

Trabalho Prático: Máquina de Busca

Douglas Silva Ferreira

Matrícula: 2015123339

Instruções para utilização e testes do trabalho(TP) prático índice invertido

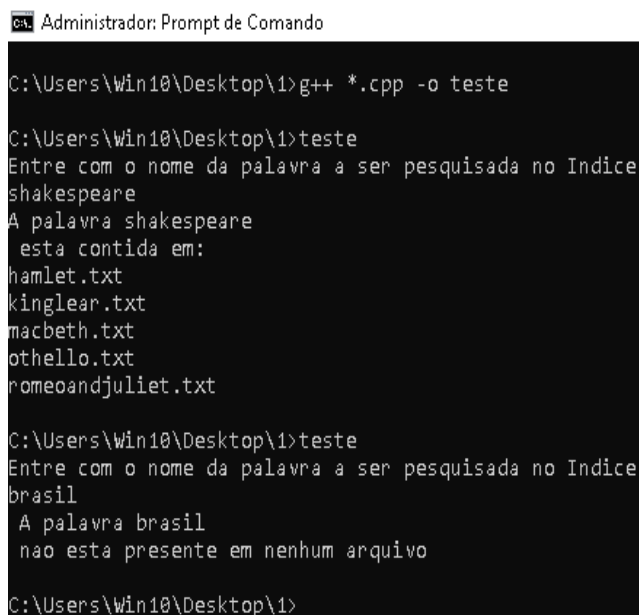
Avisos:

Aviso 1

Para pesquisar as palavras é necessário que se digite os nomes dos arquivos, de forma explícita e com extensão, na função membro void LerArquivo() da classe Indice. Os nomes dos arquivos são armazenados em um vetor de strings com um tamanho fixo relativo à quantidade de arquivos a ser aberta, caso se queira abrir mais arquivos é precisa aumentar esse tamanho.

Aviso 2

Esse trabalho prático foi desenvolvido no ide do netbeans no sistema operacional Windows. Portanto, vou disponibilizar no repositório github tanto o projeto como também os arquivos separados, incluindo os arquivos a serem lidos. Pra compilar no prompt de comando do Windows, basta que se navegue até à pasta dos arquivos com **cd "caminho"** e se digite **g++ *.cpp -o nomequalquer**. Depois digite nomequalquer pra executar o arquivo gerado na compilação. O prompt não roda os arquivos de teste do netbeans, deve ser necessário algum comando específico, portanto é necessário não deixá-los no diretório do projeto, caso se queira compilar no prompt.



```
ca. Administrador: Prompt de Comando

C:\Users\Win10\Desktop\1>g++ *.cpp -o teste

C:\Users\Win10\Desktop\1>teste
Entre com o nome da palavra a ser pesquisada no Indice
shakespeare
A palavra shakespeare
esta contida em:
hamlet.txt
kinglear.txt
macbeth.txt
othello.txt
romeoandjuliet.txt

C:\Users\Win10\Desktop\1>teste
Entre com o nome da palavra a ser pesquisada no Indice
brasil
A palavra brasil
nao esta presente em nenhum arquivo

C:\Users\Win10\Desktop\1>
```

O TP foi dividido em duas classes chamadas IsChars e Indice, com arquivos .cpp e .h. A função main cria um objeto que aciona o construtor da classe Indice, construtor esse, que faz as chamadas das funções membro dessa mesma classe:

void LerArquivo():

Essa função armazena os nomes dos arquivos em um vetor de strings e depois em um loop, os abre, e utiliza de funções auxiliares e também da classe IsChars para retirar caracteres indesejados na leitura, de modo que cada palavra é lida até o espaço. Essas palavras são transformadas em minúsculas e são armazenadas em listas(foi criado um vetor de listas e as palavras de cada arquivo são armazenadas em uma lista diferente).

void EntraComPalavra():

Essa função pede ao usuário que entre com a palavra que deseja pesquisar nos arquivos que foram abertos. As palavras digitadas no teclado, caso sejam digitadas em maiúsculo e também com caracteres especiais, passam por funções que as transformam em minúsculo e sem caracteres.

void ArmazenaNomeArquivoQContem():

Essa função percorre um vetor de listas que armazenam as palavras e identifica os arquivos que contém a palavra digitada na função void Entra_Com_Palavra() e armazena o nome desses arquivos em outra lista .

