

LISTA ESTÁTICA

Prof. Muriel Mazzetto
Estrutura de Dados

Lista Estática

2

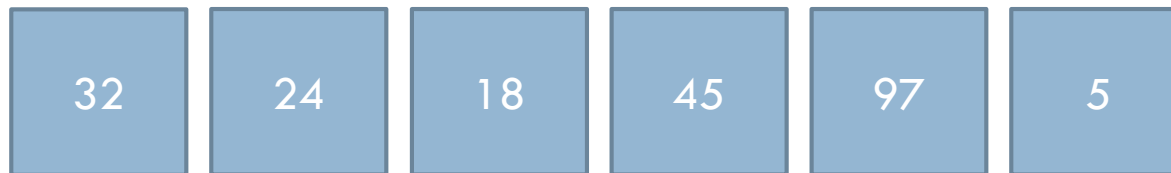
- Essa estrutura possui diferentes nomes na literatura:
 - ▣ Lista Estática.
 - ▣ Lista Linear.
 - ▣ Lista Sequencial.
- Definição: Uma estrutura do tipo “Lista” é uma sequência de elementos do mesmo tipo.
- Usam-se TADs para abstrair a sua estrutura interna.

Lista Estática

3

- Uma lista pode ser descrita como:
 - ▣ Vazia.
 - ▣ $(a_1, a_2, a_3, \dots, a_N)$.
- Exemplos:
 - ▣ Listas de contatos, de compras, etc.

LISTA



Lista Estática

4

- Características da Lista **Estática**:
 - ▣ Sequencial: o sucessor de um elemento ocupa uma posição física consecutiva na memória (vetores).
 - ▣ Espaço de memória alocado no momento da compilação.
 - ▣ Definir tamanho máximo que a lista possuirá.
 - ▣ Possuir campo chave, que diferencia os elementos (CPF, RA, Identificação).

Lista Estática

5

- Operações comuns de uma Lista:
 - ▣ Criação da lista (alocar a estrutura);
 - ▣ Verificação se a lista está vazia;
 - ▣ Inserção de elemento;
 - ▣ Exclusão de elemento;
 - ▣ Acesso a um elemento;
 - ▣ Busca de um elemento;
 - ▣ Ordenar elementos;
 - ▣ Impressão da lista;
 - ▣ Destruição da lista (desalocar toda a estrutura);

Lista Estática

6

LISTA

qtd

4

dados

32

45

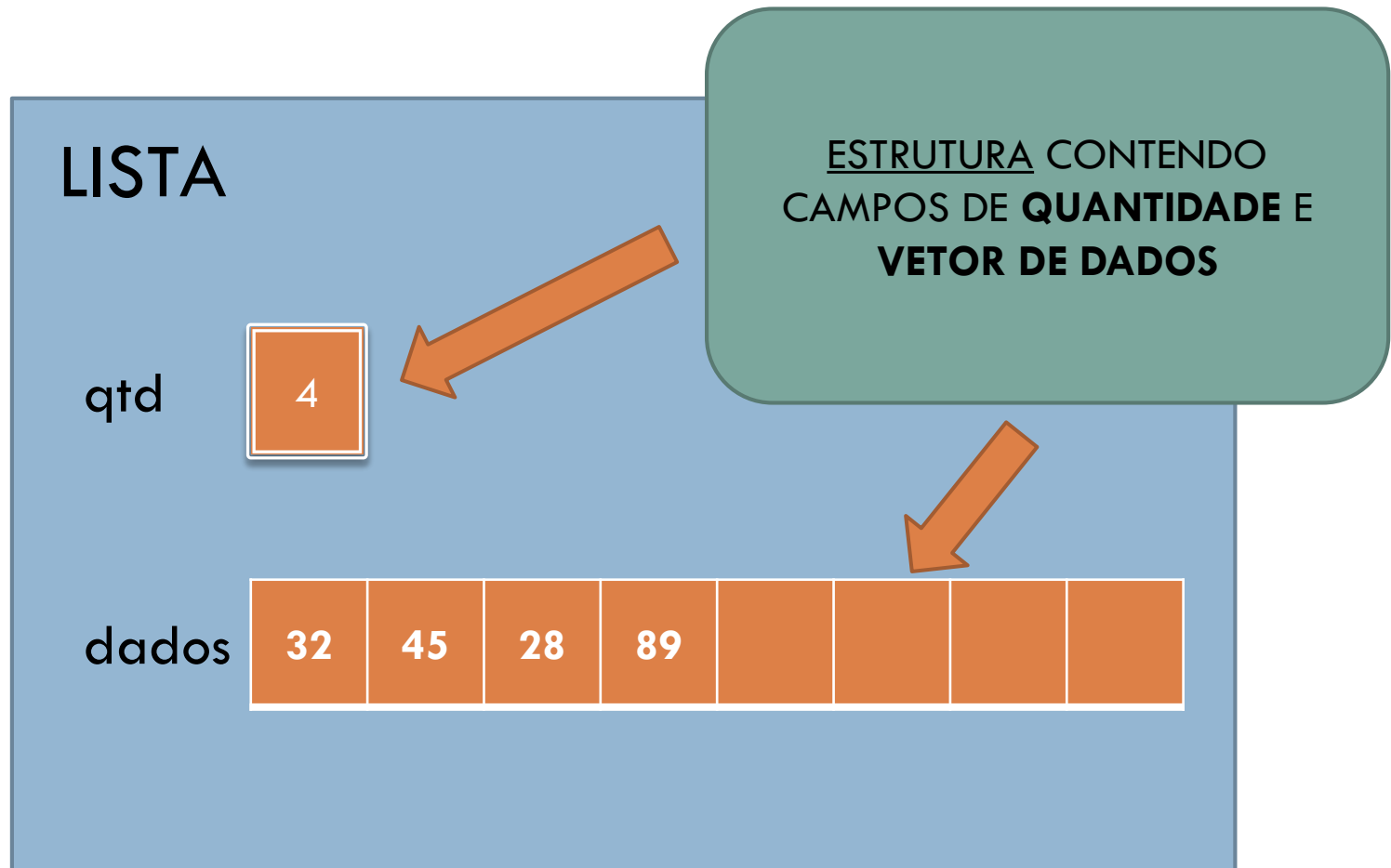
28

89

--	--	--	--	--	--	--	--

Lista Estática

7



Lista Estática

8

LISTA

qtd

4

dados

32	45	28	89				
----	----	----	----	--	--	--	--

A QUANTIDADE AUMENTA
QUANDO **INSERIR** UM
ELEMENTO, E DIMINUI QUANDO
REMOVER UM ELEMENTO

Lista Estática

9

LISTA

qtd

4

dados

32

45

28

89

0

1

2

3

4

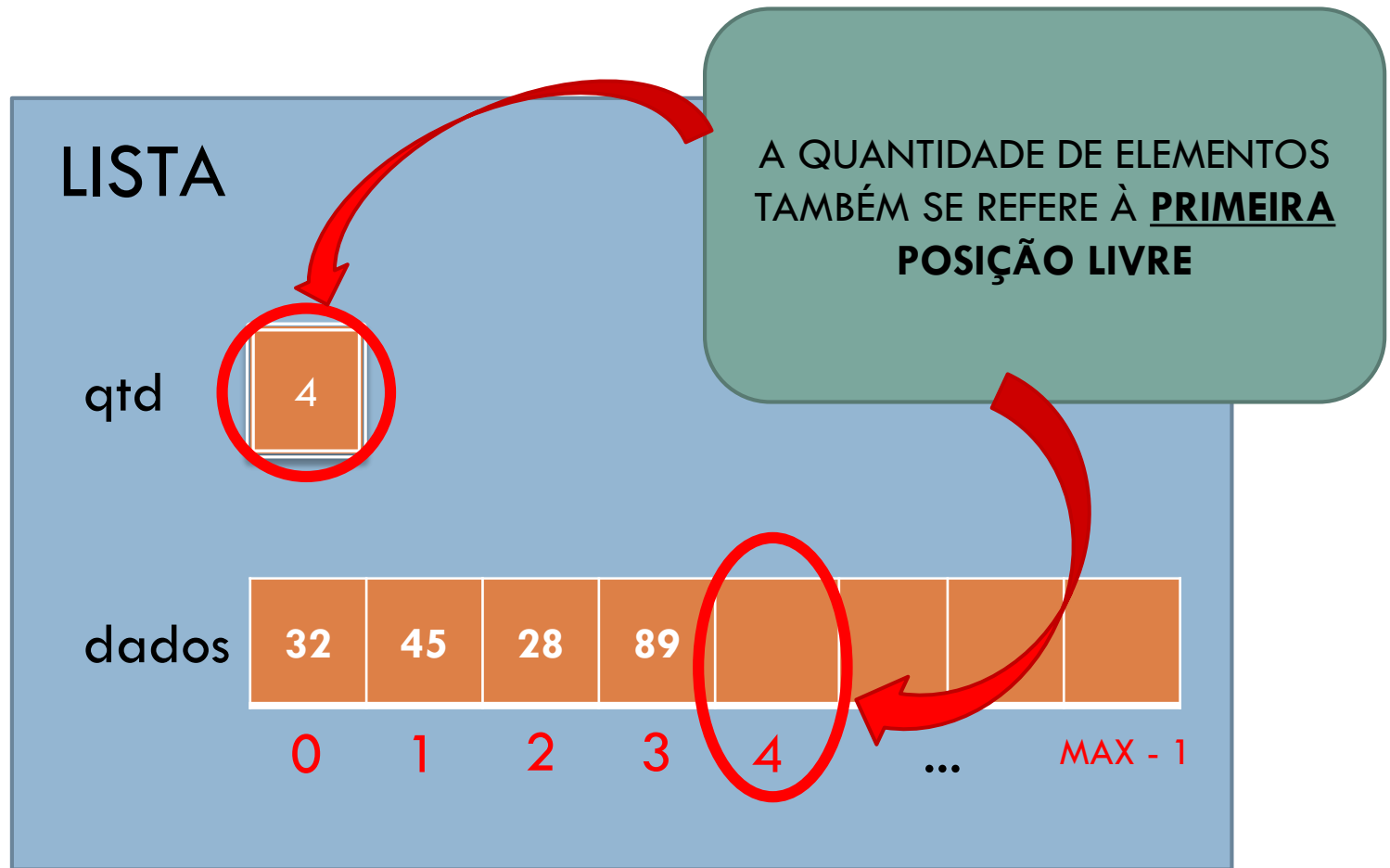
...

MAX - 1

VETOR POSSUI ÍNDICES DE
ELEMENTOS DE **0** ATÉ **MAX - 1**

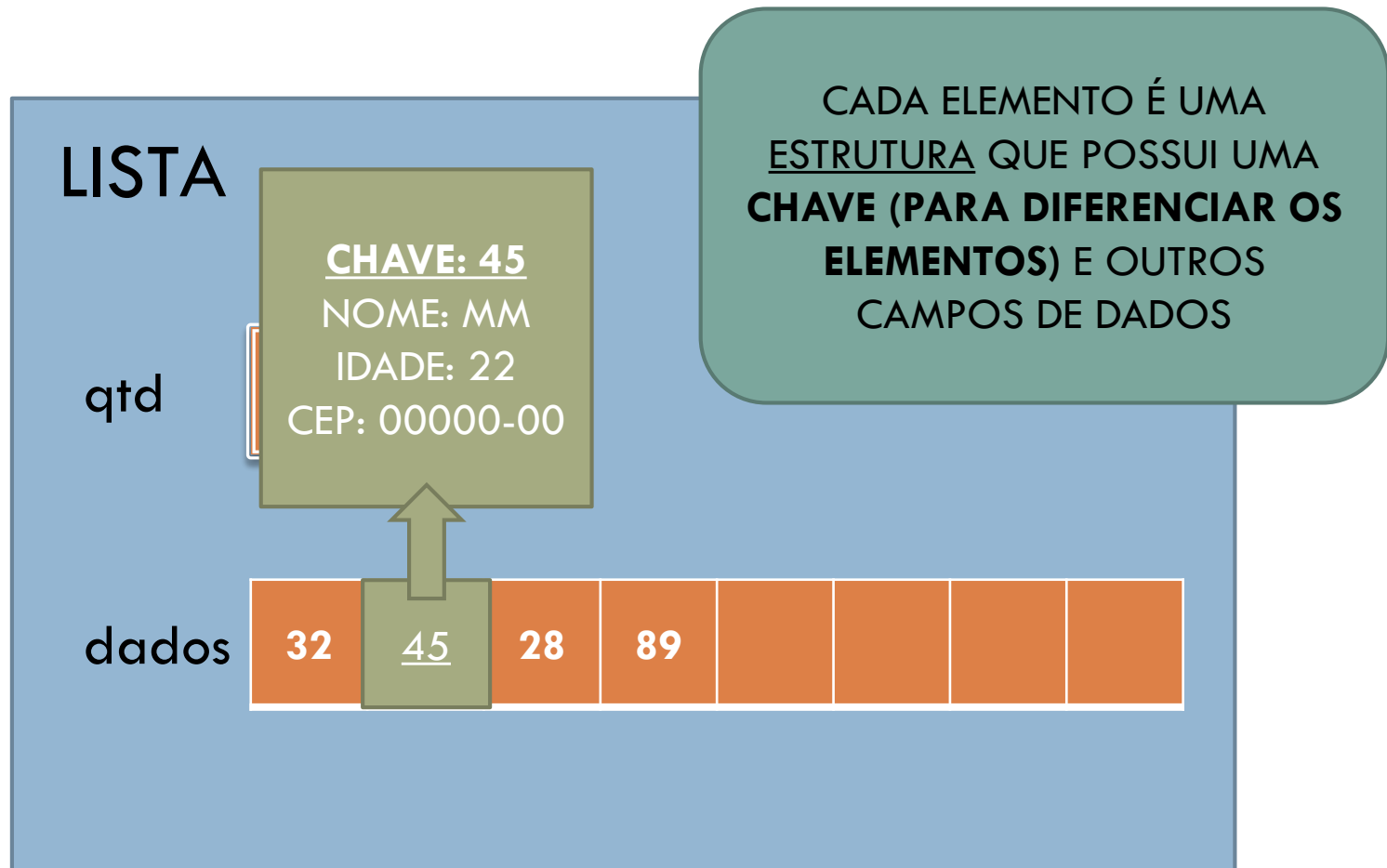
Lista Estática

10



Lista Estática

11



Lista Estática: Exemplo

12

- Lista de alunos:
 - Matricula;
 - Nome;
 - Notas;
- Operações:
 - Criar lista;
 - Deletar lista;
 - Inserir aluno ordenado;
 - Remover aluno;
 - Imprimir lista;

Lista Estática: Exemplo

13

□ Lista de alunos:

□ Mo

□ No

□ No

□ Ope

□ Cri

□ De

□ Ins

□ Re

□ Imprimir lista;

- DEFINIR ESTRUTURA (struct) DE CADA ELEMENTO (.h)
- DECLARAR O TIPO DE PONTEIRO (typedef) DA LISTA (.h)
- DEFINIR TAMANHO MÁXIMO DA LISTA (.h)
- DECLARAR ESCOPO DAS FUNÇÕES (.h)

- DEFINIR ESTRUTURA (struct) DE LISTA (.c)
- DEFINIR OPERAÇÕES (funções) DA TAD (.c)

Lista Estática: Exemplo

14

```
//Arquivo ListaSequencial.h  
#define MAX 100
```

Lista Estática: Exemplo

15

```
//Arquivo ListaSequencial.h
#define MAX 100

//ESTE TIPO DE DADO ESTÁ VIZÍVEL PARA QUEM IMPORTAR A BIBLIOTECA
//PODE SER CRIADO E ACESSADO DIRETAMENTE NA MAIN
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};
```

Lista Estática: Exemplo

16

```
//Arquivo ListaSequencial.h
#define MAX 100

//ESTE TIPO DE DADO ESTÁ VIZÍVEL PARA QUEM IMPORTAR A BIBLIOTECA
//PODE SER CRIADO E ACESSADO DIRETAMENTE NA MAIN
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};

//ESTE TIPO DE DADO PODE SER UTILIZADO APENAS COMO PONTEIRO
//A LISTA FICARÁ ENCAPSULADA NO PROGRAMA PRINCIPAL
typedef struct lista Lista;
```


Lista Estática: Exemplo

17

```
//Arquivo ListaSequencial.h
#define MAX 100

//ESTE TIPO DE DADO ESTÁ VIZÍVEL PARA QUEM IMPORTAR A BIBLIOTECA
//PODE SER CRIADO E ACESSADO DIRETAMENTE NA MAIN
struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};

//ESTE TIPO DE DADO PODE SER UTILIZADO APENAS COMO PONTEIRO
//A LISTA FICARÁ ENCAPSULADA NO PROGRAMA PRINCIPAL
typedef struct lista Lista;

Lista* cria_lista();
void libera_lista(Lista* li);
int insere_lista_ordenada(Lista* li, struct aluno al);
void imprime_lista(Lista* li);
int remove_lista_otimizado(Lista* li, int mat);
```

Lista Estática: Exemplo

18

```
//Arquivo ListaSequencial.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaEstatica.h" //inclui os Protótipos
```

Lista Estática: Exemplo

19

```
//Arquivo ListaSequencial.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaEstatica.h" //inclui os Protótipos

//Definição do tipo lista
struct lista{
    int qtd;
    struct aluno dados[MAX];
};
```

Lista Estática: Exemplo

20

```
//Arquivo ListaSequencial.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaEstatica.h" //inclui os Protótipos

//Definição do tipo lista
struct lista{
    int qtd;
    struct aluno dados[MAX];
};

Lista* cria_lista(){
    Lista *li;
    li = (Lista*) malloc(sizeof(struct lista));
    if(li != NULL)
        li->qtd = 0;
    return li;
}

void libera_lista(Lista* li){
    free(li);
}
```

Lista Estática: Exemplo

21

```
//Arquivo ListaSequencial.c
#include <stdio.h>
#include <stdlib.h>
#include "ListaEstatica.h" //inclui os Protótipos

#include <stdio.h>
#include <stdlib.h>
#include "ListaEstatica.h"
int main() {

    Lista* li = cria_lista(); // Criando o bloco inicial da lista
                             // Armazena a quantidade e o inicio da lista

    libera_lista(li);

    system("pause");
    return 0;
}

return li;
}

void libera_lista(Lista* li) {
    free(li);
}
```

Lista Estática: Exemplo

22

- **Inserção ordenada:** inserir elementos novos mantendo uma ordenação de acordo com a chave.
 - ▣ Crescentes: o próximo elemento da sequência é maior que o atual, sucessivamente.
 - ▣ Decrescente: o próximo elemento da sequência é menor que o atual, sucessivamente.

- **Procurar local** para inserir o novo elemento, mantendo a ordem.

