



Sistema de Gestión de Cursos en C++ {

```
cout<<"Josue Adrian Sosa  
Cruz";  
}
```





Contenidos

- 01 Introducción
- 02 Objetivo del Proyecto
- 03 Tecnologías Utilizadas
- 04 Arquitectura del Sistema
- 05 Diagrama UML
- 06 Funcionamiento del Programa
- 07 Resultados Obtenidos
- 08 Conclusiones



Introducción {

La gestión manual de información académica puede generar errores y dificultar el seguimiento del rendimiento estudiantil.

Este proyecto propone una solución informática desarrollada en C++ que permite administrar cursos, estudiantes y profesores de manera organizada, aplicando Programación Orientada a Objetos y manejo de archivos.

- Estructuración.
- Revisión de errores.
- Corrección.
- Asignación del equipo.
- Formateo.
- Limpieza.
- Exportación.
- Depuración.

}



Objetivo del Proyecto

Objetivo general:

Desarrollar un sistema en C++ que permita gestionar cursos, estudiantes y profesores aplicando Programación Orientada a Objetos.

Objetivos específicos:

Aplicar encapsulamiento, herencia y polimorfismo
Gestionar notas y promedios
Generar reportes en archivos de texto

```
int main() {
    int opcion;

    do {
        system("cls");
        menuPrincipalColores();
        cin >> opcion;
        system("cls");
        switch (opcion) {
            case 1:
                crearCurso();
                break;
            case 2:
                mostrarCursos();
                break;
            case 3:
                menuCurso();
                break;
            case 4:
                menuEstudiantes();
                break;
            case 5:
                menuNotas();
                break;
            case 6:
                menuReportes();
                break;
            default:
                cout << "Opción
inválida.\n";
                break;
        }
        if (opcion != 7) {
            system("pause");
        }
    } while (opcion != 7);

    return 0;
}
```



Tecnologías Utilizadas {

01

Lenguaje de
programación
: C++

02

Paradigma:
Programación
Orientada a
Objetos

03

Librerías

04

iostream

05

string

06

fstream

07

algorithm

}

Arquitectura del Sistema {

```
#ifndef PERSONA_H
#define PERSONA_H

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

class Persona {
protected:
    string nombre;
    bool activo;

public:
    Persona() {
        nombre = "";
        activo = false;
    }

    void setNombre(string n) {
        nombre = n;
        activo = true;
    }

    bool estaActivo() {
        return activo;
    }

    string getNombre() {
        return nombre;
    }

    virtual void mostrarInfo() {
        cout << "Nombre: " << nombre << endl;
    }

    virtual ~Persona() {}
};

#endif
```

```
#ifndef PROFESOR_H
#define PROFESOR_H

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

class Profesor : public Persona {
public:
    string especialidad;
    int codigo;

    Profesor() {
        especialidad = "";
        codigo = 0;
        activo = false;
    }

    void mostrarInfo() override {
        cout << "\n--- PROFESOR ---\n";
        cout << "Nombre: " << nombre << endl;
        cout << "Especialidad: " << especialidad << endl;
        cout << "Código: " << codigo << endl;
    }
};

#endif
```

