

# LISTAS

PUC MINAS

ALGORITMOS E ESTRUTURAS DE DADOS II

# LISTA LINEAR

- Estrutura de dados básica para **interligar itens** de um conjunto.
- Armazena uma **sequência** de zero ou mais itens:
  - $1^{\circ}, 2^{\circ}, 3^{\circ}, \dots, n$ -ésimo;
    - cada item da lista linear é de um determinado tipo;
    - **n** representa o **tamanho da lista linear**.

# LISTA LINEAR

- Principal **propriedade estrutural** envolve as **posições relativas dos itens** em uma dimensão.
- Assumindo  $n \geq 1$ :
  - item que ocupa a **primeira posição** da lista:
    - primeiro item da lista;
  - item que ocupa a **última posição** da lista:
    - último item da lista;
  - item que ocupa a  **$i$ -ésima posição** da lista:
    - $i$ -ésimo item da lista linear;
  - $i$ -ésimo item **precede**  $(i+1)$ -ésimo item, para  $i = 1, 2, 3, \dots, n-1$
  - $i$ -ésimo item **suced**e  $(i-1)$ -ésimo item, para  $i = 2, 3, \dots, n$

# LISTA LINEAR – EXEMPLOS

- Lista de alunos matriculados em uma turma;
- Gerência de memória em um sistema operacional;
- Índice de palavras de um documento.

# LISTA LINEAR – PRINCIPAIS OPERAÇÕES

- **Criar uma lista linear vazia;**
- **Inserir um novo item na lista;**
- **Retirar da lista o item que ocupa a  $i$ -ésima posição ou que apresenta o valor  $v$ ;**

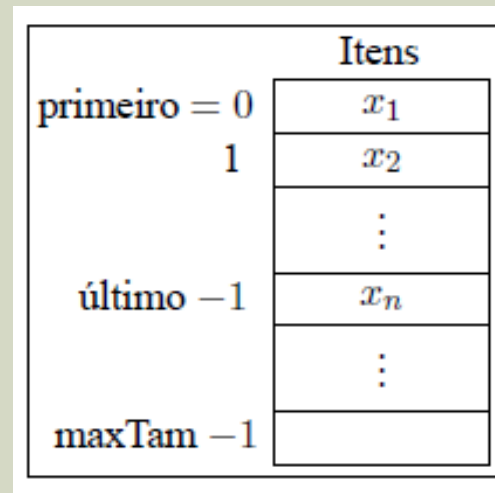
# LISTA LINEAR – PRINCIPAIS OPERAÇÕES

- Localizar o  $i$ -ésimo item da lista ou o item de valor  $v$ ;
- Verificar se a lista está vazia;
- Imprimir o conteúdo de toda a lista.

# LISTAS LINEARES E VETORES

- Itens da lista são armazenados em **posições contínuas de memória:**

- cria-se um **vetor com M posições vazias;**
- **variáveis de controle:**
  - primeiro;
  - último.



- Lista pode ser **percorrida em qualquer direção.**

# LISTAS LINEARES E VETORES – OPERAÇÕES

- **Inserir:**
  - no “**final**” da lista;
    - relativamente simples:
      - incrementar “**último**”.
  - em **outras posições** da lista;
    - deslocar itens da lista;
      - a partir da posição em que o novo item será inserido.
    - incrementar “**último**”.



# LISTAS LINEARES E VETORES – OPERAÇÕES

- **Retirar:**
  - **percorrer a lista do início até a posição do item que será retirado;**
  - **retirar o item da lista;**
  - **deslocar os itens da lista que ocupam posições posteriores à do item retirado;**
    - **para posições anteriores.**
  - **decrementar “último”.**

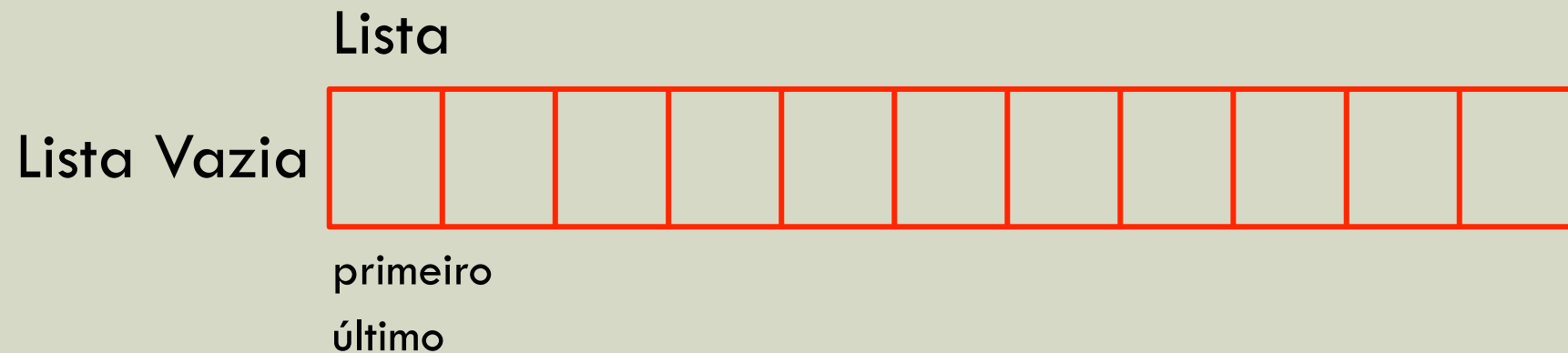
# LISTAS LINEARES E VETORES – OPERAÇÕES

- **Localizar:**
  - **percorrer a lista do início até a posição do item desejado;**
  - **ou até o final da lista;**
    - quando o item desejado não é localizado na lista.

# LISTAS LINEARES E VETORES – OPERAÇÕES

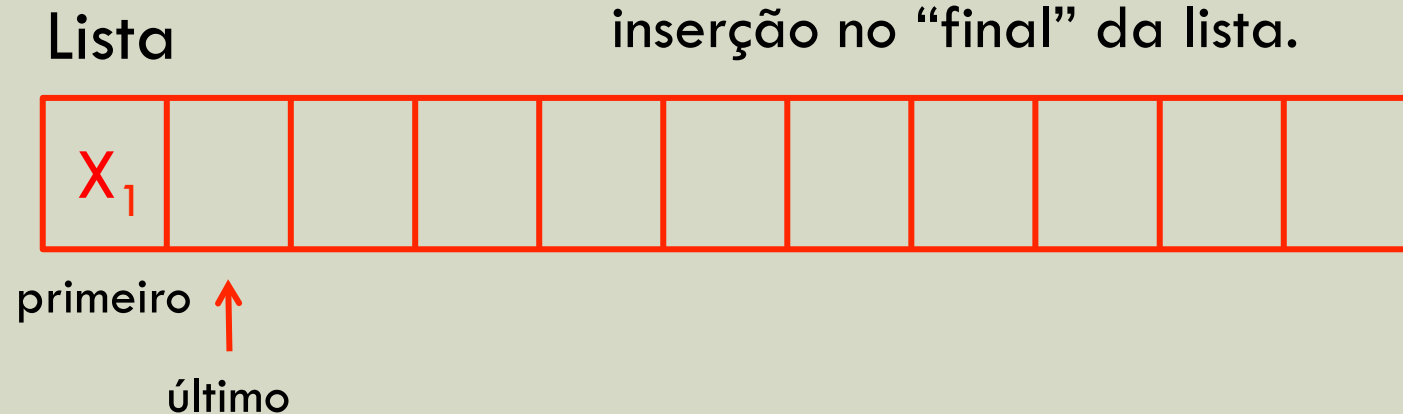
- **Verificar se a lista está vazia:**
  - “primeiro” == “último”.
- **Imprimir todo o conteúdo da lista:**
  - **laço entre “primeiro” e “último”.**

# LISTAS LINEARES COM VETORES



# LISTAS LINEARES COM VETORES

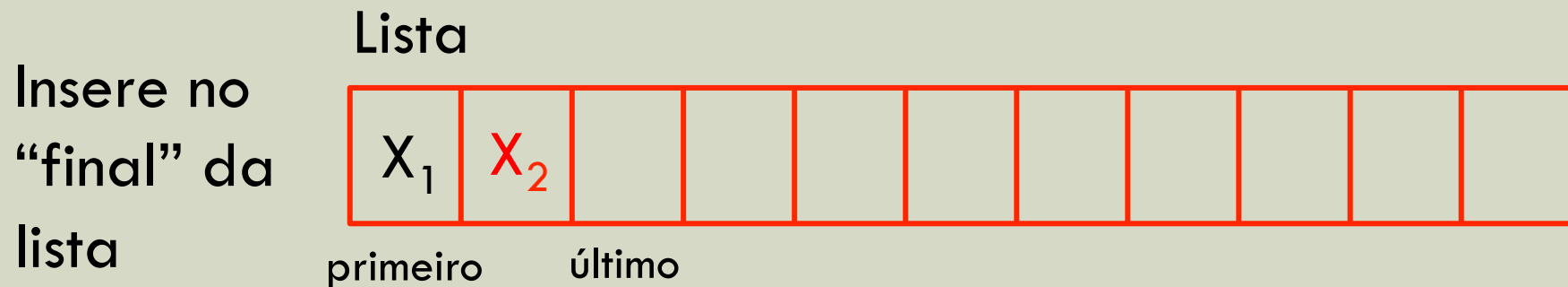
Inserir no  
“final” da  
lista



Inserção padrão na lista linear:  
inserção no “final” da lista.

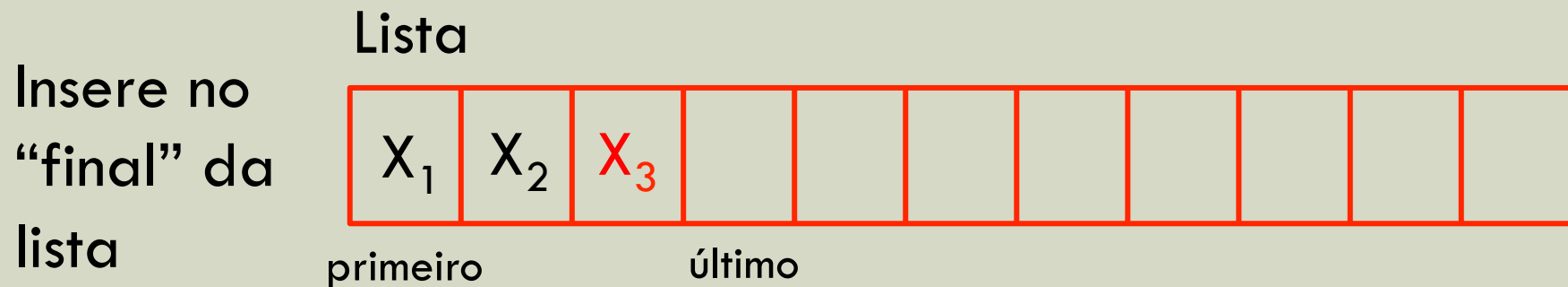
Inserção no “final” da lista linear é igual à operação “enfileirar” da fila.

# LISTAS LINEARES COM VETORES



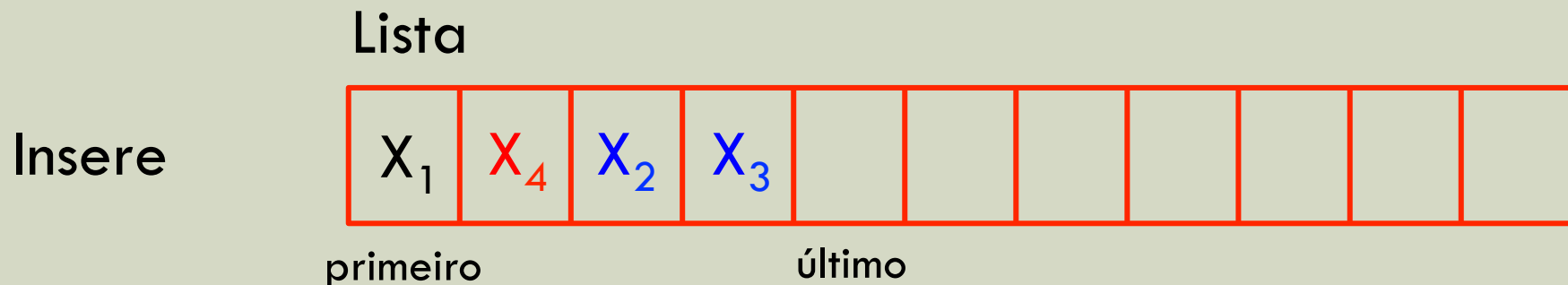
Inserção no “final” da lista linear é igual à operação “enfileirar” da fila.

# LISTAS LINEARES COM VETORES



Inserção no “final” da lista linear é igual à operação “enfileirar” da fila.

# LISTAS LINEARES COM VETORES



Inserção em outras posições da lista linear requer o **deslocamento de itens** da lista, **a partir da posição em que o novo item foi inserido.**



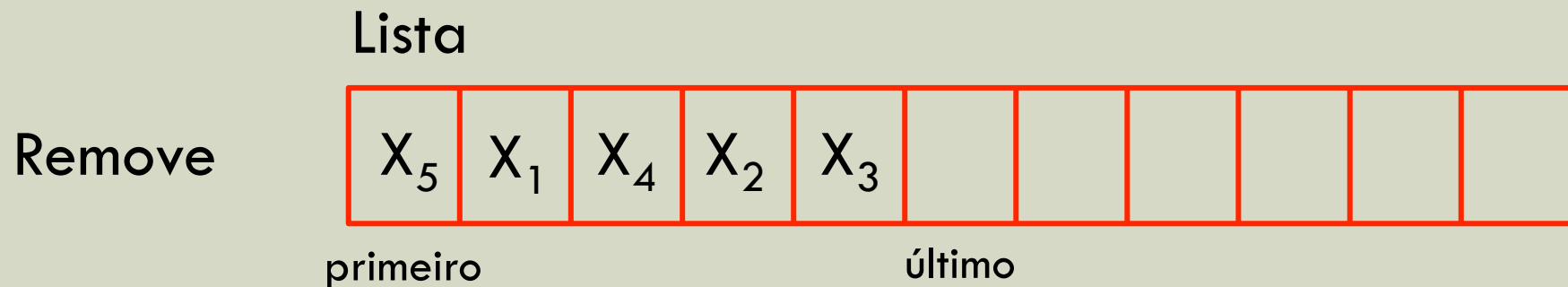
Downloaded from <http://ajphaphysocpharm.sagepub.com/> at 11:01 11 November 2014

$x_5$	$x_1$	$x_4$	$x_2$	$x_3$						
-------	-------	-------	-------	-------	--	--	--	--	--	--

primeiro

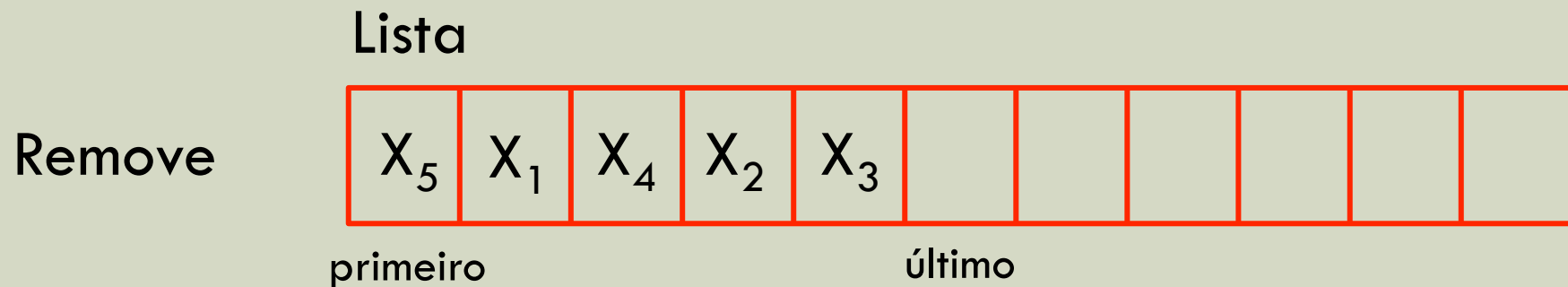
Inserção em outras posições da lista linear requer o **deslocamento de itens** da lista, a partir da posição em que o novo item foi inserido.

