

Capítulo 7

Introducción a las Interfaces Gráficas de usuario

Continuar

Introducción

Se explicará qué es una interfaz gráfica, cómo han evolucionado y cómo es que debe desarrollarse un programa que incluya una interfaz; para después explicar cómo programar interfaces en Java que hagan uso de cuadros de diálogo, etiquetas, campos de texto y botones.



Definición y evolución de las Interfaces Gráficas

Una GUI (*Graphical User Interface*, Interfaz Gráfica de Usuario), es un programa computacional que por medio de un conjunto de elementos gráficos, permite introducir y visualizar información. Esto hace que el software sea “amigable”; es decir, que sea fácil de utilizar y más agradable a la vista. Surgieron ante la necesidad de hacer más accesible a la gente el uso de la computadora, porque en un inicio sólo las personas con conocimientos técnicos podían manipularlas.

En 1945, Vanner Bush publicó el artículo “*As we may think*” (traducido en español: “como podríamos pensar”), en el cual planteaba varias ideas interesantes acerca de la interacción Hombre-Máquina; como el crear un dispositivo que permitiera realizar cálculos. En 1970, Douglas Engelbart patentó el ratón de ordenador e inventó la primera GUI basándose en las ideas de Vanner Bush. En 1973, la compañía estadounidense Xerox desarrolló la primera computadora personal con una GUI llamada “Xerox Alto”. Para 1984, se lanza al mercado la primera Macintosh creada por Steve Jobs y Jef Raskin, ésta fue la primera computadora comercial en tener una GUI de ventanas multipanel.

Proceso de desarrollo de software con Interfaces Gráficas

1. Entender el problema: En este paso, se deben obtener y comprender las especificaciones del cliente/desarrollador con las cuáles deberá cumplir el software a desarrollar.
2. Diseñar la interfaz: Es recomendable esbozar en hojas de papel la(s) pantalla(s) de la interfaz gráfica de usuario. Una buena práctica consiste en mostrar los diagramas de la interfaz al cliente antes de continuar con las demás fases del proceso de desarrollo de software; esto permite aclarar malos entendidos antes de codificar.
3. Diseñar la solución funcional: En esta fase se diseña la parte funcional del software. Esto puede realizarse a través de Diagramas de Flujo, Diagramas N-S o Pseudocódigo. Tanto el diseño de la interfaz gráfica como el de la solución funcional, se incluyen dentro de las fases conocidas como *Análisis y Diseño*.
4. Codificar: Esta fase consiste en programar la solución planteada en las etapas anteriores.
5. Realizar pruebas: Una vez que se tiene un programa, es necesario realizar pruebas para corroborar su buen funcionamiento. Actualmente, los desarrollos de software no esperan a que se termine de construir el software para probarlo, sino que se realizan pruebas constantes; lo cual es una buena práctica.

Paquetes gráficos

- AWT: tiene componentes pesados que cada plataforma (sistema operativo) representa de manera determinada; por lo que no es portable.
- Swing: tiene componentes ligeros que pueden tomar diferente aspecto y comportamiento dependiendo de la plataforma donde se ejecute un programa. Está escrito totalmente en Java y es el paquete más utilizado.

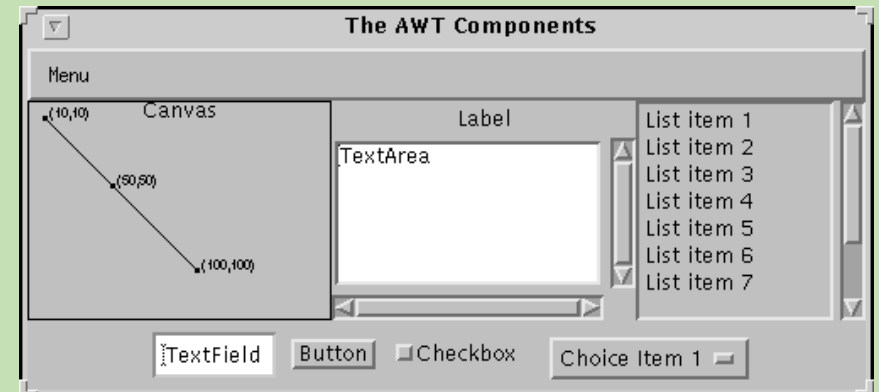
Para hacer uso de los paquetes sólo es necesario importarlos:

//Para utilizar todas las clases de Swing

```
import javax.swing.*;
```

//Para utilizar todas las clases de AWT

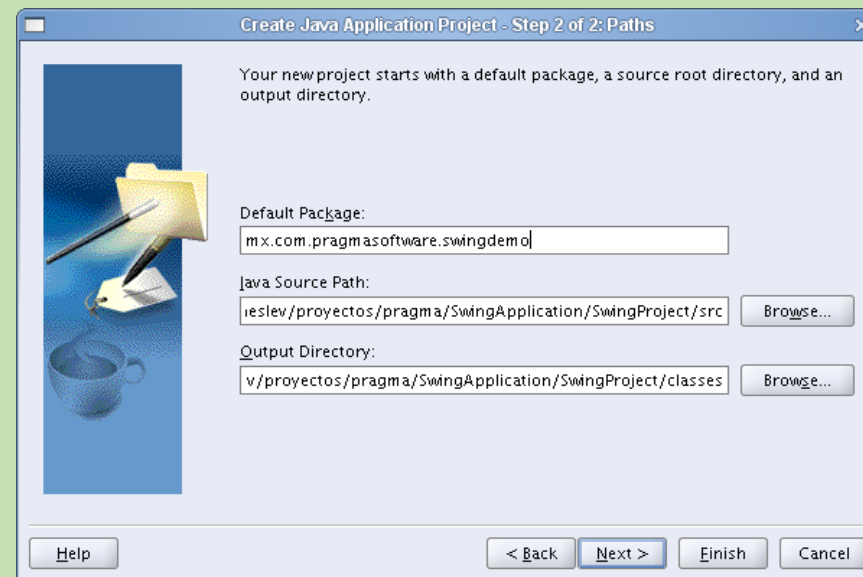
```
import java.awt.*;
```



Cuadro de diálogo

Los cuadros de diálogo son ventanas que permiten la interacción entre el usuario y el programa. Existen dos tipos: de entrada y de salida. Para utilizarlos es necesario importar la clase `JOptionPane` del paquete `Swing`.

Los cuadros de diálogos de salida, permiten mostrar información, y los cuadros de diálogos de entrada permiten recibir información introducida por el usuario.



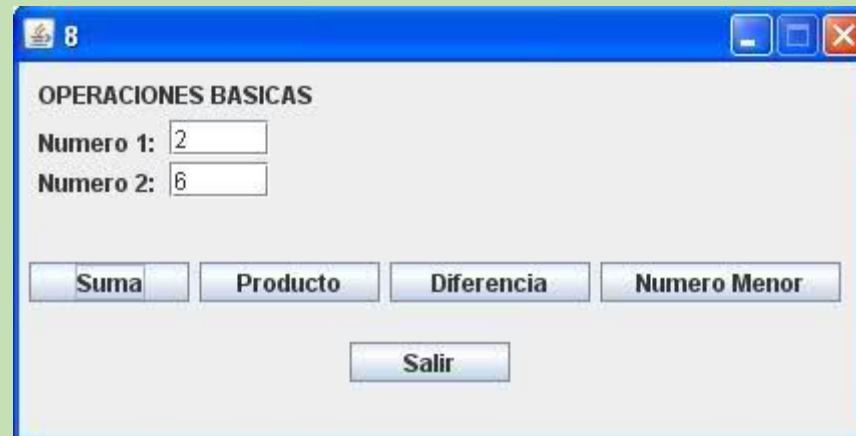
Etiquetas

Las etiquetas son elementos gráficos que permiten mostrar texto no editable en una interfaz gráfica. Para utilizar una etiqueta es necesario crear un objeto de la clase JLabel, la cual pertenece al paquete swing. Por lo que, primero es necesario importar el paquete swing. Es importante saber que cuando se utilizan elementos gráficos (como las etiquetas, cuadros de texto o botones), éstos deben estar incluidos en un panel (JFrame).



Cuadros de texto

Los cuadros de texto son elementos gráficos que permiten al usuario ingresar información a través del teclado. Los cuadros de texto funcionan de manera similar a las etiquetas, es decir se requiere importar el paquete swing. Adicionalmente deben importarse las clases `ActionEvent` y `ActionListener` para manejar los eventos de los cuadros de texto (los eventos son las acciones asociadas a los cuadros de texto). Al igual que con las etiquetas, es necesario crear el objeto, en este caso de la clase `JtextField`.



Botones

Los botones son elementos gráficos que permiten activar un evento para que se realicen ciertas acciones. Para crear un botón en Java, es necesario importar el paquete swing, las clases `ActionEvent` y `ActionListener` para manejar sus eventos y crear el objeto de la clase `JButton`.



JTextArea y JScrollPane

Los elementos de tipo JTextArea permiten contener texto plano en diferentes líneas, a diferencia de los JTextFields que solo puede contener una línea.

Para utilizar un objeto de área de texto es necesario crear un objeto de la clase JTextArea, la cual pertenece al paquete swing. Es importante mencionar que un objeto de tipo JTextArea no es capaz de manejar por si solo sus diferentes líneas a través del mecanismo de desplazamiento entre líneas. Para resolver la situación en que se desea mostrar un conjunto de líneas en un área de texto de dimensión menor a la necesaria para ciertas líneas, se puede hacer uso de un panel de desplazamiento u objeto de tipo JScrollPane, que también pertenece al paquete swing.

