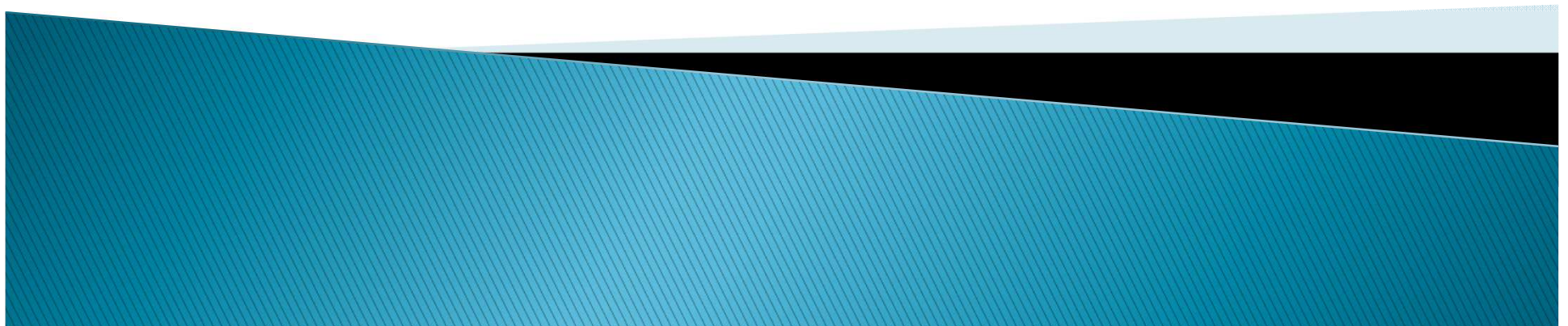




# Introducción a JavaFX 8

Interfaces Persona Computador

Depto. Sistemas Informáticos y Computación. UPV



# Índice

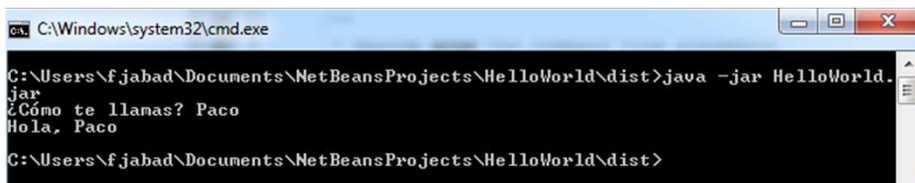
- ▶ Introducción
- ▶ Programación orientada a eventos
- ▶ Introducción a JavaFX
  - Arquitectura de JavaFX
  - Grafo de escena



# Programación de consola vs orientada a eventos

## ▶ Las aplicaciones de consola:

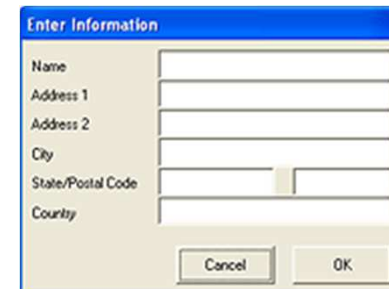
- Realizan su entrada/salida en la consola
- Dirigen el flujo de ejecución del sistema
- Dicho flujo de ejecución es normalmente lineal
- Piden información al usuario y quedan a la espera hasta que la consiguen



```
C:\Windows\system32\cmd.exe
C:\Users\fjabad\Documents\NetBeansProjects\HelloWorld\dist>java -jar HelloWorld.jar
¿Cómo te llamas? Paco
Hola, Paco
C:\Users\fjabad\Documents\NetBeansProjects\HelloWorld\dist>
```

