

Control de edición

La mayoría de los applets disponen de controles de edición en los que se introducen datos para su procesamiento por el programa. La verificación de los datos que se introducen en los controles de edición tiene gran importancia para el correcto funcionamiento del programa.

La verificación del dato, se puede realizar mientras se teclean los caracteres o bien después de haber introducido dicho dato en el control de edición. El primer caso, se emplea para eliminar los caracteres no deseados que tecleamos sin darnos cuenta, por ejemplo letras cuando se requieren caracteres numéricos o viceversa. El segundo caso, se emplea por ejemplo, para verificar que la cantidad introducida está dentro de un intervalo apropiado. Este último caso, no lo estudiaremos ya que requiere conocer primero el concepto de excepción.

El control de edición

Se crea el control de edición llamando a uno de sus constructores

```
TextField textEntero = new TextField();
```

Se establece el tamaño del control mediante *setColumns*

```
textEntero.setColumns(4);
```

Se puede habilitar o inhabilitar el control de edición, es decir, puede hacerse de lectura/escritura o de sólo lectura.

```
textEntero.setEditable(false);
```

Para poner texto en el control de edición, se llama a *setText*.

```
textEntero.setText("3");
```

Para obtener el texto del control de edición, se emplea *getText*.

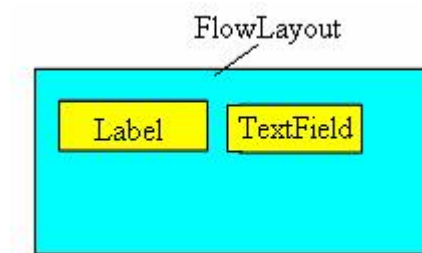
```
String texto=textEntero.getText();
```

Verificación de los datos mientras se introducen

Propósito

Filtrar los caracteres que se introducen en el control de modo que solamente se muestren caracteres no numéricos.

Diseño



Crear un applet y situar sobre el applet en el modo diseño un control etiqueta (Label) y un control de edición (TextField).

Cambiar sus propiedades en sus respectivas hojas de propiedades

Establecer *FlowLayout* como gestor de diseño del applet

Respuesta a las acciones del usuario

Para gestionar los sucesos procedentes del teclado que se producen en un componente, se han de seguir los siguientes pasos.

- Se crea una clase que implemente el interface *KeyListener*, y que ha de definir las tres funciones declaradas en dicho interface

```
public interface KeyListener extends EventListener {
    public void keyTyped(KeyEvent e);
    public void keyPressed(KeyEvent e);
    public void keyReleased(KeyEvent e);
}
```

- Mediante *addKeyListener* se conecta el componente con un objeto de la clase que maneja los sucesos originados en dicho componente.
- La información acerca del suceso viene encapsulada en un objeto de la clase *KeyEvent*.

Cuando se implementa un interface es necesario definir todas las funciones declaradas en el interface, aunque no contengan código entre las llaves de apertura y cierre.

```
class ValidaCaracter implements KeyListener{
    public void keyTyped (KeyEvent e){}
    public void keyReleased(KeyEvent e){}
    public void keyPressed(KeyEvent ev){
        int codigo=ev.getKeyCode();
        if(codigo>=KeyEvent.VK_0 && codigo<=KeyEvent.VK_9){
            ev.consume();
        }
    }
}
```

El objeto *ev* de la clase *KeyEvent* nos suministra código de la tecla que se ha pulsado que se extrae mediante su función miembro *getKeyCode* o el carácter tecleado mediante *getKeyChar*. Si dicho código corresponde a una tecla numérica, el suceso se consume, concluye su procesamiento, y por tanto, no se muestra en el control de edición.

```
int codigo=ev.getKeyCode();
if(codigo>=KeyEvent.VK_0 && codigo<=KeyEvent.VK_9){
    ev.consume();
}
```

Otra forma equivalente sería la siguiente

```
char tecla=ev.getKeyChar();
if(tecla>='0' && tecla<='9'){
    ev.consume();
}
```

Asociamos mediante *addKeyListener*, el control de edición *textIntro* en el que ocurren los sucesos relacionados con el teclado, con un objeto de la clase *ValidaCaracter* que maneja dichos sucesos.

```
ValidaCaracter valChar=new ValidaCaracter();
textIntro.addKeyListener(valChar);
```

o bien, en una sola línea

```
textIntro.addKeyListener(new ValidaCaracter());
```

De un modo análogo, se podría definir una clase que filtrase los caracteres numéricos.

El código completo de este ejemplo, es el siguiente

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

public class EdicionApplet1 extends Applet {
    TextField textIntro = new TextField();
    Label label1 = new Label();
    FlowLayout flowLayout1 = new FlowLayout();

    public void init() {
        textIntro.setColumns(10);
        label1.setText("Introducir caracteres NO numéricos");
        this.setLayout(flowLayout1);
        this.add(label1, null);
        this.add(textIntro, null);
        ValidaCaracter valChar=new ValidaCaracter();
        textIntro.addKeyListener(valChar);
    }
}

class ValidaCaracter implements KeyAdapter{
    public void keyTyped (KeyEvent e){}
    public void keyReleased(KeyEvent e){}
    public void keyPressed(KeyEvent ev){}
```

```
int codigo=ev.getKeyCode();  
if(codigo>KeyEvent.VK_0 && codigo<=KeyEvent.VK_9){  
    ev.consume();  
}  
}  
}
```