



FAESA  
CENTRO UNIVERSITÁRIO ESPÍRITO-SANTENSE/FAESA  
SISTEMAS DE INFORMAÇÃO

TASIO BRENO SANTOS GUIMARAES

Estrutura de dados: Desafio da Pilha

Trabalho Eventual

Vila Velha  
2023  
TASIO BRENO SANTOS GUIMARAES

## Sumário

2

- 1. Prints do código . . . . . 3
- 2. Arquivo com o código . . . . . 4

The image shows a Visual Studio Code editor with a C program for a stack. The code is as follows:

```

7  typedef struct {
8      int topo; //posição do elemento*/
9      int elementos[MAX];
10 } Pilha;
11
12 // Função para criar a pilha
13 void criaPilha(Pilha *p) {
14     p->topo = -1;
15 }
16
17 // Função para verificar se a pilha está vazia
18 int estaVazia(Pilha *p) {
19     return (p->topo == -1);
20 }
21
22 // Função para verificar se a pilha está cheia
23 int estaCheia(Pilha *p) {
24     return (p->topo == MAX-1);
25 }
26
27 int main() {
28     Pilha p;
29     criaPilha(&p);
30     int op;
31     do {
32         system("clear");
33         printf("\nMenu de opções:\n");
34         printf("1 - Empilhar um elemento\n");
35         printf("2 - Desempilhar o elemento do topo\n");
36         printf("3 - Consultar o elemento do topo\n");
37         printf("4 - Exibir todos os elementos da pilha\n");
38         printf("5 - Remover todos os elementos da pilha\n");
39         printf("6 - Sair do programa\n");
40         printf("Digite o valor a ser empilhado: ");
41         scanf("%d", &op);
42         switch (op) {
43             case 1:
44                 printf("Escolha uma opção:\n");
45                 printf("1 - Empilhar um elemento\n");
46                 printf("2 - Desempilhar o elemento do topo\n");
47                 printf("3 - Consultar o elemento do topo\n");
48                 printf("4 - Exibir todos os elementos da pilha\n");
49                 printf("5 - Remover todos os elementos da pilha\n");
50                 printf("6 - Sair do programa\n");
51                 scanf("%d", &op);
52                 break;
53             case 2:
54                 if (!estaVazia(&p)) {
55                     printf("Elemento desempilhado: %d\n", p.topo);
56                     p.topo--;
57                 } else {
58                     printf("A pilha está vazia.\n");
59                 }
60                 break;
61             case 3:
62                 if (!estaVazia(&p)) {
63                     printf("Elemento no topo: %d\n", p.topo);
64                 } else {
65                     printf("A pilha está vazia.\n");
66                 }
67                 break;
68             case 4:
69                 if (!estaVazia(&p)) {
70                     printf("Elementos na pilha:\n");
71                     for (int i = p.topo; i >= 0; i--) {
72                         printf("%d ", p.elementos[i]);
73                     }
74                     printf("\n");
75                 } else {
76                     printf("A pilha está vazia.\n");
77                 }
78                 break;
79             case 5:
80                 if (!estaVazia(&p)) {
81                     printf("Removendo elementos da pilha...\n");
82                     while (!estaVazia(&p)) {
83                         p.topo--;
84                     }
85                     printf("Pilha vazia.\n");
86                 } else {
87                     printf("A pilha está vazia.\n");
88                 }
89                 break;
90             case 6:
91                 printf("Saindo do programa.\n");
92                 return 0;
93             default:
94                 printf("Opção inválida.\n");
95                 break;
96         }
97     } while (op != 6);
98     return 0;
99 }

```

The terminal output shows the program running with the following sequence of events:

- Menu displayed with options 1-6.
- User input: 10 (value to be pushed).
- User input: 1 (push option).
- User input: 2 (pop option).
- User input: 3 (top element option).
- User input: 4 (display all elements option).
- User input: 5 (remove all elements option).
- User input: 6 (exit option).

The program successfully compiles and runs, demonstrating the stack operations.

```

1 - Empilhar um elemento
2 - Desempilhar o elemento do topo
3 - Consultar o elemento do topo
4 - Exibir todos os elementos da pilha
5 - Remover todos os elementos da pilha
6 - Sair do programa
5
Pilha esvaziada com sucesso!

Escolha uma opção:
1 - Empilhar um elemento
2 - Desempilhar o elemento do topo
3 - Consultar o elemento do topo
4 - Exibir todos os elementos da pilha
5 - Remover todos os elementos da pilha
6 - Sair do programa
4
Pilha: 90 40 20

Escolha uma opção:
1 - Empilhar um elemento
2 - Desempilhar o elemento do topo
3 - Consultar o elemento do topo
4 - Exibir todos os elementos da pilha
5 - Remover todos os elementos da pilha
6 - Sair do programa

```

Arquivo do trabalho com o código:



Trabalho\_Pilha.c



Trabalho\_Pilha.exe