Lista Estática: Exemplo

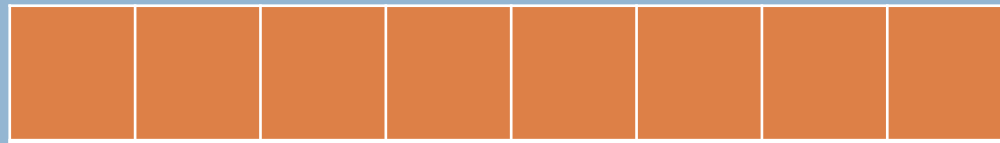
23

LISTA

qtd

0

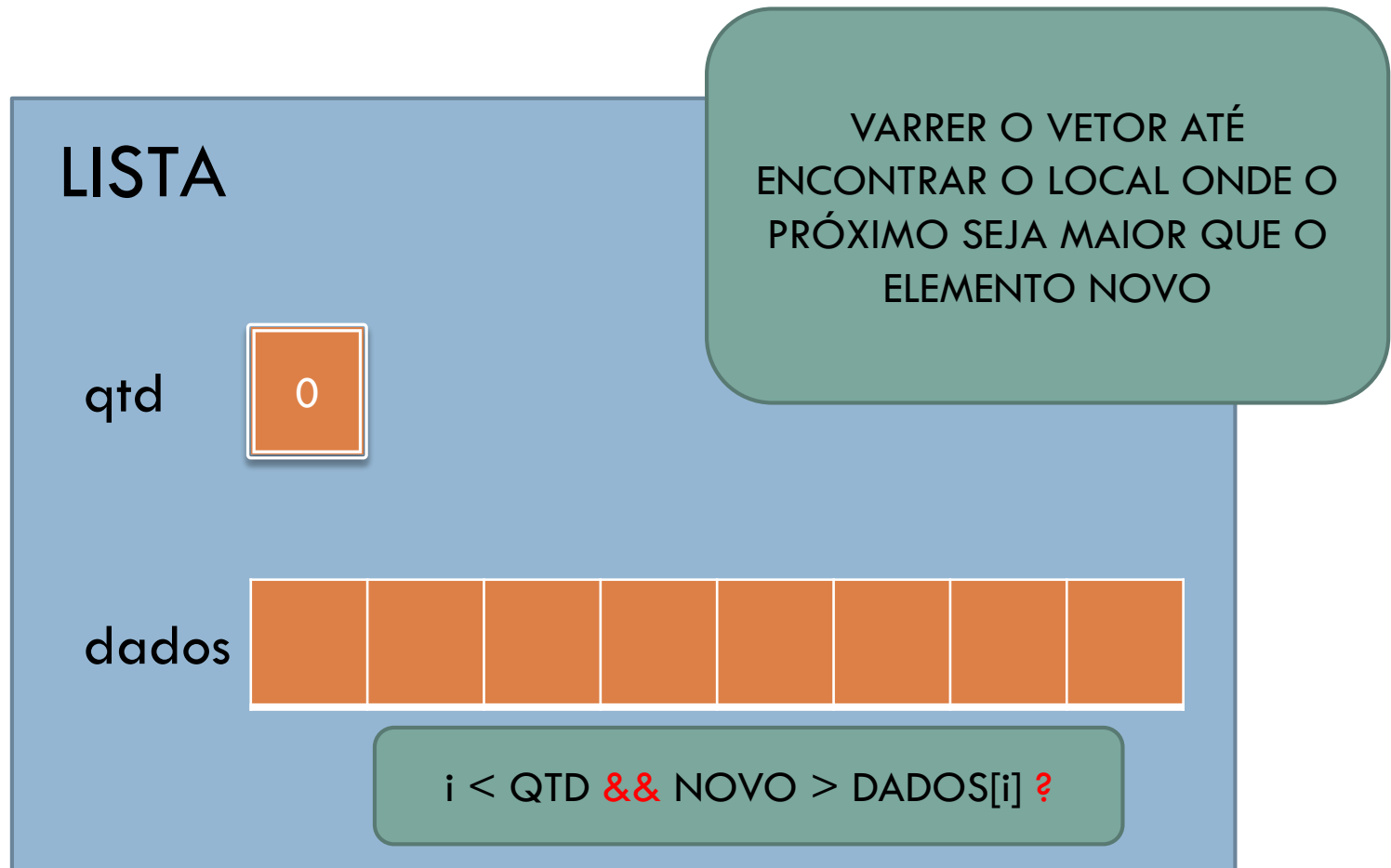
dados



VARRER O VETOR ATÉ
ENCONTRAR O LOCAL ONDE O
PRÓXIMO SEJA MAIOR QUE O
ELEMENTO NOVO

Lista Estática: Exemplo

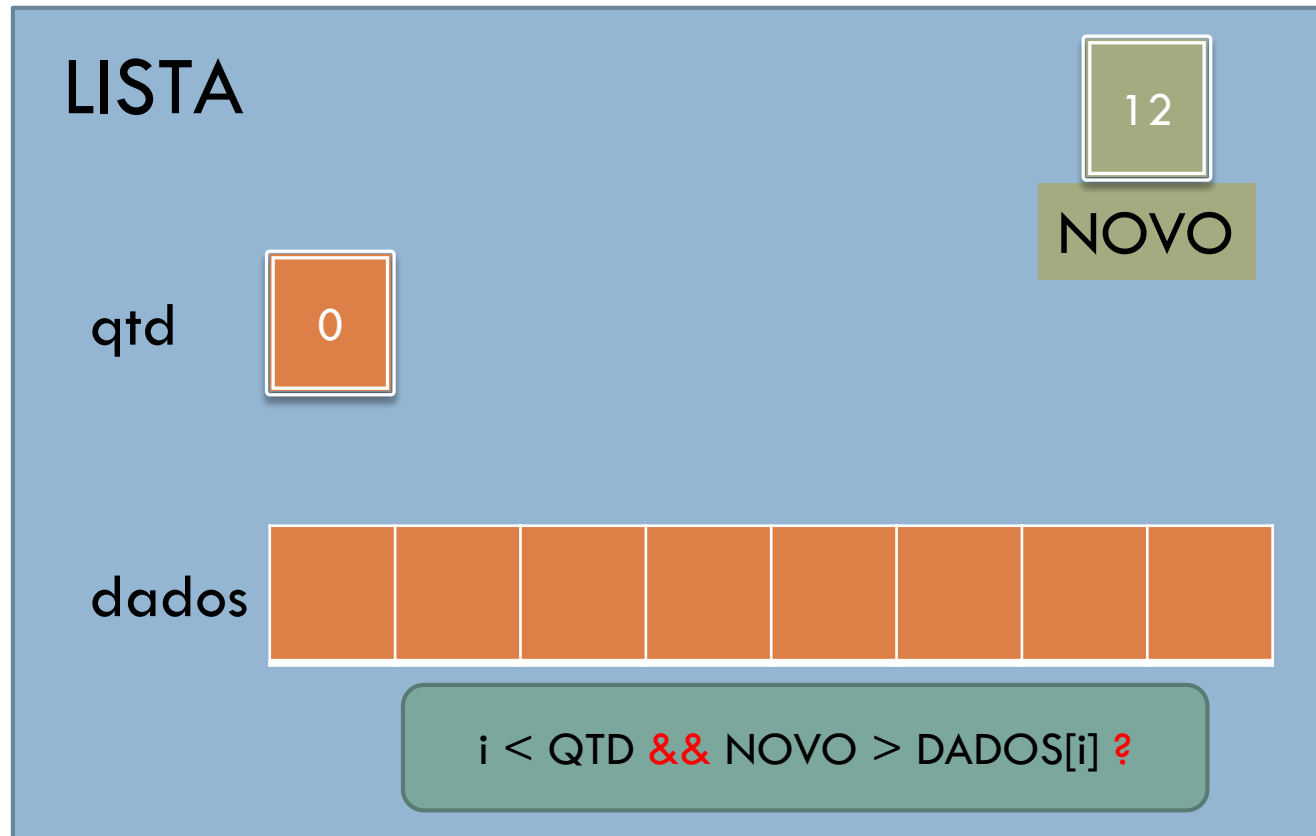
24



i É UMA VARIÁVEL DE CONTROLE, PARA ARMAZENAR A POSIÇÃO EM QUE O NOVO ELEMENTO SERÁ ARMAZENADO

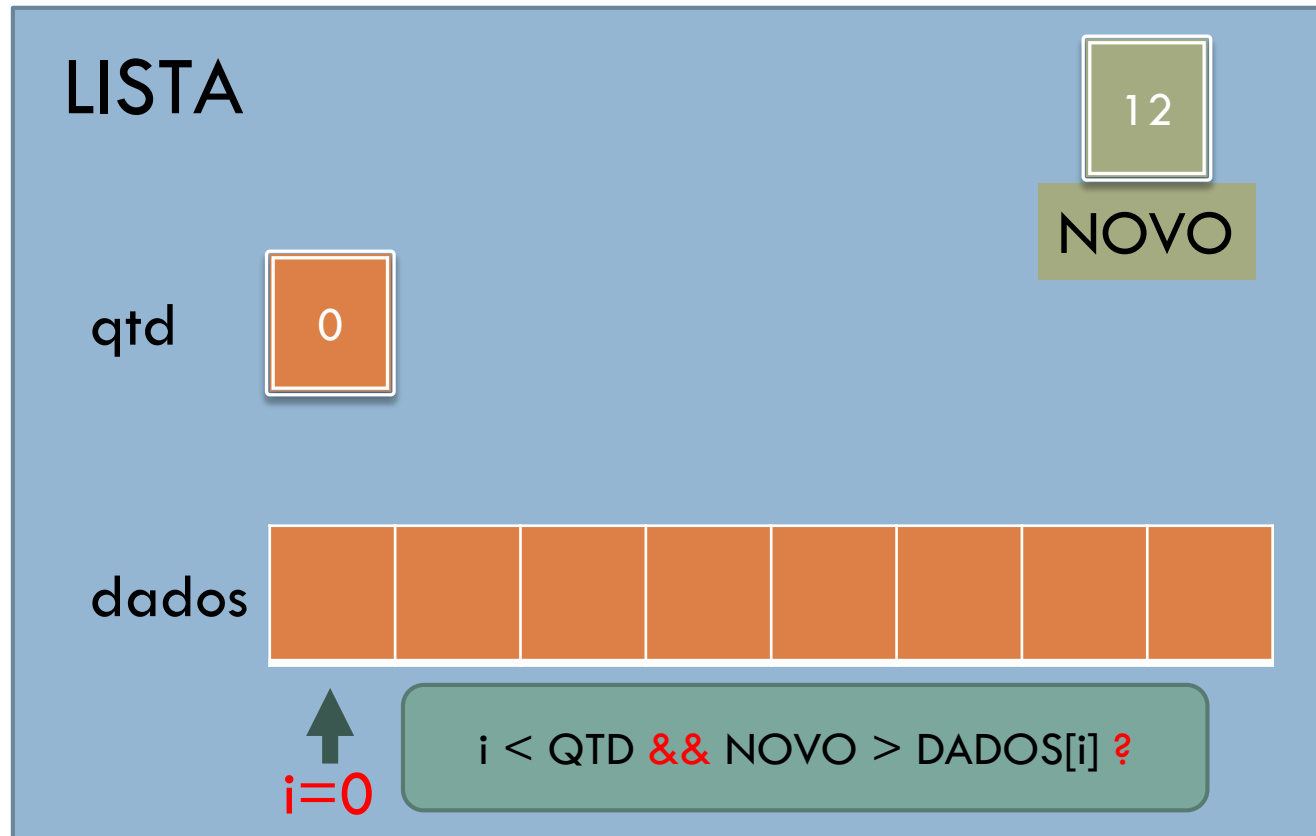
Lista Estática: Exemplo

25



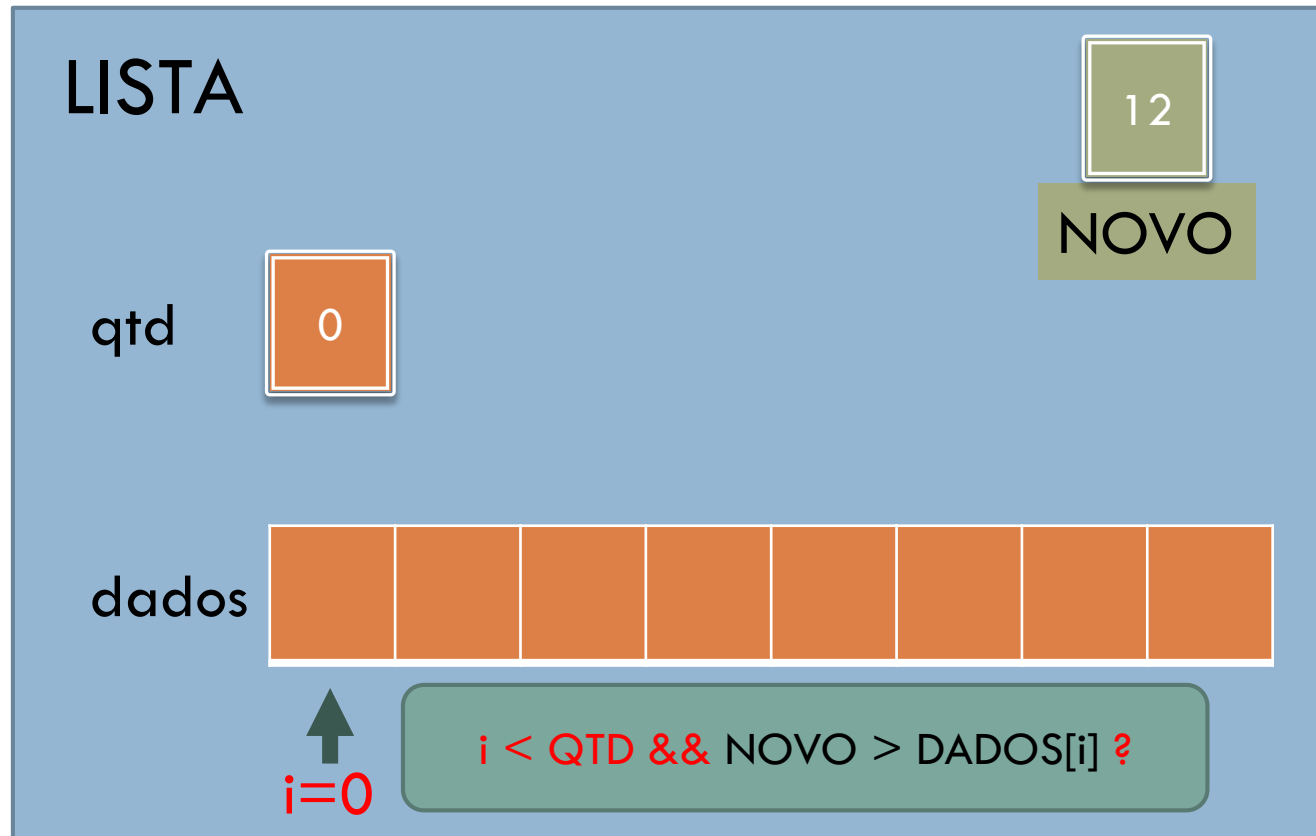
Lista Estática: Exemplo

26



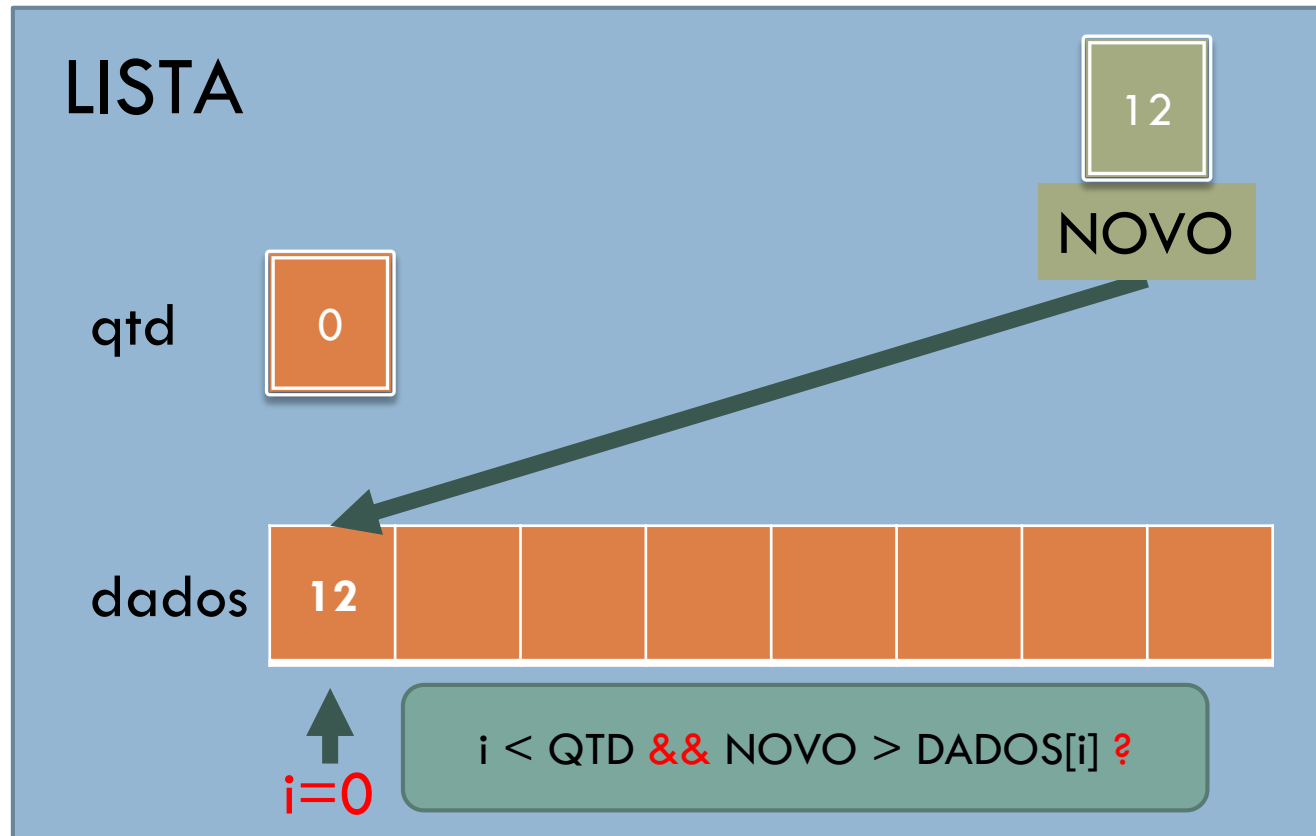
Lista Estática: Exemplo

27



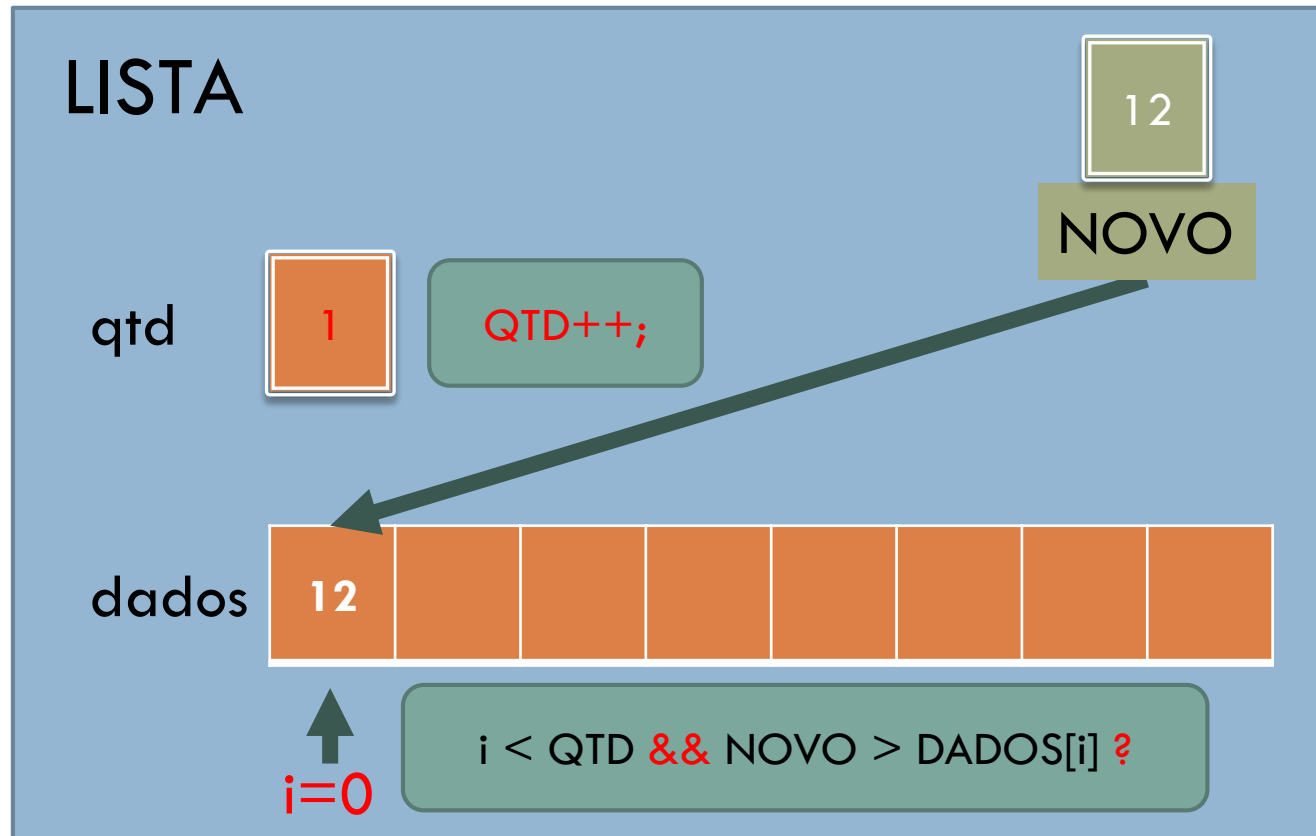
Lista Estática: Exemplo

28



Lista Estática: Exemplo

29



Lista Estática: Exemplo