# LISTAS LINEARES COM VETORES



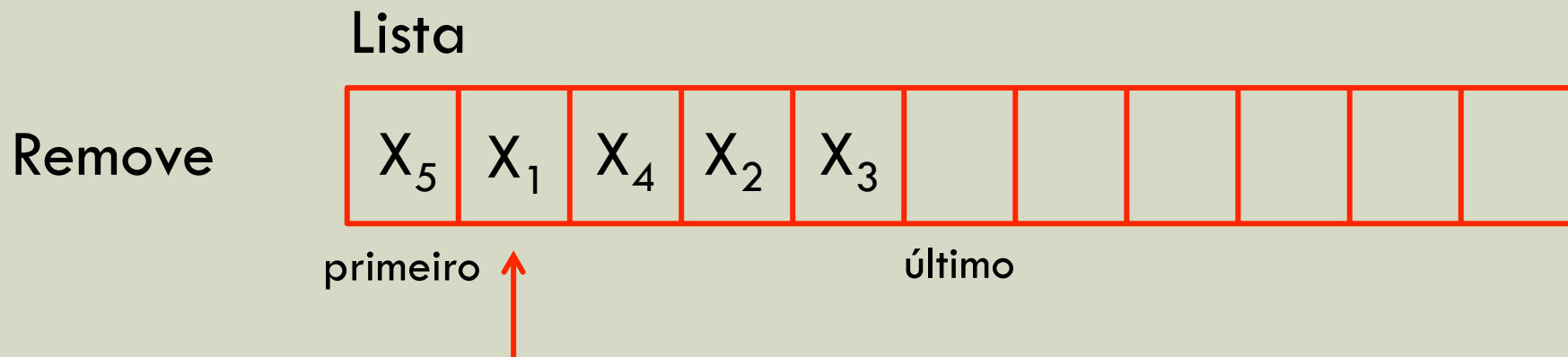
Remover: item " $X_4$ "

# LISTAS LINEARES COM VETORES



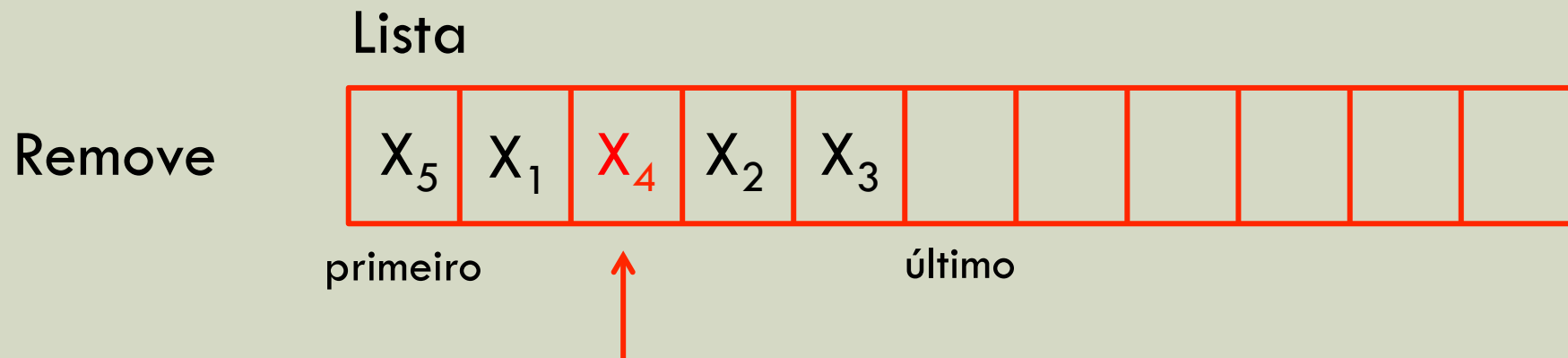
**Percorre a lista do início até encontrar o item “ $X_4$ ”.**

# LISTAS LINEARES COM VETORES



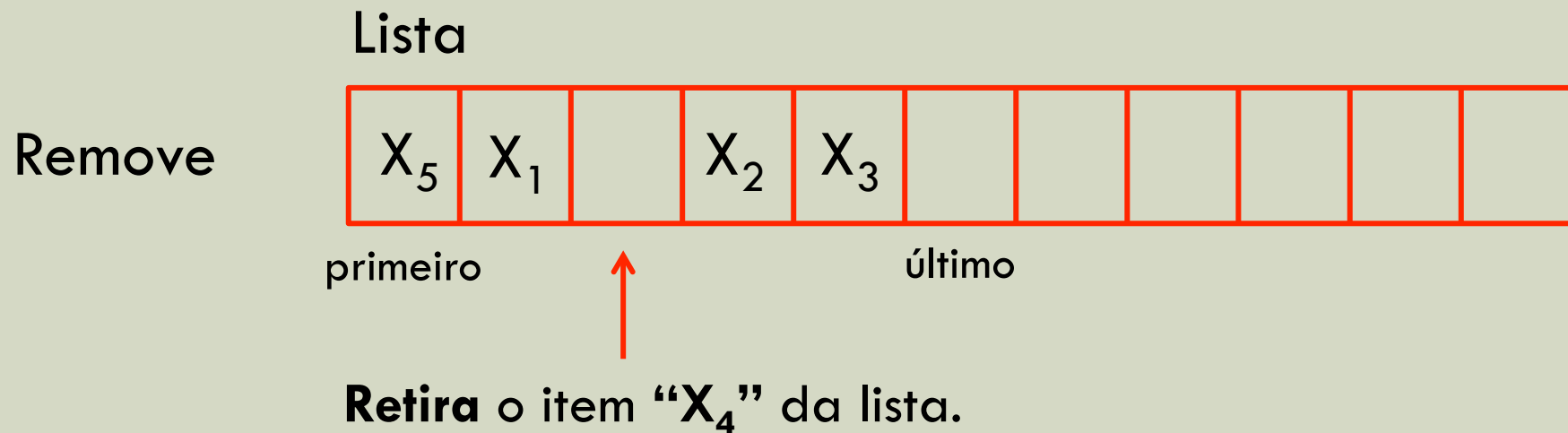
**Percorre a lista do início até encontrar o item “ $X_4$ ”.**

# LISTAS LINEARES COM VETORES

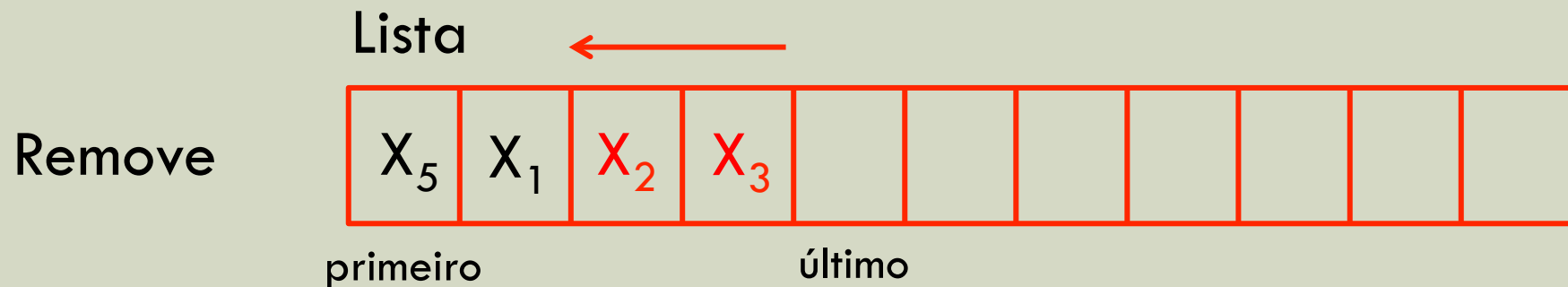


**Percorre a lista do início até encontrar o item “ $X_4$ ”.**

# LISTAS LINEARES COM VETORES



# LISTAS LINEARES COM VETORES



**Desloca os itens da lista que ocupam posições posteriores à do item “ $X_4$ ” para posições anteriores e decrementa “último”.**

# LISTAS LINEARES COM VETORES

- **Vantagem:**

- **economia de memória;**
  - apontadores implícitos.

- **Desvantagens:**

- **custo para inserir ou retirar itens da lista.**
  - Pior caso: deslocamento de todos os demais itens da lista.
- **aplicações em que não existe previsão em relação ao crescimento da lista;**
  - exige-se **realocação de memória.**

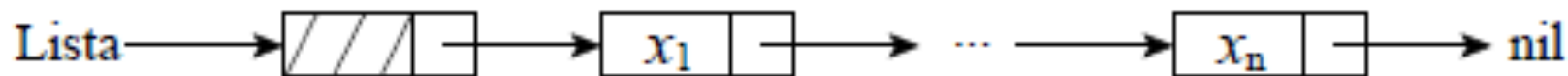


# LISTAS LINEARES IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

- Também chamadas de **listas encadeadas**.
- Estruturas **flexíveis**:
  - podem **crescer ou diminuir de tamanho** durante a **execução** de um **programa**;
  - têm **tamanho “infinito”**;
  - permitem utilizar **posições não contínuas de memória**;
    - **cada célula da lista** contém **informação** necessária para **alcançar-se a próxima célula**;
    - **pode-se inserir uma nova célula sempre que necessário**;
      - **e apontar para ela**.

# LISTAS LINEARES IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

- São adequadas em situações nas quais é difícil prever a demanda por memória para um conjunto de itens:
  - **manipulação de quantidades imprevisíveis de dados.**



# LISTAS LINEARES IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

- **Classe Lista:**
  - **Apontadores de controle:**
    - primeiro e último.
  - **Célula sentinela, nó-cabeça ou nodo-cabeça:**
    - primeiro elemento da lista;
    - serve apenas para controle;
      - para facilitar a implementação das operações inserir e retirar quando a lista está vazia.

# LISTAS LINEARES IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

- Lista implementada por meio de células.
- Cada célula contém:
  - um item da lista;
  - uma referência para a próxima célula.
- Classe Lista contém:
  - referência para a primeira célula da lista;
    - célula sentinela;
  - referência para a última célula da lista.

# LISTAS ENCADEADAS – OPERAÇÕES

- **Criar uma lista vazia:**
  - **instanciar célula sentinela;**
  - **inicializar os ponteiros de controle:**
    - **“primeiro” e “último” referenciam a mesma célula:**
      - **célula sentinela.**

# LISTAS ENCADEADAS – OPERAÇÕES

- **Inserir:**
  - no “**final**” da lista;
    - **criar uma célula nova, ligá-la à célula referenciada por “último” e colocar nela o novo item;**
    - **atualizar “último”.**

# LISTAS ENCADEADAS – OPERAÇÕES

## ■ Inserir:

- em **outras posições** da lista;
  - **localizar posição** em que o novo item será inserido;
  - **criar uma célula nova e colocar nela o novo item;**
  - **“ligá-la” às células entre as quais deve ser posicionada, atualizando referências.**

# LISTAS ENCADEADAS – OPERAÇÕES

- **Retirar:**
  - **localizar item que será retirado;**
  - **atualizar referências, “ligando” as células entre as quais localiza-se o item retirado;**
    - retirando o item da lista;
  - **se o item retirado for o último da lista;**
    - **atualizar “último”.**



# LISTAS ENCADEADAS – OPERAÇÕES

- **Localizar:**

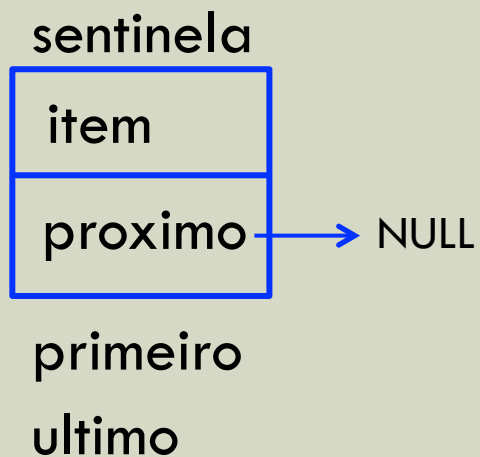
- **percorrer a lista do início até o item desejado;**
- **ou até o final da lista;**
  - quando o item desejado não é localizado na lista.

# LISTAS ENCADEADAS – OPERAÇÕES

- **Verificar se a lista está vazia:**
  - “primeiro” == “último”.
- **Imprimir todo o conteúdo da lista:**
  - **laço do início até o final da lista;**
    - imprimindo dados relevantes.

# LISTAS ENCADEADAS – OPERAÇÕES

Lista vazia: “primeiro” e “último” referenciam a célula sentinela



# LISTAS ENCADEADAS – OPERAÇÕES

sentinela

item

proximo → NULL

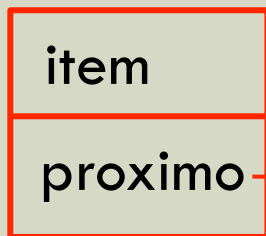
primeiro

ultimo

# LISTAS ENCADEADAS – OPERAÇÕES

Inserir no “final” da lista

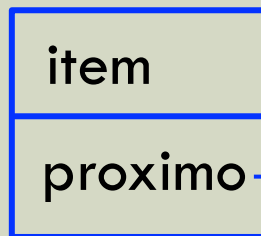
sentinela



primeiro

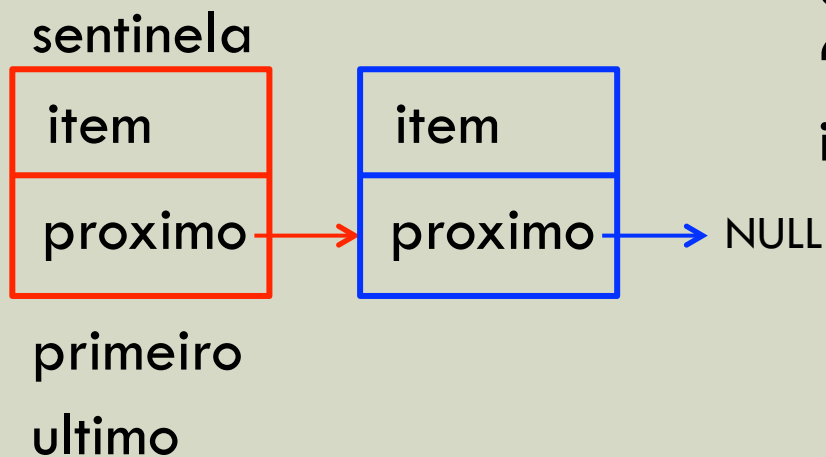
ultimo

**Criar uma célula nova**, ligá-la à célula referenciada por “último” e colocar nela o novo item



# LISTAS ENCADEADAS – OPERAÇÕES

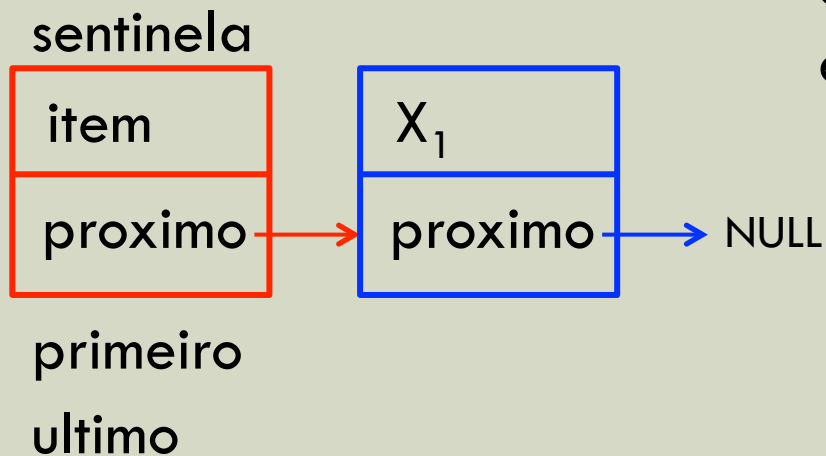
Inserir no “final” da lista



Criar uma célula nova, **ligá-la à célula referenciada por “último”** e colocar nela o novo item

