

# Programación Orientada a Objetos

Cristo Alvarado

14 de julio de 2025

## Resumen

Ejemplo Abstract.

## ÍNDICE

---

§1	Fundamentos	1
§1.1	¿Qué es la POO?	1
§1.2	Clases y Objetos	3
§1.3	Atributos	5

## LISTA DE CÓDIGOS

---

1	Declaración de Clase en Java.	4
2	Declaración de Objeto en Java.	4
3	Declaración de Objeto en Java.	4
4	Archivo con la Clase Main.	4
5	Archivo con la Clase Second.	5

## §1 FUNDAMENTOS

---

### §1.1 ¿QUÉ ES LA POO?

---

#### **Definición 1.1 (Paradigma de Programación)**

Un **paradigma de programación** es una manera o estilo de programación de software. Se trata de un conjunto de métodos sistemáticos aplicables en todos los niveles del diseño de programas para resolver problemas computacionales.

Básicamente, es una forma de pensar sobre cómo estructurar y escribir el código. Cada paradigma de programación ofrece diferentes tipos de enfoques/filosofías/reglas para abordar tareas específicas.

Existen varios paradigmas de programación, algunos de los más comunes son los siguientes:

### Ejemplo 1.1 (Paradigma Imperativo)

Este paradigma se basa en la ejecución secuencial de instrucciones, donde el programador especifica cada paso que la computadora debe realizar para lograr un resultado. Incluye lo siguiente:

- Programación estructurada.
- Programación Procedural.
- Programación Modular.

Lenguajes como C, C++, y Java pueden ser utilizados bajo este paradigma.

### Ejemplo 1.2 (Paradigma Orientado a Objetos)

En el **Paradigma Orientado a Objetos** (también llamado **Programación Orientada a Objetos** y abreviado **POO**) se construyen modelos de objetos que representan elementos (objetos) del problema a resolver, que tienen características y funciones. Permite separar los diferentes componentes de un programa, simplificando así su creación, depuración y posteriores mejoras. La programación orientada a objetos disminuye los errores y promueve la reutilización del código. Es una manera especial de programar, que se acerca de alguna manera a cómo expresaríamos las cosas en la vida real.

Lenguajes de programación orientados a objetos son Java, Python o C#.

### Observación 1.1 (Objetos)

En el paradigma orientado a objetos, podemos definir un objeto como una estructura abstracta que, de manera más fiable, describe un posible objeto del mundo real y su relación con el resto del mundo que lo rodea a través de interfaces.

La programación orientada a objetos se sirve de diferentes conceptos como:

- Abstracción de datos
- Encapsulación
- Eventos
- Modularidad
- Herencia
- Polimorfismo

La programación orientada a objetos tiene varias ventajas:

- POO es más fácil y rápida de ejecutar.
- POO provee una estructura clara de los programas.
- POO ayuda a mantener la filosofía del código Java DRY “Don’t Repeat Yourself”, y hace que el código sea más fácil de mantener, modificar y debuggear.
- POO hace posible crear aplicaciones totalmente reusables con menos código y menos tiempo de desarrollo.

### Observación 1.2 (Filosofía DRY)

La filosofía DRY trata sobre la repetición del código. En vez de repetir un código, se debe intentar reutilizar el mismo cada que sea posible.

## §1.2 CLASES Y OBJETOS

Dos aspectos fundamentales de la POO son las **clases** y los **objetos**.

### Definición 1.2 (Clase)

En POO, una **clase** es una *plantilla o modelo que define las propiedades y comportamientos comunes de un conjunto de objetos*. Es en esencia una plantilla de lo que será un objeto.

Una clase *puede contener atributos (variables) y métodos (funciones) que describen el estado y el comportamiento de los objetos que se crean a partir de ella*.

### Definición 1.3 (Objeto)

En POO, un **objeto** es una *instancia de una clase*.

Cada objeto *tiene su propio estado y comportamiento, que se define por la clase a la que pertenece*. Los objetos son entidades concretas que se crean a partir de las clases y pueden interactuar entre sí.

### Definición 1.4 (Instancia)

Una **instancia** es un *objeto específico creado a partir de una clase*.

### Ejemplo 1.3 (Clases y Objetos)

La siguiente tabla muestra ejemplos de clases y objetos:

Clase	Objeto
Fruta	Manzana
	Plátano
	Mango
Carro	Volvo EX30
	Audi A3
	Toyota C-HR

Básicamente las clases son plantillas que definen las características y comportamientos de los objetos, mientras que los objetos son instancias concretas de esas clases.

Cuando los objetos individuales son creados a partir de una clase, ellos heredan todas las variables y métodos de la clase.

### Observación 1.3 (Java y POO)

Todo en Java está asociado con clases y objetos, junto con sus métodos y variables.

En la vida real un carro tiene **atributos**, como el color y su tamaño, y **métodos**, como acelerar y frenar.

#### Observación 1.4 (Declaración de Clases en Java)

Para declarar una clase en Java usamos la palabra reservada `class`:

```
public class Ejemplo{  
    int x = 5;  
}
```

Código 1: Declaración de Clase en Java.

Siempre el nombre de la clase debe comenzar con mayúsculas y debe ser el mismo que el nombre del archivo.

Como se mencionaba anteriormente en Java un objeto es creado a partir de una clase.

#### Observación 1.5 (Creación de Objetos en Java)

Para crear un objeto debemos especificar el nombre de la clase seguido del nombre del objeto, y luego usar la palabra reservada `new`, como se muestra a continuación:

```
public class Main {  
    int x = 5;  
    public static void main(String[] args) {  
        Main myObj = new Main();  
        System.out.println(myObj.x);  
    }  
}
```

Código 2: Declaración de Objeto en Java.

También es posible crear múltiples objetos de la misma clase:

```
public class Main {  
    int x = 5;  
    public static void main(String[] args) {  
        Main myObj1 = new Main();  
        Main myObj2 = new Main();  
        System.out.println(myObj1.x);  
        System.out.println(myObj2.x);  
    }  
}
```

Código 3: Declaración de Objeto en Java.

También podemos usar múltiples clases en y acceder a alguna de ella a la otra clase. Esto es usado comunmente para la organización de clases (una clase tiene todos los atributos y métodos, mientras que la otra clase tiene el método `main()`).

#### Ejemplo 1.4

En este ejemplo tenemos dos clases, `Main` y `Second`:

```
class Main {  
    int x = 10;  
    public static void main (String[] args){  
        Second segunda = new Second();  
    }  
}
```

```
        System.out.println("El numero de la clase Second es: " + segunda.x);  
    }  
}
```

Código 4: Archivo con la Clase **Main**.

y,

```
public class Second {  
    int x = 20;  
}
```

Código 5: Archivo con la Clase **Second**.

Al ejecutar el código anterior, la ejecución mostrará el número 20.

## §1.3 ATRIBUTOS

En los ejemplos anteriores nombramos a **x** como una variable, pero realmente esta es un **atributo** de la clase.

### **Definición 1.5 (Atributo)**

En POO, un **atributo** es una *característica o propiedad de un objeto*.