

## TEORÍA ADICIONAL.- INTERFACE DE USUARIO.

### Objetivos

- Desarrollar Formularios a través de código.
- Insertar controles para mostrar y recoger información.
- Permitir el trabajo con varios formularios.

### Contenidos

#### 1.- Interfaces de Usuario

Java dispone de varias librerías de clase para implementar interfaces visuales. Vamos a utilizar los componentes del paquete Swing para el desarrollo de interfaces gráficas. Para utilizar los componentes del paquete Swing hay que importar del paquete javax.swing.

Para evitar advertencias en el código del entorno Netbeans en el desarrollo de interfaces debemos desactivar las opciones "Problematic call in the constructor" y "Parsing suspicious parameter in the constructor" del menú Herramientas / Opciones / Editor / Sugerencias / Initialization .

#### JFrame.

La clase JFrame encapsula el concepto de una ventana. Para implementar una aplicación que muestre una ventana debemos crear una clase que herede de la clase JFrame.

```
public class Formulario extends JFrame{
```

Estamos creando una clase Formulario que hereda todos los métodos y propiedades de la clase JFrame.

La clase JFrame dispone de varios métodos para dar formato a la ventana:

Método	Descripción
<b>setTitle(texto)</b>	Título de la Ventana
<b>setSize(ancho, alto)</b>	Tamaño de la Ventana
<b>setLocation(x, y)</b>	Posición de la Ventana
<b>setBounds(x, y, ancho, alto)</b>	Posición y Tamaño de la Ventana
<b>setResizable(boolean)</b>	Permitir cambio de Tamaño
<b>setLocationRelativeTo(null)</b>	Posición centro de la pantalla
<b>setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)</b>	Acción al pulsar botón Cerrar
<b>setVisible(boolean)</b>	Permitir Visibilidad

Ejemplo:

```
package formulario.jframe;
import javax.swing.*;
public class FormularioJFrame {
    public static void main(String[] args) {
        Formulario miFormulario = new Formulario();
        miFormulario.setVisible(true);
    }
}
```

```

}
class Formulario extends JFrame{
    public Formulario() {
        this.setTitle("Mi Aplicación Java");
        this.setSize(500, 300);
        this.setResizable(false);
        this.setLocationRelativeTo(null);
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}

```

### JPanel

La clase JPanel permite crear objetos contenedores, la finalidad de estos objetos es la agrupación de otros objetos como botones, campos de texto, etiquetas, selectores, etc... y permite manejar la agrupación de una mejor forma.

Podemos crear un JPanel creando una instancia de la clase con

```
JPanel miPanel = new JPanel ();
```

Una vez creado el JPanel podemos establecer el método setLayout a null para poder posicionar los componentes en las posiciones x, y que indiquemos con setLocation o setBounds.

```
miPanel.setLayout (null);
```

Una vez colocados los componentes en el panel añadimos el panel al formulario con add().

```
miFormulario.add(miPanel);
```

### JLabel

La clase JLabel permite añadir texto. Podemos crear un JLabel creando una instancia de la clase y pasando como parámetro el texto que queremos mostrar.

```
JLabel miTexto = new JLabel ("Este es el texto que se mostrará");
```

Para ubicar el texto en el JPanel utilizaremos el método setLocation o setBounds, para posteriormente añadir el texto al panel.

```
miTexto.setBounds(10, 40, 100, 30);  
miPanel.add(miTexto);
```

Podemos cambiar el tipo y tamaño de fuente creando una fuente con la clase Font y asignarlo posteriormente al texto con el método setFont(). El método setFont() recibe como parámetros el nombre de la fuente, el estilo y el tamaño. Los posibles estilos vienen definidos por las

constantes `Font.PLAIN`, `Font.BOLD`, `Font.ITALIC`, ... Deberemos importar la clase `Font` del paquete `java.awt`

```
Font miFuente = new Font("Comic Sans Ms", Font.PLAIN, 24);  
miTexto.setFont(miFuente);
```

También podemos asignar color al texto con el método `setForeground()` que recibe como parámetro el color definidos por las constantes `Color.RED`, `Color.BLUE`, `Color.GREEN`, ... o creando una instancia de `Color` y asignar la cantidad de cada una de las componentes RGB. Deberemos importar la clase `Color` del paquete `java.awt`

```
Color miColor = new Color(100, 45, 200);  
miTexto.setForeground(miColor);
```

Si queremos asignar un texto al `JLabel` podemos utilizar el método `setText()`;

```
miTexto.setText("Este es el nuevo texto");
```