# LISTAS ENCADEADAS – OPERAÇÕES

Inserir no “final” da lista

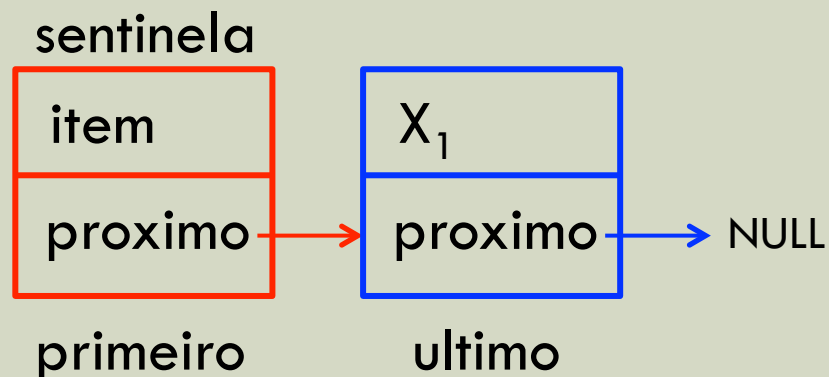


Criar uma célula nova, ligá-la à célula referenciada por “último” e **colocar nela o novo item**

# LISTAS ENCADEADAS – OPERAÇÕES

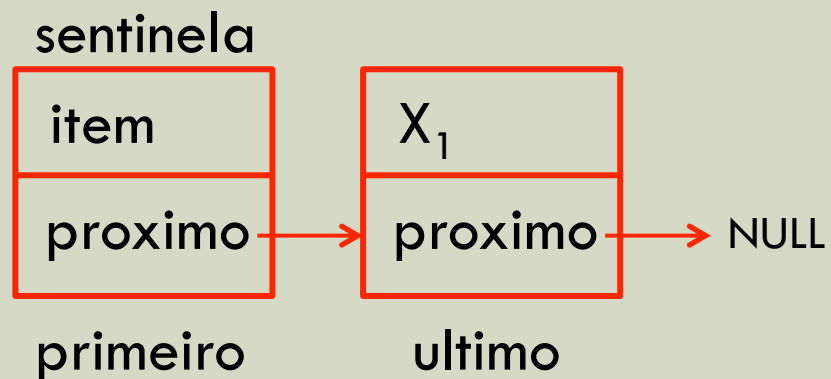
Inserir no “final” da lista

Atualizar “último”



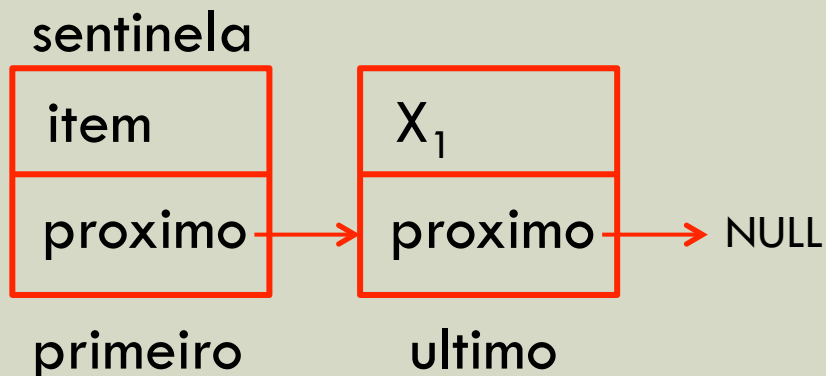


# LISTAS ENCADEADAS – OPERAÇÕES

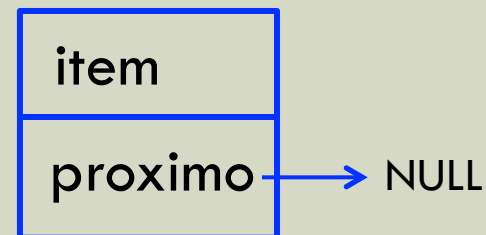


# LISTAS ENCADEADAS – OPERAÇÕES

Inserir no “final” da lista

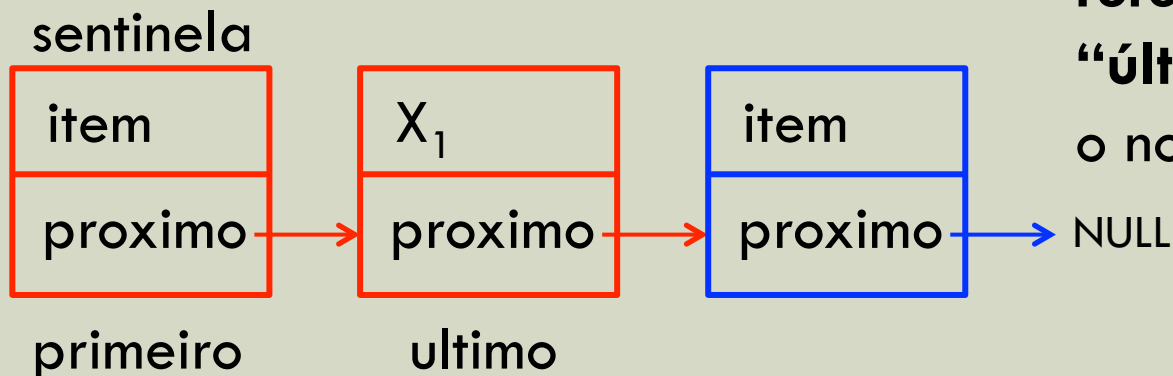


**Criar uma célula nova**, ligá-la à célula referenciada por “último” e colocar nela o novo item



# LISTAS ENCADEADAS – OPERAÇÕES

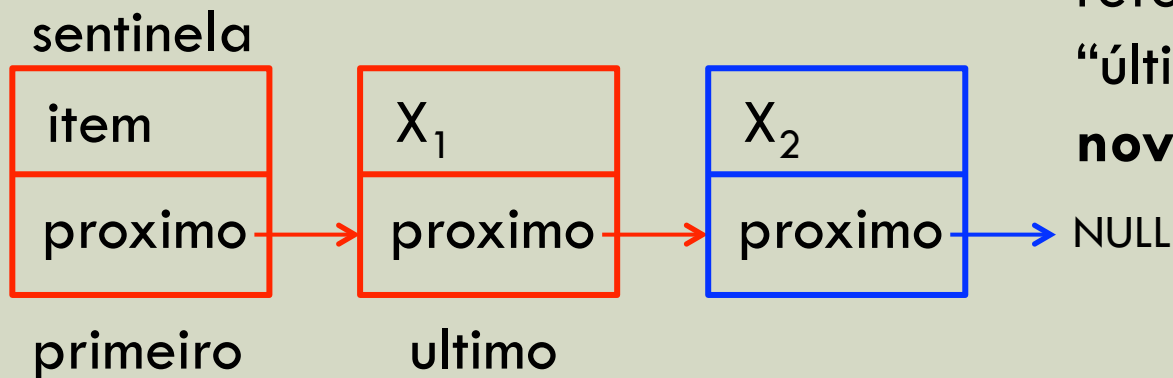
Inserir no “final” da lista



Criar uma célula nova,  
**ligá-la à célula**  
**referenciada por**  
**“último”** e colocar nela  
o novo item

# LISTAS ENCADEADAS – OPERAÇÕES

Inserir no “final” da lista

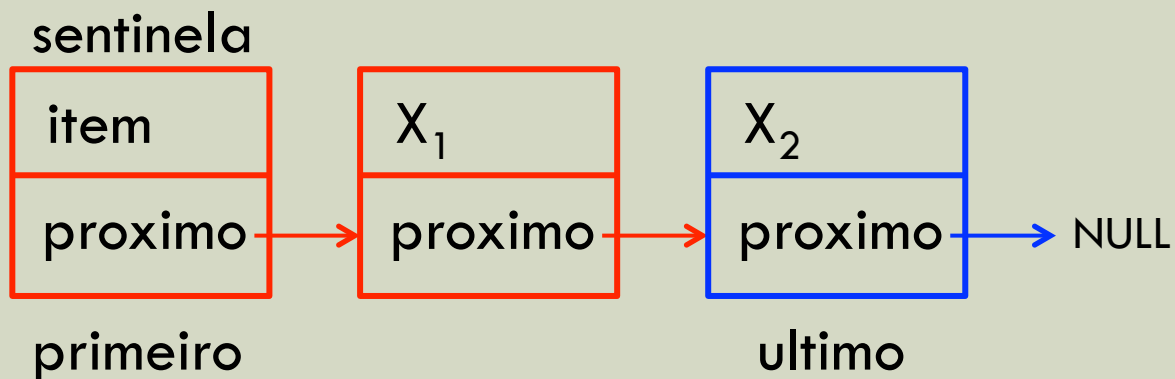


Criar uma célula nova,  
ligá-la à célula  
referenciada por  
“último” e **colocar nela o  
novo item**

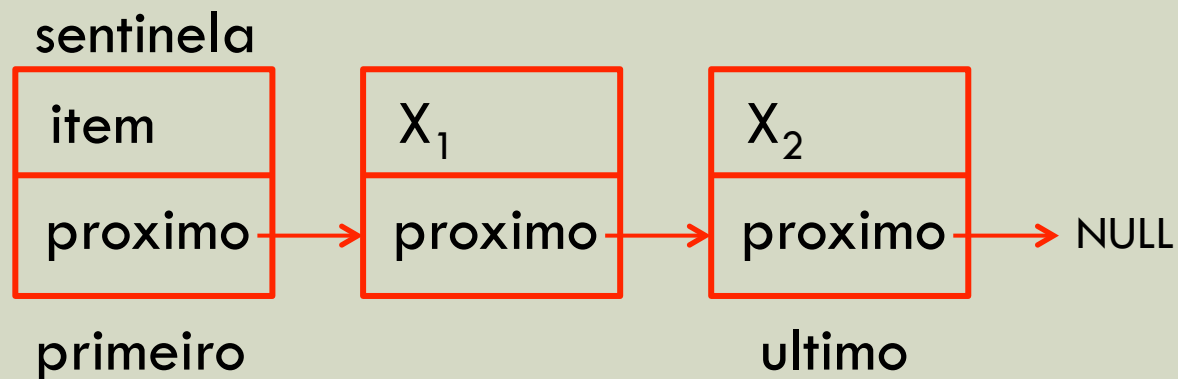
# LISTAS ENCADEADAS – OPERAÇÕES

Inserir no “final” da lista

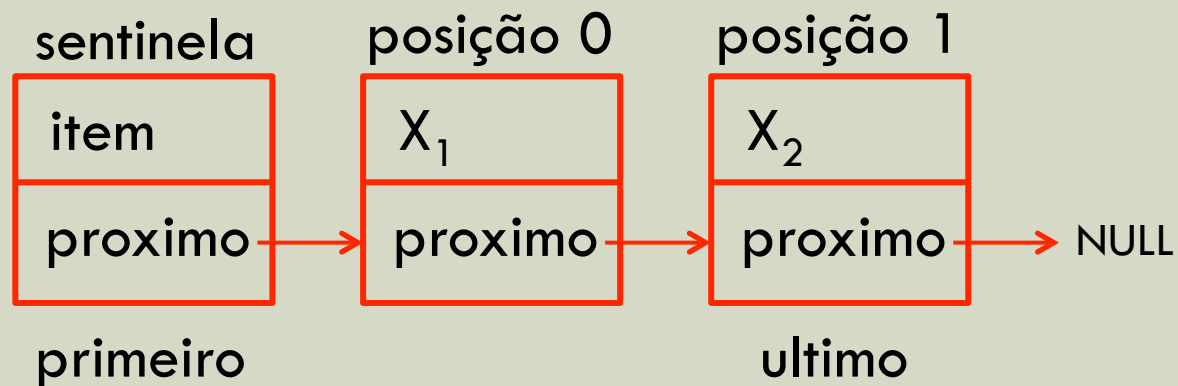
Atualizar “último”



