

Estrutura de **Dados Básicas I.**

Lista Duplamente Encadeada

Prof. Eiji Adachi M. Barbosa

Lista encadeada (Simples)

- Como visto em sala de aula:
 - Inserir no fim – Complexidade?
 - Inserir no início – Complexidade?
 - Remover no fim – Complexidade?
 - Remover no início – Complexidade?

Lista encadeada (Simples)

- Como visto em sala de aula:
 - **Inserir no fim – $O(n)$**
 - Inserir no início – $O(1)$
 - **Remover no fim – $O(n)$**
 - Remover no início – $O(1)$

Como eu poderia melhorar as operações no fim de uma Lista?

Numa remoção, é necessário saber quem é o elemento anterior na lista.

```
Estrutura Lista{  
    No início;  
    No fim;  
    tamanho;  
}
```

```
Estrutura No{  
    conteúdo;  
    No próximo;  
    No anterior;  
}
```

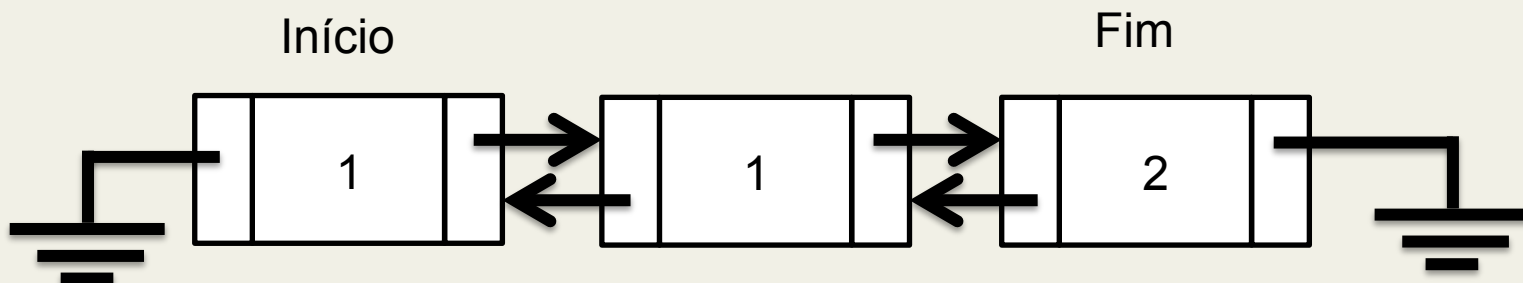
LISTA DUPLAMENTE ENCADEADA

Lista duplamente encadeada

Estrutura No

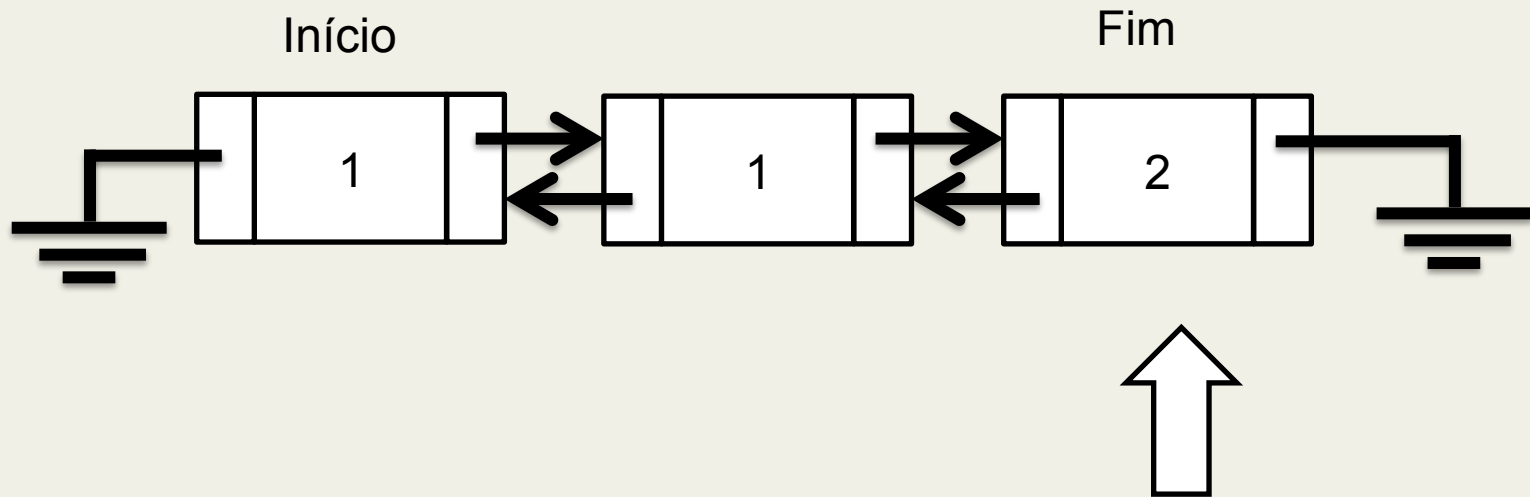


Estrutura Lista



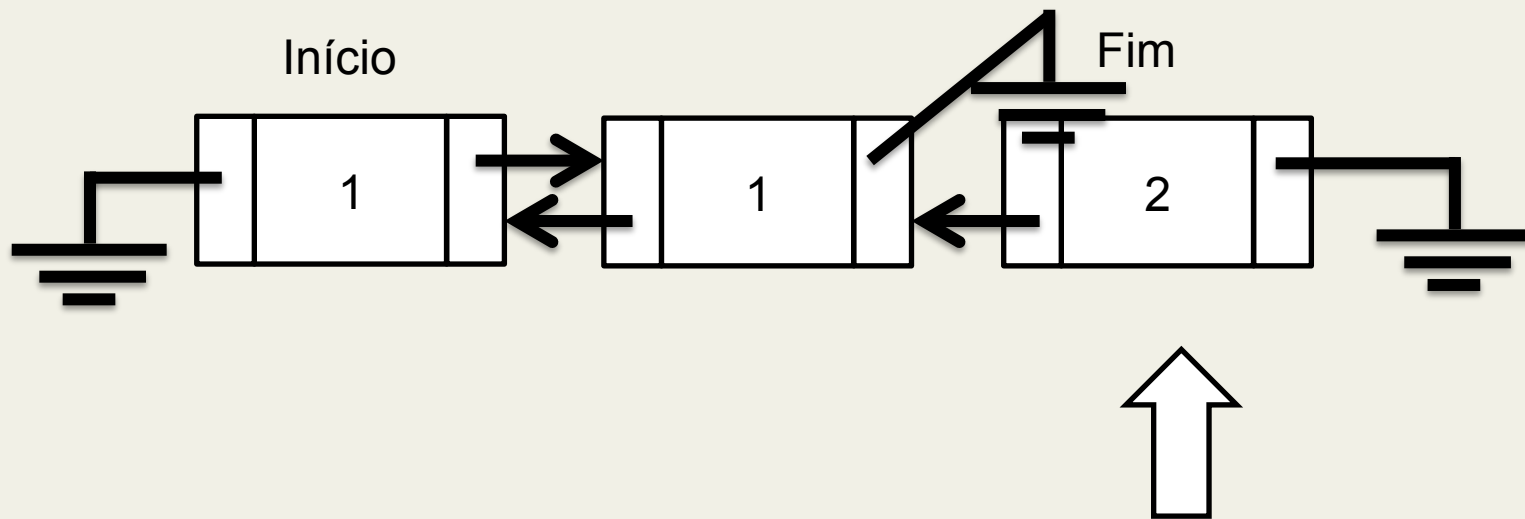
Lista duplamente encadeada – Remoção

- Remover no fim
 - RemoverFim()