## ▶ Las aplicaciones orientadas a eventos:

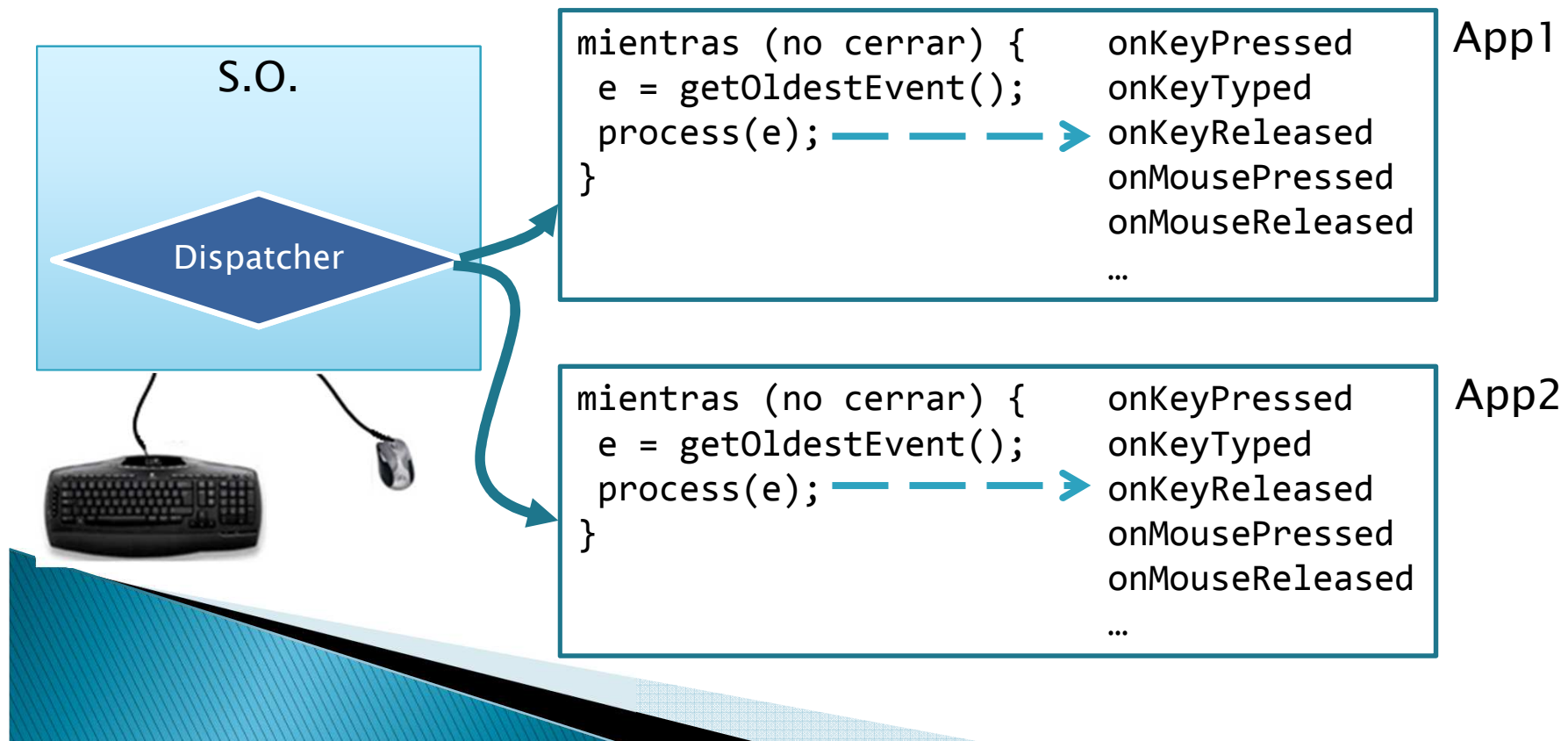
- Dejan que el usuario defina el flujo de ejecución
- Reaccionan a los eventos de entrada ejecutando un pequeño código y devolviendo el control
- Los eventos los genera el sistema (p.e., ha cambiado el tamaño de la ventana), o normalmente el usuario (ha pulsado una tecla, o el texto del control ha cambiado)



The screenshot shows a window titled "Enter Information" with a blue title bar. It contains several text input fields for "Name", "Address 1", "Address 2", "City", "State/Postal Code", and "Country". At the bottom of the window are "Cancel" and "OK" buttons.

