

# Introducción a JavaFX



# Contenido

## Introducción a JavaFX

- ❖ ¿Qué es JavaFx?
- ❖ Patrón MVC.
- ❖ Instalación de JavaFX.
- ❖ Arquitectura y estructura.
- ❖ Stage, Scene, Layouts.
- ❖ Primera aplicación.
- ❖ SceneBuilder.



# ¿Qué es JavaFX?

- ❖ *JavaFX* es una tecnología de software que, combinada con Java, permite crear Rich Internet Applications (RIAs).
- ❖ Lanzado a fines del 2008.
- ❖ Es el sucesor de **Swing** (1997), otro set de librerías de front-end de Java.
- ❖ Herramientas gráficas.
- ❖ Basado en escenas.
- ❖ Multiplataforma.
- ❖ Runtime específico.



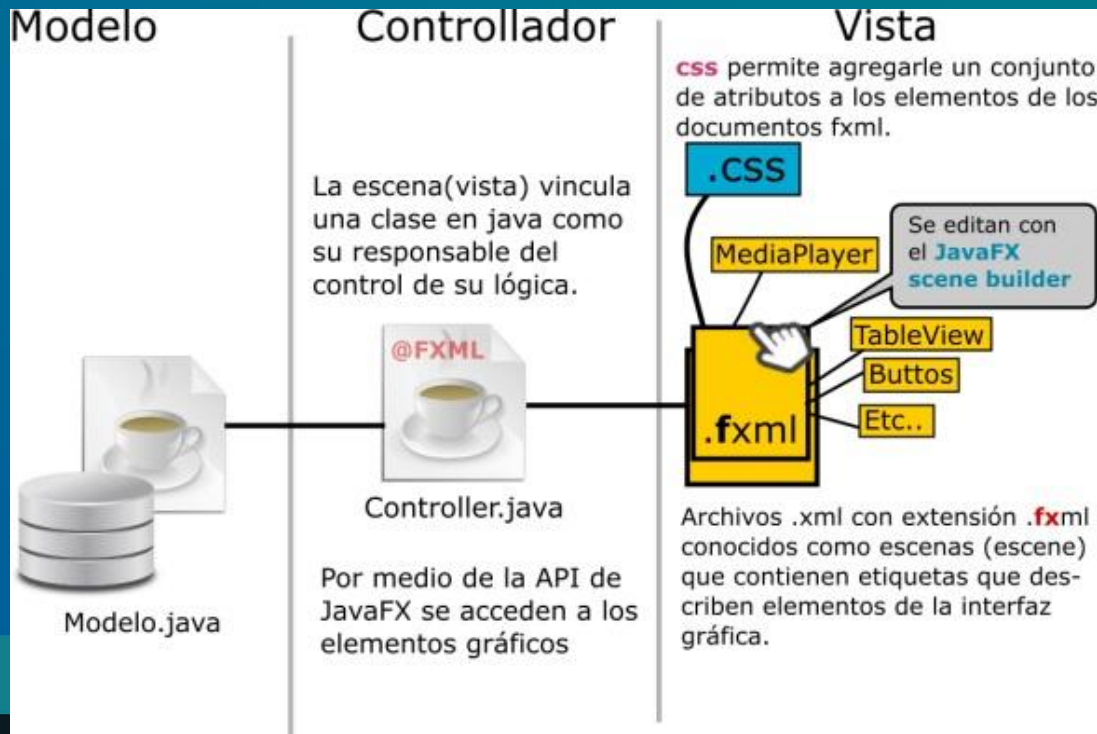


# Patrón MVC

- ❖ **Model** (Modelo)
  - Se compone de todas las clases del proyecto.
- ❖ **View** (Vista)
  - Son los archivos **.FXML** con el diseño, posición y estilo de todos los componentes de *front-end*.
- ❖ **Controller** (Controlador)
  - Es la clase de front-end que implementa la interacción de los componentes.

