



# Java

## Tema 7: JAVA FX

---

### Interfaz gráfica

1



### Contenidos

---

1. Introducción
2. Configurar entorno
3. Primer programa

2



## Introduccion

---

Las **interfaces gráficas de usuario (Graphical User Interface - GUI)** ofrecen al usuario ventanas, cuadros de diálogo, barras de herramientas, botones, listas desplegables y muchos otros elementos visuales con los que ya estamos muy acostumbrados a tratar. Son la parte del programa que permite al usuario interactuar con él.

Las aplicaciones son conducidas por **eventos** y se desarrollan haciendo uso de las clases que para ello nos ofrece la **API de Java**.

La API de Java proporciona varias librerías de clases para el desarrollo de Interfaces gráficas de usuario:

- Java **AWT** (Abstract Window Toolkit) → java.awt.\*
- Java **SWING** → javax.swing.\*
- Java **FX** → javafx.\*      **← NOS CENTRAREMOS EN ESTA**

3



## Que es JAVA FX

---

Es un **framework de java** (incluido hasta la jdk 10) para crear aplicaciones **RIA** (Rich Internet Application):

- ☐ Permite vinculación de datos entre GUI y el modelo de datos → BINDING
- ☐ Dos formas de crear GUI:
  - ☐ Código java
  - ☐ Usando **FXML** (lenguaje de script de java)  
**USAREMOS ESTA FORMA**
- ☐ Soporte para efectos y animaciones
- ☐ Soporte multimedia → reproducción audios y videos
- ☐ Contenido web

4



## Características JAVA FX

❑ Se separan la interfaz gráfica de usuario de la lógica/programación de la aplicación.

❑ **Interfaz** → usa lenguaje de script FXML ([más info](#))

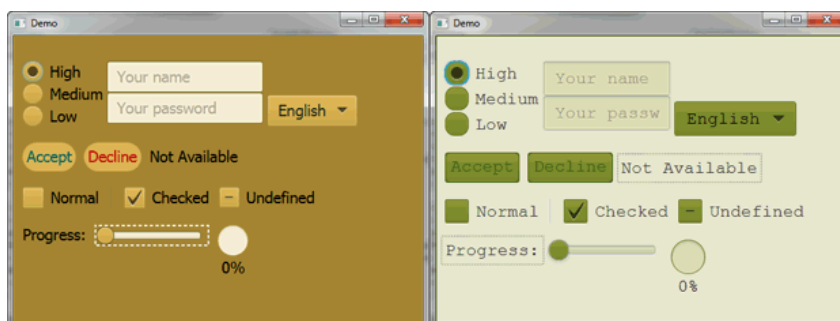
❑ **Lógica** → nuestras clases del api de java

5



## Características JAVA FX

❑ La apariencia de las aplicaciones JavaFX **se puede personalizar mediante hojas de estilos CSS**



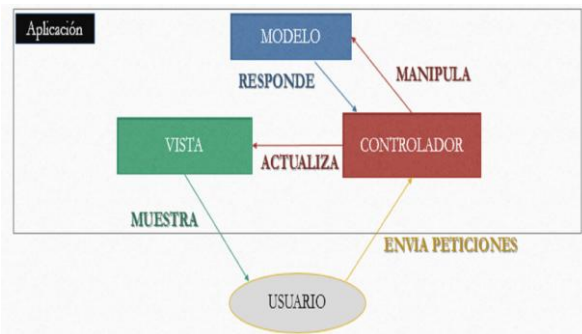
6



## Características: patron MVC

❑ Las aplicaciones Java FX siguen el patrón **MVC (Modelo Vista Controlador)**

- ❑ **Vista** → interfaz de usuario, lo que visualiza
- ❑ **Controlador** → actualiza la vista e interactúa con el modelo
- ❑ **Modelo** → Gestión de los datos (bda, archivos ...) y  
→ Lógica/reglas de negocio de la aplicación



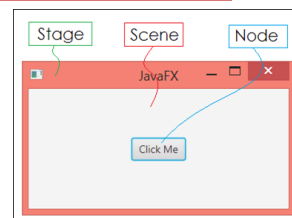
7



## Características JAVA FX

Elementos de la interfaz gráfica:

- ❑ **Stage** → Contenedor
- ❑ **Scene** → Contenedor, donde podemos incluir:



Los  
que  
más  
usare  
mos

- **Nodos(Nodes):** formas(**Shapes**) (2-D y 3-D), imágenes(**Images**), medios(**Media**), navegador web incorporado(**embedded web browser**), texto(**Text**), controles de interfaz de usuario(**Controls UI**), gráficos(**Charts**), grupos(**Groups**) y otros contenedores(**containers**).
- **Estado(State):** Transformaciones(**Transforms**) (posicionamiento y orientación de los nodos), efectos visuales, y otro estado visual del Contenido
- **Efectos(effects):** objetos simples que cambian la apariencia de nodos de gráficos de escena (**Scene Graph**), como los desenfoques (**blurs**), sombras (**shadows**) y ajuste de color (**color adjustment**)

