



## EXPERIENCIA DE APRENDIZAJE 3 INTERFAZ, ESTRUCTURA Y VALIDACIÓN

### Interfaz Gráfica

DSY1102-Desarrollo Orientado a Objetos

# CONTENIDO

**01**

GUI

**04**

EVENTOS

**02**

CONTENEDOR

**03**

COMPONENTES

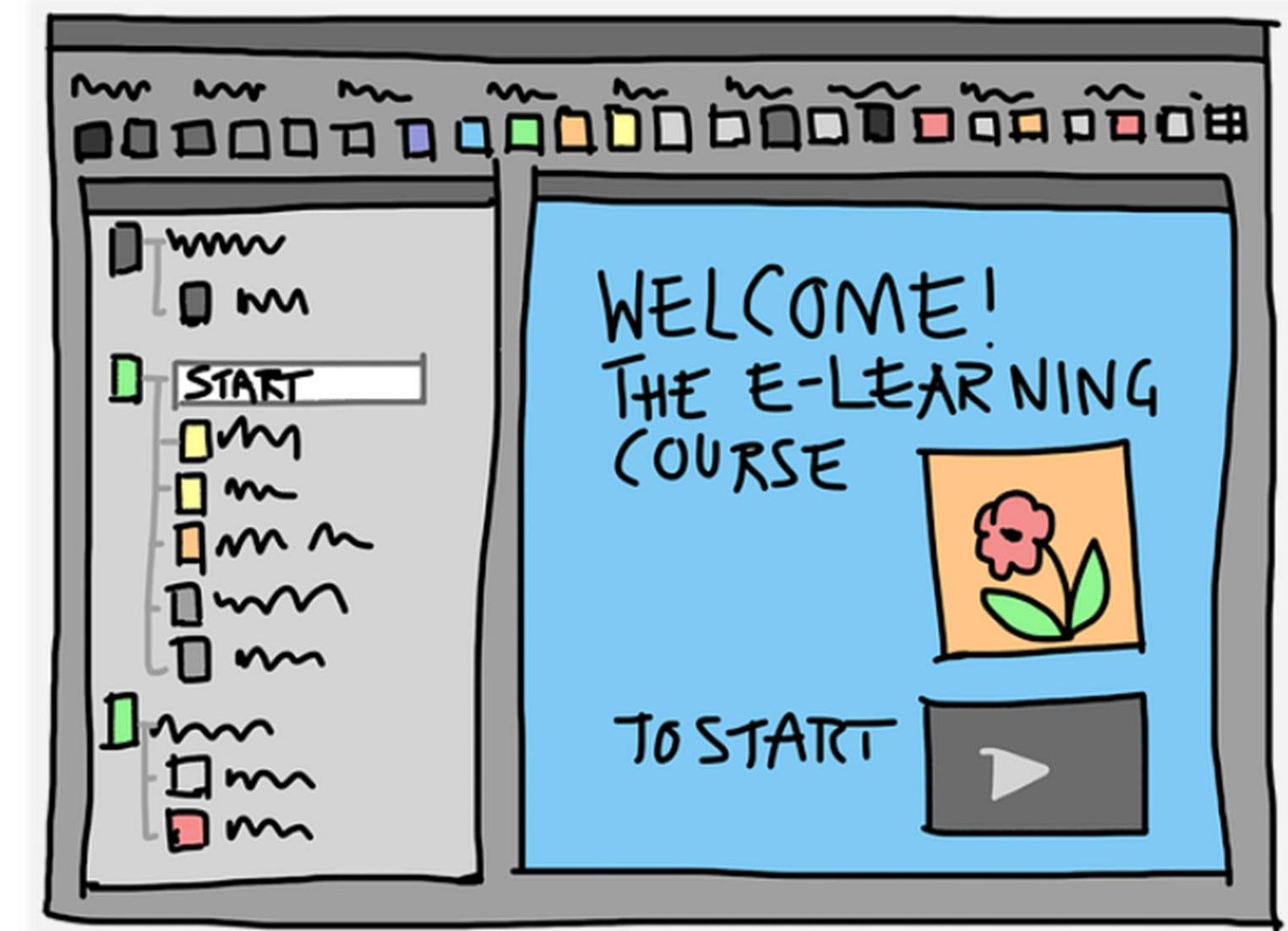
# 01

## GUI

# GUI

¿Qué es GUI?

*Interfaz Gráfica de Usuario*



Permite al usuario interactuar con una aplicación de manera amigable e intuitiva para aprender a utilizarla en el menor tiempo y así lograr una mayor productividad. Utilizan un sistema de ventanas y de elementos como: botones, cajas de textos, listas entre otros.

# GUI

**Swing**, es una evolución del AWT (Abstract Window Toolkit ), con un diseño interno orientado a componentes y un look mucho más satisfactorio.

Las GUIs de Java se creaban a partir de componentes del AWT (paquete `java.awt`), pero al ejecutar una aplicación, los componentes de la GUI se mostraban de manera distinta en cada plataforma.

Los componentes de GUI de Swing nos permiten especificar una apariencia visual uniforme para una aplicación a través de todas las plataformas. Además, son más portables y flexibles. Por esta razón, se prefieren los componentes de GUI de Swing.

