

```
1 package E701;
2
3 import java.awt.EventQueue;
4 import java.awt.GridLayout;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7
8 import javax.swing.JButton;
9 import javax.swing.JFrame;
10 import javax.swing.JLabel;
11 import javax.swing.JTextField;
12
13
14 public class E701 {
15
16     //variables globales
17     public static String valor = ""; //recoge el valor del texto introducido
18     public static Double valorEuros; //pasamos el valor a un double
19     public static boolean flag = false; //nos indica si podemos hacer la conversion a ptas
20     public static String patron = ""; //cadena a mostrar inicial
21     public static String cadena = ""; //cadena formateada. Muestra el importe convertido a ptas
22
23     public static void main(String[] args) {
24
25         EventQueue.invokeLater(
26             new Runnable() {
27                 public void run() {
28
29                     //creamos el JFrame
30                     JFrame frame = new JFrame();
31
32                     //configuramos el JFrame
33                     frame.setTitle("Conversor Jorge Victoria Andreu");
34                     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
35                     frame.setSize(400, 280);
36                     frame.setVisible(true);
37
38                     //creamos el layout del JFrame
39                     frame.getContentPane().setLayout(new GridLayout(9,1));
40
41                     //fila1
42                     JLabel label1 = new JLabel("Importe en euros", JLabel.CENTER);
43                     frame.getContentPane().add(label1);
44
45                     //fila2
46                     JLabel label2 = new JLabel();
47                     frame.getContentPane().add(label2);
48
49                     //fila3
50                     JTextField tfImporteEuros = new JTextField();
51                     frame.getContentPane().add(tfImporteEuros);
```

```

52
53 //fila4
54 JLabel label3 = new JLabel();
55 frame.getContentPane().add(label3);
56
57 //fila5
58 JButton btConvertir = new JButton("Convertir");
59 frame.getContentPane().add(btConvertir);
60
61 //fila6
62 JLabel label4 = new JLabel();
63 frame.getContentPane().add(label4);
64
65 //fila7
66 JLabel label5 = new JLabel("Pulse para obtener el importe en pesetas", JLabel.CENTER);
67 frame.getContentPane().add(label5);
68
69 //fila8
70 JLabel label6 = new JLabel();
71 frame.getContentPane().add(label6);
72
73 //fila9
74 JLabel lbImportePtas = new JLabel("0 pts", JLabel.CENTER);
75 frame.getContentPane().add(lbImportePtas);
76
77 //listener para el boton convertir
78 btConvertir.addActionListener(new ActionListener() {
79     public void actionPerformed(ActionEvent e) {
80
81         //recogemos el valor
82         valor = tfImporteEuros.getText();
83
84         //si no hay nada, lo anunciamos
85         if(valor.equals("")) lbImportePtas.setText("Introduzca un valor");
86         else {
87             //comprobamos que el valor de la cadena se puede pasar a numero
88             try {
89                 valorEuros = Double.parseDouble(valor);
90                 //sino se puede convertir, mostramos mensaje en pantalla
91             } catch(NumberFormatException ex) {
92                 lbImportePtas.setText("valor incorrecto");
93                 flag = true;
94             }
95
96             //se puede convertir, convertimos y mostramos el valor
97             if(!flag) {
98                 patron = "%.1f ptas";
99                 cadena = String.format(patron, valorEuros*166.6);
100                 lbImportePtas.setText(cadena);
101             }
102             else flag = false; //para volver a introducir datos

```

```
103         }
104         } //fin actionpertformed
105     }); //fin actionlistener
106     } //fin run
107 } //fin runnable
108 ); //fin eventqueue
109
110     } //fin main
111
112 } //fin clase
113
```

```

1  package E702;
2
3  import java.awt.Color;
4  import java.awt.EventQueue;
5
6  import javax.swing.JFrame;
7  import javax.swing.JLabel;
8  import javax.swing.SwingConstants;
9  import java.awt.Font;
10 import java.awt.event.ActionEvent;
11 import java.awt.event.ActionListener;
12
13 import javax.swing.JTextField;
14 import javax.swing.JButton;
15
16 public class E702 {
17
18     public static String valor = "";           //almacena el valor introducido
19     public static boolean flag = false;        //para ver si se puede realizar la conversion
20     public static Double valorPtas;           //para pasar de ptas a euros
21     public static String patron = "";         //patron de la cadena a mostrar
22     public static String cadena = "";         //cadena formateada
23
24     private JFrame frmConversorJorgeVictoria; //el JFrame creado
25     private JTextField tfImportePtas;        //el JTextField creado
26
27     /**
28      * Launch the application.
29      */
30     public static void main(String[] args) {
31         EventQueue.invokeLater(new Runnable() {
32             public void run() {
33                 try {
34                     E702 window = new E702();
35                     window.frmConversorJorgeVictoria.setVisible(true);
36                 } catch (Exception e) {
37                     e.printStackTrace();
38                 }
39             }
40         });
41     }
42
43     /**
44      * Create the application.
45      */
46     public E702() {
47         initialize();
48     }
49
50     /**
51      * Initialize the contents of the frame.

