

# Melhores momentos

## AULA 6

## Listas em vetor

Manter uma **lista** em um vetor sujeita a **remoções** e **inserções** pode dar muito trabalho com **movimentações**.

Veremos uma maneira alternativa que pode dar **menos trabalho** com **movimentações**, se estivermos dispostos a gastar um pouco **mais de espaço**.

# Listas encadeadas em vetores

0	Carlos	1
1	Eduardo	2
2	Helio	3
3	Joao	4
4	Luiz	5
5	Maria	6
6	Rui	7
7	Sergio	-1
8		9
9		10
10		-1

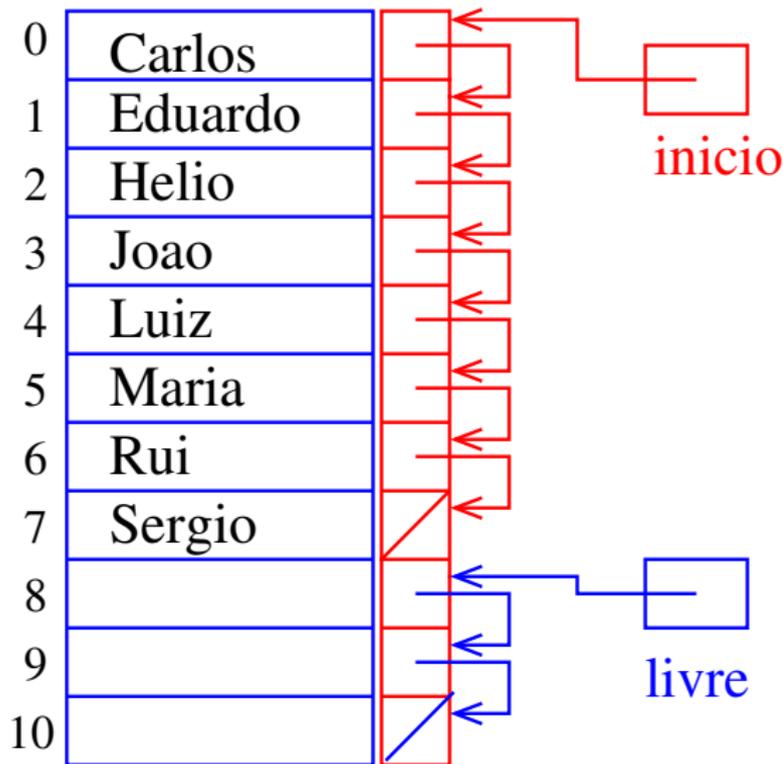
0

inicio

8

livre

# Listas encadeadas em vetores



## Estrutura de uma lista encadeada em vetor

Uma **lista encadeada** (= *linked list* = lista ligada) é uma sequência de **células**; cada **célula** contém um **objeto** de algum tipo e o **endereço** da célula seguinte.

```
struct celula {
    int conteudo;
    int prox;
};
typedef struct celula Celula;

Celula v[MAX];
int inicio;
int livre;
```

## Imprime conteúdo de uma lista

Esta função recebe o índice início de uma lista encadeada em um vetor e imprime os elementos da lista.

```
#define NULO -1
void imprime(int inicio, Celula v[]) {
    int p;
    for (p = inicio; p != NULO; p = v[p].prox)
        printf("%d ", v[p].conteudo);
    printf("\n");
}
```