# LISTAS ENCADEADAS – OPERAÇÕES



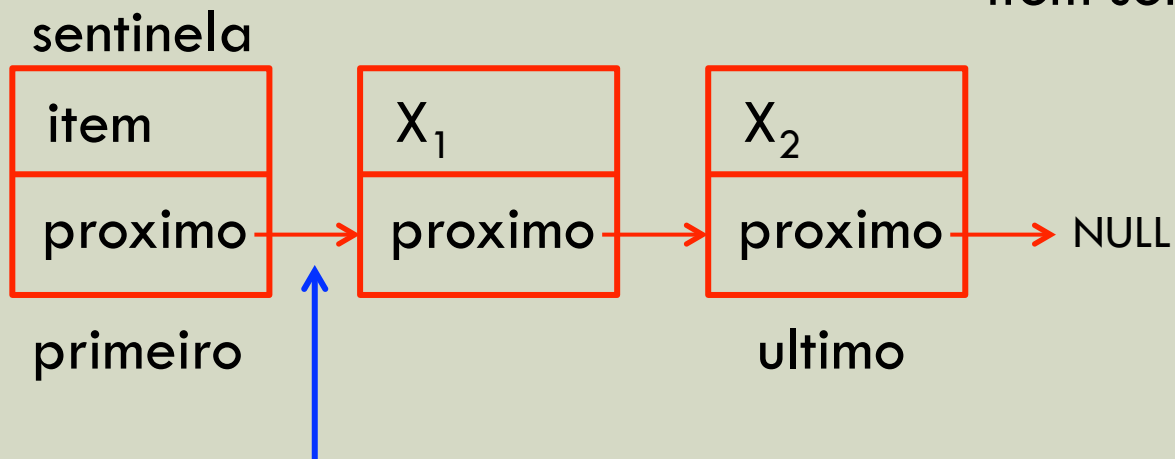
# LISTAS ENCADEADAS – OPERAÇÕES



# LISTAS ENCADEADAS – OPERAÇÕES

Inserir na posição 1 da lista

Localizar posição em que o item será inserido

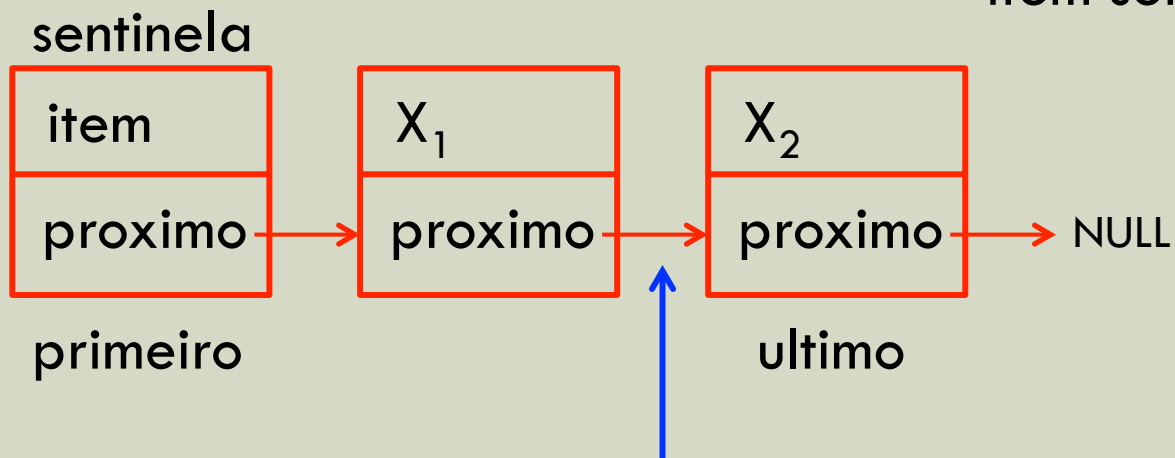




# LISTAS ENCADEADAS – OPERAÇÕES

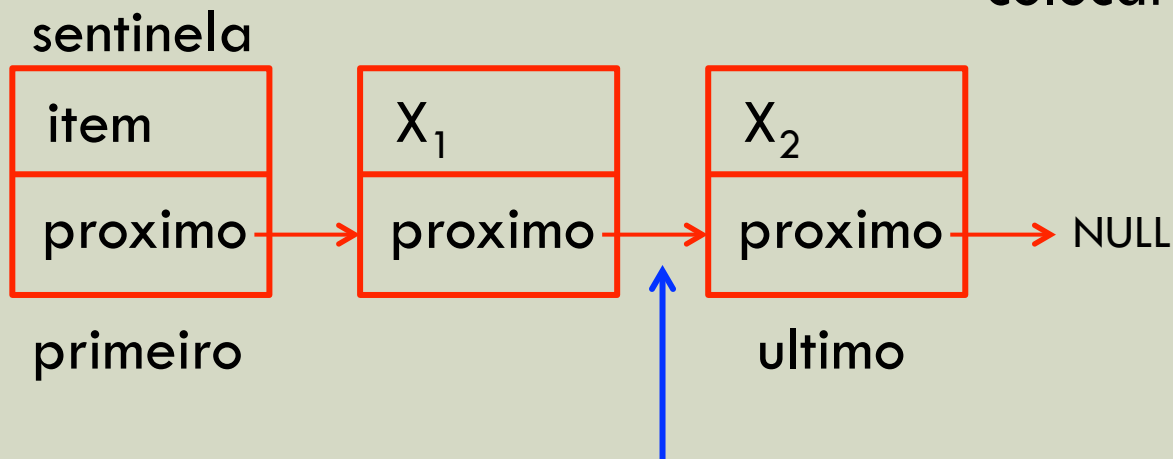
Inserir na posição 1 da lista

Localizar posição em que o item será inserido

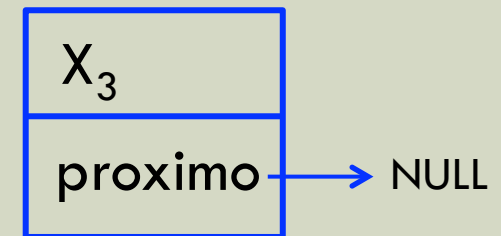


# LISTAS ENCADEADAS – OPERAÇÕES

Inserir na posição 1 da lista



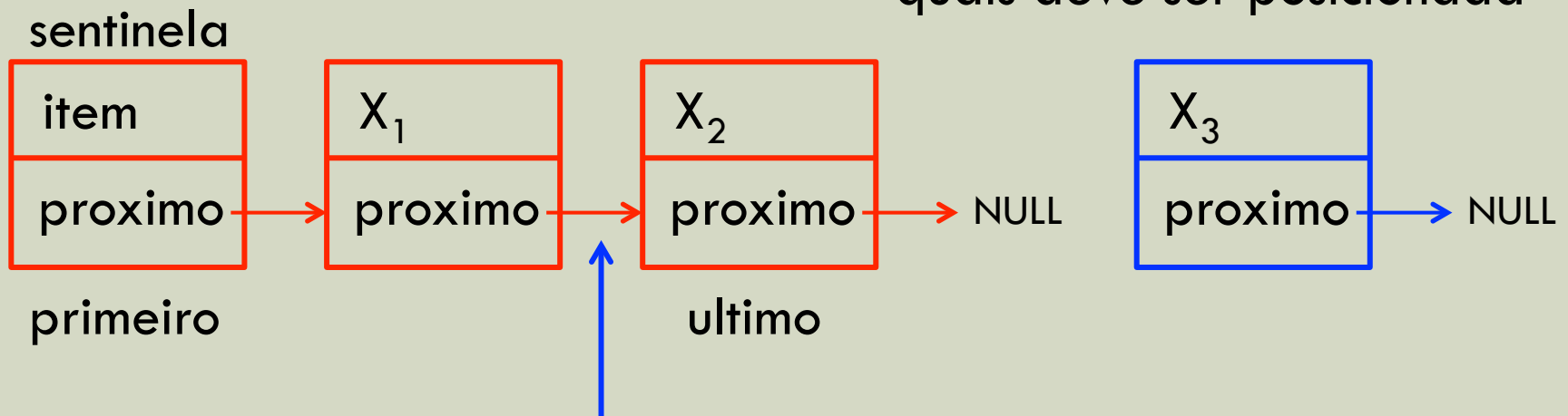
Criar uma célula nova e colocar nela o novo item



# LISTAS ENCADEADAS – OPERAÇÕES

Inserir na posição 1 da lista

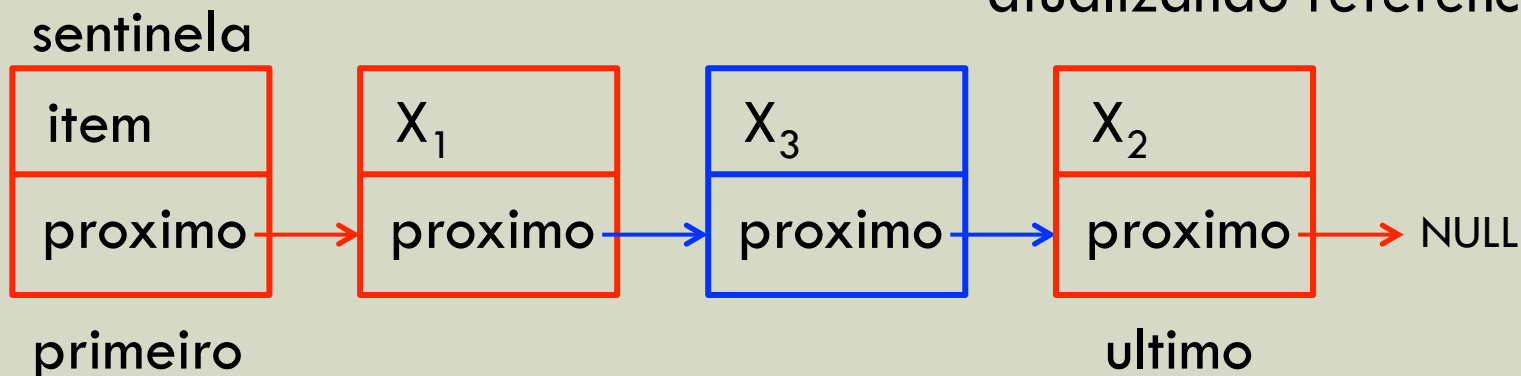
“Ligá-la” às células entre as  
quais deve ser posicionada



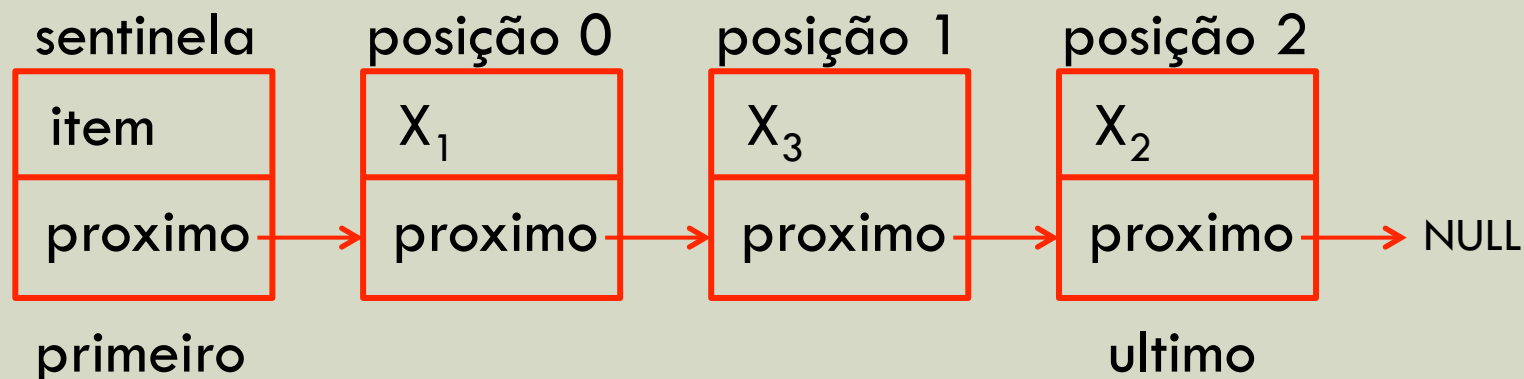
# LISTAS ENCADEADAS – OPERAÇÕES

Inserir na posição 1 da lista

“Ligá-la” às células entre as  
quais deve ser posicionada,  
atualizando referências



