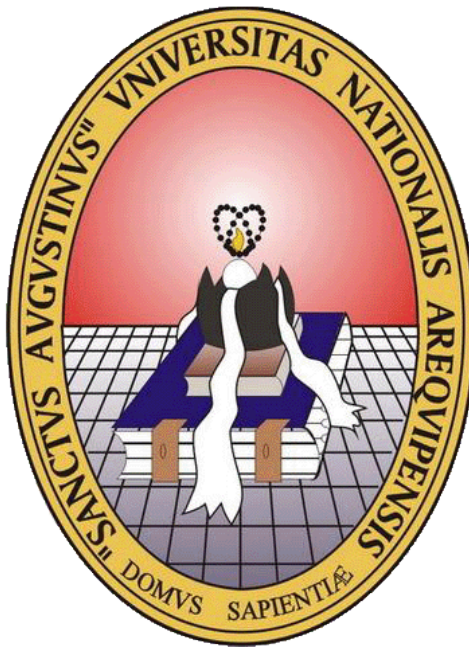


UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA  
FACULTAD DE PRODUCCIÓN Y SERVICIOS  
ESCUELA DE CIENCIAS DE LA COMPUTACIÓN



PRÁCTICA DE LABORATORIO 6  
CURSO DE CIENCIAS DE LA COMPUTACIÓN II

ESTUDIANTE:  
RUIZ MAMANI EDUARDO GERMÁN

EMAIL: [eruizm@unsa.edu.pe](mailto:eruizm@unsa.edu.pe)

CUI: 20193061

TURNO:

C

AREQUIPA- PERÚ

2021

LINK DEL REPOSITORIO: [https://github.com/EGRM23/CCII\\_20193061.git](https://github.com/EGRM23/CCII_20193061.git)

## 1. EJERCICIO

1. Implemente un programa con clases que calcule el área de un rectángulo y perímetro.

- **CÓDIGO**

```
#include <iostream>
using namespace std;

//EDUARDO GERMAN RUIZ MAMANI
//20193061

//1. Implemente un programa con clases que calcule el área de un
rectángulo y
//  perímetro.

class Rectangulo {
private:
    int an;
    int alt;
public:
    Rectangulo (int, int);
    ~Rectangulo();
    int area();
    int perimetro();
    void imprimirdat();
};

Rectangulo :: Rectangulo(int ancho, int altura) {
    an = ancho;
    alt = altura;
}

Rectangulo :: ~Rectangulo() {}

int Rectangulo :: area() {
    return an*alt;
}

int Rectangulo :: perimetro() {
    return 2 * (an + alt);
}

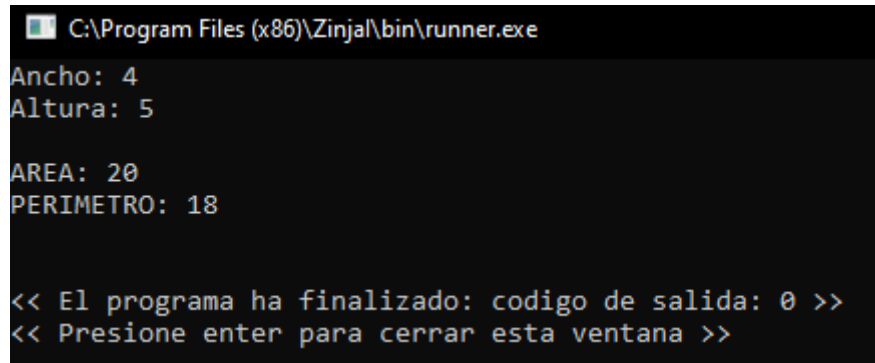
void Rectangulo :: imprimirdat() {
    cout << "Ancho: " << an << endl;
    cout << "Altura: " << alt << endl;
}

int main(int argc, char *argv[]) {
    Rectangulo r = Rectangulo(4, 5);
    r.imprimirdat();

    cout << endl;
    cout << "AREA: " << r.area() << endl;
    cout << "PERIMETRO: " << r.perimetro() << endl;
    return 0;
}
```

```
}
```

- **CAPTURAS**



```
C:\Program Files (x86)\Zinjal\bin\runner.exe
Ancho: 4
Altura: 5

AREA: 20
PERIMETRO: 18

==>

<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>
```

## 2. EJERCICIO

2. 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  $\geq 10.5$ ) o no aprobó (nota final  $< 10.5$ ) la asignatura de Ciencias de la Computación II.