Ejemplo:

```
package formulario.jframe;  
import java.awt.Color;  
import java.awt.Font;  
import javax.swing.*;  
public class FormularioJFrame {  
    public static void main(String[] args) {  
        Formulario miFormulario = new Formulario();  
        miFormulario.setVisible(true);  
    }  
}  
  
class Formulario extends JFrame{  
    JPanel miCapa;  
    JLabel textoTitulo;  
    JLabel textoSubtitulo;  
  
    public Formulario() {  
        this.setTitle("Mi Aplicación Java");  
        this.setSize(500, 300);  
        this.setResizable(false);  
        this.setLocationRelativeTo(null);  
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);  
  
        // Creamos un panel para colocar componentes en el Formulario  
        miCapa = new JPanel();  
        // Establecemos el setLayout a null para poder posicionar los  
        // componentes en las posiciones x, y indicadas en setBounds  
        miCapa.setLayout(null);  
  
        // Creamos un estilo de fuente con la clase Font y se lo asignaremos
```

```

        // con setFont a los componentes JLabel que se creen en el Panel
        // Debemos importar la clase Font del paquete java.awt
        Font miFuente = new Font("Comic Sans Ms", Font.PLAIN, 24);
        // Creamos un color con la clase Color y asignando la cantidad de cada
        // una de las componentes RGB. Posteriormente podemos asignarlo al
        // texto con el método setForeground()
        // Debemos importar la clase Color del paquete java.awt
        Color miColor = new Color(100, 45, 200);
        // Creamos un texto, la posicionamos y la añadimos a la capa
        textoTitulo = new JLabel("Curso de Programación");
        textoTitulo.setBounds(10, 20, 300, 40);
        textoTitulo.setFont(miFuente);
        textoTitulo.setForeground(miColor);
        miCapa.add(textoTitulo);
        // Creamos un texto, la posicionamos y la añadimos a la capa
        textoSubtitulo = new JLabel("Aula Campus");
        textoSubtitulo.setBounds(10, 50, 100, 30);
        miCapa.add(textoSubtitulo);
        // Añadimos la capa al Formulario
        this.add(miCapa);
    }
}

```

## JTextField

La clase JTextField permite introducir texto por teclado. Podemos crear un JTextField creando una instancia de la clase.

```
JTextField miCampoTexto = new JTextField();
```

Para ubicar el campo de texto en el JTextField utilizaremos el método setLocation o setBounds, para posteriormente añadir el texto al panel.

```
miCampoTexto.setBounds(10, 40, 100, 30);
miPanel.add(miCampoTexto);
```

Podemos acceder al contenido del JTextField con el método getText().

```
miCampoTexto.getText();
```

Ejemplo:

```

package formulario.jframe;
import javax.swing.*;
public class FormularioJFrame {
    public static void main(String[] args) {
        Formulario miFormulario = new Formulario();
        miFormulario.setVisible(true);
    }
}
class Formulario extends JFrame{
    JPanel miCapa;
    JLabel miTexto;
    JTextField miCampoTexto ;
}

```

```

public Formulario() {
    this.setTitle("Mi Aplicación Java");
    this.setSize(500, 300);
    this.setResizable(false);
    this.setLocationRelativeTo(null);
    this.setDefaultCloseOperation(EXIT_ON_CLOSE);

    // Creamos un panel para colocar componentes en el Formulario
    miCapa = new JPanel();
    // Establecemos el setLayout a null para poder posicionar los
    // componentes en las posiciones x, y indicadas en setBounds
    miCapa.setLayout(null);

    // Creamos un texto, lo posicionamos y lo añadimos a la capa
    miTexto = new JLabel("Nombre:");
    miTexto.setBounds(10, 20, 50, 30);
    miCapa.add(miTexto);
    // Creamos un Campo de Texto, lo posicionamos y añadimos a la capa
    miCampoTexto = new JTextField();
    miCampoTexto.setBounds(70, 20, 100, 30);
    miCapa.add(miCampoTexto);

    // Añadimos la capa al Formulario
    this.add(miCapa);
}
}

```

### JButton.

La clase JButton muestra un botón. Podemos crear una instancia del botón, posicionarla con el método setBounds() y añadirla al panel con add().

```

JButton miBoton = new JButton("Finalizar");
miBoton.setBounds(10, 100, 100, 30);
miPanel.add(miBoton);

```

El uso de botones precisa la captura de eventos como cuando hacemos clic sobre un botón. Java implementa el concepto de interfaces para poder llamar a métodos de una clase existente a una clase desarrollada por nosotros.

