

Algoritmos e Estrutura de Dados



Aula 06 – Estrutura de Dados: Listas
(Parte III)

Prof. Tiago A. E. Ferreira

Como já Visto...

□ Listas Estáticas

- Listas estáticas possuem tamanho fixo determinado na criação da lista e não podem ser expandidas durante a execução do programa.
- Solução: Usar listas dinâmicas

Listas Dinâmicas

- Comumente chamadas de **Listas Ligadas** ou **Listas Encadeadas**
- Estrutura de tamanho variável que utiliza apenas a quantidade de memória que precisa.
- São representadas como seqüências de dados definidas pelo encadeamento dos elementos.
- **Cada elemento** é chamado de **nó** da lista e contém os dados e um ponteiro (ou link) para o próximo nó da lista

Listas Lineares

□ **Vantagens**

- A memória é alocada e liberada quando necessário.
- A alocação dinâmica nos oferece a necessária flexibilidade para mantermos, sem um grande número de movimentações de nós na lista, a estrutura devidamente ordenada a cada inserção e/ou retirada de elemento

□ **Desvantagem**

- Maior grau de complexidade de implementação

Listas Dinâmicas

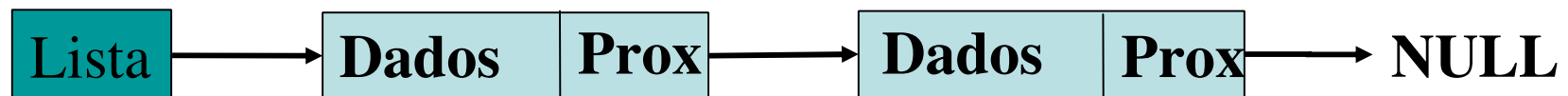
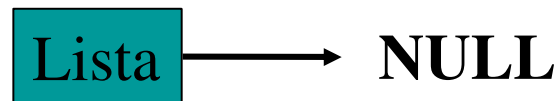
□ Aplicações

- São adequadas para aplicações onde não é possível prever a demanda por memória, permitindo a manipulação de quantidades imprevisíveis de dados, de qualquer formato

Listas Dinâmicas

- **Início da Lista**

- O início da lista é estabelecido por um ponteiro para o 1º nó da lista.
- Caso a lista esteja vazia, inicialmente o ponteiro aponta para **NULL** ou **None**





Operações Básicas

- ❑ Criar a Lista
- ❑ Inserir elemento
- ❑ Remover elemento
- ❑ Consultar elemento
- ❑ Alterar elemento
- ❑ Listar os elementos

Operações Básicas

- Inserir elemento
- Remover elemento
- Consultar elemento
- Alterar elemento
- Listar os elementos

Lista Encadeada - Criação

❑ Criando a Lista

- Inicialmente, declara-se dois ponteiros
 - ❑ Um para o início e outro para o fim da lista.
- O ponteiro para o fim da lista permite realizar inserções sem que seja necessário percorrer toda a lista
- **Inicialmente**, como a lista está vazia, ambos apontam para NULL



Lista Dinâmica - Inserção

□ Inserindo Elementos:

- Verificar se a lista ainda tem posição disponível
 - Caso afirmativo, o elemento é colocado na primeira posição vazia (e.g. no final da lista), que é indicada pela variável `qtdElementos`.
 - Posteriormente, incrementar a quantidade de elementos da lista

