

# Bootcamp Java Developer

Fase 1 - Java Analyst  
Módulo 7

# Introducción

# ¿Qué es un paradigma de programación?

Es una forma o estilo de programación de software. Existen diferentes maneras de diseñar un lenguaje de programación, y varios modos de trabajar para obtener los resultados que necesitan los programadores.

Un paradigma de programación es un **conjunto de métodos sistemáticos** aplicables en todos los niveles del diseño de programas para resolver problemas computacionales.



# Programación Orientada a Objetos (POO)

Es una forma especial de programar más cercana a cómo expresaríamos las cosas en la vida real que otros tipos de programación.

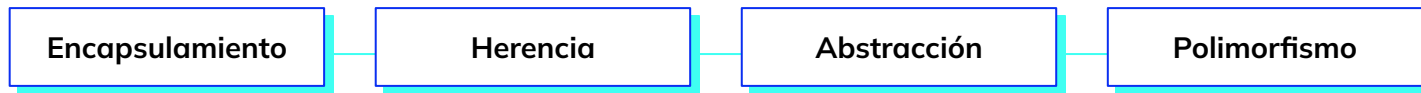
Se basa en la idea natural de un mundo lleno de objetos y la resolución de problemas se realiza mediante el modelo de objetos.

Dicho de una forma más simple, es una **forma de descomponer un problema para solucionarlo de un modo distinto**.



# Pilares de la POO

La programación **Orientada a Objetos** se basa en **4 pilares** fundamentales:



Todo pilar debe tener **bases** que debemos aprender:



# Clase

Los tipos de datos que nos proporcionan los lenguajes de programación no son suficientes para representar algo en la vida real.

**Por ejemplo:** un automóvil. Un automóvil necesita mucho más que una cadena de caracteres, un número entero e incluso hasta un arreglo para que podamos diferenciarlo de otros automóviles.

Las clases ayudan a definir este nuevo “tipo de dato” con una estructura que le daremos para solucionar un problema planteado.

También, se pueden definir como una **plantilla o molde**, en la que existen **dos categorías** de miembros a tener en cuenta:

1. **Atributos (Datos).**
2. **Métodos (Algoritmos).**

En resumen, **una clase es un conjunto de objetos con características similares.**

# Atributos

Para crear una **clase**, se debe pensar en qué es lo que hace único a cada objeto, es decir, cuáles son los **datos que diferencian una cosa de otra**. Si vemos un automóvil, ¿qué lo hace único entre tantos? Podría ser su color, marca, patente u otras características.

Para declarar esos datos, se pueden utilizar los tipos que ya conocemos (*numéricos, caracteres y booleanos*) y también utilizar otros objetos.

**Los atributos son características individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades.**

Suelen ser llamadas como: ***Variable de Instancia***, ***Miembro de Instancia*** o ***Campo de Instancia***.



# Métodos

Luego de definir los atributos, se debe analizar qué pueden hacer estos nuevos “tipos de datos”.

**Por ejemplo:** un automóvil puede encenderse, apagarse y mostrar su información.

A esto llamamos métodos, a un **conjunto de instrucciones definidas dentro de una clase, que realizan una determinada tarea y a las que podemos invocar mediante un nombre.**

Cuando se llama a un método, la ejecución del programa pasa al método, ejecuta las instrucciones que se encuentren en él y continúa a partir del punto donde se produjo el llamado, es decir, sigue su flujo natural.

La idea principal de los métodos es no escribir una misma instrucción varias veces, si la vamos a utilizar más de una vez.

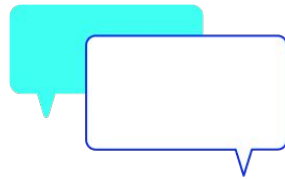


## Mensajes

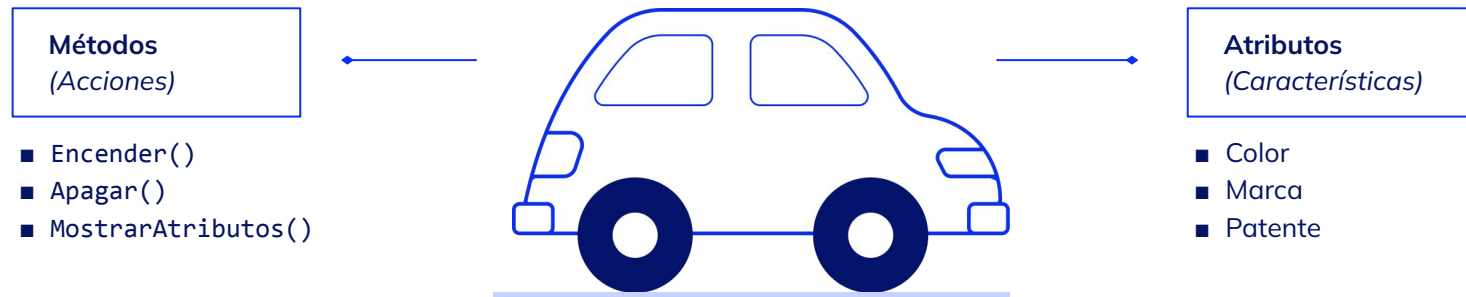
Los objetos interactúan enviándose mensajes unos a otros.

Tras la recepción de un mensaje el objeto actuará. La acción puede ser el envío de otros mensajes, el cambio de su estado, o la ejecución de cualquier otra tarea que se requiera que haga el objeto.

