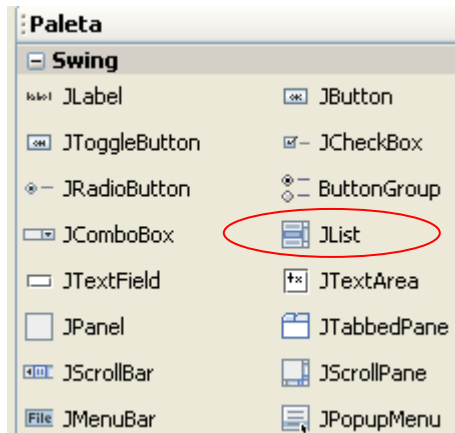
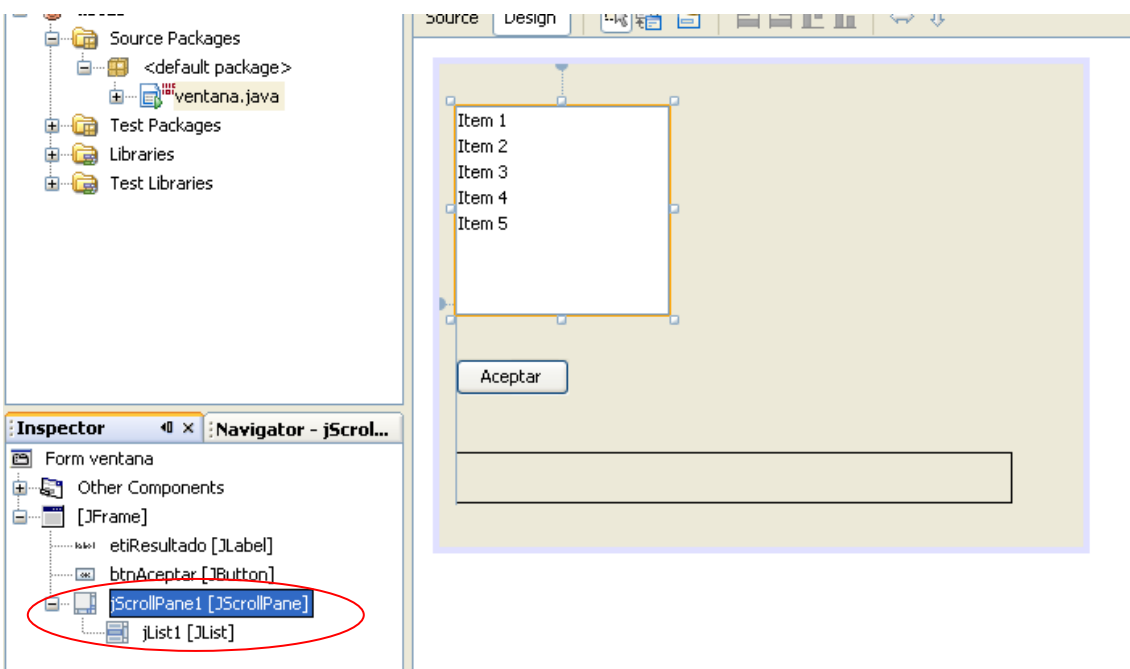


