ESTÁNDARES DEL PROYECTO

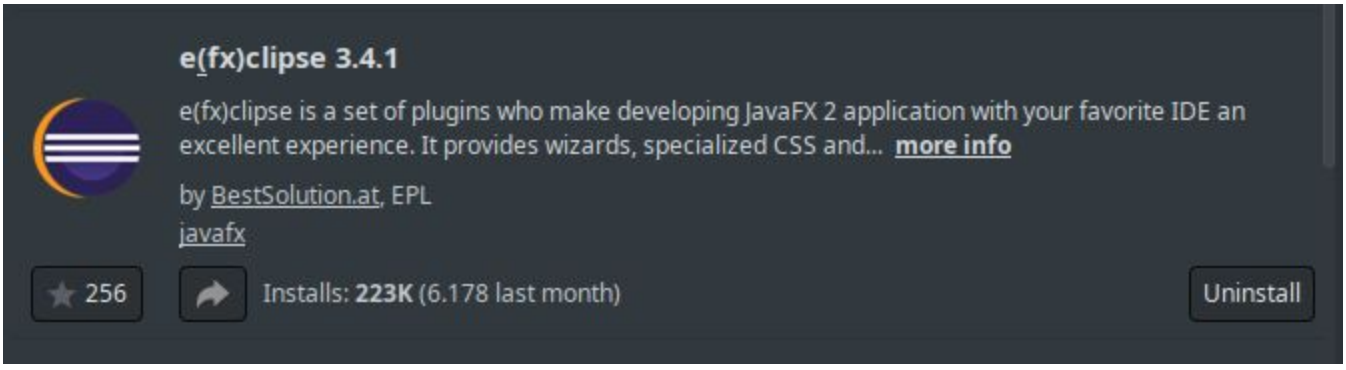
Esta es una guía completa que ayudó al desarrollo del proyecto, encontrará las herramientas usadas y los lineamientos seguidos a medida que se desarrolló. *Este documento quedó para posteridad desde **Abril-24-2019***.

Programas usados

- Java SE, JDK 8 (mirror)
- JavaFX esta viene incluida en el JDK 8
- Eclipse 2018-2
- e(fx)clipse 3.4.1 para integrar JavaFX (se descarga en el marketplace de eclipse)
- Scene Builder 2.0 de oracle
- Libreria jfoenix-8.0.8 para los efectos, (está en el repositorio)
- Librería postgresql-42.2.5, driver de postgres para java (está en el repositorio o descargar)

Configuración de Eclipse

Primero debes instalar el asistente de JavaFX, desde la pestaña help y eclipse marketplace, busca javafx y te saldra el siguiente programa que debes instalar:



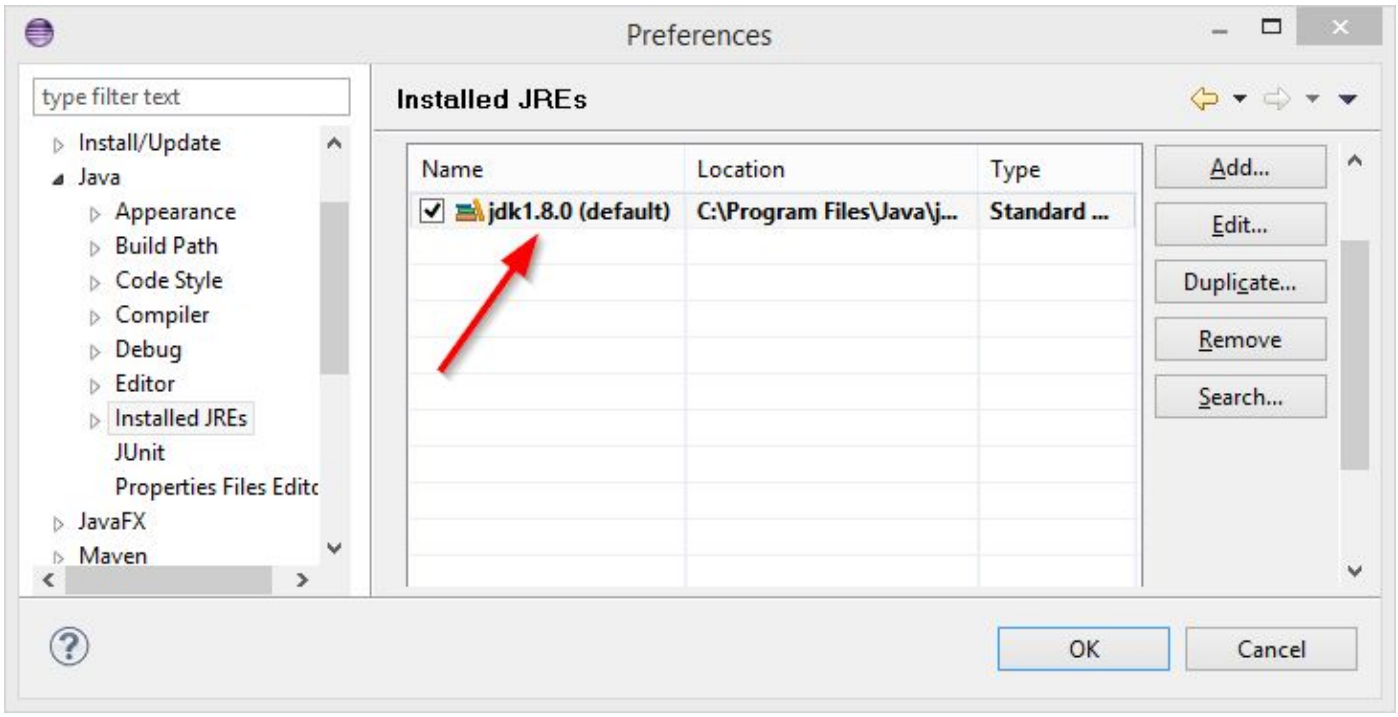
Luego te pedirá reiniciar el programa.

