

# Listas Dinâmicas

LDDE: Lista Dinâmica Duplamente  
Encadeada

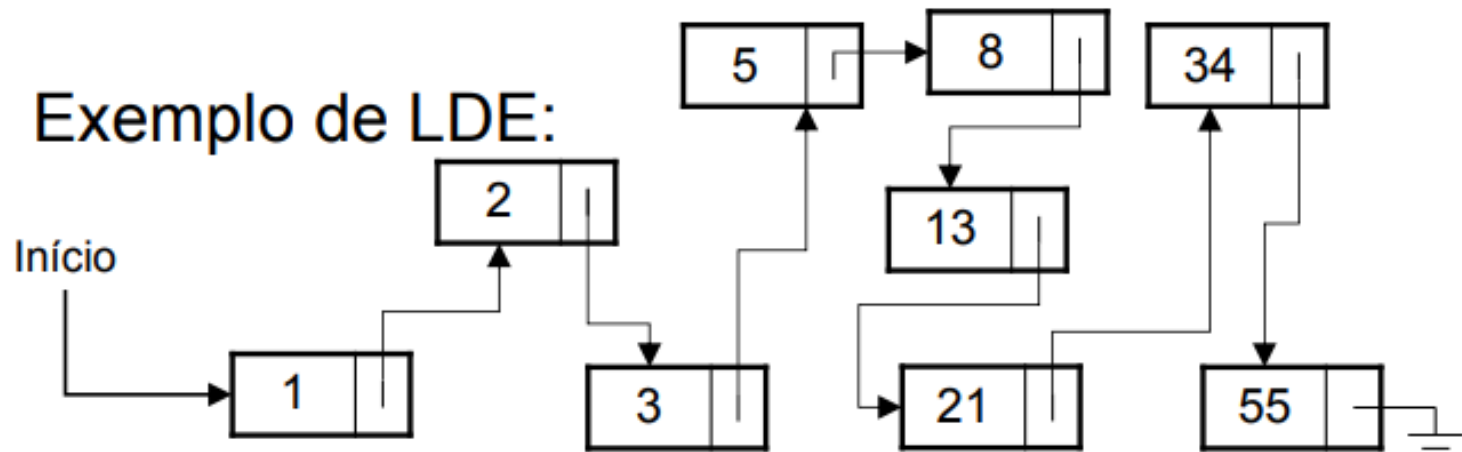
(Fonte: Material adaptado dos Slides do prof. Monael.)

# Lista Dinâmica Duplamente Encadeada

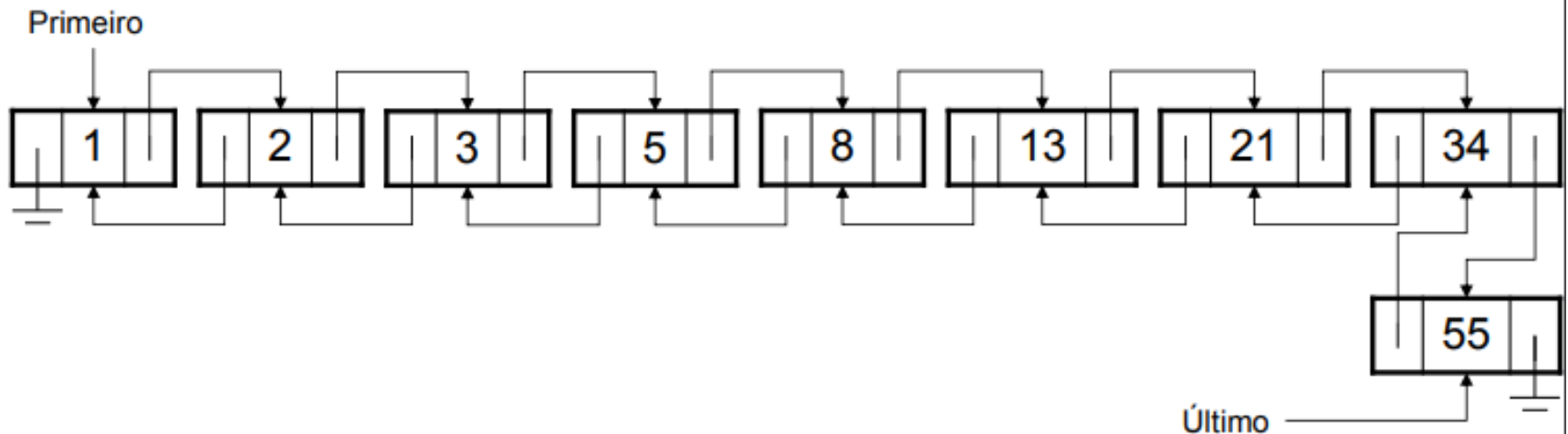
- Características LDDE:
  - Não há uma ordenação contígua.
    - Ou seja, na memória os elementos não estão ordenados de forma seqüencial como na LES.
  - A lista é ordenada através do encadeamento de seus ponteiros.
    - Embora não haja ordenação contígua, os dados devem estar ordenados se percorridos um a um através de seus apontadores.
- – Há apontadores nos dois sentidos.
  - Nas LDE cada item tem um ponteiro para o próximo item.
  - Nas LDDE cada item tem dois ponteiros.
    - Um para o próximo (como na LDE)
    - Um para o anterior.

# Lista Dinâmica Duplamente Encadeada

- Exemplo de LDE:



- Exemplo de LDDE:



# Lista Dinâmica Duplamente Encadeada

- Em Ciência da Computação uma lista dinâmica duplamente encadeada é uma estrutura de dados que:
  - Consiste de uma seqüência de registros
  - Cada registro tem um campo que contém dois ponteiros: um para o próximo item da lista e um para o item anterior da lista.

Ponteiro para o primeiro
Ponteiro para o último
quantidade

Lista

ponteiro para o anterior	<b>dados</b>	Ponteiro para o próximo
--------------------------	--------------	-------------------------

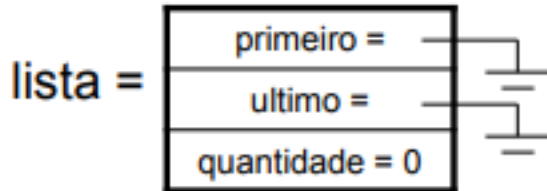
Item da lista

# Lista Dinâmica Duplamente Encadeada

- Operações Básicas
  - Inserção de um elemento na Lista
  - Eliminação de um elemento da Lista
  - Consulta da pertinência de um elemento na Lista

# Lista Dinâmica Duplamente Encadeada

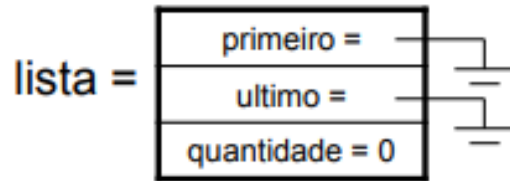
- Funcionamento (Inserção)



A priori a lista está vazia: inicio aponta para NULL e quantidade é 0.

# Lista Dinâmica Duplamente Encadeada

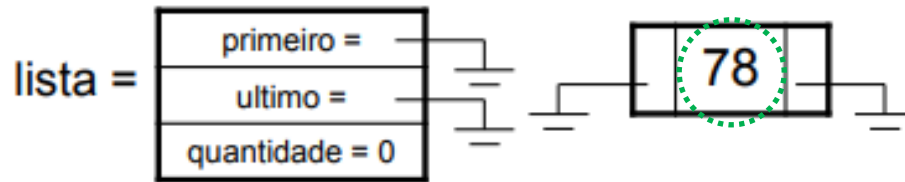
- Funcionamento (Inserção)



- 
- Inserir o elemento: 78
  - Alocar memória para um item.
  - Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
  - Caso a lista esteja vazia.
  - Faça lista no campo primeiro e campo ultimo apontar para o item
  - Incremente a quantidade em uma unidade.
-

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Inserção)

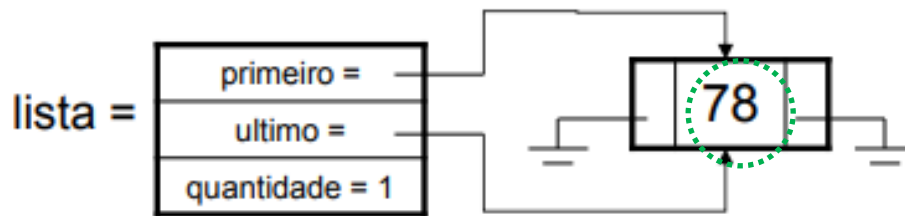


- • Inserir o elemento: 78
  - • Alocar memória para um item.
  - • Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
    - Caso a lista esteja vazia.
    - Faça lista no campo primeiro e campo ultimo apontar para o item
    - Incremente a quantidade em uma unidade.
-



# Lista Dinâmica Duplamente Encadeada

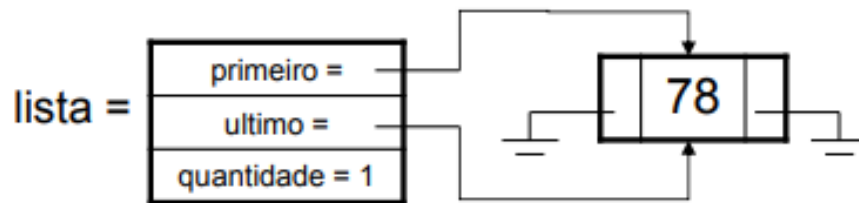
- Funcionamento (Inserção)



- Inserir o elemento: 78
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- • Caso a lista esteja vazia.
- • Faça lista no campo primeiro e campo ultimo apontar para o item
- • Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

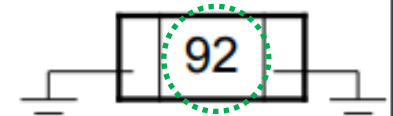
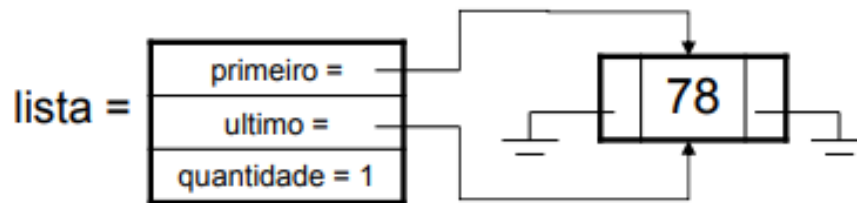
- Funcionamento (Inserção)



- • Inserir o elemento: 92
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item.
  - Atribua a lista no campo último o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

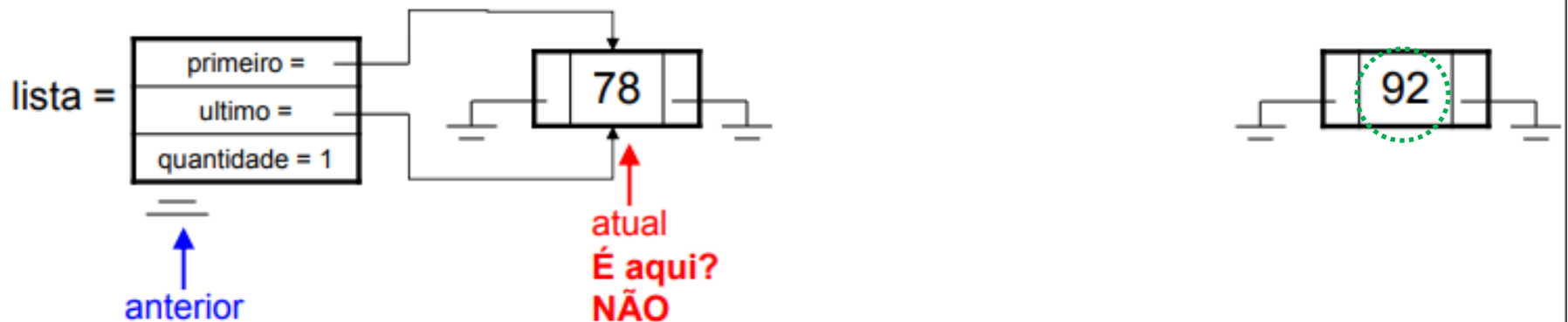
- Funcionamento (Inserção)



- • Inserir o elemento: 92
- • Alocar memória para um item.
- • Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item.
  - Atribua a lista no campo último o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Inserção)



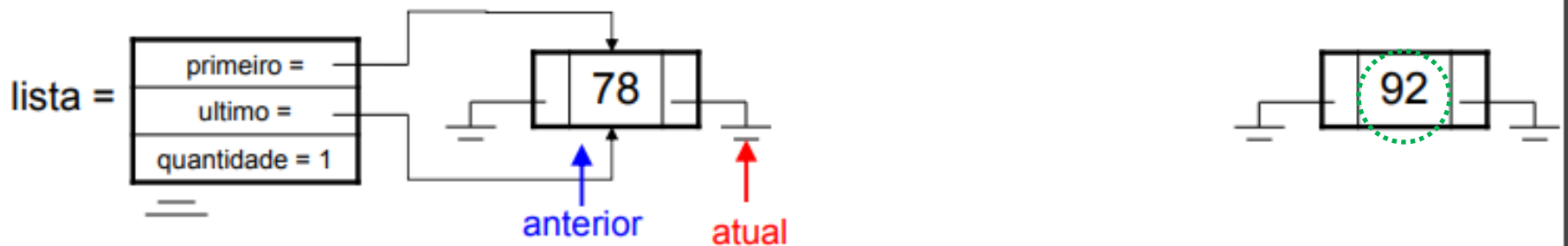
- Inserir o elemento: 92
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.

→ Se a lista não está vazia.

- • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item.
  - Atribua a lista no campo último o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

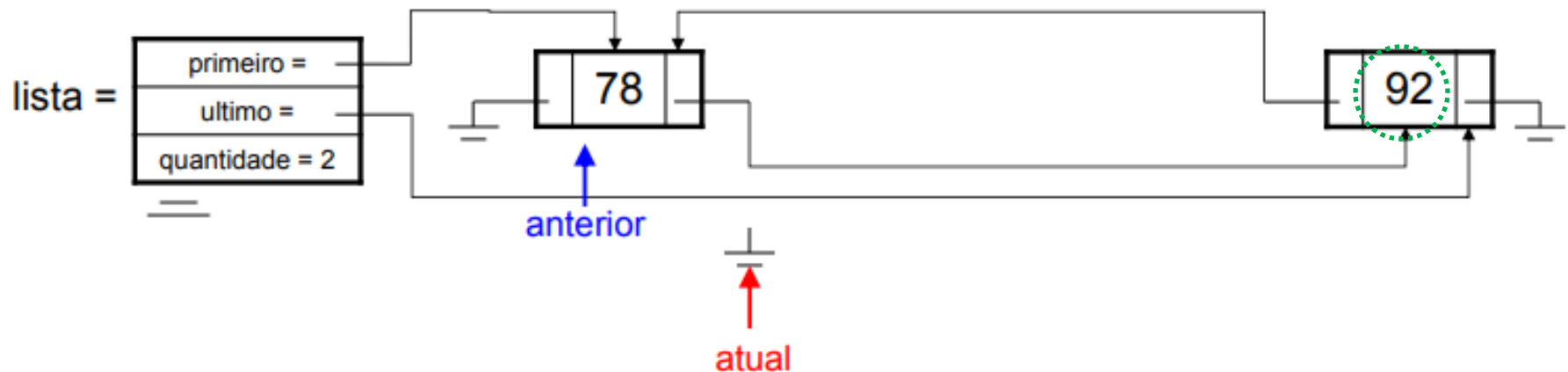
- Funcionamento (Inserção)



- Inserir o elemento: 92
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista desde o início a procura da posição correta do item ou o final da lista
    - Atribua ao campo anterior de novo item o endereço do elemento anterior.
    - Atribua ao campo próximo do elemento anterior o endereço do novo item.
    - Atribua a lista no campo último o endereço do novo item.
    - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Inserção)

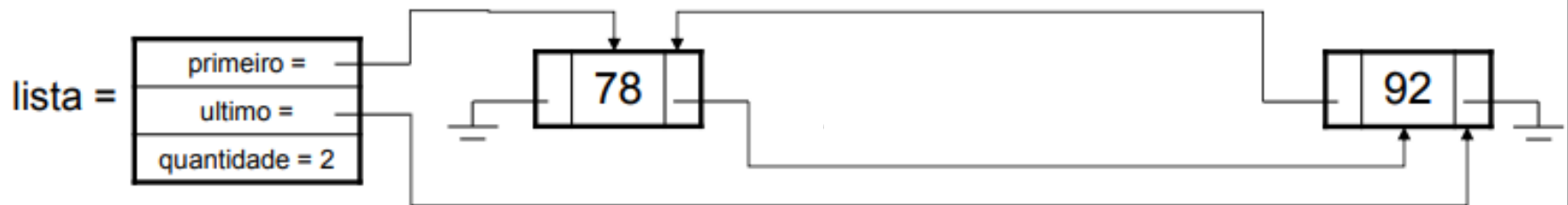


- Inserir o elemento: 92
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- • Atribua ao campo anterior de novo item o endereço do elemento anterior.
- • Atribua ao campo próximo do elemento anterior o endereço do novo item.
- • Atribua a lista no campo último o endereço do novo item.
- • Incremente a quantidade em uma unidade.



