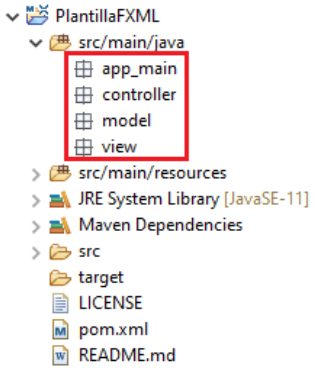
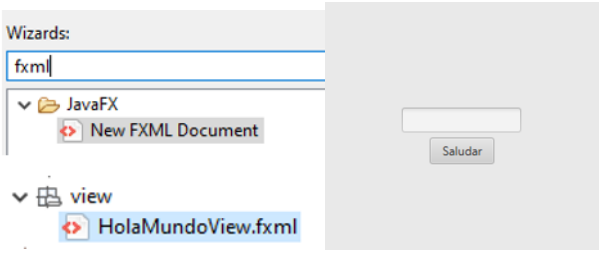
# MÓDELO VISTA CONTROLADOR – JAVAFX (FXML)

Se recomienda los siguientes paquetes:

- **View**: Ventanas sin lógica
- **Model:** Clases que se usaran, (bindeos y properties)
- **Controller**: Maneja los eventos de la ventana, donde se realiza la lógica del programa.
- **App y Main:** Encargado de cargar el controller y configurar la ventana de salida. Main ejecutará la aplicación.

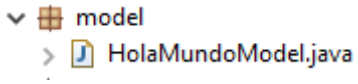



1. **View** - Crear un fichero FXML, los elementos que van a interactuar han de llevar un **identificador obligatoriamente**.



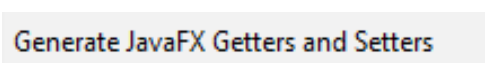

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.VBox?>
<VBox fx:id="viewHolaMundo" alignment="CENTER" minHeight="300.0" minWidth="300.0" xmlns:fx="http://javafx.com/fxml/1" xmlns="http://javafx.com/javafx/11.0.1">
   <children>
      <TextField fx:id="txt_entrada" alignment="CENTER" maxWidth="120.0" minWidth="120.0">
         <VBox.margin>
            <Insets bottom="5.0" left="5.0" right="5.0" top="5.0" />
         </VBox.margin>
      </TextField>
      <Button fx:id="btn_saludar" mnemonicParsing="false" prefHeight="27.0" prefWidth="65.0" text="Saludar" />
      <Label fx:id="lbl_saludo" prefHeight="17.0" prefWidth="134.0">
         <VBox.margin>
            <Insets bottom="5.0" left="5.0" right="5.0" top="5.0" />
         </VBox.margin>
      </Label>
   </children>
</VBox>
```

2. **Model –** Crear clase en el paquete correspondiente, en este caso incorporaremos los bindings necesarios para el proyecto.

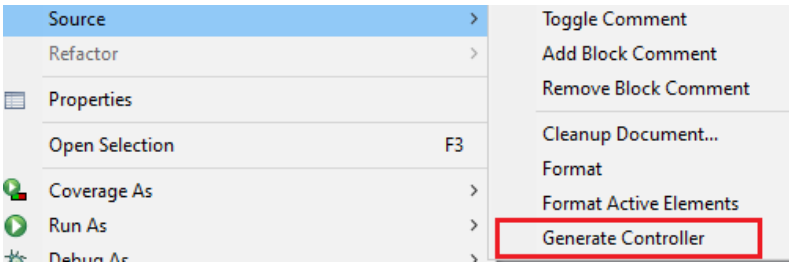


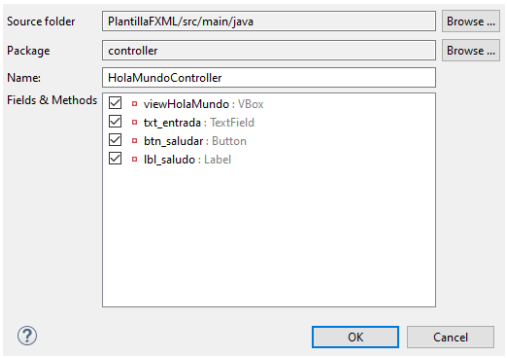

Llevará las **properties** necesarias para interactuar, y en el caso de que se necesite operar con estas un **constructor con las operaciones**. No hay que olvidarse de **añadir los JavaFX Getters & Setters.**

