

# Operações em arquivos sequenciais



## 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)

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

# Lápide

- Marca de exclusão
  - Campo (1 byte) que indica se o registro foi excluído ou permanece válido
- Os registros só são realmente apagados do arquivo quando há uma reordenação

## Exemplo de arquivo

LÁPIDE; TAMANHO; DADOS (ID; Nome; Idade)

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
			3		' '			21			1			13	'C'	'A'	'R'	'L'	'O'	'S'	' '	

Byte	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
	'C'	'H'	'A'	'G'	'A'	'S'		25	' '		17				2			9		'A'	'N'	'A'

Byte	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65
	' '	'M'	'A'	'R'	'I'	'A'		31	'*'		21				3			13		'S'	'I'	

Byte	66	67	68	69	70	71	72	73	74	75	76	77
	'L'	'V'	'I'	'O'	' '	'C'	'O'	'S'	'T'	'A'	27	

## Create (inclusão)

- Arquivo sem ordenação específica

```
01: algoritmo create(objeto)
02:   | mover o ponteiro para início do arquivo (cabeçalho)
03:   | ler últimoID
04:   | objeto.ID ← 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:   | escrever registro
10: fim-algoritmo
```

## Read (leitura)

- Algoritmo de busca sequencial

```
01: algoritmo read(ID)
02:   | mover o ponteiro para o primeiro registro (após o cabeçalho)
03:   | enquanto não atingir o fim do arquivo
04:   |   | ler próximo registro
05:   |   | se registro.lapide ≠ '*'
06:   |   |   | então extrair objeto do registro
07:   |   |   |   | se objeto.ID = ID
08:   |   |   |   |   | então retornar objeto e terminar
09:   |   |   |   |   | fim-se
10:   |   |   |   | fim-se
11:   |   | fim-enquanto
12:   | retornar objeto vazio // null
13: fim-algoritmo
```

# Read (leitura)

- **Variação:** Leitura de conjunto de objetos

```
01: algoritmo read(critérios)
02:   |   criar conjunto vazio
03:   |   mover o ponteiro para o primeiro registro (após o cabeçalho)
04:   |   enquanto não atingir o fim do arquivo
05:   |       |   ler próximo registro
06:   |       |   se registro.lapide ≠ '*'
07:   |       |       |   então extrair objeto do registro
08:   |       |       |       |   se registro atender aos critérios
09:   |       |       |       |       |   então adicionar objeto ao conjunto
10:   |       |       |       |   fim-se
11:   |       |   fim-se
12:   |   fim-enquanto
13:   |   retornar conjunto
14: fim-algoritmo
```

# Delete (exclusão)

```
01: algoritmo delete(ID)
02:   |   mover o ponteiro para o primeiro registro (após o cabeçalho)
03:   |   enquanto não atingir o fim do arquivo
04:   |       |   pos ← posição do ponteiro
05:   |       |   ler próximo registro
06:   |       |   se registro.lapide ≠ '*'
07:   |       |       |   então extrair objeto do registro
08:   |       |       |       |   se objeto.ID = ID
09:   |       |       |       |       |   então mover para pos
10:   |       |       |       |       |       |   escrever lápide como excluído
11:   |       |       |       |       |       |   retornar verdadeiro e terminar
12:   |       |       |   fim-se
13:   |       |   fim-se
14:   |   fim-enquanto
15:   |   retornar falso
16: fim-algoritmo
```

# Update (atualização)

- Se a alteração **não** implicar em mudança de tamanho
  - Escrever o registro alterado na mesma posição
- Se o registro diminuir de tamanho
  - Escrever o registro na mesma posição, mas mantendo o mesmo indicador de tamanho
- Se o registro aumentar de tamanho
  - Marcar o registro atual como excluído e criar um novo registro no fim do arquivo

```
01: algoritmo update(novoObjeto)
02:   mover para o primeiro registro do arquivo (após cabeçalho)
03:   enquanto não atingir o fim do arquivo
04:     pos ← posição do ponteiro
05:     ler próximo registro
06:     se registro.lapide ≠ '*'
07:       então extrair objeto do registro
08:       se objeto.ID = novoObjeto.ID
09:         então criar novoRegistro para novoObjeto
10:           se novoRegistro.tamanho ≤ registro.tamanho
11:             então mover para pos
12:               escrever novoRegistro mantendo ind.tam.
13:             senão mover para pos
14:               escrever lápide como excluído
15:               mover para fim do arquivo
16:               escrever novoRegistro
17:             fim-se
18:           retornar verdadeiro e terminar
19:         fim-se
20:       fim-se
21:     fim-enquanto
22:     retornar falso
23: fim-algoritmo
```

## Considerações finais

- Arquivos sequenciais dependem de acesso sequencial, o que significa que eles não são bons para quaisquer operações de acesso aleatório
- Os espaços deixados pelos registros excluídos são espaços que podem ser reaproveitados (desde que exista uma lógica para esse reaproveitamento)
- O arquivo deve ser reordenado sempre que necessário (ordenação externa)