

Unidad 9 Interfaces gráficas de usuario. JavaFX.

Calculadora simple

1.- Calculadora simple mejorada

2.- Calculadora Compleja

Estrategia para poder desarrollar la actividad

3.- No seas impaciente



Unión Europea

Fondo Social Europeo

El FSE invierte en tu futuro

Fecha	Versión	Descripción
20/04/2024	1.0.0	Versión inicial.

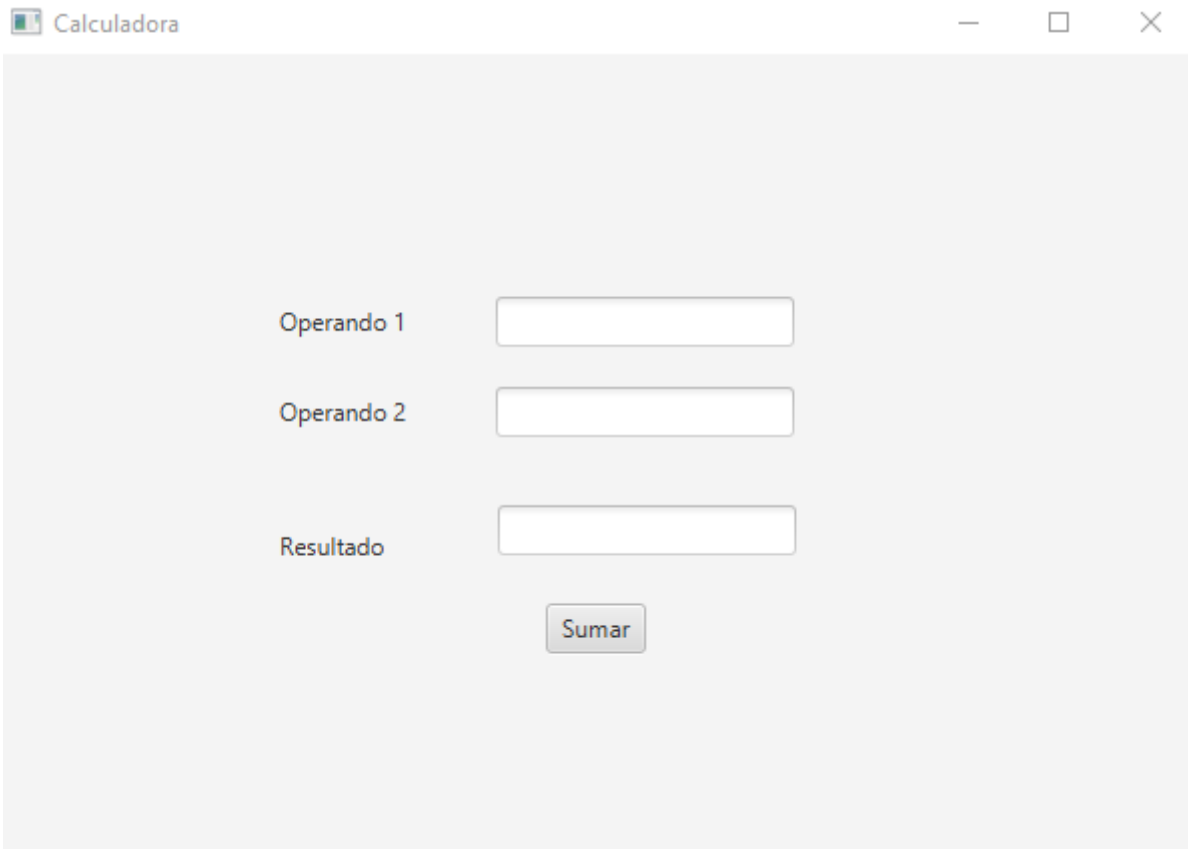
Unidad 9 Interfaces gráficas de usuario. JavaFX.

Antes de comenzar estos ejercicios deberás:

Disponer de [Scene Builder](#) instalado en tú máquina y configurado tú IDE, bien IntelliJ o NetBeans. Sigue las indicaciones de la documentación de la teoría.

Calculadora simple

Realiza una calculadora como esta:



Cuando el usuario escriba números (controlarlo con excepciones) y pulse en sumar, debe mostrar en el txtResultado el resultado final.

Crea la clase Suma, la cual deberá disponer de 2 atributos (para los 2 operandos), constructor, setters y getters, toString() así como la función que se encargará de realizar la operación de la suma.

Recuerda, necesitarás en el proyecto los siguientes elementos:

- Clase Suma: Contendrá los atributos necesarios, getters y setters así como la función sumar.
- Controlador.
- Clase aplicación.
- Vista en fichero fxml.

La operación para realizar la suma la realizaremos en el controlador.

Debemos tener en cuenta que se puede producir una excepción, por lo tanto, la operación de la suma la meteremos en un try ... catch.