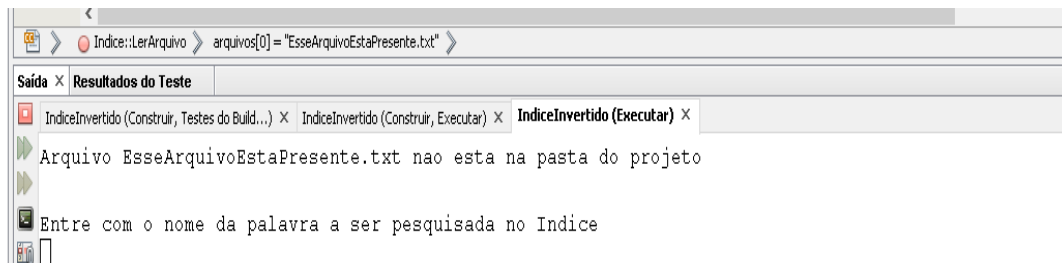
void ImprimeArquivosQContem():

A lista que armazena os nomes dos arquivos que contém a palavra a ser pesquisada, que foi criada na função void ArmazenaNomeArquivoQContem(): é percorrida e é impresso na tela os arquivos que possuem a palavra.

Abaixo as exigências, conforme a descrição do TP, são revisadas, parte por parte.

Parte 1

Os arquivos são lidos pela função ***void LerArquivo()*** nessa função foi utilizado um teste de exceção, para que, caso os arquivos a serem lidos não estiverem no diretório local, é apresentada uma mensagem “O arquivo xxx não está presente na pasta do projeto”. Como mostra a figura abaixo na tentativa de ler o arquivo EsteArquivoEstaPresente.txt, que não está na pasta no projeto.



Parte 2

Apesar de existirem outras implementações possíveis a partir das bibliotecas do c++, por exemplo utilizando a biblioteca padrão <map>, nesse trabalho foi feita a escolha de armazenar as palavras, livres de caracteres especiais e transformadas em minúsculo, em listas. Cada arquivo dispõe de uma lista de strings que faz esse armazenamento de palavras. As listas de strings estão em um vetor de tamanho equivalente à quantidade de arquivos a serem lidos, e o índice desse vetor é relacionado aos nomes dos arquivos. Quando cada uma dessas listas é lida e é constatado que a palavra está presente, o fluxo do programa passa a ler a próxima lista e o nome do arquivo relacionado à essa lista é adicionado à outra lista que contém os nomes desses arquivos que contém a palavra digitada no teclado.

Parte 3

A palavra a ser pesquisada é lida do teclado pela função ***void EntraComPalavra()***, e como explicado na Parte 3, as listas, relacionadas aos arquivos com as palavras, são lidas e é verificado a presença daquela palavra. Caso ela esteja presente o nome dos arquivos, ligados às listas, são armazenados em outra lista. Feito todo esse processo, a função ***ImprimeArquivosQContem()*** verifica se a lista que armazenou os nomes dos arquivos está vazia, se está, a função imprime que aquela palavra não está presente em nenhum dos arquivos, caso contrário, a lista é percorrida e os nomes dos arquivos são impressos na tela.

Seguem imagens.