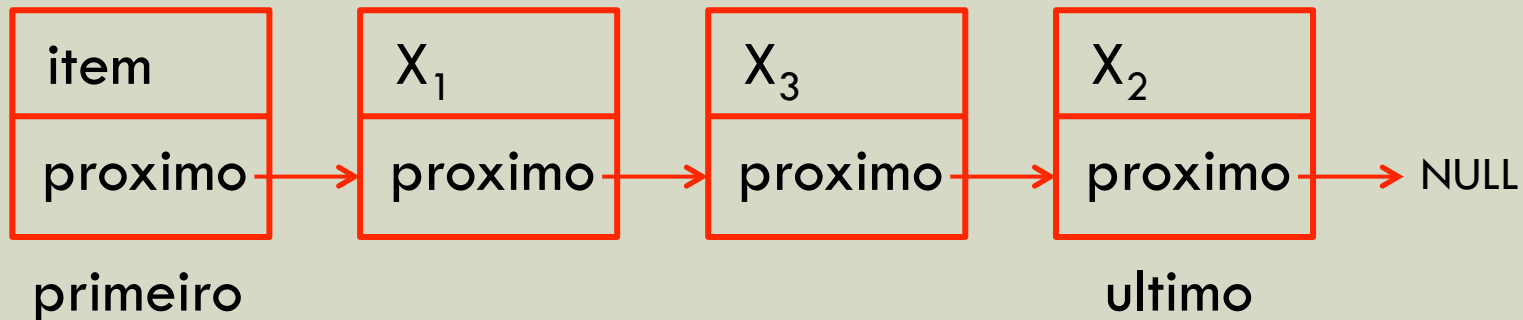
# LISTAS ENCADEADAS – OPERAÇÕES



# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_3$

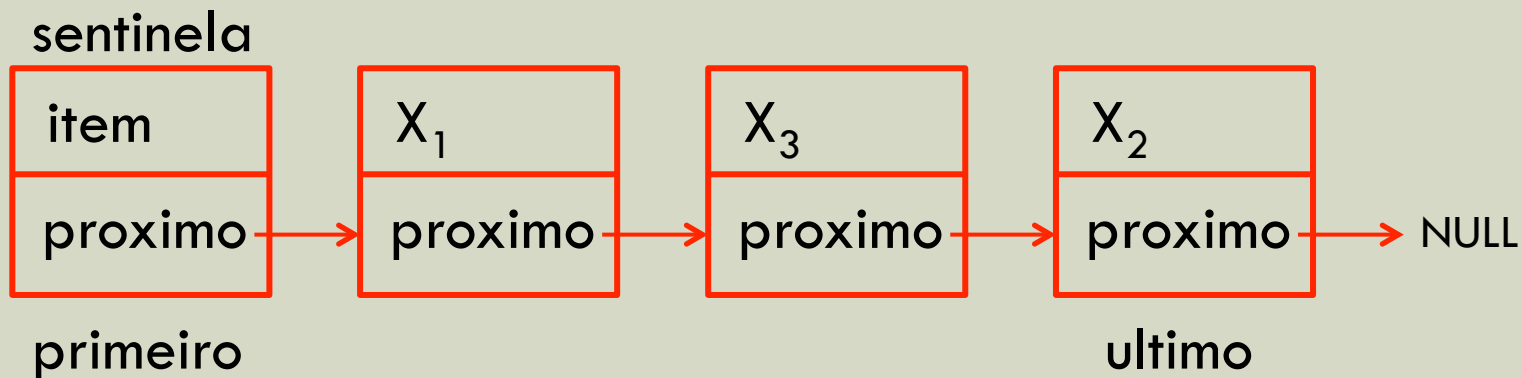
sentinela



# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_3$

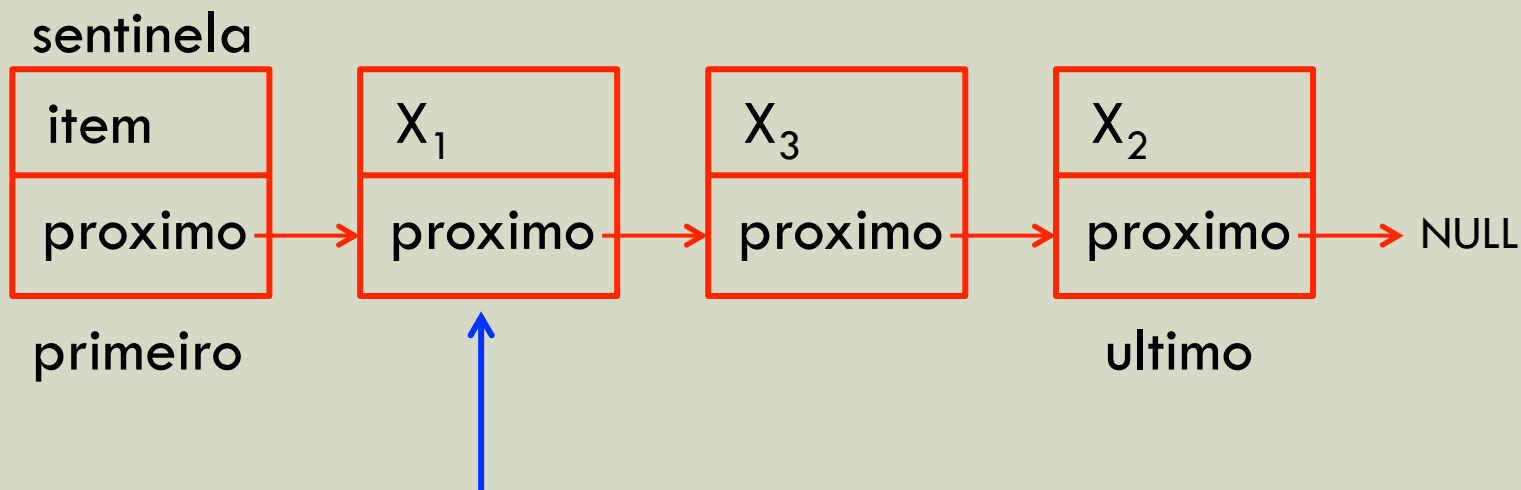
Localizar item que  
será retirado



# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_3$

Localizar item que  
será retirado

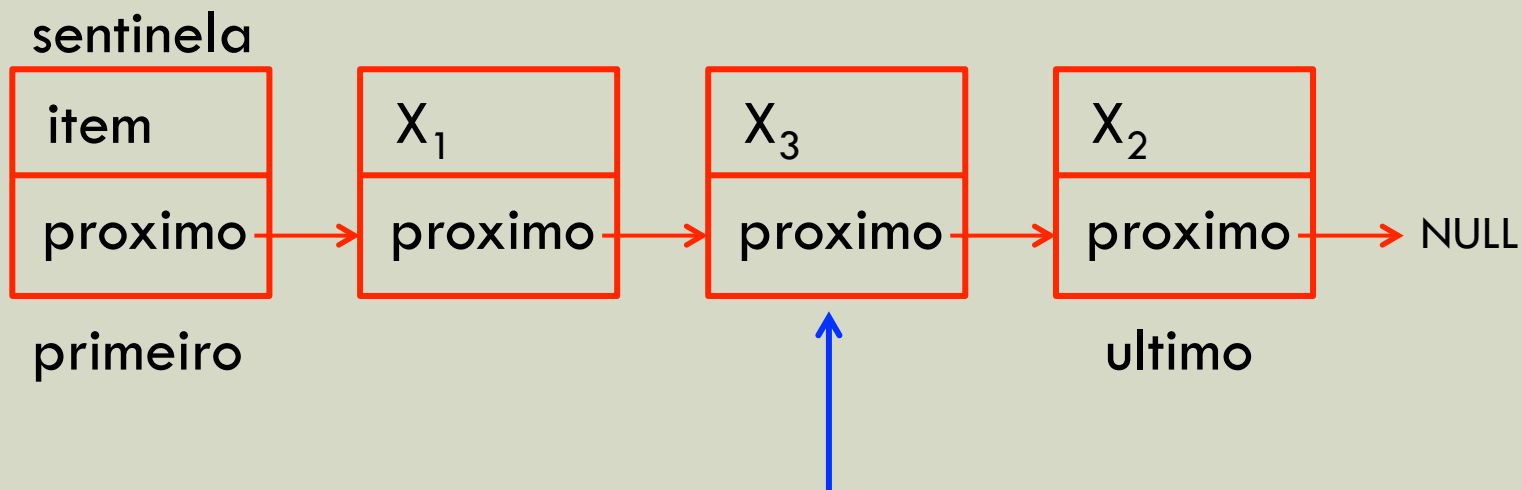




# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_3$

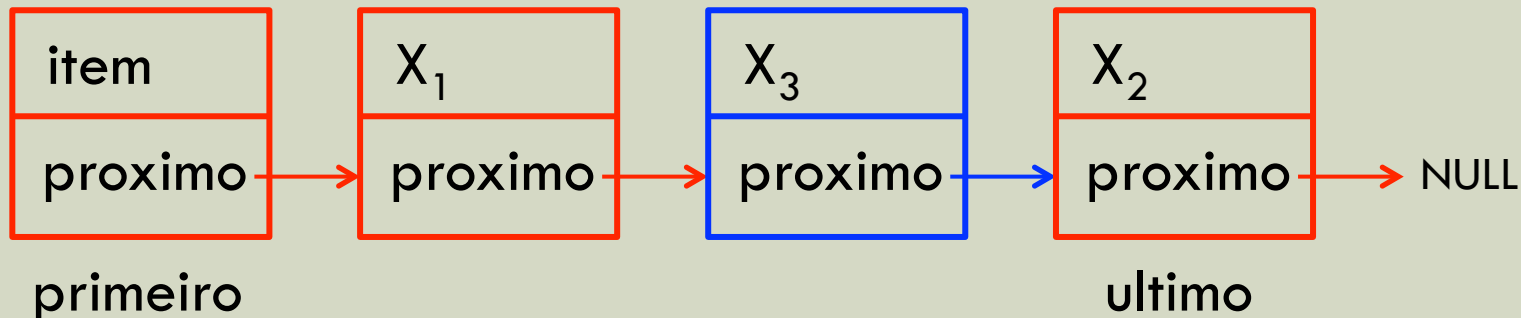
Localizar item que  
será retirado



# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_3$

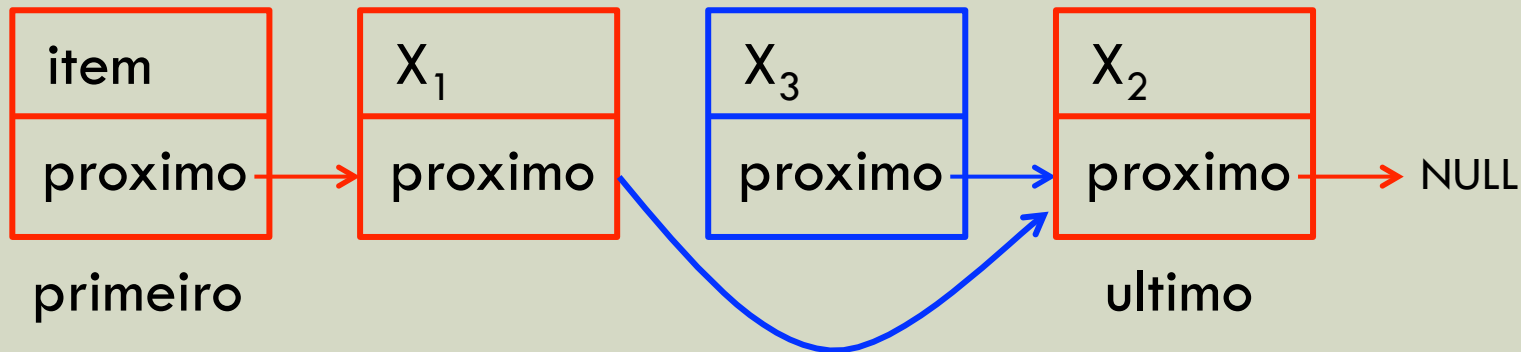
sentinela



# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_3$

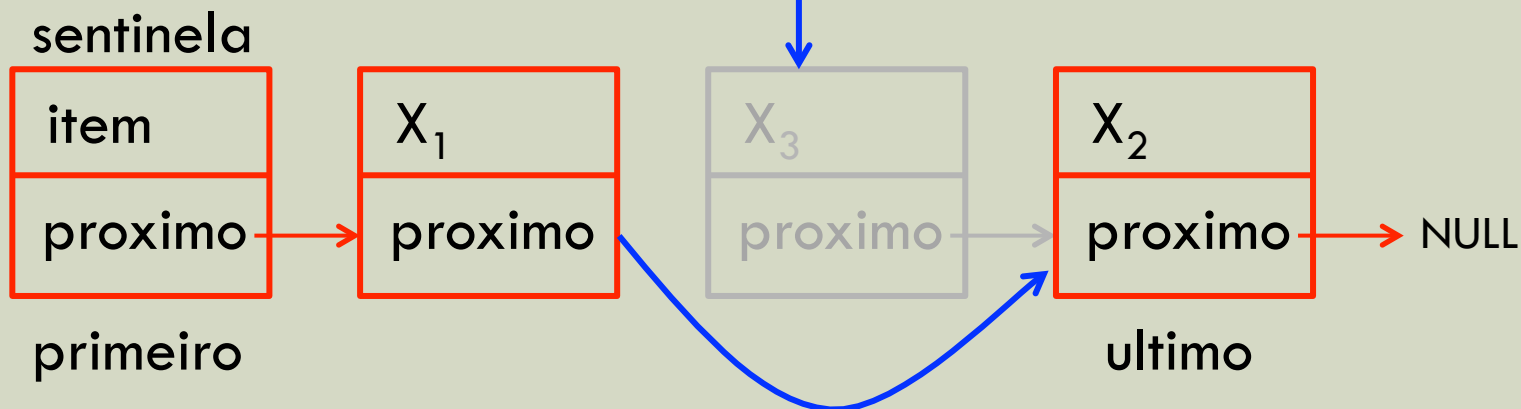
sentinela



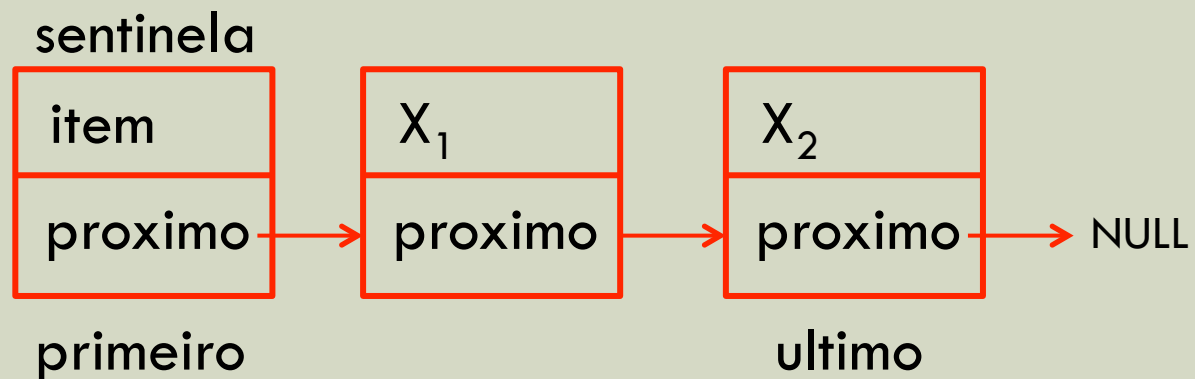
Atualizar referências, “ligando” as células entre as quais localiza-se o item retirado, retirando-o da lista

# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_3$



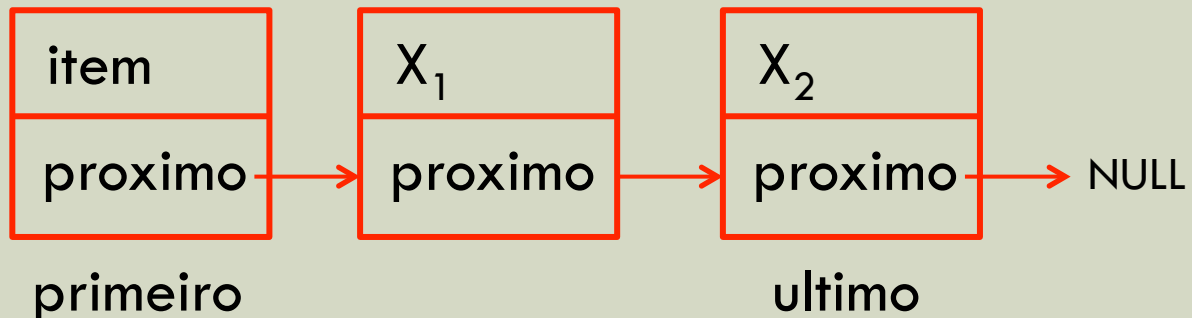
# LISTAS ENCADEADAS – OPERAÇÕES



# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_2$

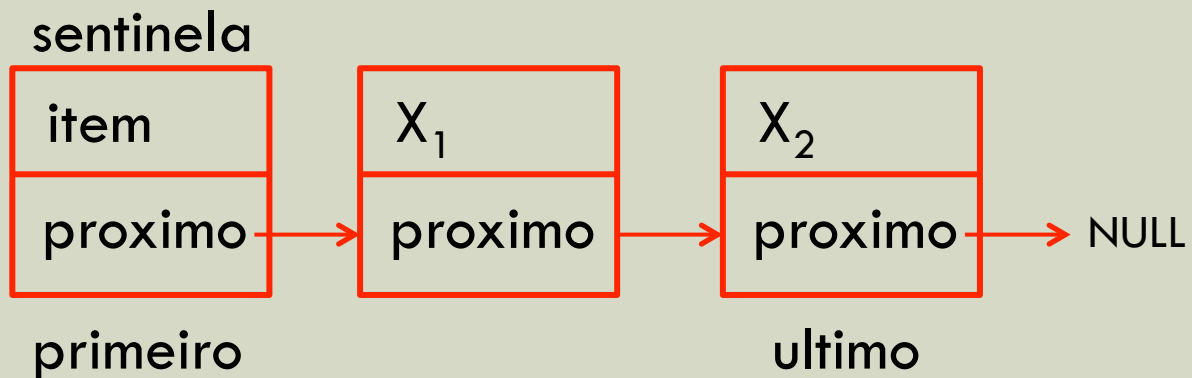
sentinela



# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_2$

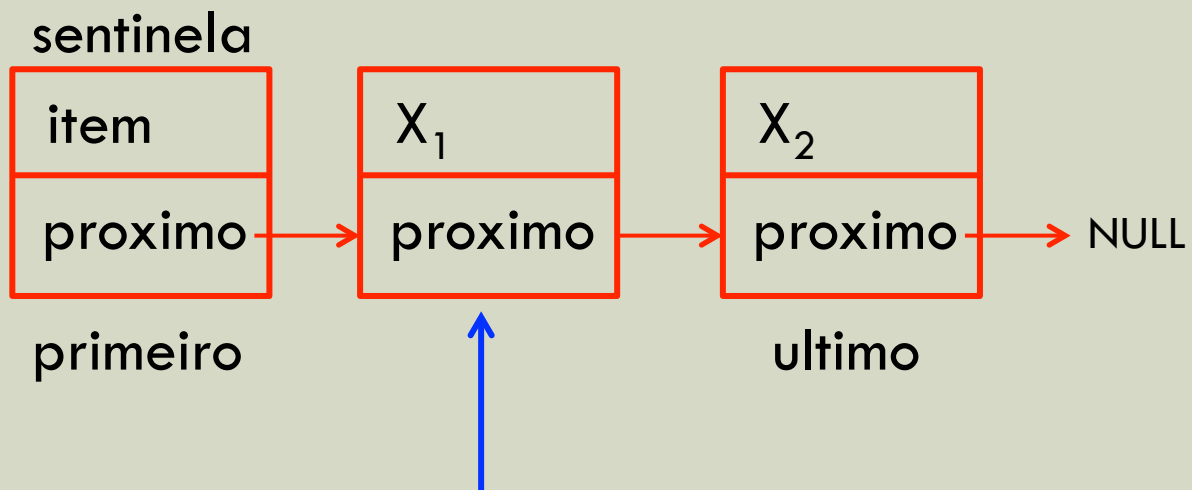
Localizar item que  
será retirado



# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_2$

Localizar item que  
será retirado

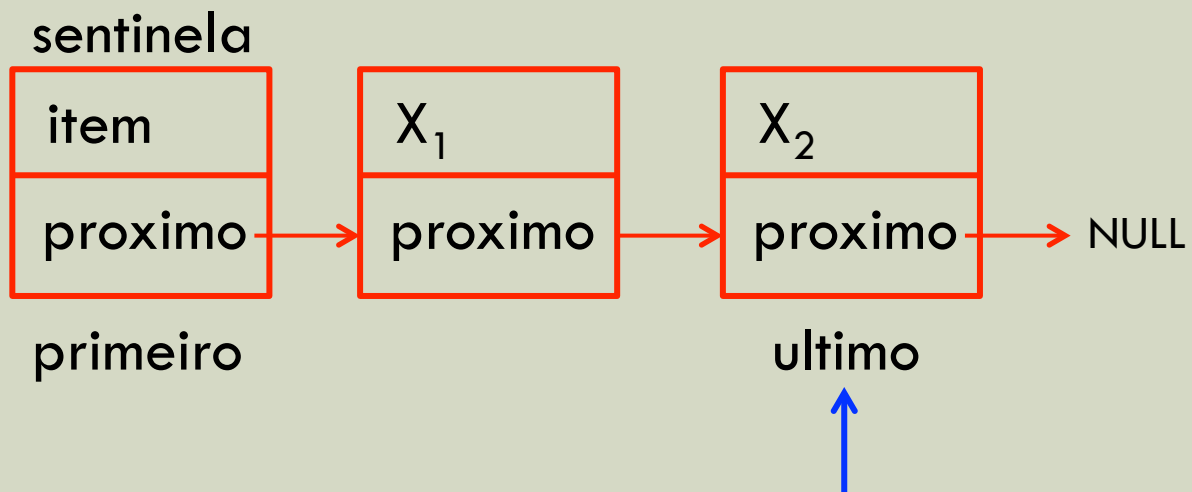




# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_2$

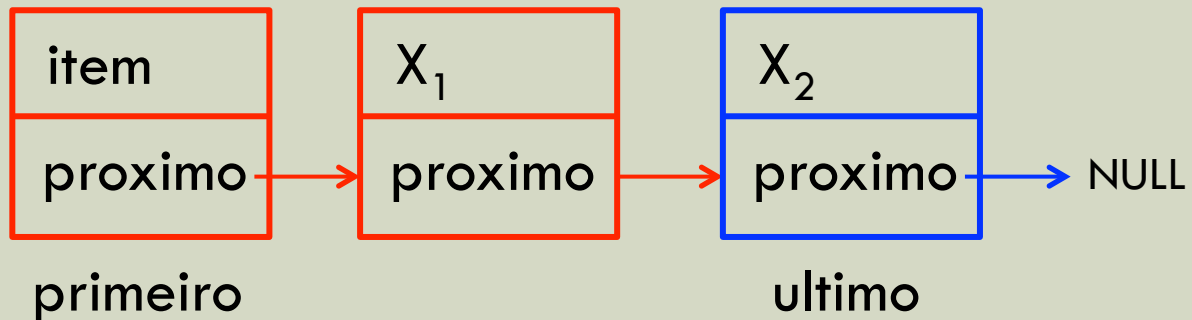
Localizar item que  
será retirado



# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_2$

sentinela

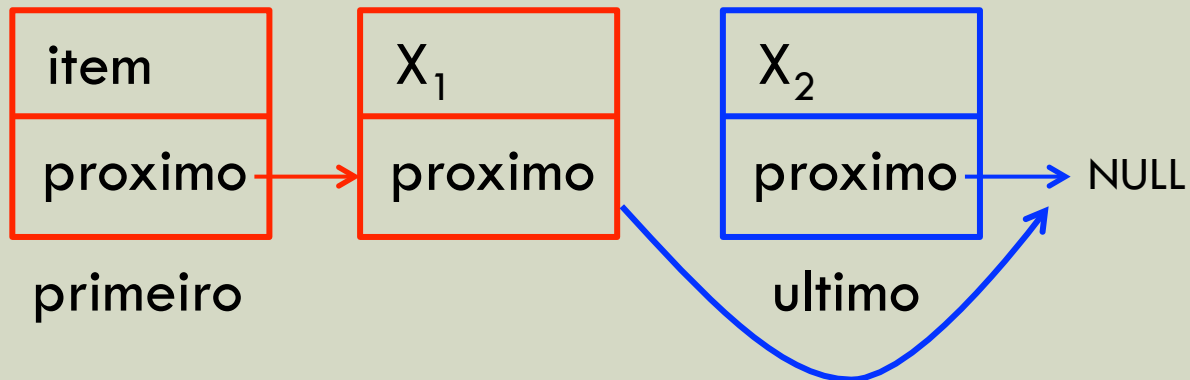


Atualizar referências, “ligando” as células entre as quais localiza-se o item retirado, retirando o item da lista

# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_2$

sentinela

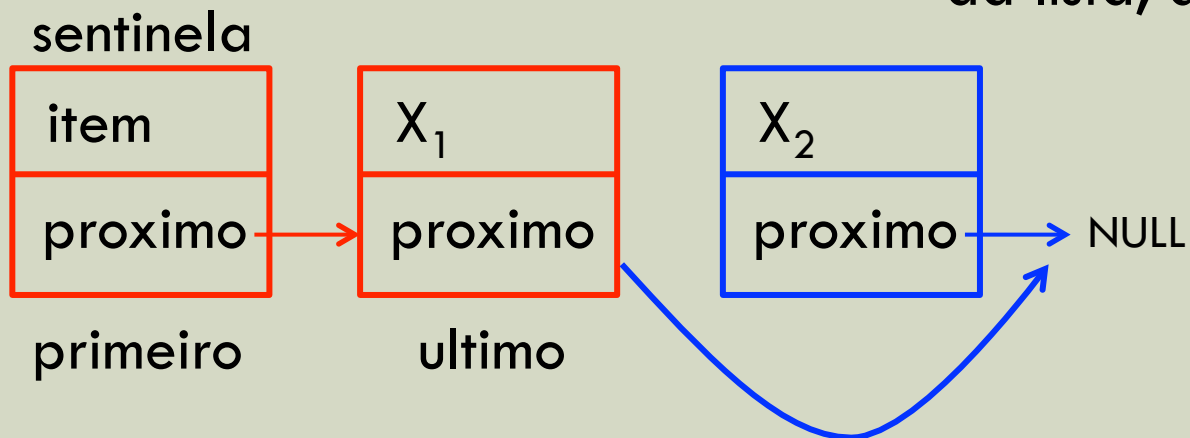


Atualizar referências, “ligando” as células entre as quais localiza-se o item retirado, retirando o item da lista

# LISTAS ENCADEADAS – OPERAÇÕES

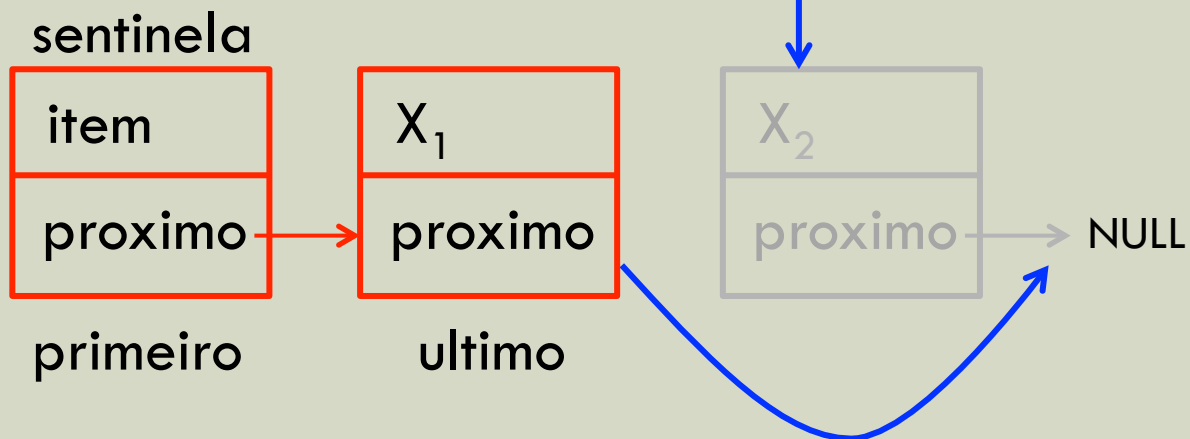
Retirar:  $X_2$

Se o item retirado for o último da lista, atualizar “último”.

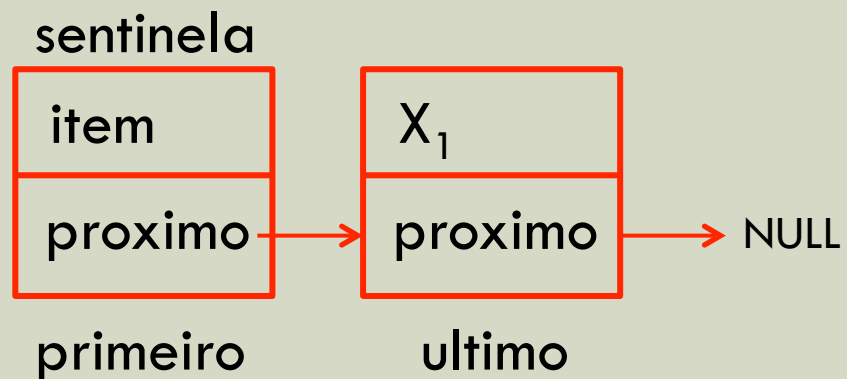


# LISTAS ENCADEADAS – OPERAÇÕES

Retirar:  $X_2$



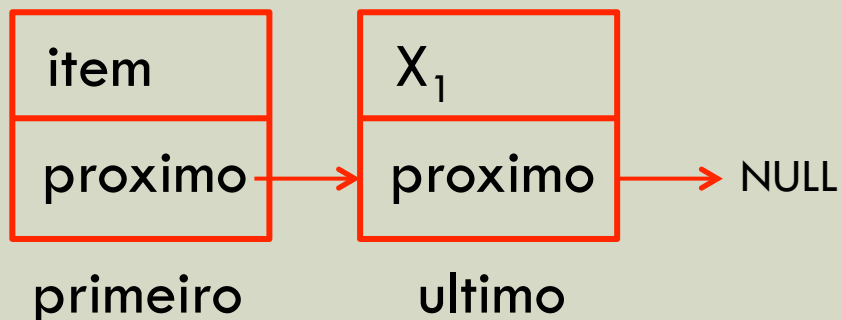
# LISTAS ENCADEADAS – OPERAÇÕES



# LISTAS ENCADEADAS – OPERAÇÕES

Localizar:  $X_1$

sentinela

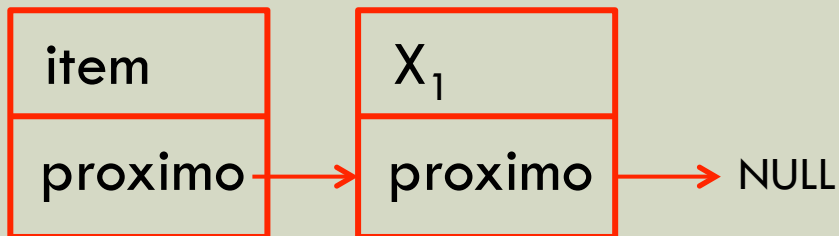


Percorrer a lista do início até o item desejado

# LISTAS ENCADEADAS – OPERAÇÕES

Localizar:  $X_1$

sentinela



primeiro

ultimo

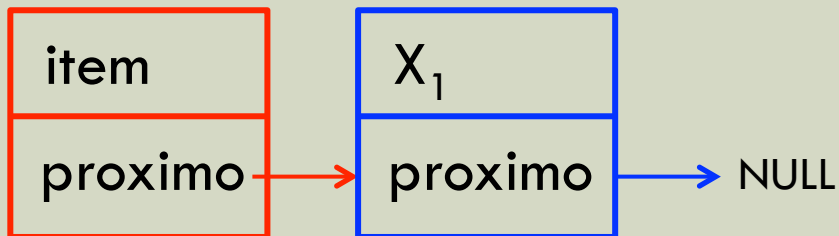
Percorrer a lista do  
início até o item  
desejado



# LISTAS ENCADEADAS – OPERAÇÕES

Localizar:  $X_1$

sentinela



primeiro

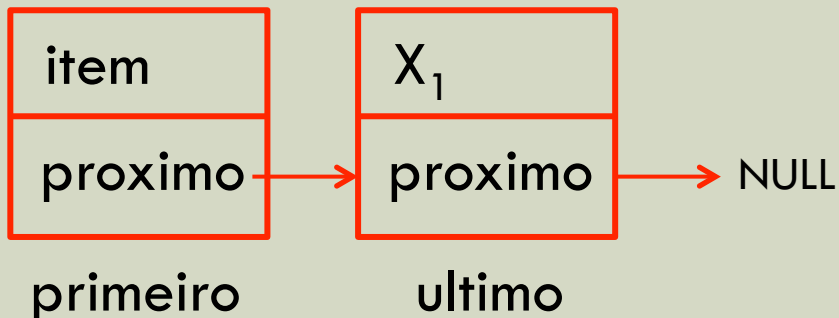
ultimo

Percorrer a lista do  
início até o item  
desejado

# LISTAS ENCADEADAS – OPERAÇÕES

Localizar:  $X_2$

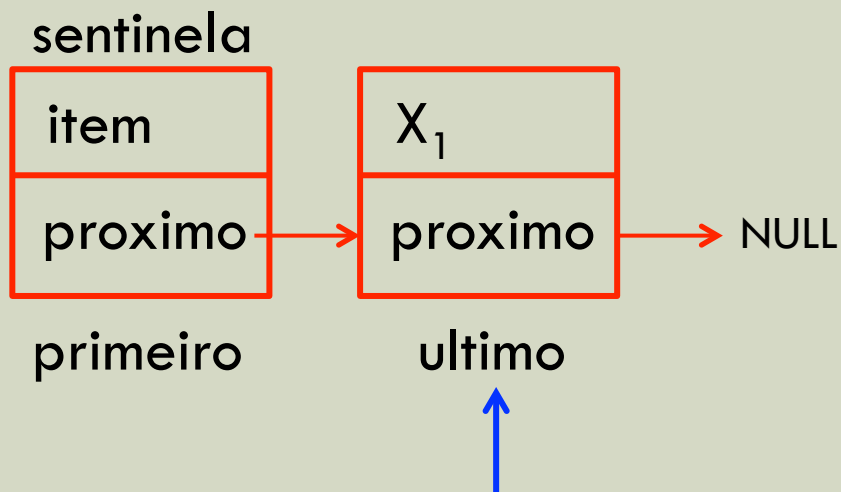
sentinela



Percorrer a lista do início até o item desejado ou até o final da lista, quando o item desejado não é localizado na lista

# LISTAS ENCADEADAS – OPERAÇÕES

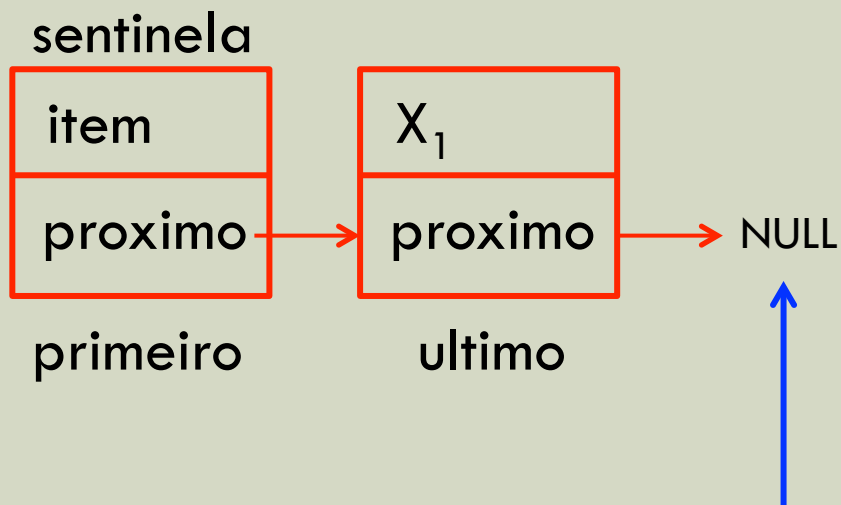
Localizar:  $X_2$



Percorrer a lista do início até o item desejado ou até o final da lista, quando o item desejado não é localizado na lista

# LISTAS ENCADEADAS – OPERAÇÕES

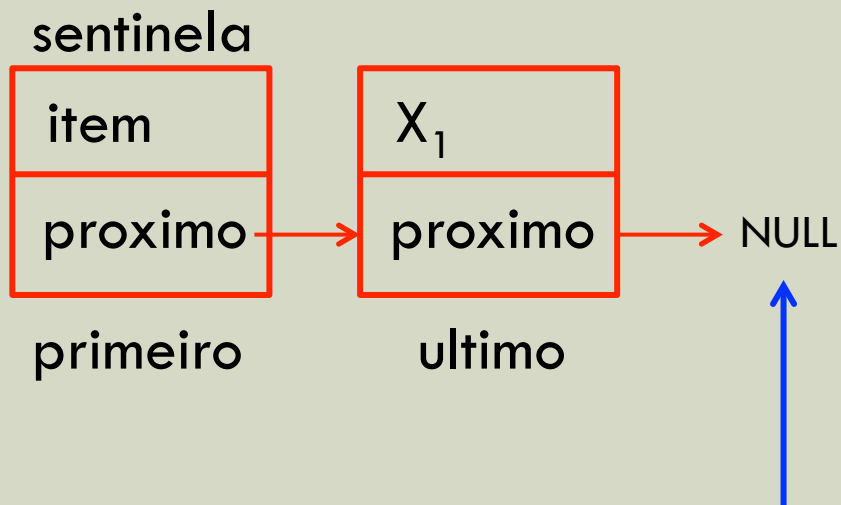
Localizar:  $X_2$



Percorrer a lista do início até o item desejado ou até o final da lista, quando o item desejado não é localizado na lista

# LISTAS ENCADEADAS – OPERAÇÕES

Localizar:  $X_2$



Percorrer a lista do início até o item desejado ou até o final da lista, quando o **item desejado não é localizado** na lista

# LISTAS ENCADEADAS

## ■ **Vantagens:**

- **custo para inserir ou retirar itens da lista;**
  - não há necessidade de deslocar os itens seguintes da lista.
- **boas para aplicações em que não existe previsão em relação ao crescimento da lista;**
  - estrutura de tamanho “infinito”.

