

# Introducción a la programación orientada a objetos (IPOO)

Objetos: Introducción

[carlos.andres.delgado@correounivalle.edu.co](mailto:carlos.andres.delgado@correounivalle.edu.co)

Carlos Andrés Delgado S.

Facultad de Ingeniería. Universidad del Valle

Abril 2017

Introducción a  
la  
programación  
orientada a  
objetos  
(IPOO)

Carlos Andrés  
Delgado S.

Introducción a  
los objetos

Atributos y  
métodos

Objetos en  
C++

## 1 Introducción a los objetos

## 2 Atributos y métodos

## 3 Objetos en C++

Introducción a  
la  
programación  
orientada a  
objetos  
(IPOO)

Carlos Andrés  
Delgado S.

Introducción a  
los objetos

Atributos y  
métodos

Objetos en  
C++

## 1 Introducción a los objetos

## 2 Atributos y métodos

## 3 Objetos en C++

## Enfoque de los objetos

- 1 Se utiliza para modelar problemas de la vida real
- 2 Un objeto tiene propiedades (atributos)
- 3 Un objeto realiza acciones (métodos)

# Introducción a los objetos

Introducción a  
la  
programación  
orientada a  
objetos  
(IPOO)

Carlos Andrés  
Delgado S.

Introducción a  
los objetos

Atributos y  
métodos

Objetos en  
C++

## ¿Que es un objeto?



# Introducción a los objetos

Introducción a  
la  
programación  
orientada a  
objetos  
(IPOO)

Carlos Andrés  
Delgado S.

Introducción a  
los objetos

Atributos y  
métodos

Objetos en  
C++

## ¿Que es un objeto?

Son entidades que describen el comportamiento de una entidad real, los objetos tiene:

- 1 Una identidad o un nombre
- 2 Una funcionalidad o comportamiento
- 3 Estado interno

## Relaciones entre objetos

Los objetos se pueden relacionar con otros objetos de distintas formas:

- 1 **Es un** (herencia)
- 2 **Es parecido** (herencia añadiendo funciones)
- 3 **Posee un** (Un objeto puede tener objetos en su estado interno)
- 4 **Conoce** (Un objeto puede tener referencias a otros objetos sin tomar propiedad)

Esto lo veremos en la próxima clase :)

# Introducción a los objetos

Introducción a  
la  
programación  
orientada a  
objetos  
(IPOO)

Carlos Andrés  
Delgado S.

Introducción a  
los objetos

Atributos y  
métodos

Objetos en  
C++

## Pensemos en objetos

Piense que estado y funciones pueden tener los siguientes objetos:

- 1 Un bombillo
- 2 Una puerta
- 3 Un dado
- 4 Un carro



## Pensemos en objetos

### 1 Un bombillo

- **Estado:** Prendido o apagado
- **Funciones:** Prender o apagar

### 2 Una puerta

- **Estado:** Abierta o cerrada
- **Funciones:** Abrir o cerrar

## Pensemos en objetos

### 3 Un dado

- **Estado:** La cara que se puede ver
- **Funciones:** Lanzar o ver cara

### 4 Un carro

- **Estado:** Prendido o apagado, posición actual, kilómetros recorridos, nivel de gasolina del carro
- **Funciones:** Encender o apagar, avanzar hasta cierto punto



# Introducción a los objetos

# Introducción a la programación orientada a objetos (IPOO)

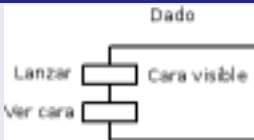
Carlos Andrés  
Delgado S.

## Introducción a los objetos

## Propiedades de los objetos

- 1 El estado interno de un objeto es inaccesible desde el exterior de ese objeto
- 2 Sólo se puede modificar el estado interno de un objeto es llamando alguna función del objeto

## Ejemplo



## Agrupación

Los objetos similares se pueden agrupar:

- 1 CD de Vallenato, CD de salsa, CD de música clásica
- 2 Mesa de noche izquierda, mesa de noche frontal.
- 3 Carro azul, carro blanco, carro verde.

Los objetos que se pueden agrupar se dicen que son de la misma **clase**

## Clase

- 1 Una clase es un conjunto que representa objetos de características similares
- 2 Cada objeto es una **instancia** de una clase en donde la Clase representa un "Tipo"

# Introducción a los objetos

Introducción a  
la  
programación  
orientada a  
objetos  
(IPOO)

Carlos Andrés  
Delgado S.

