

```
1 package Puntuacion;
2
3 import java.util.ArrayList;
4
5
6
7
8
9 public class Puntos {
10     public static ArrayList<Ciclista> listaCicPuntos;// = new ArrayList<Ciclista>();
11     public static boolean categoriaMeta = false;
12     public static boolean categoria1 = false;
13     public static boolean categoria2 = false;
14     public static boolean categoria3 = false;
15
16     public static void CrearlistaConPuntos() {
17         listaCicPuntos = new ArrayList<Ciclista>();
18         for (Ciclista i : ListaCiclistas.listaCiclistas) {
19             listaCicPuntos.add(new Ciclista(i.getDorsal(), 0,0));
20         }
21     }
22
23     public static void añadirPuntos(ArrayList<Ciclista> lista ,Boolean
    categoriaMeta,Boolean categoria3,Boolean categoria2,Boolean categoria1) {//este metodo
    sirve para calcular los puntos en funcion de que categoria es o si es meta
24         for (Ciclista i : listaCicPuntos) {
25             if (categoriaMeta) {
26                 if (i.getDorsal().toString().equals(lista.get(0).getDorsal().toString())) {
27                     i.setPuntosM(5);
28                 }
29                 if (i.getDorsal().toString().equals(lista.get(1).getDorsal().toString())) {
30                     i.setPuntosM(3);
31                 }
32                 if (i.getDorsal().toString().equals(lista.get(2).getDorsal().toString())) {
33                     i.setPuntosM(1);
34                 }
35             }
36
37             if (categoria1) {
38                 if (i.getDorsal().toString().equals(lista.get(0).getDorsal().toString())) {
39                     i.setPuntosP(20);
40                 }
41                 if (i.getDorsal().toString().equals(lista.get(1).getDorsal().toString())) {
42                     i.setPuntosP(15);
43                 }
44                 if (i.getDorsal().toString().equals(lista.get(2).getDorsal().toString())) {
45                     i.setPuntosP(10);
46                 }
47                 if (i.getDorsal().toString().equals(lista.get(3).getDorsal().toString())) {
48                     i.setPuntosP(5);
49                 }
50             }
51             if (categoria2) {
52                 if (i.getDorsal().toString().equals(lista.get(0).getDorsal().toString())) {
53                     i.setPuntosP(10);
54                 }
55                 if (i.getDorsal().toString().equals(lista.get(1).getDorsal().toString())) {
56                     i.setPuntosP(5);
57                 }
58                 if (i.getDorsal().toString().equals(lista.get(2).getDorsal().toString())) {
59                     i.setPuntosP(3);
60                 }
61             }
62         }
63         if (categoria3) {
```

```
64         if (i.getDorsal().toString().equals(lista.get(0).getDorsal().toString())) {
65             i.setPuntosP(5);
66         }
67         if (i.getDorsal().toString().equals(lista.get(1).getDorsal().toString())) {
68             i.setPuntosP(3);
69         }
70         if (i.getDorsal().toString().equals(lista.get(2).getDorsal().toString())) {
71             i.setPuntosP(1);
72         }
73     }
74 }
75 }
76 }
77 }
78 public static int[] numerosPuntuacion(String categoria) {//este metodo sirve para en
funcion de que categoria de puerto o si es meta la puntuacion que se debe mostrar y
escribir en el fichero
79     int[] num1=new int[3];
80     int[] num2 = new int[4];
81     if(categoria.equals("categoria1")) {
82         num2[0]=20;
83         num2[1]=15;
84         num2[2]=10;
85         num2[3]=5;
86         return num2;
87     }else if(categoria.equals("categoria2")) {
88         num1[0]=10;
89         num1[1]=5;
90         num1[2]=3;
91     }else if(categoria.equals("categoria3")) {
92         num1[0]=5;
93         num1[1]=3;
94         num1[2]=1;
95     }else if(categoria.equals("meta")) {
96         num1[0]=5;
97         num1[1]=3;
98         num1[2]=1;
99     }
100     return num1;
101 }
102 }
103 }
104 }
105 public static void losPrimerosM() { // este metodo ordena el arraylist en funcion de
quien tiene mas puntos en metas.
106     ListaCicPuntos.sort(Comparator.comparing(Ciclista::getPuntosM).reversed());
107 }
108 }
109 public static void losPrimerosP() { // este metodo ordena el arraylist en funcion de
quien tiene mas puntos en puertos.
110     ListaCicPuntos.sort(Comparator.comparing(Ciclista::getPuntosP).reversed());
111 }
112 }
113 public static void ordenNumDorsal(ArrayList<Ciclista>uno) {
114     uno.sort(Comparator.comparing(Ciclista::getDorsal));
115 }
116 }
117 }
118 }
119 }
```