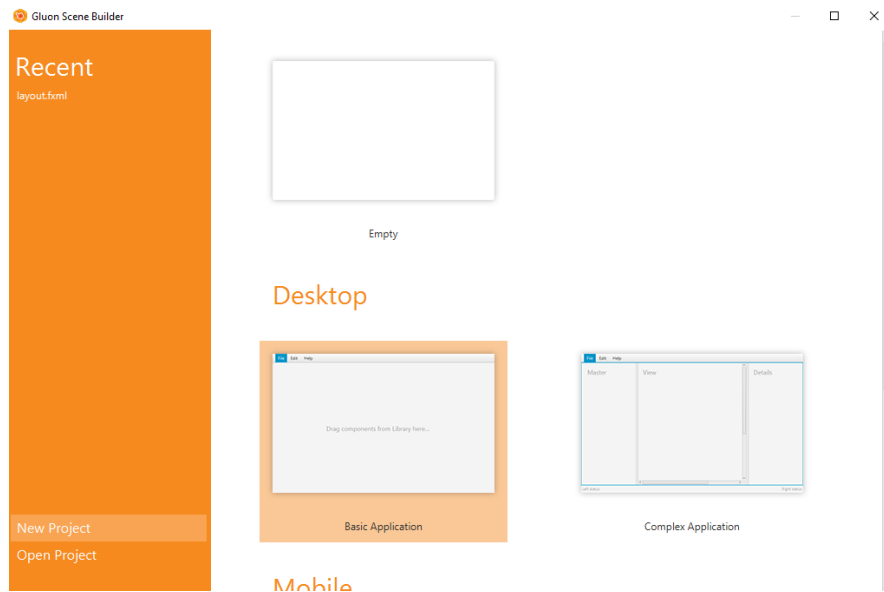
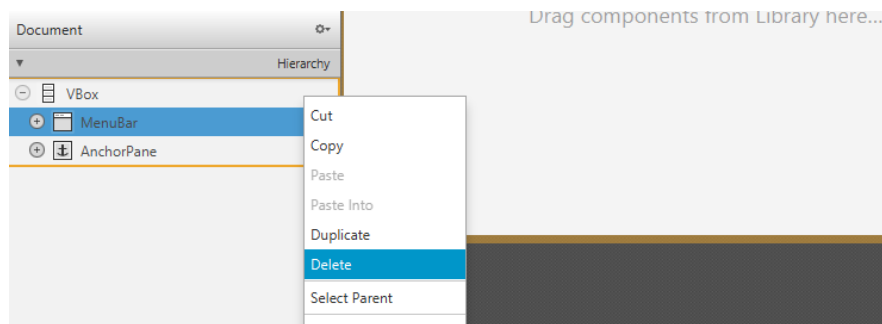


Scene Builder y VS Code

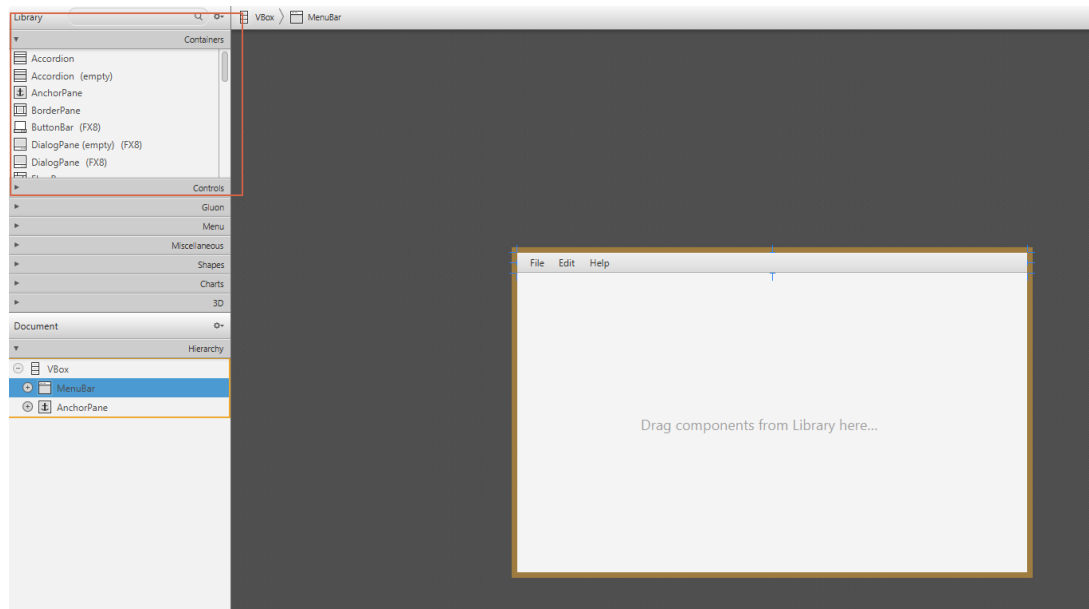
1. Instalar Scene builder <https://gluonhq.com/products/scene-builder/>
2. Abrir Scene Builder y seleccionar Desktop nuevo elemento Basic Application



