



El futuro digital
es de todos

MinTIC



Vigilada Mineducación

CICLO II: Programación Básica en Java

Misión
TIC2022





El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

Sesión 13: Introducción a Java

Aplicaciones gráficas en Java.

Misión
TIC2022



Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Usar el Editor (VS Code) para el desarrollo de aplicaciones en Java
2. Desarrollar interfaces gráficas utilizando codificación directa



¿Qué es JavaFX?

- JavaFX es una biblioteca de Java que se utiliza para desarrollar aplicaciones de escritorio y aplicaciones de Internet enriquecidas (RIA). Las aplicaciones creadas en JavaFX pueden ejecutarse en múltiples plataformas, incluidas Web, Móvil y Computadoras de escritorio.
- JavaFX presenta un lenguaje conocido como FXML, que es un lenguaje de marcado declarativo similar al HTML. El único propósito de este lenguaje es definir una interfaz de usuario.

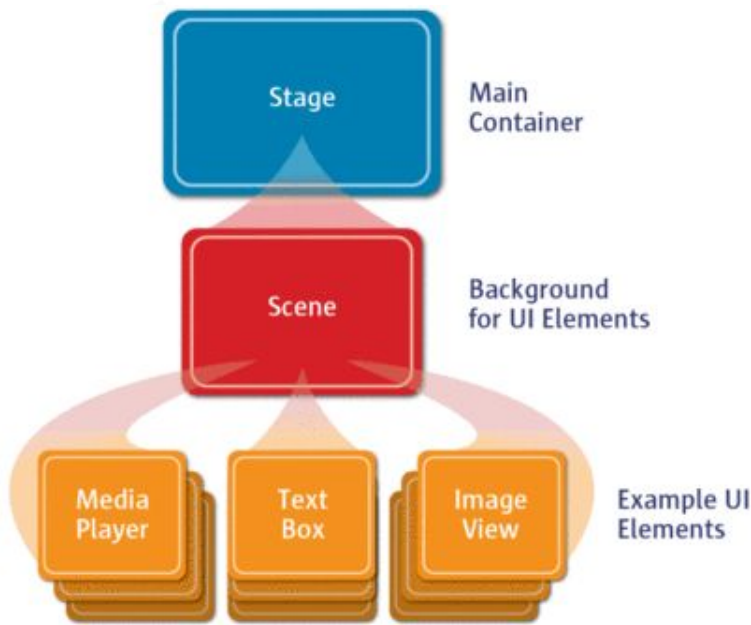


¿Qué es JavaFX?

- JavaFX está destinado a reemplazar el swing en las aplicaciones Java como un marco de GUI. Sin embargo, aporta más funcionalidades que swing. Al igual que Swing, JavaFX también proporciona sus propios componentes y no depende del sistema operativo. Es ligero y acelerado por hardware. Es compatible con varios sistemas operativos, incluidos Windows, Linux y Mac OS.



Jerarquía de Clases en JavaFX



- El Stage (escenario): es el contenedor principal, normalmente una ventana con borde y los típicos botones para maximizar, minimizar o cerrar la ventana. Dentro del Stage se puede añadir una Scene.
- El scene (escena): la cual puede cambiarse dinámicamente por otra Scene. Dentro de un Scene se añaden los nodos JavaFX, tales como AnchorPane, TextBox, etc.



Componentes de JavaFX básicos

Control	Description
Label	Label is a component that is used to define a simple text on the screen. Typically, a label is placed with the node, it describes.
Button	Button is a component that controls the function of the application. Button class is used to create a labelled button.
RadioButton	The Radio Button is used to provide various options to the user. The user can only choose one option among all. A radio button is either selected or deselected.
CheckBox	Check Box is used to get the kind of information from the user which contains various choices. User marked the checkbox either on (true) or off(false).
TextField	Text Field is basically used to get the input from the user in the form of text. <code>javafx.scene.control.TextField</code> represents TextField
PasswordField	PasswordField is used to get the user's password. Whatever is typed in the passwordfield is not shown on the screen to anyone.



HyperLink	HyperLink are used to refer any of the webpage through your application. It is represented by the class <code>javafx.scene.control.HyperLink</code>
Slider	Slider is used to provide a pane of options to the user in a graphical form where the user needs to move a slider over the range of values to select one of them.
ProgressBar	Progress Bar is used to show the work progress to the user. It is represented by the class <code>javafx.scene.control.ProgressBar</code> .
ProgressIndicator	Instead of showing the analogue progress to the user, it shows the digital progress so that the user may know the amount of work done in percentage.
ScrollBar	JavaFX Scroll Bar is used to provide a scroll bar to the user so that the user can scroll down the application pages.
Menu	JavaFX provides a Menu class to implement menus. Menu is the main component of any application.
ToolTip	JavaFX ToolTip is used to provide hint to the user about any component. It is mainly used to provide hints about the text fields or password fields being used in the application.



