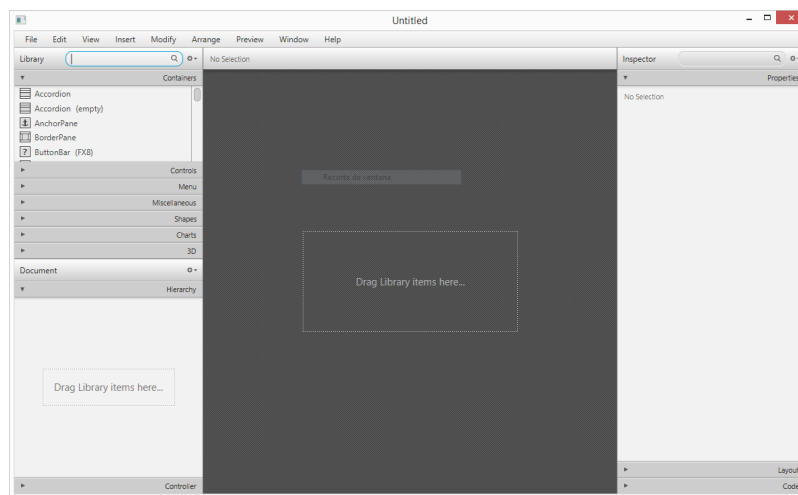


Primeros pasos con JavaFX, Scene Builder y NetBeans (/programacion-en-java/96-javafx/399-primeros-pasos-con-javafx-scene-builder-y-netbeans)

La aplicación Scene Builder permite diseñar, mediante un interfaz gráfico, las estructuras de las ventanas de las aplicaciones que queramos desarrollar usando JavaFX. En este artículo podrás conocer los fundamentos básicos para empezar a usar esta herramienta de manera integrada con el entorno de desarrollo NetBeans.

Descarga de Scene Builder: <http://gluonhq.com/open-source/scene-builder/>
(<http://gluonhq.com/open-source/scene-builder/>)



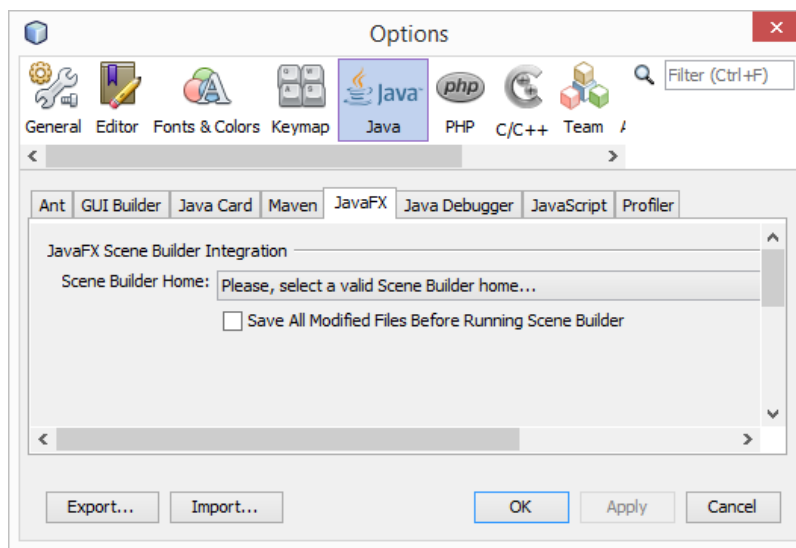
Configurar en NetBeans la localización de SceneBuilder

Con el fin de que cuando se abra un archivo FXML desde NetBeans se muestre directamente con la herramienta Scene Builder, se debe indicar en la configuración de NetBeans en qué carpeta se encuentra Scene Builder.

