

## EJERCICIO GUIADO. JAVA: VECTORES Y EVENTOS

### Asignación de eventos a los elementos de un vector de componentes

En Java se pueden crear vectores de componentes, como ya se vio en la hoja guiada anterior. Estos vectores pueden contener un grupo de etiquetas, o botones, cuadros de textos, etc.

Gracias a los vectores de componentes se pueden crear rápidamente por código estos componentes y trabajar sobre ellos sin tener que usar muchas líneas de código.

En esta hoja se verá como se pueden asignar eventos a los componentes de un vector.

### Ejercicio guiado

1. Crea un nuevo proyecto en java.
2. Empecemos definiendo desde código las características de la ventana. Crearemos el método *CreacionVentana*, y lo llamaremos desde el constructor del proyecto:

```
/** Creates new form ventana */
public ventana() {
    initComponents();
    CreacionVentana(); ←
}

public void CreacionVentana() {
    this.setSize(600,200);
    this.setTitle("Ejemplo de vectores y eventos");
}
}
```

3. Vamos a situar en la ventana 5 botones, y para ello usaremos un vector de botones, que tendremos que declarar en la zona de variables globales:

```
public class ventana extends javax.swing.JFrame {

    JButton vBotones[]; ←
}
```

4. Dentro del procedimiento *CreacionVentana* construiremos el vector y construiremos cada uno de los botones que contiene, asignando las distintas características de cada uno. Debes añadir el siguiente código a *CreacionVentana*:

```

public void CreacionVentana() {
    this.setSize(600,200);
    this.setTitle("Ejemplo de vectores y eventos");

    //texto para los botones
    String textoBotones[]={"Aceptar","Cancelar","Aplicar","Cerrar","Salir"};

    //creacion de los botones:
    vBotones=new JButton[5]; //vector de 5 botones
    int i;
    for (i=0;i<vBotones.length;i++) {
        vBotones[i]=new JButton(); //construyo cada botón
        vBotones[i].setText(textoBotones[i]);
        vBotones[i].setBounds(10+i*100,100,90,20);
        this.getContentPane().add(vBotones[i]);
    }
}

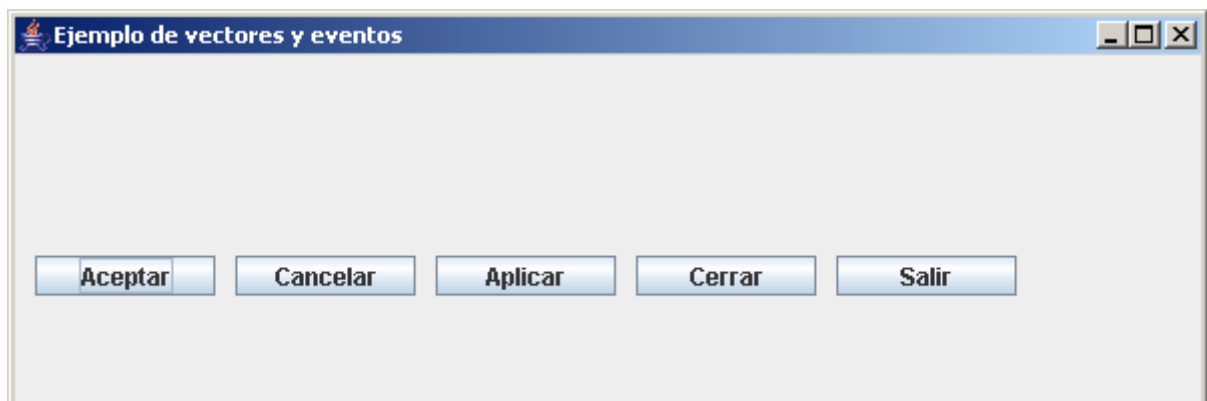
```

5. En el código anterior se crea un vector de cadenas (String) con el texto de cada botón.

Luego se construye el vector de botones, asignándole 5 botones.

A continuación, usando un for, construimos cada botón del vector, le asignamos el texto correspondiente, le asignamos su posición y su tamaño, y lo colocamos en el panel de contenido de la ventana.

6. Ejecuta el programa para observar su aspecto hasta ahora:



7. Ahora, interesaría asignar un evento *actionPerformed* a cada botón. Esto se hará aprovechando el que todos los botones pertenezcan a un vector, usando un bucle for. Añade el siguiente código al final de *CreacionVentana*:

```

for (i=0;i<vBotones.length;i++) {
    vBotones[i].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            botonesActionPerformed(evt);
        }
    });
}

```

Este código asigna a cada botón un evento *actionPerformed* que hace una llamada a un procedimiento al que se le ha llamado *botonesActionPerformed*. (Esta llamada da error porque aún no ha sido programado el procedimiento. Esto se hará a continuación.)

Hay que tener en cuenta que cuando se pulse cualquiera de los botones, se ejecutará el procedimiento *botonesActionPerformed*. Programa este procedimiento (fuera de *CreacionVentana*, por supuesto) de la siguiente forma:

```

public void botonesActionPerformed(ActionEvent evt) {
    JOptionPane.showMessageDialog(null,"Has pulsado uno de los botones");
}

```

8. Ejecuta el programa y observa el resultado. Cuando pulsas cualquiera de los botones, se ejecuta la respuesta al evento y esta lo que hace es mostrar un mensaje a través de un *JOptionPane* donde aparece el texto "Has pulsado uno de los botones".

El procedimiento *botonesActionPerformed* sirve como respuesta para todos los botones. El problema que plantea esto es, ¿cómo saber qué botón se ha pulsado? Es aquí donde entra en funcionamiento el parámetro *evt* que suelen llevar todos los eventos.

9. Modifica el procedimiento *botonesActionPerformed* de forma que quede así:

```

public void botonesActionPerformed(ActionEvent evt) {
    JButton botonPulsado = (JButton) evt.getSource();
    JOptionPane.showMessageDialog(null,"Has pulsado el botón "+botonPulsado.getText());
}

```

Este código resulta muy interesante. El parámetro *evt* tiene siempre un método llamado *getSource* que devuelve el componente de la ventana sobre el que ha sucedido el evento. De esta manera obtenemos el botón que ha sido pulsado. (Observa que es necesario hacer un *cast*) Aprovechamos esto para introducir el botón en una variable auxiliar que hemos llamado *botonPulsado*.

Luego solo hay que usar la variable *botonPulsado* como un botón normal y corriente. Por ejemplo, la usamos en el *JOptionPane* mostrando el texto que contiene el botón pulsado con el método *getText*.

10. Ejecuta el programa y observa el resultado.

## **CONCLUSIÓN**

Se pueden asignar eventos a los elementos de un vector a través de un for que recorra los distintos elementos del vector y le asigne el evento o eventos necesarios.

Cuando se asigna un mismo procedimiento de respuesta a eventos a varios componentes, es necesario usar el parámetro evt junto con el método getSource para averiguar el componente sobre el que ha sucedido el evento.

La forma general de acceder al elemento sobre el que sucedió el evento es la siguiente:

```
TipoComponente variable = (TipoComponente) evt.getSource();
```

La variable *variable* contendrá el componente activado.