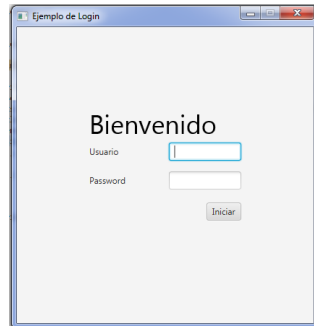


Ejemplo guiado: formulario de login

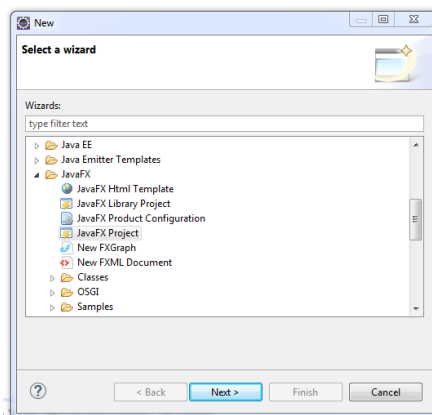
El objetivo del ejercicio es diseñar con Scene Builder una interfaz de usuario que simule el login de un usuario en una aplicación. Utilizaremos un manejador de eventos para mostrar en pantalla un texto cada vez que se pulsa el botón *Iniciar* del formulario.

La apariencia final del formulario es la que se muestra en la figura.



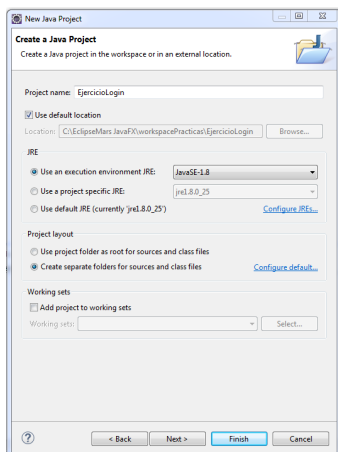
1. Crear el nuevo proyecto JavaFX con Eclipse

Dentro del entorno Eclipse utilice *File -> New -> Other -> JavaFX Project*

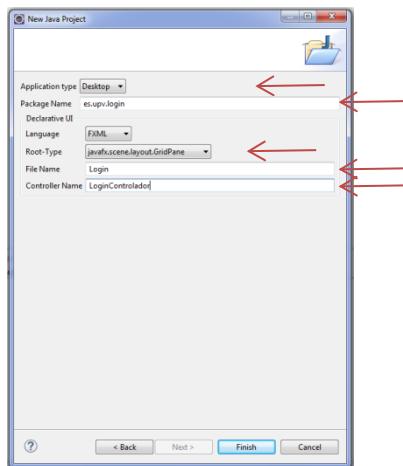


Pulse el botón *Next* y en la siguiente pantalla ponga un nombre al proyecto, *EjercicioLogin*.

Pulse *Next* y en la siguiente pantalla también *Next*.

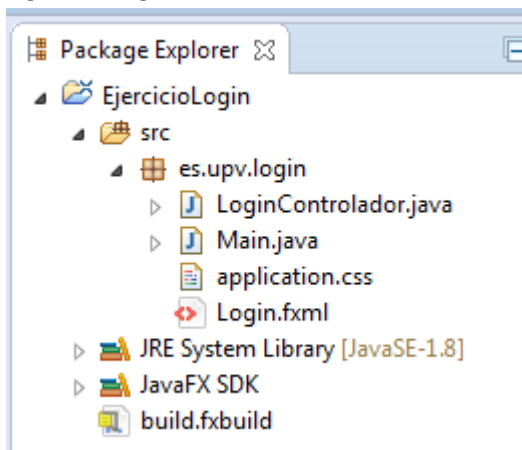


Seleccione la pestaña *Language* y marque *FXML*. En la pestaña *Root-Type* seleccione *GridPane*.

