

```
1 /*Venta de boletos aéreos para JSalvaG Airlines con una flota de 10 aviones Boeing 373, con
   vuelos a las principales ciudades venezolanas:
2  * Caracas
3  * Porlamar
4  * Valencia
5  * Maracaibo
6  * Barcelona
7  * Barquisimeto
8  * Cada aeronave cuenta con 3 clases: VIP, Ejecutiva y turística.
9 */
10 package newairlineseller;
11
12 import java.io.IOException;
13
14 /**
15  * @author JSalvaG
16 */
17 public class NewAirLineSeller {
18
19     public static void main(String[] args) throws IOException {
20         Menu menu = new Menu();
21
22         menu.menuPrincipal();
23     }
24 }
```

```

1 package newairlineseller;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStreamReader;
6
7 public class Menu {
8     BufferedReader br = new BufferedReader (new InputStreamReader(System.in));
9     Cliente c = new Cliente();
10    Manejador m = new Manejador();
11    Vuelos v = new Vuelos();
12    Boleto b = new Boleto();
13    Reportes r = new Reportes();
14
15
16    void menuPrincipal() throws IOException{
17        m.crear("vuelo");
18        m.crear("cliente");
19        System.out.println("\033[35m-----");
20        System.out.println("\033[35m----- BienVenido -----");
21        System.out.println("\033[35m-----Sistema de venta de Boletos para JSalvaG Airlines-----");
22        System.out.println("\033[35m-----");
23        System.out.println("\033[35m-----Selecciones su accion-----");
24        System.out.println("\033[35m-----");
25        boolean salir = false;
26
27        do{
28            System.out.println("\n1.-Gestion de Vuelos.");
29            System.out.println("2.-Gestion de clientes.");
30            System.out.println("3.-Gestion de boletos.");
31            System.out.println("4.-Reportes.");
32            System.out.print("5.-Salir.\nSeleccion: ");
33
34            switch (br.readLine()) {
35                case "1": //GESTion de vuelos...
36                    subMenu2();
37                    break;
38                case "2":
39                    subMenu1();
40                    break;
41                case "3":
42                    subMenu3();
43                    break;
44                case "4":
45                    subMenu4();
46                    break;
47                case "5":
48                    salir = true;
49                    System.out.println("Adios!!!!");
50                    break;
51                default:
52                    System.out.println("Opcion invalida.");

```

```

53                     break;
54                 }
55             }while(!salir);
56         }
57
58     void subMenu1() throws IOException{
59         boolean salir = false;
60         do{
61             System.out.println("-----");
62             System.out.println("- Registro y gestion de clientes -");
63             System.out.println("-----");
64             System.out.println("1.-Registrar Cliente Nuevo");
65             System.out.println("2.-Buscar Cliente Registrado");
66             System.out.println("3.-Ver lista de Clientes");
67             System.out.println("4.-Regresar al menu anterior");
68             System.out.println("-----");
69             System.out.print("Opcion: ");
70             String op = br.readLine();
71
72             switch(op){
73                 case "1":
74                     c.crearCliente();
75                     break;
76                 case "2":
77                     c.buscarCliente();
78                     break;
79                 case "3":
80                     m.leerArchivoCliente();
81                     break;
82                 case "4":
83                     salir = true;
84                     break;
85                 default:
86                     System.out.println("Opcion invalida");
87                     salir = false;
88             }
89         }while(!salir);
90     }
91
92     void subMenu2() throws IOException{//Submenu de gestion de vuelos
93         boolean salir = false;
94         do{
95             System.out.println("-----");
96             System.out.println("- Registro y gestion de vuelos -");
97             System.out.println("-----");
98             System.out.println("1.-Crear Vuelo");
99             System.out.println("2.-Buscar Vuelo");
100            System.out.println("3.-Ver lista de Vuelos");
101            System.out.println("4.-Ver Disponibilidad en Vuelo");
102            System.out.println("5.-Regresar al menu anterior");
103            System.out.println("-----");
104            System.out.print("Opcion: ");
105            String op = br.readLine();
106
107            switch(op){
108                case "1"://crear vuelo.
109                    v.crearVuelo();
110                    break;
111                case "2"://ver vuelos creados.

