

# Ejemplo sencillo de JComboBox

---

## Sumario

---

**El código completo del ejemplo con JComboBox**

**Rellenar los items del JComboBox**

**Añadir un listener al JComboBox**

**Recoger el valor seleccionado en el JComboBox**

**Meter más información en el JComboBox**

**Otros enlaces sobre JComboBox**

Un pequeño programa sencillo de ejemplo de uso del *JComboBox* de *Java*. Abre una ventana con un *JComboBox* y un *TextField*. Cada vez que seleccionemos uno de los items, se mostrará en el *TextField*. Si borras el *TextField* y seleccionas otra vez el mismo item del *JComboBox*, verás que no se pinta en el *TextField*. Esto es así porque el *addActionListener()* sólo avisa cuando hay un cambio en el item seleccionado.

Cualquier duda sobre esto o de Java en general suelo atender en el [foro de java](http://foro.chuidiang.org) (<http://foro.chuidiang.org>)

## El código completo del ejemplo con JComboBox

---

El código completo sería este. Más abajo vamos explicando con detalle los puntos interesantes.

```
package chuidiang.ejemplos;

import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JTextField;
import javax.swing.WindowConstants;

/**
 * Ejemplo de uso de JComboBox.
 * @author chuidiang
 */
public class PruebaJComboBox {

    private JTextField tf;
    private JComboBox combo;
    private JFrame v;

    /**
     * @param args
     */
    public static void main(String[] args) {
        new PruebaJComboBox();
    }

    public PruebaJComboBox()
    {
        // Creacion del JTextField
        tf = new JTextField(20);
```

```
// Creacion del JComboBox y añadir los items.
combo = new JComboBox();
combo.addItem("uno");
combo.addItem("dos");
combo.addItem("tres");

// Accion a realizar cuando el JComboBox cambia de item seleccionado.
combo.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        tf.setText(combo.getSelectedItem().toString());
    }
});

// Creacion de la ventana con los componentes
v = new JFrame();
v.getContentPane().setLayout(new FlowLayout());
v.getContentPane().add(combo);
v.getContentPane().add(tf);
v.pack();
v.setVisible(true);
v.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
}
```

## Rellenar los items del JComboBox

Hay formas más complejas de poner items en un *JComboBox*, pero la más sencilla que es la usada en este ejemplo consiste en añadir *String* de texto. Simplemente basta llamar al método *addItem()*.

```
// Creacion del JComboBox y añadir los items.
combo = new JComboBox();
combo.addItem("uno");
combo.addItem("dos");
combo.addItem("tres");
```

## Añadir un listener al JComboBox

Si nos queremos enterar de cuando el usuario cambia la selección del *JComboBox* debemos añadirle un *listener*, usando su método *addActionListener()*. A este *listener* sólo se nos avisará si hay cambio en la selección del *JComboBox*. Si el usuario abre el *JComboBox* y elige la misma opción que ya estaba seleccionada, no recibiremos ningún aviso.

```
combo.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Aquí hacemos lo que queramos hacer.
    }
});
```

## Recoger el valor seleccionado en el JComboBox

Bien en el *listener* cuando hay un cambio en la selección del *JComboBox*, bien cuando nos interese, podemos coger el item seleccionado sin más que llamar al método *getSelectedItem()*. Este método nos devolverá un *Objet* que será alguno de los *item* que pasamos en el *addItem()*. En este caso, *String*, así que podemos hacer el *cast* para saber exactamente la opción seleccionada.

```
String itemSeleccionado = (String)combo.getSelectedItem();
if ("uno".equals(itemSeleccionado) {
```

```
...  
}
```

## Meter más información en el JComboBox

Hemos comentado que hay formas más complejas de añadir items al *JComboBox*. Podemos añadir cualquier clase que queramos. El *JComboBox* llamará al método *toString()* de esa clase para saber qué pintar en la opción del menú, pero tanto *addItem()* como *getSelectedItem()* usarán el objeto completo.

Esto es útil, por ejemplo, si hemos recogido las opciones del *JComboBox* de una base de datos y queremos mantener el id de la base de datos además de un texto que sea visible en el *JComboBox*. Imagina, por ejemplo, que leemos de base de datos una lista de idiomas con id y son los que queremos mostrar en el *JComboBox*, pero sin perder el id de base de datos. Hacemos una clase como la siguiente

```
public class Pais {  
    private long id;  
    private String nombrePais;  
    // getter y setter  
    public String toString() {  
        return nombrePais;  
    }  
}
```

En el *JComboBox* podemos añadir items de esta forma

```
Pais francia = new Pais(),  
francia.setId(112234);  
francia.setNombrePais("Francia");  
...  
combo.addItem(francia);
```

El *JcomboBox* mostrará "Francia" que es lo que le devolvería el método *toString()* de la clase *Pais*. Cuando queramos recoger el item seleccionado

```
Pais paisSeleccionado = (Pais)combo.getSelectedItem();  
System.out.println("El id del pais es " + paisSeleccionado.getId());  
System.out.println("El nombre del pais es " + paisSeleccionado.getNombrePais());
```

## Otros enlaces sobre JCombobox

- [Poner iconos en un JCombobox](#)
- [JComboBox dependiente de otro JComboBox](#)

Obtenido de «[https://chuidiang.org/index.php?title=Ejemplo\\_sencillo\\_de\\_JComboBox&oldid=6437](https://chuidiang.org/index.php?title=Ejemplo_sencillo_de_JComboBox&oldid=6437)»

Esta página se editó por última vez el 25 jun 2022 a las 11:44.