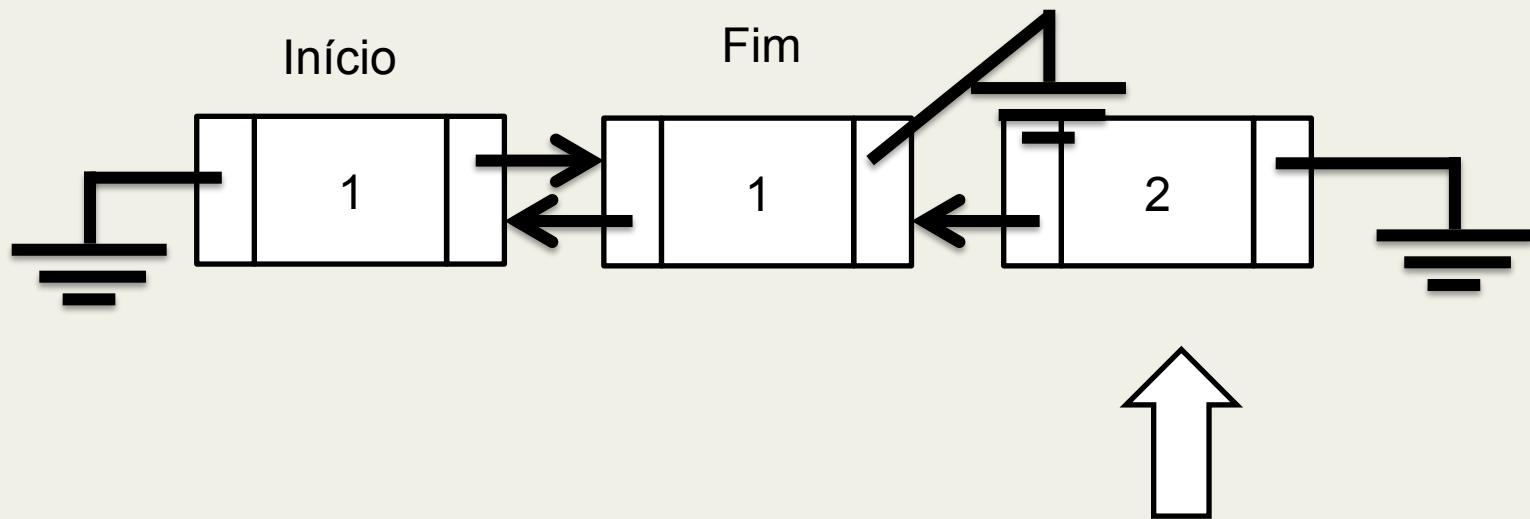
Lista duplamente encadeada – Remoção

- Remover no fim
 - RemoverFim()