# Programación orientada a eventos

- ▶ Las aplicaciones se comunican con el sistema operativo mediante una cola de mensajes
- ▶ El bucle principal de la aplicación se encarga de extraer eventos de la cola e invocar el código que se encarga de tratarlos



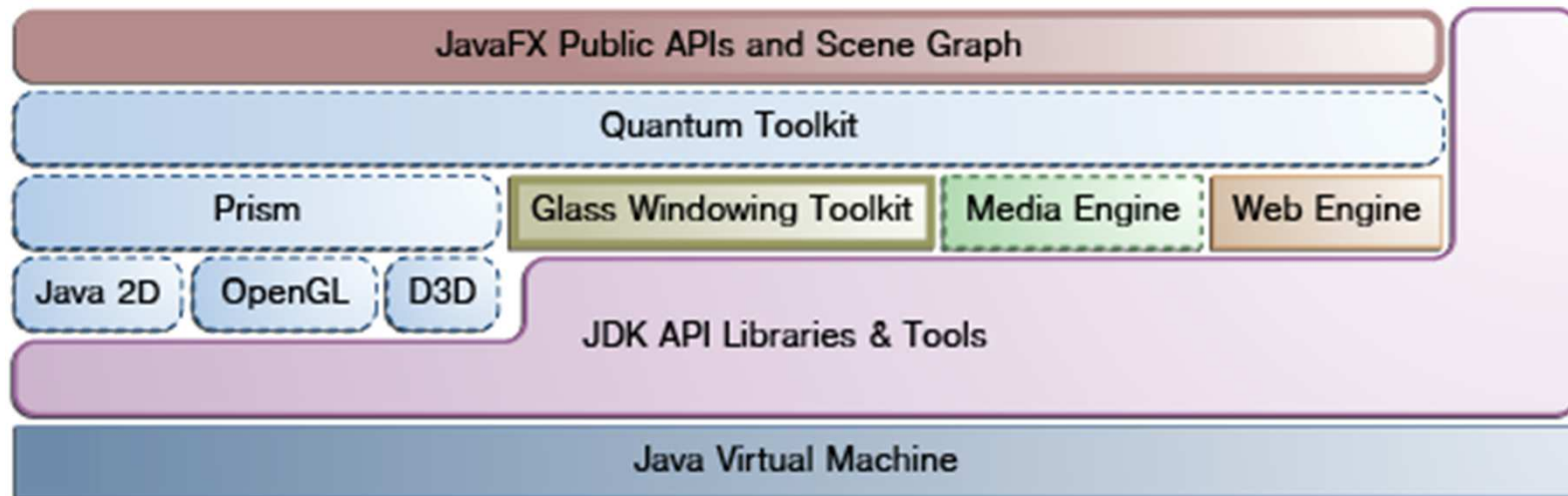
# Introducción

- ▶ JavaFX es una plataforma software para la creación de interfaces gráficas de usuario
- ▶ Está escrito en Java, y es el sustituto de Swing
- ▶ Se puede usar JavaFX para construir aplicaciones para:
  - Escritorio: Windows, Mac OS, Linux, web
  - Plataformas embebidas: Android, iOS, Raspberry, etc.(<http://gluonhq.com/open-source/javafxports>)
- ▶ Está incluido en el Java Runtime Environment (JRE), por lo que el usuario no debe instalar nada más
- ▶ Usa el hardware gráfico disponible para dibujar la interfaz
- ▶ Incorpora funciones para la programación multimedia, gráficos 3D, impresión, web, interfaces táctiles, gestos, etc.