## ■ **Desvantagem:**

- utilização de **memória adicional** para **armazenar as referências.**

# LISTAS ENCADEADAS – DIFICULDADES

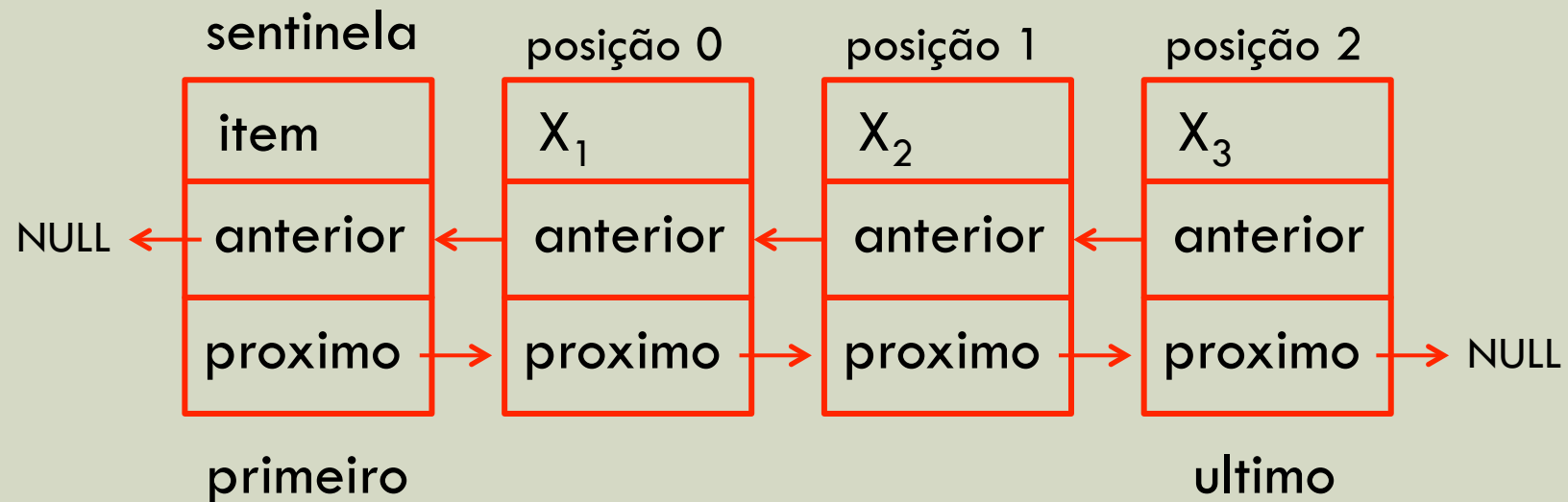
- Cada **célula** de uma **lista encadeada** contém **referência apenas para seu sucessor**;
  - não existe acesso imediato a seu antecessor.
- Para **remover a última célula** da lista encadeada;
  - deve-se **percorrer toda a lista até encontrar o penúltimo elemento**.
- Listas encadeadas longas e com muita utilização da operação de remoção da última célula;
  - podem apresentar **respostas lentas**.

# LISTAS DUPLAMENTE ENCADEADAS

- Cada célula da lista contém duas referências:
  - uma para seu **sucessor**;
    - próximo;
    - igual às listas encadeadas;
  - outra para seu **antecessor**;
    - anterior.
- Possibilitam que a lista seja percorrida nos dois sentidos.
- Permitem também **acessar os vizinhos do elemento atual**.

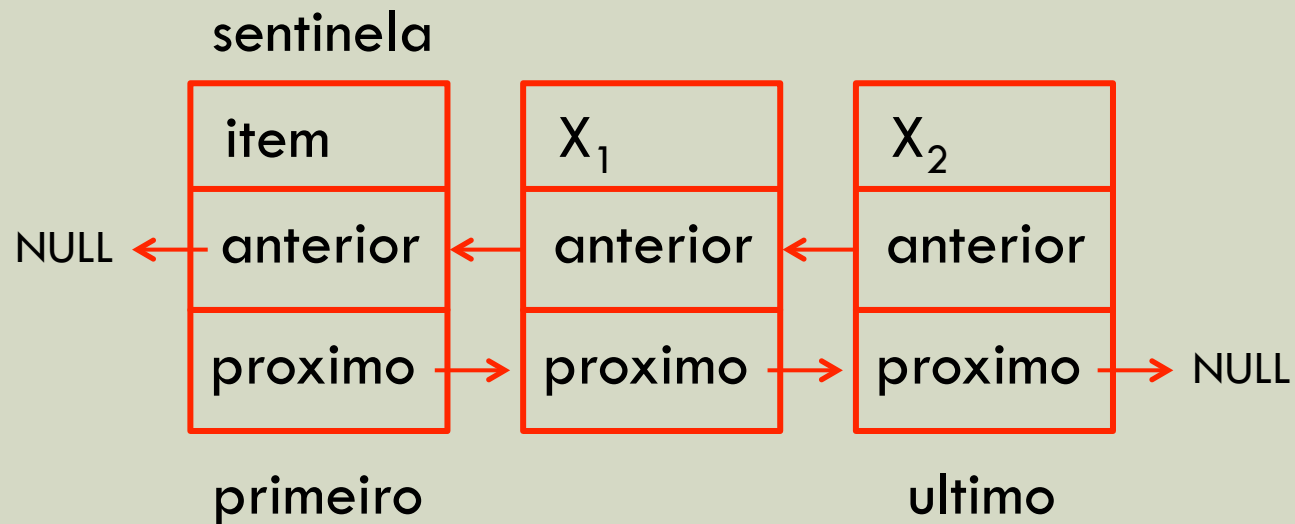


# LISTAS DUPLAMENTE ENCADEADAS



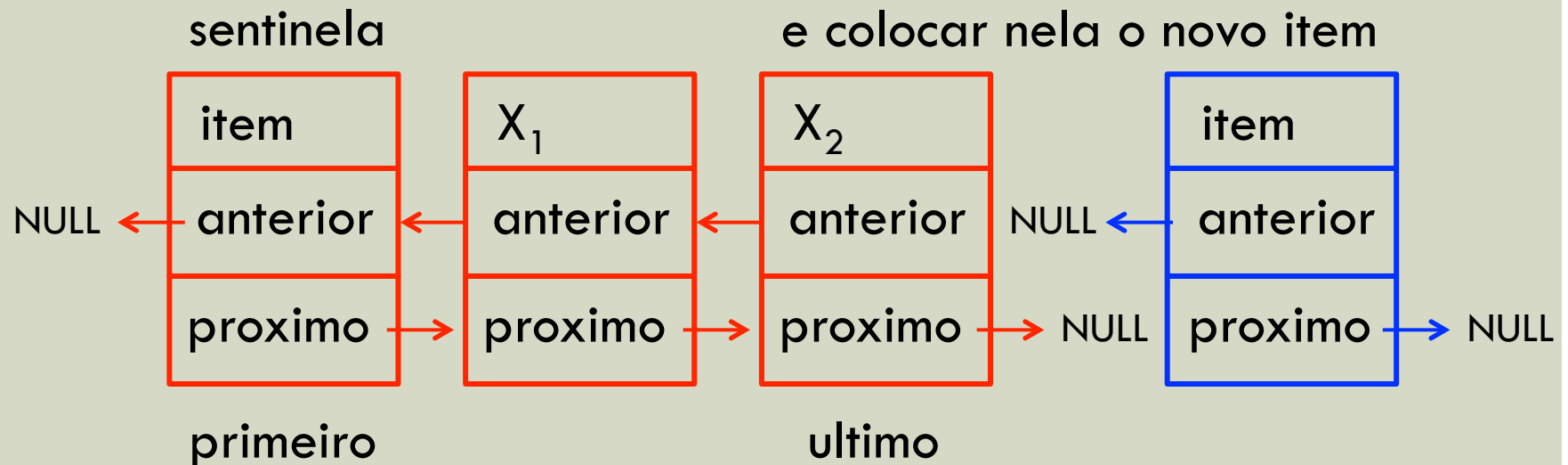
# LISTAS DUPLAMENTE ENCADEADAS – OPERAÇÕES

Inserir no “final” da lista



# LISTAS DUPLAMENTE ENCADEADAS – OPERAÇÕES

Inserir no “final” da lista



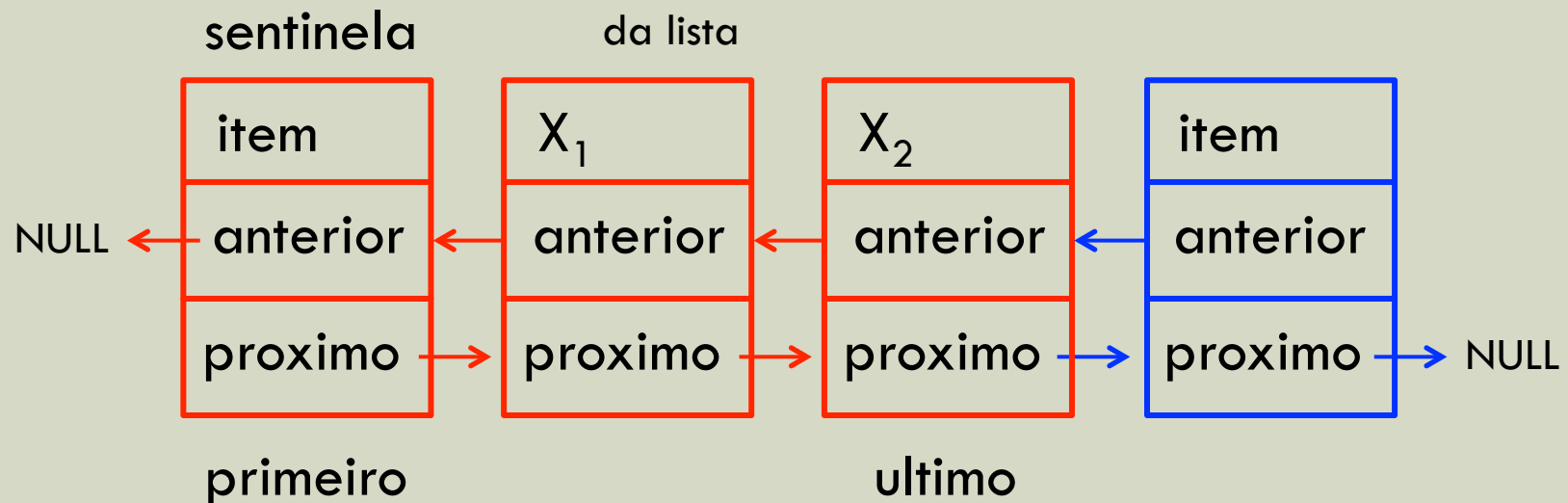
**Criar uma célula nova, ligá-la à célula referenciada por “último” e colocar nela o novo item**

# LISTAS DUPLAMENTE ENCADEADAS – OPERAÇÕES

Inserir no “final” da lista

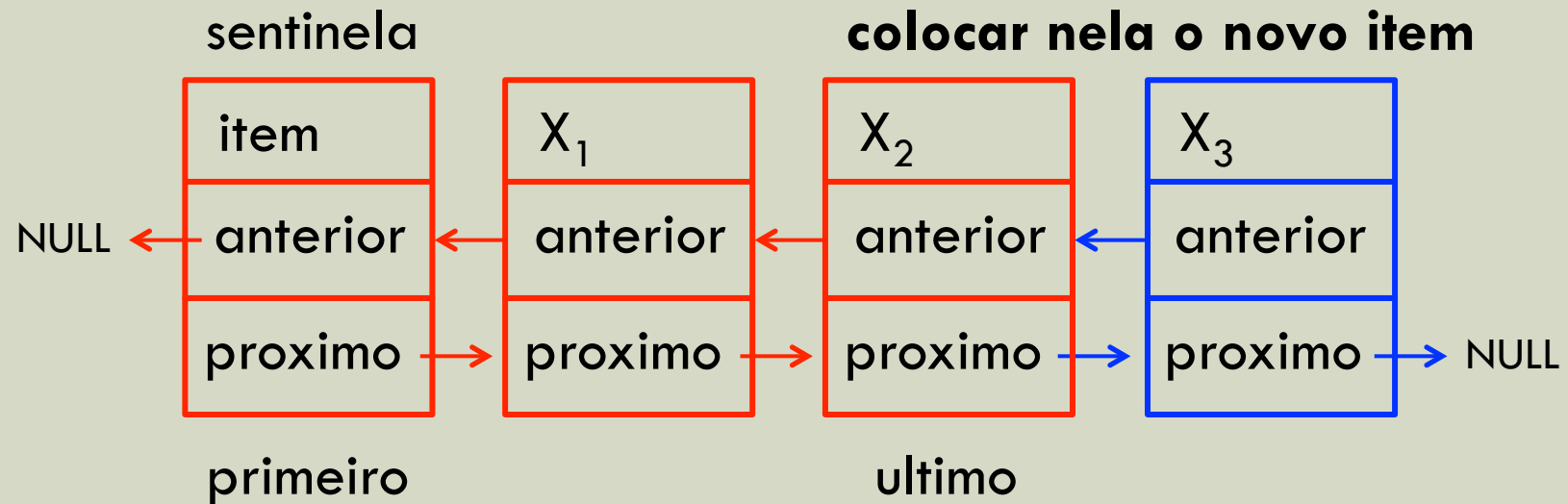
Criar uma célula nova, **ligá-la à célula referenciada por “último”** e colocar nela o novo item

“**anterior**” referencia o (até então) “**último**” elemento da lista



# LISTAS DUPLAMENTE ENCADEADAS – OPERAÇÕES

Inserir no “final” da lista

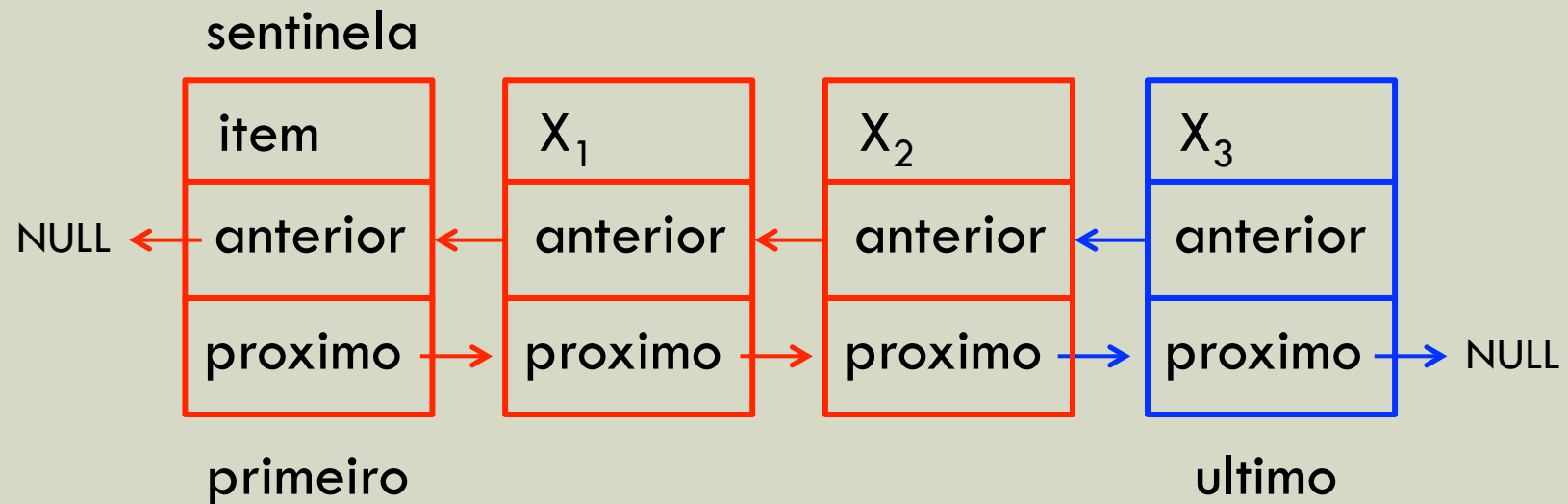


Criar uma célula nova, ligá-la à célula referenciada por “último” e **colocar nela o novo item**