Lista Dinâmica - Inserção

□ Inserindo Elementos

- **Caso 1: Lista Vazia**

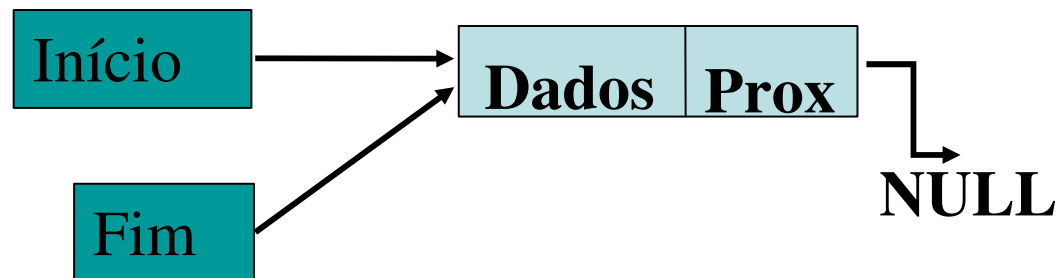
- Cria-se o novo nó e os dois ponteiros apontam para o novo nó inserido na lista, que por sua vez aponta para NULL.

- **Caso 2: Lista Não Vazia** – Inserção no final da lista

- Cria-se o novo nó que aponta para NULL; o último nó da lista aponta para o novo nó; e, ponteiro de fim aponta para o novo nó

Lista Dinâmica - Inserção

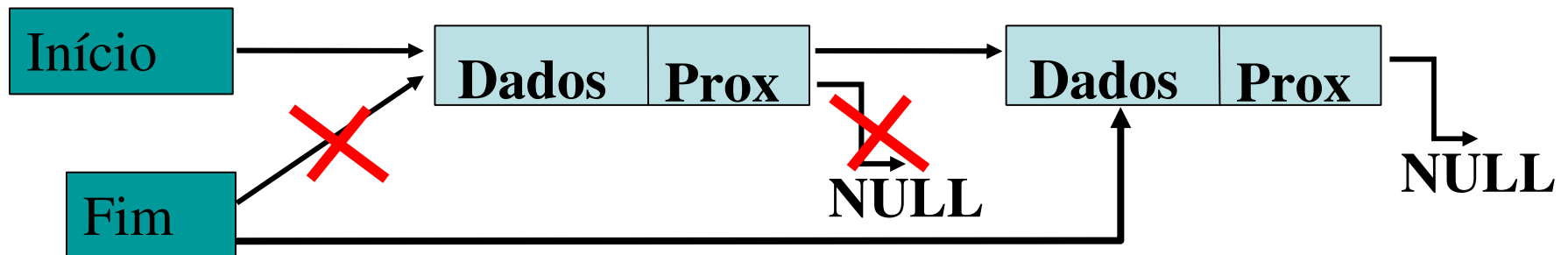
- **Inserindo Elementos**
 - **Caso 1: Lista Vazia:**



