

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
SISTEMAS DE INFORMAÇÃO**

**Roberto Costa Tupinambá
Yan Gustavo Pegyn Silva
Guilherme Francisco Silva Vidigal**

RELATÓRIO GERAÇÃO DE UM SISTEMA DE ORI

**MONTE CARMELO
2020**

Sumário

Objetivos	2
Descrição do problema	2
Estruturas de dados e limitações	2
Funções utilizadas	2

Objetivos

- Desenvolver o código fonte em Python;
- Gerar uma listagem txt com o dicionário criado;
- Elaborar um relatório PDF, padrão ABNT, com capa e sumário, descrição do problema, limitações e estruturas de dados, funções utilizadas.
- Apresentar seminário sobre a fase em aula.

Descrição do problema

- Ler os arquivos de termos e criar seus dicionários. Nenhum trabalho de padronização, stopwords e lematização deve ser feito.
- Implementar o modelo vetorial.
- Fazer buscas.
- Retornar os resultados em janela estática com os textos correspondentes (10 primeiras palavras do texto).
- Durante a apresentação dos trabalhos serão pedidas 3 buscas.

Estruturas de dados e limitações

Para a solução do problema nos utilizamos de estruturas de dados, sendo estas, listas e dicionários.

As principais limitações são: Necessidade de um arquivo “.txt” contendo a lista de documentos e o índice invertido. Além de necessitar de todos os arquivos “.pdf” que originaram o índice invertido.

Funções utilizadas

Utilizamos o módulo fitz que faz parte do pacote PyMuPDF, a função log10 do módulo math nativo do python 3.9, e o módulo tabulate para se criar uma visualização da tabela.

Definimos 15 funções novas e a função principal, sendo elas:

1. ler: Lê o arquivo de índice invertido.
2. quebrar_indice: Transforma o índice invertido em objeto.
3. quebrar_doc: Formata o número de documentos e a quantidade.
4. calc_idf: Calcula o IDF de uma palavra.

5. `calc_w`: Calcula o W de uma palavra em um documento.
6. `calc_tf_idf`: Calcula o TF_IDF de uma palavra em um documento.
7. `monta_vcons`: Monta lista do vetor de constante (pega os IDF's tudo).
8. `monta_vdoc`: Monta cada vetor de documentos.
9. `monta_mdoc`: Calcula o módulo de cada um dos documentos.
10. `monta_vdoc_normalizado`: Normaliza os vetores de documentos.
11. `calcula_cossenos`: Calcula os cossenos.
12. `ler_lista_docs`: Lê a lista de documentos e converte para objetos.
13. `ler_sumario`: Busca o sumário (10 palavras cada).
14. `rankear`: Rankear os documentos (pelos cossenos).
15. `monta_linhas_print`: Formata as linhas para a tabela.
16. `docs_presentes`: Cria a lista de quais documentos as palavras estão presentes.

Já na função principal (`__main__`), é solicitado as palavras a serem buscadas, executa as funções elaboradas para processar a busca, e assim que finalizado esses processos exibe uma tabela contendo o resultado da busca.