



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

Estructuras de datos

Pilas y Colas

Prof.: Noe Ortega Sanchez



Alumno: Edgar Antonio Delgadillo Villegas

Carrera: Ingeniería Informática

Materia: i5886 Estructura de datos I

NRC: 182881

Sección: D17

Calendario: 2021A

"Pilas y Colas"

Descripción:

En este programa se realizó un breve código en el que muestra el funcionamiento de las pilas y las colas, a través de un programa que registra nombres de libros y precios en mi caso, adicional te permitía eliminar, añadir, mostrar, vaciar y sus respectivas funcionalidades en el caso de la pila (Push, Pop, Top) y en el caso de la cola(queue y dequeue) usando una lista simple en la pila y una lista doble en la cola.

A continuación mostrare unas breves imágenes de mi código y como lo realice:

Main pila y cola

En el main realice la elaboración del menú y como se imprimiría todo, use distintos casos de switch para poder darle un mejor manejo de la información, no mostrare todo el código ya que considero pertinente para una mejor explicación y no atiborrar el contenido de puro código, adicional se adjuntara al classroom el (.zip) del código de ser necesario verlo.

```
#include <iostream>
#include "Libros.h"
#include "Pila.h"
using namespace std;

int opc, opc2;
int main()
{
    Pila *pila=new Pila;
    Libros l;

    string marca;

    do
    {
        cout<<"-----"<<endl;
        cout<<"|                               Libros                               |"<<endl;
        cout<<"-----"<<endl;
        cout<<"1) Insertar libro"<<endl;
        cout<<"2) Eliminar libro"<<endl;
    }
```



```
cout<<"9) Mostrar"<<endl;
cout<<"0) Salir"<<endl<<endl;
cout<<"Opcion: ";
cin >> opc;
cin.ignore();

system("cls");

switch (opc)
{
    case 1:
        cout << "Insertar en Pila" <<endl<<endl;
        cin >> l;
        pila->Insertar(l);
        break;
    case 2:
        cout << "Eliminar en Pila" <<endl<<endl;
        if (pila->Vacia()==1)
            cout<<"La pila esta vacia"<<endl;
        else
        {
            pila->Eliminar();
        }
        break;
    case 3:
        cout << "Buscar en Pila" <<endl<<endl;
        if (pila->Vacia()==1)
            cout<<"La pila esta vacia"<<endl;
        else
        {
            cout<<"Ingresa el libro: ";
            getline(cin, nombre);
            cin.ignore();
            pila->Buscar(nombre);    Buscar no es necesario
        }
        break;
    case 4:
        cout << "Top Pila" <<endl<<endl;
        if (pila->Vacia()==1)
            cout<<"La pila esta vacia"<<endl;
        else
            pila->Top();
        break;
    case 5:
        cout << "Vacia" <<endl<<endl;
```



```
        break;
    case 5:
        cout << "Vacía" << endl<< endl;
        if (pila->Vacía() == 1)
            cout<<"La pila está vacía"<< endl;
        else
            cout<<"La pila no está vacía"<< endl;
        break;
    case 6:
        cout << "Inicializar Pila" << endl<< endl;
        if (pila->Vacía())
        {
            cout<<"Pila está vacía"<< endl;
        }
        else
        {
            pila->Inicializa();
        }
        break;
    case 7:
        cout << "Tamaño de Pila" << endl<< endl;
        cout << "\t El tamaño de la pila es: " << pila->Tamaño() << endl;
        break;
```

```
        cout << "Vaciar Pila" << endl<< endl;
        if (pila->Vacía() == 1)
            cout<<"La pila está vacía"<< endl;
        else
            pila->Vaciar();
        break;
    case 8:
        cout << "Mostrar Toda la Pila" << endl<< endl;
        if (pila->Vacía())
            cout<<"Pila vacía, no tienes elementos para mostrar"<< endl;
        else
        {
            pila->MostrarTodo();
        }
        break;
    case 0:
        break;
    default:
        cout << "Opción Inválida"<< endl;
        break;
    }

    cout << endl;
    system("Pause");
    system("cls");
}

while(opc != 0),

return 0;
}
```

