

# Listas encadeadas

Em C, uma *lista encadeada* (ou lista ligada ou *linked list*) é composta por um conjunto de estruturas (`struct`) ligadas por ponteiros, de forma a construir uma sequência de itens de tamanho variável e que pode ser facilmente alterada.

Listas encadeadas são estruturas dinâmicas muito utilizadas em programação, por sua flexibilidade e escalabilidade. Elas são frequentemente usadas na construção de TADs como pilhas, filas e listas, por exemplo.

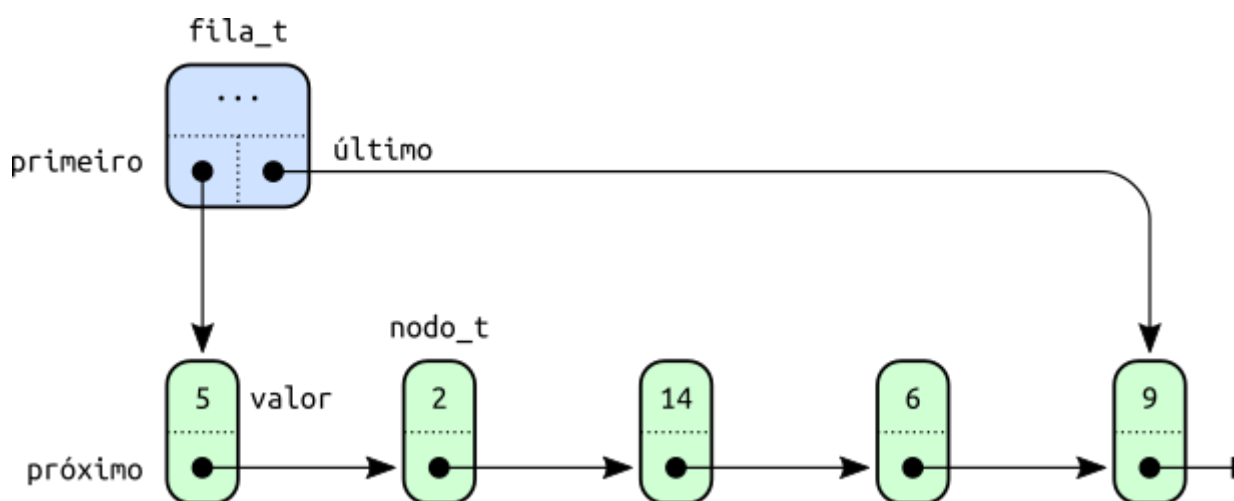
Algumas das principais operações efetuadas sobre uma lista encadeada são:

- criar/destruir uma lista
- inserir item
  - no início
  - no fim
  - em uma posição qualquer
- retirar item
  - do início
  - do fim
  - de uma posição qualquer
- percorrer a lista (para imprimir, contar, procurar, etc)
  - do início ao fim
  - do fim ao início

Esta página apresenta algumas das listas encadeadas mais usadas. Cada uma delas tem características próprias e formas distintas de implementar as operações acima definidas.

## Lista simples

Esta estrutura é frequentemente usada para construir filas (FIFO).

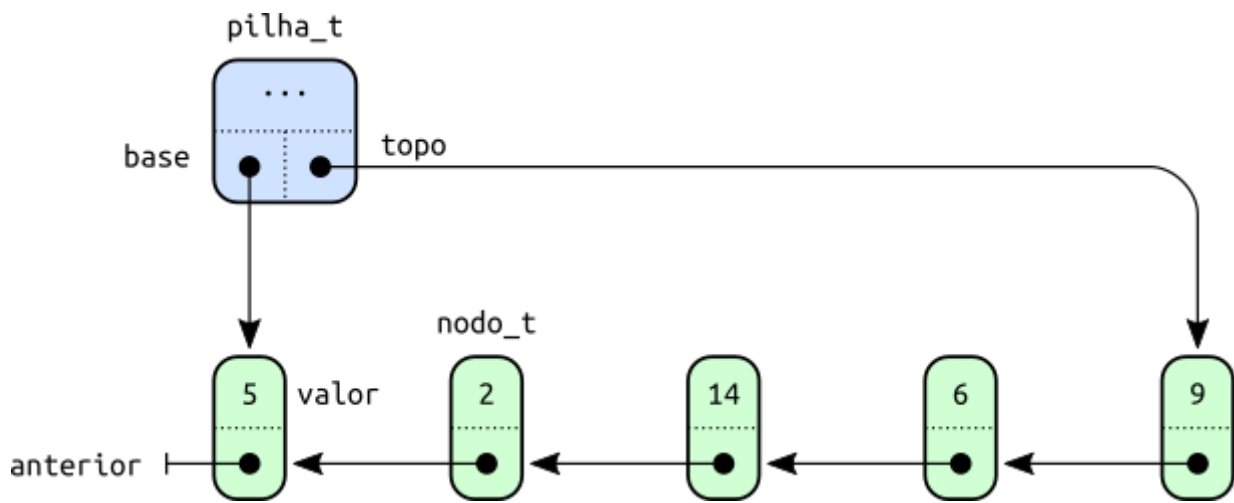


```
struct nodo_t
{
    int valor ;           // valor do item
    struct nodo_t *prox ; // próximo nodo
}
```

```
};  
  
struct fila_t  
{  
    struct nodo_t *primeiro ; // primeiro nodo  
    struct nodo_t *ultimo ; // último nodo  
    int num ; // numero de itens  
};
```

