

Primera Aplicación. Parte I

Vamos a desarrollar nuestra aplicación JavaFX. Dividiremos el desarrollo en tres partes.

- a) Creación de la interfaz
- b) Conexión a BD
- c) Control de eventos

En esta primera parte, para la creación de la interfaz, veremos que se utiliza **FXML** y **CSS**. **FXML** es un lenguaje de marcado similar a los que utilizáis en la asignatura de Lenguajes de Marcas. Con el describimos la estructura de la interfaz. Mediante **CSS** podemos modificar el aspecto y comportamiento de esta. Hay que tener en cuenta que no es un CSS genérico, como el utilizado en HTML. Muchas de las sentencias coinciden sin más que anteponerles el prefijo `-fx-`

Así, lo que en CSS sería:

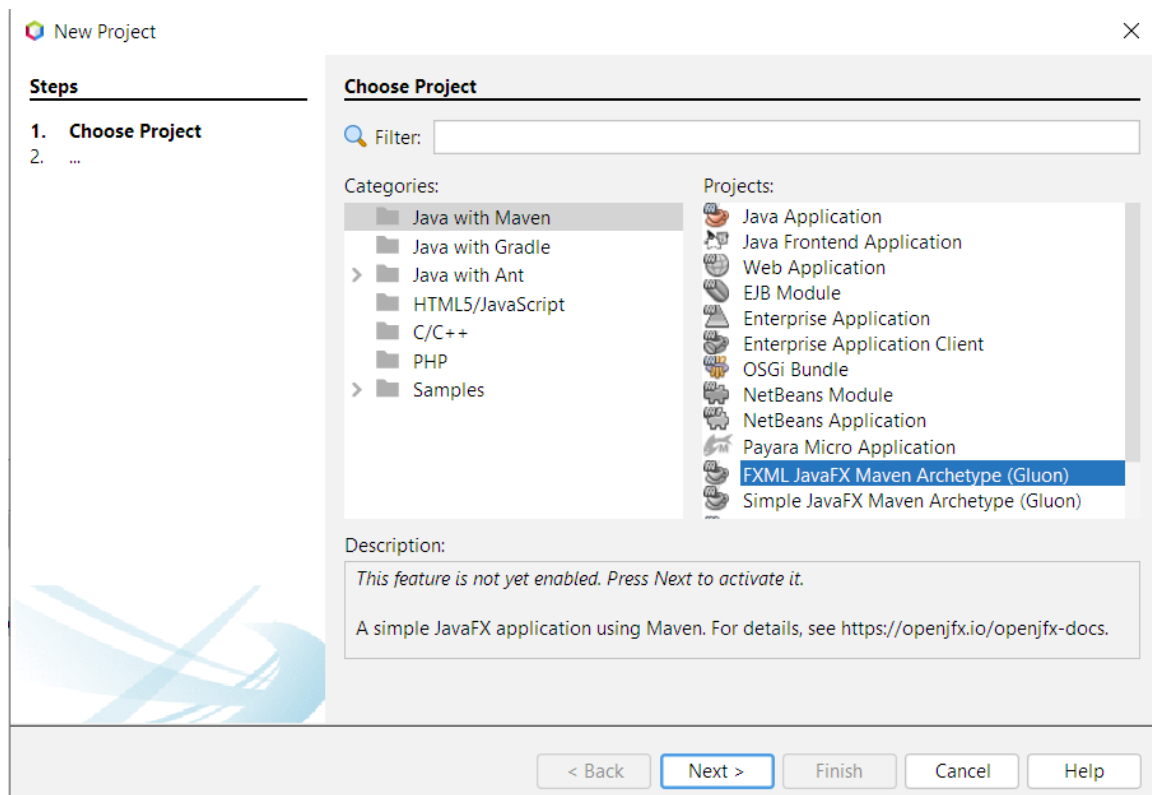
```
background-color: goldenrod;
```

Aquí lo escribiríamos como:

```
-fx-background-color: goldenrod;
```

Podemos codificar todo a mano directamente en el fichero o ficheros **.xml** de la aplicación (aquí sería: **primary.fxml**) o bien diseñarlo visualmente en **Scene Builder**. Los cambios hechos en un lado se reflejarán en el otro.

Creamos un nuevo proyecto con Maven.



En la siguiente pantalla:

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: pruebaFX2

Project Location: s:\abell\Dropbox (Personal)\1-Prog\NetBeansProjects\AxO Browse...

Project Folder: pbox (Personal)\1-Prog\NetBeansProjects\AxO\pruebaFX2

Artifact Id: pruebaFX2

Group Id: es.amador

Version: 1.0

Package: es.amador:pruebafx2 (Optional)

Additional Creation Properties:

Key	Value
javafx-version	13
javafx-maven-plugin-version	0.0.4
add-debug-configuration	Y

< Back Next > **Finish** Cancel Help

Rellenar los campos:

Project Name

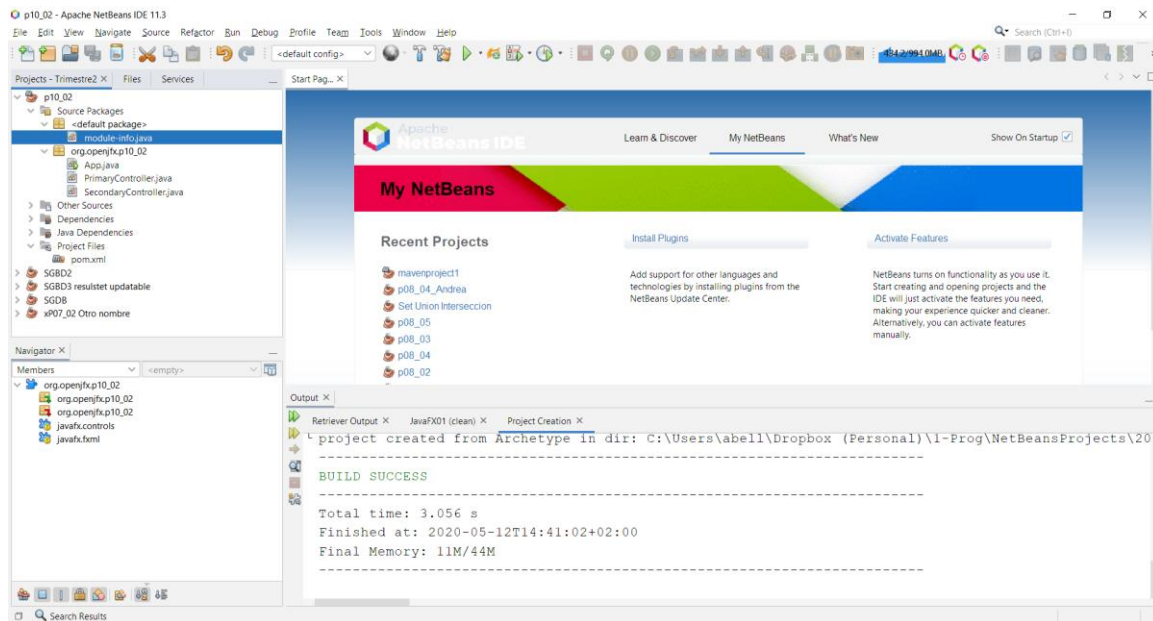
GroupID

Version

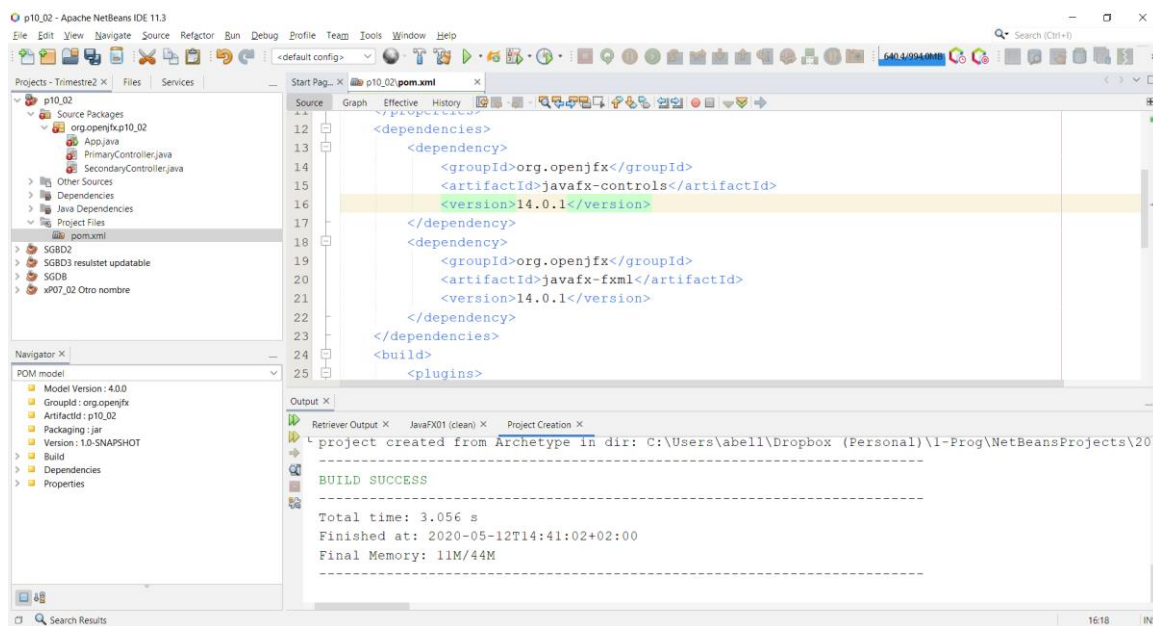
Con valores apropiados para la aplicación. Cambiar, si es necesario, Project Location.

Es interesante poner como valores de javafx-version y javaz-maven-plugin-version los últimos disponibles (siempre que no den problemas). Para esta aplicación vamos a poner los valores 16 y 0.0.5 repectivamente

Si nos fijamos en los ficheros que se crean. Hay uno denominado **module-info.java**. Este fichero solo es necesario para proyectos modulares (no es el caso), así que podemos eliminarlo.



El fichero pom.xml es donde se encuentra toda la información para que Maven realice su trabajo (compilación, ejecución, empaquetado, ...). Ahí vemos, por ejemplo, la referencias a **javafx-controls** y **javafx-fxml**, se correspondan con la versión 16 (o una subversión de ella).



Y que el plugin de **javafx-maven-plugin** se corresponde con la versión 0.0.5

The screenshot displays the Apache NetBeans IDE interface. The main editor window shows the `p10_02.pom.xml` file with the following XML content:

```

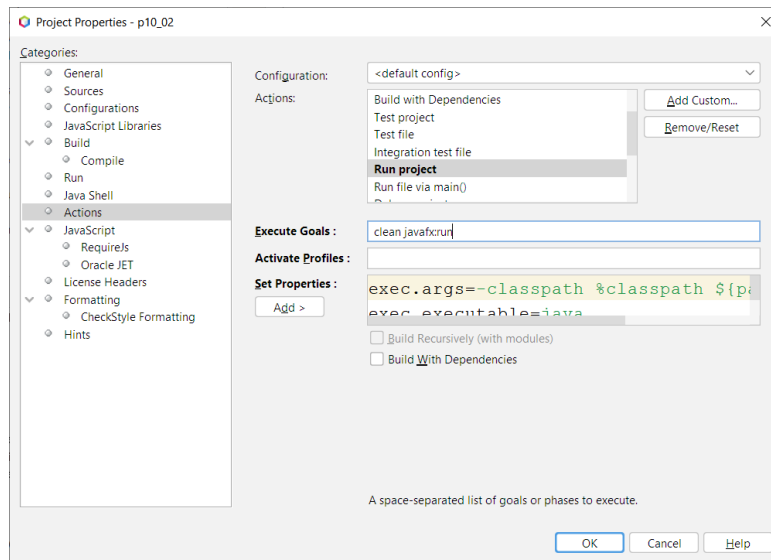
<?xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.openjfx</groupId>
  <artifactId>javafx-maven-plugin</artifactId>
  <version>0.0.4</version>
  <configuration>
    <release>11</release>
  </configuration>
  <plugin>
    <groupId>org.openjfx</groupId>
    <artifactId>javafx-maven-plugin</artifactId>
    <version>0.0.4</version>
    <configuration>
      <mainClass>org.openjfx.p10_02.App</mainClass>
    </configuration>
  </plugin>
</plugins>
</project>
  
```

The left sidebar shows the project structure for `p10_02`, including source packages, dependencies, and project files. The bottom output window shows the build log:

```

Retriever Output X   JavaFX01 (clean) X   Project Creation X
-----
project created from Archetype in dir: C:\Users\abell\Dropbox (Personal)\1-Prog\NetBeansProjects\20
BUILD SUCCESS
-----
Total time: 3.056 s
Finished at: 2020-05-12T14:41:02+02:00
Final Memory: 11M/44M
  
```

Si el proyecto no se ejecuta, hay que acceder a las propiedades del proyecto y la entrada **Actions->Run project->Execute Goals**, escribir: **clean javafx:run**



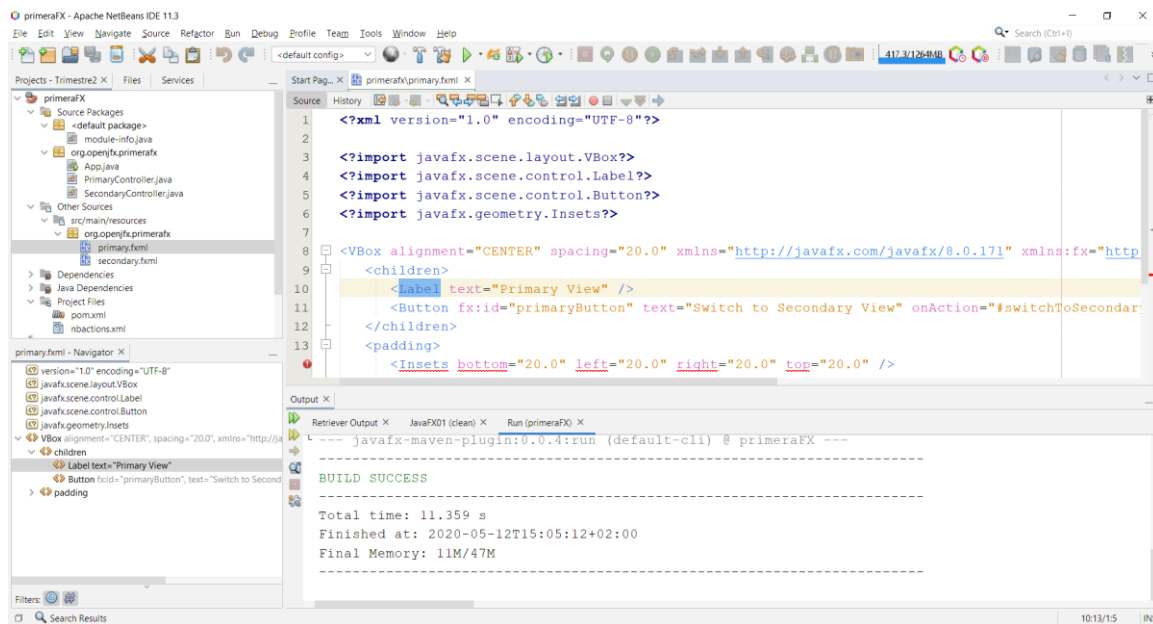
Comenzamos la edición del código. Para visualizar en NetBeans el código FXML, hay que pulsar con el botón derecho sobre el fichero, en este caso sobre **primary.fxml**, y escoger la opción **Edit**

Como solo vamos a utilizar una única ventana, podemos eliminar los ficheros:

