

Aula 9

Listas Circulares e Duplamente Encadeadas

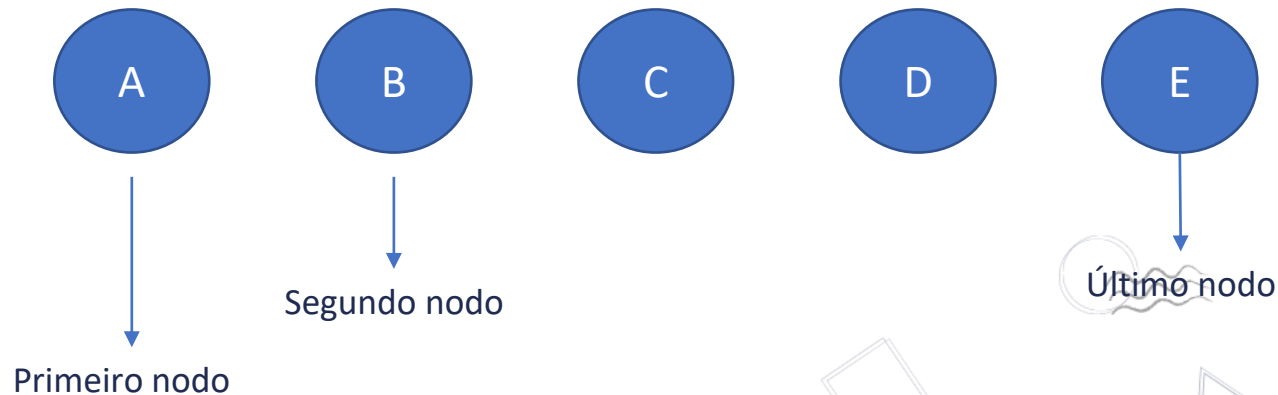
Prof.(a): William P. Santos Júnior

Listas Lineares

Recapitulando

Lista linear é um conjunto de elemento do mesmo tipo, que são denominados “nodos”, entre os quais existe uma relação de ordem linear, que é definido somente por sua posição em relação aos outros nodos. Cada nodo, pode conter um dado primitivo ou um dado composto e quando a lista apresenta todos os nodos com a mesma estrutura interna essa lista é chamada de lista homogênea o contrário chamamos de lista heterogênea.

Toda lista apresenta um nodo que encabeça a lista e que é o primeiro nodo da lista, a partir deste segue a sequencia conforme ordem definida entre eles.

