

Documentação

1. Introdução

O Trabalho Prático de PDS2 tem como objetivo principal implementar os subsistemas de Indexação e Recuperação de uma máquina de busca (usualmente subdividido em 3 etapas: Coleta, Indexação e Recuperação). O segmento de coleta já foi realizado previamente.

O desenvolvimento conjunto pelo github teve o auxílio da ferramenta Github Desktop para a sincronização, pull e push do repositório.

De início, tivemos a intenção de implementar duas classes distintas, uma de normalização e outra de pesquisa, originalmente classe `normaliza_palavra` operava da seguinte maneira:

Recolhe a string de dados requeridas e realiza o tratamento indicado no enunciado. Houveram problemas para normalizar todos os caracteres, pois o tipo `char` reconhece apenas valores ascii até 127 (128-1). O que dificulta o tratamento de caracteres como vogais acentuadas e cedilha. Realizamos o tratamento em duas funções, uma para excluir os caracteres especiais e numéricos, mais outra para transformar todos os caracteres em minúsculo usando a biblioteca `<cctype>`.

Transferimos-a para dentro da própria classe `pesquisa`, tendo em vista que economiza em linhas de código.

2. Implementação

O grupo optou por modularizar o código em duas classes principais:

`pesquisa`: Cria um índice invertido do tipo `map<string, set<string>`, ou seja, um mapa que liga uma determinada palavra chave a uma lista de documentos da qual ela pertence. O mapa é criado através do uso das bibliotecas `"filesystem"` e `"fstream"` para iterar sobre todos os arquivos da pasta `/documentos`. Além disso, o método interno `pesquisar()` envia como output na entrada padrão todos os documentos relevantes para determinada string passada como parâmetro. Como listas são naturalmente ordenadas, a função percorre a lista do início ao fim para obter o

ranqueamento lexicográfico dos resultados.

```
{  
  // See https://go.microsoft.com/fwlink/?LinkId=733558  
  // for the documentation about the tasks.json format  
  "version": "2.0.0",  
  "tasks": [  
    {  
      "label": "Compilar e Rodar",  
      "type": "shell",  
      "command": "g++ -std=c++17 main.cpp pesquisa.cpp normaliza_palavra.cpp doctest.h -lstdc++ && ./a.out",  
      "problemMatcher": [],  
      "group": {  
        "kind": "build",  
        "isDefault": true  
      }  
    }  
  ]  
}
```

A compilação foi realizada através de um task.json.

3. Conclusão

O grupo em seu Trabalho Prático obteve o ranqueamento com sucesso.

Houve um grande atraso na função de normalização, tendo em vista que a exibição de caracteres especiais varia de sistema operacional para sistema operacional, sendo necessário utilizar a biblioteca `<locale> setlocale(LC_ALL, "pt_BR_utf8")` para exibir caracteres como “ç” (c cedilha).

Houveram também dificuldade na implementação de testes no doctest, então a maioria dos testes, debug e compilação foram realizados através da plataforma virtual <https://replit.com/>.