

JavaFX
JavaFX Architecture
Release 2.2.21 E229

02-05 April 2013

The JavaFX platform is designed to enable application developers to easily create and deploy rich client applications that behave consistently across platforms.

Arquitectura y Marco de JavaFX, Versión 2.2.21

E22902-05

Copyright © 2011 Oracle y/o sus afiliados. Todos los derechos reservados.

Autor Principal: Cindy Castillo

Este software y la documentación relacionada se proporcionan bajo un acuerdo de licencia que contiene restricciones sobre el uso y la divulgación, y están protegidos por las leyes de propiedad intelectual. Excepto según lo permitido expresamente en su acuerdo de licencia o permitido por la ley, no puede utilizar, copiar, reproducir, traducir, transmitir, modificar, otorgar licencias, distribuir, exhibir, realizar, publicar o mostrar ninguna parte, en ninguna forma o por ningún medio. La ingeniería inversa, desmontaje o descompilación de este software, a menos que lo exija la ley para la interoperabilidad, está prohibida.

La información contenida aquí está sujeta a cambios sin previo aviso y no se garantiza que esté libre de errores. Si encuentra algún error, por favor, infórmenos por escrito.

Si este es un software o documentación relacionada que se entrega al Gobierno de los Estados Unidos o a alguien que lo licencia en nombre del Gobierno de los Estados Unidos, es aplicable el siguiente aviso:

USUARIOS FINALES DEL GOBIERNO DE LOS ESTADOS UNIDOS: Los programas de Oracle, incluido cualquier sistema operativo, software integrado, programas instalados en el hardware y/o documentación, entregados a usuarios finales del Gobierno de los Estados Unidos son "software informático comercial" de conformidad con la normativa federal de adquisiciones aplicable y las regulaciones específicas de la agencia. Como tal, el uso, duplicación, divulgación, modificación y adaptación de los programas, incluido cualquier sistema operativo, software integrado, programas instalados en el hardware y/o documentación, estarán sujetos a los términos de licencia y restricciones de licencia aplicables a los programas. No se otorgan otros derechos al Gobierno de los Estados Unidos.

Este software o hardware está desarrollado para uso general en una variedad de aplicaciones de gestión de información. No está desarrollado ni destinado para su uso en aplicaciones inherentemente peligrosas, incluidas las aplicaciones que pueden crear un riesgo de lesiones personales. Si utiliza este software o hardware en aplicaciones peligrosas, usted será responsable de tomar todas las medidas apropiadas de seguridad, respaldo, redundancia y otras medidas para garantizar su uso seguro. Oracle Corporation y sus afiliados renuncian a cualquier responsabilidad por los daños causados por el uso de este software o hardware en aplicaciones peligrosas.

Oracle y Java son marcas registradas de Oracle y/o sus afiliados. Otros nombres pueden ser marcas registradas de sus respectivos propietarios.

Intel e Intel Xeon son marcas comerciales o marcas comerciales registradas de Intel Corporation. Todas las marcas comerciales SPARC se utilizan bajo licencia y son marcas comerciales o marcas comerciales registradas de SPARC International, Inc. AMD, Opteron, el logotipo de AMD y el logotipo de AMD Opteron son marcas comerciales o marcas comerciales registradas de Advanced Micro Devices. UNIX es una marca registrada de The Open Group.

Este software o hardware y la documentación pueden proporcionar acceso o información sobre contenido, productos y servicios de terceros. Oracle Corporation y sus afiliados no son responsables de y renuncian expresamente a todas las garantías de cualquier tipo con respecto al contenido, productos y servicios de terceros. Oracle Corporation y sus afiliados no serán responsables de ninguna pérdida, costo o daño incurrido debido a su acceso o uso del contenido, productos o servicios de terceros.