# AULA 7

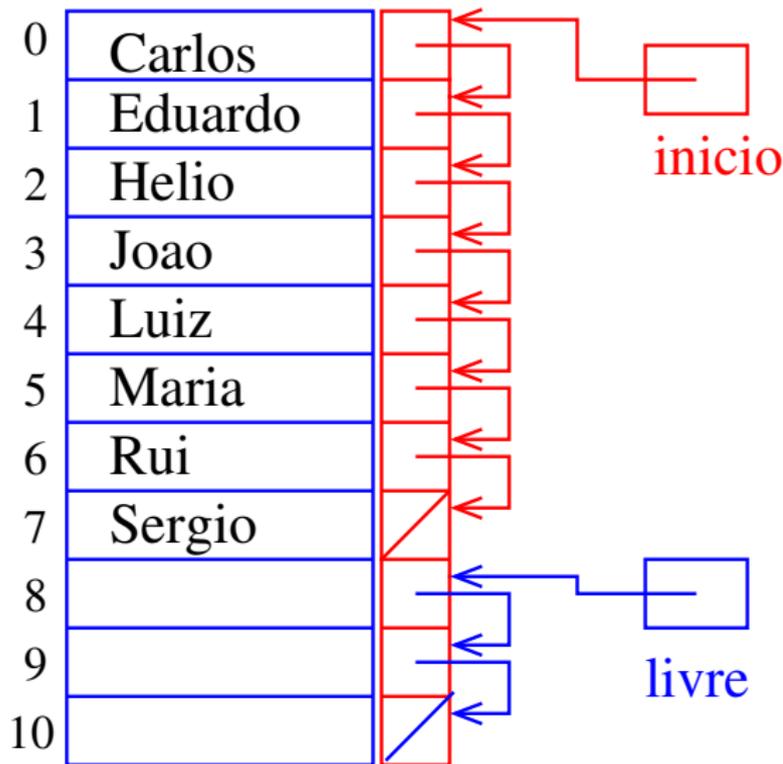
# Hoje

- ▶ listas encadeadas em C

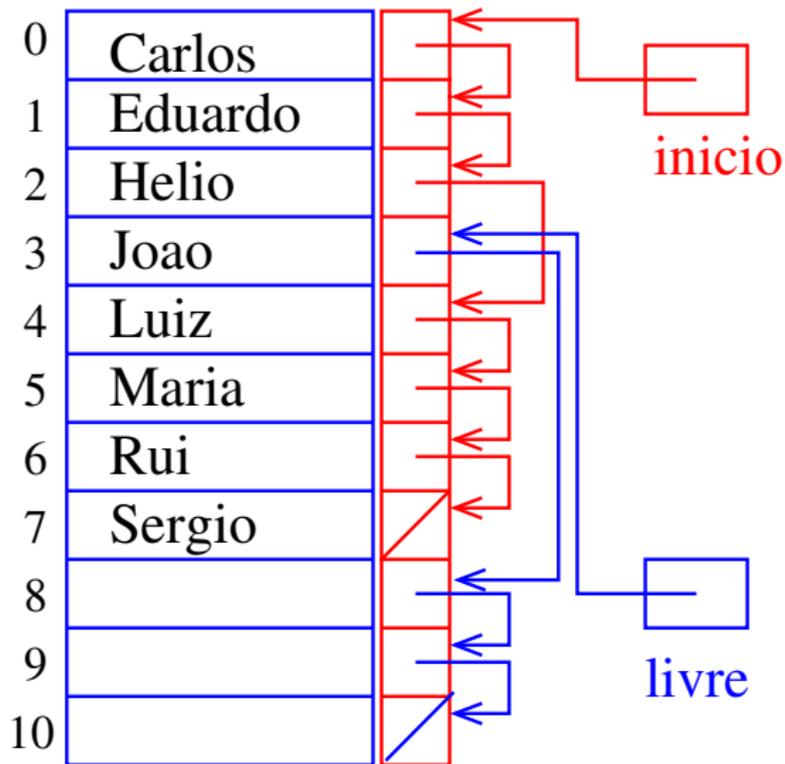


Fonte: <http://onlinefungags.com/>

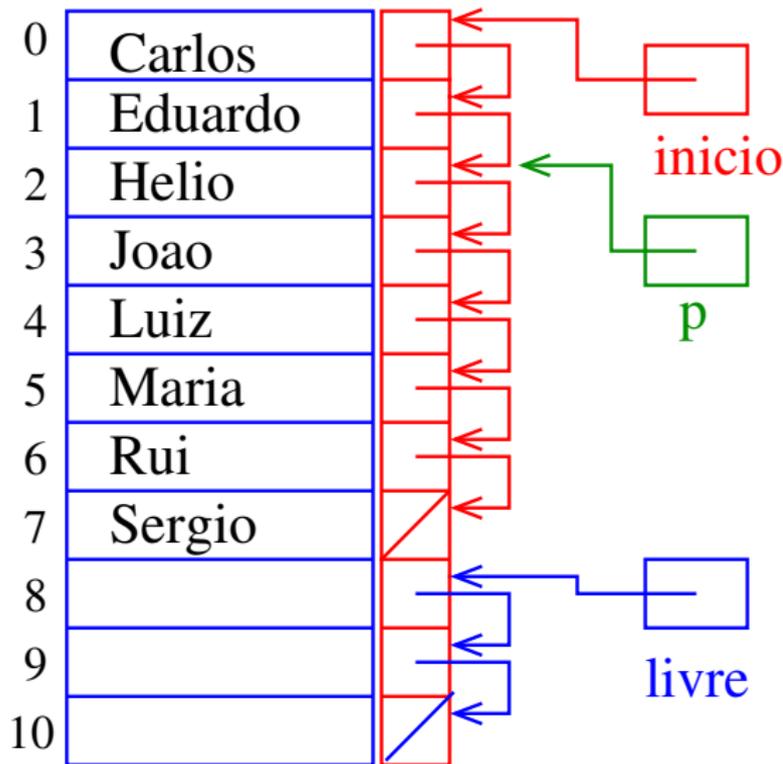
## Remove Joao



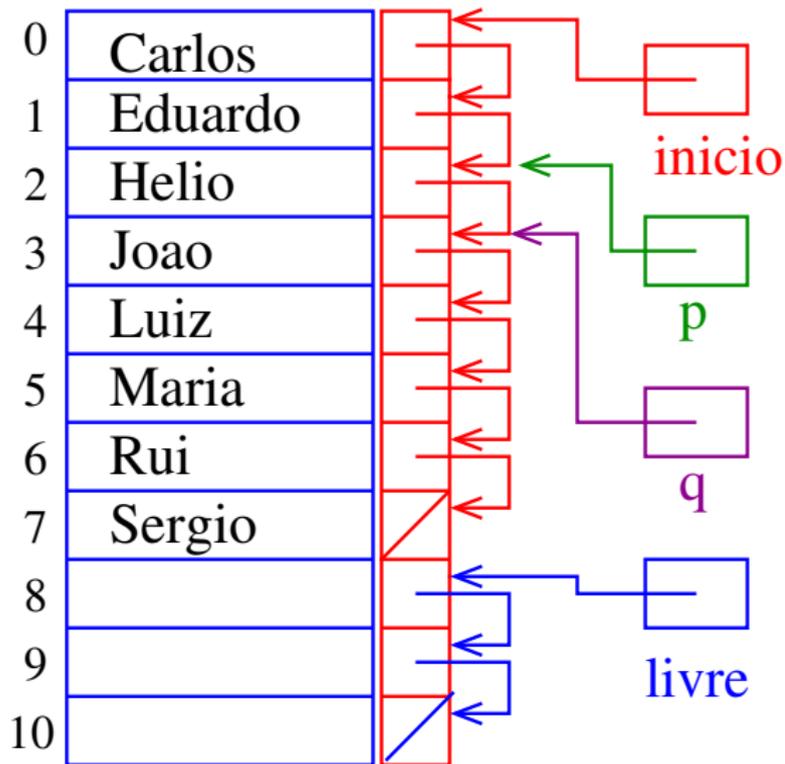
## Remove Joao



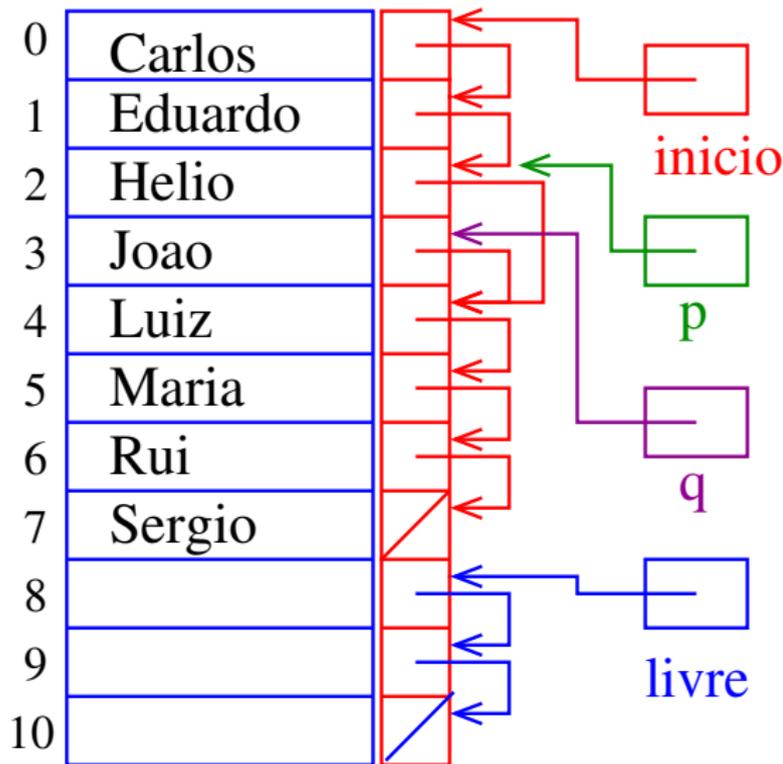
## Remove Joao



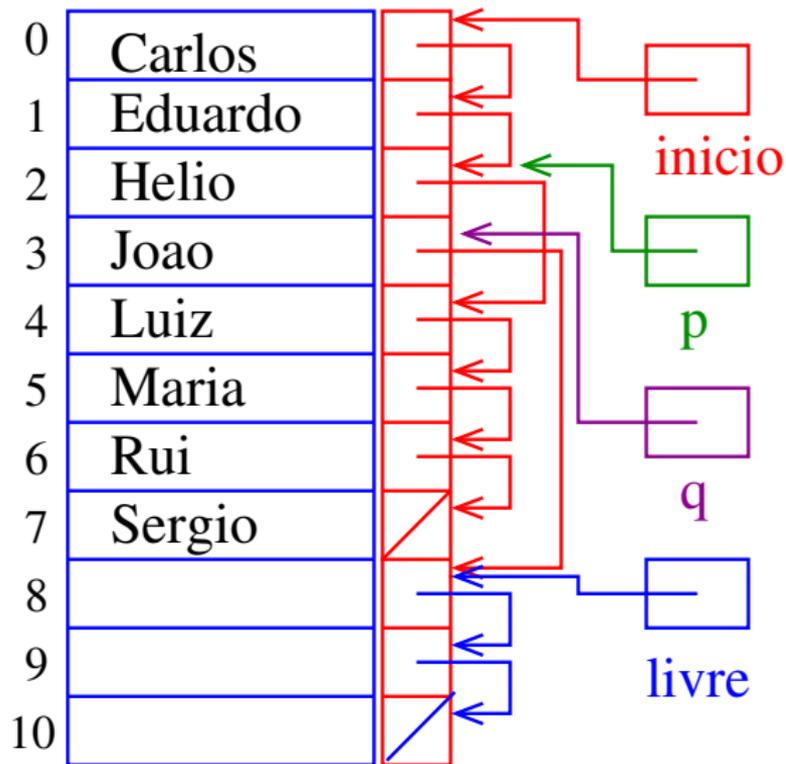
## Remove Joao



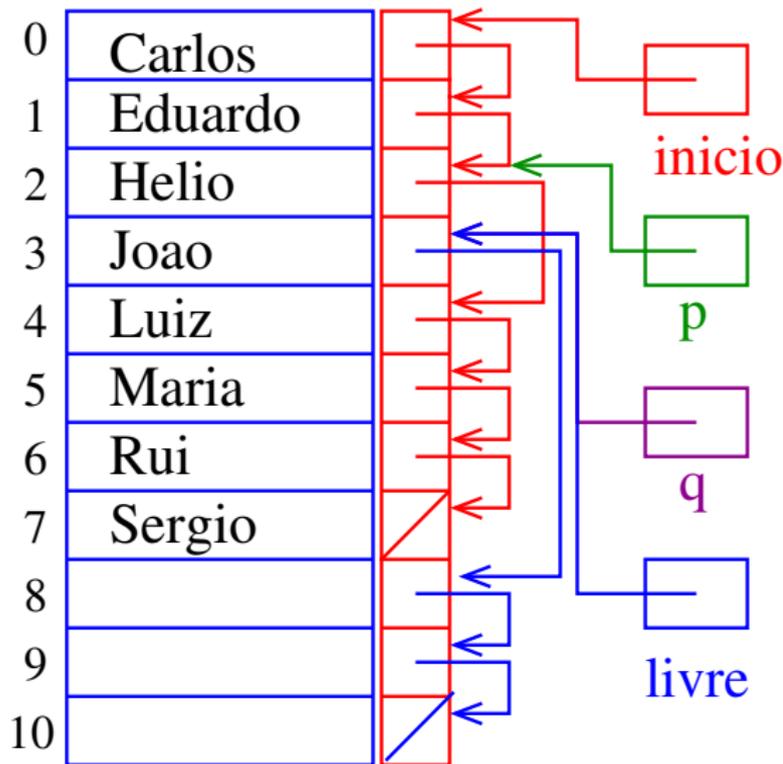
# Remove Joao



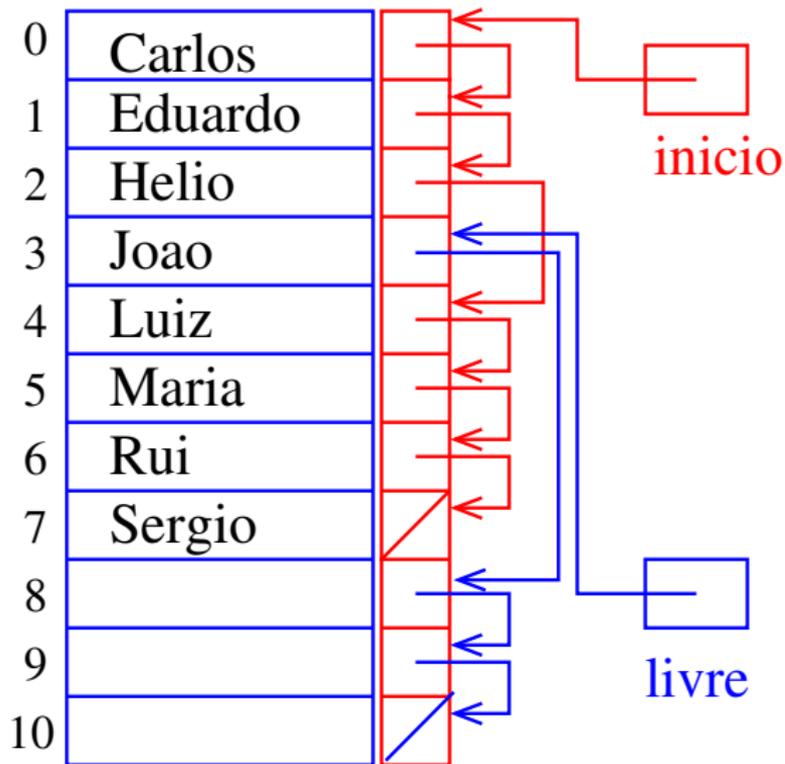
## Remove Joao



## Remove Joao



## Remove Joao



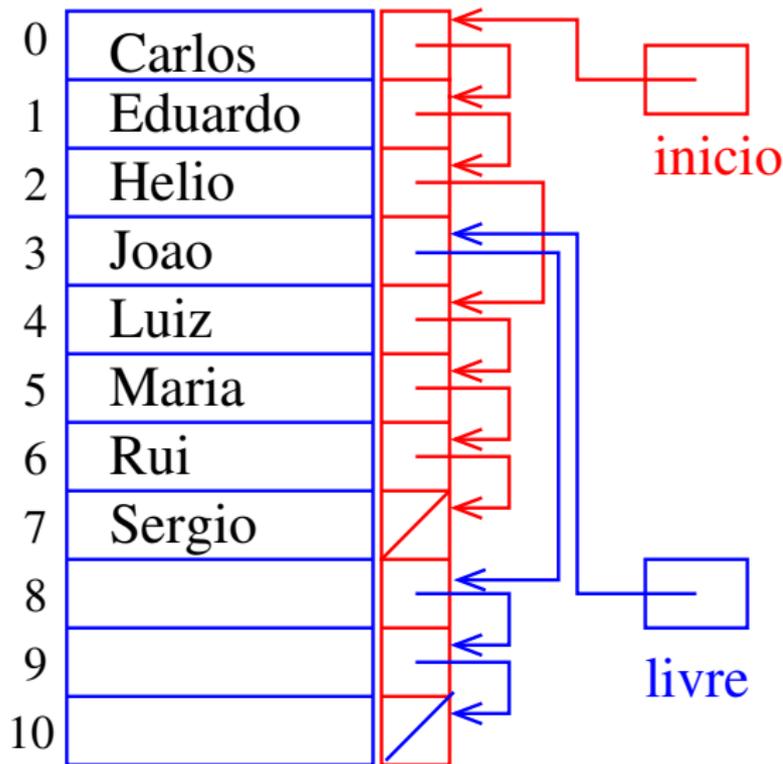
## Remoção

Dado um índice  $p$  de uma célula, o trecho de código que remove a célula de índice  $v[p].prox$  é “essencialmente” o seguinte:

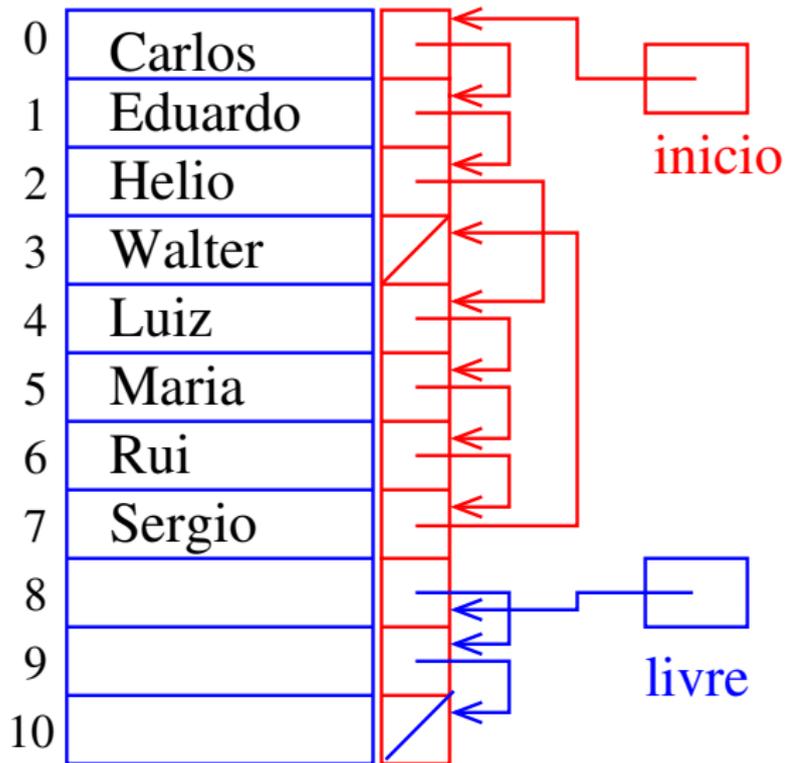
```
q = v[p].prox;  
v[p].prox = v[q].prox;  
v[q].prox = livre;  
livre = q;
```

Dizemos que o **consumo de tempo** desse trecho de código é **constante**, pois **não depende** do **tamanho da lista**.