Main cola

```
do
{

    cout<<"-----" <<endl;
    cout<<"|                               Libros                               |" <<endl;
    cout<<"-----" <<endl;

    cout <<"1) Insertar" <<endl;
    cout <<"2) Eliminar" <<endl;
    cout <<"3) Primero" <<endl;
    cout <<"4) Ultimo" <<endl;
    cout <<"5) Vacio" <<endl;
    cout <<"6) Inicializa" <<endl;
    cout <<"7) Tamano" <<endl;
    cout <<"8) Vaciar" <<endl;
    cout <<"9) Mostrar" <<endl;
    cout <<"0) Salir" <<endl;
    cout<<"Opcion: ";
    cin >> opc;
    cin.ignore(); // borrarbuffer

    system("cls"); // Limpiar pantalla-> Limpiar menu de pantalla

    switch (opc)
    {
        case 1:
            cout << "Insertar en Cola" <<endl<<endl;
            cin >> l;
            cola->Insertar(l);
            break;

        case 3:
            cout << "Primero de Cola" <<endl<<endl;
            if (cola->Vacio()==1)
                cout<<"La cola esta vacia"<<endl;
            else
                cola->Primero();
            break;

        case 4:
            cout << "Ultimo de Cola" <<endl<<endl;
            if (cola->Vacio()==1)
                cout<<"La cola esta vacia"<<endl;
            else
                cola->Ultimo();
            break;

        case 5:
            cout << "Vacio" <<endl<<endl;
            if (cola->Vacio()==1)
                cout<<"La cola esta vacia"<<endl;
            else
                cout<<"La cola no esta vacia"<<endl;
            break;

        case 6:
            cout << "Inicializar Cola" <<endl<<endl;
            if (cola->Vacio())
            {
                cout<<"Cola esta vacia"<<endl;
            }
            else
                ;
    }
}
```



```
        if (cola->Vacia()==1)
            cout<<"La Cola esta vacia"<<endl;
        else
            cola->Vaciar();
        break;
    case 9:
        cout << "Mostrar Toda la Cola" <<endl<<endl;
        if (cola->Vacia())
            cout<<"Cola vacia"<<endl;
        else
        {
            cola->MostrarTodo();
        }
        break;
    case 0:
        break;
    default:
        cout << "Opcion incorrecta"<<endl;
        break;
    }

    cout <<endl;
    system("Pause");
    system("cls");
}
while(opc != 0);

return 0;
}
```


Nodos Pila y cola

Nodo.h

```
#ifndef NODO_H
#define NODO_H

#include <iostream>
#include "Libros.h"

using namespace std;

class Nodo
{
public:
    Nodo();
    ~Nodo();
    Libros dato;
    Nodo *sig;
};

#endif // NODO_H
```

Aquí en el nodo se muestra como use la clase Libros para indicar o guardar el dato como si fuese un libro y por último se muestra sus respectivos nodo el siguiente para la Pila ya que es una lista simple y anterior y siguiente para la cola ya que es una lista doble por último.



```
#include "Nodo.h"

Nodo::Nodo()
{
    //ctor
    dato.setNombre("");
    dato.setPrecio(0);
    sig = nullptr;
}

Nodo::~~Nodo()
{
    //dtor
}
```

Nodo.cpp

```
#include "Nodo.h"

Nodo::Nodo()
{
    dato = "";
    sig = nullptr;
}

Nodo::~~Nodo()
{
}
```



```
#define NODO_H
#include <iostream>
#include "Libros.h"

using namespace std;

class Nodo
{
public:
    Nodo();
    ~Nodo();
    Libros dato;
    Nodo *sig;
    Nodo *ant;

    //protected:

    //private:
};

#endif // NODO_H
```

pila y cola(pila.h, pila.cpp, cola.h, cola.cpp)



```
class Pila
{
    public:
        Pila();
        ~Pila();
        void Inicializa(void);
        int Tamanio();
        void Vaciar();
        void MostrarTodo();
        void Buscar(string);
        void Top();
        void Insertar(Libros);
        void Eliminar();
        bool Vacia();

    private:
        Nodo *t;
};
```

```
#include "Nodo.h"
#include "Libros.h"
using namespace std;

class Cola
{
    public:
        Cola();
        ~Cola();
        bool Vacia();
        void Inicializa(void);
        int Tamanio();
        void Vaciar();
        void Eliminar();
        void Primero();
        void MostrarTodo();
        void Ultimo();
        void Insertar(Libros);
```

```
#include "Pila.h"

Pila::Pila()
{
    t=nullptr;
    //xxxx
}

Pila::~Pila()
{
    //xxxx
}

void Pila::Inicializa()
{
    t= NULL;
}

bool Pila::Vacia()
{
    if (t==nullptr)
        return true;
    else
        return false;
}

int Pila::Tamanio()
{
    int contador=0;
    Nodo *aux;
    aux = t;
    while(aux!=NULL)
    {
        contador++;
        aux = aux->sig;
    }
    return contador;
}

void Pila::Insertar(Libro a)
{
    Nodo *tmp=new Nodo();
    tmp->data=a;
    tmp->sig = this->t;
    this->t = tmp;
}

void Pila::Eliminar()
{
    Nodo *aux = t;

    t = t->sig;
    delete(aux);
    aux=t;
    cout <<"\n\tData eliminado"<<endl;
}
```



```
-  
    }  
    delete aux;  
-}  
  
void Cola::Insertar(Libro a)  
{  
    Nodo *tmp=new Nodo();  
    tmp->dato=a;  
    tmp->sig=nullptr;  
  
    if (Vacía()--1)  
    {  
        h=tmp;  
    }  
    else  
        t->sig=tmp;  
    t=tmp;  
-}  
  
bool Cola::Vacía()  
{  
    if (h==nullptr)  
        return true;  
    else  
        return false;  
-}  
  
void Cola::Inicializa(void)  
{  
    h=NULL;  
    t=NULL;  
-}
```

```
void Cola::Inicializa(void)  
{  
    h=NULL;  
    t=NULL;  
-}  
  
void Cola::Ultimo()  
{  
    cout << "Nombre: " << t->dato.getNombre() << endl;  
    cout << "Precio: " << t->dato.getPrecio() << endl;  
-}  
  
void Cola::Primero()  
{  
    cout << "Nombre: " << h->dato.getNombre() << endl;  
    cout << "Precio: " << h->dato.getPrecio() << endl;  
-}  
  
void Cola::Vaciar()  
{  
    Libro a;  
    a=h->dato;  
    Nodo *aux = h;  
  
    while (aux)  
    {  
        h=h->sig;  
        delete aux;  
        aux=h->sig;  
    }  
-}
```

```
        cout << "Precio: " << h->dato.getPrecio() << endl;
    }

void Cola::Vaciar()
{
    Libros s;
    s=h->dato;
    Nodo *aux = h;

    while (aux)
    {
        h=h->sig;
        delete aux;
        aux = h;
    }
    cout << "Pila eliminada" << endl;
}

int Cola::Tamano()
{
    int contador = 0;
    Nodo *aux=h;

    while(aux)
    {
        contador++;
        aux=aux->sig;
    }
    return contador;
}
```

