

_01_CreandoYEscribiendoEnMarcos.java

```
1 package Graficos;
2
3 import java.awt.*;
4 import javax.swing.*;
5
6 public class _01_CreandoYEscribiendoEnMarcos {
7
8     public static void main(String[] args) {
9         Marco ventana = new Marco();
10        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11    }
12 }
13
14 class Marco extends JFrame {
15     public Marco() {
16         Dimension pantalla = Toolkit.getDefaultToolkit().getScreenSize();
17         int ancho = (int) pantalla.getWidth();
18         int alto = (int) pantalla.getHeight();
19         setBounds(ancho/3, alto/3, ancho/3, alto/3);
20         setTitle(" Mi Ventana Java");
21         ImageIcon icono = new ImageIcon("src/Graficos/images/icon.png");
22         setIconImage(icono.getImage());
23         Lamina laminaObj = new Lamina();
24         add(laminaObj);
25         setVisible(true);
26     }
27 }
28
29 class Lamina extends JPanel {
30     public void paintComponent(Graphics g) {
31         super.paintComponent(g);
32         g.setFont(new Font("Roboto", Font.BOLD, 20));
33         g.setColor(new Color(153, 51, 255));
34         g.drawString("Título del Contenido", 100, 100);
35     }
36 }
```

```
1 package Graficos;
2
3 import java.awt.Color;
4 import java.awt.Graphics;
5 import java.awt.Graphics2D;
6 import java.awt.geom.Ellipse2D;
7 import java.awt.geom.Rectangle2D;
8 import javax.swing.JFrame;
9 import javax.swing.JPanel;
10
11 public class _02_PruebaDibujo_TrabajandoConColores {
12
13     public static void main(String[] args) {
14         Marco2 ventana = new Marco2();
15         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
16     }
17 }
18
19 class Marco2 extends JFrame {
20     public Marco2 () {
21         setSize(840, 840);
22         setTitle(" Dibujos y Colores");
23         setLocationRelativeTo(null);
24         add(new Lamina2());
25         setVisible(true);
26     }
27 }
28
29 class Lamina2 extends JPanel {
30     public Lamina2() {
31         setBackground(new Color(179, 255, 255));
32     }
33     @Override
34     protected void paintComponent(Graphics g) {
35         super.paintComponent(g);
36         g.drawRect(100, 200, 150, 100);
37         g.drawString("1 - Rectángulo Vacío Graphics", 100, 180);
38         g.fillRect(100, 400, 150, 100);
39         g.drawString("2 - Rectángulo Relleno Graphics", 100, 380);
40         Graphics2D g2D = (Graphics2D) g;
41         Rectangle2D rectangulo = new Rectangle2D.Double(500, 200, 150, 100);
42         g2D.draw(rectangulo);
43         g2D.drawString("3 - Rectángulo Graphics2D", 500, 120);
44         Ellipse2D elipse = new Ellipse2D.Double();
45         elipse setFrame(rectangulo);
46         g2D.draw(elipse);
47         g2D.drawString("4 - Elipse Graphics2D", 500, 140);
48         g2D.drawLine(500, 200, 650, 300);
49         g2D.drawString("5 - Línea Oblícu Graphics2D", 500, 370);
50         double centroX = rectangulo.getCenterX();
51         double centroY = rectangulo.getCenterY();
52         double radio = 90;
53         Ellipse2D circulo = new Ellipse2D.Double();
54         circulo.setFrameFromCenter(centroX, centroY, centroX+radio, centroY+radio);
55         g2D.draw(circulo);
56         g2D.drawString("6 - Círculo Graphics2D", 500, 390);
57         g.drawString("7 - Rectángulo Relleno Graphics", 500, 430);
58         g.setColor(new Color(255, 153, 255));
59         g.fillRect(500, 480, 200, 200);
60         g.setColor(new Color(0, 0, 0));
61         g.drawString("8 - Círculo Relleno Graphics", 500, 460);
62         g.setColor(new Color(179, 255, 179));
```

_02_PruebaDibujo_TrabajandoConColores.java

```
63      g.fillOval(500, 480, 200, 200);
64  }
65 }
```

_03_FuentesTipo_TrabajandoConFuentes.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.Font;
6 import java.awt.GraphicsEnvironment;
7 import javax.swing.JFrame;
8 import javax.swing.JLabel;
9 import javax.swing.JOptionPane;
10 import javax.swing.JPanel;
11 import javax.swing.SwingConstants;
12
13 public class _03_FuentesTipo_TrabajandoConFuentes {
14
15     public static void main(String[] args) {
16
17         Marco3 ventana = new Marco3();
18         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19     }
20 }
21
22 class Marco3 extends JFrame {
23     public Marco3() {
24         setSize(900, 300);
25         setTitle(" Trabajando con Fuentes");
26         setLocationRelativeTo(null);
27
28         setLayout(new BorderLayout());
29         add(new JLabel("                "), BorderLayout.NORTH);
30         add(new JLabel("                "), BorderLayout.SOUTH);
31         add(new JLabel("                "), BorderLayout.WEST);
32         add(new JLabel("                "), BorderLayout.EAST);
33
34         add(new Ventana3(), BorderLayout.CENTER);
35
36         setVisible(true);
37     }
38 }
39
40 class Ventana3 extends JPanel {
41
42     String fuenteIn;
43     boolean verificador = false;
44     JLabel encabezadoText, fuenteText;
45
46     public Ventana3() {
47
48         setLayout(new BorderLayout());
49
50         fuenteIn = JOptionPane.showInputDialog("Ingrese el nombre de la fuente a
consultar").toUpperCase();
51
52         GraphicsEnvironment e = GraphicsEnvironment.getLocalGraphicsEnvironment();
53         String Fuentes[] = e.getAvailableFontFamilyNames();
54
55         for(String fuente: Fuentes) {
56             if(fuenteIn.equalsIgnoreCase(fuente)) {
57                 verificador = true;
58                 break;
59             }
60         }
61
62     }
```

```
62         if(verificador) {
63             add(encabezadoText = new JLabel("El tipo de Fuente: " + fuenteIn + " Sí se
encuentra instalada en el sistema."), JLabel.CENTER);
64             encabezadoText.setFont(new Font(fuenteIn, Font.BOLD, 22));
65             encabezadoText.setForeground(new Color(0, 153, 51));
66             encabezadoText.setHorizontalAlignment(SwingConstants.CENTER);
67             setBackground(new Color(204, 255, 221));
68
69             System.out.println("El tipo de Fuente: " + fuenteIn + " Sí se encuentra
instalada en el sistema.");
70
71         } else {
72             add(encabezadoText = new JLabel("El tipo de Fuente: " + fuenteIn + " No está
instalada en el sistema."), JLabel.CENTER);
73             encabezadoText.setFont(new Font("Arial", Font.BOLD, 22));
74             encabezadoText.setForeground(new Color(153, 0, 61));
75             encabezadoText.setHorizontalAlignment(SwingConstants.CENTER);
76             setBackground(new Color(255, 204, 224));
77
78             System.out.println("El tipo de Fuente: " + fuenteIn + " No está instalada en
el sistema.");
79         }
80     }
81 }
```

_04_PruebaImagenes.java

```
1 package Graficos;
2
3 import java.awt.Graphics;
4 import java.awt.Image;
5 import java.io.File;
6 import javax.imageio.ImageIO;
7 import javax.swing.ImageIcon;
8 import javax.swing.JFrame;
9 import javax.swing.JPanel;
10
11 public class _04_PruebaImagenes {
12
13     public static void main(String[] args) {
14
15         Marco4 ventana = new Marco4();
16         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
17     }
18 }
19
20 class Marco4 extends JFrame {
21
22     public Marco4() {
23         setSize(600, 500);
24         setTitle(" Prueba Imágenes");
25         setLocationRelativeTo(null);
26
27         ImageIcon icono = new ImageIcon("src/Graficos/images/musicIco.png");
28         setIconImage(icono.getImage());
29
30         add(new Ventana4());
31
32         setVisible(true);
33     }
34 }
35
36 class Ventana4 extends JPanel {
37
38     private Image radioImg;
39     private Image musicImg;
40     private int anchoImg;
41     private int altoImg;
42
43     protected void paintComponent(Graphics g) {
44
45         super.paintComponent(g);
46
47         // -----MUSIC IMG-----
48
49         File musicPath = new File("src/Graficos/images/music.png");
50
51         try {
52             musicImg = ImageIO.read(musicPath);
53         } catch (Exception e) {
54             System.out.println("No File!");
55         }
56
57         g.drawImage(musicImg, 0, 0, 100, 100, null);
58
59         for (int i=0; i<600; i++) {
60             for(int j=0; j<500; j++) {
61                 g.copyArea(0, 0, 100, 100, i*100, j*100);
62             }
63         }
64     }
65 }
```

_04_PruebaImagenes.java