## Lista simples reversa

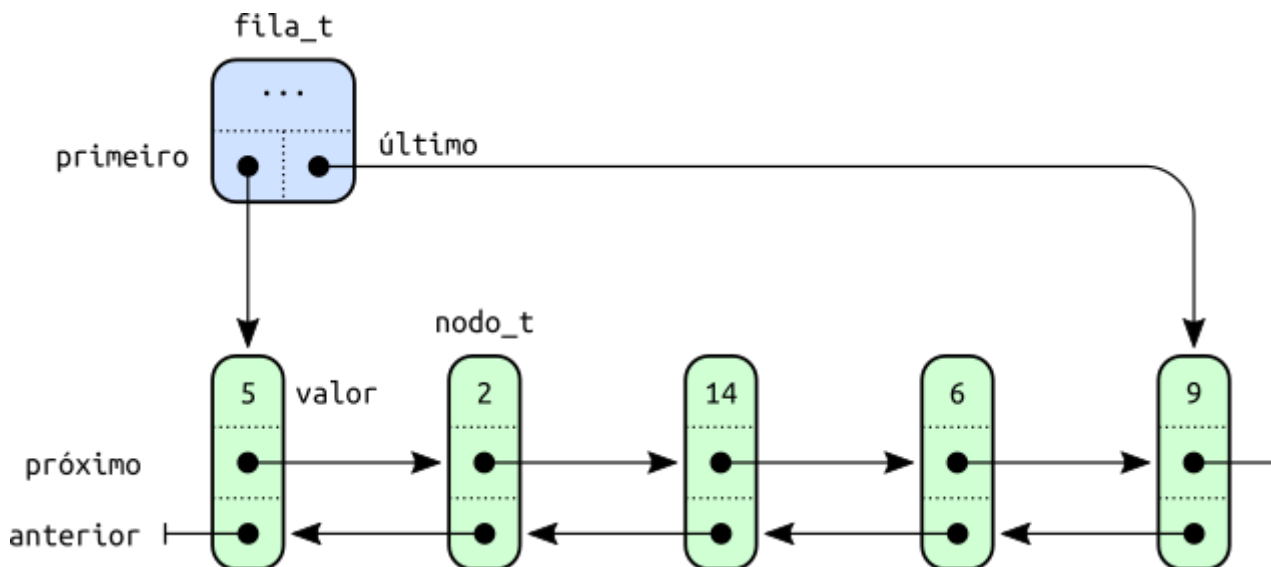
Esta estrutura é frequentemente usada para construir pilhas.



```
struct nodo_t  
{  
    int valor ; // valor do item  
    struct nodo_t *ant ; // nodo anterior  
};  
  
struct pilha_t  
{  
    struct nodo_t *base ; // primeiro nodo  
    struct nodo_t *topo ; // último nodo  
    int num ; // numero de itens  
};
```

