

```
1 package tests;
2
3 import java.util.Map;
18
19 public class TestEjercicio1 {
20
21     public static void main(String[] args) {
22         testEjercicio1("PI3E1A_DatosEntrada");
23         testEjercicio1("PI3E1B_DatosEntrada");
24     }
25
26     public static void testEjercicio1(String file) {
27
28         SimpleDirectedGraph<Persona,Relacion> dg = GraphsReader
29             .newGraph("ficheros/" + file + ".txt",
30                 Persona::ofFormat,
31                 Relacion::ofFormat,
32                 Graphs2::simpleDirectedGraph);
33
34         System.out.println("\nArchivo " + file + ".txt \n" + "Datos de entrada: " + dg);
35
36 //     Para mostrar el grafo original
37
38         GraphColors.toDot(dg,"resultados/ejercicio1/" + file + "_Datos" + ".gv",
39             x->x.nombre(), x->"", //Atributos del vertice y de la arista
40             v->GraphColors.color(Color.black), //Propiedades del vertice
41             e->GraphColors.color(Color.black)); //Propiedades de la arista
42
43
44 //     APARTADO A
45
46         Ejercicio1.apartadoA (dg, file);
47
48
49 //     APARTADO B
50
51         if (file == "PI3E1A_DatosEntrada") {
52             Persona p = Persona.of(13, "Maria", 2008, "Sevilla");
53
54             GraphColors.toDot(dg,"resultados/ejercicio1/" + file + "apartadoB" + ".gv",
55                 x->x.nombre() , x->"",
56                 v->colorea(p,v, Ejercicio1.apartadoB(dg, p)),
57                 e->GraphColors.style(Style.solid));
58             System.out.println(file + "apartadoB" + ".gv generado en " +
59 "resultados/ejercicio1");
60         } else {
61
62             Persona p = Persona.of(13, "Raquel", 1993, "Sevilla");
63
64             GraphColors.toDot(dg,"resultados/ejercicio1/" + file + "apartadoB" + ".gv",
65                 x->x.nombre() , x->"",
66                 v->colorea(p,v, Ejercicio1.apartadoB(dg, p)),
67                 e->GraphColors.style(Style.solid));
68             System.out.println(file + "apartadoB" + ".gv generado en " +
69 "resultados/ejercicio1");
70         }
71 //     APARTADO C
72
73         Persona p1 = null;
74         Persona p2 = null;
```

```
75
76     if (file == "PI3E1A_DatosEntrada") {
77         p1 = Persona.of(16, "Rafael", 2020, "Malaga");
78         p2 = Persona.of(14, "Sara", 2015, "Jaen");
79     } else {
80         p1 = Persona.of(14, "Julia", 1996, "Jaen");
81         p2 = Persona.of(6, "Angela", 1997, "Sevilla");
82     }
83     Ejercicio1.apartadoC (dg,p1,p2);
84
85 //     APARTADO D
86
87     Ejercicio1.apartadoD(dg, file);
88
89 //     APARTADO E
90
91     Graph<Persona,Relacion> g = GraphsReader
92         .newGraph("ficheros/" + file + ".txt",
93                 Persona::ofFormat,
94                 Relacion::ofFormat,
95                 Graphs2::simpleGraph);
96
97     Ejercicio1.ApartadoE(g, file);
98
99
100 }
101
102 private static Map<String,Attribute> colorea (Persona p, Persona v, Set<Persona> per){
103     Map<String,Attribute> res = GraphColors.color(Color.black);
104     if(p.equals(v)) {
105         res = GraphColors.color(Color.red);
106     }else if(per.contains(v)) {
107         res = GraphColors.color(Color.blue);
108     }
109     return res;
110 }
111
112
113
114 }
115
```