```
63     }
64
65     // -----RADIO IMG-----
66
67     File radioPath = new File("src/Graficos/images/radio.png");
68
69     try {
70         radioImg = ImageIO.read(radioPath);
71     } catch (Exception e) {
72         System.out.println("No File!");
73     }
74
75     anchoImg = radioImg.getWidth(null);
76     altoImg = radioImg.getHeight(null);
77
78     int x = (this.getWidth() - anchoImg/2) / 2;
79     int y = (this.getHeight() - altoImg/2) / 2;
80
81     g.drawImage(radioImg, x, y, anchoImg/2, altoImg/2, null);
82 }
83 }
```

```
1 package Graficos;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import javax.swing.*;
6
7 public class _05_PruebaEventos_PruebaAcciones {
8
9     public static void main(String[] args) {
10
11         Marco5 ventana = new Marco5();
12         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13     }
14 }
15
16 class Marco5 extends JFrame {
17
18     public Marco5() {
19
20         setSize(600, 500);
21         setTitle(" Eventos y Acciones");
22         setLocationRelativeTo(null);
23
24         add(new Ventana5());
25
26         setVisible(true);
27     }
28 }
29
30 class Ventana5 extends JPanel {
31
32     public Ventana5() {
33
34         setLayout(new GridBagLayout());
35         GridBagConstraints gbc = new GridBagConstraints();
36         gbc.insets = new Insets(0, 10, 0, 10);
37
38         AccionColor accionLima = new AccionColor("Lima", new
39 ImageIcon("src/Graficos/images/lima.png"), new Color(204, 255, 153));
40         AccionColor accionOrange = new AccionColor("Orange", new
41 ImageIcon("src/Graficos/images/orange.png"), new Color(255, 204, 102));
42         AccionColor accionGrape = new AccionColor("Grape", new
43 ImageIcon("src/Graficos/images/grape.png"), new Color(204, 204, 255));
44
45         add(new JButton(accionLima), gbc);
46         add(new JButton(accionOrange), gbc);
47         add(new JButton(accionGrape), gbc);
48
49         KeyStroke tecladoLima = KeyStroke.getKeyStroke("ctrl L");
50         KeyStroke tecladoOrange = KeyStroke.getKeyStroke("ctrl O");
51         KeyStroke tecladoGrape = KeyStroke.getKeyStroke("ctrl G");
52
53         InputMap mapaEntrada = getInputMap(WHEN_IN_FOCUSED_WINDOW);
54
55         mapaEntrada.put(tecladoLima, "eventoLima");
56         mapaEntrada.put(tecladoOrange, "eventoOrange");
57         mapaEntrada.put(tecladoGrape, "eventoGrape");
58
59         ActionMap mapaAccion = getActionMap();
60
61         mapaAccion.put("eventoLima", accionLima);
62         mapaAccion.put("eventoOrange", accionOrange);
63     }
64 }
```



```
60     mapaAccion.put("eventoGrape", accionGrape);
61 }
62
63 private class AccionColor extends AbstractAction {
64
65     public AccionColor(String nombre, Icon icono, Color color_boton) {
66
67         putValue(Action.NAME, nombre);
68         putValue(Action.SMALL_ICON, icono);
69         putValue(Action.SHORT_DESCRIPTION, "(Ctrl+" + nombre.charAt(0) + ") " + "Cambia
el fondo a color " + nombre);
70         putValue("color_de_fondo", color_boton);
71     }
72
73     public void actionPerformed(ActionEvent e) {
74         Color c = (Color) getValue("color_de_fondo");
75         setBackground(c);
76     }
77 }
78 }
```

```
1 package Graficos;
2
3 import java.awt.Color;
4 import java.awt.GridBagConstraints;
5 import java.awt.GridBagLayout;
6 import java.awt.event.ActionEvent;
7 import javax.swing.AbstractAction;
8 import javax.swing.ActionMap;
9 import javax.swing.ImageIcon;
10 import javax.swing.InputMap;
11 import javax.swing.JButton;
12 import javax.swing.JFrame;
13 import javax.swing.JPanel;
14 import javax.swing.KeyStroke;
15
16 public class _05_PruebaEventos_PruebaAcciones_ByPipe {
17
18     public static void main(String[] args) {
19
20         MarcoAccionesPipe ventana = new MarcoAccionesPipe();
21         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22     }
23 }
24
25 class MarcoAccionesPipe extends JFrame {
26
27     public MarcoAccionesPipe() {
28
29         setTitle(" Eventos y Acciones by Pipe");
30         setSize(600, 400);
31         setLocationRelativeTo(null);
32
33         add(new VentanaAccionesPipe());
34
35         setVisible(true);
36     }
37 }
38
39 class VentanaAccionesPipe extends JPanel {
40
41     public VentanaAccionesPipe() {
42
43         setLayout(new GridBagLayout());
44         GridBagConstraints gbc = new GridBagConstraints();
45         gbc.insets.set(0, 0, 50, 10);
46
47         Acciones accionLima = new Acciones("Lima", new
48 ImageIcon("src/Graficos/images/lima.png"), new Color(204, 255, 153));
49         Acciones accionOrange = new Acciones("Orange", new
50 ImageIcon("src/Graficos/images/orange.png"), new Color(255, 204, 102));
51         Acciones accionGrape = new Acciones("Grape", new
52 ImageIcon("src/Graficos/images/grape.png"), new Color(204, 204, 255));
53
54         add(new JButton(accionLima), gbc);
55         add(new JButton(accionOrange), gbc);
56         add(new JButton(accionGrape), gbc);
57
58         KeyStroke tecladoLima = KeyStroke.getKeyStroke("ctrl L");
59         KeyStroke tecladoOrange = KeyStroke.getKeyStroke("ctrl O");
60         KeyStroke tecladoGrape = KeyStroke.getKeyStroke("ctrl G");
61
62         InputMap mapaEntrada = getInputMap(WHEN_IN_FOCUSED_WINDOW);
```

```
60
61     mapaEntrada.put(tecladoLima, "tecladoLima");
62     mapaEntrada.put(tecladoOrange, "tecladoOrange");
63     mapaEntrada.put(tecladoGrape, "tecladoGrape");
64
65     ActionMap mapaAccion = getActionMap();
66
67     mapaAccion.put("tecladoLima", accionLima);
68     mapaAccion.put("tecladoOrange", accionOrange);
69     mapaAccion.put("tecladoGrape", accionGrape);
70 }
71
72 class Acciones extends AbstractAction {
73
74     public Acciones(String nombre, ImageIcon icon, Color color) {
75         putValue(NAME, nombre);
76         putValue(SMALL_ICON, icon);
77         putValue("dameColor", color);
78         putValue(SHORT_DESCRIPTION, "Ctrl+" + nombre.charAt(0) + " Cambia el fondo a
color: " + nombre);
79     }
80
81     @Override
82     public void actionPerformed(ActionEvent e) {
83         Color color = (Color) getValue("dameColor");
84         setBackground(color);
85     }
86 }
87 }
```

_06_EventosFocoVentanaTeclado.java

```
1 package Graficos;
2
3 import java.awt.event.KeyEvent;
4 import java.awt.event.KeyListener;
5 import java.awt.event.WindowAdapter;
6 import java.awt.event.WindowEvent;
7 import javax.swing.JFrame;
8
9 public class _06_EventosFocoVentanaTeclado {
10
11     public static void main(String[] args) {
12
13         Marco6 ventana1 = new Marco6();
14         Marco6New ventana2 = new Marco6New();
15     }
16 }
17
18 class Marco6 extends JFrame {
19
20     private String nombreVentana;
21
22     public Marco6() {
23
24         setBounds(200, 200, 500, 300);
25         setTitle(" Ventana 1ª");
26
27         nombreVentana = this.getTitle();
28
29         addWindowFocusListener(new EventosVentana(nombreVentana));
30         addWindowListener(new EventosVentana(nombreVentana));
31         addWindowStateListener(new EventosVentana(nombreVentana));
32
33         addKeyListener(new EventosTeclado());
34
35         setVisible(true);
36         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
37     }
38 }
39
40 class Marco6New extends JFrame {
41
42     private String nombreVentana;
43
44     public Marco6New() {
45
46         setBounds(900, 200, 500, 300);
47         setTitle(" Ventana 2ª");
48
49         nombreVentana = this.getTitle();
50
51         addWindowFocusListener(new EventosVentana(nombreVentana));
52         addWindowListener(new EventosVentana(nombreVentana));
53         addWindowStateListener(new EventosVentana(nombreVentana));
54
55         addKeyListener(new EventosTeclado());
56
57         setVisible(true);
58     }
59 }
60
61 class EventosVentana extends WindowAdapter {
62
```

