



Tema 4

Tutorial básico de swing

Proyecto de Análisis y Diseño de Software
2º Ingeniería Informática
Universidad Autónoma de Madrid

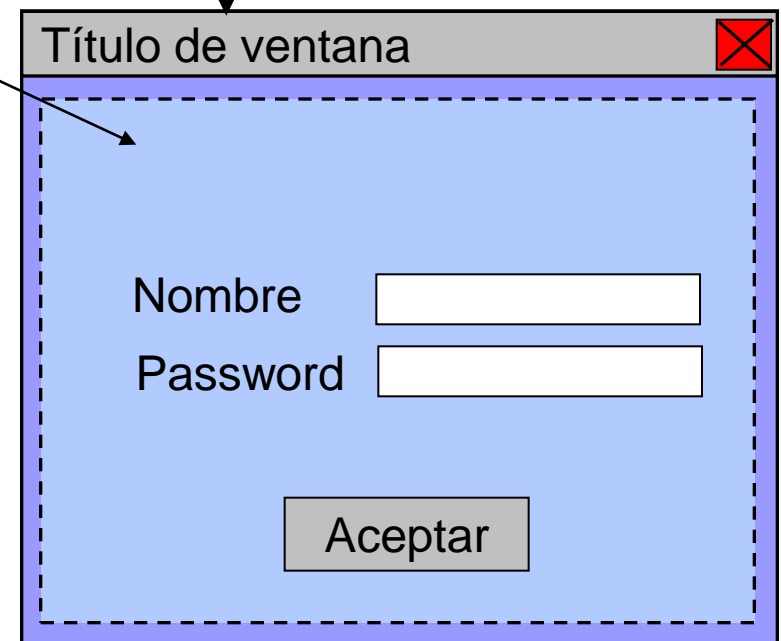
Elementos principales

■ Contenedores:

- De primer nivel (ventanas): JFrame, JDialog...
- Intermedios: JPanel...

■ Componentes:

- Etiquetas, campos de texto, botones...
- Se colocan dentro de los contenedores



Pasos para crear una GUI

1. Crear ventana (JFrame)
2. Obtener el contenedor de la ventana, y asignarle *layout* (colocación de los componentes)
3. Crear componentes
 1. Definir acciones asociadas a los componentes, por ejemplo al pulsar un botón
4. Añadir componentes al contenedor
5. Mostrar ventana

```
// crear ventana
JFrame ventana = new JFrame("Mi GUI");

// obtener contenedor, asignar layout
Container contenedor = ventana.getContentPane();
contenedor.setLayout(new FlowLayout());

// crear componentes
JLabel etiqueta = new JLabel("Nombre");
final JTextField campo = new JTextField(10);
JButton boton = new JButton("Haz click");

// asociar acciones a componentes
boton.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(null, campo.getText());
        }
    }
);

// añadir componentes al contenedor
contenedor.add(etiqueta);
contenedor.add(campo);
contenedor.add(boton);

// mostrar ventana
ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
ventana.setSize(100,140);
ventana.setVisible(true);
```

Pasos para crear una GUI

1. Crear ventana (JFrame)
2. Obtener el contenedor de la ventana, y asignarle *layout* (colocación de los componentes)
3. Crear componentes
 1. Definir acciones asociadas a los componentes, por ejemplo al pulsar un botón
4. Añadir componentes al contenedor
5. Mostrar ventana

```
// crear ventana
JFrame ventana = new JFrame("Mi GUI");

// obtener contenedor, asignar layout
Container contenedor = ventana.getContentPane();
contenedor.setLayout(new FlowLayout());

// crear componentes
JLabel etiqueta = new JLabel("Nombre");
final JTextField campo = new JTextField(10);
JButton boton = new JButton("Haz click");

// asociar acciones a componentes
boton.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(null, campo.getText());
        }
    }
);

// añadir componentes al contenedor
contenedor.add(etiqueta);
contenedor.add(campo);
contenedor.add(boton);

// mostrar ventana
ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
ventana.setSize(100,140);
ventana.setVisible(true);
```

