

Universidad Mariano Gálvez de Guatemala

Facultad de Ingeniería Matemática y Ciencias Físicas

Campus Villa Nueva, Guatemala

Ingeniería en Sistemas de información y Ciencias de la computación

Ingeniero: Carlos Alejandro Arias

Curso: Física 1

Código de Curso: 012

Código de Carrera: 5090



## Laboratorio 6

Byron Ignacio Salazar Orellana

Sección: A

No. Carné: 23-5096

Fecha: 09/03/2024

## Introducción

El siguiente trabajo presenta la documentación e informe acerca de una serie de 5 programas que se basan en programación orientada a objetos lo cual, como estudiantes es algo de vital importancia que tengamos conocimiento y practiquemos, debido a su implementación dentro del ámbito de la programación en general, el hecho de poder generar clases y en base a eso asignarle datos nos puede facilitar a la hora de realizar aplicaciones.

```

#include <iostream>
#include <string>
// para el siguiente programa haremos una sección de libros
using namespace std;
// encontramos la clase libros que engloba los datos de titulo, autor y año de publicacion
class Libro {
private:
    string titulo;
    string autor;
    int year_publicacion;
    // Aqui asignamos las variables de los datos
public:
    Libro(string t, string a, int year){
        titulo = t;
        autor = a;
        year_publicacion = year;
        // estos son los métodos para establecer datos de los libros
    }
}

```

Para comenzar, tenemos el primer programa que trata de datos sobre un libro con programación orientada a objetos, tenemos la clase libro la cual contiene métodos privados y públicos.

```

// estos son los métodos para establecer datos de los libros
}

void ingTitulo(string t){
    titulo = t;
}

void ingAutor(string a){
    autor = a;
}

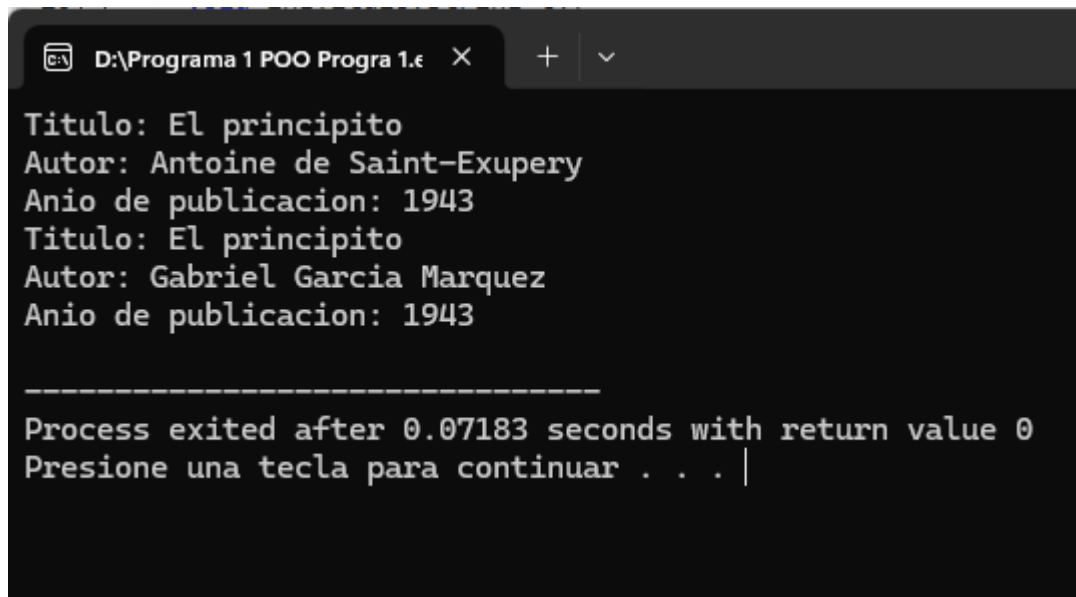
void ingYear_publicacion(int year){
    year_publicacion = year;
}
// estos metodos obtienen los datos de los libros
string egTitulo(){
    return titulo;
}
string egAutor(){
    return autor;
}
int egYear_publicacion(){
    return year_publicacion;
}
// Se asignan los datos para mostrar la informacion en el libro
void datos(){
    cout << "Titulo: " << titulo << endl;
    cout << "Autor: " << autor << endl;
    cout << "Anio de publicacion: " << year_publicacion << endl;
}

```

Ahora se asignan los datos para mostrar la información sobre el libro en la pantalla, siendo asignados a variables.

```
    }  
};  
// el programa principal muestra en la pantalla la clase y los datos  
int main(){  
    Libro libro1("El principito", "Antoine de Saint-Exupery", 1943);  
  
    libro1.datos();  
  
    libro1.ingAutor("Gabriel Garcia Marquez");  
  
    libro1.datos();  
  
    return 0;  
}
```

Por último, la función principal es el encargado de mostrar en pantalla cada uno de los datos del libro, como autor, nombre y año de publicación.



