

# UT6-7 Introducción a JavaFX y Scene Builder

José Jarones

# Instalación SceneBuilder

Si queremos crear una ventana en Java a partir de XML se puede realizar escribiendo nosotros directamente el código xml, o bien utilizando un programa con interfaz gráfica como Scene Builder.

Para descargar el programa:

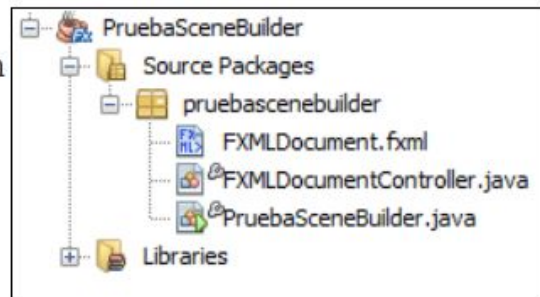
<https://gluonhq.com/products/scene-builder/#download>



# Archivo FXML y clase controladora

En netbeans, crearemos un nuevo proyecto de tipo JAVA FX y observamos que se crean los siguientes archivos:

- **FXML** (*FXMLDocument.fxml*)
  - Archivo XML que contiene el diseño con la estructura de los elementos que forman la ventana.
- **FXML Controller** (*FXMLDocumentController.java*)
  - Clase Java que se encargará de establecer las acciones que deben realizar los elementos que forman la ventana.
- **JavaFX Main Class** (*PruebaSceneBuilder.java*)
  - Clase Java que se encargará de mostrar el diseño realizado en el archivo FXML, y que permite la ejecución de la aplicación.



# Carga del FXML

En la clase principal se carga el archivo FXML

```
public void start(Stage stage) throws Exception {  
    Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));  
    Scene scene = new Scene(root);  
    stage.setScene(scene);  
    stage.show();  
}
```

De esa manera se está consiguiendo que todo el contenido del archivo FXML se cargue dentro de un contenedor genérico de tipo Pane de JavaFX (que se almacena en la variable root). Dicho contenedor Pane será el contenedor principal de los elementos gráficos de la aplicación, y por ello en la línea siguiente se le asigna a la escena (Scene) de la ventana de la aplicación.

# Archivo FXML

El archivo FXML contiene (en formato XML) la estructura de los elementos gráficos que formarán parte del interfaz de usuario.

```
<AnchorPane id="AnchorPane" prefHeight="200" prefWidth="320" xmlns:fx="http://javafx.com/fxml/1" fx:controller="pruebasценеbuilder.FXMLDocumentController">
    <children>
        <Button layoutX="126" layoutY="90" text="Click Me!" onAction="#handleButtonAction" fx:id="button" />
        <Label layoutX="126" layoutY="120" minHeight="16" minWidth="69" fx:id="label" />
    </children>
</AnchorPane>
```

De manera general observa que declara un contenedor principal de tipo **AnchorPane** (permite anclar los elementos que contenga a otros elementos), y dentro de él como hijos (**children**) aparecen un botón (**Button**) y una etiqueta de texto (**Label**).

Cada tipo de elemento de FXML tendrá sus propiedades que serán diferentes a las de otros elementos, pero **desde Scene Builder** podrás conocer de manera visual las posibles propiedades que podrás aplicar a cada elemento.

Otro aspecto **importante** a conocer del archivo FXML es la **referencia a la clase Java** que va a hacer las funciones de clase **controladora**. En este ejemplo puedes ver que es *FXMLDocumentController* como aparece en la **propiedad *fx:controller*** dentro de la etiqueta *AnchorPane*.

```
fx:controller="pruebascenebuilder.FXMLDocumentController"
```

# Clase controladora

Cada archivo FXML debe tener asociada una clase Java a la que se suele denominar "controladora". En esta clase se deben declarar las instrucciones Java que se deseen ejecutar cuando se inicialice el interfaz gráfico contenido en el archivo FXML, o también, por ejemplo, declarar el método que debe ejecutarse cuando el usuario haga clic en un botón, o el código necesario para cambiar un campo de texto, rellenar una lista de datos, etc.