Class	Description
BorderPane	Organizes nodes in top, left, right, centre and the bottom of the screen.
FlowPane	Organizes the nodes in the horizontal rows according to the available horizontal spaces. Wraps the nodes to the next line if the horizontal space is less than the total width of the nodes
GridPane	Organizes the nodes in the form of rows and columns.
HBox	Organizes the nodes in a single row.
Pane	It is the base class for all the layout classes.
StackPane	Organizes nodes in the form of a stack i.e. one onto another
VBox	Organizes nodes in a vertical column.



Eventos en JavaFX

JavaFX proporciona soporte para manejar una amplia variedad de eventos. La clase llamada Evento del paquete javafx.event es la clase base para un evento. Una instancia de cualquiera de su subclase es un evento. JavaFX proporciona una amplia variedad de eventos:

- **Mouse Event:** este es un evento de entrada que ocurre cuando se hace clic en un mouse. Está representado por la clase llamada **MouseEvent** . Incluye acciones como hacer clic con el mouse, presionar el mouse, soltar el mouse, mover el mouse, objetivo ingresado con el mouse, objetivo salido del mouse, etc.
- **Key Event:** Este es un evento de entrada que indica que la pulsación de la tecla se produjo en un nodo. Está representado por la clase llamada KeyEvent . Este evento incluye acciones como tecla presionada, tecla soltada y tecla tecleada.



Eventos en JavaFX

- **Drag Event:** Este es un evento de entrada que ocurre cuando se arrastra el mouse. Está representado por la clase llamada `DragEvent` . Incluye acciones como arrastrar ingresado, arrastrar soltado, arrastrar el objetivo ingresado, arrastrar el objetivo salido, arrastrar sobre, etc.
- **Window Event:** Este es un evento relacionado con acciones de mostrar / ocultar ventanas. Está representado por la clase llamada `WindowEvent` . Incluye acciones como ocultar ventanas, mostrar ventanas, ocultar ventanas, mostrar ventanas, etc.



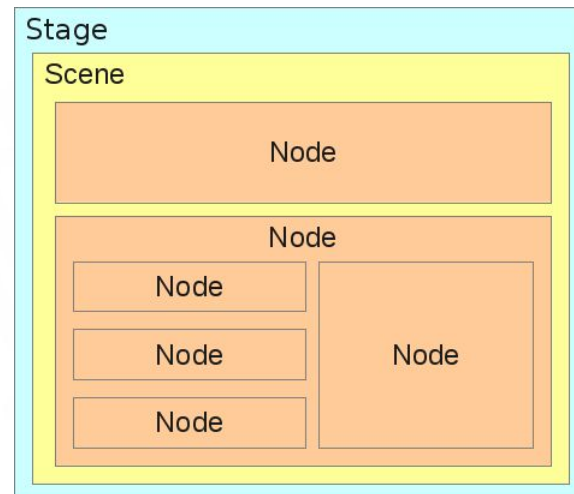
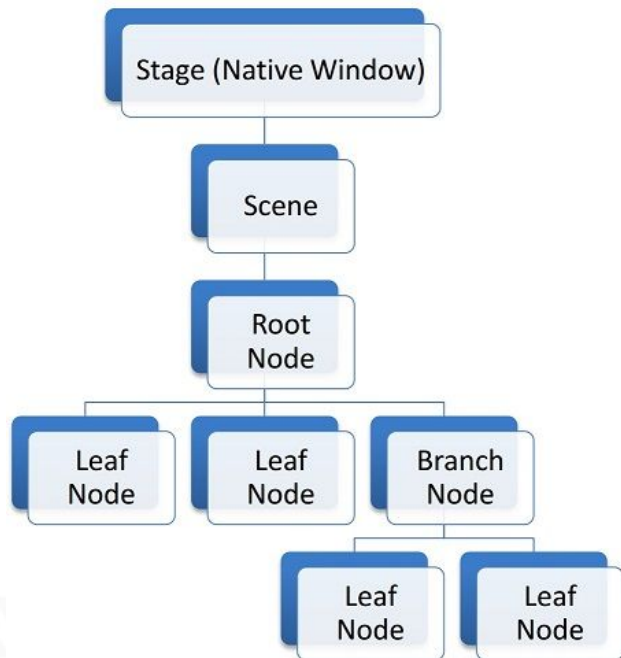
Eventos en JavaFX

