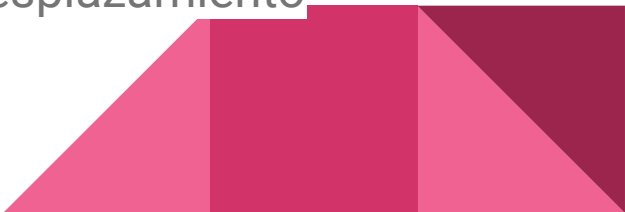


Enlazando Componentes

José Jarones

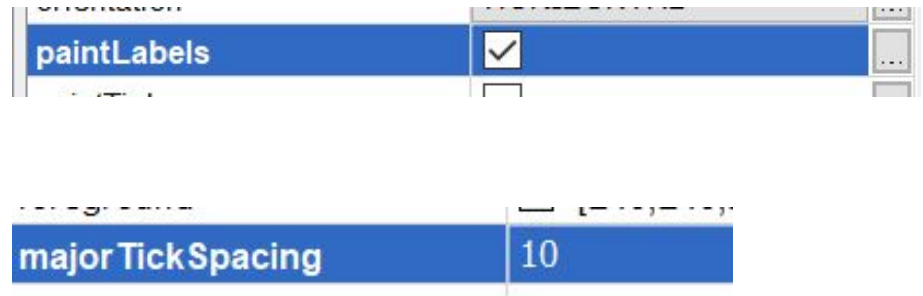
JSlider

Las propiedades importantes de JSlider en Netbeans son:

- **majorTickSpacing**: Número de valores entre dos marcas
 - **maximun**: Valor máximo
 - **minimun**: Valor mínimo
 - **paintLabels**: Muestra el valor debajo de cada marca
 - **paintTicks**: Muestra una marca según el majorTickSpacing
 - **paintTrack**: Dibuja la linea para ver la longitud de desplazamiento
- 

JSlider y JTextField

En las propiedades de JSlider marcamos `paintLabels` y `majorTickSpacing` a 10



Añadir evento StateChanged al Slider

Cada vez que cambie el estado del JSlider, actualizamos el valor del jTextField

```
private void jSlider1StateChanged(javax.swing.event.ChangeEvent evt) {  
    // TODO add your handling code here:  
    this(jTextField1.setText(""+this.jSlider1.getValue()));  
}
```



jComboBox y Datos

Vamos a enlazar un JComboBox con datos



Clase Dato.java

```
@Override
public String toString() {
    return "Dato{" + "nombre=" + nombre + ", numero=" + numero + ", bool=" + bool + '}';
}
```

```
public class Dato {

    private String nombre;
    private int numero;
    private boolean bool;

    public Dato(String nombre, int numero, boolean bool) {
        this.nombre = nombre;
        this.numero = numero;
        this.bool = bool;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

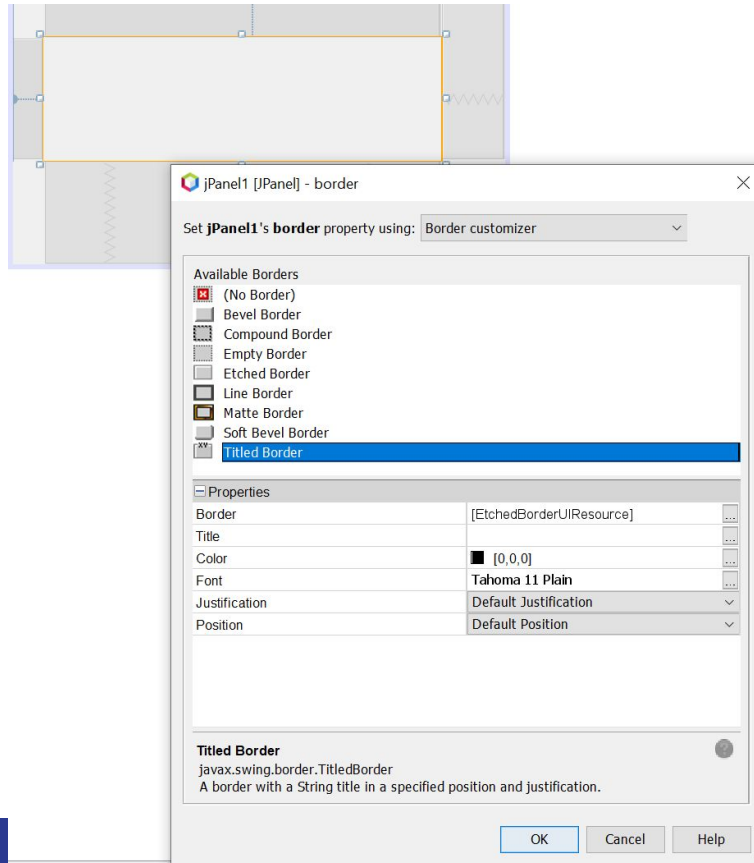
    public int getNumero() {
        return numero;
    }

    public void setNumero(int numero) {
        this.numero = numero;
    }

    public boolean isBool() {
        return bool;
    }

    public void setBool(boolean bool) {
        this.bool = bool;
    }
}
```

