

## EJERCICIO GUIADO. JAVA: SPINNER

### Introducción a los JSpinner

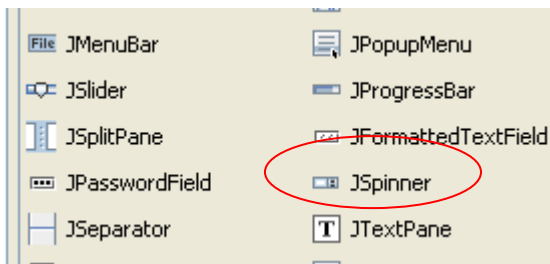
La clase JSpinner permite crear cuadros como el siguiente:



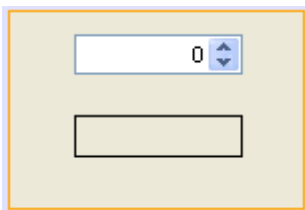
Son elementos muy comunes en los programas. A través de los dos botones triangulares se puede hacer que el valor del cuadro aumente o disminuya. También se puede escribir directamente un valor dentro del cuadro.

### Ejercicio guiado

1. Crea un nuevo proyecto.
2. Añade en él un JSpinner. Su nombre será *spiValor*.



3. Añade una etiqueta con borde. Su nombre será *etiValor*.
4. La ventana tendrá el siguiente aspecto:



5. Interesa que cuando cambie el JSpinner (ya sea porque se pulsaron los botones triangulares o porque se escribió dentro) aparezca el valor correspondiente dentro de la etiqueta. Para ello, tendrá que programar el evento *stateChanged* del JSpinner.

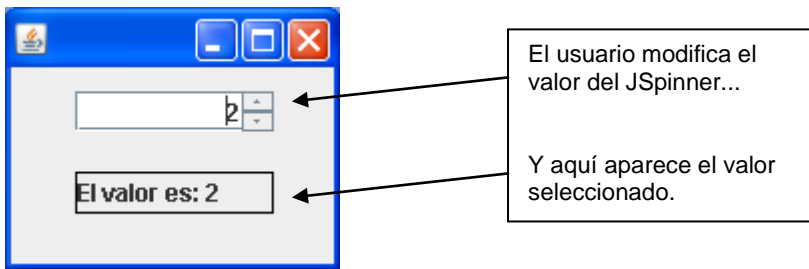
En el evento *stateChanged* introduzca el siguiente código:

```
etiValor.setText("El valor es: "+spiValor.getValue().toString());
```

6. Como puedes observar, lo que hace el programa es recoger el valor que tiene el JSpinner a través del método *getValue* y luego se lo asigna a la etiqueta con el clásico *setText*. (Es muy parecido a los deslizadores)

Debes tener en cuenta que el valor devuelto no es un número ni una cadena, así que en el ejemplo se ha usado el método *toString()* para convertirlo a una cadena.

7. Prueba el programa y observa su funcionamiento:



8. Observa como los valores del JSpinner aumentan o disminuyen en 1. Por otro lado, no parece haber un límite para los valores del JSpinner.

La pregunta ahora es: ¿Se puede modificar el contenido del JSpinner de forma que tenga unos valores concretos? La respuesta es sí. Veamos como hacerlo.

9. Entra dentro del código del programa y, dentro del constructor, añade este código debajo de  *initComponents*:

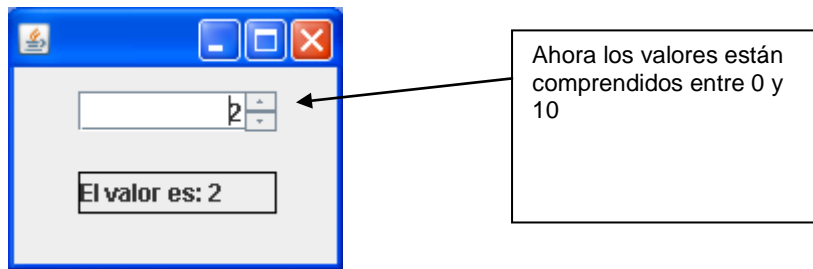
```
SpinnerNumberModel nm = new SpinnerNumberModel();  
nm.setMaximum(10);  
nm.setMinimum(0);  
spiValor.setModel(nm);
```

10. Este código hace lo siguiente:

- a. El JSpinner, al igual que los JList y los JComboBox, es un objeto que contiene otro objeto “modelo”, y es el objeto “modelo” el que contiene los números visualizados en el JSpinner.
- b. En el código anterior se crea un “modelo” para el JSpinner, se definen los valores que contendrá, y luego se asigna al JSpinner. Estudiemos las líneas del código.
- c. La primera línea crea un “modelo” llamado *nm*. Los modelos de los JSpinner son del tipo *SpinnerNumberModel*. Necesitarás incluir el import correspondiente (atento a la bombilla)

- d. En la segunda línea se define como valor máximo del modelo el 10, a través de un método llamado *setMaximum*.
- e. En la tercera línea se define como valor mínimo del modelo el 0, a través de un método llamado *setMinimum*.
- f. Finalmente se asigna el modelo creado al JSpinner.
- g. Este código, en definitiva, hará que el JSpinner muestre los valores comprendidos entre 0 y 10.

11. Prueba el programa y observa los valores que puede tomar el JSpinner.



12. Vamos a añadir otra mejora. Cambie el código del constructor por este otro. (Observa que solo se ha añadido una línea):

```
SpinnerNumberModel nm = new SpinnerNumberModel();  
nm.setMaximum(10);  
nm.setMinimum(0);  
nm.setStepSize(2);  
spiValor.setModel(nm);
```

13. La línea añadida es:

```
nm.setStepSize(2);
```

Esta línea usa un método del modelo del JSpinner que permite definir el valor de cambio del JSpinner. Dicho de otra forma, esta línea hace que los valores del JSpinner salten de 2 en 2.

14. Ejecuta el programa de nuevo y observa como cambian los valores del JSpinner.

15. El modelo del JSpinner tiene también un método llamado *setValue* que permite asignar un valor inicial al modelo. Pruebe a usar este método para hacer que el JSpinner muestre desde el principio el valor 4.

## **CONCLUSIÓN**

**Los JSpinners son objetos que permiten seleccionar un número, ya sea escribiéndolo en el recuadro, o bien a través de dos botones triangulares que permiten aumentar o disminuir el valor actual.**

**Los JSpinners son objetos con “modelo”. Es decir, este objeto contiene a su vez otro objeto “modelo” que es el que realmente contiene los datos.**

**Datos → Modelo → JSpinner**

**Para definir el contenido del JSpinner es necesario crear un modelo del tipo SpinnerNumberModel. Se le asigna al modelo los números deseados, y finalmente se une el modelo con el JSpinner.**

**El objeto modelo del JSpinner permite definir el valor mínimo y el valor máximo, así como el intervalo de aumento de los valores.**