A palavra Shakespeare está presente nos arquivos listados

```
Entre com o nome da palavra a ser pesquisada no Indice
shakespeare
A palavra shakespeare
  esta contida em:
romeoandjuliet.txt
macbeth.txt
othello.txt
kinglear.txt
hamlet.txt
```

```
EXECUTAR SUCCESSFUL (tempo total: 43s)
```

A palavra Brasil não está presente em nenhum arquivo

```
IndiceInvertido (Construir, Executar) x  IndiceInvertido (Executar) x
Entre com o nome da palavra a ser pesquisada no Indice
brasil
A palavra brasil
  nao esta presente em nenhum arquivo

EXECUTAR SUCCESSFUL (tempo total: 8s)
```

Parte 4

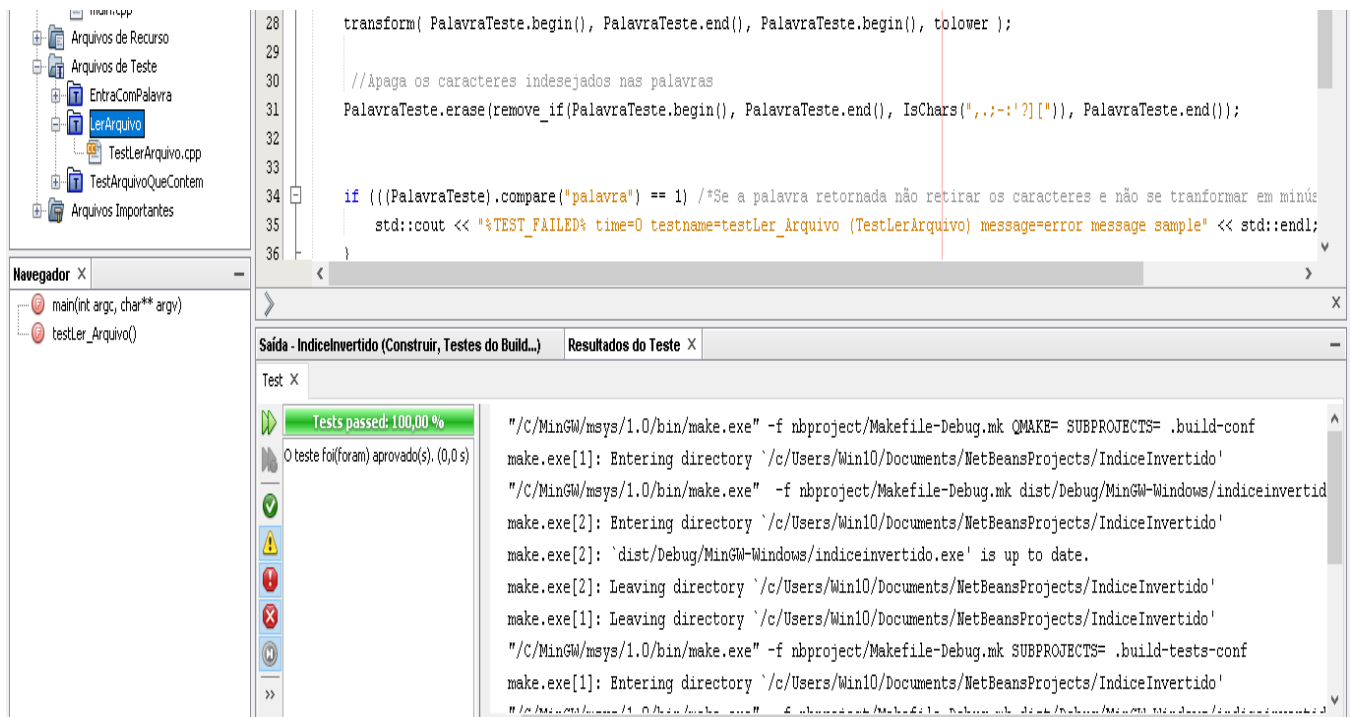
A parte 4 é essa documentação que está sendo redigida.

Parte 5

Apesar de a maioria das funções serem funções void, e isso dificultar um pouco os testes de unidade nas 3 das 4 funções membro da classe Indice foram feitos testes de unidade.

