**JavaFx**

[**¿Qué es JavaFX**](https://translate.googleusercontent.com/translate_c?depth=1&hl=es&prev=search&rurl=translate.google.com.co&sl=en&u=http://www.oracle.com/pls/topic/lookup%3Fctx%3Djavase80%26id%3DJFXST784&usg=ALkJrhhOfMe-JLgvAQbI0_YENSuEHd1lIQ)

En este capítulo se proporciona una visión general de los tipos de aplicaciones que se pueden construir utilizando las API de JavaFX, dónde descargar las bibliotecas JavaFX, y un alto nivel de información acerca de la clave JavaFX características que se entregó.

JavaFX es una familia de productos y tecnologías de Sun Microsystems, adquirida por Oracle Corporation, para la creación de Rich Internet Applications (RIAs), esto es, aplicaciones web que tienen las características y capacidades de aplicaciones de escritorio, incluyendo aplicaciones multimedia interactivas.

* [Las aplicaciones JavaFX](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#A1141718)
* [Disponibilidad](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#A1095238)
* [Características principales](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#A1131418)
* [Lo que puedo construir con JavaFX?](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#BABHCIBA)
* [¿Cómo se ejecuta una aplicación de ejemplo?](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#BABHGFAH)
* [¿Cómo realizar una muestra en un IDE?](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#BABGBBDH)
* [¿Cómo se crea una aplicación JavaFX?](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#BABDJEFB)
* [recursos](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#BABJBBHJ)

Las aplicaciones JavaFX

Desde la biblioteca JavaFX está escrito como una API Java, código de aplicación JavaFX puede hacer referencia a las API de cualquier biblioteca Java. Por ejemplo, las aplicaciones JavaFX pueden utilizar las bibliotecas de API de Java para acceder a las capacidades del sistema nativas y conectarse a las aplicaciones de middleware basadas en servidor.

La apariencia de las aplicaciones JavaFX pueden ser personalizados. Cascading Style Sheets (CSS) de apariencia independiente y estilo de la aplicación para que los desarrolladores pueden concentrarse en la codificación. Los diseñadores gráficos pueden personalizar fácilmente la apariencia y el estilo de la aplicación a través de la CSS. Si usted tiene un fondo de diseño web, o si desea separar la interfaz de usuario (UI) y la lógica de fondo, entonces se puede desarrollar los aspectos de la presentación de la interfaz de usuario en el idioma FXML secuencias de comandos y utilizar el código de Java para la aplicación lógica. Si prefiere diseñar interfaces de usuario sin necesidad de escribir código, a continuación, utilizar JavaFX Escena del constructor. A medida que diseña la interfaz de usuario, Escena Builder crea marcado FXML que puede ser portado a un entorno de desarrollo integrado (IDE) de modo que los desarrolladores pueden añadir la lógica de negocio.

Disponibilidad

La compatibilidad entre plataformas permite una experiencia de tiempo de ejecución consistente para JavaFX los desarrolladores de aplicaciones y usuarios. Oracle asegura versiones y actualizaciones sincronizadas en todas las plataformas y ofrece un [amplio programa de apoyo](http://www.oracle.com/us/technologies/java/standard-edition/support/overview/) para las empresas que ejecutan aplicaciones de misión crítica.

En la [página de descarga de JDK](http://www.oracle.com/technetwork/java/javase/downloads/) , se puede obtener un archivo zip de ejemplos de aplicaciones JavaFX. Las aplicaciones de ejemplo proporcionan muchos ejemplos de código y fragmentos que muestran por ejemplo la forma de escribir aplicaciones JavaFX. Ver ["¿Cómo se ejecuta una aplicación de ejemplo?"](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#BABHGFAH) Para obtener más información.

Características principales

Las siguientes características se incluyen en JavaFX 8 y versiones posteriores. Los productos que se introdujeron en JavaFX 8 liberación se indican en consecuencia:

* **API de Java**. JavaFX es una biblioteca Java que se compone de clases e interfaces que están escritas en código Java. Las APIs están diseñados para ser una alternativa respetuosa con las lenguas de Java Virtual Machine (JVM), como JRuby y Scala.
* **FXML y Constructor de escena**. FXML es un lenguaje de marcado declarativo basado en XML para construir una interfaz de usuario de la aplicación JavaFX. Un diseñador puede codificar en FXML o utilizar JavaFX Escena Builder para diseñar de forma interactiva la interfaz gráfica de usuario (GUI). Escena Builder genera marcado FXML que puede ser portado a un IDE que un desarrollador puede agregar la lógica de negocio.
* **WebView**. Un componente web que utiliza la tecnología WebKitHTML para hacer posible embeber páginas web dentro de una aplicación JavaFX. JavaScript se ejecuta en WebView puede llamar a las API de Java y las API de Java puede llamar a JavaScript se ejecuta en WebView. Soporte para funciones adicionales, incluyendo HTML5 Web Sockets, Trabajadores Web y Fuentes web y capacidades de impresión se han añadido en JavaFX 8. Consulte [Adición de contenido HTML para aplicaciones JavaFX](http://docs.oracle.com/javase/8/javafx/embedded-browser-tutorial/overview.htm#JFXWV135) .
* **Interoperabilidad oscilación**. Aplicaciones Swing existentes pueden actualizarse con las características de JavaFX, tales como la reproducción de medios gráficos ricos y contenido Web incorporado. El SwingNode clase, lo que le permite incrustar contenido Entra a aplicaciones JavaFX, se ha añadido en JavaFX 8. Ver la [javadoc API SwingNode](http://docs.oracle.com/javase/8/javafx/api/) y [Inserción de contenido oscilación en aplicaciones JavaFX](http://docs.oracle.com/javase/8/javafx/interoperability-tutorial/embed-swing.htm#JFXIP566) para más información.
* **Incorporado en los controles de interfaz de usuario y CSS**. JavaFX proporciona todos los principales controles de interfaz de usuario que se requieren para desarrollar una aplicación con todas las funciones. Los componentes pueden ser de piel con las tecnologías Web estándar como CSS. Los controles de interfaz de usuario DatePicker y TreeTableView están ahora disponibles con el lanzamiento de JavaFX 8. Ver [Uso de los controles de interfaz de usuario JavaFX](http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/ui_controls.htm#JFXUI336) para más información. Además, las clases CSS Styleable se han convertido API pública, permitiendo que los objetos a ser de estilo de CSS.
* **Módena tema.**El tema de Módena reemplaza el tema Caspio como valor predeterminado para JavaFX 8 aplicaciones. El tema Caspian todavía está disponible para su uso mediante la adición de la setUserAgentStylesheet (STYLESHEET\_CASPIAN) de línea de su método start () de la aplicación. Para obtener más información, consulte el [blog de ​​Módena](http://fxexperience.com/2013/01/modena-new-theme-for-javafx-8/) en fxexperience.com
* **Características de gráficos 3D**. Las nuevas clases de la API para Shape3D ( caja, cilindro, MeshView, y Esfera subclases), subescena, Material, PickResult, LightBase (AmbientLight y PointLight subclases), y SceneAntialiasing se han añadido a la biblioteca de gráficos 3D en JavaFX 8. La cámara de clase API también se ha actualizado en esta versión. Para obtener más información, consulte la [Guía de introducción a JavaFX Gráficos 3D](http://docs.oracle.com/javase/8/javafx/graphics-tutorial/javafx-3d-graphics.htm#JFXGR256)documento y la correspondiente [javadoc API](http://docs.oracle.com/javase/8/javafx/api/) para javafx.scene.shape.Shape3D, javafx.scene.SubScene, javafx.scene.paint.Material, javafx.scene.input.PickResult, y javafx.scene.SceneAntialiasing.
* **API de canvas** . La API de la lona permite dibujar directamente en una zona de la escena JavaFX que consta de un elemento gráfico (nodo).
* **API de impresión.** El javafx.print paquete ha sido añadido en Java SE 8 liberación y proporciona las clases públicas para la [API de JavaFX de impresión](http://docs.oracle.com/javase/8/javafx/api/).
* **Soporte de texto enriquecido**. JavaFX 8 trae soporte para JavaFX textos mejorados, incluyendo texto bi-direccional y secuencias de comandos de texto complejos, tales como Tailandia e hindú en los controles, y de varias líneas, texto multi-estilo en los nodos de texto.
* **Soporte multi-touch**. JavaFX proporciona apoyo para las operaciones de multitoque, basado en las capacidades de la plataforma subyacente.
* **Hi-DPI apoyo.** JavaFX 8 ahora es compatible con Hi-DPI pantallas.
* **Acelerado por hardware canal de gráficos**. JavaFX gráficos se basan en la tubería de proceso de gráficos (Prisma). JavaFX ofrece gráficos suaves que hacen rápidamente a través de Prism cuando se utiliza con una unidad de procesamiento apoyado tarjeta gráfica o de gráficos (GPU). Si un sistema no dispone de una de las GPU recomendados apoyados por JavaFX, entonces los valores predeterminados de prisma en el software de renderizado pila.
* **De alto rendimiento motor de los medios de comunicación**. El ducto multimedia es compatible con la reproducción de contenido web multimedia. Proporciona un marco estable, de baja latencia de los medios de comunicación que se basa en el marco multimedia GStreamer.
* **Autónomo de aplicación de despliegue del modelo.** Paquetes de aplicaciones auto contenidos tienen todos los recursos de la aplicación y una copia privada de los tiempos de ejecución de Java y JavaFX. Se distribuyen como paquetes instalables nativas y proporcionan la misma experiencia de instalación y puesta en marcha como aplicaciones nativas para ese sistema operativo.