```

```

112             v.verVuelos();
113             break;
114         case "3":
115             m.leerArchivoVuelo();
116             break;
117         case "4":
118             System.out.print("Numero de Vuelo aconsultar: ");
119             String nV = br.readLine();
120             v.disponibilidad(nV);
121             break;
122         case "5":
123             salir = true;
124             break;
125         default:
126             System.out.println("Opcion invalida");
127             salir = false;
128         }
129     }while(!salir);
130 }
131
132 void subMenu3() throws IOException{
133     boolean salir = false;
134     do{
135         System.out.println("-----");
136         System.out.println("- Gestion y venta de boletos -");
137         System.out.println("-----");
138         System.out.println("1.-Venta de Boletos");
139         System.out.println("2.-Ver lista de Boletos");
140         System.out.println("3.-Regresar al menu anterior");
141         System.out.println("-----");
142         System.out.print("Opcion: ");
143         String op = br.readLine();
144
145         switch(op){
146             case "1":
147                 b.crearBoleto();
148                 break;
149             case "2":
150                 b.imprimirBoleto();
151                 break;
152             case "3":
153                 salir = true;
154                 break;
155             default:
156                 System.out.println("Opcion invalida");
157                 salir = false;
158             }
159     }while(!salir);
160 }
161
162 void subMenu4() throws IOException{
163     boolean salir = false;
164     do{
165         System.out.println("-----");
166         System.out.println("- Generacion de reportes -");
167         System.out.println("-----");
168         System.out.println("1.-Ventas totales.");
169         System.out.println("2.-Vuelos vendidos.");
170         System.out.println("3.-Regresar al menu anterior");

```

```
171     System.out.println("-----");
172     System.out.print("Opcion: ");
173     String op = br.readLine();
174
175     switch(op) {
176         case "1":
177             r.imprimirTotal();
178             break;
179         case "2":
180             r.vueloVendidos();
181             break;
182         case "3":
183             salir = true;
184             break;
185         default:
186             System.out.println("Opcion invalida");
187             salir = false;
188     }
189 }while(!salir);
190
191 }
```

```

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 ficerro 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\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 | Tarifa|t|");
153             System.out.println("| "+datos[0]+"\t | "+datos[1]+" | "+datos[2]+" | "+datos[3]+" | "+datos[4]+" | "+datos[5]+"\t|");
154             System.out.println("|-----");
155         }
156         System.out.println( "Nº de vuelo:\t"+datos[0]);
157         System.out.println( "Fecha y hora:\t"+datos[2]+" "+datos[1]);
158         System.out.println("Ciudad Origen:\t"+datos[3]
159                     +"\nCiudad Destino:\t"+datos[4]);
160         System.out.println( "Tarifa Base:\t"+datos[5]);
161         System.out.println("-----");
162         System.out.println();
163     }
164
165     boolean validarCedula(String ci){
166         //valida la cedula comparandola con un patron
167         Pattern pat = Pattern.compile("[0-9]{6,8}");
168         Matcher mat = pat.matcher(ci);
169         if(mat.matches()) {return true;}
170         else{return false;}
171     }
172

```

```

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 }
```

```

1 /*Creacion y gestion del registro de clientes*/
2 package newairlineseller;
3
4 import java.io.*;
5 import java.util.logging.Level;
6 import java.util.logging.Logger;
7
8 public class Cliente {
9     boolean si, si1;
10    String nom, ape, ci, dir, telf, aux;
11    Manejador man = new Manejador();
12    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
13
14    void crearCliente() {
15 /*crea un archivo con la informacion de los clientes que compran un voleto
16 * se emplean ciclos reperitivos para verificar que los datos ingresados
17 * son correctos o tiene un formato compatible con el sistema*/
18        try {
19            man.crear("cliente");
20            do{
21                System.out.print("Cedula: \t");
22                aux = br.readLine();
23                if(man.validarCedula(aux)) {
24                    ci = aux;
25                    String[] ver = man.buscar("cliente", ci);
26
27                    if (ver==null) {
28                        pedirDatos();