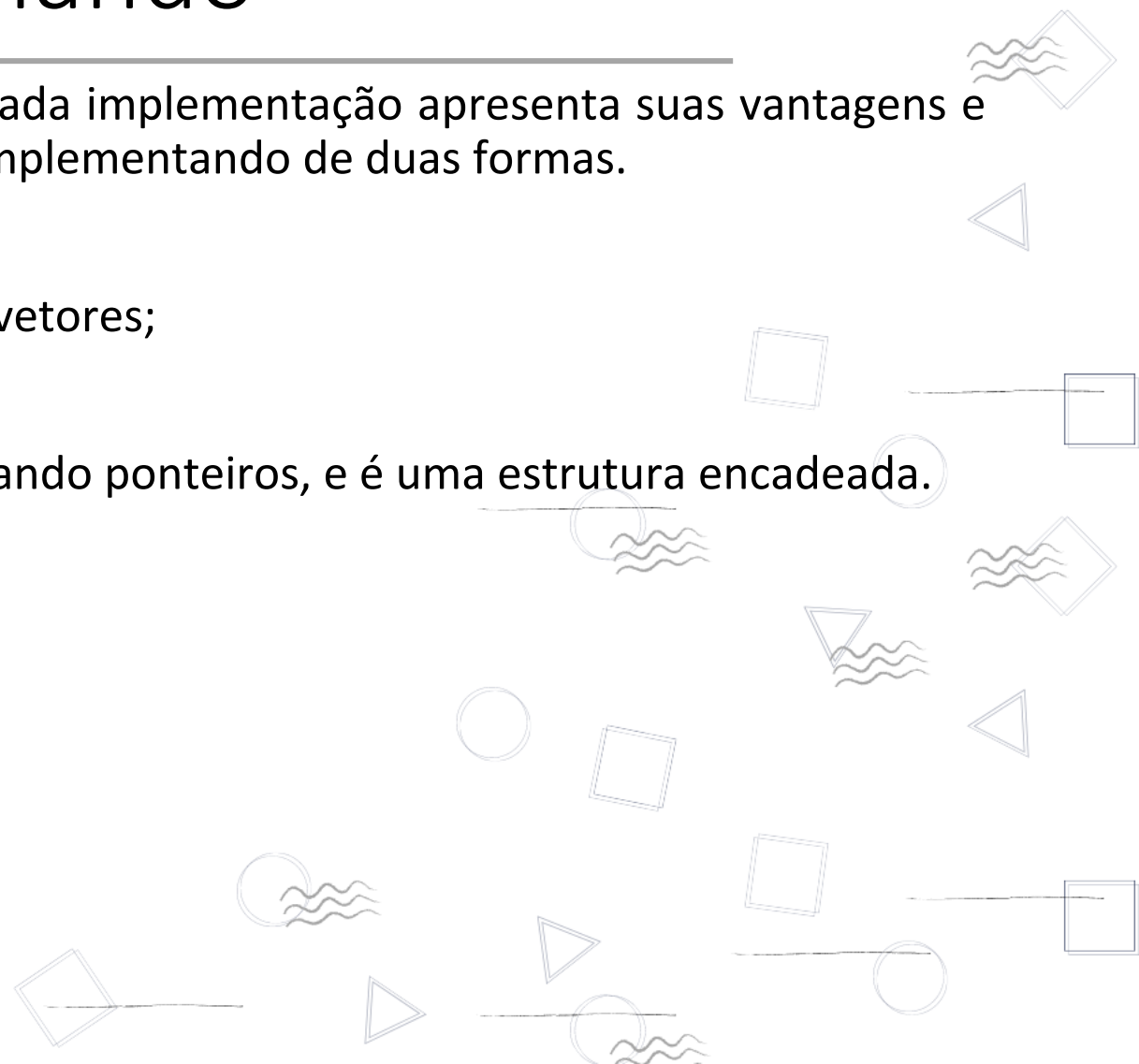


Listas Lineares

Recapitulando

Há varias maneiras de implementar listas lineares, cada implementação apresenta suas vantagens e desvantagens particulares. Vimos como usar listas implementando de duas formas.

- Usando alocação sequencial e estática: Utilizando vetores;
- Usando alocação não sequencial e dinâmica: Utilizando ponteiros, e é uma estrutura encadeada.



Listas Lineares

Recapitulando

Operações Sobre Listas Lineares:

- Criação de uma lista:
 - Primeira operação a ser executada, através da qual são alocadas as variáveis necessárias para a definição da lista.
- Inserção de um nodo:
 - É como a lista será formada, inserindo um a um cada nodo. A inserção pode ser feita em qualquer posição da lista (início, meio ou fim).
- Exclusão de um nodo:
 - É a forma de como excluir um nodo da lista, a exclusão também pode ser feita em qualquer posição da lista (início, meio ou fim).
- Acesso a um nodo:
 - Usado para consultar ou alterar valores internos.
- Destruição da lista:
 - É utilizada a destruição quando a lista existente não é mais necessária.

Listas Lineares

Recapitulando

Operações Sobre Listas Lineares:

- Criação de uma lista:
 - Primeira operação a ser executada, através da qual são alocadas as variáveis necessárias para a definição da lista.
- Inserção de um nodo:
 - É como a lista será formada, inserindo um a um cada nodo. A inserção pode ser feita em qualquer posição da lista (início, meio ou fim).
- Exclusão de um nodo:
 - É a forma de como excluir um nodo da lista, a exclusão também pode ser feita em qualquer posição da lista (início, meio ou fim).
- Acesso a um nodo:
 - Usado para consultar ou alterar valores internos.
- Destruição da lista:
 - É utilizada a destruição quando a lista existente não é mais necessária.

Listas Encadeadas

Em uma lista linear encadeada, cada elemento da lista possui, além do espaço para armazenar a informação, um espaço que guarda uma referência da localização na memória do próximo elemento da lista, ou do anterior, dependendo do caso.

Uma forma de implementar estruturas dinâmicas é através do encadeamento, onde os nodos são ligados entre si para indicar a ordem existente entre eles. Assim, a ordem dos nodos é definida por uma informação contida no próprio nodo, que informa qual o próximo nodo da lista. Esta informação está contida em um campo denominado campo de elo. Os nodos de uma lista encadeada podem estar alocados em quaisquer posições na memória, contiguas ou não, uma vez que a ordem é fornecida explicitamente através do campo de elo, sendo totalmente independente de sua posição física. Dessa forma, a contiguidade de uma lista linear encadeada é lógica.