The screenshot shows a terminal window with a dark background. The title bar at the top reads "D:\Programa 1 POO Progra 1.e" with standard window controls. The terminal output displays the program's execution results. It first shows the data for a book object created with the constructor: "Titulo: El principito", "Autor: Antoine de Saint-Exupery", and "Anio de publicacion: 1943". Then, after calling the "ingAutor" method, it shows the updated data: "Titulo: El principito", "Autor: Gabriel Garcia Marquez", and "Anio de publicacion: 1943". A horizontal line of dashes separates this from the final output, which states "Process exited after 0.07183 seconds with return value 0" and "Presione una tecla para continuar . . . |".

```
D:\Programa 1 POO Progra 1.e X + v  
Titulo: El principito  
Autor: Antoine de Saint-Exupery  
Anio de publicacion: 1943  
Titulo: El principito  
Autor: Gabriel Garcia Marquez  
Anio de publicacion: 1943  
  
-----  
Process exited after 0.07183 seconds with return value 0  
Presione una tecla para continuar . . . |
```

```

#include <iostream>
#include <string>

using namespace std;
// La clase estudiante contiene los datos de los estudiantes, como métodos
class estudiante {
private:
    string nombre;
    string apellido;
    int edad;
    string curso;
// consta de metodos publicos y privados.
public:
    estudiante(string n, string a, int e, string c){
        nombre = n;
        apellido = a;
        edad = e;
        curso = c;
    }
}

```

Para el segundo programa tenemos datos de un estudiante en un curso, como su nombre, apellido, su edad y el nombre del curso. Estos están definidos por métodos públicos y privados.

```

public:
    estudiante(string n, string a, int e, string c){
        nombre = n;
        apellido = a;
        edad = e;
        curso = c;
    }
    // esta sección establece los datos del estudiante
    void ingNombre(string n){
        nombre = n;
    }
    void ingApellido(string a){
        apellido = a;
    }
    void ingEdad(int e){
        edad = e;
    }
    void ingCurso(string c){
        curso = c;
    }
    // se obtienen los datos del estudiante retornandolo.
    string egNombre(){
        return nombre;
    }
    string egApellido(){
        return apellido;
    }
    int egEdad(){
        return edad;
    }
    string egCurso(){
        return curso;
    }
}

```

Luego de ello, se asignan cada uno de los datos a variables para luego mostrárselos al usuario en pantalla, se retornan para obtener los mismos.

```

    // método para mostrar la información del estudiante
    void datos(){
        cout << "Nombre: " << nombre << endl;
        cout << "Apellido: " << apellido << endl;
        cout << "Edad: " << edad << endl;
        cout << "Curso: " << curso << endl;
    }
};

// la función principal es la encargada de mostrar los datos
int main(){
    estudiante estudiante1("Byron","Salazar", 19, "Programacion ");
    estudiante1.datos();
    estudiante1.ingCurso("Programacion 1");
    estudiante1.datos();

    return 0;
}

```

Por último, la función principal muestra los datos del estudiante en pantalla, esto gracias a los métodos tomados como variables.

```
D:\Ejercicio 2 POO Progra 1.ex X + v
Nombre: Byron
Apellido: Salazar
Edad: 19
Curso: Programacion
Nombre: Byron
Apellido: Salazar
Edad: 19
Curso: Programacion 1

-----
Process exited after 0.1446 seconds with return value 0
Presione una tecla para continuar . . . |
```

```
#include <iostream>
// Para este programa tenemos una calculadora programada con orientación a objetos
using namespace std;
// la clase calculadora pide el ingreso de dos numetos
class calculadora {
private:
    double num1;
    double num2;
    // cuenta con metodos publicos y privados para su uso
public:
    calculadora(double numero1, double numero2) {
        num1 = numero1;
        num2 = numero2;
    }
}
```