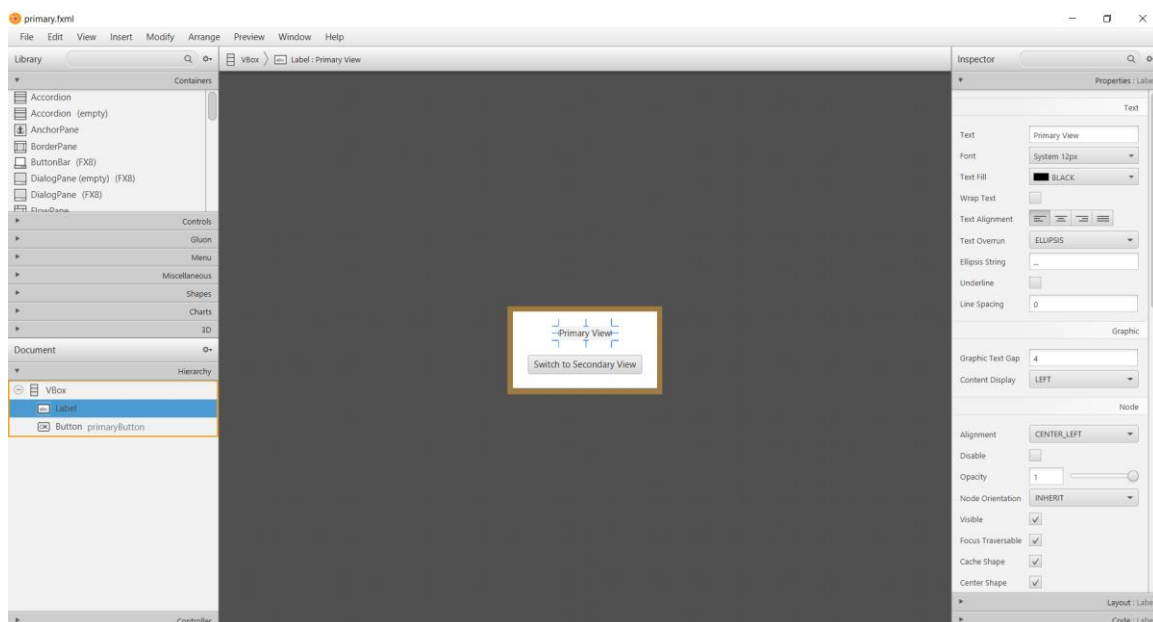
secondary.fxml

secondaryController.java

Este es el contenido de primary.fxml:



Si en lugar de escoger la opción **Edit**, escogemos **Open**, o bien hacemos doble click sobre el fichero, se abrirá Scene Builder, desde el que podemos realizar las modificaciones sobre el FXML de modo visual:



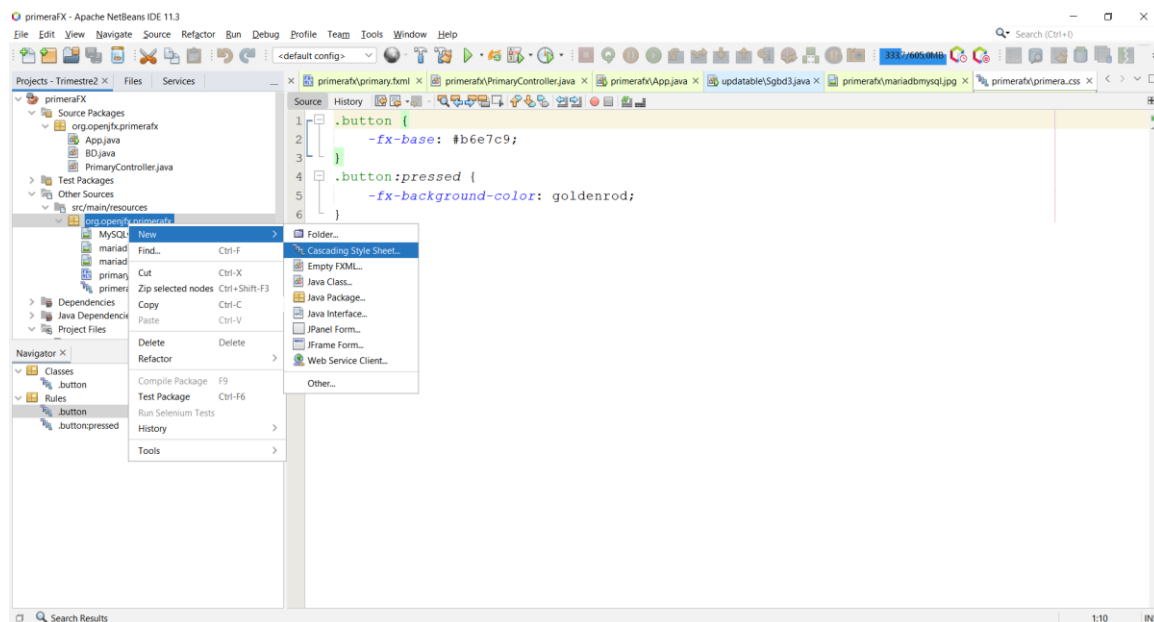
Si nos fijamos en la parte inferior izquierda, nos indica que nuestra interfaz está formada por un contenedor (VBox), sobre el que hemos colocado una etiqueta (Label) y un botón (Button).

Los componentes, similares a los que habéis visto en Lenguajes de Marcas, como áreas de texto, botones, casillas de verificación, etc., necesitan un contenedor que los albergue (en este caso un VBox). Los contenedores pueden anidarse. Nosotros vamos a utilizar un contenedor denominado **BorderPane**, que divide el área de trabajo en cinco zonas: top, bottom, center, left, right. Dedicaremos la zonas:

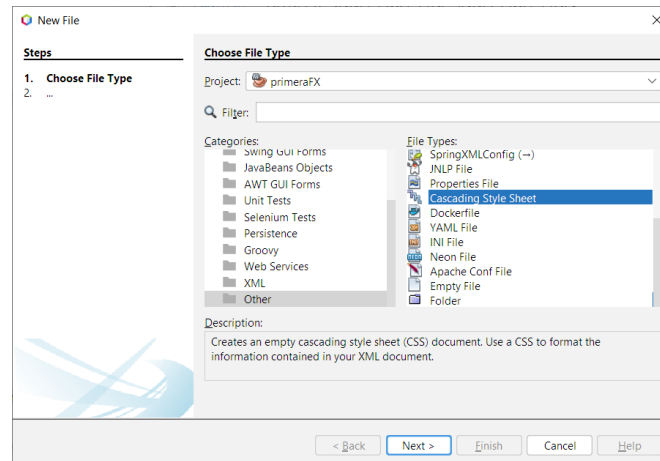
- a) superior a poner un logo y un encabezado
- b) inferior a mostrar mensajes del sistema y un botón para resetear los datos introducidos
- c) derecha para botones que ejecutarán acciones sobre la BD
- d) el izquierdo no lo usaremos y usaremos la zona que nos deja junto a la central para la introducción de datos

Sobre cada una de estas zonas situaremos un contenedor anidado, a los que asignaremos diferentes colores con la función de que se aprecien claramente las secciones. Por último sobre cada panel colocaremos los controles necesarios.