8



## Características JAVA FX

### Ejemplos de Controles UI/Componentes visuales JAVA FX



[Ver en web  
oracle](#)

9



# Java

## CONFIGURAR ENTORNO

### Para Java FX

10



## Scene Builder 10: instalar

- ❑ Scene Builder es el **IDE para diseñar UI (User Interface), interfaces gráficas de usuario**
- ❑ Bajar e instalar **Scene Builder 10 para JDK10** en tu maquina:
  - este es el [instalador](#) para windows 64bits

Download Scene Builder for Java 10		
The latest version of Scene Builder for Java 10 is 10.0.0, it was released on Jun 5, 2018.		
To be kept informed of Scene Builder releases, consider subscribing to the <a href="#">Gluon Newsletter</a>		
Product	Platform	Download
Scene Builder	Executable Jar	<a href="#">Download</a>
Scene Builder	Windows Installer	<a href="#">Download</a>
Scene Builder	Mac OS X .dmg	<a href="#">Download</a>
Scene Builder	Linux RPM	<a href="#">Download</a>
Scene Builder	Linux Deb	<a href="#">Download</a>
Scene Builder kit	Jar File	<a href="#">Download</a>

11



## Scene Builder: primer programa

- ❑ Crear nuevo proyecto **JavaFX FXML Application**, modifica el Project Name escribiendo *PrimerProgramaFX* y pulsa finish

The screenshot shows the 'New Project' dialog box in Scene Builder. The 'Steps' pane on the left shows '1. Choose Project' and '2. Name and Location'. The 'Choose Project' pane shows a tree of project types, with 'JavaFX Application' selected. The 'Name and Location' pane shows the following fields:

- Project Name: *PrimerProgramaFX*
- Project Location: *C:\Users\jagui\Desktop\PROGRAMACION JAVA\den1\den1\Temas 7 - Lectura y escritura de informacion\01 Interfaz grafica\JAVA FX*
- Project Folder: *apob\PROGRAMACION JAVA\den1\den1\Temas 7 - Lectura y escritura de informacion\01 Interfaz grafica\JAVA FX\PrimerProgramaFX*
- JavaFX Platform: *JDK 1.8 (Default)*
- ☐ Create Custom (reloader)
- ☐ Use Dedicated Folder for Storing Libraries
- ☒ Create Application Class: *primerprograma.PrimerProgramaFX*

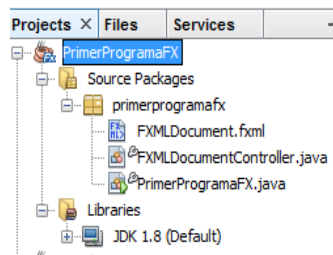
12



## Scene Builder: estructura de archivos

Se crea la siguiente estructura de archivos :

- **FXMLDocument.fxml**: define interfaz de usuario.
- **FXMLDocumentController**: controla entrada y salida de datos y es el controlador del archivo anterior.
- **PrimerProgramaFX.java**: código java para arrancar javaFX



13



## Scene Builder: probando

- **FXMLDocument.fxml**: define interfaz de usuario. Contiene los componentes visuales (botones, imágenes, texto...). Se modifica con el Scene Builder (botón derecho open)

```
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.AnchorPane?>

<AnchorPane id="AnchorPane" prefHeight="200" prefWidth="300"
  <children>
    <Button fx:id="boton" layoutX="71.0" layoutY="15"
    <Label fx:id="etiqueta" layoutX="105.0" layoutY="105"
  </children>
</AnchorPane>
```

```
1 import javafx.event.ActionEvent;
2 import javafx.fxml.FXML;
3 import javafx.fxml.Initializable;
4 import javafx.scene.control.Button;
5 import javafx.scene.control.Label;
6
7 public class FXMLDocumentController implements Initializable {
8
9     @FXML
10     private Button boton;
11     @FXML
12     private Label etiqueta;
13
14     @FXML
15     private void alPulsarBoton(ActionEvent event) {
16         System.out.println("You clicked me!");
17         etiqueta.setText("Hello World!");
18     }
19 }
```

- **FXMLDocumentController**: controla la entrada y salida de datos y es el controlador del archivo anterior. Contiene un atributo por cada componente visual con la etiqueta **@FXML**, y los controladores de cada componente visual.