Arquitectura del Sistema {

```
#ifndef ESTUDIANTE_H
#define ESTUDIANTE_H

#include "persona.h"

class Estudiante : public Persona {
public:
    float notas[5];

    Estudiante() {
        for (int i = 0; i < 5; i++)
            notas[i] = -1;
    }

    float promedio() {
        float suma = 0;
        int cont = 0;
        for (int i = 0; i < 5; i++) {
            if (notas[i] >= 0) {
                suma += notas[i];
                cont++;
            }
        }
        return (cont > 0) ? suma / cont : 0;
    }

    void mostrarInfo() override {
        cout << "Estudiante: " << nombre
            << " | Promedio: " << promedio() << endl;
    }
};

#endif
```

```
#ifndef CURSO_H
#define CURSO_H

#include "profesor.h"
#include "estudiante.h"

const int MAX_EST = 30;

class Curso {
public:
    string nombreCurso;
    Profesor profesor;
    Estudiante estudiantes[MAX_EST];
    int totalEstudiantes;
    bool activo;

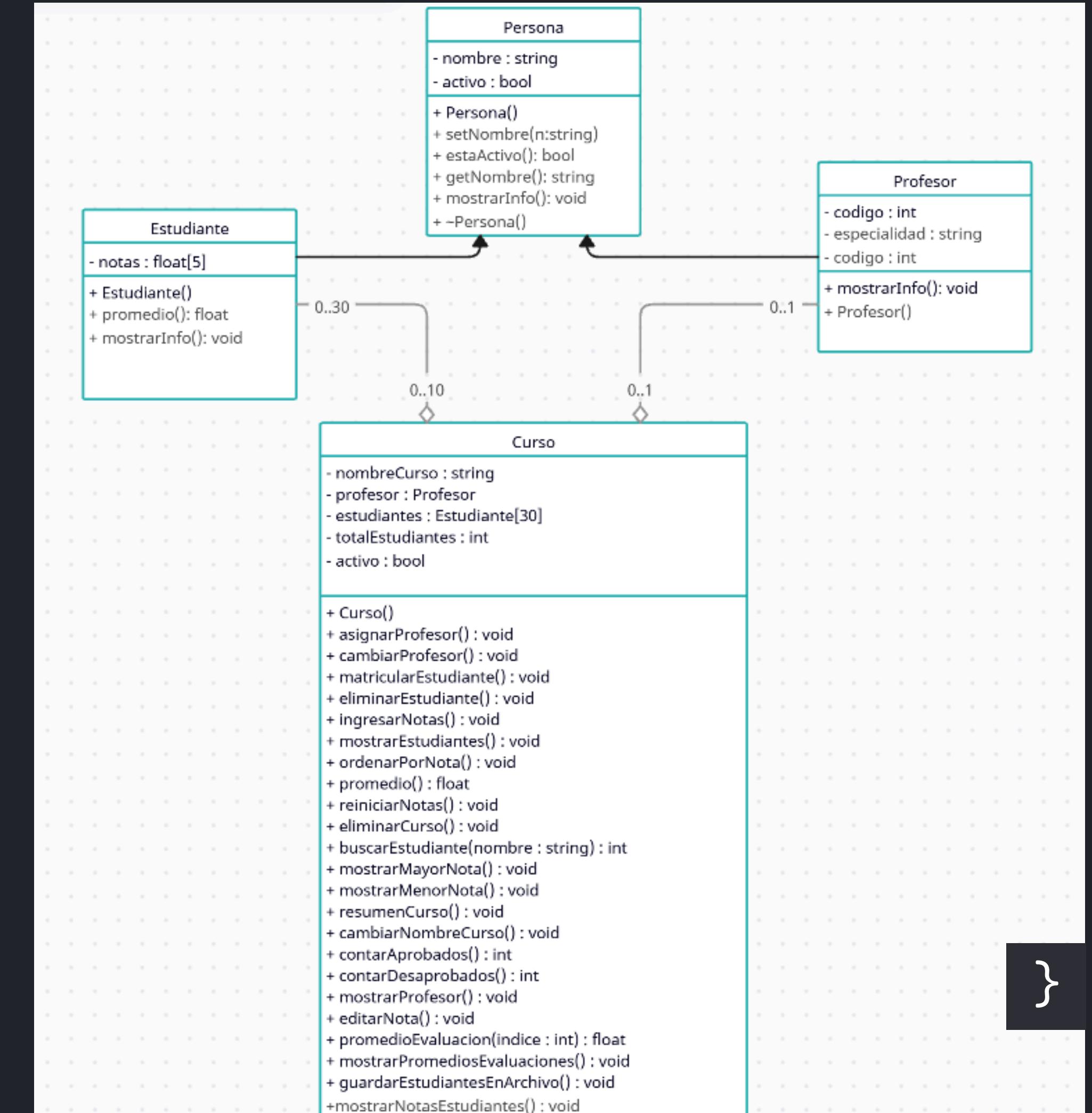
    Curso();

    void asignarProfesor();
    void cambiarProfesor();
    void matricularEstudiante();
    void eliminarEstudiante();
    void ingresarNotas();
    void mostrarEstudiantes(); |
    void ordenarPorNota();
    float promedio();
    void reiniciarNotas();
    void eliminarCurso();
    int buscarEstudiante(string nombre);
    void mostrarMayorNota();
    void mostrarMenorNota();
    void resumenCurso();
    void cambiarNombreCurso();
    int contarAprobados();
    int contarDesaprobados();
    void mostrarProfesor();
    void editarNota();
    float promedioEvaluacion(int indice);
    void mostrarPromediosEvaluaciones();
    void guardarEstudiantesEnArchivo();
};

#endif
```



Diagrama de Clases





Funcionamiento del Programa {

}



Resultados Obtenidos {

- El sistema gestiona correctamente cursos, profesores y estudiantes.
- Se aplicaron los principios de Programación Orientada a Objetos:
- encapsulamiento, herencia y polimorfismo.
- Se implementó el manejo de archivos para generar reportes en formato .txt.
- El programa funciona de forma estable mediante menús en consola.
- Se cumplieron los objetivos planteados en el curso de Programación Avanzada.