Hay que indicarle a *Eclipse* que use *JDK 8* y también dónde se encuentra el ejecutable del Scene Builder:

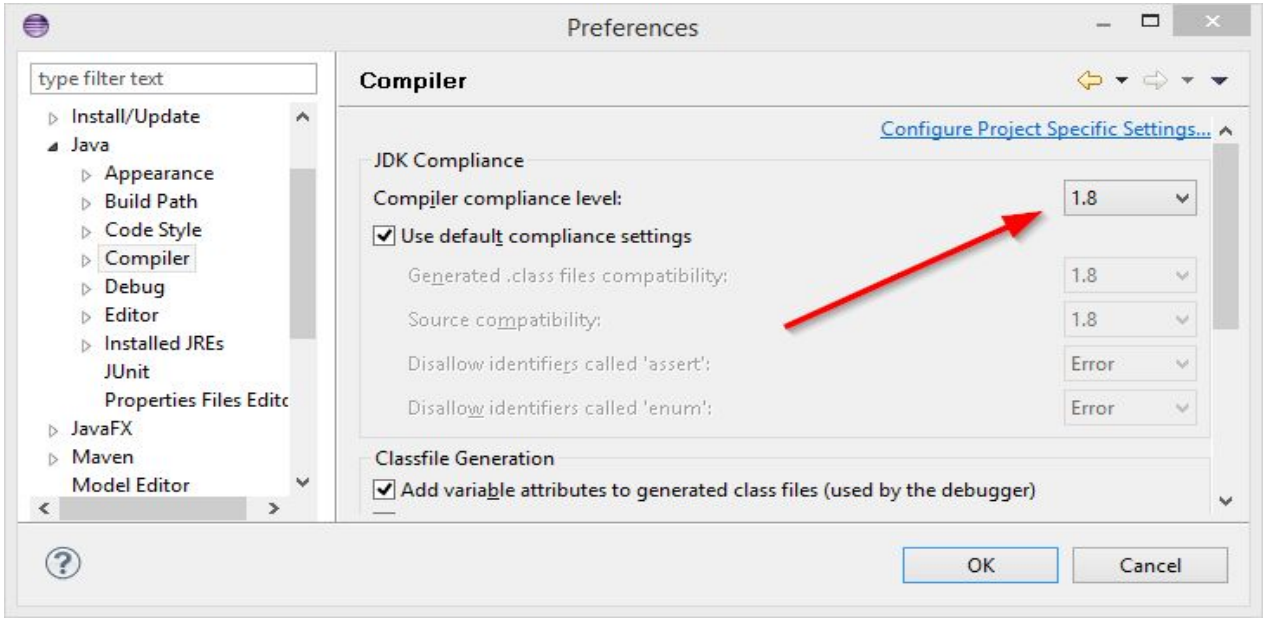
1. Abre las Preferencias de Eclipse (menú *Window* | *Preferences* y navega hasta *Java* | *Installed JREs*.

2. Si no lo tienes el jre1.8 en la lista de JREs, entonces pulsa *Add…*, selecciona *Standard VM* y elige el *Directorio* de instalación (JRE Home directory) de tu JDK 8.