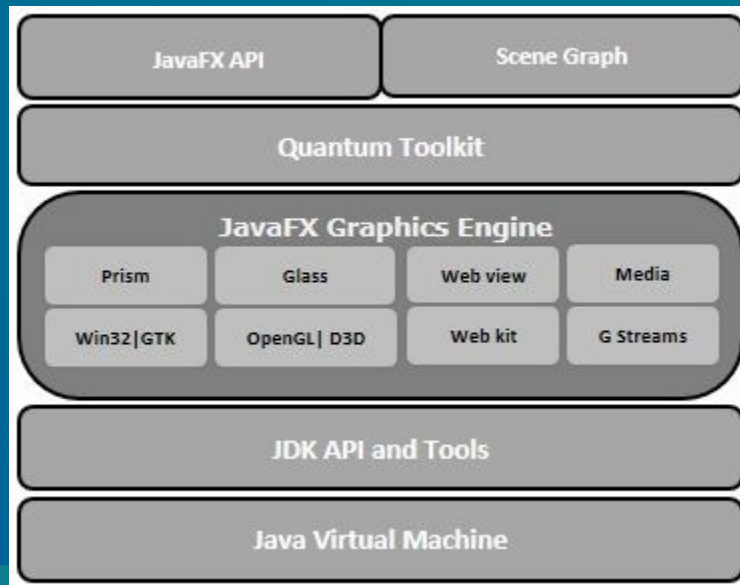
Usando **.FXML** se fuerza el uso de **MVC**.

# Patrón MVC

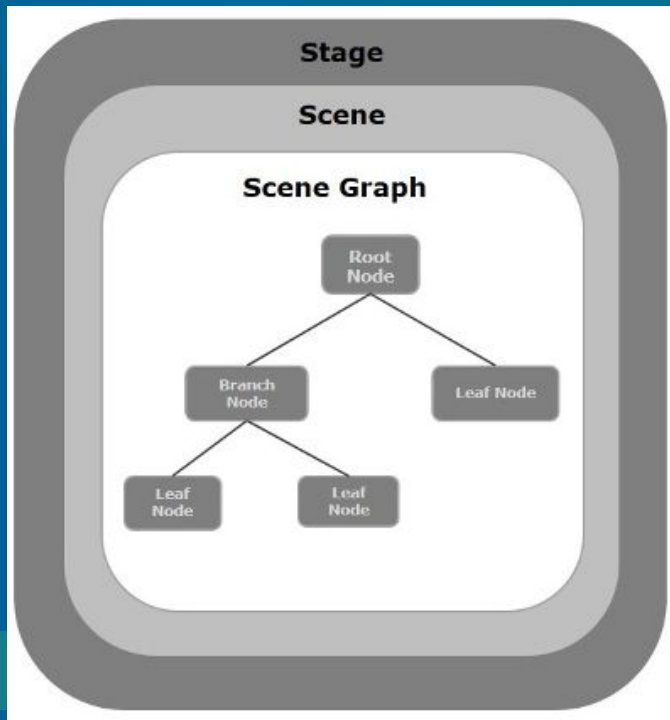


# Instalación JavaFX

# Arquitectura



# Estructura



- ❖ Cada aplicación tendrá un **Stage**.
- ❖ El Stage tendrá una **Scene**.
- ❖ La Scene tendrá un **Layout**.
- ❖ El Layout tendrá una serie de *componentes*.



