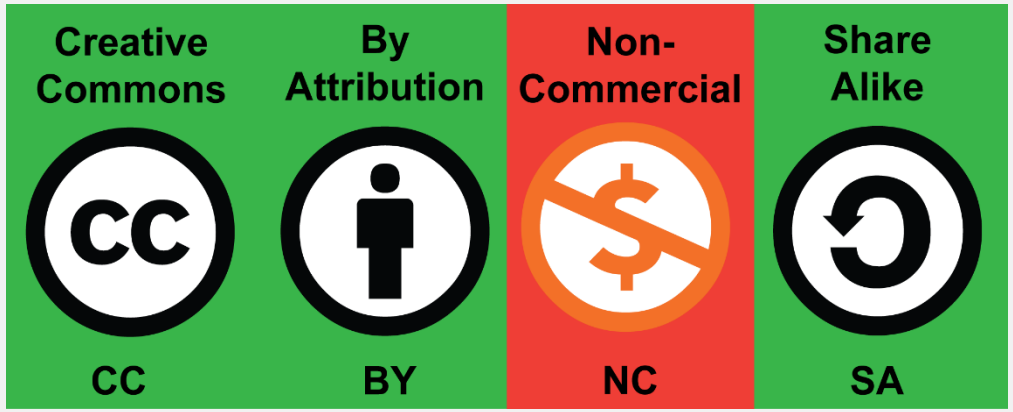
Listas Invertidas

Prof. MSc. Jackson Antonio do Prado Lima
jacksonpradolima at gmail.com

Departamento de Sistemas de Informação – DSI

UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA

Licença



UDESC

Histórico de Modificação

- Esta apresentação possui contribuição dos seguintes professores:
  - Alex Luiz de Souza (UDESC)
  - Anderson Fabiano Dums (UDESC)
  - Fernando José Muchalski (UDESC)
  - Jackson Antonio do Prado Lima (UDESC)

UDESC

Agenda

- Visão Geral
- Funcionamento
- Aplicação
- Exemplo
- Laboratório

UDESC

Visão Geral

- Estrutura de Dados que mapeia palavras-chave com seu conteúdo ou documento relacionado (***Inverted List*** ou ***Inverted Index***)
  - É uma estratégia de indexação que permite a realização de buscas precisas e rápidas
  - É uma das mais populares estratégias de sistemas para obtenção de dados, usada em larga escala em sistemas de gerenciamento de banco de dados e serviços de busca

Funcionamento

- Construída com base em uma lista tradicional de documentos, invertendo a hierarquia da informação (informação -> dado)
  - Ao invés de uma lista de documentos contendo termos, é obtida uma lista de termos, referenciando estes documentos
  - Esta referência é feita, normalmente, através de um identificador único, como uma chave primária
  - Junto deste identificador, podem ser armazenadas outras informações, conforme adequado para a natureza das buscas desejadas. Exemplo: armazenar a posição do termo no documento

Exemplo

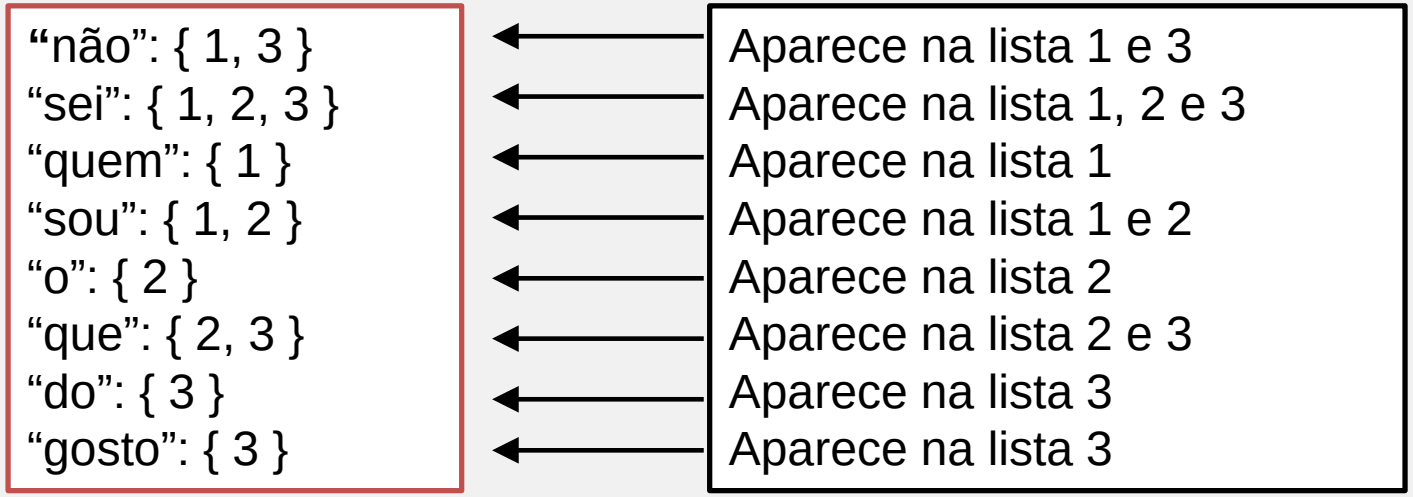
- Data a seguinte lista de documentos:

1: “Não sei quem sou”

2: “Sou o que sei”

3: “Sei do que não gosto”

- Obtemos a seguinte lista invertida:



UDESC

Aplicação

- **Listas invertidas** são um elemento central de sistemas de busca, pois estes visam trazer resultados de forma rápida e eficiente
- Buscas por termos, em uma lista tradicional, exigem percorrer **cada documento** e **cada palavra** dentro destes em busca do termo desejado
  - Com o uso do **índice inverso**, pode-se saltar diretamente para o termo procurado
  - O desempenho tende a ser cada vez mais significativo conforme aumenta a quantidade de documentos

Aplicação

- O uso de listas invertidas tem o potencial de deixar as buscas mais eficientes, dado que estas permitem que sejam armazenadas informações adicionais que, acompanhadas de algoritmos adequados, tornam fácil a classificação e ordenação dos resultados

Aplicação

- **Desvantagem**: o custo destes benefícios vem na forma de trabalho adicional para a manutenção desta lista.
  - É preciso manter a **lista invertida atualizada** (ou seja, rodar o programa gerador da lista) conforme documentos são inseridos, alterados e excluídos da lista tradicional

Exemplo: Construção

- Uma lista invertida pode ser construída com base em uma lista comum:

```
Lista_invertida = { palavra_1 = [ (arquivo, posição) ],
                    palavra_2 = [ (arquivo, posição) ],
                    ...
                    palavra_n = [ (arquivo, posição) ] }
```

- Onde:
  - **Palavra**: é uma palavra ou termo de um documento ou arquivo
  - **Arquivo**: é o número do arquivo dentro uma lista de 1 ou mais arquivos, que representam o espaço de busca
  - **Posição**: é a posição da palavra ou termo no documento ou arquivo.

Indexação

- Considerando os seguintes documentos (arquivos) como espaço de busca:

Centro de Educação do Planalto Norte	Universidade do Estado de Santa Catarina	Disciplina de Estrutura de Dados II
1.txt	2.txt	3.txt

- **Exemplo**: considerando os documentos (arquivos) acima como espaço de busca para o processo de indexação de uma lista invertida tem-se (*próximo slide*)

Busca

- O mecanismo de busca consiste em obter a localização de um ou mais termos (palavras) no espaço de busca:

```
Index ={ centro=[(1,1)], de=[(1,2),(2,4),(3,2),(3,4)],
         educacao=[(1,3)], do=[(1,4),(2,2)],
         planalto=[(1,5)], norte=[(1,6)], universidade=[(2,1)],
         estado=[(2,3)], santa=[(2,5)], catarina=[(2,6)],
         disciplina=[(3,1)], estrutura=[(3,3)],
         dados=[(3,5)], 2=[(3,6)] }
```

Busca

- Se os termos de busca forem “santa, norte, de, disciplinas, udesc” no espaço de busca do índice invertido acima, o resultado seria:

```
santa: 2.txt; norte: 1.txt; de: 1.txt, 2.txt, 3.txt;
disciplina: 3.txt; udesc: não encontrado
```

EXERCÍCIOS

UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA

Exercícios - 1

- Crie um diretório e popule este diretório com alguns arquivos texto
- Escreva algumas frases dentro destes arquivos
- Salve os arquivos

Exercícios - 1

- Defina uma tabela Hash que tenha como chave uma palavra e como valor um vetor de arquivos, ambos serão Strings
- Leia os arquivos do diretório populado na etapa de pré-requisitos, utilize a classe File e seu método list()

Exercícios - 1

- Corra a lista de arquivos encontrados
  - Para cada arquivo encontrado, utilize a classe FileReader e BufferReader para ler as linhas dos arquivos
  - Para cada linha utilize o método split para separar as palavras da linha em um novo vetor de palavras
    - Verifique através do método get se a palavra já está mapeada na tabela Hash
      - Se não crie esta palavra (chave) no HashMap
      - Inclua o documento na lista de arquivos da tabela Hash
  - Teste a tabela hash fazendo a busca por algumas palavras e verificando se encontra no arquivo (faça um tratamento também para quando não encontrar)

Exercícios - 2

- Baixe do *GitHub* o exemplo da lista invertida e teste o seu funcionamento
- Tente entender o programa e faça modificações nas listas de arquivos a serem indexados e na lista de termos procurados
- Crie uma interface (tela ou *prompt*) que permita ao usuário adicionar novos arquivos e fazer novas buscas

Exercícios - 3

- Apresente a lista invertida gerada pela indexação dos quatro arquivos abaixo. Considere que cada chave da lista invertida aponta para uma lista contendo pares de valores indicando o número do arquivo (de 1 a 4) e a respectiva posição da palavra dentro do arquivo

```
Arq_1 = “Estruturas de Dados em Java”
Arq_2 = “Conceitos de Computação com Java”
Arq_3 = “Lógica de Programação e Estruturas de Dados em Java”
Arq_4 = “Computação em Nuvem”
```

Obrigado

jacksonpradolima.github.io
github.com/ceplan

UDESC
UNIVERSIDADE
DO ESTADO DE
SANTA CATARINA