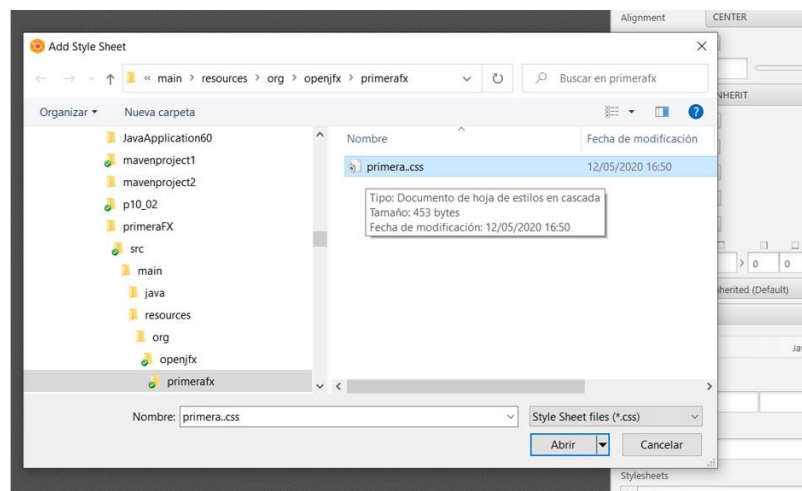
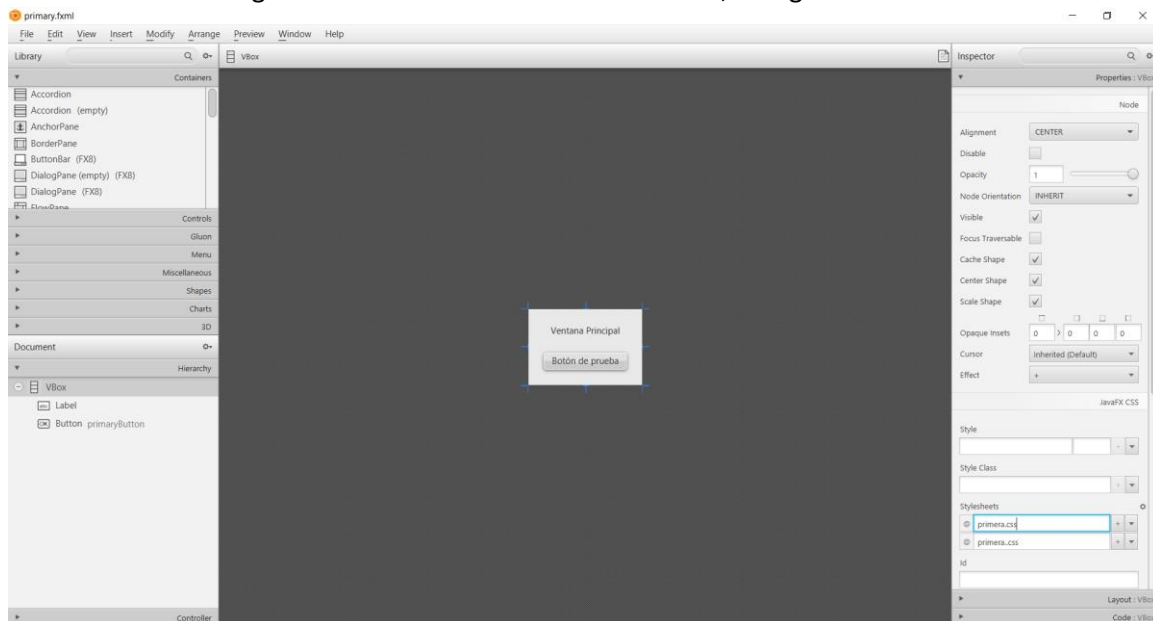
Vamos a crear un fichero CSS para ver como se aplicaría sobre nuestro diseño. En este caso nos limitamos a cambiar el aspecto de los botones. Colocados sobre **Other Sources->src/main/resources/org.openjfx.primerafx** (evidentemente esto depende del nombre que le hayáis dado a vuestro proyecto), botón derecho y escogemos **New->Cascading Style Sheet**. El contenido serían las líneas que véis a continuación:



Si no aparece la opción de Cascading Style Sheet, escogemos **Other** y se nos abrirá la siguiente ventana, donde escogeremos la opción correcta:



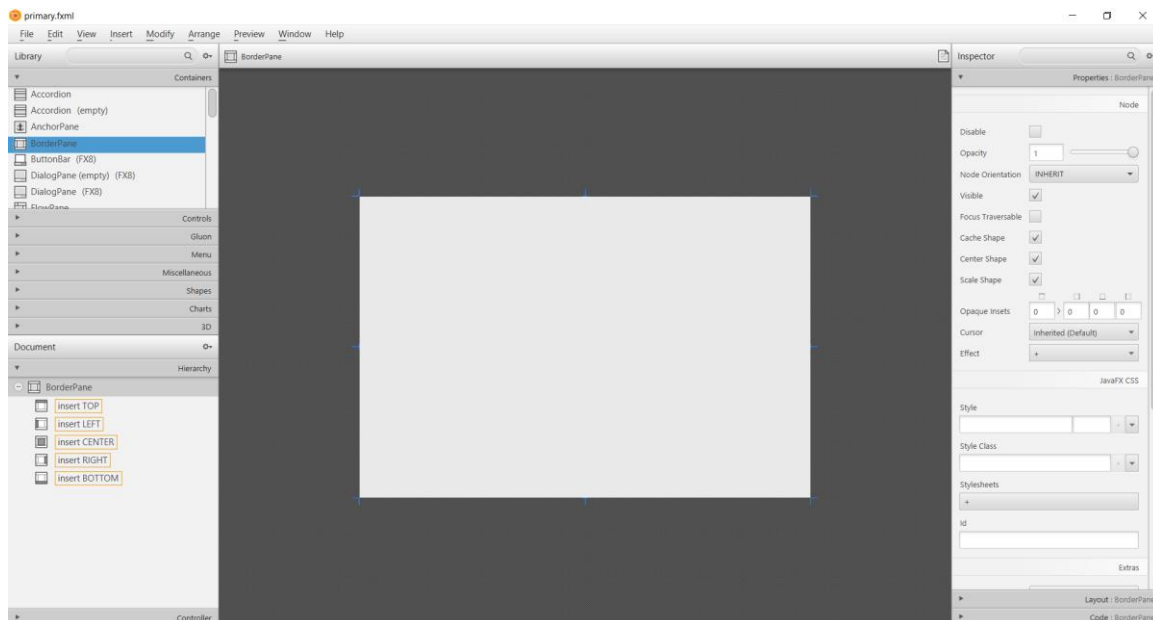
En Scene Builder, importamos el CSS pulsando el botón + en la entrada **Stylesheets** que aparece remarcada en la imagen. Se muestra un selector de ficheros, escogemos nuestro CSS



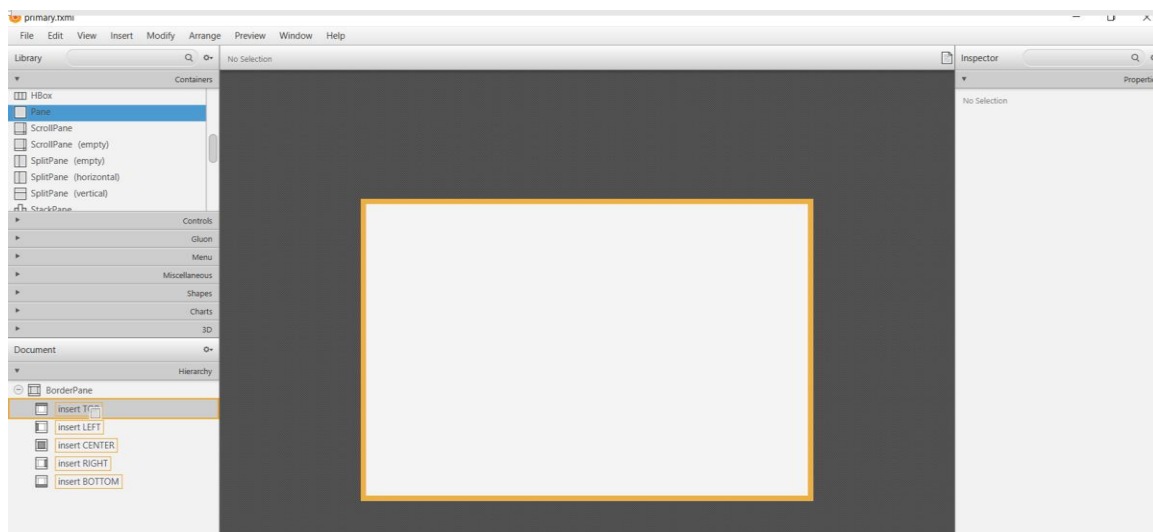
Y podremos ver como cambia el aspecto del botón.

