

Reserva.java

```

1 package Dominio;
2
3 import Persistencia.AgenteBD;
4 import java.io.Serializable;
5 import java.rmi.RemoteException;
6 import java.util.Date;
7 import java.util.List;
8
9 public class Reserva implements Serializable{ /*Se trata de la reserva de un
    servicio
    por parte de un cliente o socio*/
10
11     private int id; //ID de la reserva
12     private int servicio; //ID del servicio reservado
13     private int usuario; //ID del usuario que gestiona la reserva
14     private Date fechaActual; //fecha actual, cuando la reserva se lleva a cabo
15     private int cliente; //ID del cliente que realiza la reserva
16     private Date fechaReserva; //fecha para la que se hace la reserva
17     private int horaReserva; //hora para la que se hace la reserva
18     private int idPago; //el ID del pago relacionado con esta reserva
19
20     public Reserva() {
21         super();
22     }
23
24     public Reserva(int servicio, int usuario,
25         int cliente, Date fechaReserva, int horaReserva ,int idPago) {
26         super();
27         this.servicio = servicio;
28         this.usuario = usuario;
29         this.cliente = cliente;
30         this.fechaReserva = fechaReserva;
31         this.horaReserva=horaReserva;
32         this.idPago = idPago;
33     }
34
35     public int getId() {
36         return id;
37     }
38
39     public void setId(int id) {
40         this.id = id;
41     }
42
43     public int getServicio() {
44         return servicio;
45     }
46
47     public void setServicio(int servicio) {
48         this.servicio = servicio;
49     }
50
51     public int getUsuario() {
52         return usuario;
53     }
54
55     public void setUsuario(int usuario) {
56         this.usuario = usuario;
57     }
58
59     public Date getFechaActual() {
60         return fechaActual;
61     }

```

```

62
63     public void setFechaActual(Date fechaActual) {
64         this.fechaActual = fechaActual;
65     }
66
67     public int getCliente() {
68         return cliente;
69     }
70
71     public void setCliente(int cliente) {
72         this.cliente = cliente;
73     }
74
75     public Date getFechaReserva() {
76         return fechaReserva;
77     }
78
79     public void setFechaReserva(Date fechaReserva) {
80         this.fechaReserva = fechaReserva;
81     }
82
83     public int getHoraReserva() {
84         return horaReserva;
85     }
86
87     public void setHoraReserva(int horaReserva) {
88         this.horaReserva = horaReserva;
89     }
90
91     public int getIdPago() {
92         return idPago;
93     }
94
95     public void setIdPago(int idPago) {
96         this.idPago = idPago;
97     }
98
99     public static int crearReserva(int servicio, int usuario,
100         int cliente, Date fechaReserva, int horaReserva){
101         int res=0;
102         try {
103             AgenteBD db=new AgenteBD();
104             if(fechaReserva.compareTo(new Date())>0)
105                 res=db.crearReserva(servicio,usuario,cliente,fechaReserva,horaReser
106 va);
107             else res=-1;
108         } catch (RemoteException e) {
109             e.printStackTrace(); res=-1;
110         }
111         return res;
112     }
113
114     public static List<Reserva> consultarReservas(int idServicio,Date fecha){
115         List<Reserva> lr=null;
116         try {
117             AgenteBD db=new AgenteBD();
118             lr = db.consultarReservasFechaBD(idServicio,fecha);
119         } catch (RemoteException e) {
120             e.printStackTrace();
121         }
122         return lr;
123     }

```

Reserva.java

```
123
124     public static int anularReserva(int reserva){
125         int res=0;
126         try {
127             AgenteBD db=new AgenteBD();
128             res=db.anularReservaBD(reserva);
129         } catch (RemoteException e) {
130             e.printStackTrace(); res=-1;
131         }
132         return res;
133     }
134 }
```