El asistente de NetBeans que ha creado el proyecto genera una clase controladora con un código como el siguiente:

Observa que contiene declarado el **método *initialize***, aunque su contenido está vacío. Las instrucciones Java que indiques dentro de este método se ejecutarán cada vez que se cargue el interfaz gráfico durante la ejecución de la aplicación.

```
public class FXMLDocumentController implements Initializable {

    @FXML
    private Label label;

    @FXML
    private void handleButtonAction(ActionEvent event) {
        System.out.println("You clicked me!");
        label.setText("Hello World!");
    }

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }


}
```

# HandleButtonAction

El método `handleButtonAction` que posee la anotación `@FXML`. Esa anotación indica que el elemento que aparece a continuación de él está asociado a algún elemento del archivo FXML. En este caso, el método `handleButtonAction` aparecía asociado al botón, como puedes recordar en esta línea resumida:

```
<Button onAction="#handleButtonAction" />
```

De esta manera se consigue que **cuando el usuario haga clic en el botón** (*onAction*), se ejecuten las sentencias contenidas en el método *handleButtonAction*





# Variables


El identificador que se asigne a un elemento FXML se debe corresponder con el nombre de la variable que se declare en la clase controladora.

```
<Label fx:id="label" />
```

```
@FXML  
private Label label;
```

De esta manera, cualquier acción que se realice sobre el objeto al que hace referencia la variable tendrá efecto sobre el elemento FXML que tenga ese identificador.

```
label.setText("Hello World!");
```



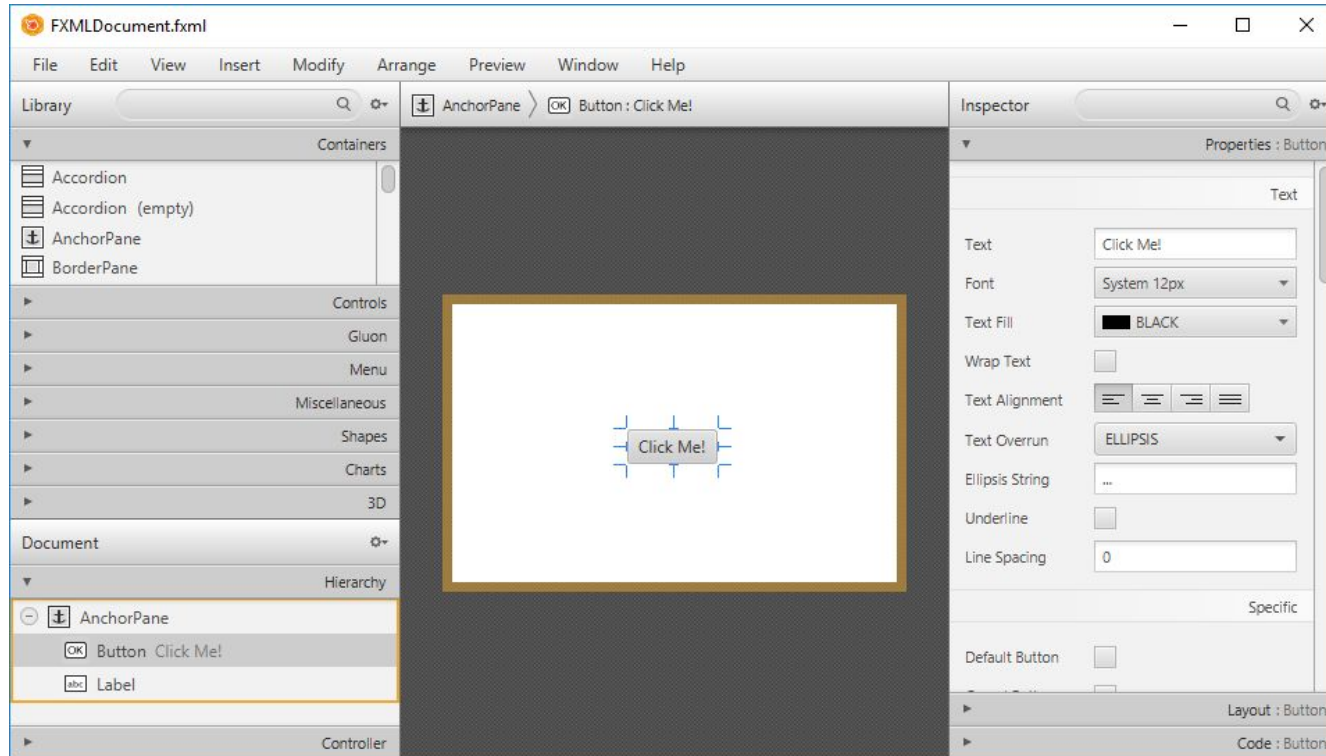
# Vista diseño en Scene Builder

Al hacer doble clic sobre fxml se abre SceneBuilder y podremos modificar el diseño.

