



Presentación:

Practica 4.

Nombre: Farfán de León José Osvaldo

Código: 214796622

Carrera: Ingeniería en computación

Materia: Estructura de Datos I

Profesor: Julio Esteban Valdés López

Sección: "D13"

Fecha de entrega: 04/11/2020

Introducción:

Objetivo de la practica: Implementar pilas estáticas.

Datos de entrada y precondiciones: El usuario ingresara los datos de cada pila.

Resultados obtenidos (captura de pantalla):

Código fuente en C++:

```

1  #include <iostream>
2  #include <cstdlib>
3  #include <windows.h>
4  #include <stdio.h>
5  #define TAM 500
6  using namespace std;
7
8  typedef int type_dato;
9
10 typedef struct{
11     type_dato datos[TAM];
12     int tope;
13 }type_pila;
14
15 void inicializa(type_pila *T){
16     T->tope = -1;
17 }
18
19 int vacia(type_pila *T){
20     return T->tope == -1;
21 }
22
23 int llena(type_pila *T){
24     return T->tope == TAM-1;
25 }
26
27 void push(type_pila *T){
28     int d;
29     if(llena(T)){
30         cout<<"Desbordamiento de datos"<<endl;
31     }else{
32         cout<<"Ingrese el dato: ";
33         cin>>d;
34         T->tope++;
35         T->datos[T->tope]=d;
36     }
37 }
38
39 int pop(type_pila *T){
40     if(vacia(T)){
41         cout<<"Error... Insuficiencia de datos"<<endl;
42         return 0;
43     }else{
44         cout<<"Se eliminara el dato["<<T->datos[T->tope]<<"] del tope ["<<T->tope<<"]
correctamente"<<endl;
45         T->tope--;
46     }
47 }
48
49 int tope(type_pila *T){
50     if(vacia(T)){
51         cout << "!Error!... No hay elementos." << endl;
52         system("pause");
53         return -1;
54     }else{
55         cout<<"El tope es: "<<T->datos[T->tope]<<endl;
56         return 0;

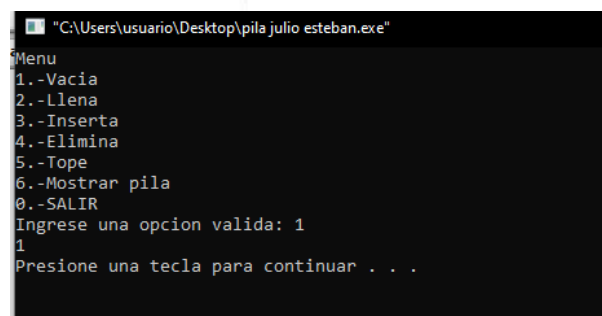
```

```

57     }
58 }
59
60 void mostrar_pila(type_pila *T){
61     for(int i=T->tope ; i>=0 ; i--){
62         printf("\t\t%c ",186,205);
63         cout<<T->datos[i]; printf(" %c\n",186);
64         printf("\t\t%c%c%c%c%c\n",204,205,205,205,185);
65     }
66 }
67
68 int menu(){
69     int opc;
70     cout<<"Menu"<<endl;
71     cout<<"1.-Vacía"<<endl;
72     cout<<"2.-Llena"<<endl;
73     cout<<"3.-Inserta"<<endl;
74     cout<<"4.-Elimina"<<endl;
75     cout<<"5.-Tope"<<endl;
76     cout<<"6.-Mostrar pila"<<endl;
77     cout<<"0.-SALIR"<<endl;
78     cout<<"Ingrese una opcion valida: ";
79     cin>>opc;
80     return opc;
81 }
82
83 int main(){
84     int opc, band=1;
85     type_pila _pila;
86     type_pila *_pila2;
87     _pila2= &_pila;
88     inicializa(_pila2);
89     do{
90         system("cls");
91         mostrar_pila(_pila2);
92         switch(menu()){
93             case 1: vacia(_pila2);cout<<vacía(_pila2)<<endl;system("pause");break;
94             case 2: llena(_pila2);cout<<llena(_pila2)<<endl;system("pause");break;
95             case 3: push(_pila2);cout<<"Dato agregado exitosamente"<<endl; break;
96             case 4: pop(_pila2);system("pause");break;
97             case 5: tope(_pila2);system("pause");break;
98             case 6: mostrar_pila(_pila2);system("pause");break;
99             case 0: system("cls");cout<<"FIN DEL PROGRAMA"<<endl;band=0;break;
100             default:
101                 cout<<"ERROR... Dato invalido"<<endl;
102                 Sleep(2000);
103         }
104     }while(band != 0);
105 }
106
107 }

```

Pantalla de ejecución:



```

C:\Users\usuario\Desktop\pila julio esteban.exe
Menu
1.-Vacía
2.-Llena
3.-Inserta
4.-Elimina
5.-Tope
6.-Mostrar pila
0.-SALIR
Ingrese una opcion valida: 1
Presione una tecla para continuar . . .

```

```
C:\Users\usuario\Desktop>pila julio esteban.exe
```

```
Menu
1.-Vacía
2.-Llena
3.-Inserta
4.-Elimina
5.-Tope
6.-Mostrar pila
0.-SALIR
Ingrese una opcion valida: 2
0
Presione una tecla para continuar . . .
```

```

C:\Users\usuario\Desktop\pila_jefe_nitchen.exe

  4
  6
  5

Menu
1.-Vacia
2.-Llena
3.-Inserta
4.-Elimina
5.-Tope
6.-Mostrar pila
0.-Salir
Ingrese una opcion valida:

```

```

C:\Users\usuario\Desktop>gda_jda_stelman.exe

      4
      |
      R
      |
      S
      |

Menu
1.-Vacía
2.-Lista
3.-Inserta
4.-Elimina
5.-Tipo
6.-Buscar Pila
a.-Salir

Ingrese una opción válida: 4
Se eliminara el dato[4] del tipo [1] correctamente
Presione una tecla para continuar . . .

```

```

C:\Users\usuario\Desktop>g++ julio_ej18han.cpp

```



```

Menu
1.-Vacia
2.-llena
3.-Inserta
4.-Elimina
5.-Tipo
6.-Mostrar pila
0.-SALIR

Ingrese una opcion valida: 5
El tipo es: 6
Presione una tecla para continuar . . .

```

```

C:\Users\usuario\Desktop>g++ lista_enlazada.cpp

Menu
1.-Vacia
2.-Llena
3.-Inserta
4.-Elimina
5.-Tipe
6.-Mostrar lista
7.-Salir
Ingrese una opcion valida: 6

6

Presione una tecla para continuar...

```

Conclusion:

Implementación de pilas estáticas en c++ utilizando los métodos de vacia, llena, inserta, elimina, tope, mostrar pila y salir, en mi caso siempre se encuentra mostrando la pila con un marco utilizado en código ascci.