Pasos para crear una GUI

1. Crear ventana (JFrame)
2. Obtener el contenedor de la ventana, y asignarle *layout* (colocación de los componentes)
3. **Crear componentes**
 1. Definir acciones asociadas a los componentes, por ejemplo al pulsar un botón
4. Añadir componentes al contenedor
5. Mostrar ventana

```
// crear ventana
JFrame ventana = new JFrame("Mi GUI");

// obtener contenedor, asignar layout
Container contenedor = ventana.getContentPane();
contenedor.setLayout(new FlowLayout());

// crear componentes
JLabel etiqueta = new JLabel("Nombre");
final JTextField campo = new JTextField(10);
JButton boton = new JButton("Haz click");

// asociar acciones a componentes
boton.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(null, campo.getText());
        }
    }
);

// añadir componentes al contenedor
contenedor.add(etiqueta);
contenedor.add(campo);
contenedor.add(boton);

// mostrar ventana
ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
ventana.setSize(100,140);
ventana.setVisible(true);
```

Pasos para crear una GUI

1. Crear ventana (JFrame)
2. Obtener el contenedor de la ventana, y asignarle *layout* (colocación de los componentes)
3. Crear componentes
 1. Definir acciones asociadas a los componentes, por ejemplo al pulsar un botón
4. Añadir componentes al contenedor
5. Mostrar ventana

```
// crear ventana
JFrame ventana = new JFrame("Mi GUI");

// obtener contenedor, asignar layout
Container contenedor = ventana.getContentPane();
contenedor.setLayout(new FlowLayout());

// crear componentes
JLabel etiqueta = new JLabel("Nombre");
final JTextField campo = new JTextField(20);
JButton boton = new JButton("Haz clic");

// asociar acciones a componentes
boton.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(null, campo.getText());
        }
    }
);

// añadir componentes al contenedor
contenedor.add(etiqueta);
contenedor.add(campo);
contenedor.add(boton);

// mostrar ventana
ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
ventana.setSize(100,140);
ventana.setVisible(true);
```

El código dentro del método *actionPerformed* se ejecutará al pulsar el botón. En este caso, se mostrará un mensaje por pantalla.

Pasos para crear una GUI

1. Crear ventana (JFrame)
2. Obtener el contenedor de la ventana, y asignarle *layout* (colocación de los componentes)
3. Crear componentes
 1. Definir acciones asociadas a los componentes, por ejemplo al pulsar un botón
4. **Añadir componentes al contenedor**
5. Mostrar ventana

```
// crear ventana
JFrame ventana = new JFrame("Mi GUI");

// obtener contenedor, asignar layout
Container contenedor = ventana.getContentPane();
contenedor.setLayout(new FlowLayout());

// crear componentes
JLabel etiqueta = new JLabel("Nombre");
final JTextField campo = new JTextField(10);
JButton boton = new JButton("Haz click");

// asociar acciones a componentes
boton.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(null, campo.getText());
        }
    }
);

// añadir componentes al contenedor
contenedor.add(etiqueta);
contenedor.add(campo);
contenedor.add(boton);

// mostrar ventana
ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
ventana.setSize(100,140);
ventana.setVisible(true);
```

Pasos para crear una GUI

1. Crear ventana (JFrame)
2. Obtener el contenedor de la ventana, y asignarle *layout* (colocación de los componentes)
3. Crear componentes
 1. Definir acciones asociadas a los componentes, por ejemplo al pulsar un botón
4. Añadir componentes al contenedor
5. **Mostrar ventana**

```
// crear ventana
JFrame ventana = new JFrame("Mi GUI");

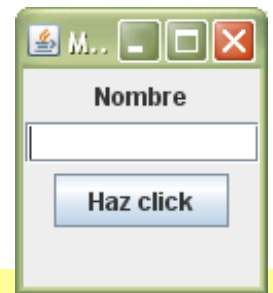
// obtener contenedor, asignar layout
Container contenedor = ventana.getContentPane();
contenedor.setLayout(new FlowLayout());

// crear componentes
JLabel etiqueta = new JLabel("Nombre");
final JTextField campo = new JTextField(10);
JButton boton = new JButton("Haz click");

// asociar acciones a componentes
boton.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(null, campo.getText());
        }
    }
);

// añadir componentes al contenedor
contenedor.add(etiqueta);
contenedor.add(campo);
contenedor.add(boton);

// mostrar ventana
ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
ventana.setSize(100,140);
ventana.setVisible(true);
```



