

Comparar objetos

DigitalHouse >
Coding School



**Certified Tech
Developer**
The Ultimate Degree

Índice

1. [Comparar objetos](#)
2. [Interface comparable](#)
3. [Su implementación en Java](#)

1

Comparar objetos



Vamos a aprender una buena práctica
de **cómo comparar objetos**.



Comparar objetos

A la hora de comparar tipos primitivos lo hacemos con los operadores "=", ">", "<", ">=", "<=", "!", "!=", pero, ¿cómo hacemos si queremos comparar dos objetos? Por ejemplo, dos pimientos:

```
int i = 5;  
int j = 6;  
If(i < j)  
    System.out.println("i < j");  
else  
    System.out.println("j >= i");
```



Para poder **comparar dos objetos** lo primero que tendremos que saber es por cuál o cuáles de sus **atributos los vamos a comparar**. Es decir, cómo responderíamos a la pregunta: ¿estos pimientos son iguales?



La primer duda que nos surgirá es si debemos considerar el **color, tipo, peso** o el **tamaño** y sobrescribir el **método equals()**.

Pero qué sucede ahora si lo que queremos saber es ¿cuál de los pimientos es **mayor** que el otro?



El método **equals()** solo nos sirve para comparar igualdad, **pero no si es mayor o menor a otro objeto.**

Método compararCon

Una solución a la problemática planteada es lograr que todos los objetos que necesite comparar tengan por ejemplo un método **compararCon** que reciban como parámetro al otro objeto con el que se desea hacer la comparación y nos devuelva, por ejemplo:

Cero: si son iguales.

Mayor a cero: si el objeto que invoca el método es mayor al recibido como parámetro.

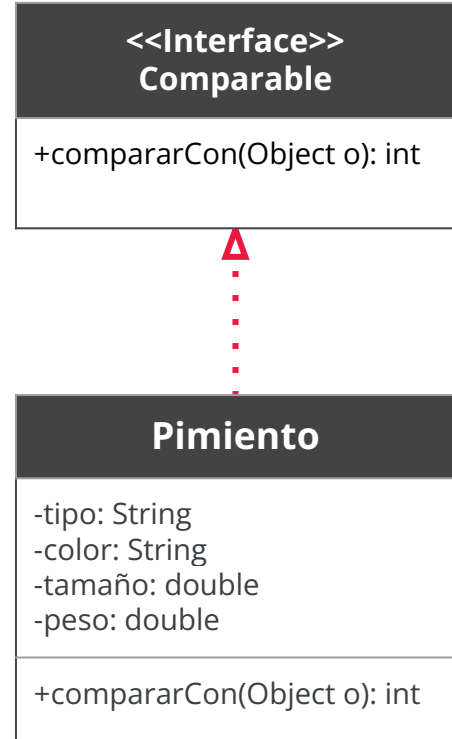
Menor a cero: si el objeto que invoca el método es menor al recibido como parámetro.



¿Cómo hacemos para **obligar** a todos los objetos que queremos comparar para que tengan un método `compararCon`?



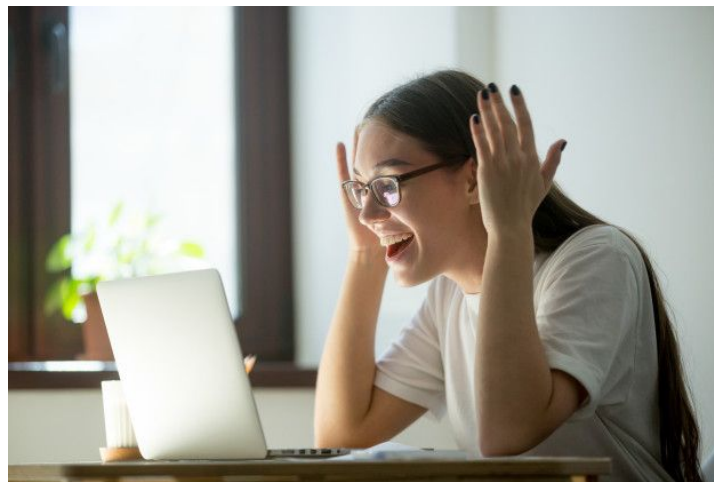
Con la **interfaces** podemos hacer que quien la implemente posea sí o sí un método **`compararCon`** y pueda establecer su propia implementación.



