

## **2º Trabalho prático de Banco de Dados**

UFAM - CIÊNCIA DA COMPPUTAÇÃO - 6º Período

# **Árvore B+ & Hashing**

## **Integrantes**

- Helder Medeiros
- Matheus Marques
- Vitor Hugo

# Notas:

- As questões estão já pré-feitas em arquivos Bash, na pasta /questoes/. Contudo, ainda é possível realizar testes diferentes (veja abaixo).
- O arquivo principal “db” está no PATH do container, então basta acessar a pasta /csv\_files/ e digitar relativamente ao caminho
  - Ex: db --insert entrada.csv
  - Ex<sup>2</sup>: db --timed --reduced --find entrada.db/ 1000
  - \*digite db --help para mais configurações.

## Estruturas

### Hashing

*O hashing foi primariamente feito baseando-se em structs (como visto no código comentado pelo programa), para manter em um único arquivo, e sendo acessado por meio de seek()'s, porém o tempo para realizar o hashing de 1M estava maior que meia hora. Por isso, foi realizado o seguinte método:*

- O hashing foi realizado utilizando uma técnica de Hash duplo.
- Seus mods, após alguns testes, foram decididos para 127 e 131, ambos primos.
- Cada um dos buckets da primeira parte do hash correspondem à uma pasta no sistema, dentro da pasta *arquivo.db*.
- E dentro de cada uma destas pastas, tem-se arquivos do tipo “.b”, que são os buckets do segundo hash.
- As colisões são checadas sequencialmente no arquivo .b final da busca.

## Árvore B+ (não implementada totalmente)

*A árvore foi armazenada em disco através de “structs” da linguagem C. Elas consistem em ponteiros entre os nós, um seek (para encontrar diretamente no arquivo csv original) e 2 ID’s, um do nó e outro do registro.*

- Foi adotado uma estratégia sem nenhum include externo para o tipo de dados, utilizando simplesmente o default do C++.
- A ramificação foi feita baseada no lado direito, tendo os nós iniciais à esquerda dos conjuntos de registros.

## Funções

**Fonte: *helper.cpp***

**Funções:**

- `bool has_suffix(const std::string &str, const std::string &suffix);`
  - Desenvolvedor: Matheus Marques;
  - Papel: Checa o sufixo do arquivo se é “.csv”;
- `void clrscr();`
  - Desenvolvedor: Matheus Marques;
  - Papel: Limpar a tela. (VISUAL)
- `bool isNumber(string s);`
  - Desenvolvedor: Matheus Marques;
  - Papel: Checar se o argumento é um número com fins de validação;

**Fonte: *hashing.cpp***

**Funções:**

- `int insertHash(string folder, string line, int mod1, int mod2);`
  - Desenvolvedor: Vitor Hugo e Matheus Marques;
  - Papel: Inserção por hash;
- `string findHash(string folder, int id, int mod1, int mod2);`
  - Desenvolvedor: Helder Medeiros e Matheus Marques;
  - Papel: Busca por hash.

## Fonte: bplus.cpp

### Funções:

- `node getNode(FILE *f, int ponteiro);`
  - Desenvolvedor: Matheus Marques e Helder Medeiros
  - Papel: Função para retirar o nó do arquivo de B+.
- `bool writeRoot(FILE *f, int ponteiro);`
  - Desenvolvedor: Vitor Hugo e Helder Medeiros
  - Papel: Função de escrever na primeira seção do arquivo, o nó raiz.
- `int getRoot(FILE *f);`
  - Desenvolvedor: Vitor Hugo
  - Papel: Retornar a primeira linha, que é o nó raiz.
- `int sizeofPack(FILE *f, int first);`
  - Desenvolvedor: Vitor Hugo
  - Papel: Retornar o tamanho do "array" de nós da B+.
- `bool writeNode(FILE *f, node nd);`
  - Desenvolvedor: Helder Medeiros
  - Papel: Escrever um nó no final do arquivo da B+;
- `bool change<X>(FILE *f, int ponteiro, int new_<X>);`
  - Desenvolvedor: Helder Medeiros
  - Papel: Modificar em disco um atributo do nó
  
- `bool subirMeio(FILE *f, int firstOfPack, int ponteiro_atual);`
  - Desenvolvedor: Matheus Marques
  - Papel: Realizar a ascensão do atributo do meio, quando o "array" estourar.
- `bool addNode(FILE *f, node nd);`
  - Desenvolvedor: Matheus Marques
  - Papel: Adicionar o nó, envolvendo toda a lógica da estrutura de dados de uma B+, no arquivo.

## Fonte: main.cpp

### Funções:

- `void *t_progress(void *lpParam);`
  - Desenvolvedor: Matheus Marques;
  - Papel: Thread para demonstrar o progresso
  -
- `int readHelp();`
  - Desenvolvedor: Matheus Marques;
  - Papel: Retorna um 'help';
- `string create_db_folder(string csv_name);`
  - Desenvolvedor: Matheus Marques;
  - Papel: Cria uma pasta para o csv, com o sufixo ".db";
- `string normalSearch(string csv, int id);`
  - Desenvolvedor: Vitor Hugo;
  - Papel: Busca sequencial usada para comparar com os outros tipos de busca implementados no trabalho.