La captura del clic sobre el objeto de la clase JButton se hace mediante la implementación de la interface ActionListener tiene la siguiente estructura:

```

interface ActionListener {
    public void actionPerformed(ActionEvent e) {
    }
}

```

Luego las clases que implementen la interface ActionListener deberán especificar el código del método actionPerformed.

Para indicar que una clase implementará una interface declaramos la clase con la sintaxis:

```

public class Formulario extends JFrame implements ActionListener {}

```

Estamos indicando que nuestra clase implementa la interface ActionListener, por lo que estamos obligados a desarrollar el código del método actionPerformed.

Podemos añadir al botón el método addActionListener() que queda esperando a que pulsemos sobre el botón. Cuando pulsemos sobre el botón, tomará el parámetro this que hace referencia al propio botón y ejecutará el método actionPerformed() que recibe como parámetro un objeto de la clase ActionEvent. La interface ActionListener y el objeto de la clase ActionEvent que llega como parámetro están definidos en el paquete java.awt.event.

```
mi Boton. addActionListener(this);
```

El método actionPerformed() mediante el uso del método getSource() del objeto que llega como parámetro analizar que botón se presionó:

```
public void actionPerformed(ActionEvent e) {  
    if (e.getSource()==mi Boton) {  
        System.exit(0);  
    }  
}
```

Si se pulsó miBoton se llama al método exit de la clase System se finaliza la ejecución del programa.

Ejemplo:

```
package formulario.jframe;  
import java.awt.event.*;  
import javax.swing.*;  
public class FormularioJFrame {  
    public static void main(String[] args) {  
        Formulario miFormulario = new Formulario();  
        miFormulario.setVisible(true);  
    }  
}  
class Formulario extends JFrame implements ActionListener{  
    JPanel miCapa;  
    JButton miBoton;  
    JButton salirBoton;  
    JLabel miTexto;  
    JLabel miMensaje;  
    JTextField miCampoTexto;  
  
    public Formulario() {  
        this.setTitle("Mi Aplicación Java");  
        this.setSize(500, 300);  
        this.setResizable(false);  
        this.setLocationRelativeTo(null);  
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);  
  
        // Creamos un panel para colocar componentes en el Formulario
```

```

miCapa = new JPanel ();
// Establecemos el setLayout a null para poder posicionar los
// componentes en las posiciones x, y indicadas en setBounds
miCapa.setLayout(null);

// Creamos un Texto, lo posicionamos y añadimos a la capa
miTexto = new JLabel ("Nombre: ");
miTexto.setBounds(10, 20, 100, 30);
miCapa.add(miTexto);

miMensaje = new JLabel ();
miMensaje.setBounds(10, 50, 100, 30);
miCapa.add(miMensaje);

// Creamos un Campo de Texto, lo posicionamos y añadimos a la capa
miCampoTexto = new JTextField();
miCampoTexto.setBounds(80, 20, 100, 30);
miCapa.add(miCampoTexto);

// Creamos un botón, lo posicionamos y añadimos a la capa
miBoton = new JButton("Mensaje");
miBoton.setBounds(10, 100, 100, 30);
miCapa.add(miBoton);

// Creamos un botón, lo posicionamos y añadimos a la capa
salirBoton = new JButton("Salir");
salirBoton.setBounds(150, 100, 100, 30);
miCapa.add(salirBoton);

miBoton.addActionListener(this);
salirBoton.addActionListener(this);

// Añadimos la capa al Formulario
this.add(miCapa);
}
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource()==miBoton) {
        String mensaje;
        mensaje = miCampoTexto.getText();
        miMensaje.setText("Hola "+mensaje);
    }
    if (e.getSource()==salirBoton) {
        System.exit(0);
    }
}
}

```

**Ejemplo:** Aplicación con interface visual que permita introducir dos números en dos cajas de texto. Debe tener un botón llamado Sumar que suma los dos números introducidos en las

