

```

1  /*Clase creada para la interaccion con los archivos de texto*/
2  package newairlineseller;
3
4  import java.io.*;
5  import java.util.regex.Matcher;
6  import java.util.regex.Pattern;
7
8  public class Manejador {
9      PrintWriter pf;
10     FileReader fr;
11     BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
12
13     void crear(String nArchivo) {
14         /*crea el archivo vacio o verifica su existencia,
15         * necesario para almacenar los datos que generan los distintos metodos
16         * nArchivo es el parametro con el nombre que tendra el fichero de texto
17         */
18         try{
19             pf = new PrintWriter(new FileWriter(nArchivo+".txt",true));
20             pf.close();
21             System.out.println("Archivo "+nArchivo+".txt se genero o existe...");
22         } catch (Exception e) {
23             System.out.println("Error en creacion de archivo.\nError: "+e);
24         }
25     }
26
27     void guardar(String nArchivo, String nDatos){
28         /*crea o verifica la existencia del archivo para agregar los datos
29         * generados en los diferentes metodos que lo invocan
30         */
31         try{
32             pf = new PrintWriter(new FileWriter(nArchivo+".txt",true));
33             pf.println(nDatos);
34             pf.close();
35             System.out.println("Archivo "+nArchivo+".txt se genero o existe...");
36         } catch (Exception e) {
37             System.out.println("Error de guardado en archivo.\nError: "+e);
38         }
39     }
40
41     String[] buscar(String nArchivo, String ID){
42         /*busca por numero de cedula una linea especifica dentro de un archivo de texto
43         */
44         String[] datos;
45         try {
46             fr = new FileReader(nArchivo+".txt");
47             BufferedReader brf = new BufferedReader(fr);
48             String linea;
49             while ((linea=brf.readLine()) != null) {
50                 datos = linea.split("-");
51                 if(datos[0].equalsIgnoreCase(ID)) {
52                     return datos;
53                 }
54             }
55         } catch (Exception e) {
56         }
57         return datos = null;
58     }

```

```

59
60     double[] buscarMontos(int nVuelo) throws IOException{
61         /*busca los montos en un archivo de texto*/
62         String[] datos;
63         double[] suma = {0,0,0,0,0,0,0,0};
64         //suma contiene{0-monto total, 1-monto clase A, 2-monto clase B, 0-monto clase C,
65         //4-cantidad total de boletos, 5-cant boletos A, 6-cant boletos B, 7-cant boletos C}
66         fr = new FileReader("boletos"+nVuelo+".txt");
67         BufferedReader brf = new BufferedReader(fr);
68         String linea;
69
70         while((linea=brf.readLine())!=null){
71             datos = linea.split("-");
72             if(datos!=null){
73                 suma[4]++;
74                 double aux = Double.parseDouble(datos[6]);
75                 suma[0] += aux;
76                 if(datos[5].equals("A")){
77                     suma[1] += aux;
78                     suma[5]++;
79                 }
80                 if(datos[5].equals("B")){
81                     suma[2] += aux;
82                     suma[6]++;
83                 }
84                 if(datos[5].equals("C")){
85                     suma[3] += aux;
86                     suma[7]++;
87                 }
88             }
89         }
90         fr.close();
91         return suma;
92     }
93
94     int buscar(String nArchivo) throws IOException{
95         /*busca en un archivo el valor de la primera posicion en cada linea para
96         * saber el siguiente numero para el el vuelo*/
97         int mayor = 0;
98         String[] datos;
99         fr = new FileReader(nArchivo+".txt");
100        BufferedReader brf = new BufferedReader(fr);
101        String linea;
102        while((linea = brf.readLine())!=null){
103            datos = linea.split("-");
104
105            if (datos.length==1){
106                return 0;
107            }else{
108                //         for (int i = 0; i < datos.length; i++) {
109                //             int indice = Integer.parseInt(datos[i]);
110                //             if(indice>= mayor){
111                //                 mayor = indice;
112                //             }else{
113                //                 return -1;
114                //             }
115                //         }
116            }
117        }