Ahora vamos a eliminar todos los componentes que se generaron automáticamente (contenedores y controles). Pulsando botón derecho sobre el componente raíz y escogemos Delete.

Y comenzamos un diseño desde cero. En la parte izquierda, dentro de Containers, escogemos **BorderPane** y lo arrastramos a la parte central de la aplicación:

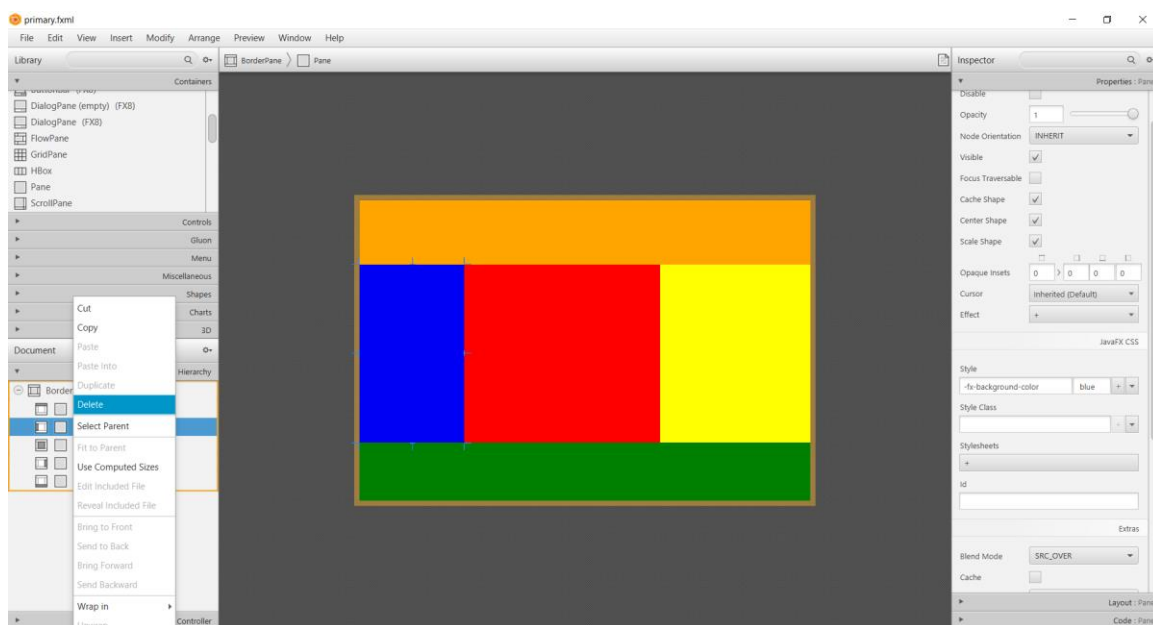
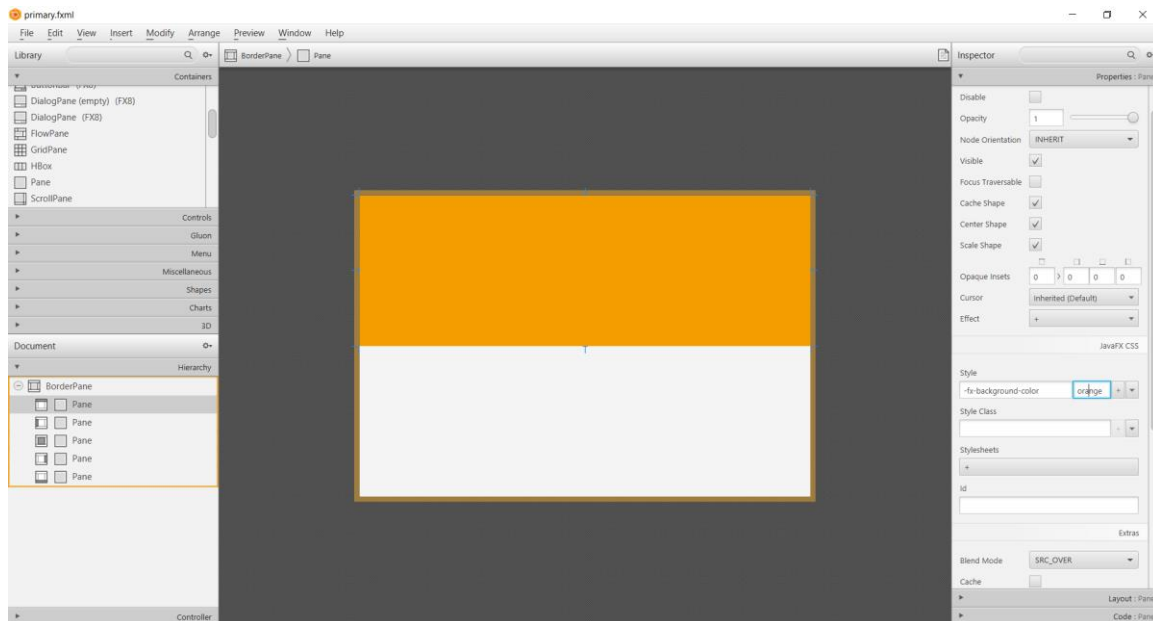


Podemos observar en la parte inferior izquierda como se han generado las cinco zonas. Colocaremos sobre cada una de ellas, salvo la izquierda, un contenedor de tipo **Pane**. Para ello lo seleccionamos en la sección de **Containers** (parte superior izquierda) y lo arrastramos a la zona correspondiente.



Una vez colocado, podemos redimensionarlo mediante arrastrar y soltar o bien poniendo manualmente los valores dentro de la sección **Layout** (en la parte derecha, debajo de **Properties**). Para que se observen claramente cada uno de los paneles, vamos a asignarles un color a cada uno. Así será más fácil darles tamaño y ver como quedaría la interfaz.

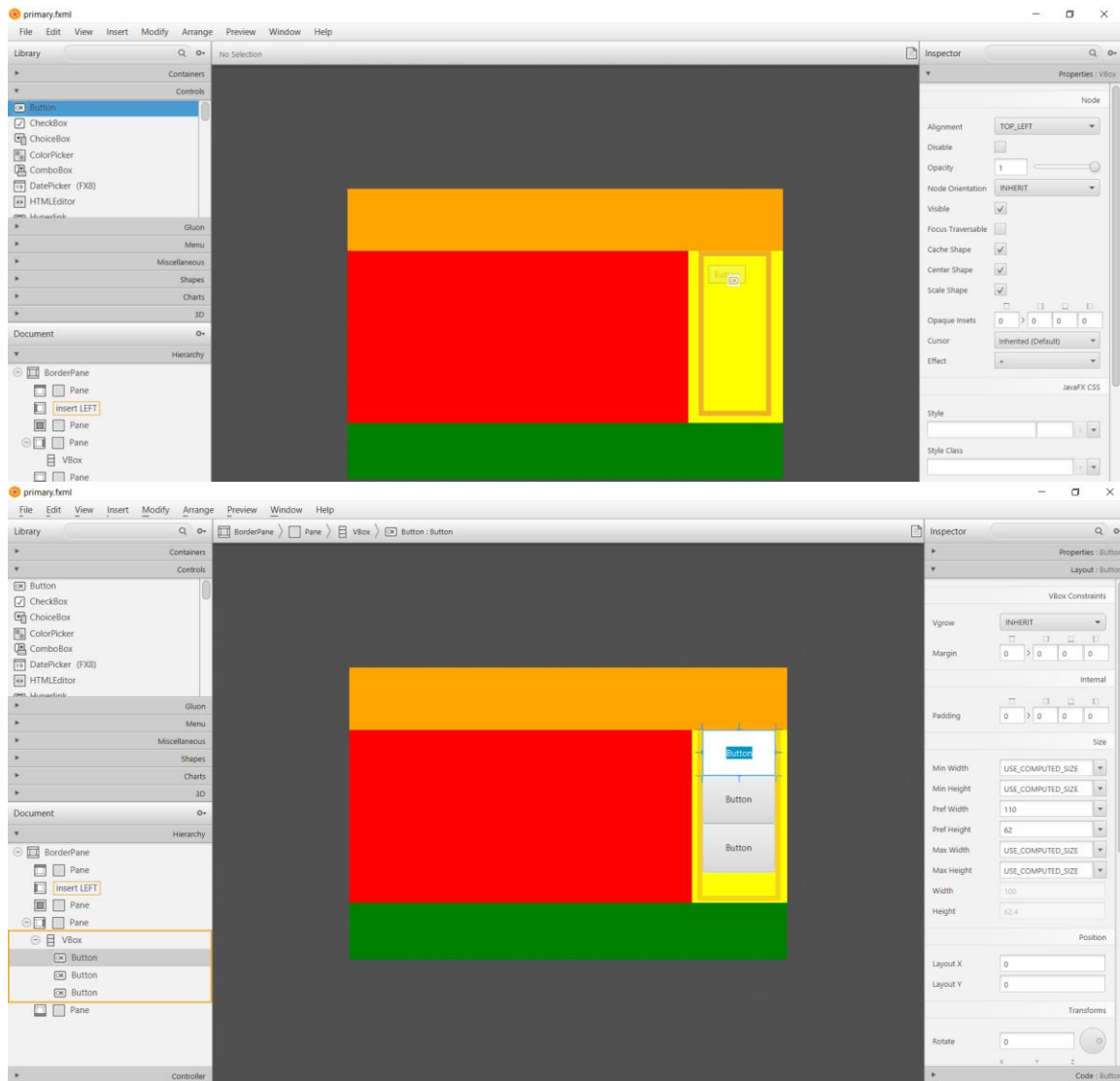
En la parte derecha, en la sección **Properties**, tenemos una entrada llamada **Style**. Si pulsamos sobre el selector (flecha hacia abajo), nos aparecen diferentes estilos CSS que podemos aplicar. Escogemos **-fx-background-color** y ponemos un nombre de color o un valor hexadecimal, que podemos obtener desde cualquier aplicación de color-picker. Esto también podríamos hacerlo incluyéndolo en el CSS, claro está. Bien, vamos a darle un color a cada zona y redimensionarlas hasta que nos quede algo parecido a lo mostrado.