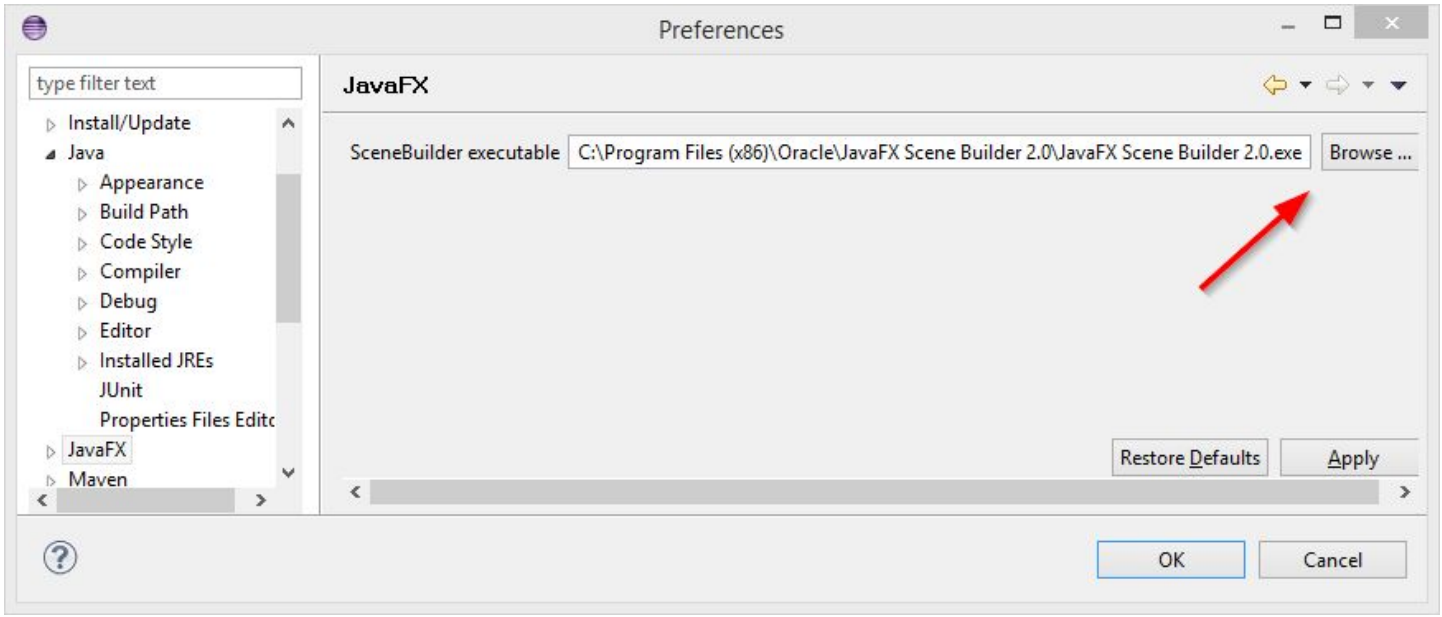
3. Elimina otros JREs o JDKs de tal manera que **JDK 8 se convierta en la opción por defecto**.



4. Navega a *Java* | *Compiler*. Establece el **nivel de cumplimiento del compilador en 1.8** (Compiler compliance level).



5. Navega hasta *Java | JavaFX*. Especifica la ruta al ejecutable del Scene Builder.



Crea un nuevo proyecto JavaFX

En Eclipse (con ***e(fx)clipse*** instalado) ve a *File | New | Other…* y elige *JavaFX Project*. Especifica el nombre del proyecto (ej. *AddressApp*) y aprieta *Finish*.

Borra el paquete *application* y su contenido si ha sido automáticamente creado.

Crea los paquetes

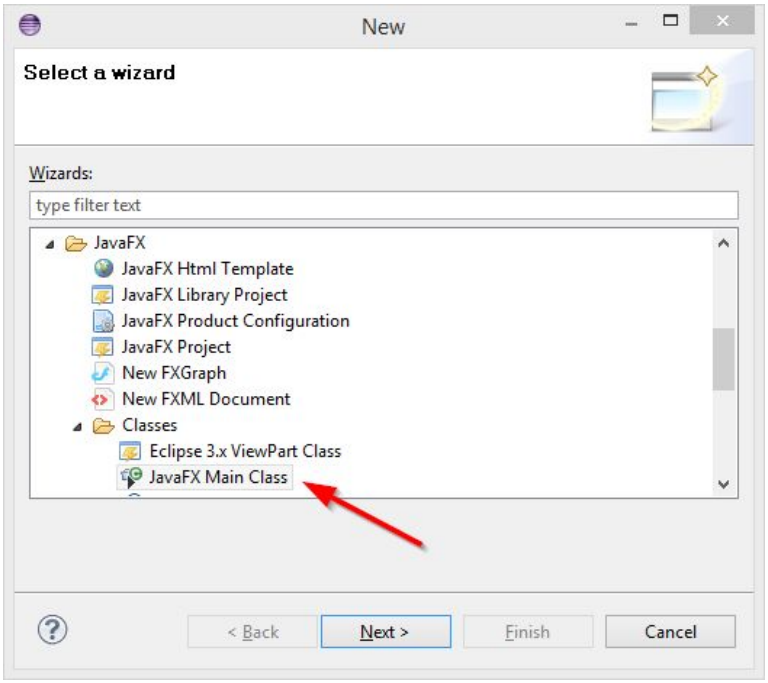
En el ratón hacemos clic derecho en la carpeta *src*, *New | Package*:

- controles - contendrá la *mayoría* de clases de control ©
- clases - contendrá las clases del modelo (M)
- vistas - contendrá las vistas (V)