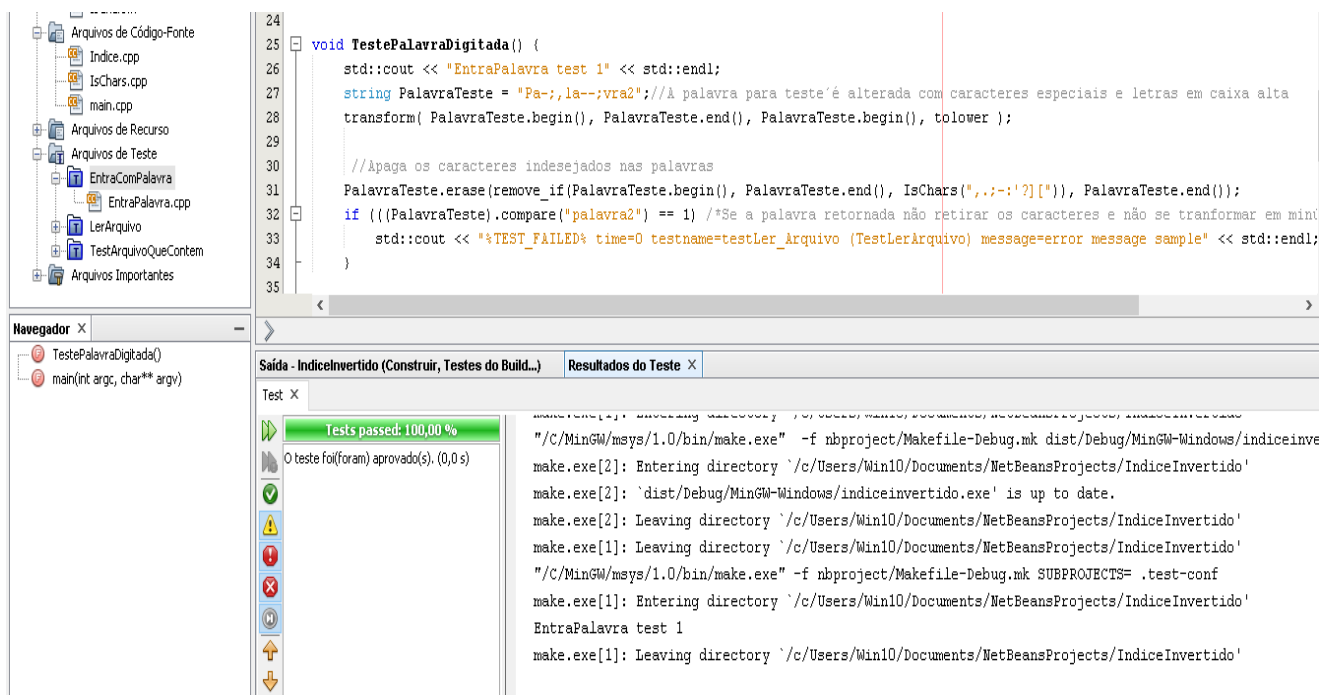
void LerArquivo():

O arquivo de teste TesteLerArquivo.cpp verifica se as palavras, digitadas com caracteres especiais e com letras maiúsculas, retornam das funções transformadoras como minúsculas e sem caracteres especiais, se a resposta for positiva o teste é retornado sem erros, se não é apresentado um erro.