2 | Interface Comparable

Interface Comparable

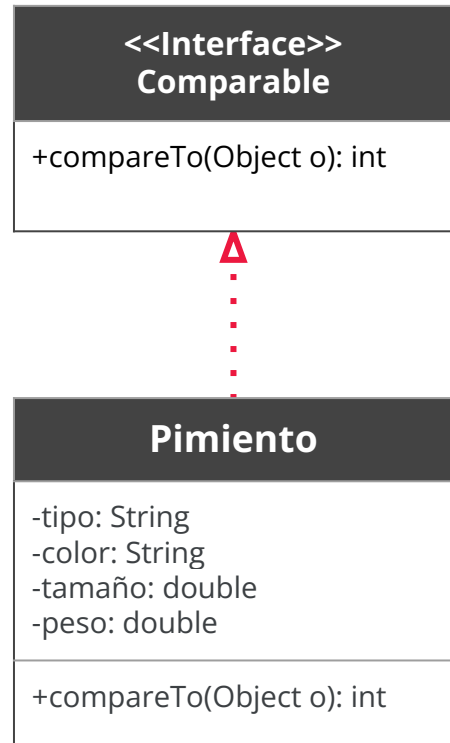
No necesitamos crear una interface para comparar objetos porque Java tiene la suya, es la interface **Comparable** y es necesaria utilizarla en otras circunstancias para comparar objetos, por ejemplo, para ordenarlos en las colecciones.



El método que obliga a implementar la interface Comparable de Java es el método **compareTo**.



Para utilizar la interface Comparable de Java debemos importar el paquete java.lang.



2 | Su implementación en Java

Su implementación en Java

```
import java.lang.*;

public class Pimiento implements Comparable{

    private String tipo;
    private String color;
    private double tamano;
    private double peso;

    public Pimiento(){
    }

    public int compareTo(Object obj){

        Pimiento p2 = (Pimiento) obj;
        int respuesta = 0;

        if(this.getPeso() > p2.getPeso())
            respuesta = 1;

        if(this.getPeso() < p2.getPeso())
            respuesta = -1;

        return respuesta;

    }
```

```
    public void setTipo(String tipo){
        this.tipo = tipo;
    }

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

    public void setTamano(double tamano){
        this.tamano = tamano;
    }

    public void setPeso(double peso){
        this.peso = peso;
    }

    public String getTipo(){
        return tipo;
    }

    public String getColor(){
        return color;
    }

    public double getTamano(){
        return tamano;
    }

    public double getPeso(){
        return peso;
    }

}
```

```
import java.lang.*;

public class Pimiento implements Comparable{

    private String tipo;
    private String color;
    private double tamano;
    private double peso;

    public Pimiento(){
    }

    public int compareTo(Object obj){

        Pimiento p2 = (Pimiento) obj;
        int respuesta = 0;

        if(this.getPeso() > p2.getPeso())
            respuesta = 1;

        if(this.getPeso() < p2.getPeso())
            respuesta = -1;

        return respuesta;
    }
}
```



El método **compareTo** debe devolver:
Si son iguales: 0.
Si es mayor: un número mayor a cero.
Si es menor: un número menor a cero.

```
public class Prueba{

    public void main(String args[]){

        Pimiento p1 = new Pimiento();
        p1.setPeso(200);
        p1.setColor("amarillo");
        Pimiento p2 = new Pimiento();
        p2.setColor("rojo");
        p2.setPeso(150);

        if(p1.compareTo(p2) > 0){
            System.out.println("Pimiento amarillo es mayor al rojo");
        }else if(p1.compareTo(p2) < 0){
            System.out.println("Pimiento rojo es mayor al amarillo");
        }else{
            System.out.println("Pimiento rojo es igual al amarillo");
        }
    }
}
```




Finalmente, por nuestra implementación, podemos decir que el pimiento amarillo es mayor al rojo.

DigitalHouse>
Coding School