

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

PROGRAMACION ORIENTADA A OBJETOS

ASIGNATURA:	POO
PROFESOR:	Ing. Yadira Franco R
PERÍODO ACADÉMICO:	2025-A

TAREA S2_1

TÍTULO:

CLASES , OBJETOS, ATRIBUTOS Y METODOS

INGRESO DE DATOS

María Girón



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 el plano o molde que define las características y comportamientos de un objeto. Se crea con la palabra clave class.

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

- Cómo se crea un constructor con y sin parámetros: Un **constructor** es un método especial que inicializa un objeto cuando se crea. Se llama automáticamente.

Constructor sin parámetros:

```
public Persona() {  
    nombre = "Desconocido";  
    edad = 0;  
}
```

Constructor con parámetros:

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

La diferencia entre ambos es que el constructor **sin parámetros** asigna valores por defecto, mientras que el constructor **con parámetros** permite personalizar los atributos al crear un objeto.

- Como se crean métodos: Los métodos permiten realizar acciones dentro de la clase.

Se crea de la siguiente forma:

```
public void mostrarInfo() {  
    System.out.println("Nombre: " + nombre + ", Edad: " + edad);  
}
```

Este método muestra los datos de la persona.

- Cómo se instancia un objeto y donde se lo ubica: Para usar la clase, se debe **instanciar** un objeto. Esto se hace en el método main o en otra clase diferente.

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        // Creando objetos de Persona
```

```
        Persona persona1 = new Persona(); // Usa el constructor sin parámetros
```

```
        Persona persona2 = new Persona("Maria", 25); // Usa el constructor con parámetros
```

```
        // Llamando a un método
```

```
        persona1.mostrarInfo();
```

```
persona2.mostrarInfo();  
}  
}
```

- Cómo se usan los atributos y métodos: Los atributos pueden modificarse y los métodos ejecutarse dependiendo de los requerimientos del programa.

```
personal.nombre = "Carlos";  
personal.edad = 30;  
personal.mostrarInfo(); // Ahora mostrará los nuevos valores
```

- ¿Cómo se reutiliza los métodos si tengo tres o más objetos?

Si se tiene varios objetos, se pueden reutilizar los métodos con cada uno de los objetos que se hayan creado:

```
Persona persona3 = new Persona("Luis", 40);  
persona3.mostrarInfo();  
  
Persona persona4 = new Persona("Ana", 35);  
persona4.mostrarInfo();
```

Cada objeto tiene su propia copia de los atributos, pero todos pueden usar los métodos definidos de la clase.

- Diferencia entre métodos constructor con parámetros y sin parámetros

CONSTRUCTOR	CARACTERÍSTICAS DEL CONSTRUCTOR
Sin parámetros	Asigna valores predeterminados al objeto.
Con parámetros	Permite definir valores personalizados al crear el objeto.

```
Persona p1 = new Persona(); // Tendrá valores por defecto
Persona p2 = new Persona("Eva", 22); // Tendrá valores personalizados
```

- Diferencia entre métodos

TIPO DE MÉTODO	CARACTERÍSTICAS
Sin retorno (void)	Realiza una acción, pero no devuelve datos.
Con retorno (int, String, etc.)	Devuelve un valor tras ejecutarse.
Con parámetros	Recibe información previa para procesarla.
Sin parámetros	No requiere información externa.

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:

- 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).
- 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.

- ¿Identifica si se creó un método constructor, y que sucede si no se lo crea??, Si no definimos ningún constructor en una clase, Java proporciona un *constructor por defecto* (sin parámetros) automáticamente, SI O NO??

```
C:\Users\XCORP\.jdk\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Users\XCORP\AppData\Local\Programs\IntelliJ IDEA\bin\idea_rt.jar=5000:C:\Users\XCORP\AppData\Local\Programs\IntelliJ IDEA\bin" -Dfile.encoding=UTF-8
Hola mi nombre es: Maria Girón
Mi profesión es: Estudiante Universitario
Mi edad es: 21
Mi hobby es: Jugar al tenis
-----
```

```
Hola mi nombre es: null
Mi profesión es: null
Mi edad es: 0
```

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.

```
C:\Users\XCORP\.jdk\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Users\XCORP\AppData\Local\Temp\jvarkit\jvarkit.jar"
Marca del auto: CHEVROLET
Color: BLANCO
Año de fabricación: 0
Precio: 24999.99
Marca del auto: null
Color: null
Año de fabricación: 0
Precio: 0.0

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.

```
C:\Users\XCORP\.jdk\openjdk-24.0.1\bin
Color: Rojo, Tamaño: Grande
Color: Azul, Tamaño: Mediano
Color: Verde, 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 (título, 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?

```
C:\Users\XCORP\.jdk\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Users\
Título: Desconocido
Autor: Desconocido
Páginas: 0
-----
Título: 1984
Autor: George Orwell
Páginas: 328
-----
Título: Cien años de soledad
Autor: Gabriel García Márquez
Páginas: 471

Process finished with exit code 0
```

Aquí se crea la clase Libro con los atributos título, autor y páginas, integrando dos constructores y un método normal para diferenciar sus usos.

El constructor sin parámetros inicializa los atributos con valores predeterminados ("Desconocido"), mientras que el constructor con parámetros permite asignar valores personalizados al momento de instanciar un objeto. Además, se define el método `imprimirInfo()`, que imprime los datos del libro, demostrando que aunque cada instancia tiene atributos únicos, comparte la estructura y comportamiento de la clase.

En `main()`, se crean objetos con ambos constructores y se llama al método `imprimirInfo()` para visualizar cómo los valores individuales son manejados dentro de un mismo esquema de clase principal.

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