Listas Encadeadas

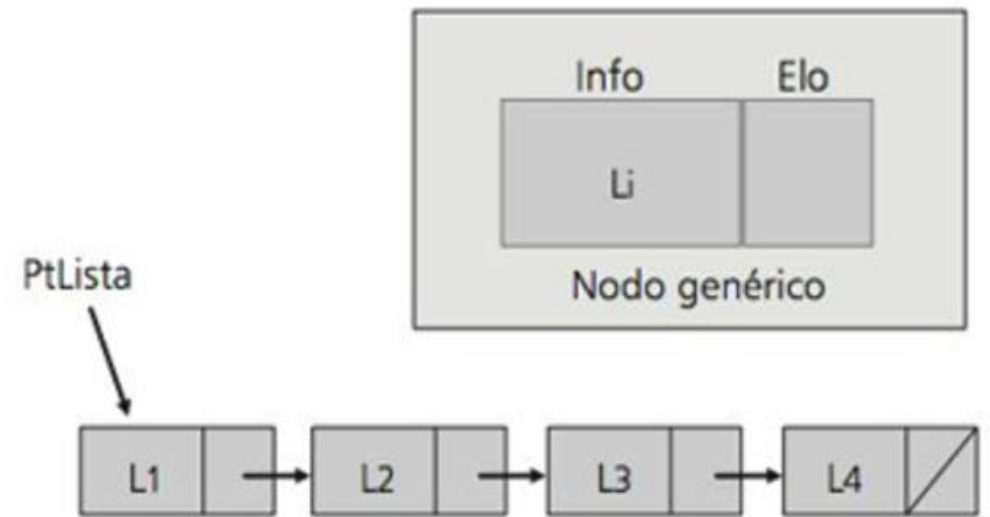
Onde:

PtLista: Ponteiro para o primeiro nodo da lista, por ele a lista poderá ser acessada.

Info: dado armazenado na lista

Elo: faz o encadeamento entre os nodos.

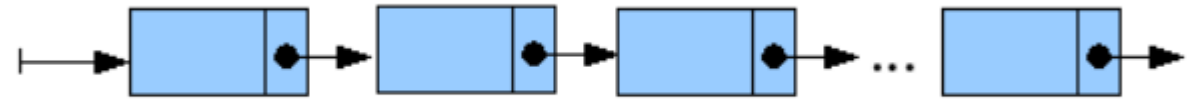
L4: Na L4 esse no “elo”, representa nulo e deve ser passível de teste para que o algoritmo detecte quando o final de lista foi alcançado.



Listas Encadeadas

Estrutura `Nodo`, foi declarado duas variáveis, uma tipo inteiro, "info", que irá armazenar a informação que será guardada e uma do tipo `struct Nodo`, que é um ponteiro para o próximo nodo da lista.

Na estrutura `ListaSimplesEnc`, a variável "`*prim`" é um ponteiro para a primeira posição da estrutura.

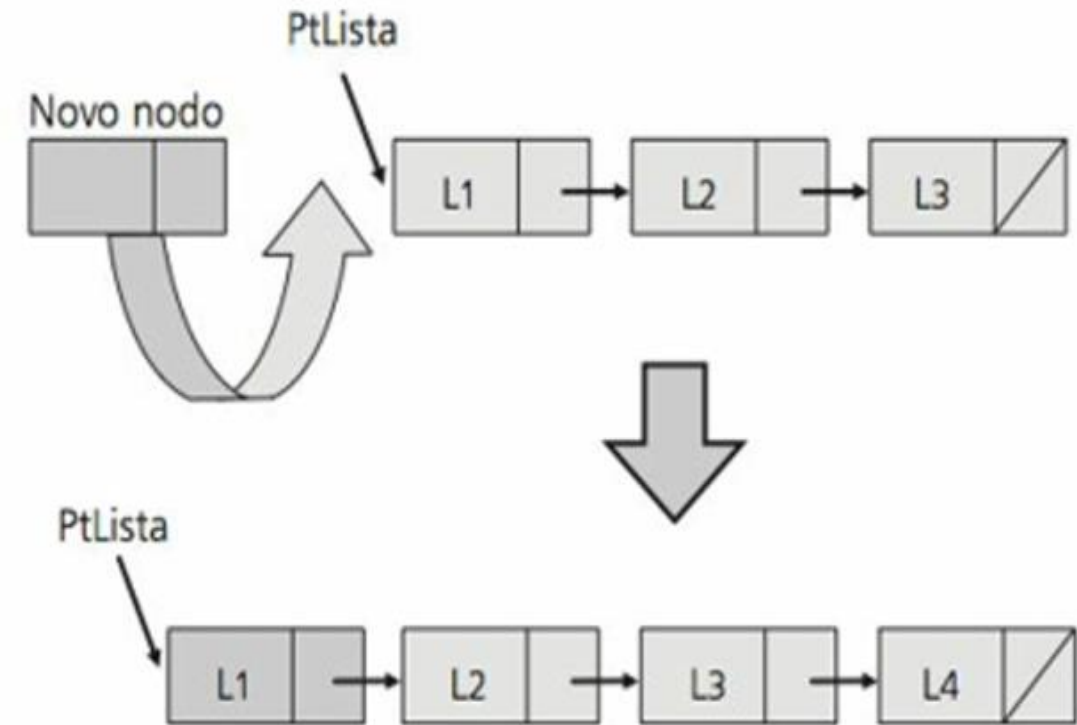


```
6 struct Nodo {  
7  
8     int info;  
9     struct Nodo *prox;  
10 };  
11  
12 struct ListaSimplesEnc {  
13  
14     struct Nodo *prim;  
15 };
```


