

# Introducción a JavaFX

DI DAM2

# Ventajas JavaFX

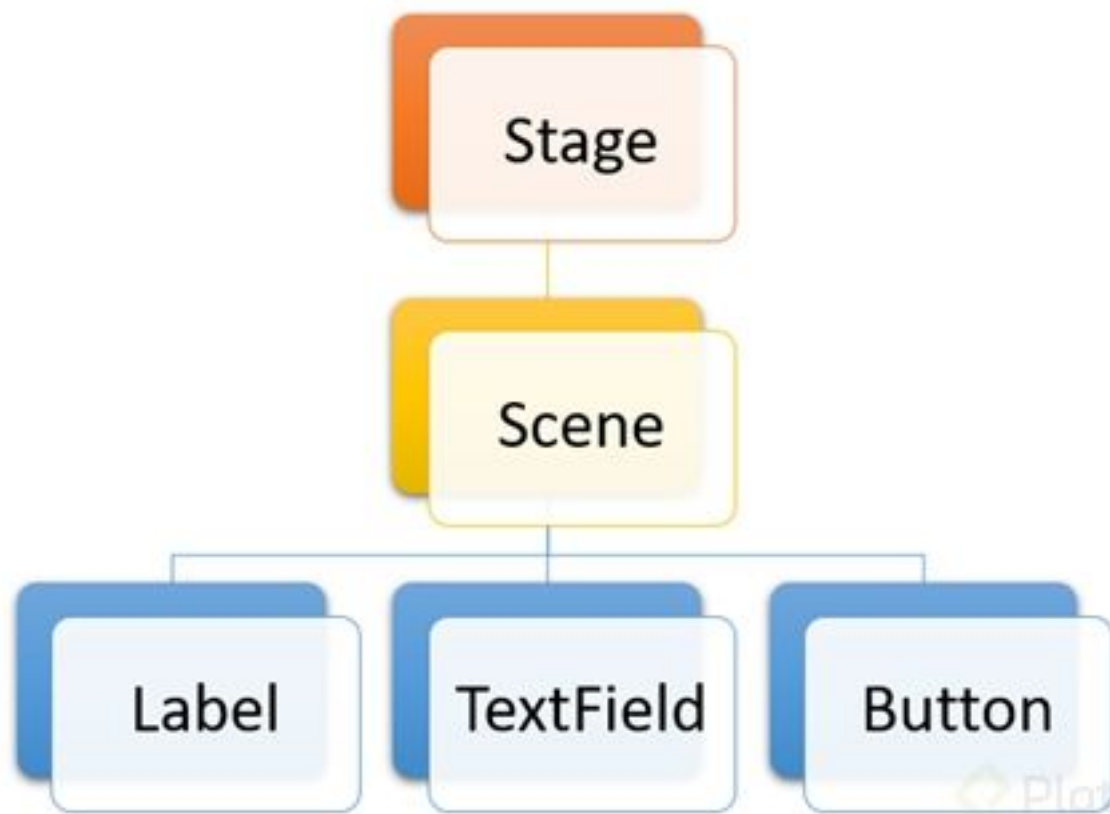
- Hojas de estilo CSS
- Soporte FXML
- Elementos organizados en una estructura de árbol

# The Stage

- Ventana de nivel superior
- Cada proyecto tiene al menos un escenario principal
- El escenario es creado automáticamente al crear el proyecto

# The Scene

- Contenedor de elementos
- Cada escenario tiene al menos una escena
- Cada escenario puede contener varias escenas e ir cambiando entre ellas

