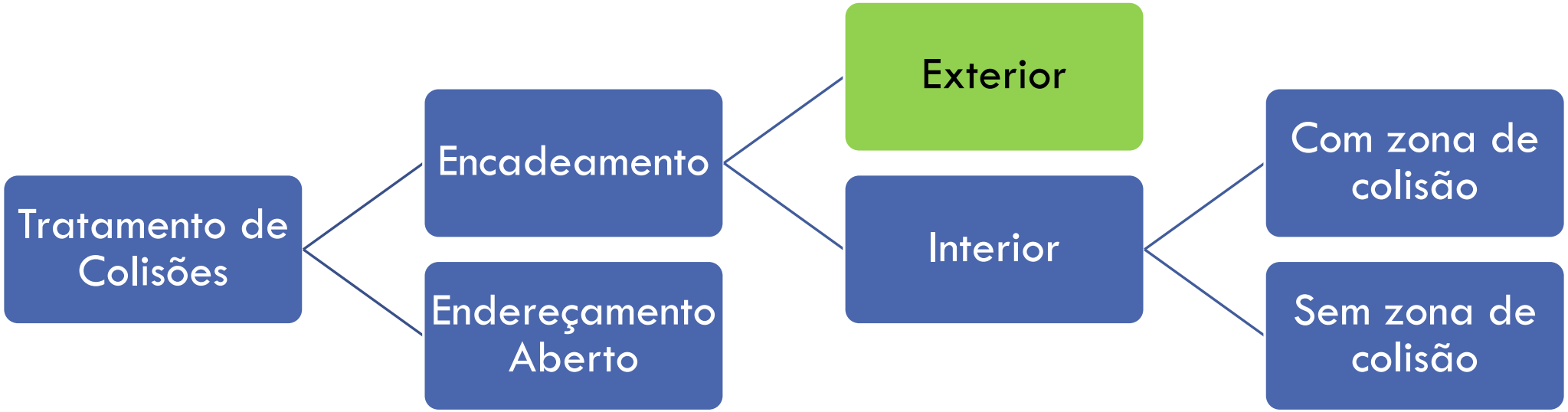
TABELAS HASH
TRATAMENTO DE COLISÕES POR ENCADEAMENTO EXTERIOR

Vanessa Braganholo

Estruturas de Dados e Seus Algoritmos


Tratamento de Colisões
Encadeamento
Endereçamento Aberto
Exterior
Interior
Com zona de colisão
Sem zona de colisão

ENCADEAMENTO EXTERIOR

Manter **m** listas encadeadas, uma para cada possível endereço base

A tabela base não possui nenhum registro, apenas os ponteiros para as listas encadeadas

Por isso chamamos de encadeamento **exterior**: a tabela base não armazena nenhum registro