30

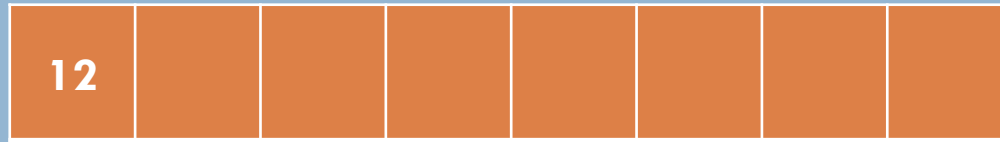
LISTA

qtd

1

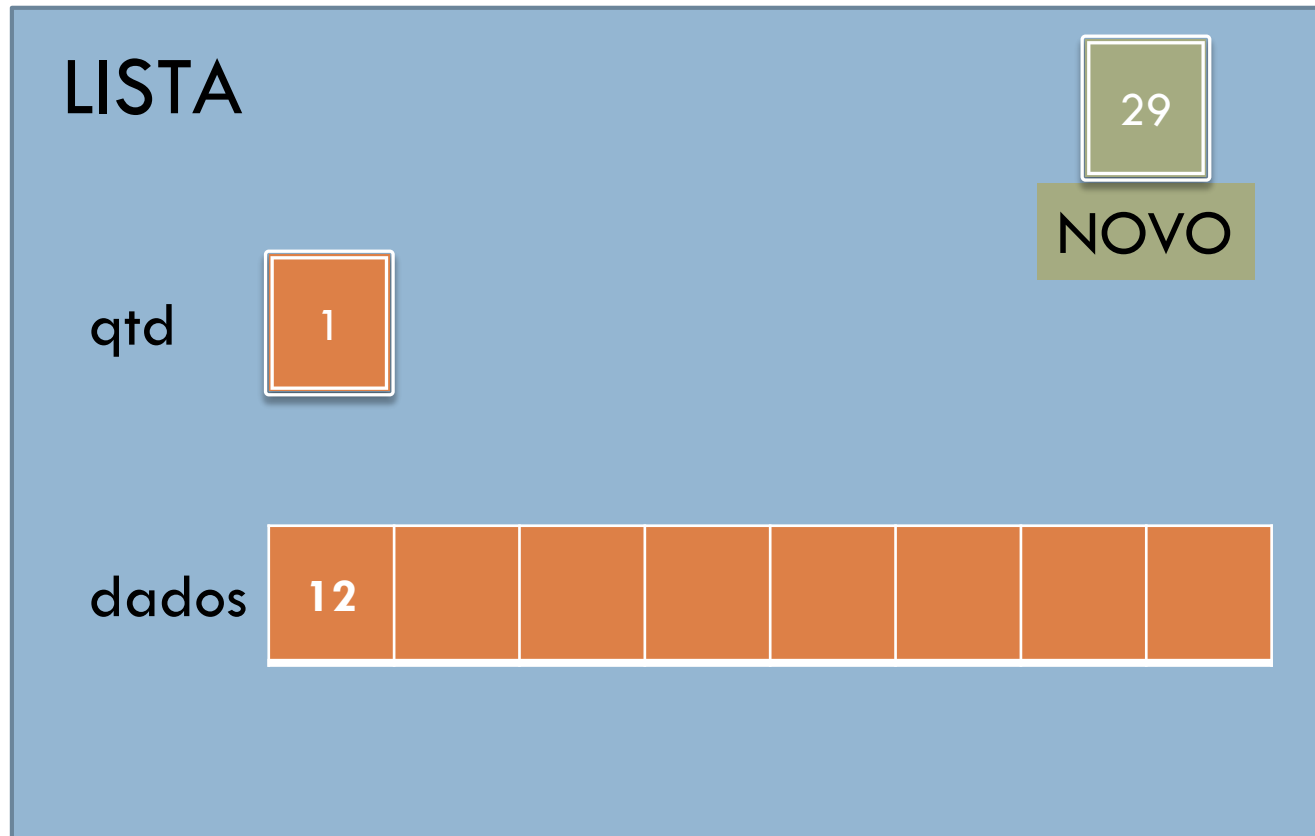
dados

12



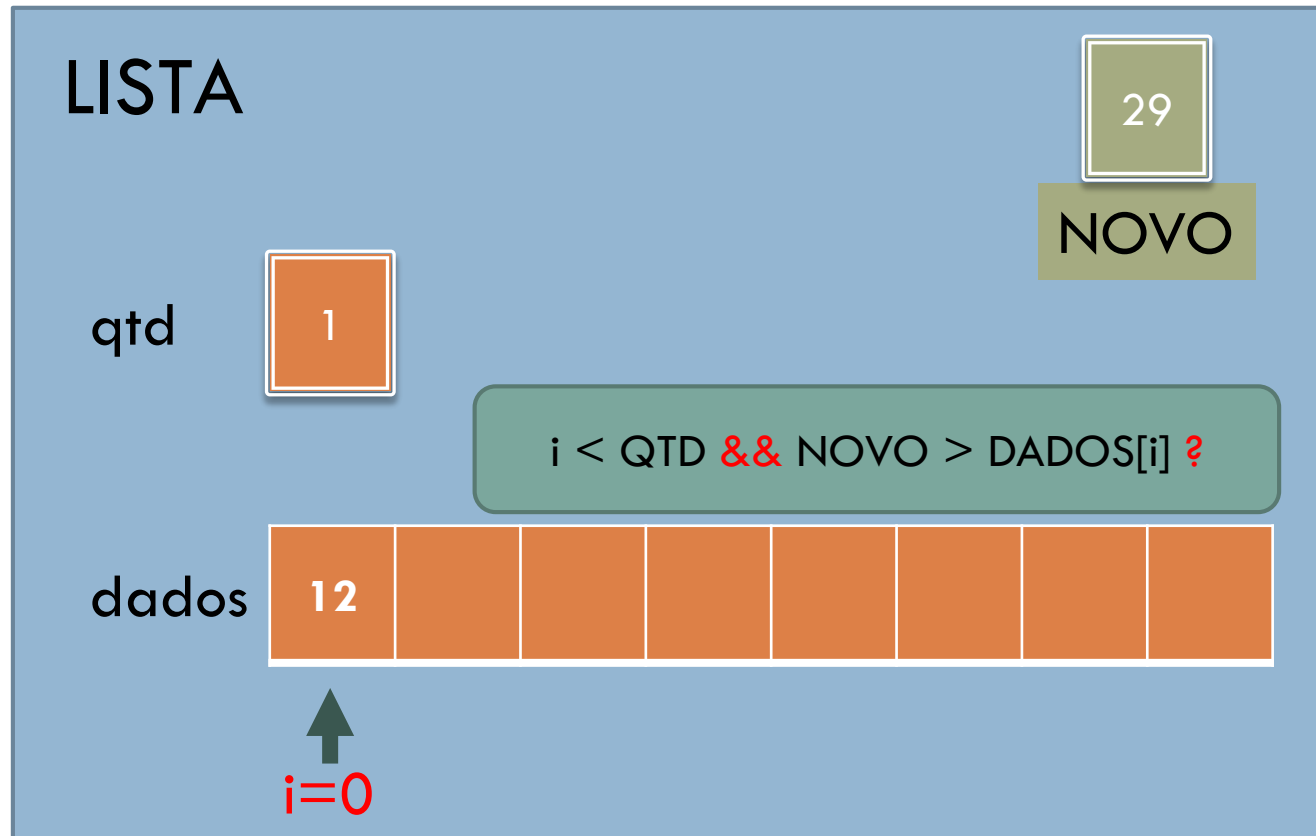
Lista Estática: Exemplo

31



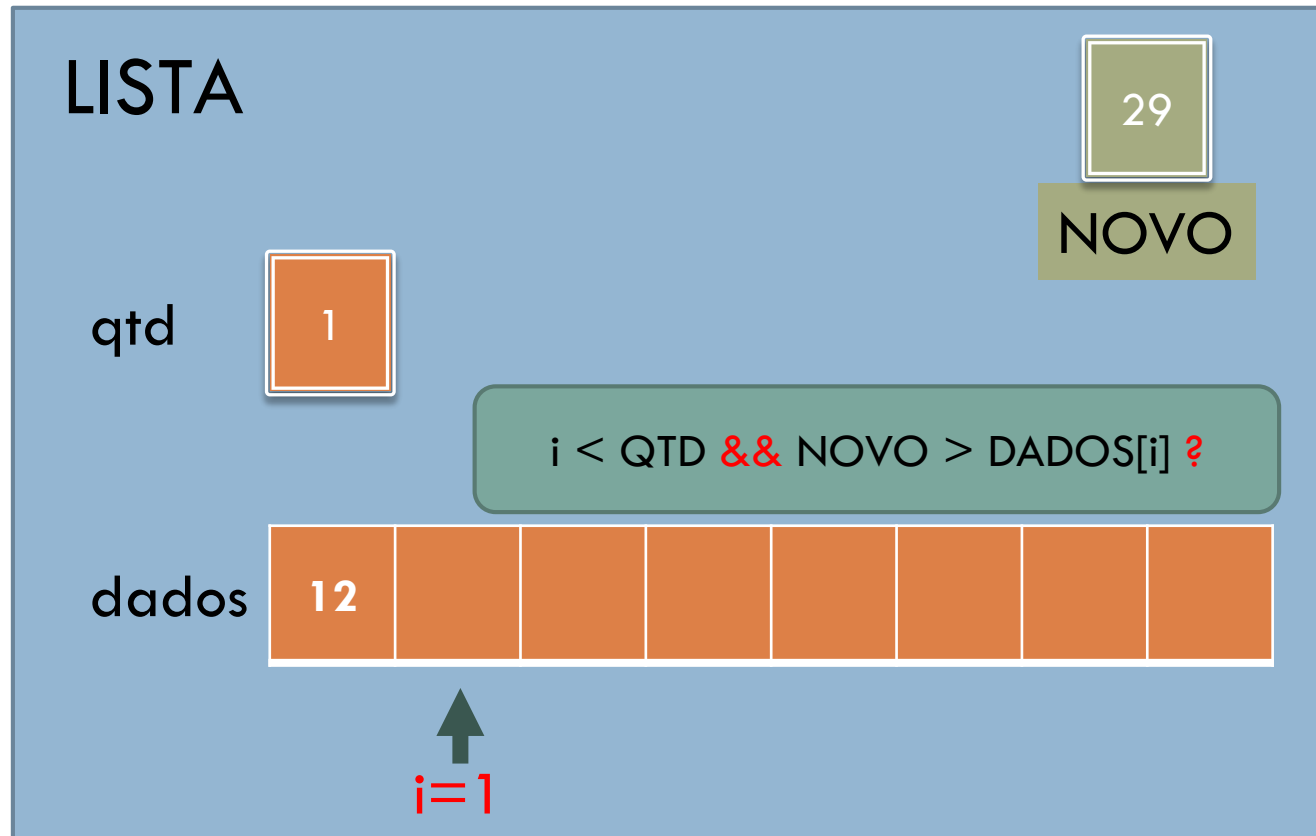
Lista Estática: Exemplo

32



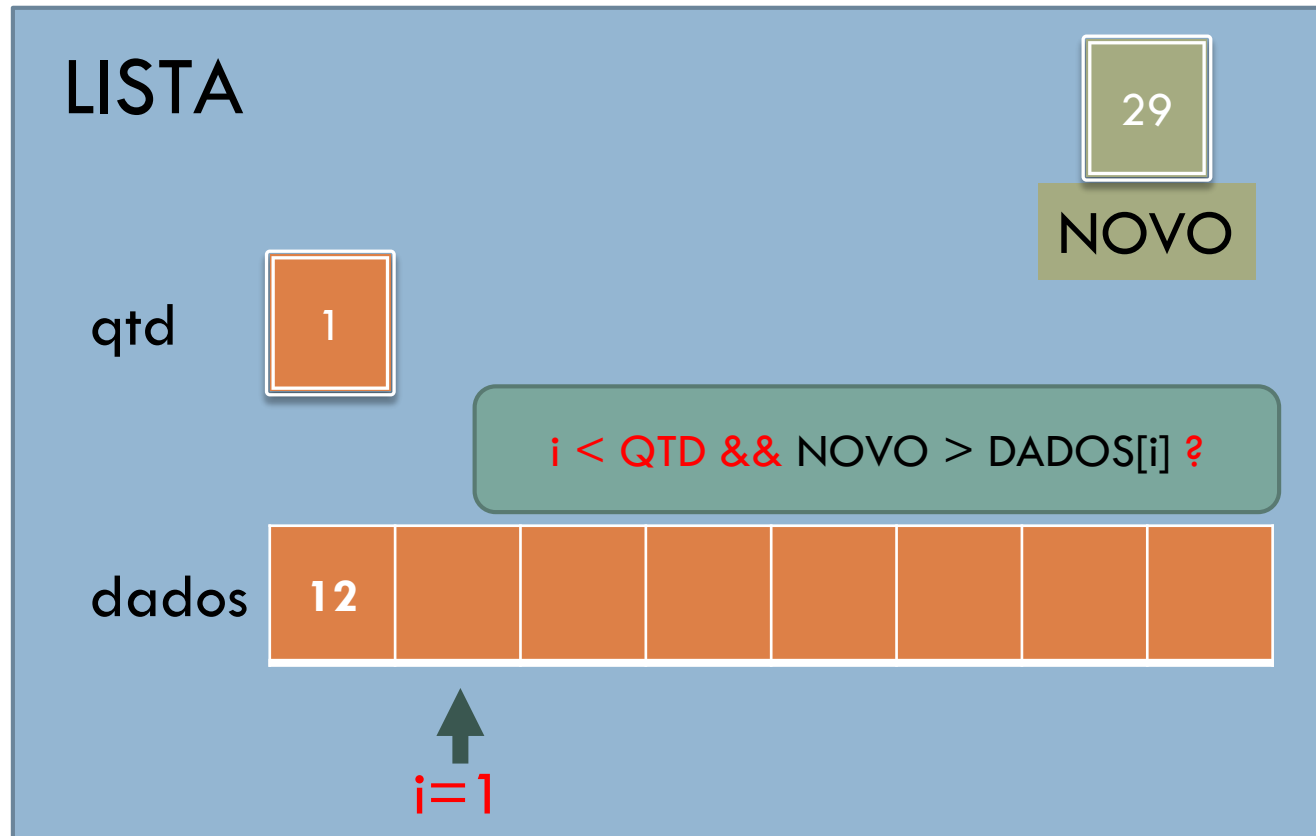
Lista Estática: Exemplo

33



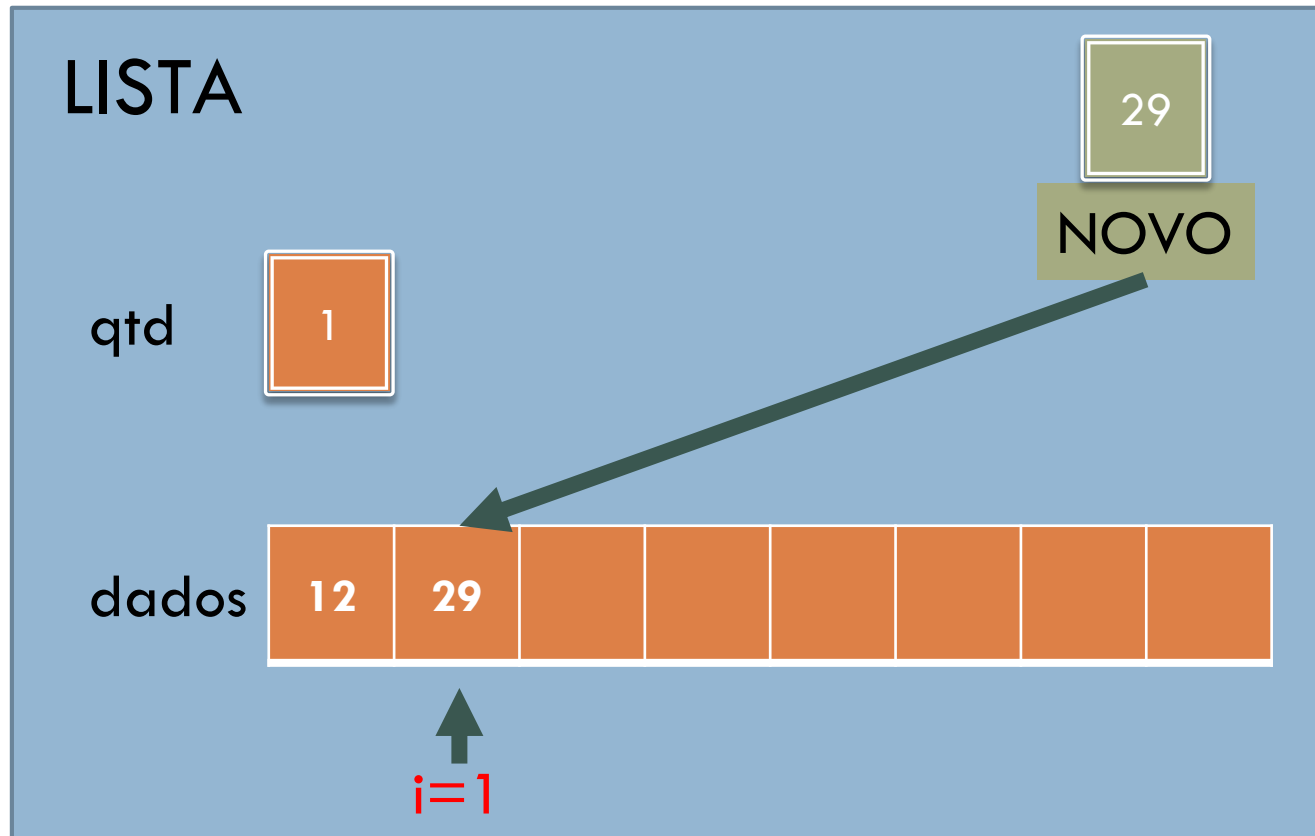
Lista Estática: Exemplo

34



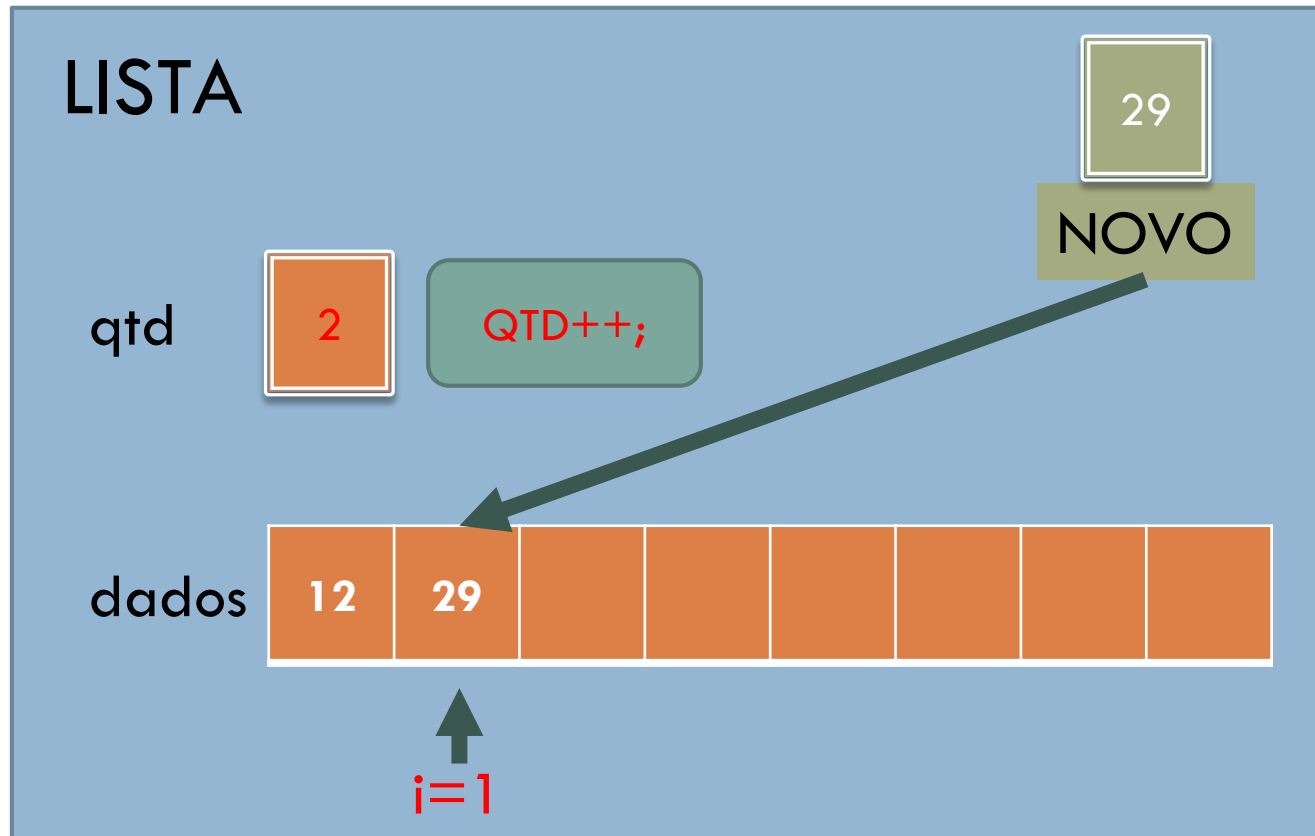
Lista Estática: Exemplo

35



Lista Estática: Exemplo

36



Lista Estática: Exemplo

37

LISTA

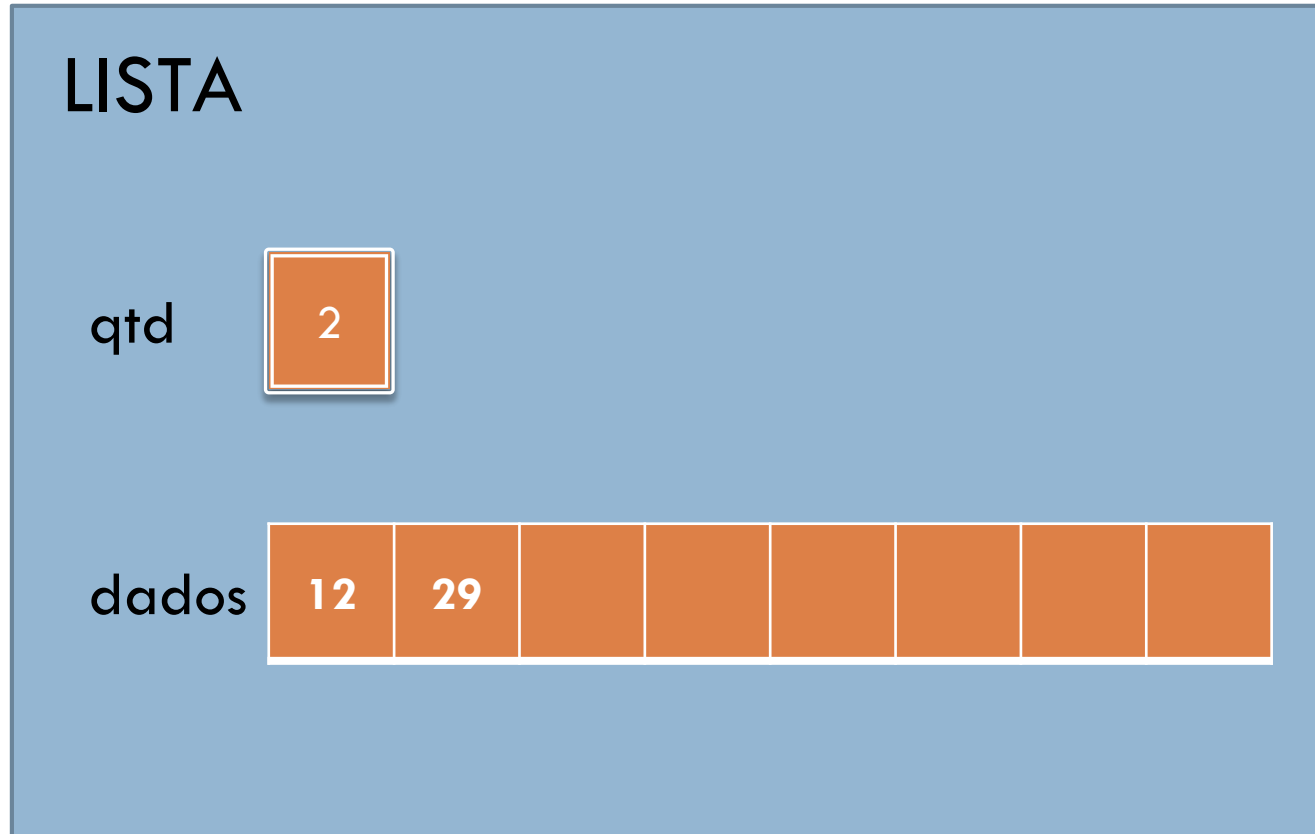
qtd

2

dados

12

29



Lista Estática: Exemplo

38

LISTA

qtd

4

dados

12

29

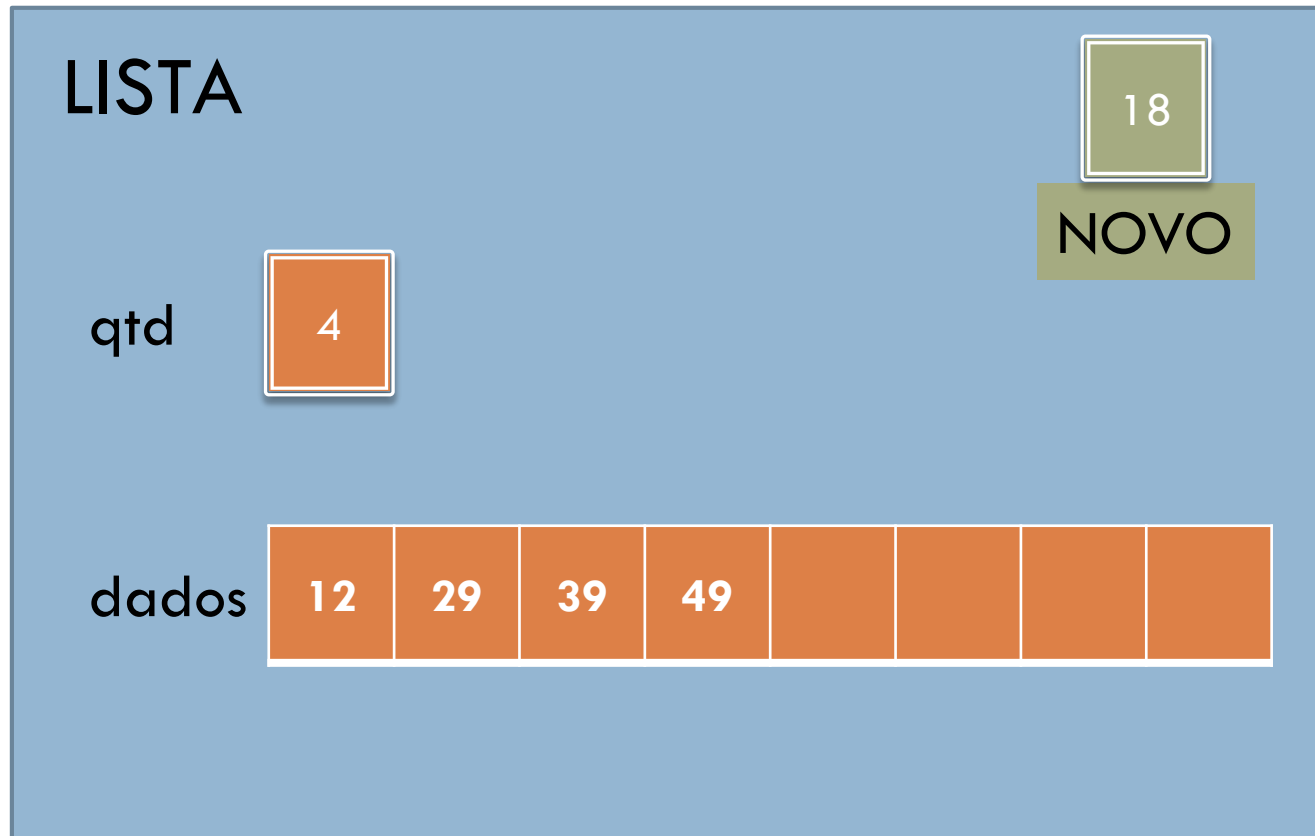
39