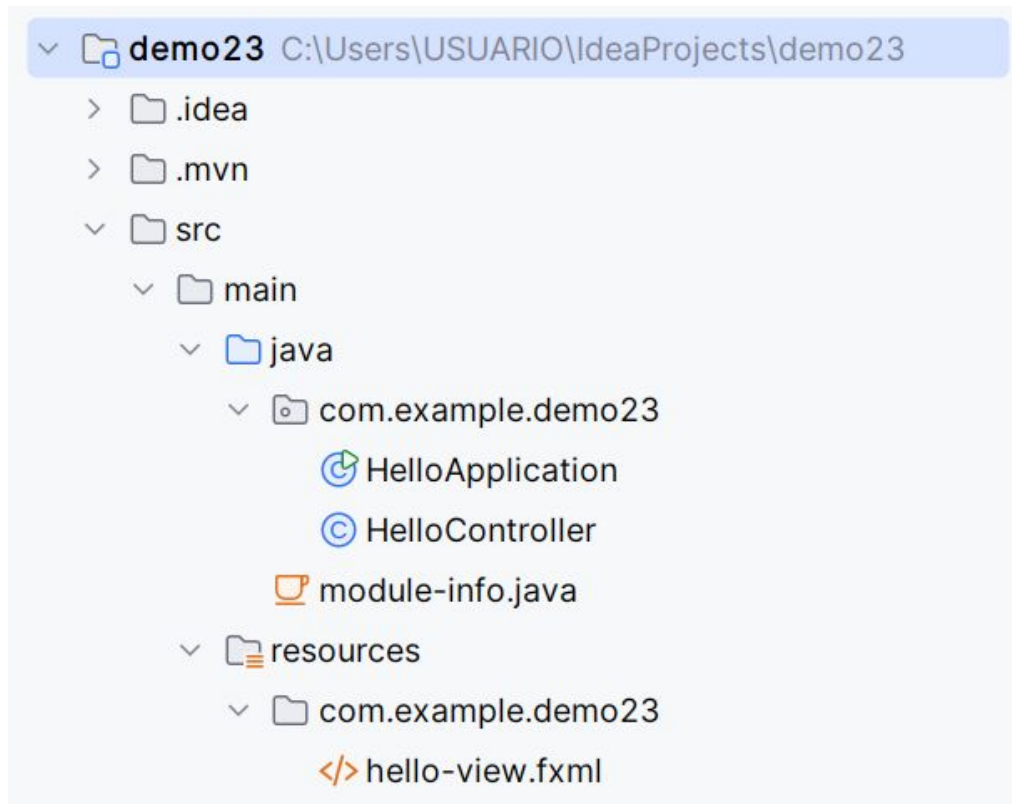


# Organización

- Stage
  - Scene
    - Root Node (Contenedor principal)
      - Parent Nodes (Contenedores internos)
        - Leaf Nodes (Controles)

# Contenido de un nuevo proyecto

- HelloApplication
- HelloController
- hello-view.fxml



# Código de HelloApplication

```
package com.example.demo23;

import ...

public class HelloApplication extends Application {
    @Override
    public void start(Stage stage) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource("hello-view.fxml"));
        Scene scene = new Scene(fxmlLoader.load(), width: 320, height: 240);
        stage.setTitle("Hello!");
        stage.setScene(scene);
        stage.show();
    }

    public static void main(String[] args) { launch(); }
}
```



# Código de HelloController

```
import javafx.fxml.FXML;
import javafx.scene.control.Label;

public class HelloController {
    @FXML
    private Label welcomeText;

    @FXML
    protected void onHelloButtonClick() { welcomeText.setText("Welcome to JavaFX Application!"); }
}
```

# Código de hello-view.fxml

```
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.VBox?>

<?import javafx.scene.control.Button?>
<VBox alignment="CENTER" spacing="20.0" xmlns:fx="http://javafx.com/fxml"
    fx:controller="com.example.demo23.HelloController">
    <padding>
        <Insets bottom="20.0" left="20.0" right="20.0" top="20.0"/>
    </padding>

    <Label fx:id="welcomeText"/>
    <Button text="Hello!" onAction="#onHelloButtonClick"/>
</VBox>
```