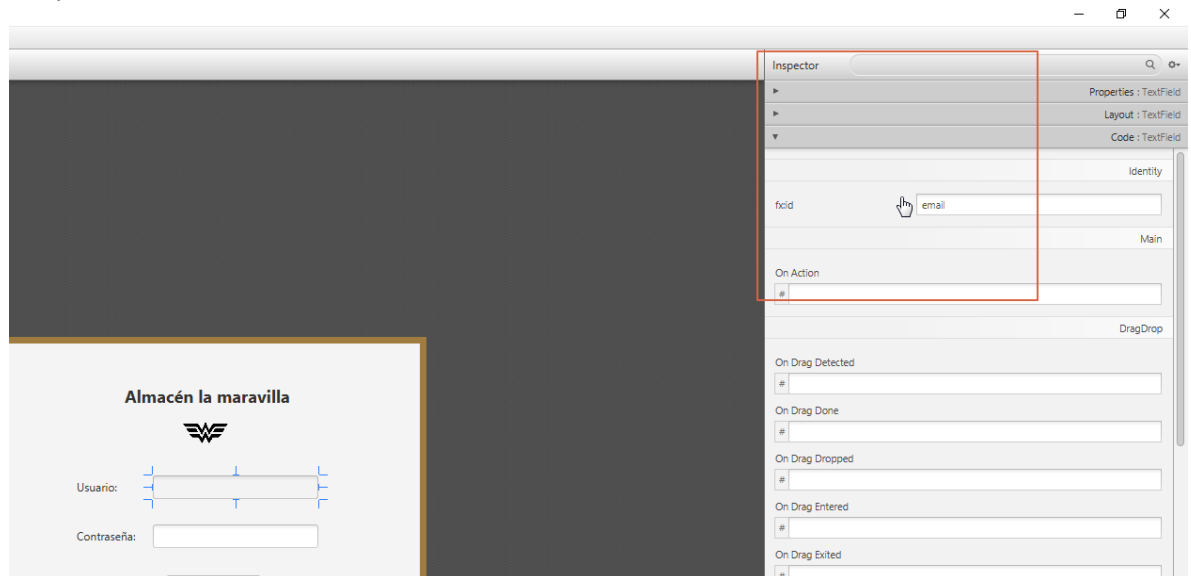
3. Eliminar elemento MenuBar en panel izquierdo de la herramienta o dejarlo en caso de desear contar con este elemento.



