

Jtext Field.

Este objeto nos permite crear tiras de texto, de pequeño tamaño, que podemos usar para que el usuario de nuestras aplicaciones teclee algo o para mostrar algún resultado. Se trata de un objeto “editable”.

Para crear un un textfield usamos el constructor:

```
textField1 = new JTextField("texto a mostrar",tamaño);
```

Disponemos de un método `getText()` para recuperar el texto y de otro `setTex(“texto”)` para escribir texto en el textfield.

Podemos usar el método `setFont` para establecer la fuente del textfied, por ejemplo:

```
textField1.setFont( new Font( "Serif", Font.PLAIN, 14 ) );
```

También podemos asignar a los textfield escuchadores de tipo `ActionListener` que normalmente se activan cundo presionamos “enter”.

Veamos un ejemplo:

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

class frame extends JFrame{

    JTextField textField1, textField2;

    JPanel panel = new JPanel();

    public frame(){

        setTitle("Ejemplo de JTextField");

        setSize(300,100);

        setDefaultCloseOperation(EXIT_ON_CLOSE);

        panel.setLayout(new GridLayout(3,1, 5,5));

        // Crear dos textfield y un button

        textField1 = new JTextField( "Teclea algo y presiona return", 20); // 20 columnas de ancho

        // Añadir un listener

        textField1.addActionListener(new TextFieldListener());

        // Asignar una línea negra de borde.

        textField1.setBorder(BorderFactory.createLineBorder(Color.black));

        textField2 = new JTextField(20);
```

```
textField2.setBorder(BorderFactory.createLineBorder(Color.blue));

JButton button = new JButton("Limpiar");

button.addActionListener(new ButtonListener());

// Añadir los elementos al panel

panel.add(textField1);

panel.add(textField2);

panel.add(button);

add(panel);

}

// escuchador del textfield

class TextFieldListener implements ActionListener {

    public void actionPerformed(ActionEvent e) {

        // coge el texto tecleado en textField1

        // y lo asigna al otro textField.

        String text = textField1.getText();

        textField2.setText(text);

    }

}

// escuchador del botón.

class ButtonListener implements ActionListener {

    public void actionPerformed(ActionEvent e) {

        // limpia los textfields.

        textField1.setText("");

        textField1.requestFocus(); // Da el foco al textField1

        textField2.setText("");

    }

}

}
```

JLabel

No es más que una etiqueta de texto que podemos colocar al lado de cualquier componente para darle una indicación al usuario de cuál es la función de dicho componente. También se puede emplear a modo de título.

Veamos un ejemplo:

```
import javax.swing.*;

import java.awt.event.*;

import java.awt.*;

class frame extends JFrame {

    public frame(){

        setTitle("Ejemplo de JLabel");

        setSize(300,200);

        setDefaultCloseOperation(EXIT_ON_CLOSE);

        JPanel panel = new JPanel();

        //cambio el layout por defecto del panel

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

        //creo tres etiquetas

        JLabel label1 = new JLabel("Primera etiqueta"); //justificación izquierda por defecto

        JLabel label2 = new JLabel("Segunda etiqueta",JLabel.CENTER); //centrado

        JLabel label3 = new JLabel("Tercera etiqueta",JLabel.RIGHT); //derecha

        //las añado al panel

        panel.add(label1);

        panel.add(label2);

        panel.add(label3);

        add(panel);

    }

}

//de frame
```