

TEMA 5 – COMPONENTES VISUALES – EJERCICIO 3



Tener en cuenta que este ejercicio es esencial realizarlo para aprender a definir y usar componentes personalizados y establecer bindings.

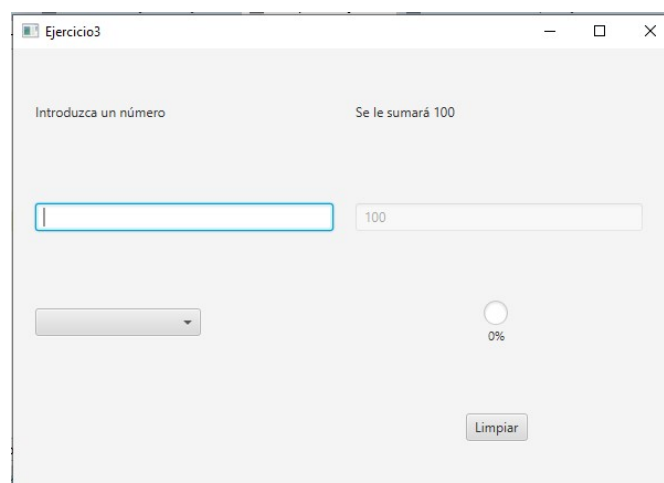
1. Personalización de campo de texto:

Debes realizar la personalización de un componente de tipo TextField cumpliendo los siguientes requisitos:

- a) Deberá tener una nueva propiedad que permitirán configurar el tipo de moneda.
- b) En el método setMoneda, se recibirá un String y:
 - i) Si tiene el valor "euro" la nueva propiedad almacenará el texto "€" y el cuadro de texto permanecerá visible.
 - ii) Si tiene el valor "dolar" la nueva propiedad almacenará el texto "\$" y el cuadro de texto permanecerá visible.
 - iii) Si llega cualquier otro String por parámetro, la propiedad no asignará ningún texto y además el campo de texto no se mostrará.
- c) Además, el TextField nunca será editable.

2. Uso del componente personalizado en nueva ventana de aplicación y bindings:

Una vez personalizado el componente, haremos uso de él (importando el jar) en una nueva ventana que debe tener el aspecto que se muestra en la imagen:

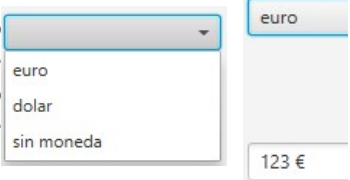


- a. La ventana organizará los elementos en un GridPane.
- b. Los dos primeros componentes son etiquetas con un texto que no cambiará.

- c. Después aparecen dos campos de texto cuyo comportamiento es el siguiente: el segundo cuadro de texto es igual al primer texto (se supone que será siempre numérico) más cien. Si el primer texto está vacío, se procesará como un 0. Este comportamiento se definirá usando bindings y listeners. El primer cuadro de texto es editable y el segundo no lo es. Más información sobre bindings al final.



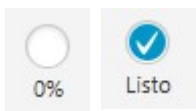

- d. El desplegable mostrará los textos: euro, dólar, sin moneda. Al escoger uno de estos valores, se deberá emplear el método setMoneda del componente personalizado. En la imagen anterior el componente no es visible porque no hay un tipo de moneda seleccionada. Si se ha seleccionado euro o dólar, se mostrará el cuadro de texto, sino permanecerá oculto.



- d. El cuadro de texto customizado, además de mostrar el símbolo de tipo de moneda seleccionado, mostrará la cantidad del primer cuadro de texto justo delante de la moneda y separada por un espacio. Este comportamiento se definirá con un listener y binding. Más información al final.



- e. El sexto elemento es un ProgressIndicator, estará al 0% cuando no haya nada de texto en el primero cuadro de texto de la ventana y pasará al estado “Listo” cuando haya al menos un carácter en el primer cuadro de texto de la ventana. Este comportamiento se definirá usando createIntegerBinding que enlazará ProgressProperty con el valor de texto del TextField.



- f. El botón “Limpiar” borra el contenido del primer cuadro de texto y el del desplegable, con su correspondiente efecto en los otros controles por la acción de los bindings creados.



3. Para definir los **bindings** que van a rellenar los valores de los cuadros de texto (el que suma 100 y el del tipo de moneda), una de las maneras más fáciles es definir en el controlador dos atributos de clase que de tipo property:

```
private IntegerProperty cantidadProperty;
private StringProperty textoMonedaProperty;
```

Estas propiedades son las que se van a enlazar a través del binding con la propiedad de texto de los cuadros, por ejemplo:

```
cuadroDeTexto.textProperty().bind(cantidadProperty.asString());
```

Estos atributos se actualizarán con el valor del cuadro de texto y del desplegable respectivamente. Por ejemplo:

```
cantidadProperty = new SimpleIntegerProperty(cantidad);
```

Donde cantidad es un entero que contiene el valor con la suma.

Ejemplos de comportamiento:

Introduzca un número Se le sumará 100

123 223

dolar

✓
Listo

123 \$ Limpiar

Introduzca un número Se le sumará 100

100

euro

○
0%

€ Limpiar

Introduzca un número Se le sumará 100

123 223

sin moneda

✓
Listo

Limpiar