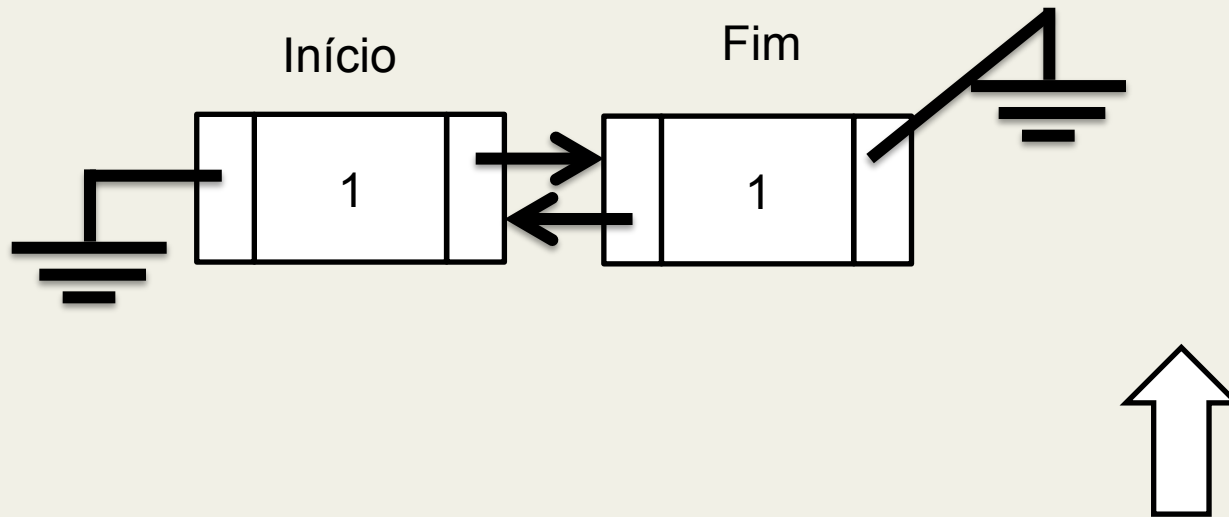
Lista duplamente encadeada – Remoção

- Remover no fim
 - RemoverFim()



Lista duplamente encadeada – Remoção

- Remover no fim
 - RemoverFim()



Lista encadeada

Simplex

- Inserir no fim – $O(n)$
- Inserir no início – $O(1)$
- **Remover no fim – $O(n)$**
- Remover no início – $O(1)$

Dupla

- Inserir no fim – $O(1)$
- Inserir no início – $O(1)$
- **Remover no fim – $O(1)$**
- Remover no início – $O(1)$

Há ganho de eficiência na remoção no fim.
As outras operações ficam um pouco mais complexas,
pois é necessário atualizar o ponteiro 'anterior'.

LISTA ENCADEADA COM SENTINELAS

Lista encadeada com sentinelas

- Sentinelas
 - São nós que não armazenam informações
 - Servem para delimitar extremos da lista, evitando usar NULO
 - Nunca devem ser modificados!
 - Atribuição aos nós sentinelas ocorrem única e exclusivamente no momento da criação da lista

