

# **INTERFACE GRÁFICA**

**EVENTS**

# INTERFAZ LISTENER

Clases	Métodos	Descripción
<b>ActionListener</b>	actionPerformed(ActionEvent e)	Ejecuta algún comando
<b>AdjustmentListener</b>	adjustmentValueChanged(AdjustmentEvent e)	Ajusta algún valor
<b>ComponentListener</b>	componentHidden(ComponentEvent e)	El componente se oculta
	componentMoved(ComponentEvent e)	El componente se mueve
	componentResized(ComponentEvent e)	El componente se redimensiona
	componentShown(ComponentEvent e)	El componente se visualiza
<b>ContainerListener</b>	componentAdded(ContainerEvent e)	Se añade un componente al contenedor
	componentRemoved(ContainerEvent e)	Se elimina un componente del contenedor
<b>FocusListener</b>	focusGained(FocusEvent e)	El componente obtiene el foco
	focusLost(FocusEvent e)	El componente pierde el foco
<b>ItemListener</b>	itemStateChanged(ItemEvent e)	Se modifica el estado de algún elemento del componente, como puede ser la elección de alguna casilla de verificación
<b>KeyListener</b>	keyPressed(KeyEvent e)	Se ha pulsado una tecla
	keyReleased(KeyEvent e)	Se ha soltado la tecla
	keyTyped(KeyEvent e)	Se ha tecleado un carácter

# INTERFAZ LISTENER

Clases	Métodos	Descripción
<b>MouseListener</b>	mouseClicked(MouseEvent e)	Se ha pulsado el botón del ratón
	mouseEntered(MouseEvent e)	El puntero del ratón ha entrado en el componente
	mouseExited(MouseEvent e)	El puntero del ratón ha salido del componente
	mousePressed(MouseEvent e)	Se ha presionado un botón del ratón
	mouseReleased(MouseEvent e)	Se ha soltado un botón del ratón
<b>MouseMotionListener</b>	mouseDragged(MouseEvent e)	Se está desplazando el ratón con el botón pulsado
	mouseMoved(MouseEvent e)	El puntero del ratón ha cambiado de posición
<b>TextListener</b>	textValueChanged(textEvent e)	El contenido del texto del componente ha cambiado
<b>WindowListener</b>	windowActivated(WindowEvent e)	La ventana ha sido activada
	windowClosed(WindowEvent e)	Se ha cerrado la ventana
	windowClosing(WindowEvent e)	Se ha solicitado cerrar la ventana
	windowDeactivated(WindowEvent e)	La ventana ha sido desactivada
	windowDeiconified(WindowEvent e)	Cuando se restaura la ventana a su tamaño original o se maximiza
	windowIconified(WindowEvent e)	Se ha minimizado la ventana
	windowOpened(WindowEvent e)	se ha abierto la ventana

# CLASES ADAPTER

- Problema de las interfaces:
  - Tenemos que definir todos sus métodos abstractos en las clases que las implementan. Por tanto, si una de nuestras clases implementa la interfaz `WindowListener`, deberá implementar todos los métodos asociados, aún cuando sólo utilicemos uno de ellos. (El resto de los métodos tendrán una implementación vacía).
  - Por esta razón surgen las clases adaptadoras. Estas clases adaptadoras se encargan de implementar todos los métodos de la clase de escucha ***Listener***. Así sólo necesitaremos redefinir aquellos métodos que nos van a ser útiles para gestionar eventos , sin preocuparnos del resto.

# CLASES ABSTRACTAS

- Sólo las clases que poseen más de un método tienen adaptador, y son las siguientes:

Interface	Clase Abstracta
ComponentListener	ComponentAdapter
ContainerListener	ContainerAdapter
FocusListener	FocusAdapter
KeyListener	KeyAdapter
MouseListener	MouseAdapter
MouseMotionListener	MouseMotionAdapter
WindowListener	WindowAdapter