# Lista Dinâmica Duplamente Encadeada

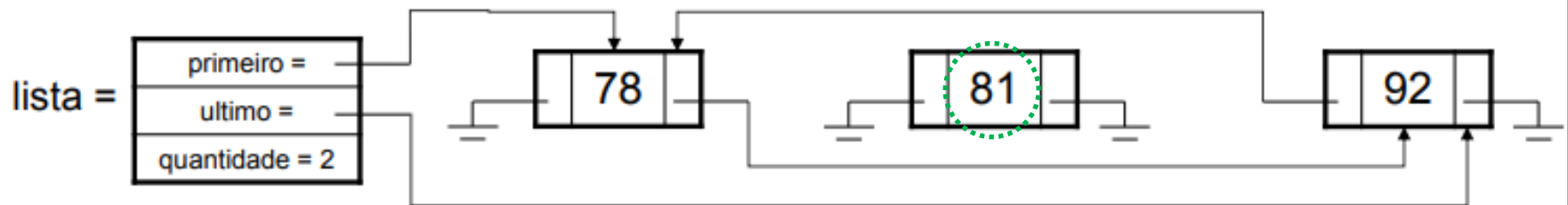
- Funcionamento (Inserção)



- • Inserir o elemento: 81
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item
  - Atribua ao campo anterior do item atual o endereço do item novo.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Inserção)

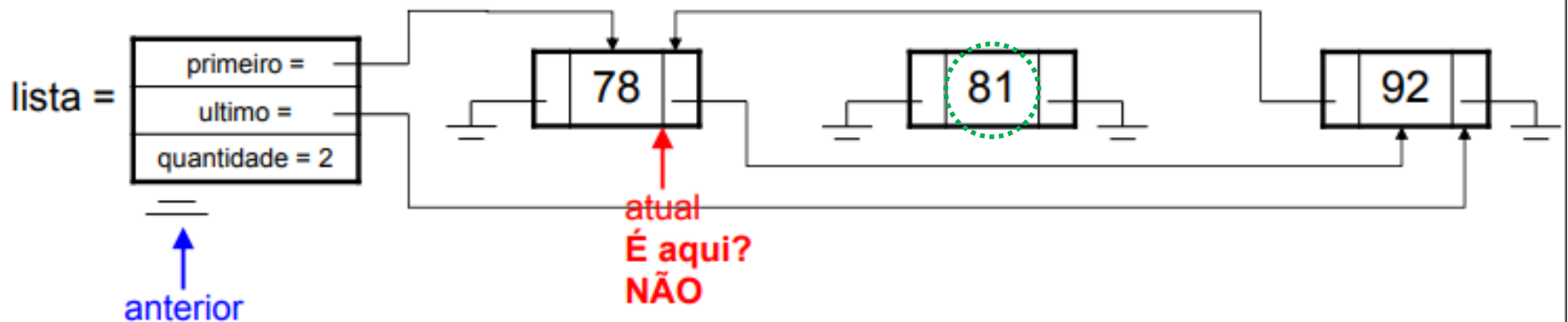


- • Inserir o elemento: 81
- • Alocar memória para um item.
- • Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item
  - Atribua ao campo anterior do item atual o endereço do item novo.
  - Incremente a quantidade em uma unidade.



# Lista Dinâmica Duplamente Encadeada

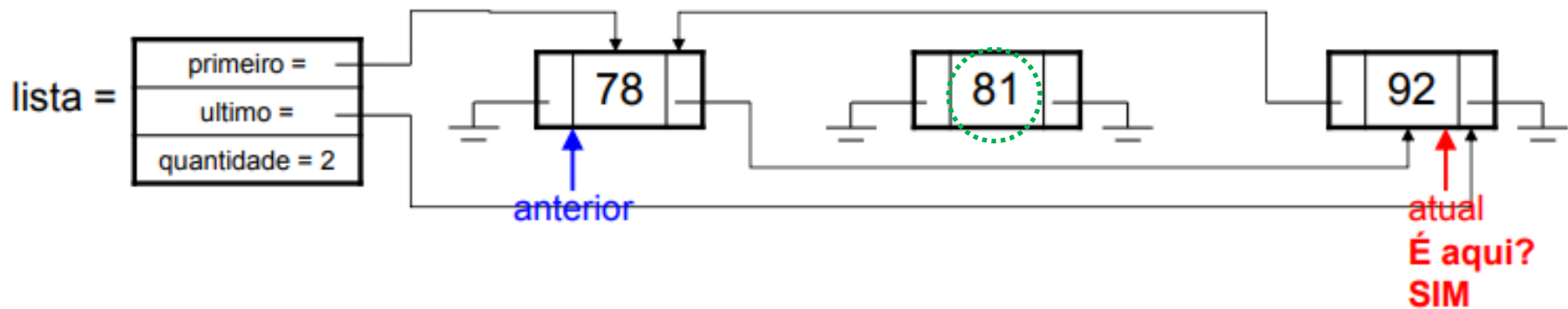
- Funcionamento (Inserção)



- Inserir o elemento: 81
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- • Se a lista não está vazia.
  - • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item
  - Atribua ao campo anterior do item atual o endereço do item novo.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

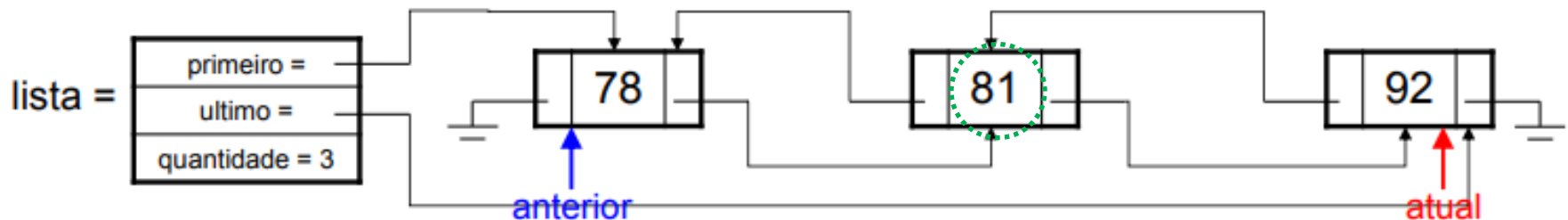
- Funcionamento (Inserção)



- Inserir o elemento: 81
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
    - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
    - Atribua ao campo anterior de novo item o endereço do elemento anterior.
    - Atribua ao campo próximo do elemento anterior o endereço do novo item
    - Atribua ao campo anterior do item atual o endereço do item novo.
    - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

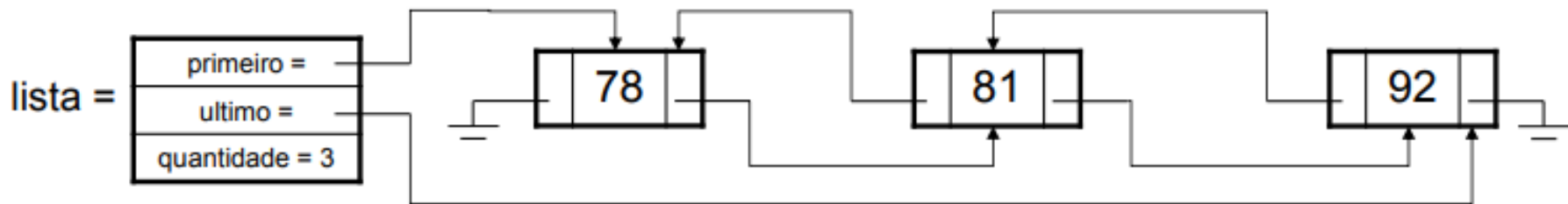
- Funcionamento (Inserção)



- Inserir o elemento: 81
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista desde o início a procura da posição correta do item ou o final da lista.
  - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item
  - Atribua ao campo anterior do item atual o endereço do item novo.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

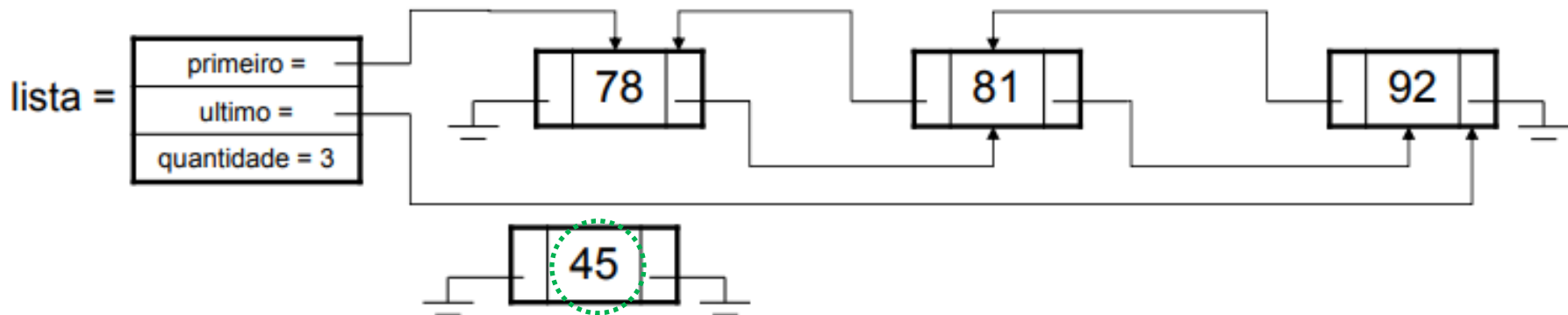
- Funcionamento (Inserção)



- • Inserir o elemento: 45
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Se o ponteiro anterior for NULL, então
    - Atribua ao campo próximo do novo item o valor de lista no campo primeiro.
    - Atribua campo primeiro da lista o endereço do novo item.
    - Atribua ao campo anterior do atual o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Inserção)

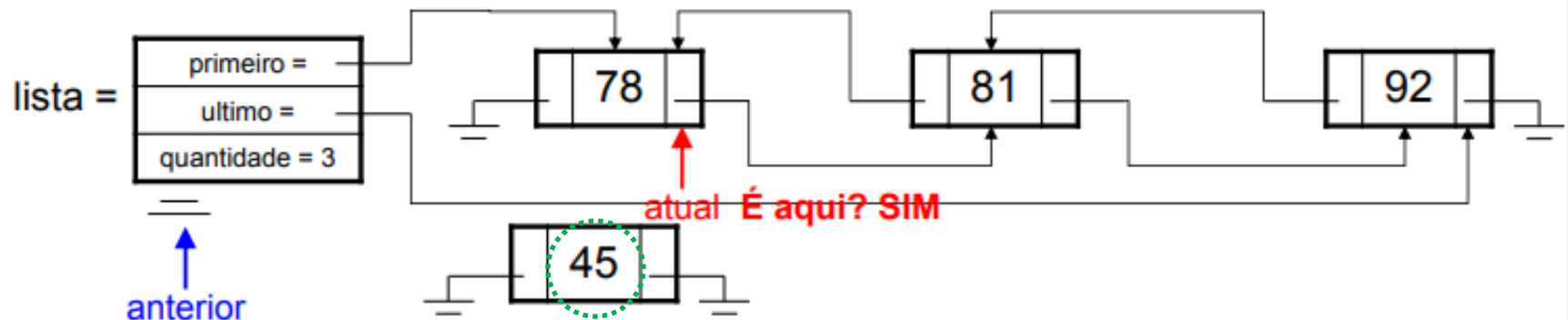


- Inserir o elemento: 45
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Se o ponteiro anterior for NULL, então
    - Atribua ao campo próximo do novo item o valor de lista no campo primeiro.
    - Atribua campo primeiro da lista o endereço do novo item.
    - Atribua ao campo anterior do atual o endereço do novo item.
  - Incremente a quantidade em uma unidade.



# Lista Dinâmica Duplamente Encadeada

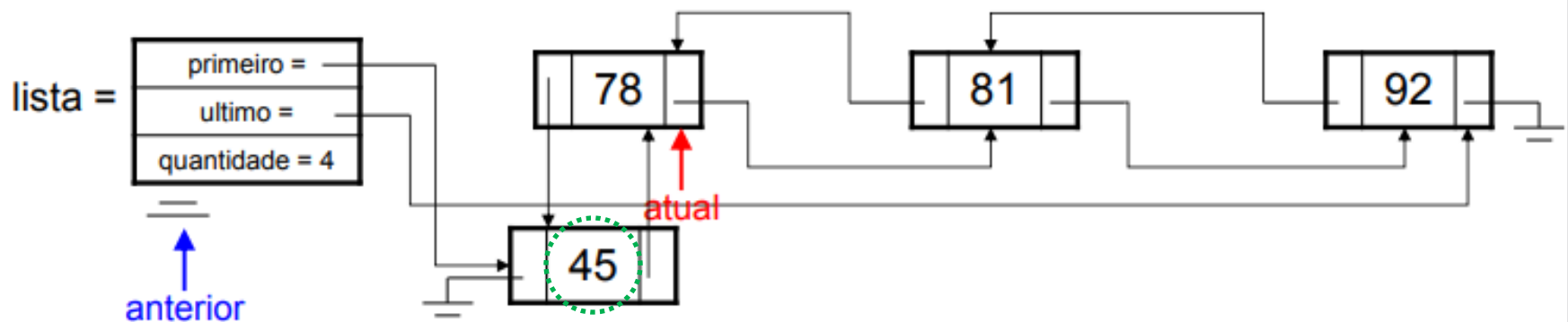
- Funcionamento (Inserção)



- Inserir o elemento: 45
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- • Se a lista não está vazia.
  - • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Se o ponteiro anterior for NULL, então
    - Atribua ao campo próximo do novo item o valor de lista no campo primeiro.
    - Atribua campo primeiro da lista o endereço do novo item.
    - Atribua ao campo anterior do atual o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Inserção)



- Inserir o elemento: 45
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- • Se o ponteiro anterior for NULL, então
  - • Atribua ao campo próximo do novo item o valor de lista no campo primeiro.
  - • Atribua campo primeiro da lista o endereço do novo item.
  - • Atribua ao campo anterior do atual o endereço do novo item.
- • Incremente a quantidade em uma unidade.

