

**Universidad Nacional de San Agustín**

**Ciencia de la Computación II**

**LABORATORIO 11**

**Nombre:** Alexandro Delgado Justo

**CUI:** 20173455

**Profesor:** Enzo Velázquez Lobatón

**Repositorio:** <https://github.com/adelgadoj/CCII-LABORATORIO11>

**1. Defina una Pila que permita insertar elementos utilizando clases.**

***pila.h***

```
#include <iostream>
#include "nodo.h"
using namespace std;
class pila
{
private:
    Nodo *head;
    int size;
public:
    pila();
    ~pila();
    void push(int);
    void print();
};
pila::pila()
{
    head = nullptr;
    size = 0;
}
pila::~~pila()
{
    Nodo *aux = head;
    while (head != nullptr)
    {
        aux = head->getNext();
        delete head;
        head = aux;
    }
}

void pila::push(int _dato)
{
    Nodo *temp = new Nodo(_dato);
    Nodo *aux = head;
    if (head == nullptr)
    {
        head = temp;
    }
}
```

```

        else
        {
            temp->setNext(aux);
            head = temp;
        }
        size++;
    }

void pila::print()
{
    Nodo *aux = head;
    while (aux)
    {
        cout << aux->getDato() << "->";
        aux = aux->getNext();
    }
    cout<<endl;
}

```

### ***nodo.h***

```

#include <iostream>
class Nodo
{
private:
    int dato;
    Nodo *next;

public:
    Nodo(int);
    int getDato();
    void setDato(int);
    Nodo *getNext();
    void setNext(Nodo *);
};

Nodo::Nodo(int _dato)
{
    dato = _dato;
    next = nullptr;
}

int Nodo::getDato()
{
    return dato;
}

void Nodo::setDato(int _dato)
{
    dato = _dato;
}

Nodo *Nodo::getNext()

```

```

{
    return next;
}
void Nodo::setNext(Nodo *_next)
{
    next = _next;
}

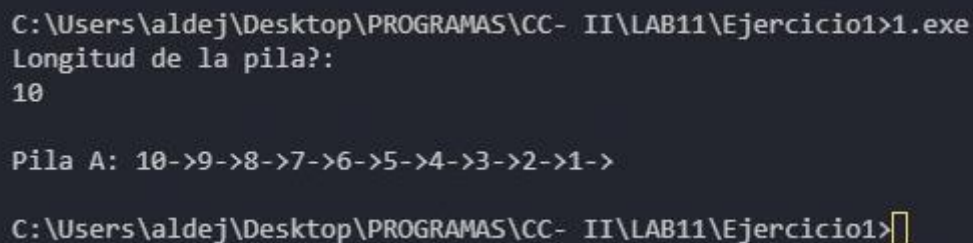
```

### **main.cpp**

```

#include <iostream>
#include "pila.h"
using namespace std;
int main(){
    pila A;
    int t;
    cout << "Longitud de la pila?: " << endl;
    cin >> t;
    for(int i = 0; i < t ; i++){
        A.push(i+1);
    }
    cout<<"\nPila A: ";
    A.print();
    A.~pila();
    return 0;
}

```



```

C:\Users\aldej\Desktop\PROGRAMAS\CC- II\LAB11\Ejercicio1>1.exe
Longitud de la pila?:
10

Pila A: 10->9->8->7->6->5->4->3->2->1->

C:\Users\aldej\Desktop\PROGRAMAS\CC- II\LAB11\Ejercicio1>

```

**2. Sobre el ejercicio anterior, adecue el programa para eliminar elementos de una Pila.**

### **pila.h**

```

#include <iostream>
#include "nodo.h"
using namespace std;
class pila
{
private:
    Nodo *head;
    int size;
public:
    pila();
    ~pila();
    void push(int);
    void pop(); //Funcion para eliminar