## Lista dupla

Esta estrutura pode ser facilmente percorrida nos dois sentidos, sendo útil para construir listas de valores.



```

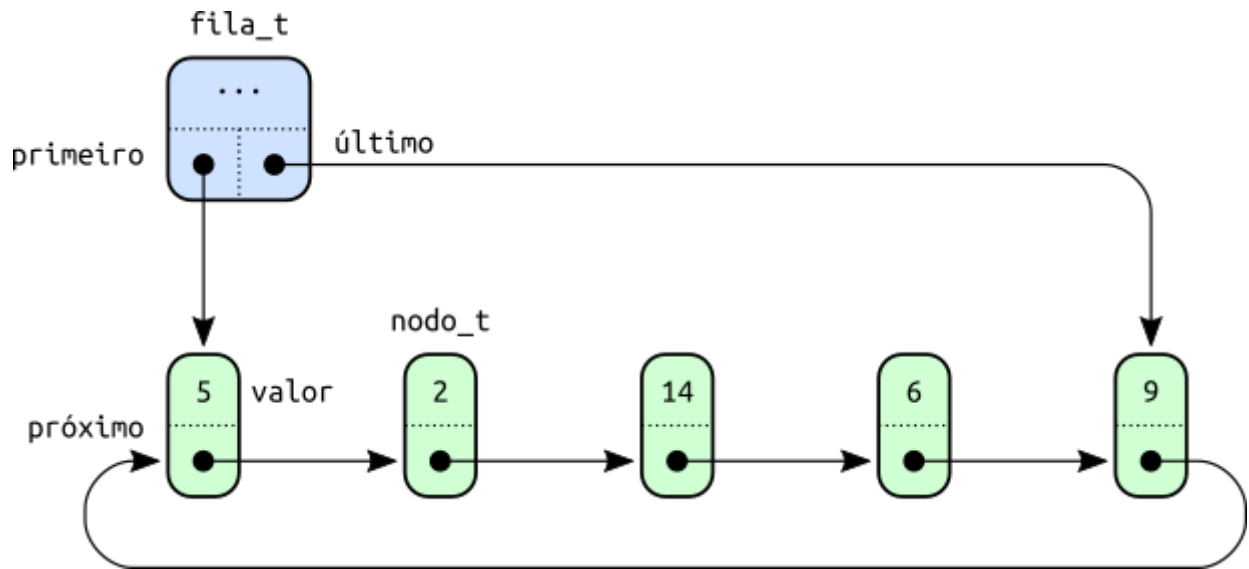
struct nodo_t
{
    int valor ;                // valor do item
    struct nodo_t *ant ;      // nodo anterior
    struct nodo_t *prox ;    // próximo nodo
} ;

struct fila_t
{
    struct nodo_t *primeiro ; // primeiro nodo
    struct nodo_t *ultimo ;  // último nodo
    int num ;                // numero de itens
} ;

```

## Lista simples circular

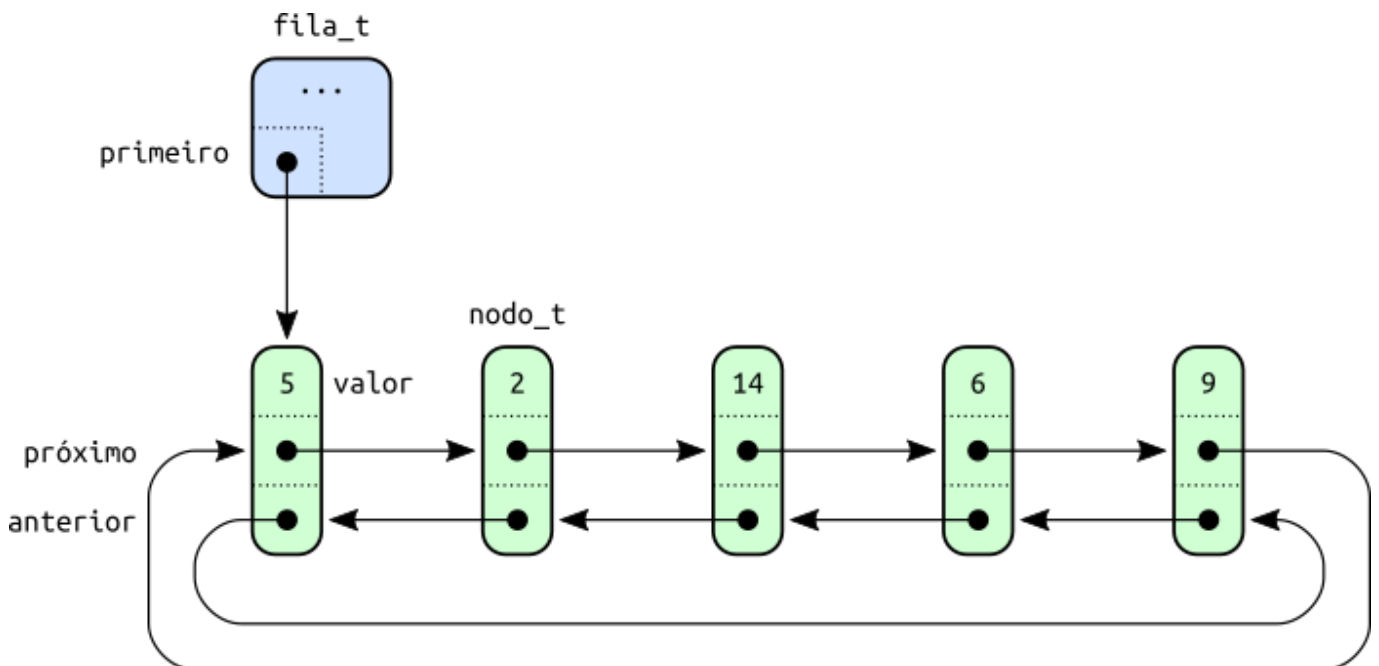
As listas circulares permitem “girar” facilmente a lista. Por exemplo, para transferir o primeiro item da lista para o fim da mesma, basta alterar os ponteiros `primeiro` e `ultimo`.



```
struct nodo_t
{
    int valor ; // valor do item
    struct nodo_t *prox ; // próximo nodo
};

struct fila_t
{
    struct nodo_t *primeiro ; // primeiro nodo
    struct nodo_t *ultimo ; // último nodo
    int num ; // numero de itens
};
```