**AWT**

Enviar



**Swing**

Enviar

# GUI

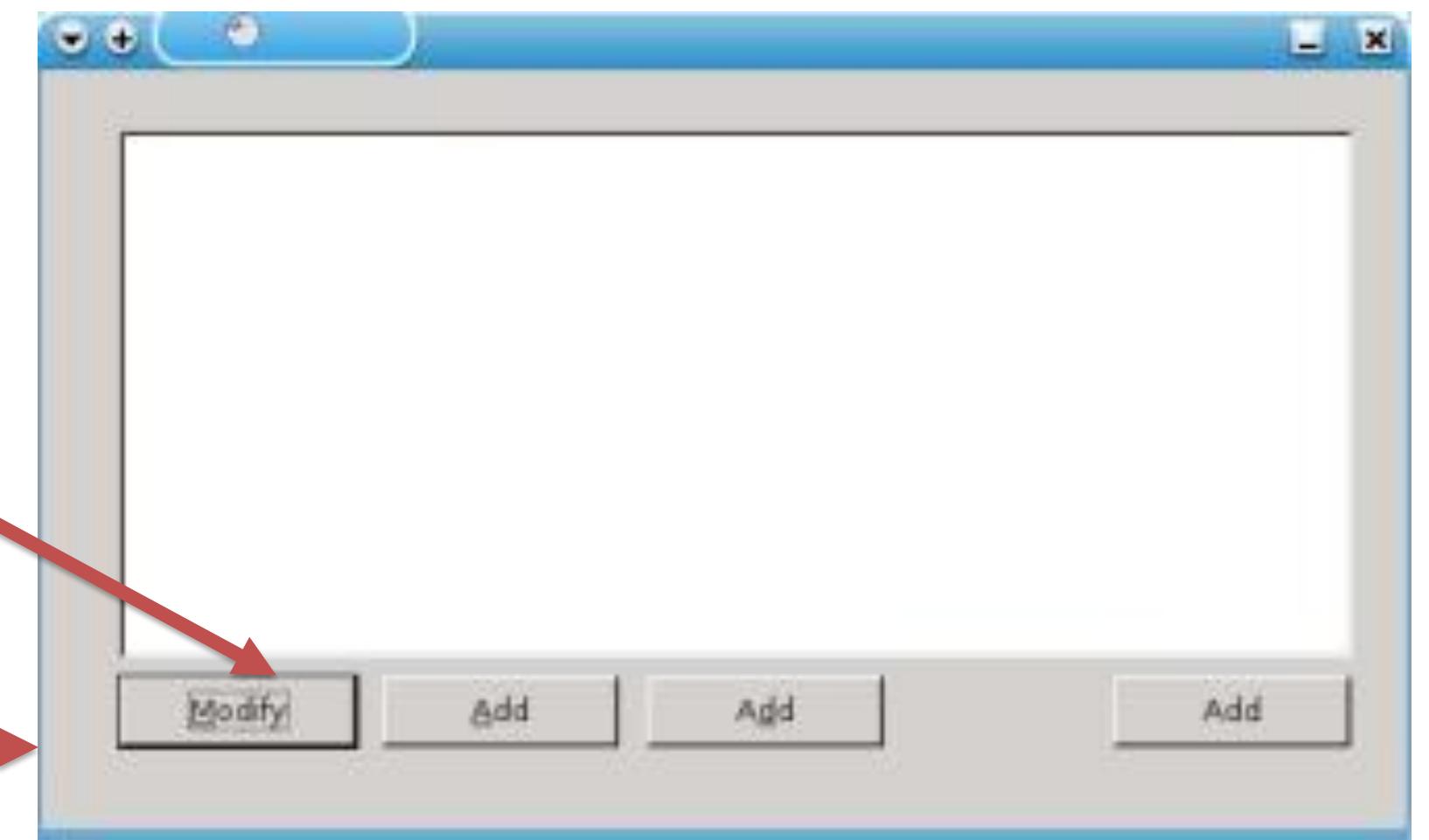
## *Elementos básicos*

- **Componentes**

Permiten al usuario interactuar con la aplicación y proporcionarle información sobre el estado de ésta. Ejemplo: botones, etiquetas, cajas de texto y listas, entre otros.