Lo que puedo construir con JavaFX?

Con JavaFX, se pueden construir muchos tipos de aplicaciones. Por lo general, se trata de aplicaciones en red en cuenta que se implementan a través de múltiples plataformas y visualización de información en una interfaz de usuario moderna de alto rendimiento que cuenta con audio, vídeo, gráficos y animación.

[La tabla 1-1](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#BABCJEGC) muestra las imágenes de algunas de las aplicaciones JavaFX ejemplo que se incluyen con el JavaFX 8. *N* liberación.

***Tabla 1-1 Ejemplos de aplicaciones JavaFX***

| **Aplicación de ejemplo** | **Descripción** |
| --- | --- |
| versión en miniatura de la instantánea de la aplicación Ensemble. [Descripción de la ilustración de conjunto-small.gif](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/img_text/ensemble-small.htm) | **JavaFX Ensemble** 8  Ensemble8 es una galería de aplicaciones de ejemplo que muestran una gran variedad de características, incluyendo JavaFX animación, gráficos y controles. Puede ver e interactuar con cada muestra que se ejecuta en todas las plataformas, y leer sus descripciones. En las plataformas de escritorio, puede copiar el código fuente de cada muestra, ajustar las propiedades de los componentes de la muestra usados ​​en varias muestras, y siga los enlaces a la documentación de la API relevante cuando se está conectado a la Internet.Ensemble8 también funciona con JavaFX para ARM. |
| versión en miniatura de instantánea de la aplicación de paneles de ventas. [Descripción de la ilustración Modena-sample.gif](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/img_text/modena-sample.htm) | **Módena**  Módena es una aplicación de ejemplo que muestra la apariencia de los componentes de interfaz de usuario utilizando el tema de Módena. Se le da la opción de contrastar temas Módena y Caspio, y explorar diversos aspectos de estos temas. |
| versión en miniatura de instantánea de la aplicación de paneles de ventas. [Descripción de la ilustración de la muestra-3dviewer.gif](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/img_text/sample-3dviewer.htm) | **Visor 3D**  3dviewer es una aplicación de ejemplo que le permite navegar y examinar una escena 3D con un ratón o un trackpad. 3dviewer tiene importadores para un subconjunto de las funciones de archivos OBJ y maya. También se ofrece la posibilidad de importar archivos de animación Maya. (Tenga en cuenta que en el caso de los archivos de Maya, la historia de la construcción debe suprimirse en todos los objetos al guardar como un archivo de Maya.)  3dviewer también tiene la capacidad de exportar el contenido de la escena como archivos Java o FXML. |

¿Cómo se ejecuta una aplicación de ejemplo?

Los pasos de esta sección se explica cómo descargar y ejecutar las aplicaciones de ejemplo que están disponibles como descarga independiente con la Plataforma Java (JDK 8).

**Nota:**

Antes de poder ejecutar una aplicación JavaFX muestra, es necesario tener las bibliotecas de ejecución de JavaFX en su máquina. Antes de continuar con estos pasos, ya sea instalar la [última versión del JDK 8](http://www.oracle.com/technetwork/java/javase/downloads/) o la última versión del [JRE](http://java.com/en/) .

**Para descargar y ejecutar las aplicaciones de ejemplo:**

1. Ir a la página de descargas de Java SE en <http://www.oracle.com/technetwork/java/javase/downloads/> .
2. Desplazarse hacia abajo para localizar el JDK 8 y JavaFX Demos y muestras sección.
3. Haga clic en el Demostraciones y Muestras **Descargar** botón para ir a la página de descargas.
4. En la página de Java SE Development Kit 8 Descargas, desplácese hacia abajo para el Ejemplos y demostraciones Descargas JavaFX sección.
5. Descargar el archivo zip para el sistema operativo correcto y extraer los archivos.

El -muestras JavaFX-8.x directorio se crea y contiene los archivos de las muestras disponibles. Los proyectos de NetBeans para las muestras están en el\ src-muestras JavaFX-8.x directorio.

1. Haga doble clic en el archivo ejecutable para una muestra.

Por ejemplo, para ejecutar la aplicación de ejemplo Ensemble8 pre-construidos, haga doble clic en el Ensemble8.jar archivo.

¿Cómo realizar una muestra en un IDE?

Puede utilizar varios entornos de desarrollo de desarrollo Java para desarrollar aplicaciones JavaFX. Los siguientes pasos se explica cómo ver y ejecutar el código fuente en el IDE NetBeans.

**Para ver y ejecutar el código fuente de la muestra en el IDE NetBeans:**

1. Descargar las muestras, como se describe anteriormente, y extraer los archivos.
2. Desde un NetBeans 7.4 o posterior IDE, cargue el proyecto para la muestra que desea ver.
   1. Desde el **Archivo** menú, seleccione **Abrir proyecto**.
   2. En el **Proyecto Abrir** cuadro de diálogo, vaya al directorio que lista las muestras. La ruta de navegación es como la siguiente:
   3. .. \ Javafx\_samples-8.x- <plataforma> \ JavaFX-muestras-8.x \ src
   4. Seleccione la muestra que desea ver.
   5. Haga clic en el **proyecto abierto** botón.
3. En la ventana Proyectos, haga clic derecho en el proyecto que acaba de abrir y seleccione **Ejecutar**.   
   Observe la ventana de resultados se actualiza y el proyecto de ejemplo se ejecuta y se despliega.

¿Cómo se crea una aplicación JavaFX?

Dado que las aplicaciones JavaFX están escritos en el lenguaje Java, puede utilizar su editor favorito o cualquier entorno de desarrollo integrado (IDE) que soporta el lenguaje Java (como NetBeans, Eclipse o IntelliJ IDEA) para crear aplicaciones JavaFX.

**Para crear aplicaciones JavaFX:**

1. Ir a la página de descargas de Java SE en <http://www.oracle.com/technetwork/java/javase/downloads/> para descargar el Oracle JDK 8 con JavaFX 8. *n* apoyo. Los enlaces a las configuraciones de sistemas certificados y notas de la versión también están disponibles en la página ..
2. Uso [Introducción a JavaFX aplicaciones de ejemplo](http://docs.oracle.com/javase/8/javafx/get-started-tutorial/get_start_apps.htm#BACECIIB) para crear aplicaciones sencillas que demuestran cómo trabajar con presentaciones, hojas de estilo, y los efectos visuales.
3. Utilizar JavaFX Escena Builder para diseñar la interfaz de usuario para su aplicación JavaFX sin necesidad de programación. Usted puede arrastrar y soltar los componentes de interfaz de usuario a un área de trabajo, modificar sus propiedades, se aplican las hojas de estilo, e integrar el código resultante con su lógica de la aplicación.
   1. Descargar la escena del constructor JavaFX JavaFX desde la página de Descargas en<http://www.oracle.com/technetwork/java/javase/downloads/> .
   2. Sigue la guía Introducción a JavaFX Escena Constructor tutorial para aprender más.