29                        String data = ci+"-"+nom+"-"+ape+"-"+telf+"-"+dir;
30                        man.guardar("cliente", data);
31                        si = true;
32                    }else{
33                        if(ver[0].equals(ci)){
34                            System.out.println("\033[30mNombre:\t\t\033[31m"+ver[1]);
35                            nom = ver[1];
36                            System.out.println("\033[30mAPELLIDO:\t\033[31m"+ver[2]);
37                            ape = ver[2];
38                            System.out.println("\033[30mTELÉFONO:\t\033[31m"+ver[3]);
39                            telf = ver[3];
40                            System.out.println("\033[30mDIRECCIÓN:\t\033[31m"+ver[4]);
41                            dir = ver[4];
42                            System.out.println("\033[31mCliente encontrado!!!\033[30m");
43                            si = true;
44                        }else{
45                            pedirDatos();
46                            String data = ci+"-"+nom+"-"+ape+"-"+telf+"-"+dir;
47                            man.guardar("cliente", data);
48                            si = true;
49                        }
50                    }
51                }else{
52                    System.out.println("Cedula invalida..");
53                    si = false;
54                }
55            }while(!si);
56        } catch (IOException ex) {
57            Logger.getLogger(Cliente.class.getName()).log(Level.SEVERE, null, ex);
58        }
}

```

```

59      }
60
61  void pedirDatos() throws IOException{
62 //      pide los datos basicos de informacion para el archivo cliente.txt
63  do{
64      System.out.print("Nombre:\t\t");
65      aux = br.readLine();
66      if(man.validarNombApe(aux)) {
67          nom = aux;
68          sil = true;
69      }else{
70          System.out.println("Nombre invalido!!!!");
71          sil = false;
72      }
73  }while(!sil);
74  do{
75      System.out.print("Apellido:\t");
76      aux = br.readLine();
77      if(man.validarNombApe(aux)) {
78          ape = aux;
79          sil = true;
80      }else{
81          System.out.println("Apellido invalido!!!!");
82          sil = false;
83      }
84  }while(!sil);
85  do{
86      System.out.print("Telefono:\t");
87      aux = br.readLine();
88      if(man.validarTelf(aux)) {
89          telf = aux;
90          sil = true;
91      }else{
92          System.out.println("Telefono invalido!!!!");
93          sil = false;
94      }
95  }while(!sil);
96  do{
97      System.out.print("Direccion:\t");
98      aux = br.readLine();
99      if(man.validarDir(aux)) {
100          dir = aux;
101          sil = true;
102      }else{
103          System.out.println("Direccion invalido!!!!");
104          sil = false;
105      }
106  }while(!sil);
107 }
108
109 String[] pedirDatos2() throws IOException{
110 /*solicita los datos de un cliente y devuelve un arreglo para utilizar
111 * los datos en otra clase*/
112     String[] datos = new String[4];
113     do{
114         System.out.print("Nombre:\t\t");
115         aux = br.readLine();
116         if(man.validarNombApe(aux)) {
117             datos[0] = aux;

```



```
176     nom = ver[1];
177     System.out.println("\033[30mApellido:\t\033[31m"+ver[2]);
178     ape = ver[2];
179     System.out.println("\033[30mTelefono:\t\033[31m"+ver[3]);
180     telf = ver[3];
181     System.out.println("\033[30mDireccion:\t\033[31m"+ver[4]);
182     dir = ver[4];
183     System.out.println("\033[31mCliente encontrado!!!\033[30m");
184     System.out.println("\033[31m-----");
185     si = true;
186 }else{
187     pedirDatos();
188     String data = ci+"-"+nom+"-"+ape+"-"+telf+"-"+dir;
189     man.guardar("cliente", data);
190     si = true;
191 }
192 }
193 }else{
194     System.out.println("Cedula invalida..");
195     si = false;
196 }
197 }while(!si);
198 }
199 }
```

```

1 /*Creacion y gestion vuelos*/
2 package newairlineseller;
3
4 import java.io.BufferedReader;
5 import java.io.IOException;
6 import java.io.InputStreamReader;
7
8 public class Vuelos {
9     String fechaVuelo, horaVuelo, ciudadOrigen, ciudadDestino, aux, color;
10    double tarifaBase;
11    boolean si, dispon;
12    int cantNaves = 10, disp = 96;
13    Manejador man = new Manejador();
14    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
15
16    void crearVuelo() throws IOException{//Crear vuelo
17        man.crear("vuelo");//Crear archivo vuelo.txt
18        int num = man.buscar("vuelo")+1;//busca en el numero del ultimo vuelo creado y le
19        suma 1 para crear el siguiente vuelo.