```
63 String nombre;
64
65 public EventosVentana(String nombre) {
66     this.nombre = nombre;
67 }
68
69 @Override
70 public void windowActivated(WindowEvent e) {
71     System.out.println("La Ventana se ha Activado " + nombre);
72     super.windowActivated(e);
73 }
74
75 @Override
76 public void windowDeactivated(WindowEvent e) {
77     System.out.println("La Ventana se ha Desactivado" + nombre);
78     super.windowDeactivated(e);
79 }
80
81 @Override
82 public void windowOpened(WindowEvent e) {
83     System.out.println("La Ventana se ha Abierto" + nombre);
84     super.windowOpened(e);
85 }
86
87 @Override
88 public void windowClosing(WindowEvent e) {
89     System.out.println("La Ventana se está Cerrando" + nombre);
90     super.windowClosing(e);
91 }
92
93 @Override
94 public void windowClosed(WindowEvent e) {
95     System.out.println("La Ventana se ha Cerrado" + nombre);
96     super.windowClosed(e);
97 }
98
99 @Override
100 public void windowIconified(WindowEvent e) {
101     System.out.println("La Ventana se ha Minimizado" + nombre);
102     super.windowIconified(e);
103 }
104
105 @Override
106 public void windowDeiconified(WindowEvent e) {
107     System.out.println("La Ventana se ha Maximizado" + nombre);
108     super.windowDeiconified(e);
109 }
110
111 @Override
112 public void windowGainedFocus(WindowEvent e) {
113     System.out.println("La Ventana ha Ganado el Foco" + nombre);
114     super.windowGainedFocus(e);
115 }
116
117 @Override
118 public void windowLostFocus(WindowEvent e) {
119     System.out.println("La Ventana ha Perdido el Foco" + nombre);
120     super.windowLostFocus(e);
121 }
122
123 @Override
124 public void windowStateChanged(WindowEvent e) {
```

```
125         System.out.println("La Ventana ha Cambiado de Estado" + nombre);
126         super.windowStateChanged(e);
127     }
128 }
129
130
131 class EventosTeclado implements KeyListener {
132
133     @Override
134     public void keyTyped(KeyEvent e) {
135         System.out.println("Se ha Tecleado la tecla: " + e.getKeyChar());
136     }
137
138     @Override
139     public void keyPressed(KeyEvent e) {
140         System.out.println("Se ha Presionado la tecla: " + e.getKeyChar());
141     }
142
143     @Override
144     public void keyReleased(KeyEvent e) {
145         System.out.println("Se ha Levantado la tecla: " + e.getKeyChar());
146     }
147 }
```

_07_EventosRaton.java

```
1 package Graficos;
2
3 import java.awt.event.MouseAdapter;
4 import java.awt.event.MouseEvent;
5 import java.awt.event.MouseWheelEvent;
6 import javax.swing.JFrame;
7
8 public class _07_EventosRaton extends JFrame {
9
10     public static void main(String[] args) {
11
12         JFrame ventana = new JFrame();
13
14         ventana.setTitle(" Eventos del Ratón");
15         ventana.setSize(500, 400);
16         ventana.setLocationRelativeTo(null);
17
18         ventana.addMouseListener(new AccionesRaton());
19         ventana.addMouseMotionListener(new AccionesRaton());
20         ventana.addMouseWheelListener(new AccionesRaton());
21
22         ventana.setVisible(true);
23         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24     }
25 }
26
27 class AccionesRaton extends MouseAdapter {
28
29     @Override
30     public void mouseClicked(MouseEvent e) {
31         int boton = e.getButton();
32         String click = "";
33         if (boton == 1) {
34             click = "Click Izquierdo";
35         } else if (boton == 2) {
36             click = "Rueda del Ratón";
37         } else if (boton == 3) {
38             click = "Click Derecho";
39         }
40         int x = e.getX();
41         int y = e.getY();
42         System.out.println("Se ha hecho Click con: " + click + " en la coordenada: X=" + x +
43 " Y=" + y);
44         super.mouseClicked(e);
45     }
46
47     @Override
48     public void mouseDragged(MouseEvent e) {
49         System.out.println("Se ha Arrastrado el ratón");
50         super.mouseDragged(e);
51     }
52
53     @Override
54     public void mouseEntered(MouseEvent e) {
55         System.out.println("El ratón ha Entrado en la ventana");
56         super.mouseEntered(e);
57     }
58
59     @Override
60     public void mouseExited(MouseEvent e) {
61         System.out.println("El ratón ha Salido de la ventana");
62         super.mouseExited(e);
63     }
64 }
```

_07_EventosRaton.java

```
62     }
63
64     @Override
65     public void mouseMoved(MouseEvent e) {
66 //         System.out.println("Se ha Movido el ratón");
67         super.mouseMoved(e);
68     }
69
70     @Override
71     public void mousePressed(MouseEvent e) {
72         System.out.println("Se ha Presionado el ratón");
73         super.mousePressed(e);
74     }
75
76     @Override
77     public void mouseReleased(MouseEvent e) {
78         System.out.println("Se ha Soltado la tecla del ratón");
79         super.mouseReleased(e);
80     }
81
82     @Override
83     public void mouseWheelMoved(MouseWheelEvent e) {
84         System.out.println("Se ha Movido la Rueda del ratón");
85         super.mouseWheelMoved(e);
86     }
87 }
```



```
1 package Graficos;
2 import java.awt.Color;
3 import java.awt.GridBagLayout;
4 import java.awt.GridLayout;
5 import java.awt.event.FocusEvent;
6 import java.awt.event.FocusListener;
7 import javax.swing.JFrame;
8 import javax.swing.JLabel;
9 import javax.swing.JPanel;
10 import javax.swing.JTextField;
11
12 public class _08_FocoEvento {
13
14     public static JLabel aviso;
15
16     public static void main(String[] args) {
17
18         Color fondo = new Color(230, 204, 255);
19         JTextField email;
20
21         JFrame ventana = new JFrame();
22         ventana.setTitle(" Foco Evento");
23         ventana.setSize(600, 400);
24         ventana.setLocationRelativeTo(null);
25         ventana.setLayout(new GridBagLayout());
26         ventana.getContentPane().setBackground(fondo);
27
28         JPanel lamina = new JPanel();
29         lamina.setBackground(fondo);
30
31         CampoFoco oyenteFoco = new CampoFoco();
32
33         lamina.setLayout(new GridLayout(10,1));
34
35         lamina.add(new JLabel("eMail: ")).setForeground(Color.GRAY.darker());
36         lamina.add(email = new JTextField(30));
37         email.addFocusListener(oyenteFoco);
38         lamina.add(new JLabel("")).setForeground(Color.GRAY.darker());
39         lamina.add(avisos = new JLabel("
40         lamina.add(new JLabel("")).setForeground(Color.GRAY.darker());
41         lamina.add(new JLabel("Contraseña: ")).setForeground(Color.GRAY.darker());
42         lamina.add(new JTextField(30));
43         lamina.add(new JLabel("")).setForeground(Color.GRAY.darker()); //
44         Estos labels son sólo para aumentar espacio
45         lamina.add(new JLabel("")).setForeground(Color.GRAY.darker());
46         lamina.add(new JLabel("")).setForeground(Color.GRAY.darker());
47
48         ventana.add(lamina);
49
50         ventana.setVisible(true);
51         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
52     }
53
54     static class CampoFoco implements FocusListener {
55
56         @Override
57         public void focusGained(FocusEvent e) {
58             // TODO Auto-generated method stub
59         }
60
61         @Override
```

```
61     public void focusLost(FocusEvent e) {
62         JTextField emailObj = (JTextField) e.getSource();
63         String emailTxt = emailObj.getText();
64         int arroba = 0;
65
66         boolean punto = false;
67
68         for (int i=0; i<emailTxt.length(); i++) {
69             if (emailTxt.charAt(i) == '@') {
70                 arroba++;
71             }
72             if (emailTxt.charAt(i) == '.') {
73                 punto = true;
74             }
75         }
76
77         if (arroba == 1 && punto) {
78             aviso.setText(" ");
79             System.out.println("Email correcto");
80         } else {
81             aviso.setText("eMail Incorrecto, corrija e intente nuevamente");
82             System.out.println("Email Incorrecto");
83             arroba = 0;
84         }
85     }
86 }
87 }
```

_09_VariosOyentes.java

```
1 package Graficos;
2
3 import java.awt.Color;
4 import java.awt.Font;
5 import java.awt.GridBagLayout;
6 import java.awt.event.ActionEvent;
7 import java.awt.event.ActionListener;
8 import java.util.Random;
9 import javax.swing.JButton;
10 import javax.swing.JFrame;
11 import javax.swing.JLabel;
12 import javax.swing.JPanel;
13
14 public class _09_VariosOyentes {
15
16     public static void main(String[] args) {
17         Ventana9 ventana = new Ventana9();
18     }
19 }
20
21 class Ventana9 extends JFrame {
22     public Ventana9() {
23         setSize(400, 300);
24         setTitle(" Varios Oyentes");
25         setBounds(1200, 200, 400, 300);
26         add(new Lamina9());
27         setVisible(true);
28         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
29     }
30 }
31
32 class Lamina9 extends JPanel {
33     JButton btn1, btn2;
34     public Lamina9() {
35         setBackground(new Color(255, 255, 153));
36         setLayout(new GridBagLayout());
37         add(btn1 = new JButton("Nueva Ventana"));
38         add(btn2 = new JButton("Cerrar Ventanas"));
39         btn1.addActionListener(new OyenteListener());
40     }
41
42     public class OyenteListener implements ActionListener {
43         @Override
44         public void actionPerformed(ActionEvent e) {
45             VentanaEmergente marco = new VentanaEmergente(btn2);
46         }
47     }
48
49     int numero = 1;
50
51     class VentanaEmergente extends JFrame {
52
53         public VentanaEmergente(JButton btnCerrar) {
54             setTitle("Ventana " + numero);
55             setBounds(40*numero, 40*numero, 250, 200);
56
57             add(new NombreAleatorio());
58
59             setVisible(true);
60             numero++;
61             btnCerrar.addActionListener(new OyenteCerrar());
62         }
63     }
64 }
```