# Estructura - Scene

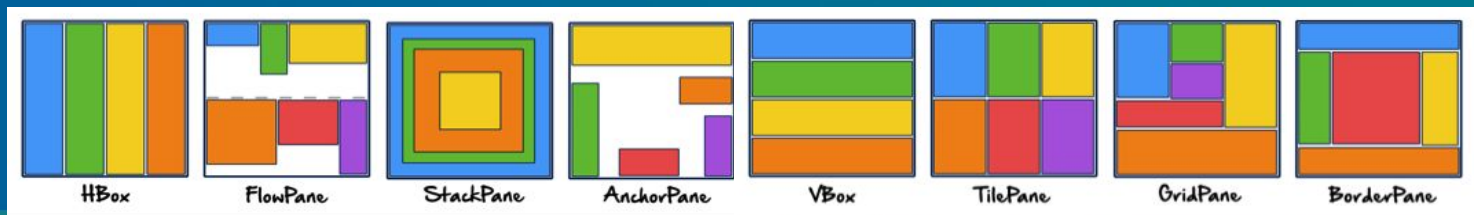
- ❖ Un **Scene** representa el contenido físico de una aplicación JavaFX. Contiene toda la parte gráfica de la escena.
- ❖ La clase **Scene** del paquete `javafx.scene` representa el objeto de Scene. Como una instancia, el objeto de Scene se agrega a un **Stage**.
- ❖ Un *gráfico de escena* es una estructura de datos en forma de árbol (jerárquica) que representa el contenido de una escena. Por el contrario, un *nodo* es un objeto visual/gráfico de un gráfico de escena.

# Estructura - Stage

- ❖ Un **Stage** (ventana contenedora) contiene todos los objetos de una aplicación JavaFX (*javafx.stage*).
- ❖ El Stage principal lo crea la propia plataforma. El objeto Stage creado se pasa como argumento al método abstracto **start()** de la clase **Application** (explicado más adelante).
- ❖ Un Stage se divide en:
  - Área de contenido.
  - Decoraciones (barra de títulos, bordes, etc.).

# Estructura - Layouts

- ❖ Si bien una aplicación JavaFX se puede configurar *manualmente* la interfaz de usuario, la opción más fácil es hacer uso de *paneles de diseño* (**layouts**).
- ❖ Hay varios *layouts* para una configuración sencilla de diseños clásicos como filas, columnas, pilas, mosaicos y otros.



# Controles de usuario

- ❖ Los **controles UI** son los elementos gráficos que permiten a los usuarios interactuar con una aplicación o un sitio web.  
Incluyen *botones, menús, campos de texto, casillas de verificación, botones de opción* entre otros.
- **UI elements:** son los principales elementos con los que el usuario interactúa. JavaFX proporciona una lista enorme de dichos elementos.
- **Layouts:** definen cómo se deben organizar los elementos de la interfaz de usuario en la pantalla y proporcionan una apariencia final a la GUI.
- **Behavior:** son eventos que ocurren cuando el usuario interactúa con elementos de la interfaz de usuario.

# Primera aplicación



# Aplicación JavaFX

- ❖ La clase **Application** del paquete *javafx.application* es el punto de entrada de la aplicación en JavaFX.
- ❖ Para crear una aplicación JavaFX, se debe heredar de esta clase e implementar su método abstracto **start()**.
- ❖ En el método **main()**, se debe iniciar la aplicación utilizando el método **launch()**.  
Este método llama internamente al método `start()` de la clase `Application`.