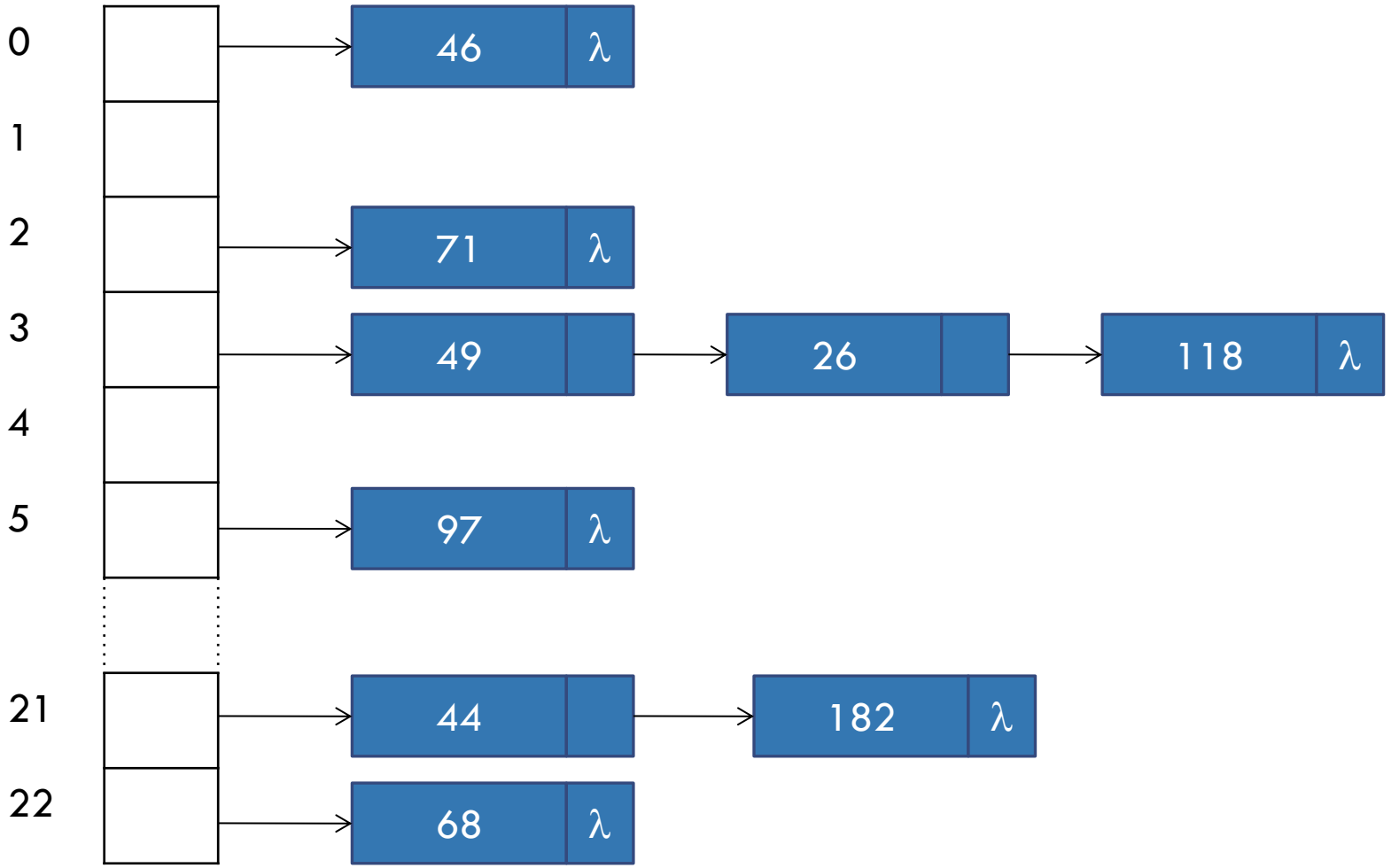
NÓS DA LISTA ENCADEADA

Cada nó da lista encadeada contém:

- um registro
- um ponteiro para o próximo nó

EXEMPLO: ENCADEAMENTO EXTERIOR

h(x) = x mod 23



Fonte: Fig. 10.5, pag 240

BUSCA

Busca por um registro de chave x:

1. Calcular o endereço aplicando a função h(x)
2. Percorrer a lista encadeada associada ao endereço
3. Comparar a chave de cada nó da lista encadeada com a chave x, até encontrar o nó desejado
4. Se final da lista for atingido, registro não está lá

INSERÇÃO

Inserção de um registro de chave x

1. Calcular o endereço aplicando a função h(x)
2. Buscar registro na lista associada ao endereço h(x)
3. Se registro for encontrado, sinalizar erro
4. Se o registro não for encontrado, inserir no final da lista

EXCLUSÃO

Exclusão de um registro de chave x

1. Calcular o endereço aplicando a função h(x)
2. Buscar registro na lista associada ao endereço h(x)
3. Se registro for encontrado, excluir registro
4. Se o registro não for encontrado, sinalizar erro

COMPLEXIDADE NO PIOR CASO

É necessário percorrer uma lista encadeada até o final para concluir que a chave não está na tabela

Comprimento de uma lista encadeada pode ser O(n)

Complexidade no pior caso: **O(n)**

COMPLEXIDADE NO CASO MÉDIO

Assume que função hash é uniforme

Número médio de comparações feitas na **busca sem sucesso** é igual ao fator de carga da tabela $\alpha = n/m$

Número médio de comparações feitas na **busca com sucesso** também é igual a $\alpha = n/m$

Se assumirmos que o número de chaves **n** é proporcional ao tamanho da tabela **m**

- $\alpha = n/m = O(1)$
- **Complexidade constante!**

IMPLEMENTAÇÃO EM MEMÓRIA PRINCIPAL

Ver implementação no site da disciplina

IMPLEMENTAÇÃO EM DISCO

Normalmente, usa-se um arquivo para armazenar os compartimentos da tabela, e outro para armazenar as listas encadeadas

