

Síntesis conceptual

Asignatura: Programación
Unidad: 15. Creación de interfaces gráficas

Resumen

Java, Abstract Window Toolkit, es la librería más nativa que permite la creación de interfaces gráficas con el paquete `java.awt`, al darnos acceso a la clase común `Component`, las cuales definen los controles y la pantalla de la aplicación en desarrollo respectivamente.

La clase raíz `Component` contienen todos los componentes que intervienen en una interfaz gráfica. Estos pueden ser contenedores o controles.

- Los contenedores: usan de la clase principal `Container`, definen la estructura de ventanas de una aplicación y pueden albergar otros componentes o controles en su interior:
 - `Windows`: Contenedor en forma de ventana sin bordes, título ni barras de menú.
 - `Frame`: Ventana que usa el contenedor `Windows`, tiene barra de título y menú.
 - `Dialog`: Ventana que usa el contenedor `Windows`, tiene borde y título.
 - `FileDialog`: Ventana de diálogo que permite al usuario seleccionar archivos.
 - `Panel`: Usa de la Ventana `Frame`, permite agrupar controles dentro de él.
- Los controles: Componentes contenidos en un contenedor, permiten definir acciones o visualizar contenido en una interfaz:
 - `Button`: Botón con título, permite realizar acciones, como hacer click sobre él.
 - `Checkbox`: Botón de selección. Permite seleccionar varias opciones.
 - `Choice`: Desplegable con diferentes opciones.
 - `Label`: Etiqueta de información.
 - `TextField`: Campo para introducir texto.

Los eventos son acciones que se producen cuando un usuario interactúa con un componente. La principal clase que contienen la funcionalidad de los eventos es `EventObject`, para que estos realicen una acción requieren de un segundo elemento que capture esa acción.

Los contenedores pueden tener asociado un layout, estos permiten modificar la localización de los componentes y su tamaño, si no se usan los elementos se situarán en posiciones absolutas. Algunos de los layout son:

- `BorderLayout`: Permite colocar elementos en el centro de la ventana, así como en los cuatro bordes.
- `CardLayout`: Gestiona los componentes en el mismo espacio, se añaden en una lista y se pueden ir mostrando.
- `FlowLayout`: Coloca cada componente horizontalmente de izquierda a derecha. Si sobrepasa la fila comienza en la siguiente.
- `GridLayout`: Similar a `FlowLayout`, pero esta vez sigue el patrón de una tabla.

Los menús se sitúan en la parte de arriba de la ventana y permite organizar acciones de la aplicación. La clase principal para desarrollar un menú es `MenuComponent`.

Swing es una evolución de AWT que incorpora nuevas herramientas al tiempo que elimina algunas limitaciones, dándonos una mayor libertad de trabajo, además de permitir cambiar la apariencia con que se muestran los controles. La principal diferencia es que Swing dispone de todos los controles e incluso puede definir nuevos. La gran mayoría de componentes descritos para AWT existen para Swing y se pueden diferenciar al apreciar la letra J delante del nombre de la clase del componente, como JFrame, JDialog o JPanel.

Se añaden nuevos controles como:

- JComboBox: Lista de opciones seleccionables.
- JColorChooser: Permite seleccionar una paleta de colores.
- JPasswordField: Es un campo de texto, pero lo que se escriba se oculta con "*".
- JProgressBar: Una barra de progreso.
- JTable: Representa una tabla para mostrar datos.
- JTabbedPane: Agrupación de pestañas y por cada pestaña una agrupación de controles.
- JTree: Representa una estructura en jerarquía.

Se añaden layout como GridBagLayout, permitiendo colocar cada componente en la celda deseada de una cuadrícula ya hecha.

Trabajar directamente interfaces gráficas puede ser problemático debido a la cantidad de código que se genera por la multitud de controles y propiedades que existen. Para evitar esto surgen herramientas gráficas que nos facilitan esta tarea, permitiendo hacer *drag a drop* de los controles y que se genere automáticamente el código.

Conceptos fundamentales

- **Listener:** Elemento que capta un evento producido, por ejemplo, pulsando un botón, e inicia la acción consecuente.
- **NetBeans:** Entorno de desarrollo integrado con licencia CDDL libre diseñado para trabajar con Java.
- **Eclipse:** Entorno de desarrollo integrado con licencia pública de código abierto, que cuentan.
- **Design:** pestaña de IDE como NetBeans o Eclipse donde podemos diseñar una interfaz mediante el empleo de una interfaz gráfica.
- **Source:** pestaña de IDE como NetBeans o Eclipse donde podemos diseñar una interfaz mediante el empleo de una vista de código.