Lista encadeada com sentinelas

Lista sem sentinelas

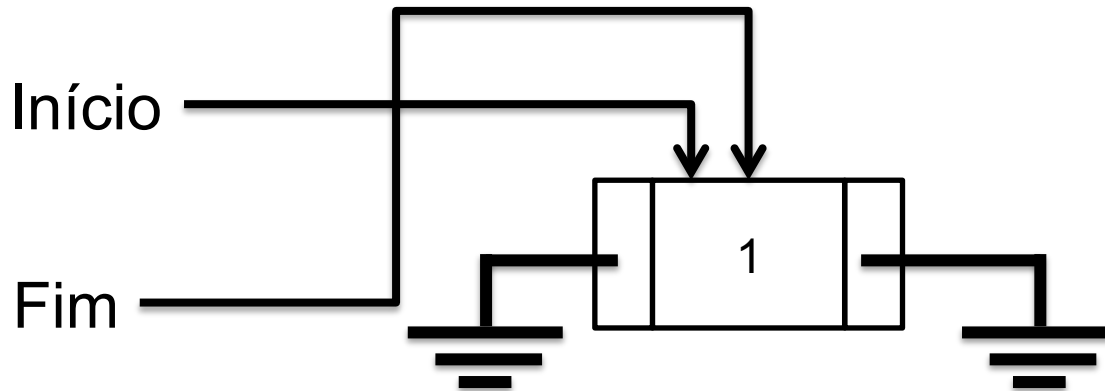
Início 

Fim 

Tamanho = 0

Lista encadeada com sentinelas

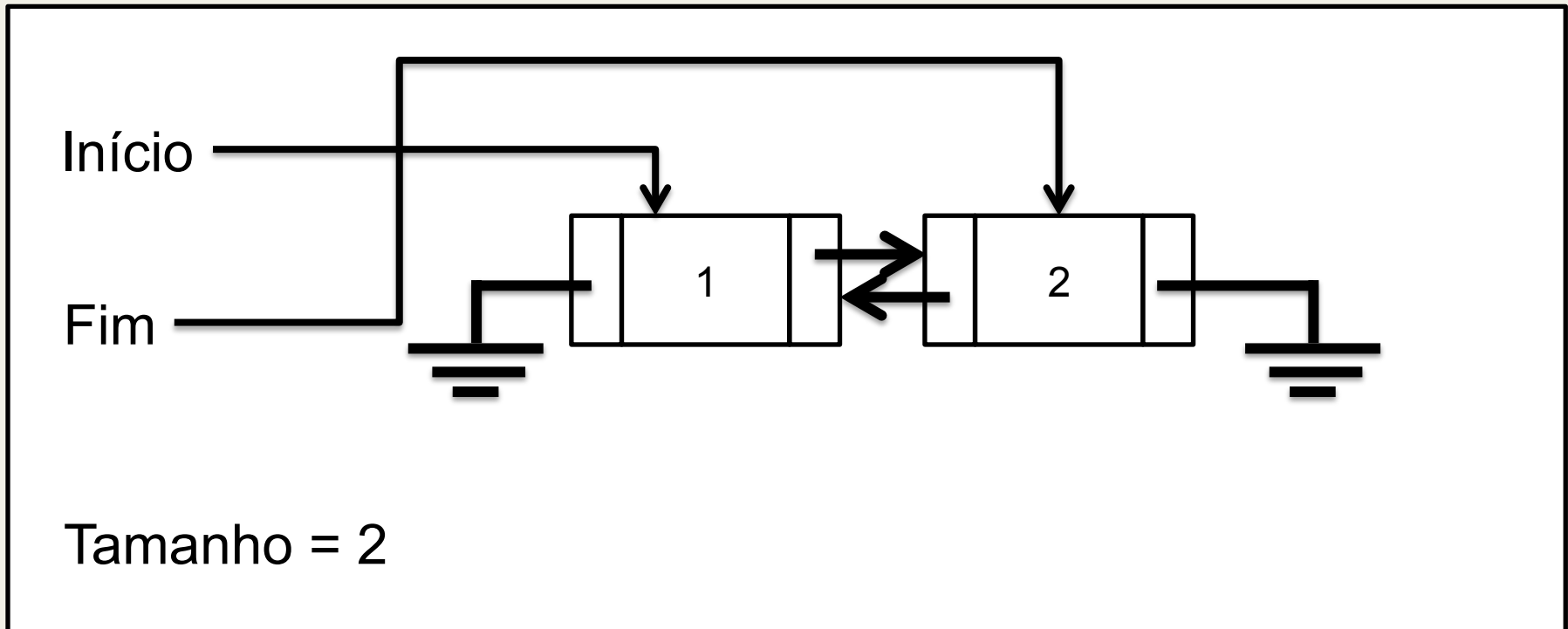
Lista sem sentinelas



Tamanho = 1

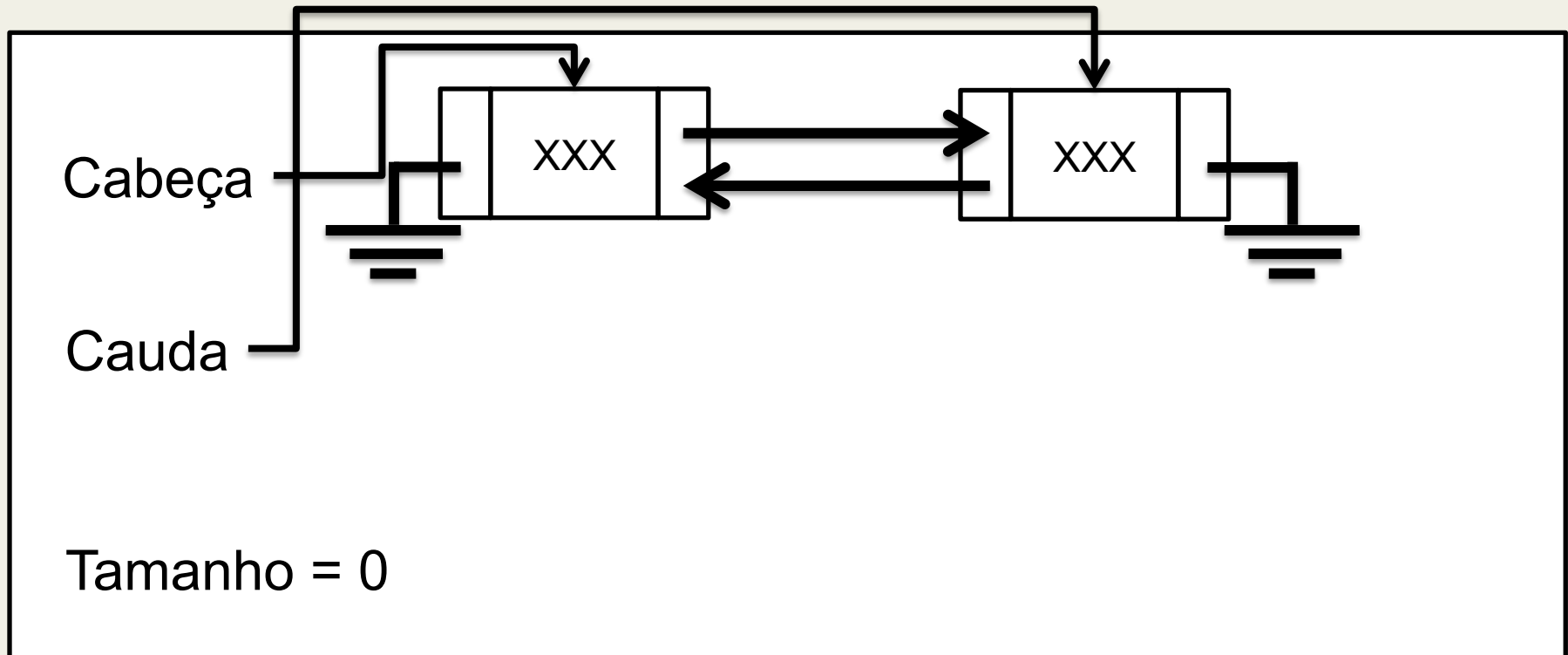
Lista encadeada com sentinelas

Lista sem sentinelas



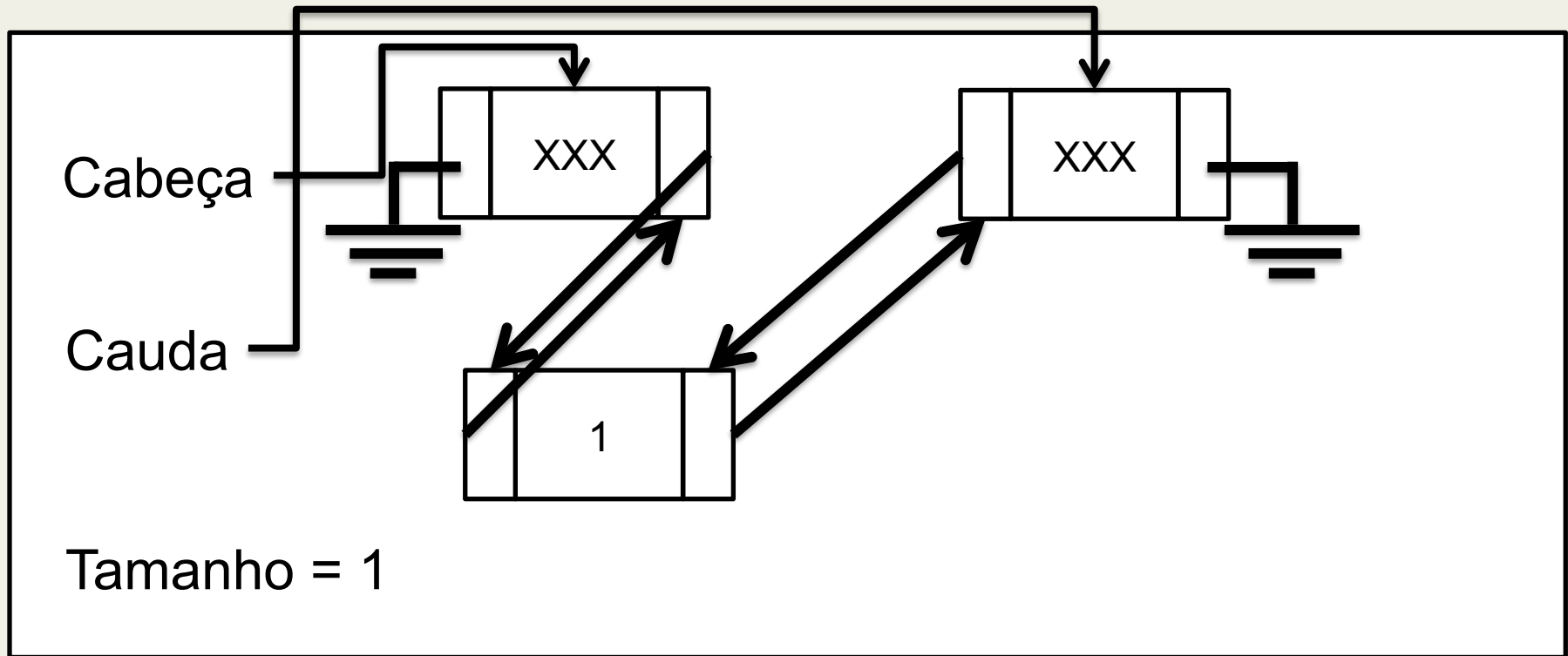
Lista encadeada com sentinelas

Lista com sentinelas



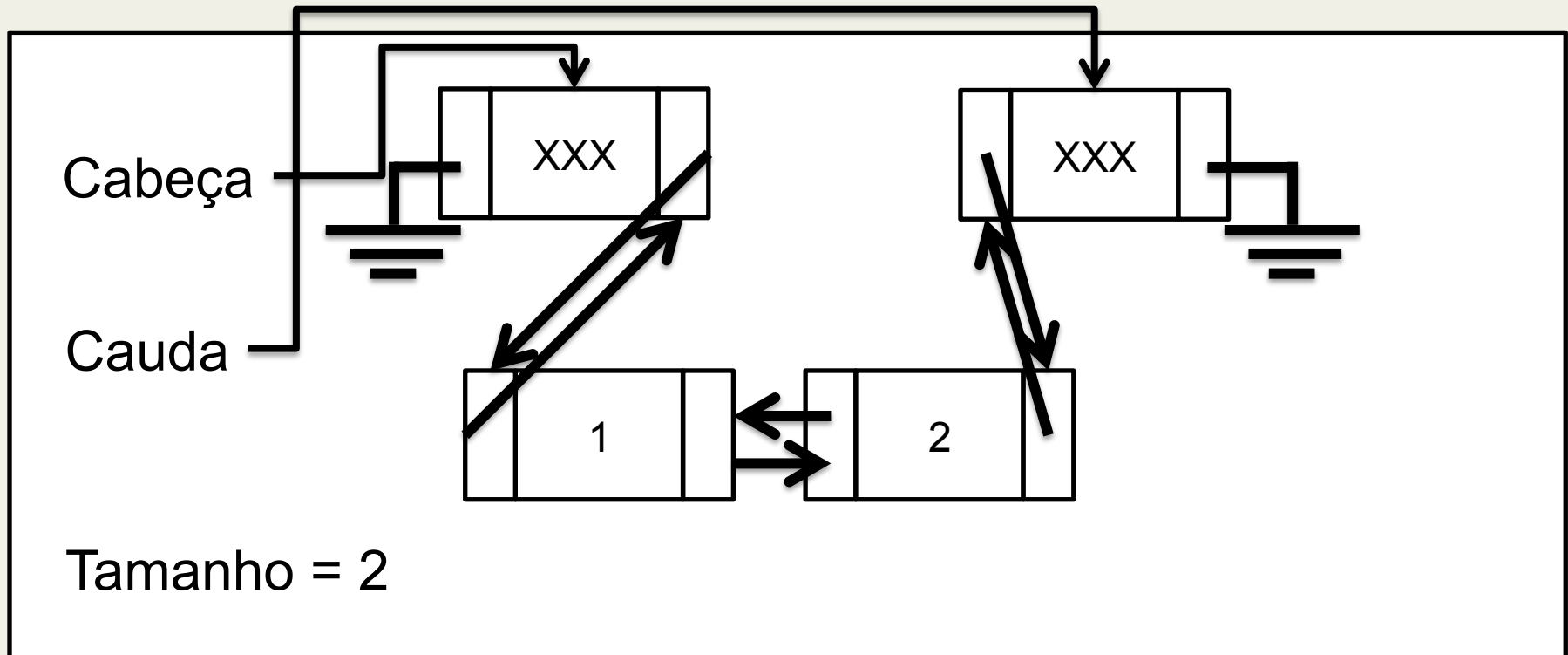
Lista encadeada com sentinelas

Lista com sentinelas



Lista encadeada com sentinelas

Lista com sentinelas



Lista encadeada com sentinelas

```
Estrutura Lista{  
    No cabeça;  
    No cauda;  
    tamanho;  
}
```

```
Estrutura No{  
    conteúdo;  
    No próximo;  