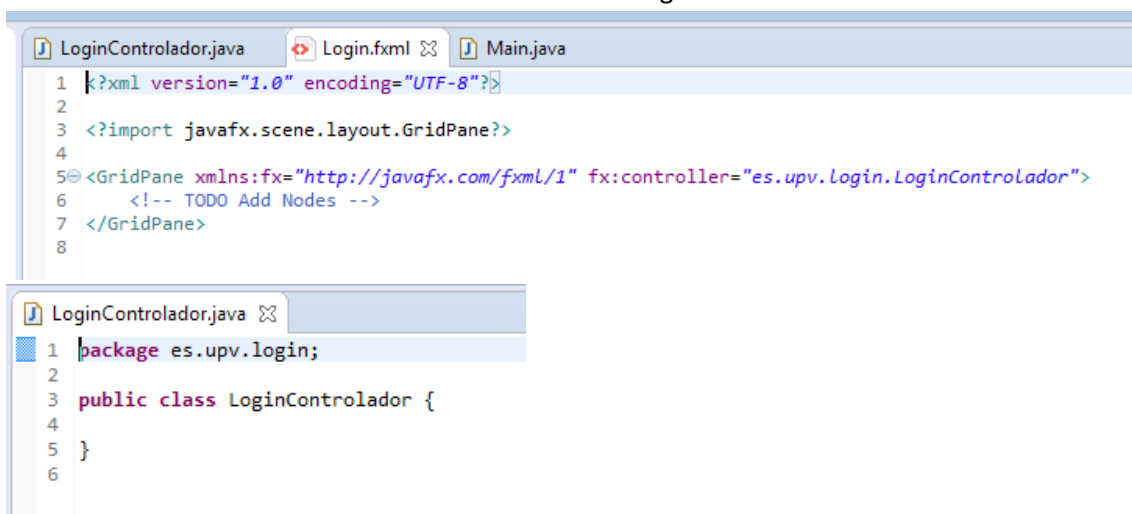


Este será el contenedor de layout de alto nivel de la ventana. Por último en *FieldName* escriba *Login* y en *ControllerName* *LoginControlador*. Si aparece algún icono de advertencia en la figura anterior, ignórela y pulse *Finish*.

Eclipse ha creado la estructura de directorios del proyecto, tal como se muestra en la siguiente figura.



Los archivos recién creados con el asistente tienen la siguiente información:



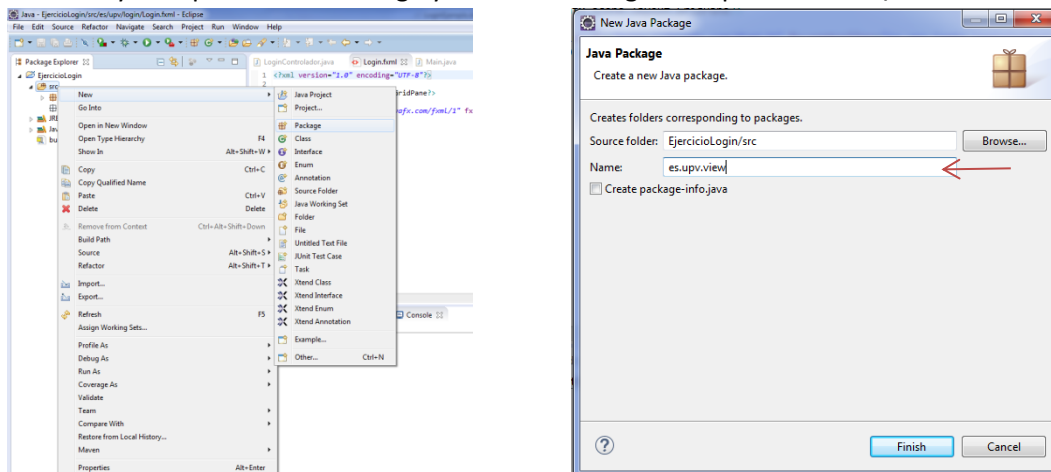
```

1 package es.upv.login;
2
3 import javafx.application.Application;
4
5
6
7
8
9
10 public class Main extends Application {
11     @Override
12     public void start(Stage primaryStage) {
13         try {
14             BorderPane root = (BorderPane)FXMLLoader.Load(getClass().getResource("Login.fxml"));
15             Scene scene = new Scene(root,400,400);
16             scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
17             primaryStage.setScene(scene);
18             primaryStage.show();
19         } catch (Exception e) {
20             e.printStackTrace();
21         }
22     }
23
24     public static void main(String[] args) {
25         Launch(args);
26     }
27 }
28

```

Vamos a cambiar la estructura del proyecto, definiendo paquetes adicionales. Ahora crearemos un paquete llamado `es.upv.view` y arrastraremos los archivos de la interfaz de usuario, en este caso el archivo `Login.fxml`, a esa ubicación.

