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 de bienvenida cada vez que se pulsa el botón *Iniciar* del formulario.
La apariencia final del formulario es la que se muestra en la figura.

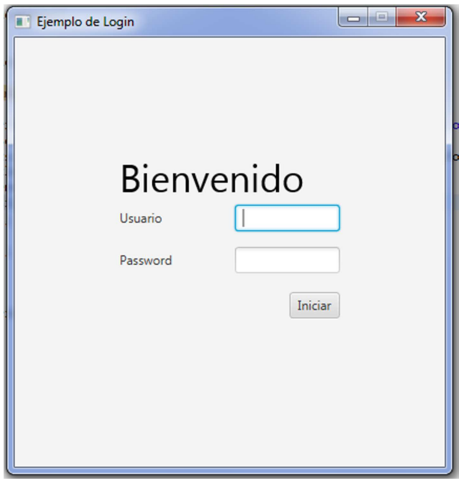


Ilustración 1: Interfaz final

1. Crear el nuevo proyecto JavaFX con Netbeans

Dentro del entorno NetBeans utilice *File -> NewProject* y seleccione un proyecto *JavaFX FXML Application*

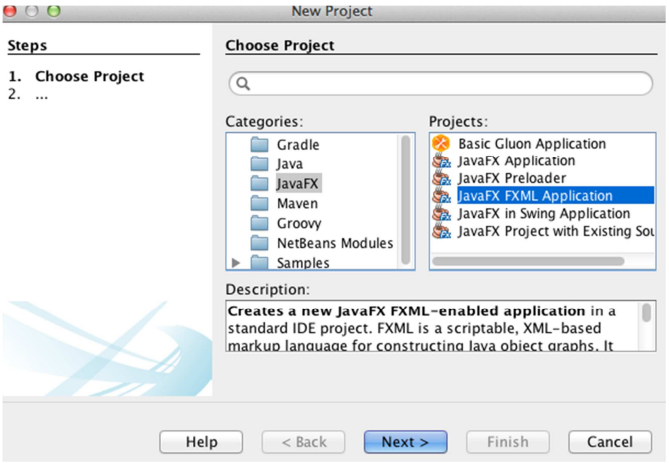


Ilustración 2: Tipos de proyectos

Pulse el botón *Next* y en la siguiente pantalla ponga un nombre al proyecto, *EjercicioLogin*. Cambie el nombre por defecto del fichero FXML *FXMLDocument* por *FXMLLogin* y después pulse *Finish.*

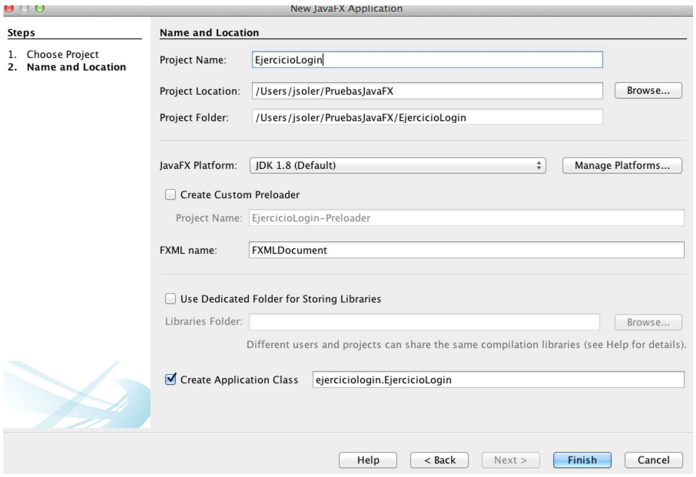


Ilustración 3: Propiedades del proyecto a crear

NetBeans ha creado la estructura del proyecto, tal como se muestra en la siguiente figura.