```

```

52 */
53 private void initialize() {
54     //Jframe
55     frmConversorJorgeVictoria = new JFrame();
56     frmConversorJorgeVictoria.setTitle("Conversor Jorge Victoria Andreu");
57     frmConversorJorgeVictoria.getContentPane().setFont(new Font("Comic Sans MS", Font.BOLD, 11));
58     frmConversorJorgeVictoria.setBounds(100, 100, 450, 300);
59     frmConversorJorgeVictoria.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
60     frmConversorJorgeVictoria.getContentPane().setLayout(null);
61
62     //Jlabel
63     JLabel lbl = new JLabel("Importe en pesetas");
64     lbl.setFont(new Font("Comic Sans MS", Font.BOLD, 12));
65     lbl.setHorizontalAlignment(SwingConstants.CENTER);
66     lbl.setBounds(134, 37, 154, 14);
67     frmConversorJorgeVictoria.getContentPane().add(lbl);
68
69     //campo de texto para introducir las ptas
70     tfImportePtas = new JTextField();
71     tfImportePtas.setHorizontalAlignment(SwingConstants.RIGHT);
72     tfImportePtas.setBounds(74, 78, 280, 20);
73     frmConversorJorgeVictoria.getContentPane().add(tfImportePtas);
74     tfImportePtas.setColumns(10);
75
76     //Jlabel
77     JLabel lbl2 = new JLabel("Pulse para convertir a texto");
78     lbl2.setFont(new Font("Comic Sans MS", Font.BOLD, 12));
79     lbl2.setHorizontalAlignment(SwingConstants.CENTER);
80     lbl2.setBounds(124, 117, 184, 20);
81     frmConversorJorgeVictoria.getContentPane().add(lbl2);
82
83     //boton para convertir
84     JButton btConvertir = new JButton("Convertir");
85     btConvertir.setFont(new Font("Comic Sans MS", Font.BOLD, 12));
86     btConvertir.setBounds(134, 158, 174, 23);
87     frmConversorJorgeVictoria.getContentPane().add(btConvertir);
88
89     //zona donde se muestra la conversion a euros
90     JLabel lblImporteEuros = new JLabel("0 Euros");
91     lblImporteEuros.setFont(new Font("Comic Sans MS", Font.BOLD, 12));
92     lblImporteEuros.setHorizontalAlignment(SwingConstants.CENTER);
93     lblImporteEuros.setBounds(165, 216, 123, 14);
94     frmConversorJorgeVictoria.getContentPane().add(lblImporteEuros);
95
96     //listener para el boton convertir
97     btConvertir.addActionListener(new ActionListener() {
98         public void actionPerformed(ActionEvent e) {
99
100             //recogemos el valor
101             valor = tfImportePtas.getText();
102

```

```
103         //si no hay nada, lo anunciamos, cambiamos el color de fondo y el color de la fuente
104         if(valor.equals("")) {
105             tfImportePtas.setBackground(Color.red);
106             lbImporteEuros.setForeground(Color.red);
107             lbImporteEuros.setText("Introduzca un valor");
108         }
109         else {
110             //comprobamos que el valor de la cadena se puede pasar a numero
111             try {
112                 valorPtas = Double.parseDouble(valor);
113                 //sino se puede convertir, mostramos mensaje en pantalla
114             } catch(NumberFormatException ex) {
115                 lbImporteEuros.setText("valor incorrecto");
116                 tfImportePtas.setBackground(Color.red);
117                 lbImporteEuros.setForeground(Color.red);
118                 flag = true;
119             } //fin try catch
120
121             //se puede convertir, convertimos y mostramos el valor
122             if(!flag) {
123                 patron = "%.2f Euros";
124                 cadena = String.format(patron, valorPtas/166.6);
125                 tfImportePtas.setBackground(Color.white);
126                 lbImporteEuros.setForeground(Color.black);
127                 lbImporteEuros.setText(cadena);
128             }
129             else flag = false; //para volver a introducir datos
130         } //fin else
131
132     } //fin actionPerformed
133 }); //fin listener
134
135
136     } //fin initialize
137 } //fin clase
138
139
```

```

1  package E703;
2
3  import java.awt.EventQueue;
4
5  import javax.swing.JFrame;
6  import java.awt.BorderLayout;
7  import javax.swing.JTextField;
8  import java.awt.Font;
9  import javax.swing.SwingConstants;
10 import java.awt.Color;
11 import javax.swing.border.EmptyBorder;
12 import javax.swing.border.LineBorder;
13 import javax.swing.JPanel;
14 import java.awt.GridLayout;
15 import java.awt.FlowLayout;
16 import javax.swing.JButton;
17 import java.awt.CardLayout;
18 import java.awt.GridBagLayout;
19 import java.awt.GridBagConstraints;
20 import java.awt.Insets;
21 import java.awt.event.ActionListener;
22 import java.awt.event.ActionEvent;
23 import java.awt.event.MouseAdapter;
24 import java.awt.event.MouseEvent;
25 import javax.swing.ButtonGroup;
26
27 public class E703 {
28
29     //variables
30     private JFrame frmCalculadoraJorgeVictoria; //el JFrame sobre el que vamos a trabajar
31     private JTextField tfPantalla; //pantalla de la calculadora
32     private String valorA = ""; //primer operando
33     private String valorB = ""; //segundo operando
34     private String signo = ""; //signo de la operacion
35     private boolean bloqueoPantalla = false; //cuando imprimimos el resultado, se bloquea la pantalla
36     private double potenciaDouble; //para calcular la potencia cuando el numero tiene decimales
37     private int potenciaInteger; //para calcular la potencia cuando el numero es un entero
38     private double raizCuadrada; //calcula la raiz cuadrada
39
40     /**
41      * Launch the application.