```
63
64     class OyenteCerrar implements ActionListener {
65         @Override
66         public void actionPerformed(ActionEvent e) {
67             dispose();
68             numero = 1;
69         }
70     }
71 }
72 }
73
74 class NombreAleatorio extends JPanel {
75
76     JLabel nombre;
77
78     Random random = new Random();
79     int aleatorio1 = random.nextInt(100)+155;
80     int aleatorio2 = random.nextInt(100)+155;
81     int aleatorio3 = random.nextInt(100)+155;
82
83     int vocalNro1 = random.nextInt(9);
84     int vocalNro2 = random.nextInt(9);
85     int vocalNro3 = random.nextInt(9);
86     int vocalNro4 = random.nextInt(9);
87     int vocalNro5 = random.nextInt(9);
88     int vocalNro6 = random.nextInt(9);
89     int consonanteNro1 = random.nextInt(27);
90     int consonanteNro2 = random.nextInt(27);
91     int consonanteNro3 = random.nextInt(27);
92     int consonanteNro4 = random.nextInt(27);
93     int consonanteNro5 = random.nextInt(27);
94     int consonanteNro6 = random.nextInt(27);
95
96     String vocales[] = {"A","E","I","O","U","AA","EE","II","OO","UU"};
97     String consonantes[] =
98         {"B","C","D","F","G","H","J","K","L","M","N","Ñ","P","Q","R","S","T","V","W","X","Y","Z","L",
99         "L","BB","RR","MM","PP","SS",};
100
101     String vocalRandom1;
102     String vocalRandom2;
103     String vocalRandom3;
104     String vocalRandom4;
105     String vocalRandom5;
106     String vocalRandom6;
107     String consonanteRandom1;
108     String consonanteRandom2;
109     String consonanteRandom3;
110     String consonanteRandom4;
111     String consonanteRandom5;
112     String consonanteRandom6;
113
114     public NombreAleatorio() {
115         // -----VOCALES-----
116         for(int i=0; i<vocales.length; i++) {
117             if(i == vocalNro1) {
118                 vocalRandom1 = vocales[i];
119             }
120         }
121         for(int i=0; i<vocales.length; i++) {
122             if(i == vocalNro2) {
123                 vocalRandom2 = vocales[i];
```

```
123     }
124 }
125 for(int i=0; i<vocales.length; i++) {
126     if(i == vocalNro3) {
127         vocalRandom3 = vocales[i];
128     }
129 }
130 for(int i=0; i<vocales.length; i++) {
131     if(i == vocalNro4) {
132         vocalRandom4 = vocales[i];
133     }
134 }
135 for(int i=0; i<vocales.length; i++) {
136     if(i == vocalNro5) {
137         vocalRandom5 = vocales[i];
138     }
139 }
140 for(int i=0; i<vocales.length; i++) {
141     if(i == vocalNro6) {
142         vocalRandom6 = vocales[i];
143     }
144 }
145 // -----CONSONANTES-----
146 for(int i=0; i<consonantes.length; i++) {
147     if(i == consonanteNro1) {
148         consonanteRandom1 = consonantes[i];
149     }
150 }
151 for(int i=0; i<consonantes.length; i++) {
152     if(i == consonanteNro2) {
153         consonanteRandom2 = consonantes[i];
154     }
155 }
156 for(int i=0; i<consonantes.length; i++) {
157     if(i == consonanteNro3) {
158         consonanteRandom3 = consonantes[i];
159     }
160 }
161 for(int i=0; i<consonantes.length; i++) {
162     if(i == consonanteNro4) {
163         consonanteRandom4 = consonantes[i];
164     }
165 }
166 for(int i=0; i<consonantes.length; i++) {
167     if(i == consonanteNro5) {
168         consonanteRandom5 = consonantes[i];
169     }
170 }
171 for(int i=0; i<consonantes.length; i++) {
172     if(i == consonanteNro6) {
173         consonanteRandom6 = consonantes[i];
174     }
175 }
176
177 setBackground(new Color(aleatorio1, aleatorio2, aleatorio3));
178 System.out.println("El color RGB de la ventana nueva es: " + aleatorio1 + " " +
    aleatorio2 + " " + aleatorio3);
179 System.out.println("Letras aleatorias: " + vocalRandom1 + consonanteRandom1 +
    vocalRandom2 + consonanteRandom2 + vocalRandom3);
180
181 setLayout(new GridBagLayout());
182
```

_09_VariosOyentes.java

```
183         add(nombre = new JLabel(vocalRandom1 + consonanteRandom1 + vocalRandom2 +  
consonanteRandom2 + vocalRandom3));  
184         nombre.setFont(new Font("Roboto", Font.PLAIN, 24));  
185     }  
186 }
```

_10_Layouts.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.FlowLayout;
6 import javax.swing.JButton;
7 import javax.swing.JFrame;
8 import javax.swing.JPanel;
9
10 public class _10_Layouts {
11
12     public static void main(String[] args) {
13
14         Ventana10 ventana = new Ventana10();
15     }
16 }
17
18 class Ventana10 extends JFrame {
19
20     public Ventana10() {
21
22         setSize(600, 500);
23         setTitle(" Layouts");
24         setLocationRelativeTo(null);
25
26         add(new Lamina10());
27
28         setVisible(true);
29         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
30     }
31 }
32
33 class Lamina10 extends JPanel {
34
35     public Lamina10() {
36         setLayout(new BorderLayout());
37         setBackground(new Color(204, 255, 102));
38
39         JPanel lamina1 = new JPanel();
40         add(lamina1, BorderLayout.NORTH);
41         lamina1.setLayout(new BorderLayout());
42
43         lamina1.add(new JButton("Botón 1"), BorderLayout.NORTH);
44         lamina1.add(new JButton("Botón 2"), BorderLayout.SOUTH);
45         lamina1.add(new JButton("Botón 3"), BorderLayout.EAST);
46         lamina1.add(new JButton("Botón 4"), BorderLayout.WEST);
47         lamina1.add(new JButton("Botón 5"), BorderLayout.CENTER);
48
49         JPanel lamina2 = new JPanel();
50         add(lamina2, BorderLayout.SOUTH);
51         lamina2.setLayout(new FlowLayout(FlowLayout.RIGHT));
52
53         lamina2.add(new JButton("Botón 1"));
54         lamina2.add(new JButton("Botón 2"));
55         lamina2.add(new JButton("Botón 3"));
56     }
57 }
```

_11_Calculadora.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.Dimension;
6 import java.awt.Font;
7 import java.awt.GridLayout;
8 import java.awt.event.ActionEvent;
9 import java.awt.event.ActionListener;
10 import javax.swing.JButton;
11 import javax.swing.JFrame;
12 import javax.swing.JPanel;
13 import javax.swing.SwingConstants;
14
15 public class _11_Calculadora {
16
17     public static void main(String[] args) {
18
19         Ventana11 ventana = new Ventana11();
20     }
21 }
22
23 class Ventana11 extends JFrame {
24
25     public Ventana11() {
26         setTitle(" Calculadora");
27         setSize(300, 350);
28         setLocationRelativeTo(null);
29
30         add(new Lamina11());
31
32         setVisible(true);
33         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
34     }
35 }
36
37 class Lamina11 extends JPanel {
38
39     JPanel lamina1, lamina2;
40     JButton display, btn1, btn2, btn3, btn4, btn5, btn6, btn7, btn8, btn9, btn0, btnSum,
41         btnRes, btnMul, btnDiv, btnClr, btnEqu;
42
43     public Lamina11() {
44         setLayout(new BorderLayout());
45
46         add(lamina1 = new JPanel(), BorderLayout.NORTH);
47         lamina1.setLayout(new BorderLayout());
48         lamina1.add(display = new JButton("0"));
49         display.setHorizontalAlignment(SwingConstants.RIGHT);
50         display.setFont(new Font("Arial", Font.PLAIN, 24));
51         display.setPreferredSize(new Dimension(300, 80));
52         display.setEnabled(false);
53
54         add(lamina2 = new JPanel(), BorderLayout.CENTER);
55         lamina2.setLayout(new GridLayout(4, 4));
56         lamina2.add(btn1 = new JButton("1"));
57         lamina2.add(btn2 = new JButton("2"));
58         lamina2.add(btn3 = new JButton("3"));
59         lamina2.add(btnSum = new JButton("+"));
60         lamina2.add(btn4 = new JButton("4"));
61         lamina2.add(btn5 = new JButton("5"));
62         lamina2.add(btn6 = new JButton("6"));
```


_11_Calculadora.java