Añadimos JPanel en propiedad Border elegimos titulo



Añadimos 3 datos

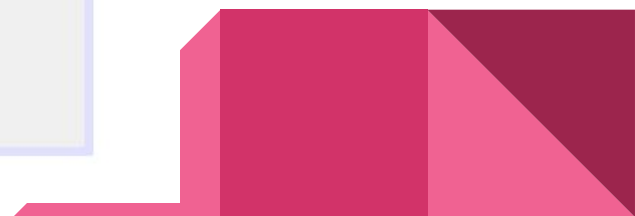
Item 1

Datos

Nombre:

Número:

Booleano:



Creamos un array de datos en el JFrame

```
] private void iniData(){  
    List<Dato> datos = new ArrayList<>();  
    for(int i =1; i<10; i++){  
        Dato dato = new Dato("Dato"+i, i, i%2==0);  
        datos.add(dato);  
    }  
    DefaultComboBoxModel model = new DefaultComboBoxModel(datos.toArray());  
    this.jComboBox1.setModel(model);  
- }
```

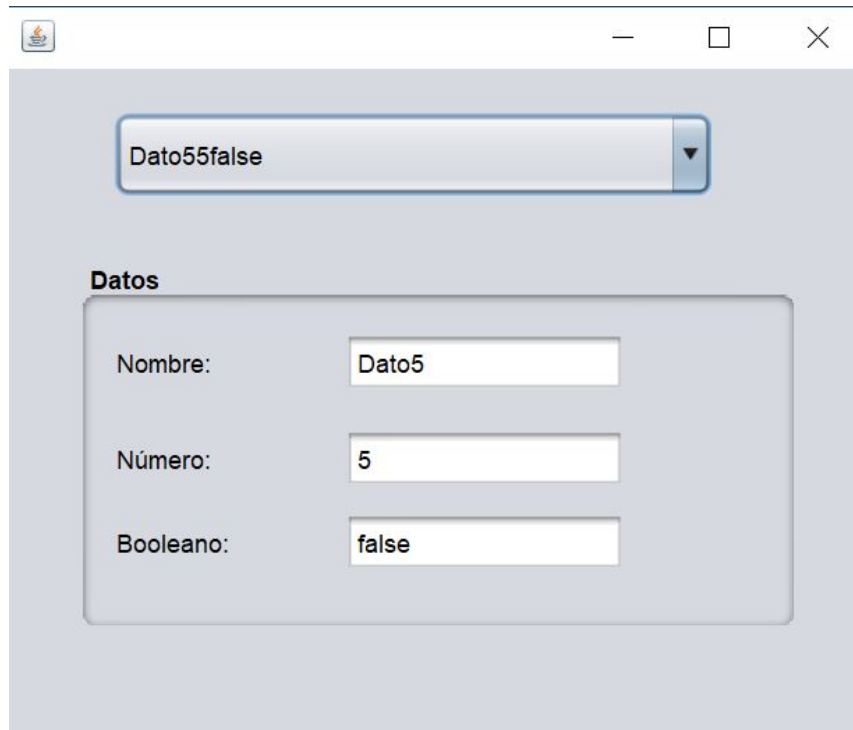
```
public BeanBind() {  
    initComponents();  
    iniData();  
}
```

Añadimos el método después de initComponents

```
2 4  
public BeanBind() {  
    initComponents();  
    iniData();  
}
```

Actualizamos los datos

```
private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    Dato dato= (Dato)this.jComboBox1.getSelectedItem();  
    this.jTextField1.setText(dato.getNombre());  
    this.jTextField2.setText(Integer.toString(dato.getNumero()));  
    this.jTextField3.setText(Boolean.toString(dato.isBool()));  
}
```



A screenshot of a Java Swing window. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main content area has a light gray background. At the top, there is a text field containing the text "Dato5false". Below this, there is a section titled "Datos" in bold black font. Inside this section, there are three rows of labels and text fields:

- Label: "Nombre:", Text field content: "Dato5"
- Label: "Número:", Text field content: "5"
- Label: "Booleano:", Text field content: "false"