42      */
43     public static void main(String[] args) {
44         EventQueue.invokeLater(new Runnable() {
45             public void run() {
46                 try {
47                     E703 window = new E703();
48                     window.frmCalculadoraJorgeVictoria.setVisible(true);
49                 } catch (Exception e) {
50                     e.printStackTrace();
51                 }

```

```

52         }
53     });
54 }
55
56 /**
57  * Create the application.
58  */
59 public E703() {
60     initialize();
61 }
62
63 /**
64  * Initialize the contents of the frame.
65  */
66 private void initialize() {
67
68     //Jframe y sus parametros
69     frmCalculadoraJorgeVictoria = new JFrame();
70     frmCalculadoraJorgeVictoria.setResizable(false);
71     frmCalculadoraJorgeVictoria.setTitle("Calculadora Jorge Victoria");
72     frmCalculadoraJorgeVictoria.setBounds(100, 100, 455, 300);
73     frmCalculadoraJorgeVictoria.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
74     frmCalculadoraJorgeVictoria.getContentPane().setLayout(null);
75
76     //pantalla de la calculadora
77     tfPantalla = new JTextField();
78     tfPantalla.setBorder(new LineBorder(new Color(0, 0, 0)));
79     tfPantalla.setEditable(false);
80     tfPantalla.setBackground(new Color(154, 205, 50));
81     tfPantalla.setHorizontalAlignment(SwingConstants.RIGHT);
82     tfPantalla.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
83     tfPantalla.setText("0");
84     tfPantalla.setBounds(10, 0, 420, 50);
85     frmCalculadoraJorgeVictoria.getContentPane().add(tfPantalla);
86     tfPantalla.setColumns(10);
87
88     //boton para el digito 7
89     JButton bt7 = new JButton("7");
90     bt7.setBorder(new LineBorder(new Color(0, 0, 0)));
91     bt7.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
92     bt7.setBounds(10, 61, 80, 40);
93     frmCalculadoraJorgeVictoria.getContentPane().add(bt7);
94
95     //boton para el digito 8
96     JButton bt8 = new JButton("8");
97     bt8.setBorder(new LineBorder(new Color(0, 0, 0)));
98     bt8.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
99     bt8.setBounds(95, 61, 80, 40);
100    frmCalculadoraJorgeVictoria.getContentPane().add(bt8);
101
102    //boton para el digito 9

```



```
103 JButton bt9 = new JButton("9");
104 bt9.setBorder(new LineBorder(new Color(0, 0, 0)));
105 bt9.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
106 bt9.setBounds(180, 61, 80, 40);
107 frmCalculadoraJorgeVictoria.getContentPane().add(bt9);
108
109 //boton para borrar digito
110 JButton btDel = new JButton("DEL");
111 btDel.setToolTipText("Borrar Digito");
112 btDel.setFont(new Font("Comic Sans MS", Font.BOLD, 10));
113 btDel.setBorder(new LineBorder(new Color(0, 0, 0)));
114 btDel.setBackground(new Color(255, 215, 0));
115 btDel.setBounds(265, 61, 80, 40);
116 frmCalculadoraJorgeVictoria.getContentPane().add(btDel);
117
118 //boton para limpiar la pantalla
119 JButton btLimpiar = new JButton("CL");
120 btLimpiar.setToolTipText("Limpiar Pantalla");
121 btLimpiar.setFont(new Font("Comic Sans MS", Font.BOLD, 10));
122 btLimpiar.setBorder(new LineBorder(new Color(0, 0, 0)));
123 btLimpiar.setBackground(new Color(255, 215, 0));
124 btLimpiar.setBounds(350, 61, 80, 40);
125 frmCalculadoraJorgeVictoria.getContentPane().add(btLimpiar);
126
127 //boton para el digito 4
128 JButton bt4 = new JButton("4");
129 bt4.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
130 bt4.setBorder(new LineBorder(new Color(0, 0, 0)));
131 bt4.setBounds(10, 112, 80, 40);
132 frmCalculadoraJorgeVictoria.getContentPane().add(bt4);
133
134 //boton para el digito 5
135 JButton bt5 = new JButton("5");
136 bt5.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
137 bt5.setBorder(new LineBorder(new Color(0, 0, 0)));
138 bt5.setBounds(95, 112, 80, 40);
139 frmCalculadoraJorgeVictoria.getContentPane().add(bt5);
140
141 //boton para el digito 6
142 JButton bt6 = new JButton("6");
143 bt6.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
144 bt6.setBorder(new LineBorder(new Color(0, 0, 0)));
145 bt6.setBounds(180, 112, 80, 40);
146 frmCalculadoraJorgeVictoria.getContentPane().add(bt6);
147
148 //boton para la operacion suma
149 JButton btSuma = new JButton("+");
150 btSuma.setToolTipText("Sumar");
151 btSuma.setForeground(new Color(255, 255, 255));
152 btSuma.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
153 btSuma.setBorder(new LineBorder(new Color(0, 0, 0)));
```

```
154 btsuma.setBackground(new Color(0, 0, 255));
155 btsuma.setBounds(265, 112, 80, 40);
156 frmCalculadoraJorgeVictoria.getContentPane().add(btsuma);
157
158 //boton para la operacion resta
159 JButton btresta = new JButton("-");
160 btresta.setToolTipText("Restar");
161 btresta.setForeground(new Color(255, 255, 255));
162 btresta.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
163 btresta.setBorder(new LineBorder(new Color(0, 0, 0)));
164 btresta.setBackground(new Color(0, 0, 255));
165 btresta.setBounds(350, 112, 80, 40);
166 frmCalculadoraJorgeVictoria.getContentPane().add(btresta);
167
168 //boton para el digito 1
169 JButton bt1 = new JButton("1");
170 bt1.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
171 bt1.setBorder(new LineBorder(new Color(0, 0, 0)));