```

```

    void print();
};
pila::pila()
{
    head = nullptr;
    size = 0;
}
pila::~~pila()
{
    Nodo *aux = head;
    while (head != nullptr)
    {
        aux = head->getNext();
        delete head;
        head = aux;
    }
}

void pila::push(int _dato)
{
    Nodo *temp = new Nodo(_dato);
    Nodo *aux = head;
    if (head == nullptr)
    {
        head = temp;
    }
    else
    {
        temp->setNext(aux);
        head = temp;
    }
    size++;
}

void pila::pop()
{
    Nodo *aux = head;
    if (head == nullptr)
    {
        cout << "##Pila vacia##" << endl;
        system("pause");
        return;
    }
    else
    {
        Nodo *aux = head;
        head = aux->getNext();
        delete aux;
        size--;
    }
}

void pila::print()
{

```

```

    Nodo *aux = head;
    while (aux)
    {
        cout << aux->getDato() << "->";
        aux = aux->getNext();
    }
    cout<<endl;
}

```

### ***nodo.h***

```

#include <iostream>
class Nodo
{
private:
    int dato;
    Nodo *next;

public:
    Nodo(int);
    int getDato();
    void setDato(int);
    Nodo *getNext();
    void setNext(Nodo *);
};
Nodo::Nodo(int _dato)
{
    dato = _dato;
    next = nullptr;
}
int Nodo::getDato()
{
    return dato;
}

void Nodo::setDato(int _dato)
{
    dato = _dato;
}

Nodo *Nodo::getNext()
{
    return next;
}
void Nodo::setNext(Nodo *_next)
{
    next = _next;
}

```

### **main.cpp**

```
#include <iostream>
#include "pila.h"
using namespace std;
int main(){
    pila A;
    int t;
    cout << "Longitud de la pila?: " << endl;
    cin >> t;
    for(int i = 0; i < t; i++){
        A.push(i+1);
    }
    cout<<"\nPila A: ";
    A.print();
    cout << "\n\tEliminando Datos.."<<endl;
    for (int j = 0; j <= t; j++)
    {
        A.pop();
        A.print();
    }
    A.~pila();
    return 0;
}
```

```
C:\Users\aldej\Desktop\PROGRAMAS\CC- II\LAB11\Ejercicio2>2.exe
Longitud de la pila?:
10

Pila A: 10->9->8->7->6->5->4->3->2->1->

      Eliminando Datos..
9->8->7->6->5->4->3->2->1->
8->7->6->5->4->3->2->1->
7->6->5->4->3->2->1->
6->5->4->3->2->1->
5->4->3->2->1->
4->3->2->1->
3->2->1->
2->1->
1->

##Pila vacia##
Presione una tecla para continuar . . .
```

### 3. Implemente un algoritmo para buscar elementos de la Pila.

#### **pila.h**

```
#include <iostream>
#include "nodo.h"
using namespace std;
class pila
{
private:
    Nodo *head;
    int size;
public:
    pila();
    ~pila();
    void push(int);
    void pop();
    void busqueda(int);
    int getSize();
    void print();
    int top();
};

pila::pila()
{
    head = nullptr;
    size = 0;
}

pila::~~pila()
{
    Nodo *aux = head;
    while (head != nullptr)
    {
        aux = head->getNext();
        delete head;
        head = aux;
    }
}

void pila::push(int _dato)
{
    Nodo *temp = new Nodo(_dato);
    Nodo *aux = head;
    if (head == nullptr)
    {
        head = temp;
    }
    else
    {
        temp->setNext(aux);
        head = temp;
    }
    size++;
}
```

```

}
void pila::pop()
{
    Nodo *aux = head;
    if (head == nullptr)
    {
        cout << "##Pila vacia##" << endl;
        system("pause");
        return;
    }
    else
    {
        Nodo *aux = head;
        head = aux->getNext();
        delete aux;
        size--;
    }
}

void pila::busqueda(int _d) // Funcion de busqueda
{
    Nodo *aux = head;
    int p = 0;
    bool b = false;
    while (aux)
    {
        if(aux->getDato() == _d){
            cout<<"Se encontro Dato en la posicion: "<< p <<endl;
            b = true;
            return;
        }
        else{
            aux = aux->getNext();
            p++;
        }
    }

    if(b == false){
        cout << "No se encontro dato :(" << endl;
    }
}

void pila::print()
{
    Nodo *aux = head;
    while (aux)
    {
        cout << aux->getDato() << "->";
        aux = aux->getNext();
    }
    cout<<endl;
}

int pila::getSize()