Event Handling: El manejo de eventos es el mecanismo que controla el evento y decide qué debe suceder, si ocurre un evento. Este mecanismo tiene el código que se conoce como controlador de eventos que se ejecuta cuando ocurre un evento. JavaFX proporciona controladores y filtros para manejar eventos. En JavaFX cada evento tiene:

- **Target:** el nodo en el que ocurrió un evento. Un objetivo puede ser una ventana, una escena y un nodo.
- **Source:** la fuente a partir de la cual se genera el evento será la fuente del evento. En el escenario anterior, el mouse es la fuente del evento.
- **Type:** Tipo de evento ocurrido; en el caso de un evento de mouse, el mouse presionado, el mouse soltado son el tipo de eventos.



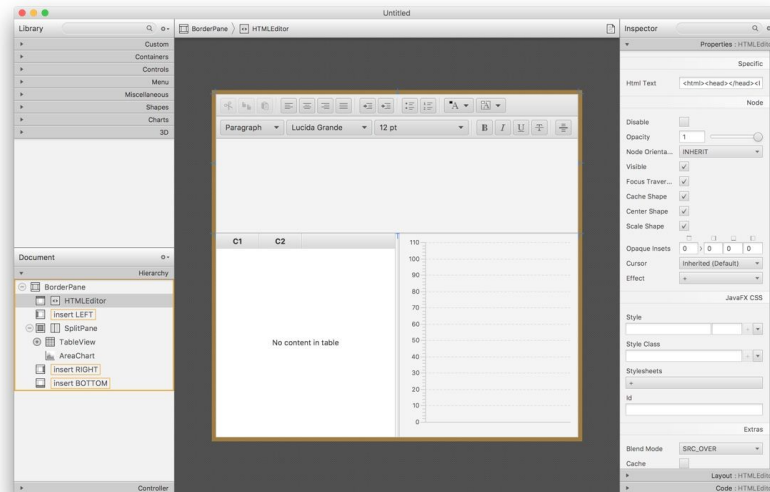
Estructura de una aplicación GUI





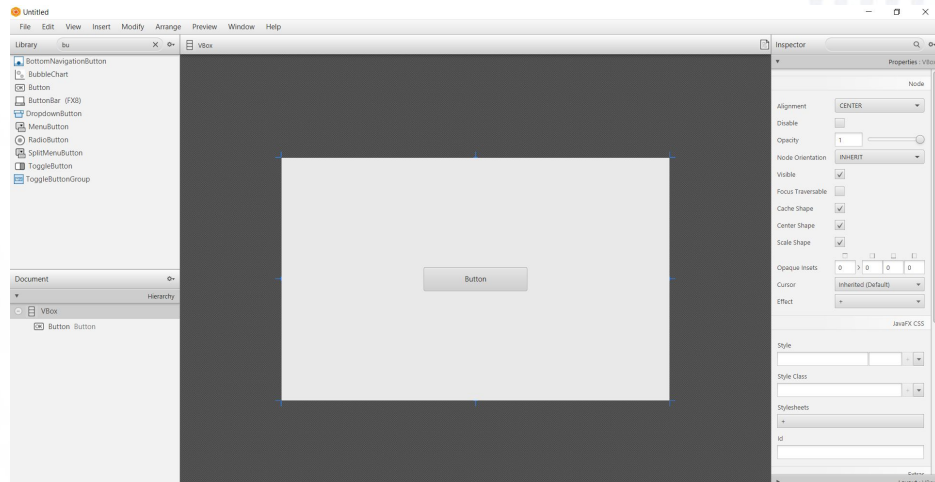
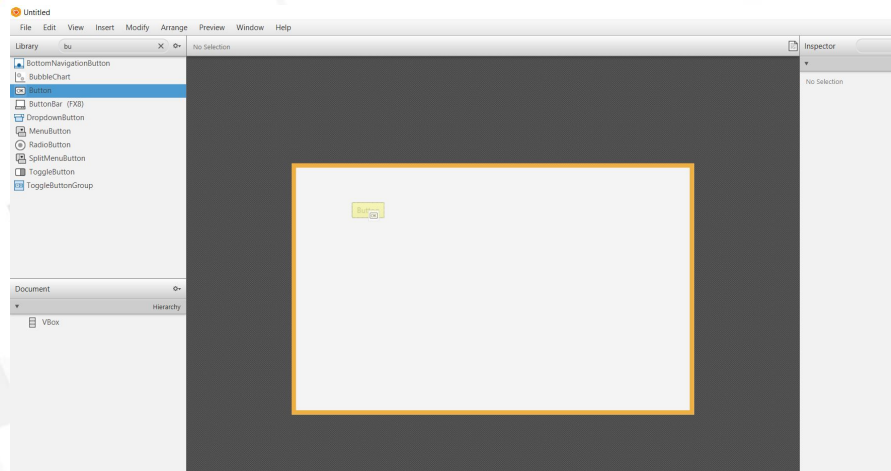
Scene Builder

