

Tópicos Avançados em Estrutura de Dados

Atividade 9

Bruna Galastri Guedes	18.00189-0
Daniel Ughini Xavier	18.00022-3
Rodolfo Cochi Bezerra	18.00202-0
Vítor Martin Simoni	18.00050-9
Leonardo Cury Haddad	18.00442-3
Leonardo de Barros Rodrigues	18.02401-7

01/06/2020

Questão 1

Segue o programa em C que implementa as operações básicas de Listas Ligadas:

```
#include <stdio.h>
#include <stdlib.h>

typedef struct Elemento
{
    int info;
    struct Elemento *prox;
} TElemento;

typedef struct Lista
{
    TElemento *inicio;
    TElemento *fim;
} TLista;

void inicializa_lista(TLista *lista)
{
    lista->inicio = NULL;
    lista->fim = NULL;
}

int inserir_elemento(TLista *lista)
{
    TElemento *novo = malloc(sizeof(*lista));
    printf("\n Informe valor: ");
    scanf("%d", &novo->info);
    novo->prox = NULL;

    if (lista->inicio == NULL)
    {
        lista->inicio = novo;
        lista->fim = novo;
    }
    else
    {
        lista->fim->prox = novo;
        lista->fim = novo;
    }
}

void apresentar_elementos(TLista *lista)
{
    TElemento *aux;
    aux = lista->inicio;

    if (aux == NULL)
    {
        printf("\n Lista Vazia!");
    }
    else
    {

```

```

        while (aux != NULL)
        {
            printf("\n Valor: %d", aux->info);
            aux = aux->prox;
        }
    }
}

void excluir_elemento(TLista *lista)
{
    int valor;
    TElemento *anterior, *atual;

    printf("\n Informe elemento a ser excluido: ");
    scanf("%d", &valor);

    if(lista->inicio == NULL)
    {
        printf("\n Lista Vazia!");
    }
    else
    {
        anterior = lista->inicio;
        atual = lista->inicio;
        while(atual != NULL)
        {
            if(atual->info == valor)
            {
                if(atual == lista->inicio)
                {
                    lista->inicio = lista->inicio->prox;
                    free(atual);
                    break;
                }
                else
                {
                    if(atual == lista->fim)
                    {
                        lista->fim = anterior;
                    }

                    anterior->prox = atual->prox;
                    free(atual);
                    break;
                }
            }
            else
            {
                anterior = atual;
                atual = atual->prox;
            }
        }
    }
}

int main()
{

```

```

TLista L1;
inicializa_lista(&L1);

int opcao;
do
{
    printf("\nOpcoes: \n\n");
    printf("1 - Inserir novo elemento \n");
    printf("2 - Listar todos os elementos \n");
    printf("3 - Excluir elemento \n");
    printf("0 - para sair \n\n");
    scanf("%d", &opcao);

    switch (opcao)
    {
        case 1: inserir_elemento(&L1); break;
        case 2: apresentar_elementos(&L1); break;
        case 3: excluir_elemento(&L1); break;
        case 0: break;
    }
}while (opcao != 0);

return 0;
}

```