

**UNIVERSIDAD NACIONAL DE SAN  
AGUSTÍN DE AREQUIPA**

**FACULTAD DE INGENIERIA DE PRODUCCION Y  
SERVICIOS**

**ESCUELA PROFESIONAL DE CIENCIAS DE LA  
COMPUTACIÓN**



**LABORATORIO 05 – Punteros.**

**DOCENTE:**

Enzo Edir Velásquez Lobatón

**ALUMNO:**

Owen Haziél Roque Sosa.

**FECHA:**

13/05/2022

**Arequipa – Perú**

```

#include <iostream>
#include "rect.h"
using namespace std;
// 1. Implemente un programa con clases que calcule
// el área de un rectángulo y perímetro.

int main()
{
    int b, h;
    cout<<"Base y altura del rectángulo: "<<endl;
    cin>>b>>h;
    rect Rectangle(b,h);
    Rectangle.area();
    Rectangle.perimetro();

    return 0;
}

```

1. main.cpp

```

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

rect::rect(int _b, int _h){
    b = _b;
    h = _h;
}

rect::~rect(){}

void rect::area(){
    cout << "Area: " << b*h << endl;
}

void rect::perimetro(){
    cout << "Perimetro: " << 2*(b+h) << endl;
}

```

1.1. rect.cpp

```
#ifndef RECT_H
#define RECT_H

class rect{
private:
    int b;
    int h;
public:
    rect(int, int);
    ~rect();
    void area();
    void perimetro();
};
#endif
```

1.2. *rect.h*

```
Base y altura del rectangulo:
8
6
Area: 48
Perimetro: 28
```

1.3. *Resultado*

```

#include <iostream>
#include "alumno.h"
using namespace std;
//Se conoce de un alumno de la Universidad Nacional de San Agustín: CUI, nombre
//completo y tres notas parciales (nota1, nota2, nota3). El programa con clases
//debe imprimir: CUI, el primer nombre, el promedio de las tres notas e indique
//con un mensaje si el alumno aprobó (nota final >= 10.5) o no aprobó
//(nota final < 10.5) la asignatura de Ciencias de la Computación II.
int main()
{
    float n[3], cui;
    string nombres, apellidos;
    cout<<"CUI del alumno: ";
    cin>>cui;
    cin.ignore();
    cout<<"Nombres: ";
    getline(cin,nombres);
    cout<<"Apellidos: ";
    getline(cin,apellidos);

    float* nPtr;
    nPtr = n;
    cout<<"Ingrese las notas del alumno: \n";
    for (int i=0;i<3;i++){
        int x;
        cout<<"\tNota " <<i+1<<": ";
        cin>>x;
        n[i]=x;
    }

    alumno Alumno(cui,nombres,apellidos,nPtr);
    Alumno.output();
    return 0;
}

```

## 2. main.cpp

```

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

using namespace std;

alumno::alumno(int _cui, string _n, string _ap, float* _notas){
    CUI = _cui;
    nombres = _n;
    apellidos = _ap;
    notas = _notas;
}

alumno::~alumno(){}

void alumno::promedio_notas(){
    float res=0;
    for(int i=0;i<3;i++){
        res+=notas[i];
    }
    res=(float)res/3;
    cout<<res<<endl;
    if (res>=10.5) {
        cout<<"El alumno aprobo la asignatura de Ciencias de la Computacion II."<<endl;
    }
    else {
        cout<<"El alumno reprobo la asignatura de Ciencias de la Computacion II."<<endl;
    }
}

void alumno::output(){
    //primer nombre
    int pos = nombres.find(" ");
    string pnombre = nombres.substr(0,pos);
    cout<<"Nota promedio del alumno " <<pnombre<<": ";
    //promedio
    promedio_notas();
}

```

### 2.1. alumno.cpp

```

#ifndef ALUMNO_H
#define ALUMNO_H
#include <iostream>
using namespace std;

class alumno{
private:
    int CUI;
    string nombres;
    string apellidos;
    float* notas;
public:
    alumno(int, string, string, float*);
    ~alumno();
    void promedio_notas();
    void output();
};
#endif

```

## 2.2. alumno.h

```

CUI del alumno: 20211819
Nombres: Owen Haziell
Apellidos: Roque Sosa
Ingrese las notas del alumno:
    Nota 1: 20
    Nota 2: 18
    Nota 3: 17
Nota promedio del alumno Owen: 18.3333
El alumno aprobo la asignatura de Ciencias de la Computacion II.

