

Operações em arquivos indexados



Interface do CRUD

- `ID ← arquivo.create(novo_objeto)`
- `objeto ← arquivo.read(ID)`
- `ok ← arquivo.update(objeto_atualizado)`
- `ok ← arquivo.delete(ID)`

O ID passado nesse novo objeto será ignorado (usar -1)

Observação: A interface do CRUD não deve oferecer acesso direto aos índices usados

O ID deve permanecer o mesmo, pois IDs nunca são alterados.

Create

```
01: algoritmo create(objeto)
02:   mover o ponteiro para início do arquivo (cabeçalho)
03:   ler últimoID
04:   objeto.ID  $\leftarrow$  últimoID + 1
05:   mover o ponteiro para início do arquivo
06:   escrever objeto.ID
07:   criar registro para o objeto
08:   mover para o fim do arquivo
09:   pos  $\leftarrow$  posição do ponteiro
10:   escrever registro
11:   inserir o par (objeto.ID, pos) no índice
12: fim-algoritmo
```

Read

```
01: algoritmo read(ID)
02:   pos  $\leftarrow$  buscar o ID no índice
03:   se pos  $\neq$  -1
04:     mover ponteiro para pos
05:     ler registro
06:     se registro.lapide  $\neq$  '*' // checagem dupla
07:       então extrair objeto do registro
08:       retornar objeto e terminar
09:     fim-se
10:   fim-se
11:   retornar objeto vazio // null
12: fim-algoritmo
```

Delete

```
01: algoritmo delete(ID)
02:   pos ← buscar o ID no índice
03:   se pos ≠ -1
04:     então mover ponteiro para pos
05:     ler registro
06:     se registro.lapide ≠ '*'
07:       então extrair objeto do registro
08:       mover para pos
09:       escrever lápide como excluído
10:       remover o ID do índice
11:       retornar verdadeiro e terminar
12:     fim-se
13:   fim-se
14:   retornar falso
15: fim-algoritmo
```

Update

```
01: algoritmo update(novoObjeto)
02:   pos ← buscar o ID no índice
03:   se pos ≠ -1
04:     então mover ponteiro para pos
05:     ler registro
06:     se registro.lapide ≠ '*'
07:       então extrair objeto do registro
08:       criar novoRegistro para novoObjeto
09:       se novoRegistro.tamanho ≤ registro.tamanho
10:         então mover para pos
11:         escrever novoRegistro (manter ind.tam.)
12:       senão mover para pos
13:         escrever lápide como excluído
14:         mover para fim do arquivo
15:         pos ← posição do ponteiro
16:         escrever novoRegistro
17:         atualizar o endereço para o ID no índice
18:         retornar verdadeiro e terminar
19:       fim-se
20:     fim-se
21:   fim-se
22:   retornar falso
23: fim-algoritmo
```

Considerações finais

- O algoritmo apresentado possui apenas um índice secundário (por ID), direto e denso. Se os arquivos tiverem mais índices, esse algoritmo deve ser modificado.
- Se houver algum índice indireto, a atualização deve ser feita por meio da remoção do par (chave, ID) no índice e inserção do novo par, pois o valor de chave pode ter sido alterado.
- Os índices podem ser gerenciados fora do CRUD, mas isso requer muita atenção extra.