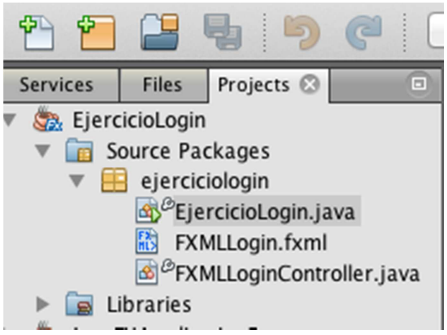


Ilustración 4: Estructura del proyecto

Por defecto Netbeans ha creado el fichero XML con un *Anchorpane* que contiene *un Button* y un *Label* . Los archivos recién creados con el asistente tienen la siguiente información:

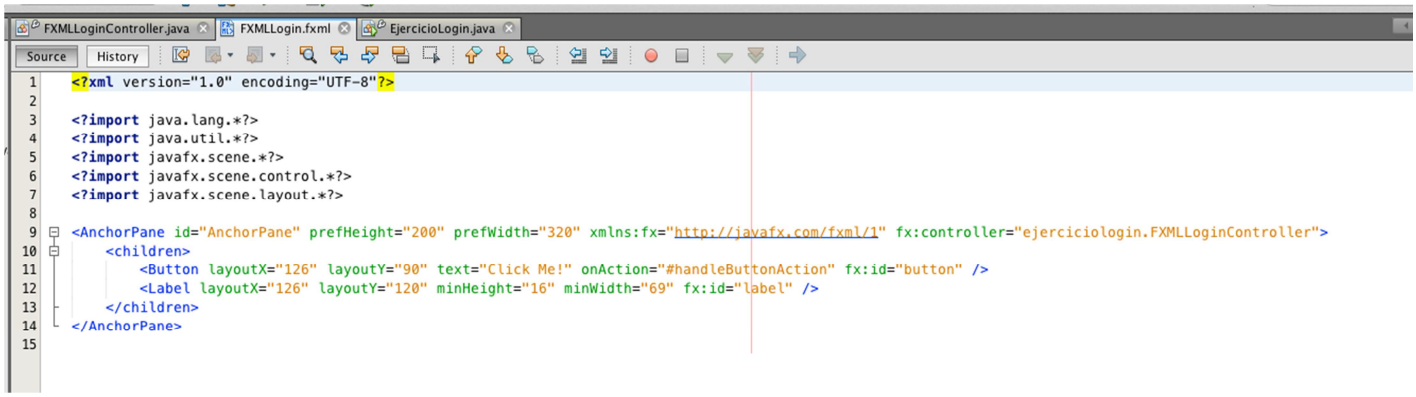


Ilustración 5: Fichero FXML

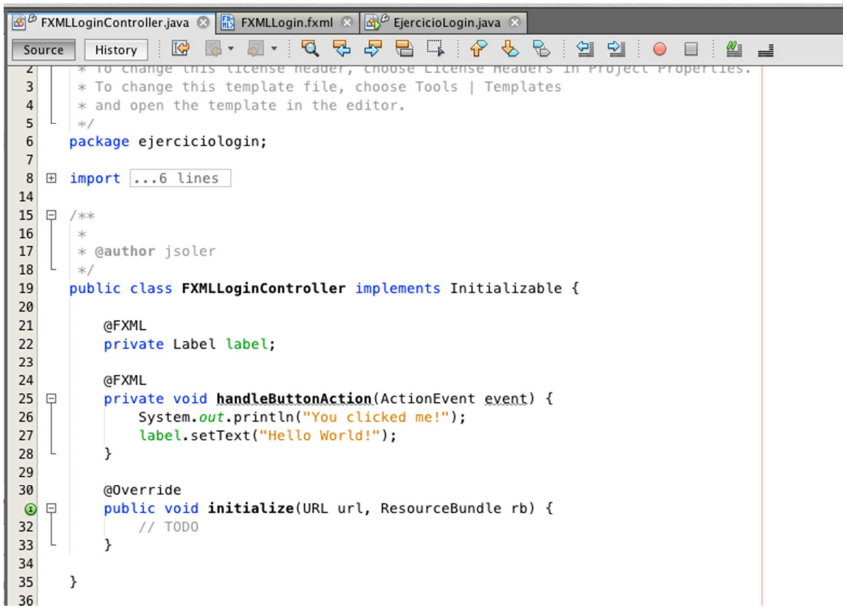


Ilustración 6: Clase de Java "Controlador"