En estos cuatro, en lista.h, en lista.cpp, en cola .h y en cola.cpp se muestra que métodos utilice tanto el push, como el pop, queue, dequeue, vaciar, mostrar, tope entre otros en el cpp se muestra como lo implemente e hice que funcionara, como comente anteriormente, no mostrare el código completo para no atiborrar el ensayo de puro código pero adicional anexare el (.zip) al classrom de ser necesario verlo.

Libro.h y libro.cpp



```
#include "Libros.h"
#include <iostream>
using namespace std;

Libros::Libros()
{
    //set
    nombre = "-";
    precio = 0;
}

Libros::~Libros()
{
    //set
}

void Libros::setNombre(const string& valor)
{
    nombre = valor;
}

string Libros::getNombre() const
{
    return nombre;
}

float Libros::getPrecio() const
{
    return precio;
}

void Libros::setPrecio(float valor)
{
}

class LIBROS
{
public:
    Libros();
    ~Libros();

    float getPrecio() const;
    void setPrecio(float valor);

    void setNombre(const string& valor);
    string getNombre() const;

private:
    string nombre;
    float precio;
};

#endif // SMARTPHONES_H
```

```
//xxx
nombre = "-";
precio = 0;
}

Libros::~Libros()
{
    //xxx
}

void Libros::setNombre(const string& valor)
{
    nombre = valor;
}

string Libros::getNombre() const
{
    return nombre;
}

float Libros::getPrecio() const
{
    return precio;
}

void Libros::setPrecio(float valor)
{
    precio = valor;
}

#ifdef NODU_H
#define NODO_H
#include <iostream>
#include "Libros.h"

using namespace std;

class Nodo
{
public:
    Nodo();
    ~Nodo();
    Libros dato;
    Nodo *sig;
    Nodo *ant;

    //protected:
    //private:
};

#endif // NODO_H
```

Aquí se muestra como use la clase libro para indicar o guardar el dato como si fuese un libro, con sus respectivos nodos dependiendo si es cola o pila.



Compilando el Programa

```
!                               Libros                               !
-----
1> Insertar libro<push>
2> Eliminar libro<pop>
3> Buscar
4> Top
5> Vacía
6> Inicializa
7> Tamaño
8> Vaciar
9> Mostrar
0> Salir

Opcion:
```

```
Insertar en Pila
Nombre del libro: el mejor libro
Precio: 2
Presione una tecla para continuar . . .
```

```
Dato eliminado
Presione una tecla para continuar . . .
```



```
Marca: el mejor libro  
Precio: $1
```

```
Presione una tecla para continuar . . .
```

```
Tamano de Pila
```

```
El tamano de la pila es: 1
```

```
Presione una tecla para continuar . . .
```

```
Vacia
```

```
La pila esta vacia
```

```
Presione una tecla para continuar . . .
```



```
!                               Libros                               !
-----
1) Insertar(queue)
2) Eliminar(dequeue)
3) Primero
4) Ultimo
5) Vacia
6) Inicializa
7) Tamano
8) Vaciar
9) Mostrar
0) Salir
Opcion:
```

```
Insertar en Cola
Nombre del libro:
el libro
Precio: 2
Presione una tecla para continuar . . .
```

Eliminar en Cola

Se elimino el primer dato ingresado:

Presione una tecla para continuar . . .

Tamano de Cola

El tamaño de la pila es: 0

Presione una tecla para continuar . . .

Opinión;

Este programa por lo menos para mí fue un poco más sencillo, aun siento que me falta pulir un poco más la teoría de listas dobles, pero siento que poco a poco se seguirá implementando y será más fácil, fue sencillo para mí la Cola y la Pila ya que es básicamente como Las listas ya sea doble y simple, obviamente con sus respectivas indicaciones e implementaciones propias de cada una.