- **CÓDIGO**

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

//EDUARDO GERMAN RUIZ MAMANI
//20193061

//2. 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  $\geq 10.5$ ) o no
aprobó (nota final  $< 10.5$ ) la
//    asignatura de Ciencias de la Computación II.

class Alumno {
private:
    string CUI;
    string nombres;
    string apellidos;
    string curso;
    int *notas;
    int cantnot;
public:
    Alumno(string, string, string, string, int*, int);
    ~Alumno();
    float notafin();
    bool aprobacion();
    void imprimirdat();
};
```

```

};

Alumno :: Alumno(string cod, string n, string a, string c, int
*notes, int tam) {
    CUI = cod;
    nombres = n;
    apellidos = a;
    curso = c;
    notas = notes;
    cantnot = tam;
}

Alumno :: ~Alumno() {}

float Alumno :: notafin() {
    int promedio = 0;
    for (int i = 0; i < cantnot; i++)
        promedio += notas[i];

    return (float) promedio / 3;
}

bool Alumno :: aprobacion() {
    if (notafin() >= 10.5)
        return true;
    else
        return false;
}

void Alumno :: imprimirdat () {
    cout << "DATOS DEL ESTUDIANTE\n" << endl;
    cout << "CUI: " << CUI << endl;
    cout << "NOMBRE: ";

    for (char letra : nombres) {
        if (letra == ' ') {
            cout << endl;
            break;
        } else
            cout << letra;
    }

    cout << "CURSO: " << curso << endl;
    cout << "NOTA FINAL: " << notafin() << endl;
    cout << "ESTADO: ";
    if (aprobacion() == true)
        cout << "APROBADO";
    else
        cout << "DESAPROBADO";
}

int main(int argc, char *argv[]) {
    int notas[] = {10,15,3,4,5};
    int tam = sizeof(notas) / sizeof(notas[0]);

```

```

        Alumno est_1 = Alumno("20193061", "Eduardo German", "Ruiz
Mamani", "Ciencias de la Computación II", notas, tam);

        est_1.imprimirdat();
        return 0;
    }

```

- **CAPTURAS**

```

C:\Program Files (x86)\Zinjal\bin\runner.exe
DATOS DEL ESTUDIANTE

CUI: 20193061
NOMBRE: Eduardo
CURSO: Ciencias de la Computación II
NOTA FINAL: 12.3333
ESTADO: APROBADO

<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>

```

### 3. EJERCICIO

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.

- **CÓDIGO**

```

#include <iostream>
#include <time.h>
using namespace std;

//EDUARDO GERMAN RUIZ MAMANI
//20193061

//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.

class Persona {
private:
    string nombre;
    int d;
    int m;
    int a;
public:
    Persona();
    ~Persona();
    bool EsBisiesto();
    bool Validacion();
    int DiasDelMes();
    void Edad();
    void imprimirdat();
};

Persona :: Persona () {

```

```

cout << "Nombre: ";
cin >> nombre;

do {
    cout << "FECHA DE NACIMIENTO\n";
    cout << "Año? ";
    cin >> a;
    cout << "Mes? ";
    cin >> m;
    cout << "Dia? ";
    cin >> d;

    if (Validacion() == false){
        cout<<"FECHA INVALIDA! Ingrese los datos de
nuevo\n";
        cout<<endl;
    } else
        break;

} while (true);
}

Persona :: ~Persona() {}

bool Persona :: EsBisiesto() {
    if(a%4 == 0)
        if(a%100 == 0 && a%400 != 0)
            return false; //No es bisiesto
        else {
            return true; //Si es bisiesto
        }
    else
        return false; //No es bisiesto
}

bool Persona :: Validacion() {
    if (m > 0 && m <= 12){
        if (d > 0 && d <= 31){
            if (m == 4 || m == 6 || m == 9 || m == 11){
                if (d != 31)
                    return true;
                else
                    return false;
            } else if(m == 2){
                if (d == 30 || d == 31)
                    return false;
                else if (EsBisiesto() == true || d ==
29)
                    return false;
                else
                    return false;
            } else
                return true;
        } else
            return false;
    }
}

```

