

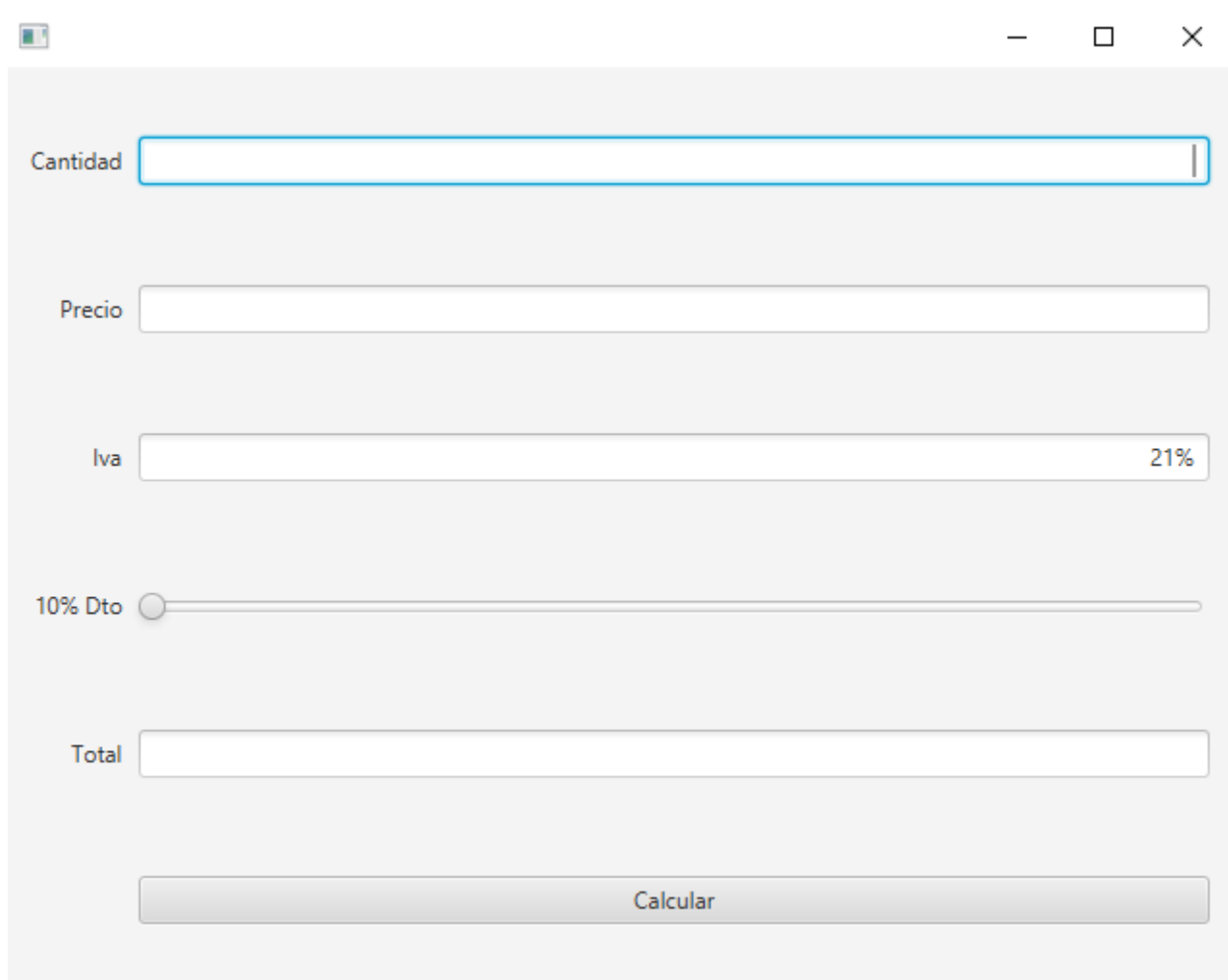
PRÁCTICA 2: INTERFAZ JAVAFX CON FUNCIONALIDAD

Objetivos:

- Iniciar al alumado en el desarrollo y manejo de interfaces gráficas con JavaFX
- Iniciar al alumnado en el manejo de la herramienta visual Scene Builder

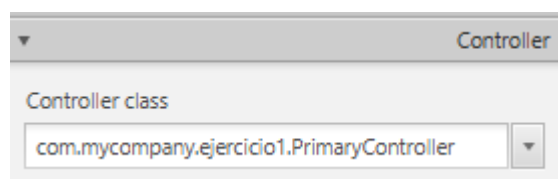
Secuencia y desarrollo:

1. Diseña la siguiente interfaz con JavaFX



2. Pasos necesarios para dar funcionalidad.

1. Debes tener vinculado la interfaz con el archivo controlador



2. Todos los componentes gráficos que necesiten ser modificados o que vayan a tener funcionalidad se le va a asignar un identificador, para ello debemos crear en el archivo Controller las siguientes variables

@FXML

Label etiquetaDto;

@FXML

TextField campoCantidad;

@FXML

TextField campoPrecio;

@FXML

TextField campoIva;

@FXML

TextField campoTotal;

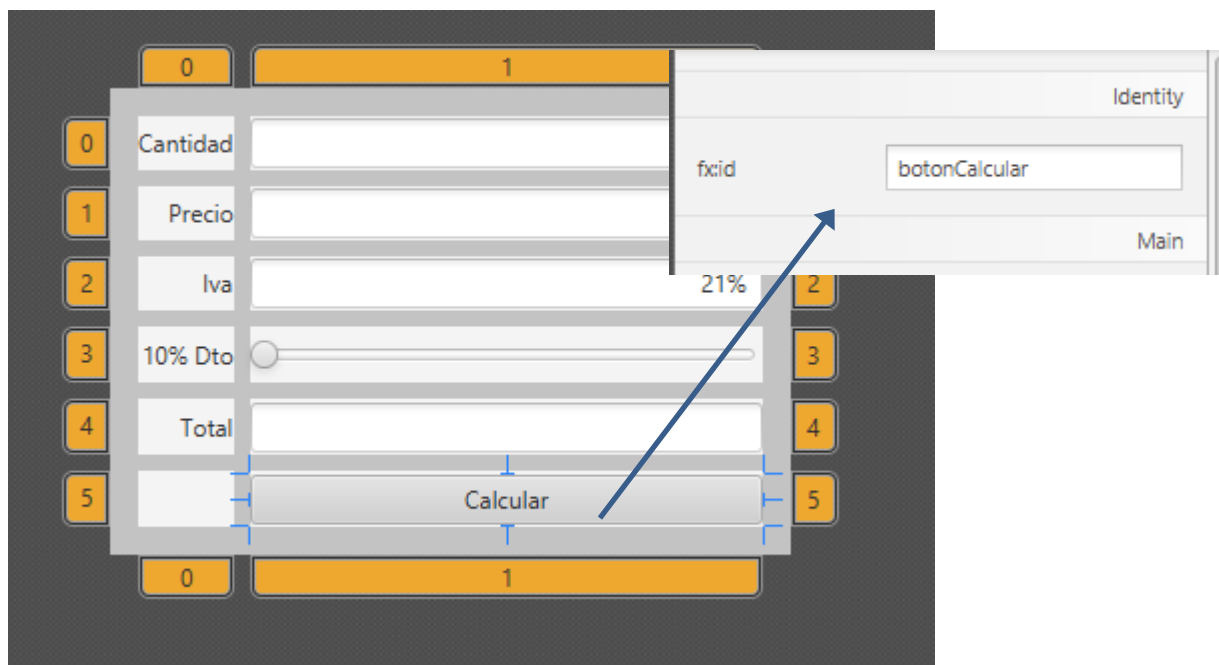
@FXML

Slider sliderDescuento;

@FXML

Button botonCalcular;

3. A continuación con el SceneBuilder vinculamos cada componentes con su identificador (Nota: Debes guardar los cambios que haces en cada fichero)



3. Por último hay que darle funcionalidad. Para ello debes declarar las funciones en el fichero Controller y asociar la funcionalidad con el evento específico

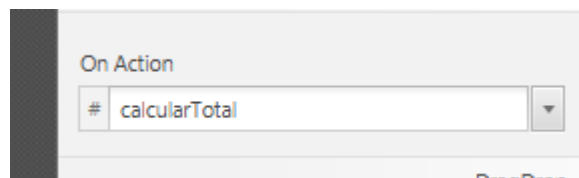
public class PrimaryController implements Initializable {