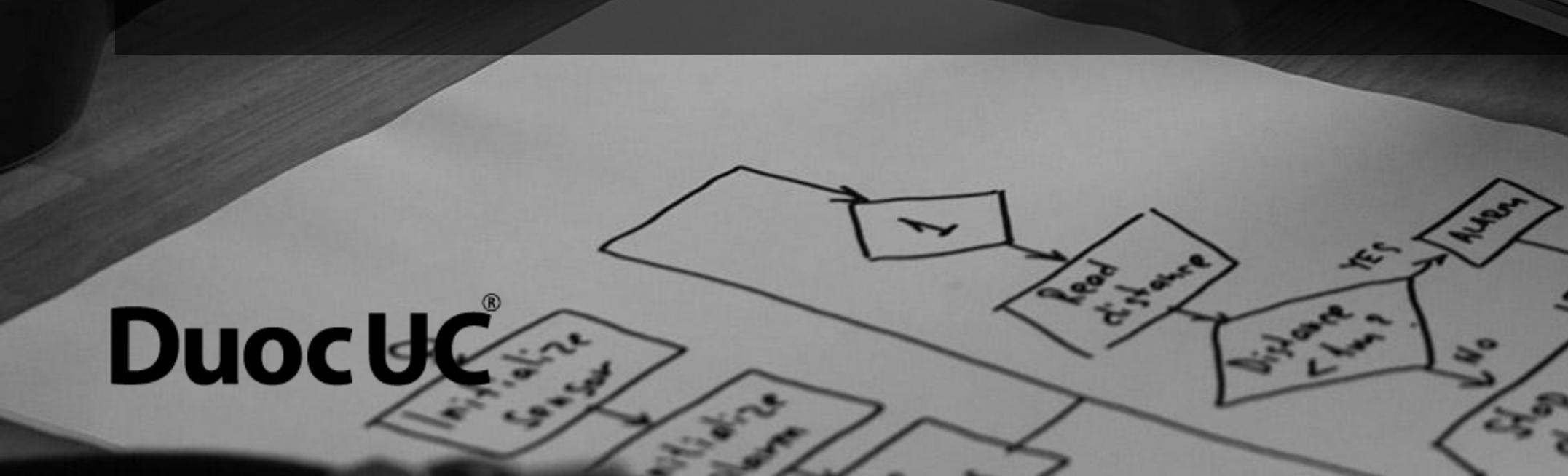
- **Contenedores**

Sirven para contener y organizar otros componentes. Ejemplo: ventanas y paneles.