Seleccione en la ventana *package explorer* el directorio `src`, pulse con el botón derecho del ratón y marque `New -> Package` y escriba en la siguiente pantalla `es.upv.view`



Mantenga con el botón de ratón en el explorador de proyecto el archivo `Login.fxml` y arrástrelo al paquete recién creado.

Para que pueda ejecutarse la aplicación, que en estos momentos únicamente tiene una ventana vacía, cambie en el archivo `Main.java` para indicar la ubicación del archivo `Login.fxml`. Déjelo como aparece en la figura

```

package es.upv.login;

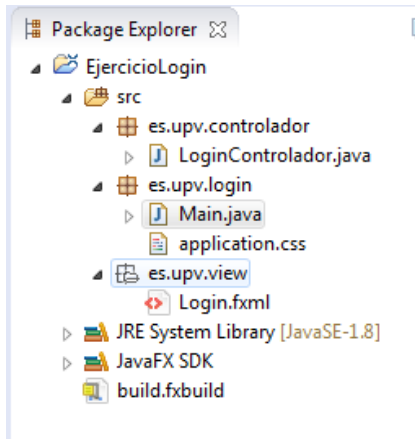
import javafx.application.Application;

public class Main extends Application {
    @Override
    public void start(Stage primaryStage) {
        try {
            GridPane root = (GridPane)FXMLLoader.Load(getClass().getResource("/es/upv/view/Login.fxml"));
            Scene scene = new Scene(root, 400, 400);

```

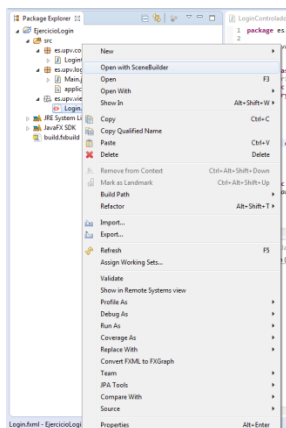
Crearemos ahora el paquete para incluir la clase controladora, cree el paquete `es.upv.controlador` y copie arrastrando a su interior el archivo `LoginControlador.java`.

El proyecto debe quedar como en la siguiente figura.

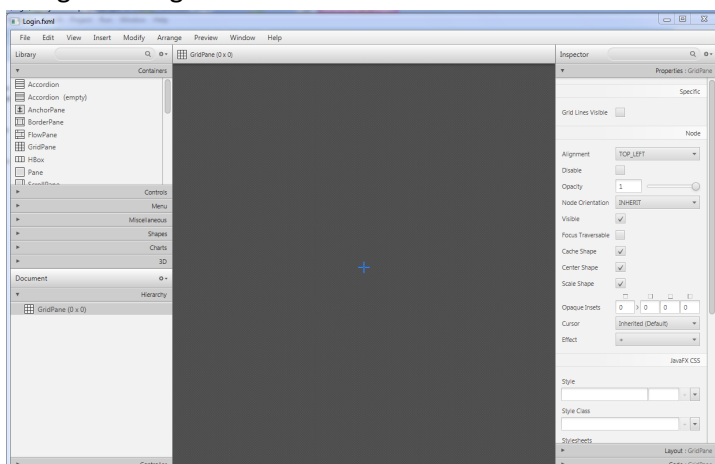


2. Crear la interfaz con Scene Builder

Ahora diseñaremos la interfaz de usuario, para ello marcamos el archivo `Login.fxml` y con el botón derecho del ratón en el menú contextual *Open with Scene Builder*.

