

Variables y métodos de clase

DigitalHouse >
Coding School



**Certified Tech
Developer**
The Ultimate Degree

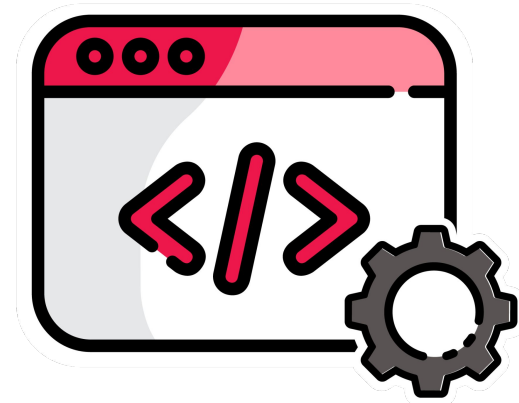
Índice

1. [Las clases y los objetos](#)
2. [Variables de clase](#)
3. [Métodos de clase](#)
4. [Ejemplo de uso](#)

1 | Las clases y los objetos

Las clases

Vimos que el enfoque de la programación orientada a objetos se basa en identificar objetos con sus atributos y responsabilidades. Entonces, encontramos que hay grupos de objetos que aunque tienen diferentes estados —valores de los atributos—, tienen en común cuáles son los atributos y cuáles son sus responsabilidades. Entonces este “*molde*” es lo que llamamos **clases**.



Los objetos



Ingeniero

- nombre: String
- matricula: int
- salario: float

- + construir()
- + cobrar()

Clase Ingeniero

Juan Milutti
Mat. 434244
120.000 \$/mes

Miguel Zapata
Mat. 1133242
134.000 \$/mes

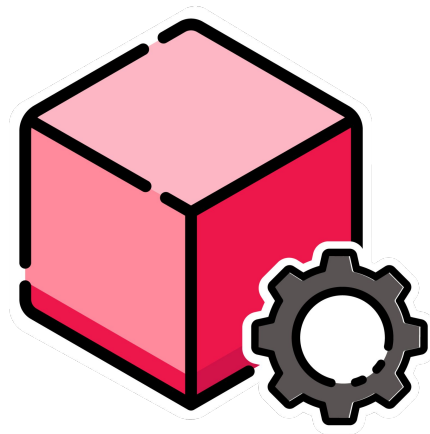
Claudia Vasques
Mat. 775543
140.000 \$/mes

Muriel Fringuesti
Mat. 242352
136.000 \$/mes

Instancias u objetos Ingeniero

¿Qué comparten los objetos?

Todos los objetos de una clase tienen la misma estructura: los mismos atributos y el mismo comportamiento, es decir, pueden hacer lo mismo. Pero cada objeto tiene sus propios atributos, puede tener distintos valores en sus atributos, tiene un estado propio.



2 | Variables de clase

Un valor común para todos los objetos

¿Todos los objetos pueden tener un valor en común?



Todos los diferentes camiones usan el mismo valor del combustible.



Todos los corredores de 100 mts llanos tratan de batir el mismo récord mundial.

Variable de clase

Vamos a llamar **variables de clase** a aquellas variables —atributos— que guardan valores comunes a todos los objetos. El combustible vale lo mismo para cualquier objeto Camion.

| Camión | |
|--------|---------------------------------------|
| - | marca: String |
| - | patente: String |
| - | <u>valorCombustible: double</u> |
| + | Camion (String marca, String patente) |
| + | transportar() |
| + | gastoCombustible(int litros) |

El nombre de variable va subrayado para indicar que es una variable de clase.

3 | Métodos de clase

Métodos de clase

Un método de clase se puede utilizar, sin necesidad de instanciar o crear un objeto, directamente con la clase.

Definimos `cambiarPrecioCombustible(double precio)`, como un método de clase. Para indicar que es un método de clase también debemos subrayarlo.

| Camión | |
|--------|--|
| - | marca: String |
| - | patente: String |
| - | <u>valorCombustible: double</u> |
| <hr/> | |
| + | Camion (String marca, String patente) |
| + | transportar() |
| + | gastoCombustible(int litros) |
| + | <u>cambiarPrecioCombustible(double precio)</u> |

4 | Ejemplo de uso

La clase Camión

```
public class Camion {  
    private String marca;  
    private String patente;  
    static private double valorCombustible;  
  
    public Camion(String marca, String patente){  
        this.marca=marca;  
        this.patente=patente;  
    }  
    public double gastoCombustible(int litros){  
        return litros*Camion.valorCombustible;  
    }  
    static public void cambiarPrecioCombustible(double precio){  
        Camion.valorCombustible=precio;  
    }  
}
```

{ } Código

```
public class Camion {  
    private String marca;  
    private String patente;  
    static private double valorCombustible;  
  
    public Camion(String marca, String patente){  
        this.marca=marca;  
        this.patente=patente;  
    }  
    public double gastoCombustible(int litros){  
        return litros*Camion.valorCombustible;  
    }  
    static public void cambiarPrecioCombustible(double precio){  
        Camion.valorCombustible=precio;  
    }  
}
```

Definimos la variable como static, esto hace que no se pueda usar con un objeto.

{ } Código

```
public class Camion {  
    private String marca;  
    private String patente;  
    static private double valorCombustible;  
  
    public Camion(String marca, String patente){  
        this.marca=marca;  
        this.patente=patente;  
    }  
    public double gastoCombustible(int litros){  
        return litros*Camion.valorCombustible;  
    }  
    static public void cambiarPrecioCombustible(double precio){  
        Camion.valorCombustible=precio;  
    }  
}
```

Definimos el método como static, esto hace que no se pueda usar con un objeto.

`{ }` Código

```
public class Camion {  
    private String marca;  
    private String patente;  
    static private double valorCombustible;  
  
    public Camion(String marca, String patente){  
        this.marca=marca;  
        this.patente=patente;  
    }  
    public double gastoCombustible(int litros){  
        return litros*Camion.valorCombustible;  
    }  
    static public void cambiarPrecioCombustible(double precio){  
        Camion.valorCombustible=precio;  
    }  
}
```

Accedemos a la variable de clase y podemos cambiar su valor.

El Main

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Camion miCamion = new Camion("Ford","AB XXX CD");  
  
        Camion.cambiarPrecioCombustible(98.50);  
  
        Sytem.out.println("Gasto " + miCamion.gastoCombustible(40));  
  
    }  
}
```

{ } Código

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Camion miCamion = new Camion("Ford", "AB XXX CD");  
  
        Camion.cambiarPrecioCombustible(98.50);  
  
        Sytem.out.println("Gasto " + miCamion.gastoCombustible(40));  
  
    }  
}
```

Creamos un objeto de la clase Camion.

`{}` Código

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Camion miCamion = new Camion("Ford", "AB XXX CD");  
  
        Camion.cambiarPrecioCombustible(98.50);  
  
        Sytem.out.println("Gasto " + miCamion.gastoCombustible(40));  
  
    }  
}
```

Utilizamos el método de clase a través de la clase y no del objeto.

DigitalHouse>
Coding School