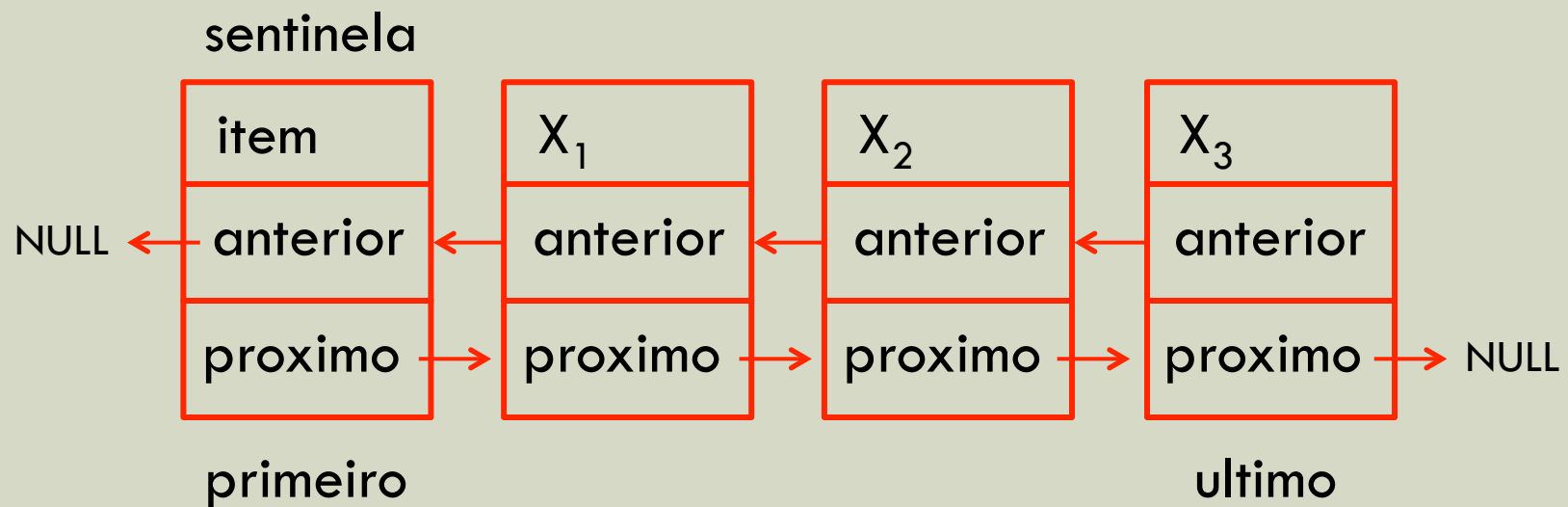
# LISTAS DUPLAMENTE ENCADEADAS – OPERAÇÕES

Inserir no “final” da lista

Atualizar “último”



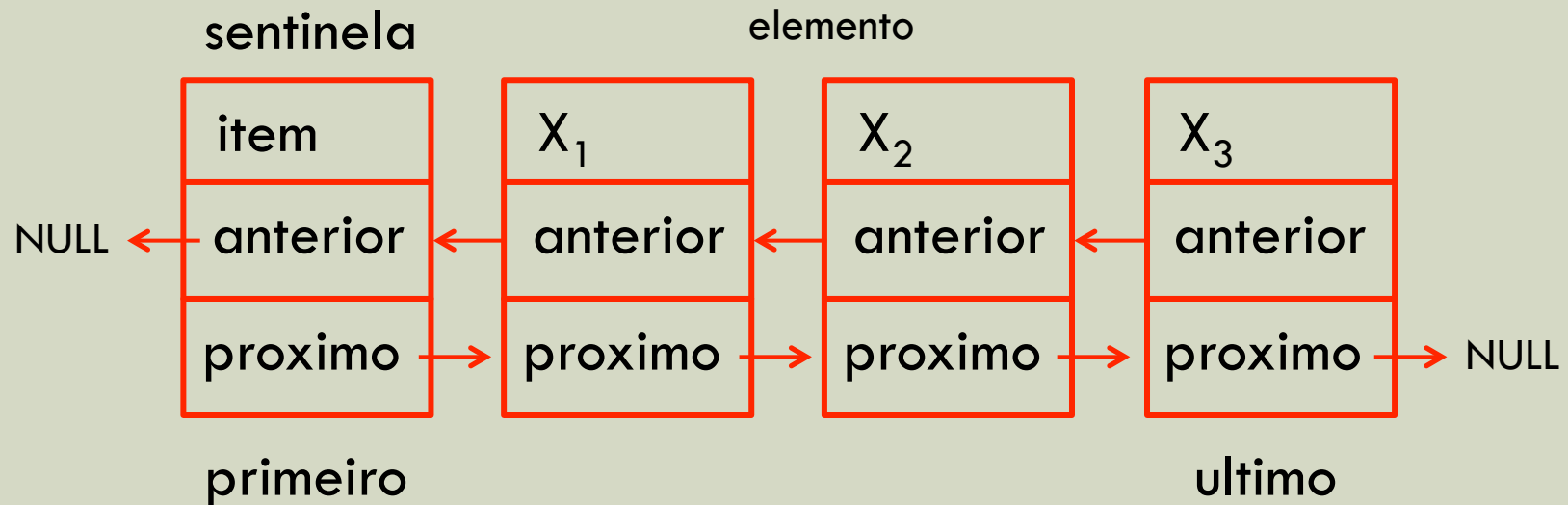
# LISTAS DUPLAMENTE ENCADEADAS – OPERAÇÕES



# LISTAS DUPLAMENTE ENCADEADAS – OPERAÇÕES

Remover no “final” da lista

Operação simplificada, pois a **última célula da lista referencia sua antecessora**. Assim, nenhum laço é necessário para a remoção do último elemento

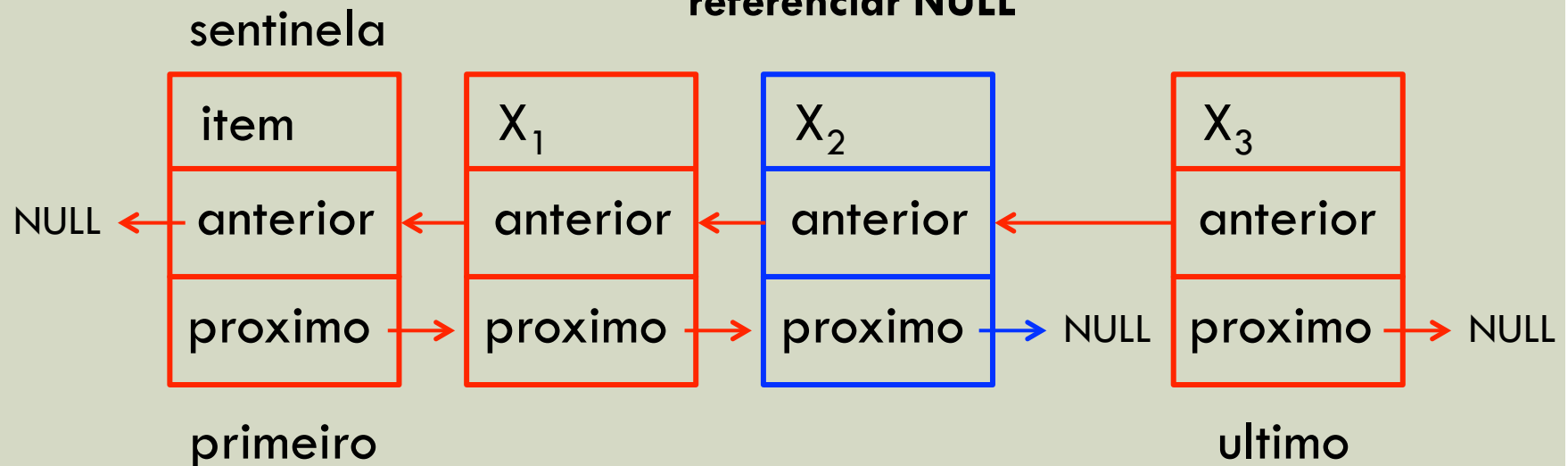




# LISTAS DUPLAMENTE ENCADEADAS – OPERAÇÕES

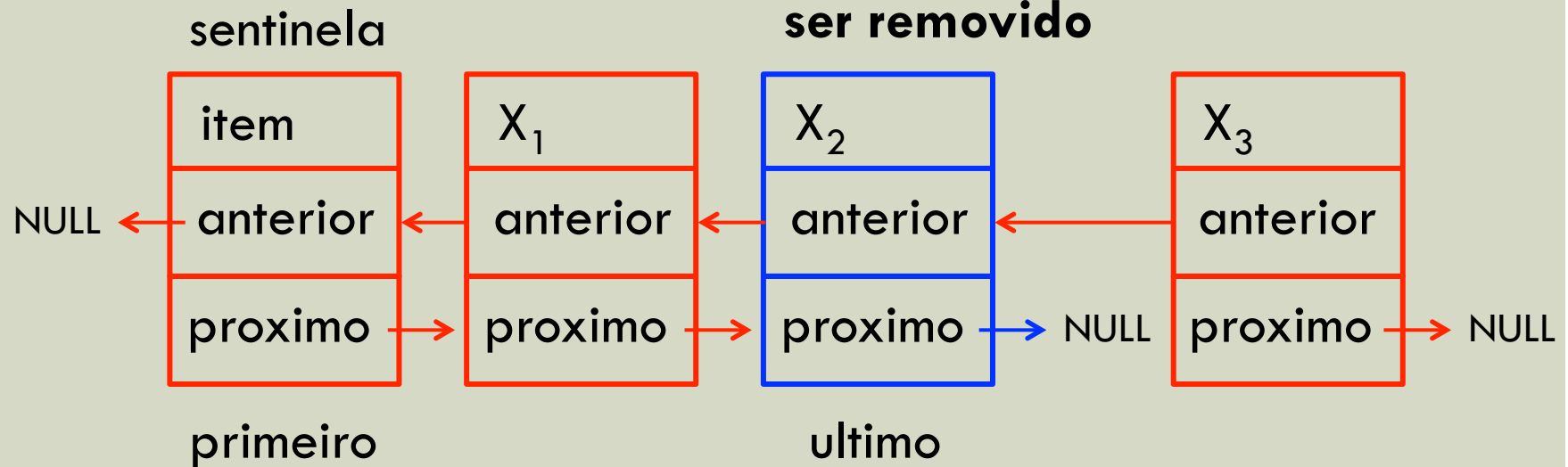
Remover no “final” da lista

Após recuperar o elemento a ser removido,  
“**próximo**” do antecessor passa a  
referenciar **NULL**



# LISTAS DUPLAMENTE ENCADEADAS – OPERAÇÕES

Remover no “final” da lista

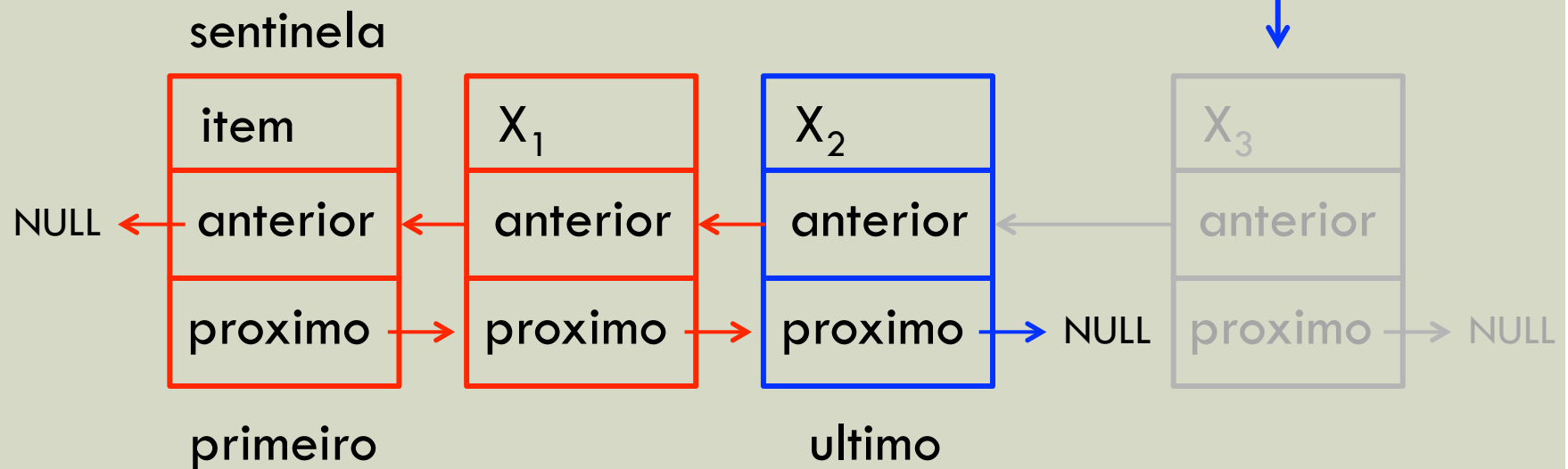


**Atualizar “último”** que passa a ser o **antecessor do elemento a ser removido**

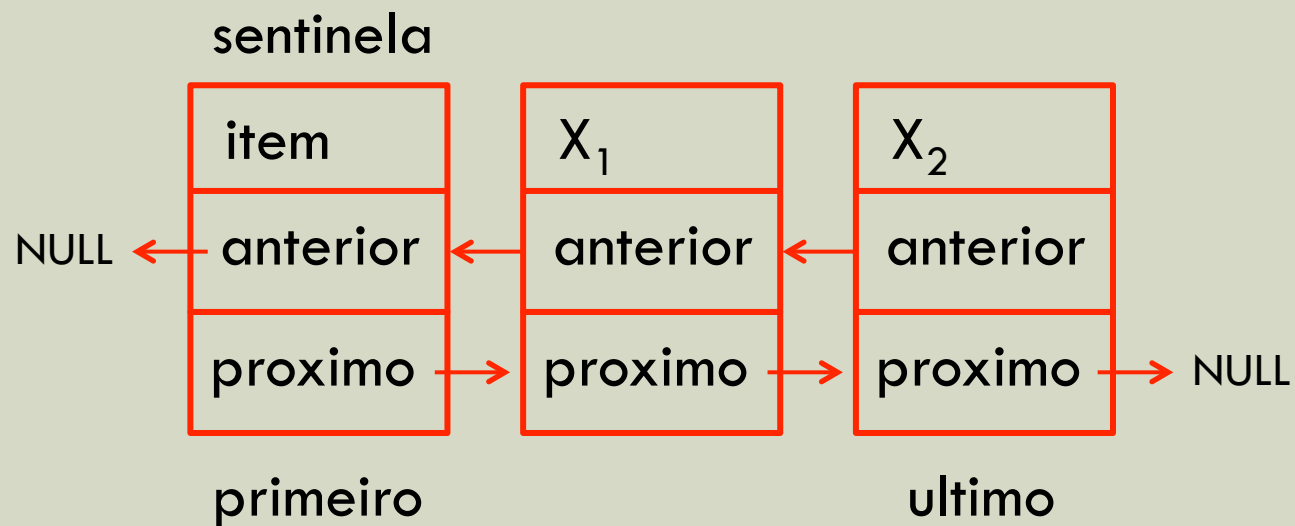
# LISTAS DUPLAMENTE ENCADEADAS – OPERAÇÕES

Remover no “final” da lista

Inacessível a partir da lista



# LISTAS DUPLAMENTE ENCADEADAS – OPERAÇÕES



# LISTAS DUPLAMENTE ENCADEADAS – OPERAÇÕES

- Devido ao **acesso imediato à última célula** de uma lista duplamente encadeada e à **sua antecessora**;
- a operação de **remoção do último elemento** é executada **sem a necessidade de percorrer-se toda a lista**.