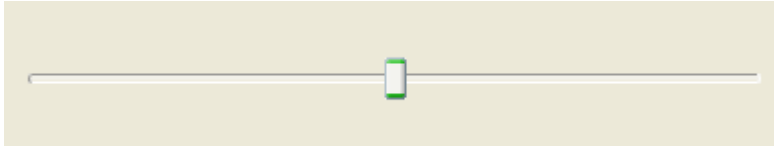


EJERCICIO GUIADO. JAVA: SLIDERS

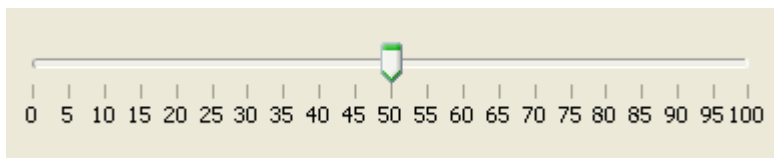
Introducción a los JSliders

La clase JSlider permite crear objetos como el siguiente:

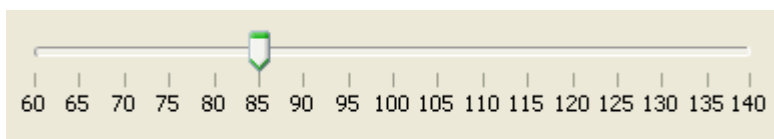


Estos elementos tienen un pequeño recuadro que se puede arrastrar a derecha o izquierda. Según la posición del recuadro, el JSlider tendrá un valor concreto.

El JSlider se puede configurar para que muestre los distintos valores que puede tomar:



También se puede configurar de forma que los valores mínimo y máximo sean distintos:

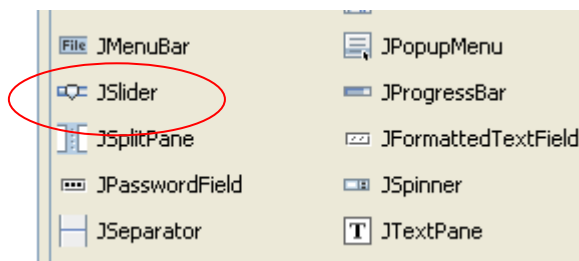


El valor que tiene un JSlider es el valor al que apunta el recuadro del JSlider. En la imagen anterior, el JSlider tiene un valor de 85.

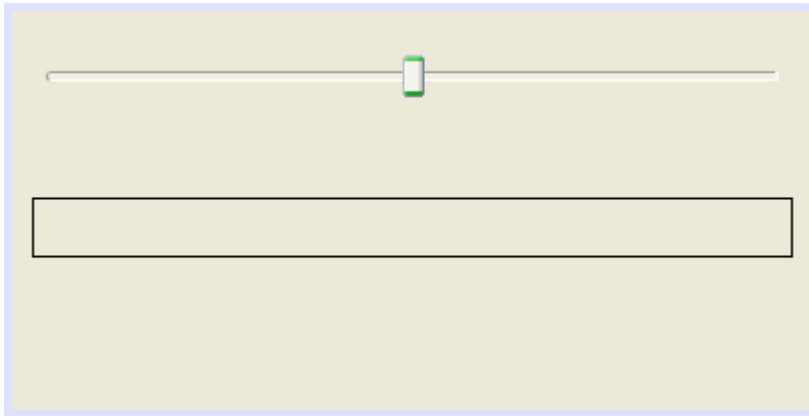
Se verá a continuación las características más interesantes de los JSlider y como programarlos.

Ejercicio guiado

1. Crea un nuevo proyecto.
2. Añade en él un JSlider. Su nombre será *slDeslizador*.



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



5. Un JSlider tiene un valor mínimo y un valor máximo. El valor mínimo es el valor que tiene cuando el recuadro está pegado a la parte izquierda, mientras que el valor máximo es el valor que tiene cuando el recuadro está pegado a la parte derecha.