```

        } else
            return false;
    }

    int Persona :: DiasDelMes() {
        if (m == 4 || m == 6 || m == 9 || m == 11)
            return 30;
        else if (m == 2){
            if(EsBisiesto() == 1)
                return 29;
            else
                return 28;
        } else
            return 31;
    }

    void Persona :: Edad() {
        int dE = 0, mE = 0, aE = 0, dA, mA, aA;

        time_t t = time(NULL);
        struct tm tiempoLocal = *localtime(&t);
        dA = tiempoLocal.tm_mday;
        mA = (tiempoLocal.tm_mon) + 1;
        aA = (tiempoLocal.tm_year) + 1900;

        if (aA == a) {
            if (mA == m)
                dE = dA - d;
            else {
                if (mA - 1 != m)
                    mE = mA - m - 1;

                if (dA >= d) {
                    mE++;
                    dE = dA - d;
                } else
                    dE = DiasDelMes() - d + dA;
            }
        } else {
            if (aA - 1 != a)
                aE = aA - a - 1;

            if (mA > m) {
                aE++;
                mE = mA - m;

                if (dA >= d)
                    dE = dA - d;
                else {
                    dE = DiasDelMes() - d + dA;
                    mE--;
                }
            } else if (mA == m) {
                if (dA >= d) {
                    aE++;

```

```

        dE = dA - d;
    }
    else {
        mE = 12 - m + mA - 1;
        if (dA >= d) {
            mE++;
            dE = dA - d;
        } else
            dE = DiasDelMes() - d + dA;
    }
}

if (mE == 12)
    aE++;
}

cout << "Fecha Actual: " << dA << "-" << mA << "-" << aA
<< endl;
cout << "Edad: " << aE << " años, " << mE << " meses y "
<< dE << " días\n";
}

void Persona :: imprimirdat() {
    cout << "DATOS DE LA PERSONA\n";
    cout << "Nombre: " << nombre << endl;
    cout << "Fecha de nacimiento: " << d << "-" << m << "-" <<
a << endl;
    Edad();
}

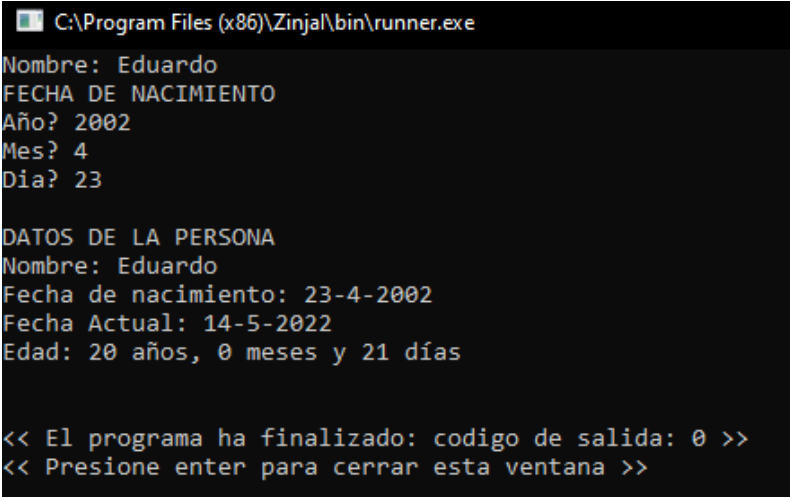
int main(int argc, char *argv[]) {
    Persona pers_1;

    cout << endl;
    pers_1.imprimirdat();

    return 0;
}

```

- **CAPTURAS**



```

C:\Program Files (x86)\Zinjal\bin\runner.exe
Nombre: Eduardo
FECHA DE NACIMIENTO
Año? 2002
Mes? 4
Día? 23

DATOS DE LA PERSONA
Nombre: Eduardo
Fecha de nacimiento: 23-4-2002
Fecha Actual: 14-5-2022
Edad: 20 años, 0 meses y 21 días

<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>

```



#### 4. EJERCICIO

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.

- **CÓDIGO (no pude culminar, se me complicó el void)**

```
#include <iostream>
using namespace std;

//EDUARDO GERMAN RUIZ MAMANI
//20193061

//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.