- Scene Builder permite construir la interfaz gráfica de una aplicación de escritorio Java de forma más sencilla.
- El diseño de la interfaz de usuario de arrastrar y soltar permite una iteración rápida.
- JavaFX Scene Builder genera archivos descriptores FXML que podemos cargar en la aplicación evitando la tediosa y no sencilla tarea de construir la interfaz gráfica mediante código.





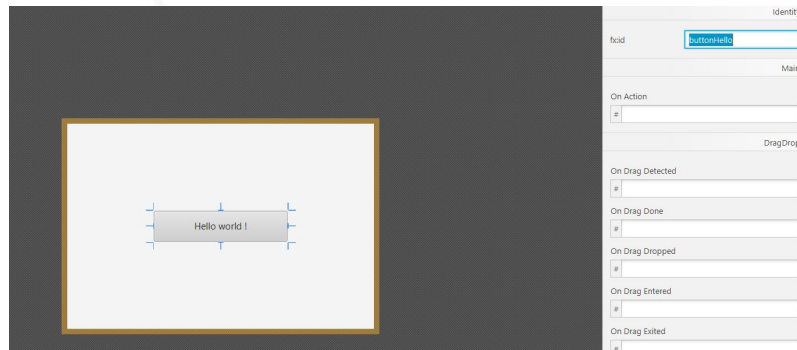
Ejemplo de una aplicación GUI utilizando Scene Builder



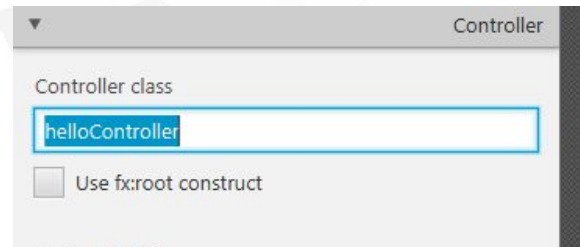
Se construye el layout arrastrando los componentes



Ejemplo de una aplicación GUI utilizando Scene Builder



Asignar identificadores únicos a cada componente

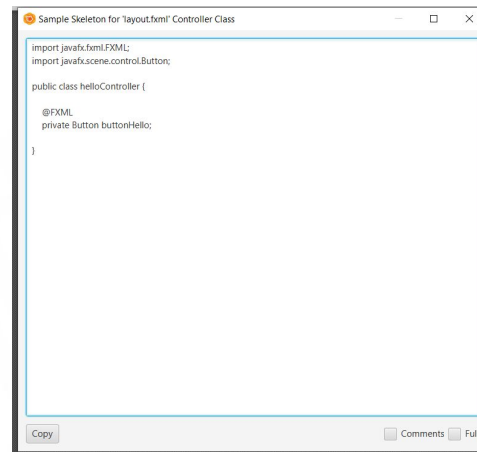
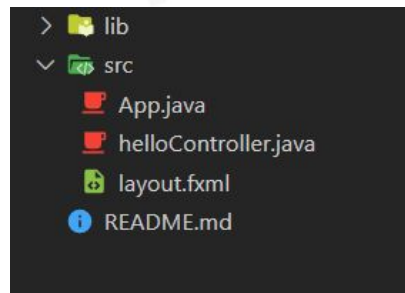


Asignar nombre al controlador



Ejemplo de una aplicación GUI utilizando Scene Builder

Guardar archivo FXML en la carpeta src de
nuestro proyecto en visual studio code



Copiar el esqueleto del controlador sugerido por la
herramienta en nuestro proyecto con el mismo nombre y
extensión .java



