

PROGRAMACION ORIENTADA A OBJETOS

ASIGNATURA:

POO

PROFESOR: Ing. Yaira Franco R PERÍODO ACADÉMICO: 2025-A

TAREA S2_1

TÍTULO:

CLASES, OBJETOS, ATRIBUTOS Y METODOS

INGRESO DE DATOS

Nombre de estudiante

Alan Logroño Soria



2025-A

REVISAR LOS VIDEOS

<https://www.youtube.com/watch?v=pXX3c8BloY0> <https://www.youtube.com/watch?v=fdXkiuVHp4>

This en el Metodo constructor

https://www.youtube.com/watch?v=aUX_rQO1ac8

Método Constructor <https://www.youtube.com/watch?v=4plcVlxh0lw>

Creación de Objetos <https://www.youtube.com/watch?v=zhUTqZk3-ks>

NO OLVIDAR ¿Qué debe quedar muy claro?

- Primero va la clase.
- Dentro de la clase van los atributos y métodos.
- Después se crea el constructor.
- Luego en main() se instancia el objeto con new.
- Se usan los atributos y métodos a través del objeto.

INVESTIGE Y explique claramente PASO A PASO:

- Cómo se crea una clase.

Una clase es una plantilla o modelo que describe cómo será un objeto. En Java, una clase puede tener:

Atributos (también llamados propiedades o variables) → describen las características.

Métodos → son las acciones o comportamientos que puede hacer ese objeto.

```
public class Persona {  
  
    String nombre;  
    int edad;  
}
```

- Cómo se crea un constructor con y sin parámetros

SIN parámetros:

Un constructor es un método especial que inicializa el objeto cuando se crea.

```
public class Persona {
    String nombre;
    int edad;

    // Constructor sin parámetros
    public Persona() {
        nombre = "Sin nombre";
        edad = 0;
    }
}
```

Con Parámetros:

```
public class Persona {
    String nombre;
    int edad;

    // Constructor con parámetros
    public Persona(String nombre, int edad) {
        this.nombre = nombre;
        this.edad = edad;
    }
}
```

- **Como se crear métodos**

Un método es una función dentro de una clase (acciones que el objeto puede realizar).

```
public void saludar() {
    System.out.println("Hola, mi nombre es " + nombre);
}
```

- **Cómo se instancia un objeto y donde se lo ubica.**

Instanciar = crear un objeto usando la clase.

```
Persona p1 = new Persona();
```

- **Cómo se usan los atributos y métodos.**

Después de instanciar un objeto, puedes acceder a sus atributos y métodos:

```
p1.nombre = "Ana";
p1.saludar();
```

- **Como se reutiliza los métodos si tengo tres o mas objetos**

Cada objeto puede usar los mismos métodos, pero con sus propios datos:

Cada objeto usa su propia versión de los atributos dentro del mismo método saludar().

```
p1 = Persona("Ana", 25)
p2 = Persona("Pepe", 30)
p3 = Persona("Marta", 20)

p1.saludar()
p2.saludar()
p3.saludar()
```

- **Diferencia entre métodos constructor con parámetros y sin parámetros**

Constructor sin parámetros	Constructor con parámetros
No necesita datos al crear el objeto.	Necesita datos específicos al crear el objeto.
Usa valores por defecto.	Personaliza cada objeto al momento de la creación.
Ejemplo: Persona()	Ejemplo: Persona("Ana", 25)

- **Diferencia entre métodos**

Tipo de Método	Descripción	Ejemplo
Sin retorno (void)	Realizan acciones pero no devuelven valores.	saludar()
Con retorno	Devuelven un valor al programa.	calcularEdadFutura() que devuelve un int
Con parámetros	Reciben datos para trabajar.	cambiarNombre(String nuevoNombre)
Sin parámetros	No reciben datos.	mostrarDatos()

Prácticas Progresivas para Entender el Flujo de POO en Java

La siguiente serie de **cuatro prácticas** te ayudará a comprender paso a paso cómo funciona la Programación Orientada a Objetos (POO) en Java. Iremos desde lo más básico (crear una clase y un objeto) hasta utilizar múltiples objetos y distinguir claramente entre **constructores** y **métodos** normales. Cada práctica incluye un objetivo, una actividad propuesta y un pequeño ejemplo o guía de referencia. ¡Manos a la obra!

Práctica 1: Mi Primera Clase y Objeto

Objetivo: Entender qué es una **clase** y qué es un **objeto**, y aprender a definir una clase simple con atributos y métodos. Se busca que comprendas que primero se crea una *clase* (el plano o molde), y luego podemos crear *objetos* (instancias concretas) a partir de esa clase.

