

Componentes Swing

1. Validar si un mail está correctamente escrito o no. El mail se ha de introducir dentro de un campo de texto y ha de informar la aplicación si es válido o no.

```
package swing;

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Ejercicio1 {

    public static void main(String[] args) {
        ValidarEmail mail = new ValidarEmail();
        mail.setVisible(true);
    }
}

class ValidarEmail extends JFrame {
    private JTextField emailTextField;
    private JLabel resultLabel;

    public ValidarEmail() {
        setTitle("Validación de Correo Electrónico");
        setSize(300, 150);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(3, 1));

        JLabel label = new JLabel("Ingresa un correo electrónico:");
        emailTextField = new JTextField(20);
        JButton validateButton = new JButton("Validar");
        resultLabel = new JLabel();

        validateButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String email = emailTextField.getText();
                if (isValidEmail(email)) {
                    resultLabel.setText("Correo electrónico válido.");
                    resultLabel.setForeground(Color.GREEN);
                } else {
                    resultLabel.setText("Correo electrónico
inválido.");
                    resultLabel.setForeground(Color.RED);
                }
            }
        });

        panel.add(label);
        panel.add(emailTextField);
        panel.add(validateButton);

        add(panel, BorderLayout.CENTER);
    }
}
```

```
        add(resultLabel, BorderLayout.SOUTH);
    }

    private boolean isValidEmail(String email) {
        String emailRegex = "[A-Za-z0-9+_.-]+@(.+)$";
        Pattern pattern = Pattern.compile(emailRegex);
        Matcher matcher = pattern.matcher(email);
        return matcher.matches();
    }
}
```

2. Crear una aplicación que contenga un área de texto en el cuál podamos escribir y el texto convertirlo a negrita o cursiva.

```
package swing;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Ejercicio2 {

    public static void main(String[] args) {
        EditorTexto editor = new EditorTexto();
        editor.setVisible(true);
    }
}

class EditorTexto extends JFrame implements ActionListener {
    private JTextArea textArea;
    private JCheckBox boldCheckBox;
    private JCheckBox italicCheckBox;

    public EditorTexto() {
        setTitle("Editor de texto");
        setSize(600, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        textArea = new JTextArea();
        JScrollPane scrollPane = new JScrollPane(textArea);
        getContentPane().add(scrollPane, BorderLayout.CENTER);

        boldCheckBox = new JCheckBox("Negrita");
        italicCheckBox = new JCheckBox("Cursiva");

        JPanel checkBoxPanel = new JPanel();
        checkBoxPanel.add(boldCheckBox);
        checkBoxPanel.add(italicCheckBox);
        getContentPane().add(checkBoxPanel, BorderLayout.NORTH);

        boldCheckBox.addActionListener(this);
        italicCheckBox.addActionListener(this);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        int style = Font.PLAIN;
```

```
        if (boldCheckBox.isSelected()) {
            style |= Font.BOLD;
        }
        if (italicCheckBox.isSelected()) {
            style |= Font.ITALIC;
        }

        Font font = textArea.getFont();
        Font newFont = new Font(font.getName(), style,
font.getSize());
        textArea.setFont(newFont);
    }
}
```

3. Crea una aplicación con tres checkbox y un botón, al seleccionar un checkbox y pulsar el botón que muestre en una ventana la descripción del checkbox seleccionado.

```
package swing;

import javax.swing.*;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Ejercicio3 {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                createAndShowGUI();
            }
        });
    }

    private static void createAndShowGUI() {
        JFrame frame = new JFrame("Ejercicio Checkbox");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(3, 1)); // 3 filas, 1 columna

        JCheckBox checkBox1 = new JCheckBox("Checkbox 1");
        JCheckBox checkBox2 = new JCheckBox("Checkbox 2");
        JCheckBox checkBox3 = new JCheckBox("Checkbox 3");

        panel.add(checkBox1);
        panel.add(checkBox2);
        panel.add(checkBox3);

        JButton showStatusButton = new JButton("Mostrar estado");
        showStatusButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                StringBuilder status = new StringBuilder();
                status.append("Checkbox 1:");
                status.append(checkBox1.isSelected()).append("\n");
            }
        });
    }
}
```

```
        status.append("Checkbox 2:");
    ").append(checkBox2.isSelected()).append("\n");
        status.append("Checkbox 3:");
    ").append(checkBox3.isSelected()).append("\n");
        JOptionPane.showMessageDialog(frame,
status.toString(), "Estado Checkbox",
JOptionPane.INFORMATION_MESSAGE);
    }
    });

    frame.add(panel, BorderLayout.CENTER);
    frame.add(showStatusButton, BorderLayout.SOUTH);

    frame.pack();
    frame.setVisible(true);
}
}
```