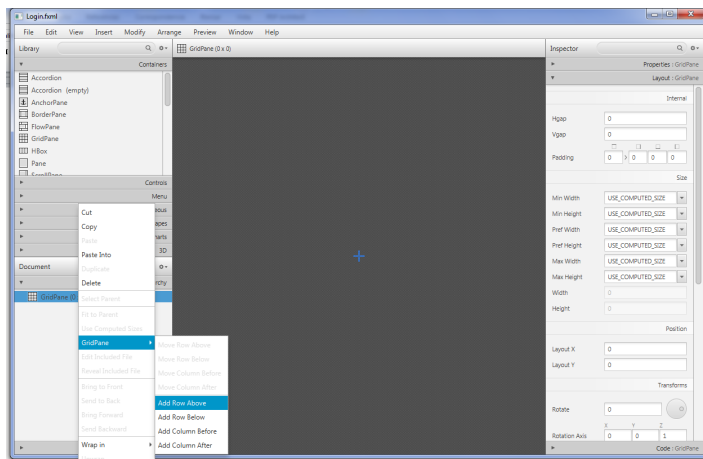


La siguiente figura muestra la interfaz de usuario de Scene Builder.

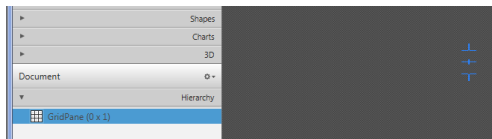


El layout *GridPane* representa una matriz de filas y columnas en las cuales se pueden situar componentes de la interfaz de usuario. Definiremos una matriz de 3 filas y 2 columnas (2x3).

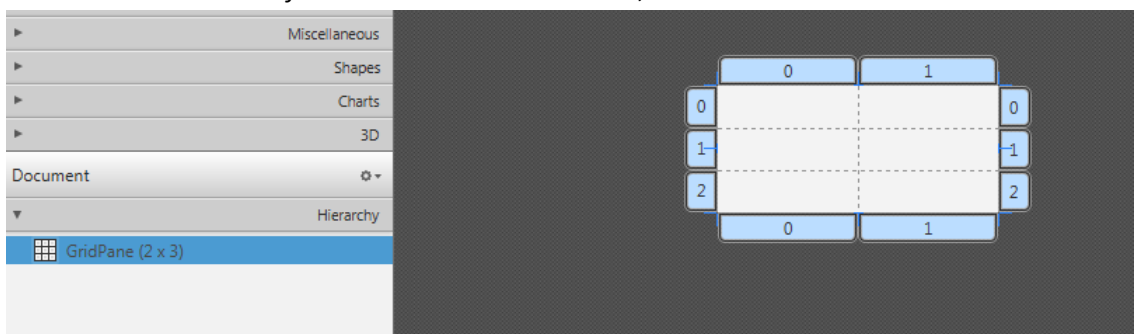
Para ello dentro de *Scene Builder* sobre el panel izquierdo *Hierarchy* usando el menú contextual con el botón derecho del ratón seleccionamos *Add Row Above*.¹



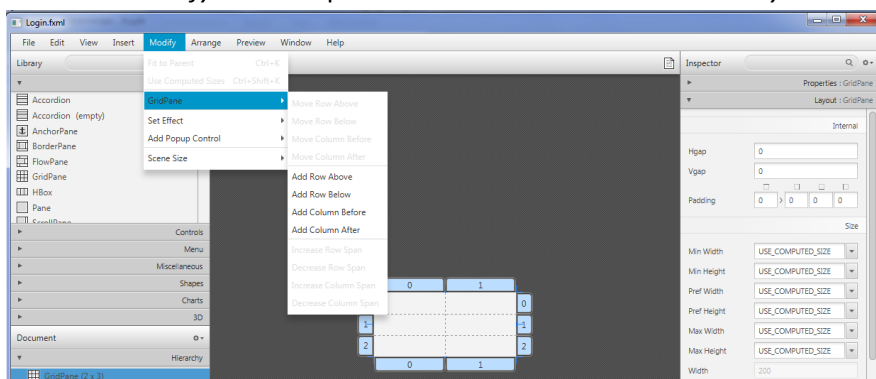
El efecto se muestra en la siguiente pantalla



Añada 2 veces más una fila a la matriz. Ahora seleccionamos de nuevo la matriz y usamos *Grid Pane* → *Add Column Before*. Esto lo hacemos 2 veces, ahora tenemos una matriz de 2 x 3.