class datos {
private:
    int arr[5] = {};
    int tam;
public:
    datos();
    ~datos();
    void insertar();
    void eliminar();
    void imprimirdat();
    void menu();
};

datos :: datos() {
    tam = 0;
}

datos :: ~datos() {}

void datos :: insertar() {
    if (tam == 5)
```

```

        cout << "\nERROR! El arreglo está lleno, no se
pueden ingresar más datos\n";
    else {
        cout << "Numero a insertar: ";
        cin >> arr[tam];
        tam++;
    }
}

void datos :: eliminar() {
    if (tam == 0)
        cout << "\nERROR! El arreglo está vacío, no hay
dato a eliminar\n";
    else {
        int p;
        do {
            cout << "Posicion del numero que quiere
eliminar: ";
            cin >> p;
            if (p > tam || p < 1)
                cout << "ERROR! La posicion indicada
está fuera del rango del arreglo\n";
        } while(p > tam || p < 1);

        for (int i = p-1; i < tam; i++) {
            if (i == tam-1)
                arr[i] = 0;
            else
                arr[i] = arr[i+1];
        }

        tam--;
    }
}

void datos :: imprimirdat() {
    cout << "Datos = [ ";
    for (int i = 0; i < tam; i++)
        cout << arr[i] << " ";
    cout << "]\n";
}

void datos :: menu() {
    int op = 0;
    imprimirdat();
    cout << endl;

    do {
        do {
            cout << "Operaciones\n";
            cout << "1. Insertar dato\n";
            cout << "2. Eliminar dato\n";
            cout << "3. Salir\n";
            cout << "Numero de operacion: ";
            cin >> op;

```

```

        if (op != 1 && op != 2 && op != 3)
            cout << "ERROR! Ingrese un numero de
operacion valido\n" << endl;
    } while(op != 1 && op != 2 && op != 3);

    if (op == 3) {
        cout << "\n Gracias por confiar en
nosotros!";
        break;
    }

    switch (op) {
        case 1:
            insertar();
            cout << endl;
            break;
        case 2:
            eliminar();
            cout << endl;
            break;
    }

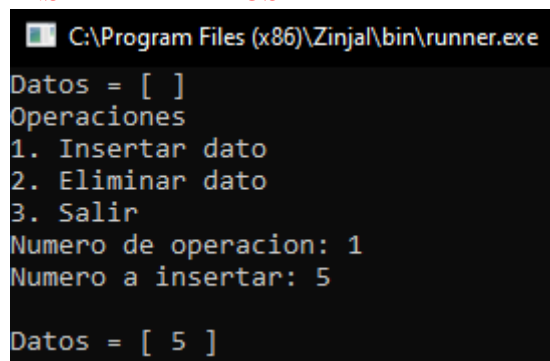
    imprimirdat();
} while (true);
}

int main(int argc, char *argv[]) {
    datos d;
    d.menu();

    return 0;
}

```

- CAPTURAS
  - INSERTAR DATOS



```

C:\Program Files (x86)\Zinjal\bin\runner.exe
Datos = [ ]
Operaciones
1. Insertar dato
2. Eliminar dato
3. Salir
Numero de operacion: 1
Numero a insertar: 5
Datos = [ 5 ]

```

- INSERTAR DATOS CUANDO ESTÁ LLENO

```
C:\Program Files (x86)\Zinjal\bin\runner.exe
Datos = [ 5 7 9 2 5 ]
Operaciones
1. Insertar dato
2. Eliminar dato
3. Salir
Numero de operacion: 1
ERROR! El arreglo está lleno, no se pueden ingresar más datos
```

- ELIMINAR DATOS

```
C:\Program Files (x86)\Zinjal\bin\runner.exe
Datos = [ 5 7 9 2 5 ]
Operaciones
1. Insertar dato
2. Eliminar dato
3. Salir
Numero de operacion: 2
Posicion del numero que quiere eliminar: 5
Datos = [ 5 7 9 2 ]
```

- ELIMINAR DATOS CUANDO NO HAY ELEMENTOS

```
C:\Program Files (x86)\Zinjal\bin\runner.exe
Datos = [ ]
Operaciones
1. Insertar dato
2. Eliminar dato
3. Salir
Numero de operacion: 2
ERROR! El arreglo está vacío, no hay dato a eliminar
Datos = [ ]
```

- FINALIZACION

```
C:\Program Files (x86)\Zinjal\bin\runner.exe
Datos = [ ]
Operaciones
1. Insertar dato
2. Eliminar dato
3. Salir
Numero de operacion: 3

Gracias por confiar en nosotros!