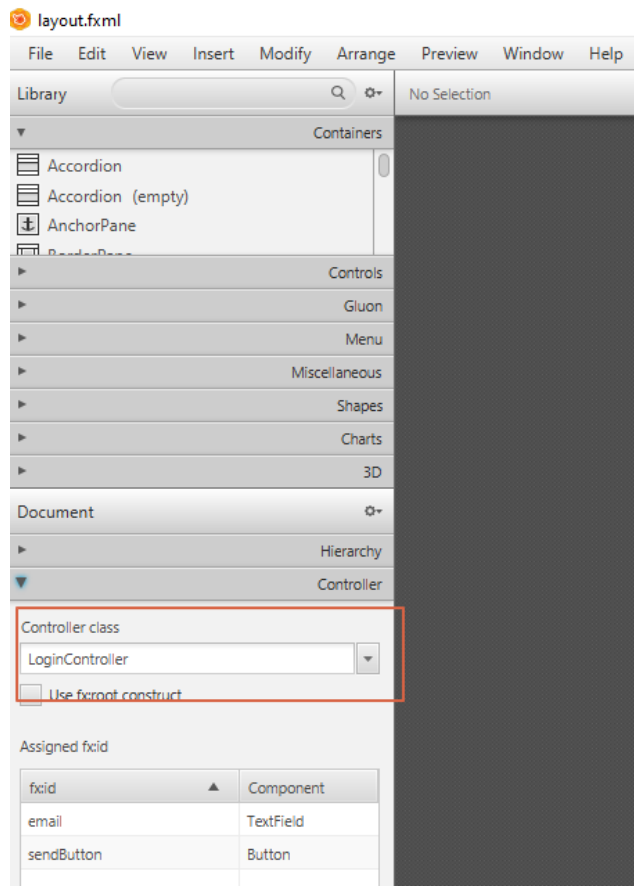
4. Arrastrar desde Library a Pane componentes escogidos para la vista a crear



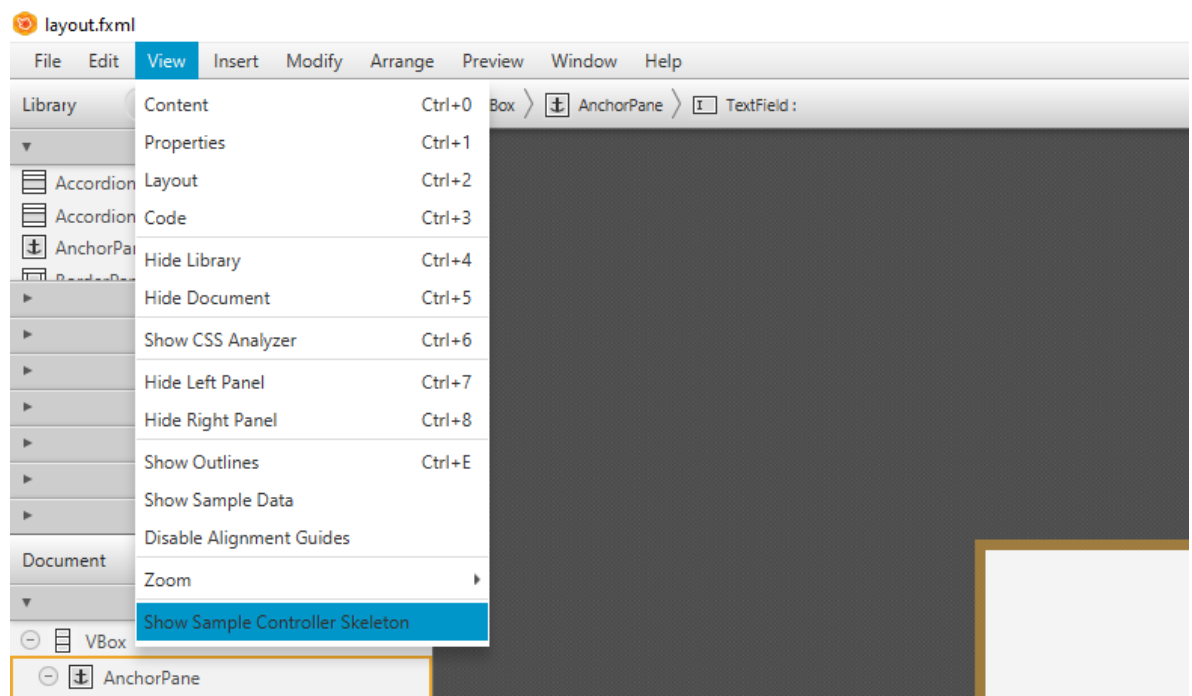
5. Asignar identificadores únicos a cada componente. Seleccionar elemento y desde inspector, panel de Code asignar fx:id o en caso de necesitar eventos asignar nombre en campo On Action.