EJERCICIO GUIADO. JAVA: CUADROS DE LISTA

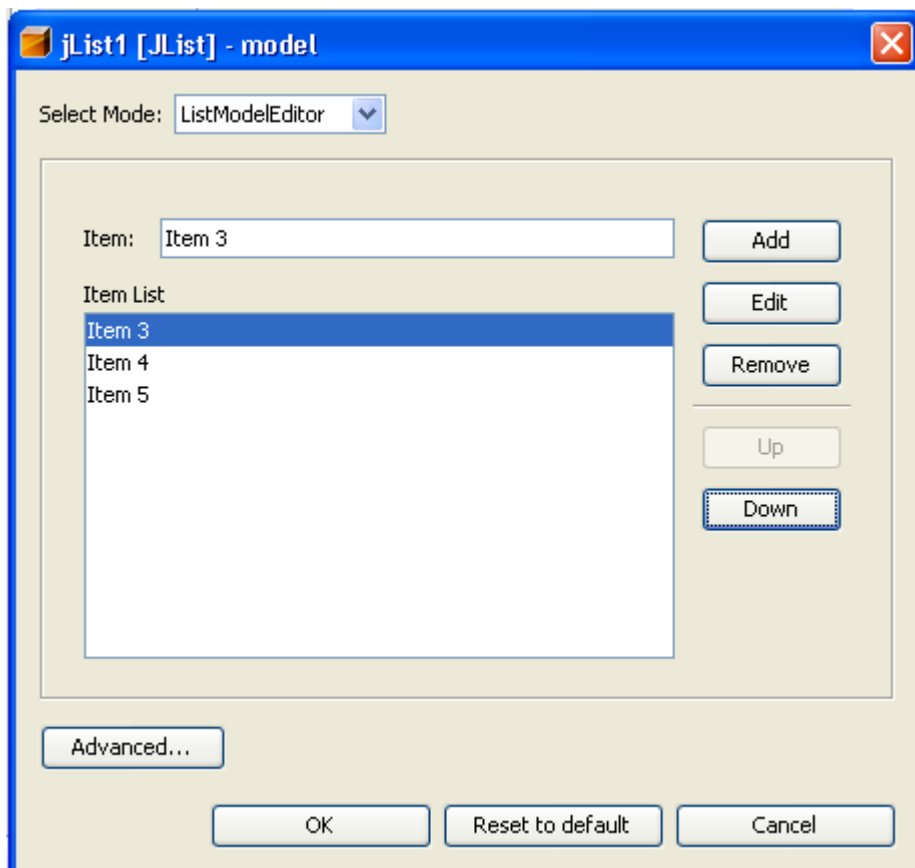
1. Realiza un nuevo proyecto.
2. En la ventana principal debes añadir lo siguiente:
 - a. Un botón “Aceptar” llamado btnAceptar.
 - b. Una etiqueta con borde llamada etiResultado.
3. Añade un cuadro de lista. Los cuadros de listas son objetos JList.



4. Cámbiale el nombre al JList. Ten cuidado, ya que en los JList aparecen siempre dentro de otro objeto llamado JScrollPane. Si miras en el Inspector, verás que al pulsar en el botón + del JScrollPane aparecerá tu JList:



5. Aprovecha para cambiarle el nombre al JList. El nuevo nombre será IstColores.
6. Si te fijas en el JList, consiste en un cuadro que contiene una serie de Items. Estos elementos pueden ser cambiados a través de la propiedad Model del JList.
7. Busca la propiedad Model y haz clic en el botón de los tres puntos. Aparecerá un cuadro de diálogo parecido al siguiente. Solo tienes que seleccionar los elementos que quieras y pulsar el botón “Borrar” (Remove) para eliminarlos de la lista.
8. Puedes añadir elementos escribiéndolos en el cuadro Artículo y luego pulsando el botón “Añadir” (Add).



9. Debes hacer que la lista sea la siguiente:

Rojo
Verde
Azul

10. Ahora programaremos el *actionPerformed* del botón Aceptar. Debes introducir el siguiente código:

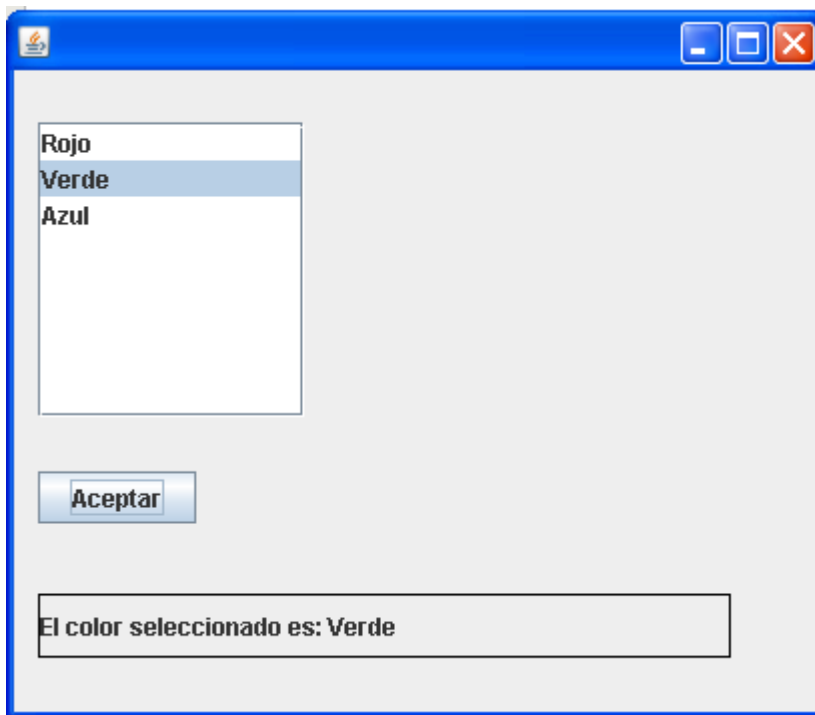
```
String mensaje;  
  
mensaje="El color seleccionado es: "+lstColores.getSelectedValue().toString();  
etiResultado.setText(mensaje);
```

11. Observa el código:

- Se crea una variable de cadena llamada *mensaje*.
- Y dentro de esta variable se introduce una concatenación de cadenas.
- Observa la parte: `lstColores.getSelectedValue()`, esta parte devuelve el valor seleccionado de la lista.
- Hay que tener en cuenta que este valor no es una cadena, por eso hay que convertirla a cadena añadiendo `.toString()`.
- De esta manera puedes extraer el elemento seleccionado de un cuadro de lista.
- Luego simplemente ponemos la cadena *mensaje* dentro de la etiqueta.

