

```
1 package E706;
2
3 import java.awt.EventQueue;
4
5 import javax.swing.JFrame;
6 import javax.swing.JButton;
7 import javax.swing.JFileChooser;
8
9 import java.awt.Font;
10 import java.awt.Graphics;
11 import java.awt.Graphics2D;
12 import java.awt.List;
13 import java.awt.event.ActionEvent;
14 import java.awt.event.ActionListener;
15 import java.awt.geom.Line2D;
16 import java.io.BufferedReader;
17 import java.io.File;
18 import java.io.FileNotFoundException;
19 import java.io.FileReader;
20 import java.io.IOException;
21 import java.util.ArrayList;
22
23 import javax.swing.JTextField;
24 import javax.swing.filechooser.FileNameExtensionFilter;
25 import javax.swing.JLabel;
26 import javax.swing.JPanel;
27 import java.awt.Color;
28
29 public class E706 {
30
31     private JFrame frame;
32     private BufferedReader br = null;
33     private File fichero;
34     private int[] notas = new int[11];
35     private int[] porcentaje = new int[11];
36     private ArrayList<Double> list = new ArrayList<Double>(); //para almacenar las notas para la desviacion
37     private int contador=0;
38     private double nota;
39     private String palabra ="";
40     private double notaTotal=0;
41     private JPanel panel;
42     private double desviacion;
43     private double sumaDesv=0;
44
45     /**
46      * Launch the application.
47      */
48     public static void main(String[] args) {
49         EventQueue.invokeLater(new Runnable() {
50             public void run() {
51                 try {
```

```

52         E706 window = new E706();
53         window.frame.setVisible(true);
54     } catch (Exception e) {
55         e.printStackTrace();
56     }
57 }
58 });
59 }
60
61 /**
62  * Create the application.
63  */
64 public E706() {
65     initialize();
66 }
67
68 /**
69  * Initialize the contents of the frame.
70  */
71 private void initialize() {
72     frame = new JFrame();
73     frame.setResizable(false);
74     frame.setBounds(100, 100, 454, 371);
75     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
76     frame.getContentPane().setLayout(null);
77
78     JButton btAbrir = new JButton("Abrir Fichero");
79     btAbrir.setFont(new Font("Tahoma", Font.BOLD, 12));
80     btAbrir.setBounds(10, 11, 142, 23);
81     frame.getContentPane().add(btAbrir);
82
83     JLabel lbFichero = new JLabel("");
84     lbFichero.setFont(new Font("Tahoma", Font.BOLD, 12));
85     lbFichero.setBounds(162, 11, 201, 23);
86     frame.getContentPane().add(lbFichero);
87
88     panel = new JPanel() {
89         public void paint(Graphics g) {
90             super.paint(g);
91             g.setColor(Color.black);
92
93             //ejes horizontal y vertical
94             g.drawLine(30, 250, 360, 250);
95             g.drawLine(30, 250, 30, 16);
96
97             //marcas eje horizontal
98             int ancho = 60;
99             for(int i = 0; i <= 10; i++) {
100                 g.drawString(Integer.toString(i), ancho-20, 266);
101                 g.drawLine(ancho, 248, ancho, 252);
102                 ancho += 30;

```

```

103     }
104
105     //marcas eje vertical
106     ancho = 250;
107     for(int i = 0; i <= 26; i++) {
108         if(i== 0 || i == 10 || i == 20 || i == 26) {
109             g.drawString(Integer.toString(i) + "%", 0, ancho+5);
110             g.drawLine(26, ancho, 34, ancho);
111         }
112         else g.drawLine(28, ancho, 32, ancho);
113         ancho -= 9;
114     }
115
116     }
117 };
118 panel.setBackground(Color.WHITE);
119 panel.setBounds(10, 45, 415, 276);
120 frame.getContentPane().add(panel);
121
122 //listener para el jfilechooser
123 btAbrir.addActionListener(new ActionListener() {
124     public void actionPerformed(ActionEvent e) {
125         //buscamos el fichero
126         fichero = leerFichero();
127         //ponemos nombre del fichero
128         lbFichero.setText(fichero.getName());
129         //recorremos el fichero y hacemos cosas con él
130         recorrerNotas(fichero);
131         //rellenamos con el grafico
132         rellenarGrafico();
133     }
134 });
135
136 }
137
138 public static File leerFichero() {
139
140     JFileChooser fd = new JFileChooser();
141     fd.setAcceptAllFileFilterUsed(false);
142     FileNameExtensionFilter filter = new FileNameExtensionFilter("archivos CSV", "csv");
143     fd.addChoosableFileFilter(filter);
144     fd.setDialogTitle("Selecciona el fichero a leer");
145     fd.setSelectedFile(null);
146     int opcion = fd.showOpenDialog(null);
147
148     if (opcion != JFileChooser.APPROVE_OPTION) return null;
149
150     File f = fd.getSelectedFile();
151
152     return f;
153