172 bt1.setBounds(10, 163, 80, 40);
173 frmCalculadoraJorgeVictoria.getContentPane().add(bt1);
174
175 //boton para el digito 2
176 JButton bt2 = new JButton("2");
177 bt2.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
178 bt2.setBorder(new LineBorder(new Color(0, 0, 0)));
179 bt2.setBounds(95, 163, 80, 40);
180 frmCalculadoraJorgeVictoria.getContentPane().add(bt2);
181
182 //boton para el digito 3
183 JButton bt3 = new JButton("3");
184 bt3.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
185 bt3.setBorder(new LineBorder(new Color(0, 0, 0)));
186 bt3.setBounds(180, 163, 80, 40);
187 frmCalculadoraJorgeVictoria.getContentPane().add(bt3);
188
189 //boton para la operacion producto
190 JButton btproducto = new JButton("*");
191 btproducto.setToolTipText("Producto");
192 btproducto.setForeground(new Color(255, 255, 255));
193 btproducto.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
194 btproducto.setBorder(new LineBorder(new Color(0, 0, 0)));
195 btproducto.setBackground(new Color(0, 0, 255));
196 btproducto.setBounds(265, 163, 80, 40);
197 frmCalculadoraJorgeVictoria.getContentPane().add(btproducto);
198
199 //boton para la operacion division
200 JButton btdivision = new JButton("/");
201 btdivision.setToolTipText("Division");
202 btdivision.setForeground(new Color(255, 255, 255));
203 btdivision.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
204 btdivision.setBorder(new LineBorder(new Color(0, 0, 0)));
```

```
205     btdivision.setBackground(new Color(0, 0, 255));
206     btdivision.setBounds(350, 163, 80, 40);
207     frmCalculadoraJorgeVictoria.getContentPane().add(btdivision);
208
209     //boton para el digito 0
210     JButton bt0 = new JButton("0");
211     bt0.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
212     bt0.setBorder(new LineBorder(new Color(0, 0, 0)));
213     bt0.setBounds(10, 214, 80, 40);
214     frmCalculadoraJorgeVictoria.getContentPane().add(bt0);
215
216     //boton para el digito decimal
217     JButton btpunto = new JButton(".");
218     btpunto.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
219     btpunto.setBorder(new LineBorder(new Color(0, 0, 0)));
220     btpunto.setBounds(95, 214, 80, 40);
221     frmCalculadoraJorgeVictoria.getContentPane().add(btpunto);
222
223     //boton para el signo igual
224     JButton btIgual = new JButton("=");
225     btIgual.setToolTipText("Igual");
226     btIgual.setForeground(new Color(255, 255, 255));
227     btIgual.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
228     btIgual.setBorder(new LineBorder(new Color(0, 0, 0)));
229     btIgual.setBackground(new Color(0, 0, 255));
230     btIgual.setBounds(180, 214, 80, 40);
231     frmCalculadoraJorgeVictoria.getContentPane().add(btIgual);
232
233     //boton para la operacion potencia
234     JButton btCuadrado = new JButton("^");
235     btCuadrado.setToolTipText("Cuadrado");
236     btCuadrado.setForeground(new Color(255, 255, 255));
237     btCuadrado.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
238     btCuadrado.setBorder(new LineBorder(new Color(0, 0, 0)));
239     btCuadrado.setBackground(new Color(0, 0, 255));
240     btCuadrado.setBounds(265, 214, 80, 40);
241     frmCalculadoraJorgeVictoria.getContentPane().add(btCuadrado);
242
243     //boton para la operacion raiz cuadrada
244     JButton btRaiz = new JButton("R");
245     btRaiz.setToolTipText("Raiz Cuadrada");
246     btRaiz.setForeground(new Color(255, 255, 255));
247     btRaiz.setFont(new Font("Comic Sans MS", Font.BOLD, 18));
248     btRaiz.setBorder(new LineBorder(new Color(0, 0, 0)));
249     btRaiz.setBackground(new Color(0, 0, 255));
250     btRaiz.setBounds(350, 214, 80, 40);
251     frmCalculadoraJorgeVictoria.getContentPane().add(btRaiz);
252
253     //listener de los digitos
254     ActionListener digitos = new ActionListener() {
255         public void actionPerformed(ActionEvent e) {
```

```

256         JButton btn = (JButton) e.getSource();
257         if(!bloqueoPantalla) {
258             //si solo esta el cero, lo quitamos y añadimos el digito. Sino, añadimos el digito
259             if (tfPantalla.getText().charAt(0) == 48 && tfPantalla.getText().length() == 1)
                tfPantalla.setText(btn.getText());
            else tfPantalla.setText(tfPantalla.getText() + btn.getText());
        }
    }
};

//listener para poner un punto
ActionListener decimal = new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JButton btn = (JButton) e.getSource();
        if(!bloqueoPantalla) {
            //vemos si existe un punto y sino, lo podemos añadir
            if (!tfPantalla.getText().contains(".")) tfPantalla.setText(tfPantalla.getText() + btn.getText());
        }
    }
};

//listener para borrar digito
ActionListener borrarDigito = new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JButton btn = (JButton) e.getSource();
        if (!bloqueoPantalla) {
            //si solo hay un caracter, pero no es el cero. Si solo hay un cero en pantalla, nada que borrar
            if (tfPantalla.getText().charAt(0) != 48 && tfPantalla.getText().length() == 1) tfPantalla.setText("0");
            //si hay mas de un caracter, vamos borrando por la derecha.