4. Crear una aplicación similar al ejercicio anterior pero en lugar de tres checkbox van a ser tres jradiobutton y un botón, al seleccionar un jradiobutton y pulsar el botón que muestre la descripción del jradiobutton seleccionado

```
package swing;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Ejercicio4 {

    public static void main(String[] args) {

        JRadioButtonEjercicio jradio = new JRadioButtonEjercicio();

        jradio.setVisible(true);
    }
}

class JRadioButtonEjercicio extends JFrame {
    private JRadioButton option1, option2, option3;
    private JButton submitButton;
    private JLabel resultLabel;

    public JRadioButtonEjercicio() {
        setTitle("Ejercicio JRadioButton");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 200);
        setLocationRelativeTo(null);

        option1 = new JRadioButton("Opción 1");
        option2 = new JRadioButton("Opción 2");
        option3 = new JRadioButton("Opción 3");

        ButtonGroup group = new ButtonGroup();
        group.add(option1);
        group.add(option2);
        group.add(option3);

        submitButton = new JButton("Enviar");
    }
}
```

```
resultLabel = new JLabel("Selecciona una opción y pulsa en
Enviar.");

submitButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String result = "Selecciona opción: ";
        if (option1.isSelected()) {
            result += "Opción 1";
        } else if (option2.isSelected()) {
            result += "Opción 2";
        } else if (option3.isSelected()) {
            result += "Opción 3";
        } else {
            result += "No has seleccionado ninguna opción";
        }
        resultLabel.setText(result);
    }
});

setLayout(new GridLayout(5, 1));
add(option1);
add(option2);
add(option3);
add(submitButton);
add(resultLabel);

setVisible(true);
}
```

5. Crear un combobox

```
package swing;

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Ejercicio5 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Ejercicio ComboBox");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);

        JPanel panel = new JPanel();

        JLabel label = new JLabel("Selecciona una opción:");
        panel.add(label);

        String[] opciones = {"Opción 1", "Opción 2", "Opción 3"};
        JComboBox<String> comboBox = new JComboBox<>(opciones);
        panel.add(comboBox);

        JButton boton = new JButton("Mostrar selección");
        boton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String seleccion = (String)
                    comboBox.getSelectedItem();
            }
        });
    }
}
```

```
        JOptionPane.showMessageDialog(frame, "Has  
seleccionado: " + seleccion);  
    }  
    });  
    panel.add(boton);  
  
    frame.add(panel);  
    frame.setVisible(true);  
}  
}
```

6. Crear una aplicación que contenga un JSlider y un texto, de forma que si aumentamos o disminuimos el JSlider el texto cambie de tamaño

```
package swing;  
  
import javax.swing.*;  
import javax.swing.event.ChangeEvent;  
import javax.swing.event.ChangeListener;  
import java.awt.Font;  
  
public class Ejercicio6 {  
    public static void main(String[] args) {  
        SwingUtilities.invokeLater(() -> {  
            JFrame frame = new JFrame("Ejercicio JSlider");  
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
            frame.setSize(300, 200);  
  
            JPanel panel = new JPanel();  
  
            JLabel label = new JLabel("Aprendiendo a programar en  
Java");  
            panel.add(label);  
  
            JSlider slider = new JSlider(JSlider.HORIZONTAL, 10, 50,  
20);  
            slider.setMajorTickSpacing(10);  
            slider.setMinorTickSpacing(1);  
            slider.setPaintTicks(true);  
            slider.setPaintLabels(true);  
            panel.add(slider);  
  
            slider.addChangeListener(new ChangeListener() {  
                public void stateChanged(ChangeEvent e) {  
                    JSlider source = (JSlider) e.getSource();  
                    if (!source.getValueIsAdjusting()) {  
                        int fontSize = (int) source.getValue();  
                        label.setFont(new Font("Arial", Font.PLAIN,  
fontSize));  
                    }  
                }  
            });  
  
            frame.add(panel);  
            frame.setVisible(true);  
        });  
    }  
}
```