El tipo de excepción que debemos capturar es **NumberFormatException**. Dentro del mismo utilizaremos un objeto de tipo Alert. Si, este no lo hemos visto, pero su funcionamiento es el siguiente:

```
Alert alert = new Alert(Alert.AlertType.WARNING);
alert.setHeaderText(null);
alert.setTitle("Error en la entrada de datos");
alert.setContentText("Error en el formato");
alert.showAndWait();
```

Creamos el objeto de tipo Alert seleccionando el tipo de Alerta.

Establecemos su cabecera a null

Indicamos el título

Mostramos la información del problema.

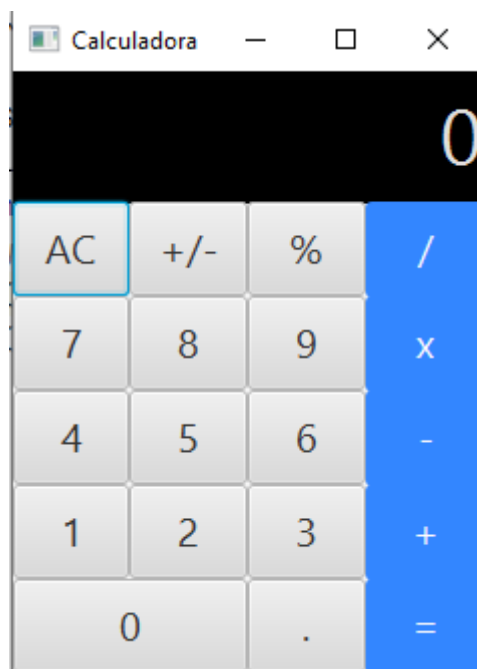
1.- Calculadora simple mejorada

Una vez hecha nuestra calculadora simple. Añade a esta 3 nuevas funcionalidades: Restar, Multiplicar y Dividir.

2.- Calculadora Compleja

Una vez hemos visto como implementar una calculadora simple, vamos a implementar una calculadora un poquito más compleja.

Para ello implementaremos una calculadora similar a esta:

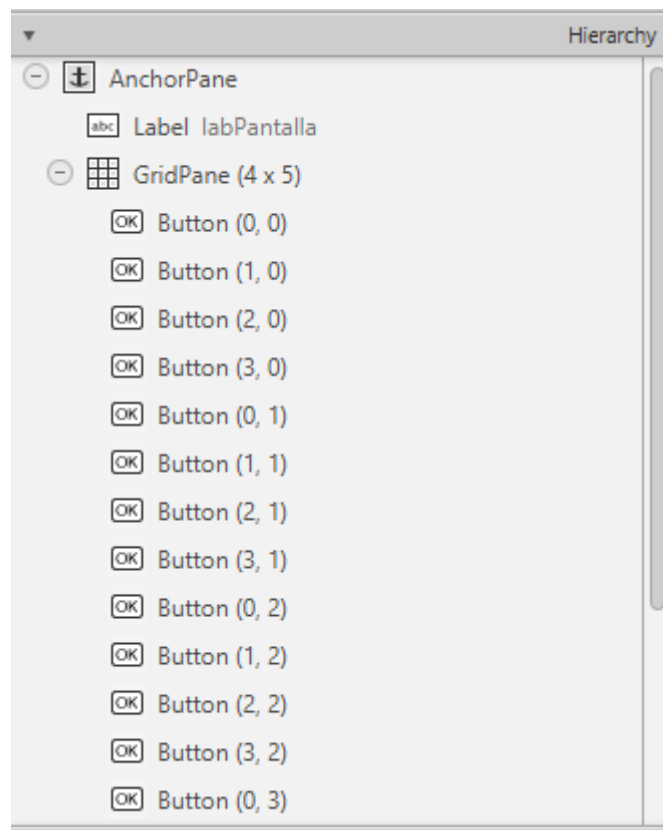


Las funciones que llevará a cabo son las siguientes:

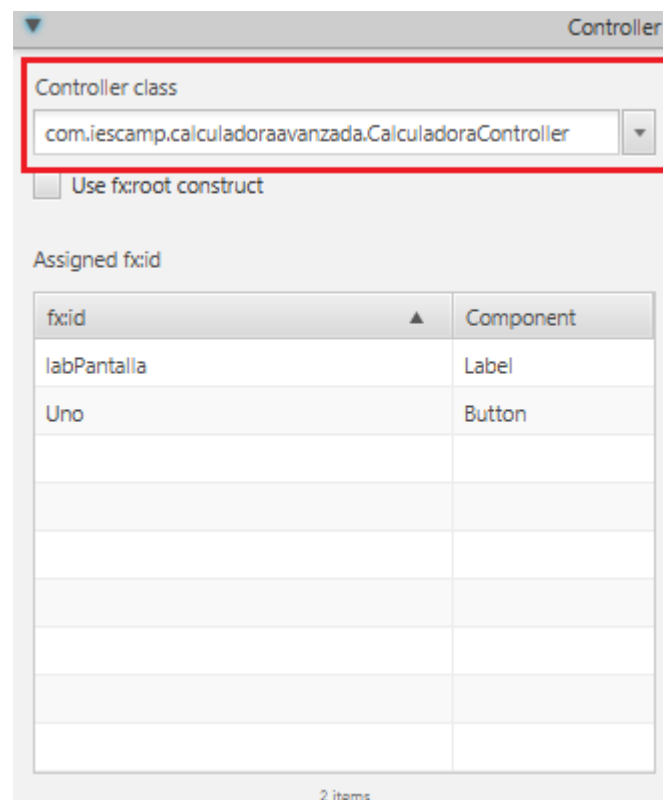
- AC: Realizará el borrado de pantalla.
- +/-: Añadirá al número el símbolo negativo.
- %: Calculará con una regla de 3 el porcentaje del número introducido.
- /: Realizará la división entre números (recuerda que en la calculadora se introducen también números enteros)
- x: Multiplicación.
- -: Resta.
- +: Suma.
- =: Nos permitirá obtener el resultado de la operación.