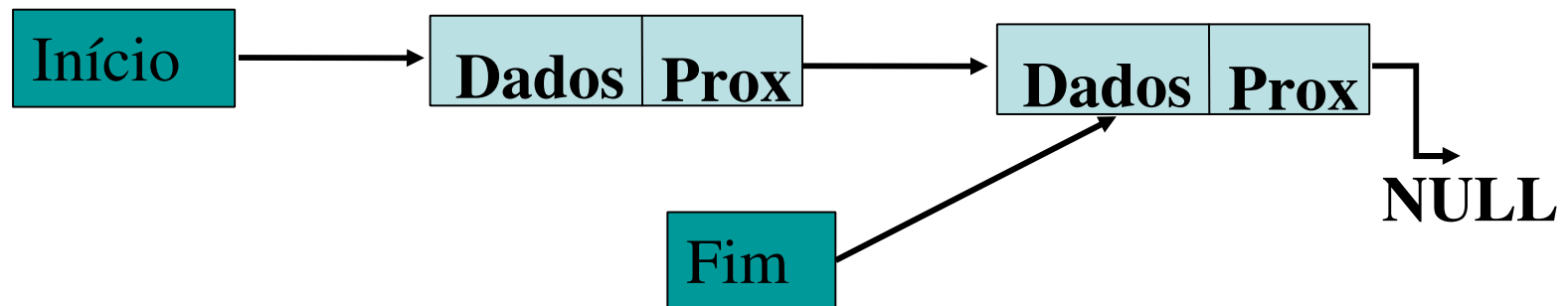
Lista Dinâmica - Inserção

□ Inserindo Elementos

- Caso 2: Lista Não Vazia – Inserção no final da Lista



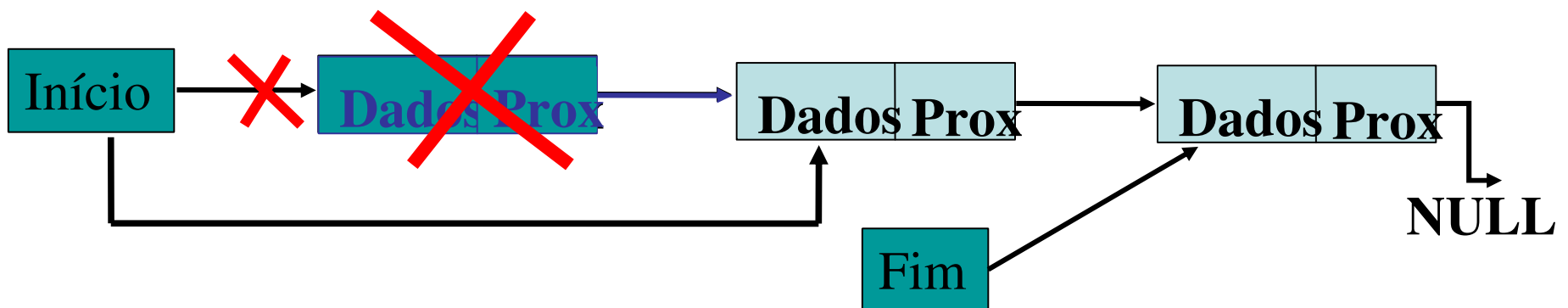
- Resultado Final:



Lista Encadeada - Remoção

Removendo Elementos

- **Caso 1:** Remover primeiro elemento da lista
 - O elemento a ser removido é marcado.
 - O ponteiro do início aponta para o próximo elemento.
 - A memória é liberada