La clase principal en JavaFX

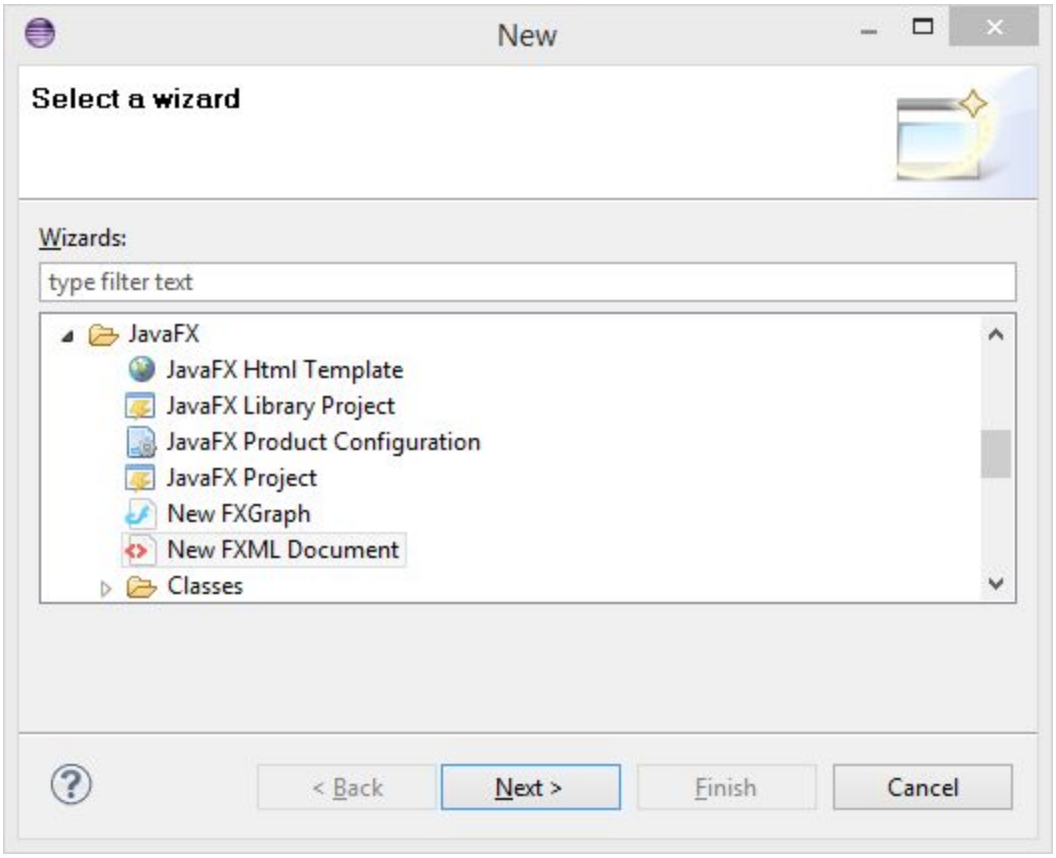
Ahora necesitamos crear la **clase java principal**, la cual iniciará nuestra aplicación

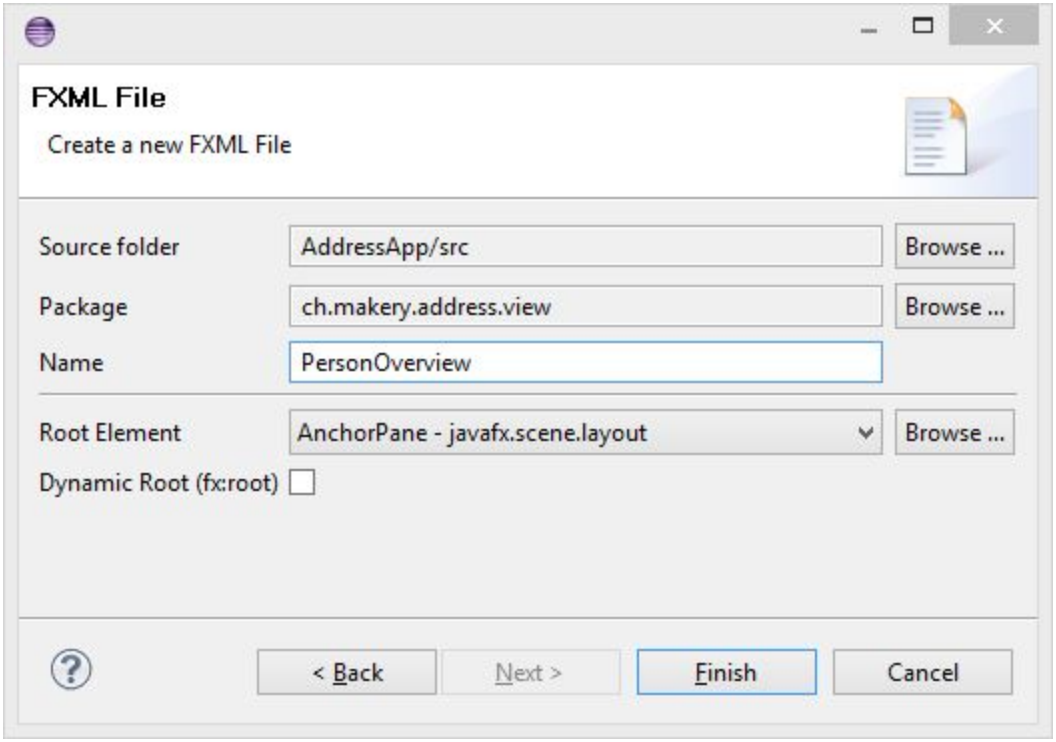
1. Haz clic derecho en el proyecto y elige *New | Other | JavaFX | classes | JavaFX Main Class*.



