

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

Nayely Del Rocio Ayol Guanoluisa



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.



Definir clase

- Crear un nuevo proyecto en java.
- Sobre el archivo Main.java
- Nombralo en base a la clase que vas a desarrollar.



Definir variables de instancia

- Se agregan las variables que perteneceran a cada objeto creado.
- Se definen dentro de la clase.



Definir métodos

- Los métodos servirán para poder trabajar con las variables.
- Estos sirven para realizar procesos específicos y mostrar resultados.
- Se usan verbos en infinitivo.



Usar la clase

- Esta clase puede ser usada en diferentes partes del programa.
- Debes crear un objeto para la clase.
- Luego el objeto sera usado para acceder a las variables y usar los métodos de clase.

Cómo se crea un constructor con y sin parámetros

Sin parámetros

Declarar la clase



- ☐ Primero se declara la clase que se va a usar.
- ☐ De preferencia usar un nombre que este relacionado con lo que se está desarrollando.

Declarar un constructor



- ☐ Este constructor es implementado sin parámetros.
- ☐ No recibe ningún valor cuando se crea un objeto.

Inicializar los atributos



- ☐ Recomendable darle a los parámetros valores predeterminados.
- ☐ De esta forma en caso de que se cree un objeto sin información, los valores predeterminados aparecerán.

Crear el objeto



- ☐ Se llama al constructor y no se le proporciona ninguna información de los parámetros.
- ☐ El objeto se inicializa con la información que fue implementada en el constructor.

Con parámetros

Declarar la clase



- ☐ Se declara la clase que se va a usar.
- ☐ El nombre debe estar relacionado con lo que se este desarrollando.

Declarar un constructor



- ☐ En este constructor se incluyen parámetros.
- ☐ Estos parámetros serán asignados al constructor cuando se realiza un objeto.

Inicializar los atributos



- ☐ Dentro del constructor se usan los parámetros para inicializar los atributos.
- ☐ De esta forma los objetos que se creen se quedan con los valores que se designen al constructor.

Crear el objeto



- ☐ En este caso el objeto ya tiene los valores que corresponden a los parámetros.
- ☐ De esta forma el objeto usa los atributos que se le han asignado de forma específica.

Como se crean métodos



Cómo se instancia un objeto y donde se lo ubica.

Crear una clase

- La clase es necesaria para instanciar un objeto.

Método Main

- Aquí se crean las instancias de los objetos.

Instancia del objeto

- Se usa 'new' para crear una nueva instancia de clase y asignarla a una variable.

Ubicación

- El objeto se guarda en la memoria.

Uso del objeto

- La forma en la que se puede usar el objeto es combinándolo con los métodos que se hayan creado.

Cómo se usan los atributos y métodos.

Atributos

Declaración

- Son declarados dentro de la clase creada.

Inicialización

- Se inicializan con el constructor de la clase.

Acceso

- Se acceden a estos a través de los métodos.

Métodos

- Los métodos se definen en la clase.

Uso

- Es necesario un objeto para poder llamarlo.

Los métodos pueden ser usados para realizar cálculos y devolver el resultado.

Como se reutiliza los métodos si tengo tres o más objetos

Reutilización	Objetos	Uso
<ul style="list-style-type: none">• Los métodos pueden reutilizarse gracias a que, tanto estos como los objetos pertenecen a la misma clase.	<ul style="list-style-type: none">• Los objetos pueden fácilmente usar los métodos de la clase, solo hace falta que los invoquen.	<ul style="list-style-type: none">• Siempre y cuando el método exista solo es necesario crear un objeto que pueda usarlo.

Diferencia entre métodos constructor con parámetros y sin parámetros

MÉTODOS CON PARÁMETROS	MÉTODOS SIN PARÁMETROS
<ul style="list-style-type: none">- Necesitan datos para ejecutarse.	<ul style="list-style-type: none">- No necesitan ningún dato para ejecutarse.
<ul style="list-style-type: none">- Se especifican dentro del paréntesis del método.	<ul style="list-style-type: none">- Los paréntesis suelen estar vacíos.
<ul style="list-style-type: none">- Se pueden moldear las acciones del objeto con datos específicos.	<ul style="list-style-type: none">- Las acciones son fijas, es decir, no dependen de otra información.