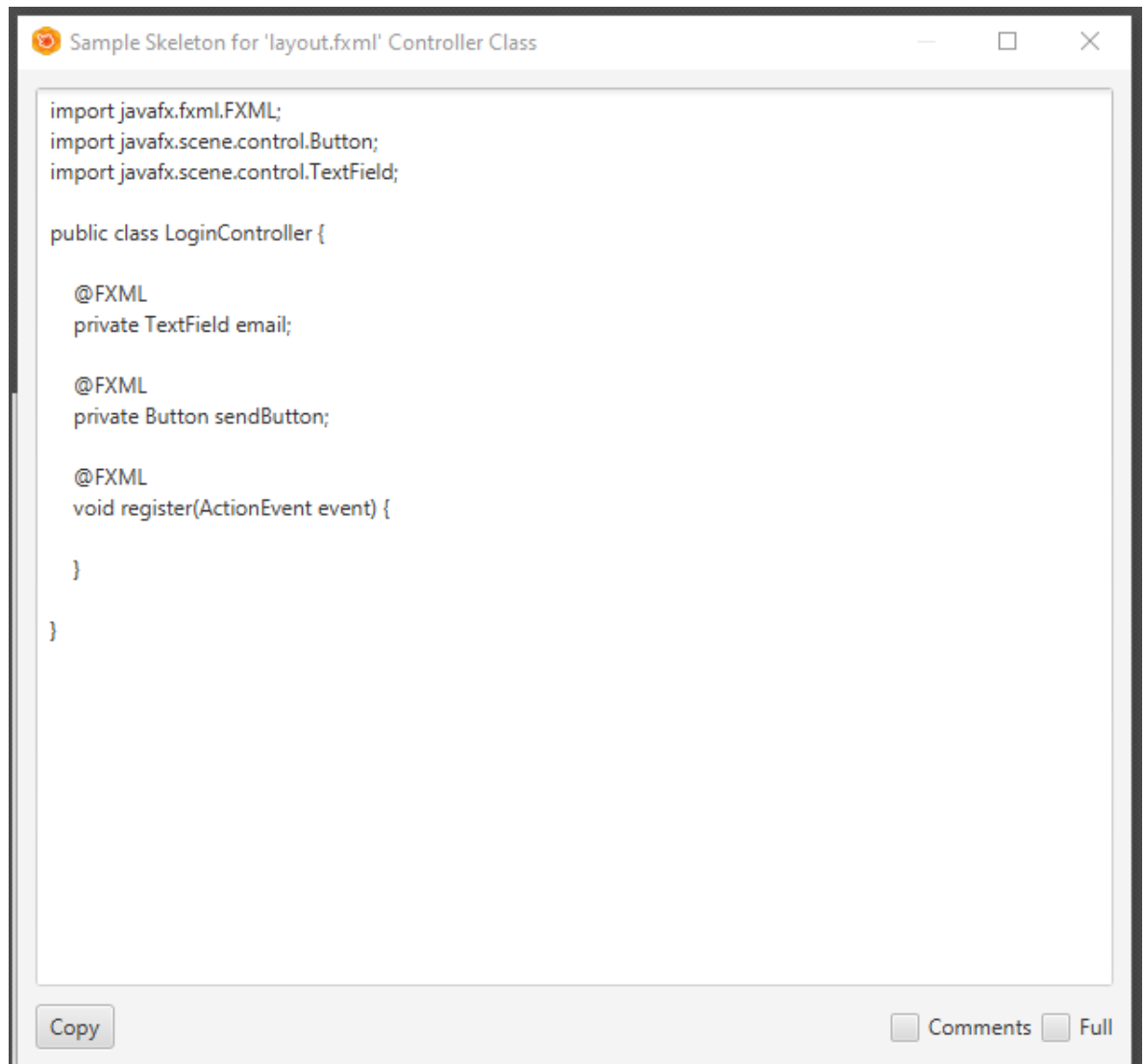
6. Asignar nombre a controlador de la vista en creación. Panel lateral izquierdo, Menú Controller.