## COTENEDOR 02



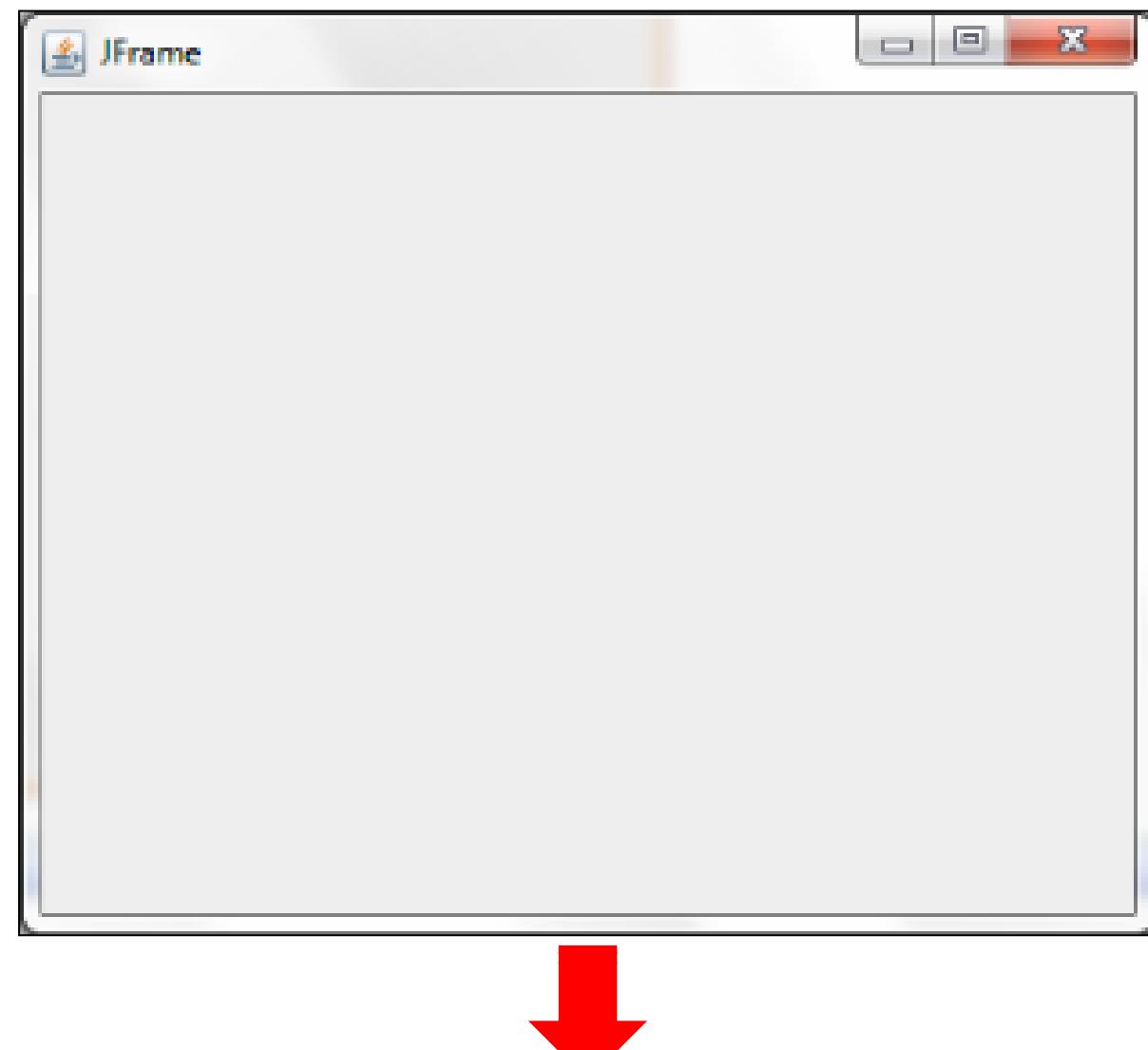
DuocUC®



# GUI

## Configurar el Contenedor

**JFrame**: Representa una ventana básica, capaz de contener otros componentes.



### Propiedades

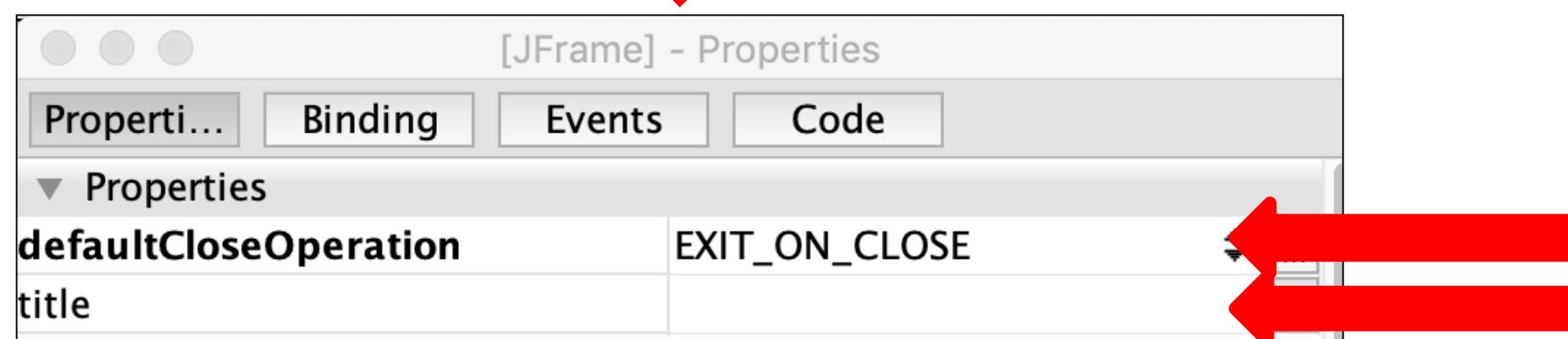
#### **defaultCloseOperation**

EXIT\_ON\_CLOSE: Termina la aplicación.

DISPOSE: Cierra sólo la ventana.