CONSTRUCTOR CON PARÁMETROS	CONSTRUCTOR SIN PARÁMETROS
<ul style="list-style-type: none">- Los valores son proporcionados de forma específica.	<ul style="list-style-type: none">- Los valores son predeterminados.
<ul style="list-style-type: none">- Nombre de la clase + ()	<ul style="list-style-type: none">- Nombre de la clase + (Parámetros)
<ul style="list-style-type: none">- Objetos con información.	<ul style="list-style-type: none">- Objetos sin información.

Diferencia entre métodos

ESTÁTICOS	NO ESTÁTICOS
- Se accede a él a través del nombre de la clase.	- Se usa una instancia de clase para acceder a él.
- No dependen del estado de un objeto específico.	- Trabajan con un objeto específico de la clase.
- Como su nombre lo dice, solo pueden acceder a atributos estáticos.	- Pueden acceder a atributos estáticos y no estáticos.

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

- Sí, si el método constructor no es creado, java nos da uno por defecto y sin parámetros. Si bien este no se muestra de forma explícita, podemos comprobar que existe al inicializar un objeto con los valores de los atributos predeterminados. De esta forma se crean objetos sin pasar parámetros al constructor.

PRÁCTICA 1

ProgramaPersona.java

```
public class ProgramaPersona {
    public static void main(String[] args) {
        //Crear una nueva instancia
        Persona persona1=new Persona();

        //Dar valores a los atributos
        persona1.nombre="Liam";
        persona1.nacionalidad="Italia";
        persona1.edad=20;

        //Llamar al metodo para mostrar la informacion
        persona1.presentarse();
    }
}
```

Persona.java

```
public class Persona {
    //definir los atributos
    public String nombre, nacionalidad;
    public int edad;

    //Crear un metodo para mostrar la informacion
    public void presentarse(){
        System.out.println("\n-----!Hola a todos!-----\n");
        System.out.println("Mi nombre es "+nombre+"\n"+"Soy de "+nacionalidad+"\n"+"Y tengo "+edad);
    }
}
```

SALIDA


```
C:\Users\PC\.jdk\openjdk-24

-----!Hola a todos;-----

Mi nombre es Liam
Soy de Italia
Y tengo 20
```

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.

PRACTICA 2

Main.java

```
public class Main {
    public static void main(String[] args) {
        //Instancia del objeto con el constructor sin parametros
        Coche coche1=new Coche();
        //Instancia del objeto con el constructor con parametros
        Coche coche2=new Coche("Chevrolet","Negro",2023,3000.50);

        //Invocacion del metodo
        System.out.println("\n---INFORMACION DE LOS AUTOS---\n");
        coche1.mostrarinfo();
        System.out.println("\n-----\n");
        coche2.mostrarinfo();
    }
}
```

Coche.java

```

public class Coche {
    //Atributos del coche
    public String marca, color;
    public int anio;
    public double precio;

    //Constructor sin parametros
    public Coche(){
        marca="Toyota";
        color="Azul";
        anio=2019;
        precio=2500.50;
    }

    //Constructor con parametros
    public Coche(String marca, String color, int anio, double precio){
        this.marca=marca;
        this.color=color;
        this.anio=anio;
        this.precio=precio;
    }

    //Metodo para mostrar la informacion
    public void mostrarinfo(){
        System.out.println("Marca: "+marca+"\n"+"Color: "+color+"\n"+"Año: "+anio+"\n"+"Precio: "+precio);
    }
}

```

SALIDA

```

---INFORMACION DE LOS AUTOS---

Marca: Toyota
Color: Azul
Año: 2019
Precio: 2500.5

-----

Marca: Chevrolet
Color: Negro
Año: 2023
Precio: 3000.5

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.

PRACTICA 3

Main.java

```
public class Main {
    public static void main(String[] args) {
        //Instanciar objetos
        Pelota pelota1=new Pelota("Grande", "Morado",12,1.50);
        Pelota pelota2=new Pelota("Mediana", "Celeste", 15,1.75);
        Pelota pelota3=new Pelota("Pequeña", "Negra", 5,1.20);

        //Llamar al primer metodo
        System.out.println("\n-----DETALLES DEL PRODUCTO-----\n");
        pelota1.mostrar();
        pelota2.mostrar();
        pelota3.mostrar();

        //Llamar al tercer metodo
        System.out.println("\n-----FACTURA-----\n");
        pelota1.facturar();
        pelota2.facturar();
        pelota3.facturar();
    }
}
```

Pelota.java

