Elaboración de interfaces mediante documentos XML.

# 1.- Introducción.

JavaFX es una biblioteca de Java multiplataforma utilizada para construir aplicaciones dinámicas de Internet. Se pueden desarrollar con ella aplicaciones para diversos dispositivos, como ordenadores de sobremesa, móviles, televisores, tabletas, etc.

Para desarrollar GUI Applications usando el lenguaje de programación Java, los programadores se basan en las bibliotecas, tales como Advanced Windowing Tool kit y Swings. Después de la llegada de JavaFX, estos programadores de Java pueden ahora desarrollar aplicaciones GUI eficazmente con contenido rico.

**Rich Internet Applications** (RIA) son aquellas **aplicaciones web que proporcionan características y experiencia como la de las aplicaciones de escritorio similar**. Ofrecen una mejor experiencia visual en comparación con las aplicaciones web normales a los usuarios. Estas aplicaciones se entregan como los complementos del navegador o como una máquina virtual.

A diferencia de las aplicaciones de escritorio tradicionales, RIA no requiere tener ningún software adicional para funcionar. Como alternativa, se debería instalar un software como ActiveX, Java, Flash, dependiendo de la aplicación.

En RIA, la representación gráfica se maneja en el lado del cliente gracias al plugin correspondiente, mientras que la manipulación de datos en un RIA se lleva a cabo en el lado del servidor.

Tenemos **tres tecnologías principales para desarrollar un RIA**:

* Adobe Flash.
* Microsoft Silverlight.
* JavaFX.

## 1.1.- Adobe Flash.

Se utiliza en la creación de aplicaciones dinámicas de Internet. Junto con estos, también se puede construir otras aplicaciones, como vector, animación, juegos de navegador, aplicaciones de escritorio, aplicaciones móviles y juegos, etc. Esta es la plataforma más utilizada para el desarrollo y ejecución de Ria con una tasa de penetración en el navegador de escritorio del 96%.

## 1.2.- Microsoft Silverlight.

Inicialmente este marco se utilizó para el streaming de medios de comunicación. Las versiones actuales soporte multimedia, gráficos y animación así.

Esta plataforma se utiliza muy poco (66% de tasa de penetración de mercado).

## 1.3.- JavaFX.

JavaFX es una biblioteca de Java multiplataforma utilizada para construir aplicaciones dinámicas de Internet. Se pueden desarrollar con ella aplicaciones para diversos dispositivos, como ordenadores de sobremesa, móviles, televisores, tabletas, etc.

Para desarrollar GUI Applications usando el lenguaje de programación Java, los programadores se basan en las bibliotecas, tales como Advanced Windowing Tool kit y Swings. Después de la llegada de JavaFX, estos programadores de Java pueden ahora desarrollar aplicaciones GUI eficazmente con contenido rico.

## 1.4.- Necesidad de JavaFX.

## 1.5.- Características de JavaFX.

## 1.5.- Historia de JavaFX.

# 2.- Instalación del entorno de Trabajo.

## 2.1.- Instalación Java8.

## 2.2.- Configuración de la Ruta PATH para Windows.

## 2.3.- Ajustes del entorno de NetBeans para JavaFX.

## 2.4.- La instalación de JavaFX en Eclipse.

# 3.- Arquitectura de JavaFX.

## 3.1.- Escenario gráfico.

## 3.2.- Prisma.

## 3.3.- GWT (Glass Windowing Toolkit).

## 3.4.- Quantum Tookit.

## 3.5.- WebView.

## 3.6.- Media Engine.

# 4.- Esquema de una aplicación básica.

## 4.1.- Escenario.

## 4.2.- Escena.

## 4.3.- Gráfico escena y Nodos.

## 4.4.- Crear una aplicación JavaFX.

## 4.5.- Clase de aplicaciones.

## 4.6.- Preparando el escenario gráfico.

## 4.7.- Preparación de la escena.

## 4.8.- Preparación de la Etapa.

## 4.9.- Ciclo de vida de una aplicación JavaFX.

## 4.10.- Terminar la aplicación JavaFX.

## 4.11.- Ejemplo 1. Creación de una ventana vacía.

### 4.11.1.- Paso 1. Creación de una clase.

### 4.11.2.- Paso 2. Creación de un objeto Grupo.

### 4.11.3.- Paso 3. Crear un objeto de la escena.

### 4.11.4.- Paso 4. Ajuste del título de la Etapa.

### 4.11.5.- Paso 5. Adición de escenas a la Etapa.

### 4.11.6.- Paso 6. Visualización del contenido de la Etapa.

### 4.11.7.- Inicio de la aplicación.

### 4.11.8.- Ejemplo.

## 4.12.- Ejemplo 2. Dibujando una línea recta.

import javafx.application.Application;

import javafx.scene.Group;

import javafx.scene.Scene;

import javafx.scene.shape.Line;

import javafx.stage.Stage;

public class DrawingLine extends Application {

@Override

public void start(Stage stage) {

// Creating a line object

Line line = new Line();

// Setting the properties to a line

line.setStartX(100.0);

line.setStartY(150.0);

line.setEndX(500.0);

line.setEndY(150.0);

// Creating a Group

Group root = new Group(line);

// Creating a Scene

Scene scene = new Scene(root, 600, 300);

// Setting title to the scene

stage.setTitle("Sample application");

// Adding the scene to the stage

stage.setScene(scene);

// Displaying the contents of a scene

stage.show();

}

public static void main(String args[]) {

launch(args);

}

}

## 4.13.- Ejemplo 3. Visualización de texto.

import javafx.application.Application;

import javafx.collections.ObservableList;

import javafx.scene.Group;

import javafx.scene.Scene;

import javafx.scene.text.Font;

import javafx.scene.text.Text;

import javafx.stage.Stage;

public class DisplayingText extends Application {

@Override

public void start(Stage stage) {

// Creating a Text object

Text text = new Text();

// Setting font to the text

text.setFont(new Font(45));

// Setting the position of the text

text.setX(50);

text.setY(150);

// Setting the text to be added

text.setText("Welcome to Tutorialspoint");

// Creating a Group object

Group root = new Group();

// Retrieving the observable list object

ObservableList list = root.getChildren();

// Setting the text object as a node to the group object

list.add(text);

// Creating a scene object

Scene scene = new Scene(root, 600, 300);

// Setting title to the Stage

stage.setTitle("Sample Application");

// Adding scene to the stage

stage.setScene(scene);

// Displaying the contents of the stage

stage.show();

}

public static void main(String args[]) {

launch(args);

}

}