

Trabalho Prático de **Programação e** **Desenvolvimento de Software** **II**

Pesquisa de palavras em um ou mais textos

Alunos: João Victor de Carvalho Silva / Lucas Rodrigues dos Santos

Sumário

1 - Introdução	2
2 - Desenvolvimento	2
3 - Descrição do Algoritmo	3
4 - Testes realizados.....	3
5 - Controle de versão	4
6 - Conclusão.....	4

1 - Introdução

A ideia do trabalho prático era fazer um programa que consegue encontrar, em um ou mais textos, palavras que forem digitadas e fornecidas por um usuário. Precisávamos desenvolver um programa que recebesse as informações, procurasse nos arquivos e devolvesse para o usuário o(s) nome(s) do(s) arquivo(s) onde há ocorrência daqueles termos fornecidos por ele, contemplando os conhecimentos adquiridos em sala e tendo em vista as boas práticas de programação.

Por termos um pouco de conhecimento na linguagem Java, optamos por ela, dada as facilidades para conseguirmos desenvolver um programa com as características acima levantadas.

2 - Desenvolvimento

Para desenvolvimento, criamos a classe MyMap, que contempla o construtor e os métodos “tratamento”, “pesquisa” e “print”, explicados individualmente abaixo:

- O construtor contém uma ArrayList chamada “documentos”, que será responsável por guardar os nomes dos arquivos que contém a palavra informada, e um array de strings chamado “arrayDeArquivos”, que recebe o nome dos arquivos que estão no diretório e serão percorridos e pesquisados em busca das palavras que estão armazenadas em “documentos”.
- TRATAMENTO: É o método que padroniza as palavras dos documentos e a entrada do usuário, com as funções “replaceAll”, que substitui todas as palavras que contém algum termo não alfanumérico por uma expressão em branco e “toLowerCase”, que transforma todas as letras maiúsculas em minúsculas (caso existam). Por exemplo, se o usuário digitar “GUARDA-CHUVA”, a função tratamento retorna “guardachuva”, o que torna possível a busca por essa palavra nos documentos.
- PESQUISA: Percorre todos os documentos atrás da palavra informada pelo usuário. Faz uso do método “tratamento”. Quando acha a palavra no arquivo, adiciona o nome do arquivo para retornar, no futuro, juntamente com os outros que contiverem a palavra.
- PRINT: Retorna quais dos documentos contém a(s) palavra(s) informada(s) pelo usuário.

Na nossa função MyMapMain, o usuário informa quais palavras serão pesquisadas. As palavras passam pelo método “tratamento” e depois ocorre a busca nos arquivos (através do método “pesquisa”), seguida pela execução do método “print”. Ao final, o usuário é perguntado se deseja realizar mais uma pesquisa. Se a resposta for afirmativa, ele informa as palavras e o procedimento se repete. Caso a resposta seja negativa, o programa se encerra.

3 - Descrição do Algoritmo

Para desenvolver a máquina de busca com índice invertido, por questões de eficiência, optamos por pesquisar a palavra direta no arquivo, sem a necessidade de armazenamento na memória do computador. Portanto, assim que a palavra era lida, primeiro era padronizada com a função tratamento e depois comparada com a palavra do usuário (também padronizada pela mesma função). Se as palavras são iguais, o nome do arquivo é armazenado em uma lista. Um ponto importante a se destacar, é o fato de que, quando a palavra era encontrada, a leitura do arquivo atual é interrompida para começar a pesquisa em outro arquivo. Este processo justifica a escolha pela pesquisa direto no arquivo, pois não há necessidade de percorrer o arquivo inteiro, caso a palavra já tenha sido encontrada. Após a impressão do resultado da busca, os elementos contidos na lista são apagados, para que uma nova pesquisa seja feita.