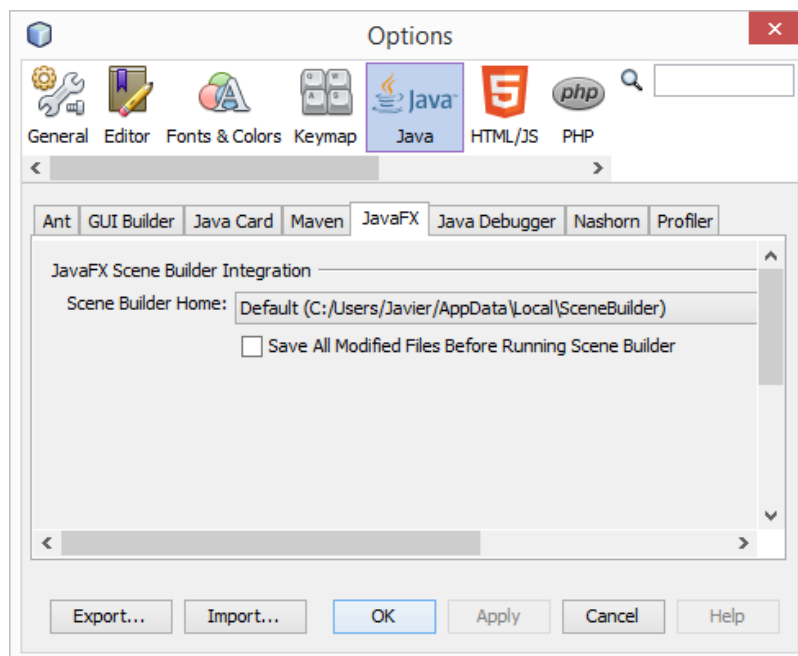
En el artículo Using Scene Builder with NetBeans IDE (<https://docs.oracle.com/javase/8/scene-builder-2/work-with-java-ides/sb-with-nb.htm#CHEEHIDG>) de la web de Oracle se puede obtener también información sobre los pasos a seguir.

La versión 8.0 de **Scene Builder** se encuentra instalada por defecto en la carpeta **C:\Users\TU_USUARIO\AppData\Local\SceneBuilder\SceneBuilder.exe**

Puedes acceder a las opciones de configuración de NetBeans en el menú **Tools > Options**. Ahí accede a la **sección Java y la pestaña JavaFX**:

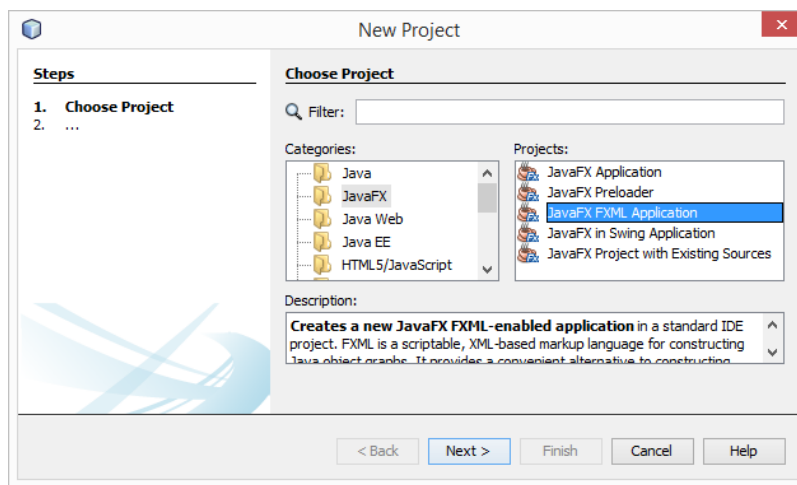


En NetBeans 8.0.2 no se reconoce la carpeta de instalación de Scene Builder. En cambio en la versión 8.1 RC de NetBeans que es la más moderna a la que se puede acceder en el momento de crear este artículo, se detecta automáticamente la carpeta de instalación de Scene Builder:

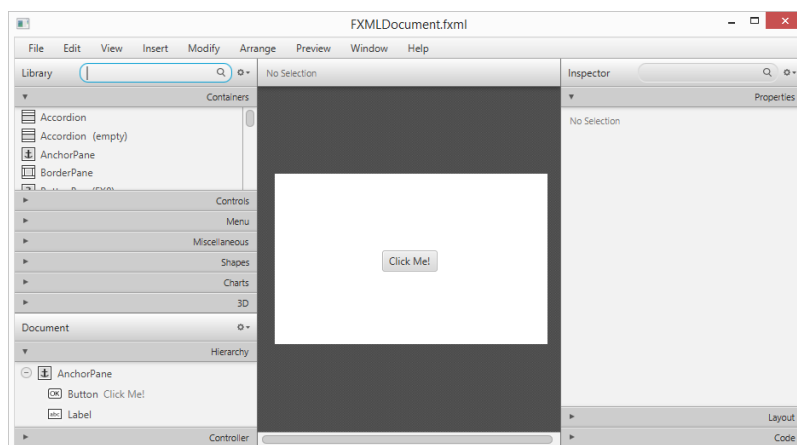
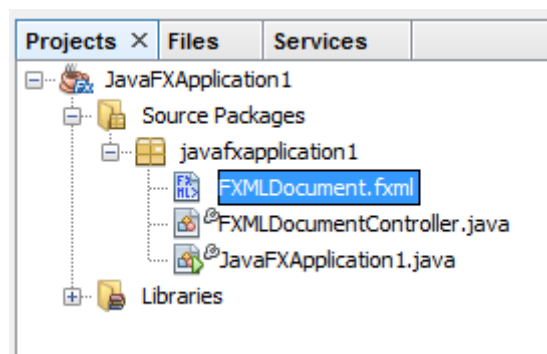