El valor mínimo y máximo del JSlider se puede cambiar. Busca las propiedades *maximum* y *minimum* del JSlider y asigna los siguientes valores:

Máximo: 500


Mínimo: 100

| | |
|---|-------------------------------------|
| font | Tahoma 11 Pla |
| foreground | <input type="checkbox"/> [236,233,2 |
| majorTickSpacing | 0 |
| maximum | 500 |
| minimum | 100 |
| minorTickSpacing | 0 |
| minorTickSpacing | |
| (int) Sets the number of values between mir | |

6. Se puede asignar un valor inicial al JSlider a través de su propiedad *value*. Busque esta propiedad y asigne un valor de 400. Observe donde se sitúa el recuadro del JSlider.

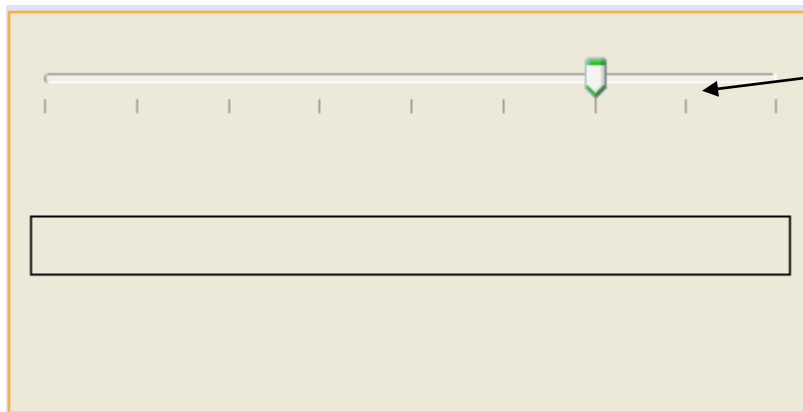
| | |
|------------------|--------------------------|
| snapToTicks | <input type="checkbox"/> |
| toolTipText | null |
| value | 400 |
| Other Properties | |
| UIClassID | Other propertie |

7. Se puede mejorar el JSlider definiendo unas divisiones (medidas) Por ejemplo, haremos que cada 50 unidades aparezca una división. Para ello use la propiedad *majorTickSpacing* y asigne un 50.

| | |
|-------------------------|---|
| foreground |  [236,233, |
| majorTickSpacing | 50 |
| maximum | 500 |

8. Esto, en realidad, no produce ningún cambio en el JSlider. Para que las divisiones se vean, es necesario que active también la propiedad *paintTicks*. Esta propiedad pintará divisiones en el JSlider:

| | |
|-------------------|-------------------------------------|
| paintLabels | <input type="checkbox"/> |
| paintTicks | <input checked="" type="checkbox"/> |
| paintTrack | <input checked="" type="checkbox"/> |
| snapToTicks | <input type="checkbox"/> |

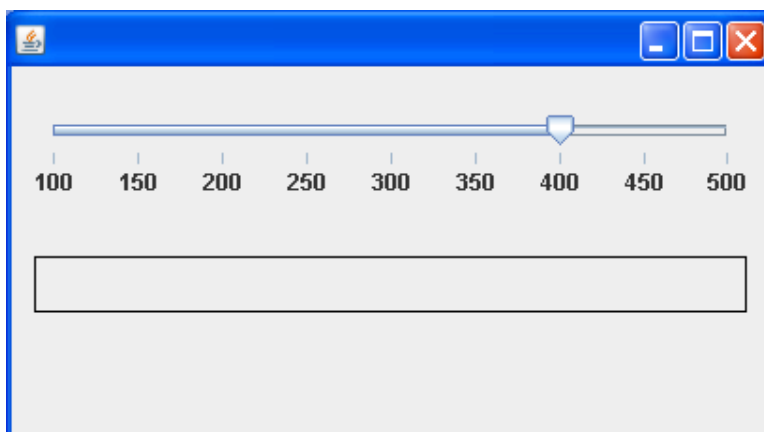


Medidas cada 50 unidades

9. Aún se puede mejorar la presentación del JSlider, si hacemos que aparezca el valor de cada división. Para ello debes activar la propiedad *paintLabel*.

| | |
|--------------------|-------------------------------------|
| orientation | HORIZONTAL |
| paintLabels | <input checked="" type="checkbox"/> |
| paintTicks | <input checked="" type="checkbox"/> |
| paintTrack | <input checked="" type="checkbox"/> |

10. Ejecuta el programa para ver el funcionamiento del Deslizador y su aspecto. Debe ser parecido al siguiente:



11. Bien. Ahora se pretende que cuando el usuario arrastre el deslizador, en la etiqueta aparezca el valor correspondiente. Para ello tendrá que programar el evento *stateChanged* del JSlider.

