

## Edificio.java

```
1 package consorcio_F;
2 import java.util.ArrayList;
3
4 public class Edificio {
5
6     private String direccion;
7     private int cantDptos;
8     private ArrayList <Departamento> listaDptos;
9
10    public Edificio(String suDirec,int cuantosDptos) {
11        this.setCantDptos(cuantosDptos);
12        this.setDireccion(suDirec);
13        this.listaDptos = new ArrayList<>();
14    }
15
16    private void setDireccion(String suDirec) {
17        this.direccion = suDirec;
18    }
19
20    private void setCantDptos(int cuantosDptos) {
21        this.cantDptos = cuantosDptos;
22    }
23
24    public String getDireccion() {
25        return this.direccion;
26    }
27
28    public int getCantDptos() {
29        return this.cantDptos;
30    }
31
32    public Departamento buscarDpto(int unidadBuscada) {
33        Departamento dptoEncontrado = null;
34        int i=0;
35        /* para cada elemento de la lista compara su unidad con la
36           recibida
37        * por param. */
38        while (i<this.listaDptos.size() && dptoEncontrado==null)
39        {
40            if(this.listaDptos.get(i).getUnidad()
41                ==unidadBuscada) {
42
43            }
44        }
45        /* si coincide (encontro), se asigna a la var. que va a
```

## Edificio.java

```
retornar*/
40         dptoEncontrado = this.listaDptos.get(i);
41     } else {
42         i++;
43     }
44 }
45 return dptoEncontrado;
46 }
47
48 public int agregarUnidad(int unidadNue,String
    nombreNue,String dniNue) {
49     int codRetorno = -1;    // significa que no pudo agregar
50     if (unidadNue<=this.cantDptos) {
51 /* nro. de unidad valida. Entonces se intenta agregar*/
52         Departamento dptoNuevo = null;
53         dptoNuevo = buscarDpto(unidadNue);
54         if(dptoNuevo==null) {
55 /* no lo encuentra. Podemos crearlo*/
56             dptoNuevo = new
                Departamento(unidadNue,nombreNue,dniNue);
57             this.listaDptos.add(dptoNuevo);
58             codRetorno = 0;    // significa que pudo
agregar bien
59         }
60     } else {
61 /* nro. de unidad invalido. No se intenta agregar*/
62         codRetorno = 1;    // significa nro. unidad
invalida
63     }
64     return codRetorno;
65 }
66
67 public int actualizarVoto(int unidadQueVota,String suVoto) {
68     int codRetorno = -1;
69     Departamento dptoBuscado = null;
70     dptoBuscado = buscarDpto(unidadQueVota);
71     if (dptoBuscado!=null &&
        dptoBuscado.getVoto().equals("")) {
72         dptoBuscado.setVoto(suVoto);
73         codRetorno = 0;
74     }
75     return codRetorno;
```

## Edificio.java

```
76     }
77
78     public void listarPositivos() {
79         System.out.println("listado de dptos. que votaron
positivo");
80         for (int i=0;i<this.listaDptos.size();i++) {
81             if (this.listaDptos.get(i).getVoto().equals("S")) {
82                 System.out.println(this.listaDptos.get(i).toStri
ng());
83             }
84         }
85         for (Departamento dep : this.listaDptos) {
86             if (dep.getVoto().equals("S")) {
87                 System.out.println(dep.toString());
88             }
89         }
90     }
91
92     public void calcularPorcent() {
93         int cuantasVotaron = 0;
94         for (Departamento dep : this.listaDptos) {
95             if (!dep.getVoto().equals("")) {
96                 cuantasVotaron++;
97             }
98         }
99         double percent =
cuantasVotaron*100/this.listaDptos.size();
100         System.out.println("el percent. de unidades que votaron
es de "+percent+"%");
101     }
102 }
103
```