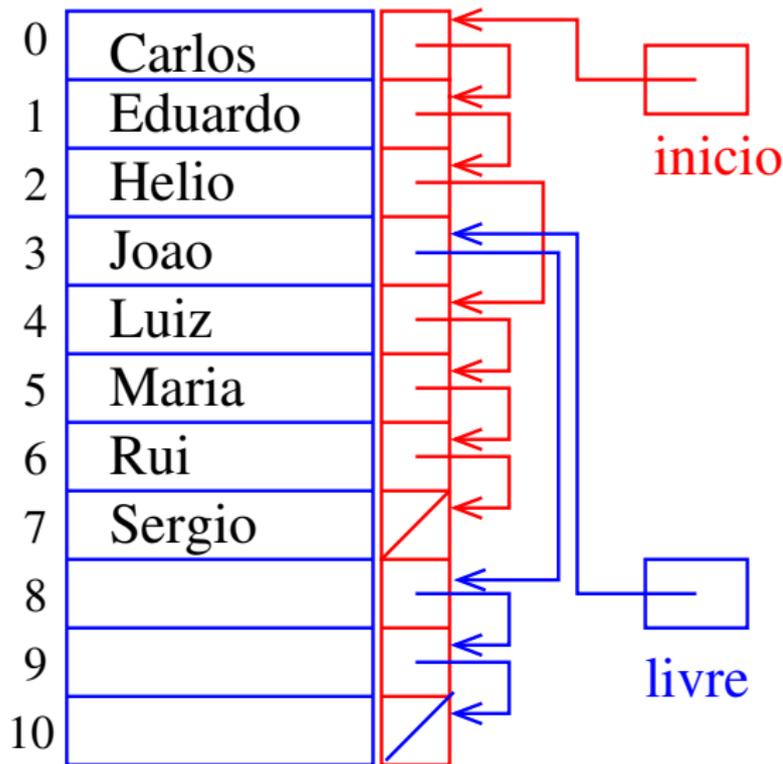
## Inserir Walter



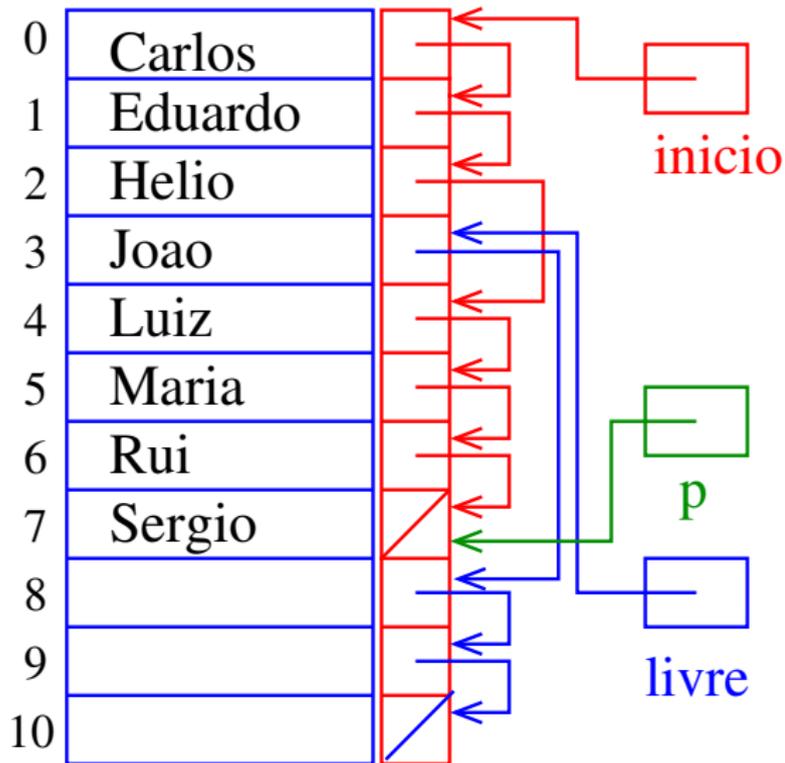
# Inserir Walter



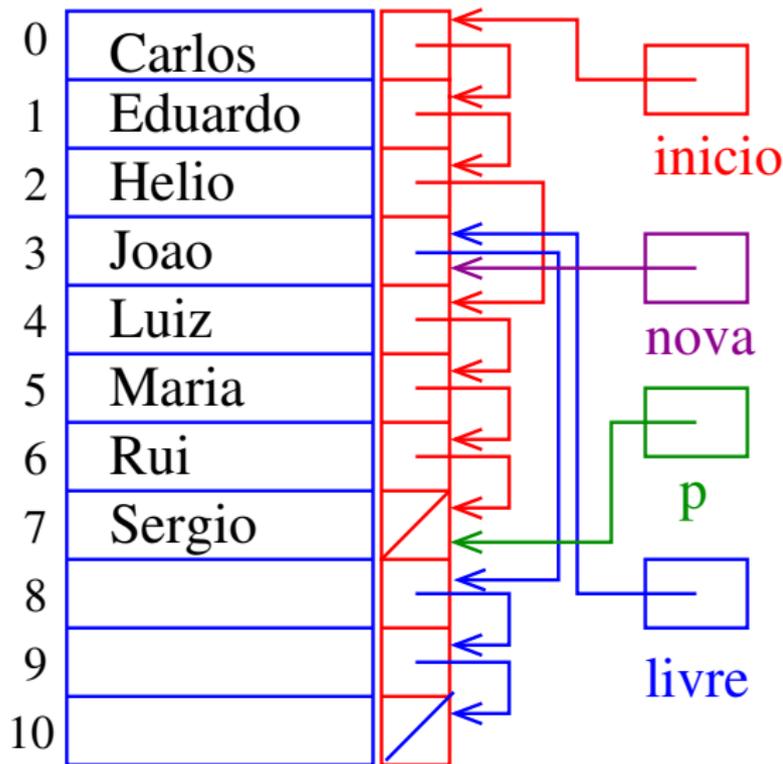
## Inserir Walter



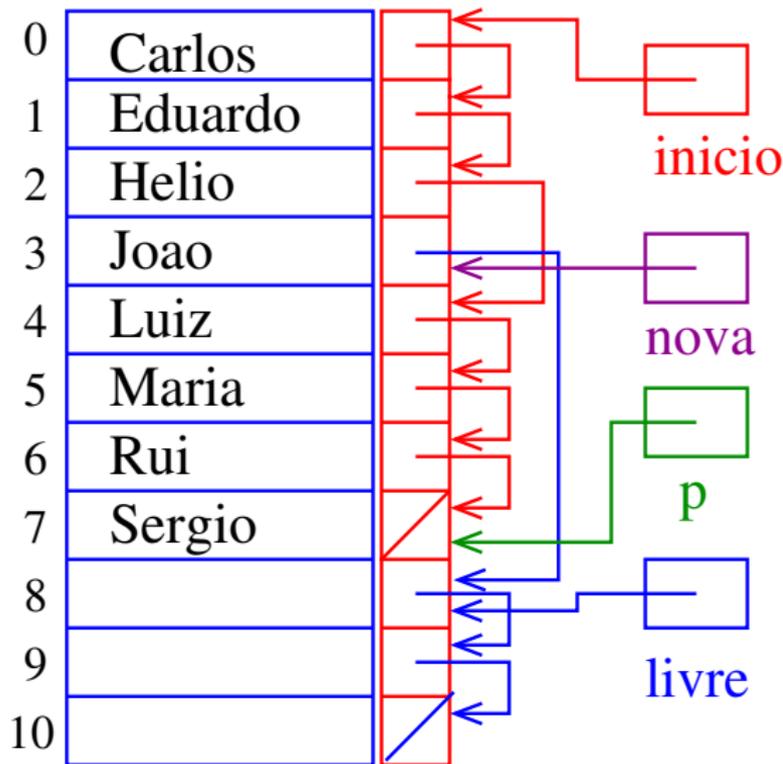
# Inserir Walter



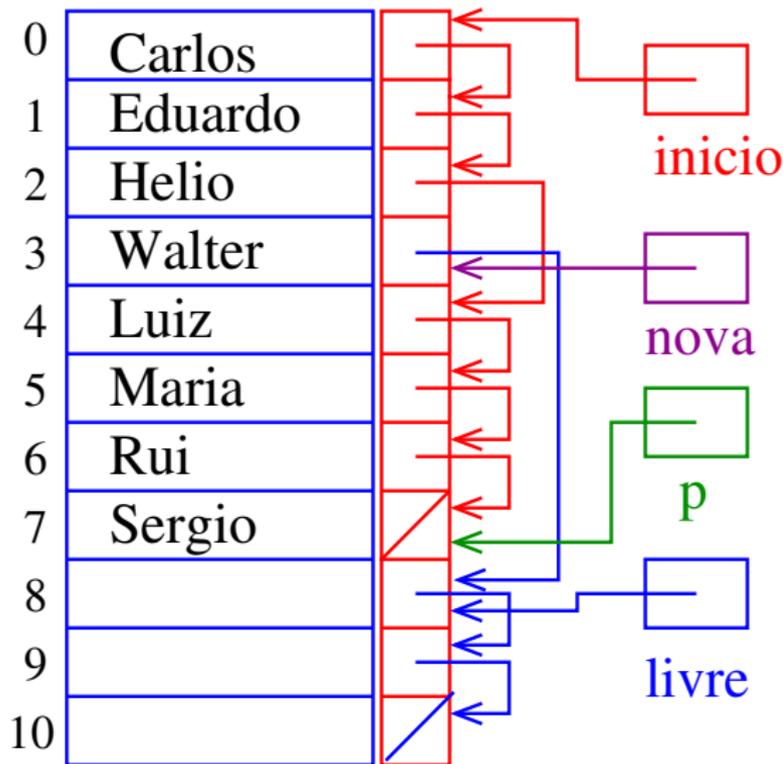
# Inserir Walter



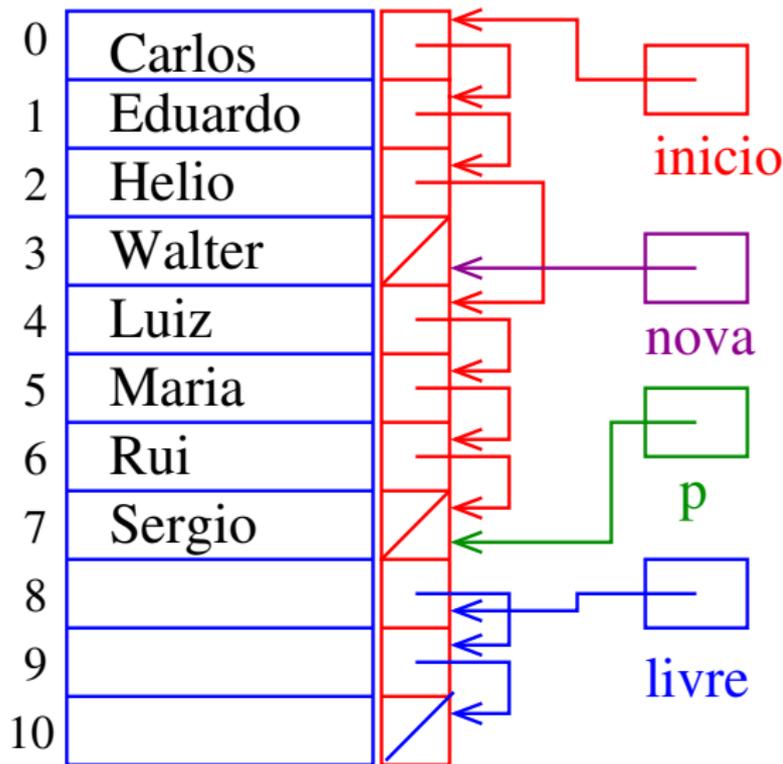
# Inserir Walter



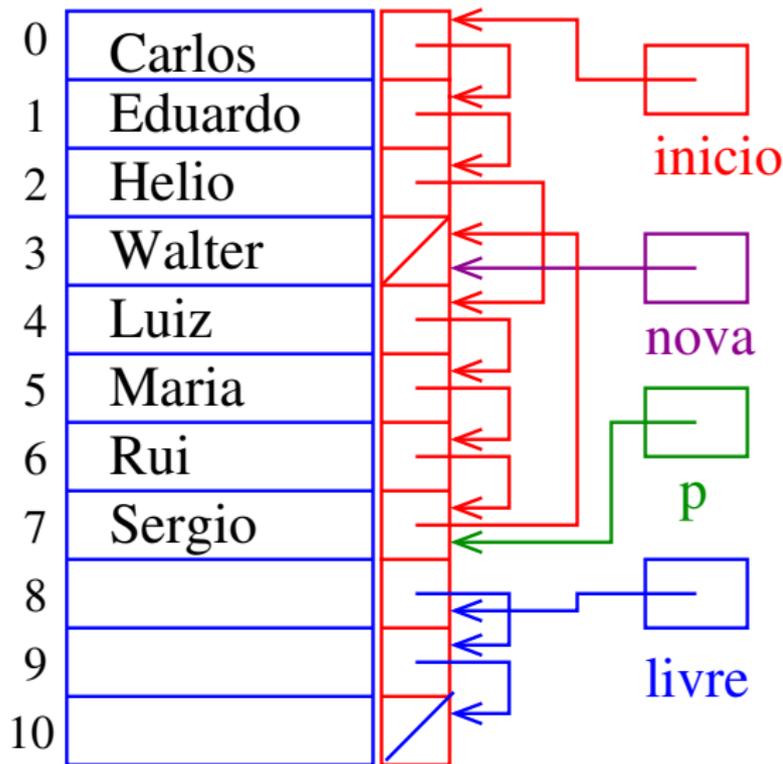
# Inserir Walter



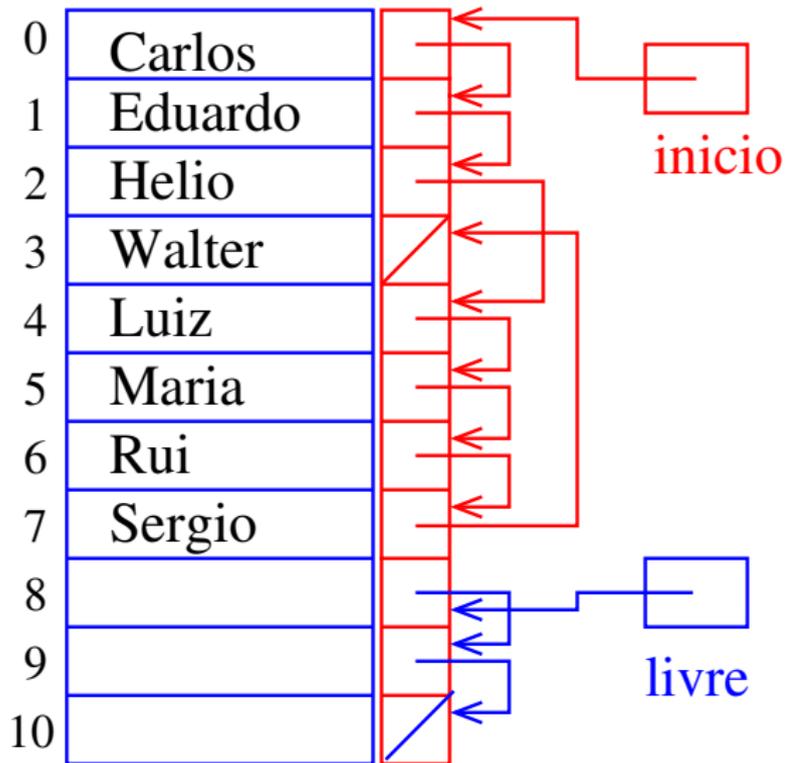
# Inserir Walter



# Inserir Walter



# Inserir Walter



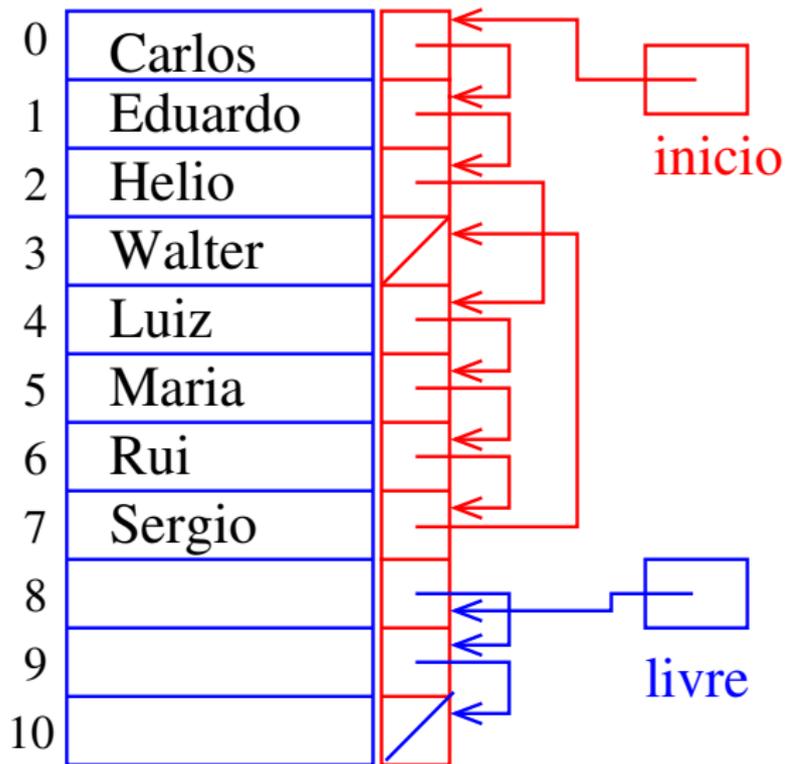
## Inserção

O trecho de código para inserir uma nova célula com o elemento `x` entre as células de índice `p` e `v[p].prox` é “essencialmente” o seguinte