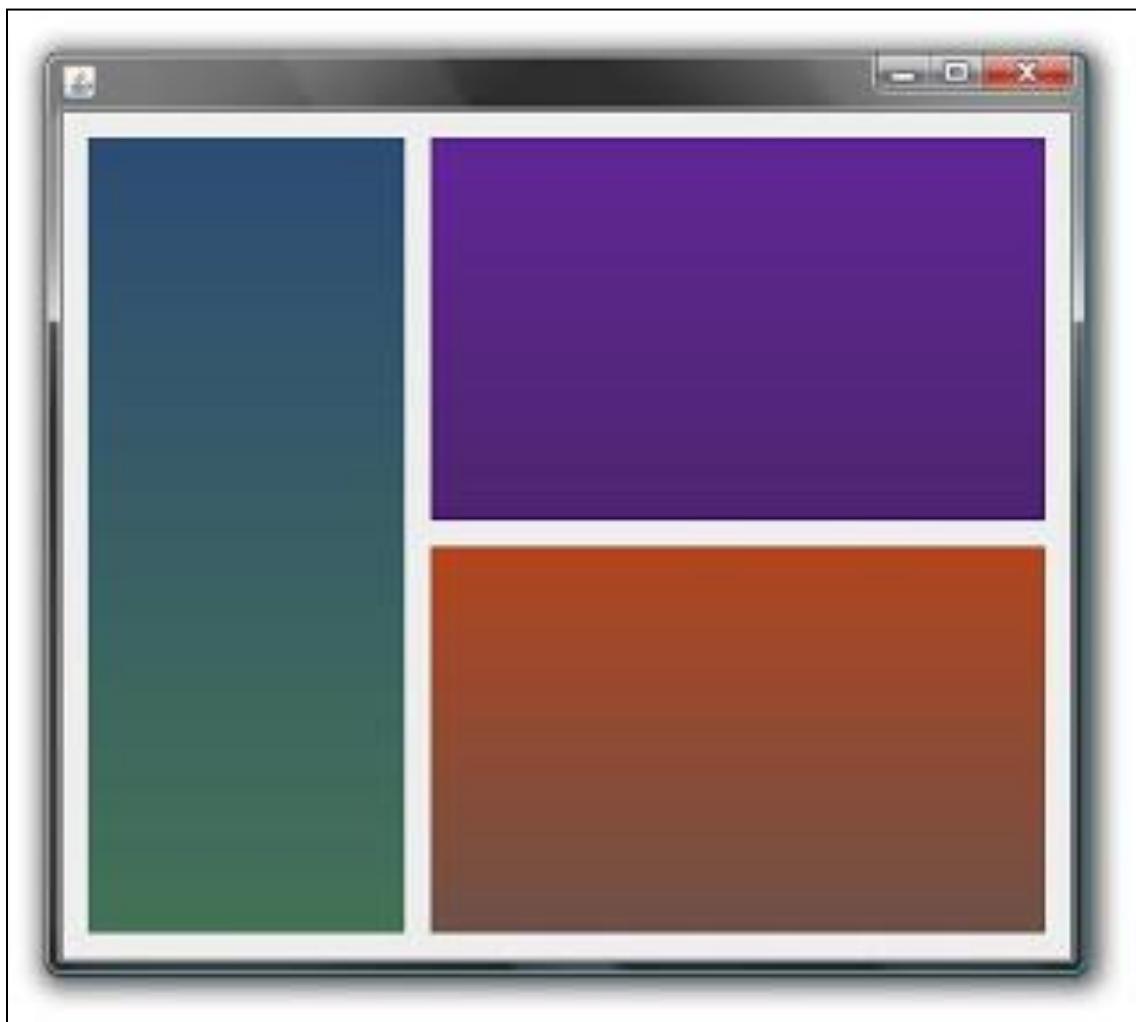
#### **title**

Título de la ventana.



# GUI

**JPanel:** Un panel sirve para agrupar y organizar otros componentes. Puede estar decorado mediante un borde y una etiqueta.



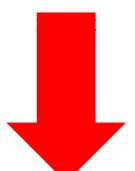
## *Propiedades*

### **background**

Color del fondo del panel.

### **border**

Borde el panel.



<b>background</b>	<input type="color" value="#0,204,255"/> [0,204,255]
<b>border</b>	[LineBorder]

# **03**

# **COMPONENTES**

# Componentes

## Añadir Componentes a los Contenedores

Las interfaces gráficas de usuario se construyen con componentes, cada uno de los cuales está preparado para responder a distintos tipos de eventos.

- **JLabel**: etiqueta para mostrar texto.
- **JTextField** y **JTextArea**: cajas de texto para la entrada de datos.
- **JButton**: botón.
- **JCheckBox**: caja de comprobación, para elegir opciones.
- **JRadioButton**: para elegir opciones mutuamente excluyentes.
- **JComboBox**: lista desplegable de opciones.
- **JMenuBar**, **JMenu**, **JMenuItem**: barras de menú.
- **JOptionPane**: ventanas de diálogo.

# Componentes

## **JLabel**

Con la clase JLabel, se puede mostrar texto no seleccionable e imágenes.

**Prefijo:** lbl

Nombre:

### **Propiedades**

text: Permite modificar el texto de la etiqueta

icon: Permite agregar una imagen

### **Métodos**

setText(String): Cambia el texto de la etiqueta

getText(): Obtiene el texto de la etiqueta

setVisible(boolean): Cambia el estado visible/invisible

setEnabled(boolean): Cambia el estado habilitado/deshabilitado

# Componentes

## **JTextField**

Un campo de texto es un control básico que permite al usuario ingresar texto.

**Prefijo:** txt

**Propiedades**

**text:** Permite modificar el texto del campo de texto

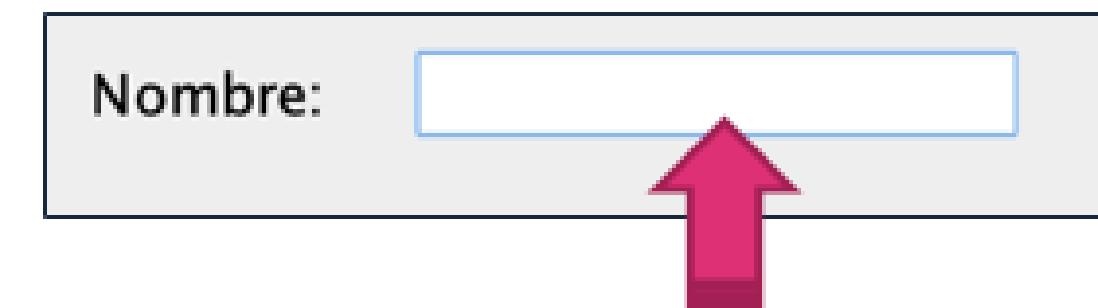
**Métodos**

**setText(String):** Cambia el texto del campo de texto

**getText():** Obtiene el texto escrito en el campo

**setVisible(boolean):** Cambia el estado visible/invisible

**setEnabled(boolean):** Cambia el estado habilitado/deshabilitado



# Componentes

## **JTextArea**

Un JTextArea es un componente básico del Swing de Java y su función principal es la de capturar texto ingresado por el teclado. Su funcionamiento es idéntico a JTextField, con la diferencia que tiene la propiedad de ser multilínea.

**Prefijo:** txa

### **Propiedades**

**text:** Permite modificar el texto del área de texto



### **Métodos**

**setText(String):** Cambia el texto del área de texto

**getText():** Obtiene el texto escrito en el textarea

**setVisible(boolean):** Cambia el estado visible/invisible

**setEnabled(boolean):** Cambia el estado habilitado/deshabilitado

# Componentes

## **JButton**

Es el botón.

Enviar

**Prefijo:** btn

## **Propiedades**

**text:** Permite modificar el texto del botón

## **Métodos**

**setText(String):** Cambia el texto del botón

**getText():** Obtiene el texto del botón

**setVisible(boolean):** Cambia el estado visible/invisible

**setEnabled(boolean):** Cambia el estado habilitado/deshabilitado

**doClick():** Ejecuta el ActionListener (click) asociado al botón

# Componentes

## JCheckBox

Se utilizan frecuentemente como botones de estado. Proporcionan información true si la caja se ha seleccionado y false en otro caso. Permite seleccionar más de una opción.