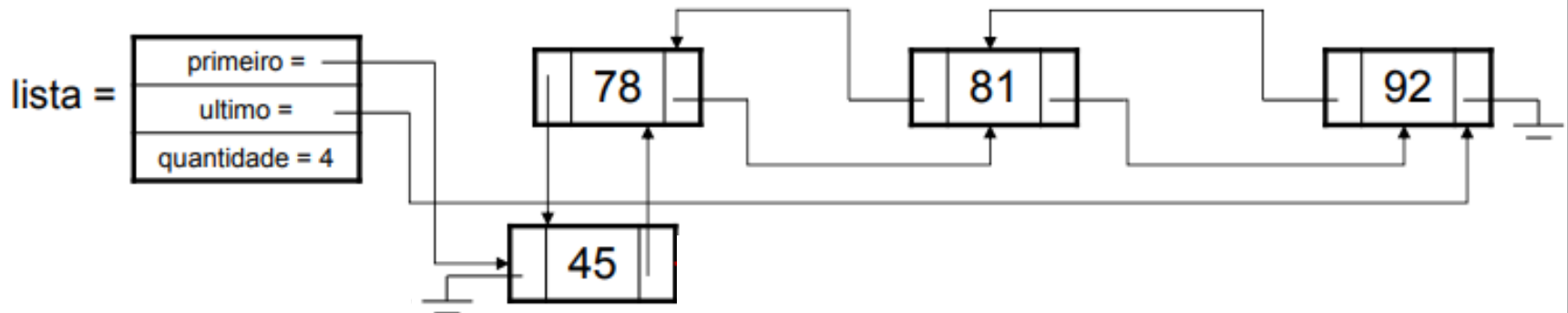
# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Inserção)
  - Observe cada situação da inserção
    - Inserir no início da Lista
      - Vazia
      - Não Vazia
    - Inserir no meio da Lista
    - Inserir no final da Lista



# Lista Dinâmica Duplamente Encadeada

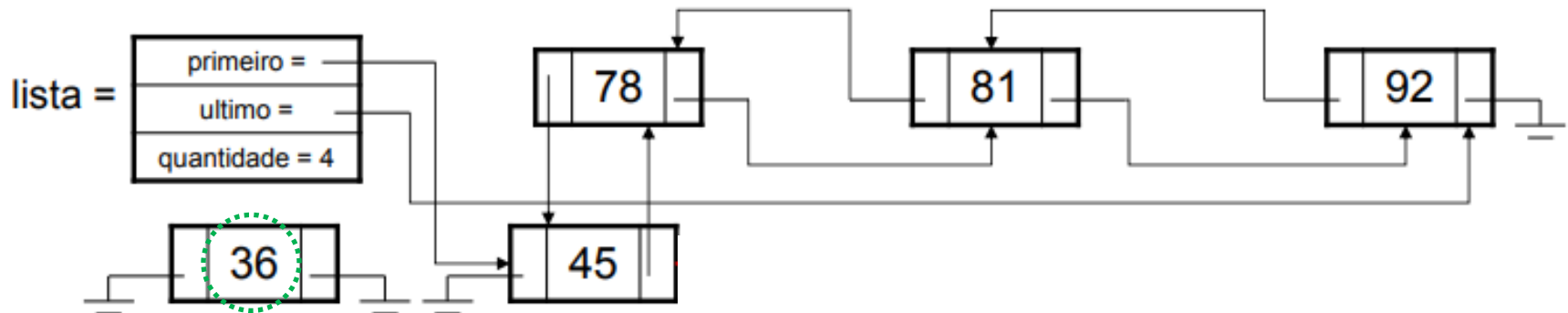
- Funcionamento (Inserção no início da lista)



- • Inserir o elemento: 36
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Se o ponteiro anterior for NULL, então
    - Atribua ao campo próximo do novo item o valor de lista no campo primeiro.
    - Atribua campo primeiro da lista o endereço do novo item.
    - Atribua ao campo anterior do atual o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

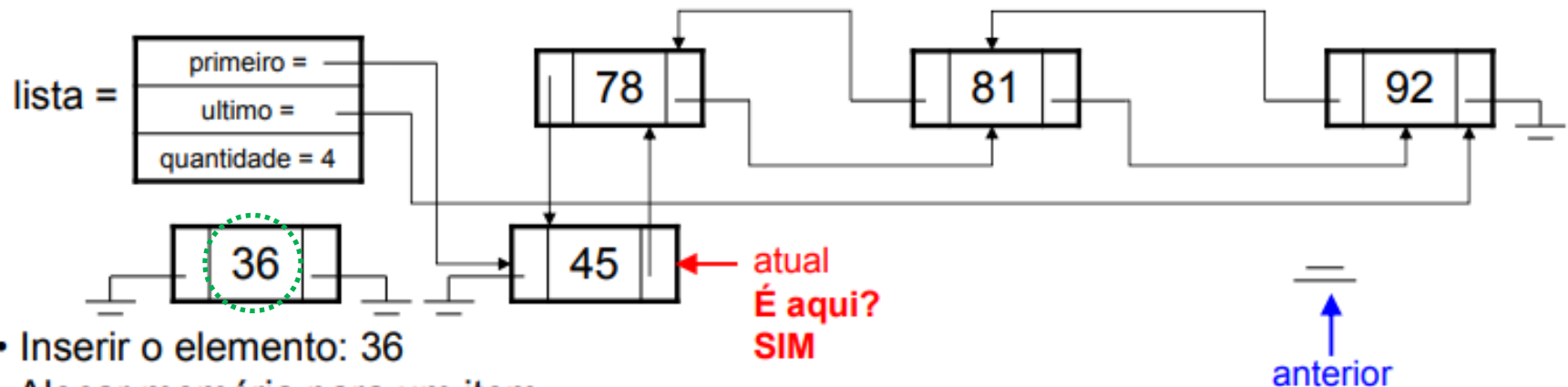
- Funcionamento (Inserção no início da lista)



- • Inserir o elemento: 36
- • Alocar memória para um item.
- • Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Se o ponteiro anterior for NULL, então
    - Atribua ao campo próximo do novo item o valor de lista no campo primeiro.
    - Atribua campo primeiro da lista o endereço do novo item.
    - Atribua ao campo anterior do atual o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

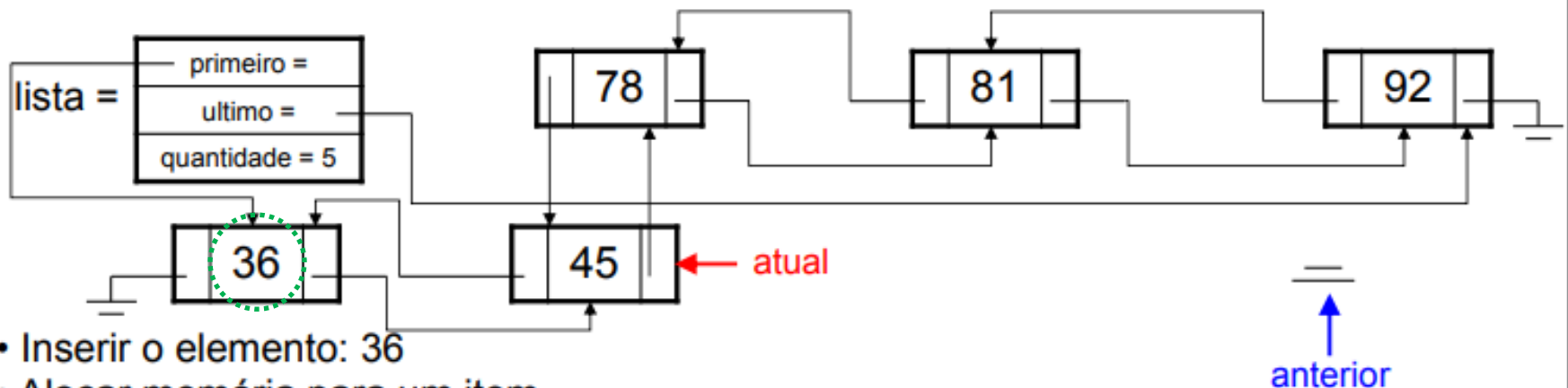
- Funcionamento (Inserção no início da lista)



- Inserir o elemento: 36
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Se o ponteiro anterior for NULL, então
    - Atribua ao campo próximo do novo item o valor de lista no campo primeiro.
    - Atribua campo primeiro da lista o endereço do novo item.
    - Atribua ao campo anterior do atual o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

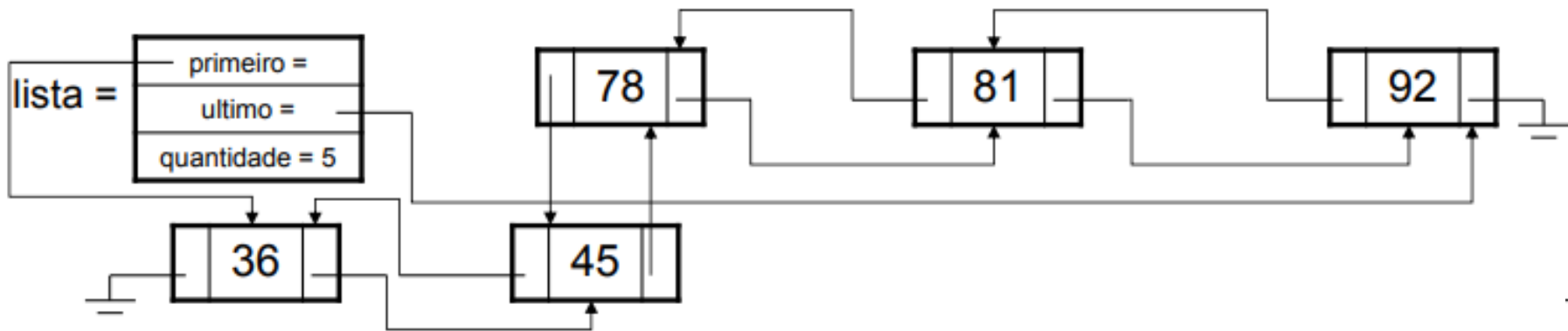
- Funcionamento (Inserção no início da lista)



- Inserir o elemento: 36
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- • Se o ponteiro anterior for NULL, então
  - • Atribua ao campo próximo do novo item o valor de lista no campo primeiro.
  - • Atribua campo primeiro da lista o endereço do novo item.
  - • Atribua ao campo anterior do atual o endereço do novo item.
- • Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Inserção no final da lista)

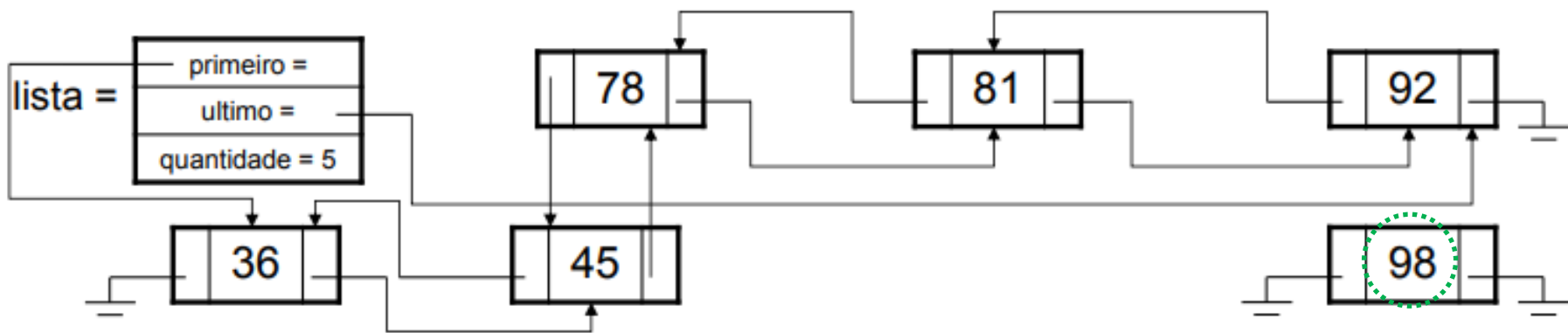


→ Inserir o elemento: 98

- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista desde o início a procura da posição correta do item ou o final da lista
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item.
  - Atribua a lista no campo último o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

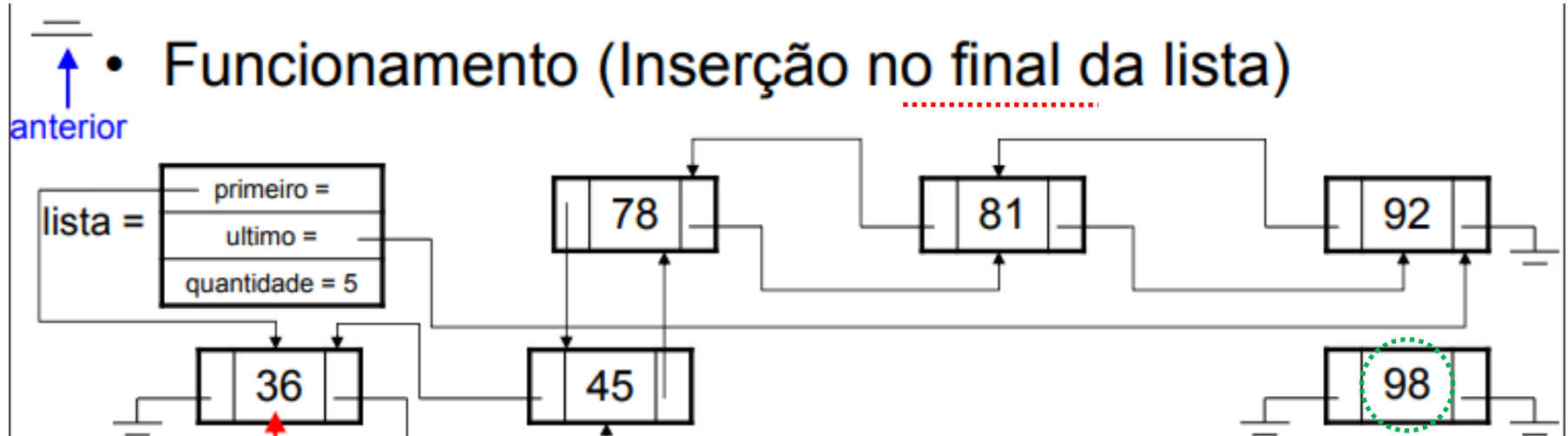
- Funcionamento (Inserção no final da lista)



- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista desde o início a procura da posição correta do item ou o final da lista
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item.
  - Atribua a lista no campo último o endereço do novo item.
  - Incremente a quantidade em uma unidade.



# Lista Dinâmica Duplamente Encadeada

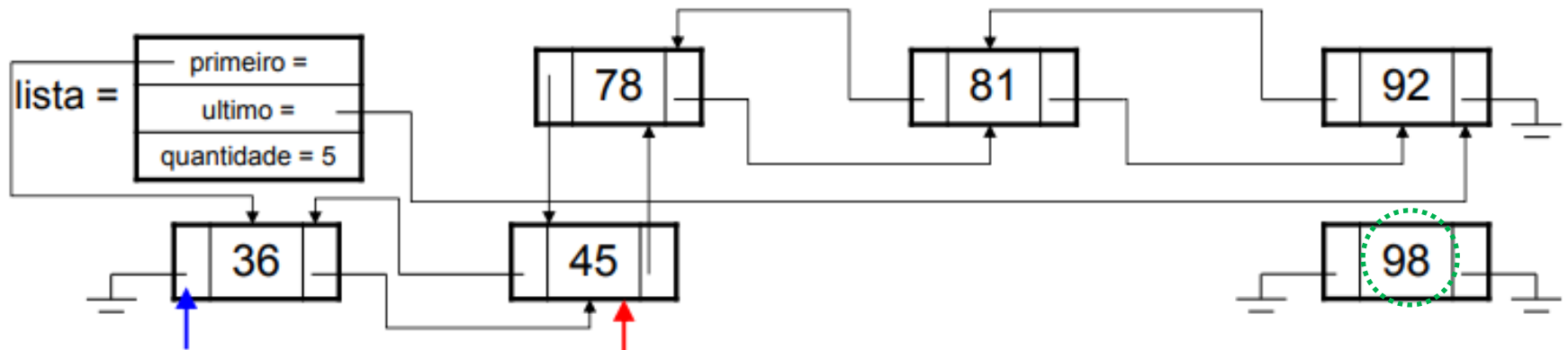


## • Funcionamento (Inserção no final da lista)

- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item.
  - Atribua a lista no campo último o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Inserção no final da lista)

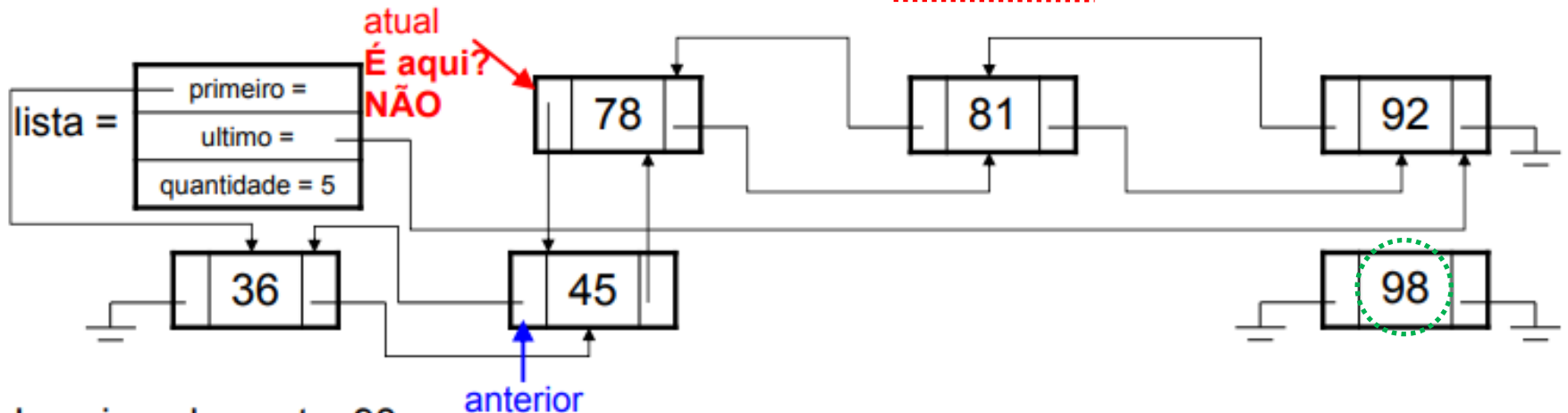


- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- • Se a lista não está vazia.
  - • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item.
  - Atribua a lista no campo último o endereço do novo item.
  - Incremente a quantidade em uma unidade.



