrados a partir da organização definida, bem como futuras implementações que podem ser feitas como melhorias do sistema.

2. Funcionalidades

2.1. Inserção de Arquivos

O sistema comporta dois tipos de inserção de arquivos, aqueles que são considerados na base de dados, ou seja, os que contém as palavras que os mecanismos de busca acessam, e os que são considerados uma lista negra de palavras (blacklist), esta contém palavras que devem ser ignoradas na pesquisa, como por exemplo, palavras de baixo calão. Quando o programa analisa uma palavra de determinado arquivo, é verificado se ela pertence a blacklist, se isto ocorrer, a palavra é descartada, caso contrário é considerada válida.

Quando um novo arquivo é inserido na base de dados, o programa realiza um processo de indexação, o que consiste em separar todas as palavras encontradas no arquivo, verificar se cada uma delas são válidas, e se for, a palavra é adicionada em uma árvore digital Trie armazenada em memória, as buscas acessam essa estrutura de dados, os resultados são mostrados ao usuário por meio de uma interface gráfica.

2.2. Listagem de Arquivos

O usuário tem acesso a uma interface que contém uma lista com todos os arquivos adicionados no sistema.

2.3. Remoção de Arquivos

O sistema permite ao usuário remover arquivos que foram anteriormente adicionados com as palavras que são consideradas pelo mecanismo de busca. Para tanto, por meio da interface gráfica de listagem dos arquivos, o usuário pode selecionar o arquivo que deseja remover e acionar o mecanismo de remoção.

Quando uma chamada de remoção é feita, o sistema busca todas as palavras associadas ao arquivo selecionado e remove os nós ou índices associados na árvore digital Trie, que possui o armazenamento das palavras.

2.4. Atualização de Arquivos

Quando um arquivo é adicionado uma vez ao sistema, não é permitido que seja adicionado novamente, ao menos que tenha sido excluído. Assim, caso o conteúdo do arquivo seja alterado, as novas palavras (se existiram) não serão incluídas na estrutura e, portanto, não serão vistas pelos mecanismos de busca.

Para isso, na interface de listagem dos arquivos o usuário pode selecionar um arquivo e acionar a atualização. Desta forma, o sistema verifica se há alguma palavra no arquivo que não está adicionada na estrutura de dados e a inclui.

2.5. Busca de Palavras

Há três tipos de busca que podem ser realizadas:
- Busca Simples
- Busca AND
- Busca OR

Há duas interfaces gráficas correspondentes à busca. Na inicial, o usuário tem um campo de texto no qual uma palavra pode ser digitada. Ao acionar o mecanismo de busca, a palavra é buscada na árvore digital Trie, levando em consideração o tipo Busca Simples e o resultado é listado logo acima do buscador.

Há uma interface gráfica de busca avançada, na qual a organização segue a mesma da inicial, contudo é permitido ao usuário a busca por mais de uma palavra, possibilitando acionar o tipo Busca AND ou Busca OR. Na AND, são listados apenas os arquivos que contém todas as palavras solicitadas na busca, enquanto na OR são exibidas todas as ocorrências que contém as palavras, não necessariamente todas que foram solicitadas.

2.6. Listagem do resultado da(s) palavra(s) buscada(s)

Na mesma interface que é permitido acionar a busca por palavras, há um campo reservado para a listagem com os resultados. Assim, quando o usuário aciona a funcionalidade de busca, automaticamente é chamada a função de listagem dos resultados, que lista, na interface, o resultado encontrado.

O formato da listagem dos resultados segue apresentando o título do arquivo está associada, a quantidade de ocorrências identificadas da palavra buscada (na mesma linha) e a linha do arquivo na qual ela se encontra.

3. Descrição da solução

diagrama

4. Estruturas de Dados

A estrutura de dados escolhida para a implementação deste tipo de sistema foi a árvore digital Trie. Essa estrutura permite realizar a busca por uma palavra em $O(klog(m)+k)$, onde k é o tamanho da chave buscada e m é o número de caracteres no alfabeto.

A principal característica de uma árvore digital é que uma chave não é mantida em um único nó, mas sim obtida através de uma sequência de nós que começa da raiz e vai até um determinado nó v que é denominado terminal. Além disso, o caminho da raiz até um nó qualquer é chamado de prefixo.

Cada nó corresponde a um único dígito e a aresta entre o próprio nó e seu pai representa o caractere que compõe a chave. Para implementar a estrutura de acordo com esta propriedade, é utilizada uma árvore m-ária e a recuperação/atribuição do caractere é feita uma busca binária para definir cada filho do nó.

Dessa forma, ao buscar uma palavra no sistema, não é feita a comparação de toda a chave em cada nó, o que resultaria em $O(log(n))$ comparações e cada operação geraria um custo proporcional ao comprimento da palavra, mas sim de cada dígito que forma o prefixo da chave na Trie. É essa particularidade que torna a árvore digital mais eficiente para este tipo de aplicação que as demais.

Para facilitar o armazenamento das ocorrências de uma certa palavra nos arquivos do sistema, cada nó terminal carrega também a informação de em quais arquivos a contém, bem como as linhas e o número de vezes que o termo se repete naquela linha.

5. Reflexão

Ao longo da disciplina, foi aprendido sobre a importância de efetuar um bom planejamento do projeto, com discussões sobre o que precisa ser desenvolvido, descrição das principais funcionalidades, antes da implementação, e elaboração de um diagrama de classes do sistema.

O projeto foi desenvolvido levando em consideração essas abordagens. Deste modo, antes de iniciarmos a codificação, foi feito um levantamento das funcionalidades que o sistema deveria ter, descrição e investigação sobre a relação entre as classes e funções que poderiam ser utilizadas por mais de uma funcionalidade, bem como o modo como isso seria visualizado (interfaces gráficas). Além disso, foi elaborado um diagrama de classes inicial para o sistema.

Após isso, foi gerado e documentado uma versão inicial (stub) das classes do sistema, com a declaração e descrição da composição (campos e comportamentos) de cada uma das classes que foram planejadas no diagrama de classes. Essas primeiras etapas do projeto foram realizadas com a equipe reunida e, depois de concluído, houve a divisão de tarefas entre os membros do grupo. A implementação do sistema foi realizada utilizando como suporte uma ferramenta apresentada em sala: o sistema de controle de versão de arquivos (git) e o sistema de hospedagem de projetos (github), que possui, também, funções extras aplicadas ao git.

Um dos problemas que não foi solucionado é o da implementação do autocorrect, isto é, a identificação de palavras semelhantes à que foi digitada na busca, de modo a tratar erros como o de digitação incorreta por parte do usuário.