```
62     lamina2.add(btnRes = new JButton("-"));
63     lamina2.add(btn7 = new JButton("7"));
64     lamina2.add(btn8 = new JButton("8"));
65     lamina2.add(btn9 = new JButton("9"));
66     lamina2.add(btnMul = new JButton("*"));
67     lamina2.add(btnClr = new JButton("clr"));
68     lamina2.add(btn0 = new JButton("0"));
69     lamina2.add(btnEqu = new JButton("="));
70     lamina2.add(btnDiv = new JButton("/"));
71
72     OyenteNumeros oyenteNro = new OyenteNumeros();
73
74     btn1.addActionListener(oyenteNro);
75     btn2.addActionListener(oyenteNro);
76     btn3.addActionListener(oyenteNro);
77     btn4.addActionListener(oyenteNro);
78     btn5.addActionListener(oyenteNro);
79     btn6.addActionListener(oyenteNro);
80     btn7.addActionListener(oyenteNro);
81     btn8.addActionListener(oyenteNro);
82     btn9.addActionListener(oyenteNro);
83     btn0.addActionListener(oyenteNro);
84
85     OyenteOperandos oyenteOperando = new OyenteOperandos();
86
87     btnSum.addActionListener(oyenteOperando);
88     btnRes.addActionListener(oyenteOperando);
89     btnMul.addActionListener(oyenteOperando);
90     btnDiv.addActionListener(oyenteOperando);
91     btnClr.addActionListener(oyenteOperando);
92     btnEqu.addActionListener(oyenteOperando);
93
94     btn1.setBackground(new Color(230, 255, 204));
95     btn2.setBackground(new Color(230, 255, 204));
96     btn3.setBackground(new Color(230, 255, 204));
97     btn4.setBackground(new Color(230, 255, 204));
98     btn5.setBackground(new Color(230, 255, 204));
99     btn6.setBackground(new Color(230, 255, 204));
100    btn7.setBackground(new Color(230, 255, 204));
101    btn8.setBackground(new Color(230, 255, 204));
102    btn9.setBackground(new Color(230, 255, 204));
103    btn0.setBackground(new Color(230, 255, 204));
104
105    btnSum.setBackground(new Color(230, 204, 255));
106    btnRes.setBackground(new Color(230, 204, 255));
107    btnMul.setBackground(new Color(230, 204, 255));
108    btnDiv.setBackground(new Color(230, 204, 255));
109    btnClr.setBackground(new Color(230, 204, 255));
110    btnEqu.setBackground(new Color(230, 204, 255));
111 }
112
113 class OyenteNumeros implements ActionListener {
114     @Override
115     public void actionPerformed(ActionEvent e) {
116
117         JButton btnNro = (JButton) e.getSource();
118         double nro = Double.parseDouble(btnNro.getText());
119
120         if(display.getText() == "0") {
121             display.setText("");
122             display.setText(btnNro.getText());
123         } else {
```

```
124         display.setText(display.getText() + btnNro.getText());
125     }
126 }
127 }
128
129 class OyenteOperandos implements ActionListener {
130
131     double resultado = 0;
132     String simbolo = "";
133
134     @Override
135     public void actionPerformed(ActionEvent e) {
136
137         JButton operandoBtn = (JButton) e.getSource();
138         String operando = operandoBtn.getText();
139
140         if(operando.equals("+")) {
141             resultado = resultado + Double.parseDouble(display.getText());
142             display.setText("0");
143             simbolo = "+";
144         }
145         if(operando.equals("-")) {
146             if(resultado == 0) {
147                 resultado = Double.parseDouble(display.getText());
148                 display.setText("0");
149                 simbolo = "-";
150             } else {
151                 resultado = resultado - Double.parseDouble(display.getText());
152                 display.setText("0");
153                 simbolo = "-";
154             }
155         }
156         if(operando.equals("*")) {
157             if(resultado == 0) {
158                 resultado = (resultado+1) * Double.parseDouble(display.getText());
159                 display.setText("0");
160                 simbolo = "*";
161             } else {
162                 resultado = resultado * Double.parseDouble(display.getText());
163                 display.setText("0");
164                 simbolo = "*";
165             }
166         }
167         if(operando.equals("/")) {
168             if(resultado == 0) {
169                 resultado = Double.parseDouble(display.getText());
170                 display.setText("0");
171                 simbolo = "/";
172             } else {
173                 resultado = resultado / Double.parseDouble(display.getText());
174                 display.setText("0");
175                 simbolo = "/";
176             }
177         }
178
179         if(operando.equals("=")) {
180             if(simbolo.equals("+")) {
181                 resultado = resultado + Double.parseDouble(display.getText());
182                 display.setText(String.valueOf(resultado));
183                 resultado = 0;
184             }
185             if(simbolo.equals("-")) {
```

_11_Calculadora.java

```
186         resultado = resultado - Double.parseDouble(display.getText());
187         display.setText(String.valueOf(resultado));
188         resultado = 0;
189     }
190     if(simbolo.equals("*")) {
191         resultado = resultado * Double.parseDouble(display.getText());
192         display.setText(String.valueOf(resultado));
193         resultado = 0;
194     }
195     if(simbolo.equals("/")) {
196         resultado = resultado / Double.parseDouble(display.getText());
197         display.setText(String.valueOf(resultado));
198         resultado = 0;
199     }
200 }
201
202 if(operando.equals("clr")) {
203     resultado = 0;
204     display.setText("0");
205     simbolo = " ";
206 }
207 }
208 }
209 }
```

_12_PruebaTexto.java

```
1 package Graficos;
2
3 import java.awt.Color;
4 import java.awt.GridBagConstraints;
5 import java.awt.GridBagLayout;
6 import java.awt.GridLayout;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9 import javax.swing.JButton;
10 import javax.swing.JFrame;
11 import javax.swing.JLabel;
12 import javax.swing.JPanel;
13 import javax.swing.JTextField;
14
15 public class _12_PruebaTexto {
16
17     public static void main(String[] args) {
18
19         Marco12 ventana = new Marco12();
20         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21     }
22 }
23
24 class Marco12 extends JFrame {
25
26     Color fondo = new Color(51, 204, 204);
27
28     public Marco12() {
29
30         setSize(600, 500);
31         setTitle(" Prueba Texto");
32         setLocationRelativeTo(null);
33         getContentPane().setBackground(fondo);
34
35         setLayout(new GridBagLayout());
36         GridBagConstraints gbc = new GridBagConstraints();
37         gbc.insets.set(0, 0, 60, 0);
38
39         add(new Ventana12(), gbc);
40
41         setVisible(true);
42     }
43
44     class Ventana12 extends JPanel {
45
46         JTextField campoTexto;
47         JLabel aviso;
48         JButton btn;
49
50         public Ventana12() {
51
52             setBackground(fondo);
53
54             setLayout(new GridLayout(7,1));
55
56             add(new JLabel("Ingrese su eMail:"));
57             add(new JLabel(""));
58             add(campoTexto = new JTextField(30));
59             add(new JLabel(""));
60             add(aviso = new JLabel(""));
61             add(new JLabel(""));
62             add(btn = new JButton("Enviar"));
```

```
63
64     btn.addActionListener(new ClaseOyente());
65 }
66
67 class ClaseOyente implements ActionListener {
68
69     @Override
70     public void actionPerformed(ActionEvent e) {
71
72         int arroba = 0;
73         boolean punto = false;
74
75         for (int i = 0; i < campoTexto.getText().length(); i++) {
76             if(campoTexto.getText().charAt(i) == '@') {
77                 arroba++;
78             }
79             if(campoTexto.getText().charAt(i) == '.') {
80                 punto = true;
81             }
82         }
83         if(arroba == 1 && punto) {
84             aviso.setForeground(new Color(102, 153, 0));
85             aviso.setText("eMail correcto, Bienvenido!");
86             arroba = 0;
87         } else {
88             aviso.setForeground(new Color(255, 0, 102));
89             aviso.setText("eMail incorrecto, corrija e intente nuevamente");
90             arroba = 0;
91         }
92     }
93 }
94 }
95 }
```

_13_CampoPassword.java

```
1 package Graficos;
2
3 import java.awt.Color;
4 import java.awt.GridBagLayout;
5 import java.awt.GridLayout;
6 import javax.swing.JButton;
7 import javax.swing.JFrame;
8 import javax.swing.JLabel;
9 import javax.swing.JPanel;
10 import javax.swing.JPasswordField;
11 import javax.swing.JTextField;
12 import javax.swing.event.DocumentEvent;
13 import javax.swing.event.DocumentListener;
14
15 public class _13_CampoPassword {
16
17     public static void main(String[] args) {
18
19         Marco13 ventana = new Marco13();
20         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21     }
22 }
23
24 class Marco13 extends JFrame {
25
26     public Marco13() {
27
28         setSize(550, 600);
29         setTitle(" Campo Password");
30         setLocationRelativeTo(null);
31
32         add(new Ventana13());
33
34         setVisible(true);
35     }
36 }
37
38 class Ventana13 extends JPanel {
39
40     JLabel emailTxt, passTxt, avisoMail, avisoPass;
41     JTextField emailField;
42     JPasswordField passField;
43     JButton enviarBtn;
44     JPanel lamina;
45     Color fondo = new Color(204, 255, 153);
46
47     public Ventana13() {
48
49         setBackground(fondo);
50
51         setLayout(new GridBagLayout());
52
53         add(lamina = new JPanel());
54         lamina.setBackground(fondo);
55         lamina.setLayout(new GridLayout(10,1));
56
57         lamina.add(emailTxt = new JLabel("Email:"));
58         lamina.add(emailField = new JTextField(40));
59         lamina.add(avisoMail = new JLabel(" "));
60         lamina.add(new JLabel(" "));
61         lamina.add(passTxt = new JLabel("Contraseña:"));
62         lamina.add(passField = new JPasswordField(40));
```

_13_CampoPassword.java