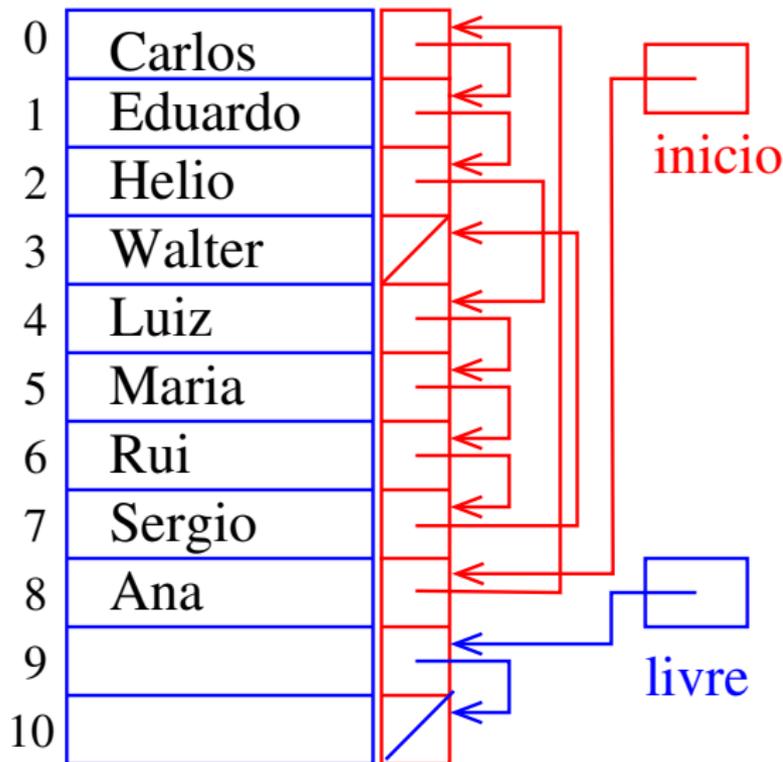
```
nova = livre;  
livre = v[livre].prox;  
v[nova].conteudo = x;  
v[nova].prox = v[p].prox;  
v[p].prox = nova;
```

Dizemos que o **consumo de tempo** desse trecho de código é **constante**, pois **não depende** do **tamanho da lista**.