# Lista Dinâmica Duplamente Encadeada

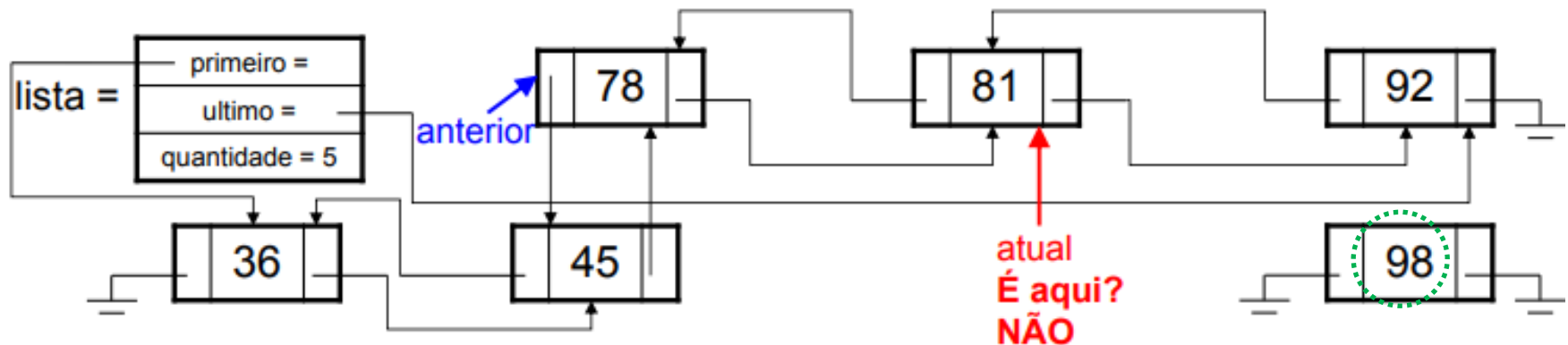
- Funcionamento (Inserção no final da lista)



- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- • Se a lista não está vazia.
  - • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item.
  - Atribua a lista no campo último o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

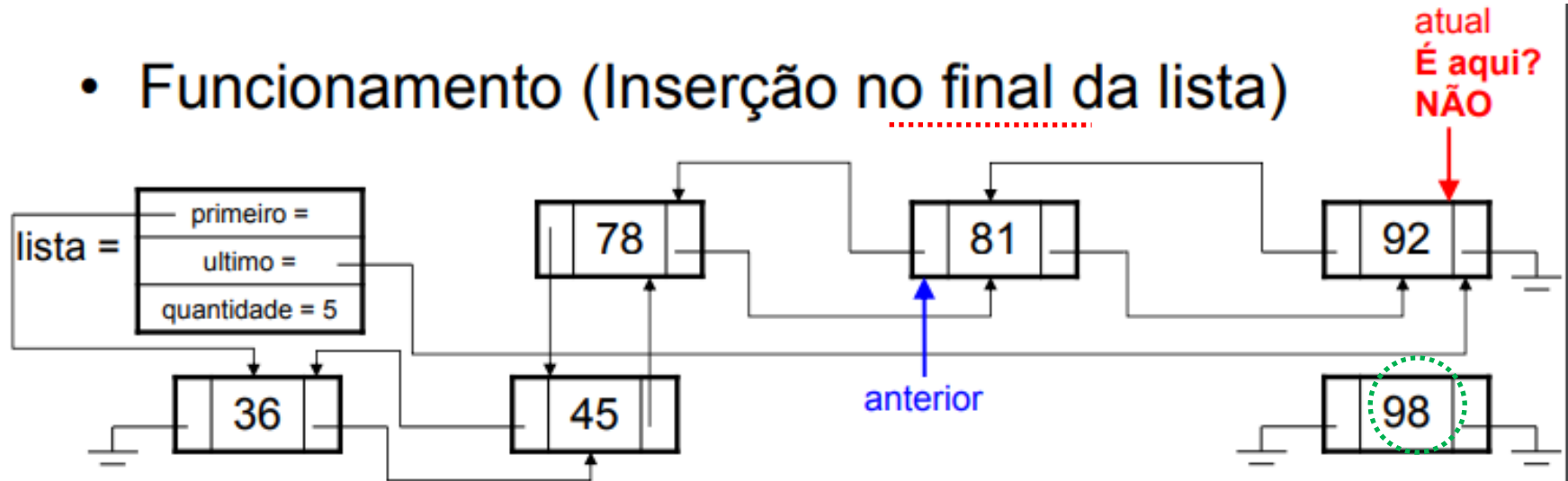
- Funcionamento (Inserção no final da lista)



- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- • Se a lista não está vazia.
  - • Percorra a lista deste o início a procura da posição correta do item ou o final da lista.
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item.
  - Atribua a lista no campo último o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

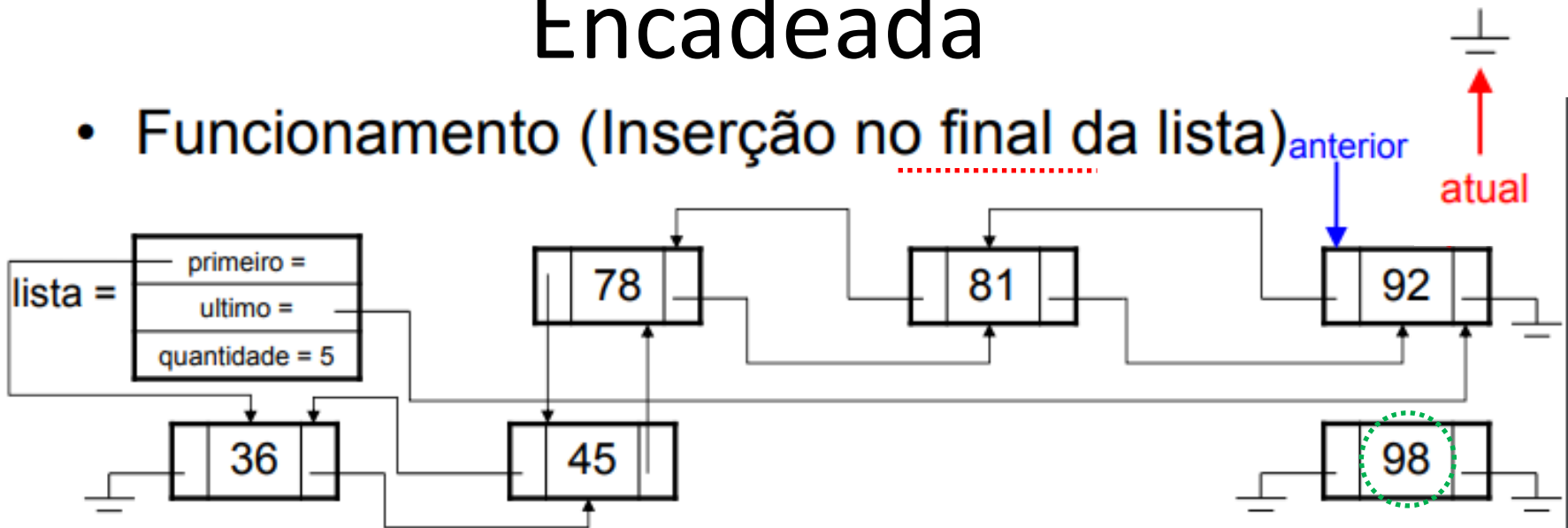
- Funcionamento (Inserção no final da lista)



- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- • Se a lista não está vazia.
  - • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item.
  - Atribua a lista no campo último o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

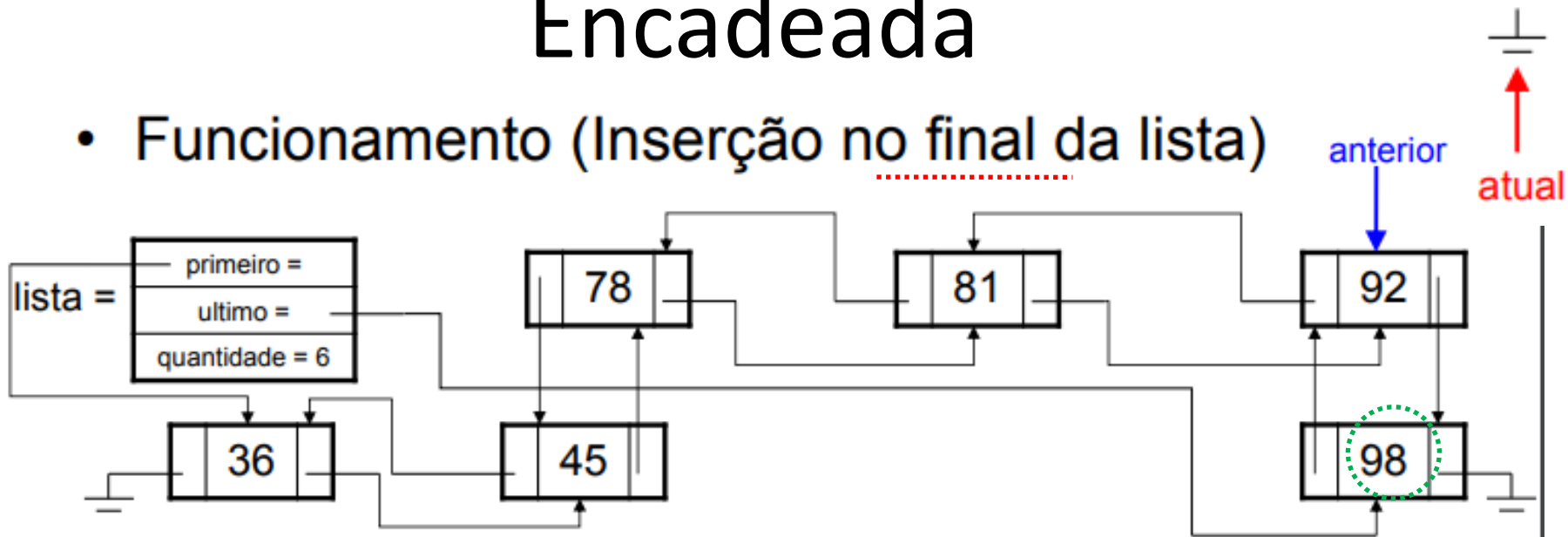
- Funcionamento (Inserção no final da lista)



- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item.
  - Atribua a lista no campo último o endereço do novo item.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

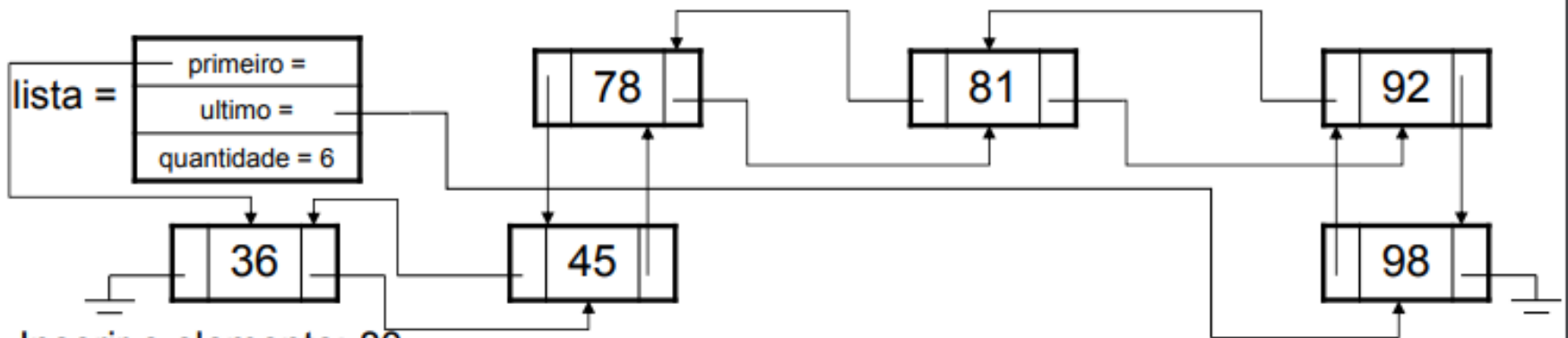
- Funcionamento (Inserção no final da lista)



- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista desde o início a procura da posição correta do item ou o final da lista
  - • Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - • Atribua ao campo próximo do elemento anterior o endereço do novo item.
  - • Atribua a lista no campo último o endereço do novo item.
  - • Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Inserção no meio da lista)

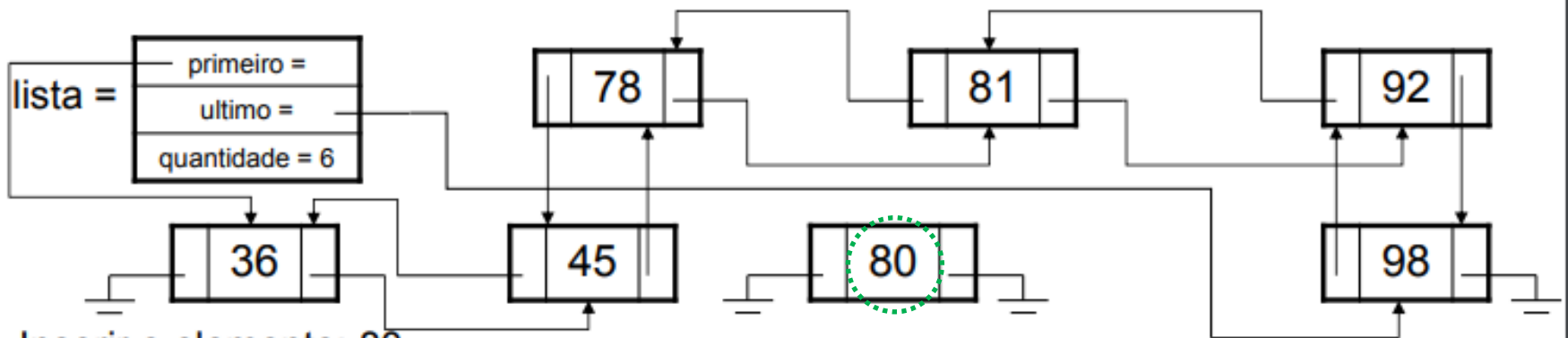


- Inserir o elemento: 80
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item
  - Atribua ao campo anterior do item atual o endereço do item novo.
  - Incremente a quantidade em uma unidade.



