

```

1 package Dominio;
2
3 import java.awt.Dimension;
4 import java.awt.Toolkit;
5 import java.io.ByteArrayOutputStream;
6 import java.io.IOException;
7 import java.io.InputStream;
8 import java.io.Serializable;
9 import java.sql.Blob;
10 import java.sql.SQLException;
11 import java.util.Calendar;
12 import java.util.GregorianCalendar;
13 import java.util.Iterator;
14 import java.util.List;
15 import javax.swing.JFrame;
16
17 public class Utilidades implements Serializable{
18
19     public static String FechayHoraActual(){
20         String fyh="";
21         Calendar calendario = new GregorianCalendar();
22         int horas, minutos, segundos;
23         int dia,mes,año;
24         dia =calendario.get(Calendar.DAY_OF_MONTH);
25         mes =calendario.get(Calendar.MONTH)+1;
26         año =calendario.get(Calendar.YEAR);
27         horas =calendario.get(Calendar.HOUR_OF_DAY);
28         minutos = calendario.get(Calendar.MINUTE);
29         segundos = calendario.get(Calendar.SECOND);
30         String dd,mm,aaa,h,m,s;
31         dd=ceros(String.valueOf(dia));
32         mm=ceros(String.valueOf(mes));
33         aaa=ceros(String.valueOf(año));
34         h=ceros(String.valueOf(horas));
35         m=ceros(String.valueOf(minutos));
36         s=ceros(String.valueOf(segundos));
37         fyh=dd+"/"+mm+"/"+aaa+" "+h+": "+m+": "+s;
38         return fyh;
39     }
40
41     public static void CentrarJFrame(JFrame v){
42         Dimension pantalla, cuadro;
43         pantalla = Toolkit.getDefaultToolkit().getScreenSize();
44         cuadro = v.getSize();
45         v.setLocation(((pantalla.width - cuadro.width)/2),
46                     (pantalla.height - cuadro.height)/2);
47     }
48
49     public static String ceros(String f){
50         String res=f;
51         if(f.equals("0")) res="00";
52         else if(f.equals("1")) res="01";
53         else if(f.equals("2")) res="02";
54         else if(f.equals("3")) res="03";
55         else if(f.equals("4")) res="04";
56         else if(f.equals("5")) res="05";
57         else if(f.equals("6")) res="06";
58         else if(f.equals("7")) res="07";
59         else if(f.equals("8")) res="08";
60         else if(f.equals("9")) res="09";
61         return res;
62     }

```

```

63
64     public static String[][] construirMatrizTablaClientes(List<Cliente> l){
65         String [][]m=new String [l.size()][3];
66         Iterator<Cliente> iter = l.iterator();
67         int i=0;
68         while (iter.hasNext()){
69             Cliente c=(Cliente)iter.next();
70             m[i][0]=String.valueOf(c.getDni());
71             m[i][1]=c.getNombre();
72             m[i][2]=c.getApellidos();
73             i++;
74         }
75         return m;
76     }
77
78     public static String[][] construirMatrizTablaUsuarios(List<Usuario> l){
79         String [][]m=new String [l.size()][3];
80         Iterator<Usuario> iter = l.iterator();
81         int i=0;
82         while (iter.hasNext()){
83             Usuario u=(Usuario)iter.next();
84             m[i][0]=String.valueOf(u.getDni());
85             m[i][1]=u.getNombre();
86             m[i][2]=u.getApellidos();
87             i++;
88         }
89         return m;
90     }
91
92     public static String[]construirArrayServicios(List<Servicio>l){
93         String [] res=new String[l.size()];
94         Iterator<Servicio> iter = l.iterator();
95         int i=0;
96         while (iter.hasNext()){
97             Servicio s=(Servicio)iter.next();
98             if(s.getHabilitado()==0){
99                 res[i]=String.valueOf(s.getId())+" - "+s.getDescripcion()+"
100             (DESHABILITADO)";
101             }
102             else{
103                 res[i]=String.valueOf(s.getId())+" - "+s.getDescripcion();
104             }
105             i++;
106         }
107         return res;
108     }
109
110     public static String [][] construirTablaReservasDia(List<Reserva>l){
111         String [][]m=new String [13][5];
112         int i=0;
113         for (int j = 9; j <= 21; j++) {
114             if(i<l.size() && l.get(i).getHoraReserva()==j){
115                 m[j-9][0]=String.valueOf(l.get(i).getHoraReserva());
116                 m[j-9][1]=String.valueOf(l.get(i).getFechaReserva());
117                 m[j-9][2]=String.valueOf(l.get(i).getCliente());
118                 m[j-9][3]=String.valueOf(l.get(i).getUsuario());
119                 m[j-9][4]=String.valueOf(l.get(i).getId());
120                 i++;
121             }
122             else{
123                 m[j-9][0]=String.valueOf(j);
124                 m[j-9][1]=" - ";

```

```

124         m[j-9][2]="        -        ";
125         m[j-9][3]="        -        ";
126         m[j-9][4]="        -        ";
127     }
128 }
129 return m;
130 }
131
132 public static String[][] construirMatrizTablaPagos(List<Pago> l){
133     String [][]m=new String [l.size()][7];
134     Iterator<Pago> iter = l.iterator();
135     int i=0;
136     while (iter.hasNext()){
137         Pago p=(Pago)iter.next();
138         m[i][0]=String.valueOf(p.getId());
139         m[i][1]=String.valueOf(p.getCliente());
140         m[i][2]=String.valueOf(p.getCantidad());
141         m[i][3]=String.valueOf(p.getConcepto());
142         m[i][4]=String.valueOf(p.getTipoPago());
143         m[i][5]=String.valueOf(p.getFecha());
144         m[i][6]=String.valueOf(p.getLiquidado());
145         i++;
146     }
147     return m;
148 }
149
150 public static byte[] toByteArray(Blob fromBlob) {
151     ByteArrayOutputStream baos = new ByteArrayOutputStream();
152     try {
153         return toByteArrayImpl(fromBlob, baos);
154     }
155     catch (SQLException e) {
156         throw new RuntimeException(e);
157     }
158     catch (IOException e) {
159         throw new RuntimeException(e);
160     }
161     finally {
162         if (baos != null) {
163             try {
164                 baos.close();
165             }
166             catch (IOException ex) {
167             }
168         }
169     }
170 }
171
172 private static byte[] toByteArrayImpl(Blob fromBlob, ByteArrayOutputStream
baos)
173     throws SQLException, IOException {
174     byte[] buf = new byte[100000];
175     InputStream is = fromBlob.getBinaryStream();
176     try {
177         for (;;) {
178             int dataSize = is.read(buf);
179
180             if (dataSize == -1)
181                 break;
182             baos.write(buf, 0, dataSize);
183         }
184     }

```

```
185         finally {
186             if (is != null) {
187                 try {
188                     is.close();
189                 }
190                 catch (IOException ex) {
191                 }
192             }
193         }
194         return baos.toByteArray();
195     }
196
197     public static String[][] construirMatrizTablaServicios(List<Servicio> l) {
198         String [][]m=new String [l.size()][4];
199         Iterator<Servicio> iter = l.iterator();
200         int i=0;
201         while (iter.hasNext()){
202             Servicio s=(Servicio)iter.next();
203             m[i][0]=String.valueOf(s.getId());
204             m[i][1]=String.valueOf(s.getDescripcion());
205             m[i][2]=String.valueOf(s.getPrecioHora());
206             m[i][3]=String.valueOf(s.getHabilitado());
207             i++;
208         }
209         return m;
210     }
211 }
```