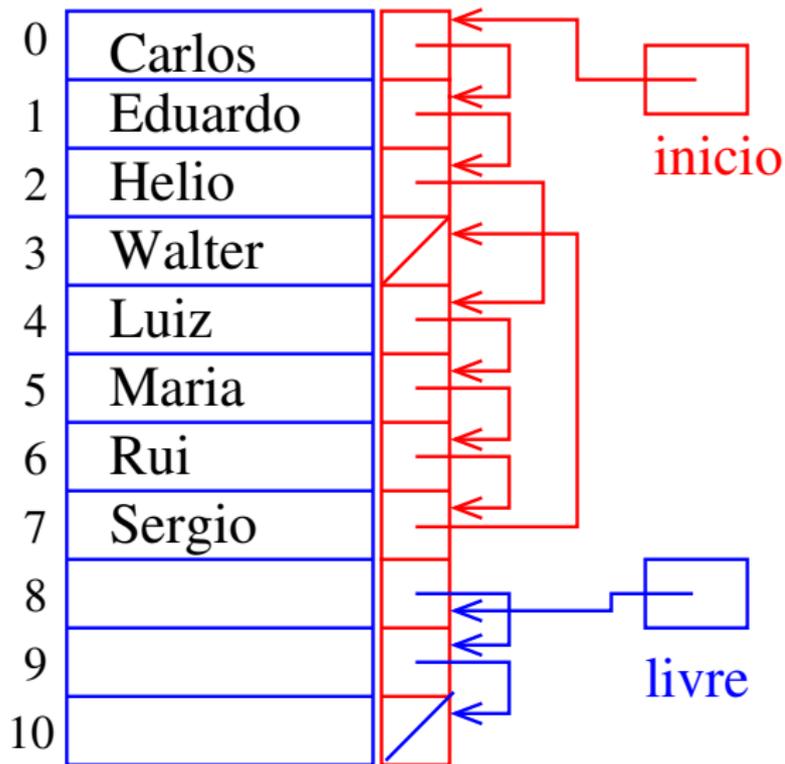
## Inserir Ana



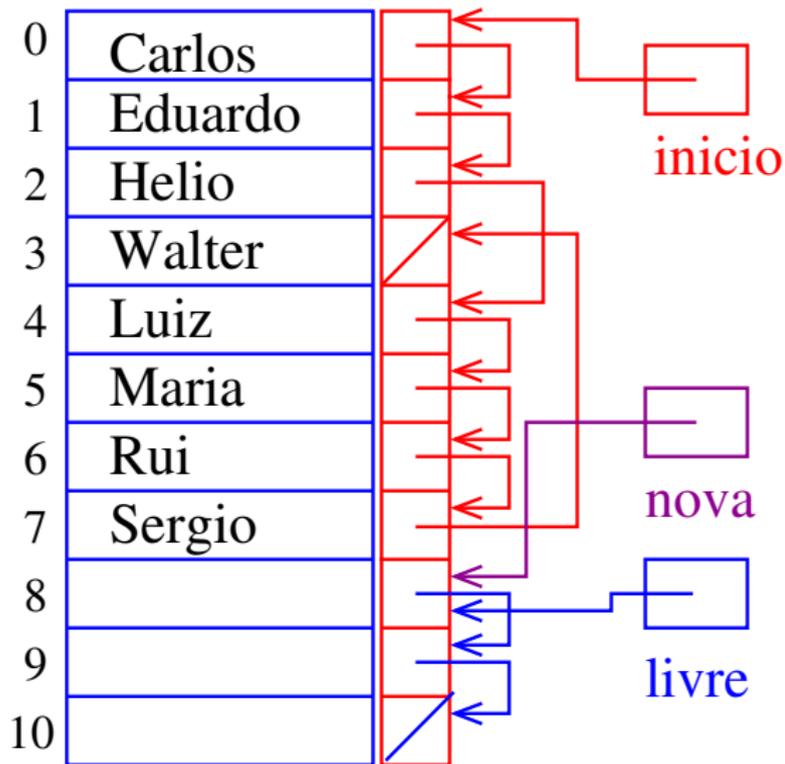
# Inserir Ana



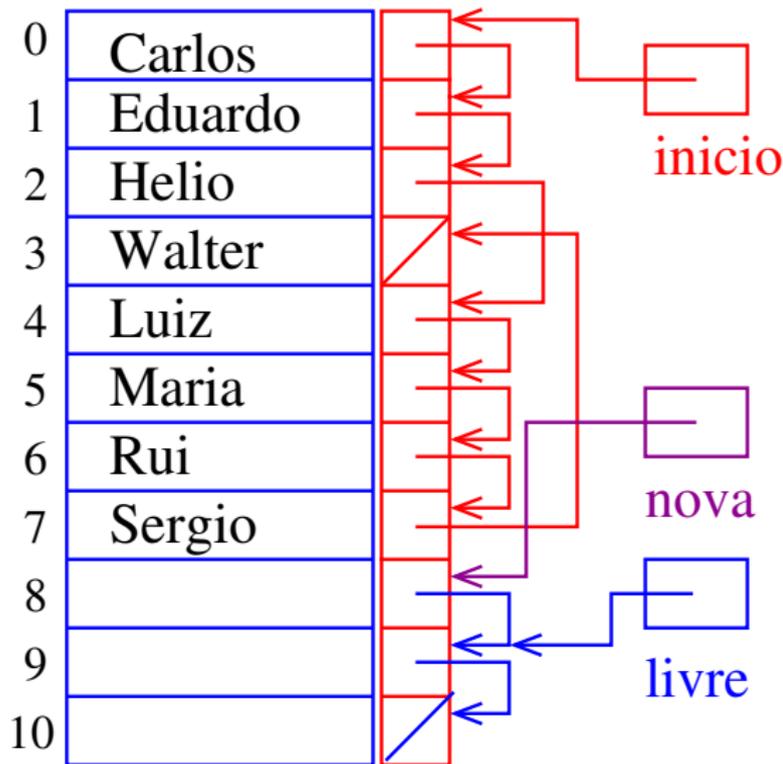
## Inserir Ana



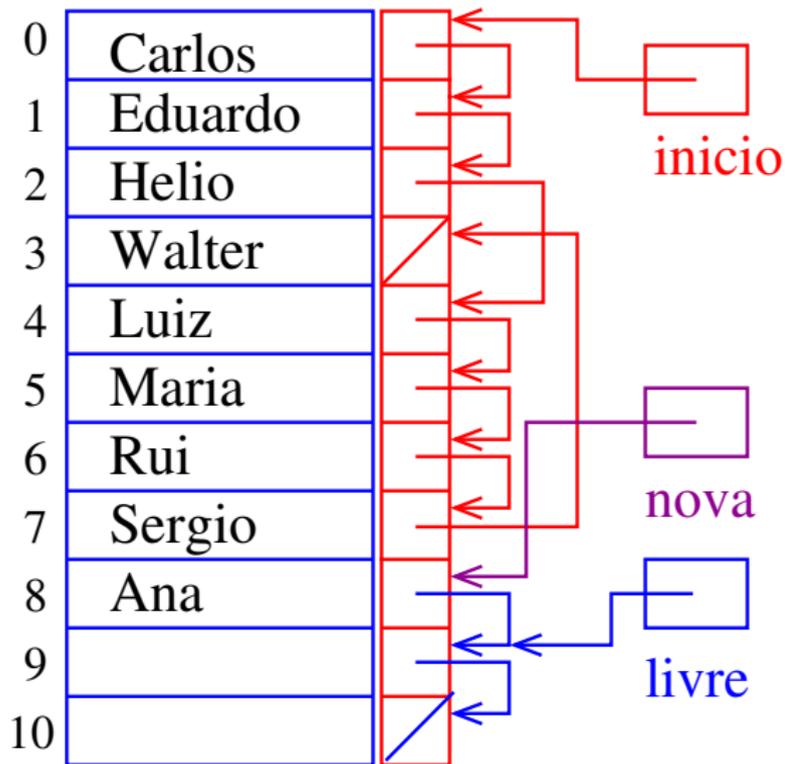
# Inserir Ana



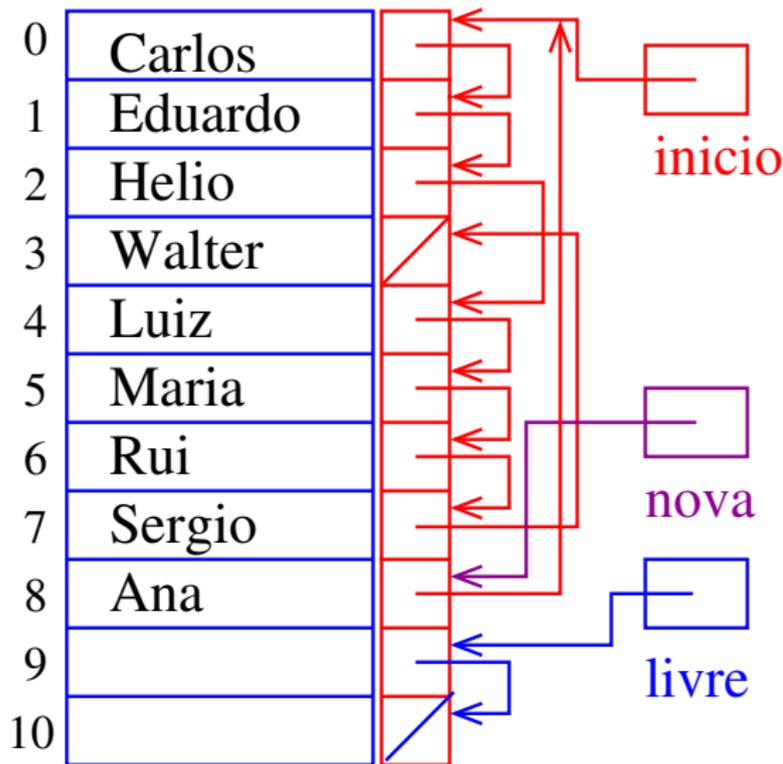
# Inserir Ana



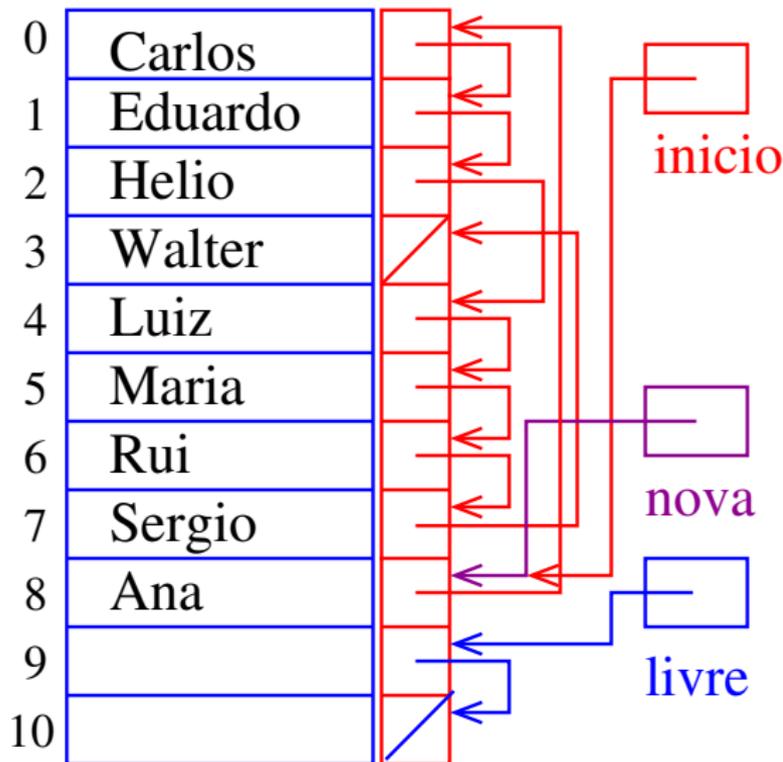
# Inserir Ana



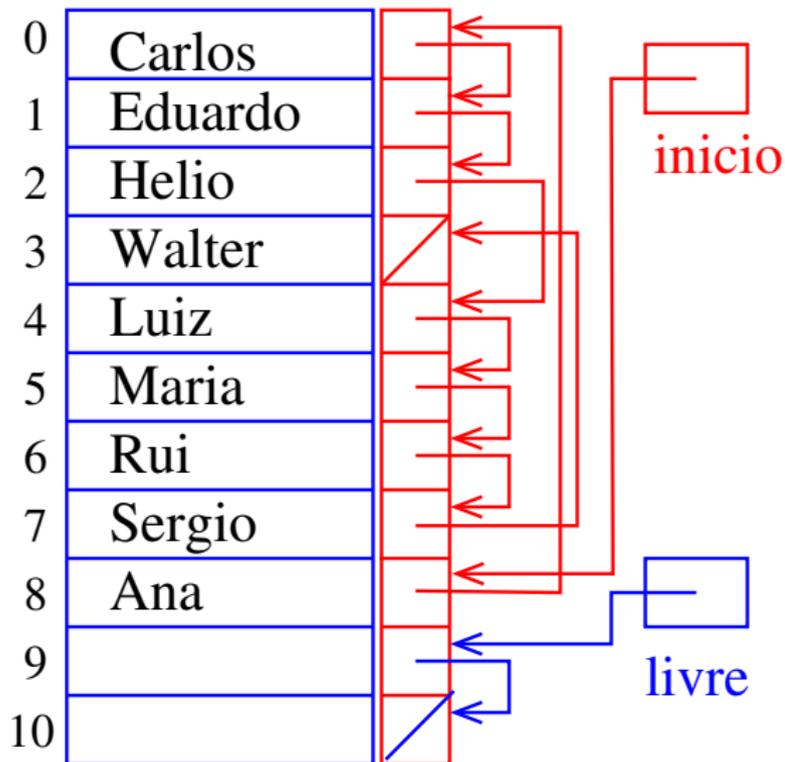
## Inserir Ana



# Inserir Ana



# Inserir Ana



## Inserção no início

O trecho de código para inserir uma nova célula com o elemento `x` no **início** da lista;

```
nova = livre;  
livre = v[livre].prox;  
v[nova].conteudo = x;  
v[nova].prox = inicio;  
inicio = nova;
```

Dizemos que o **consumo de tempo** desse trecho de código é **constante**, pois **não depende** do **tamanho da lista**.

# Listas encadeadas em C

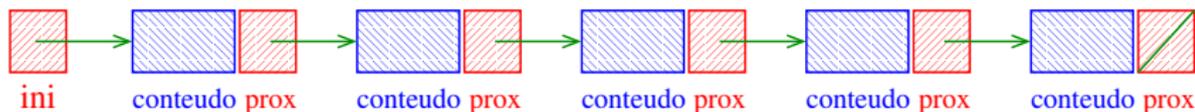
PF 4, S 3.3

<http://www.ime.usp.br/~pf/algoritmos/aulas/lista.html>

# Listas encadeadas

Uma **lista encadeada** (= *linked list* = lista ligada) é uma sequência de **células**; cada **célula** contém um **objeto** de algum tipo e o **endereço** da célula seguinte.

Ilustração de uma **lista encadeada** (“sem cabeça”)



## Estrutura para listas encadeadas em C

```
struct celula {  
    int conteudo;  
    struct celula *prox;  
};  
typedef struct celula Celula;  
  
Celula *ini;  
/* inicialmente a lista esta vazia */  
ini = NULL;
```



ini

# Endereços e listas encadeadas

O endereço de uma lista encadeada é o endereço de sua primeira célula.

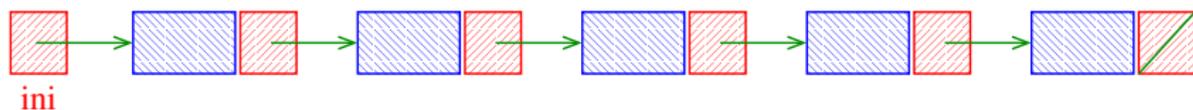
Se  $p$  é o endereço de uma lista, às vezes dizemos que “ $p$  é uma lista”.

Se  $p$  é uma lista então

- ▶  $p == \text{NULL}$  ou
- ▶  $p \rightarrow \text{prox}$  é uma lista.

## Imprime conteúdo de uma lista

Esta função **imprime** o **conteudo** de cada célula de uma lista encadeada **ini**.



```
void imprima (Celula *ini) {  
    Celula *p;  
    for (p = ini; p != NULL; p = p->prox)  
        printf("%d\n", p->conteudo);  
}
```

## Busca em listas encadeadas

Esta função **recebe** um inteiro **x** e uma lista **ini**, e **devolve** o endereço de uma célula que contém **x**. Se tal célula não existe, a função **devolve** **NULL**.

```
Celula *busca (int x, Celula *ini) {  
    Celula *p;  
  
    p = ini;  
    while (p != NULL && p->conteudo != x)  
        p = p->prox;  
  
    return p;  
}
```

## Inserção no início de uma lista encadeada

Cria uma célula para guardar um elemento `x` e insere esta célula no início da lista `ini`.

```
void insere (int x, Celula *ini) {  
    Celula nova;  
    nova.conteudo = x;  
    nova.prox = ini;  
    ini = &nova;  
}
```

## Inserção no início de uma lista encadeada

Cria uma célula para guardar um elemento `x` e insere esta célula no início da lista `ini`.

```
void insere (int x, Celula *ini) {  
    Celula nova;  
    nova.conteudo = x;  
    nova.prox = ini;  
    ini = &nova;  
}
```

**ERRADO!**

A estrutura `nova` é uma **variável local** (=automática). Ela será devolvida ao sistema no final da execução da função.

## Inserção no início de uma lista encadeada

Cria uma célula para guardar um elemento `x` e insere esta célula no início da lista `ini`.

```
void insere (int x, Celula *ini) {  
    Celula *nova;  
    nova = mallocSafe(sizeof(Celula));  
    nova->conteudo = x;  
    nova->prox = ini;  
    ini = nova;  
}
```

## Inserção no início de uma lista encadeada

Cria uma célula para guardar um elemento `x` e insere esta célula no início da lista `ini`.

```
void insere (int x, Celula *ini) {  
    Celula *nova;  
    nova = mallocSafe(sizeof(Celula));  
    nova->conteudo = x;  
    nova->prox = ini;  
    ini = nova;  
}
```

**ERRADO!**

Mesmo erro! O parâmetro `ini` é uma **variável local** (=automática) inicializada com valor do argumento.

## Inserção no início de uma lista encadeada

**Cria** uma célula para guardar um elemento `x`  
e **insere** esta célula no início da lista `ini`.

```
Celula *insere (int x, Celula *ini) {  
    Celula *nova;  
    nova = mallocSafe(sizeof(Celula));  
    nova->conteudo = x;  
    nova->prox = ini;  
    return nova;  
}
```

**Certo!** Exemplos de chamadas da função:

```
ini = insere(11, ini);  
ini = insere(valor+3, ini);
```

## Inserção no início de uma lista encadeada

**Cria** uma célula para guardar um elemento `x`  
e **insere** esta célula no início da lista `ini`.

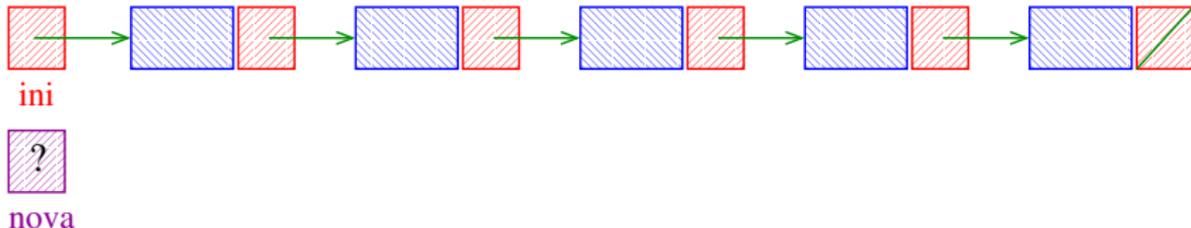
```
void insere (int x, Celula **ini) {  
    Celula *nova;  
    nova = mallocSafe(sizeof(Celula));  
    nova->conteudo = x;  
    nova->prox = *ini;  
    *ini = nova;  
}
```

**Certo!** Exemplos de chamadas da função:

```
insere(11, &ini);  
insere(valor+3, &ini);
```

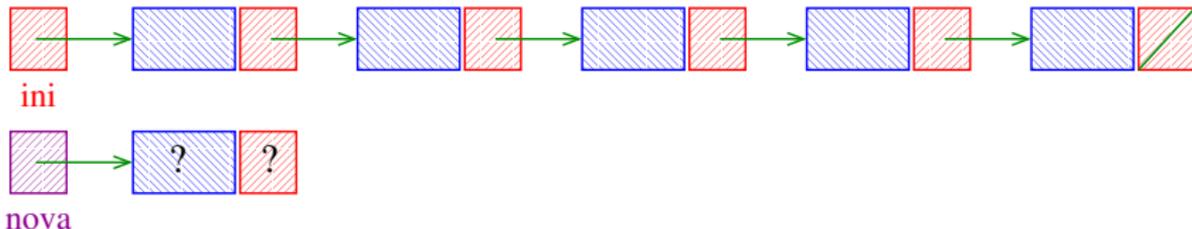
# Inserção no início de uma lista encadeada

**Cria** uma célula para guardar um elemento  $x$   
e **insere** esta célula no início da lista **ini**.



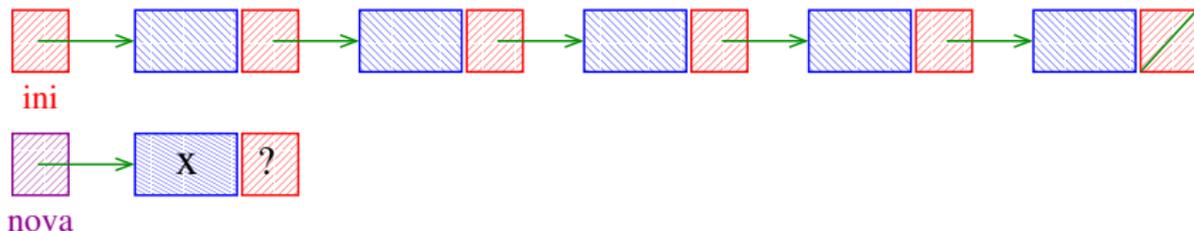
# Inserção no início de uma lista encadeada

**Cria** uma célula para guardar um elemento  $x$   
e **insere** esta célula no início da lista **ini**.



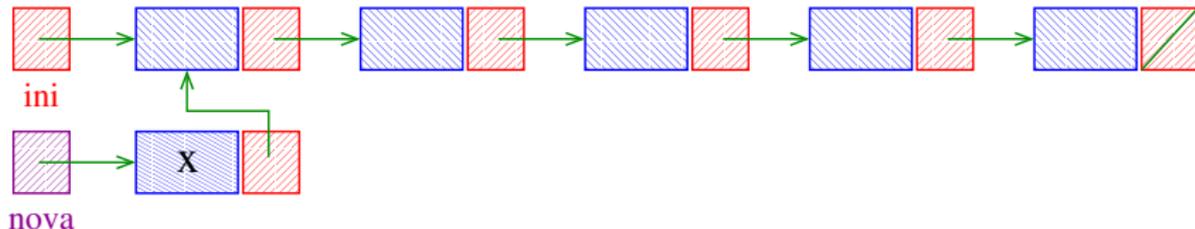
# Inserção no início de uma lista encadeada

**Cria** uma célula para guardar um elemento  $x$   
e **insere** esta célula no início da lista **ini**.



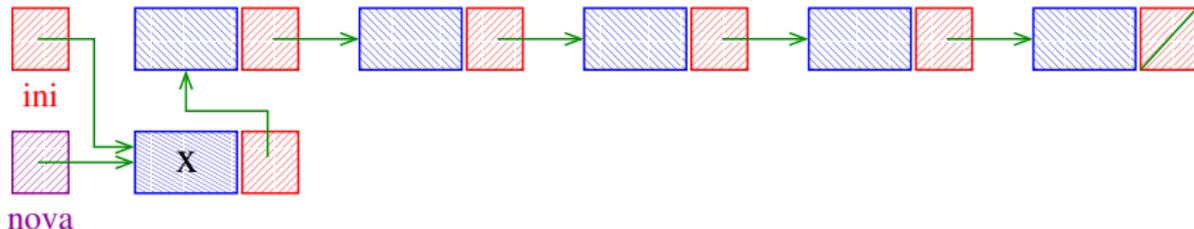
# Inserção no início de uma lista encadeada

**Cria** uma célula para guardar um elemento  $x$   
e **insere** esta célula no início da lista **ini**.



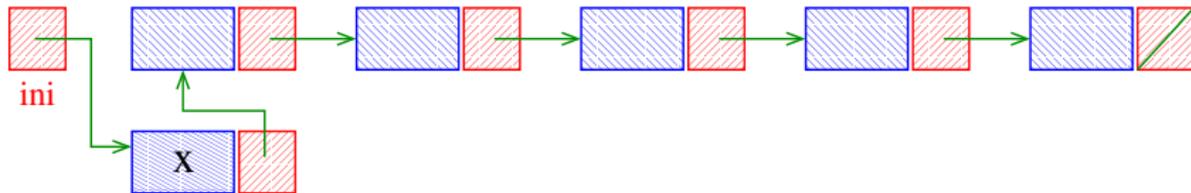
# Inserção no início de uma lista encadeada

**Cria** uma célula para guardar um elemento  $x$   
e **insere** esta célula no início da lista **ini**.



# Inserção no início de uma lista encadeada

**Cria** uma célula para guardar um elemento  $x$   
e **insere** esta célula no início da lista **ini**.



## Remoção em uma lista encadeada

Recebe o endereço `p` de uma célula de uma lista encadeada e `remove` da lista a célula `p->prox`.