- **PrimerProgramaFX.java**: código java para arrancar javaFX y mostrar el archivo FXMLDocument

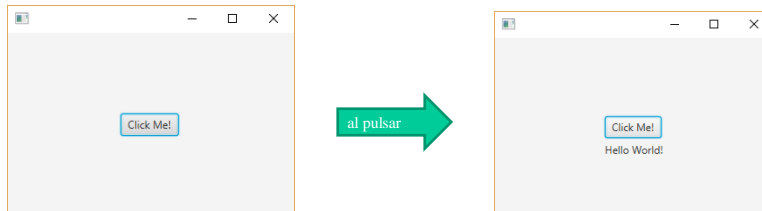
```
public class JavaFXApplication2 extends Application {
    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }
}
```

14



## Scene Builder: ejecución

- ❑ **Ejecútalo** (Run) sin modificar nada y pulsa el botón "Click Me!". Revisa el código

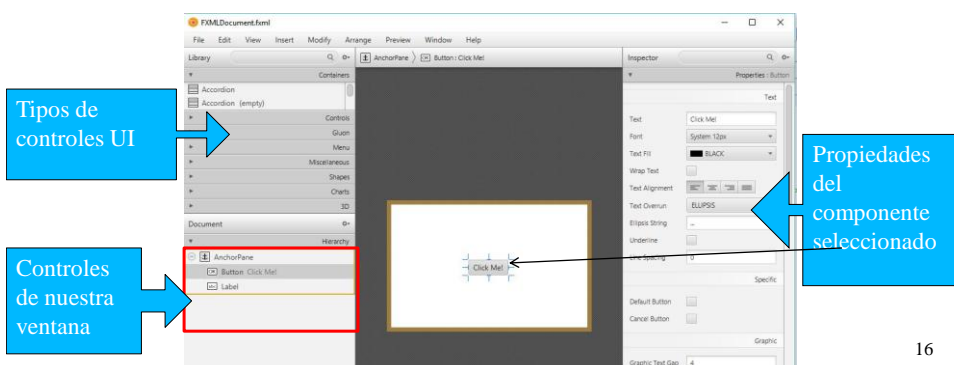


15



## Modificar archivo FXMLDocument.fxml

- ❑ Para modificar y diseñar la ventana gráfica/visual, añadir nuevos componentes o modificar propiedades, doble click sobre el archivo o botón derecho y pulsa **open** → se abrirá en el **SceneBuilder**



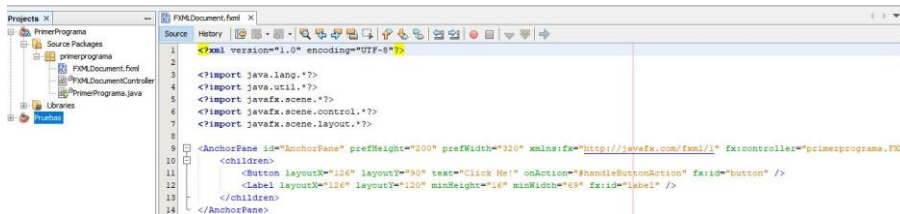
16





## Archivo FXMLDocument.fxml

- ❑ Para editar su código fxml, botón derecho sobre el archivo y pulsa **edit** → **ESTO NO ES RECOMENDABLE** (SceneBuilder se encarga de este archivo)



```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import java.lang.*?>
4 <?import java.util.*?>
5 <?import javafx.scene.*?>
6 <?import javafx.scene.control.*?>
7 <?import javafx.scene.layout.*?>
8
9 <AnchorPane id="AnchorPane" prefHeight="200" prefWidth="320" xmlns:fx="http://javafx.com/fxml/" fx:controller="primerprograma.FXMLDocumentController">
10     <children>
11         <Button layoutX="126" layoutY="150" text="Click Me" onAction="#handleButtonAction" fx:id="button" />
12         <Label layoutX="126" layoutY="120" minHeight="16" minWidth="65" fx:id="label" />
13     </children>
14 </AnchorPane>
```

17



## Scene Builder: probando

Si modificamos el archivo fxml desde Scene Builder, aparece en la zona superior un triángulo amarillo:



- ❑ El triángulo amarillo indica que **se debe actualizar el controller**. Por tanto debemos hacer:

1. En **Scene Builder**: pulsar **File/Save**
2. En **NetBeans**: **selecciona el archivo \*.fxml**, botón derecho y pulsa **Source/Make Controller** → esto actualiza el controlador XXXController.java añadiendo nuevos componentes del \*.fxml. Esto hará que se actualice los elementos y manejadores/métodos de la clase **FXMLDocumentController**

18



# Java

## PRIMER PROGRAMA

19



### Primer programa

- ❑ Video tutorial para **crear una aplicación simple:**

[https://www.youtube.com/watch?v=AKYnfYJxO8U&index=7&list=PL2QCh\\_slcGPNjMIHqdAkYhVlcI3UZ-8O2](https://www.youtube.com/watch?v=AKYnfYJxO8U&index=7&list=PL2QCh_slcGPNjMIHqdAkYhVlcI3UZ-8O2)

#### Nosotros seguiremos otro orden:

##### 1.En Scene Builder:

- Crear los componentes visuales y asignarles id.
- Definir nombre de los métodos que gestionarán los eventos. (pej. En el botón "mostrar" escribir, en la propiedad "on Action", accionMostrar)
- grabar (Save)

**2.En NetBeans:** Seleccionar \*.fxml botón derecho y **"make controller"** para que se añadan los componentes como atributos y los métodos en el controlador. Programar ahí las funcionalidades

**3.** El comboBox de "año nacimiento" lo rellenaremos de otra manera (más adelante lo veremos)