### Lista dupla circular



```

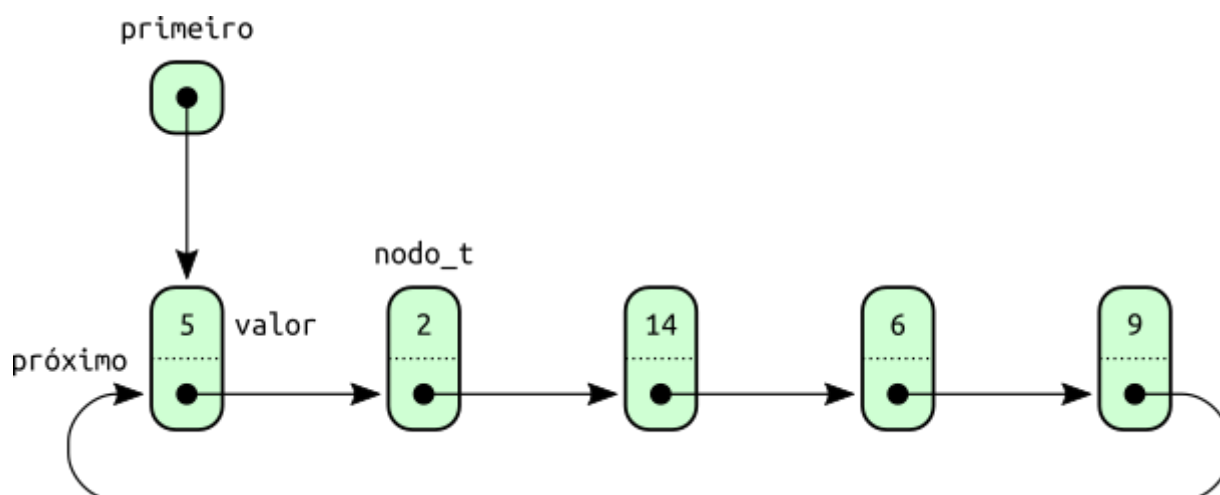
struct nodo_t
{
    int valor ;           // valor do item
    struct nodo_t *ant ; // nodo anterior
    struct nodo_t *prox ; // próximo nodo
} ;

struct fila_t
{
    struct nodo_t *primeiro ; // primeiro nodo
    int num ;                 // numero de itens
} ;

```

## Listas "sem cabeça"

A implementação de listas também pode ser feita usando somente estruturas do tipo `nodo_t`, sem uma estrutura para a cabeça da lista. É uma implementação mais simples, porém menos flexível.



```

struct nodo_t
{
    int valor ;           // valor do item
    struct nodo_t *ant ; // nodo anterior
    struct nodo_t *prox ; // próximo nodo
} ;

struct nodo_t *primeiro ; // início da lista

```

## Exercícios

- [Esta página](#) do prof. Paulo Feofiloff (IME/USP) tem uma ótima apresentação de listas encadeadas, com muitos exercícios.

From:

<https://wiki.inf.ufpr.br/maziero/> - **Prof. Carlos Maziero**

Permanent link:

[https://wiki.inf.ufpr.br/maziero/doku.php?id=prog1:listas\\_encadeadas](https://wiki.inf.ufpr.br/maziero/doku.php?id=prog1:listas_encadeadas)

Last update: **2024/10/23 14:07**