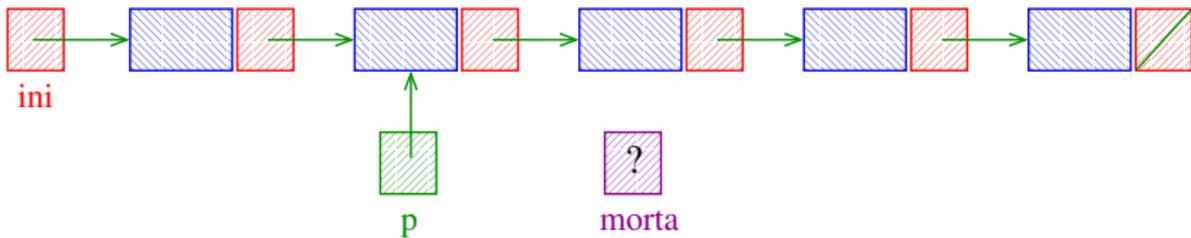
A função supõe que `p != NULL` e  
`p->prox != NULL`.

```
void remove (Celula *p) {  
    Celula *morta;  
    morta = p->prox;  
    p->prox = morta->prox;  
    free(morta);  
}
```

# Remoção em uma lista encadeada

Recebe o endereço  $p$  de uma célula de uma lista encadeada e remove da lista a célula  $p \rightarrow prox$ .

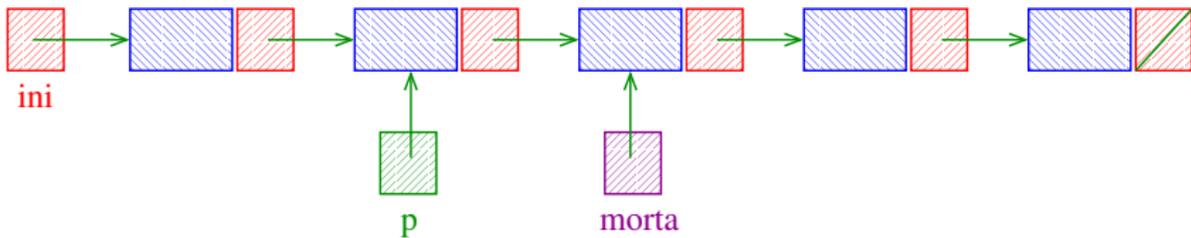
A função supõe que  $p \neq NULL$  e  $p \rightarrow prox \neq NULL$ .



## Remoção em uma lista encadeada

Recebe o endereço  $p$  de uma célula de uma lista encadeada e remove da lista a célula  $p \rightarrow \text{prox}$ .

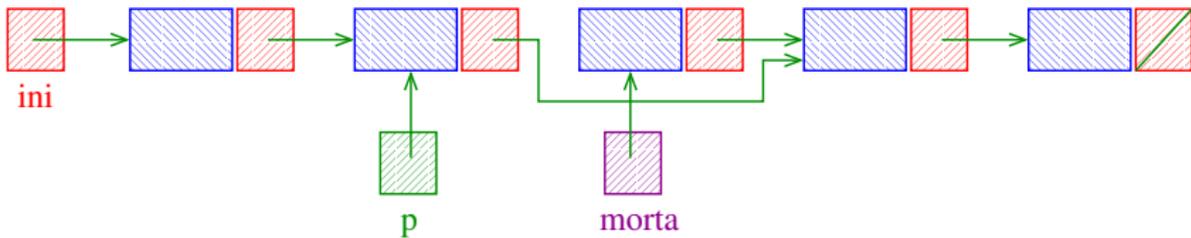
A função supõe que  $p \neq \text{NULL}$  e  $p \rightarrow \text{prox} \neq \text{NULL}$ .



# Remoção em uma lista encadeada

Recebe o endereço  $p$  de uma célula de uma lista encadeada e remove da lista a célula  $p \rightarrow \text{prox}$ .

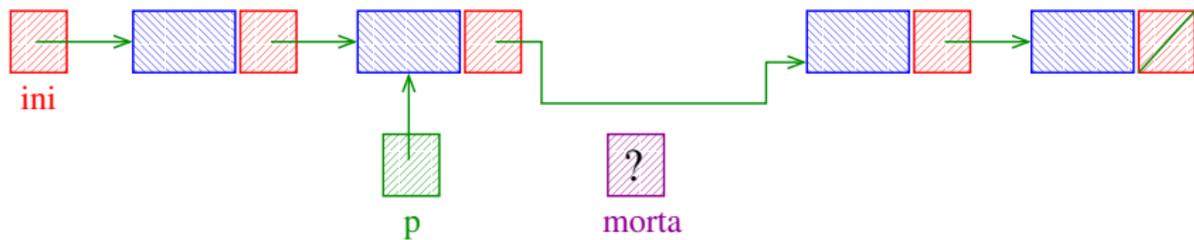
A função supõe que  $p \neq \text{NULL}$  e  $p \rightarrow \text{prox} \neq \text{NULL}$ .



# Remoção em uma lista encadeada

Recebe o endereço  $p$  de uma célula de uma lista encadeada e remove da lista a célula  $p \rightarrow prox$ .

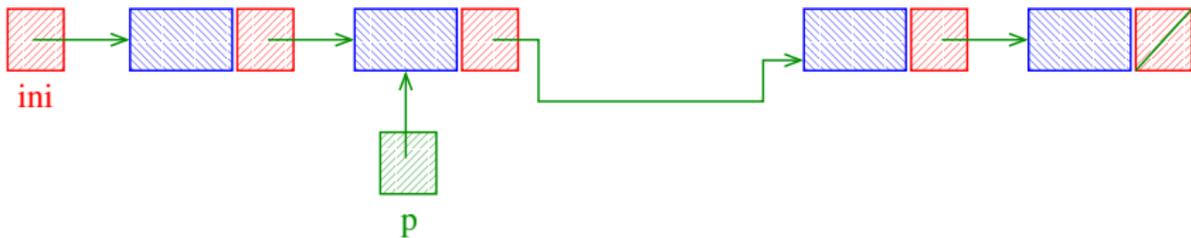
A função supõe que  $p \neq NULL$  e  $p \rightarrow prox \neq NULL$ .



## Remoção em uma lista encadeada

Recebe o endereço  $p$  de uma célula de uma lista encadeada e remove da lista a célula  $p \rightarrow \text{prox}$ .

A função supõe que  $p \neq \text{NULL}$  e  $p \rightarrow \text{prox} \neq \text{NULL}$ .



## Busca e Inserção em uma lista

Recebe lista `ini` e insere célula de conteúdo `x` antes da primeira célula de conteúdo `y`. Se nenhuma célula contém `y`, insere a célula com `x` no final da lista.

`Celula *`

```
buscaInsere(int x, int y, Celula *ini) {  
    Celula *p, *q, *nova;  
    nova = mallocSafe(sizeof(Celula));  
    nova->conteudo = x;  
    if (ini == NULL || ini->conteudo == y) {  
        nova->prox = ini;  
        ini = nova;  
    }  
}
```

## Busca e Inserção em uma lista

```
else {
    p = ini;
    q = p->prox;
    while (q != NULL && q->conteudo != y) {
        p = q;
        q = p->prox;
    }
    p->prox = nova;
    nova->prox = q;
}
return ini;
}
```

## Chamadas de buscaInsere

```
Celula *ini, *ini2;  
ini = ini2 = NULL;
```

[...manipulação das listas ...]

```
ini = buscaInsere(22, 33, ini);  
ini2 = buscaInsere(x+1, y, ini2);  
ini2 = buscaInsere(x, 2*y, ini2);  
ini = buscaInsere(valor, meio, ini);
```

## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, insere a célula com **x** no final da lista.

Caso em que **ini** está vazia.



ini

## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, insere a célula com **x** no final da lista.

Caso em que **ini** está vazia.



ini

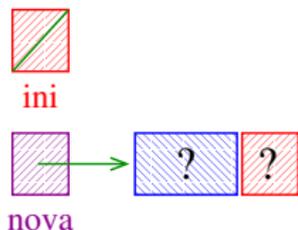


nova

## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, insere a célula com **x** no final da lista.

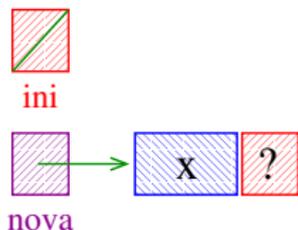
Caso em que **ini** está vazia.



## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, insere a célula com **x** no final da lista.

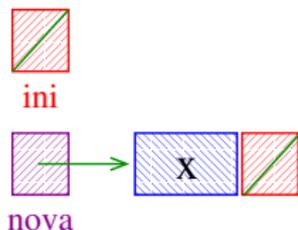
Caso em que **ini** está vazia.



## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, insere a célula com **x** no final da lista.

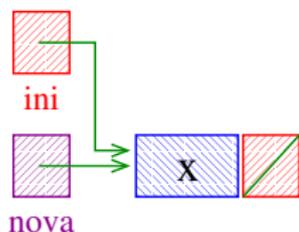
Caso em que **ini** está vazia.



## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, insere a célula com **x** no final da lista.

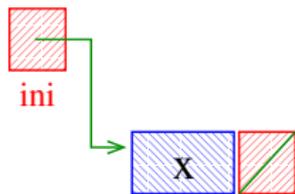
Caso em que **ini** está vazia.



## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, insere a célula com **x** no final da lista.

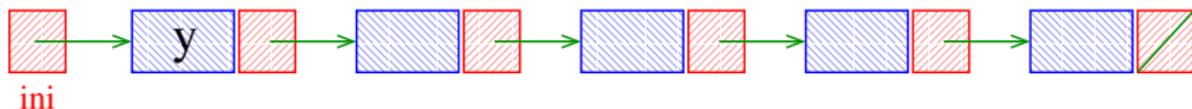
Caso em que **ini** está vazia.



## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

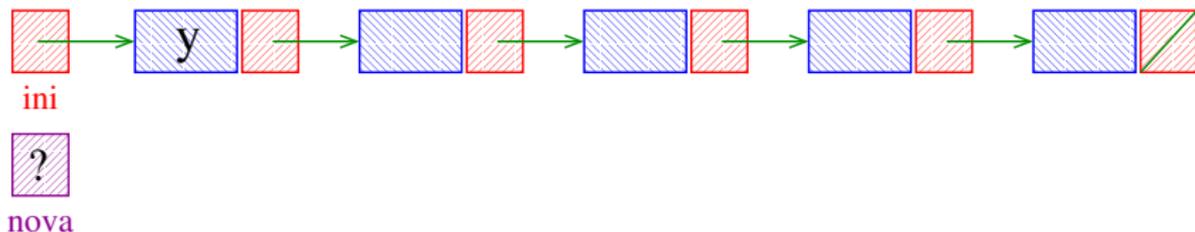
Caso em que a primeira célula contém  $y$ .



## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

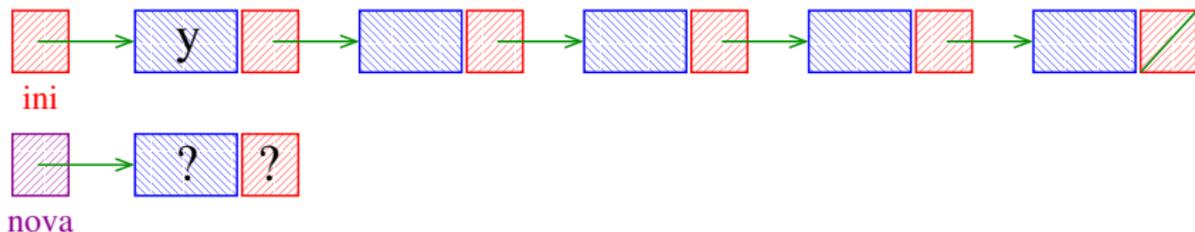
Caso em que a primeira célula contém  $y$ .



## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

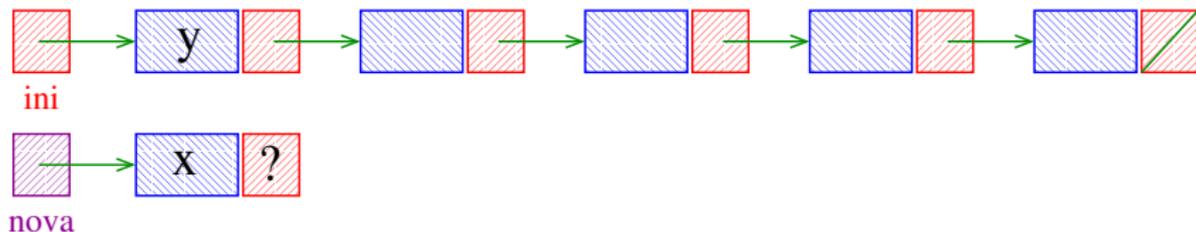
Caso em que a primeira célula contém  $y$ .



## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, insere a célula com **x** no final da lista.