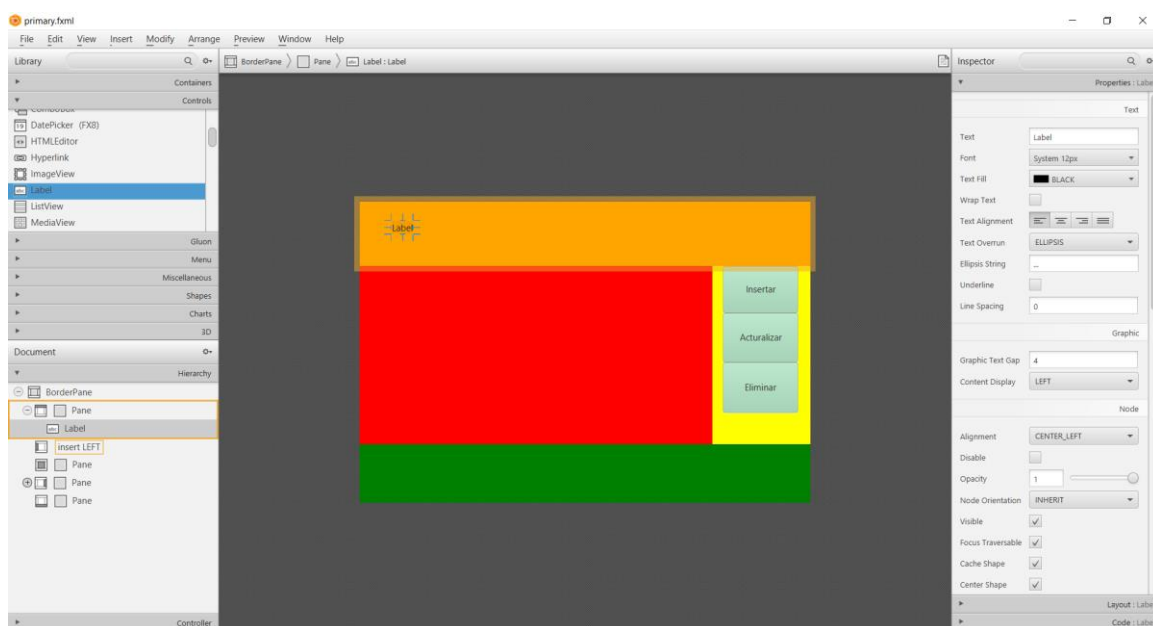
Sobre el panel derecho vamos a colocar otro contenedor. Un VBox que nos sirve para colocar elementos verticalmente. Aquí vamos a colocar los botones de acciones de la BD.



Para colocar los botones, abrimos la sección Controls (en la parte izquierda, debajo de la sección Containers), buscamos el control Button, y arrastramos sobre el VBox. Vamos a colocar 3 botones. Con las propiedades de Spacing podemos fijar el espacio entre los botones, y con padding, un margen interno del VBox.

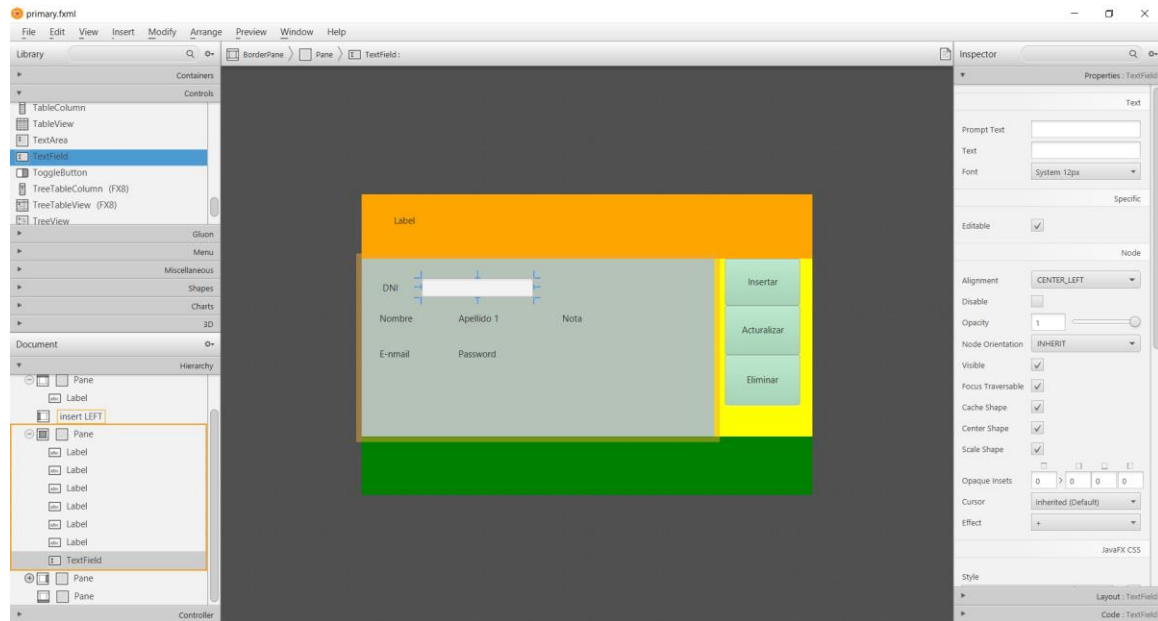


Si hacemos doble click sobre el botón, podemos cambiar el texto que se muestra.