```
63     lamina.add(avisopass = new JLabel(" "));
64     lamina.add(new JLabel(" "));
65     lamina.add(enviarBtn = new JButton("Enviar"));
66
67     emailField.getDocument().addDocumentListener(new OyenteEmail());
68     passField.getDocument().addDocumentListener(new OyentePassword());
69 }
70
71 // -----OYENTE EMAIL-----
72 class OyenteEmail implements DocumentListener {
73     @Override
74     public void insertUpdate(DocumentEvent e) {
75         String email = emailField.getText();
76         int arroba = 0;
77         boolean punto = false;
78
79         for (int i = 0; i < email.length(); i++) {
80             if (email.charAt(i) == '@') {
81                 arroba++;
82             }
83             if (email.charAt(i) == '.') {
84                 punto = true;
85             }
86         }
87         if (email.length() > 8) {
88             if (arroba == 1 && punto) {
89                 emailField.setBackground(Color.WHITE);
90                 avisoMail.setText(" ");
91                 arroba = 0;
92             } else {
93                 emailField.setBackground(new Color(255, 102, 153));
94                 avisoMail.setForeground(new Color(255, 102, 153));
95                 avisoMail.setText("eMail incorrecto, corrija!");
96             }
97         }
98     }
99     @Override
100    public void removeUpdate(DocumentEvent e) {
101        String email = emailField.getText();
102        int arroba = 0;
103        boolean punto = false;
104
105        for (int i = 0; i < email.length(); i++) {
106            if (email.charAt(i) == '@') {
107                arroba++;
108            }
109            if (email.charAt(i) == '.') {
110                punto = true;
111            }
112        }
113        if (email.length() > 8) {
114            if (arroba == 1 && punto) {
115                emailField.setBackground(Color.WHITE);
116                avisoMail.setText(" ");
117                arroba = 0;
118            } else {
119                emailField.setBackground(new Color(255, 102, 153));
120                avisoMail.setForeground(new Color(255, 102, 153));
121                avisoMail.setText("eMail incorrecto, corrija!");
122            }
123        }
124    }
```

_13_CampoPassword.java

```
125     @Override
126     public void changedUpdate(DocumentEvent e) {
127         String email = emailField.getText();
128         int arroba = 0;
129         boolean punto = false;
130
131         for (int i = 0; i < email.length(); i++) {
132             if (email.charAt(i) == '@') {
133                 arroba++;
134             }
135             if (email.charAt(i) == '.') {
136                 punto = true;
137             }
138         }
139         if (email.length() > 8) {
140             if (arroba == 1 && punto) {
141                 emailField.setBackground(Color.WHITE);
142                 avisoMail.setText(" ");
143                 arroba = 0;
144             } else {
145                 emailField.setBackground(new Color(255, 102, 153));
146                 avisoMail.setForeground(new Color(255, 102, 153));
147                 avisoMail.setText("eMail incorrecto, corrija!");
148             }
149         }
150     }
151 }
152
153 // -----OYENTE
154 class OyentePassword implements DocumentListener {
155     @Override
156     public void insertUpdate(DocumentEvent e) {
157         char password[] = passField.getPassword();
158         for (int i = 0; i < password.length; i++) {
159             if (password.length < 6 || password.length > 12) {
160                 passField.setBackground(new Color(255, 102, 153));
161                 avisoPass.setForeground(new Color(255, 102, 153));
162                 avisoPass.setText("La contraseña debe tener mínimo 6 y máximo 12
163 letras!");
164             } else {
165                 passField.setBackground(Color.WHITE);
166                 avisoPass.setText(" ");
167             }
168         }
169     }
170     @Override
171     public void removeUpdate(DocumentEvent e) {
172         char password[] = passField.getPassword();
173         for (int i = 0; i < password.length; i++) {
174             if (password.length < 6 || password.length > 12) {
175                 passField.setBackground(new Color(255, 102, 153));
176                 avisoPass.setForeground(new Color(255, 102, 153));
177                 avisoPass.setText("La contraseña debe tener mínimo 6 y máximo 12
178 letras!");
179             } else {
180                 passField.setBackground(Color.WHITE);
181                 avisoPass.setText(" ");
182             }
183         }
184     }
185     @Override
```


_13_CampoPassword.java

```
184     public void changedUpdate(DocumentEvent e) {
185         char password[] = passField.getPassword();
186         for (int i = 0; i < password.length; i++) {
187             if(password.length < 6 || password.length > 12) {
188                 passField.setBackground(new Color(255, 102, 153));
189                 avisoPass.setForeground(new Color(255, 102, 153));
190                 avisoPass.setText("La contraseña debe tener mínimo 6 y máximo 12
letras!");
191             } else {
192                 passField.setBackground(Color.WHITE);
193                 avisoPass.setText(" ");
194             }
195         }
196     }
197 }
198 }
```

_14_EjemploArea.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.FlowLayout;
6 import java.awt.event.ActionEvent;
7 import java.awt.event.ActionListener;
8 import javax.swing.JButton;
9 import javax.swing.JFrame;
10 import javax.swing.JLabel;
11 import javax.swing.JPanel;
12 import javax.swing.JScrollPane;
13 import javax.swing.JTextArea;
14
15 public class _14_EjemploArea {
16
17     public static void main(String[] args) {
18
19         Marco14 ventana = new Marco14();
20         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21     }
22 }
23
24 class Marco14 extends JFrame {
25
26     public Marco14() {
27
28         setSize(350, 500);
29         setTitle(" Ejemplo Área");
30         setLocationRelativeTo(null);
31         getContentPane().setBackground(new Color(153, 204, 255));
32
33         setLayout(new BorderLayout());
34
35         // -----Espacios en los Bordes-----
36         add(new JLabel(" "), BorderLayout.NORTH);
37         add(new JLabel(" "), BorderLayout.SOUTH);
38         add(new JLabel(" "), BorderLayout.EAST);
39         add(new JLabel(" "), BorderLayout.WEST);
40         // -----Espacios en los Bordes-----
41
42         add(new Ventana14(), BorderLayout.CENTER);
43
44         setVisible(true);
45     }
46 }
47
48 class Ventana14 extends JPanel {
49
50     JPanel lamina;
51     JTextArea areaIn, areaOut;
52     JLabel textoOut;
53     JButton btnEnviar;
54     JScrollPane vistaScroll;
55
56     public Ventana14() {
57
58         setBackground(new Color(153, 204, 255));
59
60         setLayout(new FlowLayout(FlowLayout.LEFT, 20,20));
61
62         add(new JLabel(" "));
```

_14_EjemploArea.java

```
63     add(new JLabel(" "));
64     add(areaIn = new JTextArea(8,20));
65     areaIn.setText(lorem);
66     areaIn.setLineWrap(true);
67     add(vistaScroll = new JScrollPane(areaIn));
68     add(btnEnviar = new JButton("Enviar"));
69     add(new JLabel(" "));
70     add(new JLabel(" "));
71     add(areaOut = new JTextArea(8,20));
72     areaOut.setLineWrap(true);
73     areaOut.setEnabled(false);
74     areaOut.setDisabledTextColor(Color.DARK_GRAY);
75     areaOut.setBackground(new Color(179, 204, 230));
76     add(vistaScroll = new JScrollPane(areaOut));
77     add(textoOut = new JLabel(" "));
78
79     btnEnviar.addActionListener(new ClaseOyente());
80 }
81
82 class ClaseOyente implements ActionListener {
83
84     @Override
85     public void actionPerformed(ActionEvent e) {
86
87         String entrada = areaIn.getText();
88         areaOut.setText(entrada);
89
90         textoOut.setText("Data enviada!");
91     }
92 }
93
94 String lorem = "There are many variations of passages of Lorem Ipsum available, but the
majority have suffered alteration in some form, by injected humour, or randomised words
which don't look even slightly believable. If you are going to use a passage of Lorem Ipsum,
you need to be sure there isn't anything embarrassing hidden in the middle of text. All the
Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making
this the first true generator on the Internet. It uses a dictionary of over 200 Latin words,
combined with a handful of model sentence structures, to generate Lorem Ipsum which looks
reasonable. The generated Lorem Ipsum is therefore always free from repetition, injected
humour, or non-characteristic words etc.";
95 }
```