**Prefijo:** chk

### Propiedades

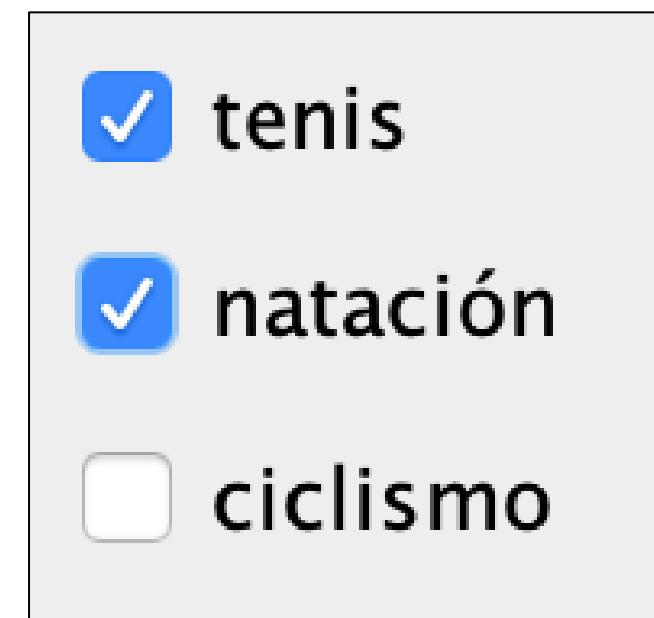
**selected:** Permite marcar el objeto por defecto

### Métodos

**getText():** Obtiene el texto del objeto

**isSelected():** devuelve true si el objeto se encuentra seleccionado, false en caso contrario

**setSelected(boolean):** permite seleccionar un objeto (true) o desmarcarlo (false)

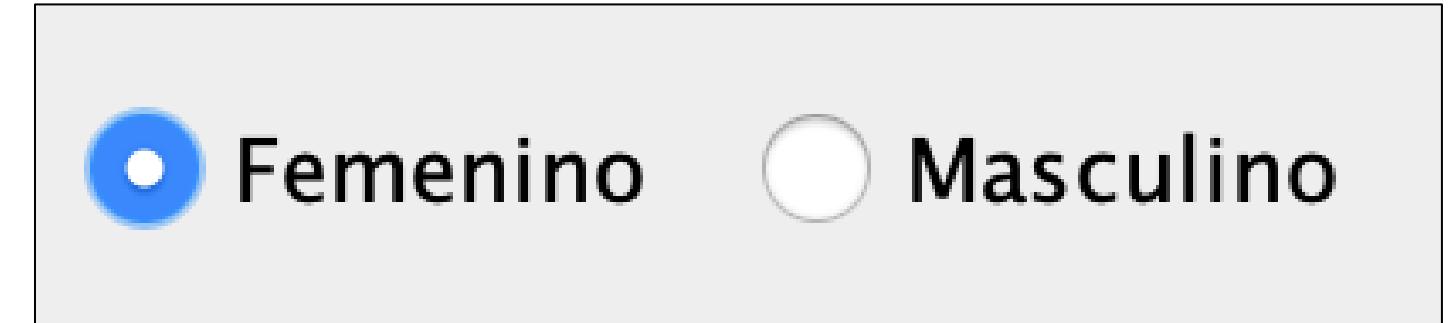


# Componentes

## **JRadioButton**

Proporcionan información true si se ha seleccionado y false en otro caso. Es un elemento que permite seleccionar una opción y sólo una, sobre un conjunto de posibilidades.

**Prefijo:** opt



## **Propiedades**

selected: Permite marcar el objeto por defecto

ButtonGroup: asocia el objeto a un grupo



## **Métodos**

getText(): Obtiene el texto del objeto

**isSelected()**: devuelve true si el objeto se encuentra seleccionado, false en caso contrario

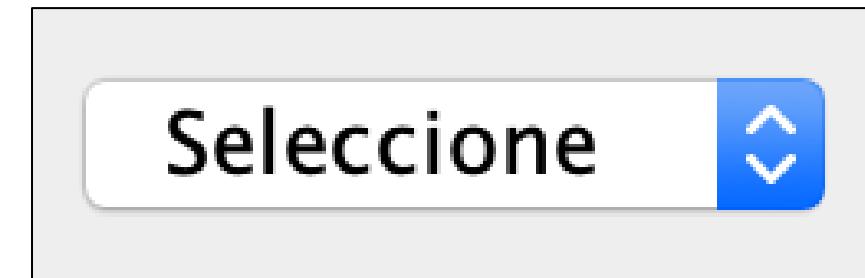
setSelected(boolean): permite seleccionar un objeto

# Componentes

## **JComboBox**

Nos permite definir un conjunto de datos o valores asociados a una caja de selección, así el usuario tendrá la oportunidad de seleccionar un dato del conjunto de datos o respuestas ya predefinido.