Para el tercer programa tenemos una calculadora con programación orientada a objetos, donde se pide el ingreso de dos números.

```

// en esta sección establecemos las operaciones, retornando los resultados
double suma() {
    return num1 + num2;
}

double resta() {
    return num1 - num2;
}

double multiplicacion() {
    return num1 * num2;
}

// para la división tomamos en cuenta que no se puede dividir en 0
double division() {
    if (num2 != 0)
        return num1 / num2;
    else {
        cout << "No se puede dividir entre 0" << endl;
        return 0;
    }
}

```

Para obtener los datos se retornan las operaciones según el símbolo de operación, tomamos en cuenta que no se puede dividir entre 0.

```

// la función principal nos muestra el resultado de las operaciones
// también se realiza el ingreso de datos
int main() {
    double num1, num2;

    cout << "Ingrese el primer numero: ";
    cin >> num1;

    cout << "Ingrese el segundo numero: ";
    cin >> num2;

    calculadora calcular(num1, num2);

    cout << "Suma: " << calcular.suma() << endl;
    cout << "Resta: " << calcular.resta() << endl;
    cout << "Multiplicacion: " << calcular.multiplicacion() << endl;
    cout << "Division: " << calcular.division() << endl;

    return 0;
}

```

Por último la función principal le muestra al usuario los resultados de las operaciones según los números que ingresó.



```
D:\Programa 3 POO Progra 1.1 X + v
Ingrese el primer numero: 10
Ingrese el segundo numero: 5
Suma: 15
Resta: 5
Multiplicacion: 50
Division: 2

-----
Process exited after 5.738 seconds with return value 0
Presione una tecla para continuar . . . |
```

Funcionamiento de la aplicación.

```
1  #include <iostream>
2  #include <string>
3  // Para el cuarto programa se nos solicita hacer una cuenta bancaria
4  using namespace std;
5  // la clase cuenta_banco representa la cuenta de banco, que tiene
6  // datos como el titular, numero de cuenta y el saldo
7  class cuenta_banco{
8      private:
9          string nombre_titular;
10         string numero_cuenta;
11         double saldo;
12
13     public:
14         cuenta_banco(string nombre, string numero, double saldo_actual){
15             nombre_titular = nombre;
16             numero_cuenta = numero;
17             saldo = saldo_actual;
18         }
19 }
```

En este programa, de una cuenta de banco con programación orientada a objetos, obtendremos datos como el nombre del titular, cuenta de banco y el saldo, esto para poder hacer transacciones.

```

// en esta sección podremos ver el depósito que realiza hacia la cuenta de banco.
void deposito (double monto){
    saldo += monto;
    cout << "Deposito de " << monto << " realizado con exito" << endl;
}
// en esta sección podremos retirar dinero de la cuenta siempre y cuando el saldo sea mayor
// que la cantidad a retirar
void retiro(double monto){
    if (saldo >= monto){
        saldo -= monto;
        cout << "Retiro de " << monto << " realizado con exito" << endl;
    } else {
        cout << "Saldo insuficiente" << endl;
    }
}
// consultar el saldo de la cuenta
double consulta_saldo(){
    return saldo;
}

```

Podremos hacer transacciones como depósitos o retiros, este último teniendo en cuenta el saldo de la cuenta que el saldo sea mayor que el monto a retirar. Luego podremos consultar el saldo de cuenta retornando el valor del mismo.

```

39 // la funcion principal muestra los datos de la cuenta de banco.
40 int main(){
41     cuenta_banco cuenta("Byron Salazar", "213455", 500.00);
42     cuenta.deposito(500.00);
43     cuenta.retiro(200.00);
44
45     cout << "Saldo actual: " << cuenta.consulta_saldo() << endl;
46     return 0;
47 }

```

Por último, la función principal es la encargada de enseñarle al usuario de los datos de la cuenta de banco.

```
Deposito de 500 realizado con exito
Retiro de 200 realizado con exito
Saldo actual: 800

-----
Process exited after 0.2179 seconds with return value 0
Presione una tecla para continuar . . . |
```

Funcionamiento de la aplicación.

```
1  #include <iostream>
2  #include <string>
3  // para el último programa, se nos pide programar sobre películas
4  using namespace std;
5  // la clase película es la que contiene datos sobre las mismas
6  // como título, director, año de lanzamiento.
7  class Pelicula {
8  private:
9      string titulo;
10     string director;
11     int year_lanzamiento;
12
13 public:
14     Pelicula(string t, string d, int year) {
15         titulo = t;
16         director = d;
17         year_lanzamiento = year;
18     }
```