20        System.out.println("\033[30mVuelo #\033[31m"+num+"\033[30m.");//Imprime en pantalla
21        el numero de vuelo
22        do{//bucle para verificacion de datos ingresados para la cracion del vuelo
23            System.out.print("Fecha del Vuelo: ");
24            aux = br.readLine();
25            if(man.validarFecha(aux)){//evalua mediante una funcion si el dato fue ingrado
26                fechaVuelo = aux;
27                si = true;
28            }else{
29                System.out.println("Fecha invalida.");
30                si = false;
31            }
32            //si el dato es correcto se carga el valor de la variable auxiliar en la
33            //variable definitiva.
34            //la variable "si" se evalua a verdadero.
35        }while(!si); //El bucle se reperita mientras la variable "si" se evalue falsa.
36        do{//bucle de virificacion del formato de la hora.
37            System.out.print("Hora de salida: ");
38            aux = br.readLine();
39            if(man.validarHora(aux)){
40                horaVuelo = aux;
41                si = true;
42            }else{
43                System.out.println("Hora invalida.");
44            }
45        }while(!si);
46
47        System.out.println("Ciudad de origen: ");
48        ciudadOrigen = man.selecCiudad(); //mediante la fucion selecCiudad carga un codigo de
49        ciudad en la variable correspondiente
50        System.out.println("Ciudad destino: ");

```

```

51 ciudadDestino = man.selecCiudad();
52
53 do{
54     System.out.print("Tarifa base: ");
55     aux = br.readLine();
56     if(man.validarTarifa(aux)){
57         tarifaBase = Double.parseDouble(aux);
58         si = true;
59     }else{
59         System.out.println("Fecha invalida.");
60         si = false;
61     }
62 }
63 while(!si);
64 String data = num+"-"+fechaVuelo+"-"+horaVuelo+"-"+ciudadOrigen
65     +"-"+ciudadDestino+"-"+tarifaBase;
66 man.guardar("vuelo", data);
67 }
68
69 void verVuelos() throws IOException{//muestra los vuelos creados
70     System.out.print("Introduzca el numero del vuelos: ");
71     aux = br.readLine();
72     String[] datos = man.buscar("vuelo", aux);
73     //si la funcion buscar() devuelve un null significa que el archivo esta vacio
74     if(datos!=null){
75         System.out.println("\033[34m|-----|");
76         System.out.println("\033[34m|Vuelo #\033[31m"+datos[0]+\033[34m\t\t\t |");
77         System.out.println("\033[34m|Con fecha: \033[31m"+datos[1]+\033[34m - a las
    \033[31m"+datos[2]+\033[34m |");
78         System.out.println("\033[34m|De \033[31m"+datos[3]+\033[34m a
    \033[31m"+datos[4]+\033[34m\t\t\t |");
79         System.out.println("\033[34m|Con una tatira base de
    \033[31m"+datos[5]+\033[34m\t\t |");
80         System.out.println("\033[34m|-----|");
81     }else{
82         System.out.println("Vuelo no encontrado...!");
83     }
84 }
85
86 }
87
88 boolean verificarVuelo(String n){//verifica la existencia de un vuelo en le archivo
vuelo.txt
89     String[] data = man.buscar("vuelo", n);
90     if(data!=null){
91         if(data[0].equals(n)){
92             return true;
93         }else{
94             return false;
95         }
96     }else{
97         return false;
98     }
99 }
100
101 String[] infoVuelo(String n){//envia un arreglo con la informacion de un buelo buscado
102     String[] data;
103     data = man.buscar("vuelo",n);
104     return data;
105 }
```

```

106
107     void disponibilidad(String numV) {
108 /*imprime en pantalla una matriz con la disposicion de los diferentes asientos
109 * en un vuelo. Se incican en color negro los asientos disponibles, y en
110 * color rojo los asientos ocupados*/
111     int i,j,x;
112
113     for(x=1; x<=16; x++) {
114         if(x<=9) {
115             System.out.print("\033[32m |"+x+"| ");
116         }else{
117             System.out.print("\033[32m|"+x+"| ");
118         }
119     }
120
121     System.out.println();