Actividad: Crea una clase sencilla y utilízala para instanciar (crear) un objeto en Java. Sigue estos pasos básicos:

1. **Definir una clase:** Por ejemplo, una clase Persona. Dentro de la clase, declara uno o más **atributos** (datos que describen a la persona, por ejemplo nombre) y un **método** (una acción o comportamiento, por ejemplo un método que haga que la persona se presente).
2. **Instanciar la clase en main:** En otra clase con método main (por ejemplo ProgramaPersona), crea un objeto de tipo Persona usando la palabra clave new.
3. **Usar el objeto:** A través del objeto instanciado, asigna valor a sus atributos y llama a sus métodos. Observa cómo el método puede usar los atributos del objeto.

¿Identifica si se creó un método constructor, y que sucede si no se lo crea??

Si, puesto que se creó puesto que en la clase Persona se crearon 2 constructores, de edad y nombre, en caso de no suceder java proporcionara un constructor por defecto.

¿Si no definimos ningún constructor en una clase, ¿Java proporciona un constructor por defecto (sin parámetros) automáticamente, SI O NO??

Si, en el caso de no definir un constructor explícitamente, java lo proporcionara un constructor por defecto sin parámetros.

```
public class Persona {
    static class LaPersona {
        String nombre;
        int edad;

        void presentarse() {
            System.out.println("Hola, mi nombre es " + nombre + " y tengo "
+ edad + " años.");
        }
    }
    public static void main(String[] args) {
        LaPersona personal = new LaPersona();
    }
}
```

```
personal.nombre = "Alan Logroño Soria";
personal.edad = 20;

personal.presentarse();
}
```

```
C:\Users\Usuario\.jdk\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Users\Usuario\AppData\Local\Temp\jvarkit\jvarkit.jar"
Hola, mi nombre es Alan Logroño Soria y tengo 20 años.

Process finished with exit code 0
```

Práctica 2: Introducción a los Constructores (sin y con Parámetros)

Objetivo: Aprender qué es un **método constructor** en Java, para qué sirve y cómo definir dos tipos de constructores: uno *sin parámetros* (constructor por defecto) y otro *con parámetros*. Comprenderás cómo los constructores inicializan los atributos al crear un objeto y cómo se diferencian de los métodos normales en su definición y uso.

Actividad: Define una nueva clase que incluya **constructores**. Por ejemplo, crea una clase Coche con algunos atributos, y proporciona dos constructores:

- Un constructor **sin parámetros** que asigne valores predeterminados a los atributos.
- Un constructor **con parámetros** que reciba valores para inicializar los atributos.

Además, incluye en la clase al menos un método normal (por ejemplo, mostrarInfo) para mostrar el estado del objeto. Luego, en el método main, instancia **dos objetos**: uno usando el constructor sin parámetros y otro usando el constructor con parámetros, y utiliza el método para verificar que los atributos fueron inicializados correctamente en cada caso.

```
public class Coche {
    String marca;
    String modelo;
    int anio;

    public Coche() {
        this.marca = "Toyota";
        this.modelo = "Corolla";
        this.anio = 2020;
    }

    public Coche(String marca, String modelo, int anio) {
        this.marca = marca;
        this.modelo = modelo;
    }
}
```

```

        this.anio = anio;
    }

    public void mostrarInfo() {
        System.out.println("Marca: " + marca);
        System.out.println("Modelo: " + modelo);
        System.out.println("Año: " + anio);
        System.out.println("-----");
    }

    public static void main(String[] args) {
        Coche coche1 = new Coche();
        System.out.println("Coche 1:");
        coche1.mostrarInfo();

        Coche coche2 = new Coche("Honda", "Civic", 2023);
        System.out.println("Coche 2:");
        coche2.mostrarInfo();
    }
}

```

```

C:\Users\Usuario\.jdk\openjdk-24.0.1\bin\java.exe "
Coche 1:
Marca: Toyota
Modelo: Corolla
Año: 2020
-----
Coche 2:
Marca: Honda
Modelo: Civic
Año: 2023
-----

Process finished with exit code 0

```

Práctica 3: Usando Múltiples Objetos y Reutilización de Métodos (Ejemplo Mejorado)

Objetivo:

Comprobar cómo una misma **clase** puede servir de plantilla para **múltiples objetos** y cómo los **métodos definidos** en la clase se reutilizan en cada objeto.