cajas de texto y muestra el resultado en otra caja de texto no editable. Por último un botón de Salir permite cerrar la aplicación.

```
package formulario.jframe;
import java.awt.event.*;
import javax.swing.*;

public class FormularioJFrame {
    public static void main(String[] args) {
        Formulario miFormulario = new Formulario();
        miFormulario.setVisible(true);
    }
}

class Formulario extends JFrame implements ActionListener{
    JPanel miCapa;
    JButton sumarB;
    JButton salirB;
    JLabel num1L;
    JLabel num2L;
    JLabel resultadoL;
    JTextField num1TF;
    JTextField num2TF;
    JTextField resultadoTF;

    public Formulario() {
        this.setTitle("Calculadora");
        this.setSize(500, 300);
        this.setResizable(false);
        this.setLocationRelativeTo(null);
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);

        // Creamos un panel para colocar componentes en el Formulario
        miCapa = new JPanel();
        // Establecemos el setLayout a null para poder posicionar los
        // componentes en las posiciones x, y indicadas en setBounds
        miCapa.setLayout(null);

        // Creamos un Texto, lo posicionamos y añadimos a la capa
        num1L = new JLabel("Numero 1: ");
        num1L.setBounds(10, 20, 100, 30);
        miCapa.add(num1L);

        num2L = new JLabel("Numero 2: ");
        num2L.setBounds(10, 60, 100, 30);
        miCapa.add(num2L);

        resultadoL = new JLabel("Resultado: ");
        resultadoL.setBounds(10, 100, 100, 30);
        miCapa.add(resultadoL);

        // Creamos un Campo de Texto, lo posicionamos y añadimos a la capa
```



```

        num1TF = new JTextField();
        num1TF.setBounds(80, 20, 100, 30);
        miCapa.add(num1TF);

        // Creamos un Campo de Texto, lo posicionamos y añadimos a la capa
        num2TF = new JTextField();
        num2TF.setBounds(80, 60, 100, 30);
        miCapa.add(num2TF);

        resultadoTF = new JTextField();
        resultadoTF.setBounds(80, 100, 100, 30);
        resultadoTF.setEditable(false);
        miCapa.add(resultadoTF);

        // Creamos un botón, lo posicionamos y añadimos a la capa
        sumarB = new JButton("Sumar");
        sumarB.setBounds(140, 180, 100, 30);
        miCapa.add(sumarB);

        // Creamos un botón, lo posicionamos y añadimos a la capa
        salirB = new JButton("Salir");
        salirB.setBounds(250, 180, 100, 30);
        miCapa.add(salirB);

        sumarB.addActionListener(this);
        salirB.addActionListener(this);

        // Añadimos la capa al Formulario
        this.add(miCapa);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource()==sumarB) {
            Double num1;
            Double num2;
            Double calculo;

            num1=Double.parseDouble(num1TF.getText());
            num2=Double.parseDouble(num2TF.getText());
            calculo = num1+num2;
            resultadoTF.setText(calculo.toString());
        }
        if (e.getSource()==salirB) {
            System.exit(0);
        }
    }
}

```

### JComboBox.

```

package despl egable;
import java.awt.event.*;
import javax.swing.*;
public class Despl egable {
    public static void main(String[] args) {
        Ventana mi Ventana = new Ventana();
        mi Ventana.setVisible(true);
    }
}

class Ventana extends JFrame implements ActionListener{
    JPanel mi Panel ;
    JComboBox<String> mi Combo;
    JButton comprobar;
    JButton salir;
    JLabel mensaje;

    public Ventana(){
        this.setResizable(false);
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setTitle("Cines");
        this.setSize(500, 400);
        this.setLocationRelativeTo(null);

        mi Panel = new JPanel ();
        mi Panel .setLayout(null);

        mi Combo = new JComboBox<>();
        mi Combo.addItem("Casper");
        mi Combo.addItem("Piratas del Caribe");
        mi Combo.addItem("Batman");
        mi Combo.addItem("La forma del agua");

        mi Combo.setBounds(30, 30, 150, 30);
        mi Panel .add(mi Combo);

        mensaje = new JLabel ();
        mensaje.setBounds(30, 200, 600, 30);
        mi Panel .add(mensaje);

        comprobar = new JButton("Comprobar");
        comprobar.setBounds(230, 30, 100, 30);
        mi Panel .add(comprobar);

        salir = new JButton("Salir");
        salir.setBounds(350, 30, 100, 30);
        mi Panel .add(salir);

        this.add(mi Panel );
        comprobar.addActionListener(this);
        salir.addActionListener(this);
    }
}

```

```

@Override
public void actionPerformed(ActionEvent e) {
    if(e.getSource() == comprobar){
        String seleccion;
        seleccion = (String)miCombo.getSelectedItem();
        int indice;
        indice = miCombo.getSelectedIndex();
        mensaje.setText("Se ha seleccionado la opción "+seleccion+" que
está en la posición "+indice);
    }
    if(e.getSource() == salir){
        System.exit(0);
    }
}
}