122     System.out.println("\033[32m-----");
-----");
123     for (i=0;i<=5;i++){
124         if(i==3){
125             System.out.println();
126         }
127         for (j=0;j<=15;j++) {
128             //funcion de ver disponibilidad
129             if(man.verDispon(j, i, numV)){
130                 dispon = false;
131             }else{
132                 dispon = true;
133             }
134             if(dispon){
135                 disp--;
136                 color = "\033[31m";
137             }else{
138                 color = "\033[30m";
139             }
140             switch(i){
141                 case 0 : System.out.print(color+" |A| "); break;
142                 case 1 : System.out.print(color+" |B| "); break;
143                 case 2 : System.out.print(color+" |C| "); break;
144                 case 3 : System.out.print(color+" |D| "); break;
145                 case 4 : System.out.print(color+" |E| "); break;
146                 case 5 : System.out.print(color+" |F| "); break;
147             }
148         }
149         System.out.println();
150     }
151     System.out.println("\033[32m-----");
-----");
152     System.out.println("Lugares disponibles: \033[31m"+disp);
153     disp = 96;
154 }
155
156 }
```

```

1 /*Gestion y venta de boletos*/
2 package newairlineseller;
3
4 import java.io.BufferedReader;
5 import java.io.File;
6 import java.io.IOException;
7 import java.io.InputStreamReader;
8
9
10 public class Boleto {
11     String nom, ape, ci, dir, telf, nVuel, fVuelo, hVuelo, nBoleto,
12         clase, ciudadOrigen, ciudadDestino, aux, asiento;
13     boolean si;
14     double tarifa, base;
15     int fil, col;
16     Manejador man = new Manejador();
17     Vuelos v = new Vuelos();
18     Cliente c = new Cliente();
19     BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
20
21     double tarifaBase(String nV) {
22 //busca en el archivo vuelo la tarifa base correspondiente a un vuelo especifico
23         double b;
24         String[] vl = man.buscar("vuelo", nV);
25         b = Double.parseDouble(vl[5]);
26         return b;
27     }
28
29     void crearBoleto() throws IOException{
30 /*inicia la recoleccion de datos para la cracion del boleto y su
31 * correspondiente registro en archivo*/
32         String nomArch = "boletos";
33 /*se inicializa la variable para evitar que se cree un archivo con
34 * nombre incorrecto*/
35         si = false;
36
37         do{
38             System.out.print("Nuemro de vuelo a vender: ");
39             aux = br.readLine();
40             if(v.verificarVuelo(aux)) {
41                 nVuel = aux;
42                 nomArch = "boletos"+nVuel;
43 /*se crea un archivo con el nombre boleto y la referencia del numero de vuelo*/
44                 man.crear(nomArch);
45                 nBoleto = Integer.toString(man.buscar(nomArch)+1);
46                 System.out.println("Venta de boleto para el vuelo nº
\033[31m"+nVuel+"\033[30m.");
47                 System.out.println("\033[30mBoleto #\033[31m"+nBoleto+"\033[30m.");
48                 System.out.print("Cedula: ");
49                 ci = br.readLine();
50                 String[] ver = man.buscar("cliente", ci);
51
52                 if(ver!=null){
53                     if(ver[0].equals(ci)){
54                         System.out.println("\033[30mNombre:\t\t\033[31m"+ver[1]);
55                         nom = ver[1];
56                         System.out.println("\033[30mApellido:\t\033[31m"+ver[2]);
57                         ape = ver[2];

```

```

58         System.out.println("\033[30mTelefono:\t\033[31m"+ver[3]);
59         telf = ver[3];
60         System.out.println("\033[30mDireccion:\t\033[31m"+ver[4]);
61         dir = ver[4];
62         System.out.println("\033[31mCliente encontrado!!!\033[30m");
63
64         si = true;
65     }else{
66         String[] datos = c.pedirDatos2();
67         nom = datos[0]; ape = datos[1]; telf = datos[2]; dir = datos[3];