```

## 2.3. Resultado

```

#include<iostream>
#include "persona.h"
using namespace std;
//3. Implemente un programa con clases que lea
//la fecha de nacimiento y la fecha de hoy;
//y muestre por pantalla el nombre y
//la edad de la persona.
int main (int argc, char *argv[]) {
    string name;
    int year, month;
    cout<<"Nombre de la persona: ";
    cin>>name;
    cout<<"Fecha Nacimiento:\n";
    cout<<"\tAño: ";
    cin>>year;
    cout<<"\tMes: ";
    cin>>month;
    persona p(name,0);
    p.calculoEdad(year,month);
    p.mostrar();
    return 0;
}

```

```

#include<iostream>
#include <iomanip>
#include <ctime>
#include <sstream>
#include "persona.h"
using namespace std;

persona::persona(string _nombre, int _edad) {
    nombre = _nombre;
    _edad = edad;
}

persona::~~persona() {}

void persona::calculoEdad(int yy, int mm) {
    //obtener año y mes actual
    auto t = time(nullptr);
    auto tm = *localtime(&t);
    ostringstream oss;
    oss << put_time(&tm, "%m-%Y");
    string time = oss.str();
    int pos = time.find("-");
    string m = time.substr(0, pos);
    int actmonth = stoi(m);
    string y = time.substr(pos+1);
    int actyear = stoi(y);
    //calculoEdad
    if (actmonth < mm)
        actyear -= 1;
    edad = actyear - yy;
}

void persona::mostrar() {
    cout << "Nombre: " << nombre << endl;
    cout << "Edad: " << edad << endl;
}

```

```
#ifndef PERSONA_H
#define PERSONA_H
using namespace std;
class persona{
private:
    string nombre;
    int edad;

public:
    persona(string,int);
    ~persona();
    void calculoEdad(int,int);
    void mostrar();
};
#endif
```

3.2. *persona.h*

```
Nombre de la persona: Owen
Fecha Nacimiento:
    Anio: 2004
    Mes: 6
Nombre: Owen
Edad: 17
```

3.3. *Resultado*



```

#include<iostream>
#include "arreglo.h"
using namespace std;
// 4. Implemente una aplicación con clases donde una clase contiene un arreglo
// en el que se pueden almacenar como máximo 5 datos. Debe tener un nuevo dato
// para almacenarlo, debe existir un método que verifique que el arreglo tenga
// espacio disponible para guardar el dato. En caso contrario se debe mostrar
// un mensaje indicando que el arreglo está lleno.
// Asimismo, debe tener un método para retirar un dato del arreglo, este método
// debe verificar que, si haya algo para sacar del arreglo, es decir que el
// arreglo no vaya a estar vacío, en cuyo caso debe desplegar un mensaje
// diciendo que no hay nada para sacar de arreglo.
int main (int argc, char *argv[]) {
    arreglo Arr;
    char opt;
    int pos = 0;
    do {
        cout<<"***** Arreglo *****\n";
        cout<<"a| AgregarDato\n";
        cout<<"b| RetirarDato\n";
        cout<<"c| Mostrar Array\n";
        cout<<"d| Salir\n";
        cout<<"Ingrese su opcion: ";
        cin>>opt;
        switch (opt){
            case 'a':
                int v;
                cout<<"Valor a ingresar: ";
                cin>>v;
                Arr.agregarDato(v,pos);
                pos++;
                break;
            case 'b':
                int b;
                cout<<"Valor a retirar: ";
                cin>>b;
                Arr.removerDato(b);
                pos--;
                break;
            case 'c':
                Arr.mostrar();
                break;
            case 'd':
                break;
        }
    }
    while (opt!='d');
    cout<<"Array final: \n";
    Arr.mostrar();
    return 0;
}

