

Relatório do Trabalho Prático II

Davi R. Cruvel¹, Emidio J. Aranha¹, Enzo H. Albuquerque¹

¹Instituto de Computação – Universidade Federal do Amazonas (UFAM)

{drcc,ejacf,ehsa}@icomp.ufam.edu.br

1. Estruturas

Estrutura do artigo:

```
typedef struct {  
    unsigned int id;  
    char titulo[300];  
    unsigned int ano;  
    char autor[100];  
    unsigned int citacoes;  
    char atualizacao[20];  
    char snippet[1000];  
}Artigo;
```

Estrutura do Bloco:

```
typedef struct {  
    unsigned int numRegistros;  
    char cabecalho[365];  
    char corpo[3724];  
}Bloco;
```

2. Fontes dos programas

O programa upload é responsável por fazer a carga inicial dos dados através da leitura de um csv, ele é composto pelos fontes: parser.hpp, arvoreBMaisPrimaria.hpp e arvoreB-MaisSecundaria.hpp. O programa findrec é responsável por buscar um registro através do ID informado, ele é formado pelo fonte parser.hpp.

O seek1 retorna um registro também pelo ID pesquisando em um arquivo de índice primário, ele é formado pelos fontes: arvoreBMaisPrimaria.hpp, parser.hpp. E o seek2 retorna um registro pelo seu título pesquisando em um arquivo de índice secundário, ele é formado pelos fontes: arvoreBMaisSecundaria.hpp, parser.hpp.

3. Funções das fontes:

arvoreBMaisPrimaria.cpp:

```
void atualizaBlocoNoDisco(tipoNoInterno *bloco);  
void atualizaCabecario();  
tipoNoInterno* alocaNovoNoInterno();  
tipoNoInterno* carregaBlocoIndiceDoDisco(int posicao);  
void carregaCabecario();  
int adicionaNovoBlocoNoDisco(tipoNoInterno *bloco);  
void criaCabecario();
```

```

int abreArquivoIndices(const char *caminhoArquivoIndice);
int abreArquivoDados(const char *caminhoArquivoDados);
void fechaArquivos();
void criaArquivoIndice(const char *caminhoArquivoDados, const char *caminhoArquivoIndice);
void povoaArquivoIndices();
void insereFolhaComEspaco(tipoNoInterno *no, int chave, int ponteiro);
void insereInternoComEspaco(tipoNoInterno *no, int chave, int ponteiro);
tipoNoAuxiliar* insereFolhaSemEspaco(tipoNoInterno *no, int chave, int ponteiro);
tipoNoAuxiliar* insereInternoSemEspaco(tipoNoInterno *no, int chave, int ponteiro);
tipoNoAuxiliar* insereNaArvoreInterno(tipoNoInterno *no, int chave, int ponteiro);
void insereNaArvore(int chave, int ponteiro);
int buscaNaArvore(const char *caminhoArquivoDados, const char *caminhoArquivoIndice, int chave);
void seek1(const char *caminhoArquivoDados, const char *caminhoArquivoIndice, int chave);

```

arvoreBMaisSecundaria.cpp:

```

void insereNaArvore2(char chave[300], int ponteiro);
void atualizaBlocoNoDisco2(tipoNoInterno2 *bloco);
void atualizaCabecalho2();
tipoNoInterno2* alocaNovoNoInterno2();
tipoNoInterno2* carregaBlocoIndiceDoDisco2(int posicao);
void carregaCabecalho2();
int adicionaNovoBlocoNoDisco2(tipoNoInterno2 *bloco);
void criaCabecalho2();
int abreArquivoIndices2(const char *caminhoArquivoIndice);
int abreArquivoDados2(const char *caminhoArquivoDados);
void fechaArquivos2();
void criaArquivoIndice2(const char *caminhoArquivoDados, const char *caminhoArquivoIndice);
void povoaArquivoIndices2();
void uploadBMaisSecundaria(const char *caminhoArquivoDados, const char *caminhoArquivoIndice);
void insereFolhaComEspaco2(tipoNoInterno2 *no, char chave[300], int ponteiro);
void insereInternoComEspaco2(tipoNoInterno2 *no, char chave[300], int ponteiro);
tipoNoAuxiliar2* insereFolhaSemEspaco2(tipoNoInterno2 *no, char chave[300], int ponteiro);
tipoNoAuxiliar2* insereInternoSemEspaco2(tipoNoInterno2 *no, char chave[300], int ponteiro);
tipoNoAuxiliar2* insereNaArvoreInterno2(tipoNoInterno2 *no, char chave[300], int ponteiro);
void insereNaArvore2(char chave[300], int ponteiro);
int buscaNaArvore2(const char *caminhoArquivoDados, const char *caminhoArquivoIndice, char chave[300]);
void seek2(const char *caminhoArquivoDados, const char *caminhoArquivoIndice, char chave[300]);