    No anterior;  
}
```

Lista encadeada com sentinelas

- Criar lista

```
CriarLista( ):  
    Lista lista = Alocar Memória Para Lista  
    lista.cabeça = CriarNo( valor-qualquer )  
    lista.cauda  = CriarNo( valor-qualquer )  
  
    lista.cabeça.próximo = lista.cauda  
    lista.cauda.anterior = lista.cabeça  
  
    lista.cabeça.anterior = NULO  
    lista.cauda.próximo  = NULO  
  
    lista.tamanho = 0  
FIM
```

Lista encadeada com sentinelas

- Percorrer lista

Evita usar NULO

```
PercorrerLista( lista ):  
  No atual = lista.cabeça.próximo  
  ENQUANTO atual != lista.cauda FAÇA  
    // Faça alguma operação com nó da lista  
    atual = atual.próximo  
  FIM_ENQUANTO  
FIM
```

Lista encadeada com sentinelas

- Inserir no início

```
InserirInício( lista, valor ):  
  No nó = CriarNo( valor )  
  No cabeça = lista.cabeça  
  
  nó.próximo = cabeça.próximo  
  nó.anterior = cabeça  
  
  nó.próximo.anterior = nó  
  nó.anterior.próximo = nó  
  
  lista.tamanho = lista.tamanho+1  
FIM
```

Não precisa checar
se é NULO

Lista encadeada com sentinelas

- Inserir no Fim

```
InserirFim( lista, valor ):  
    No nó = CriarNo( valor )  
    No cauda = lista.cauda  
  
    nó.próximo = cauda  
    nó.anterior = cauda.anterior  
  
    nó.próximo.anterior = nó  
    nó.anterior.próximo = nó  
  
    lista.tamanho = lista.tamanho+1  
FIM
```

Não precisa checar
se é NULO

Lista encadeada com sentinelas

- Remover no início

```
RemoverInício( lista ):  
    SE lista está vazia ENTÃO  
        RETORNE ERRO-ListaVazia  
    FIM_SE
```

```
No aRemover = lista.cabeça.próximo  
valor = aRemover.conteúdo
```

```
nó.anterior.próximo = nó.próximo  
nó.próximo.anterior = nó.anterior  
delete nó
```

```
    lista.tamanho = lista.tamanho-1  
FIM
```

Lista encadeada com sentinelas

- Remover no fim

```
RemoverFim( lista ):  
    SE lista está vazia ENTÃO  
        RETORNE ERRO-ListaVazia  
    FIM_SE
```

```
No aRemover = lista.cauda.anterior  
valor = aRemover.conteúdo
```

```
nó.anterior.próximo = nó.próximo  
nó.próximo.anterior = nó.anterior  
delete nó
```

```
    lista.tamanho = lista.tamanho-1  
FIM
```

Estrutura de Dados Básicas I.

Lista Duplamente Encadeada

Prof. Eiji Adachi M. Barbosa