Crea el archivo FXML de diseño

Haz clic derecho el paquete *vistas* y crea un nuevo archivo FXML (*New | Other | FXML | New FXML Document*) EJ: llamado GUI.



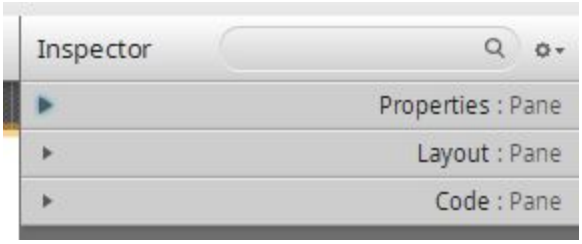


Haz clic derecho sobre `GUI.fxml` y elige *Open with Scene Builder*. O ábrelo manualmente en scene. **En Root Element deben especificar el tipo de layout del elemento padre, en este caso, el elemento padre es un BorderPane.**

Uso de Scene Builder

Descargar scene builder 2.0 de oracle y lo instalas. Lo que encuentras en la parte izquierda (imagen 1):

En la parte derecha, encuentra la propiedades básicas, el tipo de *layout* y *code*, el *code* contiene todos los posibles eventos del elemento



Una vez realizados los cambios, guarda, mira como en tu archivo GUI.fxml en eclipse, tiene todos los cambios.