Ejemplo de una aplicación GUI utilizando Scene Builder

```
public void start(Stage primaryStage) throws  
Exception {  
  
    //FXMLLoader fxmLoader = new  
    FXMLLoader(getClass().getResource("ruta absoluta  
    del archivo fxm a mostrar"));  
  
    FXMLLoader fxmLoader = new  
    FXMLLoader(getClass().getResource("layout.fxml"));  
  
    Parent root = fxmLoader.load();  
  
    Scene scene = new Scene(root);  
  
  
    primaryStage.setTitle("Hello world!");  
  
    primaryStage.setScene(scene);  
  
    primaryStage.show();  
  
}
```

En el archivo principal App.java se define el layout que tendrá el stage principal

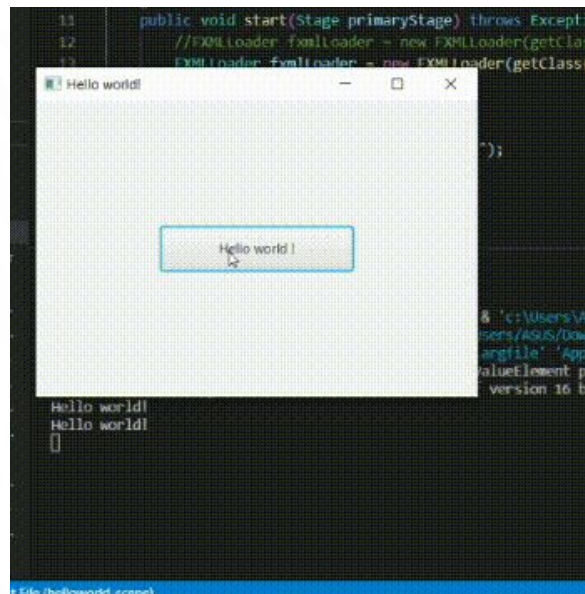


Ejemplo de una aplicación GUI utilizando Scene Builder

Se le agrega el evento `onClick` a el button y se define la acción en el controlador.

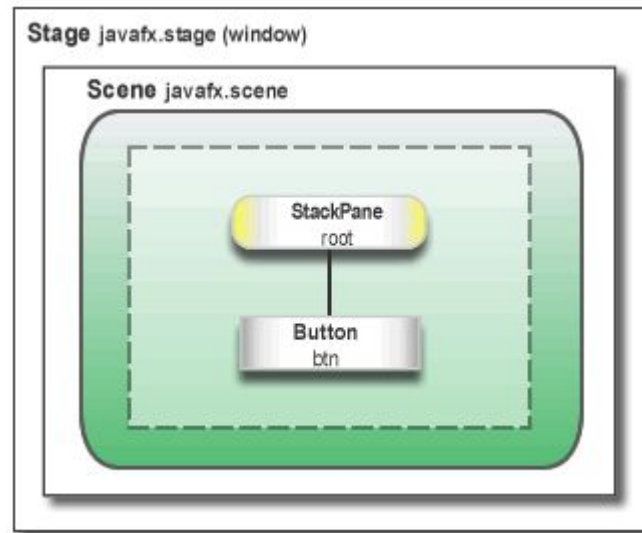
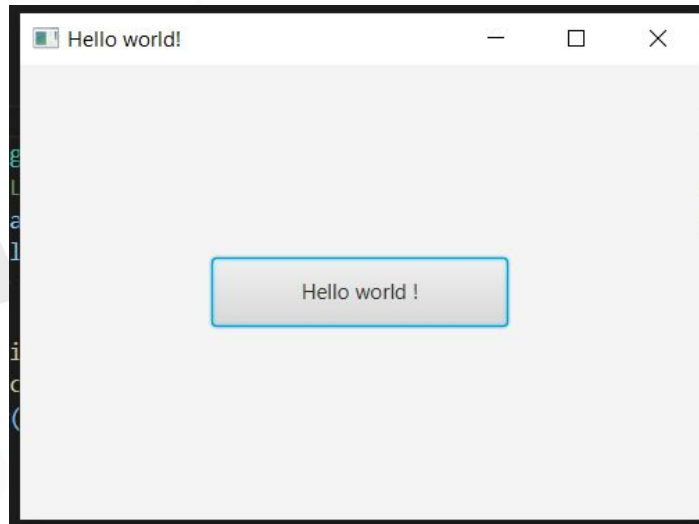
```
public void buttonClicked() {  
    System.out.println("Hello world!");  
}
```

```
onAction="#buttonClicked"
```





Ejemplo de una aplicación GUI utilizando Scene Builder





El futuro digital
es de todos

MinTIC



Vigilada Mineducación

Ejercicios para practicar





El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

¡GRACIAS
POR SER PARTE DE
ESTA EXPERIENCIA
DE APRENDIZAJE!



Misión
TIC 2022