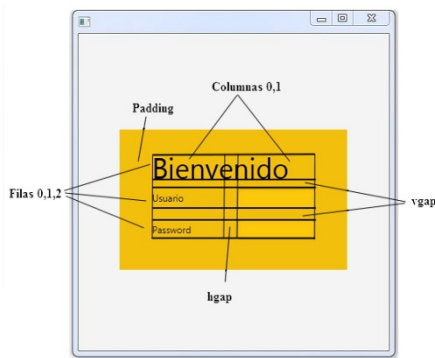


El mismo efecto de añadir columnas y fila se puede conseguir desde el menú principal utilizando *Modify* una vez que se tiene seleccionado en *Hierarchy* el elemento *GridPane*.

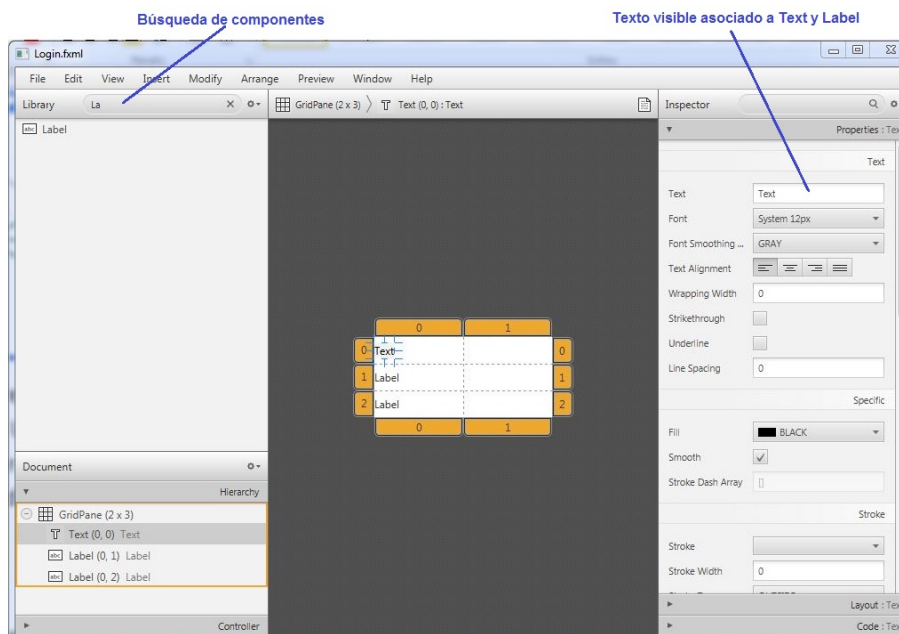


El objetivo es diseñar la siguiente ventana.

¹ Al estar la matriz vacía se podría haber usado también *Add Row Below*

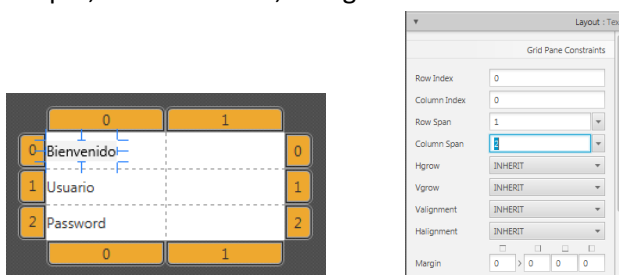


Empezaremos añadiendo desde Scene Builder algunos de los componentes de la interfaz de usuario. Usaremos un componente *Text* para el mensaje de bienvenida y dos componentes *Label* para las etiquetas *usuario* y *password*. Los seleccionamos en la paleta (puede usar la función de búsqueda en Library) y los dejamos respectivamente en las tres filas de la matriz.