            if (tfPantalla.getText().length() > 1) {
                tfPantalla.setText(tfPantalla.getText().substring(0, tfPantalla.getText().length()-1));
            }
            //puede ocurrir que borremos todos los caracteres, pues ponemos un cero
            if (tfPantalla.getText().length() == 0) tfPantalla.setText("0");
        }
    }
};

//listener para las operaciones binarias
ActionListener operacion = new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JButton btn = (JButton) e.getSource();
        //almacenamos el texto
        signo = btn.getText();
        //solo pasamos el valor en caso de que no lo hayamos hecho
        //si valorA tuviese un valor, el cambio de signo va cambiando hasta que no pulsemos =
        //una vez leído el primer operando, ponemos el digito en 0
        if(valorA.equals("")) {
            valorA = tfPantalla.getText();
            tfPantalla.setText("0");
        }
    }
};

```

```

306     }
307
308     }
309 };
310
311 //listener para el boton igual
312 ActionListener igual = new ActionListener() {
313     public void actionPerformed(ActionEvent e) {
314         if(!bloqueoPantalla) {
315             //para recoger el valorB, valorA debe tener un valor
316             if (valorA.length() > 0) {
317                 valorB = tfPantalla.getText();
318                 //llamada al metodo para realizar la operacion binaria correspondiente
319                 opBinaria();
320                 bloqueoPantalla = true;
321             }
322         }
323     }
324 }
325
326 //metodo que realiza una operacion binaria en funcion del signo almacenado
327 //he intentado tener en cuenta la posibilidad de operaciones con y sin decimales
328 private void opBinaria() {
329
330     //suma
331     if(signo.equals("+") && (valorA.contains(".") || valorB.contains(".")))
332         tfPantalla.setText(String.valueOf(Double.parseDouble(valorA) + Double.parseDouble(valorB)));
333     else if(signo.equals("+")) tfPantalla.setText(String.valueOf(Integer.parseInt(valorA) +
334     Integer.parseInt(valorB)));
335
336     //resta
337     if(signo.equals("-") && (valorA.contains(".") || valorB.contains(".")))
338         tfPantalla.setText(String.valueOf(Double.parseDouble(valorA) - Double.parseDouble(valorB)));
339     else if(signo.equals("-"))tfPantalla.setText(String.valueOf(Integer.parseInt(valorA) -
340     Integer.parseInt(valorB)));
341
342     //producto
343     if(signo.equals("*") && (valorA.contains(".") || valorB.contains(".")))
344         tfPantalla.setText(String.valueOf(Double.parseDouble(valorA) * Double.parseDouble(valorB)));
345     else if(signo.equals("*")) tfPantalla.setText(String.valueOf(Integer.parseInt(valorA) *
346     Integer.parseInt(valorB)));
347
348     //division
349     if(signo.equals("/") && (Double.parseDouble(valorA)%Double.parseDouble(valorB) != 0))
350         tfPantalla.setText(String.valueOf(Double.parseDouble(valorA) / Double.parseDouble(valorB)));
351     else if(signo.equals("/")) tfPantalla.setText(String.valueOf(Integer.parseInt(valorA) /
352     Integer.parseInt(valorB)));
353 }
354 };

```

```

353
354
355 //listener para operaciones unarias
356 ActionListener unario = new ActionListener() {
357     public void actionPerformed(ActionEvent e) {
358         JButton btn = (JButton) e.getSource();
359
360         //elevado al cuadrado
361         if(!bloqueoPantalla) {
362             //para doubles
363             if(btn.getText().equals("^") && (tfPantalla.getText().contains("."))) {
364                 potenciaDouble = Double.parseDouble(tfPantalla.getText());
365                 potenciaDouble = Math.pow(potenciaDouble, 2);
366                 tfPantalla.setText(Double.toString(potenciaDouble));
367
368                 //para enteros
369             } else if(btn.getText().equals("^")) {
370                 potenciaInteger = Integer.parseInt(tfPantalla.getText());
371                 potenciaInteger = (int) Math.pow(potenciaInteger, 2);
372                 tfPantalla.setText(Integer.toString(potenciaInteger));
373             }
374
375             //raiz cuadrada
376             if(btn.getText().equals("R")){
377                 raizCuadrada = Double.parseDouble(tfPantalla.getText());
378                 raizCuadrada = Math.sqrt(raizCuadrada);
379                 if(raizCuadrada < 0) tfPantalla.setText("Err");
380                 if ((int)raizCuadrada%raizCuadrada==0)
381                     tfPantalla.setText(Integer.toString((int)raizCuadrada));
382                 else if(Double.toString(raizCuadrada).contains("."))
