

Ejercicio U7_B6B_E1:

```
public int compareTo(Articulo a){
    if(this.cantidad<a.cantidad)
        return -1;
    else if(this.cantidad==a.cantidad)
        return 0;
    else
        return 1;
}
```

Ejercicio U7_B6B_E2:

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

class ComparadorTelefonos implements Comparator<String>{

    @Override
    public int compare(String s1, String s2) {
        if (s1.equals(s2)) return 0;
        if (s1.startsWith("+") && !s2.startsWith("+")) return -1;
        if (!s1.startsWith("+") && s2.startsWith("+")) return 1;

        //si o1 y o2 empiezan ambos por + o ninguno de ellos empieza por + se comparan numéricamente
        //el + no le estorba a parseLong() por que lo entiende como indicación de número positivo
        long l1=Long.parseLong(s1);
        long l2=Long.parseLong(s2);
        if (l2<l1)
            return -1;
        else if (l1>l2)
            return 1;
        else
            return 0;
    }
}

public class App {
    public static void main(String[] args) {
        ArrayList<String> telefonos = new ArrayList<>();
        telefonos.add("981555555");
        telefonos.add("+34981565656");
        telefonos.add("666666666");
        telefonos.add("+34666666666");

        Collections.sort(telefonos, new ComparadorTelefonos());
        for(String tlf: telefonos)
            System.out.print(tlf+ " ");
    }
}
```

También es posible en este caso trabajar todo con Strings de forma que decido primero por la longitud del string y si son igual de largos comparo alfabéticamente.

```
public int compare(String s1, String s2) {
```

```

if (s1.equals(s2)) return 0;
if (s1.startsWith("+") && !s2.startsWith("+")) return -1;
if (!s1.startsWith("+") && s2.startsWith("+")) return 1;

if(s1.length()>s2.length())
    return 1; //si s1 es más largo es mayor
else if (s1.length()<s2.length())
    return -1;
//si son igual de largos hago comparación alfabética
else if (s1.compareTo(s2)<0)
    return 1;
else if (s1.compareTo(s2)>0)
    return -1;
else
    return 0;
}

```

Ejercicio U7_B6B_E3:

```

import java.util.PriorityQueue;

class Enfermo implements Comparable<Enfermo>{
    String nombre;
    int gravedad;//de 1 a 3. 3 más que grave que 2. 2 más que 1.

    public Enfermo(String nombre, int gravedad) {
        this.nombre = nombre;
        this.gravedad = gravedad;
    }

    @Override
    public String toString() {
        return "Enfermo{" + "nombre=" + nombre + ", gravedad=" + gravedad + '}';
    }

    @Override
    public int compareTo(Enfermo t) {
        return this.gravedad>t.gravedad?-1:1;
    }
}

public class App {

    public static void main(String args[]) {
        PriorityQueue<Enfermo> enfermos= new PriorityQueue<>();
        enfermos.add(new Enfermo("Cristina",2));
        enfermos.add(new Enfermo("Juan",1));
        enfermos.add(new Enfermo("Ana",3));
        enfermos.add(new Enfermo("Oscar",3));
        System.out.println("");
        Enfermo e=enfermos.poll();
        while(e!=null){
            System.out.println(e);
            e=enfermos.poll();
        }

    }

}

```

