

Sesión 4: Consultores y Modificadores

CONSULTORES

Los **consultores** son los métodos encargados de **devolver el valor de un atributo** o hacer una operación devolviéndolo modificado. Estos consultores pueden ser de **clase o de instancia**.

Esto se hace para que los atributos sean privados y no se puedan acceder desde cualquier clase, entonces solo dejamos a clases determinadas acceder a esos métodos.

JAVA

En Java se suele denominar con **getAtributo()**. Para los consultores de **clase**, se utiliza **static** en el int.

```
public class Coche {
    private String color;
    private int velocidad;
    private String modelo;
    private static int MAYORIAEDAD = 18; // Atributo de clase

    Coche(String color, int velocidad, String modelo){
        this.color = color;
        this.velocidad = velocidad;
        this.modelo = modelo;
    }
    public String getColor(){
        return color;
    }

    public static int getMayoriaEdad(){
        return MAYORIAEDAD;
    }
}
```

```
Coche coche = new Coche("Rojo",250,"BMW");
System.out.println(coche.getColor()); //Imprime el color
System.out.println(Coche.getMayoriaEdad()); // Imprime la mayoria de edad
(Clase, no del objeto)
```

RUBY

En Ruby existen unos consultores sin necesidad de hacer un método como tal, es **attr_reader**. Pero también se puede crear un método consultor.

```
class Coche
    @@MAYORIA_EDAD = 18 #Atributo de clase
    def initialize (color, velocidad, modelo)
        @color = color
        @velocidad = velocidad
    end
end
```

```

        @modelo = modelo
    end
    attr_reader :color # consultor

    def velocidad # consultor
        @velocidad
    end

    def self.MAYORIA_EDAD # consultor de clase
        @@MAYORIA_EDAD
    end
end
end

```

```

coche = Coche.new("Rojo",250,"BMW")
puts coche.color #Imprime "Rojo"
puts coche.velocidad #Imprime 250
puts Coche.MAYORIA_EDAD #Imprime 18

```

MODIFICADORES

Son métodos que se encargan de **modificar el valor de un atributo**. Pueden ser de **clase o de instancia** y pueden controlar otras restricciones sobre el atributo.

JAVA

En Java se utiliza **setAtributo()**.

```

public class Coche {
    private String color;
    private int velocidad;
    private String modelo;
    private static int MAYORIAEDAD = 18; // Atributo de clase

    Coche(String color, int velocidad, String modelo){ // Constructor
        this.color = color;
        this.velocidad = velocidad;
        this.modelo = modelo;
    }

    public String setColor(String color){ //modificador
        this.color = color;
    }

    public static void setMayoriaEdad(int mayor){ //modificador de clase
        MAYORIAEDAD = mayor;
    }
}

```

```

Coche coche = new Coche ("Rojo",250,"BMW");
coche.setColor("Rojo"); //Pongo el color del coche a rojo
Coche.setMayoriaEdad(21); // Cambio la mayoria de edad sobre la clase

```

RUBY

En Ruby se puede utilizar los modificadores en forma de método o sin necesidad de ellos.

El **modificador de clase** se utiliza con **self**.

```
class Coche
  @@MAYORIA_EDAD = 18 #Atributo de clase
  def initialize (color, velocidad, modelo)
    @color = color
    @velocidad = velocidad
    @modelo = modelo
  end
  attr_accessor :color #consultor y modificador
  attr_reader :velocidad #modificador

  def velocidad=veloc # modificador
    @velocidad = veloc
  end
  def self.MAYORIA_EDAD=e #modificador de clase
    @@MAYORIA_EDAD = e
  end
end
```

```
coche = Coche.new("Rojo",250,"BMW")
puts coche.velocidad #Imprime 250
coche.velocidad = 260 # Le asigna a velocidad el valor 260
puts coche.velocidad #Imprime 260
coche.color = "Verde" #Cambia color a verde
puts coche.color #Imprime verde
Coche.MAYORIA_EDAD = 21 #Cambia la mayoria de edad (cuidado que estoy tocando la
clase, no el objeto)
```