Ponteiros para NULL são representados por -1

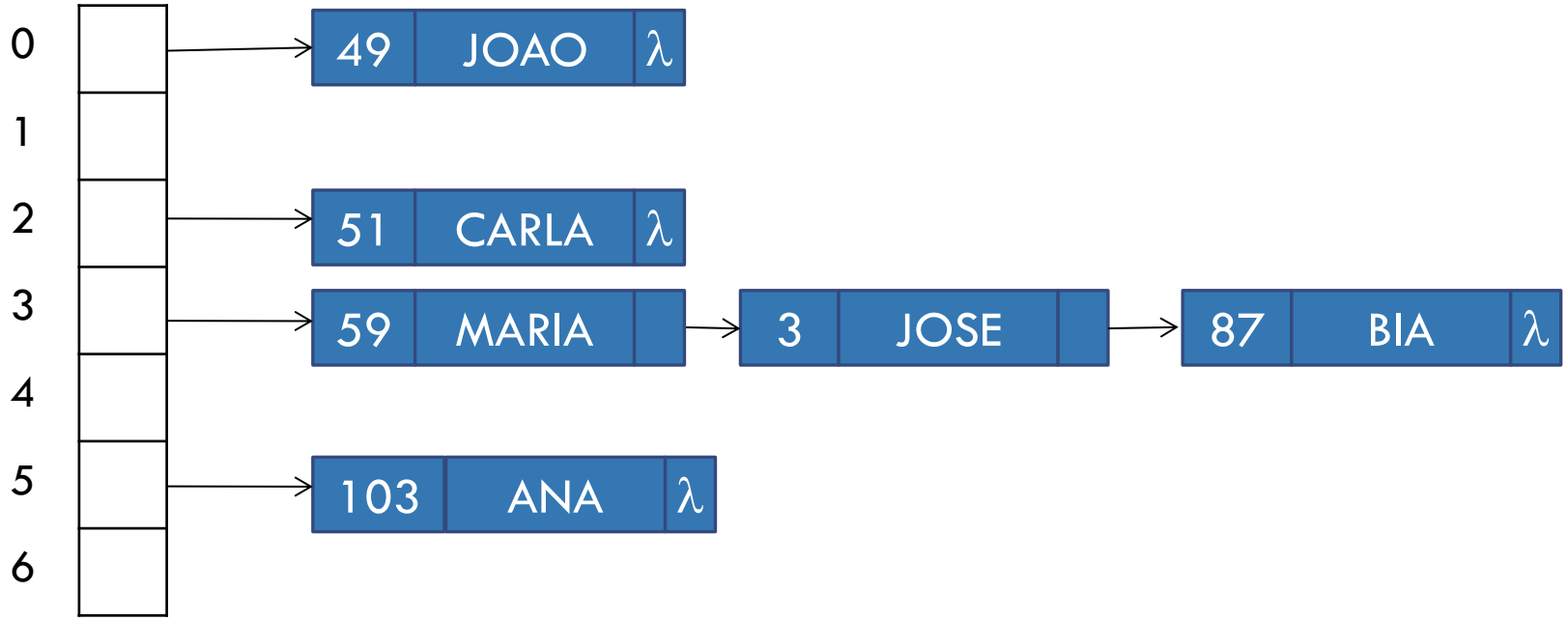
USO DE FLAG INDICADOR DE STATUS

Para facilitar a manutenção da lista encadeada, pode-se adicionar um flag indicador de **status** a cada registro (chamaremos esse flag de **ocupado**)