_15_PruebaArea.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7 import javax.swing.JButton;
8 import javax.swing.JFrame;
9 import javax.swing.JPanel;
10 import javax.swing.JScrollPane;
11 import javax.swing.JTextArea;
12
13 public class _15_PruebaArea {
14
15     public static void main(String[] args) {
16
17         Marco15 ventana = new Marco15();
18         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19     }
20 }
21
22 class Marco15 extends JFrame {
23
24     public Marco15() {
25
26         setSize(400, 300);
27         setTitle(" Prueba Área");
28         setLocationRelativeTo(null);
29
30         add(new Ventana15());
31
32         setVisible(true);
33     }
34 }
35
36 class Ventana15 extends JPanel {
37
38     JTextArea areaTxt;
39     JScrollPane vistaScroll;
40     JPanel lamina;
41     JButton btn1, btn2;
42     String lorem = "Al contrario del pensamiento popular, el texto de Lorem Ipsum no es
43 simplemente texto aleatorio. | Al contrario del pensamiento popular, el texto de Lorem Ipsum
44 no es simplemente texto aleatorio. | Al contrario del pensamiento popular, el texto de Lorem
45 Ipsum no es simplemente texto aleatorio. | Al contrario del pensamiento popular, el texto de
46 Lorem Ipsum no es simplemente texto aleatorio. | ";
47
48     public Ventana15() {
49
50         setLayout(new BorderLayout());
51
52         add(areaTxt = new JTextArea(), BorderLayout.CENTER);
53         areaTxt.setBackground(new Color(204, 255, 255));
54         areaTxt.setForeground(new Color(153, 0, 115));
55         add(vistaScroll = new JScrollPane(areaTxt));
56
57         add(lamina = new JPanel(), BorderLayout.SOUTH);
58         lamina.setBackground(new Color(204, 204, 255));
59
60         lamina.add(btn1 = new JButton("Agregar texto"));
61         lamina.add(btn2 = new JButton("Insertar salto de línea"));
```

_15_PruebaArea.java

```
59     btn1.setBackground(new Color(204, 204, 255));
60     btn2.setBackground(new Color(204, 204, 255));
61
62     btn1.addActionListener(new ClaseOyente());
63     btn2.addActionListener(new ClaseOyente());
64 }
65
66 class ClaseOyente implements ActionListener {
67
68     @Override
69     public void actionPerformed(ActionEvent e) {
70
71         if (e.getSource() == btn1) {
72             areaTxt.append(lorem);
73         }
74         if (e.getSource() == btn2) {
75             if (areaTxt.getLineWrap() == false) {
76                 areaTxt.setLineWrap(true);
77                 btn2.setText("Quitar salto de línea");
78                 btn2.setBackground(new Color(255, 153, 204));
79             } else {
80                 areaTxt.setLineWrap(false);
81                 btn2.setText("Insertar salto de línea");
82                 btn2.setBackground(new Color(153, 255, 204));
83             }
84         }
85     }
86 }
87 }
```

_16_PruebaChecks.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.Font;
6 import java.awt.GridLayout;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9 import javax.swing.JCheckBox;
10 import javax.swing.JFrame;
11 import javax.swing.JLabel;
12 import javax.swing.JPanel;
13 import javax.swing.SwingConstants;
14
15 public class _16_PruebaChecks {
16
17     public static void main(String[] args) {
18
19         Marco16 ventana = new Marco16();
20         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21     }
22 }
23
24 class Marco16 extends JFrame {
25
26     JLabel east;
27     Color fondo = new Color(179, 255, 179);
28
29     public Marco16() {
30
31         setSize(900, 400);
32         setTitle(" Prueba Checks");
33         setLocationRelativeTo(null);
34
35         setLayout(new BorderLayout());
36
37         // -----Espacios en los Bordres-----
38         add(new JLabel(" "), BorderLayout.NORTH);
39         add(new JLabel(" "), BorderLayout.SOUTH);
40         add(east = new JLabel(" ..... "), BorderLayout.EAST);
41         east.setForeground(new Color(0, 0, 0, 0));
42         add(new JLabel(" "), BorderLayout.WEST);
43         // -----Espacios en los Bordres-----
44
45         add(new Ventana16(fondo), BorderLayout.CENTER);
46
47         getContentPane().setBackground(fondo);
48
49         setVisible(true);
50     }
51 }
52
53 class Ventana16 extends JPanel {
54
55     JLabel texto1, texto2;
56     JCheckBox check1, check2;
57     JPanel lamina1, lamina2;
58
59     public Ventana16(Color fondo) {
60
61         setLayout(new BorderLayout());
62
63     }
```

_16_PruebaChecks.java

```
63     add(lamina1 = new JPanel(), BorderLayout.CENTER);
64     lamina1.setLayout(new GridLayout(7,1));
65     lamina1.setBackground(fondo);
66
67     lamina1.add(new JLabel(""));
68     lamina1.add(new JLabel(""));
69     lamina1.add(new JLabel(""));
70     lamina1.add(texto1 = new JLabel("\nEs capaz el que piensa que es capaz.\n"));
71     texto1.setFont(new Font("Verdana", Font.PLAIN, 34));
72     texto1.setHorizontalAlignment(SwingConstants.RIGHT);
73     lamina1.add(new JLabel(""));
74     lamina1.add(texto2 = new JLabel("Buda"));
75     texto2.setFont(new Font("Verdana", Font.PLAIN, 24));
76     texto2.setHorizontalAlignment(SwingConstants.RIGHT);
77
78     add(lamina2 = new JPanel(), BorderLayout.SOUTH);
79     lamina2.setBackground(fondo);
80
81     lamina2.add(check1 = new JCheckBox("Negrita", false));
82     check1.setFont(new Font("Verdana", Font.PLAIN, 18));
83     check1.setBackground(fondo);
84     lamina2.add(check2 = new JCheckBox("Cursiva", false));
85     check2.setFont(new Font("Verdana", Font.PLAIN, 18));
86     check2.setBackground(fondo);
87
88     check1.addActionListener(new ClaseOyente());
89     check2.addActionListener(new ClaseOyente());
90 }
91
92 class ClaseOyente implements ActionListener {
93
94     @Override
95     public void actionPerformed(ActionEvent e) {
96
97         int constante = 0;
98
99         if (check1.isSelected() && !check2.isSelected()) {
100             constante = 1;
101         } else if (!check1.isSelected() && check2.isSelected()) {
102             constante = 2;
103         } else if (check1.isSelected() && check2.isSelected()) {
104             constante = 3;
105         } else {
106             constante = 0;
107         }
108
109         texto1.setFont(new Font("Verdana", constante, 34));
110     }
111 }
112 }
```

_17_SintaxisRadio_EjemploRadio_PruebaCombo_MarcoSlider_MarcoSpinner.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.Dimension;
6 import java.awt.Font;
7 import java.awt.GraphicsEnvironment;
8 import java.awt.GridLayout;
9 import java.awt.event.ActionEvent;
10 import java.awt.event.ActionListener;
11 import javax.swing.ButtonGroup;
12 import javax.swing.JComboBox;
13 import javax.swing.JFrame;
14 import javax.swing.JLabel;
15 import javax.swing.JPanel;
16 import javax.swing.JRadioButton;
17 import javax.swing.JSlider;
18 import javax.swing.JSpinner;
19 import javax.swing.SpinnerListModel;
20 import javax.swing.SpinnerNumberModel;
21 import javax.swing.event.ChangeEvent;
22 import javax.swing.event.ChangeListener;
23
24 public class _17_SintaxisRadio_EjemploRadio_PruebaCombo_MarcoSlider_MarcoSpinner {
25
26     public static void main(String[] args) {
27
28         Marco17 ventana = new Marco17();
29         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
30     }
31 }
32
33 class Marco17 extends JFrame {
34
35     Color fondo1 = new Color(179, 255, 224);
36
37     public Marco17() {
38
39         setSize(1000, 900);
40         setTitle(" Ejercicio 3.17");
41         setLocationRelativeTo(null);
42         getContentPane().setBackground(fondo1);
43
44         setLayout(new BorderLayout());
45
46         // -----Espacios en los Bordes-----
47         add(new JLabel(" "), BorderLayout.NORTH);
48         add(new JLabel(" "), BorderLayout.SOUTH);
49         add(new JLabel(" "), BorderLayout.EAST);
50         add(new JLabel(" "), BorderLayout.WEST);
51         // -----Espacios en los Bordes-----
52
53         add(new Ventana17(fondo1), BorderLayout.CENTER);
54
55         setVisible(true);
56     }
57 }
58
59 class Ventana17 extends JPanel {
60
61     JPanel lamina1, lamina2, laminaGrid, laminaGroup1, laminaGroup2, laminaSeparator,
        laminaSpinners;
```


_17_SintaxisRadio_EjemploRadio_PruebaCombo_MarcoSlider_MarcoSpinner.java