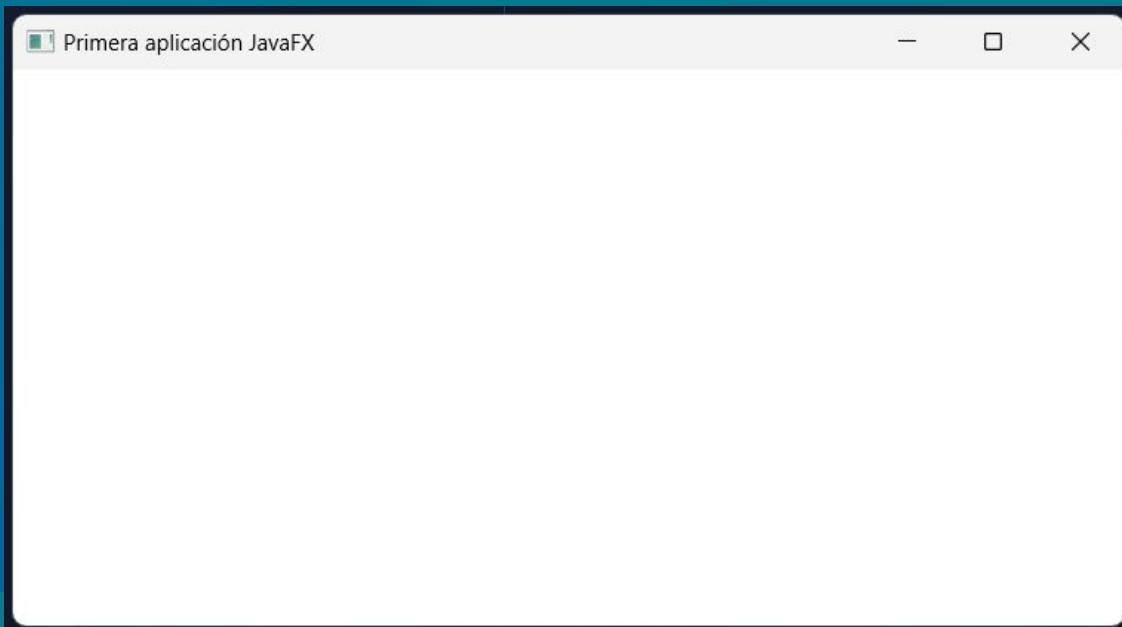
# Aplicación JavaFX

```
public class JFXTest extends Application {  
  
    @Override  
    public void start(Stage stage) throws Exception {  
  
        // Agregar código para Stage, Scene y Scene graph  
    }  
  
    public static void main(String[] args) {  
  
        JFXTest.launch(args);  
    }  
}
```



# Aplicación JavaFX

```
public class JFXTest extends Application {  
  
    @Override  
    public void start(Stage stage) throws Exception {  
  
        // Crear grupo  
        Group group = new Group();  
  
        // Crear una escena pasando el grupo, alto y ancho  
        Scene scene = new Scene(group, 600, 300);  
  
        // Establecer el título al escenario  
        primaryStage.setTitle("Primera aplicación JavaFX");  
  
        // Agregar la escena al escenario  
        primaryStage.setScene(scene);  
  
        // Mostrar el contenido del escenario  
        primaryStage.show();  
  
    }  
  
    public static void main(String[] args) {  
  
        JFXTest.launch(args);  
  
    }  
}
```