# Lista Dinâmica Duplamente Encadeada

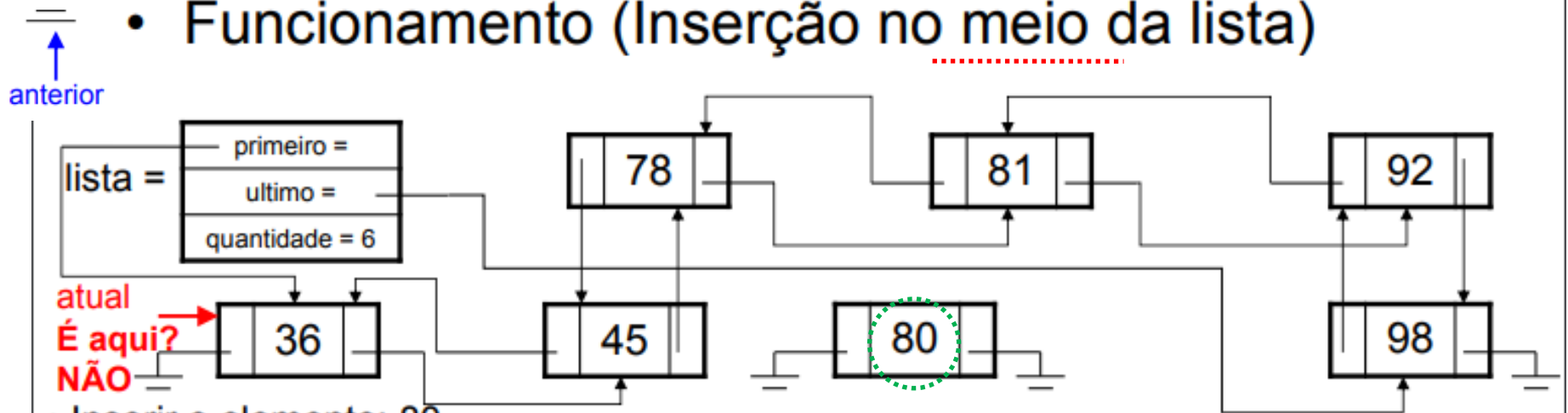
- Funcionamento (Inserção no meio da lista)



- Inserir o elemento: 80
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item
  - Atribua ao campo anterior do item atual o endereço do item novo.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

## • Funcionamento (Inserção no meio da lista)

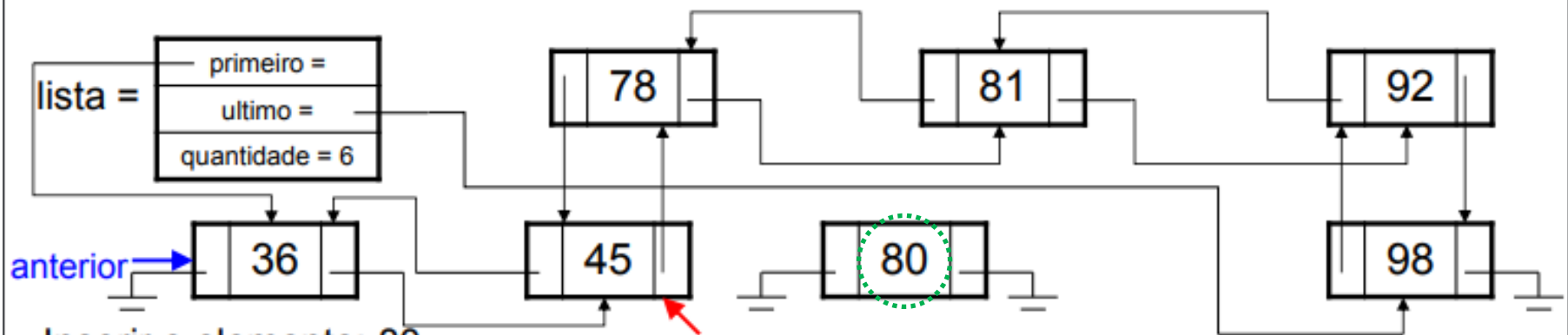


- Inserir o elemento: 80
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item
  - Atribua ao campo anterior do item atual o endereço do item novo.
  - Incremente a quantidade em uma unidade.



# Lista Dinâmica Duplamente Encadeada

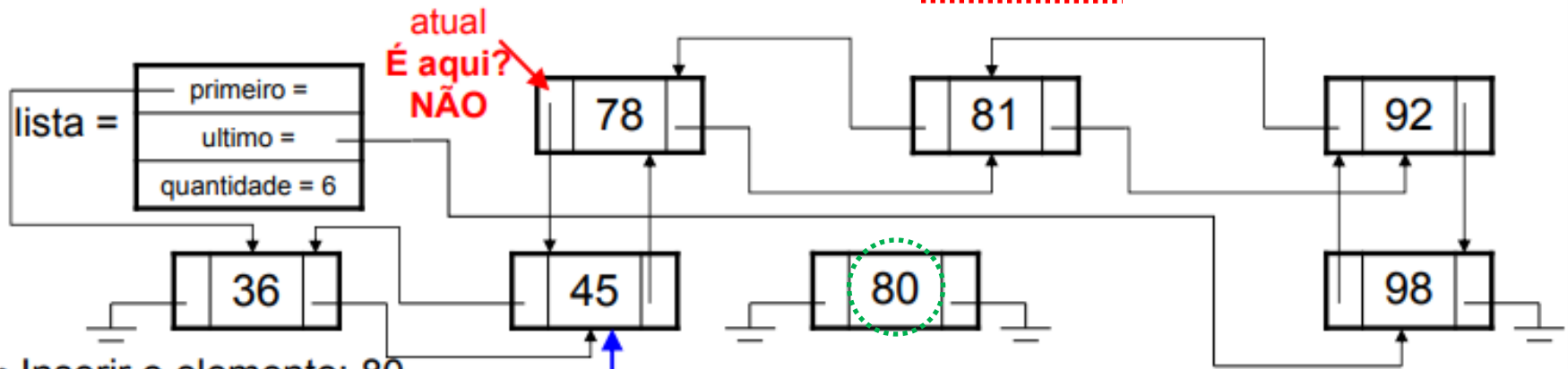
- Funcionamento (Inserção no meio da lista)



- Inserir o elemento: 80
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item
  - Atribua ao campo anterior do item atual o endereço do item novo.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

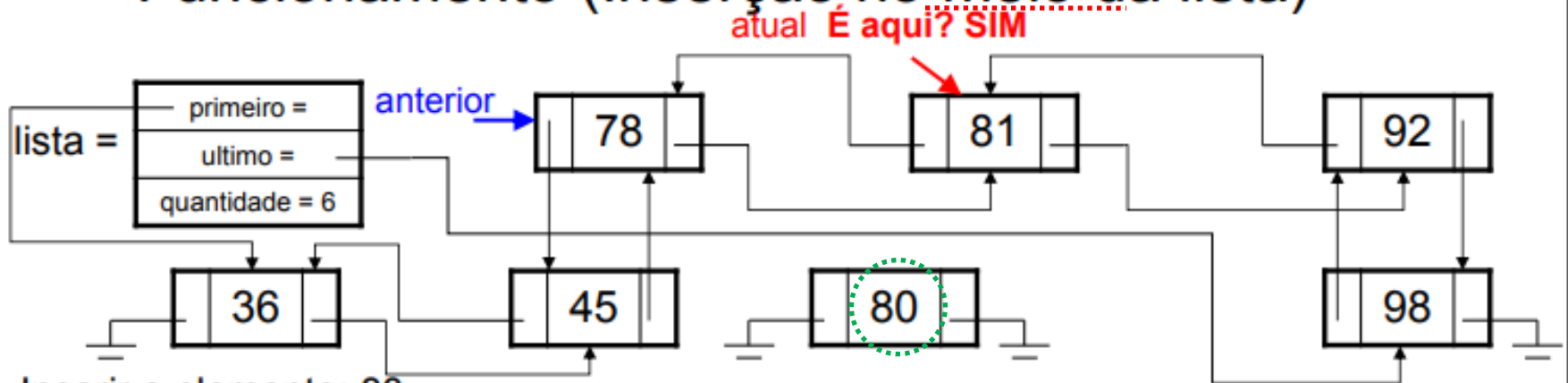
- Funcionamento (Inserção no meio da lista)



- Inserir o elemento: 80
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
  - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
  - Atribua ao campo anterior de novo item o endereço do elemento anterior.
  - Atribua ao campo próximo do elemento anterior o endereço do novo item
  - Atribua ao campo anterior do item atual o endereço do item novo.
  - Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

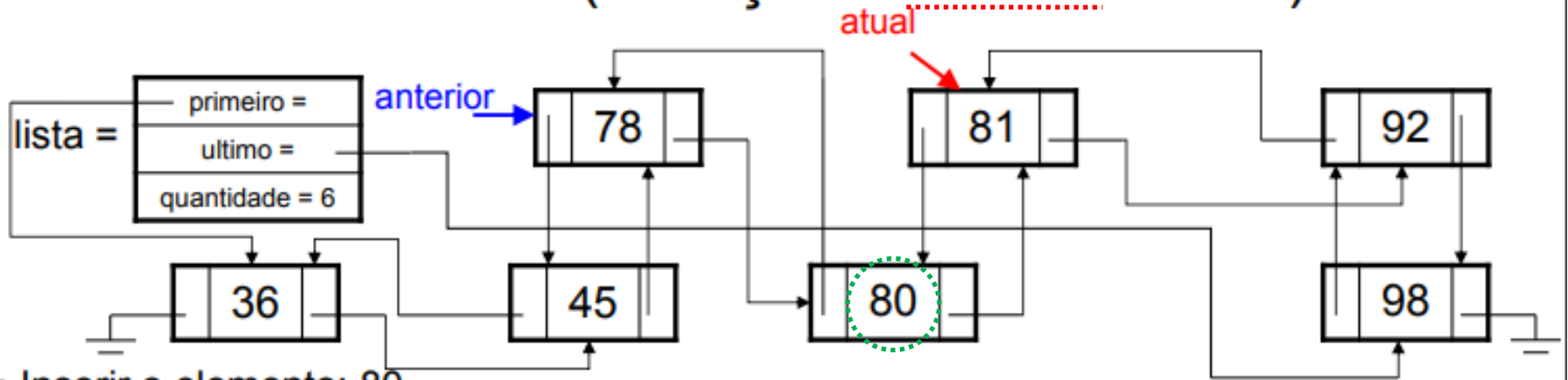
- Funcionamento (Inserção no meio da lista)



- Inserir o elemento: 80
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- • Se a lista não está vazia.
- • Percorra a lista deste o início a procura da posição correta do item ou o final da lista.
- Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
- Atribua ao campo anterior de novo item o endereço do elemento anterior.
- Atribua ao campo próximo do elemento anterior o endereço do novo item
- Atribua ao campo anterior do item atual o endereço do item novo.
- Incremente a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Inserção no meio da lista)



- Inserir o elemento: 80
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo e anterior o valor nulo.
- Se a lista não está vazia.
  - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
- Atribua ao campo anterior de novo item o endereço do elemento anterior.
- Atribua ao campo próximo do elemento anterior o endereço do novo item
- Atribua ao campo anterior do item atual o endereço do item novo.
- Incremente a quantidade em uma unidade.

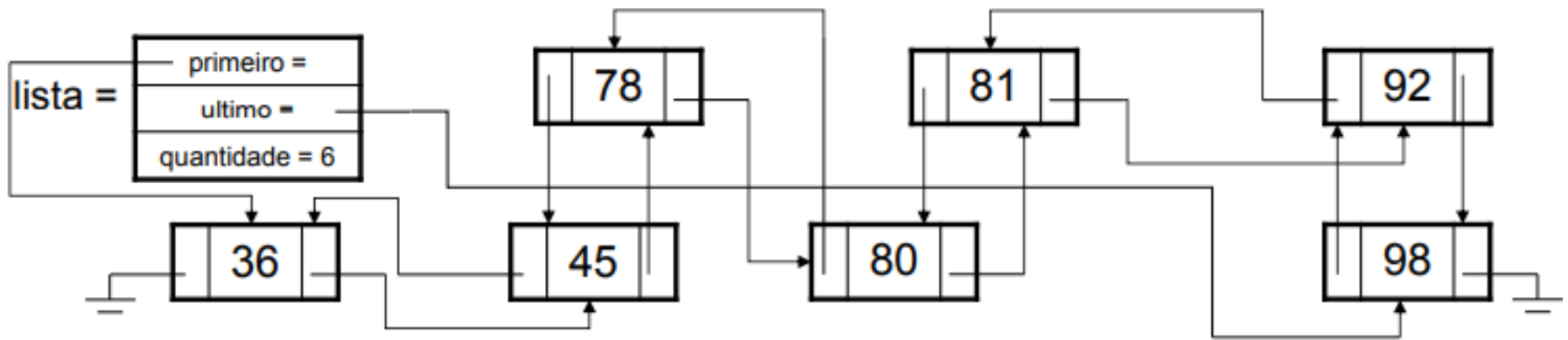
# Lista Dinâmica Duplamente Encadeada

- Inserção: (Lembrete...)
  - Caso 1: Inserir primeiro elemento em uma lista vazia:
    - Checar ponteiro para primeiro e para ultimo, se ambos forem NULL.
      - apontar ambos para o novo item.
  - Caso 2: Inserir primeiro elemento de uma lista não vazia:
    - Quando encontrar a posição de inserção, verificar se o ponteiro para anterior é NULL, caso seja verdade:
      - Campo próximo do item novo recebe o atual.
      - Campo primeiro da lista recebe o endereço do novo item.
      - Campo anterior do atual recebe o endereço do novo item.
  - Caso 3: Inserir o último elemento de uma lista:
    - Quando chegar ao final da lista (atual é NULL).
      - Campo próximo do apontado por anterior recebe o endereço do novo item.
      - Campo anterior do novo item recebe o anterior.
  - Caso 4: Inserir no meio da lista:
    - Quando encontra posição de inserção e anterior não é NULL.
      - Campo próximo do item novo recebe atual.
      - Campo anterior do atual recebe o item novo.
      - Campo próximo do anterior recebe novo item.
      - Campo anterior do novo item recebe anterior.
  - Em todos os casos incrementa-se a quantidade em uma unidade.



# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Busca)

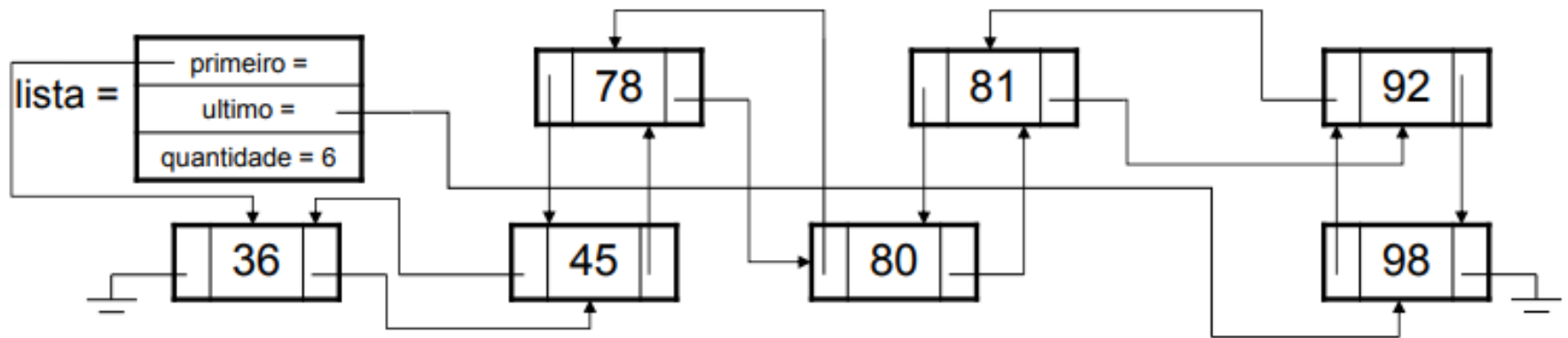


**Como faremos a busca de um item na lista ?**

- A lista está ordenada. Busca Binária ? **X**
- Busca Seqüencial (Linear) **✓**

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Busca)



## Como faremos a busca de um item na lista ?

- Comece com o apontado por lista início.
  - Verifica se é o buscado.
  - Se não for, vá para o próximo.
  - Repita até que encontre NULL

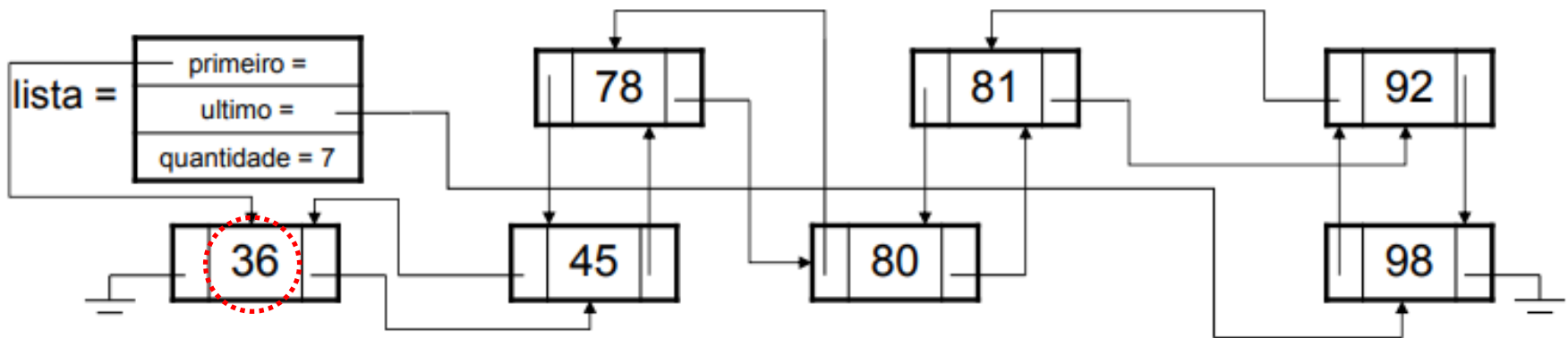


# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Remoção)
  - Observe cada situação da remoção
    - Remover do início da Lista
    - Remover do meio da Lista
    - Remover do final da Lista