49

--	--	--	--	--	--	--	--

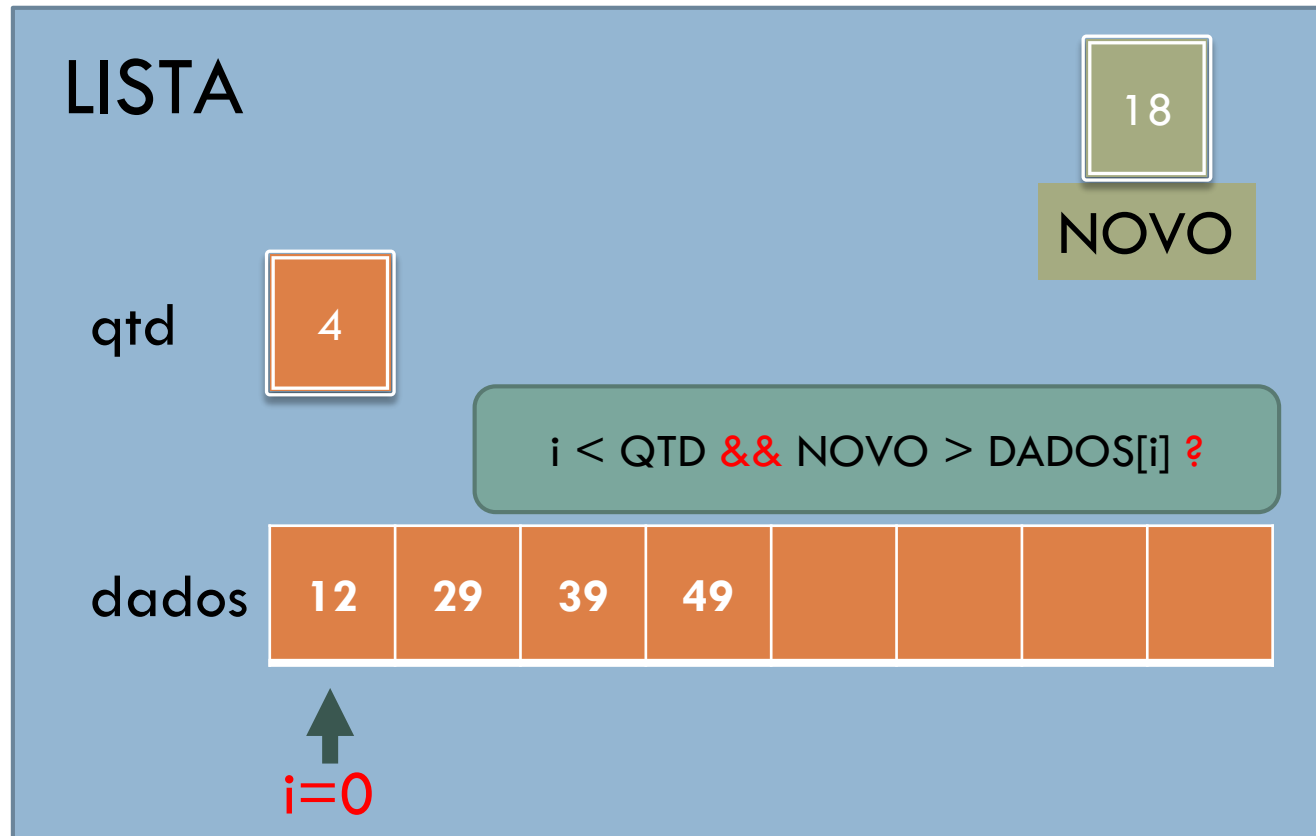
Lista Estática: Exemplo

39



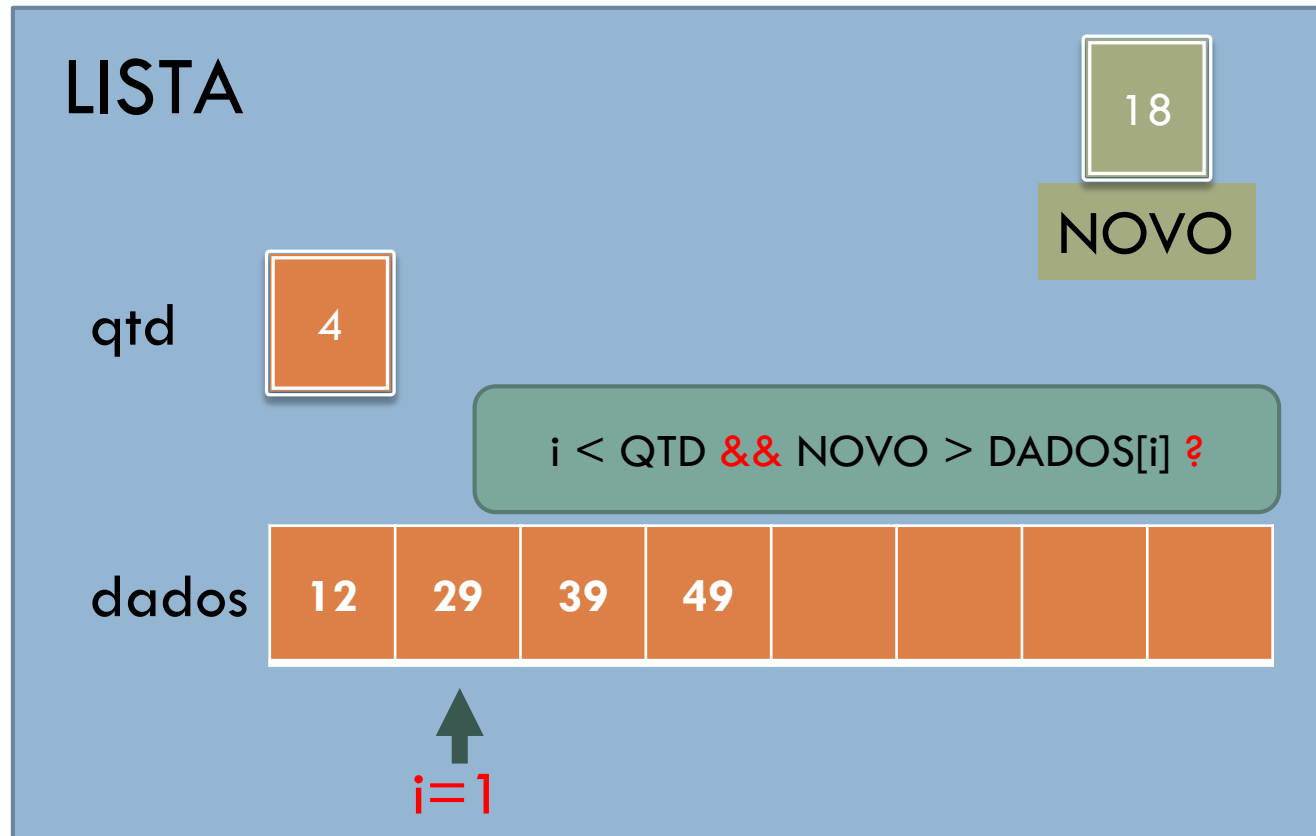
Lista Estática: Exemplo

40



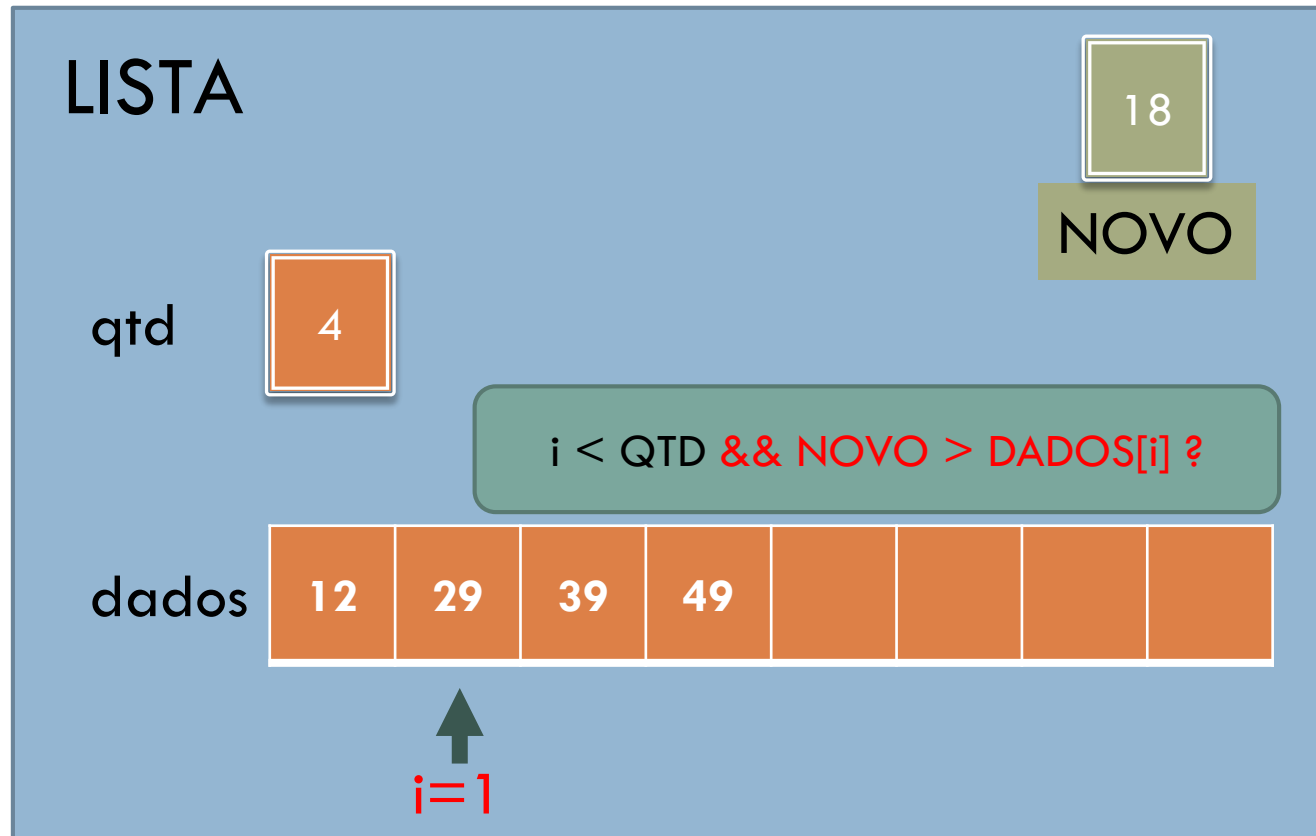
Lista Estática: Exemplo

41



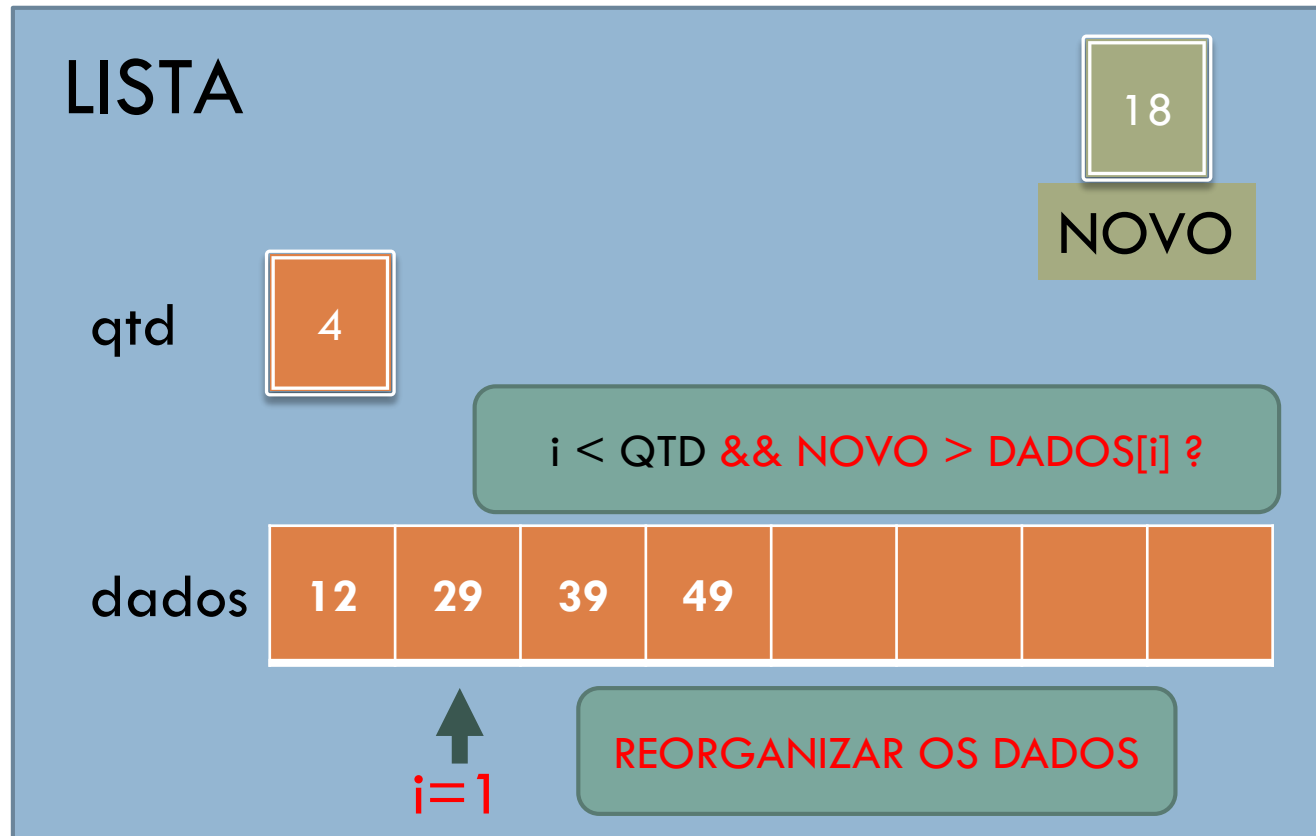
Lista Estática: Exemplo

42



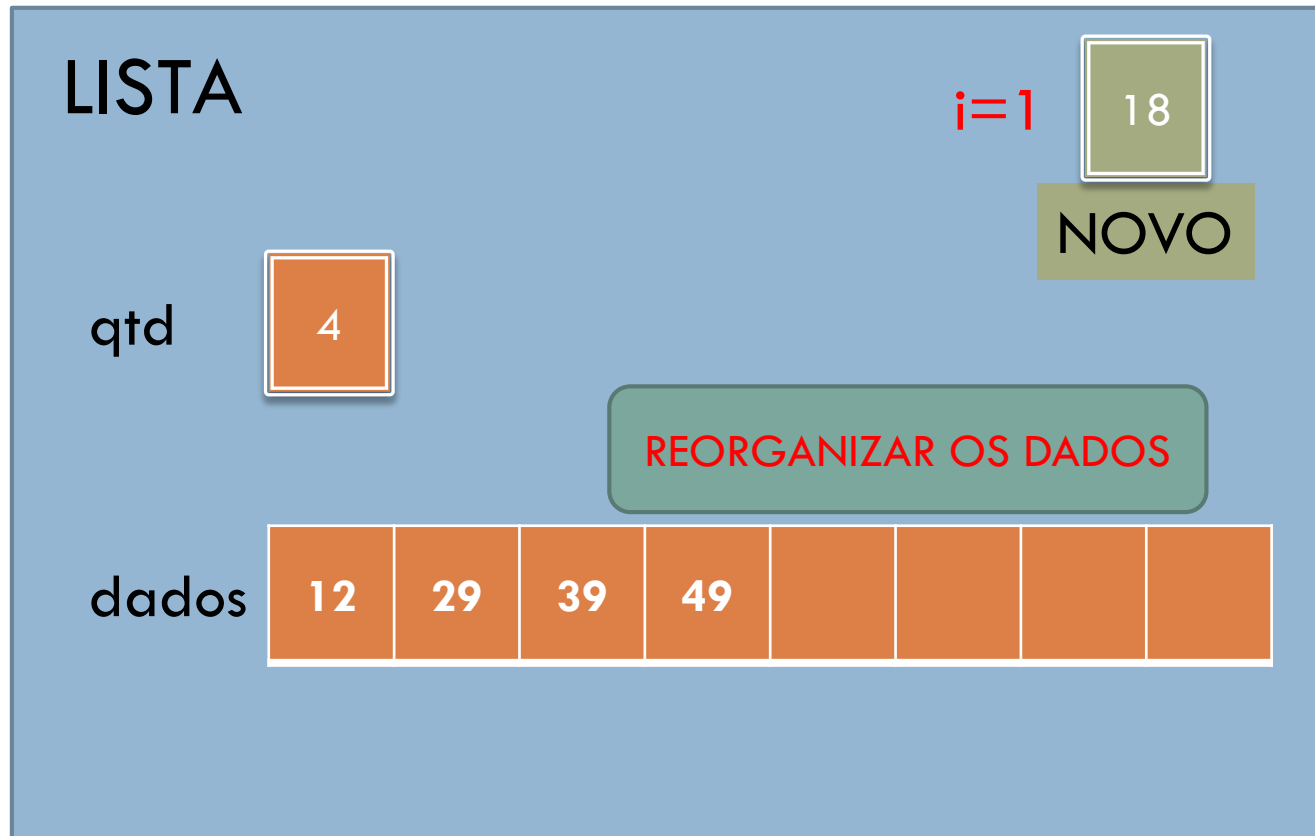
Lista Estática: Exemplo

43



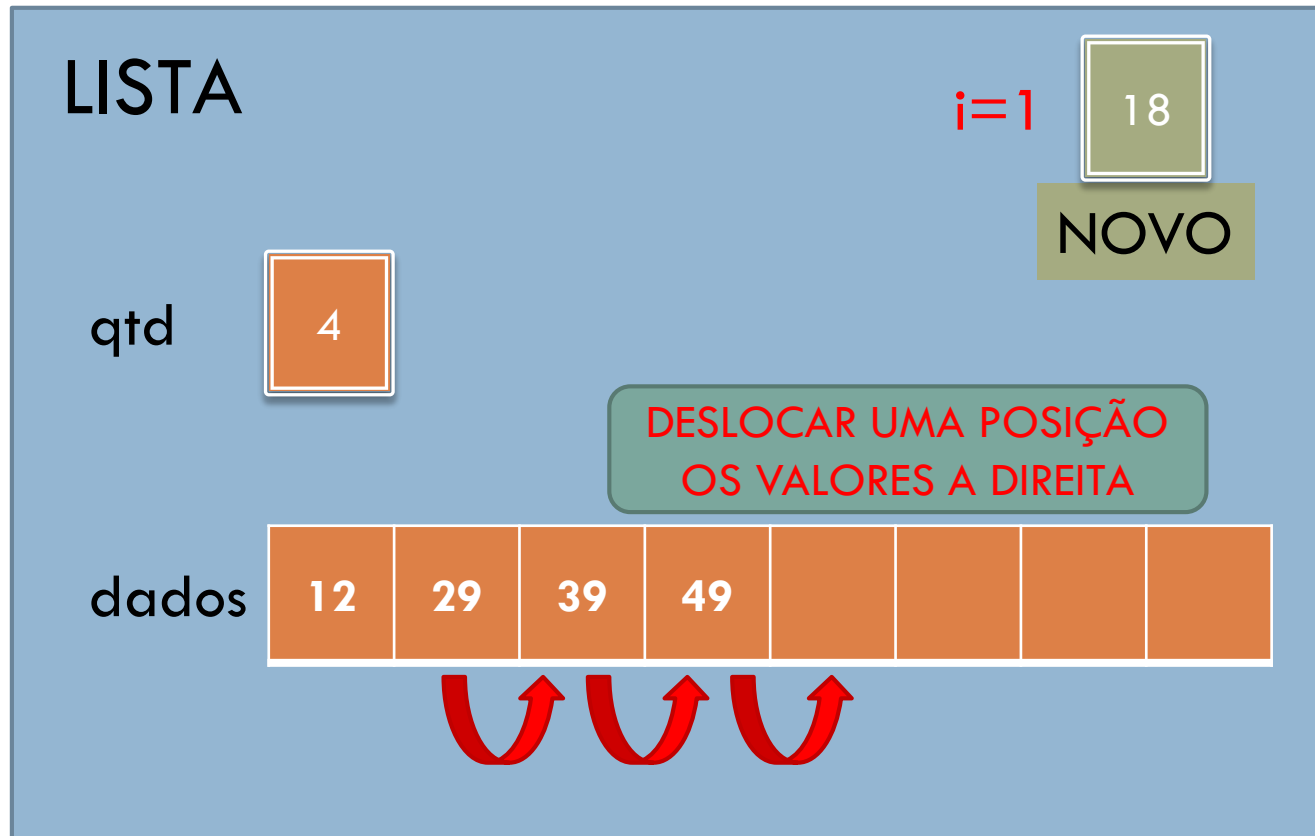
Lista Estática: Exemplo

44



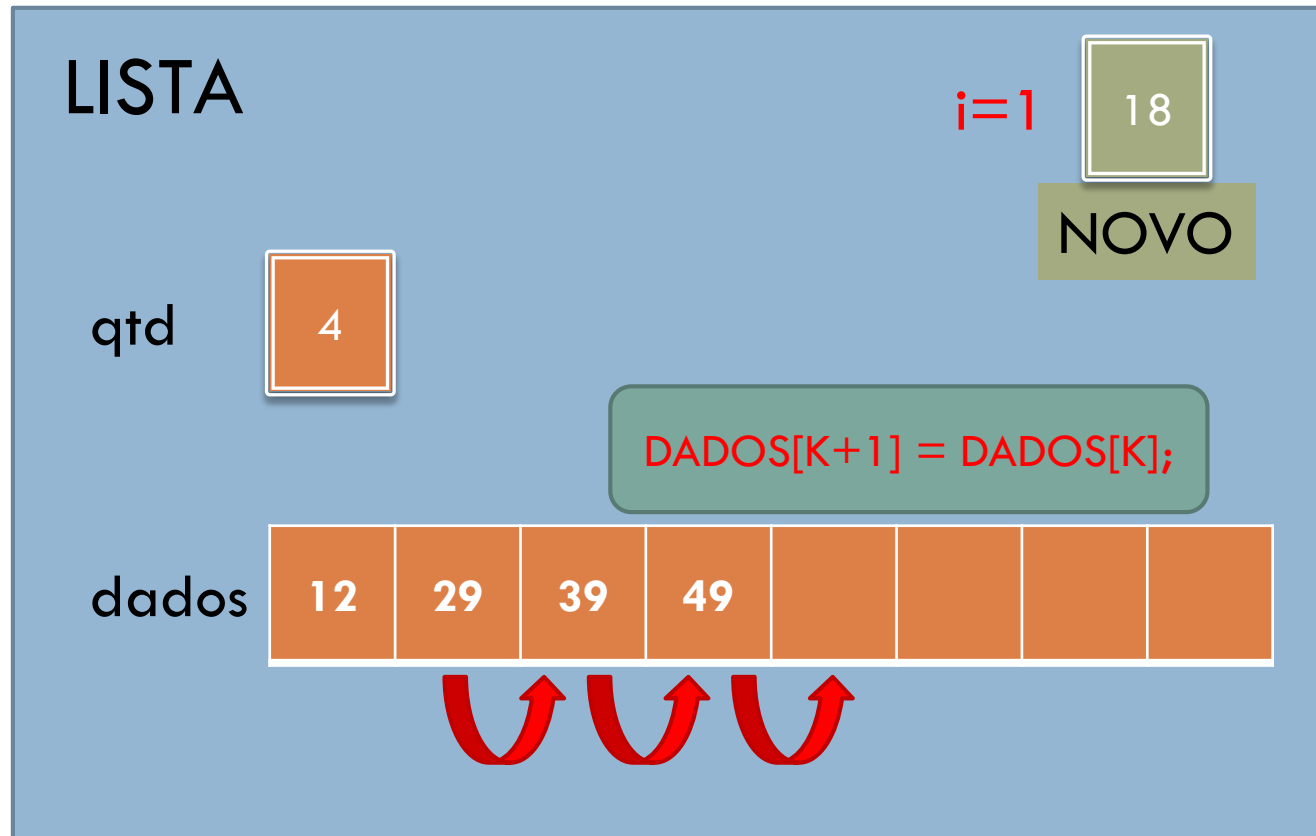
Lista Estática: Exemplo

45



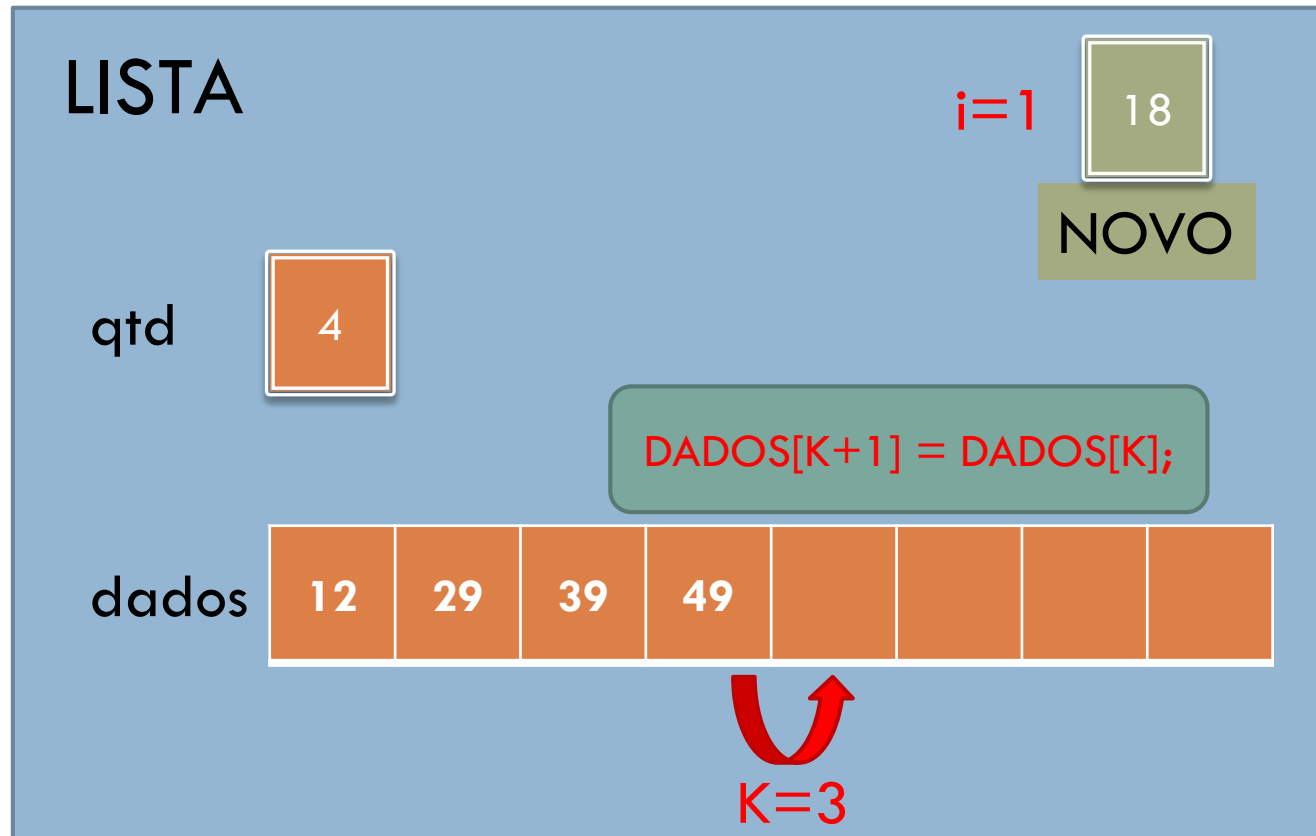
Lista Estática: Exemplo

46



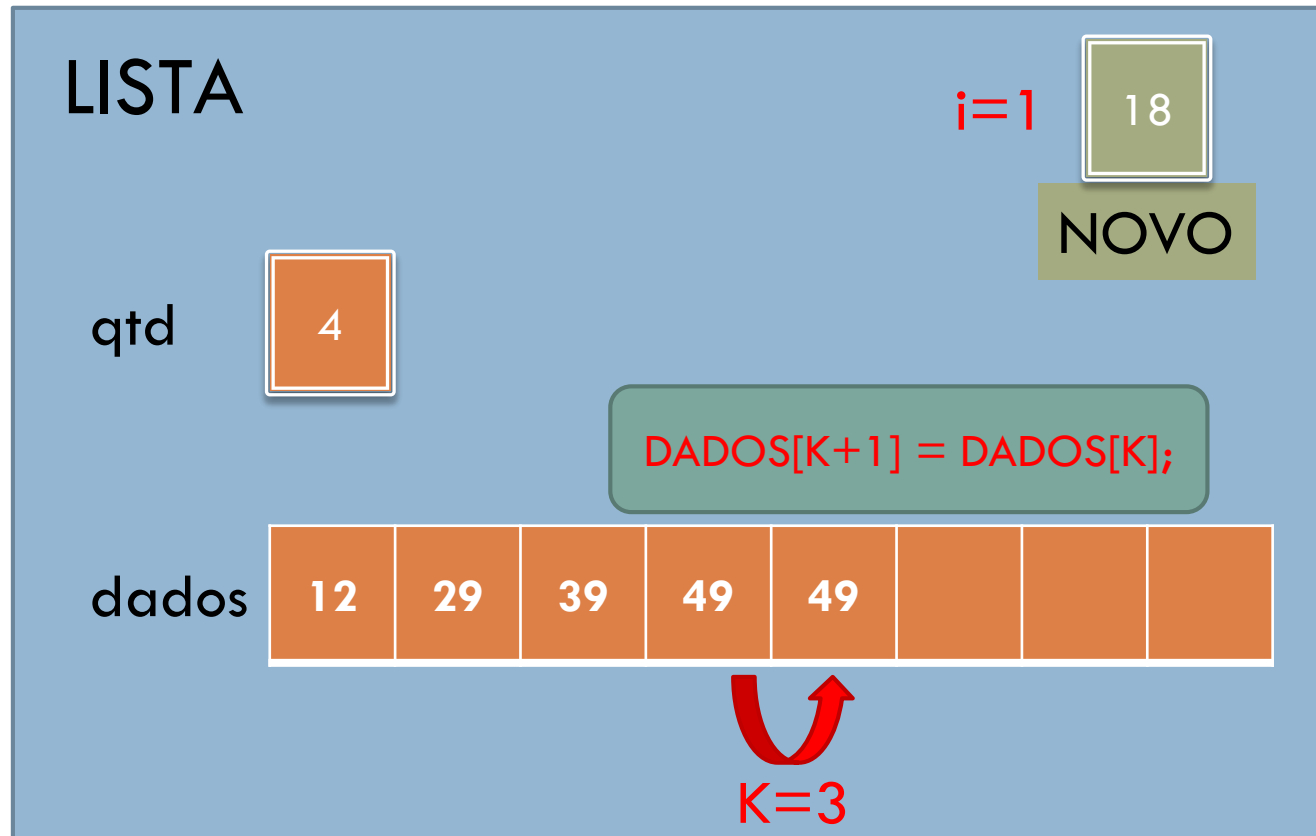
Lista Estática: Exemplo

47



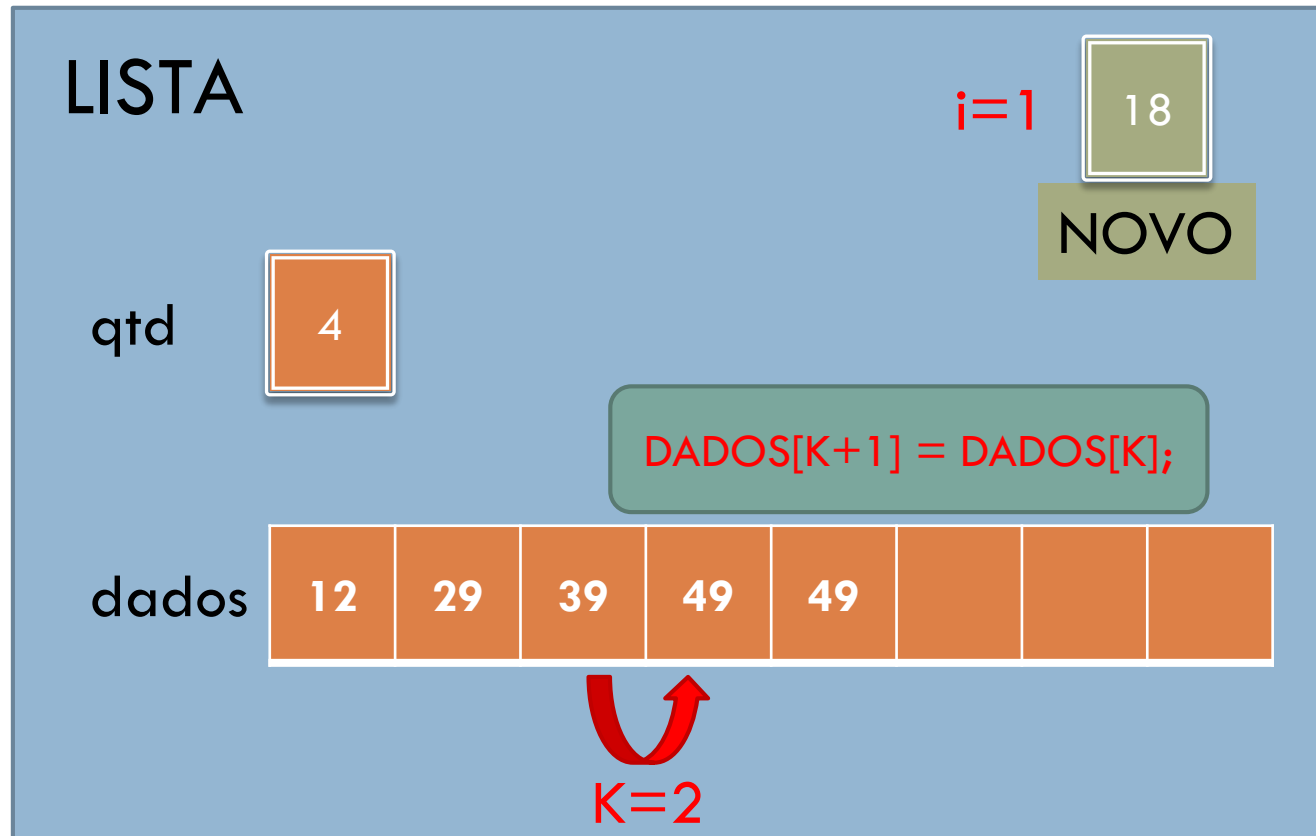