```



```

{
    return size;
}
int pila::top()
{
    return head->getDato();
}

```

### ***nodo.h***

```

#include <iostream>
class Nodo
{
private:
    int dato;
    Nodo *next;

public:
    Nodo(int);
    int getDato();
    void setDato(int);
    Nodo *getNext();
    void setNext(Nodo *);
};
Nodo::Nodo(int _dato)
{
    dato = _dato;
    next = nullptr;
}
int Nodo::getDato()
{
    return dato;
}

void Nodo::setDato(int _dato)
{
    dato = _dato;
}

Nodo *Nodo::getNext()
{
    return next;
}
void Nodo::setNext(Nodo *_next)
{
    next = _next;
}

```

### **main.cpp**

```
#include <iostream>
#include "pila.h"
using namespace std;
int main(){
    pila A;
    int t, _dato;
    cout << "Longitud de la pila?: " << endl;
    cin >> t;
    for(int i = 0; i < t ; i++){
        A.push(i+1);
    }
    cout<<"\nPila A: ";
    A.print();
    cout<<"Que dato desea buscar?: "<<endl;
    cin >> _dato;
    A.búsqueda(_dato);
    A.~pila();
    return 0;
}
```

```
C:\Users\aldej\Desktop\PROGRAMAS\CC- II\LAB11\Ejercicio3>3.exe
Longitud de la pila?:
10

Pila A: 10->9->8->7->6->5->4->3->2->1->
Que dato desea buscar?:
2
Se encontro Dato en la posicion: 8

C:\Users\aldej\Desktop\PROGRAMAS\CC- II\LAB11\Ejercicio3>3.exe
Longitud de la pila?:
2

Pila A: 2->1->
Que dato desea buscar?:
4
No se encontro dato :(
```

#### 4. Escribir un programa que dé la solución al problema de las Torres de Hanoi para

**N discos, utilizando pilas, las cuales representen cada uno de los postes:**

***pila.h***

```
#include <iostream>
#include "nodo.h"
using namespace std;
class pila
{
private:
    Nodo *head;
    int size;
public:
    pila();
    ~pila();
    void push(int);
    void pop();
    void busqueda(int);
    int getSize();
    void print();
    int top();
};

pila::pila()
{
    head = nullptr;
    size = 0;
}

pila::~~pila()
{
    Nodo *aux = head;
    while (head != nullptr)
    {
        aux = head->getNext();
        delete head;
        head = aux;
    }
}

void pila::push(int _dato)
{
    Nodo *temp = new Nodo(_dato);
    Nodo *aux = head;
    if (head == nullptr)
    {
        head = temp;
    }
    else
    {
        temp->setNext(aux);
        head = temp;
    }
}
```

```

        size++;
    }
void pila::pop()
{
    Nodo *aux = head;
    if (head == nullptr)
    {
        cout << "##Pila vacia##" << endl;
        system("pause");
        return;
    }
    else
    {
        Nodo *aux = head;
        head = aux->getNext();
        delete aux;
        size--;
    }
}
void pila::busqueda(int _d)
{
    Nodo *aux = head;
    int p = 0;
    bool b = false;
    while (aux)
    {
        if (aux->getDato() == _d)
        {
            cout << "Se encontro Dato en la posicion: " << p << endl;
            b = true;
            return;
        }
        else
        {
            aux = aux->getNext();
            p++;
        }
    }
    if (b == false)
    {
        cout << "No se encontro dato :(" << endl;
    }
}
void pila::print()
{
    Nodo *aux = head;
    while (aux)
    {
        cout << aux->getDato() << "->";
        aux = aux->getNext();
    }
    cout<<endl;
}