Recursos

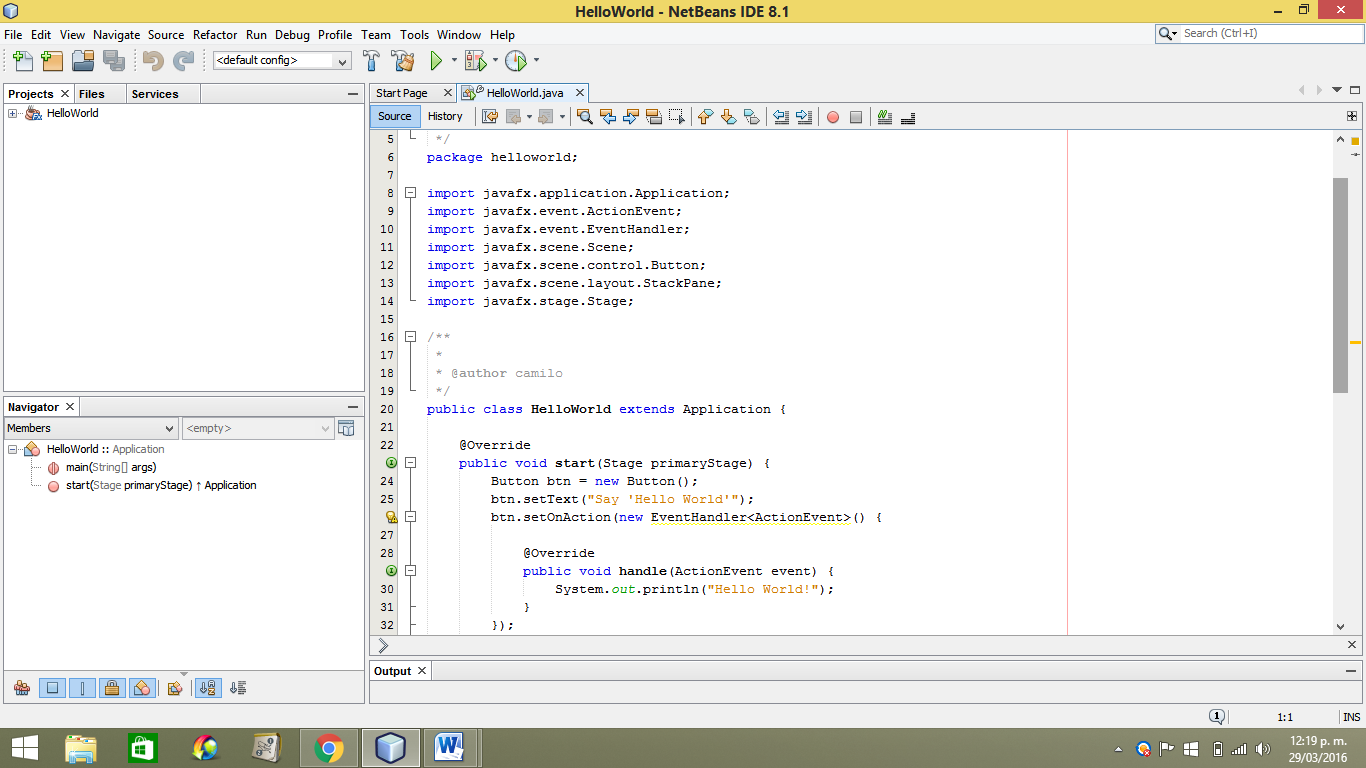
Utilice los siguientes recursos para aprender más sobre la tecnología JavaFX.

* Descargar la última versión del JDK 8 y las muestras de JavaFX desde la página de descargas de Java SE en: <http://www.oracle.com/technetwork/java/javase/downloads/> .

Solución parte 1

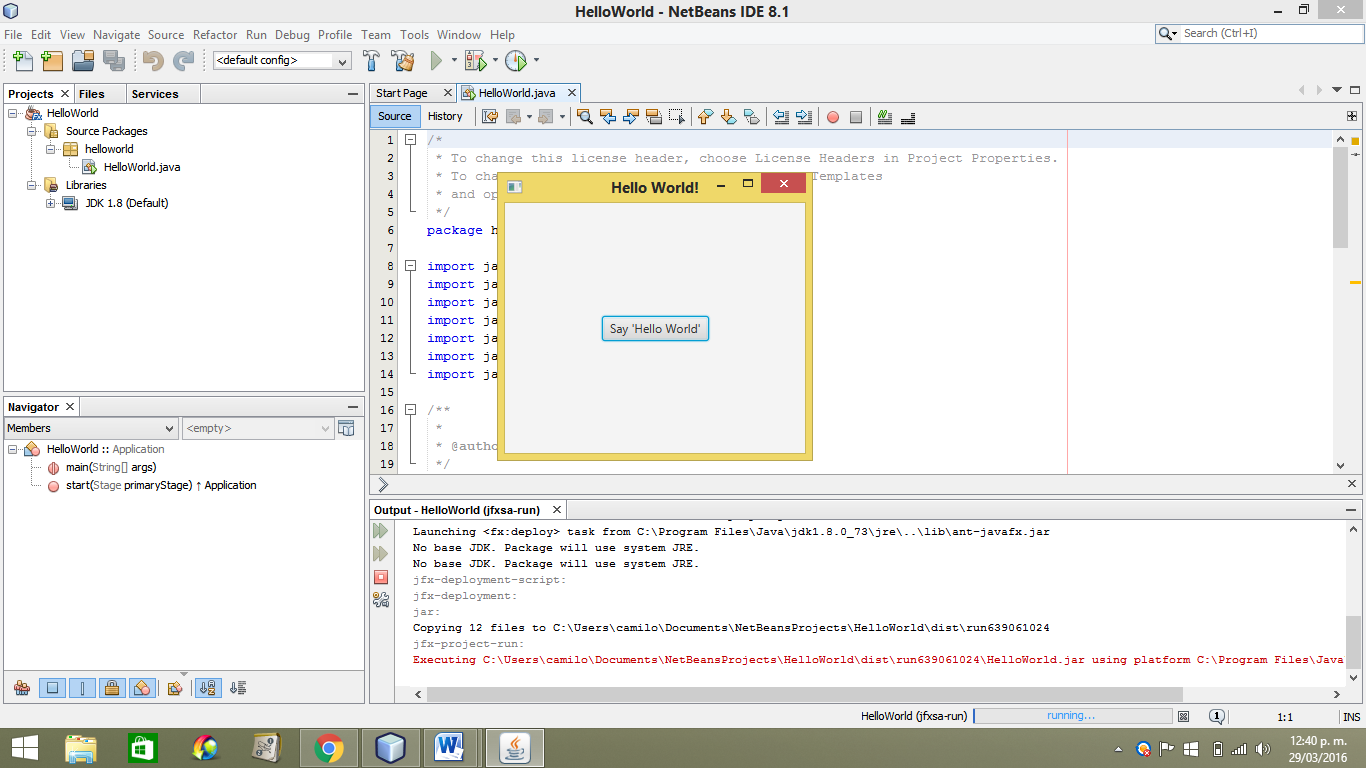
Creación del proyecto hello world javafx

La mejor manera de enseñarle lo que es como para crear y construir una aplicación JavaFX es con una aplicación " Hello World " . Un beneficio adicional de este tutorial es que le permite probar que su tecnología JavaFX está instalado correctamente .