GUI con varias pantallas

1. Crear ventana (JFrame)
2. Obtener el contenedor de la ventana, y asignarle *layout* (colocación de los componentes)
3. **Crear un JPanel por pantalla**
 1. La primera pantalla contiene un botón, y al pulsarlo “vamos” a la segunda pantalla
4. Añadir pantallas al contenedor
5. Mostrar ventana

```
// crear ventana (como antes)
// obtener contenedor, asignar layout
...
```

La primera pantalla, que contiene un botón, está inicialmente visible.

```
// crear pantalla 1
final JPanel pantalla1 = new JPanel();
JButton boton = new JButton("siguiente");
pantalla1.add(boton);
pantalla1.setVisible(true);
```

```
// crear pantalla 2
final JPanel pantalla2 = new JPanel();
JLabel etiqueta = new JLabel("segunda pantalla");
pantalla2.add(etiqueta);
pantalla2.setVisible(false);
```

```
// pulsar el botón oculta la pantalla 1 y muestra la 2
boton.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            pantalla1.setVisible(false);
            pantalla2.setVisible(true);
        }
    }
);
```

```
// añadir pantallas al contenedor
contenedor.add(pantalla1);
contenedor.add(pantalla2);
```

```
// mostrar ventana (como antes)
...
```

GUI con varias pantallas

1. Crear ventana (JFrame)
2. Obtener el contenedor de la ventana, y asignarle *layout* (colocación de los componentes)
3. **Crear un JPanel por pantalla**
 1. La primera pantalla contiene un botón, y al pulsarlo “vamos” a la segunda pantalla
4. Añadir pantallas al contenedor
5. Mostrar ventana

```
// crear ventana (como antes)
// obtener contenedor, asignar layout (como antes)
...
```

```
// crear pantalla 1
final JPanel pantalla1 = new JPanel();
JButton boton = new JButton("siguiente");
pantalla1.add(boton);
pantalla1.setVisible(true);
```

La segunda pantalla está inicialmente oculta.

```
// crear pantalla 2
final JPanel pantalla2 = new JPanel();
JLabel etiqueta = new JLabel("segunda pantalla");
pantalla2.add(etiqueta);
pantalla2.setVisible(false);
```

```
// pulsar el botón oculta la pantalla 1 y muestra la 2
boton.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            pantalla1.setVisible(false);
            pantalla2.setVisible(true);
        }
    }
);
```

```
// añadir pantallas al contenedor
contenedor.add(pantalla1);
contenedor.add(pantalla2);
```

```
// mostrar ventana (como antes)
...
```

GUI con varias pantallas

1. Crear ventana (JFrame)
2. Obtener el contenedor de la ventana, y asignarle *layout* (colocación de los componentes)
3. Crear un JPanel por pantalla
 1. **La primera pantalla contiene un botón, y al pulsarlo “vamos” a la segunda pantalla**
4. Añadir pantallas al contenedor
5. Mostrar ventana

```
// crear ventana (como antes)
// obtener contenedor, asignar layout (como antes)
...

// crear pantalla 1
final JPanel pantalla1 = new JPanel();
JButton boton = new JButton("siguiente");
pantalla1.add(boton);
pantalla1.setVisible(true);

// crear pantalla 2
final JPanel pantalla2 = new JPanel();
JLabel etiqueta = new JLabel("segunda pantalla");
pantalla2.add(etiqueta);
pantalla2.setVisible(false);

// acción asociada al botón
boton.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            pantalla1.setVisible(false);
            pantalla2.setVisible(true);
        }
    }
);

// añadir pantallas al contenedor
contenedor.add(pantalla1);
contenedor.add(pantalla2);

// mostrar ventana (como antes)
...
```

Al pulsar el botón se oculta la pantalla 1 y se muestra la 2.

GUI con varias pantallas

1. Crear ventana (JFrame)
2. Obtener el contenedor de la ventana, y asignarle *layout* (colocación de los componentes)
3. Crear un JPanel por pantalla
 1. La primera pantalla contiene un botón, y al pulsarlo “vamos” a la segunda pantalla
4. **Añadir pantallas al contenedor**
5. Mostrar ventana

```
// crear ventana (como antes)
// obtener contenedor, asignar layout (como antes)
...

// crear pantalla 1
final JPanel pantalla1 = new JPanel();
JButton boton = new JButton("siguiente");
pantalla1.add(boton);
pantalla1.setVisible(true);

// crear pantalla 2
final JPanel pantalla2 = new JPanel();
JLabel etiqueta = new JLabel("segunda pantalla");
pantalla2.add(etiqueta);
pantalla2.setVisible(false);

// acción asociada al botón
boton.addActionListener(
    new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            pantalla1.setVisible(false);
            pantalla2.setVisible(true);
        }
    }
);

// añadir pantallas al contenedor
contenedor.add(pantalla1);
contenedor.add(pantalla2);

// mostrar ventana (como antes)
...
```

