

Curso:

Disciplina:

Código/Turma:

Professor/a:

Data:

Aluno/a:

Matrícula:

## INSTRUÇÕES PARA O DESENVOLVIMENTO DO TRABALHO

- Trabalho é em Equipe (até 4 pessoas).
- O Trabalho vale 30% da nota da disciplina e o trabalho será apresentado pela equipe. Será marcado um dia para a equipe mostrar a implementação e a execução da aplicação.

### *Implementar um índice hash estático*

**1. Interface gráfica obrigatória** ilustrando as estruturas de dados e o funcionamento de um índice hash estático.

**2. Funcionalidades principais:**

- a. Construção do índice;
- b. Busca por uma tupla a partir da entrada de uma chave de busca usando o índice construído;
- c. Fazer um table scan até encontrar a chave de busca fornecida.

**3. Estruturas a serem implementadas:**

- a. **Página:** estrutura de dados que representa a divisão e alocação física da tabela na mídia de armazenamento.
- b. **Bucket:** estrutura de dados que mapeia chaves de busca em endereços de páginas.
- c. **Função hash:** mapeia uma chave de busca em um endereço de bucket. Deve ser escolhida/projetada pela equipe.

**4. Parâmetros:**

- a) **Arquivos de dados:** contém os dados que serão usados para popular uma ou mais tabelas. Para este trabalho será usado um arquivo com 466 mil palavras do idioma Inglês, disponível em: <https://github.com/dwyl/english-words>. Este arquivo txt tem somente uma palavra por linha. Esta única palavra é única no arquivo, podendo ser considerada chave.
- b) **Tamanho da página:** **entrada de usuário** que determina o tamanho de cada página individualmente.
- c) **Quantidade de páginas:** (calculado) número máximo de páginas usadas <sup>1</sup> para dividir o conteúdo da tabela.
- d) **Número de buckets (NB):** (calculado) onde  $NB > NR / FR$ . NR é a cardinalidade da tabela (número de tuplas) e FR é o número de tuplas endereçadas por bucket.
- e) **Tamanho dos buckets (FR):** (definido pela equipe conforme função hash) número máximo de tuplas endereçadas por bucket, depende da função hash implementada.
- f) **Chave de busca de uma tupla específica:** depois que o índice é construído o usuário pode entrar com uma chave de busca para que o sistema retorne a tupla associada. Deve existir um local na interface gráfica para ser digitada a chave que será buscada. O registro retornado e em qual página está localizado devem ser mostrados na interface gráfica
- g) **Botão para o Table Scan:** Após o usuário digitar a chave de busca, habilita um botão que pode ser usado para fazer o Table Scan em que serão listados na interface os registros até encontrar o a chave de busca. Informar o custo da leitura destes dados.

<sup>1</sup> Se o usuário escolheu o tamanho da página então a quantidade de páginas é um parâmetro calculado. Os dois parâmetros são mutuamente exclusivos como entrada.

## 5. Problemas:

- A implementação do índice deve levar em consideração as colisões, ou seja, deve ser implementado um algoritmo de resolução de colisões.
- A implementação do índice deve levar em consideração o transbordamento dos buckets (bucket overflow), ou seja, deve ser implementado um algoritmo de resolução de overflow.

## 6. Estatísticas:

- Deve ser calculada e mostrada a taxa de colisões (%).
- Deve ser calculada e mostrada a taxa de overflows (%).
- Deve ser calculado e mostrado uma estimativa de custo (quantidade de acessos a disco – leitura de páginas), quando uma chave de busca é entrada.
- Deve ser calculado e mostrado a estimativa de custo (quantidade de páginas lidas) para o table scan.
- Deve ser mostrada a diferença de tempo entre a busca usando o índice e o table scan.

## 7. Funcionamento em passos:

Carga do programa:

- O arquivo de dados é carregado em memória.
- As linhas da tabela devem ser divididas em páginas, de acordo com o tamanho das páginas.
- Deve ser mostrada na interface a primeira e a última página carregada, mostrando o nome/numero da página e os registros que estão em cada uma das páginas.
- NB buckets de tamanho FR são criados ( $NB > NR/FR$ , onde NR representa o número de tuplas e FR representa o número de tuplas por Bucket).
- Para a construção do índice deve-se percorrer página por página, pegando cada uma das chaves de busca que foram carregadas nas páginas.
- A função hash é aplicada à chave de busca de cada tupla; a chave de busca e o endereço da página onde a tupla foi armazenada são adicionadas ao bucket cujo endereço foi calculado pela função hash.

Uso do programa:

- Uma chave de busca é fornecida na interface.
- A função hash deve ser aplicada na chave de busca fornecida encontrando a página de dados a ser lida
- A página deve ser lida e será informado na interface gráfica se a chave foi encontrada, o número da página e o custo da leitura.
- O botão table scan deve ser acionado e o resultado da leitura das páginas deve ser mostrado na interface gráfica, além do número da página onde a chave foi encontrada e o custo da leitura do table scan.
- Deve ser mostrada a diferença de tempo entre a busca usando a chave e o table Scan.

Critério	Notas
1. Interface gráfica	1,0
2. Carga de Dados nas páginas	1,5
3. Entrada para Tamanho da página	1,0
4. Cálculo da quantidade de páginas	1,0
5. Construção e uso da função hash	1,0
6. Cálculo da quantidade de buckets	0,5
7. Funcionamento com pesquisa com o uso do índice	2,0
8. Deve ser calculada e mostrada a taxa de colisões.	0,5
9. Deve ser calculada e mostrada a taxa de overflows.	0,5
10. Execução do Table Scan	0,5
11. Deve ser calculado e mostrado uma estimativa de custo (acessos a disco) e o comparativo de tempo entre a busca com índice e o table scan	0,5