```

parser.cpp:

```
void copyStringToStr(char* a, string b, int tam);
void imprimirArtigo(Artigo art);
void copiaArtigo(Artigo *destino, Artigo *fonte);
void pause();
string getCampo(ifstream *arq);
Bloco getBucketHashFileByID(fstream *arq, int id);
Bloco * getBucketHashFileByPosition(fstream *arq, int position);
Artigo* getArtigoByPositionID(fstream *arq, int position, int id);
Artigo* getArtigoByPositionTitle(fstream *arq, int position, char title[300]);
int hashSimples(int id);
int inserirNoHashFile(fstream *arq, Artigo artigo);
Artigo findrec(fstream *arq, int id, bool imprimir);
Artigo getArtigo(ifstream *arq);
int getColisoes();
```

4. Trabalho da Equipe:

Cada integrante ficou responsável por 3 ou 4 fontes no trabalho, e com isso o desenvolvimento das funções responsáveis pelo funcionamento, a divisão de tarefas sucedeu-se assim:

Davi:

```
arvoreBMaisPrimaria.cpp
arvoreBMaisPrimaria.hpp
seek1.cpp
```

Emidio:

```
parser.cpp
parser.hpp
seek2.cpp
upload.cpp
```

Enzo:

```
arvoreBMaisSecundaria.cpp
arvoreBMaisSecundaria.hpp
findrec.cpp
```

5. Papel das Funções:

- **void atualizaBlocoNoDisco2(tipoNoInterno2 *bloco);** //atualiza os blocos no disco com um arquivo de índice secundário.
- **void atualizaCabecalho2();** //atualiza o cabeçalho com um arquivo de índice secundário.
- **tipoNoInterno2* alocaNovoNoInterno2();** //Caso seja precisa essa função aloca um novo nó interno.

- **tipoNoInterno2* carregaBlocoIndiceDoDisco2(int posicao);** // Carrega um bloco de índice do arquivo de índice secundário.
- **void carregaCabecalho2();** // Carrega um cabeçalho do arquivo de índice secundário.
- **int adicionaNovoBlocoNoDisco2(tipoNoInterno2 *bloco);** // Caso seja necessário, esta função adiciona novos blocos no disco.
- **void criaCabecalho2();** //Cria um cabeçalho e atualiza o mesmo.
- **int abreArquivoIndices2(const char *caminhoArquivoIndice);** // Abre um arquivos de índices secundário.
- **int abreArquivoDados2(const char *caminhoArquivoDados);** // Abre um arquivo de dados.
- **void fechaArquivos2();** //Fecha um arquivo de dados e índice secundário.
- **void criaArquivoIndice2(const char *caminhoArquivoDados, const char *caminhoArquivoIndice);** //Cria um novo arquivo de índices secundário.
- **void povoaArquivoIndices2();** //Insere os índices no arquivo de índices secundário.
- **void uploadBMaisSecundaria(const char *caminhoArquivoDados, const char *caminhoArquivoIndice);** // Dar um upload na árvore B+ secundária.
- **void insereFolhaComEspaco2(tipoNoInterno2 *no, char chave[300], int ponteiro);** // Insere uma folha na árvore B+ secundária.
- **void insereInternoComEspaco2(tipoNoInterno2 *no, char chave[300], int ponteiro);** // Insere um nó interno em um grupo com espaços na árvore
- **tipoNoAuxiliar2* insereFolhaSemEspaco2(tipoNoInterno2 *no, char chave[300], int ponteiro);** // Insere uma folha em um grupo sem espaços na árvore.
- **tipoNoAuxiliar2* insereInternoSemEspaco2(tipoNoInterno2 *no, char chave[300], int ponteiro);** // Insere um nó interno em um grupo sem espaços na árvore.
- **tipoNoAuxiliar2* insereNaArvoreInterno2(tipoNoInterno2 *no, char chave[300], int ponteiro);** // Insere na parte mais interna da árvore B+ secundária
- **void insereNaArvore2(char chave[300], int ponteiro);** // Insere um dado na árvore B+ secundária