Lista Estática: Exemplo

48



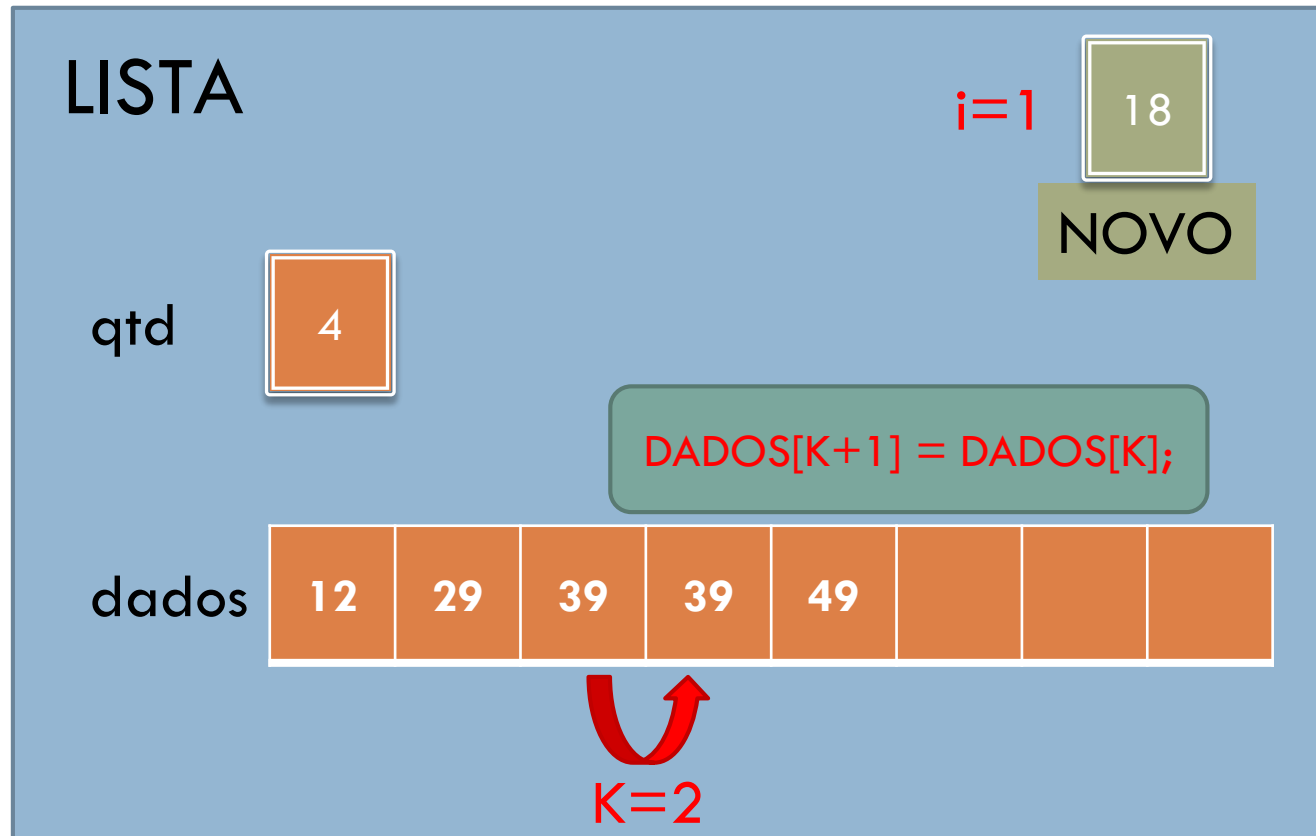
Lista Estática: Exemplo

49



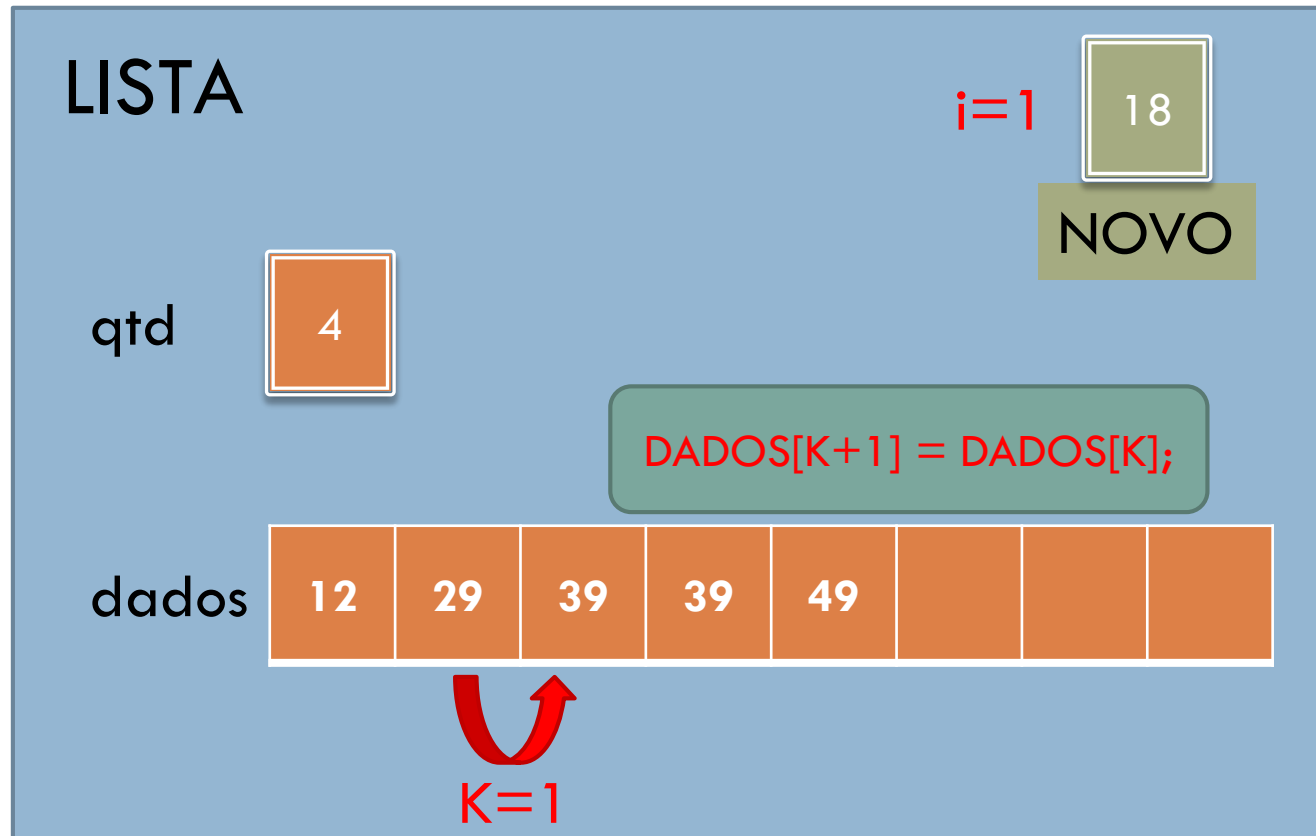
Lista Estática: Exemplo

50



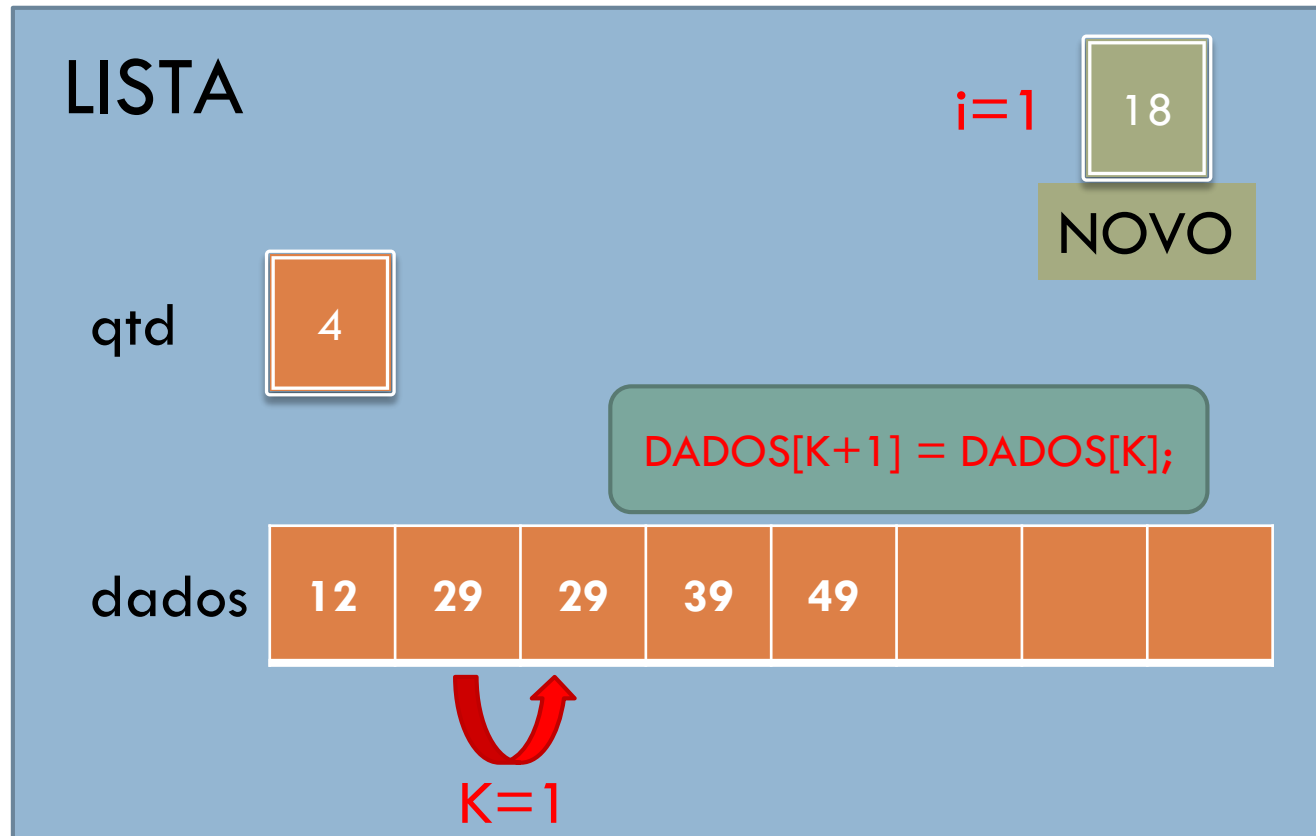
Lista Estática: Exemplo

51



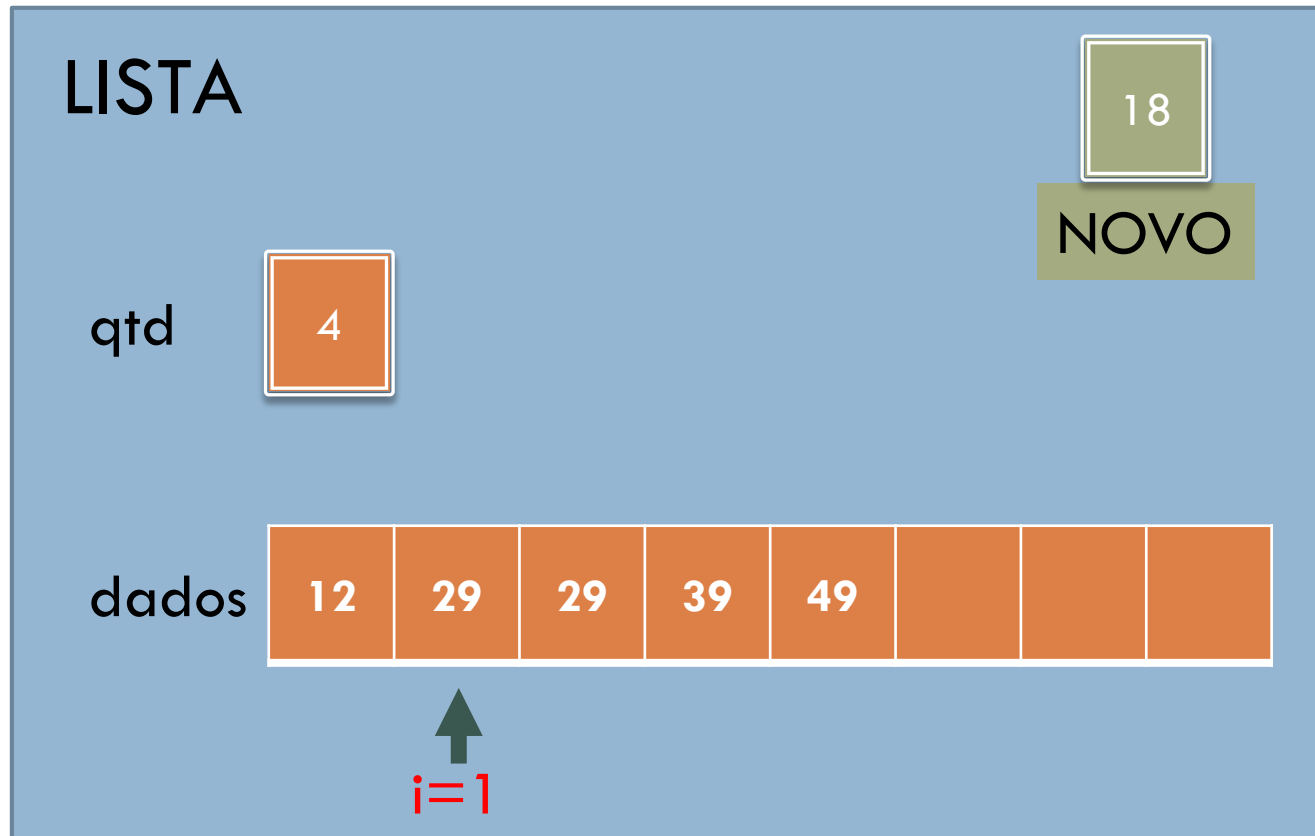
Lista Estática: Exemplo

52



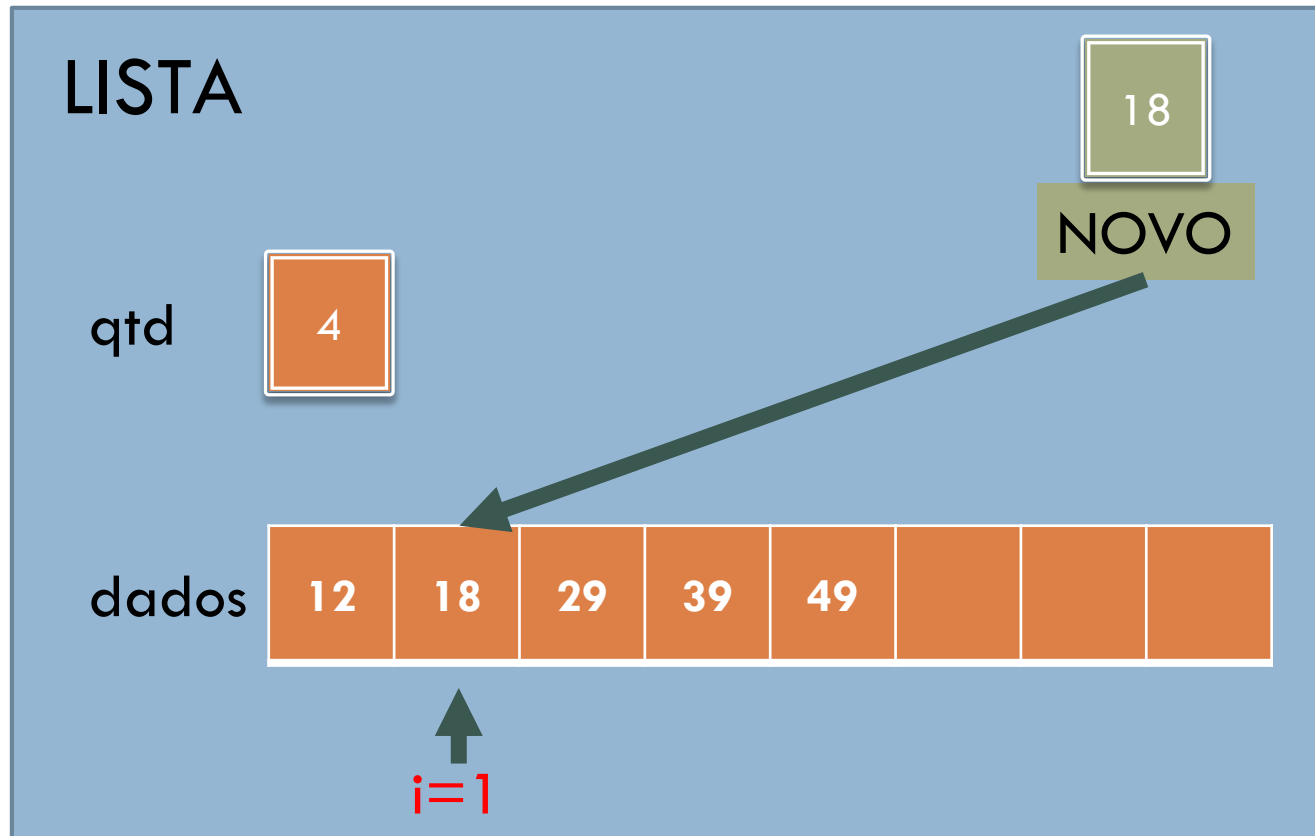
Lista Estática: Exemplo

53



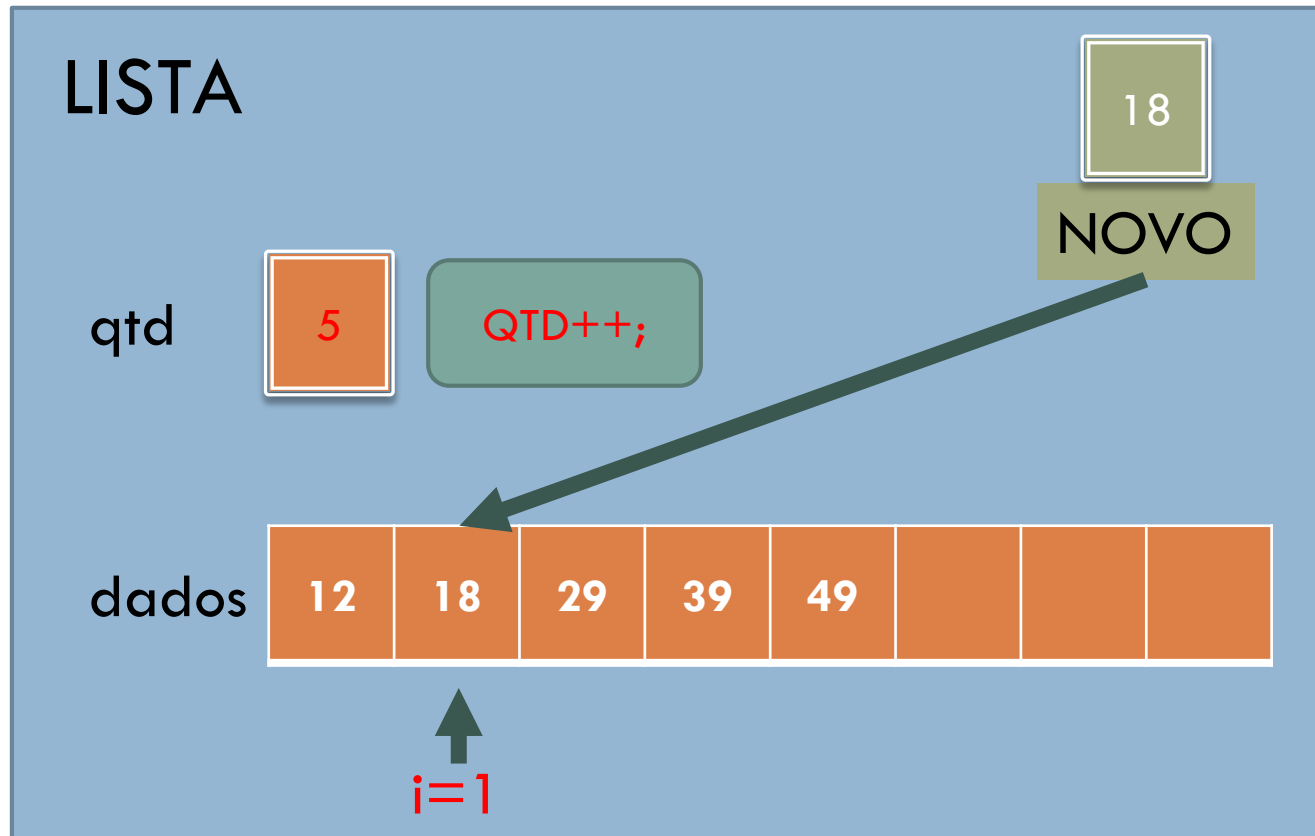
Lista Estática: Exemplo

54



Lista Estática: Exemplo

55



Lista Estática: Exemplo

56

LISTA

qtd

5

dados

12

18

29

39

49

--	--	--	--	--	--	--	--

Lista Estática: Exemplo

57

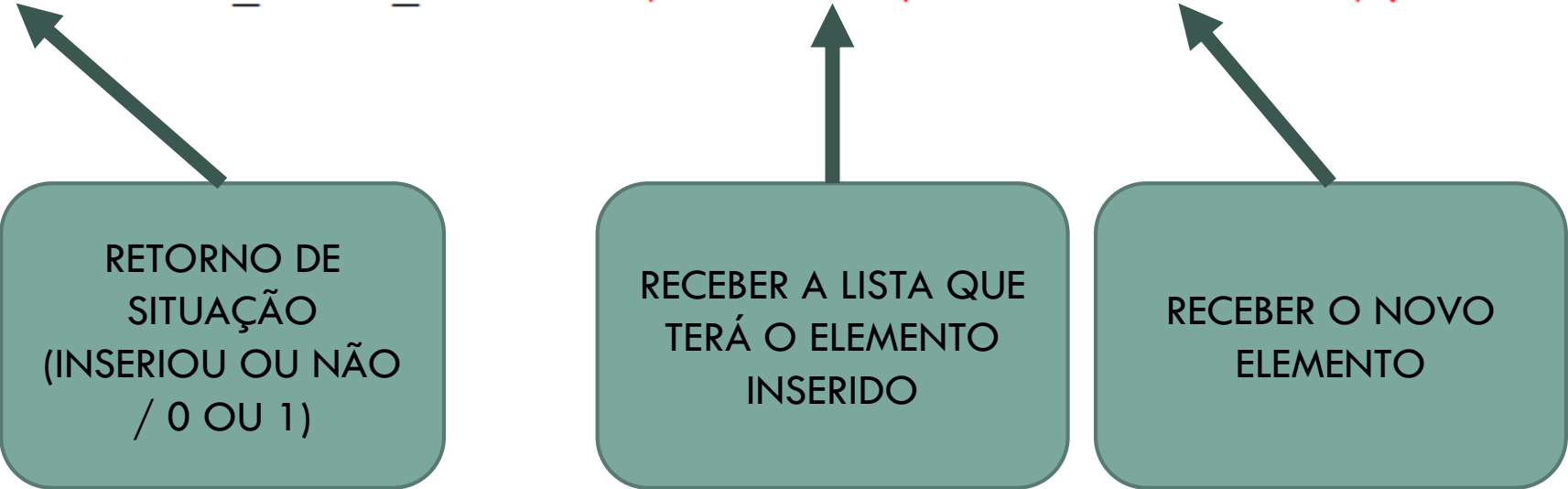
```
int insere_lista_ordenada(Lista* li, struct aluno al){
```