Listas Encadeadas

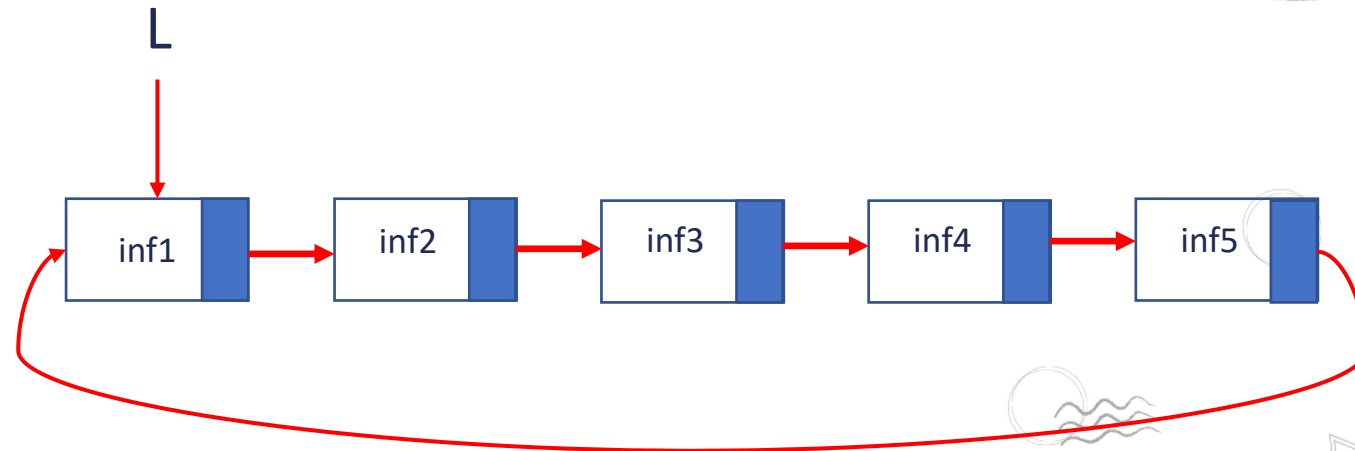
Para inserir um novo nodo em uma lista, deve-se manipular a referência ao próximo nó da lista.

```
35 void inserirIni(struct ListaSimplesEnc *pList, int v){  
36     struct Nodo *novo;  
37     novo = (struct Nodo*) malloc (sizeof (struct Nodo));  
38     novo -> info = v;  
39     novo -> prox = pList -> prim;  
40     pList -> prim = novo;  
41 }
```