68         String data = ci+"-"+nom+"-"+ape+"-"+telf+"-"+dir;
69         man.guardar("cliente", data);
70         si = true;
71     }
72
73 }else{
74     String[] datos = c.pedirDatos2();
75     nom = datos[0]; ape = datos[1]; telf = datos[2]; dir = datos[3];
76     String data = ci+"-"+nom+"-"+ape+"-"+telf+"-"+dir;
77     man.guardar("cliente", data);
78     si = true;
79 }
80 }else{
81     System.out.print("No se ha creado el vuelo.\n"
82                     + "Desea crear un nuevo vuelo? (s/n): ");
83     String aux1 = br.readLine();
84     if(aux1.equals("s")){
85         v.crearVuelo();
86         si = false;
87     }else{
88         si = true;
89     }
90 }
91 }while(!si);
92
93 v.disponibilidad(nVuel);
94
95 System.out.println("Asiento");
96 do{
97     System.out.print("Linea: ");
98     aux = br.readLine();
99     fil = Integer.parseInt(aux);
100    tarifa = tarifaBase(nVuel);
101    if((fil>0)&&(fil<17)){
102        if(fil<3){
103            tarifa += tarifa*0.25; clase = "A";
104            si = true;
105        }
106        if((fil>=3)&&(fil<8)){
107            tarifa += tarifa*0.15; clase = "B";
108            si = true;
109        }
110        if(fil>=8){
111            tarifa = tarifaBase(nVuel); clase = "C";
112            si = true;
113        }
114    }else{
115        System.out.println("Linea invalida");
116        si = false;

```

```

117         }
118     }while(!si);
119
120     do{
121         System.out.print("Letra: ");
122         aux = br.readLine();
123         asiento = fil+aux.toUpperCase();
124         switch(aux.toLowerCase()){
125             case "a" : col = 0; si=true; break;
126             case "b" : col = 1; si=true; break;
127             case "c" : col = 2; si=true; break;
128             case "d" : col = 3; si=true; break;
129             case "e" : col = 4; si=true; break;
130             case "f" : col = 5; si=true; break;
131             default:
132                 System.out.println("Letra invalida.");
133                 si = false; break;
134         }
135
136     }while(!si);
137
138     String data = nBoleto+"-"+nVuel+"-"+ci+"-"+(fil-1)+"-"+col+"-"+clase+"-"+tarifa;
139     man.guardar(nomArch, data);
140
141     imprimirBoleto(nBoleto, nVuel, ci);
142 }
143
144 void imprimirBoleto(String Bol, String Vuel, String CI){
145     String[] vuelo= man.buscar("vuelo", Vuel);
146     String[] cliente= man.buscar("cliente", CI);
147     System.out.println("\033[31m-----\n-----");
148
149     System.out.println("\033[31mBoleto\t#\033[30m"+Bol+".\t\t\t\033[31mVuelo\t#\033[30m"+nVuel);
150     System.out.println("\033[31mNombre:\t\033[30m"+cliente[2]+",
151 "+cliente[1]+\t\t\033[31mFecha de Vuelo:\t\033[30m"+vuelo[1]);
152     System.out.println("\033[31mCedula:\t\033[30m"+cliente[0]+\t\t\033[31mHora de
153 Vuelo:\t\033[30m"+vuelo[2]+".");
154
155     System.out.println("\033[31mClase:\t\033[30m"+clase+"\t\033[31mAsiento:\033[30m"+asiento+"\t
156 \033[31mPrecio:\t\033[30m"+tarifa);
157     System.out.println("\033[31mCiudad origen:\033[30m"+vuelo[3]+\t\033[31mCiudad
158 destino:\t\033[30m"+vuelo[4]);
159     System.out.println("\033[31m-----\n-----");
160
161
162     void imprimirBoleto() throws IOException{
163         int i = 1;
164         System.out.print("Introduzca numero de vuelo a consultar: ");
165         String n = br.readLine();
166         String[] vuelo= man.buscar("vuelo", n);
167
168         while(true){
169             if(vuelo==null){
170                 System.out.println("\033[31m-----\n-----");
171                 System.out.println("El vuelo #"+n+" no existe.");
172                 break;
173             }

```

```

168     String[] boleto = man.buscar("boletos"+n, Integer.toString(i));
169     i++;
170     if(boleto!=null) {
171         String[] cliente= man.buscar("cliente", boleto[2]);
172         String aux1 = "";
173
174         switch(boleto[4]){
175             case "0" : aux1 = "A"; si=true; break;