# Arquitectura de JavaFX

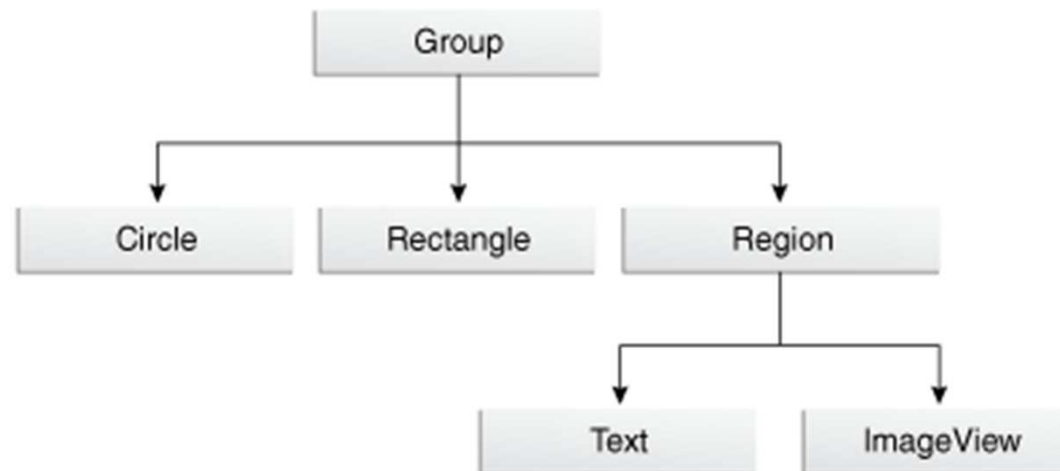
- ▶ Las aplicaciones usan la API pública de JavaFX para construir el grafo de escena, que es la representación jerárquica de la interfaz. Por debajo hay una serie de componentes no accesibles sobre los que se apoya JavaFX.