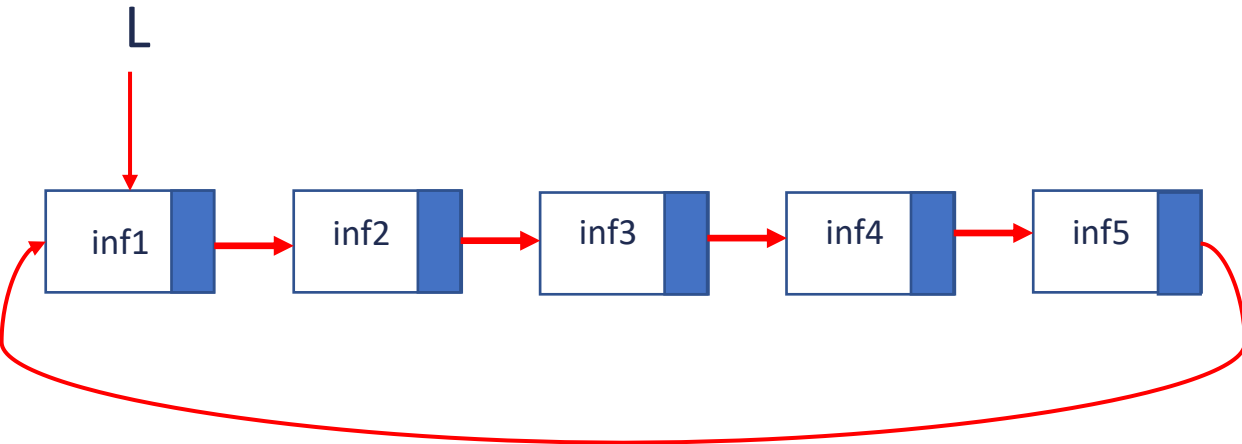


Listas Encadeadas Circulares

- Possui a mesma estrutura de uma lista simplesmente Encadeada;
- Diferença é que o último item da lista aponta para primeiro item da lista;
- Passamos a ter o nó atual ao invés do nó cabeça da lista.



Listas Encadeadas Circulares

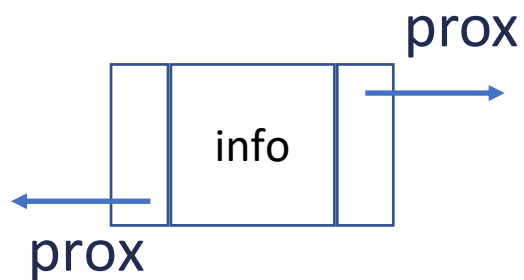


```
1  /* insere no início da Lista*/
2
3  Lista *insereLista(Lista *l, int i)
4  {
5      Lista *aux;
6      Lista *novo = (Lista*) malloc(sizeof(Lista));
7      novo->info= i;
8      novo->prox= l;
9      if(l == NULL)
10         novo->prox = novo;
11     else
12     {
13         aux = l;
14         do
15         {
16             aux = aux->prox;
17         }while(aux->prox != l);
18         aux->prox = novo;
19     }
20     return novo;
21 }
```

Listas Duplamente Encadeadas

Duplamente encadeada

Em uma lista duplamente encadeada, cada elemento(nó), possui além de um espaço para armazenar a informação, um espaço que guarda o endereço de memória do próximo elemento e outro espaço para guardar o endereço do elemento anterior.

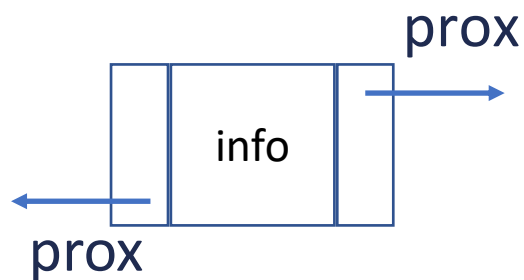
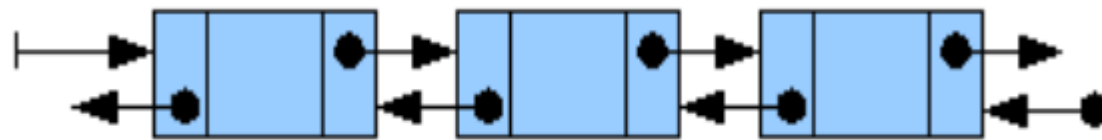


```
1 struct nodo{  
2     int inf;  
3     struct nodo *ant;  
4     struct nodo *prox;  
5 }no;
```

Listas Duplamente Encadeadas

Duplamente encadeada

Em uma lista duplamente encadeada, cada elemento(nó), possui além de um espaço para armazenar a informação, um espaço que guarda o endereço de memória do próximo elemento e outro espaço para guardar o endereço do elemento anterior.



```
1 struct nodo{
2     int inf;
3     struct nodo *ant;
4     struct nodo *prox;
5 }no;
```

Atividades

Em Duplas fazer a atividade pré-aula – Valor 10Pts.



UniEVANGÉLICA
UNIVERSIDADE EVANGÉLICA DE GOIÁS