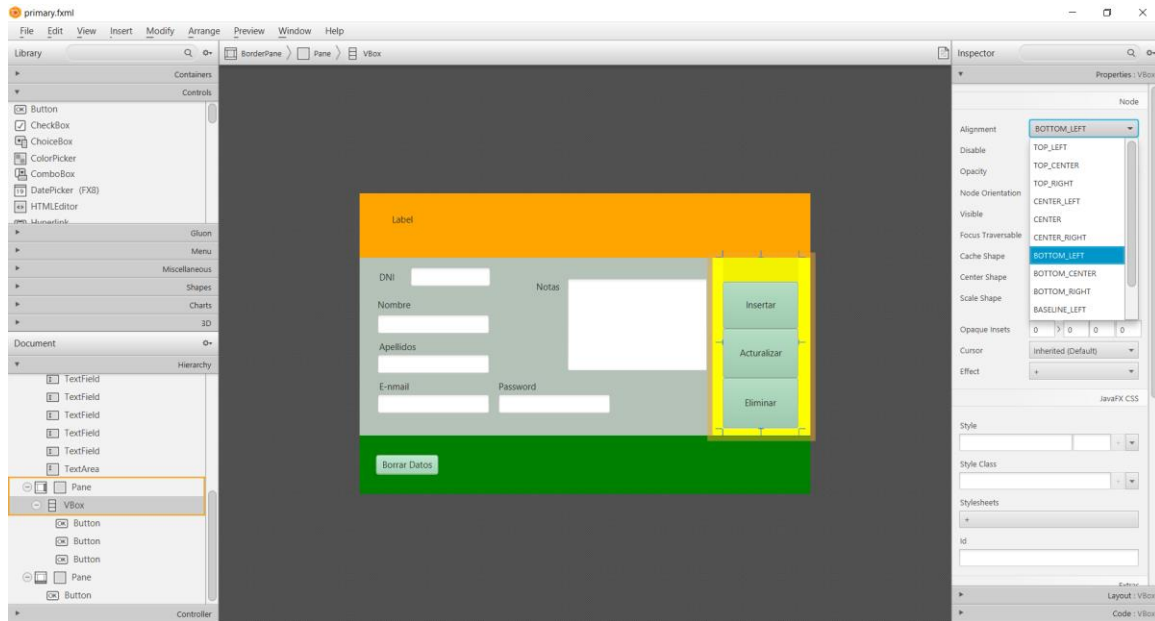
Del mismo modo, vamos a ir colocando etiquetas (control Label) y cambiando sus textos (DNI, Nombre, Apellidos, E-mail, ...)

Para la introducción de datos vamos a utilizar los controles TextField, TextArea (para las Notas) y PasswordField (para la password)

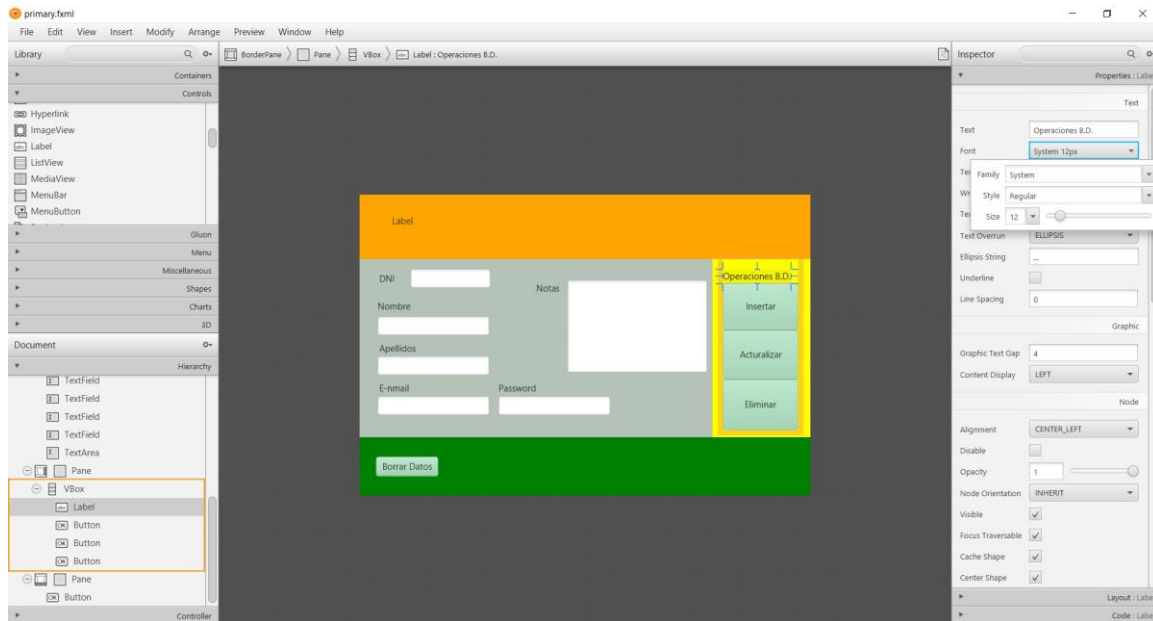


Añadimos también un botón en la parte inferior, que servirá para resetear datos introducidos y mensajes.

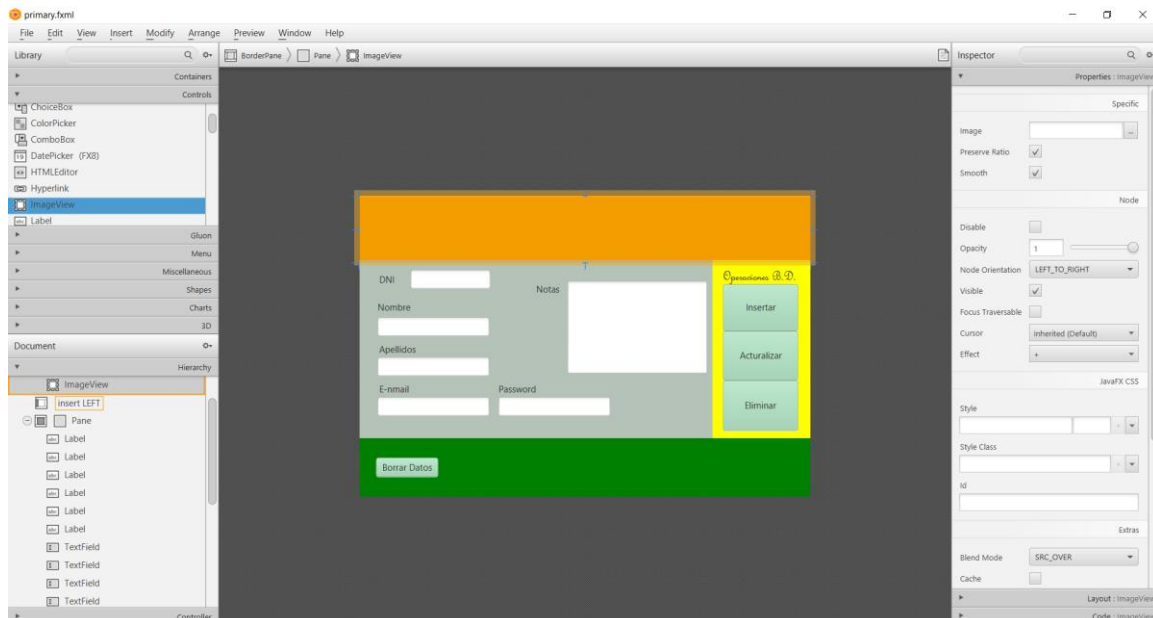
En Properties del VBox, lo ajustamos a la parte inferior, para dejar sitio para colocar una etiqueta sobre los botones.