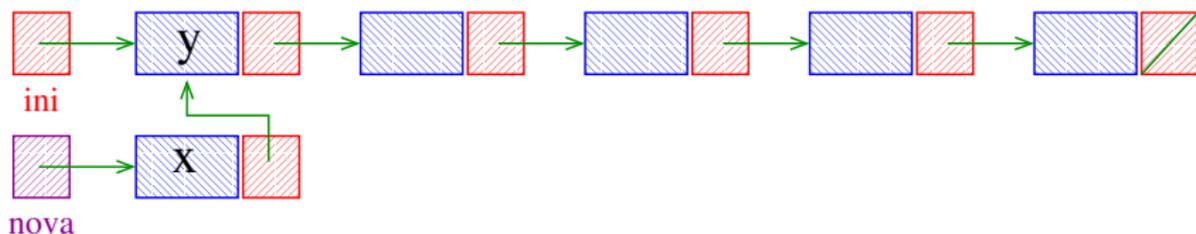
Caso em que a primeira célula contém **y**.



## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, insere a célula com **x** no final da lista.

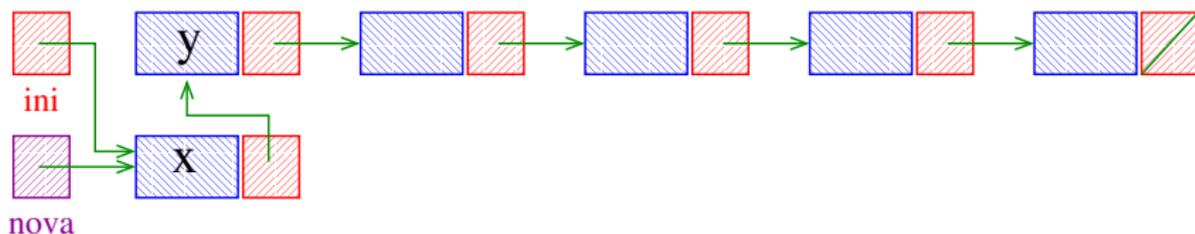
Caso em que a primeira célula contém **y**.



## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, insere a célula com **x** no final da lista.

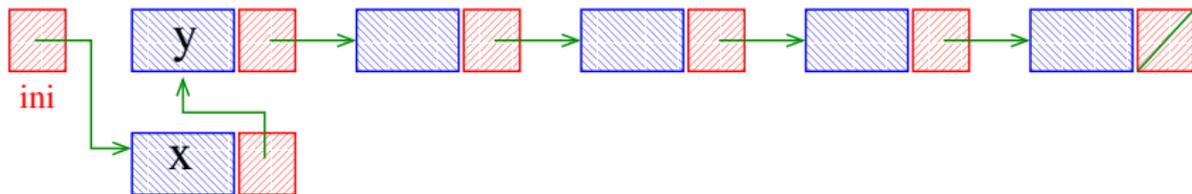
Caso em que a primeira célula contém **y**.



## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, insere a célula com **x** no final da lista.

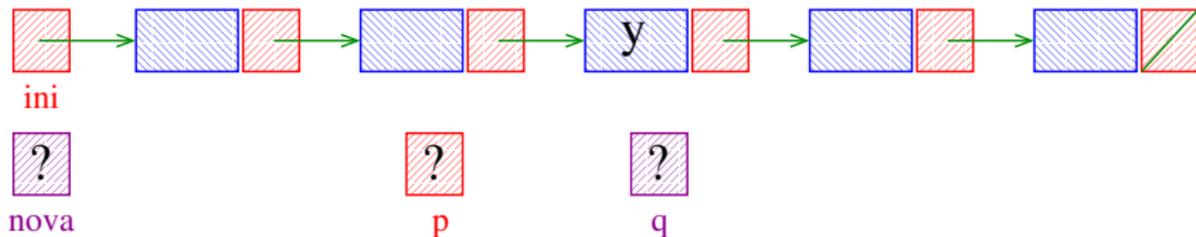
Caso em que a primeira célula contém **y**.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

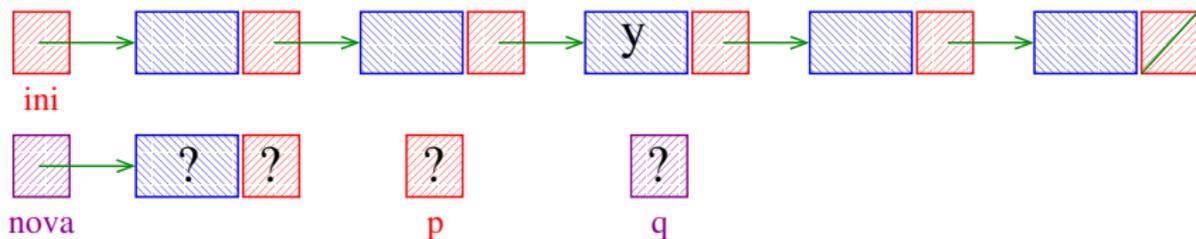
Caso geral:  $y$  está na lista, mas não na primeira célula.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

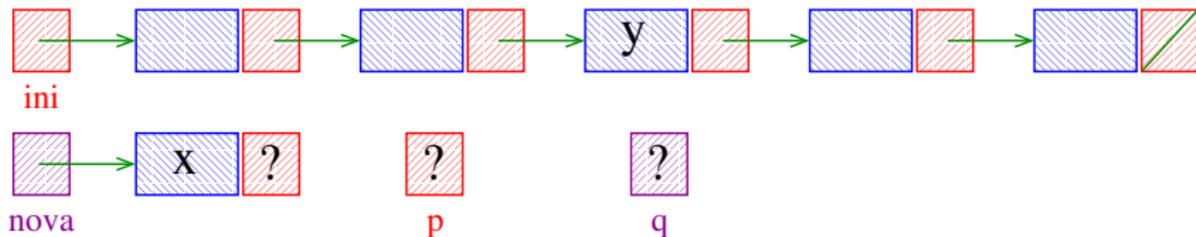
Caso geral:  $y$  está na lista, mas não na primeira célula.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

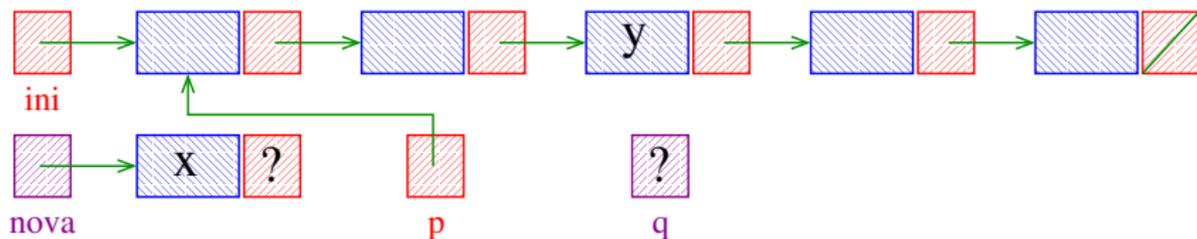
Caso geral:  $y$  está na lista, mas não na primeira célula.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

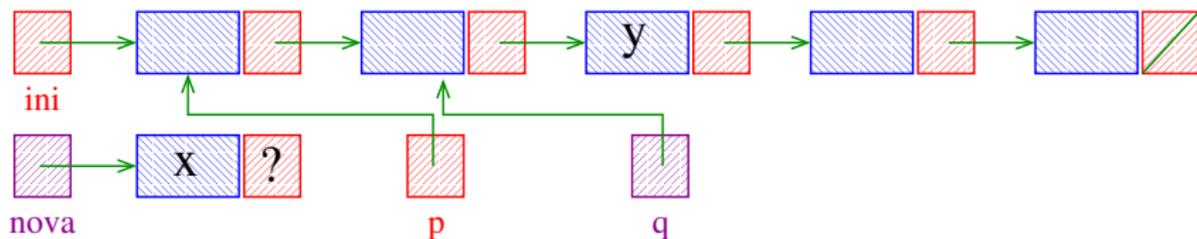
Caso geral:  $y$  está na lista, mas não na primeira célula.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

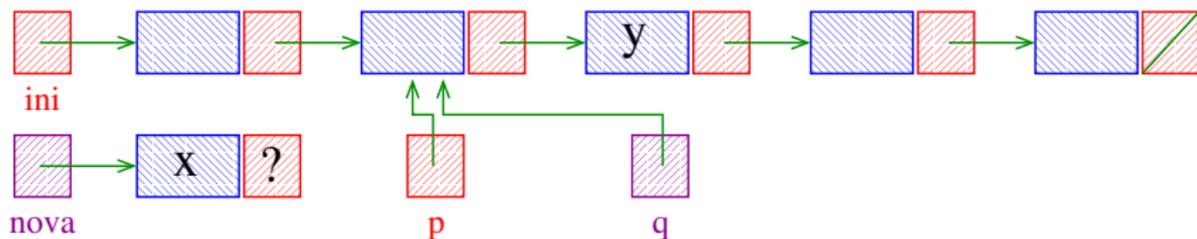
Caso geral:  $y$  está na lista, mas não na primeira célula.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

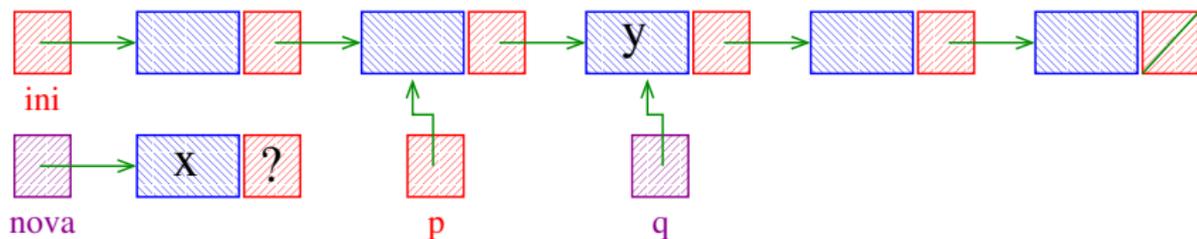
Caso geral:  $y$  está na lista, mas não na primeira célula.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

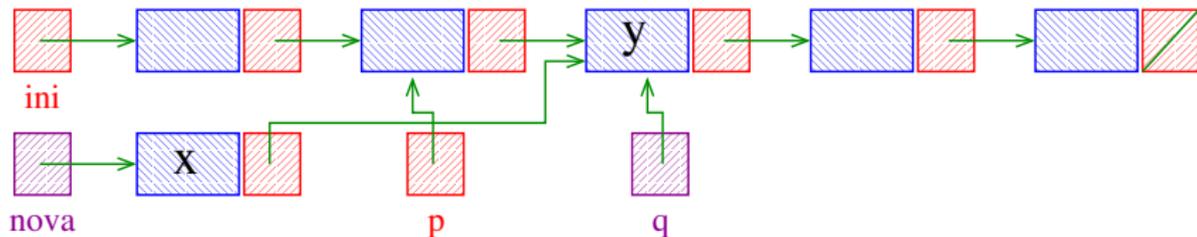
Caso geral:  $y$  está na lista, mas não na primeira célula.



## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, insere a célula com **x** no final da lista.

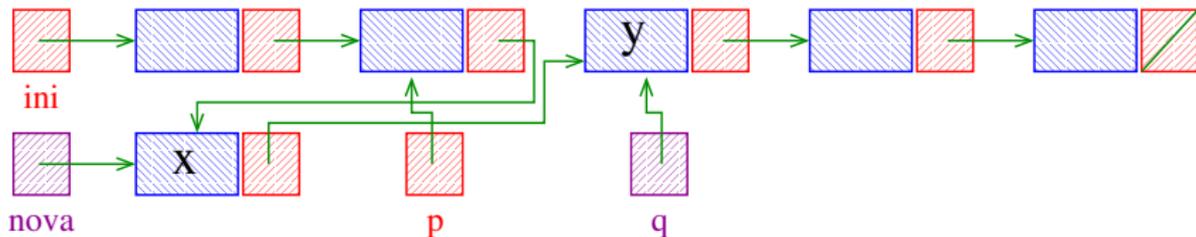
Caso geral: **y** está na lista, mas não na primeira célula.