```

```

118         return mayor; //retorna el valor del ultimo vuelo
119     }
120
121     void leerArchivoCliente() throws IOException{
122         /*Imprime en pantalla una lista entendible con el contenido del archivo
123         * de clientes
124         */
125         fr = new FileReader("cliente.txt");
126         BufferedReader brf = new BufferedReader(fr);
127         String linea;
128         System.out.println("Litado de clientes");
129         while ((linea=brf.readLine()) != null) {
130             String[] datos = linea.split("-");
131             System.out.println("\033[34m-----");
132             System.out.println("\033[34mCedula:\t\t\033[31m"+datos[0]);
133             System.out.println("\033[34mNombre:\t\t\033[31m"+datos[2]+" "+datos[1]);
134             System.out.println("\033[34mTelefono:\t\t\033[31m"+datos[3]);
135             System.out.println("\033[34mDireccion:\t\t\033[31m"+datos[4]);
136         }
137         System.out.println("\033[34m-----");
138
139     }
140
141     void leerArchivoVuelo() throws IOException{
142         /*Imprime en pantalla una lista entendible con el contenido del archivo
143         * de clientes
144         */
145         fr = new FileReader("vuelo.txt");
146         BufferedReader brf = new BufferedReader(fr);
147         String linea;
148         System.out.println("\033[31mLitado de vuelos");
149         System.out.println("|-----");
150         while ((linea=brf.readLine()) != null) {
151             String[] datos = linea.split("-");
152             System.out.println("|Vuelo n° | Fecha | Hora | Origen | Destino |
153             Tarifa\t|");
154             System.out.println("| "+datos[0]+" \t | "+datos[1]+" | "+datos[2]+" |
155             "+datos[3]+" | "+datos[4]+" | "+datos[5]+" \t|");
156             System.out.println("|-----");
157             // System.out.println( "N° de vuelo:\t"+datos[0]);
158             // System.out.println( "Fecha y hora:\t"+datos[2]+" "+datos[1]);
159             // System.out.println("Ciudad Origen:\t"+datos[3]
160             // +"\nCiudad Destino:\t"+datos[4]);
161             // System.out.println( "Tarifa Base:\t"+datos[5]);
162             // System.out.println("-----");
163             // System.out.println();
164         }
165     }
166
167     boolean validarCedula(String ci){
168         //valida la cedula comparandola con un patron
169         Pattern pat = Pattern.compile("[0-9]{6,8}");
170         Matcher mat = pat.matcher(ci);
171         if(mat.matches()) {return true;}
172         else{return false;}
173     }

```

```

173 boolean validarNombApe(String nom){
174     //valida el nombre y el apellido comparandolo con un patron
175     boolean patF = Pattern.matches("[a-zA-Z]{1,100}",nom);
176     return patF;
177 }
178
179 boolean validarTelf(String telf){
180     //valida el telefono comparandolo con un patron
181     boolean patF = Pattern.matches("[0-9]{11}",telf);
182     return patF;
183 }
184
185 boolean validarDir(String dir){
186     //valida la direccion comparandola con un patron
187     boolean patF = Pattern.matches("[a-zA-Z ]{1,200}",dir);
188     return patF;
189 }
190
191 boolean validarTarifa(String t){
192     boolean patT = Pattern.matches("\\d+\\.\\d*",t);
193     return patT;
194 }
195
196 boolean validarFila(String fil){
197     //verifica que el numero de fila este entre 1 y 16
198     boolean patT = Pattern.matches("[1-16]",fil);
199     return patT;
200 }
201
202 boolean validarHora(String vHora){
203     //verifica que la hora ingresada tenga un formato valido.
204     boolean patH = Pattern.matches("[0-2]?\\d:[0-5]?\\d",vHora);
205     return patH;
206 }
207
208 boolean validarFecha(String vFecha){
209     //verifica que la fecha tenga un formato valido.
210     boolean patF = Pattern.matches("\\d+\\/\\d+\\/\\d{4}",vFecha);
211     return patF;
212 }
213
214 String selecCiudad() throws IOException{
215     /*muestra una lista de ciudades disponibles para los vuelos y devuelve
216     * el codigo para una de ella*/
217     String ciudad = "";
218     boolean val=false;
219     do {
220         System.out.println("1 - Caracas.");
221         System.out.println("2 - Barquisimeto.");
222         System.out.println("3 - Valencia.");
223         System.out.println("4 - Barcelona.");
224         System.out.println("5 - Maracaibo.");
225         System.out.println("6 - Porlamar");
226         System.out.print("Seleccione Ciudad: ");
227
228         String aux3 = br.readLine();
229         if(Pattern.matches("[1-6]", aux3)){
230             switch(aux3){
231                 case "1" : ciudad = "CCS"; val=true; break;

```

```

232         case "2" : ciudad = "BRM"; val=true; break;
233         case "3" : ciudad = "VLN"; val=true; break;
234         case "4" : ciudad = "BLA"; val=true; break;
235         case "5" : ciudad = "MAR"; val=true; break;
236         case "6" : ciudad = "POR"; val=true; break;
237     }
238     }else{
239         System.out.println("Seleccion invalida.");
240         val=false;
241     }
242     } while (!val);
243     return ciudad;
244 }
245
246 boolean verDispon(int fil, int col, String numVuelo){
247 //     verifica la disponibilidad de un asiento en un vuelo especifico
248     String[] datos;
249     try {
250         fr = new FileReader("boletos"+numVuelo+".txt");
251         BufferedReader brf = new BufferedReader(fr);
252         String linea;
253         while((linea=brf.readLine())!=null){
254             datos = linea.split("-");
255             if((datos[3].equals(fil+"") && (datos[4].equals(col+""))){
256                 return false;
257             }
258         }
259     } catch (Exception e) {
260         System.err.println(e);
261     }
262     return true;
263 }
264
265 }

```