```

```

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

arreglo::arreglo(){
    for(int i=0;i<5;i++){
        arr[i]=0;
    }
    //arr = {0, 0, 0, 0, 0};
}

arreglo::~arreglo(){}

void arreglo::agregarDato(int value,int &pos){
    if (arr[4]!=0) //todos los valores pretedeterminados a 0
        cout<<"Array lleno! elimine elementos para agregar nuevos datos."<<endl;
    else {
        arr[pos]=value;
    }
    cout<<endl;
}

void arreglo::removerDato(int value){
    bool vacio=true;
    for(int i=0;i<5;i++){
        if (arr[i]!=0){
            vacio=false;
            break;
        }
    }
    if (vacio)
        cout<<"Array Vacio! agregue elementos al arreglo."<<endl;
    else {
        for(int i=0;i<5;i++){
            if(arr[i]==value){
                arr[i]=0;
                int j=i;
                while (j!=4){
                    int aux=arr[j+1];
                    arr[j+1]=arr[j];
                    arr[j]=aux;
                    j++;
                }
                break;
            } //para eliminar solo al 1er dato coincidente
        }
        cout<<endl;
    }
}

void arreglo::mostrar(){
    for(int i=0;i<5;i++)
        cout<<arr[i]<<" ";
    cout<<endl;
}

```

```

#ifndef ARREGLO_H
#define ARREGLO_H
class arreglo{
private:
    int arr[5];
public:
    arreglo();
    ~arreglo();
    void agregarDato(int,int&);
    void removerDato(int);
    void mostrar();
};
#endif

```

#### 4.2. arreglo.h

<pre> ***** Arreglo ***** a  AgregarDato b  RetirarDato c  Mostrar Array d  Salir Ingrese su opcion: a Valor a ingresar: 8 ***** Arreglo ***** a  AgregarDato b  RetirarDato c  Mostrar Array d  Salir Ingrese su opcion: a Valor a ingresar: 9 ***** Arreglo ***** a  AgregarDato b  RetirarDato c  Mostrar Array d  Salir Ingrese su opcion: a Valor a ingresar: 1 </pre>	<pre> ***** Arreglo ***** a  AgregarDato b  RetirarDato c  Mostrar Array d  Salir Ingrese su opcion: c 8 9 1 0 0 ***** Arreglo ***** a  AgregarDato b  RetirarDato c  Mostrar Array d  Salir Ingrese su opcion: b Valor a retirar: 9 ***** Arreglo ***** a  AgregarDato b  RetirarDato c  Mostrar Array d  Salir Ingrese su opcion: c 8 1 0 0 0 </pre>	<pre> ***** Arreglo ***** a  AgregarDato b  RetirarDato c  Mostrar Array d  Salir Ingrese su opcion: a Valor a ingresar: 3 ***** Arreglo ***** a  AgregarDato b  RetirarDato c  Mostrar Array d  Salir Ingrese su opcion: d Array final: 8 1 3 0 0 </pre>
---	--	---

#### 4.3. Resultado

```

#include<iostream>
#include "matriz.h"
using namespace std;
// 5. Implementar un programa con clases que haga la búsqueda de un dato
// almacenado en una matriz de 3x3 que tiene los números de 1 al 9 ingresados
// aleatoriamente y debe indicar la posición donde se encuentra el dato.
int main (int argc, char *argv[]) {
    matriz ma;
    ma.mostrar();
    int v;
    cout<<"Ingrese el dato a buscar: ";
    cin>>v;
    ma.buscar(v);
    return 0;
}

```

## 5. main.cpp

```

#include "matriz.h"
#include<iostream>
#include <stdlib.h>
#include <time.h>
using namespace std;
matriz::matriz() {
    srand(time(NULL));
    for(int i=0;i<3;i++)
        for(int j=0;j<3;j++)
            matrix[i][j]=1 + rand() % (10 - 1) ;
}

matriz::~~matriz(){}

void matriz::mostrar(){
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            cout<<matrix[i][j]<<" ";
        }
        cout<<endl;
    }
    cout<<endl;
}

void matriz::buscar(int value){
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            if(matrix[i][j]==value){
                cout<<"Valor Encontrado en: ";
                cout<<"["<<i<<"] ["<<j<<"]\n";
            }
        }
    }
    cout<<endl;
}

```

## 5.1. matriz.cpp

```

#ifndef MATRIZ_H
#define MATRIZ_H

class matriz {
public:
    matriz();
    ~matriz();
    void mostrar();
    void buscar(int);
private:
    int matrix[3][3];
};

#endif

```

5.2. *matriz.h*

```

4 6 7
7 2 4
5 4 5

Ingresa el dato a buscar: 7
Valor Encontrado en: [0] [2]
Valor Encontrado en: [1] [0]

```

5.3. *Resultado*