7. Crear una aplicación que contenga un JSpinner y un botón, al seleccionar uno de los valores de JSpinner y pulsar el botón que muestre una ventana de JOptionPane y nos informe del valor seleccionado en el JSpinner

```
package swing;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Ejercicio7 {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            createAndShowGUI();
        });
    }

    private static void createAndShowGUI() {
        JFrame frame = new JFrame("Ejercicio JSpinner");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);

        JPanel panel = new JPanel();
        frame.add(panel);

        // Crear un modelo de spinner con valores de 0 a 100 y paso de
1
100, 1);
SpinnerNumberModel spinnerModel = new SpinnerNumberModel(0, 0,
100, 1);
JSpinner spinner = new JSpinner(spinnerModel);

panel.add(spinner);

JButton button = new JButton("Mostrar Valor");
panel.add(button);

button.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Obtener el valor actual del spinner
        int valor = (int) spinner.getValue();
        JOptionPane.showMessageDialog(frame, "Valor
seleccionado: " + valor);
    }
});

frame.setVisible(true);
}
}
```

8. Crear una barra de menús

```
package swing;

import javax.swing.*;
import java.awt.event.*;
```

```
public class Ejercicio8 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Ejercicio Menú");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JMenuBar menuBar = new JMenuBar();
        frame.setJMenuBar(menuBar);

        JMenu fileMenu = new JMenu("Archivo");
        menuBar.add(fileMenu);

        JMenuItem openItem = new JMenuItem("Abrir");
        openItem.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // Lógica para manejar la acción de "Abrir"
                JOptionPane.showMessageDialog(null, "Se ha
seleccionado Abrir");
            }
        });
        fileMenu.add(openItem);

        JMenuItem exitItem = new JMenuItem("Salir");
        exitItem.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // Lógica para manejar la acción de "Salir"
                System.exit(0);
            }
        });
        fileMenu.add(exitItem);

        frame.setSize(300, 200);
        frame.setVisible(true);
    }
}
```

9. Crear un menú emergente

```
package swing;

import javax.swing.*;
import java.awt.event.*;

public class Ejercicio9 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Ejercicio Menú Emergente");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);

        // Crear el menú emergente
        JPopupMenu popupMenu = new JPopupMenu();
        JMenuItem menuItem1 = new JMenuItem("Opción 1");
        JMenuItem menuItem2 = new JMenuItem("Opción 2");
        JMenuItem menuItem3 = new JMenuItem("Opción 3");
        popupMenu.add(menuItem1);
        popupMenu.add(menuItem2);
        popupMenu.add(menuItem3);

        // Agregar el menú emergente a un componente (por ejemplo, un
panel)
        JPanel panel = new JPanel();
```



```
panel.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseReleased(MouseEvent e) {
        if (e.isPopupTrigger()) {
            popupMenu.show(e.getComponent(), e.getX(),
e.getY());
        }
    }
});

frame.add(panel);
frame.setVisible(true);
}
}
```

10. Crear una aplicación que contenga un menú, dos submenús: “Copiar” y “Pegar” y un área de texto. De forma que al escribir en el área de texto podamos copiar y pegar el texto utilizando los atajos de teclado.

```
package swing;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;

public class Ejercicio10 {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> createAndShowGUI());
    }

    private static void createAndShowGUI() {
        JFrame frame = new JFrame("Atajos de Teclado");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JTextArea textArea = new JTextArea();
        frame.add(textArea);

        JMenuBar menuBar = new JMenuBar();
        JMenu editMenu = new JMenu("Editar");
        editMenu.setMnemonic(KeyEvent.VK_E);

        JMenuItem copyMenuItem = new JMenuItem("Copiar");

        copyMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_C,
Toolkit.getDefaultToolkit().getMenuShortcutKeyMask()));
        copyMenuItem.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                textArea.copy();
            }
        });
        editMenu.add(copyMenuItem);

        JMenuItem pasteMenuItem = new JMenuItem("Pegar");
```

```
pasteMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_V,  
Toolkit.getDefaultToolkit().getMenuShortcutKeyMask()));  
pasteMenuItem.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        textArea.paste();  
    }  
});  
editMenu.add(pasteMenuItem);  
  
menuBar.add(editMenu);  
frame.setJMenuBar(menuBar);  
  
frame.setPreferredSize(new Dimension(400, 300));  
frame.pack();  
frame.setVisible(true);  
}  
}
```