Si queremos modificar como texto, pulsamos sobre el botón derecho sobre el archivo y hacemos clic en edit.

En caso de que no se abra Scene Builder directamente desde NetBeans al abrir un archivo FXML, **abre el menú *Tools > Options* de NetBeans** y accede a la **sección *Java*** y desde ahí a la **pestaña *JavaFX***. En la **lista desplegable *Scene Builder Home*** selecciona la **opción *Browse*** y localiza la carpeta donde se encuentre instalado *Scene Builder*.

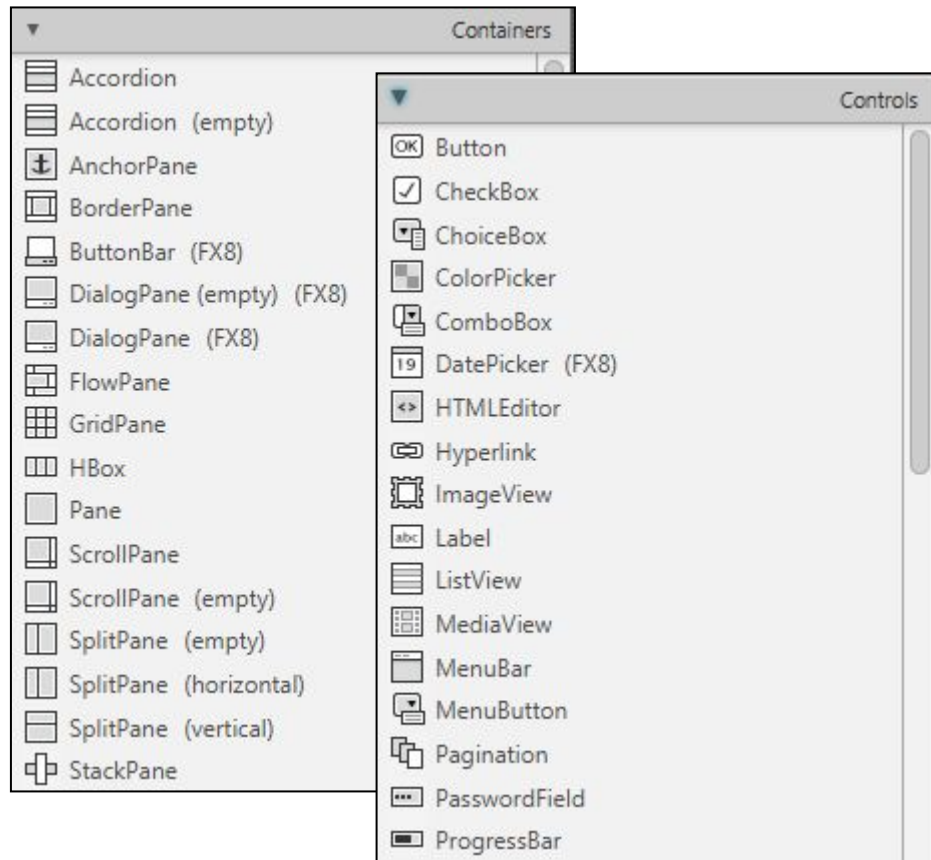
# SceneBuilder



# Sección Library

En la parte derecha de la ventana de Scene Builder puedes encontrar esta sección que engloba varios apartados como: Containers, Controls, Gluis, Menu, etc. Ahí puedes encontrar los elementos gráficos que puedes añadir al interfaz gráfico que estés diseñando.

Dentro de todos esos apartados los más importantes pueden ser el de Containers, donde se encuentran los distintos tipos de contenedores, como HBox, VBox, BorderPane, GridPane, ScrollPane, etc, que permiten organizar los elementos que se quieran añadir al diseño, y la sección Controls que contiene los elementos más comunes de la mayoría de los interfaces gráficos como: botones, etiquetas de texto, campos de texto, casillas de verificación, listas, etc.