Aquí afianzarás la idea de que cada objeto tiene **sus propios valores** en los atributos, pero **comparte la estructura y comportamientos** de la clase.

Actividad:

- Crea una clase que modele algo sencillo.
- Instancia **tres o más objetos** de esa clase y usa sus métodos.
- Por ejemplo, modela una **Pelota**:
 - Atributos: color y tamaño.
 - Método: `mostrarDetalles()`, que imprime el color y tamaño de la pelota.
- En el método `main`, crea al menos tres pelotas con diferentes características.
- Llama al método de cada pelota para **mostrar sus detalles**.

Observa:

El **mismo método** se ejecuta en cada objeto, pero usando **los valores propios** de cada instancia.

```
public class Pelota {
    String color;
    String tamaño;

    public Pelota(String color, String tamaño) {
        this.color = color;
        this.tamaño = tamaño;
    }

    public void mostrarDetalles() {
        System.out.println("Pelota de color " + color + " y tamaño " +
tamaño);
    }

    public static void main(String[] args) {
        Pelota pelota1 = new Pelota("rojo", "grande");
        Pelota pelota2 = new Pelota("azul", "mediano");
        Pelota pelota3 = new Pelota("verde", "pequeño");

        pelota1.mostrarDetalles();
        pelota2.mostrarDetalles();
        pelota3.mostrarDetalles();
    }
}
```



```
C:\Users\Usuario\.jdk\openjdk-24.0.1\bin\  
Pelota de color rojo y tamaño grande  
Pelota de color azul y tamaño mediano  
Pelota de color verde y tamaño pequeño  
  
Process finished with exit code 0
```

Práctica 4: Consolidando Conceptos – Constructores vs. Métodos Normales

Objetivo: Diferenciar claramente entre **métodos constructores** (con o sin parámetros) y **métodos normales** de una clase, consolidando todo el flujo POO aprendido. Tras esta práctica, sabrás identificar qué hace cada tipo de método, cuándo se ejecuta cada uno y cómo utilizarlos correctamente al crear y manipular objetos.

Actividad: Diseña una clase que integre todo lo aprendido. Por ejemplo, crea una clase Libro con varios atributos (titulo, autor, paginas). Provee **dos constructores**: uno **sin parámetros** (que inicialice los atributos con valores por defecto, p.ej. "Desconocido") y otro **con parámetros** (que reciba valores para cada atributo). Agrega también al menos un **método normal** – por ejemplo, `imprimirInfo()` que muestre los datos del libro formateados. En el método main, instancia objetos de tipo Libro usando ambos constructores y prueba el método normal. Esto te permitirá comparar:

¿cómo se usan y cuándo se ejecutan el constructor vs. el método normal?

El constructor: El constructor se ejecuta cuando creas un objeto con `new`. Se utiliza para inicializar los atributos del objeto en el momento en que el objeto es creado.

El método normal `imprimirInfo()`: Este método se llama una vez que el objeto ha sido creado e inicializado, y permite interactuar con los atributos del objeto.

```
public class Libro {  
    String titulo;  
    String autor;  
    int paginas;  
  
    public Libro() {  
        this.titulo = "Desconocido";  
        this.autor = "Desconocido";  
        this.paginas = 0;  
    }  
  
    public Libro(String titulo, String autor, int paginas) {  
        this.titulo = titulo;  
        this.autor = autor;  
        this.paginas = paginas;  
    }  
}
```

```

public void imprimirInfo() {
    System.out.println("Título: " + titulo);
    System.out.println("Autor: " + autor);
    System.out.println("Páginas: " + paginas);
    System.out.println("-----");
}

public static void main(String[] args) {
    Libro libro1 = new Libro();
    System.out.println("Libro 1:");
    libro1.imprimirInfo();

    Libro libro2 = new Libro("Cien Años de Soledad", "Gabriel García
Márquez", 417);
    System.out.println("Libro 2:");
    libro2.imprimirInfo();
}
}

```

```

C:\Users\Usuario\.jdk\openjdk-24.0.1\bin\java.exe "-ja
Libro 1:
Título: Desconocido
Autor: Desconocido
Páginas: 0
-----
Libro 2:
Título: Cien Años de Soledad
Autor: Gabriel García Márquez
Páginas: 417
-----

Process finished with exit code 0

```

GitHub:

https://github.com/Alan-Logrono-S/Tarea-S2_1---Logro-o.git

Subir la práctica de manera individual al git, tanto los programas como la investigación, enviar el enlace
TAREA2_Apellido_Nombre