

Ministerio de Educación Superior, Ciencia y Tecnología (Mescyt)

Centro de Tecnología y Educación Permanente TEP

Pontificia Universidad Católica Madre y Maestra (PUCMM)

Diplomado en Programación en Lenguaje JAVA

Facilitador: Ing. Eudris Cabrera

Interfaces gráficas de usuario en Java

Una pregunta que siempre hacen las personas que están iniciándose en el mundo de la programación de Java es: **Cómo puedo hacer Interfaces gráficas de usuario (GUI) en Java?**.

A continuación vamos a debatir los componentes que existen en Java para Interfaces gráficas de usuario (GUI).

La mayor parte de las aplicaciones que se utilizan hoy en día incluyen interfaces de usuario más sofisticadas que las de las aplicaciones que se ejecutan en modo consola.

Java incluye, como parte de su biblioteca de clases estándar, un conjunto de componentes para crear interfaces gráficas de usuario (GUI).

Los componentes que se utilizan en Java para crear interfaces gráficas de usuario se agrupan en dos paquetes (y sus correspondientes subpaquetes):

a) java.awt (Abstract Window Toolkit)

Los componentes AWT dependen de las facilidades gráficas ofrecidas por cada sistema operativo: los programas escritos con AWT tendrán un “look and feel” distinto en Windows y en UNIX.

b) javax.swing

SWING es 100% Java y, por tanto, completamente independiente de la plataforma: los componentes gráficos se pintan en tiempo de ejecución (por lo que las aplicaciones SWING suelen ser algo más lentas que las AWT).

En la práctica, las aplicaciones Java con interfaces gráficas de usuario suelen mezclar AWT y SWING (porque AWT se encarga de toda la gestión de eventos y SWING ofrece

componentes más sofisticados).

Componentes Swing

Las clases cuyo nombre comienza por la letra **J** forman parte de Swing. Todas las demás están incluidas en AWT (**Abstract Window Toolkit**).

Algunos términos que debemos de conocer son los siguientes:

Component: es una clase abstracta que representa a cualquier componente con representación gráfica.

Container: es un componente que puede contener a otros componentes gráficos (p.ej. JPanel).

Componentes estándar

Los frames (como JFrame) son contenedores, por lo que incluyen un “panel de contenido” (content panel) al cual se le pueden añadir componentes gráficos (etiquetas, botones, cajas de texto, etc.) y otros contenedores (como paneles JPanel).

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

Algunos de los componentes incluidos en Swing son:

- **JLabel** (etiqueta para mostrar texto)
- **JTextBox & JTextArea** (cajas de texto para la entrada de datos)
- **JButton** (botón)
- **JCheckBox** (caja de comprobación, para elegir opciones)
- **JRadioButton** (para elegir opciones mutuamente excluyentes)
- **JList** (lista de opciones)
- **JComboBox** (lista desplegable de opciones)
- **JScrollBar** (barra de scroll)
- **JTree** (árbol)
- **JTable** (tabla)
- **Menús:** JMenuBar, JMenu, JMenuItem
- **Ventanas de diálogo:** JOptionPane
- **Ventanas estándar:** JFileChooser, JcolorChooser

Distribución del espacio (layout management): antes de construir una interfaz gráfica de usuario es importante saber cómo se distribuyen especialmente los componentes.

Los **layout managers** controlan la forma en la que colocan los componentes dentro de un contenedor.

Para mayores informaciones puede visitar este enlace:

<https://docs.oracle.com/javase/tutorial/uiswing/components/jcomponent.html>

Manejo de Eventos:

En las aplicaciones en modo consola, nuestro método main determina el orden en el que se ejecutan las operaciones de nuestro programa.

En las aplicaciones con interfaces gráficas de usuario, el orden en el que se ejecutan las operaciones dependerá de las acciones del usuario.

Nosotros sólo debemos de preocuparnos de qué acciones va a realizar nuestra aplicación, cuándo han de realizarse y definir los correspondientes manejadores de eventos, que serán invocados automáticamente cuando sus eventos asociados se produzcan.

El modelo de eventos de Java Cada lenguaje de programación tiene su propio modelo de eventos:

En **Visual Basic**, cada componente genera eventos específicos y nosotros redefinimos la respuesta del componente a cada evento (modelo simple pero bastante inflexible).

En **C**, todos los eventos se sitúan en una cola: debemos de ir leyendo los eventos de esta cola y decidir qué acción realizar en función del evento extraído de la cola (tedioso de programar).

En Java, se definen clases auxiliares (**event listeners**) que pueden recibir eventos de tipos específicos (p.ej. el clic del botón del ratón). Estas clases se asocian luego a componentes específicos.

A diferencia de VB, el manejador no viene predeterminado (**mayor flexibilidad**).
A diferencia de C, cada tipo de evento se puede tratar por separado.

En términos de objetos y métodos, el manejo de eventos en Java funciona de la siguiente forma:

1. Los manejadores de eventos (**event listeners**) se registran en las distintas fuente de eventos (**event source**).
2. Una fuente de eventos (el ratón, un botón, una ventana...) envía objetos de tipo **EventObject** a todos los manejadores registrados cuando se produce un nuevo evento.
3. Cada manejador de eventos utiliza la información recibida a través del objeto de tipo **EventObject** para realizar las acciones que estime adecuadas.

En Java, los eventos están organizados en una jerarquía de clases dentro del paquete **java.awt.event**

La clase **java.util.EventObject** es la clase base de todos los eventos en Java.

La subclase **java.awt.AWTEvent** es la clase base de todos los eventos que se utilizan en la construcción de GUIs.

Cada tipo de evento **XxxEvent** tiene asociada una interfaz **XxxListener** que es la que nos permite definir manejadores de eventos.

Para simplificar la implementación de algunos manejadores de eventos, el paquete **java.awt.event** incluye clases base **XxxAdapter** que implementan las interfaces **XxxListener**.

Eudris Cabrera Rodriguez

Ingeniero Telemático

Consultor / Desarrollador Informático

LinkedIn: <http://www.linkedin.com/in/eudriscabrera>

Mayo 2016, Santiago de los Caballeros, R. D.