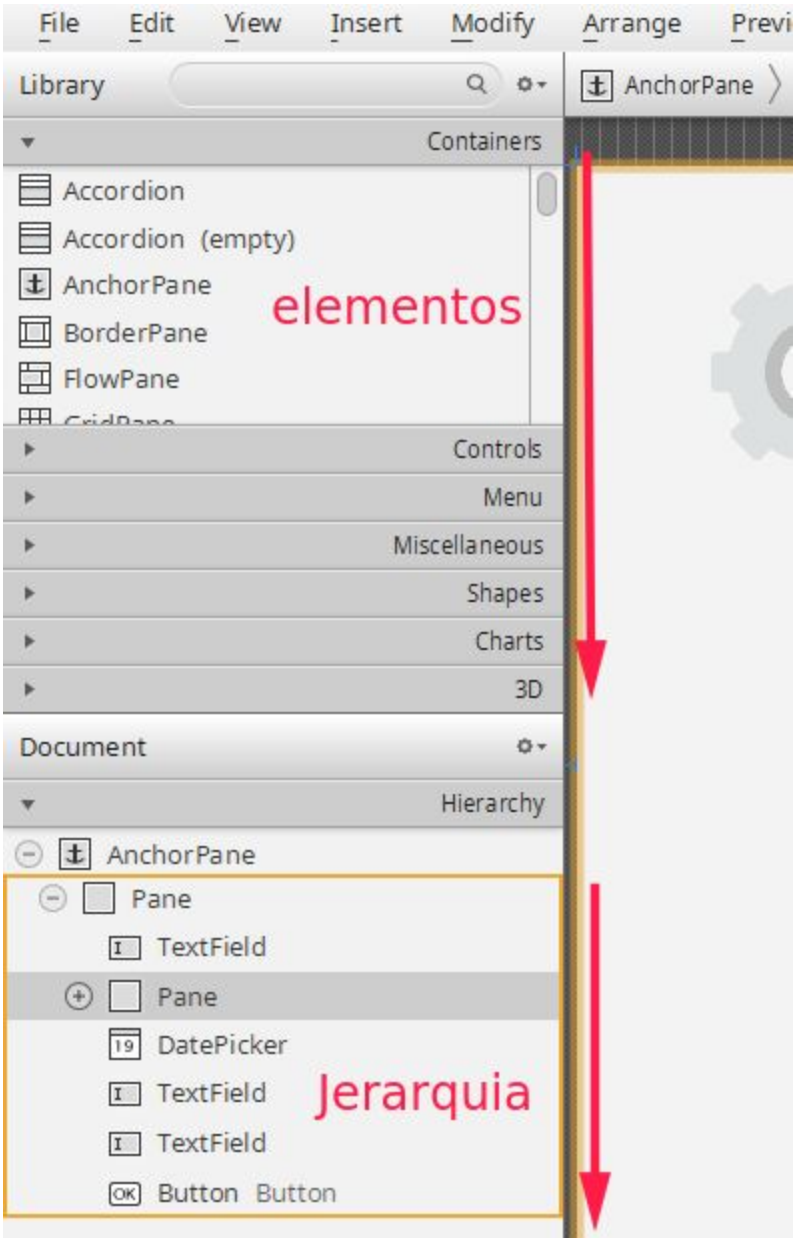


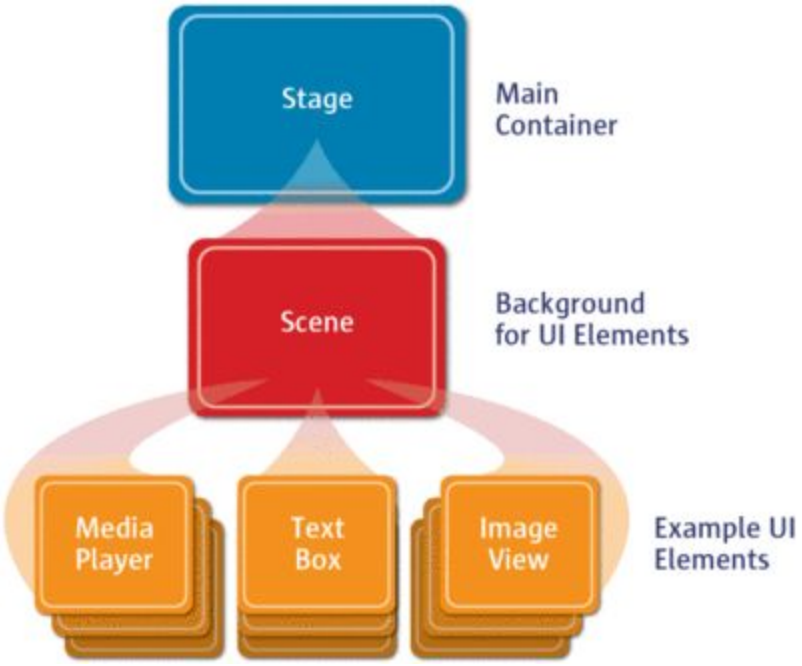
Imagen 1: Jerarquía de Scene Builder

¿Cómo funciona la interfaz?

La clase generada extiende a la clase `Application` y contiene dos métodos. La parte más importante para nosotros es el método `start(Stage primaryStage)`. Este método que tiene un elemento `Stage` como parámetro, es invocado automáticamente cuando la aplicación es lanzada desde el método `main`.

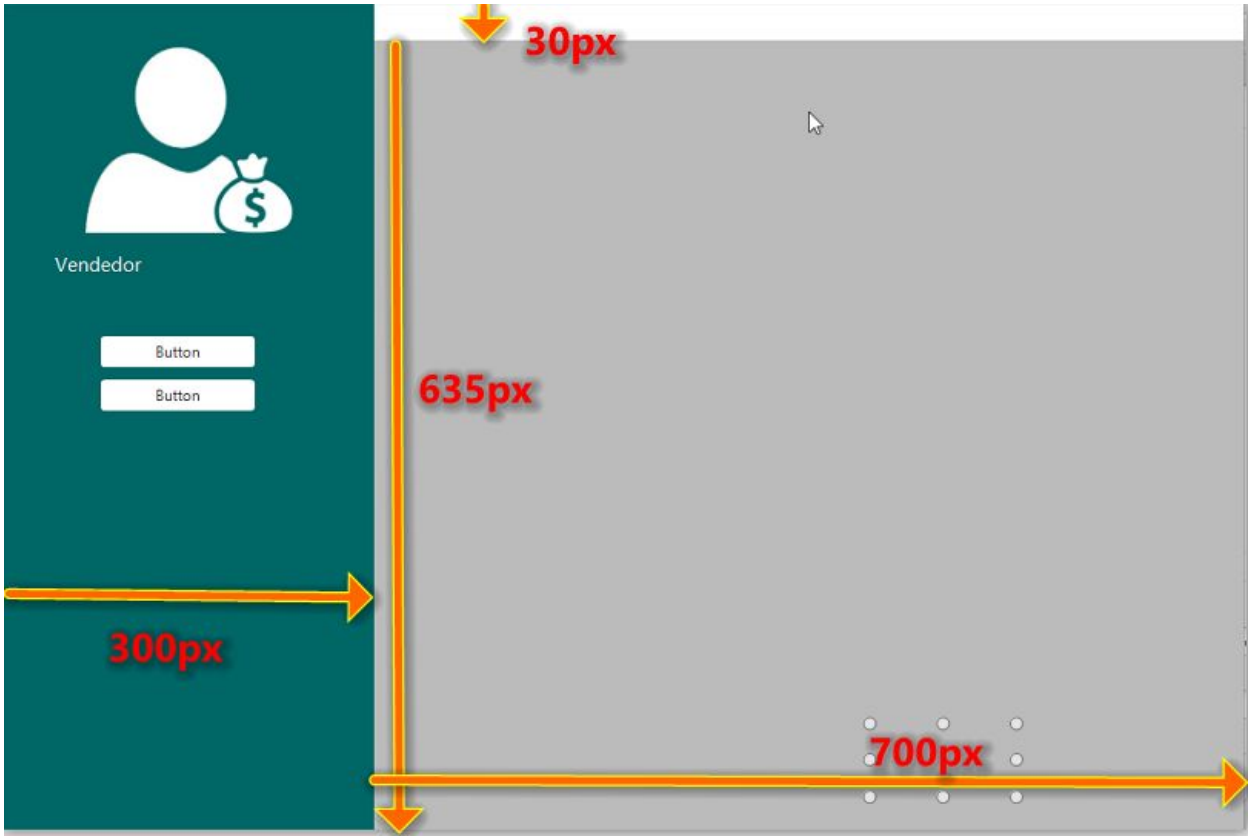
El `Stage` (escenario) es el contenedor principal, normalmente una ventana con borde y los típicos botones para maximizar, minimizar o cerrar la ventana, en este ejemplo sería `GUI.fxml`. Dentro del `Stage`