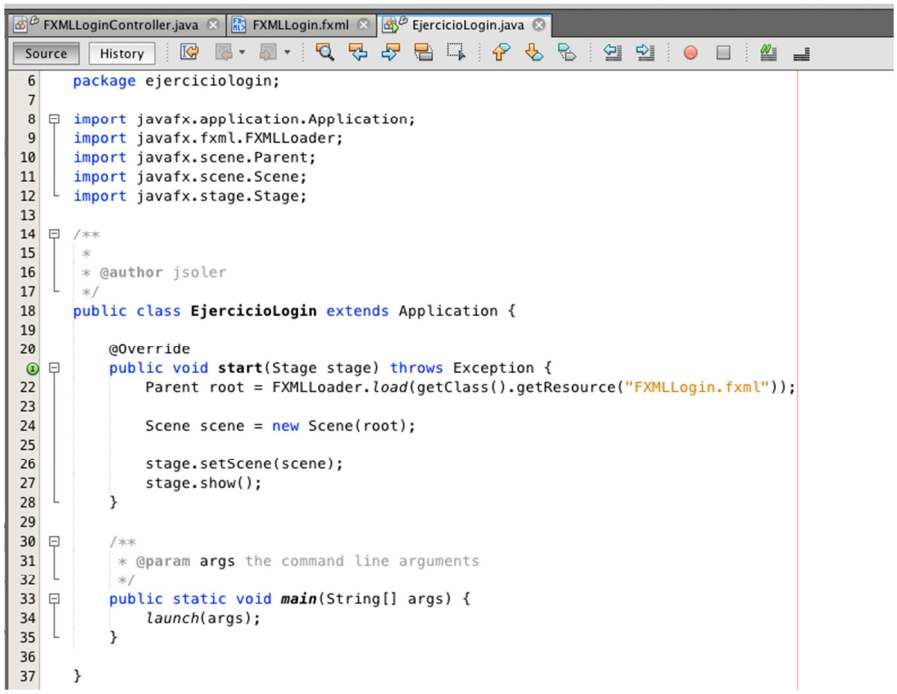


Ilustración 7: Clase principal de la aplicación

Podemos ejecutar el proyecto creado con la opción de menú *Run Project.*

2. Crear la interfaz con Scene Builder

Para crear la interfaz vamos a reutilizar el fichero FXML generado por defecto. El fichero FXMLLoginController.java no nos va a resultar útil, antes de seguir lo eliminaremos del explorador del proyecto.

Ahora diseñaremos la interfaz de usuario, para ello marcamos el archivo `FXMLLogin.fxml` y con el botón derecho del ratón en el menú contextual *Open* (ó simplemente doble click)