Lista Estática: Exemplo

58

```
int insere_lista_ordenada(Lista* li, struct aluno al){
```

RETORNO DE
SITUAÇÃO
(INSERIOU OU NÃO
/ 0 OU 1)



RECEBER A LISTA QUE
TERÁ O ELEMENTO
INSERIDO

RECEBER O NOVO
ELEMENTO

Lista Estática: Exemplo

59

```
int insere_lista_ordenada(Lista* li, struct aluno al){  
  
    //VERIFICAÇÃO SE A LISTA É NULA OU SE ATINGIU O MÁXIMO  
    if(li == NULL)  
        return 0;  
    if(li->qtd == MAX)  
        return 0;
```

Lista Estática: Exemplo

60

```
int insere_lista_ordenada(Lista* li, struct aluno al){  
  
    //VERIFICAÇÃO SE A LISTA É NULA OU SE ATINGIU O MÁXIMO  
    if(li == NULL)  
        return 0;  
    if(li->qtd == MAX)  
        return 0;  
  
    int k,i = 0;  
    while(i<li->qtd && li->dados[i].matricula < al.matricula)  
        i++;
```

Lista Estática: Exemplo

61

```
int insere_lista_ordenada(Lista* li, struct aluno al){
```

```
//V
```

```
if
```

```
if
```

AUMENTA O VALOR DE **i** **ENQUANTO** NÃO ATINGIR O FINAL DA LISTA E O **VALOR NOVO** MENOR QUE O **VALOR ATUAL** DE **i**



```
int k,i = 0;
```

```
while(i<li->qtd && li->dados[i].matricula < al.matricula)  
    i++;
```

KIMO

Lista Estática: Exemplo

62

```
int insere_lista_ordenada(Lista* li, struct aluno al){  
  
    //VERIFICAÇÃO SE A LISTA É NULA OU SE ATINGIU O MÁXIMO  
    if(li == NULL)  
        return 0;  
    if(li->qtd == MAX)  
        return 0;  
  
    int k,i = 0;  
    while(i<li->qtd && li->dados[i].matricula < al.matricula)  
        i++;
```



A VARIÁVEL **i** ARMAZENA O **ÍNDICE** IDEAL PARA **INSERIR** O NOVO VALOR, MANTENDO A ORDEM CRESCENTE

Lista Estática: Exemplo

63

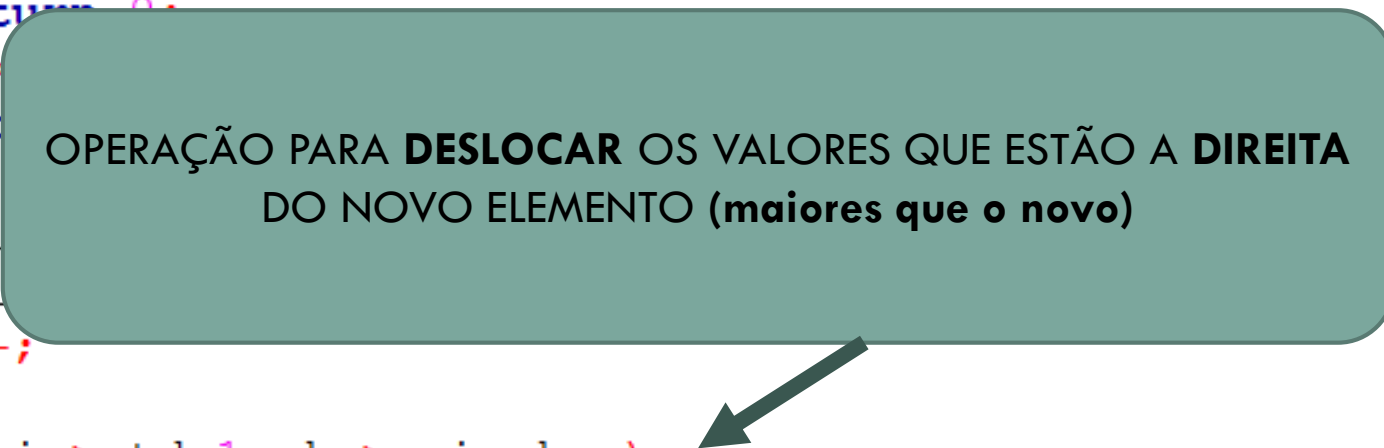
```
int insere_lista_ordenada(Lista* li, struct aluno al){  
  
    //VERIFICAÇÃO SE A LISTA É NULA OU SE ATINGIU O MÁXIMO  
    if(li == NULL)  
        return 0;  
    if(li->qtd == MAX)  
        return 0;  
  
    int k,i = 0;  
    while(i<li->qtd && li->dados[i].matricula < al.matricula)  
        i++;  
  
    for(k=li->qtd-1; k >= i; k--)  
        li->dados[k+1] = li->dados[k];
```

Lista Estática: Exemplo

64

```
int insere_lista_ordenada(Lista* li, struct aluno al){  
    //VERIFICAÇÃO SE A LISTA É NULA OU SE ATINGIU O MÁXIMO  
    if(li == NULL)  
        return 0;  
    if(li->qtde == MAXIMO)  
        return 0;  
    int k, i;  
    while(i < li->qtde - 1 && li->dados[i] < al.dados[0])  
        i++;  
    for(k=li->qtde-1; k >= i; k--)  
        li->dados[k+1] = li->dados[k];  
    li->dados[i] = al;  
    li->qtde++;  
    return 1;  
}
```

OPERAÇÃO PARA **DESLOCAR** OS VALORES QUE ESTÃO A **DIREITA** DO NOVO ELEMENTO (**maiores que o novo**)

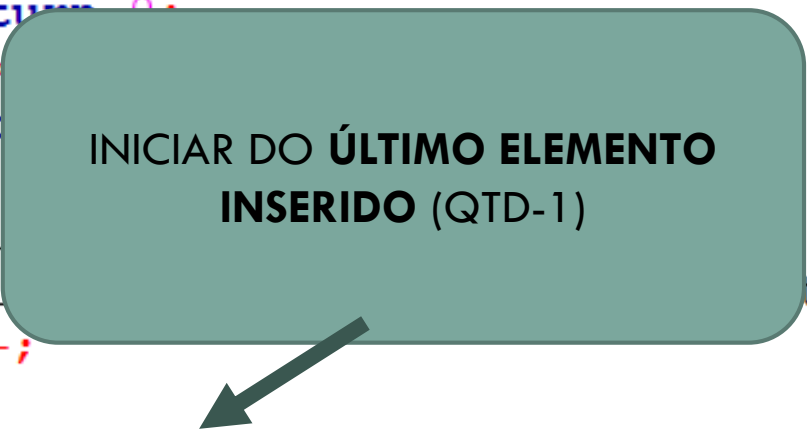


Lista Estática: Exemplo

65

```
int insere_lista_ordenada(Lista* li, struct aluno al){  
    //VERIFICAÇÃO SE A LISTA É NULA OU SE ATINGIU O MÁXIMO  
    if(li == NULL)  
        return 0;  
    if(li->qtde == MAXIMO)  
        return 0;  
    int k, i;  
    while(i < li->qtde & al.matricula < al.matricula){  
        i++;  
    }  
    for(k=li->qtde-1; k >= i; k--)  
        li->dados[k+1] = li->dados[k];
```

INICIAR DO ÚLTIMO ELEMENTO
INSERIDO (QTD-1)

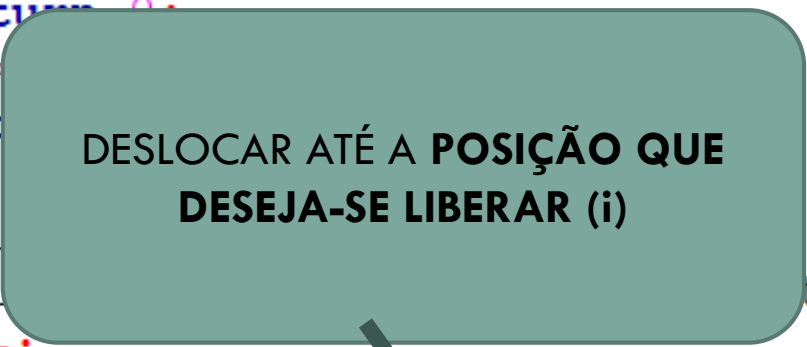


Lista Estática: Exemplo

66

```
int insere_lista_ordenada(Lista* li, struct aluno al){  
    //VERIFICAÇÃO SE A LISTA É NULA OU SE ATINGIU O MÁXIMO  
    if(li == NULL)  
        return 0;  
    if(li->qtde == MAXIMO)  
        return 0;  
    int k, i;  
    while(i < li->qtde & al.matricula < al.matricula)  
        i++;  
    for(k=li->qtde-1; k >= i; k--)  
        li->dados[k+1] = li->dados[k];
```

DESLOCAR ATÉ A POSIÇÃO QUE
DESEJA-SE LIBERAR (i)



Lista Estática: Exemplo

67

```
int insere_lista_ordenada(Lista* li, struct aluno al){  
  
    //VERIFICAÇÃO SE A LISTA É NULA OU SE ATINGIU O MÁXIMO  
    if(li == NULL)  
        return 0;  
    if(li->qtd == MAX)  
        return 0;  
  
    int k,i = 0;  
    while(i<li->qtd && li->dados[i].matricula < al.matricula)  
        i++;  
  
    for(k=li->qtd-1; k >= i; k--)  
        li->dados[k+1] = li->dados[k];  
  
    li->dados[i] = al;  
    li->qtd++;  
    return 1;  
}
```

Lista Estática: Exemplo

68

```
int insere_lista_ordenada(Lista* li, struct aluno al){  
  
    //VERIFICAÇÃO SE A LISTA É NULA OU SE ATINGIU O MÁXIMO  
    if(li == NULL)  
        return 0;  
    if(li->qtd == MAX)  
        return 0;  
  
    int k,i = 0;  
    while(i<li->qtd && li->dados[i].matricula < al.matricula)  
        i++;  
  
    for(k=li->qtd-1; k >= i; k--)  
        li->dados[k+1] = li->dados[k];  
  
    li->dados[i] = al;  
    li->qtd++;  
    return 1;  
}
```

INSERIR NOVO ELEMENTO NA
POSIÇÃO IDEAL;

INCREMENTAR QUANTIDADE DA LISTA

Lista Estática: Exemplo

69

```
int insere_lista_ordenada(Lista* li, struct aluno al){
```

```
//Declarando os dados estaticamente
//Iniciando os valores na ordem de descrição da estrutura
//OBS: Você pode substituir essa etapa por uma leitura de valores do teclado
struct aluno a[4] = {{1, "Andre", 9.5, 7.8, 8.5},
                     {3, "Ricardo", 7.5, 8.7, 6.8},
                     {0, "Bianca", 9.7, 6.7, 8.4},
                     {2, "Ana", 5.7, 6.1, 7.4}};

Lista* li = cria_lista();// Criando o bloco inicial da lista
                        // Armazena a quantidade e o inicio da lista

int i;
for(i=0; i < 4; i++)
    insere_lista_ordenada(li, a[i]);
//OBS: uma leitura de dados pode ser aplicada aqui
    li->dados[i] = al;
    li->qtd++;
    return 1;
}
```

Lista Estática: Exemplo

70

- **Remoção:** remover elementos de acordo com a chave passada.
- Procurar a chave na lista (varredura) e **substituir pelo último** elemento da lista.
 - ▣ Não mantém a ordem da inserção.
 - ▣ É mais rápido na execução.