25%

Bugs reportados la semana anterior.

256

Horas invertidas en solución de errores.

187.000

Líneas de código formateadas y optimizadas.

458

Datos duplicados.

387

Horas invertidas en las apis.

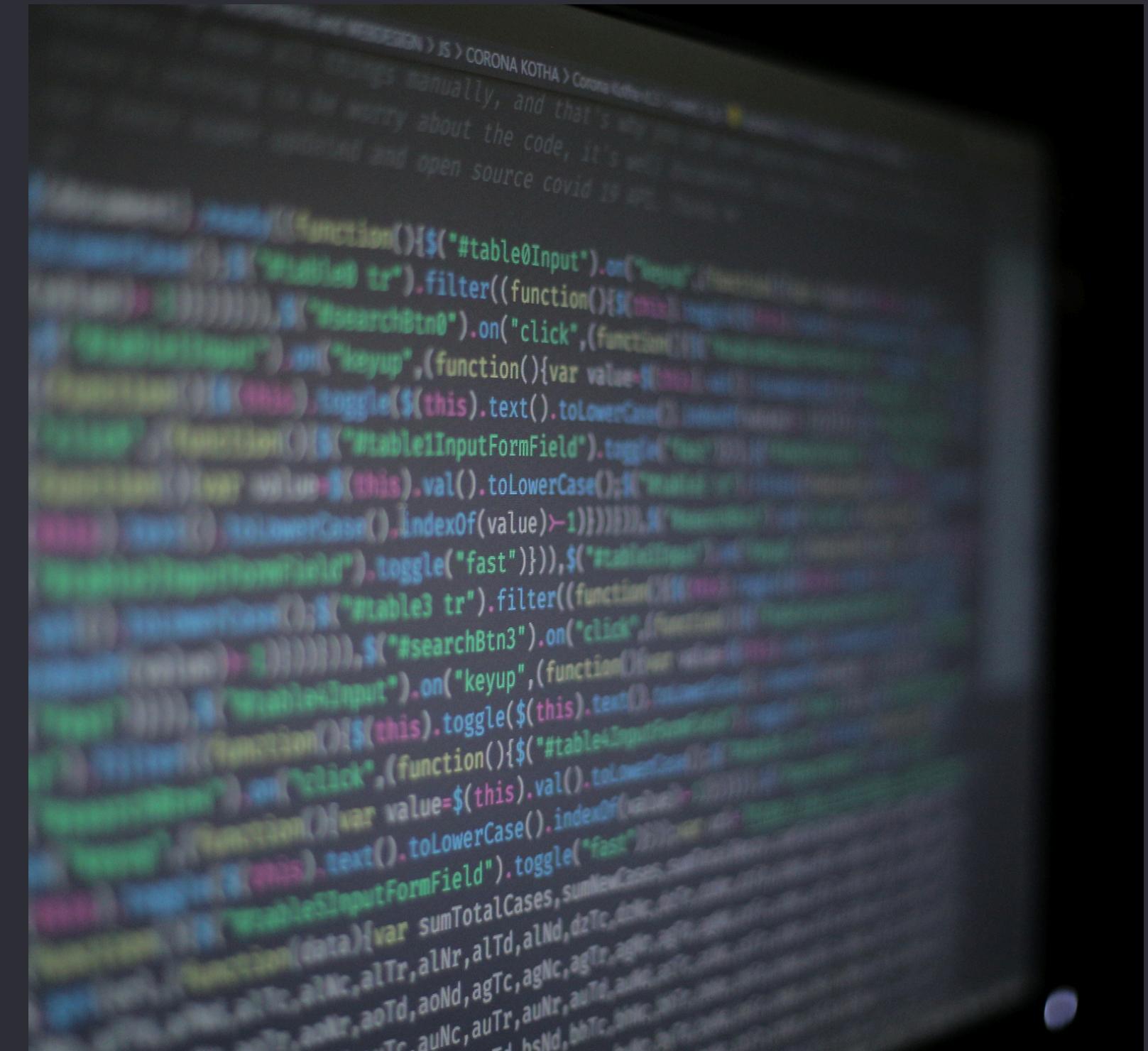
}



Conclusiones {

El proyecto permitió aplicar de manera práctica los conceptos de Programación Orientada a Objetos en C++, integrando herencia, polimorfismo y manejo de archivos.

El sistema desarrollado es funcional, organizado y cumple con los objetivos del curso de Programación Avanzada, sirviendo como base para futuros sistemas más complejos.



}



<!--Estudio Shonos-->

Gracias {

cout<<"Josue Adrian Sosa

Cruz";

}