383                     tfPantalla.setText(Double.toString(raizCuadrada));
384
385             }
386
387             bloqueoPantalla = true;
388         }
389     }
390 };
391
392 //listener para limpiar la pantalla
393 ActionListener limpiarPantalla = new ActionListener() {
394     public void actionPerformed(ActionEvent e) {
395         //desbloqueamos la pantalla e inicializamos todo
396         bloqueoPantalla = false;
397         tfPantalla.setText("0");
398         valorA = "";
399         valorB = "";
400         signo = "";
401     }
402 };
403

```

```
404
405
406
407 //llamadas a listeners
408 bt0.addActionListener(digitos);
409 bt1.addActionListener(digitos);
410 bt2.addActionListener(digitos);
411 bt3.addActionListener(digitos);
412 bt4.addActionListener(digitos);
413 bt5.addActionListener(digitos);
414 bt6.addActionListener(digitos);
415 bt7.addActionListener(digitos);
416 bt8.addActionListener(digitos);
417 bt9.addActionListener(digitos);
418
419 btPunto.addActionListener(decimal);
420
421 btDel.addActionListener(borrarDigito);
422
423 btSuma.addActionListener(operacion);
424 btResta.addActionListener(operacion);
425 btProducto.addActionListener(operacion);
426 btDivision.addActionListener(operacion);
427
428 btIgual.addActionListener(igual);
429
430 btCuadrado.addActionListener(unario);
431 btRaiz.addActionListener(unario);
432
433 btLimpiar.addActionListener(limpiarPantalla);
434
435
436 }
437 }
438
```

```

1  package E705;
2
3  import java.awt.EventQueue;
4  import java.awt.Graphics;
5  import java.awt.Image;
6  import java.awt.color.ColorSpace;
7  import java.awt.event.ActionEvent;
8  import java.awt.event.ActionListener;
9  import java.awt.image.BufferedImage;
10 import java.awt.image.ColorConvertOp;
11 import java.awt.image.WritableRaster;
12 import java.io.File;
13 import java.io.IOException;
14
15 import javax.swing.JFrame;
16 import javax.swing.JLabel;
17 import javax.swing.JButton;
18 import javax.swing.JFileChooser;
19
20 import java.awt.BorderLayout;
21
22 import javax.imageio.ImageIO;
23 import javax.swing.ImageIcon;
24 import javax.swing.JScrollPane;
25 import javax.swing.border.LineBorder;
26 import javax.swing.filechooser.FileNameExtensionFilter;
27
28 import java.awt.Color;
29 import javax.swing.JPanel;
30
31 public class E705 {
32
33     private JFrame ecualizador;           //JFrame
34     private JPanel pCopia;                //Jpanel donde metemeos el histograma de la copia
35     private File fichero;                 //para guardar el fichero que leemos
36     private JScrollPane SPfotoOriginal;   //ScrollPane de la foto original
37     private JScrollPane SPfotoCopia;      //ScrollPane de la foto que es copia
38     private BufferedImage bi;              //imagen con la que vamos a trabajar
39     private ImageIcon ii;                  //paso intermedio de file a bufferedImage
40     private double[] histo = new double[256]; //histograma de los pixeles
41     private double[] histoAcu = new double[256]; //histograma acumulado
42     private JPanel pOriginal;              //Jpanel donde metemos el histograma de la foto original
43
44     /**
45      * Launch the application.
46      */
47     public static void main(String[] args) {
48         EventQueue.invokeLater(new Runnable() {
49             public void run() {
50                 try {
51                     E705 window = new E705();

```



```

52         window.ecualizador.setVisible(true);
53     } catch (Exception e) {
54         e.printStackTrace();
55     }
56 }
57 });
58 }
59
60 /**
61  * Create the application.
62  */
63 public E705() {
64     initialize();
65 }
66
67 /**
68  * Initialize the contents of the frame.
69  */
70 private void initialize() {
71     ecualizador = new JFrame();
72     ecualizador.setTitle("Jorge Victoria Andreu");
73     ecualizador.setBounds(50, 50, 800, 700);
74     ecualizador.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
75     ecualizador.getContentPane().setLayout(null);
76
77     JButton btAbrir = new JButton("Abrir");
78     btAbrir.setIcon(new ImageIcon("C:\\Users\\jvand\\OneDrive\\Estudios\\Workspace\\Eclipse\\Tema
79 07\\src\\E705\\iconos\\carpeta.png"));
80     btAbrir.setBounds(10, 11, 120, 23);
81     ecualizador.getContentPane().add(btAbrir);
82
83     JButton btEqualizar = new JButton("Eequalizar");
84     btEequalizar.setIcon(new ImageIcon("C:\\Users\\jvand\\OneDrive\\Estudios\\Workspace\\Eclipse\\Tema
85 07\\src\\E705\\iconos\\ecualizador.png"));
86     btEequalizar.setEnabled(false);
87     btEequalizar.setBounds(140, 11, 120, 23);
88     ecualizador.getContentPane().add(btEequalizar);
89
90     JButton btGuardar = new JButton("Guardar");
91     btGuardar.setEnabled(false);
92     btGuardar.setIcon(new ImageIcon("C:\\Users\\jvand\\OneDrive\\Estudios\\Workspace\\Eclipse\\Tema
93 07\\src\\E705\\iconos\\disquette.png"));
94     btGuardar.setBounds(270, 11, 120, 23);
95     ecualizador.getContentPane().add(btGuardar);
96
97     SPfotoCopia = new JScrollPane();
98     SPfotoCopia.setBounds(404, 45, 370, 255);
99     ecualizador.getContentPane().add(SPfotoCopia);
100