Para elaborar la misma deberemos crear con Scene Builder nuestro formulario. Llámalo **Calculadora.fxml** y contendrá los siguientes elementos:

1. Un AnchorPane que actuará de panel central de nuestra calculadora.
2. Una label, que actuará de pantalla.
3. Un GridPanel con 4 columnas y 5 filas (dentro del cual ubicaremos los botones)



Antes de dejar Scene Builder le asociaremos al formulario **Calculadora.fxml** el controlador que crearemos bajo:



Una vez tenemos el diseño de la pantalla, crearemos el controlador de la misma, al cual llamaremos **CalculadoraController**.

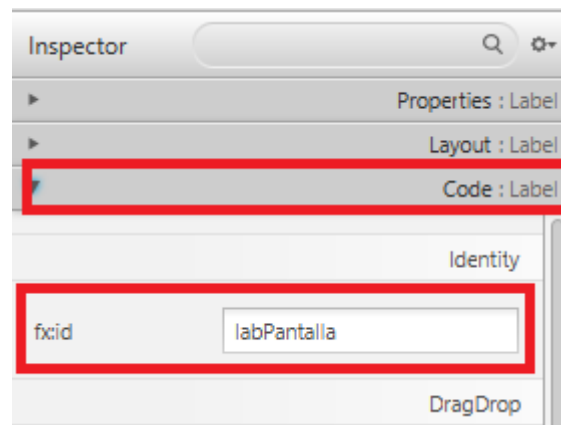
Las únicas variables (globales) que utilizaremos serán las siguientes:

```
@FXML
private Label labPantalla;

private boolean digito = false;
private boolean punto = false;
private int numOperandos=0;
private double operando1, operando2;
private char operador=' ';
```

labPantalla es el fix:id como he identificado yo a la label que actúa de pantalla. Si le llamáis de otra forma recordad cambiar esta.

Recordad que cada uno de los elementos con los que queremos interactuar o acceder desde el controlador, en la vista (**Calculadora.fxml**) deben tener un identificador:



Estrategia para poder desarrollar la actividad

1. Primero diseña la vista (Calculadora.fxml) (Recuerda asociarle el controlador)
2. Una vez hecho esto, prueba la visualización desde el Scene Builder para ver que el diseño de la vista es correcta.
3. Una vez está diseñada la vista, pasa a desarrollar el controlador. Si vamos a modificar el valor de algún elemento de la vista deberemos definir la respectiva variable en el controlador. Se realiza con una etiqueta superior también conocido como decorador (@FXML):

```
@FXML
private Label labPantalla;
```

4. Define una función que se encargue de mostrar los datos por pantalla, la cual recibirá una cadena, que es el dígito asociado a cada botón:

```
public void resultadoPantalla(String resultado)
```

En esta función deberás tener en cuenta que si no es un dígito (mira las variables que he dejado arriba) y el parámetro que se le pasa (resultado) es un "0" no deberá hacer nada.

Si es un dígito deberá añadirlo a lo existente.

Si no es un dígito, colocamos la pantalla sin nada y marcamos el punto como falso y en caso contrario es que es un dígito, por lo tanto, añadiremos lo que le pasamos en resultado a la pantalla y dígito lo colocaremos a true.

Por ejemplo:

```
- 12 +dígito 3 = 123 (añadimos el 3 al número)
- 0 +dígito . = 0.   (añadimos el decimal al número)
- 0.2 +dígito . = 0.2 (como ponemos el decimal y ya existe no hace nada)
```

5. Si nos fijamos, la forma de proceder para los botones de suma, resta, división, multiplicación e igual, son muy parecidos. Define una función que invoquemos cuando hagamos un click en un botón.

```
public void evaluarOperador(String operador)
```

En este caso de que sea un dígito, incrementará el valor del número de operandos, si es el número de operandos igual a 1, lo asignará a la variable operando1 y en el caso que el valor de número de operandos sea igual a 2, lo asignará a la variable operando2, para a continuación realizar la operación (switch) en función del operador que se le pase.

6. El resto de métodos os lo dejo a vuestra creación.

7. **ES MUY IMPORTANTE QUE TENGÁIS EN CUENTA SI ES UN DÍGITO, SI ES UN PUNTO, SI ES UN OPERADOR.**

3.- No seas impaciente