//Crear variables estáticas

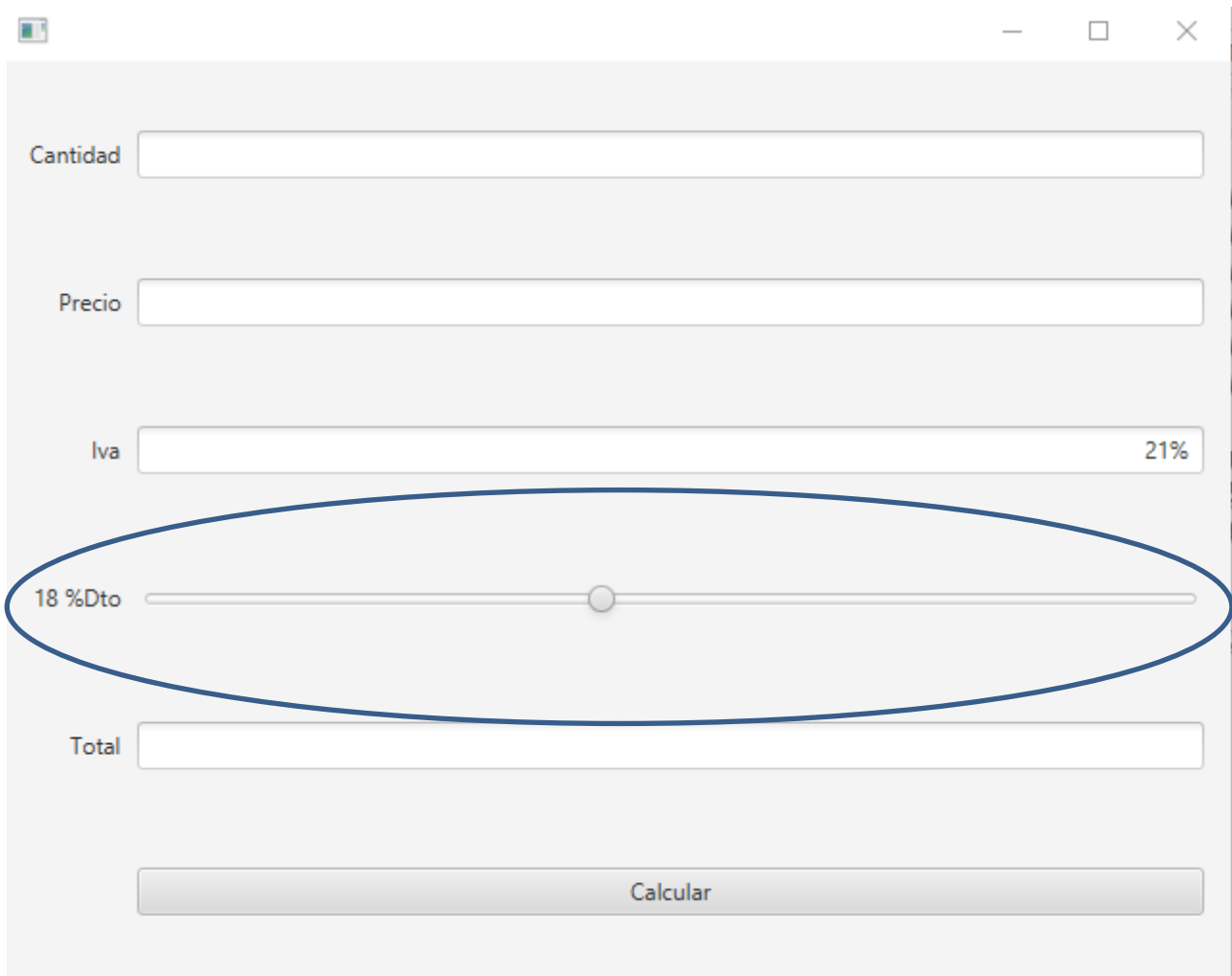
```
private static final NumberFormat moneda  
    = NumberFormat.getCurrencyInstance();  
private static final NumberFormat descuento  
    = NumberFormat.getPercentInstance();  
private BigDecimal descuentoCliente  
    = new BigDecimal(0.10);  
private BigDecimal porcentajelva  
    = new BigDecimal(0.21);
```

@FXML

```
private void calcularTotal(ActionEvent event) {  
    try {  
        BigDecimal cantidad = new BigDecimal(campoCantidad.getText());  
        BigDecimal precio = new BigDecimal(campoPrecio.getText());  
        BigDecimal importe = cantidad.multiply(precio);  
        BigDecimal descuento = precio.multiply(descuentoCliente);  
        BigDecimal iva = precio.multiply(porcentajelva);  
        BigDecimal total = (precio.subtract(descuento)).add(iva);  
  
        campoTotal.setText(moneda.format(total));  
    } catch (NumberFormatException e) {  
        campoCantidad.setText("Introduzca una cantidad correcta");  
    }  
    campoCantidad.selectAll();  
    campoCantidad.requestFocus();  
  
}
```



4. Por último debes dar funcionalidad al componente Slider, el cual debe modificar su valor al ser desplazado.



The screenshot shows a web form with the following elements:

- Cantidad**: A text input field.
- Precio**: A text input field.
- Iva**: A text input field with the value **21%** displayed on the right.
- 18 %Dto**: A slider control with a range from 0 to 100. The current value is 18%, indicated by a blue oval around the slider and the text "18 %Dto" on the left.
- Total**: A text input field.
- Calcular**: A button at the bottom of the form.