```
public class HolaMundoModel {

    private StringProperty txt_entrada = new SimpleStringProperty();
    private StringProperty lbl_salida = new SimpleStringProperty();
```



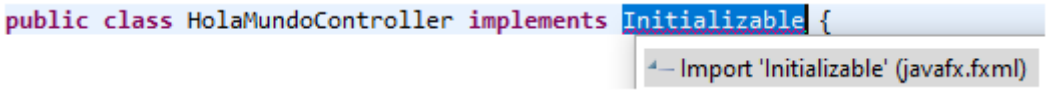
3. **Controller** - se creará a partir del View que ya hemos hecho, para ello dentro de este haremos lo siguiente:




Se seleccionarán los elementos con los que se va a interactuar y generaremos el controller en el paquete correspondiente. Esto creara los atributos correspondientes, al view y los eventos creados en la ventana.

A continuación implementaremos intializable:




Se creará un constructor, que cargará la vista a través de su ruta relativa(*Copiable*):

```
publico HolaMundoController() throws IOException {
                FXMLLoader loader = new
FXMLLoader(getClass().getResource("/fxml/HolaMundoView.fxml"));
                loader.setController(this);
                loader.load();
}
```

Dentro del método sobreescrito, se realizan Bindeos y listeners nuevos:

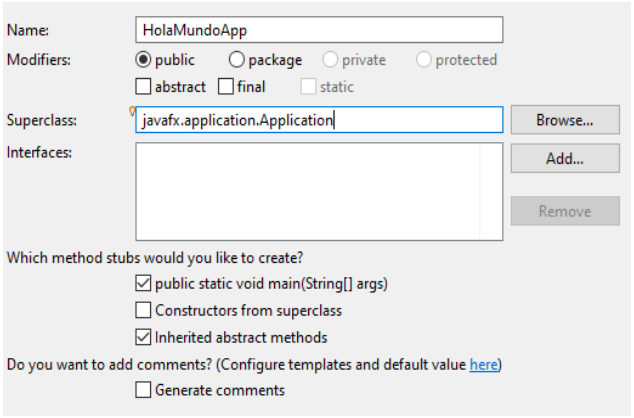
```
@Override
public void initialize(URL location, ResourceBundle resources) {

    //Aquí haremos los bindeos necesarios
    model.txt_entradaProperty().bind(txt_entrada.textProperty());
    model.lbl_salidaProperty().bind(lbl_saludo.textProperty());

    //Listeners propios
    btn_saludar.setOnAction(e -> onClickSaludarActionEvent(e));

}
```

Se creará un getter, para poder acceder a la vista

```
public VBox getViewHolaMundo() {
    return viewHolaMundo;
}
```

Crear funciones necesarias…

4. **Application** – Se creará una nueva clase de la siguiente manera:




Tendrá un atributo controller, dentro del start se instanciará el controller, y se creará una nueva escena y se configurará el stage. Y por se pasarán los argumentos a través de un main

```
import controller.HolaMundoController;

public class HolaMundoApp extends Application {

    private HolaMundoController controller;

    @Override
    public void start(Stage primaryStage) throws Exception {

        controller = new HolaMundoController();

        Scene scene = new Scene(controller.getViewHolaMundo(), 320, 200);

        primaryStage.setTitle("HolaMundo con FXML");
        primaryStage.setScene(scene);
        primaryStage.show();


    }

    public static void main(String[] args) {
        launch(args);

    }

}
```

Por último en un main, se ejecutará la aplicación, llamando a los argumentos de la clase App.

```
package app_main;

public class Main {

    public static void main(String[] args) {

        HolaMundoApp.main(args);

    }

}
```

**Resultado final:**