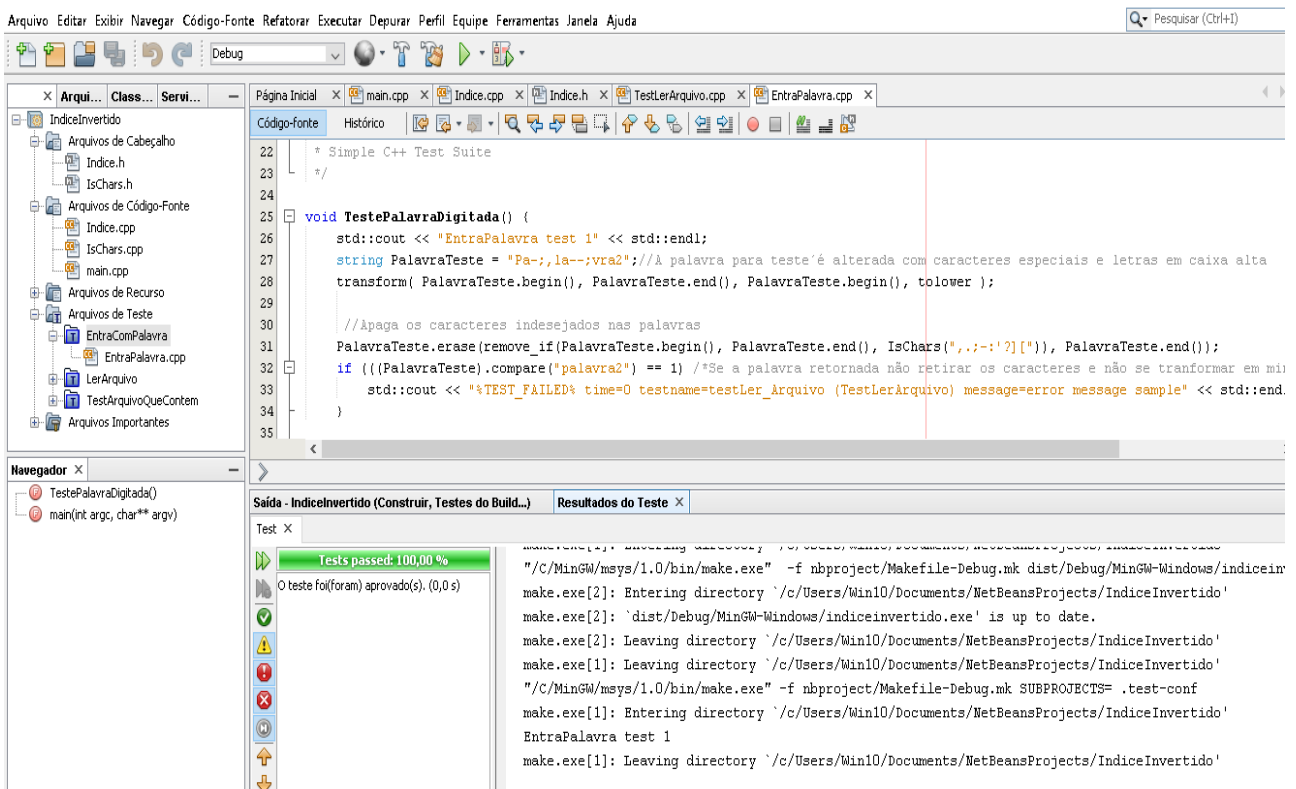
void EntraComPalavra():

O teste de unidade dessa função em síntese é igual ao que foi apresentado no item anterior. O arquivo de teste EntraPalavra.cpp faz o serviço de teste.



void ArmazenaNomeArquivoQContem():

O arquivo de teste TestArquivoQueContem testa se ocorre uma resposta positiva, a respeito da presença de uma palavra em uma lista, é garantido que a palavra está contida na estrutura. Se o retorno é positivo, o teste passa, se não, falha.



Não foi feito teste de unidade na função membro ***ImprimeArquivosQContem()***.

Essa função, apenas, percorre a lista e imprime as strings.

Esses testes foram feitos com uma ferramenta nativa da ide netbeans(Novo teste simples c++), com alguns pequenos ajustes, para atender as necessidades desse projeto.

Os arquivos de testes serão mantidos no projeto que será disponibilizado na entrega do trabalho.

Bibliografia

<https://stackoverflow.com/questions/5891610/how-to-remove-certain-characters-from-a-string-in-c/21014485>

<http://www.cplusplus.com/reference/list/list/>

<https://www.geeksforgeeks.org/list-cpp-stl/>