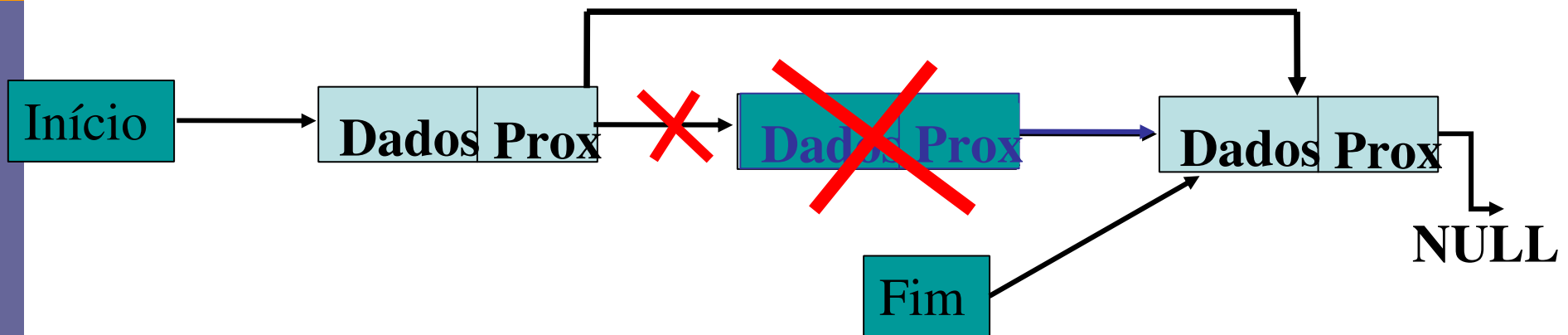


Lista Encadeada - Remoção

- Removendo Elementos

- Caso 2: Remover elemento do meio da lista

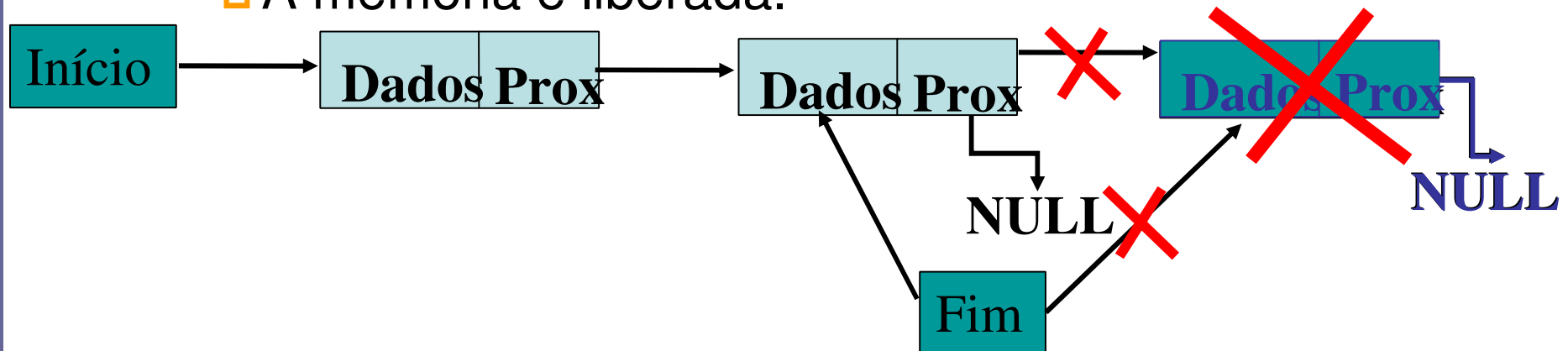
- ▣ O elemento a ser removido é marcado.
- ▣ O elemento anterior ao removido aponta para onde o removido apontava.
- ▣ A memória é liberada.



Lista Encadeada - Remoção

Removendo Elementos

- Caso 3: Remover elemento do final da lista
 - O elemento a ser removido é marcado.
 - O elemento anterior ao removido aponta para NULL.
 - O ponteiro para fim aponta para o anterior.
 - A memória é liberada.



Lista Encadeada Circular

□ Listas Encadeadas

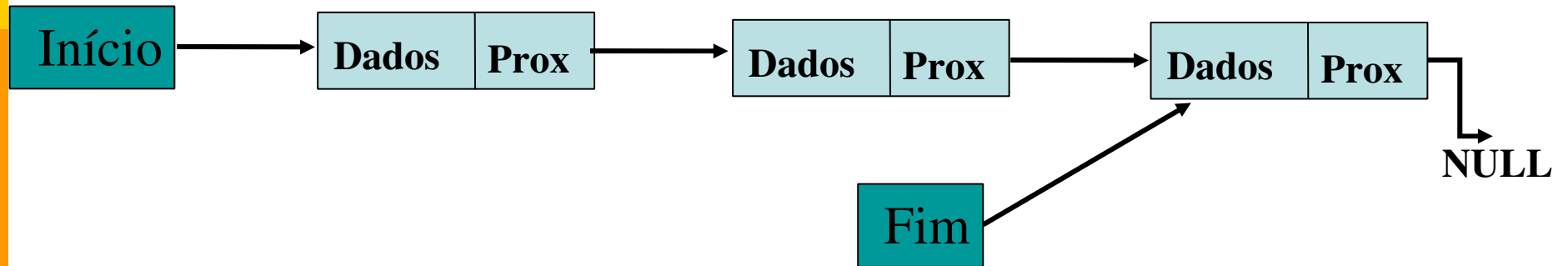
- Cada Nodo (ou Nó) tem dois campos:
 - Dado
 - Prox
- Para o último Nodo, o campo prox aponta para NULL