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

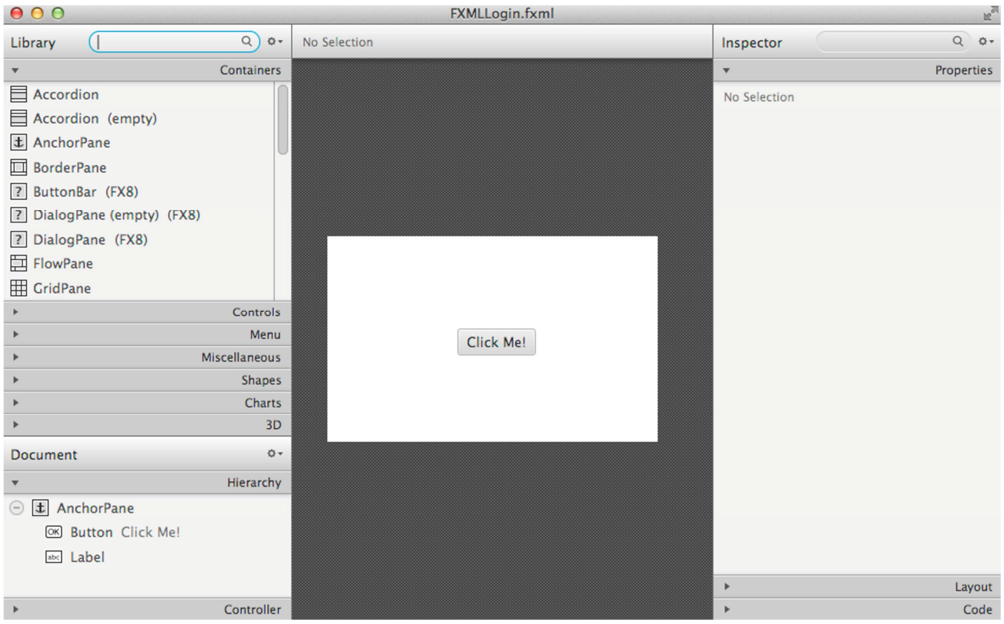


Ilustración 8: Scene Builder

Ahora vamos a eliminar todos los elementos gráficos del fichero (AnchorPane, Button, Label) y empezaremos a diseñar la interfaz de login.

Vamos a empezar añadiendo un layout *GridPane* que 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 seleccionaremos un GridPane de la librería de controles y lo arrastraremos a la zona de trabajo. Por defecto se creará un grid de 2x3.

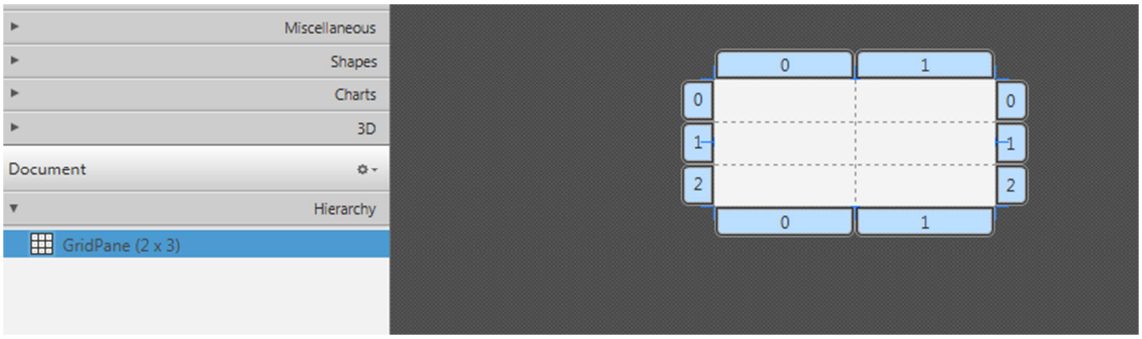


Ilustración 9: GridPane

Si necesitamos más filas o columnas, sobre el panel izquierdo *Hierarchy* y usando el menú contextual (el botón derecho del ratón) *nos aparecerán todas las opciones necesarias*. De la misma manera, en la zona de trabajo podemos seleccionar una fila o columna y con el botón derecho del ratón nos aparecerán las opciones para modificar el GridPane.