Para el último programa se nos solicita hacer algo similar al primero, pero esta vez con películas. Tenemos métodos públicos y privados.

```

// esta sección establece los datos para llevarlos com variables
void ingTitulo(string t) {
    titulo = t;
}

void ingDirector(string d) {
    director = d;
}

void ingYear_lanzamiento(int year) {
    year_lanzamiento = year;
}
// esta sección obtiene los datos sobre la pelicula
string egTitulo() {
    return titulo;
}

string egDirector() {
    return director;
}

int egYear_lanzamiento() {
    return year_lanzamiento;
}

```

El código convierte cada uno de los métodos en variables para así mostrarlas en el programa principal, como datos de la película.

```

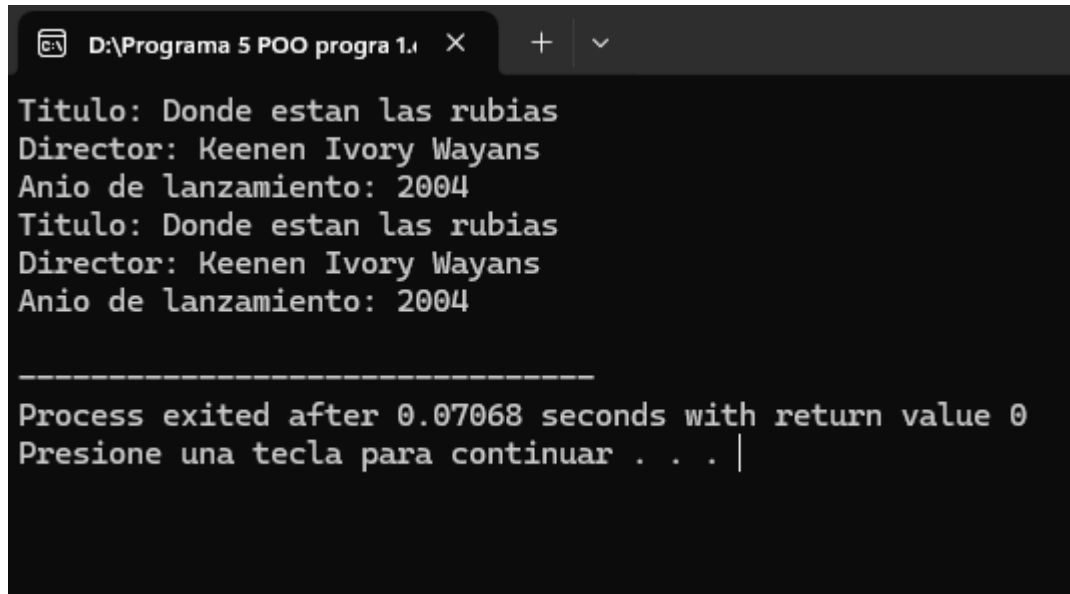
// esta sección escribe los datos sobre la pelicula
void datos() {
    cout << "Titulo: " << titulo << endl;
    cout << "Director: " << director << endl;
    cout << "Año de lanzamiento: " << year_lanzamiento << endl;
}
};
// la funcion principal se encarga de mostrar los datos de la pelicula al usuario.
int main() {
    Pelicula peli("Donde estan las rubias", "Keenen Ivory Wayans", 2004);

    peli.datos();
    peli.ingDirector("Keenen Ivory Wayans");
    peli.datos();

    return 0;
}

```

Por último, la función principal nos permite mostrar los datos de la película al usuario. Todo esto con programación orientada a objetos.



```
D:\Programa 5 POO progra 1.1 X + v
Titulo: Donde estan las rubias
Director: Keenen Ivory Wayans
Anio de lanzamiento: 2004
Titulo: Donde estan las rubias
Director: Keenen Ivory Wayans
Anio de lanzamiento: 2004

-----
Process exited after 0.07068 seconds with return value 0
Presione una tecla para continuar . . . |
```

Funcionamiento del programa.

## Conclusiones

Podemos decir que la programación orientada a objetos tiene diferentes usanzas dentro del ámbito de la creación de aplicaciones, en un lenguaje como c++ el cual es sumamente popular, podremos apoyarnos de la POO para la fácil creación de programas como bases de datos, en donde veríamos todo el potencial que tiene este tipo de programación.