101     SPfotoOriginal = new JScrollPane();
102     SPfotoOriginal.setBounds(10, 45, 370, 255);

```

```

100     ecualizador.getContentPane().add(SPfotoOriginal);
101
102     pOriginal = new JPanel() {
103
104         public void paint(Graphics g) {
105             super.paint(g);
106             g.setColor(Color.black);
107
108             //ejes horizontal y vertical
109             g.drawLine(30, 270, 286, 270);
110             g.drawLine(30, 270, 30, 20);
111
112             g.drawString("0", 28, 290 );
113
114             int ancho = 28;
115             //dibujamos marcas y numeros
116             for(int i = 1; i <= 5; i++) {
117                 ancho = ancho + 50;
118                 g.drawString(String.valueOf(i*50), ancho-5, 290);
119                 g.drawLine(ancho, 268, ancho, 272);
120             }
121         }
122     };
123
124     pOriginal.setBorder(new LineBorder(new Color(0, 0, 0)));
125     pOriginal.setBackground(Color.WHITE);
126     pOriginal.setBounds(10, 311, 370, 300);
127     ecualizador.getContentPane().add(pOriginal);
128
129     pCopia = new JPanel() {
130         public void paint(Graphics g) {
131             super.paint(g);
132             g.setColor(Color.black);
133
134             //ejes horizontal y vertical
135             g.drawLine(30, 270, 286, 270);
136             g.drawLine(30, 270, 30, 20);
137
138             g.drawString("0", 28, 290 );
139
140             int ancho = 28;
141             //dibujamos marcas y numeros
142             for(int i = 1; i <= 5; i++) {
143                 ancho = ancho + 50;
144                 g.drawString(String.valueOf(i*50), ancho-5, 290);
145                 g.drawLine(ancho, 268, ancho, 272);
146             }
147         }
148     };
149     pCopia.setBorder(new LineBorder(new Color(0, 0, 0)));
150     pCopia.setBackground(Color.WHITE);

```

```

151     pCopia.setBounds(404, 311, 370, 300);
152     ecualizador.getContentPane().add(pCopia);
153
154     //listener para el jfilechooser
155     btAbrir.addActionListener(new ActionListener() {
156         public void actionPerformed(ActionEvent e) {
157             //buscamos el fichero
158             fichero = leerFichero();
159
160             //cargamos la imagen
161             cargaImagen();
162
163             //vamos a crear el histograma
164             calculoHistograma();
165
166             //vamos a pintar el histograma
167             pintaGrafica(pOriginal);
168
169             //ponemos el boton de ecualizar activo
170             btEcualizar.setEnabled(true);
171         }
172     });
173
174     //listener para el ecualizador
175     btEcualizar.addActionListener(new ActionListener() {
176         public void actionPerformed(ActionEvent e) {
177             //ecualizamos la imagen
178             bi = ecualizar(bi);
179
180             //cargamos la imagen
181             cargaImagenCopia();
182
183             //vamos a crear el histograma
184             calculoHistograma();
185
186             //vamos a pintar el histograma
187             pintaGrafica(pCopia);
188
189             //ponemos el boton de guardar activo
190             btGuardar.setEnabled(true);
191         }
192     });
193
194     //listener para el guardado
195     btGuardar.addActionListener(new ActionListener() {
196         public void actionPerformed(ActionEvent e) {
197             guardarFichero();
198         }
199     });
200 }
201

```

```
202 //metodo para leer fichero con jfileChooer
203 public static File leerFichero() {
204
205     JFileChooser fd = new JFileChooser();
206     fd.setAcceptAllFileFilterUsed(false);
207     FileNameExtensionFilter filter = new FileNameExtensionFilter("archivos imagen", "jpg", "png", "gif");
208     fd.addChoosableFileFilter(filter);
209     fd.setDialogTitle("Selecciona el fichero a leer");
210     fd.setSelectedFile(null);
211     int opcion = fd.showOpenDialog(null);
212
213     if (opcion != JFileChooser.APPROVE_OPTION) return null;
214
215     File f = fd.getSelectedFile();
216
217     return f;
218 }
219
220 //metodo para cargar la imagen en pantalla
221 public void cargaImagen() {
222
223     try {
224         bi = ImageIO.read(fichero);
225     } catch (IOException e1) {
226         // TODO Bloque catch generado automáticamente
227         e1.printStackTrace();
228     }
229
230
231     //llamamos al metodo para pasar a escala de grises
232     bi = escalarGrises(bi);
233
234     //escalamos la imagen
235     ii = new ImageIcon(bi.getScaledInstance(240, 291, Image.SCALE_SMOOTH));
236
237     //construimos un jlabel para poder incrustar la imagen en el jpanel
238     JLabel etiqueta = new JLabel(ii);
239
240     //cargamos la foto en el jscrollpane
241     SPfotoOriginal.setViewportView(etiqueta);
242 }
243
244 //metodo para cargar la imagen ecualizada en pantalla
245 public void cargaImagenCopia() {
246
247     //escalamos la imagen
248     ii = new ImageIcon(bi.getScaledInstance(240, 291, Image.SCALE_SMOOTH));
249
250     //construimos un jlabel para poder incrustar la imagen en el jpanel
251     JLabel etiqueta = new JLabel(ii);
252 }
```

```