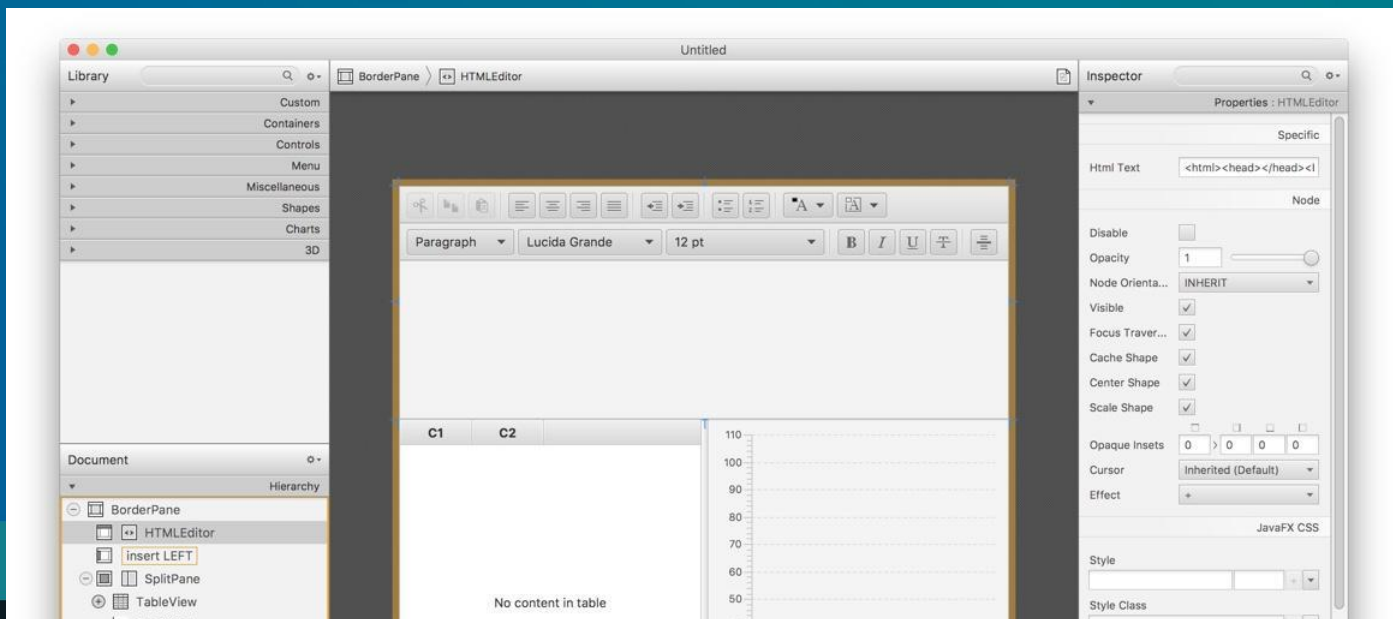


# SceneBuilder

# SceneBuilder



- ❖ Aplicación para crear de forma interactiva archivos de vista (.FXML)



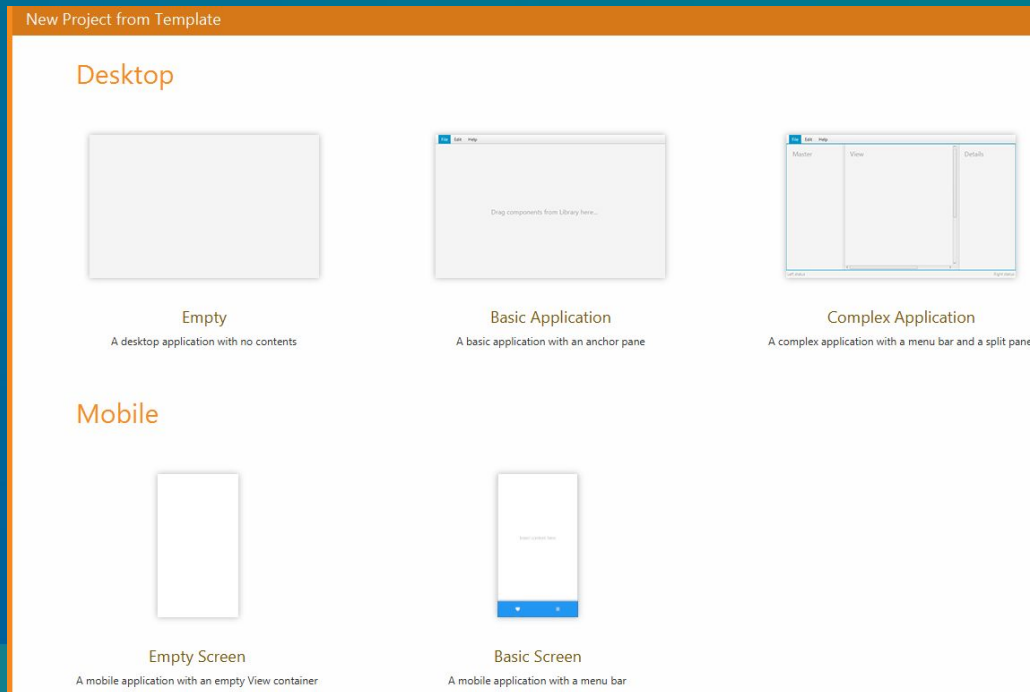
# SceneBuilder

- ❖ Aplicación de diseño de formularios
  - Drag & Drop.
  - AnchorPane.
  - Propiedades, Layouts y eventos.
  - FXML.
  