176             case "1" : aux1 = "B"; si=true; break;
177             case "2" : aux1 = "C"; si=true; break;
178             case "3" : aux1 = "D"; si=true; break;
179             case "4" : aux1 = "E"; si=true; break;
180             case "5" : aux1 = "F"; si=true; break;
181         }
182
183         String lugar = (Integer.parseInt(boleto[3])+1)+aux1+"";
184         System.out.println("\033[31m-----");
185         System.out.println("\033[31mBoleto\t#\033[30m"+boleto[1]+"\t\t\t\033[31mVuelo\t#\033[30m"+n);
186         System.out.println("\033[31mNombre:\t\033[30m"+cliente[2]+",
187 "+cliente[1]+\t\033[31mFecha de Vuelo:\t\033[30m"+vuelo[1]);
188         System.out.println("\033[31mCedula:\t\033[30m"+cliente[0]+"\t\t\033[31mHora
189 de Vuelo:\t\033[30m"+vuelo[2]+".");
190         System.out.println("\033[31mClase:\t\033[30m"+boleto[5]+\t\033[31mAsiento:\033[30m"+lugar+
191 \t\033[31mPrecio:\t\033[30m"+boleto[6]);
192         System.out.println("\033[31mCiudad
193 origen:\033[30m"+vuelo[3]+\t\033[31mCiudad destino:\t\033[30m"+vuelo[4]);
194     } else{
195         System.out.println("\033[31m-----");
196     }
197 }
```

```

1 /*Generacion de reportes de ventas de boletos*/
2 package newairlineseller;
3
4 import java.io.File;
5 import java.io.IOException;
6
7 public class Reportes {
8     Manejador man = new Manejador();
9
10    double[] sumarMontos() throws IOException{
11        double[] monto = {0,0,0,0,0,0,0,0}, datos;
12        int i=1;
13
14        while(new File("boletos"+i+".txt").exists()) {
15            datos = man.buscarMontos(i);
16            monto[0] += datos[0];//total general
17            monto[1] += datos[1];//total clase A
18            monto[2] += datos[2];//total clase B
19            monto[3] += datos[3];//total clase C
20            monto[4] += datos[4];//total boletos
21            monto[5] += datos[5];//total boletos A
22            monto[6] += datos[6];//total boletos B
23            monto[7] += datos[7];//total boletos C
24
25            i++;
26        }
27        return monto;
28    }
29
30    void imprimirTotal() throws IOException{
31        double[] total = sumarMontos();
32        System.out.println("\033[30mEl monto total recaudado en \033[31m"+(int)total[4]
33                                +"\033[30m boletos vendidos es: \033[31m"+total[0]+\033[30m");
34        System.out.println("\033[30mEl monto total recaudado en la clase A con \033[31m"
35                                +(int)total[5]+\033[30m boletos es: \033[31m"+total[1]+\033[30m");
36        System.out.println("\033[30mEl monto total recaudado en la clase B con \033[31m"
37                                +(int)total[6]+\033[30m boletos es: \033[31m"+total[2]+\033[30m");
38        System.out.println("\033[30mEl monto total recaudado en la clase C con \033[31m"
39                                +(int)total[7]+\033[30m boletos es: \033[31m"+total[3]+\033[30m");
40    }
41
42    void vueloVendidos() throws IOException{
43        double[] datos;
44        int i = 1;
45        int v = man.buscar("vuelo");
46        System.out.println("\033[36m-----\033[30m");
47        System.out.println("Existen \033[31m"+v+"\033[30m vuelos creados.");
48        while (new File("boletos"+i+".txt").exists()) {
49            datos = man.buscarMontos(i);
50            System.out.println("\033[36m-----\033[30m");
51            System.out.println("Para el vuelo #\033[31m"+i+"\033[30m, se vendieron:");
52            System.out.println("Clase A \033[31m"+(int)datos[5]+\033[30m boletos, que suman
53 : \033[31m"+datos[1]+\033[30m Bs.");
54            System.out.println("Clase B \033[31m"+(int)datos[6]+\033[30m boletos, que suman
55 : \033[31m"+datos[2]+\033[30m Bs.");
56            System.out.println("Clase C \033[31m"+(int)datos[7]+\033[30m boletos, que suman
57 : \033[31m"+datos[3]+\033[30m Bs.");
58            i++;
59        }
60    }

```

```
57         System.out.println("\033[36m-----\033[30m");
58     }
59 }
```