7. Acceder a menú View opción Show Sample Controller Skeleton



8. Copiar esqueleto del controlador y cerrar



9. Ir a opción File y guardar archivo .fxml en la carpeta src del proyecto Java, asignar nombre correspondiente a la vista. Ejemplo: layout
10. Abrir proyecto en VS Code donde se desea añadir la vista y Desde App.java o la clase donde deseemos acceder a la vista creada, cargamos y mostramos el archivo fxml correspondiente. Seguir la siguiente estructura:

```
import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;
```

```

import javafx.scene.Scene;

import javafx.stage.Stage;

public class App extends Application {

    @Override

    public void start(Stage primaryStage) throws Exception {

        FXMLLoader fxmlLoader = new

        FXMLLoader(getClass().getResource("layout.fxml"));

        Parent root = fxmlLoader.load();

        Scene scene = new Scene(root);

        primaryStage.setTitle("Ejemplo Scene Builder");

        primaryStage.setScene(scene);

        primaryStage.show();

    }

    public static void main(String[] args) {

        launch(args);

    }

}

```

En el caso de contar con sistema operativo **MacOs**, el archivo App.java deberá contener lo siguiente:

```

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

```

```

import javafx.stage.Stage;

public class App extends Application {

    @Override

    public void start(Stage primaryStage) throws Exception {

        FXMLLoader fxmlLoader = new
FXMLLoader(getClass().getResource("layout.fxml"));

        Parent root = fxmlLoader.load();

        Scene scene = new Scene(root, 300, 250);

        scene.getRoot().setStyle("-fx-font-family: 'serif'");

        primaryStage.setTitle("Ejemplo Scene Builder");

        primaryStage.setScene(scene);

        primaryStage.show();

    }

    public static void main(String[] args) {

        launch(args);

    }

}

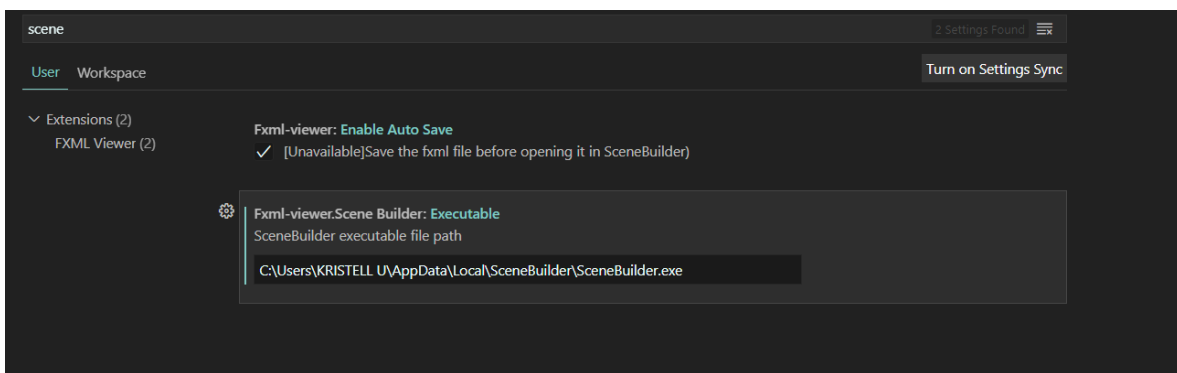
```

11. Una vez cargado el archivo en nuestro proyecto, crear un nuevo file .java con el nombre del controlador creado en el paso 6. Ejemplo: LoginController.java y dentro de este archivo pegar el texto de paso 8.
12. En este punto, el proyecto java cuenta con acceso a la vista fxml creada y tiene acceso a los componentes a los cuales se les asignó id en Scene Builder. Añadir al controlador correspondiente validaciones o funcionalidades según considere y guardar.

13. Ejecutar proyecto. Tener en cuenta que cada vez que realice alguna modificación del archivo fxml layout desde Scene builder y guarde los cambios, este cambio se verá reflejado en el proyecto inmediatamente. En caso de modificar identificadores de componentes, debe actualizar su archivo controlador.

TIP EXTRA:

14. Acceder a la configuración de la extensión FXMLViewer y en la opción Fxml-viewer.Scene Builder: Executable añadir la ruta en nuestro equipo del ejecutable de la herramienta Scene Builder. Ejemplo:



Al añadir esta configuración reiniciamos VS Code y posteriormente podremos contar con una conexión visible entre las herramientas y abrir directamente desde VS Code la herramienta de Scene Builder para ver las vistas creadas. Al abrir un archivo .fxml en la vista superior derecha tendremos el icono de un ojo en el cual al dar click abrirá scene builder.