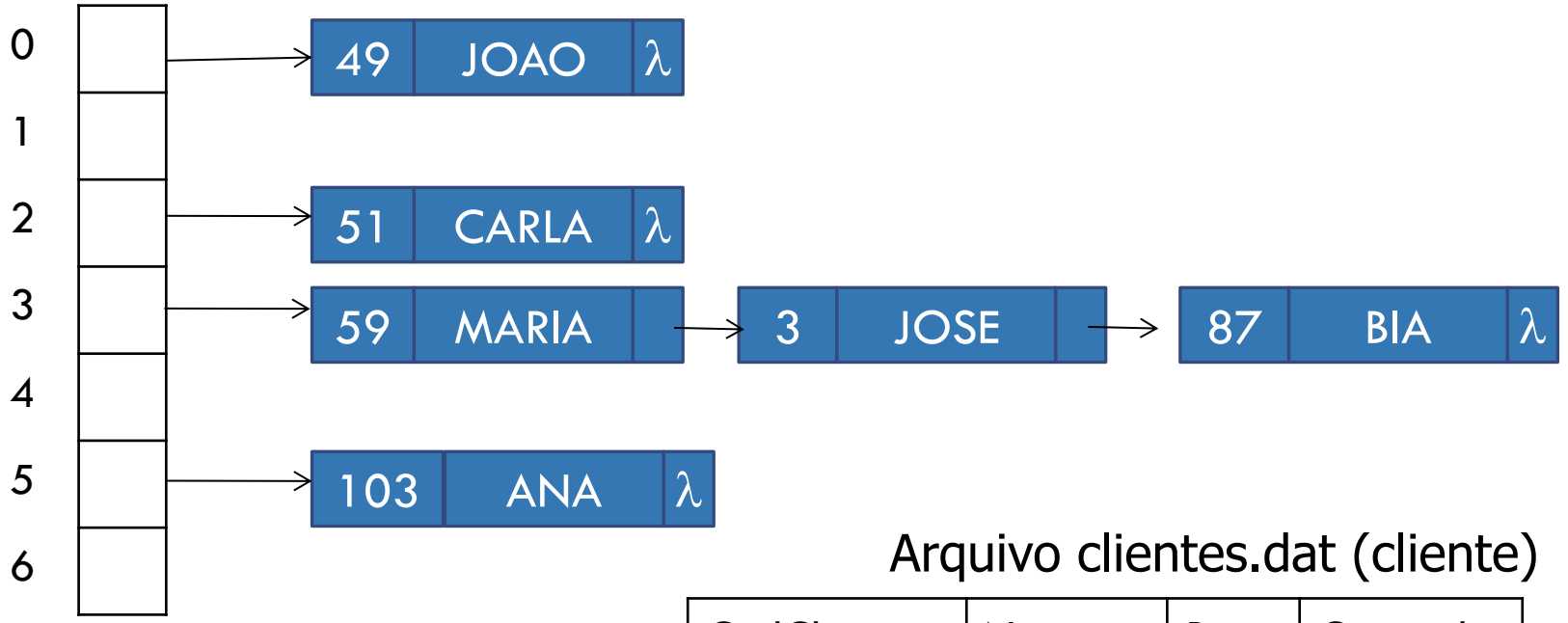
O flag **ocupado** pode ter os seguintes valores:

- TRUE: quando o compartimento tem um registro
- FALSE: quando o registro que estava no compartimento foi excluído

EXEMPLO



ESTRUTURA DOS ARQUIVOS



Arquivo tabHash.dat
(compartimento_hash)

0	0
1	-1
2	4
3	1
4	-1
5	2
6	-1

m = 7

Arquivo clientes.dat (cliente)

	CodCliente	Nome	Prox	Ocupado
0	49	JOAO	-1	TRUE
1	59	MARIA	3	TRUE
2	103	ANA	-1	TRUE
3	3	JOSE	5	TRUE
4	51	CARLA	-1	TRUE
5	87	BIA	-1	TRUE
6				
7				
...				

REFLEXÃO:

Como seriam os procedimentos para inclusão e exclusão?

IMPLEMENTAÇÃO DE EXCLUSÃO

- Ao excluir um registro, marca-se o flag de ocupado como FALSE (ou seja, marca-se que o compartimento está liberado para nova inserção)

IMPLEMENTAÇÃO DE INSERÇÃO (OPÇÃO 1)

Para inserir novo registro

- Inserir o registro no final da lista encadeada, se ele já não estiver na lista

- De tempos em tempos, re-arrumar o arquivo para ocupar as posições onde o flag de ocupado é FALSE

IMPLEMENTAÇÃO DE INSERÇÃO (OPÇÃO 2)

Para inserir novo registro

- Ao passar pelos registros procurando pela chave, guardar o endereço **p** do primeiro nó marcado como LIBERADO (flag ocupado = FALSE)
- Se ao chegar ao final da lista encadeada, a chave não for encontrada, gravar o registro na posição **p**, ou no final da lista, caso não tenha sido encontrado compartimento livre

EXERCÍCIO

Desenhe a tabela hash (em disco) resultante das seguintes operações (cumulativas) usando o algoritmo de inserção em **Tabela Hash com Encadeamento Exterior.**

Considere que a tabela tem **tamanho 7** e a função de hash usa o **método da divisão.**

Inserir as chaves 10, 3, 5, 7, 12, 6, 14, 4, 8

REFERÊNCIA

Szwarcfiter, J.; Markezon, L. Estruturas de Dados e seus Algoritmos, 3a. ed. LTC. Cap. 10