Como puede verse, un contenedor puede contener otros contenedores (pantalla1 y pantalla2 son contenedores)

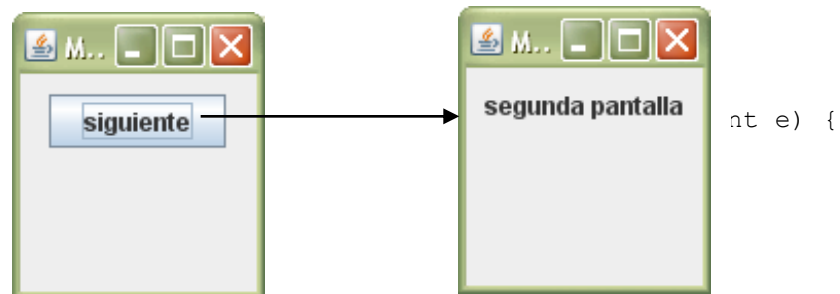
GUI con varias pantallas

1. Crear ventana (JFrame)
2. Obtener el contenedor de la ventana, y asignarle *layout* (colocación de los componentes)
3. Crear un JPanel por pantalla
 1. La primera pantalla contiene un botón, y al pulsarlo “vamos” a la segunda pantalla
4. Añadir pantallas al contenedor
5. Mostrar ventana

```
// crear ventana (como antes)
// obtener contenedor, asignar layout (como antes)
...

// crear pantalla 1
final JPanel pantalla1 = new JPanel();
JButton boton = new JButton("siguiente");
pantalla1.add(boton);
pantalla1.setVisible(true);

// crear pantalla 2
final JPanel pantalla2 = new JPanel();
JLabel etiqueta = new JLabel("segunda pantalla");
pantalla2.add(etiqueta);
pantalla2.setVisible(false);
```



```
// añadir pantallas al contenedor
contenedor.add(pantalla1);
contenedor.add(pantalla2);

// mostrar ventana (como antes)
...
```

Construir componentes GUI

Creamos subclases de los componentes swing

```
public class MiPanel extends JPanel {  
  
    MiPanel () {  
        // asignar layout  
        this.setLayout(new FlowLayout());  
  
        // crear componentes  
        JLabel etiqueta = new JLabel("Nombre");  
        final JTextField campo = new JTextField(10);  
        JButton boton = new JButton("Haz click");  
  
        // asociar acciones a componentes  
        boton.addActionListener(  
            new ActionListener() {  
                public void actionPerformed(ActionEvent e) {  
                    JOptionPane.showMessageDialog(null,  
                        campo.getText());  
                }  
            }  
        );  
  
        // añadir componentes  
        // al contenedor  
        this.add(etiqueta);  
        this.add(campo);  
        this.add(boton);  
    }  
}
```

Los componentes del panel se crean en el constructor. Puedo tener constructores con distintos parámetros.

Las subclases pueden usarse como un componente swing

```
// Éste es el ejemplo de la transparencia 3,  
// usando el nuevo panel que nos hemos creado  
  
// crear ventana  
JFrame ventana = new JFrame("Mi GUI");  
  
// obtener contenedor, asignar layout  
Container contenedor = ventana.getContentPane();  
contenedor.setLayout(new FlowLayout());  
  
// crear componentes  
JPanel panel = new MiPanel();  
  
// añadir componentes al contenedor  
contenedor.add(panel);  
  
// mostrar ventana  
ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
ventana.pack();  
ventana.setVisible(true);
```

Es recomendable crear componentes GUI, en vez de tener toda la interfaz de la aplicación en una sola clase.



Referencias

- Tutorial de Swing:

<http://docs.oracle.com/javase/tutorial/uiswing/>

- JavaDoc del API de Swing:

<http://download.oracle.com/javase/6/docs/api/javax/swing/package-summary.html>

- Colección de ejemplos Swing:

<http://download.oracle.com/javase/tutorial/uiswing/examples/components/index.html>