

Interfaces gráficas de usuarios simples

La interfaz de usuario es todo lo que ve el usuario del programa, permitiendo con ello que puedan interactuar entre sí. Dicha interfaz va desde simples mensajes (como los utilizados hasta ahora) hasta las interfaces gráficas; estas últimas se caracterizan porque están formadas por imágenes y objetos gráficos. El éxito de un programa depende en una gran medida de su interfaz.

Es muy importante distinguir en el diseño de una interfaz gráfica:

Objetos gráficos: botones, etiquetas, listas, barras de desplazamiento, campos de texto...

Evento: un evento se produce cuando el usuario pulsa sobre alguno de los objetos gráficos de los que está formada la interfaz, haciendo que el programa reaccione llevando a cabo la tarea que corresponda.

Trabajando con código:

Vamos a crear una ventana donde aparezca una etiqueta "Hola Mundo" y un botón para cerrar la ventana: (Proyecto PrimeraVentana)

Se crean dos clases: Ventana y Principal.

Ventana incluirá el constructor de ventanas que llamará a un método **visible()**, en el que se incluirán todas las instrucciones para dar forma a la ventana e inicializar los objetos.

Para que la clase Ventana sea una clase capaz de generar una ventana de Windows tiene que heredar de la clase JFrame, clase que proporciona los métodos necesarios para trabajar en un marco adecuado.

Dentro de la clase Ventana se van añadiendo las siguientes instrucciones una a una, ejecutando el proyecto cada vez que se añada una para ver los cambios que se van produciendo sobre la ventana.

```
this.setVisible(true);
```

```
this.setSize(400,400); //Probar a cambiar medidas
```

```
this.setLocation(300, 30); //Probar a cambiar medidas
```

```
this.setTitle("La ventana de ...");
```

¿Podemos hacer `this.setColor(yellow);`? ¿Algún método para darle color al fondo?

Para añadir objetos gráficos a una ventana, es aconsejable que dicha ventana tenga un contenedor, ya que un objeto `JFrame` es como el marco de una pizarra sin fondo. Si queremos añadir un solo objeto a dicho programa no habrá problema, no haría falta el contenedor, ya que dicho objeto ocuparía toda la ventana, pero si queremos colocar más de un objeto hace falta que tenga un contenedor. El contenedor sería el fondo de dicha pizarra.

```
JPanel contenedor=(JPanel) this.getContentPane(); //this hace referencia a la propia ventana
```

A partir de aquí podemos añadir todos los objetos que nos sean necesarios.

Colocaremos una etiqueta con el fin de visualizar “Hola Mundo”; para ello creamos el objeto etiqueta y lo añadimos al contenedor:

```
JLabel etiqueta=new JLabel(“mensaje que queremos ver en la etiqueta”);  
contenedor.add(etiqueta);
```

Vamos a hacer lo mismo con un botón, sobre el que se lea “Salir”. Queremos que, si el usuario pulsa sobre dicho botón, visualice un mensaje por la salida estándar y cierre el programa.

Nos damos cuenta de que al añadirle un nuevo componente gráfico al programa hay que decir dónde queremos añadir los componentes para no pisar unos a otros, así que necesitamos establecer el Layout del contenedor, que en principio puede ser null, para indicarle después en qué columna y fila, respectivamente, se va a colocar la esquina izquierda superior del objeto en la ventana, así como el ancho y el largo del objeto:

```
contenedor.setLayout(null);  
  
boton.setBounds (25, 75, 100, 50); //Probar a cambiar medidas  
etiqueta.setBounds(25, 25, 75, 20); //Probar a cambiar medidas
```

Si no le decimos dónde queremos colocar los objetos, por defecto ocuparían toda la ventana y sólo se vería el último.

Hay otras formas de colocar los diferentes objetos que aparecen en la ventana, mucho más fáciles, ya que no es necesario proporcionar la posición exacta donde lo queremos colocar. alguna de estas formas:

```
contenedor.setLayout(new FlowLayout());  
contenedor.setLayout(new BorderLayout());
```

Creación de controladores de eventos

Una vez que tengamos diseñada la ventana hay que hacer que el programa reaccione ante la acción del usuario (sobre la interfaz). Es lo que se conoce como creación de controladores de eventos. Para poner a la escucha a un componente gráfico de la ventana (controlar el evento), hay que realizar tres pasos.

En nuestro ejemplo queremos que al pulsar el objeto botón, se visualice un mensaje y se cierre el programa.

1. El objeto hay que ponerlo a la escucha sobre una determinada acción.

El compilador nos mostrará los métodos que podemos ejecutar para poner el objeto a la escucha, es decir todos aquellos que contienen la palabra "listener".

boton.addActionListener (instaciacion de un objeto de tipo ActionListener)

2. Cuando elegimos el método que pone a la escucha a dicho objeto, dicho método requiere como argumento un objeto que implemente dicha interfaz. En nuestro ejemplo, la interfaz ActionListener tiene un único método: `actionPerformed(ActionEvent e)`, así que este método que tenemos que definir es el que se ejecutará cuando pulsemos el botón.

3. En dicho método se escribirá el código que queremos que se ejecute cuando se pulse el botón.

En nuestro caso como queremos que cuando se pulse el botón se visualice un mensaje y se cierre el programa habrá que poner, por ejemplo:

```
public void actionPerformed (ActionEvent e){  
    System.out.println("Se ha pulsado el botón cerrar");  
    System.exit(0);  
}
```

Observación

Si se cierra la ventana (x), la ventana se cierra, pero no el programa. Si queremos cerrar el programa una opción es cambiar una de las propiedades de la ventana con el código siguiente:

```
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Trabajando en modo diseño:

Vamos a crear la misma ventana esta vez utilizando las herramientas de diseño de NetBeans (proyecto SegundaVentana).

Se crea un proyecto, esta vez incluye una clase VentanaGrafica.

VentanaGrafica es un formulario JFrame en el que veremos que se crea código por defecto. Por ejemplo:

```
public VentanaGrafica() {  
  
    initComponents();  
  
}
```

La clase VentanaGrafica incluirá el constructor de ventanas que llamará a un método **initComponents()**, en el que se incluirán todas las instrucciones para dar forma a la ventana e inicializar los objetos.

Aquello que se diseñe gráficamente irá generando el código correspondiente.

Una vez creada visualmente la ventana, se añadirán las acciones necesarias para generar los eventos asociados a los objetos colocados.