El evento *stateChanged* sucede cuando el usuario arrastra el recuadro del deslizador.

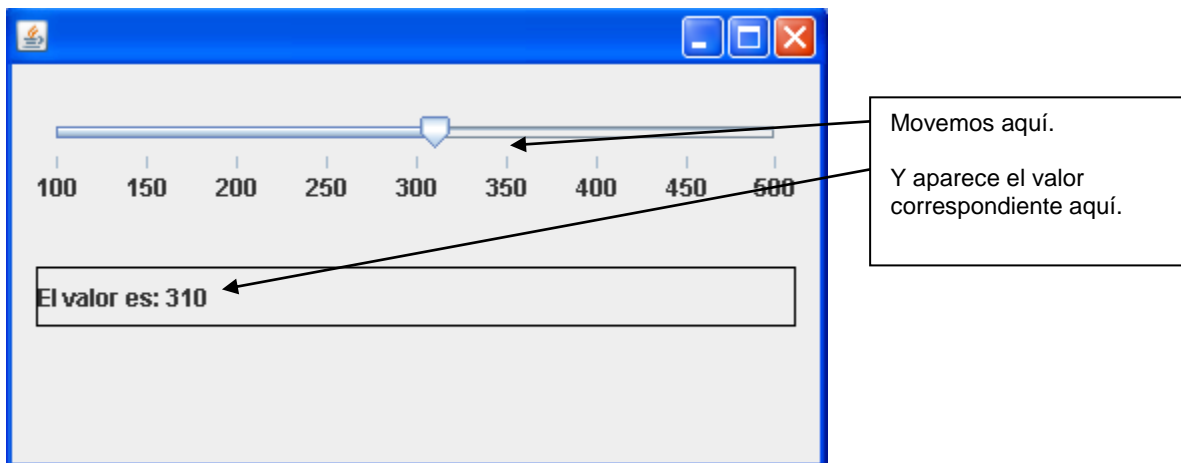
En este evento programe lo siguiente:

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

12. Ejecute el programa y observe lo que sucede cuando arrastra el deslizador.

13. La explicación del código es la siguiente:

- El método *getValue* del deslizador nos devuelve el valor que tiene actualmente el deslizador.
- Este valor es concatenado a la cadena "El valor es:" y es mostrado en la etiqueta a través del conocido *setText*.



14. A continuación se mencionan otras propiedades interesantes de los JSlider que puedes probar por tu cuenta:

orientation

Permite cambiar la orientación del JSlider. Podrías por ejemplo hacer que el JSlider estuviera en vertical.

minorTickSpacing

Permite asignar subdivisiones a las divisiones ya asignadas. Prueba por ejemplo a asignar un 10 a esta propiedad y ejecuta el programa. Observa las divisiones del JSlider.

snapToTicks

Cuando esta propiedad está activada, no podrás colocar el deslizador entre dos divisiones. Es decir, el deslizador siempre estará situado sobre una de las divisiones. Prueba a activarla.

paintTrack

Esta propiedad permite pintar o no la línea sobre la que se desliza el JSlider. Prueba a desactivarla.

CONCLUSIÓN

Los JSliders son objetos “deslizadores”. Permiten elegir un valor arrastrando un pequeño recuadro de derecha a izquierda o viceversa.

El valor de un JSliders puede ser obtenido a través de su método *getValue*.

Si quieres programar el cambio (el arrastre) en el deslizador, tienes que programar el evento llamado *stateChanged*.