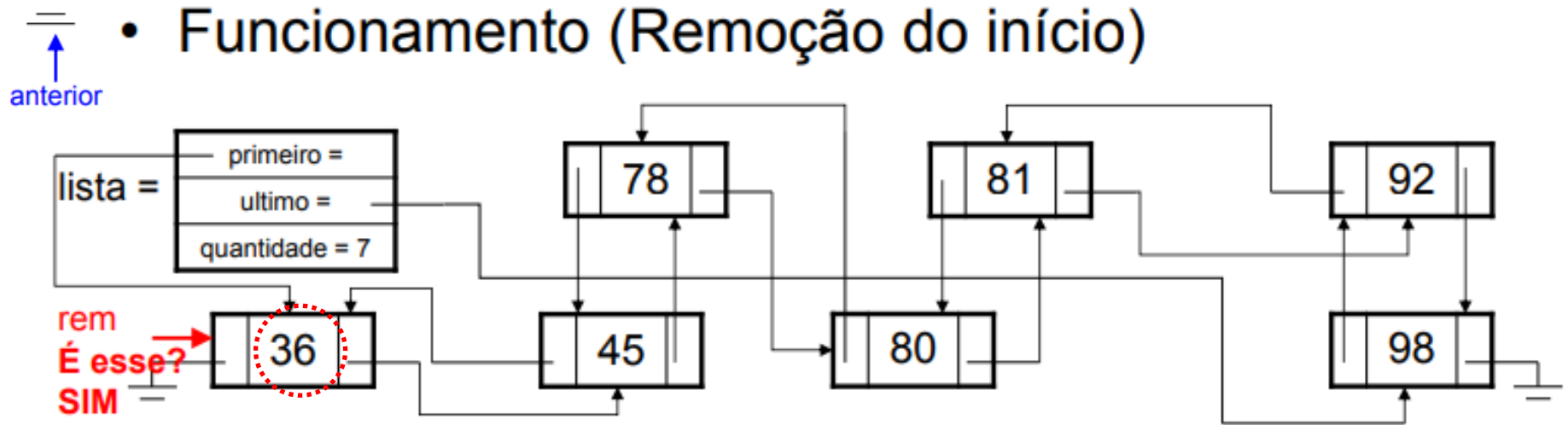
# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Remoção do início)



- Remover o item: 36
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for igual a NULL (excluindo o primeiro)
    - Campo inicio da lista recebe o campo próximo do item a ser removido.
    - Campo anterior do próximo recebe o anterior do atual.
      - (atual->proximo)->anterior = atual->anterior;
  - Desaloca a memória do item a ser removido.
  - Decrementa a quantidade em uma unidade

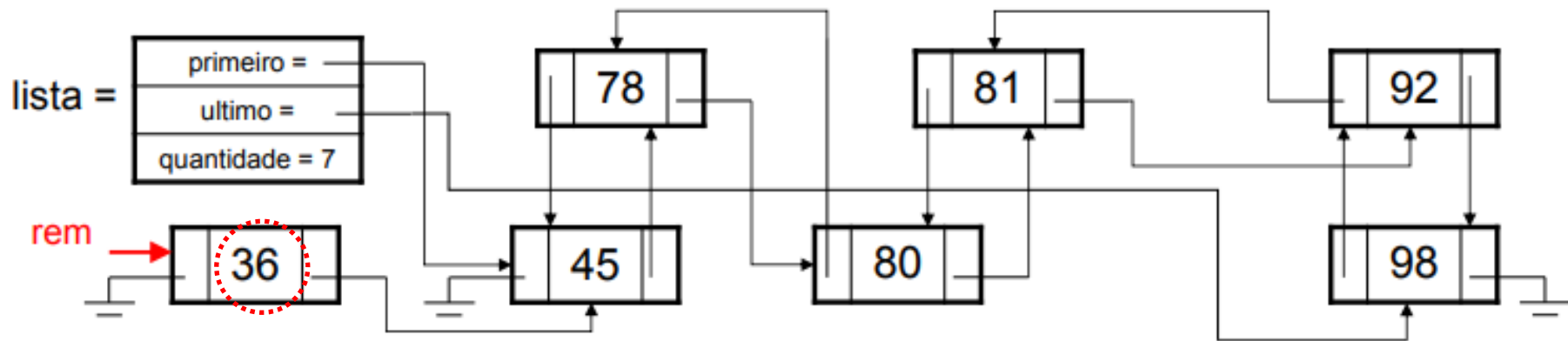
# Lista Dinâmica Duplamente Encadeada



- Remover o item: 36
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for igual a NULL (excluindo o primeiro)
    - Campo inicio da lista recebe o campo próximo do item a ser removido.
    - Campo anterior do próximo recebe o anterior do item a ser removido.
      - $(rem \rightarrow proximo) \rightarrow anterior = rem \rightarrow anterior;$
  - Desaloca a memória do item a ser removido.
  - Decrementa a quantidade em uma unidade

# Lista Dinâmica Duplamente Encadeada

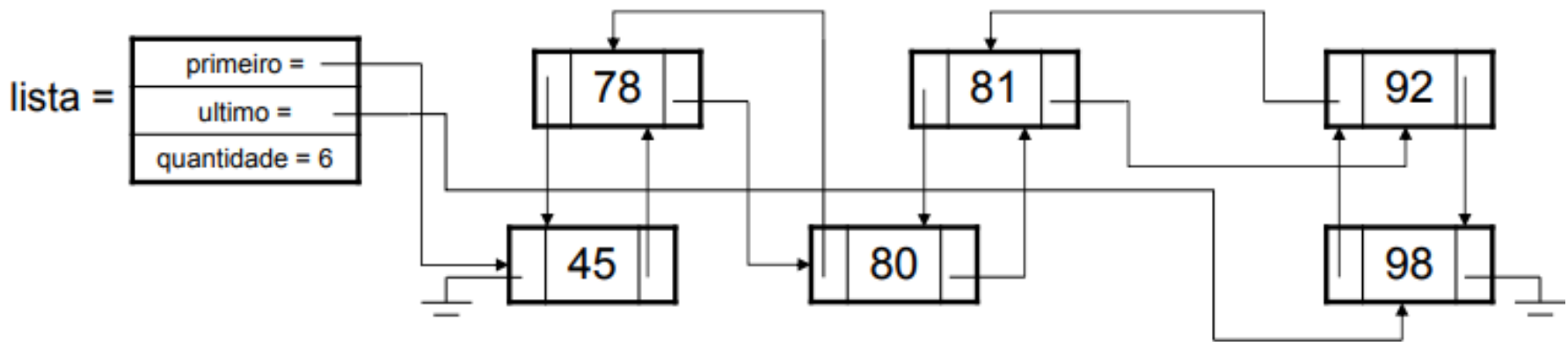
- Funcionamento (Remoção do início)



- Remover o item: 36
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for igual a NULL (excluindo o primeiro)
    - • Campo inicio da lista recebe o campo próximo do item a ser removido.
    - • Campo anterior do próximo recebe o anterior do item a ser removido.
      - $(rem \rightarrow proximo) \rightarrow anterior = rem \rightarrow anterior;$
  - Desaloca a memória do item a ser removido.
  - Decrementa a quantidade em uma unidade

# Lista Dinâmica Duplamente Encadeada

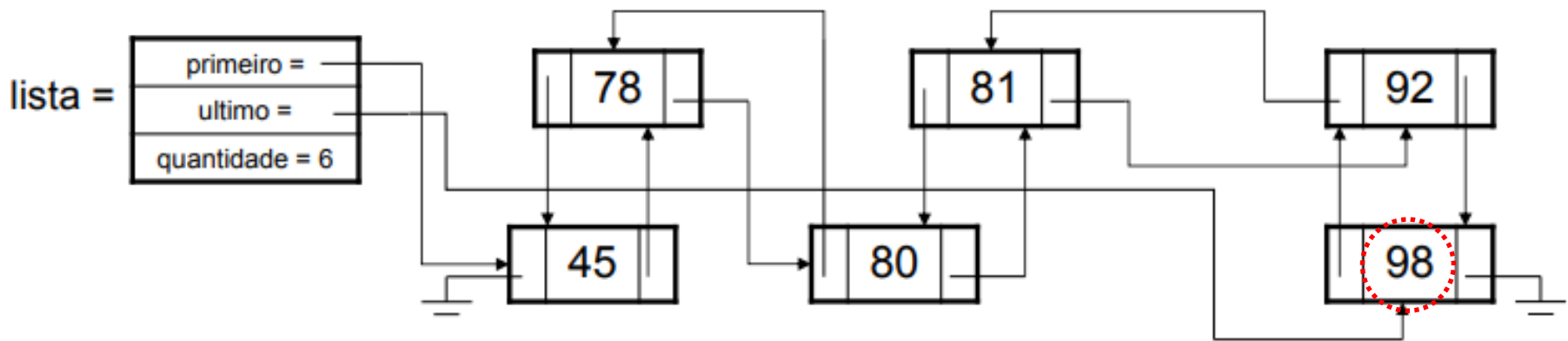
- Funcionamento (Remoção do início)



- Remover o item: 36
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for igual a NULL (excluindo o primeiro)
    - Campo inicio da lista recebe o campo próximo do item a ser removido.
    - Campo anterior do próximo recebe o anterior do item a ser removido.
      - $(rem \rightarrow proximo) \rightarrow anterior = rem \rightarrow anterior;$
- • Desaloca a memória do item a ser removido.
- • Decrementa a quantidade em uma unidade

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Remoção do final da lista)

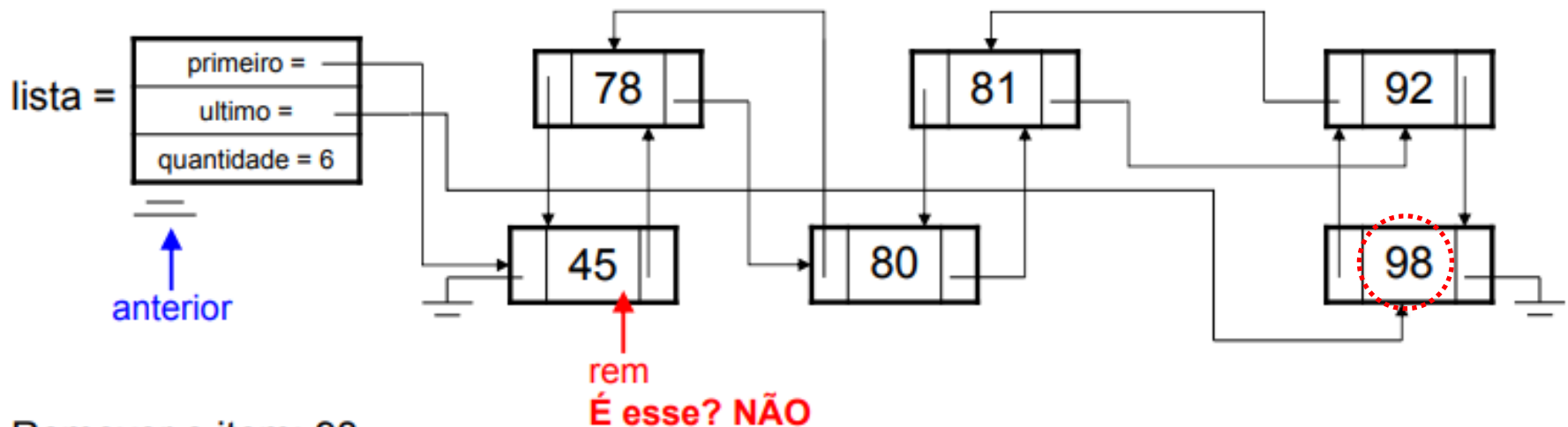


- Remover o item: 98
  - Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
  - Se encontrar o item a ser removido.
    - Se anterior for diferente de NULL
      - Campo próximo do anterior recebe o campo próximo do item a ser removido.
      - Lista no campo ultimo recebe o anterior do item a ser removido.
  - Desaloca a memória do item a ser removido.
  - Decrementa a quantidade em uma unidade



# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Remoção do final da lista)

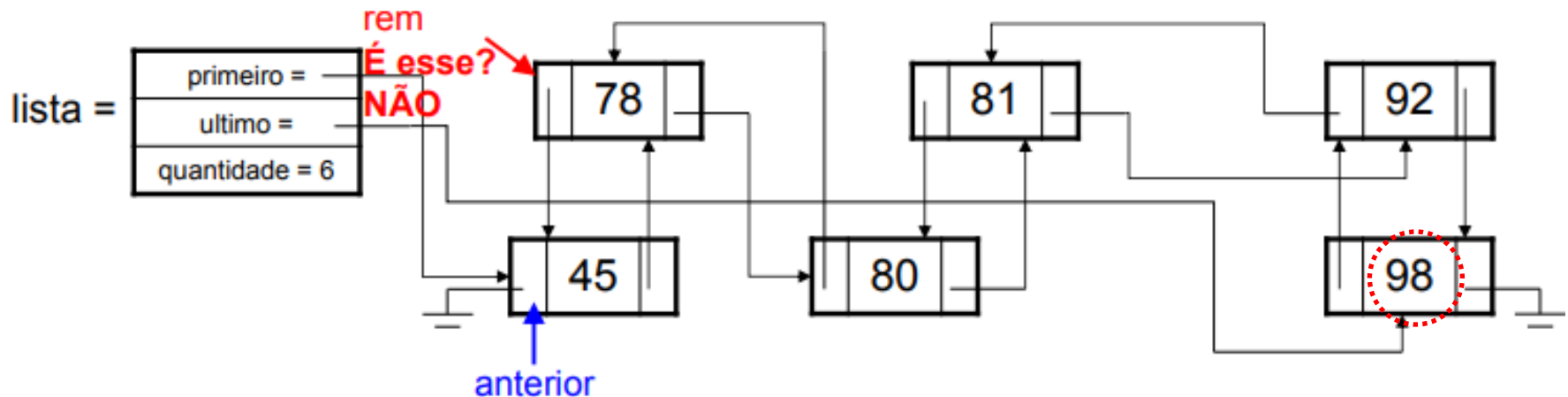


- Remover o item: 98
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for diferente de NULL
    - Campo próximo do anterior recebe o campo próximo do item a ser removido.
    - Lista no campo ultimo recebe o anterior do item a ser removido.
  - Desaloca a memória do item a ser removido.
  - Decrementa a quantidade em uma unidade



# Lista Dinâmica Duplamente Encadeada

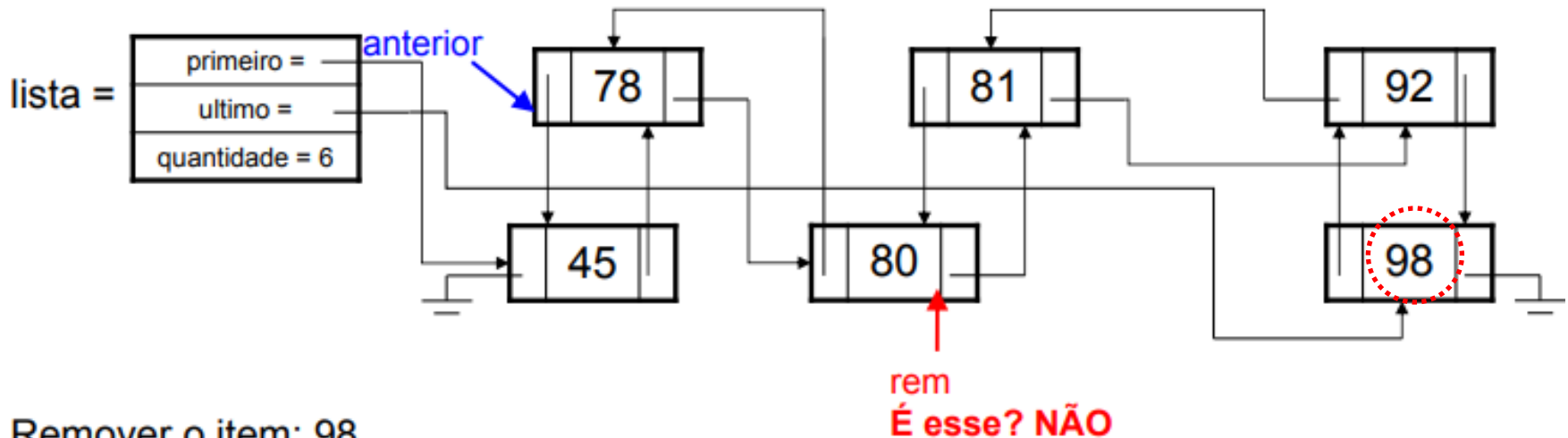
- Funcionamento (Remoção do final da lista)