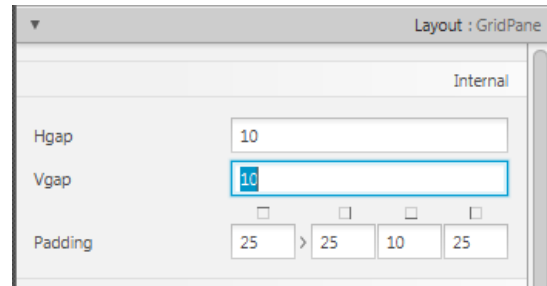
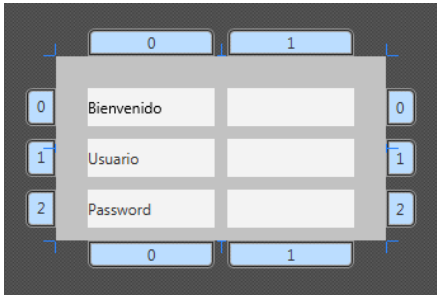
Escriba en la propiedad *Text* de cada uno de los componentes: *Bienvenido*, *Usuario* y *Password*.

Ahora cambiaremos algunas propiedades del componente *Text* (*Bienvenido*). Abra dentro del inspector la pestaña *Layout* y escriba en *Columnspan* 2. Esto permite que el componente pueda ocupar, si es necesario, la segunda columna.

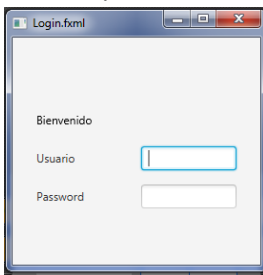


Ahora definiremos la separación entre filas (*Vgap*) y entre columnas (*Hgap*). Seleccione el *GridPane* y en *Layout* teclee los valores 10, 10 en *Vgap* y *Hgap*. Fijaremos también la el margen

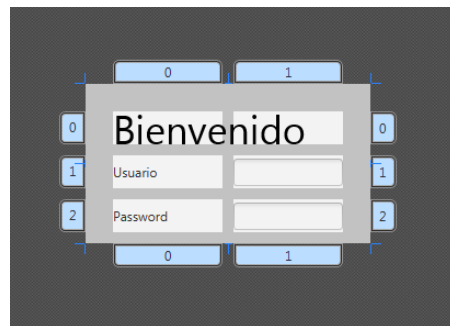
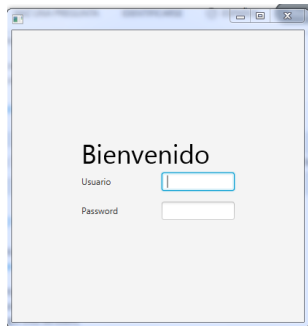
interno (padding) del gridPane. En Padding introduzca 25,25,10,25. El aspecto del componente es el de la siguiente figura.



Añadiremos ahora un componente para la entrada de texto (*TextField*) y otro para la entrada del password (*PasswordField*). Los arrastramos a sus respectivas posiciones. En Preview puede ver el aspecto de la interfaz.



Vamos a cambiar el tamaño de la fuente para el mensaje Bienvenido, de modo que pueda ocupar las dos columnas. Seleccione el componente Text y cambie en Properties -> Font el valor a 36.



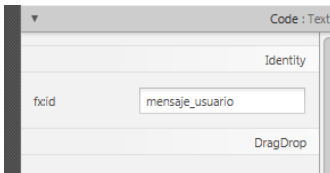
Lo último que queda por añadir es un botón y el texto que muestra el inicio de sesión. Añada una nueva fila al final de la matriz y sitúe dentro en la posición 1,3 un panel HBox y dentro de éste un botón. Ponga el alineamiento del panel a Bottom-Right y cambie el texto del botón a Iniciar. Para el mensaje que recibirá el usuario utilizaremos un componente Text que se ubicará en una nueva fila de la matriz. Sitúe el componente Text en la primera columna y última fila, cambie la propiedad Fill a color rojo, use para eso la paleta de colores.