Lista Estática: Exemplo

71

LISTA

qtd

5

dados

12

18

29

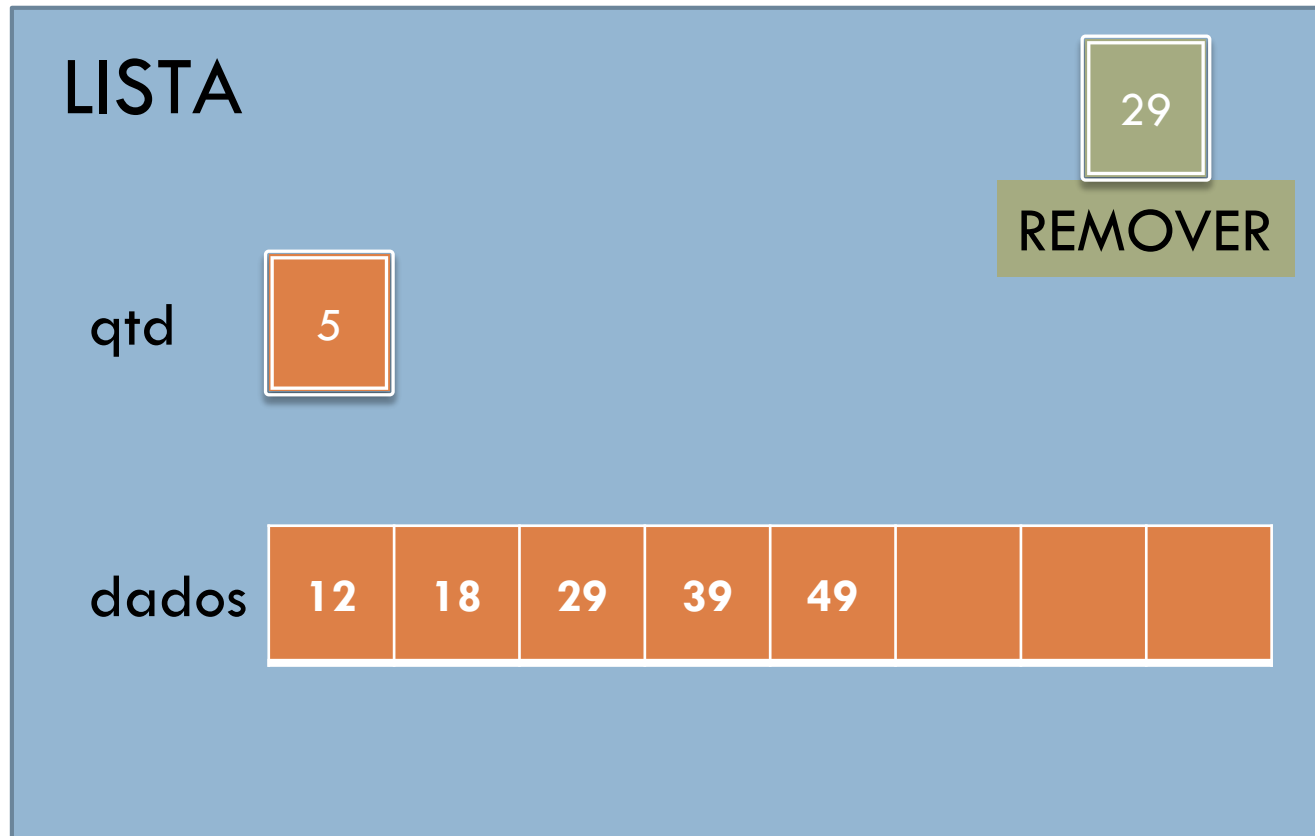
39

49

--	--	--	--	--	--	--	--

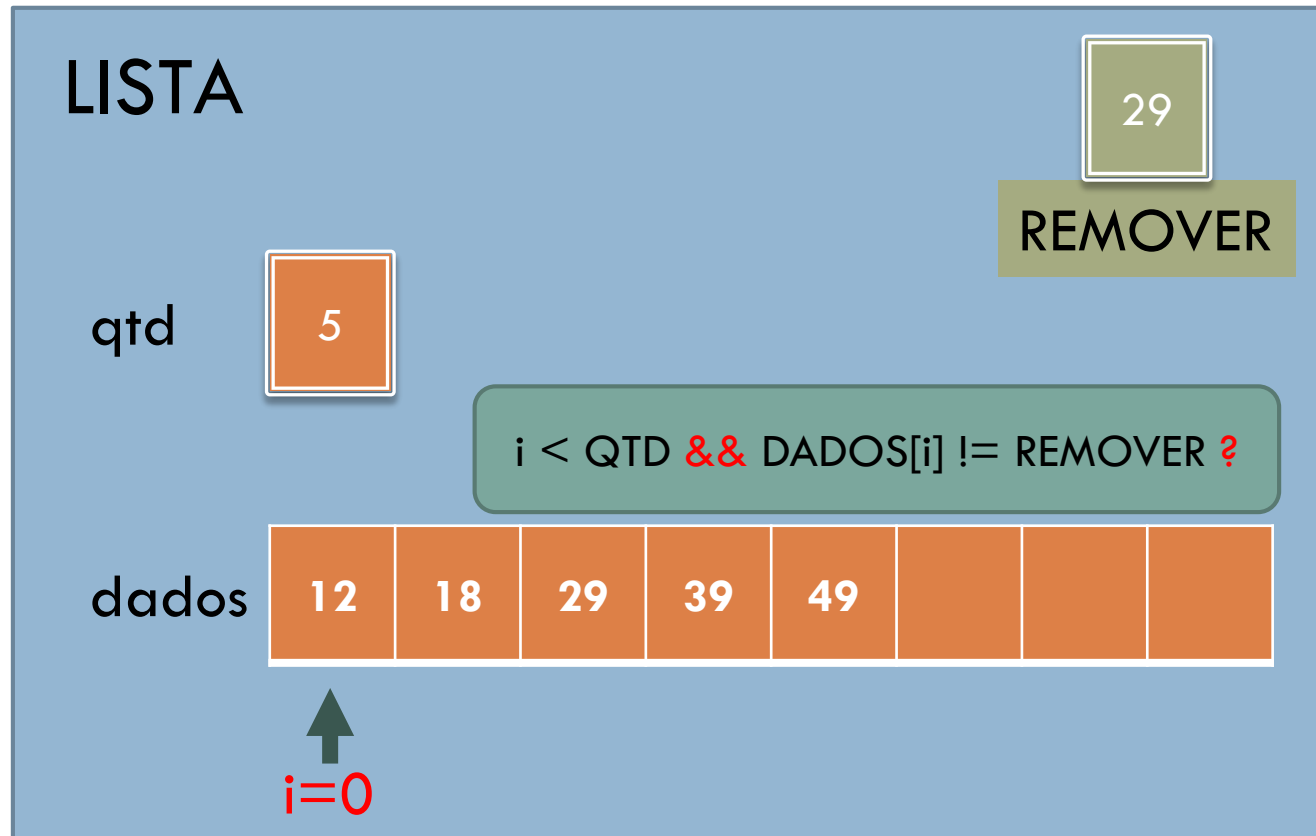
Lista Estática: Exemplo

72



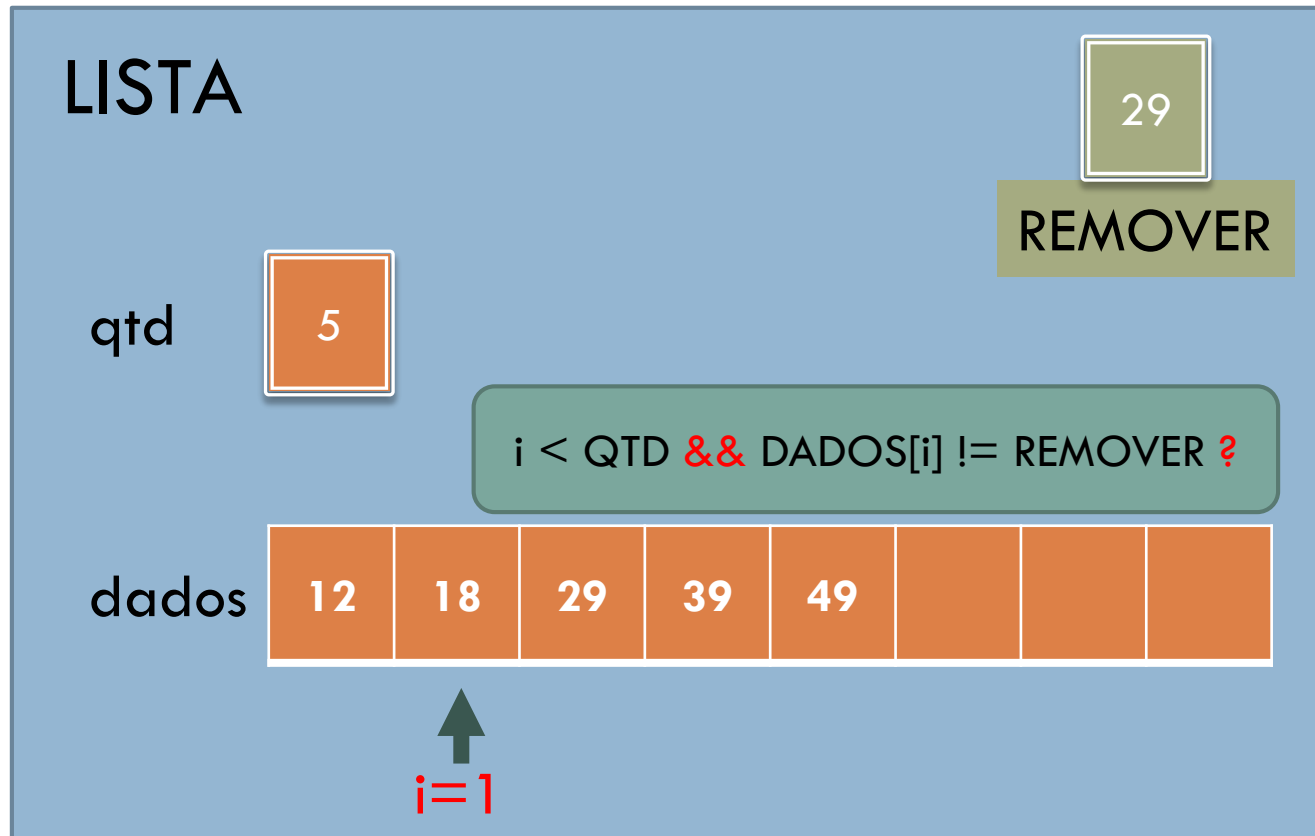
Lista Estática: Exemplo

73



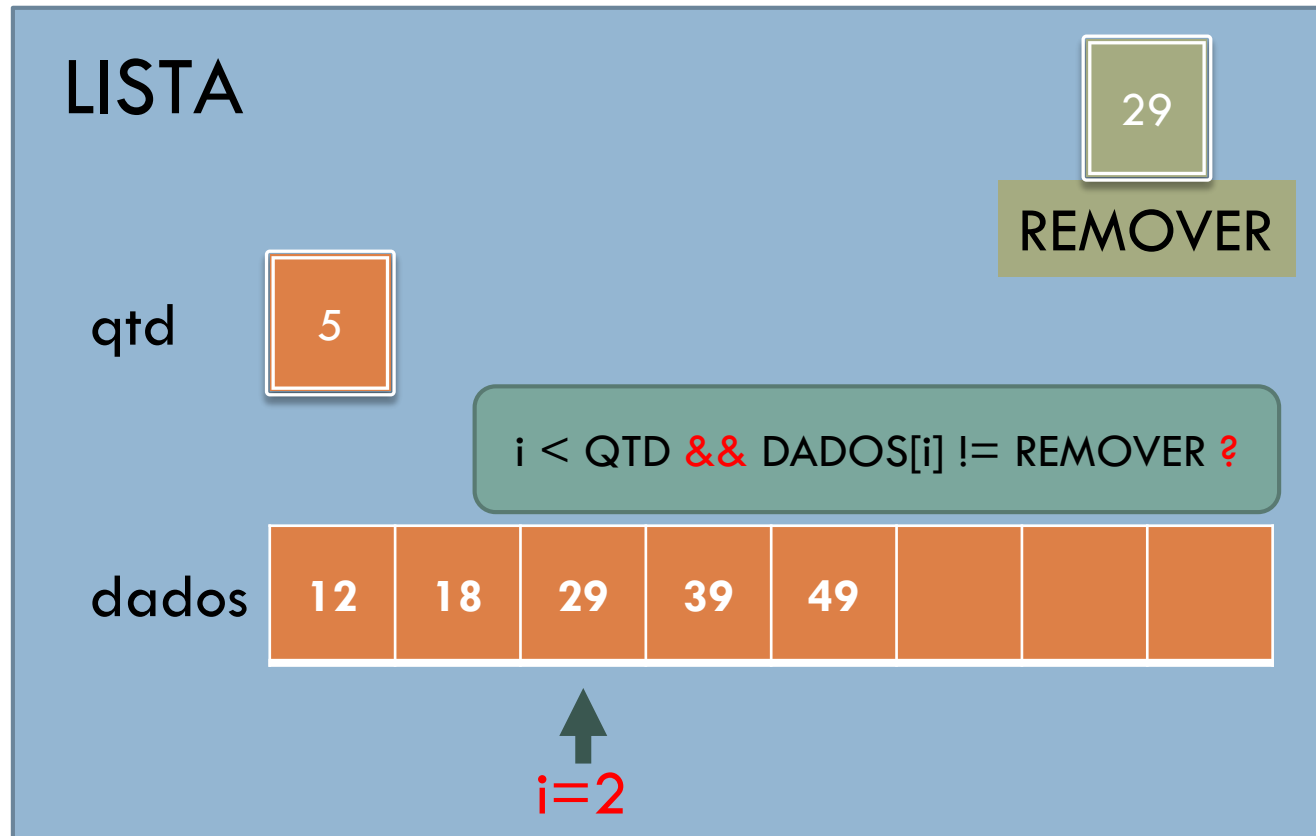
Lista Estática: Exemplo

74



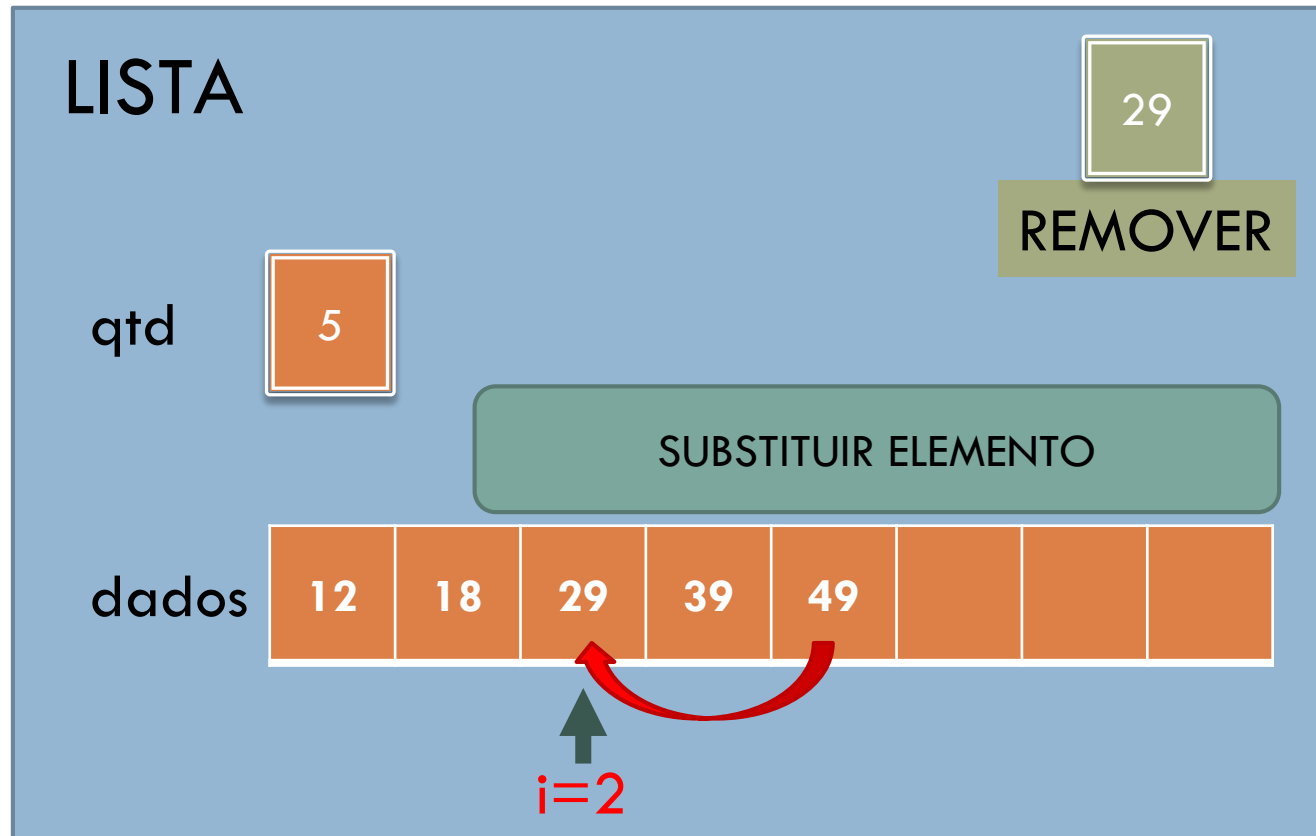
Lista Estática: Exemplo

75



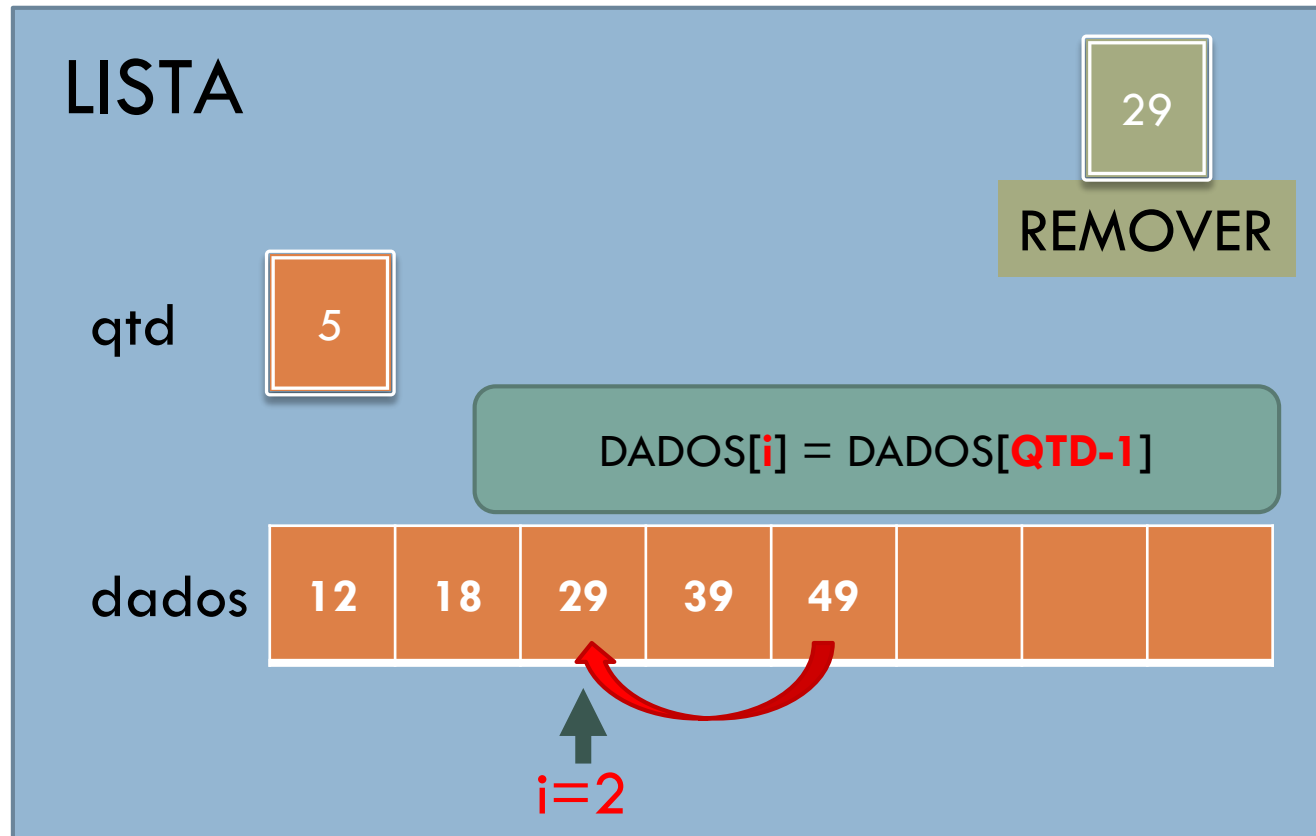
Lista Estática: Exemplo

76



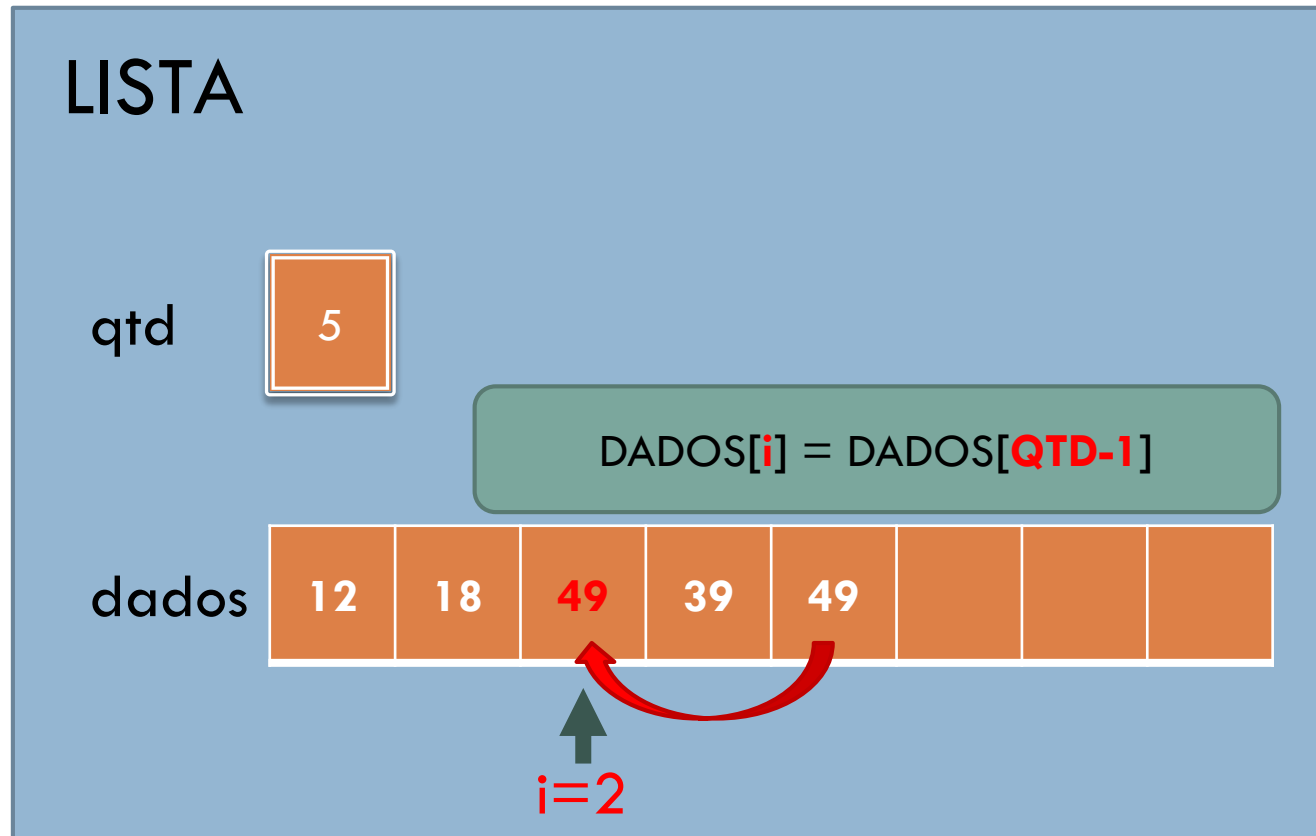
Lista Estática: Exemplo

77



Lista Estática: Exemplo

78



Lista Estática: Exemplo

79

LISTA

qtd

4

QTD--;