```

public class Pelota {
    //Atributos de la clase pelota
    public String tamaño, color;
    public int cantidad;
    public double precio;

    //Constructor con parametros
    public Pelota(String tamaño, String color, int cantidad, double precio){
        this.tamaño=tamaño;
        this.color=color;
        this.cantidad=cantidad;
        this.precio=precio;
    }

    //Primer metodo para mostrar informacion
    public void mostrar(){
        System.out.println("Tamaño: "+tamaño+"\nColor: "+color+"\nCantidad: "+cantidad+"\nPrecio: "+precio+"\n");
    }

    //Segundo metodo para realizar calculos
    public void facturar(){
        System.out.println("Producto: Pelota "+tamaño+"\nColor: "+color+"\nTotal a pagar: "+(cantidad*precio)+"\n");
    }
}

```

SALIDA

-----DETALLES DEL PRODUCTO-----

```

Tamaño: Grande
Color: Morado
Cantidad: 12
Precio: 1.5
|
Tamaño: Mediana
Color: Celeste
Cantidad: 15
Precio: 1.75

Tamaño: Pequeña
Color: Negra
Cantidad: 5
Precio: 1.2

```

-----FACTURA-----

```

Producto: Pelota Grande
Color: Morado
Total a pagar: 18.0

Producto: Pelota Mediana
Color: Celeste
Total a pagar: 26.25

Producto: Pelota Pequeña
Color: Negra
Total a pagar: 6.0

```

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?

PRACTICA 4

Main.java

```
public class Main {
    public static void main(String[] args) {
        //Se crea una instancia con un constructor sin parametros
        Libro libro1=new Libro();
        //Se crea otra instancia con un constructor con parametros
        Libro libro2=new Libro("Donde viven las musas","Mariela de dos Santos",215,27.50);

        //Se llama al primer metodo creado
        System.out.println("\n-----LIBROS EN STOCK-----");
        libro1.mostrarInfo();
        System.out.println("-----");
        libro2.mostrarInfo();

        //Se llama al segundo metodo creado
        System.out.println("-----FACTURA-----");
        libro1.facturar();
        System.out.println("-----");
        libro2.facturar();
    }
}
```

Libro.java

```

public class Libro {
    //Se da lugar a los atributos
    public String titulo, autor;
    public int num_paginas;
    public double precio;

    //Se crea un constructor sin parametros
    public Libro(){
        titulo="Desconocido";
        autor="Anonino";
        num_paginas=0;
        precio=0;
    }

    //Se crea un constructor con parametros
    public Libro(String titulo, String autor, int num_paginas, double precio){
        this.titulo=titulo;
        this.autor=autor;
        this.num_paginas=num_paginas;
        this.precio=precio;
    }

    //Se crea un metodo para mostrar los detalles del libro
    public void mostrarInfo(){
        System.out.println("\nTitulo: "+titulo+"\nAutor: "+autor+"\nPaginas: "+num_paginas+"\nPrecio: "+precio+"\n");
    }