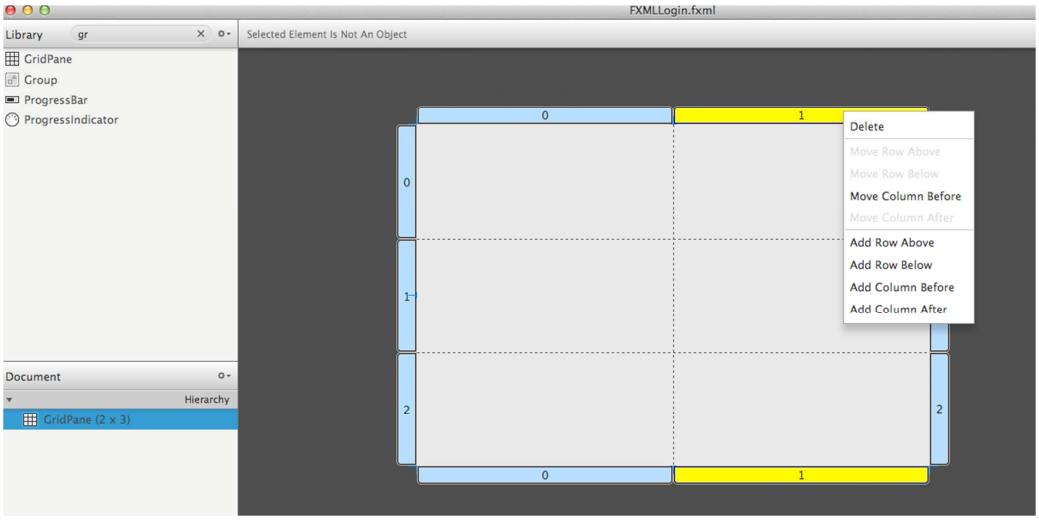
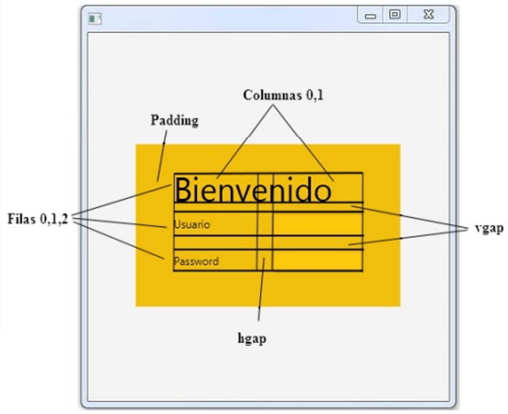
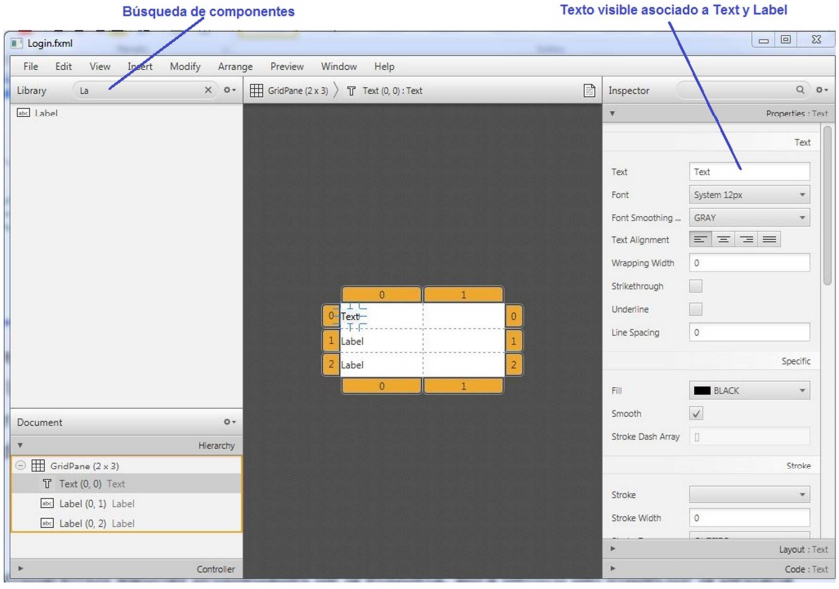


Ilustración 10: Modificar las características del GridPane

El objetivo ahora es diseñar la siguiente ventana.