dados

12

18

49

39

49

--	--	--	--	--	--	--

Lista Estática: Exemplo

80

LISTA

qtd

4

POSIÇÃO LIVRE /
DESCONSIDERA O
ELEMENTO

dados

12

18

49

39

49



Lista Estática: Exemplo

81

LISTA

qtd

4

dados

12

18

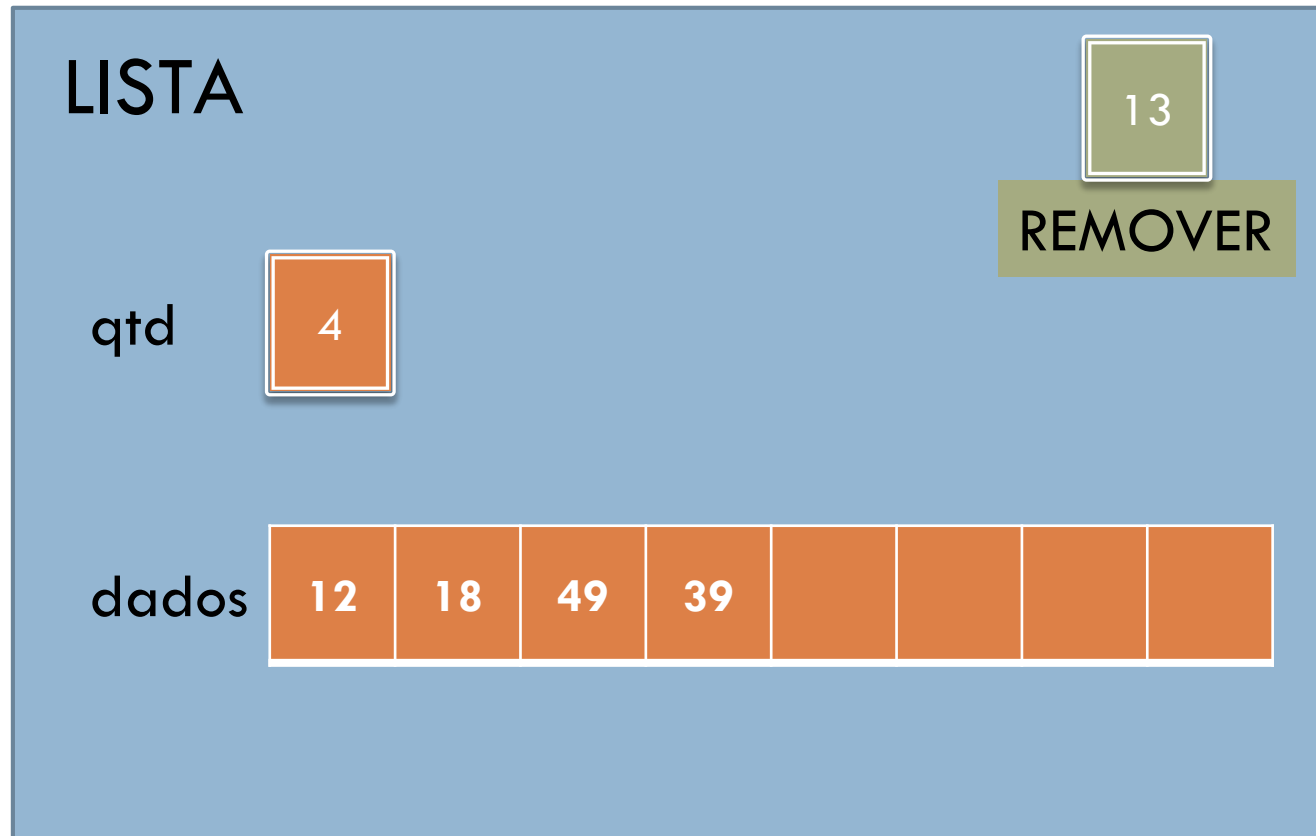
49

39

--	--	--	--	--	--	--	--

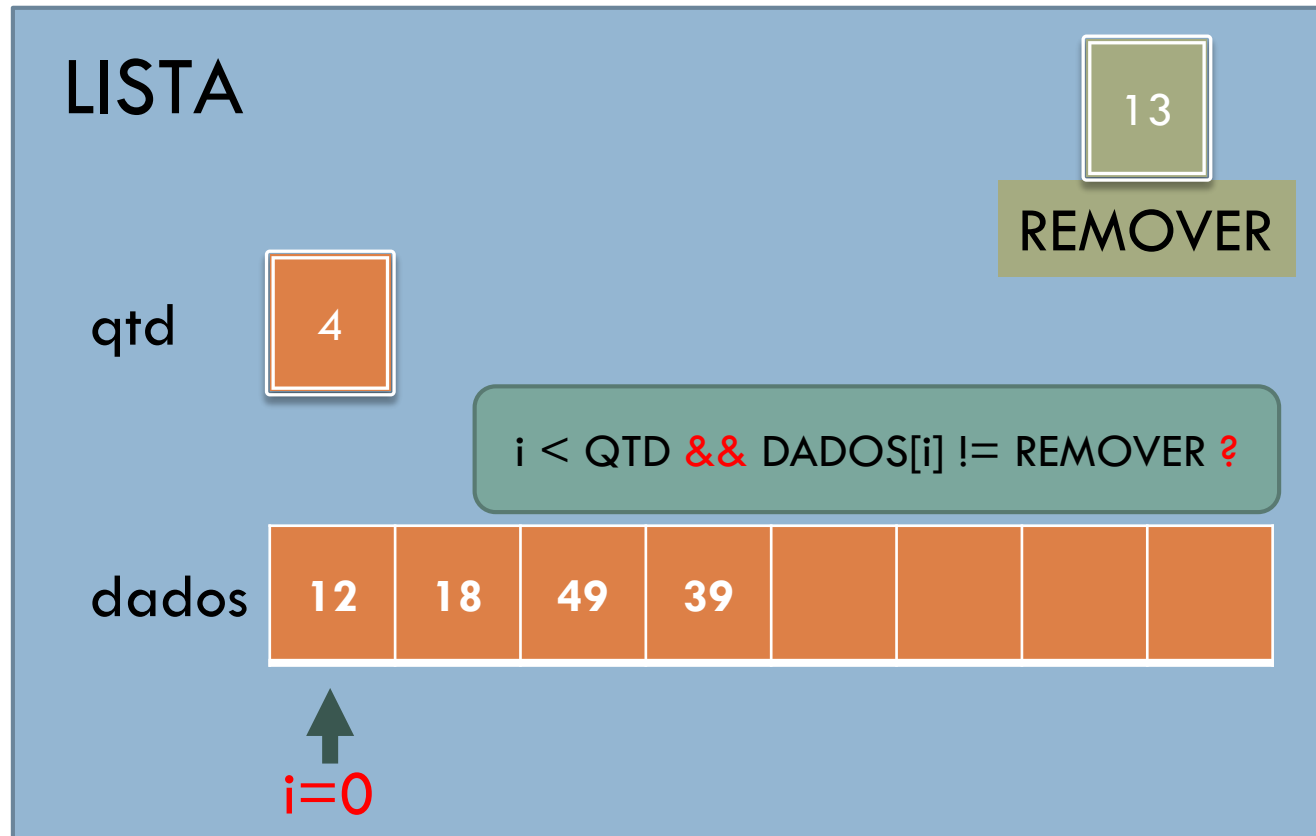
Lista Estática: Exemplo

82



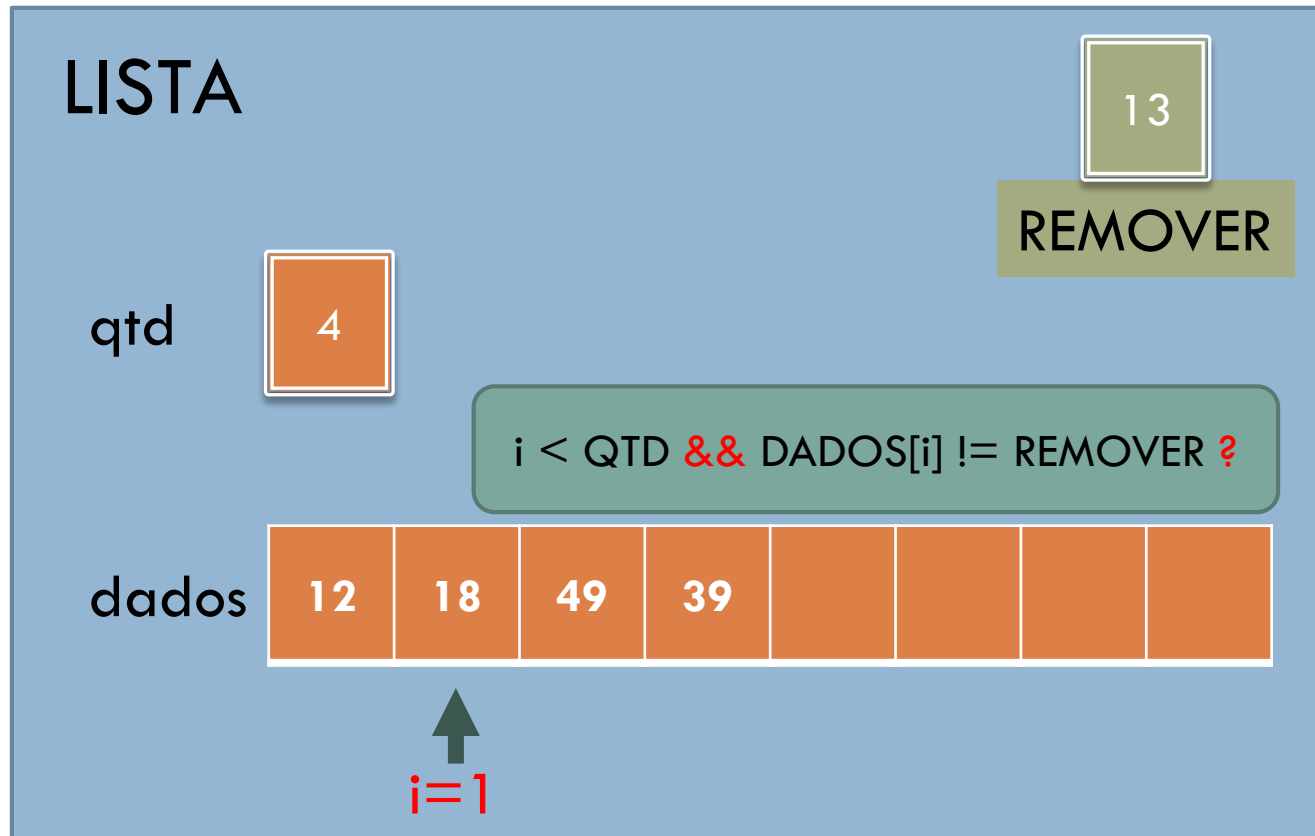
Lista Estática: Exemplo

83



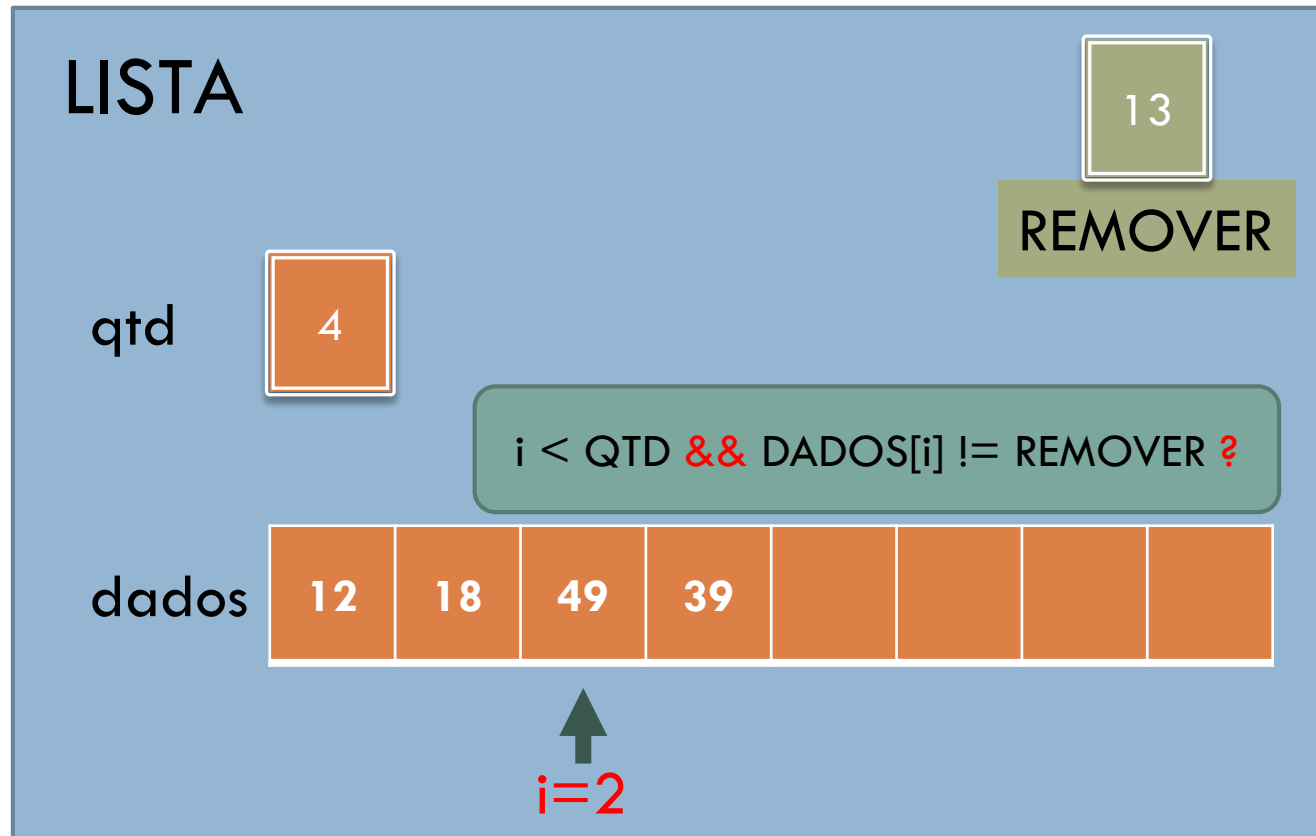
Lista Estática: Exemplo

84



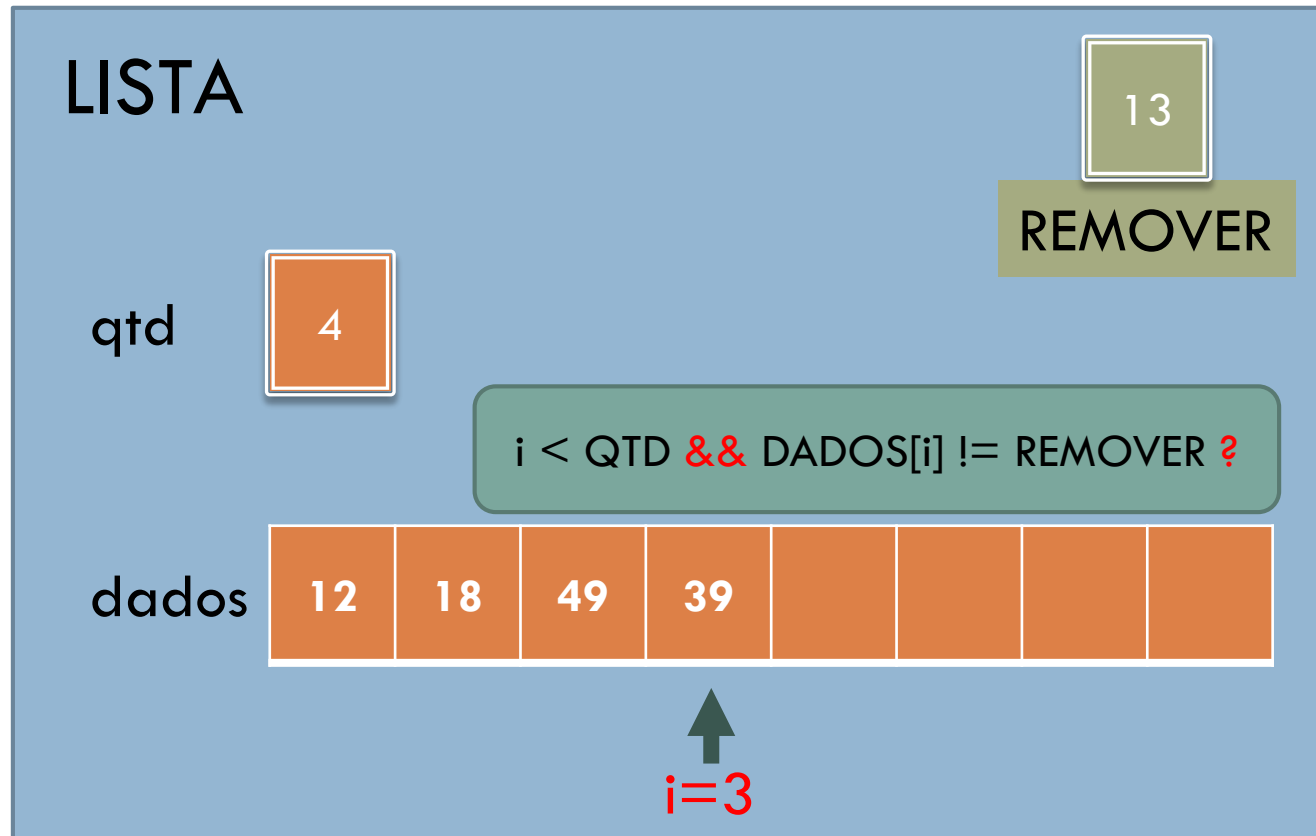
Lista Estática: Exemplo

85



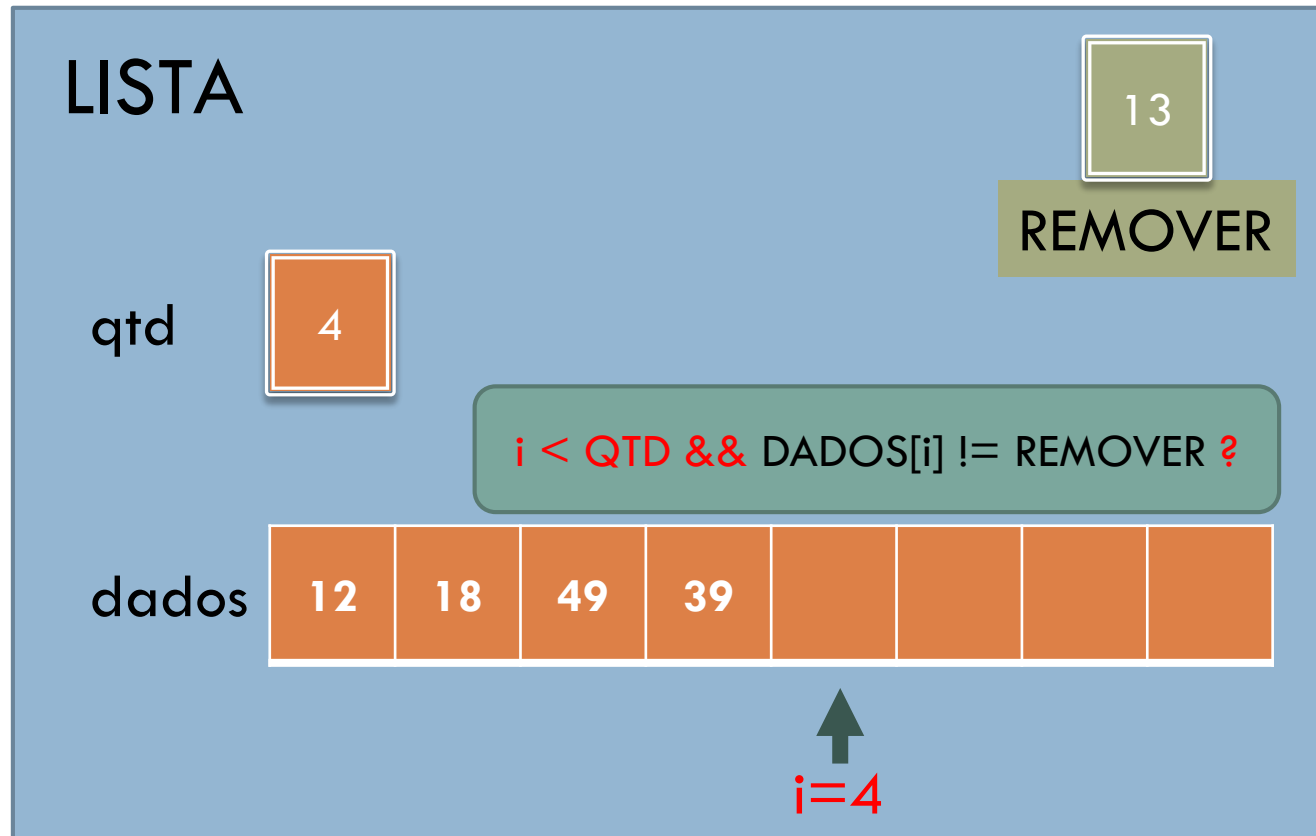
Lista Estática: Exemplo

86



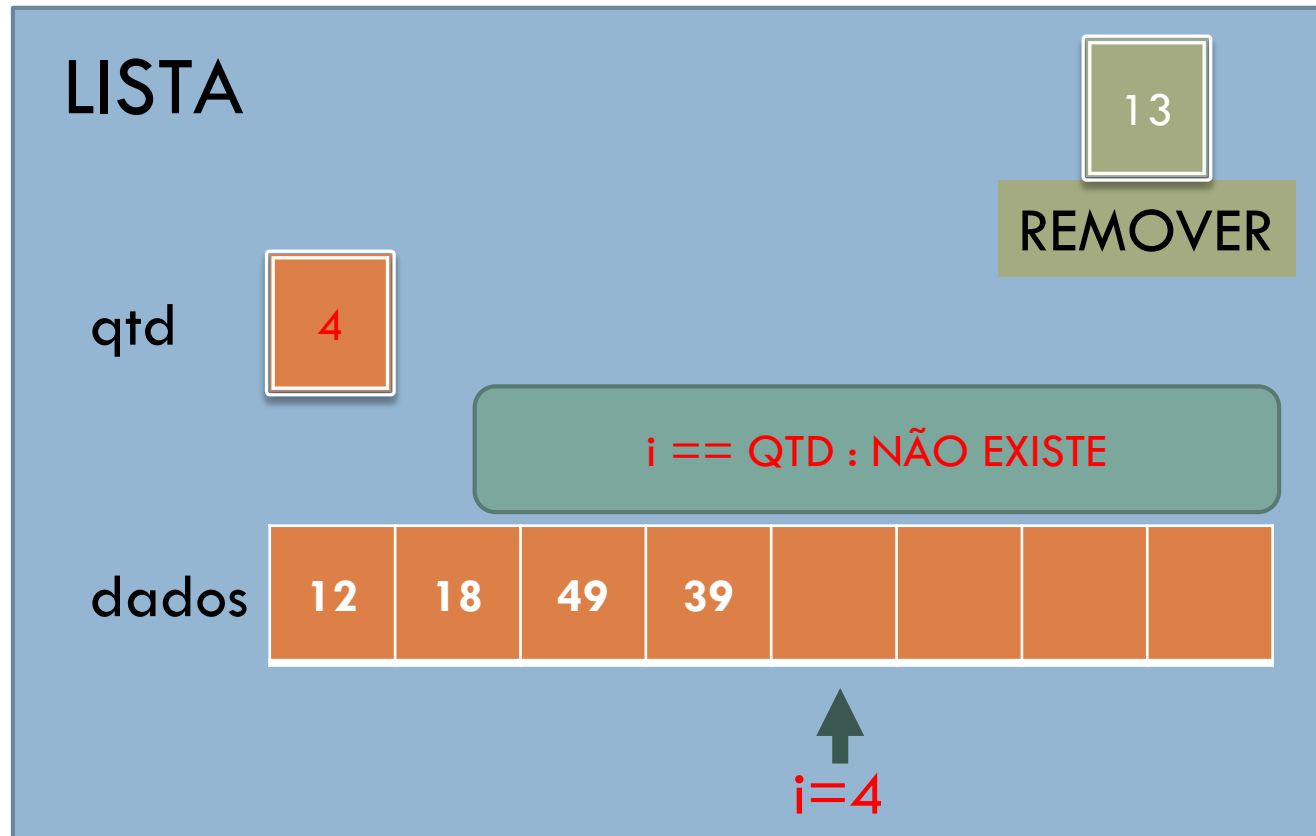
Lista Estática: Exemplo

87



Lista Estática: Exemplo

88

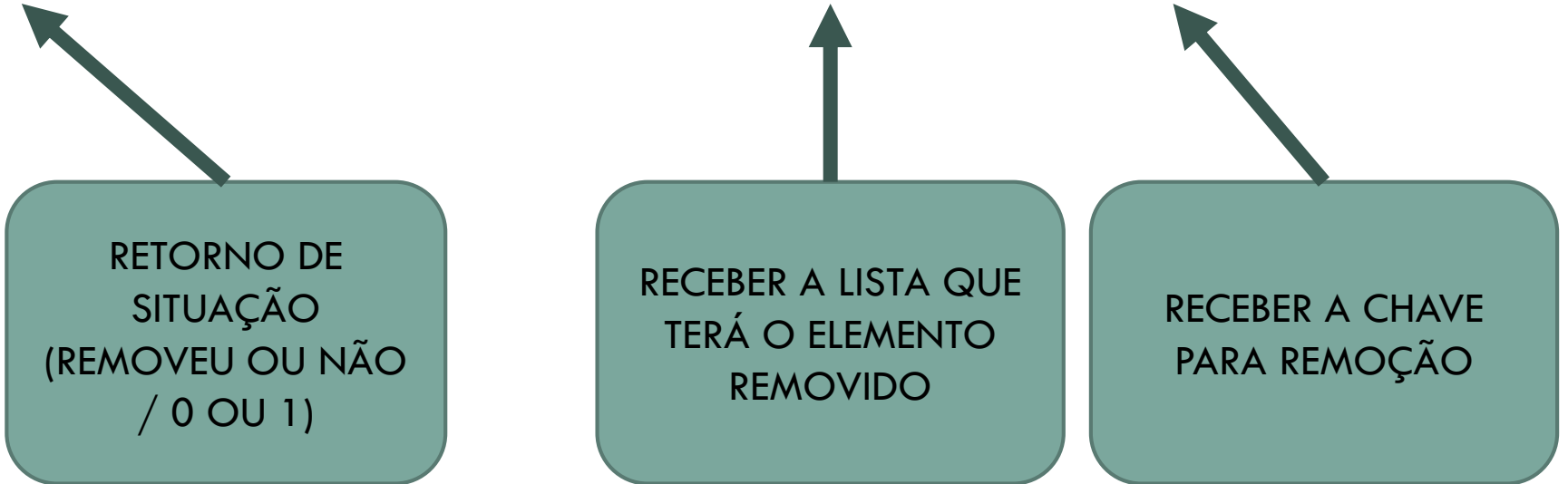


Lista Estática: Exemplo

89

```
int remove_lista_otimizado(Lista* li, int mat){
```

RETORNO DE
SITUAÇÃO
(REMOVEU OU NÃO
/ 0 OU 1)



```
graph BT; A[RETORNO DE SITUAÇÃO (REMOVEU OU NÃO / 0 OU 1)] --> B[int]; C[RECEBER A LISTA QUE TERÁ O ELEMENTO REMOVIDO] --> D[Lista*]; E[RECEBER A CHAVE PARA REMOÇÃO] --> F[int]; B --> G[remove_lista_otimizado(Lista* li, int mat)]; D --> G; F --> G;
```


RECEBER A LISTA QUE
TERÁ O ELEMENTO
REMOVIDO

RECEBER A CHAVE
PARA REMOÇÃO

Lista Estática: Exemplo

90

```
int remove_lista_otimizado(Lista* li, int mat){  
    if(li == NULL)  
        return 0;  
    if(li->qtd == 0)  
        return 0;
```



VERIFICAÇÃO SE A
LISTA É NULA OU SE
ESTÁ VAZIA

Lista Estática: Exemplo

91

```
int remove_lista_otimizado(Lista* li, int mat){  
    if(li == NULL)  
        return 0;  
    if(li->qtd == 0)  
        return 0;  
    int i = 0;  
    while(i < li->qtd && li->dados[i].matricula != mat)  
        i++;
```



AUMENTA O VALOR DE **i** **ENQUANTO** NÃO ATINGIR O FINAL DA
LISTA E A CHAVE NÃO FOR ENCONTRADA

Lista Estática: Exemplo

92

```
int remove_lista_otimizado(Lista* li, int mat){  
    if(li == NULL)  
        return 0;  
    if(li->qtd == 0)  
        return 0;  
    int i = 0;  
    while(i < li->qtd && li->dados[i].matricula != mat)  
        i++;
```




A VARIÁVEL **i** ARMAZENA O **ÍNDICE** (posição) DO VALOR QUE
SERÁ REMOVIDO

Lista Estática: Exemplo

93

```
int remove_lista_otimizado(Lista* li, int mat){  
    if(li == NULL)  
        return 0;  
    if(li->qtd == 0)  
        return 0;  
    int i = 0;  
    while(i < li->qtd && li->dados[i].matricula != mat)  
        i++;  
    if(i == li->qtd) //elemento nao encontrado  
        return 0;
```



VARREU TODA A
LISTA E NÃO
ENCONTROU

Lista Estática: Exemplo

94

```
int remove_lista_otimizado(Lista* li, int mat){
    if(li == NULL)
        return 0;
    if(li->qtd == 0)
        return 0;
    int i = 0;
    while(i < li->qtd && li->dados[i].matricula != mat)
        i++;
    if(i == li->qtd) //elemento nao encontrado
        return 0;

    li->qtd--;
    li->dados[i] = li->dados[li->qtd]; // Preenche espaço
    return 1;                          // com o último elemento
}
```

Lista Estática: Exemplo

95

```
int remove_lista_otimizado(Lista* li, int mat){
    if(li == NULL)
        return 0;
    i = 0;
    // Caso nao encontrou o elemento, retorna 0
    if(!remove_lista_otimizado(li, mat))
        printf("Erro\n");
    if(i == li->qtd) //elemento nao encontrado
        return 0;

    li->qtd--;
    li->dados[i] = li->dados[li->qtd]; // Preenche espaço
    return 1;                          // com o último elemento
}
```

Lista Estática: Exemplo

96

- **Impressão:** apresentar na tela todos os dados dos elementos da lista.
- Varredura e impressão de cada elemento do vetor de dados..

Lista Estática: Exemplo

97

```
void imprime_lista(Lista* li) {
    if (li == NULL)
        return;
    int i;
    for (i=0; i< li->qtd; i++) {
        printf("Matricula: %d\n", li->dados[i].matricula);
        printf("Nome: %s\n", li->dados[i].nome);
        printf("Notas: %f %f %f\n", li->dados[i].n1,
                                                    li->dados[i].n2,
                                                    li->dados[i].n3);
        printf("-----\n");
    }
}
```

Lista Estática

98

□ Vantagens:

- ▣ Acesso direto pela posição (índice do vetor);
- ▣ Tempo constante.
- ▣ Possibilita busca binária (se a lista estiver ordenada).

□ Desvantagens:

- ▣ Movimento de dados na Inserção e Exclusão.
- ▣ Tamanho máximo deve ser determinado na compilação.
- ▣ Risco de overflow.

Lista Estática: Exercícios

99

- Baseado na lista estática do exemplo, faça as implementações necessárias para adicionar as seguintes operações:
 - ▣ Descobrir se a lista está cheia (retornar);
 - ▣ Descobrir se a lista está vazia (retornar);
 - ▣ Descobrir o tamanho da lista (retornar);
 - ▣ Inserir elemento no final da lista;
 - ▣ Inserir elemento no início da lista;
 - ▣ Remover elemento do final da lista;
 - ▣ Removei elemento do início da lista;

Lista Estática: Exercícios (cont.)

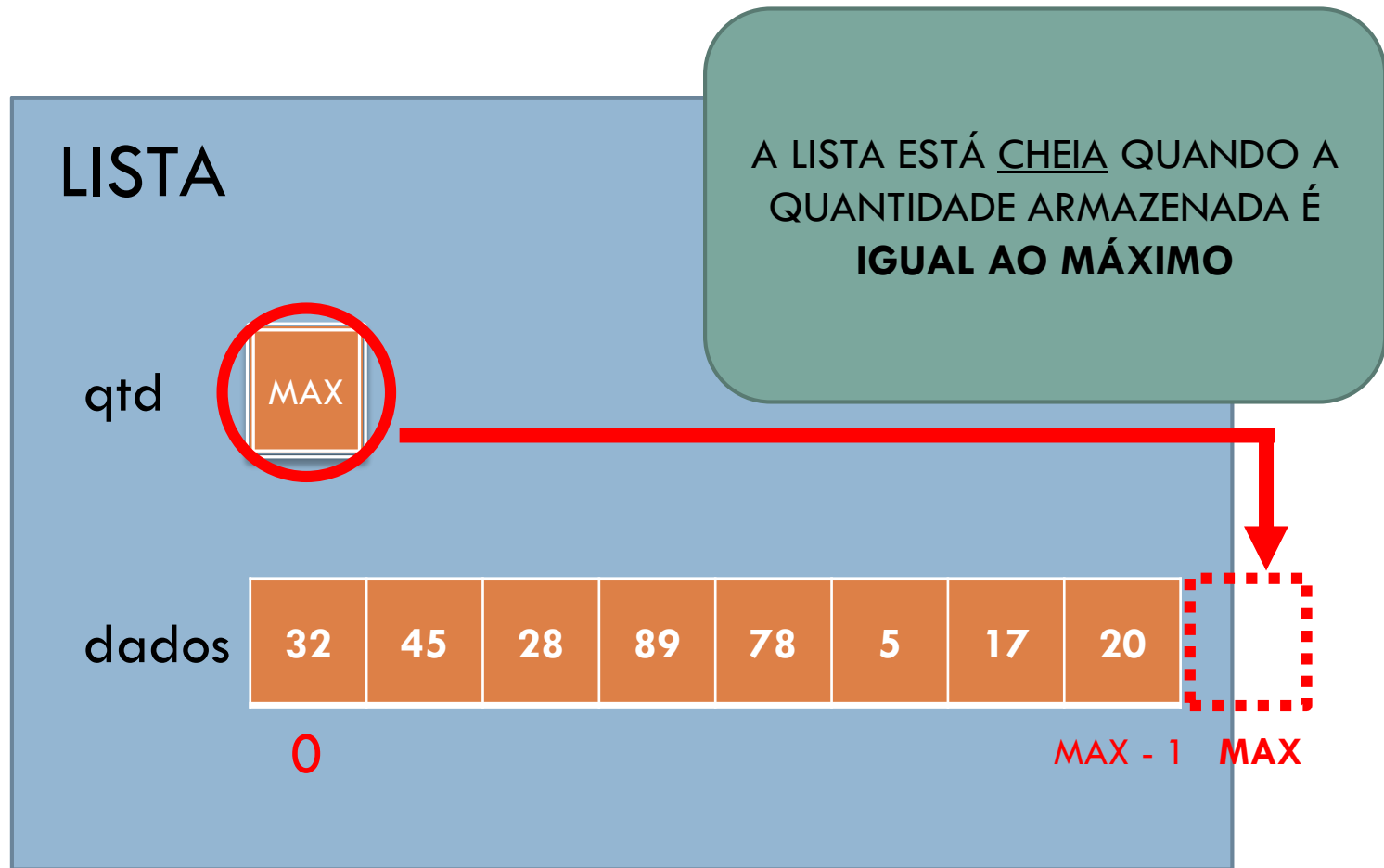
100

- Baseado na lista estática do exemplo, faça as implementações necessárias para adicionar as seguintes operações:
 - ▣ Remover elemento mantendo a ordenação;
 - ▣ Consultar um elemento:
 - Passe como parâmetros **a lista, a matrícula que será buscada e um ponteiro do elemento.**
 - Não retorne o elemento encontrado pelo retorno da função, armazene o endereço no ponteiro do parâmetro.

```
int consulta(Lista* li, int mat, struct aluno *al)
```

Lista Estática: Exercícios

101



Lista Estática: Exercícios

102

LISTA

qtd

MAX

dados

32

45

28

89

78

5

17

20

0

MAX - 1 MAX

```
int lista_cheia(Lista* li){  
    if(li == NULL)  
        return -1;  
    return (li->qtd == MAX);  
}
```