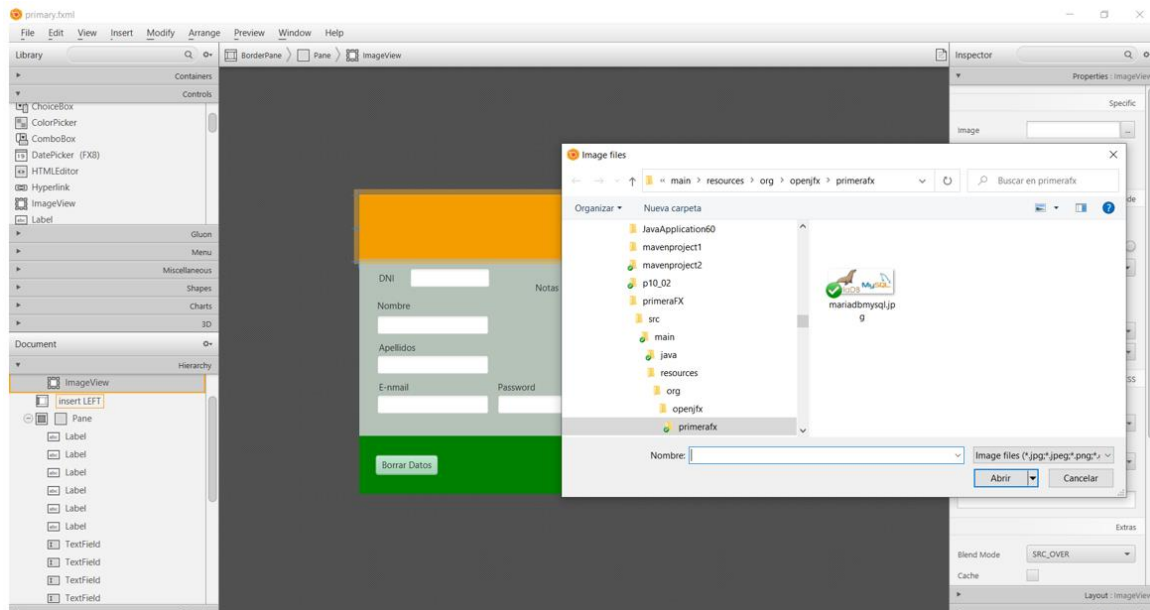


Podemos cambiar el tipo de letra pulsando en la entrada Font. De paso ya voy cambiando un poco los colores para que no quede tan....



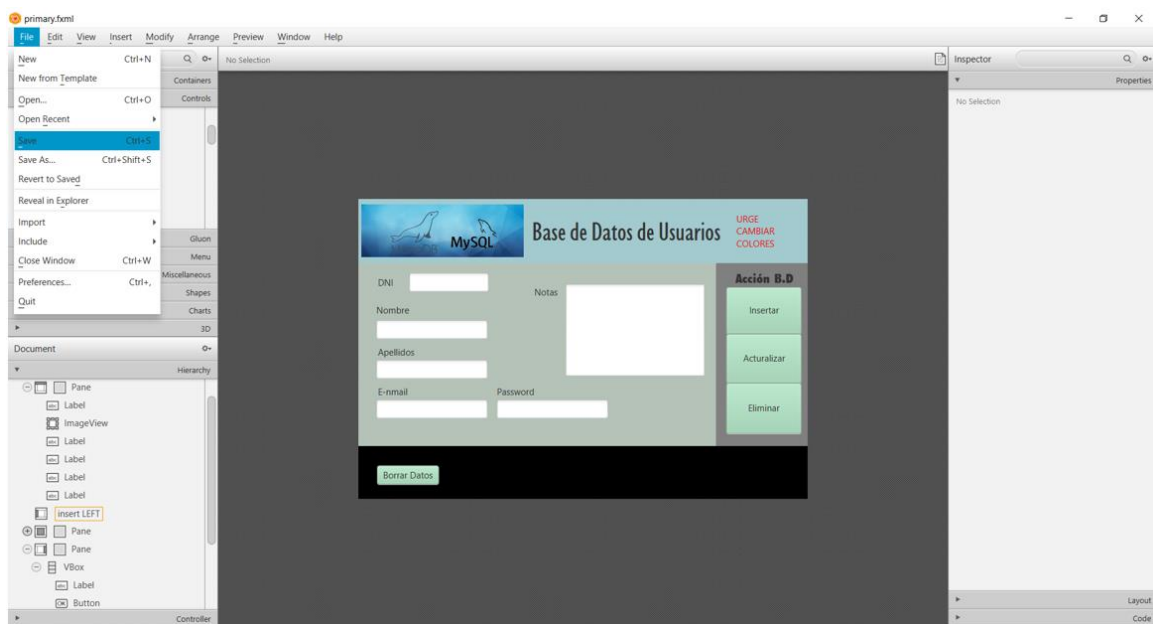
En la zona superior vamos a insertar un logo, una imagen, a la izquierda y a continuación una etiqueta. Para la imagen escogemos el control **ImageView** y lo arrastramos sobre el panel superior y le damos la dimensión que necesitemos. Para coger la imagen pulsamos sobre la entrada Image en la sección Properties (lado derecho) y se abrirá un selector de ficheros. Esta imagen debería estar almacenada en el mismo directorio que el **.css** y el **.fxml** ("**NombreDelProyecto**"/**src/main/resources/.....**)





Y esta es la pantalla final, después de unos cuantos cambios. Para guardar los cambios: **File->Save**

Si queremos ver como quedaría la interfaz: **Preview -> Show Preview in Window**



Para finalizar vamos a NetBeans. En la imagen siguiente se muestra el programa en ejecución y en segundo plano vemos como el fichero **primary.fxml** se ha modificado en función de las acciones realizadas en **Scene Builder**. Podemos modificar el tamaño por defecto de la ventana en el fichero **App.java**:

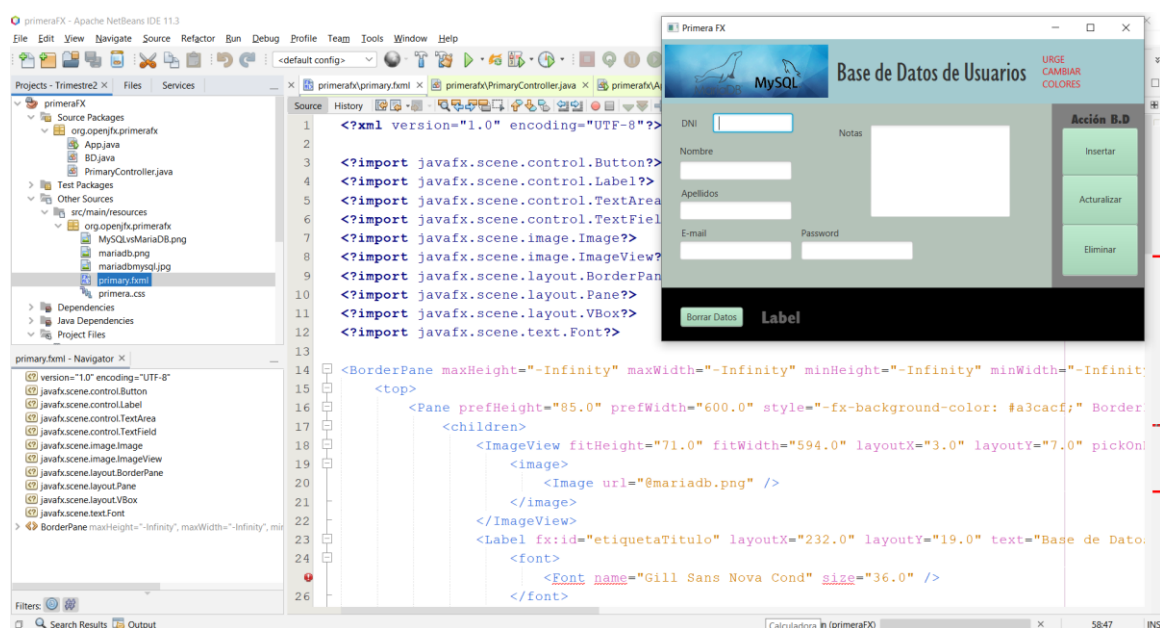
```
scene = new Scene(loadFXML("primary"), 640, 400);
```

Y para darle un nombre a la ventana de la aplicación:

```
stage.setTitle("Probando JavaFX y MariaDB");
```

Si queremos cambiar su icono. Colocamos el fichero del icono en la carpeta de 'resources', y:

```
stage.getIcons().add(new Image(getClass().getResourceAsStream("lineas.png")));
```



En el siguiente documento veremos cómo integrar la conexión a BD en nuestra aplicación.