Introducción a  
los objetos

Atributos y  
métodos

Objetos en  
C++

## Ejemplos: Diseño con objetos

Nombre objeto	Nombre clase
cdArjona	CdDeMusica
cdLosEnanos	CdDeMusica
tepeteNegro	Tapete
tapeteBlanco	Tapete
camaDeMadera	Cama
camaMetalica	Cama

## Los objetos en C++

En C++ existen mecanismos que relacionan los objetos unos con otros:

- 1 **Herencia:** La relación se crea para siempre
- 2 **Punteros:** Es una relación modificable, se puede cambiar a donde apunta un puntero

## Los objetos en C++

- 1 En C++ el mecanismo más usado son los punteros, ya que son flexibles
- 2 En otros lenguajes (Como java) el mecanismo más usado es la herencia
- 3 Cada mecanismo tiene sus ventajas y desventajas. A ustedes les toca aprender todos



# Introducción a los objetos

Introducción a  
la  
programación  
orientada a  
objetos  
(IPOO)

Carlos Andrés  
Delgado S.

Introducción a  
los objetos

Atributos y  
métodos

Objetos en  
C++

## Otros conceptos de O.O

- 1 **Algoritmo:** Secuencia de pasos elementales en los cuales se realiza una tarea compleja
- 2 **Implementación:** Secuencia de instrucciones, que conforman un programa

## Aclaración

Ambos conceptos son similares

- 1 Los algoritmos son abstractos (ideas)
- 2 Los algoritmos finalmente, se deben implementar en un lenguaje de programación

## Programación con objetos

En un lenguaje orientado a objetos (O.O) los programas se construyen:

- Construyendo clases
- Definiendo objetos a partir de las clases
- Definiendo relaciones entre esos objetos
- Programando el intercambio de mensajes entre esos objetos

## Diseño con objetos

Algunas reglas de diseño con objetos.

- 1 Los nombres de **objetos** inician en minúsculas, si hay espacio entre las palabras, a partir de la segunda palabra cada palabra inicia con mayúscula
- 2 Los nombres de **clases** trabajan de igual manera que los objetos pero comienzan en mayúscula.

Esto es importante tenerlo en cuenta, algunos lenguajes de programación, se exige el correcto nombramiento de las clases y objetos. Mayor información

[https://es.wikipedia.org/wiki/Convenci%C3%B3n\\_de\\_nombres\\_\(programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Convenci%C3%B3n_de_nombres_(programaci%C3%B3n))

Introducción a  
la  
programación  
orientada a  
objetos  
(IPOO)

Carlos Andrés  
Delgado S.

Introducción a  
los objetos

Atributos y  
métodos

Objetos en  
C++

1 Introducción a los objetos

2 Atributos y métodos

3 Objetos en C++

## Atributos

¿Que características tienen los siguientes clases de objetos?

- 1 CdDeMusica
- 2 Cama
- 3 Tapete

## Métodos

¿Que características tienen los siguientes clases de objetos?

- 1 CdDeMusica: Titulo de CD, Nombre de artista, año de grabación, número de canciones
- 2 Cama: Largo, ancho, material
- 3 Tapete: Color, material, largo, ancho

A cada una de estas características se le llama **atributo**.

## Métodos

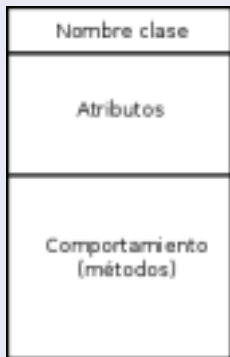
¿Que operaciones podemos realizar con las siguientes clases de objetos?

- 1 CdDeMusica:
- 2 Cama: Mover
- 3 Tapete: Colocar

A cada una de estas operaciones se le llama **métodos**.

## Atributos

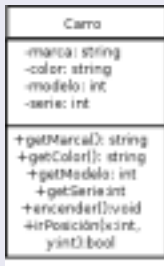
Existe una notación gráfica que permite definir las clases:





## Atributos

### Ejemplo:



Introducción a  
la  
programación  
orientada a  
objetos  
(IPOO)

Carlos Andrés  
Delgado S.