□ Listas Encadeadas Circular

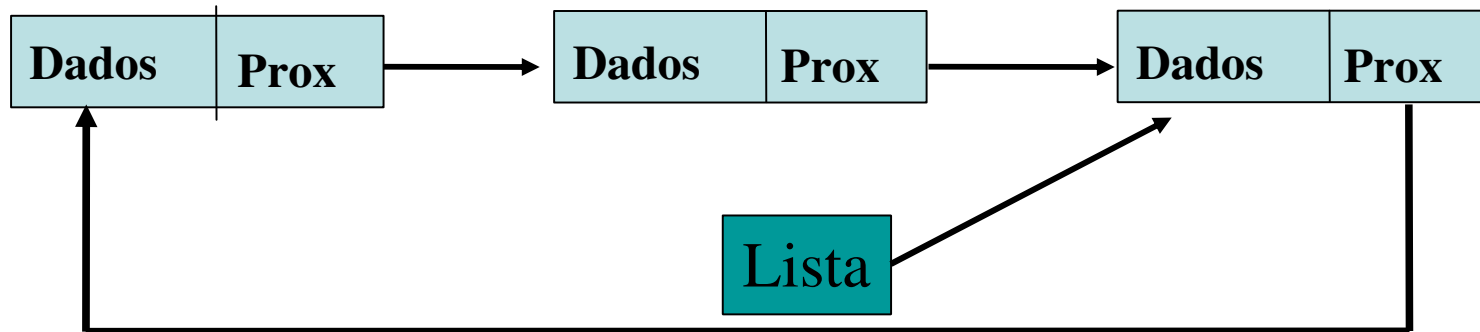
- Para estas listas, o campo prox do último Nodo aponta para o primeiro Nodo da lista, formando um “circulo” de encadeamento

Lista Encadeada Circular

Lista simplesmente encadeada



Lista com encadeamento circular



Encadeamento Circular

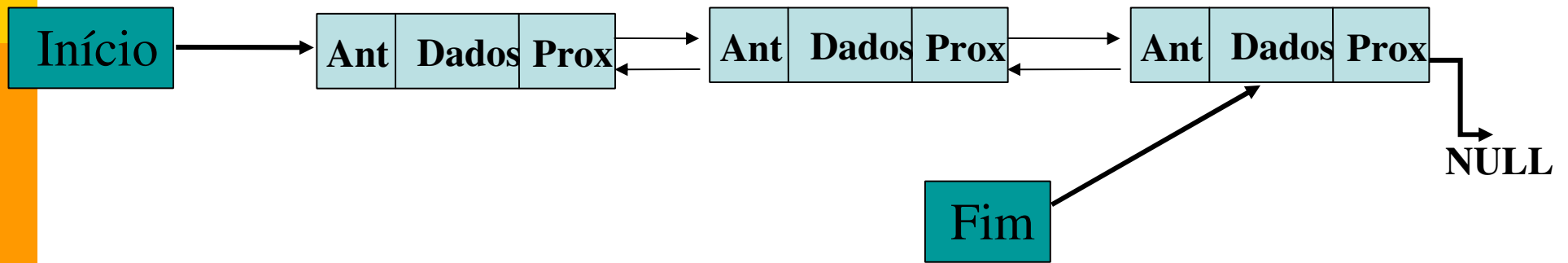
- A grande vantagem nesta lista é que só é necessário o uso de um ponteiro para o final da lista
 - Se a estrutura não for ordenada, o elemento deverá ser inserido sempre no primeiro lugar.
 - Ao remover um elemento, deve-se estar atento para o caso de a fila ter somente um único elemento, pois a estrutura torna-se vazia

Encadeamento duplo

- Para uma lista com encadeamento duplo, cada Nodo tem três campos:
 - Dados
 - Prox. (Referência ao próximo Nodo)
 - Ant. (Referência ao Nodo Anterior)

Encadeamentos Duplos

Lista duplamente encadeada



Lista com encadeamento duplo circular