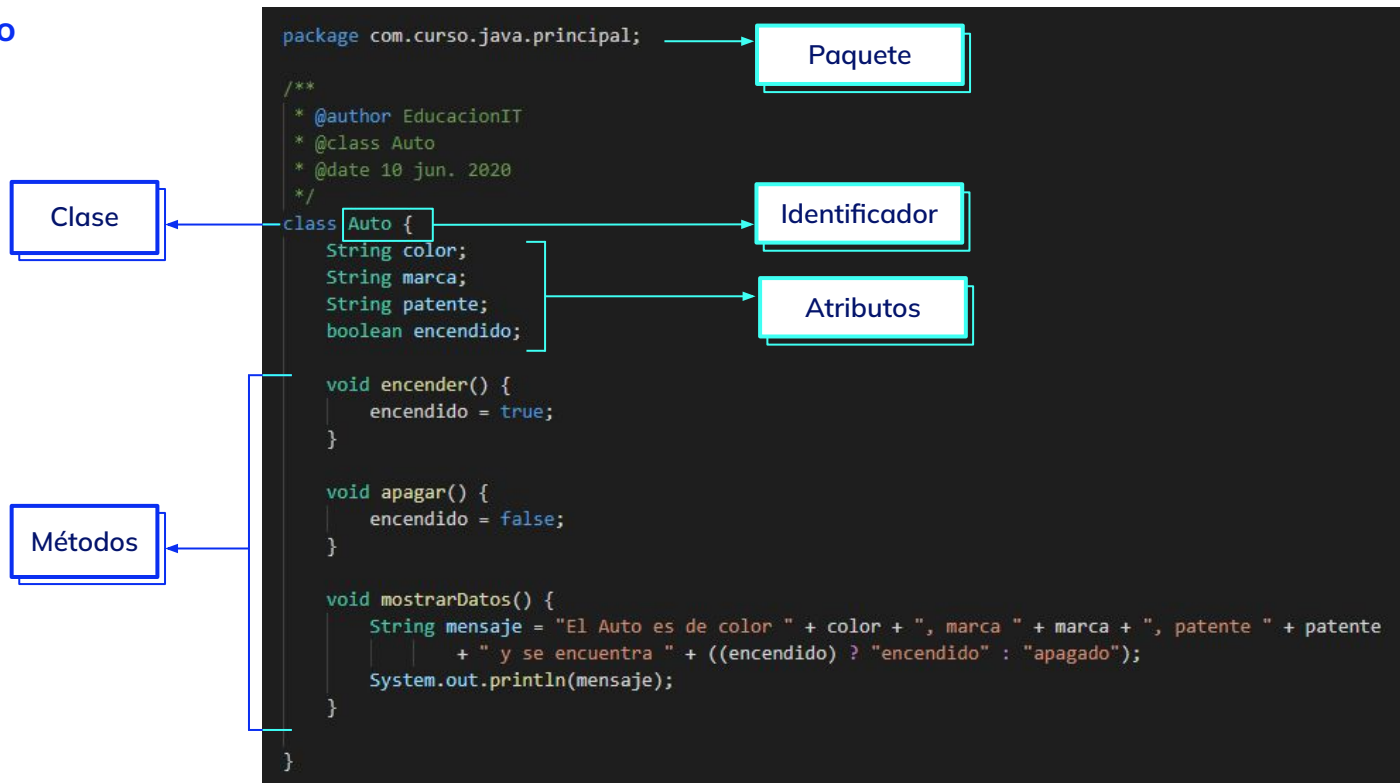
En resumen: **a un objeto se le envía un mensaje cuando una instancia utiliza un método.**



## Ejemplo



## Ejemplo



# Instancia y objeto

Recordemos un poco; al crear una variable o constante debemos definir: el tipo de dato, su identificador y su valor inicial. Con eso logramos poder acceder a ella más adelante y cambiar su valor. Lo mismo ocurre con un objeto, lo primero que debemos hacer es declarar: tipo de dato, identificador y asignarle un espacio de memoria. Esto significa: indicarle al software que debe **guardar bajo un mismo nombre una serie de atributos y comportamientos.**



Pero como ahora el **dato** es más complejo que una simple cadena de caracteres debemos hacerlo de una forma diferente:

- Tipo de dato: clase de donde tomará el molde o plantilla.
- Identificador.
- La sentencia “**new**”.
- Volver a decir el tipo de dato, como en los arreglos.
- Y cerrar con paréntesis “**()**”

```
Auto auto1 = new Auto();
```

Entonces, ahora podemos decir que:

- Una **instancia** es una ocurrencia de la clase. Se logra cuando le decimos a Java que cree un **new Auto()**.
- Al momento de crear un objeto se produce la instanciación.
- Un **objeto** es una **instancia de una clase específica**.

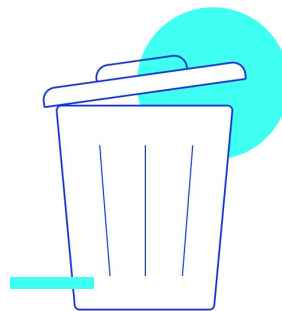


## Ejemplo



# Cómo se destruye un objeto en Java

Quien se encarga de destruir o liberar la memoria de los objetos que ya no se utilizan es el **Garbage Collector (GC)** y lo hace de **forma automática administrando la memoria**.



**¡Sigamos  
trabajando!**