```

### JTable.

```

package tabla;
import java.awt.event.*;
import javax.swing.*;
public class Tabla {
    public static void main(String[] args) {
        Formulario miFormulario = new Formulario();
        miFormulario.setVisible(true);
    }
}
class Formulario extends JFrame implements ActionListener{
    JPanel miCapa;
    JPanel tuCapa;
    JTable miTabla;
    JScrollPane scrollTable;
    JButton miBoton;

    public Formulario(){
        // Asignamos las características del JFrame
        this.setTitle("Tabla");
        this.setSize(700, 600);
        this.setResizable(false);
        this.setLocationRelativeTo(null);
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);

        // Array cabeceras de la tabla
        String[] campos = new String[] {"Id", "Nombre", "Apellidos", "Edad"};
        // Array de datos de la Tabla
        Object[][] datos = {
            {1, "Luis", "Ros Mas", 35},
            {2, "Rosa", "Sancho Molina", 25}
        };

        // Creamos la Capa
        miCapa = new JPanel();
    }
}

```

```

// Asignamos sistema de posicionamiento por coordenadas
miCapa.setLayout(null);

// Creamos la tabla
miTabla = new JTable(datos, campos);

//Creamos Panel con Scroll y se le asignamos la tabla
scrollTable = new JScrollPane(miTabla);
// Asignamos tamaño al Panel con Scroll que contiene la tabla
scrollTable.setBounds(10, 10, 660, 390);
// Añadimos el Panel con Scroll que contiene la tabla a la Capa
miCapa.add(scrollTable);

// Creamos el botón y le damos tamaño
miBoton = new JButton("Salir");
miBoton.setBounds(300, 430, 100, 30);
// Añadimos el botón a la capa
miCapa.add(miBoton);
// Ponemos el botón a la escucha
miBoton.addActionListener(this);
// Añadimos la capa al Formulario
this.add(miCapa);
}
@Override
public void actionPerformed(ActionEvent e){
    // Si pulsamos el botón Salir cerramos la aplicación.
    if(e.getSource()==miBoton){
        System.exit(0);
    }
}
}

```

### Trabajo con varios formularios.

```

package interfaces;
import java.awt.event.*;
import javax.swing.*;
public class interfaces {
    public static void main(String[] args) {
        Formulario1 miFormulario = new Formulario1();
        miFormulario.setVisible(true);
    }
}
class Formulario1 extends JFrame implements ActionListener{
    JPanel miCapa;
    JLabel textoTitulo;
    JLabel textoSubtitulo;
    JLabel miTexto;
    JTextFiled miCampoTexto;

```

```

JButton mi Boton;
JButton botonSalir;

public Formulario1(){
    this.setTitle("Java 1");
    this.setSize(170, 300);
    this.setResizable(false);
    this.setLocationRelativeTo(null);
    this.setDefaultCloseOperation(EXIT_ON_CLOSE);

    miCapa = new JPanel();
    miCapa.setLayout(null);

    miBoton = new JButton("Formulario 2");
    miBoton.setBounds(10, 140, 150, 30);
    miCapa.add(miBoton);

    miBoton.addActionListener(this);

    this.add(miCapa);
}
@Override
public void actionPerformed(ActionEvent e){
    if(e.getSource()==miBoton){
        Formulario2 miFormulario = new Formulario2();
        miFormulario.setVisible(true);
        this.dispose();
    }

    if(e.getSource()==botonSalir){
        System.exit(0);
    }
}
}

class Formulario2 extends JFrame implements ActionListener{
    JPanel miCapa;
    JLabel textoTitulo;
    JLabel textoSubtitulo;
    JLabel miTexto;
    JTextField miCampoTexto;
    JButton miBoton;
    JButton botonSalir;

    public Formulario2(){
        this.setTitle("Java 2");

```

```

        this.setSize(170, 300);
        this.setResizable(false);
        this.setLocationRelativeTo(null);
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);

        miCapa = new JPanel();
        miCapa.setLayout(null);

        miBoton = new JButton("Formulario 1");
        miBoton.setBounds(10, 140, 150, 30);
        miCapa.add(miBoton);

        botonSalir = new JButton("Salir");
        botonSalir.setBounds(10, 190, 150, 30);
        miCapa.add(botonSalir);

        miBoton.addActionListener(this);
        botonSalir.addActionListener(this);

        this.add(miCapa);
    }
    @Override
    public void actionPerformed(ActionEvent e){
        if(e.getSource()==miBoton){
            Formulario1 miFormulario = new Formulario1();
            miFormulario.setVisible(true);
            this.dispose();
        }
        if(e.getSource()==botonSalir){
            System.exit(0);
        }
    }
}

```