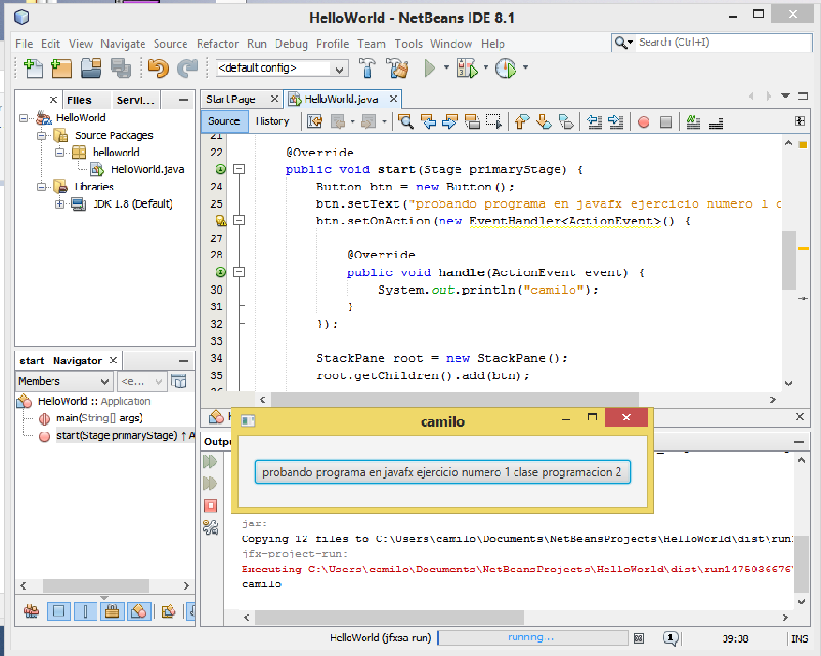


Proyecto hello world javafx reproduciendo

Se creó un programa de una ventana personalizada con un botón para seleccionar y se imprime en el programa netbeans



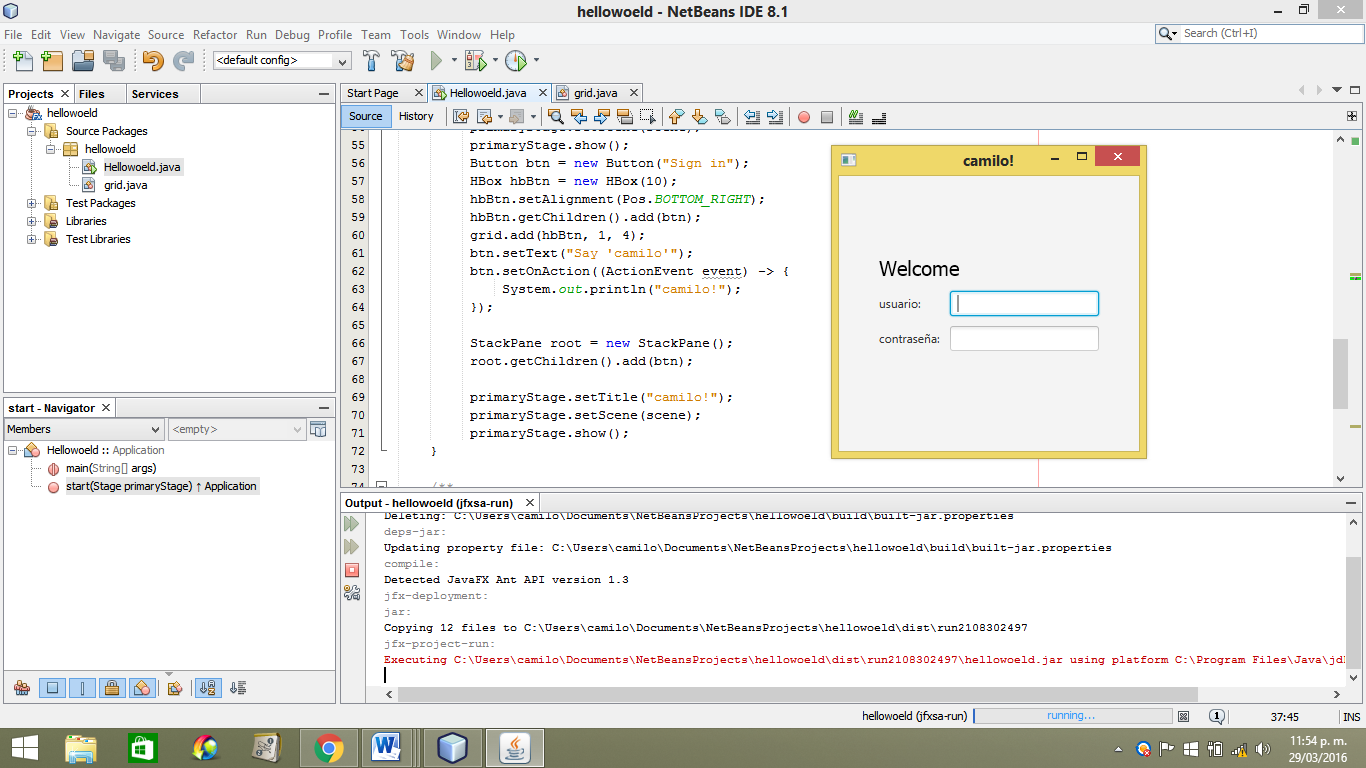
Ejemplo:



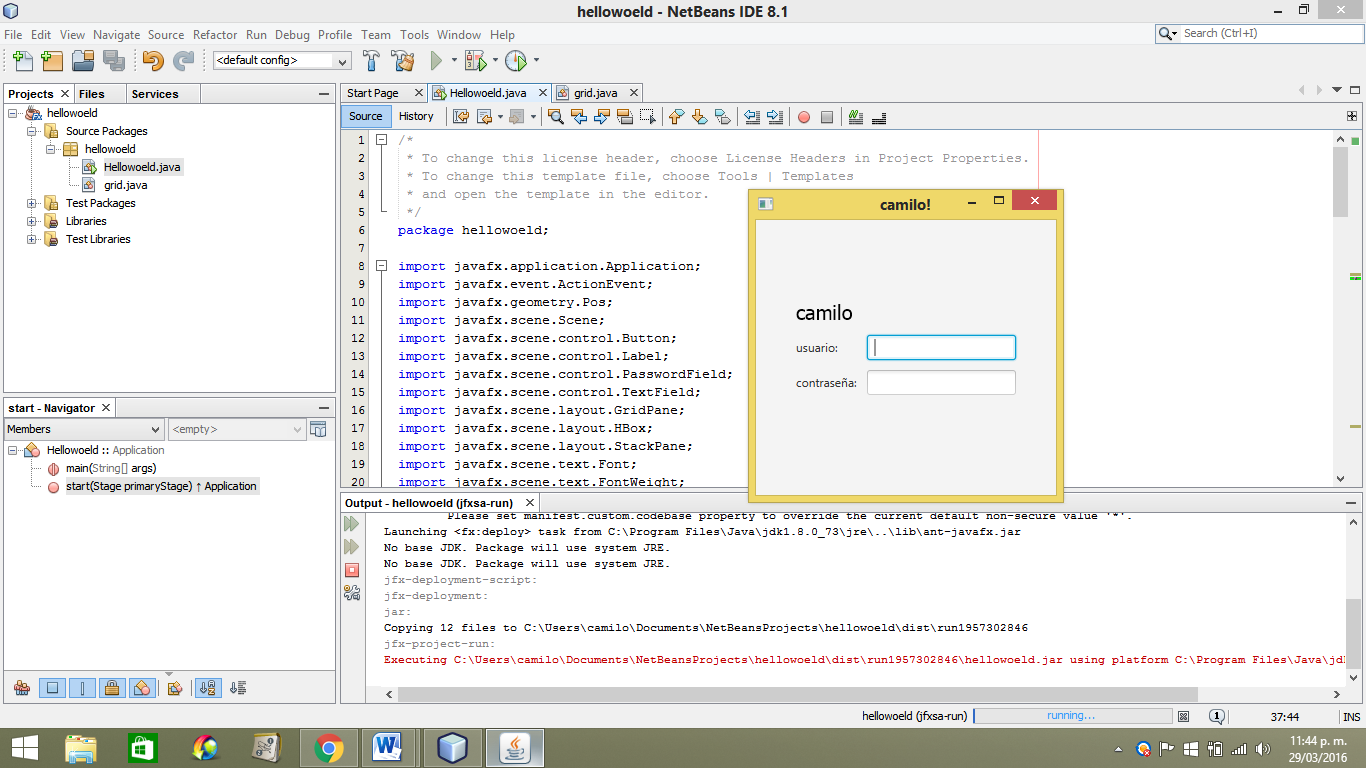
Segunda parte 2:

Creación de un formulario es una actividad común en el desarrollo de una aplicación. Este tutorial te enseña los conceptos básicos de diseño de la pantalla , cómo añadir controles a un panel de diseño , y cómo crear eventos de entrada, se va a utilizar JavaFX para construir la forma de la conexión

Creación de un formulario en JavaFX



Usuario personalizado:



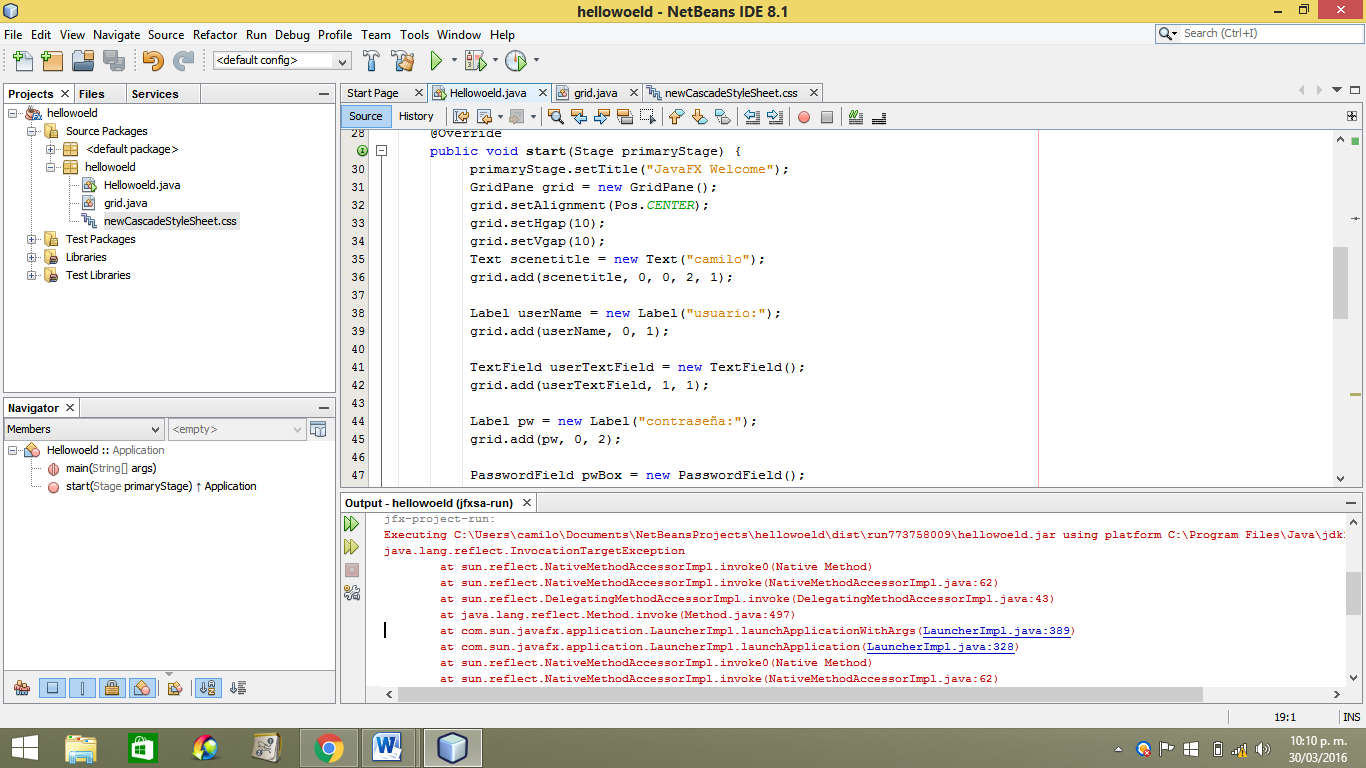
Tercera parte 3:

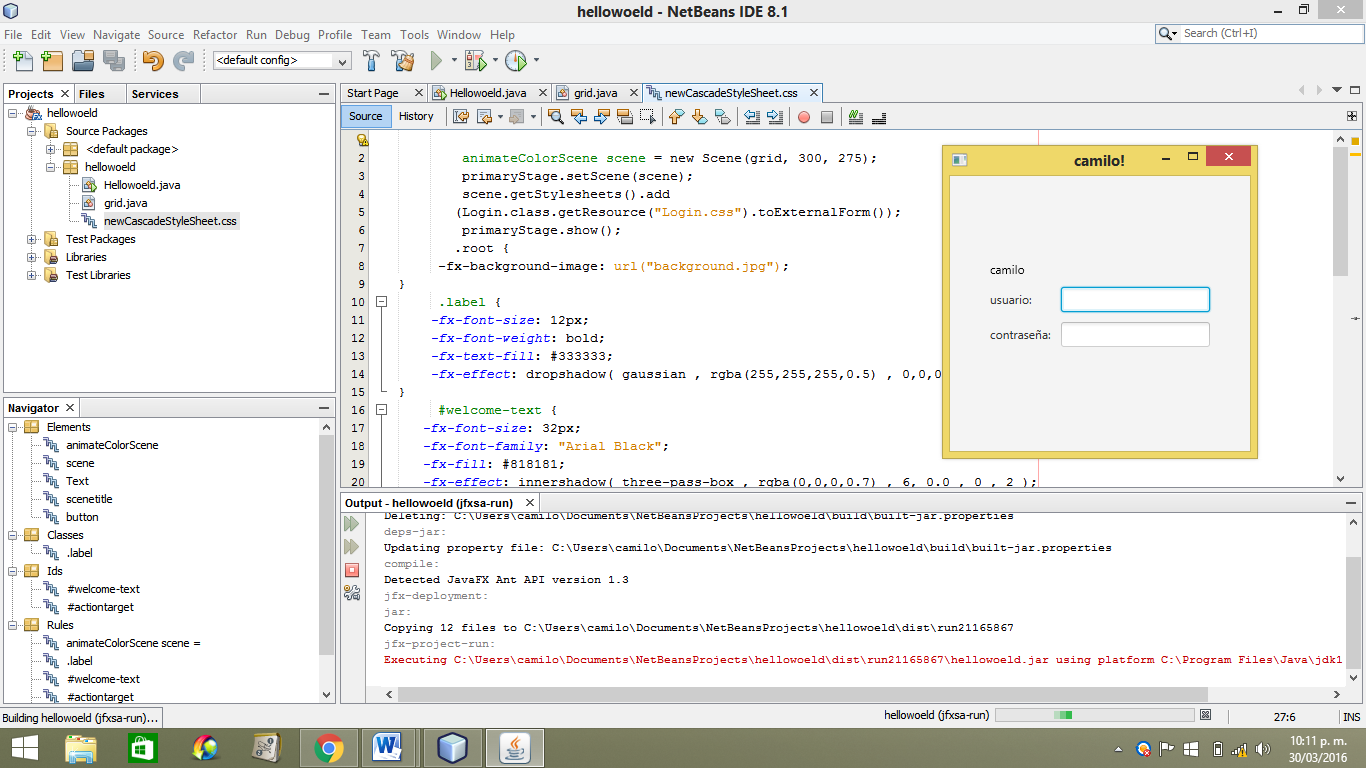
Formas de lujo con JavaFX:

se trata de hacer su aplicación JavaFX, es atractivo mediante la adición de una hoja de estilo en cascada ( CSS ) . A desarrollar un diseño, crea un archivo .css , y aplicar los nuevos estilos, se llevará a un formulario de acceso que utiliza estilos predeterminados para etiquetas , botones, y el color de fondo , y , con algunas modificaciones simples CSS , convertirlo en una aplicación estilizada

ejemplo:

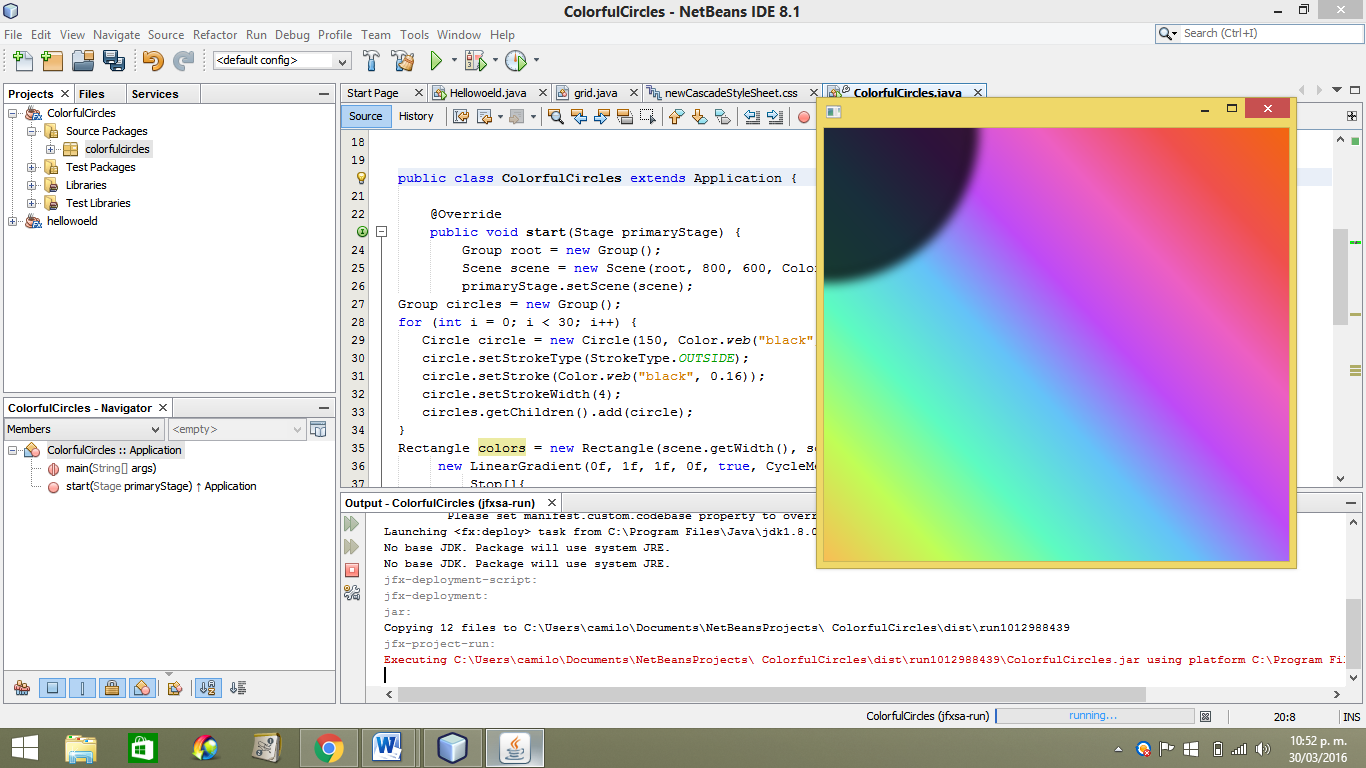






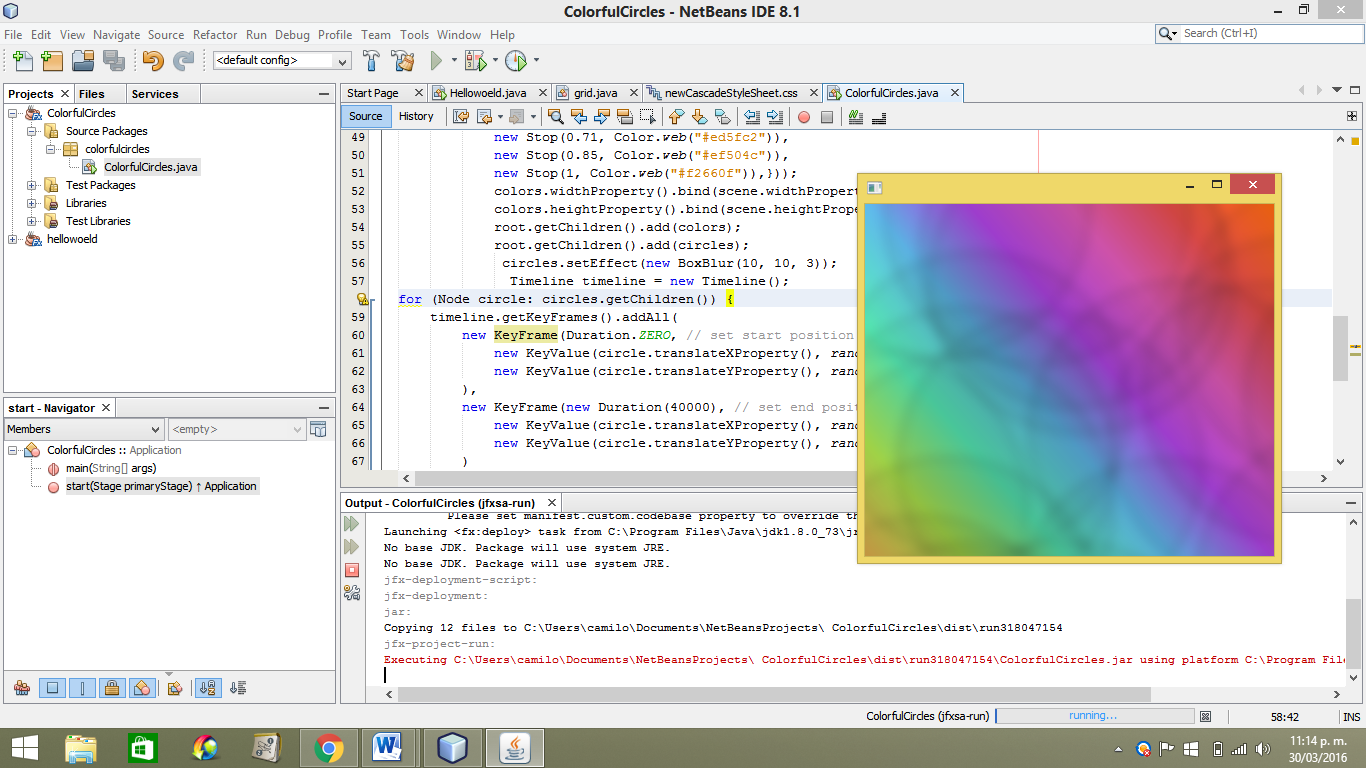
**Animación y efectos visuales en JavaFX**

Puede utilizar JavaFX para desarrollar rápidamente aplicaciones con experiencias de usuario ricas . En este tutorial de introducción , usted aprenderá a crear objetos animados y lograr efectos complejos con muy poca codificación . La Figura 7-1 muestra la aplicación que se creará. Figura 7-1 círculos de colores Aplicación



**Animación es impulsado por una línea de tiempo ,**

por lo que este código se crea una línea de tiempo , a continuación, utiliza un bucle for para añadir dos fotogramas clave para cada uno de los 30 círculos . El primer fotograma clave a 0 segundos utiliza las propiedades translateXProperty y translateYProperty para establecer una posición aleatoria de los círculos dentro de la ventana. El segundo fotograma clave en 40 segundos hace lo mismo. Por lo tanto , cuando se reproduce la línea de tiempo , se anima a todos los círculos de una posición al azar a otro durante un período de 40 segundos . La figura 7-8 muestra los 30 círculos de colores en movimiento, que completa la aplicación. Para el código fuente completo , consulte el archivo ColorfulCircles.java , que se incluye en el archivo descargable ColorfulCircles.zip



Guía de implementación

Título e información de copyright Prefacio Audiencia Accesibilidad documentación Documentos relacionados Convenciones 1 Novedades