# Contenedores

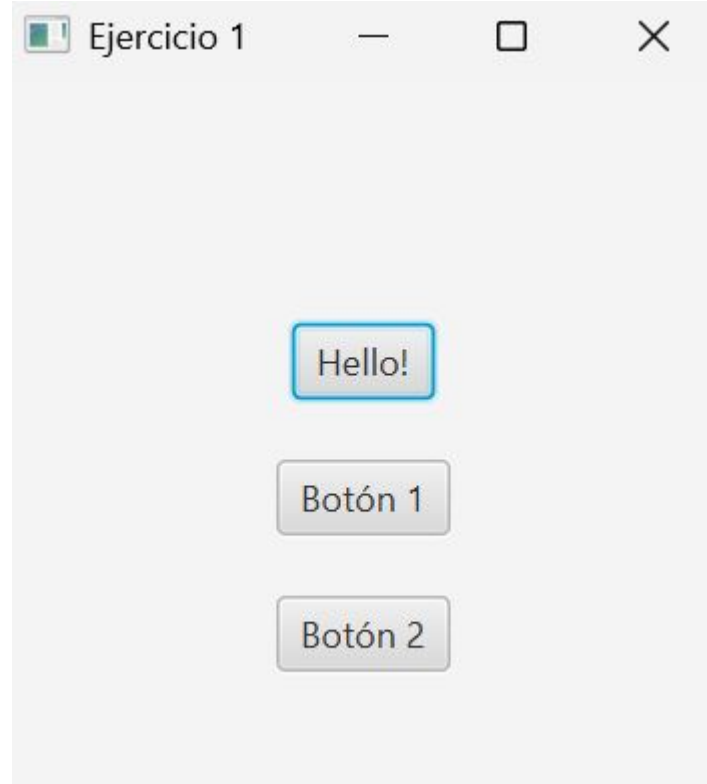
- AnchorPane
- VBox
- HBox
- GridPane

# Controls

- Button
- Label
- TextField

# Ejercicio 1

- Probar distintas medidas para la escena
- Cambiar el título de la interfaz
- Añadir 2 botones
- Cambiar *VBox* por *HBox*



# Ejercicio utilizando xml



# Añadir un evento a un botón

- Al pulsar un botón, en la etiqueta aparece la letra correspondiente



# Ficheros FXML

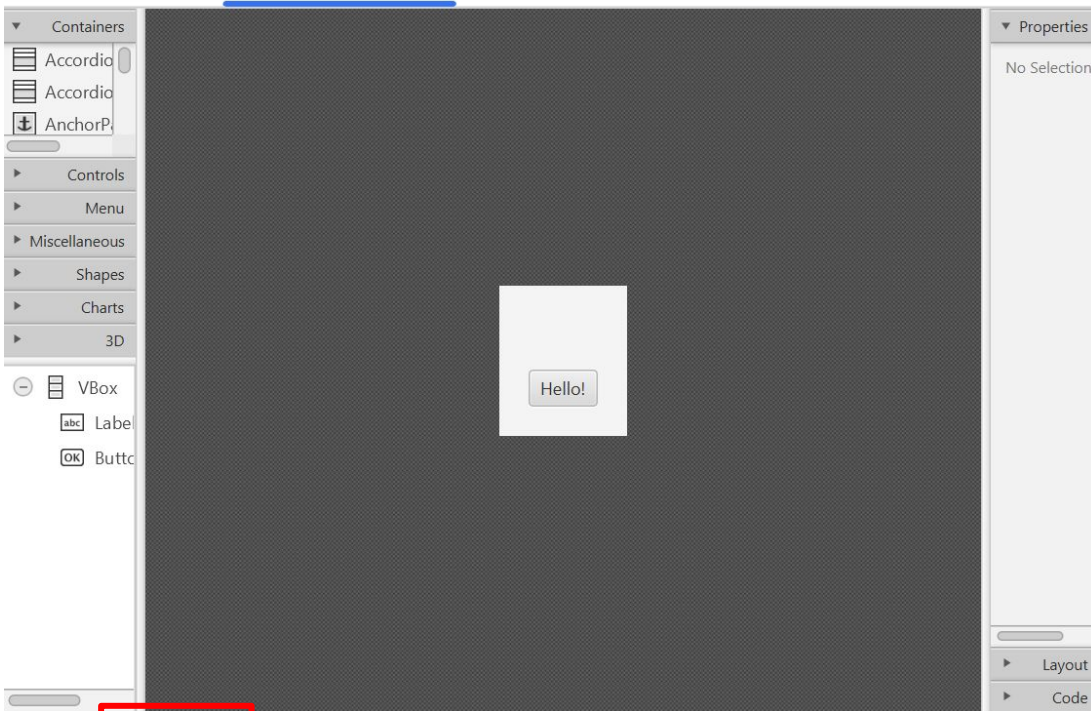
pom.xml (Prueba2) hello-view.fxml x HelloController.java

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.geometry.Insets?>
4 <?import javafx.scene.control.Label?>
5 <?import javafx.scene.layout.VBox?>
6
7 <?import javafx.scene.control.Button?>
8 <VBox alignment="CENTER" spacing="20.0" xmlns:fx="http://javafx.com/fxml"
9     fx:controller="org.example.prueba2.HelloController">
10     <padding>
11         <Insets bottom="20.0" left="20.0" right="20.0" top="20.0"/>
12     </padding>
13
14     <Label fx:id="welcomeText"/>
15     <Button text="Hello!" onAction="#onHelloButtonClicked"/>
16 </VBox>
17
```

Text

Scene Builder

pom.xml (Prueba2) hello-view.fxml x HelloController.java HelloApplication.java



Text

Scene Builder



# Abrir el fichero FXML en Scene Builder