```

```
154     }
155
156     public void recorrerNotas(File fichero) {
157
158         //rellenamos el array de notas a 0
159         for(int i = 0; i < 11; i++) {
160             notas[i] = 0;
161         }
162
163         try {
164             //nos creamos un filereader envuelto en un buffer de lectura
165             br = new BufferedReader(new FileReader(fichero));
166
167             //leemos la primera linea
168             String linea = br.readLine();
169
170             //mientras hayan lineas que leer
171             while (linea != null) {
172
173                 //vamos contando lineas leidas
174                 contador++;
175
176                 //construimos un array con los campos de la linea. Separamos los campos con split
177                 String [] fields = linea.split(";");
178
179                 //comprobamos que cada linea tiene los dos campos correctos
180                 if(fields.length == 2) {
181                     try {
182                         //comprobamos la conversion
183                         nota = Double.parseDouble(fields[1]);
184                     } catch (NumberFormatException exception) {
185                         //como nos vamos a encontrar con comas en vez de puntos, reconstruimos la linea
186                         //ademas elimino unas comillas que aparecen al final de cada linea
187                         //primero vemos si hay un 10
188                         if (fields[1].charAt(0) == 49 && fields[1].charAt(1) == 48 ) { //es un 10
189                             palabra = palabra + fields[1].charAt(0) + fields[1].charAt(1) + "." + fields[1].charAt(3) +
190                                 fields[1].charAt(4);
191                         }
192                         //no es un 10
193                         else palabra = palabra + fields[1].charAt(0) + "." + fields[1].charAt(2) + fields[1].charAt(3);
194                         //con la nueva linea correctamente creada, ya podemos hacer la conversion
195                         nota = Double.parseDouble(palabra);
196
197                         list.add(nota);
198
199                         notaTotal += nota; //almacenamos en el total de notas
200                         palabra = ""; //inicializamos la cadena
201                     }
202
203                     //vamos a incrementar el valor en la posicion correspondiente del array segun la nota
```

```

204         //obtenemos la parte entera
205         int num = (int)nota;
206
207         //incrementamos la posicion en el array
208         notas[num]++;
209
210         //calcular porcentajes de cada nota
211         for(int i = 0; i < 11; i++) {
212             porcentaje[i] = notas[i]*100/contador;
213         }
214
215
216
217     }
218
219     //leemos la siguiente linea
220     linea = br.readLine();
221 }
222
223 } catch (FileNotFoundException e) {
224     System.out.println(e.getMessage());
225 } catch (IOException e) {
226     System.out.println(e.getMessage());
227 }
228
229 finally { //cerramos cosas
230     // br.close();
231 }
232 }
233
234 //este metodo permite crear la grafica de barras despues de todos los calculos
235 public void rellenarGrafico() {
236
237     Graphics gr = panel.getGraphics();
238     gr.setColor(Color.GREEN);
239
240     //pintamos
241     int ejeX = 32;
242     for(int i = 0; i <= 10; i++) {
243         gr.fillRect(ejeX, 250-porcentaje[i]*9, 26, porcentaje[i]*9);
244         ejeX += 30;
245     }
246
247     gr.setColor(Color.black);
248
249     gr.drawString("Media:" + notaTotal/contador, 40, 10);
250
251     for(int i = 0; i < list.size(); i++) {
252         sumaDesv = sumaDesv + Math.pow((list.get(i)-notaTotal/contador),2);
253     }
254

```

```
255         sumaDesv = sumaDesv/contador;
256         desviacion = Math.sqrt(sumaDesv);
257
258         gr.drawString("Desviacion: " + String.valueOf(desviacion), 200, 10);
259     }
260 }
261
```