```
62     JLabel texto1, texto2;
63     JRadioButton radioPequeño, radioMediano, radioGrande, radioMuyGrande, radioNegrita,
        radioCursiva, radioNormal;
64     JComboBox comboFuentes;
65     JSlider slider;
66     JSpinner spinnerFuente, spinnerTamano;
67     String fuentes[] =
        GraphicsEnvironment.getLocalGraphicsEnvironment().getAvailableFontFamilyNames();
68     String fuenteTipo = "Arial";
69     int fuenteFormat = 0;
70     int fuenteSize = 30;
71
72     public Ventana17(Color fondo1) {
73
74         setBackground(fondo1);
75         add(laminaGrid = new JPanel());
76         laminaGrid.setLayout(new GridLayout(2,1));
77
78         // -----LÁMINA 1-----
79         laminaGrid.add(lamina1 = new JPanel());
80         lamina1.setPreferredSize(new Dimension(800,200));
81         lamina1.setLayout(new GridLayout(2,1));
82         lamina1.setBackground(fondo1);
83
84         // -----TEXTO PRINCIPAL-----
85         lamina1.add(texto1 = new JLabel("\nTodo es un pensamiento.\n"));
86         texto1.setFont(new Font(fuenteTipo, fuenteFormat, fuenteSize));
87         texto1.setHorizontalAlignment(JLabel.CENTER);
88         lamina1.add(texto2 = new JLabel("Buda          "));
89         texto2.setFont(new Font("Arial", Font.PLAIN, 22));
90         texto2.setHorizontalAlignment(JLabel.RIGHT);
91
92         // -----LÁMINA 2-----
93         laminaGrid.add(lamina2 = new JPanel());
94         lamina2.setLayout(new GridLayout(9,1));
95
96         // -----GRUPO 1 RADIOS-----
97         lamina2.add(laminaGroup1 = new JPanel());
98         laminaGroup1.add(radioPequeño = new JRadioButton("Pequeña", false));
99         laminaGroup1.add(radioMediano = new JRadioButton("Mediana", true));
100        laminaGroup1.add(radioGrande = new JRadioButton("Grande", false));
101        laminaGroup1.add(radioMuyGrande = new JRadioButton("Muy Grande", false));
102
103        ButtonGroup grupo1 = new ButtonGroup();
104        grupo1.add(radioPequeño);
105        grupo1.add(radioMediano);
106        grupo1.add(radioGrande);
107        grupo1.add(radioMuyGrande);
108
109        lamina2.add(laminaSeparator = new JPanel());
110        laminaSeparator.add(new JLabel(""));
111        laminaSeparator.add(new JLabel(""));
112
113        // -----GRUPO 2 RADIOS-----
114        lamina2.add(laminaGroup2 = new JPanel());
115        laminaGroup2.add(radioNegrita = new JRadioButton("Negrita", false));
116        laminaGroup2.add(radioCursiva = new JRadioButton("Cursiva", false));
117        laminaGroup2.add(radioNormal = new JRadioButton("Normal", true));
118
119        ButtonGroup grupo2 = new ButtonGroup();
120        grupo2.add(radioNegrita);
121        grupo2.add(radioCursiva);
```

```
122     grupo2.add(radioNormal);
123
124     lamina2.add(laminaSeparator = new JPanel());
125     laminaSeparator.add(new JLabel(""));
126
127     // -----SPINNERS-----
128     lamina2.add(laminaSpinners = new JPanel());
129     laminaSpinners.add(spinnerFuente = new JSpinner(new SpinnerListModel(fuentes)));
130     spinnerFuente.setPreferredSize(new Dimension(150,30));
131
132     laminaSpinners.add(spinnerTamanio = new JSpinner(new SpinnerNumberModel(30, 10, 60,
2)))));
133     spinnerTamanio.setPreferredSize(new Dimension(50,30));
134
135     lamina2.add(laminaSeparator = new JPanel());
136     laminaSeparator.add(new JLabel(""));
137
138     // -----SLIDER-----
139     lamina2.add(slider = new JSlider(10, 60, 30));
140     slider.setMajorTickSpacing(5);
141     slider.setMinorTickSpacing(1);
142     slider.setPaintLabels(true);
143     slider.setPaintTicks(true);
144
145     lamina2.add(laminaSeparator = new JPanel());
146     laminaSeparator.add(new JLabel(""));
147
148     // -----COMBOBOX-----
149     lamina2.add(comboFuentes = new JComboBox());
150     for (int i = 0; i < fuentes.length; i++) {
151         comboFuentes.addItem(fuentes[i]);
152     }
153
154     // -----OYENTES-----
155     ActionOyente oyenteAction = new ActionOyente();
156
157     radioPequeño.addActionListener(oyenteAction);
158     radioMediano.addActionListener(oyenteAction);
159     radioGrande.addActionListener(oyenteAction);
160     radioMuyGrande.addActionListener(oyenteAction);
161     radioNegrita.addActionListener(oyenteAction);
162     radioCursiva.addActionListener(oyenteAction);
163     radioNormal.addActionListener(oyenteAction);
164     comboFuentes.addActionListener(oyenteAction);
165
166     ChangeOyente oyenteChange = new ChangeOyente();
167
168     slider.addChangeListener(oyenteChange);
169     spinnerFuente.addChangeListener(oyenteChange);
170     spinnerTamanio.addChangeListener(oyenteChange);
171 }
172
173 // -----OYENTES ACTION-----
174 class ActionOyente implements ActionListener {
175
176     public ActionOyente() {
177
178     }
179     @Override
180     public void actionPerformed(ActionEvent e) {
181
182         if (e.getSource() == comboFuentes) fuenteTipo =
```

```
(String)comboFuentes.getSelectedItemAt();
183
184         if (radioNegrita.isSelected()) fuenteFormat = 1;
185         if (radioCursiva.isSelected()) fuenteFormat = 2;
186         if (radioNormal.isSelected()) fuenteFormat = 0;
187
188         if (radioPequeño.isSelected()) fuenteSize = 20;
189         if (radioMediano.isSelected()) fuenteSize = 30;
190         if (radioGrande.isSelected()) fuenteSize = 40;
191         if (radioMuyGrande.isSelected()) fuenteSize = 50;
192
193         texto1.setFont(new Font(fuenteTipo, fuenteFormat, fuenteSize));
194     }
195 }
196
197 // -----OYENTES CHANGE-----
198 class ChangeOyente implements ChangeListener {
199
200     @Override
201     public void stateChanged(ChangeEvent e) {
202
203         if (e.getSource() == slider) fuenteSize = (int) slider.getValue();
204         if (e.getSource() == spinnerTamanio) fuenteSize = (int)
spinnerTamanio.getValue();
205         if (e.getSource() == spinnerFuente) fuenteTipo = (String)
spinnerFuente.getValue();
206
207         texto1.setFont(new Font(fuenteTipo, fuenteFormat, fuenteSize));
208     }
209 }
210 }
```

_18_MarcoMenu.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import javax.swing.JFrame;
6 import javax.swing.JLabel;
7 import javax.swing.JMenu;
8 import javax.swing.JMenuBar;
9 import javax.swing.JMenuItem;
10 import javax.swing.JPanel;
11 import javax.swing.JSeparator;
12
13 public class _18_MarcoMenu {
14
15     public static void main(String[] args) {
16
17         Marco18 ventana = new Marco18();
18         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19     }
20 }
21
22 class Marco18 extends JFrame {
23
24     Color fondo = new Color(230, 204, 255);
25
26     public Marco18() {
27
28         setSize(600, 500);
29         setTitle(" Marco Menú");
30         setLocationRelativeTo(null);
31         getContentPane().setBackground(fondo);
32
33         setLayout(new BorderLayout());
34
35         // -----Espacios en los Bordes-----
36         add(new JLabel(" "), BorderLayout.NORTH);
37         add(new JLabel(" "), BorderLayout.SOUTH);
38         add(new JLabel(" "), BorderLayout.EAST);
39         add(new JLabel(" "), BorderLayout.WEST);
40         // -----Espacios en los Bordes-----
41
42         add(new Ventana18(fondo), BorderLayout.CENTER);
43
44         setVisible(true);
45     }
46 }
47
48 class Ventana18 extends JPanel {
49
50     JMenuBar barra;
51     JMenu menu1, menu2, menu3, menu4, menu5;
52     JMenuItem item1, item2, item3, item4, item5, item6, item7, item8, item9, item10;
53     JSeparator separador;
54
55     public Ventana18(Color fondo) {
56
57         setBackground(fondo);
58
59         add(barra = new JMenuBar());
60
61         barra.add(menu1 = new JMenu("Inicio"));
62         barra.add(menu2 = new JMenu("Servicios"));
```

_18_MarcoMenu.java

```
63 barra.add(menu3 = new JMenu("Galería"));
64 barra.add(menu4 = new JMenu("Contacto"));
65
66 menu1.add(item1 = new JMenuItem("Ofertas"));
67 menu1.add(item2 = new JMenuItem("Productos"));
68 menu1.addSeparator();
69 menu1.add(menu5 = new JMenu("Novedades"));
70 menu5.add(item10 = new JMenuItem("Año 2020"));
71
72 menu2.add(item4 = new JMenuItem("Básicos"));
73 menu2.add(item5 = new JMenuItem("Exclusivos"));
74 menu2.add(item6 = new JMenuItem("Outlet"));
75
76 menu3.add(item7 = new JMenuItem("Clientes"));
77 menu3.add(item8 = new JMenuItem("Eventos"));
78
79 menu4.add(item9 = new JMenuItem("Reservas"));
80 }
81 }
```