Creación de un proyecto JavaFX

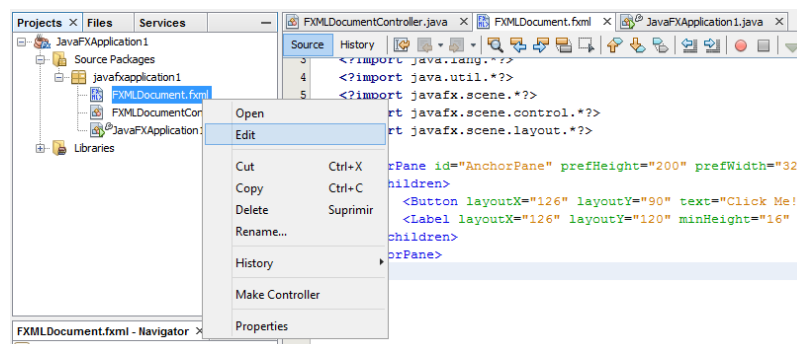
Utiliza el asistente de creación de proyectos de NetBeans para crear un proyecto de tipo **JavaFX > JavaFX FXML Application**, que crear automáticamente la estructura básicas de una aplicación JavaFX con un archivo FXML básico para la estructura de la pantalla.



Al hacer doble clic sobre el archivo FXML que se crea, o bien seleccionando la opción Open de su menú contextual, verás que se abre automáticamente la aplicación Scene Builder con ese documento abierto.

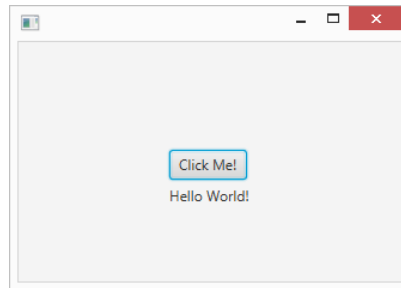


También puede optar por ver el código fuente de dicho archivo desde NetBeans. Si no lo encuentras en las pestañas de los archivos que se encuentren abiertos, puedes usar la opción **Editar** de su menú contextual. Como puedes observar, se trata de un archivo XML que contiene la estructura de la ventana, la cual puedes diseñar **editando este código o de manera visual desde la aplicación Scene Builder**.



Ejecución de la aplicación

Como cualquier otra aplicación Java, al usar el **botón *Run*** de NetBeans, podrás ver en ejecución la aplicación tal como se encuentre desarrollada hasta ese momento. Al hacerlo sobre el proyecto recién creado podrás ver una ventana con un botón "*Click Me!*" que al pulsarlo muestra debajo el mensaje "*Hello World!*".



Estructura del código fuente del proyecto

En la estructura del proyecto que se acaba de crear, puedes ver que se han generado automáticamente 3 archivos:

- FXMLDocument.fxml
- FXMLDocumentController.java
- JavaFXApplication1.java

Esta es la manera en la que están **relacionados entre ellos**:

La clase java **JavaFXApplication1** es la principal en este proyecto, y será la que inicia la ejecución de la aplicación. Dentro de su código fuente puedes observar que se encarga de **cargar la estructura de la ventana** contenida en el archivo **FXMLDocument.fxml**:

```

package javafxapplication1;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class JavaFXApplication1 extends Application {

    @Override (http://docs.oracle.com/javase/7/docs/api/java/lang/Override.html)
    public void start(Stage stage) throws Exception (http://docs.oracle.com/javase/7/docs/api/java/lang/Exception.html) {
        Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));

        Scene scene = new Scene(root);

        stage.setScene(scene);
        stage.show();
    }

    public static void main(String (http://docs.oracle.com/javase/7/docs/api/java/lang/String.html)[] args) {
        launch(args);
    }
}

