#### Etapa 3 - Hashing Extensível

O objetivo desta etapa é unir as etapas anteriores, utilizando uma trie para armazenar o diretório dos blocos no arquivo. Com isso, como é possível agora estender o diretório, há a possibilidade de tratar colisões quando o bloco fica cheio. Nesse caso, o bloco deverá ser dividido e o tamanho da chave ampliado (para todos os elementos).

Ao ser aberto, o programa abrirá o arquivo e permitirá as seguintes operações:

1. Inserir um novo objeto no arquivo.

2. Remover um objeto do arquivo. Fica a critério do grupo não apagar o objeto diretamente, mas marcar o espaço para reutilização. Nesse caso, a inserção deverá obrigatoriamente reutilizar espaços disponibilizados por remoção.

3. Consultar a um objeto no arquivo, usando busca binária ou sequencial.

4. Imprimir o arquivo, com todo seu conteúdo, na ordem de armazenamento.

5. Imprimir os registros de um dado bloco de modo ordenado, caso o armazenamento não seja feito de forma ordenada.

Os dados deverão ser armazenados em um arquivo de blocos indexado por uma tabela hash armazenada em uma trie extensível. Cada bloco irá armazenar até TAM\_BLOCO registros, sendo que TAM\_BLOCO é um valor constante definido no código. Para este projeto prático, considere que cada bloco armazena **quatro registros**.

Para efeito de simplificação, recomenda-se que o diretório dos dados (a tabela hash / trie) seja armazenada em um arquivo auxiliar (que pode ser binário ou texto) e seja carregado integralmente para a memória primária. O arquivo de dados nunca é lido integralmente, mas um bloco por vez, com todos os registros contidos neste bloco. Ao procurar um dado registro, por exemplo, pode ser necessário percorrer o bloco inteiro, caso os dados não sejam armazenados de forma ordenada em cada bloco. Em qualquer momento da execução do sistema, no máximo três blocos poderão estar alocados na memória primária.

O diretório de dados armazena, para cada chave, o endereço relativo do bloco no arquivo. O hash será calculado a partir do ID do objeto sendo armazenado, gerando uma codificação em **número binário**. A forma de representação desse binário é de escolha do grupo (vetor de caracteres, um inteiro, etc.). A partir dessa chave, será associada uma posição relativa (índice do bloco no vetor de blocos) no disco contendo o bloco de dados. A chave possui tamanho redimensionável, sendo que, quando da criação do primeiro bloco, ela tem tamanho 1, sendo este tamanho ampliado para atender necessidades de novas inserções. Poderá ser assumido, a critério do grupo, que o diretório nunca terá mais de MAX\_DIR entradas, em que MAX\_DIR será um valor constante definido no código.

|  |
| --- |
| **Data de Entrega da Etapa 3:** 13 de julho de 2018 |