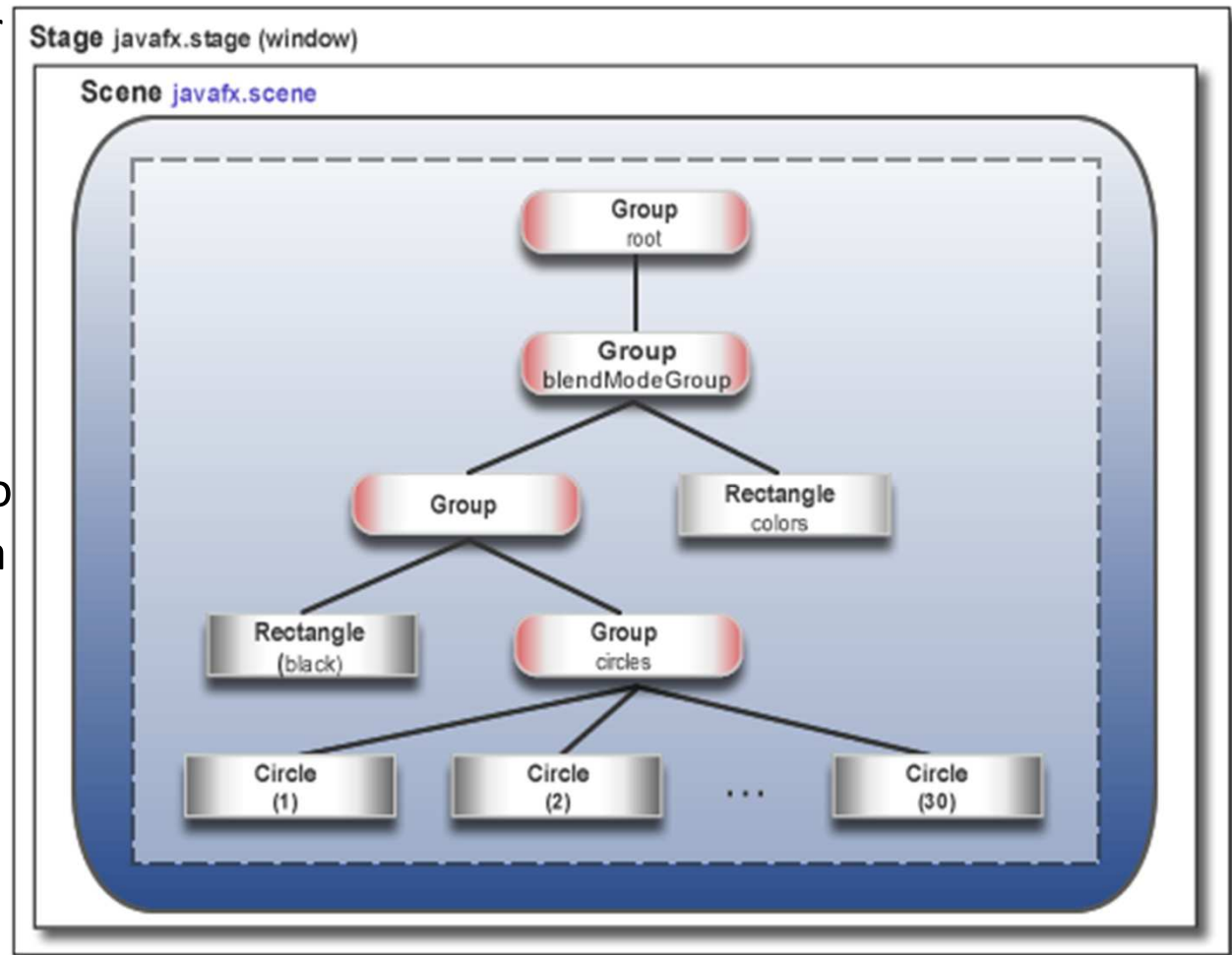
# Arquitectura de JavaFX

- ▶ Grafo de escena (*Scene Graph*)
  - Es una estructura jerárquica (que puede contener varios árboles) que contiene la descripción de los elementos que forman la interfaz gráfica.



# Arquitectura de JavaFX

- ▶ Stage: es el contenedor principal y representa la ventana del S.O.
- ▶ Scene: sirve como contenedor donde colocar el grafo de escena
- ▶ Root: es la raíz del grafo de escena que se desea mostrar

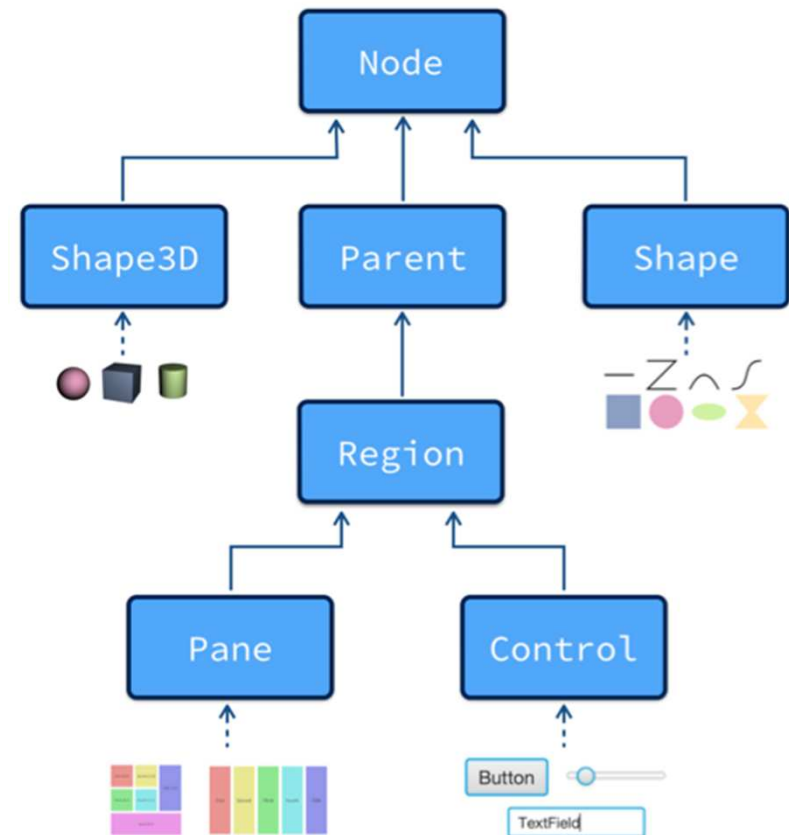




# Arquitectura de JavaFX

## ► Grafo de escena

- Cada nodo en el grafo de escena tiene un identificador, otros atributos y una serie de métodos para manejarlo.
- Hay dos tipos de nodos:
  - Contenedores: de tipo Parent (del que derivan Group, Region y Control, y otros), y que pueden contener 0 o más hijos
  - Hoja: que pueden ser primitivas gráficas como rectángulos, o texto y componentes como imágenes o medios.