- • Remover o item: 98
- • Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for diferente de NULL
    - Campo próximo do anterior recebe o campo próximo do item a ser removido.
    - Lista no campo ultimo recebe o anterior do item a ser removido.
  - Desaloca a memória do item a ser removido.
  - Decrementa a quantidade em uma unidade

# Lista Dinâmica Duplamente Encadeada

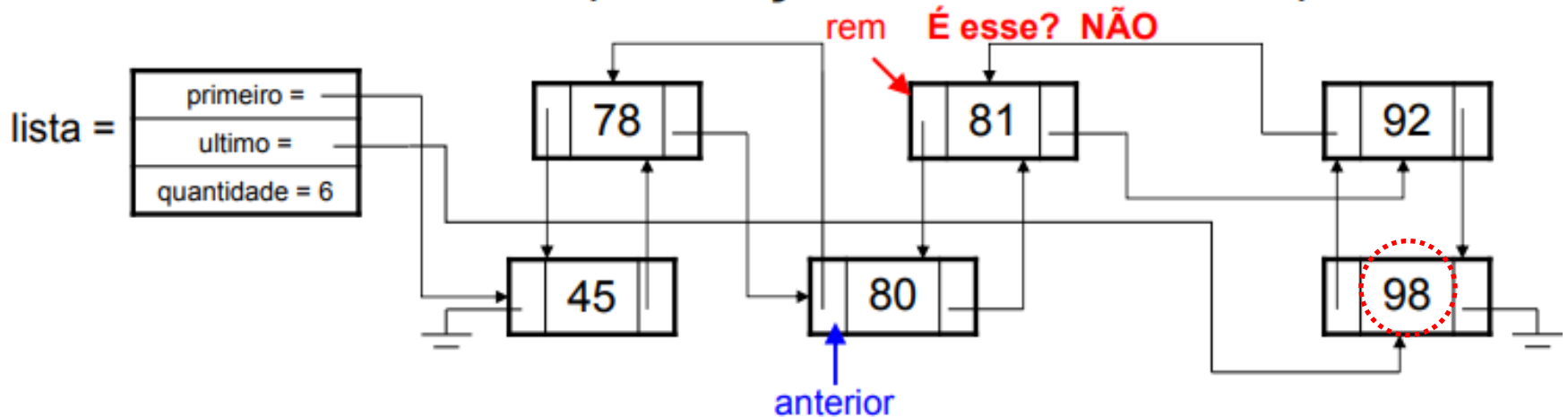
- Funcionamento (Remoção do final da lista)



- • Remover o item: 98
- • Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for diferente de NULL
    - Campo próximo do anterior recebe o campo próximo do item a ser removido.
    - Lista no campo ultimo recebe o anterior do item a ser removido.
  - Desaloca a memória do item a ser removido.
  - Decrementa a quantidade em uma unidade

# Lista Dinâmica Duplamente Encadeada

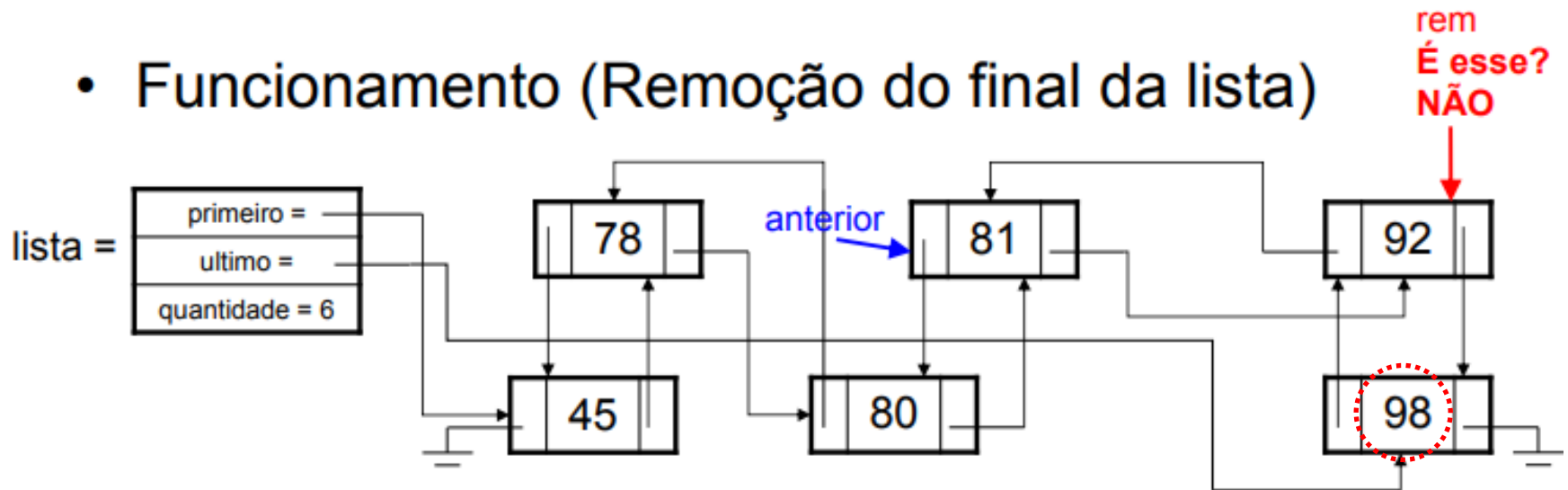
- Funcionamento (Remoção do final da lista)



- • Remover o item: 98
- • Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
  - Se encontrar o item a ser removido.
    - Se anterior for diferente de NULL
      - Campo próximo do anterior recebe o campo próximo do item a ser removido.
      - Lista no campo ultimo recebe o anterior do item a ser removido.
    - Desaloca a memória do item a ser removido.
    - Decrementa a quantidade em uma unidade

# Lista Dinâmica Duplamente Encadeada

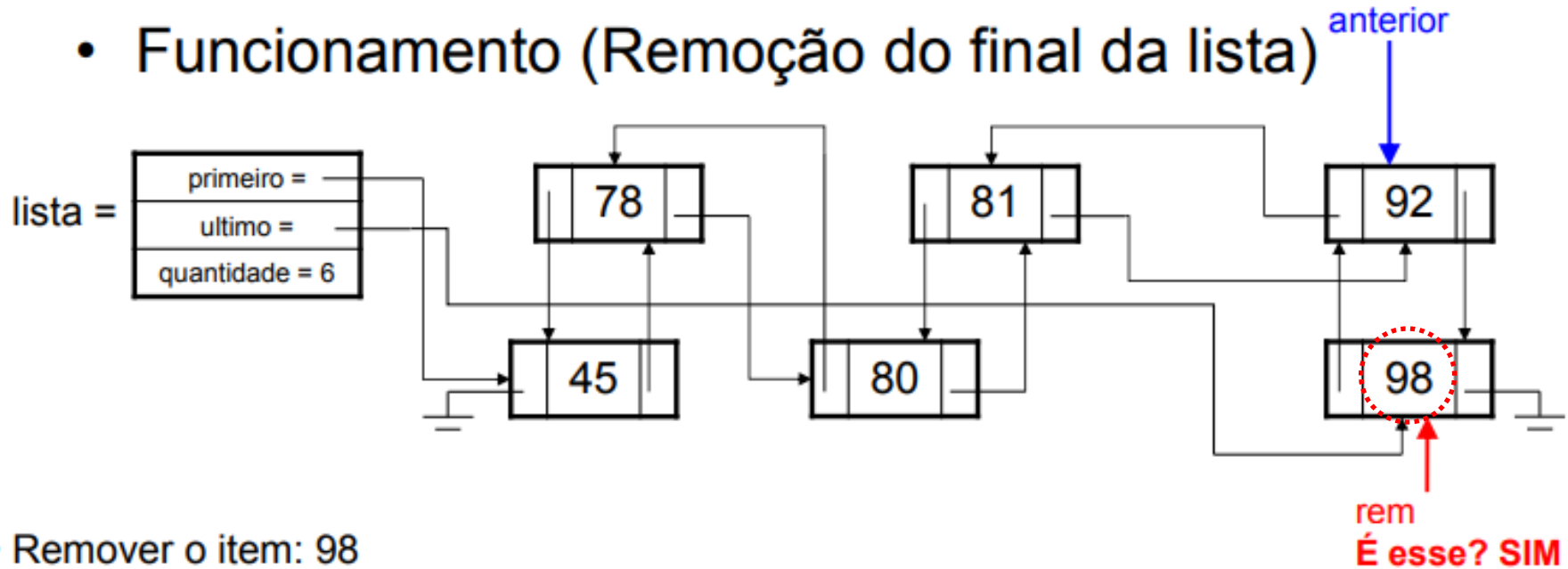
- Funcionamento (Remoção do final da lista)



- Remover o item: 98
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for diferente de NULL
    - Campo próximo do anterior recebe o campo próximo do item a ser removido.
    - Lista no campo ultimo recebe o anterior do item a ser removido.
  - Desaloca a memória do item a ser removido.
  - Decrementa a quantidade em uma unidade

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Remoção do final da lista)

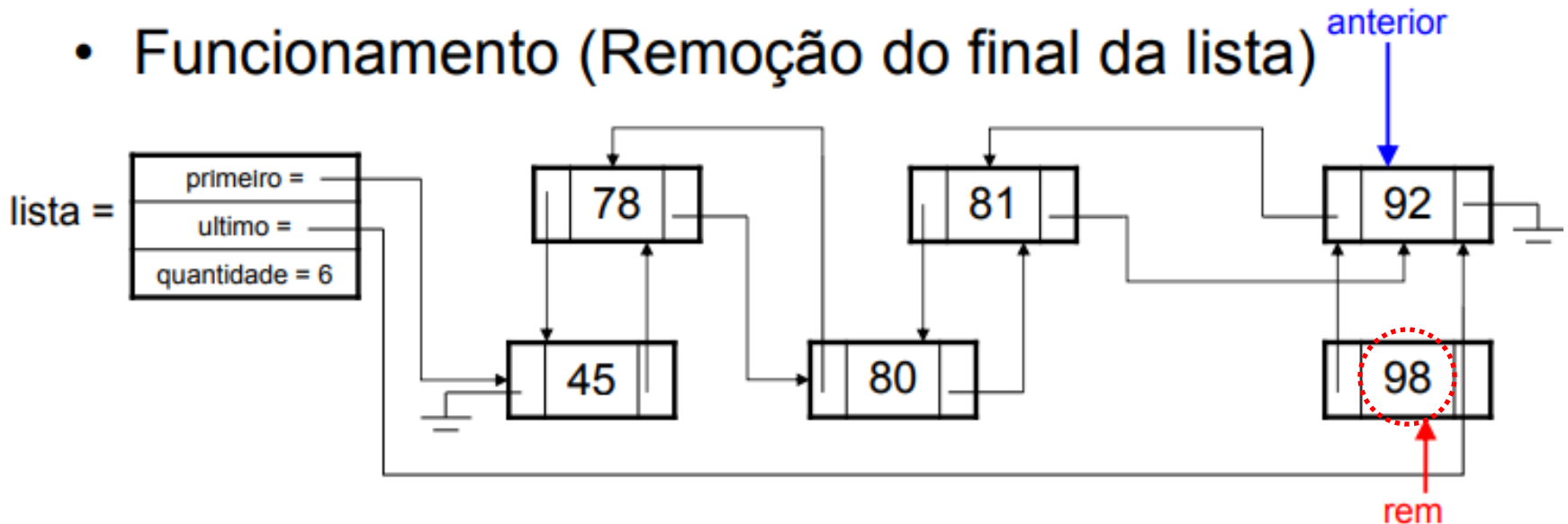


rem  
É esse? SIM

- Remover o item: 98
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for diferente de NULL
    - Campo próximo do anterior recebe o campo próximo do item a ser removido.
    - Lista no campo ultimo recebe o anterior do item a ser removido.
  - Desaloca a memória do item a ser removido.
  - Decrementa a quantidade em uma unidade

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Remoção do final da lista)

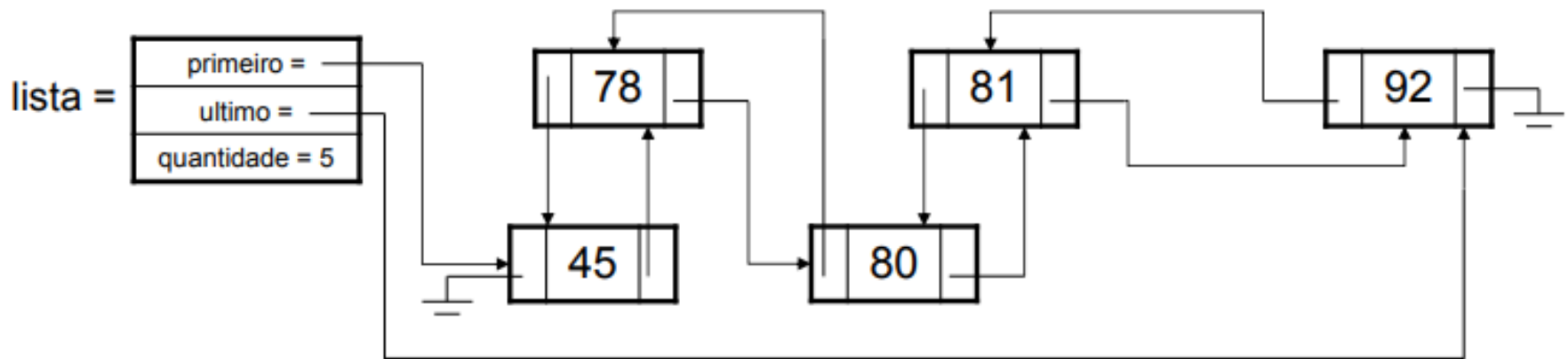


- Remover o item: 98
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- • Se encontrar o item a ser removido.
- • Se anterior for diferente de NULL
  - • Campo próximo do anterior recebe o campo próximo do item a ser removido.
  - • Lista no campo ultimo recebe o anterior do item a ser removido.
- Desaloca a memória do item a ser removido.
- Decrementa a quantidade em uma unidade



# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Remoção do final da lista)

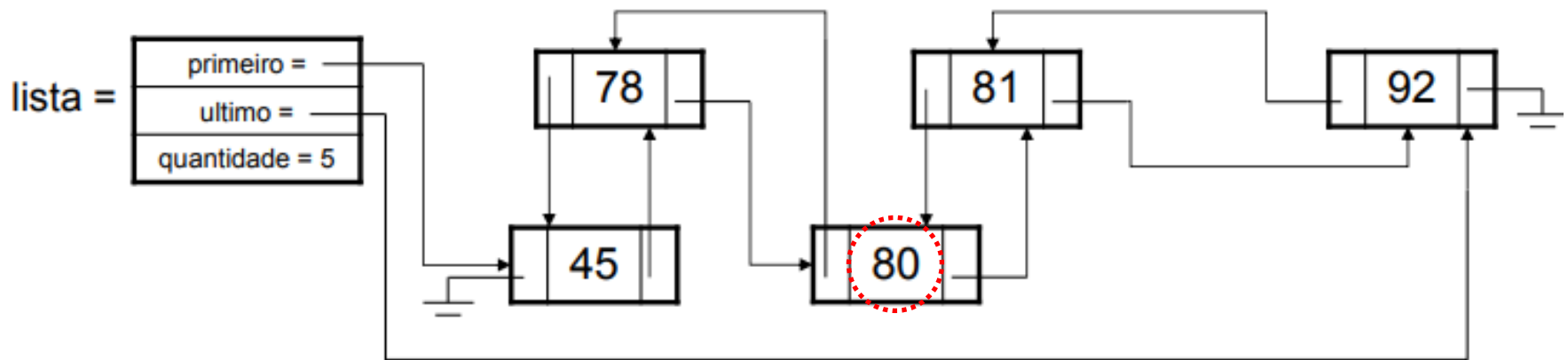


- Remover o item: 98
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for diferente de NULL
    - Campo próximo do anterior recebe o campo próximo do item a ser removido.
    - Lista no campo ultimo recebe o anterior do item a ser removido.
- • Desaloca a memória do item a ser removido.
- • Decrementa a quantidade em uma unidade



