CIÊNCIA DA COMPUTAÇÃO
Estrutura de Dados
Prof. Me. Fahad Kalil

PROJETO - INDEXAÇÃO DE DADOS

(1) Instruções
- Este trabalho vale 60% da nota G2;
- O trabalho deverá ser desenvolvido individualmente ou em grupos com até 3 integrantes;
- A data de entrega é: **03/12/2020 (até 23:59 / mesmo dia da Apresentação)**
- Deverá ser enviado pelo AVA, em arquivo **.zip**

(2) Objetivos
- Compreender a criação de índice invertido, proporcionando busca rápida em documentos textuais;

(3) Entregáveis:

Código-fonte do app contendo interação via linha de comando com as opções descritas no final do arquivo, bem como a pasta docs/ contendo os documentos TXT de exemplo.

Apresentação do aplicativo por todos os integrantes do grupo, no horário normal de aula.

(4) Plágio:
- Caso seja detectado plágio entre trabalhos, por meio de ferramenta de comparação de códigos, os envolvidos terão avaliação zero!

ROTEIRO:

** CRIAR SUA PRÓPRIA BASE DE ARQUIVOS DE TEXTO **
|___ Definir um tema e criar, pelo menos, 10 arquivos contendo notícias e salvos em TXT (padrão UTF-8)

- **Criar** um menu inicial que irá conter as opções:

```
------- INDEXAÇÃO -------

1. Criar Novo Documento
2. Indexar documentos '.txt' presentes na pasta docs/
3. Realizar consultas
     1. Usando operador OR
     2. Usando operador AND
     3. [opcional] Usando expressões booleanas
4. Mostrar Índice Invertido (para debug / print)
```

  - Detalhes:
    - **No item 1**: Ao criar o novo documento, solicitar ao usuário: nome do arquivo (incluindo extensão txt) e solicitar que seja digitado um conteúdo. Salvar na pasta docs/
    - **No item 2**: Ler os documentos .txt para indexação de uma pasta chamada docs/, limpando o índice atual, caso já contenha entradas, antes de indexar.
      - Nesta etapa deverão ser executados, para cada documento lido, os passos de Tokenização, Normalização (remover pontuações e deixar tudo em minúsculas), Remoção de StopWords, Stemming e por fim, a inclusão dos termos e documento no índice invertido.
      - Salvar o índice (dicionário) em um arquivo usando a biblioteca Pickle.

- **No item 3.1**, Exemplo de busca é: `brasil deputado`
  - Equivalente a: brasil OR deputado
  - Mostrar o nome do arquivo dos documentos que possuem quaisquer destes termos, sem duplicatas e maneira ordenada.
    - Usar método de UNION

- **No item 3.2**, Exemplo de busca é: `brasil deputado`
  - Equivalente a: brasil AND deputado
  - Mostrar a lista de documentos que possui ambos os termos.
    - Usar método de INTERSECT
- **No item 3.3**:
  - EXTRA (se implementado será acrescido 0,5 na nota deste projeto)
  - Exemplo de busca é: (prefeito OR deputado) AND brasil
  - Avaliar a expressão e resolver passo a passo cada expressão, como no exemplo de busca citado:
    - 1º) Obter lista de documentos que possuem a palavra 'brasil'
    - 2º) Obter a lista de documentos que possui a palavra 'prefeito'
    - 3º) Obter a lista de documentos que possui a palavra 'deputado'
    - Resolver primeiro a expressão (prefeito OR deputado) e então realizar a operação AND do resultado obtido versus a lista de documentos obtida a etapa 1.

  - DICA: Usar pilha para resolver as expressões com parênteses.

- **No item 4:** Para mostrar o índice que representa a lista de termos e documentos:
  - É um print padrão!
  - { "*Termo*" : ["docs\\doc1.txt", "docs\\doc2.txt", ...] }