<https://dzone.com/refcardz/javafx-8-1>

# Arquitectura de JavaFX

- ▶ Los nodos del grafo de escena tienen:
  - ▶ Propiedades:
    - ▶ Visuales (tamaño, posición, color, etc.)
    - ▶ Contenido (texto, valores, data sets, etc.)
  - ▶ Pueden generar eventos:
    - ▶ Eventos de la interfaz de usuario (ratón pulsado, tecla pulsada, etc.)
    - ▶ Se generan usualmente después de alguna acción del usuario
  - ▶ Manejadores de eventos
    - ▶ Código que se ejecutará ante un evento en la IU (teclado, ratón, etc.)
  - ▶ Estilos de visualización:
    - ▶ Mediante hojas de estilo en cascada (CSS)
- ▶ Se pueden aplicar efectos a los nodos, los objetos pueden cambiar de apariencia, aplicar sombras, ajuste de color
- ▶ Todos los nodos tienen un padre (salvo el nodo raíz) y cero o más hijos



# Ejemplo

```
public class Hola extends Application {  
    @Override  
    public void start(Stage primaryStage) { // ventana principal  
        Text texto= new Text("Hola a TODOS!!!"); // Nodo de tipo Text  
        StackPane raiz = new StackPane(); // Nodo de tipo StackPane, RAIZ  
        raiz.getChildren().add(texto); // texto es ahora hijo de raiz  
        Scene scene = new Scene(raiz, 300, 250); // Escena que muestra raiz  
        primaryStage.setScene(scene); // Mostramos la escena en la ventana  
        primaryStage.show();  
    }  
    public static void main(String[] args) {  
        launch(args);  
    }  
}
```

launch  
invocará a  
start

Punto de  
entrada de  
Java



# Ejemplo



```
public class Hola extends Application {  
    @Override  
    public void start(Stage primaryStage) { // ventana principal  
        Text texto= new Text("Hola a TODOS!!!"); // Nodo de tipo Text  
        StackPane raiz = new StackPane(); // Nodo de tipo StackPane, RAIZ  
        raiz.getChildren().add(texto); // texto es ahora hijo de raiz  
        Scene scene = new Scene(raiz, 300, 250); // Escena que muestra raiz  
        primaryStage.setScene(scene); // Mostramos la escena en la ventana  
        primaryStage.show();  
    }  
    public static void main(String[] args) {  
        launch(args);  
    }  
}
```

Crea los nodos,  
construye el  
árbol y crea la  
escena

Asigna la  
escena a la  
ventana

Dibuja la  
ventana y cede  
el control al  
S.O.

# Construyendo la escena en JavaFX

- ▶ En JavaFX las interfaces se pueden definir de dos formas:
  - mediante código, como en el ejemplo anterior
  - mediante FXML, un lenguaje similar al XML
- ▶ FXML
  - permite describir la interfaz, el aspecto de nuestra aplicación
  - la interfaz se almacena en un fichero aparte, que se lee en tiempo de ejecución para construir la interfaz
  - el fichero FXML tiene formato de texto, y se puede crear a mano o mediante una herramienta *drag and drop* llamada SceneBuilder



# Describiendo la escena en FXML

```
public class HolaFXM extends Application {  
    @Override  
    public void start(Stage stage) throws Exception {  
        Parent raiz =  
            FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));  
        Scene scene = new Scene(raiz, 300, 250);  
        stage.setScene(scene);  
        stage.show();  
    }  
    public static void main(String[] args) {  
        launch(args);  
    }  
}
```



```
<?xml version="1.0" encoding="UTF-8"?>  
...  
<StackPane id="Raiz" prefHeight="200" prefWidth="320"  
  xmlns:fx="http://javafx.com/fxml/1" xmlns="http://javafx.com/javafx/8" >  
  <children>  
    <Text layoutX="110.0" layoutY="97.0" strokeType="OUTSIDE" strokeWidth="0.0"  
      text="Hola a TODOS!!!" id="texto"/>  
  </children>  
</StackPane>
```