<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>
```

## 5. EJERCICIO

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.

- **CÓDIGO**

```
#include <iostream>
#include <stdlib.h>
#include <time.h>
using namespace std;

//EDUARDO GERMAN RUIZ MAMANI
//20193061

//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.

class matriz {
private:
    int **m;
    int f;
    int c;
public:
    matriz(int,int);
    ~matriz();
    void llenaraleatorio();
    void buscar();
    void imprimir();
};

matriz :: matriz(int filas, int columnas) {
    f = filas;
    c = columnas;

    m = new int *[f];
    for (int i = 0; i < f; i++)
        m[i] = new int[c] {};

    llenaraleatorio();
}

matriz :: ~matriz() {
    for (int i = 0; i < f; i++)
        delete m[i];

    delete m;
}

void matriz :: llenaraleatorio() {
    int tam = f * c;
    int *dat = new int[tam];
```

```

        srand(time(NULL));

        for (int i = 0; i < tam; i++)
            dat[i] = i+1;

        int p;
        for (int i = 0; i < f; i++) {
            for (int j = 0; j < c; j++) {
                p = rand()%(tam);
                m[i][j] = dat[p];

                for (int k = p; k < tam-1; k++)
                    dat[k] = dat[k+1];

                dat[tam-1] = 0;
                tam--;
            }
        }

        delete dat;
    }

void matriz :: buscar() {
    int d = 0;

    do {
        cout << "Dato a buscar: ";
        cin >> d;

        if (d < 1 || d > 9)
            cout << "ERROR! Dato no encontrado\n";
    } while(d < 1 || d > 9);

    for (int i = 0; i < f; i++) {
        for (int j = 0; j < c; j++) {
            if (d == m[i][j]) {
                cout << "Posicion: Fila " << i+1 << "
- Columna " << j+1;

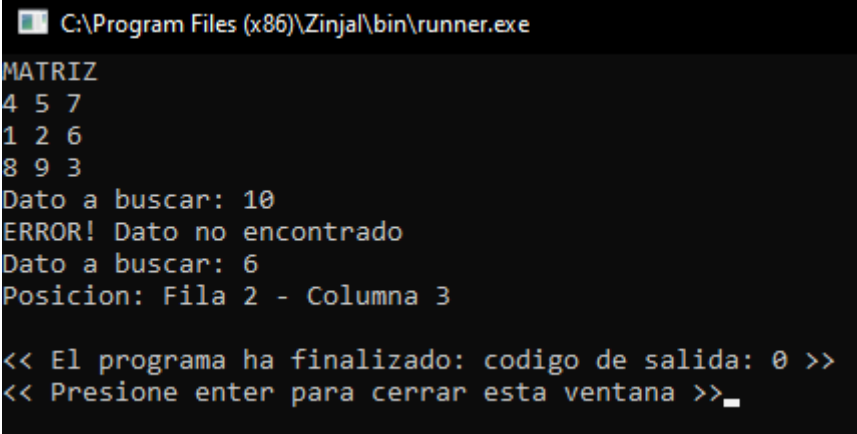
                d = 0;
                break;
            }
        }
        if (d == 0)
            break;
    }
}

void matriz :: imprimir() {
    cout << "MATRIZ\n";
    for (int i = 0; i < f; i++) {
        for (int j = 0; j < c; j++) {
            cout << m[i][j] << " ";
        }
        cout << endl;
    }
}

```

```
}  
  
int main(int argc, char *argv[]) {  
    matriz m_1(3,3);  
    m_1.imprimir();  
    m_1.buscar();  
  
    m_1.~matriz();  
    return 0;  
}
```

- **CAPTURAS**



```
C:\Program Files (x86)\Zinjal\bin\runner.exe  
MATRIZ  
4 5 7  
1 2 6  
8 9 3  
Dato a buscar: 10  
ERROR! Dato no encontrado  
Dato a buscar: 6  
Posicion: Fila 2 - Columna 3  
  
<< El programa ha finalizado: codigo de salida: 0 >>  
<< Presione enter para cerrar esta ventana >>_
```