## Busca e Inserção em uma lista

**Inserir** uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, insere a célula com **x** no final da lista.

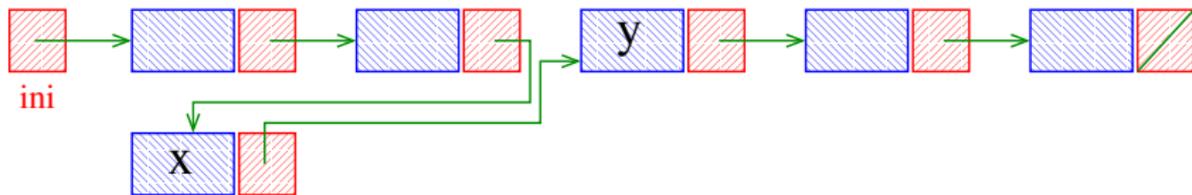
Caso geral: **y** está na lista, mas não na primeira célula.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

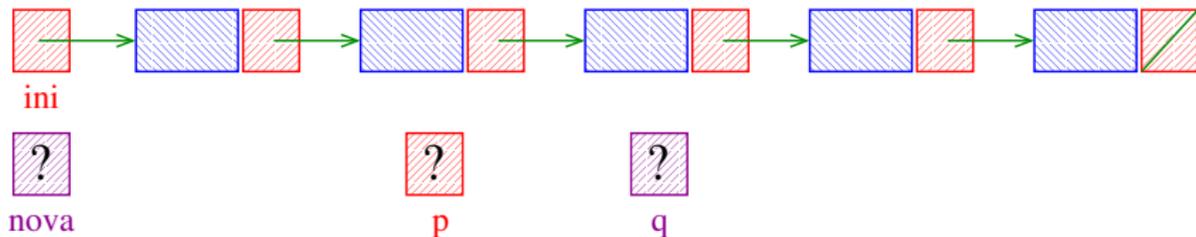
Caso geral:  $y$  está na lista, mas não na primeira célula.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

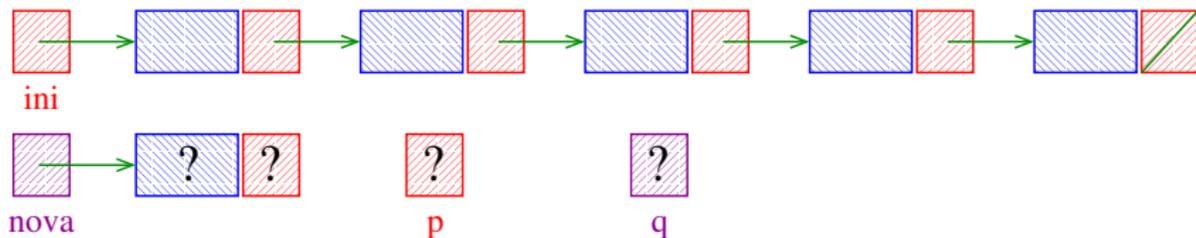
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

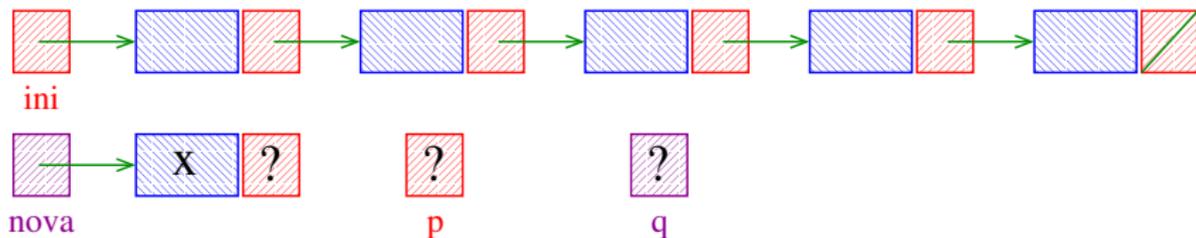
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

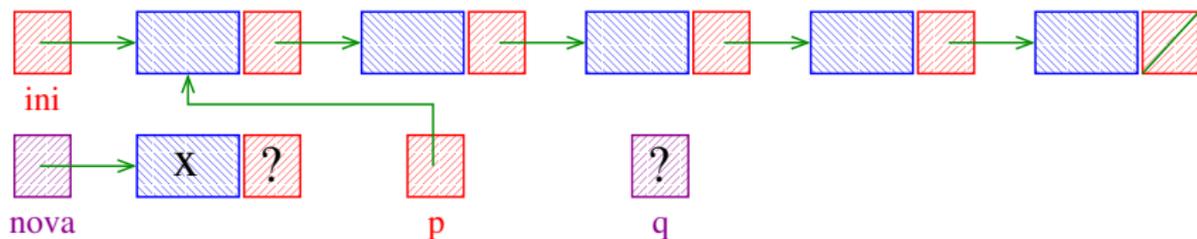
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

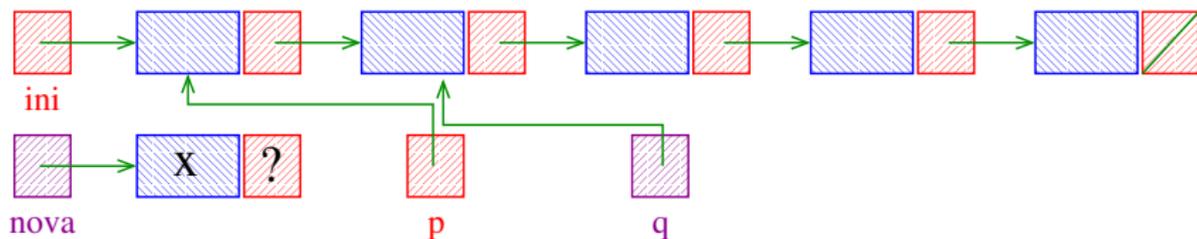
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

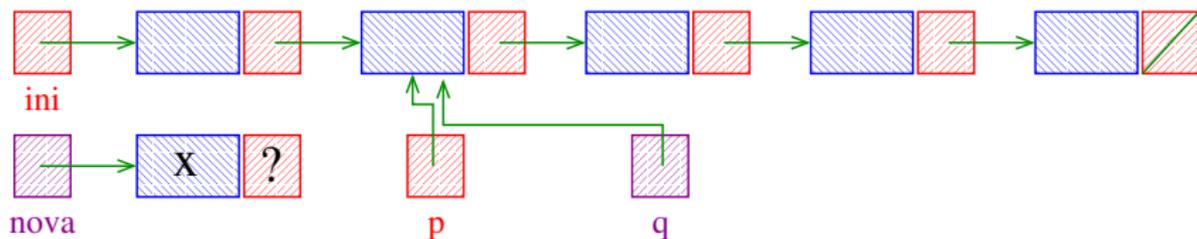
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

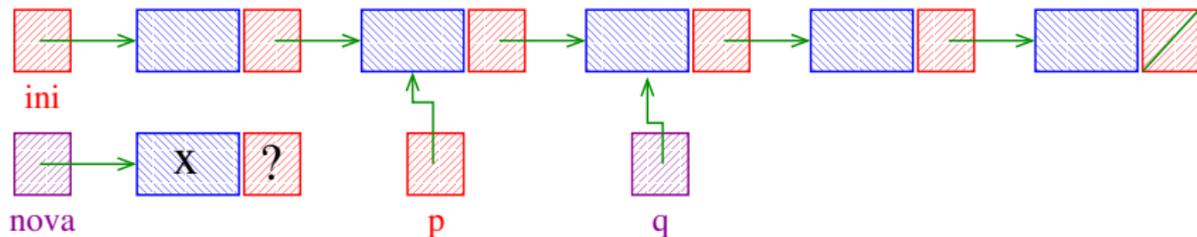
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

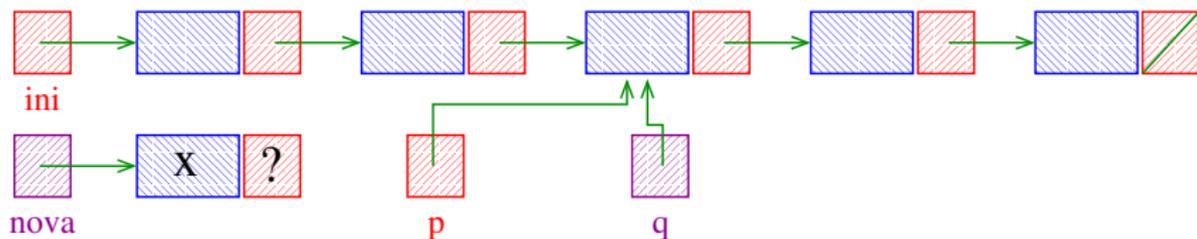
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

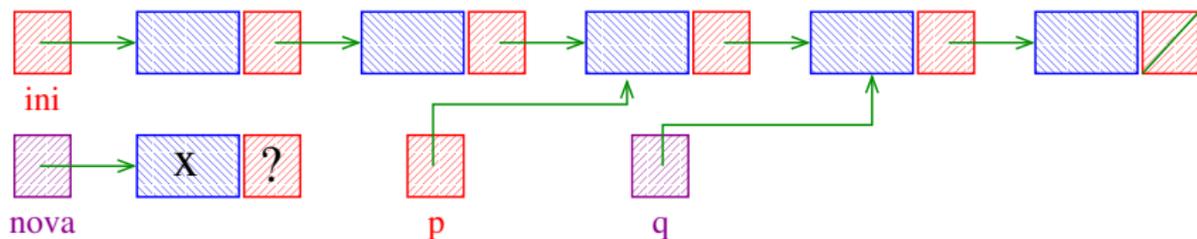
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

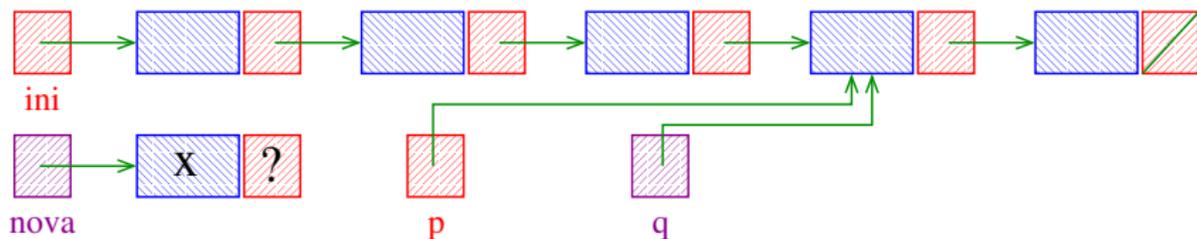
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

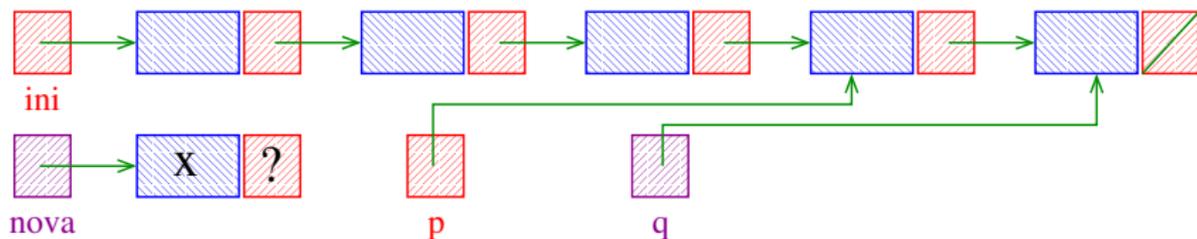
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

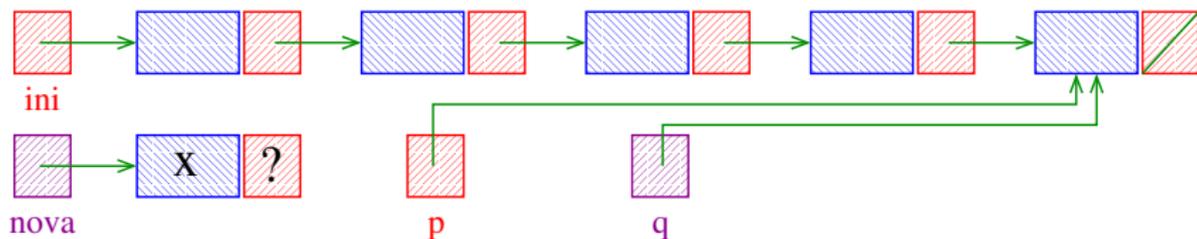
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

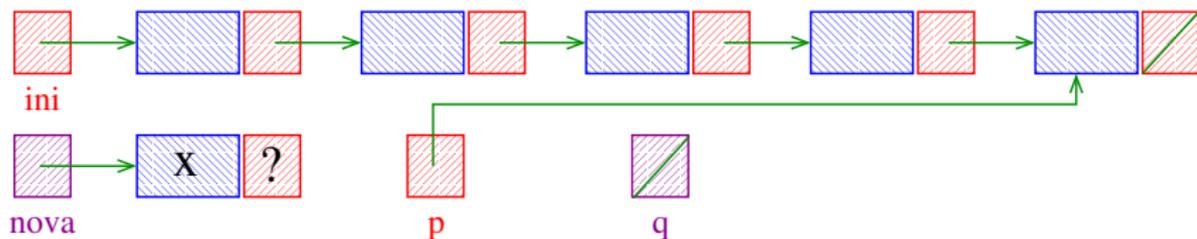
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

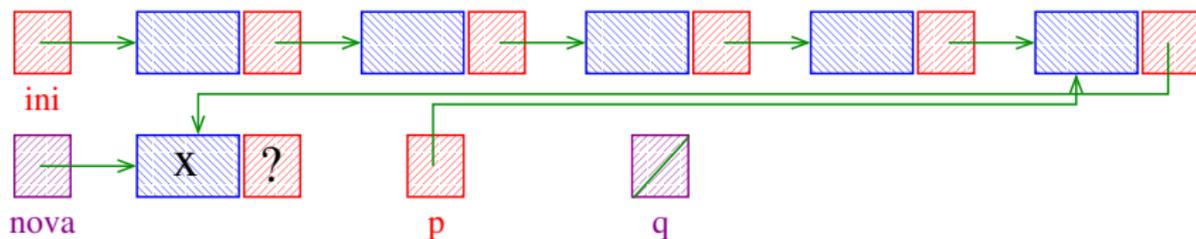
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

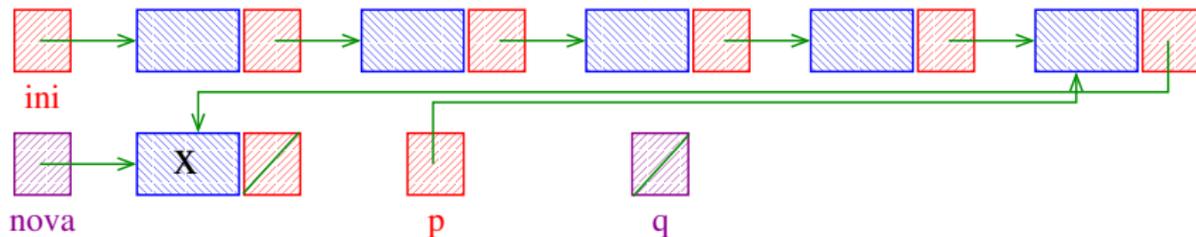
Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

Caso em que  $y$  não aparece na lista.



## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $x$  no final da lista.

Caso em que  $y$  não aparece na lista.

