



PILAS

Estructura de datos

Nombre: Daniela Beatriz Martínez Montes

Grupo: 2.B

Profesor: Oscar Flores

PILAS

Una pila es una colección de datos a los cuales se les puede acceder mediante un extremo, que se conoce generalmente como tope.

Las pilas son estructuras de datos que tienen dos operaciones básicas: push (para insertar un elemento) y pop (para extraer un elemento). Su característica fundamental es que al extraer se obtiene siempre el último elemento que acaba de insertarse.

Una posible implementación mediante listas enlazadas sería insertando y extrayendo siempre por el principio de la lista. La variable que llama al mismo procedimiento en el que está, habrá que guardarla así como el resto de variables de la nueva llamada, para a la vuelta del ciclo ir sacándolas. Existen muchos casos prácticos en los que se utiliza la idea de pila. Ejemplo: pila de platos, en el supermercado latas, además, las pilas se utilizan en muchas aplicaciones que utilizamos con frecuencia. Por ejemplo, la gestión de ventanas en Windows (cuando cerramos una ventana siempre recuperamos la que teníamos detrás). Otro ejemplo es la evaluación general de cualquier expresión matemática para evitar tener que calcular el número de variables temporales que hacen falta.

Por ejemplo:

$$3 + 4 * (8 - 2 * 5)$$

5				
-2	-10			
8	8	-2		
4	4	4	-8	
3	3	3	3	-5

Un dato importante es que al utilizar arreglos para implementar pilas se tiene la limitación de que se debe reservar el espacio en memoria con anticipación. Una vez dado un máximo de capacidad a la pila no es

posible insertar un número de elementos mayor que el máximo establecido. Si esto ocurre, en otras palabras si la pila esta llena y se intenta insertar un nuevo elemento, se producirá un error conocido como desbordamiento –overflow

Una posible solución a este tipo de inconvenientes consiste en definir pilas de gran tamaño, pero esto resultará ineficiente y costoso si solo se utilizarán algunos elementos. No siempre es viable saber con exactitud el número de elementos a tratar, y siempre existe la posibilidad de que ocurra un error de desbordamiento

Un ejemplo de pila en c++ es el siguiente:

```
#include <iostream.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>

struct pilas
{
int d;
pilas *a;
}*c,*e;

void menu(void);
void ingresar(void);
void sacar (void);
void actualizar_pila(void);

main()
{
menu();
}
void menu(void)
{
int y,opc;
for(;;)
{
cout<<"\n1. Ingresar datos";
cout<<"\t2. Sacar datos";
cout<<"\t0. Terminar";
cout<<"\n Ingrese opcion: ";cin>>opc;

switch(opc)
{
case 1:
ingresar();
break;
case 2: sacar();
break;
case 0: exit(1);
default: cout<<"\n Opcion no valida!!"; bre.
}
actualizar_pila();
cout<<"\n\nOprima una tecla para continuar";
getch();
}
}

void ingresar (void)
{
if(!c)
{
c=new(pilas);
cout<<"Ingrese elemento: ";
cin>>c->d;
c->a=NULL;
return;
}

e=new(pilas);
cout<<"\nIngrese elemento: ";
cin>>e->d;
e->a=c;
c=e;
}

void sacar(void)
{
if(!c)
{
cout<<"\n\nNo hay elementos!!";
return;
}
```

```
e=new(pilas);
cout<<"\nIngrese elemento: ";
cin>>e->d;
e->a=c;
c=e;
}

void sacar(void)
{
    if(!c)
    {
        cout<<"\n\nNo hay elementos!!";
        return;
    }

    e=new(pilas);
    e=c;
    cout<<"\n\nElemento eliminado: " <<e->d;
    c=e->a;
    delete(e);
}

void actualizar_pila(void)
{
    int y=2,i,ca=0;
    e=c;
    while(e)
    {
        ca++;
        e=e->a;
    }

    for(i=0;i<=ca;i++)
    {
        cout<<" ";
    }
    //muestro lo que tiene la pila!!
    i=0;
    e=c;
    while(e)
    {
        cout<<"\n";
        cout<<++i<<" - " <<e->d;
        e=e->a;
    }
}
```