# Lista Dinâmica Duplamente Encadeada

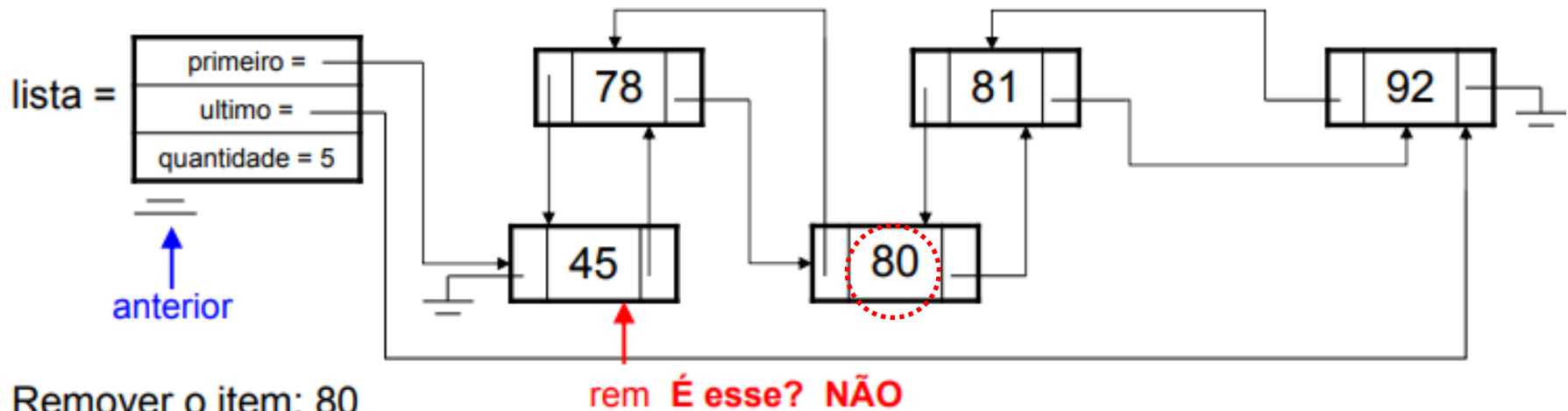
- Funcionamento (Remoção do meio da lista)



- • Remover o item: 80
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for diferente de NULL
    - Campo próximo do anterior recebe o campo próximo do item a ser removido.
    - Campo anterior do próximo recebe o anterior do item a ser removido.
      - $(rem \rightarrow proximo) \rightarrow anterior = rem \rightarrow anterior;$
  - Desaloca a memória do item a ser removido.
  - Decrementa a quantidade em uma unidade

# Lista Dinâmica Duplamente Encadeada

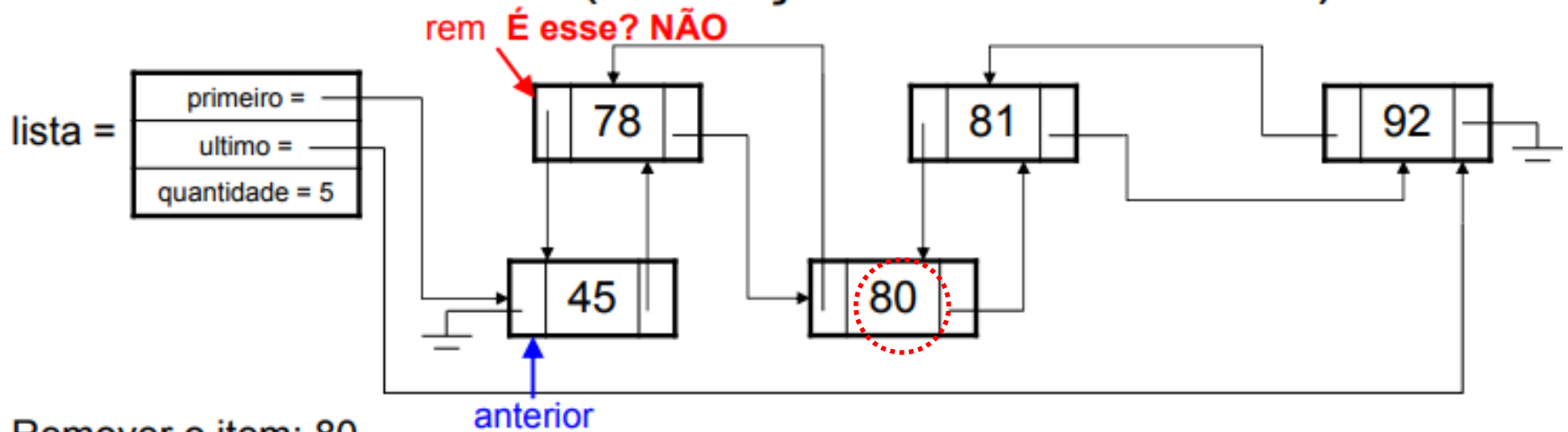
- Funcionamento (Remoção do meio da lista)



- Remover o item: 80
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
  - Se encontrar o item a ser removido.
    - Se anterior for diferente de NULL
      - Campo próximo do anterior recebe o campo próximo do item a ser removido.
      - Campo anterior do próximo recebe o anterior do item a ser removido.
        - (rem->proximo)->anterior = rem->anterior;
    - Desaloca a memória do item a ser removido.
    - Decrementa a quantidade em uma unidade

# Lista Dinâmica Duplamente Encadeada

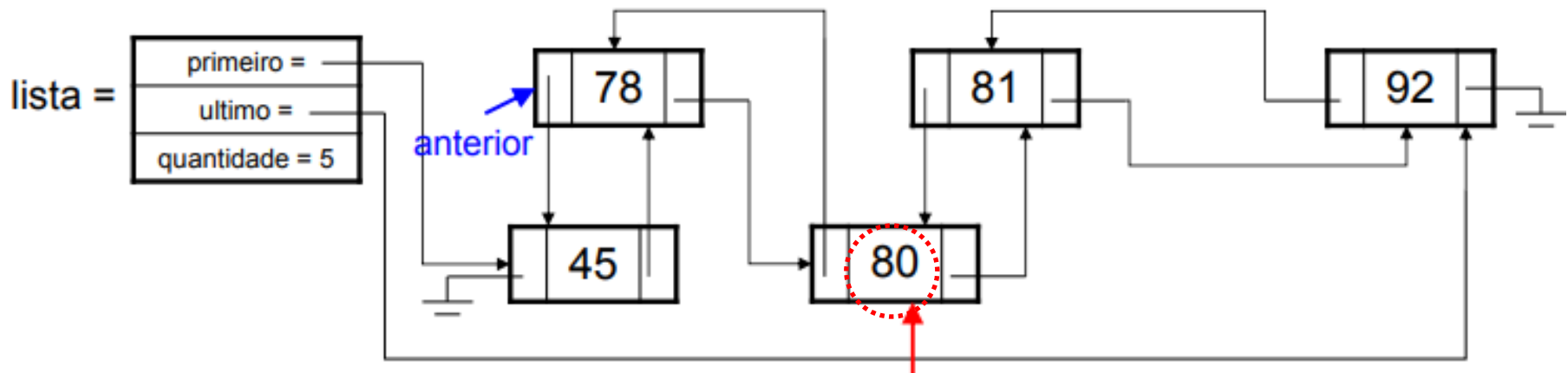
- Funcionamento (Remoção do meio da lista)



- ➔ Remover o item: 80
- ➔ Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for diferente de NULL
    - Campo próximo do anterior recebe o campo próximo do item a ser removido.
    - Campo anterior do próximo recebe o anterior do item a ser removido.
      - $(rem \rightarrow proximo) \rightarrow anterior = rem \rightarrow anterior;$
  - Desaloca a memória do item a ser removido.
  - Decrementa a quantidade em uma unidade

# Lista Dinâmica Duplamente Encadeada

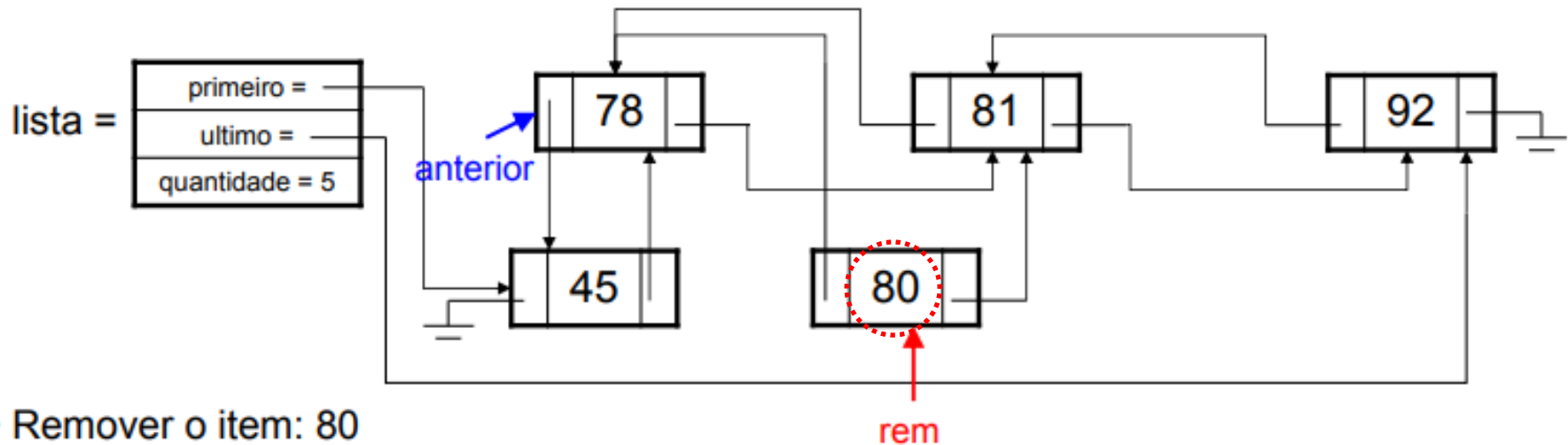
- Funcionamento (Remoção do meio da lista)



- Remover o item: 80
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for diferente de NULL
    - Campo próximo do anterior recebe o campo próximo do item a ser removido.
    - Campo anterior do próximo recebe o anterior do item a ser removido.
      - $(rem \rightarrow proximo) \rightarrow anterior = rem \rightarrow anterior;$
  - Desaloca a memória do item a ser removido.
  - Decrementa a quantidade em uma unidade

# Lista Dinâmica Duplamente Encadeada

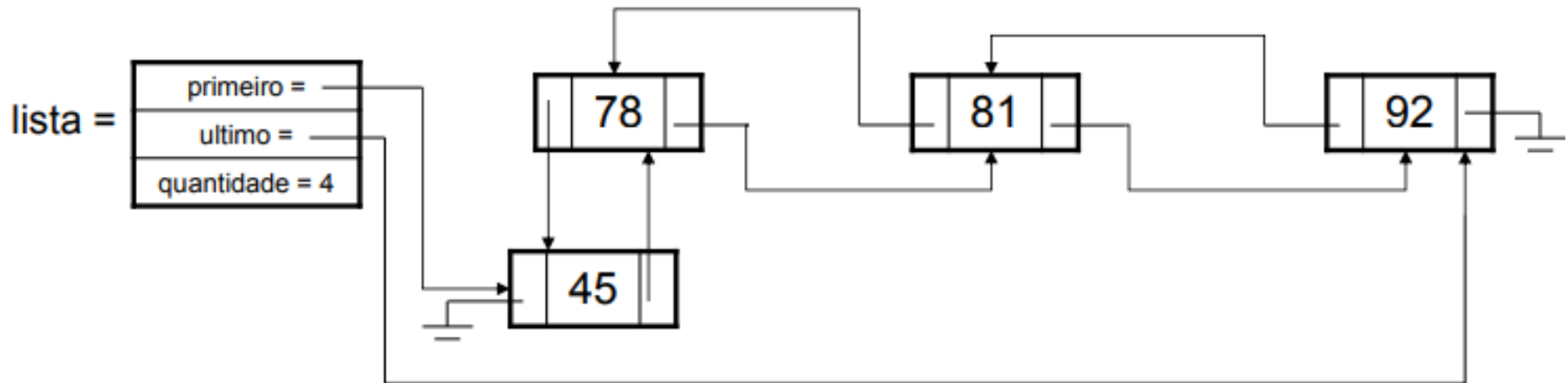
- Funcionamento (Remoção do meio da lista)



- Remover o item: 80
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- • Se encontrar o item a ser removido.
- • Se anterior for diferente de NULL
  - • Campo próximo do anterior recebe o campo próximo do item a ser removido.
  - • Campo anterior do próximo recebe o anterior do item a ser removido.
    - $(rem \rightarrow proximo) \rightarrow anterior = rem \rightarrow anterior;$
- Desaloca a memória do item a ser removido.
- Decrementa a quantidade em uma unidade

# Lista Dinâmica Duplamente Encadeada

- Funcionamento (Remoção do meio da lista)



- Remover o item: 80
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
  - Se anterior for diferente de NULL
    - Campo próximo do anterior recebe o campo próximo do item a ser removido.
    - Campo anterior do próximo recebe o anterior do item a ser removido.
      - $(rem \rightarrow proximo) \rightarrow anterior = rem \rightarrow anterior;$
- • Desaloca a memória do item a ser removido.
- • Decrementa a quantidade em uma unidade



# Lista Dinâmica Duplamente Encadeada

- Remoção: (Lembretes...)
  - Caso 1: Remover primeiro elemento de uma lista:
    - Quando encontrar o item a ser removido, verificar se o ponteiro para anterior é NULL, caso seja verdade:
      - Campo anterior do próximo do item a ser removido, recebe o anterior do item a ser removido.
      - Campo primeiro da lista recebe o campo próximo do item a ser removido.
  - Caso 2: Remover o último elemento de uma lista:
    - Quando o campo próximo do item a ser removido for igual ao campo último da lista:
      - Campo próximo do anterior, recebe campo próximo do item a ser removido.
      - Campo último da lista, recebe campo anterior do item a ser removido.
  - Caso 3: Remover no meio da lista:
    - Quando encontrar o item a ser removido e o ponteiro para anterior não for NULL e o campo próximo do item a ser removido não é igual ao campo último da lista.
      - Campo anterior do próximo do item a ser removido, recebe o anterior do item a ser removido.
      - Campo próximo do anterior, recebe campo próximo do item a ser removido.
  - Em todos os casos decrementa-se a quantidade em uma unidade.

# Lista Dinâmica Duplamente Encadeada

- Estruturas:
  - tLista
  - tItem

# Lista Dinâmica Duplamente Encadeada

- Funções:

- `struct tLista * criaLista (void);`
- `struct tItem * criaItem (void);`
- `void lerItem (struct tItem *);`
- `void insere (struct tLista *, struct tItem *);`
- `struct tItem * remover (struct tLista *);`
- `struct tItem * busca (struct tLista *)`

# Lista Dinâmica Duplamente Encadeada

- Implementação da Estrutura da Lista

```
1. struct tLista
2. {
3.     int quantidade;
4.     struct tItem *primeiro, *ultimo;
5. };
```

- Inicialização da Lista

```
1. struct tLista * criaLista(void)
2. {
3.     struct tLista * lista = (struct tLista*) malloc(sizeof(struct tLista));
4.     if (lista != NULL)
5.     {
6.         lista->primeiro = NULL;
7.         lista->ultimo = NULL;    lista->quantidade = 0;
8.     }
9.     return lista;
10. }
```

# Lista Dinâmica Duplamente Encadeada

- Implementação da Estrutura dos itens

```
1. struct tItem
2. {
3.     int chave;
4.     struct tItem *proximo, *anterior;
5. };
```

Por simplificação do exemplo nossos itens terão apenas um valor inteiro que será a chave, aqui pode-se acrescentar outros campos para a estrutura.

- Inicialização do item

```
1. struct tItem * criaItem (int x)
2. {
3.     struct tItem * it = (struct tItem*) malloc(sizeof(struct tItem));
4.     if (it != NULL) {
5.         it->anterior = NULL;
6.         it->proximo = NULL;
7.         it->chave = x;
8.     }
9.     return it;
10. }
```