```

```

}
int pila::getSize()
{
    return size;
}
int pila::top()
{
    return head->getDato();
}

```

### ***nodo.h***

```

#include <iostream>
class Nodo
{
private:
    int dato;
    Nodo *next;

public:
    Nodo(int);
    int getDato();
    void setDato(int);
    Nodo *getNext();
    void setNext(Nodo *);
};
Nodo::Nodo(int _dato)
{
    dato = _dato;
    next = nullptr;
}
int Nodo::getDato()
{
    return dato;
}

void Nodo::setDato(int _dato)
{
    dato = _dato;
}

Nodo *Nodo::getNext()
{
    return next;
}
void Nodo::setNext(Nodo *_next)
{
    next = _next;
}

```

### **main.cpp**

```
#include <iostream>
#include "pila.h"
using namespace std;
void show(pila &origen, pila &destino, pila &auxiliar)
{
    cout << "-----" << endl;
    cout << "\nA: " << endl;
    origen.print();
    cout << "B: " << endl;
    auxiliar.print();
    cout << "C: " << endl;
    destino.print();
}

void hanoi(int n, pila &origen, pila &destino, pila &auxiliar)
{
    if (n == 1)
    {
        int top = origen.top();
        destino.push(top);
        origen.pop();
        show(origen, destino, auxiliar);
        return;
    }
    else
    {
        //{mover n-1 discos de la torre origen a la torre auxiliar}
        hanoi(n - 1, origen, auxiliar, destino); //{llamada recursiva}
        // Mover un disco de la torre origen a la torre destino
        int top = origen.top();
        destino.push(top);
        origen.pop();
        show(origen, destino, auxiliar);
        //{mover n-1 discos de la torre auxiliar a la torre destino}
        hanoi(n - 1, auxiliar, destino, origen); //{llamada recursiva}
    }
    return;
}

int main()
{
    int n;
    pila origen;
    pila destino;
    pila auxiliar;
    cout<<"*****\tTORRE DE HANOI\t*****"<<endl;
    cout<<"Ingresa el numero de discos:\n";
    cin >> n;
    cout << "\nIniciando la Torre A con disco(s) del 1 al " <<n<< endl;
    for (int i = 0; i < n; i++)
```

```
{
    origen.push(n - i);
}
cout<<"\nLos elementos de la Torre A son: ";
origen.print();
show(origen,destino,auxiliar);
hanoi(n, origen, destino, auxiliar);
cout<< "\n\tDiscos movidos exitosamente";
cout<< "\nLos elementos de la Torre C son: ";
destino.print();
system("pause");
return 0;
}
```

```

Presione una tecla para continuar . . .
C:\Users\aldej\Desktop\PROGRAMAS\CC- II\LAB11\Ejercicio4>g++ main.cpp nodo.h pila.h -o hanoi.exe

C:\Users\aldej\Desktop\PROGRAMAS\CC- II\LAB11\Ejercicio4>hanoi.exe
***** TORRE DE HANOI *****
Ingresa el numero de discos:
3

Iniciando la Torre A con disco(s) del 1 al 3

Los elementos de la Torre A son: 1->2->3->
*****

A:
1->2->3->
B:

C:

*****

A:
2->3->
B:

C:
1->
*****

A:
3->
B:
1->
C:
2->
*****

A:

B:
3->
C:
1->2->
*****

A:

B:
1->2->
C:
3->
*****

A:
2->
B:
3->
C:
1->
*****

A:

B:
1->
C:
2->3->
*****

A:

B:

C:
1->2->3->

Discos movidos exitosamente
Los elementos de la Torre C son: 1->2->3->
Presione una tecla para continuar . . .

```