TP PDSII - Classe para Busca em Arquivos

- **Dupla:** André Domingos Cursino e Roberto Demétrio Neves Júnior
- **Turma:** TFI - Engenharia de Controle e Automação

Documentação da Classe

Métodos Públicos

void Inserir_Livro(string nome, string caminho)

Nome: Título do livro

Caminho: Caminho até o arquivo txt do livro. (Ex: C:\\livro.txt)

Essa função é utilizada para inserir um novo livro ao grupo de livros nos quais as buscas serão realizadas(pool de busca).

void Remover_Livro(string nome)

Nome: Título do livro

Essa função é utilizada para remover um livro do pool de busca.

int Quantidade_Livros()

Essa função retorna um inteiro que representa a quantidade de livros já inseridos no pool de busca.

void Limpa_Livros()

<

p>Essa função é utilizada para remover todos os livros do pool de busca.

std::map\<string, std::set\<string>> Buscar(string fraseDigitada)

Frase Digitada: Grupo de palavras que o usuário deseja procurar, separadas por espaços

Essa função é responsável por realizar todo o mecanismo de busca que é o objetivo final da classe. A função recebe uma string como argumento e realiza a busca das palavras dessa string dentro do pool de livros.

A função retorna uma estrutura do tipo MAP, na qual a chave é a palavra buscada e o valor é uma estrutura do tipo SET, que contém os nomes dos livros que contêm aquela chave.

Métodos Privados (Fins acadêmicos)

void Buscar_Frase(string fraseBusca)

Frase Busca: String a ser procurada

Essa função realiza a busca propriamente dita, entretanto, não retorna valor, somente armazena o resultado dentro do MAP (estrutura explicitada na justificativa). Essa separação (dessa função e a função pública que retorna o mapa) busca viabilizar a utilização do mecanismo de busca por outros métodos sem a necessidade do retorno de uma variável tão específica quanto um MAP\. Ou seja, busca deixar um caminho para a construção de uma classe mais dinâmica, suscetível a futuras alterações.

string Arrumar_Palavra(string palavra)

Palavra: Palavra que será ajustada de acordo com as requisições do trabalho.

Essa função é responsável por ajustar a input do usuário, que, para correto funcionamento da classe deve ser constituída por letras minúsculas e números.

std::set\<string> Busca_Nos_Arquivos(string palavra)

Palavra: Palavra que será buscada dentro de todos os arquivos.

Essa função organizará um SET com todos os arquivos que contêm a palavra passada como argumento. Ela não faz a varredura dos arquivos, somente trigga a função responsável, informando os arquivos a serem varridos e organizando as respostas.

bool Varre_Arquivo(string caminho, string palavra)

Caminho: Caminho do arquivo que será aberto e varrido.

Palavra: Palavra que será procurada dentro daquele arquivo.

Essa função realiza a abertura e varredura dos livros, retornando se o livro contém ou não aquela palavra.

Justificativa da Estruturação

Nesse projeto, buscamos aumentar o nível de abstração ao máximo, de forma que todo o processo de busca seja invisível(transparente) ao usuário da classe.

Foi utilizada a estrutura **MAP** para organizar a busca, primeiramente por ser uma requisição do trabalho, mas também por ser a estruturação mais lógica para realizar a tarefa de relacionar as palavras com os resultados das buscas.

Também foi utilizado **MAP** para estruturar os livros no pool de busca, buscando viabilizar a "Nomeação" do livro. Com essa estrutura, o usuário pode ter um arquivo de texto com o nome diferente do escolhido como título do livro. (**Exemplo:** Livro: *João e Maria* Arquivo: *joaoemaria.txt*)

Como Valor relacionada a cada chave na estrutura **MAP** de busca foi utilizada a estrutura **SET**, a qual foi uma forma encontrada de relacionar os múltiplos títulos de livros que resultaram da busca, sem perder a individualidade de cada um, ou seja, cada título do resultado pode ser obtido separadamente ainda que todos estejam relacionados em um mesmo SET.

Por fim, nota-se que o processo de busca é separado em diversas pequenas funções privadas, as quais são todas chamadas quando o usuário chama a função pública Buscar. Essa divisão em múltiplas funções se deu para facilitar o processo de testes e troubleshooting, e não é visível ao usuário.