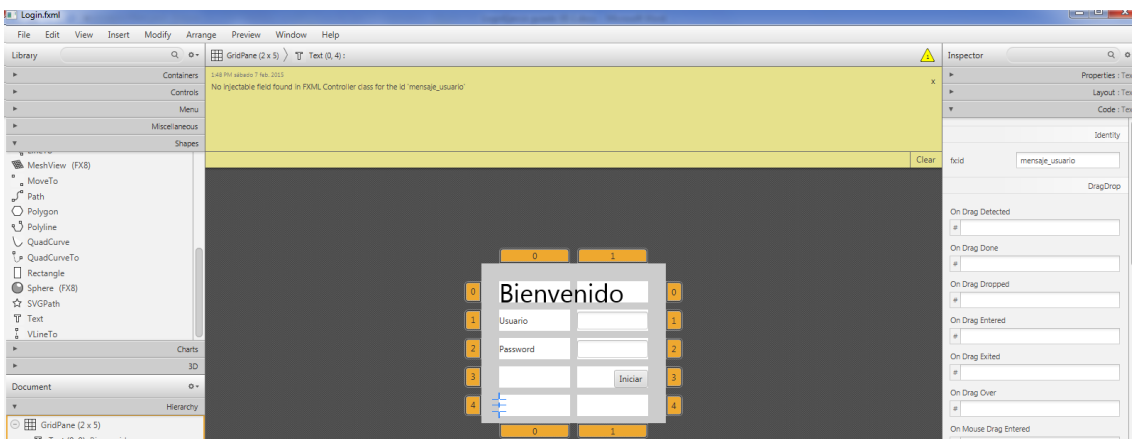
Cambie la propiedad Text del componente Text al string vacío.

Queremos, como hemos comentado anteriormente, que el programa responda con un mensaje cuando el usuario pulsa el botón iniciar.

Para que el texto cambie cuando se ejecuta el programa y después de pulsar el botón Inicial, el componente debe tener un ID, que luego será referenciado desde la clase controladora de la interfaz de usuario. Seleccione el componente y en la pestaña Code escriba en el campo fx:id, debajo de Identity, mensaje_usuario.



En Scene Builder pinchando sobre el icono amarillo puede ver un mensaje de advertencia. No se ha encontrado en la clase controladora un campo que pueda ser inyectado en la interfaz de usuario.



Ahora añadiremos funcionalidad al botón, definiendo un manejador de evento que se ejecutará cuando éste resulte pulsado.

Seleccione el botón Iniciar y abra la pestaña del Inspector Code y busque On Action, incluya allí un nombre descriptivo como por ejemplo “pulsadoIniciar”. Esto es un nombre de método que posteriormente tiene que ser implementado en el controlador.

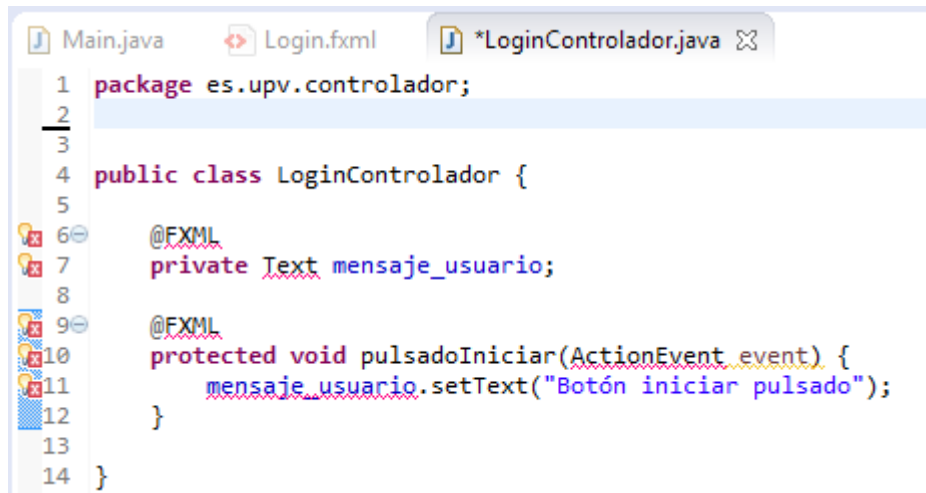


La figura anterior muestra el efecto en el archivo fxml de incluir el nombre del manejador de evento pulsadoIniciar.