    //Se crea un metodo para realizar una factura
    public void facturar(){
        System.out.println("\nTitulo: "+titulo+"\nIva: "+(precio*0.12)+"\nTotal a pagar: "+(precio+(precio*0.12))+"\n");
    }
}

```

SALIDA

-----LIBROS EN STOCK-----

Titulo: Desconocido
 Autor: Anonino
 Paginas: 0
 Precio: 0.0

Titulo: Donde viven las musas
 Autor: Mariela de dos Santos
 Paginas: 215
 Precio: 27.5

-----FACTURA-----

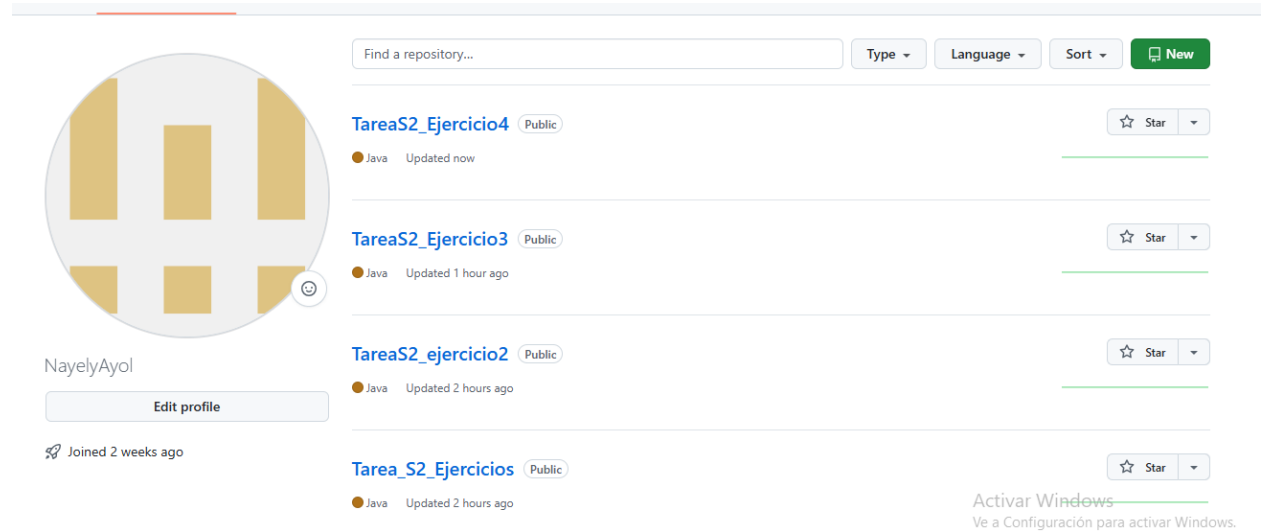
Titulo: Desconocido
 Iva: 0.0
 Total a pagar: 0.0

Titulo: Donde viven las musas
 Iva: 3.3
 Total a pagar: 30.8

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

LINK DE LA TAREA

- Dentro del Ejercicio 4 esta la investigación
- <https://github.com/NayelyAyol>



The screenshot shows the GitHub profile of a user named NayelyAyol. The profile includes a circular avatar with a yellow and grey geometric pattern, the username 'NayelyAyol', an 'Edit profile' button, and a note 'Joined 2 weeks ago'. The repository list on the right contains four entries, all in Java and marked as 'Public':

- TareaS2_Ejercicio4**: Updated now
- TareaS2_Ejercicio3**: Updated 1 hour ago
- TareaS2_ejercicio2**: Updated 2 hours ago
- Tarea_S2_Ejercicios**: Updated 2 hours ago

Each repository entry has a 'Star' button and a dropdown arrow. A search bar at the top of the repository list is labeled 'Find a repository...'. At the bottom right, there is a Windows watermark that reads 'Activar Windows' and 'Ve a Configuración para activar Windows.'

BIBLIOGRAFÍA

- Perez, A., & Perez, A. (2023, 8 octubre). Crear clases en java. Blog. <https://www.lineadecodigo.es/crear-clases-en-java/>
- Rodríguez, A. (s. f.). Cómo crear constructores con o sin parámetros. Inicializar objetos Java. Ejercicio ejemplo resuelto (CU00639B). [aprenderaprogramar.com](https://www.aprenderaprogramar.com). https://www.aprenderaprogramar.es/index.php?option=com_content&view=article&id=496:como-crear-constructores-con-o-sin-parametros-inicializar-objetos-java-ejercicio-ejemplo-resuelto-cu00639b
- Blog. (2023, 27 marzo). Crea métodos en Java: Guía paso a paso. DonComo. <https://doncomo.com/crea-metodos-en-java-guia-paso-a-paso/>
- Berenguer, M. C., & Berenguer, M. C. (2018, 29 marzo). Práctica: instanciar un objeto en Java. JavAutodidacta. <https://javautodidacta.es/instanciar-un-objeto-en-java/>
- Elprofesoremilio. (2025, 27 marzo). Atributos y Métodos en Java: da sentido a tus clases. Modo Debug. <https://mododebug.com/atributos-metodos-java/>
- Rodríguez, A. (s. f.-b). Tipos de métodos y constructores en Java: métodos con parámetros y métodos sin parámetros. (CU00625B). [aprenderaprogramar.com](https://www.aprenderaprogramar.com). https://www.aprenderaprogramar.es/index.php?option=com_content&view=article&id=436:tipos-de-metodos-y-constructores-en-java-metodos-con-parametros-y-metodos-sin-parametros-cu00625b
- La diferencia entre el método estático y el método no estático en Java, y la relación entre la clase de herencia - programador clic. (s. f.). <https://programmerclick.com/article/39302892730/>