se puede añadir una Scene (escena), la cual puede cambiarse dinámicamente por otra Scene, este será otro archivo fxml. Dentro de un Scene se añaden los nodos JavaFX, tales como AnchorPane, TextBox, etc.



Tamaño de la interfaz

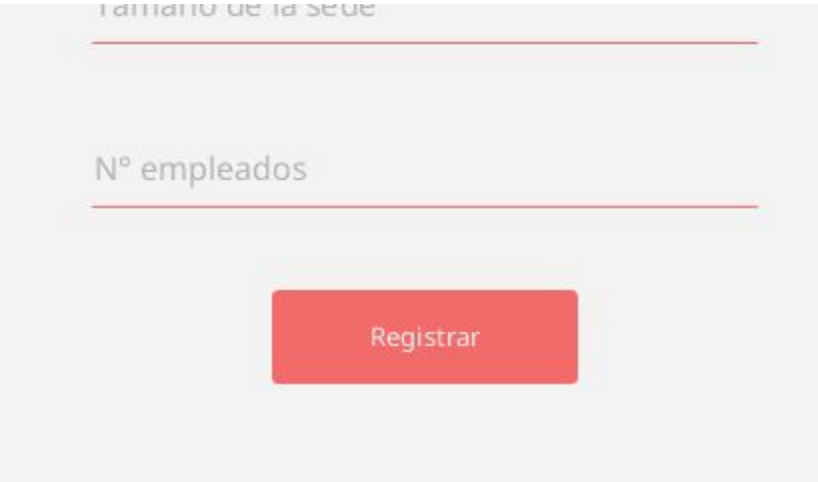
La zona de scenes cambian dinámicamente según el boton escogido en la zona de botones



Solo deben construir la scene según su HU, luego estas scenes serán seteadas en la zona gris, (está gris para diferenciarla de la raíz)

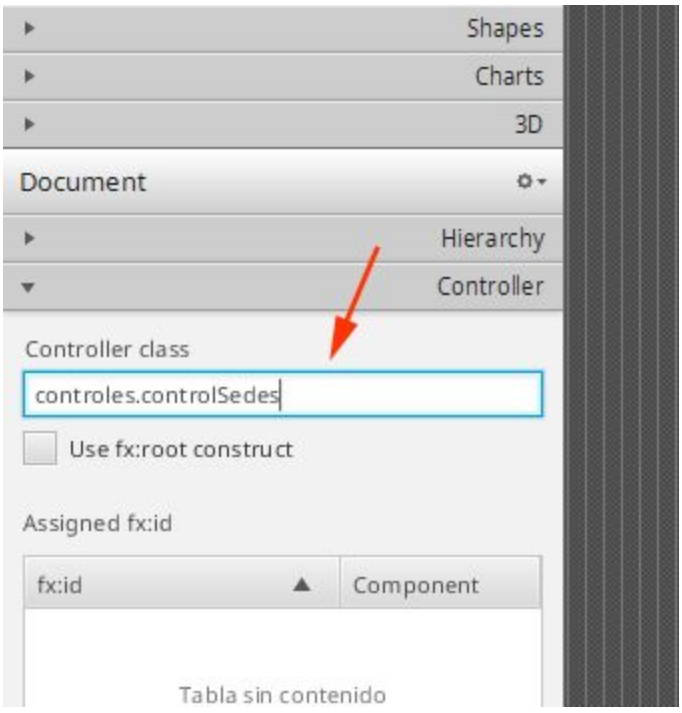
Usar un elemento via controlador

Abre tu archivo fxml mediante scene builder y haz tus cambios. Por ejemplo, vamos a añadir el evento al presionar el siguiente botón:

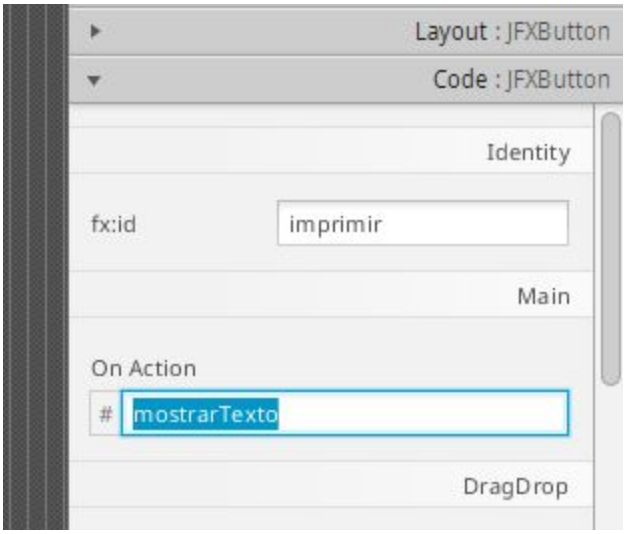


En el proyecto debes tener una clase controladora, para este ejemplo la clase es *controlSedes.java*

1. Abre la clase controladora (aquí abriré *controlSedes.java*)
2. mira el package al principio del archivo, el de este ejemplo es *controles*
3. copia ese nombre y únelo con el nombre la clase controladora y un punto quedara asi: *controles.controlSedes*
4. ve a scene builder y el aparte de controller lo pegas y enter



5. eso enlazo el controlador con el archivo fxml gráfico
6. ahora hay que asignar un id al botón para usarlo, selecciona el botón y ve a code en la parte derecha, asigna un id, aquí pondremos imprimir.
7. abajo del id, esta los eventos del botón, usaremos el evento On Action, pon un nombre, aquí pondremos *mostrarTexto*, enter y guarda



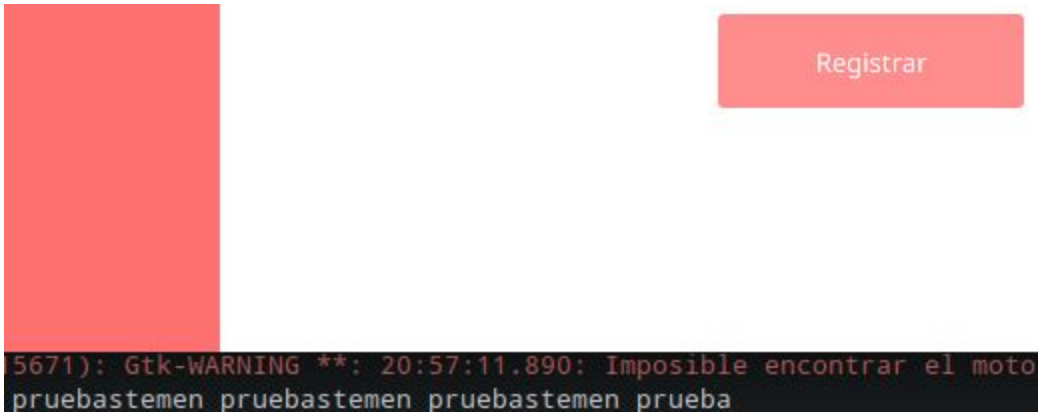
Lo siguiente es ir a la pestaña view de scene builder y seleccionar ***show sample controller skeleton***
Para este ejemplo nos arroja esto:

```
public class controlSedes {

    @FXML
    private JFXButton imprimir;



    @FXML
    void mostrarTexto(ActionEvent event) {

    }
```

Copia esos códigos y los pegas en tu clase controladora. ahora solo queda poner en el evento ***mostrarTexto*** lo que quieres hacer:

```
@FXML
void mostrarTexto(ActionEvent event) {
    System.out.print("stemen prueba");
}
```

Al presionar el botón me imprime el texto:



Características

- **Implementación del sistema de login**
  Se inicia sesión con los gerentes, vendedores, jefes de talleres registrados, además hay un solo administrador quien se encarga de crear más gerentes. Cuenta con las validaciones pertinentes.

Rol	Nombre	Contraseña
Administrador	Kimi Sagi	kspassword
Gerente	Ximena Guzman	xgpassword
Vendedor	Sara Ortega	sopassword
Jefe de Taller	Diana Lopez	dlpassword

- **Implementación y conexión con la bases de datos**
  El paquete ***BaseDatos*** contiene la clase ***FachadaDB.java***, encargada de conectar y cerrar conexión con la base, la clase ***DaoLogin.java*** es usada por el login para hacerle las solicitudes a la base a través de ***FachadaDB,*** en este último archivo, estan los parámetros:

```
FachadaDB(){
    url="jdbc:postgresql://localhost:5432/postgres";
    usuario="postgres";
    password="stemen";
}
```

Cambia la contraseña por la que uses en algun usuario en tu postgres.

Nota: Es necesario descargar el archivo **QueryBase.sql**
Para implementar la base debes iniciar *pgadmin*, luego acceder a tu base principal, de ahí puedes elegir la opción *Query tool* como se muestra en la imagen:

Luego debes dar en la carpeta, elegir **QueryBase.sql** y dar en el icono de rayo, esto importará el esquema de la base. Listo, ya tienes la base implementada.