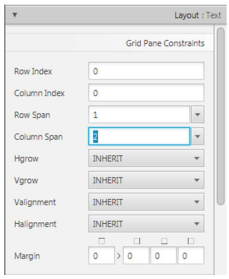


Empezaremos añadiendo desde Scene Builder algunos de los componentes de la interfaz de usuario. Usaremos un componente *Text* para el mensaje de bienvenida y dos componentes *Label* para las etiquetas *usuario* y *password*. Los seleccionamos en la paleta (puede usar la función de búsqueda en Library)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

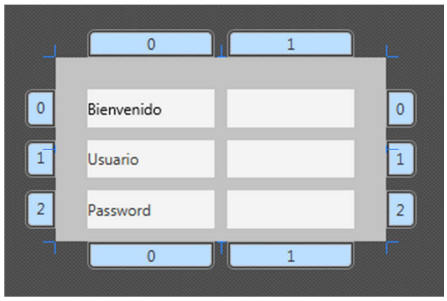


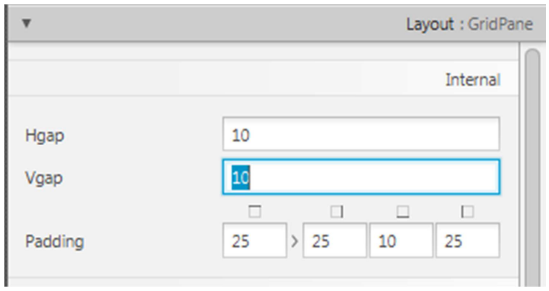
Escriba en la propiedad Text de cada uno de los componentes: Bienvenido, Usuario y Password.

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

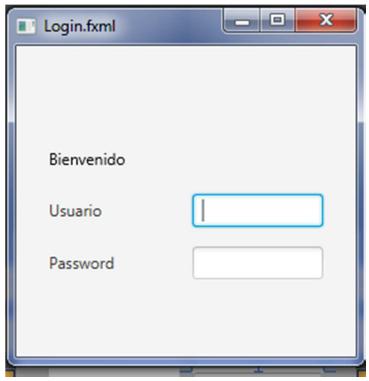


Ahora definiremos la separación entre filas (Vgap) y entre columnas (Hgap). Seleccione el GridPane y en Layout teclee los valores 10, 10 en Vgap y Hgap. Fijaremos también la el margen interno (padding) del gridPane. En Padding introduzca 25,25,10,25. El aspecto del componente es el de la siguiente figura.

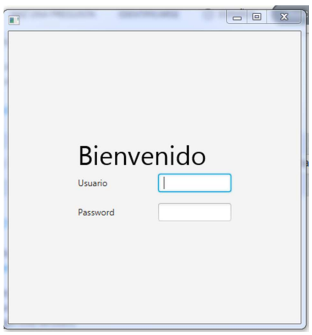


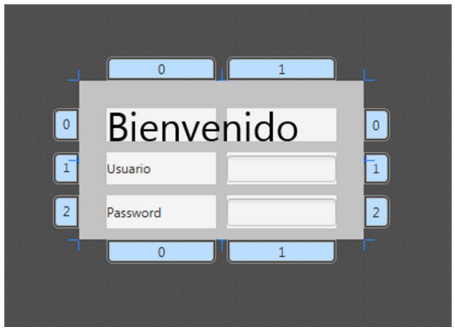


Añadiremos ahora un componente para la entrada de texto (*TextField*) y otro para la entrada del password (*PasswordField*). Los arrastramos a sus respectivas posiciones. En Preview puede ver el aspecto de la interfaz.



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, usando la paleta de colores.

Cambie la propiedad Text del componente Text a la cadena vacía.