12. ~~Ejecuta~~ el programa y observa su funcionamiento. Por ejemplo, si seleccionas el color verde y pulsas aceptar el resultado será el siguiente:





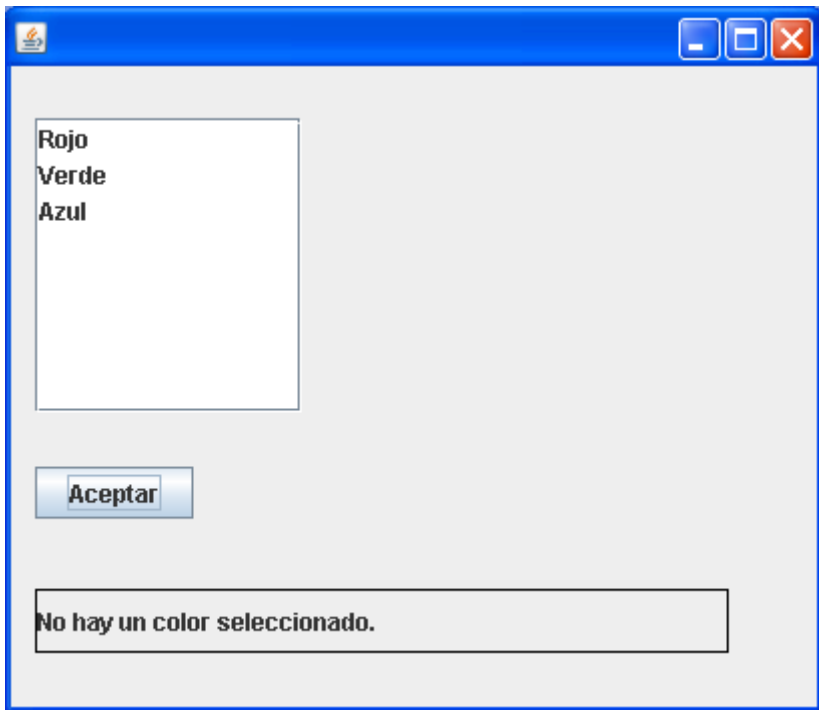
13. Vamos a mejorar el programa. Puede suceder que el usuario no seleccione ningún valor del cuadro de lista, y sería interesante en este caso que el programa avisara de ello. Cambie el código del botón Aceptar por este otro código:

```
String mensaje;  
  
if (lstColores.getSelectedIndex()==-1) {  
    mensaje="No hay un color seleccionado.";  
} else {  
    mensaje="El color seleccionado es: "+lstColores.getSelectedValue().toString();  
}  
etiResultado.setText(mensaje);
```

14. Observa el código:

- El método `getSelectedIndex` me dice el índice del elemento que está seleccionado.
- Por ejemplo, si está seleccionado el primero el índice es 0, si está seleccionado el segundo el índice es 1, etc.
- Si este método devuelve -1, entonces es señal de que no hay ningún elemento seleccionado.
- Aprovecho esto para mostrar un mensaje indicando lo sucedido.

15. Si ejecuta el programa y pulsa el botón Aceptar sin seleccionar nada el resultado debería ser el siguiente:



16. Se podría haber prescindido del botón aceptar si el código anterior se hubiera puesto en el evento `MouseClicked` del cuadro de lista en vez de en el *`actionPerformed`* del botón Aceptar. En este caso, cada vez que se seleccionara un elemento de la lista, automáticamente aparecería el mensaje en la etiqueta.

Se anima a que realice esta modificación.

CONCLUSIÓN

El objeto `JList` permite crear cuadros de lista. Estos objetos contienen una serie de elementos que pueden ser seleccionados.

A través del método `getSelectedValue` se puede obtener el elemento que está seleccionado. (Recuerda convertirlo a cadena con `toString`)

A través del método `getSelectedIndex` se puede saber la posición del elemento seleccionado. Si este índice es `-1`, entonces sabremos que no hay ningún elemento seleccionado.