- ❖ Paquete para construir componentes.
  - Clases estáticas.
  - Patrón MVC.
  - Formato de árbol.

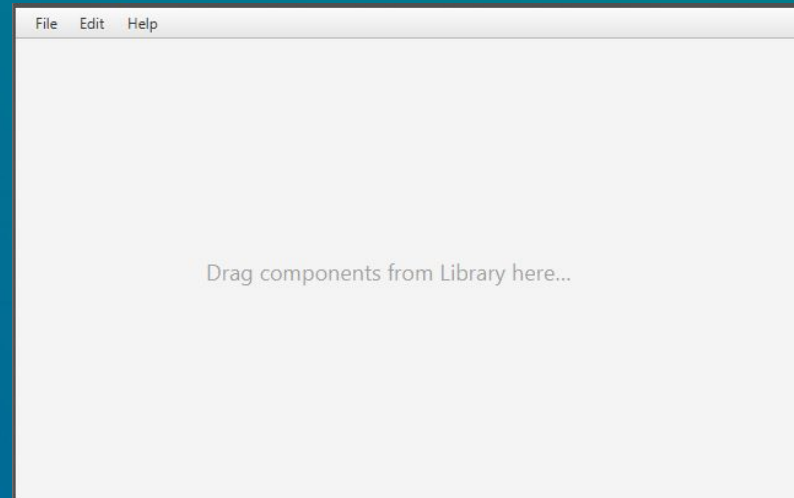
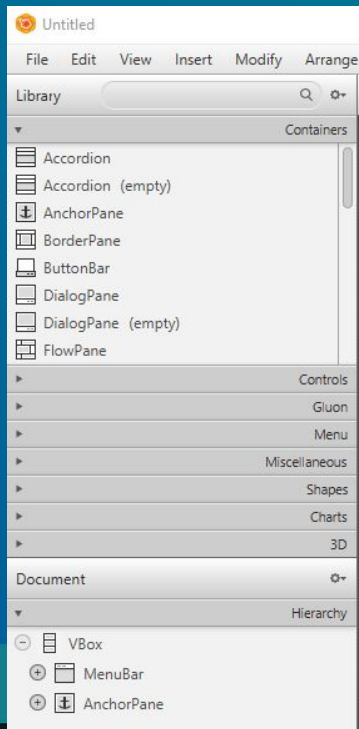
# SceneBuilder

- ❖ Patrón MVC
  - FXML son archivos de Vista.
  - Admite la clase Controller.
  - Con métodos de tipo evento ***void evento(ActionEvent)*** para responder las acciones de los controles.
- ❖ FXML
  - Atributos **on**
    - ***onAction="#evento"***

# SceneBuilder



# SceneBuilder



# SceneBuilder



The screenshot shows the JavaFX SceneBuilder interface with several UI components and their corresponding JavaFX classes labeled with handwritten text:

- Accordion:** A container with three titled panes. One pane is expanded, showing a **ChoiceBox** and a **ColorPicker** (labeled with a color swatch and the text "Schwarz").
- Button:** A standard rectangular button.
- CheckBox:** A checkbox with the label "CheckBox" and a checked state.
- ContextMenu:** A context menu containing:
  - MenuItem**
  - CheckMenuItem** (checked)
  - RadioMenuItem 1**
  - RadioMenuItem 2** (checked)
- ProgressBar:** A horizontal progress bar.
- PasswordField:** A text field with masked characters (dots).
- ProgressIndicator:** A circular progress indicator.
- Pagination:** A pagination control showing a sequence of numbers from 1 to 10, with "1/10" indicating the current page.

# Ejercitación