# Arquitectura de JavaFX

## Algunos controles de JavaFX

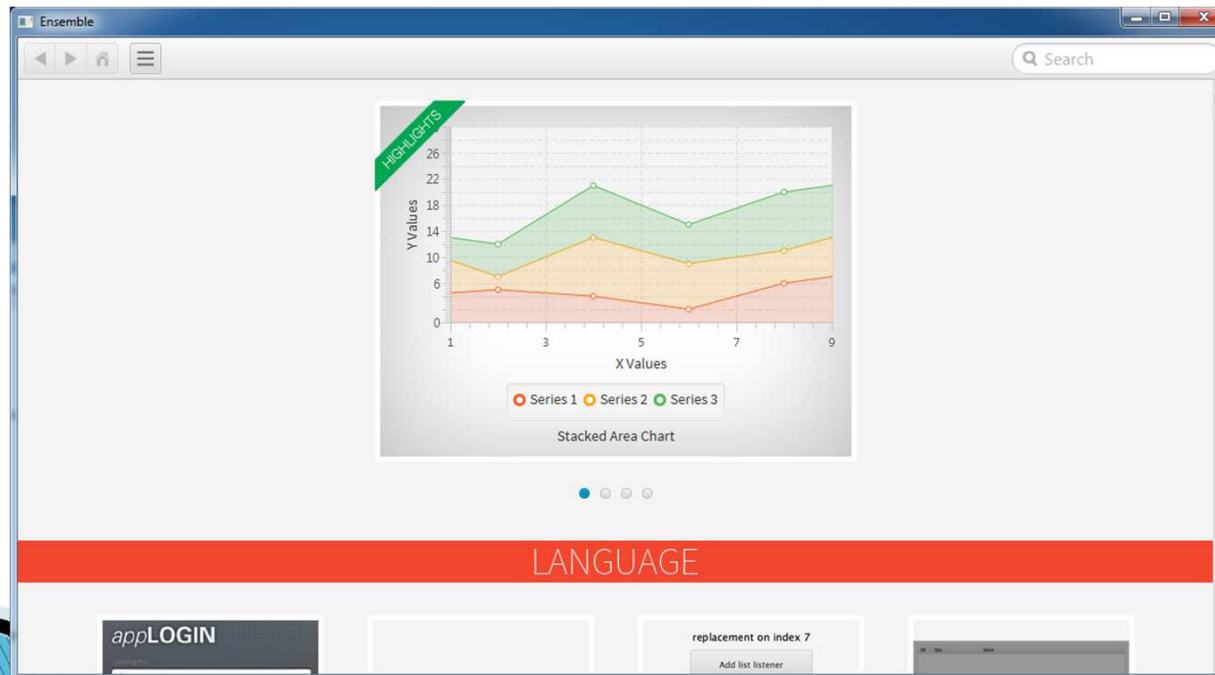


## Algunas gráficas de JavaFX



# Demo

- ▶ JavaFX Ensemble (in Java SE Demos and Samples Downloads)
  - <http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>
  - jdk1.8.xxx\demo\javafx\_samples\Ensemble8.jar
  - Descargar desde Oracle o ejecutar dentro del navegador



# Instalando Netbeans, JDK y Scene Builder

- ▶ Para trabajar en las prácticas en casa, necesitarás instalar el siguiente software:
  - Java JDK 8: (Elije el paquete para tu plataforma, por ejemplo, Windows x64)
    - <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
  - Netbeans: entorno de desarrollo (IDE)
    - <https://netbeans.org>
  - SceneBuilder: editor *drag and drop* de interfaces JavaFX
    - <http://gluonhq.com/open-source/scene-builder>
- ▶ Instalación y primeros pasos en:
  - <https://goo.gl/0BGQeC> (instala las últimas versiones disponibles)
  - <https://goo.gl/vC5H1r>

Revisa estos vídeos  
\*antes\* de ir al  
laboratorio

# Bibliografía

- ▶ Puedes encontrar más información en:
  - <https://docs.oracle.com/javase/8/javase-clienttechnologies.htm>
  - [https://docs.oracle.com/javase/8/javafx/get-started-tutorial/get\\_start\\_apps.htm](https://docs.oracle.com/javase/8/javafx/get-started-tutorial/get_start_apps.htm)
- ▶ Documentación online:
  - Java: <http://docs.oracle.com/javase/8/docs/api>
  - JavaFX: <http://docs.oracle.com/javase/8/javafx/api>
- ▶ Carl Dea y otros. JavaFX 8. Introduction by Example. Apress 2014.

