

Código da pilha

Murilo Dantas

Operações numa pilha

1ª operação
A pilha está vazia

topo
null

2ª operação
Inserção do nº 9 na pilha

topo
800
9
null

3ª operação
Inserção do nº 3 na pilha

topo
650
3
800

↓
800
9
null

4ª operação
Inserção do nº 5 na pilha

topo
750
5
650

↓
650
3
800

↓
800
9
null

5ª operação
Remoção da pilha

topo
650
3
800

↓
800
9
null

6ª operação
Remoção da pilha

topo
800
9
null

Código

```
#include <iostream.h>
#include <conio.h>

//Definindo o registro que representará cada elemento
//cada elemento da pilha

struct PILHA
{
    int num;
    PILHA *prox;
};

void main()
{
    // a pilha está vazia, logo,
    // o ponteiro topo tem o valor null
    // as operações de inserção e remoção
    // acontecem no TOPO
    PILHA *topo = NULL;
    // o ponteiro aux é um ponteiro auxiliar
    PILHA *aux;
    // apresentando o menu de opções
    int op;
    do
    {
        clrscr();
        cout<<"\nMENU DE OPÇÕES\n";
        cout<<"\n1 - Inserir na pilha";
        cout<<"\n2 - Consultar toda a pilha";
        cout<<"\n3 - Remover da pilha";
        cout<<"\n4 - Esvaziar a pilha";
        cout<<"\n5 - Sair";
        cout<<"\nDigite sua opção: ";
        cin>>op;
```

```
if (op < 1 || op > 5)
    cout<<"Opção inválida!!";
if (op == 1)
{
    cout<<"Digite o número a ser inserido na
    ↩ pilha: ";
    PILHA *novo = new PILHA();
    cin>>novo->num;
    novo->prox = topo;
    topo = novo;
    cout<<"Número inserido na pilha!!";
}
if (op == 2)
{
    if (topo == NULL)
    {
        // a pilha está vazia
        cout<<"Pilha vazia!!";
    }
    else
    {
        // a pilha contém elementos e
        // estes serão mostrados
        // do último inserido ao primeiro
        cout<<"\nConsultando toda a pilha\n";
        aux = topo;
        while (aux != NULL)
        {
            cout<<aux->num<<" ";
            aux = aux->prox;
        }
    }
}
```

Código

```
if (op == 3)
{
    if (topo == NULL)
    {
        // a pilha está vazia
        cout<<"Pilha vazia!!";
    }
    else
    {
        // a pilha contém elementos
        // e o último elemento inserido
        // será removido
        aux = topo;
        cout<<"Número "<<topo->num<<" ↪ removido";
        topo = topo->prox;
        delete(aux);
    }
}
if (op == 4)
{
    if (topo == NULL)
    {
        // a pilha está vazia
        cout<<"Pilha vazia!!";
    }
    else
    {
        // a pilha será esvaziada
        aux=topo;
```

```
while(aux!=NULL)
{
    topo = topo->prox;
    delete(aux);
    aux=topo;
}
cout<<"Pilha esvaziada";
}
}
getch();
}
while (op != 5);
}
```

Perguntas?

Bibliografia da aula

- ASCENCIO, A. F. G.; ARAÚJO, G. S. Estrutura de dados. Algoritmos, análise da complexidade e implementação em Java e C/C++. Pearson. 2010.