**Prefijo:** cbo



## **Propiedades**

**model:** Permite ingresar la lista de datos

## **Métodos Importantes:**

**getSelectedIndex()**: Obtiene la posición del dato seleccionado

**getSelectedItem()**: Obtiene el dato seleccionado

**setSelectedIndex(int)**: Selecciona el dato de la posición por parámetro

**setSelectedItem(Object)**: Selecciona el dato por parámetro

# 04 EVENTOS



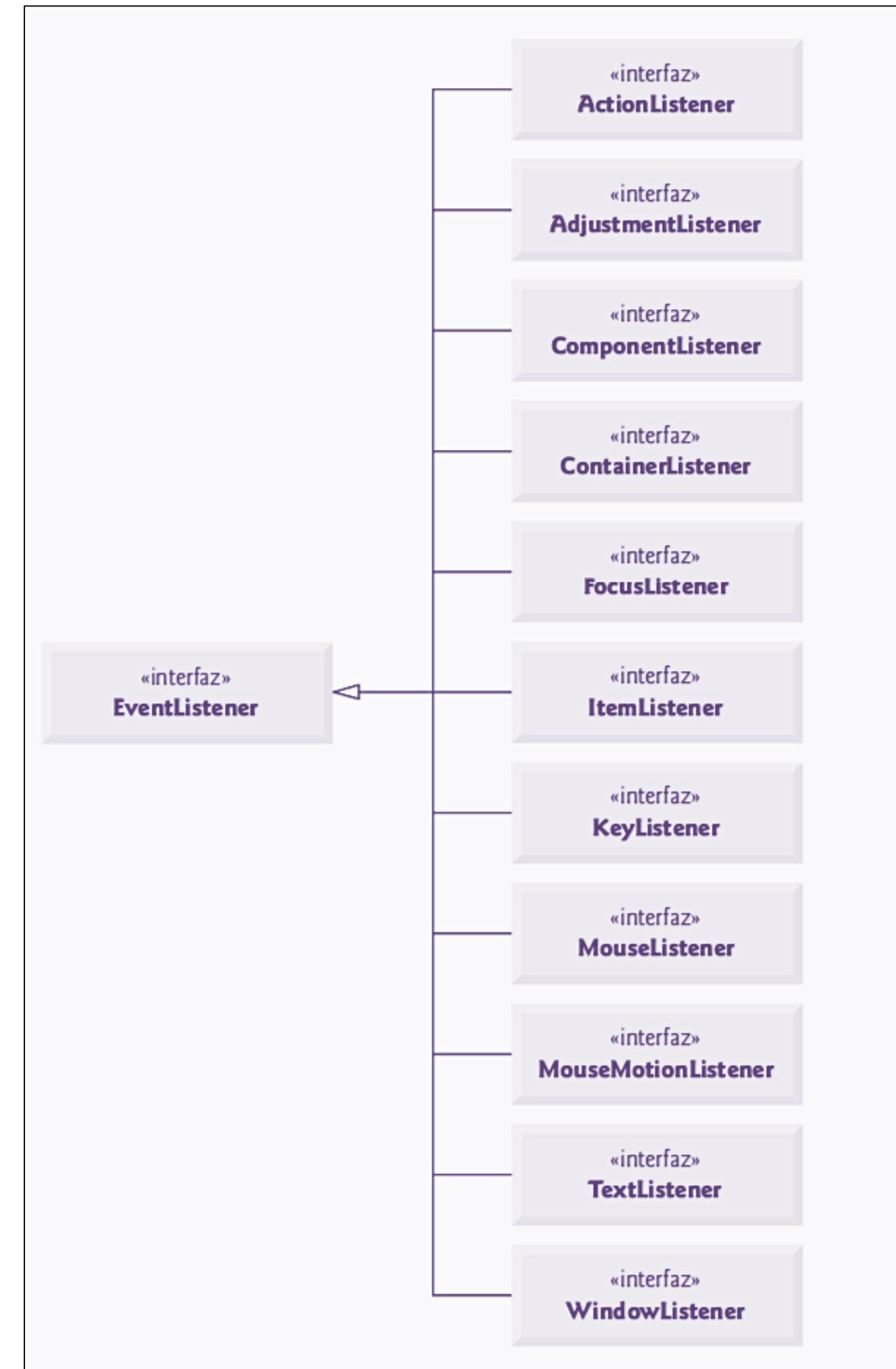
# Eventos

## Manejando Eventos

Cada vez que el usuario pulsa una tecla, un botón, mueve el mouse, entre otras, se produce un evento. Es posible notificar los eventos para que el programa actúe en consecuencia. Para ello, se crean unos objetos llamados “**listeners**” que capturan el evento realizado.