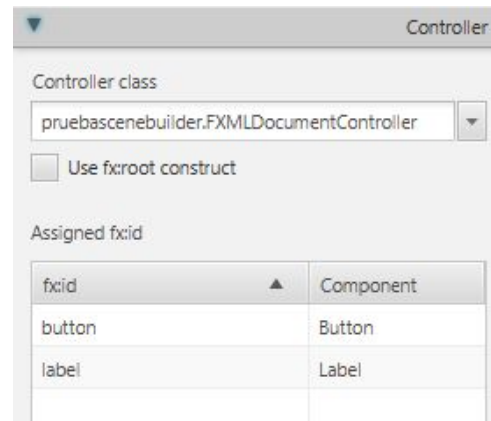
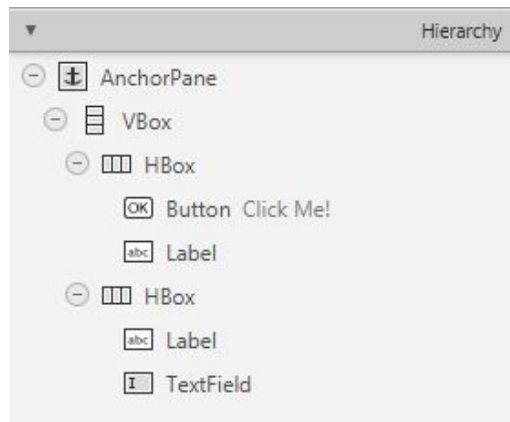


# Sección Document

Esta sección sólo incluye los apartados Hierarchy (jerarquía) y Controller.

En el apartado Hierarchy puedes ver de manera detallada qué elementos están contenidos dentro de otros, y puede resultar de bastante utilidad cuando desees reorganizar los elementos dentro de los distintos contenedores que tengas en el diseño.

En el apartado Controller puedes ver y modificar el nombre de la clase que va a hacer las funciones de controladora de este archivo FXML.



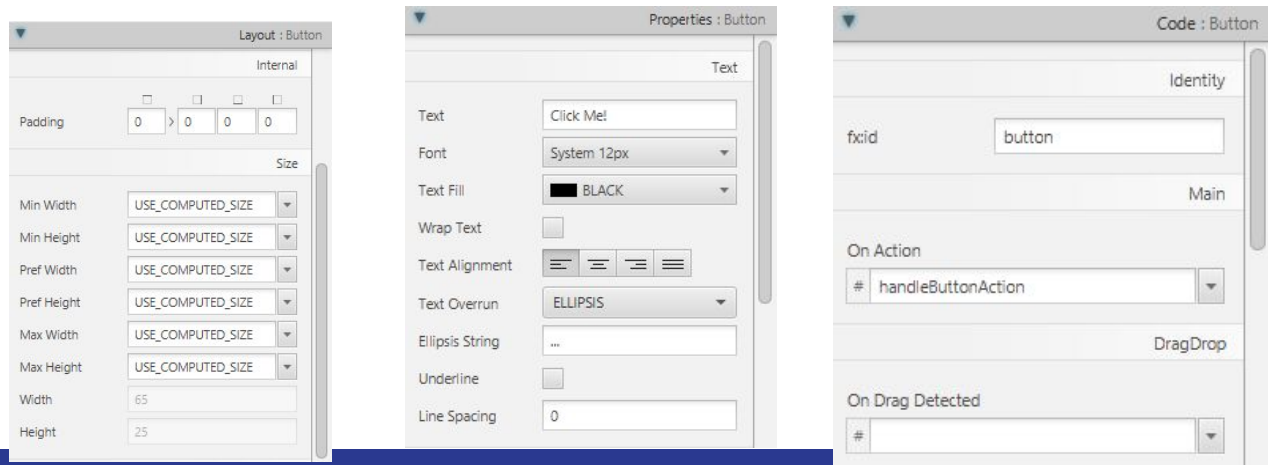
# Sección Inspector

Contiene los apartados Properties, Layout, y Code.

En el apartado Properties podrás consultar y modificar los valores asignados a las propiedades que ofrece el elemento que se encuentre seleccionado. Según el tipo de elemento que sea, se mostrarán distintas propiedades. Aquí podrás cambiar, por ejemplo, el texto de un elemento, su color, tipo de fuente, alineaciones, o asignarle un estilo CSS de JavaFX.

El apartado Layout permite consultar y modificar las aspectos relacionados con la distribución del elemento en pantalla. Es decir, ahí podrás asignar márgenes, tamaños, escalas, etc.

Por último, el apartado Code ofrece las opciones relacionadas con el código Java asociado al elemento seleccionado. Es importante conocer que ahí es donde se le puede asignar un identificador (fx:id). También aquí se indicará el nombre del método que se desea ejecutar cuando se produzca un determinado evento sobre el elemento, por ejemplo, cuando se haga clic en un botón (evento On Action).



# Make Controller

Tras cualquier cambio que realices en el apartado Code, y tras guardar los cambios del archivo en Scene Buiderr, debes actualizar la información en la clase controladora de NetBeans. Esto debes hacerlo con la opción Make Controller del menú contextual del archivo FXML.

