

## Hoja de ejercicios 80\_81:

- La colección `TreeSet<T>`.
- Métodos `add(Object)`, `iterator.hasNext()`, `iterator.next()`, `clear()`
- Sobreescritura del método `compareTo(Object)`

80. Escribe un programa que utilizando una colección `TreeSet` lea desde teclado una serie de números enteros:

- El total de números es indefinido y serán leídos en una única línea y separados por espacios.
- Se podrán introducir números repetidos pero no serán almacenados, aunque sí se mostrará un aviso por cada repetido.
- Ante cualquier error en la lectura de un número, se procesará el número siguiente.
- Finalmente se mostrarán los números ordenados ascendentemente y sin repetir.
- Ejemplo de funcionamiento:

Introduce números:

```
10 5 X 99 0 5 40 12 7 340 87 33 2888 9
```

El número 5 está repetido.

Los números ordenados ascendentemente y sin repetidos son:

```
0 5 7 9 10 12 33 40 87 99 340 2888
```

81. Escribe un programa que utilizando una colección `TreeSet` permita gestionar objetos de la clase Jugador:

- La clase Jugador estará definida de la siguiente manera:

Atributos:

```
private String nombre;
```

```
private int estatura;
```

Constructor:

```
public Jugador(String nombre, int estatura)
```

Métodos:

```
public String getNombre()
```

```
public int getEstatura()
```

- El programa principal ofrecerá el siguiente menú de opciones:

Opciones:

```
1.-Introducir jugador
```

```
2.-Listar jugadores
```

```
3.-Eliminar jugador
```

```
4.-Borrar todos
```

```
5.-Salir
```

- El número de jugadores es indeterminado.
- Los nombres no se podrán repetir y serán insensibles a mayúsculas/minúsculas.
- La lista estará en todo momento ordenada alfabéticamente por el nombre.
- Para eliminar un jugador se pedirá el nombre del mismo.