```

A su vez, el archivo **FXMLDocument.fxml** hace referencia en su código a la clase Java **FXMLDocumentController** que va a hacer las funciones de controlador, gestionando las acciones que realice el usuario sobre los elementos de la ventana. Puedes ver el código fuente contenido en este archivo editándolo desde NetBeans:

```

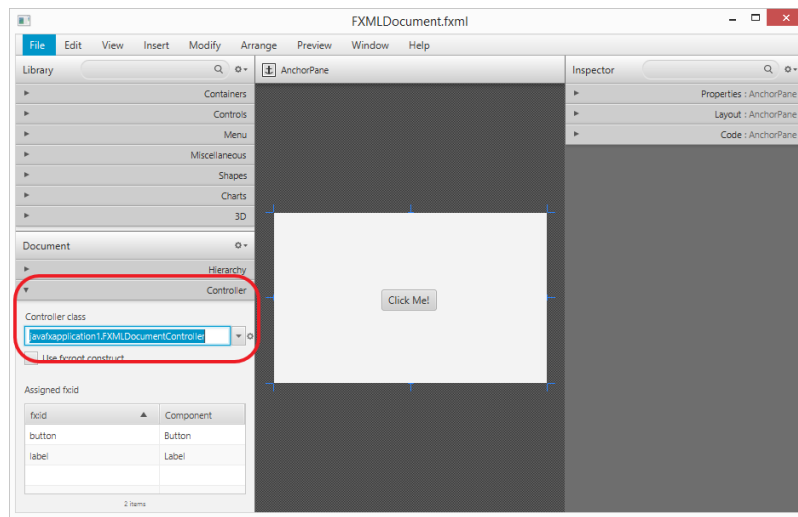
<?xml version="1.0" encoding="UTF-8"?>

<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<AnchorPane id="AnchorPane" prefHeight="200" prefWidth="320"
    xmlns:fx="http://javafx.com/fxml/1"
    fx:controller="javafxapplication1.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>

```

Desde Scene Builder también puedes **ver o modificar** la clase Java que va a hacer las funciones de **controlador**, concretamente desde el apartado **Controller** que puedes encontrar en la parte inferior derecha.



Dentro del trozo de **código correspondiente al botón** (*Button*) en el archivo que se acaba de mostrar, puedes observar que se hace referencia al **método que se ejecutará** cuando el usuario de la aplicación haga clic en él. En concreto se indica que se ejecute el método ***handleButtonAction*** que se encuentra dentro del controlador *FXMLElementController*.

Este es el código fuente del controlador (**FXMLElementController.java**):

```
package javafxapplication1;

import java.net.URL;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Label;

public class FXMLElementController implements Initializable {

    @FXML
    private Label (http://docs.oracle.com/javase/7/docs/api/java/awt/Label.html) label;

    @FXML
    private void handleButtonAction(ActionEvent (http://docs.oracle.com/javase/7/docs/api/java/awt/event/ActionEvent.html) event) {
        System (http://docs.oracle.com/javase/7/docs/api/java/lang/System.html).out.println("You clicked me!");
        label.setText("Hello World!");
    }

    @Override (http://docs.oracle.com/javase/7/docs/api/java/lang/Override.html)
    public void initialize(URL (http://docs.oracle.com/javase/7/docs/api/java/net/URL.html) url, ResourceBundle (http://docs.oracle.com/javase/7/docs/api/java/util/ResourceBundle.html) rb) {
        // TODO
    }
}
```

Observa que para que un **método pueda ser invocado desde el archivo FXML**, debe indicarse antes de la declaración de dicho método la **anotación @FXML**.

De manera similar, observa que antes de la declaración del elemento *label* también se ha usado esa anotación. Eso es necesario también en ese caso, ya que desde el código Java se está haciendo referencia al elemento con ese mismo nombre (*label*) para cambiar el texto que contiene (recuerda que aparece el texto *Hello World!*)

En el archivo FXML has podido ver que se había declarado

```
<Label layoutX="126" layoutY="120" minHeight="16"
        minWidth="69" fx:id="label" />
```

En el controlador Java se ha declarado como:


```
@FXML
private Label (http://docs.oracle.com/javase/7/docs/api/java/awt/Label.html) label;
```


Y se ha cambiado su contenido con:


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

Detalles

Categoría: JavaFX (/programacion-en-java/96-javafx)

 Publicado: 10 Octubre 2015

 Última actualización: 10 Octubre 2015

 Visto: 39030