

## Unidade 1

### Seção 3

Acesse este conteúdo  
pelo smartphone



O que é isso?  
Clique no código e saiba  
mais.

# Algoritmos e Estrutura de Dados

# Webaula 3

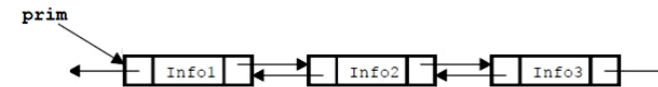
Listas duplamente ligadas

Listas ligadas são estruturas de dados ordenados sequencialmente, com ligação simples entre seus elementos, de forma que cada nó possui um ponteiro direcionando para o próximo elemento que se encontra na lista.

Com a estrutura de listas ligadas não há como percorrer os elementos de forma inversa na lista, a partir do final. Além disso, apesar de ser possível retirar um elemento de uma lista com encadeamento simples, é um processo trabalhoso, pois é necessário percorrer toda a lista para encontrar o elemento anterior, já que o ponteiro sempre indicará o elemento posterior, não o anterior. (Celes, Cerqueira, Rangel, 2004).

## Definição de lista duplamente ligada

Nessa estrutura de dados cada nó apresenta um elemento com informações, um ponteiro direcionando para o próximo elemento e um ponteiro direcionando para o elemento anterior. Dessa forma, é possível acessar tanto o próximo elemento como o elemento anterior da lista e percorrer a lista na ordem inversa, do final até o início.



Fonte: adaptado de CELES; CERQUEIRA; RANGEL. 2004, p. 149.

Na criação de uma lista duplamente ligada, é necessário criar, além do tipo de dado que será utilizado em sua implementação e do ponteiro direcionando para o próximo elemento, o ponteiro direcionando para o elemento anterior da lista.

Clique na imagem para visualizar um exemplo.

```
struct lista {  
    int info;  
    struct lista* ant;  
    struct lista* prox;  
};  
typedef struct lista Lista;
```

Fonte: elaborado pelo autor (2017).

Assim como na lista ligada simples, também precisamos inicializar a lista duplamente ligada para a utilizarmos após a sua declaração.

Uma das possíveis formas de inicialização é criar uma função que retorne a lista como nula.

```
/* Função para retornar uma lista  
vazia */  
Lista* inicializa (void)  
{  
    return NULL;  
}
```

## Adicionar elementos à lista duplamente ligada

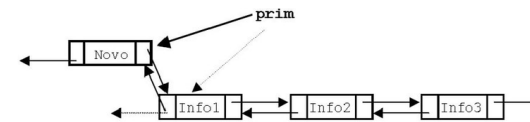
É possível adicionar novos elemento à lista duplamente ligada. Se a lista estiver vazia, tanto o elemento anterior ao inserido como o posterior serão o valor NULL.

Já se a lista tiver elementos inseridos, ao adicionar um novo elemento, o elemento antigo passa a ser o próximo elemento da lista e o anterior passa a receber o valor NULL.

## Adicionar elementos no início da lista duplamente ligada

Explore a galeria e veja como adicionar um novo elemento no início da lista duplamente ligada.

Adição de um novo elemento no início da lista.



Fonte: adaptado de CELES; CERQUEIRA; RANGEL. 2004, p. 150.

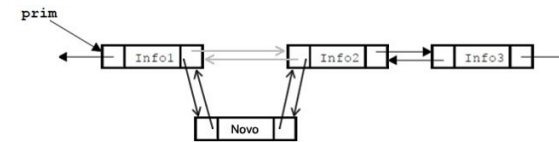




## Adicionar elementos no meio da lista duplamente ligada

Explore a galeria e veja como adicionar um novo elemento no meio da lista duplamente ligada.

Adição de um novo elemento no meio da lista.



Fonte: elaborado pelo autor (2017).

## Adicionar elementos no fim da lista duplamente ligada

Trecho de código que representa a inserção de um novo elemento no final da lista.

```
Lista* inserirFim(Lista* l, int v){
    int cont = 1;
    Lista *p = l;
    Lista* novo = (Lista*)malloc(sizeof(Lista));

    while (p -> prox != NULL) {
        p = p -> prox;
        cont++;
    }

    novo -> info = v;
    novo -> prox = NULL;
    novo -> ant = p;
    p -> prox = novo;
    return l;
}
```

Fonte: elaborado pelo autor (2017).

## Remover elementos da lista duplamente ligada

A função de remoção permite remover um elemento da lista duplamente ligada e, para executá-la, basta conhecer o ponteiro indicando tal elemento.