253     //cargamos la foto en el jscrollpane
254     SPfotoCopia.setViewportView(etiqueta);
255 }
256
257 //metodo para escalar la imagen a grises, si viene en color
258 public BufferedImage escalarGrises(BufferedImage imagen) {
259     int mediaPixeles;
260     int colorSRGB;
261     Color nuevoColor;
262
263     //Recorremos la imagen píxel a píxel
264     for( int i = 0; i < imagen.getWidth(); i++ ){
265         for( int j = 0; j < imagen.getHeight(); j++ ){
266             //Almacenamos el color del píxel
267             nuevoColor=new Color(imagen.getRGB(i, j));
268             //Calculamos la media de los tres canales (rojo, verde, azul)
269             mediaPixeles=(int) ((nuevoColor.getRed()+nuevoColor.getGreen()+nuevoColor.getBlue())/3);
270             //Cambiamos a formato sRGB
271             colorSRGB=(mediaPixeles << 16) | (mediaPixeles << 8) | mediaPixeles;
272             //Asignamos el nuevo valor al BufferedImage
273             imagen.setRGB(i, j,colorSRGB);
274         }
275     }
276     //Retornamos la imagen
277     return imagen;
278 }
279
280 //metodo para calcular los histogramas
281 public void calculoHistograma() {
282     int nivel;
283     //ponemos el array de colores a 0
284     for(int i = 0; i < 256; i++) {
285         histo[i] = 0;
286     }
287
288     //Recorremos la imagen píxel a píxel y rellenamos el array histo
289     for( int i = 0; i < bi.getWidth(); i++ ){
290         for( int j = 0; j < bi.getHeight(); j++ ){
291             Color color =new Color( bi.getRGB(i, j));
292             nivel = (color.getRed() +color.getGreen() +color.getBlue() )/3;
293             histo[nivel]++;
294         }
295     }
296
297     //ahora rellenamos el array acumulado
298     //primero rellenamos el array de 0
299     for(int i = 0; i < 256; i++) {
300         histoAcu[i] = 0;
301     }
302
303     //ponemos la posicion 0

```

```

304     histoAcu[0] = histo[0];
305
306     //y ya rellenamos el resto
307     for(int i = 1; i < 256; i++) {
308         histoAcu[i] = histoAcu[i-1] + histo[i];
309     }
310 }
311
312 //metodo para pintar las graficas en pantalla
313 public void pintaGrafica(JPanel original ) {
314
315     Graphics gr = original.getGraphics();
316
317     gr.setColor(Color.red);
318
319     //pintamos barra con el resultado de las posiciones del arry histo
320     for(int i = 0; i < 256; i++) {
321         gr.drawLine(30+i, 270-(int)histo[i]/20, 30+i, 270);
322     }
323
324     gr.setColor(Color.black);
325
326     //pintamos linea histograma historico acumulado
327     for(int i = 0; i < 256; i++) {
328         gr.drawString("-", 30+i, 270-(int)histoAcu[i]/600);
329     }
330 }
331
332 }
333
334 //metodo para ecualizar la imagen
335 public BufferedImage ecualizar(BufferedImage imagen) {
336     int mediaPixeles;
337     int colorSRGB;
338     Color nuevoColor;
339
340     for (int i = 0; i < imagen.getWidth(); i++) {
341         for (int j = 0; j < imagen.getHeight(); j++) {
342             //Almacenamos el color del píxel
343             nuevoColor=new Color(imagen.getRGB(i, j));
344             //Calculamos la media de los tres canales (rojo, verde, azul)
345             mediaPixeles=(int) ((nuevoColor.getRed()+nuevoColor.getGreen()+nuevoColor.getBlue())/3);
346
347             imagen.setRGB(i,j,(int) (255*histoAcu[mediaPixeles]/histoAcu[255]));
348         }
349     }
350     return imagen;
351 }
352
353
354 //metodo para guardar el fichero

```

```
355     public void guardarFichero() {
356
357         JFileChooser fd = new JFileChooser();
358         fd.setFileSelectionMode(JFileChooser.FILES_ONLY);
359         fd.setAcceptAllFileFilterUsed(false);
360         fd.setDialogTitle("Guardar imagen");
361         fd.setApproveButtonText("Aceptar");
362         int opcion = fd.showSaveDialog(null);
363         FileNameExtensionFilter filter = new FileNameExtensionFilter("archivos imagen", "jpg");
364         if(opcion==JFileChooser.APPROVE_OPTION){
365             File f = fd.getSelectedFile();
366             String test = f.getAbsolutePath();
367             try {
368                 ImageIO.write(bi,"jpg",new File(test));
369             } catch (IOException e1) {
370                 // TODO Auto-generated catch block
371                 e1.printStackTrace();
372             }
373         }
374     }
375 }
376
```

```
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) +
                                fields[1].charAt(4);
190                         }
191                         //no es un 10
192                         else palabra = palabra + fields[1].charAt(0) + "." + fields[1].charAt(2) + fields[1].charAt(3);
193                         //con la nueva linea correctamente creada, ya podemos hacer la conversion
194                         nota = Double.parseDouble(palabra);
195
196                         list.add(nota);
197
198                         notaTotal += nota; //almacenamos en el total de notas
199                         palabra = ""; //inicializamos la cadena
200                     }
201
202                     //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
```