- **int buscaNaArvore2(const char *caminhoArquivoDados,const char *caminhoArquivoIndice, char chave[300]);** // Busca um dado na árvore B+ secundária.
- **void seek2(const char *caminhoArquivoDados,const char *caminhoArquivoIndice, char chave[300]);** // Chama o programa seek2 que busca na árvore pelo título, através de um arquivo de índice secundário.
- **void copyStringToStr(char* a, string b, int tam);** //Passa o conteúdo de **b** para **a** via ponteiro.
- **void imprimirArtigo(Artigo art);** //Imprime os dados do artigo em questão.
- **void copiaArtigo(Artigo *destino, Artigo *fonte);** //Faz uma copia do artigo.
- **void pause();** //Pausa o parser.
- **string getCampo(ifstream *arq);** //Pega os campos do argumento passado. verifica se possui aspas, se há espaçamento ou puladas de linha e ignora salvando somente o conteúdo requerido.
- **Bloco getBucketHashFileByID(fstream *arq, int id);** // Faz uma busca via ID no hashfile e retorna o bucket com ID passado.
- **Bloco * getBucketHashFileByPosition(fstream *arq, int position);** Retorna o bucket que está na posição position no hashfile
- **Artigo* getArtigoByPositionID(fstream *arq, int position, int id);** //Pega o artigo através da posição em que se encontra no hashfile e verifica se o id do artigo e o id de entrada são iguais.
- **Artigo* getArtigoByPositionTitle(fstream *arq, int position, char title[300]);** //Pega o artigo no arquivo hash através da posição em que se encontra e verifica se o título do artigo e o título de entrada são iguais.
- **Artigo getArtigo(ifstream *arq) ;** //Retorna o artigo do arquivo
- **int hashSimples(int id) ;** //Retorna o bucket que possui o id de entrada.
- **int inserirNoHashFile(fstream *arq, Artigo artigo);** //Insere no arquivo hash o ponteiro que identifica tal artigo
- **Artigo findrec(fstream *arq,int id, bool imprimir);** // Busca no arquivo de dados por um registro com o ID informado, se existir, e retornar os campos do registro, a quantidade de blocos lidos para encontrá-lo e a quantidade total de bloco do arquivo de dados;
- **int getColisoies();** //Retorna as colisões no arquivo hash.

- **void atualizaBlocoNoDisco(tipoNoInterno *bloco);** //Atualiza o bloco no disco que foi passado no parâmetro.
- **void atualizaCabecario();** //Atualiza o cabeçalho no ponto em que a função é chamada.
- **tipoNoInterno* alocaNovoNoInterno();** //Função que cria um novo nó interno, iniciando todas as variáveis do nó.
- **tipoNoInterno* carregaBlocoIndiceDoDisco(int posicao);** //É carregado um bloco do disco pela posição passada no parâmetro.
- **void carregaCabecario();**//Aloca memória para o cabeçalho
- **int adicionaNovoBlocoNoDisco(tipoNoInterno *bloco);** //É passado um bloco como parâmetro da função e ela o adiciona no disco.
- **void criaCabecario();** //Aloca espaço para o cabeçalho e inicializa seus campos;
- **int abreArquivoIndices(const char *caminhoArquivoIndice);** // Função que abre um arquivo de índices.
- **int abreArquivoDados(const char *caminhoArquivoDados);** // Função que abre um arquivo de dados.
- **void fechaArquivos();** // Função que fecha um arquivo, liberando memória.
- **void criaArquivoIndice(const char *caminhoArquivoDados, const char *caminhoArquivoIndice);** // Lê um caminho de arquivo de índice e a partir dele é criado o arquivo de índice.
- **void povoaArquivoIndices();** // Insere registros num arquivo.
- **void insereFolhaComEspaco(tipoNoInterno *no, int chave, int ponteiro);** // Insere um nó interno na árvore quando tem espaço para isso.
- **tipoNoAuxiliar* insereFolhaSemEspaco(tipoNoInterno *no, int chave, int ponteiro);** // Insere uma folha na árvore quando não tem espaço para ela, fazendo as manipulações necessárias.
- **tipoNoAuxiliar* insereInternoSemEspaco(tipoNoInterno *no, int chave, int ponteiro);** // Função usada pra inserir um nó interno quando não se tem espaço na árvore, fazendo as manipulações necessárias para ser possível a inserção correta do nó.
- **void insereNaArvore(int chave, int ponteiro);** // Faz uma inserção simples na árvore.

- **int buscaNaArvore(const char *caminhoArquivoDados,const char *caminhoArquivoIndice, int chave);** // A função faz uma busca pela árvore, buscando a chave passada no parâmetro.
- **void seek1(const char *caminhoArquivoDados,const char *caminhoArquivoIndice, int chave);** // Faz uma chamada ao programa seek1 que busca na árvore o id informado através de um arquivo de índice primário.