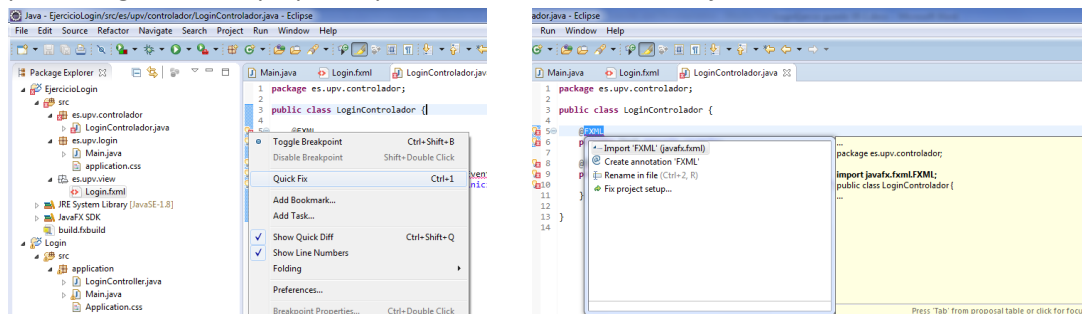
Si ahora ejecutamos la aplicación desde Eclipse, aparecerá un mensaje de error indicando que el código del manejador de evento no se ha encontrado.

`javafx.fxml: LoadException: Error resolving onMouseClicked="#pulsadoIniciar", either the event handler is not in the Namespace or there is an error in the script.`
`/C:/EclipseMars%20JavaFX/workspacePracticas/EjercicioLogin/bin/es/upv/view/Login.fxml:34`

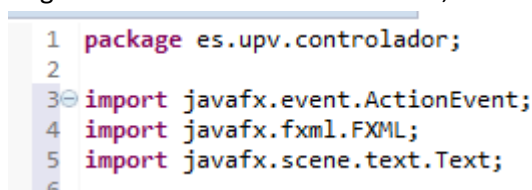
El código de manejo del evento lo situamos en la clase controladora, en Eclipse abra el archivo LoginControlador y teclee la siguiente información.



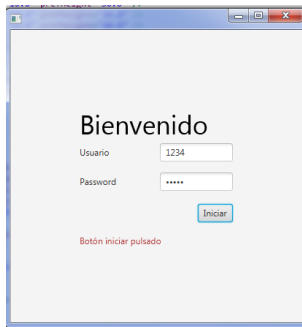
Los mensajes de error que aparecen están originados por no estar incluidas las librerías de javafx en este paquete. Seleccione la línea 5 y con el botón derecho del ratón marque Quick Fix y de las sugerencias importe la que contiene una librería JavaFX, en algunas situaciones le puede sugerir varios paquetes pero use el relacionado con javafx.



Haga lo mismo con todos los errores, al final quedará la siguiente lista de importaciones:



Ahora podemos ejecutar la aplicación desde Eclipse, seleccione la clase Main.java y pulse Run.



Si queremos que la ventana tenga un título tenemos que añadir el siguiente código (`primaryStage.setTitle("Ejemplo de Login")`) en el Main.java.

```

10 public class Main extends Application {
11     @Override
12     public void start(Stage primaryStage) {
13         try {
14             GridPane root = (GridPane)FXMLLoader.load(getClass().getResource("/es/upv/view/Login.fxml"));
15             Scene scene = new Scene(root,400,400);
16             scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
17             primaryStage.setTitle("Ejemplo de Login");
18             primaryStage.setScene(scene);
19             primaryStage.setResizable(false);
20             primaryStage.show();
21         } catch (Exception e) {
22             e.printStackTrace();
23         }
24     }
25
26     public static void main(String[] args) {
27         launch(args);
28     }
29 }
30

```

La instrucción `primaryStage.setResizable(false)` impide el redimensionamiento del formulario.