Acciones necesarias para añadir el comportamiento

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

Ilustración 11: identificador del componente Text

Vamos a hacer lo mismo en el TextField en el que se introduce el usuario, pondremos en el campo fx:id ***texto_usuario***

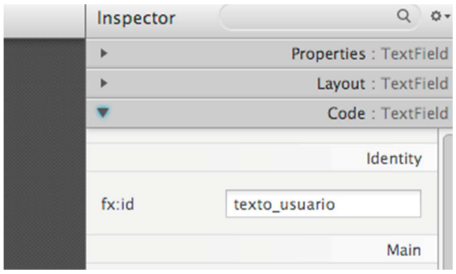


Ilustración 12: Identificador del TextField

Para finalizar 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 será el nombre de método que posteriormente tiene que ser implementado en la clase controlador.

Ilustración 13: Asociar un manejador en Scene Builder y en el fichero FXML

La figura anterior muestra el efecto en el archivo fxml de incluir el nombre del manejador de evento *pulsadoIniciar*.
Antes de abandonar SceneBuilder es necesario salvar el fichero con el que estamos trabajando.

3. Crear la clase de Java que se asocia como controlador

El código de manejo del evento lo situamos en la clase controladora que vamos a crear de manera automatica en NetBeans.
Para ello seleccionaremos en el explorador del proyecto el fichero FXMLLogin y con el boton derecho de ratón invocaremos a Make Controller

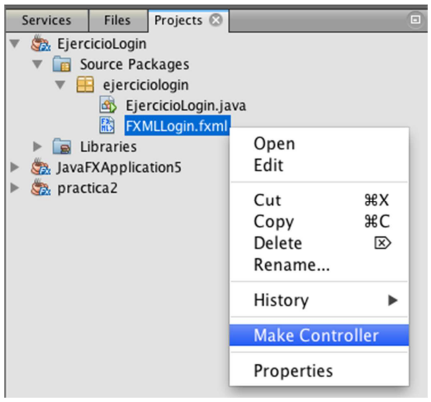


Ilustración 14: Generación automática de la clase controlador

Se crea automaticasmente el fichero FXMLLoginController, si ya existe se actualiza con los cambios introducidos en el fichero FXML. Este es el resultado:

```
6   package ejerciciologin;
7
8   import ...7 lines
15
16  /**
17   * FXML Controller class
18   *
19   * @author jsoler
20   */
21  public class FXMLLoginController implements Initializable {
22
23      @FXML
24      private TextField texto_usuario;
25      @FXML
26      private Text mensaje_usuario;
27
28      /**
29       * Initializes the controller class.
30       */
31      @Override
32      public void initialize(URL url, ResourceBundle rb) {
33          // TODO
34      }
35
36      @FXML
37      private void pulsadoIniciar(ActionEvent event) {
38      }
39
```

Ilustración 15: Código generado

Como veis aparecen debajo de las anotación @FXML los elementos que en el fichero FXML tienen asignado un valor en la campo fx:id asi como el método que hemos indicado en el campo onAction del boton.

Para que la aplicación reaccione al click del boton tendremos que añadir la siguiente linea de código dentro del metodo pulsadoIniciar:

```
35
36      @FXML
37      private void pulsadoIniciar(ActionEvent event) {
38          mensaje_usuario.setText("Bienvenido "+ texto_usuario.getText());
39      }
40
```

Ilustración 16: Código del manejador de eventos

Ahora podemos ejecutar la aplicación desde NetBeans mediante *Run Project*

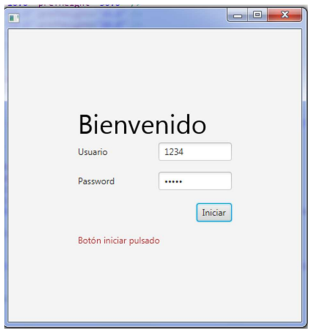


Ilustración 17: Resultado final

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

```
20      @Override
        public void start(Stage stage) throws Exception {
22          Parent root = FXMLLoader.load(getClass().getResource("FXMLLogin.fxml"));
23
24          Scene scene = new Scene(root);
25
26          stage.setScene(scene);
27
28          stage.setTitle("Login");
29          stage.setResizable(false);
30
31          stage.show();
32      }
```

Ilustración 18: Cambios en la clase principal

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