

Cliente.java

```
1 package Dominio;
2 import java.io.Serializable;
3 import java.rmi.RemoteException;
4 import java.util.LinkedList;
5 import java.util.List;
6 import Persistencia.AgenteBD;
7
8 public class Cliente implements Serializable{ /*Un cliente es una persona que
9 está registrada en el sistema pero no paga un recibo mensual. Cuando tiene que
10 pagar algo, lo hace en metálico en el momento de realizar la compra*/
11
12     private int dni;
13     private String nombre;
14     private String apellidos;
15     private String domicilio;
16     private String email;
17     private int esSocio;
18
19     public Cliente(int dni, String nombre, String apellidos, String domicilio,
20         String email, int esSocio) {
21         super();
22         this.dni = dni;
23         this.nombre = nombre;
24         this.apellidos = apellidos;
25         this.domicilio = domicilio;
26         this.email = email;
27         this.esSocio = esSocio;
28     }
29     public Cliente() {
30         super();
31     }
32     public int getDni() {
33         return dni;
34     }
35     public void setDni(int dni) {
36         this.dni = dni;
37     }
38     public String getNombre() {
39         return nombre;
40     }
41     public void setNombre(String nombre) {
42         this.nombre = nombre;
43     }
44     public String getApellidos() {
45         return apellidos;
46     }
47     public void setApellidos(String apellidos) {
48         this.apellidos = apellidos;
49     }
50     public String getDomicilio() {
51         return domicilio;
52     }
53     public void setDomicilio(String domicilio) {
54         this.domicilio = domicilio;
55     }
56     public String getEmail() {
57         return email;
58     }
59     public void setEmail(String email) {
60         this.email = email;
61     }
62     public int getEsSocio() {
```

Cliente.java

```

63         return esSocio;
64     }
65     public void setEsSocio(int esSocio) {
66         this.esSocio = esSocio;
67     }
68     public static int registrarCliente(int id,String n,String a, String d,String
e,int s){
69         int res=0;
70         Cliente c= new Cliente(id,n,a,d,e,s);
71         try {
72             AgenteBD db=new AgenteBD();
73             res=db.crearClienteBD(c);
74         } catch (RemoteException el) {
75             el.printStackTrace();
76         }
77         return res;
78     }
79
80     public static List<Cliente> consultarClientes(){
81         List<Cliente> listaClientes=null;
82         try {
83             AgenteBD db=new AgenteBD();
84             listaClientes = db.consultarClientesBD();
85         } catch (RemoteException e) {
86             e.printStackTrace();
87         }
88         return listaClientes;
89     }
90
91     public static Cliente consultarCliente(int dni){
92         Cliente c=null;
93         try {
94             AgenteBD db=new AgenteBD();
95             c = db.consultarClienteBD(dni);
96         } catch (RemoteException e) {
97             e.printStackTrace();
98         }
99         return c;
100     }
101
102     public static int modificarCliente(int id,String n,String a, String d,String
e,int s){
103         int res=0;
104         Cliente c;
105         try {
106             AgenteBD db=new AgenteBD();
107             c = db.consultarClienteBD(id);
108             if(c.getEsSocio()==1){
109                 if(s==1){
110                     c.setNombre(n);
111                     c.setApellidos(a);
112                     c.setDomicilio(d);
113                     c.setEmail(e);
114                     res=db.modificarClienteBD(c);
115                 }
116                 else{
117                     c.setNombre(n);
118                     c.setApellidos(a);
119                     c.setDomicilio(d);
120                     c.setEmail(e);
121                     Socio.bajaSocio(id);
122                 }

```

Cliente.java

```

123         }
124         else{
125             c.setNombre(n);
126             c.setApellidos(a);
127             c.setDomicilio(d);
128             c.setEmail(e);
129             c.setEsSocio(s);
130             res=db.modificarClienteBD(c);
131         }
132     } catch (RemoteException e1) {
133         e1.printStackTrace();
134     }
135     return res;
136 }
137
138 public static int borrarCliente(int id){
139     int res=0;
140     Cliente c=consultarCliente(id);
141     if(c.getEsSocio()==1){
142         res=-1;
143     }
144     else{
145         try {
146             AgenteBD db=new AgenteBD();
147             res=db.borrarClienteBD(id);
148         } catch (RemoteException e) {
149             e.printStackTrace(); res=-1;
150         }
151     }
152     return res;
153 }
154
155 public static List<Cliente> consultarClientesNoSocios() {
156     List<Cliente> listaClientes=null;
157     try {
158         AgenteBD db=new AgenteBD();
159         listaClientes = db.consultarClientesNoSociosBD();
160     } catch (RemoteException e) {
161         e.printStackTrace();
162     }
163     return listaClientes;
164 }
165 public static List<Cliente> consultarClientesSociosTitulares() {
166     List<Socio> ls=null;
167     try {
168         AgenteBD db=new AgenteBD();
169         ls = db.consultarSociosTitularesBD();
170     } catch (RemoteException e) {
171         e.printStackTrace();
172     }
173     LinkedList<Cliente> lc=new LinkedList<Cliente> ();
174     for (int i = 0; i < ls.size(); i++) {
175         try {
176             AgenteBD db=new AgenteBD();
177             lc.add(db.consultarClienteBD(ls.get(i).getDni()));
178         } catch (RemoteException e) {
179             e.printStackTrace();
180         }
181     }
182     return lc;
183 }
184

```

Cliente.java

```

185     public static List<Cliente> consultarClientesGrupo(int dniTit) {
186         List<Cliente> listaClientes=null;
187         try {
188             AgenteBD db=new AgenteBD();
189             listaClientes = db.consultarClientesGrupoBD(dniTit);
190         } catch (RemoteException e) {
191             e.printStackTrace();
192         }
193         return listaClientes;
194     }
195
196     @Override
197     public int hashCode() {
198         final int prime = 31;
199         int result = 1;
200         result = prime * result
201             + ((apellidos == null) ? 0 : apellidos.hashCode());
202         result = prime * result + dni;
203         result = prime * result
204             + ((domicilio == null) ? 0 : domicilio.hashCode());
205         result = prime * result + ((email == null) ? 0 : email.hashCode());
206         result = prime * result + esSocio;
207         result = prime * result + ((nombre == null) ? 0 : nombre.hashCode());
208         return result;
209     }
210
211     @Override
212     public boolean equals(Object obj) {
213         if (this == obj)
214             return true;
215         if (obj == null)
216             return false;
217         if (getClass() != obj.getClass())
218             return false;
219         final Cliente otro = (Cliente) obj;
220         if (apellidos == null) {
221             if (otro.apellidos != null)
222                 return false;
223         } else if (!apellidos.equals(otro.apellidos))
224             return false;
225         if (dni != otro.dni)
226             return false;
227         if (domicilio == null) {
228             if (otro.domicilio != null)
229                 return false;
230         } else if (!domicilio.equals(otro.domicilio))
231             return false;
232         if (email == null) {
233             if (otro.email != null)
234                 return false;
235         } else if (!email.equals(otro.email))
236             return false;
237         if (esSocio != otro.esSocio)
238             return false;
239         if (nombre == null) {
240             if (otro.nombre != null)
241                 return false;
242         } else if (!nombre.equals(otro.nombre))
243             return false;
244         return true;
245     }
246 }

```