1.1 Mejoras en JDK 8u40

1.2 Mejora en el JDK 8u31

1.3 Mejoras en JDK 8u20

1.4 Mejoras en JDK 8u11

1.5 Mejoras en JDK 8u5

1.6 Mejoras en JDK 8 Parte I Fundamentos de implementación

2 Primeros pasos

2.1 Pasos básicos

2.2 Selecciona el entorno de ejecución

2.3 Crear el paquete

2.3.1 Herramientas de Presentación

2.4 Creación de la Página Web

2,5 distribuir su aplicación

2.6 Conceptos más avanzados

3 modos de ejecución de aplicaciones

3.1 modos de ejecución

3.2 Las diferencias de las características Descripción

3.2.1 Soporte para aplicaciones JavaFX Precargador

3.2.2 Soporte de proxy incorporado

3.2.3 Integración de escritorio a través de acceso directo

3.2.4 Ejecutar en el recinto de seguridad

3.2.5 Actualizaciones automáticas

3.2.6 Deployment Toolkit

3.2.7 Comunicar a la página Web Host

3.2.8 Gestión de la plataforma Dependencias

4 de inicio de aplicaciones Consideraciones Experiencia del usuario

4.1 4.2 Proceso de inicio de aplicaciones, experiencia y personalización Proceso

4.2.1 De inicio

4.2.2 La experiencia de usuario predeterminado

4.2.3 Opciones de personalización para aplicaciones JavaFX

4.2.4 Opciones de personalización para los applets de Java

4.3 Los usuarios Ayudar a iniciar la aplicación

4.3.1 No se JRE

4.3.2 Errores de ejecución Parte II Packaging

5 Fundamentos de embalaje

5.1 Java Resumen Embalaje

5.2 Base del paquete de aplicaciones

5.3 Visión general de las tareas de embalaje

5.3.1 Java Herramientas de Presentación

5.4 Conversión de hoja de estilos

5.5 Creación del archivo JAR de la aplicación Principal

5.6 Firma el JAR archivos

5.7 Ejecutar la tarea de implementar o Comando

5.7.1 Configurar el descriptor de despliegue

5.7.2 recursos de aplicaciones

5.7.3 Acciones paquete personalizado de JavaScript

5.7.4 Plantillas de página Web Cookbook

5.8 Embalaje

5.8.1 Paso de parámetros a la Aplicación Configuración

5.8.2 Personalización de JVM

5.8.3 Embalaje Solicitudes Complejas

5.8.4 Publicación de una aplicación que llena la ventana del navegador Sintonía

5.9 Rendimiento de implementación web

5.9.1 Antecedentes comprobar la actualización de la Aplicación

5.9.2 Insertar el descriptor de despliegue en la página web

5.9.3 Insertar certificado de firma en descriptor de despliegue Método de firma

5.9.4 Uso Alternativo

6 JavaFX en aplicaciones Swing

6.1 Información general

6.2 Embalaje con JavaFX las tareas Ant

6.2.1 Habilitación de una pantalla HTML Splash

6.3 Embalaje sin las herramientas de empaquetado

6.3.1 Uso del Deployment Toolkit

[**Los pasos para familiarizarse con JavaFX Arquitectura**](https://translate.googleusercontent.com/translate_c?depth=1&hl=es&prev=search&rurl=translate.google.com.co&sl=en&u=http://www.oracle.com/pls/topic/lookup%3Fctx%3Djavase80%26id%3DJFXST788&usg=ALkJrhi4vaFFC77ZGNMMvN_GZLxxrNzHDQ)

# Entender la Arquitectura JavaFX

El capítulo proporciona una descripción de alto nivel de la arquitectura JavaFX.

Las secciones describen el diagrama de cada componente y cómo las partes de interconexión. Por debajo de las API públicas JavaFX se encuentra el motor que funciona su código JavaFX. Se compone de subcomponentes que incluyen un motor de gráficos de alto rendimiento JavaFX, llamado Prism.

## Gráfico escena

Es un árbol jerárquico de nodos que representa a todos los elementos visuales de la interfaz de usuario de la aplicación. Se puede manejar la entrada y puede ser prestado.

Un solo elemento en un escenario gráfico se llama un nodo. Cada nodo tiene un ID, clase de estilo, y el volumen de delimitación. Con la excepción del nodo raíz de un escenario gráfico, cada nodo en un gráfico de escena tiene un solo padre y cero o más hijos. También puede tener lo siguiente:

* Efectos, como los desenfoques y sombras
* Opacidad
* transformadas
* Los controladores de eventos (como el ratón, y la clave de método de entrada)
* Un estado de aplicación específica
* **Nodos:** Formas (2-D y 3-D), imágenes, multimedia, navegador web integrado, texto, controles de interfaz de usuario, gráficos, grupos y contenedores
* **Estado:** Transformadas (posicionamiento y orientación de nodos), efectos visuales, y otra del estado visual del contenido
* **Efectos:** Objetos simples que cambian la apariencia de los nodos del escenario gráfico, como los desenfoques, sombras y ajuste de color

package scenegraphdemo;

import javafx.application.Application;

import javafx.scene.Group;

import javafx.scene.Scene;

import javafx.scene.paint.Color;

import javafx.stage.Stage;

public class Main extends Application {

@Override

public void start(Stage stage) {

Group root = new Group();

Scene scene = new Scene(root, 500, 500, Color.BLACK);

stage.setTitle("JavaFX Scene Graph Demo");

stage.setScene(scene);

stage.show();

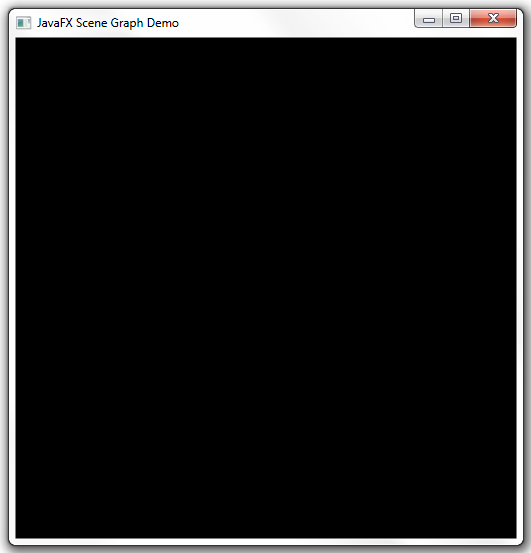
}

public static void main(String[] args) {

launch(args);

}

}



package scenegraphdemo;

import javafx.application.Application;

import javafx.scene.Group;

import javafx.scene.Scene;

import javafx.scene.paint.Color;

import javafx.scene.shape.Rectangle;

import javafx.stage.Stage;

public class Main extends Application {

@Override

public void start(Stage stage) {

Group root = new Group();

Scene scene = new Scene(root, 500, 500, Color.BLACK);

Rectangle r = new Rectangle(25,25,250,250);

r.setFill(Color.BLUE);

root.getChildren().add(r);

stage.setTitle("JavaFX Scene Graph Demo");

stage.setScene(scene);

stage.show();

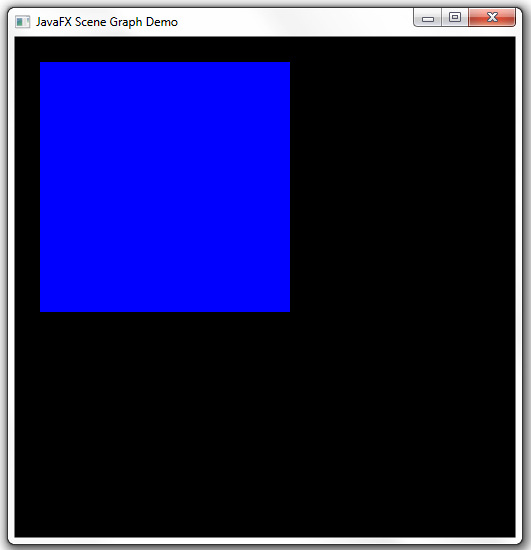
}

public static void main(String[] args) {

launch(args);

}

}



package scenegraphdemo;

import javafx.animation.FillTransition;

import javafx.application.Application;

import javafx.scene.Group;

import javafx.scene.Scene;

import javafx.scene.paint.Color;

import javafx.scene.shape.Rectangle;

import javafx.stage.Stage;

import javafx.animation.Timeline;

import javafx.animation.ParallelTransition;

import javafx.animation.RotateTransition;

import javafx.animation.ScaleTransition;

import javafx.animation.TranslateTransition;

import javafx.util.Duration;

public class Main extends Application {

@Override

public void start(Stage stage) {

Group root = new Group();

Scene scene = new Scene(root, 500, 500, Color.BLACK);

Rectangle r = new Rectangle(0, 0, 250, 250);

r.setFill(Color.BLUE);

root.getChildren().add(r);

TranslateTransition translate =

new TranslateTransition(Duration.millis(750));

translate.setToX(390);

translate.setToY(390);

FillTransition fill = new FillTransition(Duration.millis(750));

fill.setToValue(Color.RED);

RotateTransition rotate = new RotateTransition(Duration.millis(750));

rotate.setToAngle(360);

ScaleTransition scale = new ScaleTransition(Duration.millis(750));

scale.setToX(0.1);

scale.setToY(0.1);

ParallelTransition transition = new ParallelTransition(r,

translate, fill, rotate, scale);

transition.setCycleCount(Timeline.INDEFINITE);

transition.setAutoReverse(true);

transition.play();

stage.setTitle("JavaFX Scene Graph Demo");

stage.setScene(scene);

stage.show();

}

public static void main(String[] args) {

launch(args);

}

}

## API Java Públicas en Equipamiento JavaFX

Estas API proporcionan la libertad y flexibilidad sin precedentes para construir aplicaciones de cliente enriquecido. La plataforma JavaFX combina las mejores capacidades de la plataforma Java con funcionalidad integrales

## Sistema de gráficos

Es un detalle de implementación debajo de la capa del escenario gráfico JavaFX. Es compatible tanto con escenarios gráficos 2-D y 3-D. Proporciona la representación del software cuando el hardware de gráficos en un sistema es insuficiente para soportar el hardware de renderizado acelerado.