# Eventos



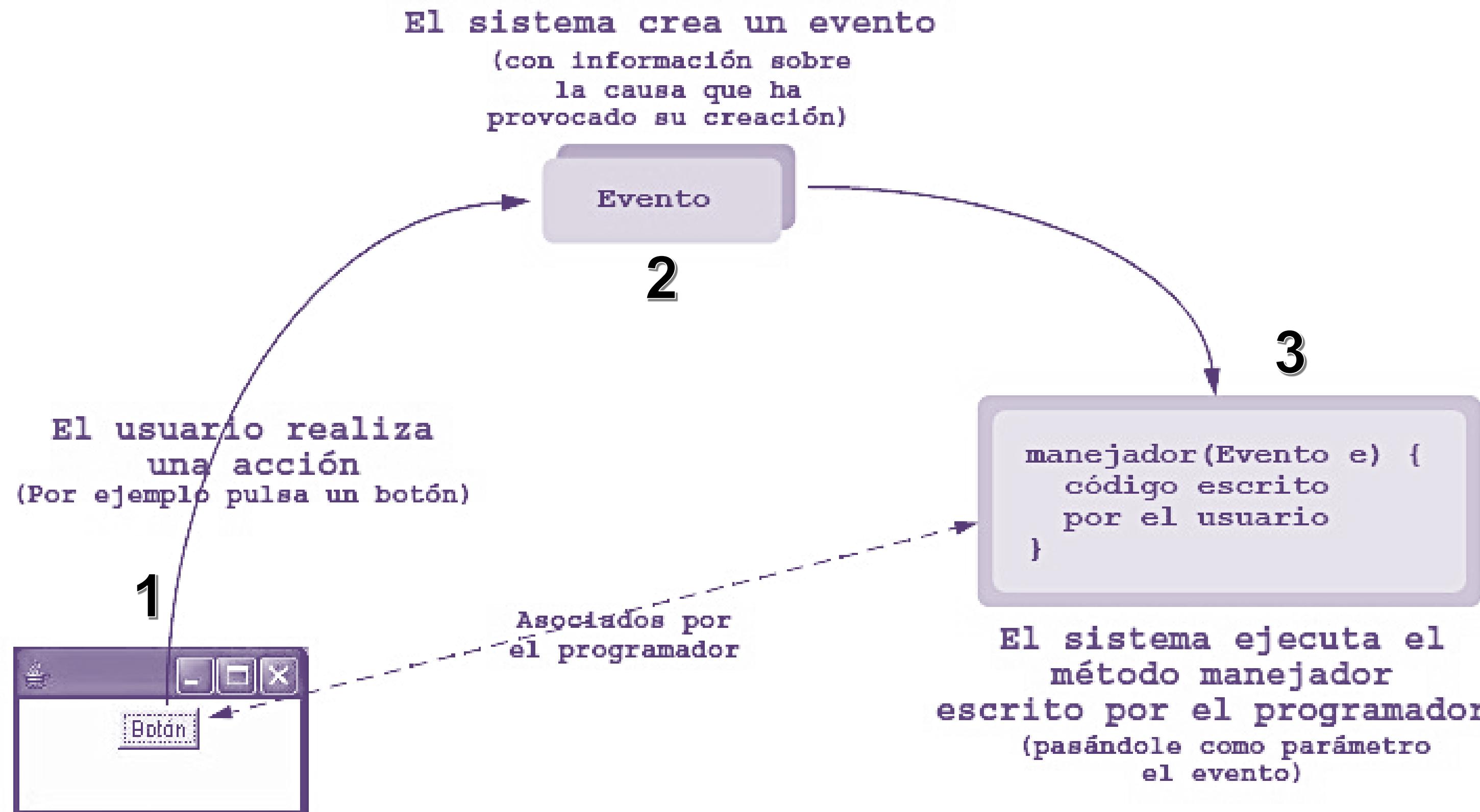
# Eventos

Listener	Acción	Eventos
ActionListener	Se produce al hacer click en un componente, también si se pulsa Enter teniendo el foco en el componente.	<b>JButton:</b> click o pulsar Enter con el foco activado en él.  <b>JList:</b> doble click en un elemento de la lista.  <b>JMenuItem:</b> selecciona una opción del menú.  <b>JTextField:</b> al pulsar Enter con el foco activado.
KeyListener	Se produce al pulsar una tecla.	<b>keyTyped:</b> al pulsar y soltar la tecla.  <b>keyPressed:</b> al pulsar la tecla.  <b>keyReleased:</b> al soltar la tecla.
FocusListener	Se produce cuando un componente gana o pierde el foco, que está seleccionado.	Recibir o perder el foco.

# Eventos

Listener	Acción	Eventos
MouseListener	Se produce cuando realizamos una acción con el ratón.	<b>mouseClicked</b> : pinchar y soltar.  <b>mousePressed</b> : presionar el botón.  <b>mouseReleased</b> : soltar el botón.
MouseMotionListener	Se produce con el movimiento del mouse.	<b>mouseDragged</b> : click y arrastrar un componente.  <b>mouseMoved</b> : al mover el puntero sobre un elemento.
WindowListener	Se produce cuando se escoge una ventana.	<b>windowOpened</b> : la primera vez que una ventana se hace visible.  <b>windowClosed</b> : cuando una ventana se ha cerrado.
ListSelectionListener	Se produce cuando se cambia la selección en una lista o en una tabla	Cambiar la selección de una lista.

# Eventos



# ¿Qué hemos aprendido?

- ✓ Crear interfaces gráficas de usuario (GUI).
- ✓ Crear botones, etiquetas, listas, campos de texto y paneles.
- ✓ Utilizar contenedores para ordenar los componentes de las GUIs.
- ✓ Programar los eventos generados por las interacciones de los usuarios.
- ✓ Solucionar un problema a través de una interfaz gráfica de usuario.



¿Qué te resultó difícil  
entender?