
TEMA 5 – COMPONENTES VISUALES – EJERCICIO 2



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

1. Personalización de área de texto:

Debes realizar la personalización de un componente de tipo TextArea, siguiendo el mismo procedimiento mostrado en los ejemplos anteriores, y cumpliendo los siguientes requisitos:

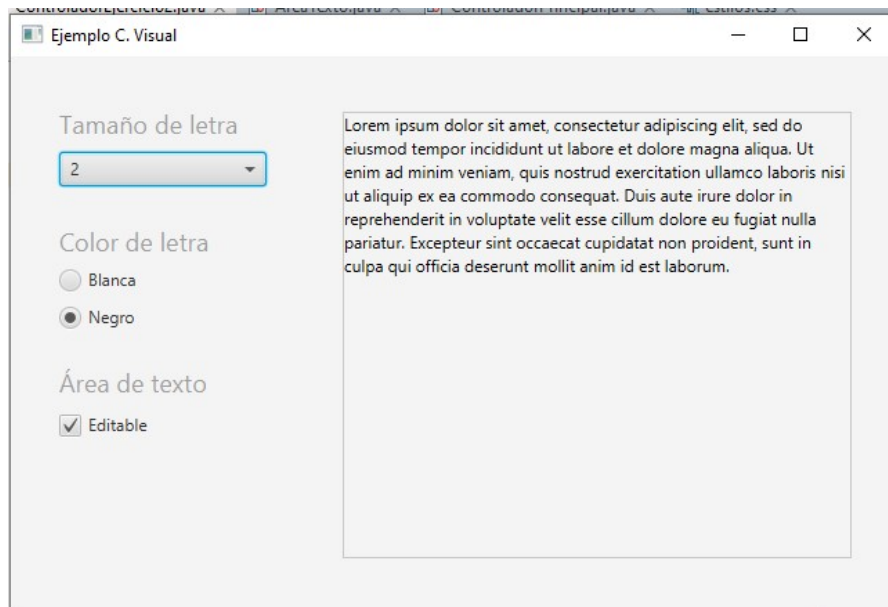
- Deberá tener dos nuevas propiedades que permitirán configurar:
 - a) El **tamaño** de la fuente (propiedad numérica llamada talla), cuyos valores podrán ser:
 - i) 1: (Pequeña) tamaño 10.
 - ii) 2: (Mediana) tamaño 12.
 - iii) 3: (Grande) tamaño 18.
 - b) El tamaño por defecto será el mediano.
 - c) El **color** de la letra (propiedad booleana para indicar si el color de letra es negro). El color por defecto será negro.
- El componente tendrá la propiedad Wrap Text siempre activa.
- Para simplificar el ejemplo y centrarnos en lo que trabajamos este tema, se permite prescindir de archivo css. pero lo ideal sería hacer uso de él.
- El texto por defecto que se mostrará en el área será "Lorem Ipsum", puedes obtenerlo en: <https://es.lipsum.com/feed/html>

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

Una vez personalizado el componente, haremos uso de él (importándolo como vimos en los ejemplos anteriores) en una nueva ventana que debe tener el aspecto que se muestra en la imagen (más adelante) y que permitirá:

- a. Modificar el tamaño de la fuente a través de un ChoiceBox con los valores: pequeña, mediana, grande.
- b. Cambiar el color, que se escogerá mediante dos RadioButtons con los valores blanco y negro.
- c. Ambos componentes deben mostrar al abrir la ventana los valores por defecto de las propiedades del área de texto.

En función a los valores del desplegable y de los RadioButtons, se actualizará automáticamente el aspecto del área de texto (no es obligatorio el uso de Bindings en esta parte del ejercicio).



3. Bindings

- Se deben usar Bindings que definan los siguientes comportamientos:
 - a. Siempre que el CheckBox esté activo, se permitirá editar el texto que contiene el área. Cuando el componente esté inactivo, no se permitirá editar el texto. Usar `editableProperty` y `selectedProperty`.
 - b. Cuando el área de texto no contenga texto alguno, deberán deshabilitarse los RadioButton que permiten modificar el color de letra (seguirán siendo visibles pero no podrán usarse). Cuando haya texto, volverán a estar habilitados. Usar `disableProperty` y `textProperty` con `createBooleanBinding`.
 - c. Cuando el área de texto no contenga texto alguno, el ChoiceBox que modifica el tamaño de letra debe deshabilitarse (seguirá siendo visible pero no podrá usarse). Cuando haya texto, el componente deberá de volver a habilitarse. Usar `disableProperty` y `textProperty` con `createBooleanBinding`.

createBooleanBinding

```
public static BooleanBinding createBooleanBinding(Callable<Boolean> func,
                                                Observable... dependencies)
```

Helper function to create a custom BooleanBinding.

Parameters:

`func` - The function that calculates the value of this binding

`dependencies` - The dependencies of this binding

Returns:

The generated binding

Pista para createBooleanBinding:

- El primer parámetro es el que devuelve el valor true o false según corresponda (puedes usar aquí una función lambda).
- El segundo parámetro es la propiedad cuyo cambio dispara la ejecución (en este caso la propiedad correspondiente al texto del área).