## Kit de herramientas de vidrio de ventanas

Es el nivel más bajo de la pila de gráficos JavaFX. Su principal responsabilidad es la de proporcionar servicios operativos nativos, tales como la gestión de las ventanas, temporizadores, y las superficies. Sirve como la capa dependiente de la plataforma que conecta la plataforma JavaFX para el sistema operativo nativo.

### **Trapos**

El sistema funciona de dos o más de los siguientes hilos en cualquier momento dado.

* **Subproceso de la aplicación JavaFX:** Este es el hilo principal utilizado por los desarrolladores de aplicaciones JavaFX. Cualquier escena en vivo
* **Prisma render hilo:** Este hilo se encarga de la prestación por separado del despachador de eventos. Permite la trama N, mientras que pasarán a ser la trama N + 1 se está procesando. Esta capacidad de realizar el procesamiento simultáneo es una gran ventaja, especialmente en sistemas modernos que tienen varios procesadores.
* **Hilo de los medios de comunicación:** Este hilo se ejecuta en segundo plano y sincroniza las últimas tramas a través del escenario gráfico utilizando el hilo de aplicaciones JavaFX.

### **Legumbres**

Es un evento que indica al escenario gráfico JavaFX que es el momento para sincronizar el estado de los elementos en el escenario gráfico con Prisma. Un pulso es estrangulado a 60 cuadros por segundo máximo (fps) y se dispara cada vez que las animaciones se están ejecutando en el escenario gráfico. Incluso cuando no se está ejecutando la animación, un pulso se programa cuando se cambia algo en el escenario gráfico. Por ejemplo, si se cambia una posición de un botón, está previsto un pulso.

Cuando se dispara un pulso, el estado de los elementos en el escenario gráfico está sincronizada hacia abajo a la capa de representación. Un pulso permite a los desarrolladores de aplicaciones un método para controlar los eventos de forma asíncrona. Esta importante característica permite que el sistema de lotes y ejecutar eventos en el pulso.

## Medios e Imágenes

La funcionalidad multimedia JavaFX está disponible a través de los javafx.scene.media API. JavaFX es compatible con los medios visuales y de audio. Se proporciona soporte para archivos de audio MP3, AIFF, y WAV y archivos de vídeo FLV. la funcionalidad multimedia JavaFX se ofrece como tres componentes separados: el objeto de Medios representa un archivo multimedia, el MediaPlayer reproduce un archivo multimedia, y un MediaView es un nodo que muestra los medios de comunicación.

## Componente web

El componente web es un control JavaFX interfaz de usuario, basado en Webkit, que proporciona un visor Web y navegación por completo a través de su API. Este componente del motor Web, que se muestra en color naranja se basa en WebKit, que es un motor de renderizado de código abierto que soporta HTML5, CSS, JavaScript, DOM, y SVG. Permite a los desarrolladores para implementar las siguientes características en sus aplicaciones Java:

* Representar el contenido HTML de URL local o remota
* Historial de soporte y proporcionar Atrás y navegación hacia adelante
* Actualizar el contenido
* Aplicar efectos al componente Web
* Editar el contenido HTML
* Ejecutar comandos de JavaScript
* gestionar eventos

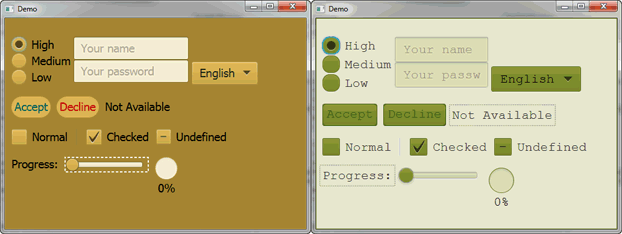
Este componente navegador integrado se compone de las siguientes clases:

* WebEngine proporciona capacidades básicas de navegación página web.
* WebView encapsula un objeto WebEngine, incorpora el contenido HTML en escena de una aplicación, y proporciona campos y métodos para aplicar efectos y transformaciones. Es una extensión de un Node clase.

## CSS

JavaFX Cascading Style Sheets (CSS) ofrece la posibilidad de aplicar un estilo personalizado a la interfaz de usuario de una aplicación JavaFX sin cambiar ninguna de código fuente de la aplicación. CSS se puede aplicar a cualquier nodo en el gráfico de la escena JavaFX y se aplican a los nodos de forma asincrónica. JavaFX estilos CSS también se pueden asignar fácilmente a la escena en tiempo de ejecución, lo que permite la aparición de una aplicación para cambiar dinámicamente.

***hoja de estilo***



## Los controles de interfaz de usuario

Los controles de interfaz de usuario JavaFX disponibles a través de la API de JavaFX se construyen mediante el uso de nodos en el grafo de la escena. Pueden sacar el máximo provecho de las características de gran riqueza visual de la plataforma JavaFX y son portátiles a través de diferentes plataformas. JavaFX CSS permite a temas y máscaras de los controles de interfaz de usuario.

***controles de la muestra***



## Diseño

recipientes o paneles de diseño pueden ser utilizadas para permitir a soluciones flexibles y dinámicas de los controles de interfaz de usuario dentro de un escenario gráfico de una aplicación JavaFX. La API de JavaFX Diseño incluye las siguientes clases de contenedores que automatizan los modelos de diseño comunes:

* El BorderPane clase expone sus nodos de contenido en la parte superior, inferior, derecha, izquierda, centro o región.
* El HBox clase organiza sus nodos de contenido horizontalmente en una sola fila.
* El VBox clase organiza sus nodos de contenido verticalmente en una sola columna.
* El StackPane clase coloca sus nodos de contenido en una sola pila de regreso a la delantera.
* El GridPane clase permite al desarrollador crear una cuadrícula flexible de las filas y columnas en la que se diseña nodos de contenido.
* El FlowPane clase organiza sus nodos de contenido, ya sea horizontal o vertical "flujo", envolviendo a la anchura especificada (para horizontal) o altura (vertical) de las fronteras.
* El TilePane clase coloca sus nodos de contenido en las celdas de diseño de tamaño uniforme o azulejos
* El AnchorPane clase permite a los desarrolladores crear nodos de anclaje a la parte superior, inferior, lateral izquierdo o en el centro de la disposición.

Para lograr una estructura de disposición deseada, diferentes contenedores se pueden anidar dentro de una aplicación JavaFX.

## 2-D y 3-D Transformaciones

Cada nodo en el grafo de la escena JavaFX se puede transformar en la coordenada xy usando los siguientes javafx.scene.tranform clases:

* translate - Mover un nodo de un lugar a otro a lo largo de las x, y, z planos relativos a su posición inicial.
* scale - Cambiar el tamaño de un nodo que aparezca más grande o más pequeño en las x, y, z aviones, dependiendo del factor de escala.
* shear - Girar un eje de manera que el eje x y el eje y ya no son perpendiculares. Las coordenadas del nodo se desplazan por los factores especificados.
* rotate - Girar un nodo sobre un punto de giro determinado de la escena.
* affine - Realizar una aplicación lineal de 2-D / 3-D coordina con otros 2-D / 3-D coordina preservando al mismo tiempo las propiedades "rectas" y "paralelo" de las líneas. Esta clase se debe utilizar con Translate , Scale , Rotate , o Shear transforman clases en lugar de ser utilizado directamente.

## Efectos visuales

El desarrollo de interfaces de cliente enriquecido en el escenario gráfico JavaFX implica el uso de efectos visuales o efectos para mejorar el aspecto de las aplicaciones JavaFX en tiempo real. Los efectos JavaFX son principalmente basado en imagen de píxeles y, por lo tanto, toman el conjunto de nodos que se encuentran en el escenario gráfico, hacen que sea como una imagen, y se aplican los efectos establecidos para ella.

Algunos de los efectos visuales disponibles en JavaFX incluyen el uso de las siguientes clases:

* Drop Shadow - Renders una sombra de un determinado contenido detrás del contenido al que se aplica el efecto.
* Reflection - Renders una versión reflejada del contenido a continuación el contenido real.
* Lighting - Simula una fuente de luz que brilla en un contenido dado y puede dar un objeto plano un aspecto más realista, tridimensional.