Igualdad y ordenamiento en las colecciones

DigitalHouse >
Coding School

Certified Tech Developer
The Ultimate Degree

Índice

A continuación, descubriremos qué necesitan las colecciones para determinar la igualdad y el orden de sus elementos.

1 Elementos iguales

Elementos iguales

Las colecciones **Set** (HashSet, LinkedHashSet, TreeSet) no aceptan valores repetidos o sea iguales ni nulos, lo mismo sucede con las Key de las **Map**. Pero, ¿cómo hacemos, por ejemplo, en objetos de una clase Persona para determinar si una persona en la colección es igual a otra?

Persona
-nombre: String -apellido: String -edad: int
+getNombreCompleto(): String +esMayorEdad(): boolean

Elementos iguales

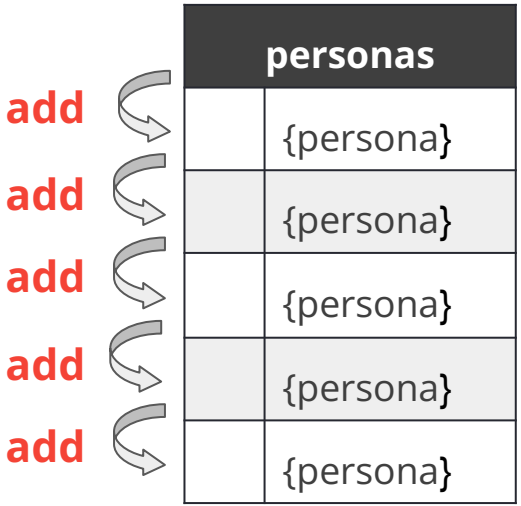
Debemos establecer el criterio por el cual una persona es igual a otra, si ese criterio no se cumple, diremos que son distintas.

En Java para determinar si dos objetos son iguales se deben sobreescribir los métodos **equals()** y **hashCode()**.

Persona
-nombre: String -apellido: String -edad: int
+getNombreCompleto(): String +esMayorEdad(): boolean +equals(o: Object): boolean +hashCode(): int

Elementos iguales

De esta manera, al sobreescribir los métodos **equals** y **hashCode**, las colecciones pueden determinar si el elemento que almacenan es igual a otro. En el caso de las Set servirán para no permitir su inserción.



Persona
-nombre: String -apellido: String -edad: int
+getNombreCompleto(): String +esMayorEdad(): boolean **+equals(o: Object): boolean** **+hashCode(): int**

2 Orden entre elementos

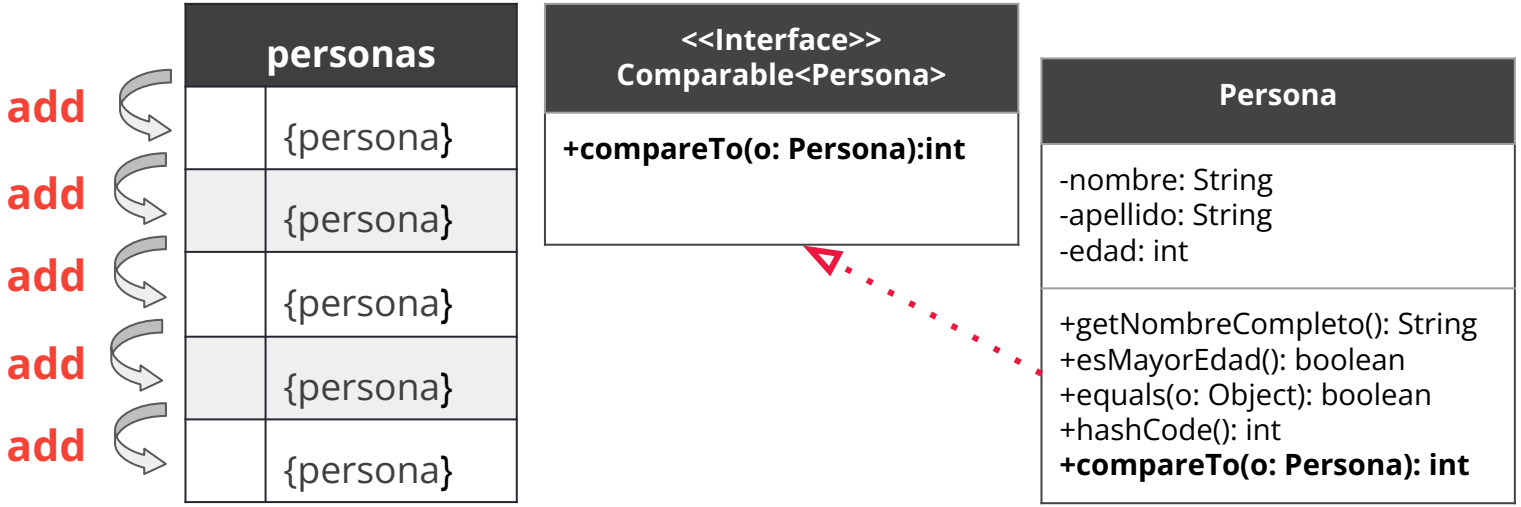
Orden entre elementos

En el caso de las colecciones **TreeSet** y **TreeMap** los elementos se almacenan en forma ordenada. No nos alcanza con determinar la igualdad, en este caso, además debemos **compararlos**, evaluando cuál es mayor, menor o igual a otro.

En Java para comparar objetos en las colecciones debemos implementar la **interface Comparable.**

Orden entre elementos

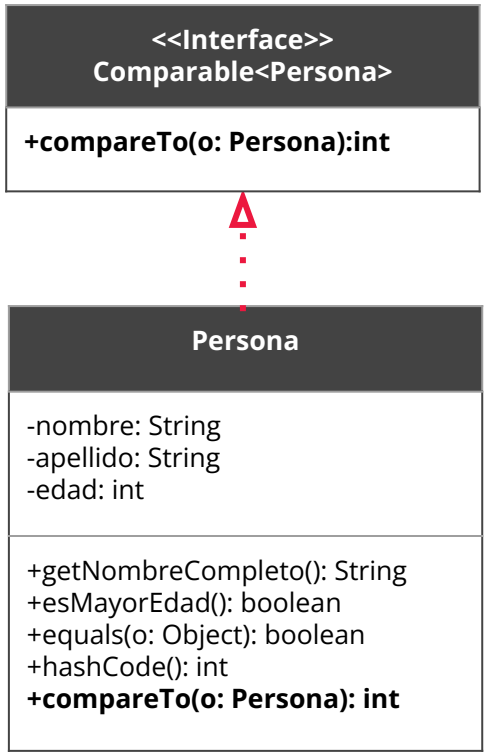
En el caso de las colecciones **TreeSet** y **TreeMap** los elementos se almacenan en forma ordenada y para ello debemos implementar la interface de Java **Comparable**.



Orden entre elementos

Podemos utilizar la interface Comparable para ordenar los elementos de una **List** (ArrayList o LinkedList) invocando su método **sort()**.

```
personas.sort(null);
```




DigitalHouse>
Coding School