Introducción a  
los objetos

Atributos y  
métodos

Objetos en  
C++

1 Introducción a los objetos

2 Atributos y métodos

3 Objetos en C++

## Definición

Para trabajar objetos en C++ necesitamos:

- Un archivo de cabecera (.h) para la **definición** de clases, atributos y métodos
- Un archivo de implementación (.cpp) (.cc) para la implementación

Esto permite separar la definición de la implementación. Un archivo .h no debe contener implementación sólo definiciones.

## Definición

Un archivo de cabecera permiten establecer un contrato entre una librería (programa que usted construye) y lo que permite se use de ella. En otras palabras es indicar **lo que hace una librería**

## En C++

En C++ vamos a incluir un nuevo tipo de archivo, cuya extensión es **.h**

## Definición

Cuando se construye un archivo de cabecera, el compilador obliga a imponer un contrato, es decir se deben declarar todas las clases, variables y métodos antes de poder ser usados. Con esto se garantiza la consistencia y evitar errores.

## Importante

Los archivos de cabecera deben cumplir una regla: **únicamente declaraciones**, es decir no se deben definir los valores de las variables ni los cuerpos de las funciones.

## Estructura

La estructura de los archivos de cabecera en C++ es la siguiente:

```
#ifndef HEADER_FLAG
#define HEADER_FLAG
    class <nombre>{
        //Atributos
        public: //Son accesibles por todo el mundo

        private: //unicamente el objeto los conoce

        protected: //Se explicara en relacion de herencia
        //Métodos
    };
#endif
```

## Estructura

Vamos a definir una clase. Archivo **Mochila.h**

```
#ifndef MOCHILA.H
#define MOCHILA_H

class Mochila{
    private:
        //Atributos deben ser privados
        int numeroElementos;

    public:
        //Metodos
        Mochila(); //Constructor
        ~Mochila(); //Destructor
        void ingresarElemento();
        bool eliminarElemento();
};

#endif
```

## Estructura

Sigamos con el programa **Mochila.cpp**.

```
#include "Mochila.h"

Mochila::Mochila(){
    numeroElementos = 0;
}

Mochila::~Mochila(){}

void Mochila::ingresarElemento(){
    numeroElementos++;
}

bool Mochila::eliminarElemento(){
    if (numeroElementos > 0){
        numeroElementos--;
        return true;
    }
    else{
        return false;
    }
}
```



## Estructura

Y finalmente, nuestro archivo principal. **Main.cpp**

```
//Importamos nuestro objeto (declaracion e implementacion)
#include "Mochila.h"

int main(int argc, char *[] argv){
    //DECLARACION ESTATICA
    //Al crear el objeto de inmediato se ejecuta el constructor
    Mochila mochilaDeCarlos;
    //Ejecutamos el metodo ingresar elemento
    mochilaDeCarlos.ingresarElemento();

    //DECLARACION DINAMICA
    //Al asignar espacio de memoria, ejecutamos el constructor
    Mochila * mochilaDeJuan = new Mochila();
    //Ejecutamos un metodo
    mochilaDeJuan->ingresarElemento();

    //Al terminar, liberemos la memoria
    delete mochilaDeJuan;
    //Al hacer esto se ejecuta el destructor
}
```

## Ejercicio

Construir una clase Bicicleta

### Atributos

- 1) Color
- 2) Marca
- 3) Peso
- 4) Tamaño (Size)

### Metodos

- 1) GetColor, getMarca, getPeso, getSize
- 2) SetColor, setMarca, setPeso, setSize

Instanciar 3 objetos de bicicleta

## Ejercicio

Genere un programa **calculadora** bajo el enfoque orientado a objetos:

- 1 Los métodos que se tienen son: sumar, restar, multiplicar, dividir y módulo. Todos reciben dos números reales y retornan un número real. Recuerde en el módulo y división hacer la verificación de cero
- 2 Bajo el enfoque orientado a objetos, los objetos **no deberían enviar mensajes directamente al usuario**, es decir evite usar cout y cin dentro de los objects
- 3 Genere un menú, cuyas opciones son: Sumar, restar, multiplicar, dividir, módulo y salir. En las operaciones siempre solicita dos números, se hace la operación y se retorna el resultado.

# ¿Preguntas?

Introducción a  
la  
programación  
orientada a  
objetos  
(IPOO)

Carlos Andrés  
Delgado S.

Introducción a  
los objetos

Atributos y  
métodos

Objetos en  
C++