Para facilitar a localização de um elemento na lista, podemos utilizar a função de busca, em seguida, ajustar o encadeamento da lista e, por fim, liberar o elemento da alocação de memória (Celes, Cerqueira, Rangel, 2004).

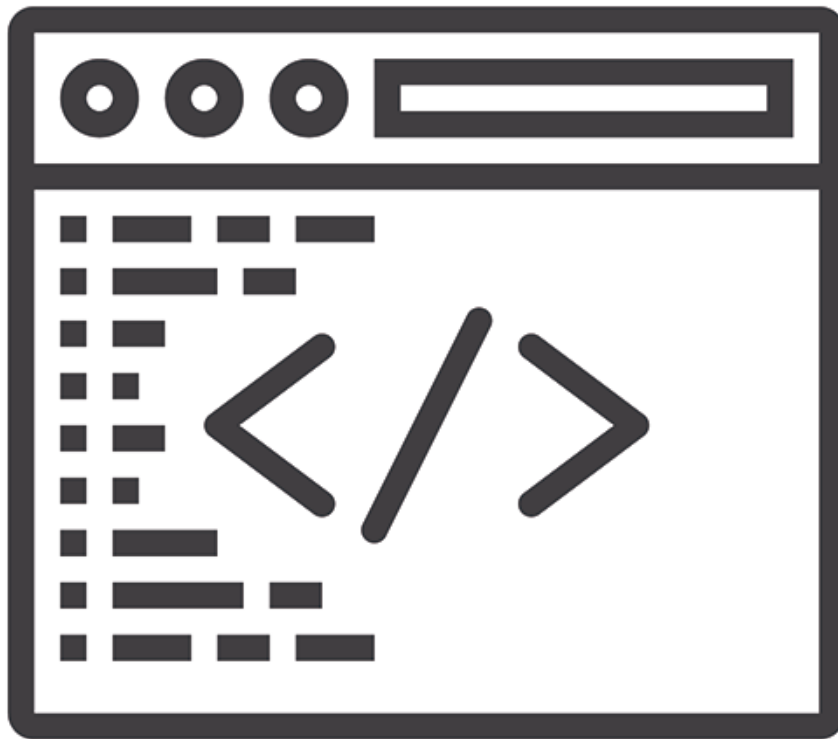
## Remover elementos da lista duplamente ligada

Explore a galeria a seguir e veja as opções de remoção de elementos da lista duplamente ligada.

Trecho de código de uma função de busca.

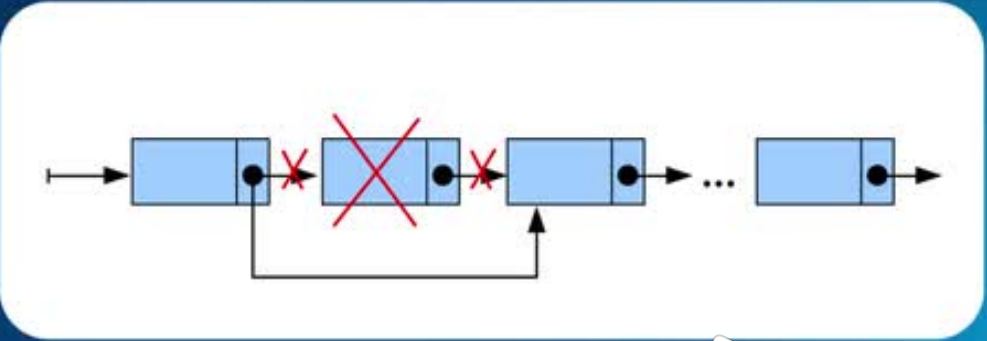
```
Lista* busca(Lista* l, int v){  
    Lista* p;  
  
    for (p = l; p != NULL; p = p -> prox) {  
        if (p -> info == v)  
            return p;  
    }  
  
    return NULL;  
}
```

Fonte: elaborado pelo autor (2017).



Nessa webaula, vimos como podemos realizar operações de adição e remoção de elementos da lista duplamente ligada.

## Vídeo de encerramento



The diagram illustrates the removal of a node from a linked list. It shows a sequence of nodes, each represented as a box divided into two parts: the left part for data and the right part for a pointer to the next node. The first node's pointer points to the second node, and the second node's pointer points to the third node. The second node is marked with a large red 'X', indicating it is to be removed. A new arrow originates from the pointer field of the first node and points directly to the third node, bypassing the second node. This action effectively removes the second node from the sequence. Ellipses (...) between the third and fourth nodes indicate that the list continues beyond what is shown.

▶ Atualizar o ponteiro  
Remover > ~~Elemento~~





Bons estudos!