4 - Testes realizados

```
G:\Users\joao.silva\Documents\TP PDS2 - João Victor e Lucas>java MyMapMain
insira uma palavra: while
hamlet.txt
kinglear.txt
macbeth.txt
othello.txt
romeoandjuliet.txt

deseja realizar mais uma pesquisa?
digite o numero da opcao: <1>-sim <2>-nao
1
insira uma palavra: WHI-L-E
hamlet.txt
kinglear.txt
macbeth.txt
othello.txt
romeoandjuliet.txt

deseja realizar mais uma pesquisa?
digite o numero da opcao: <1>-sim <2>-nao
1
insira uma palavra: ham let
hamlet.txt

deseja realizar mais uma pesquisa?
digite o numero da opcao: <1>-sim <2>-nao
1
insira uma palavra: before dark
nenhum dos documentos contem a palavra
deseja realizar mais uma pesquisa?
digite o numero da opcao: <1>-sim <2>-nao
1
insira uma palavra: dark
hamlet.txt
kinglear.txt
macbeth.txt
othello.txt
romeoandjuliet.txt

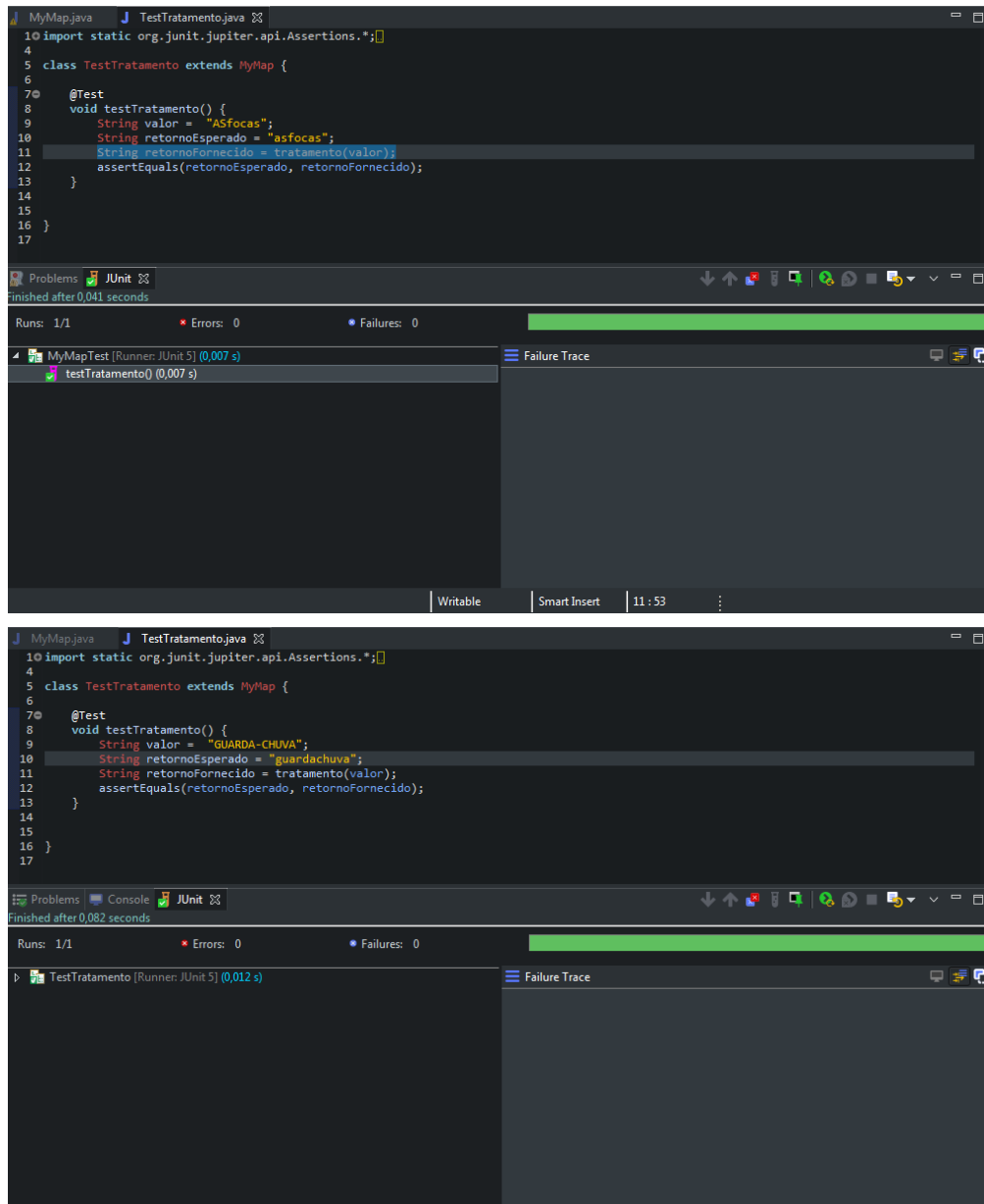
deseja realizar mais uma pesquisa?
digite o numero da opcao: <1>-sim <2>-nao
1
insira uma palavra: d a r k
hamlet.txt
kinglear.txt
macbeth.txt
othello.txt
romeoandjuliet.txt
```

Testamos o caso em que ocorre uma exceção. No caso, retiramos o arquivo “hamlet.txt” do diretório, mas mantivemos o nome do arquivo no código do programa:

```
insira uma palavra: peace
arquivo hamlet.txt nao encontrado
kinglear.txt
macbeth.txt
othello.txt
romeoandjuliet.txt

deseja realizar mais uma pesquisa?
digite o numero da opcao: (1)-sim (2)-nao
2
```

Também fizemos testes de unidade com o método “tratamento”, conforme imagem abaixo.



O método foi o único a contemplar testes de unidade pois era o único método com retorno, todos os outros eram sem retorno (void). Os testes saíram como o esperado e o arquivo encontra-se junto dos demais com o nome de “TestTratamento.java”.

5 - Controle de versão

Criamos uma versão primária onde o programa era composto apenas pela classe `MyMapMain`, que realizava todos os métodos. Depois, separamos as duas classes e adicionamos o tratamento de exceção. Como era um programa simples, com apenas duas versões, não achamos necessário utilizar algum programa para realizar esse controle. Porém, depois que colocamos o programa no GitHub, fizemos algumas alterações no código e na documentação.

6 - Conclusão

O trabalho nos proporcionou algumas experiências inusitadas, principalmente em se tratando de tratamento de exceção e teste unitário, que não tínhamos um grande conhecimento prévio. Aprendemos algumas novas funções do Java que nos ajudou bastante, como a `Scanner`, que foi primordial para leitura dos arquivos que recebemos. De maneira geral, nos ajudou a compreender a

disciplina e aplicar as principais práticas adquiridas através das discussões em sala de aula.