unioeste

Universidade Estadual do Oeste do Paraná

Listas Simplesmente Encadeada Implementação

Prof.ª Ms. Fabiane Sorbar

Criação da Estrutura da Lista / Criação da Cabeça

unioeste
Universidade Estadual do Oeste do Paraná

```
4  struct  Node{//criando o tipo do elemento da lista
5      int num;//dado
6      struct Node *prox;
7  };
8  typedef struct Node node;//apelido
9  typedef node *LISTA;//declaração ponteiro da cabeça da lista
10
```

Criação da Lista

unioeste
Universidade Estadual do Oeste do Paraná

```
12  LISTA* criarLista(){
13      LISTA *inicio =(LISTA*) malloc (sizeof(LISTA));//aloca cabeça da lista
14      if(inicio != NULL){
15          *inicio=NULL;
16      }else{
17          printf("Erro na alocação...\n");
18          exit(0);
19      }
20      return inicio;
21  }
```

Insere Início

unioeste
Universidade Estadual do Oeste do Paraná

```
23  void insereInicio(LISTA* lista){
24      node *novo = (node*) malloc (sizeof(node));//declara e aloca
25      if(novo == NULL){
26          printf("Erro na alocação...\n");
27          exit(0);
28      }
29      printf("Informe valor do Novo elemento\n");
30      scanf("%d",&novo->num);
31      novo->prox=(*lista);
32      *lista = novo;
33  }
```

Insere no Final

unioeste
Universidade Estadual do Oeste do Paraná

```
void insereFinal(LISTA* lista){
    node *novo = (node*) malloc (sizeof(node));//declara e aloca
    if(novo == NULL){
        printf("Erro na alocação...\n");
        exit(0);
    }
    printf("Informe valor do Novo elemento\n");
    scanf("%d",&novo->num);
    novo->prox = NULL; // novo final da lista
    if ((*lista) == NULL){//se lista vazia
        *lista = novo;
    }else{
        node *tmp;// no temporário
        tmp = (*lista);
        while (tmp->prox != NULL){//busca o ultimo elemento
            tmp = tmp->prox;
        }
        tmp->prox = novo;
    }
}
```

Exibir Lista

unioeste
Universidade Estadual do Oeste do Paraná

```
35  void exibe(LISTA* lista){
36      if( (*lista) == NULL){
37          printf("LISTA VAZIA...\n");
38      }else{
39          node *tmp;// no temporário
40          tmp = (*lista);
41          while (tmp != NULL){
42              printf("%5d", tmp->num);
43              tmp = tmp->prox;
44          }
45      }
46      printf("\n");
47  }
```

Libera (desaloca) Lista

unioeste
Universidade Estadual do Oeste do Paraná

```
69  void libera(LISTA* lista){
70      if((*lista) == NULL){
71          printf("LISTA VAZIA...\n");
72      }else{
73          node *tmp;
74          while ((*lista) != NULL){
75              tmp = *lista;
76              *lista = (*lista)->prox;
77              free(tmp);
78          }
79          *lista=NULL;
80      }
81  }
```

Remove Início

unioeste
Universidade Estadual do Oeste do Paraná

```
105 int removeInicio(LISTA* lista){
106     if((*lista) != NULL){
107         node *proxNode, *tmp;
108         tmp = *lista;
109         *lista = tmp->prox;
110         free(tmp);
111         printf("Elemento removido com sucesso...\n");
112     }else{
113         printf("Lista ja esta vazia... \n");
114     }
115 }
```

Remove Fim

unioeste
Universidade Estadual do Oeste do Paraná

```
117 int removeFinal(LISTA* lista){
118     node *antNode, *tmp;
119     tmp = *lista;
120     if ((*lista) != NULL && tmp->prox == NULL){//se só existe um elemento na lista
121         *lista = NULL;
122         free (tmp);
123         printf("Elemento removido com sucesso...\n");
124     }else if((*lista) != NULL){
125         while (tmp->prox !=NULL){
126             antNode = tmp;
127             tmp = tmp->prox;
128         }
129         antNode->prox = NULL;
130         free(tmp);
131         printf("Elemento removido com sucesso...\n");
132     }else{
133         printf("Lista ja esta vazia... \n");
134     }
135 }
```