Contenidos

Arquitectura de JavaFX.....	5
Grafo de Escena	6
APIs Públicas de Java para Características de JavaFX	7
Sistema Gráfico	8
Kit de Ventanas Glass	8
Hilos	9
Pulso	10
Medios e Imágenes	10
Componente Web	10
CSS	11
Controles de IU	12
Diseño	13
Transformaciones 2D y 3D	14
Efectos Visuales	15

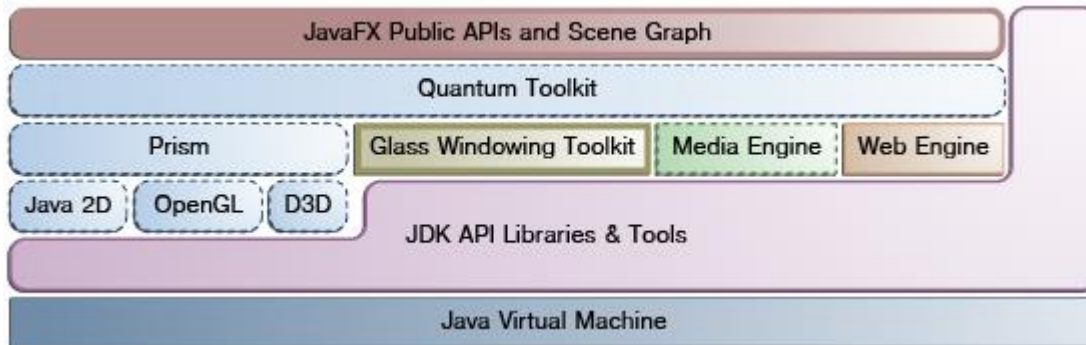
Arquitectura JavaFX

JavaFX es un conjunto de bibliotecas Java diseñadas para permitir a los desarrolladores crear e implementar aplicaciones cliente enriquecidas que se comportan de manera consistente en todas las plataformas. Consulte el documento [¿Qué es JavaFX 2?](#) para obtener un resumen de lo que JavaFX tiene para ofrecer.

[La Figura 1](#) ilustra los componentes arquitectónicos de la plataforma JavaFX. Las secciones que siguen al diagrama describen cada componente y cómo se interconectan las piezas. Debajo de las API públicas de JavaFX se encuentra el motor que ejecuta su código JavaFX. Está compuesto por subcomponentes que incluyen el nuevo motor gráfico de alto rendimiento JavaFX, llamado Prism; el nuevo pequeño y eficiente sistema de ventanas, denominado Glass; un motor de medios y un motor web. Aunque estos componentes no se exponen públicamente, sus descripciones pueden ayudarle a comprender mejor qué ejecuta una aplicación JavaFX.

- ["Gráfico de escena"](#)
- ["API públicas de Java para funciones JavaFX"](#)
- ["Sistema de gráficos"](#)
- ["Kit de herramientas para ventanas de vidrio"](#)
- ["Medios e imágenes"](#)
- ["Componente web"](#)
- ["CSS"](#)
- ["Controles de interfaz de usuario"](#)
- ["Disposición"](#)
- ["Transformaciones 2D y 3D"](#)
- ["Efectos visuales"](#)

Figura 1 Diagrama de arquitectura JavaFX



Descripción de "Figura 1 Diagrama de arquitectura JavaFX"

Gráfico de escena

El gráfico de escena JavaFX, que se muestra como parte de la capa superior en [la Figura 1](#), es el punto de partida para construir una aplicación JavaFX. Es un árbol jerárquico de nodos que representa todos los elementos visuales de la interfaz de usuario de la aplicación. Puede manejar entradas y puede renderizarse.

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

- Efectos, como desenfocos y sombras.
- Opacidad
- Transforma
- Controladores de eventos (como mouse, tecla y método de entrada)
- Un estado específico de la aplicación

A diferencia de Swing y Abstract Window Toolkit (AWT), el gráfico de escena JavaFX también incluye gráficos primitivos, como rectángulos y texto, además de tener controles, contenedores de diseño, imágenes y medios.

Para la mayoría de los usos, el gráfico de escena simplifica el trabajo con las IU, especialmente cuando se utilizan IU enriquecidas. La animación de varios gráficos en el gráfico de escena se puede lograr rápidamente utilizando las API `javafx.animation`, y los métodos declarativos, como el documento XML, también funcionan bien.

La `javafx.sceneAPI` permite la creación y especificación de varios tipos de contenido, tales como:

- **Nodos** : formas (2D y 3D), imágenes, medios, navegador web integrado, texto, controles de interfaz de usuario, gráficos, grupos y contenedores.
- **Estado** : transformaciones (posicionamiento y orientación de nodos), efectos visuales y otros estados visuales del contenido.
- **Efectos** : objetos simples que cambian la apariencia de los nodos del gráfico de escena, como desenfoques, sombras y ajuste de color.

Para obtener más información, consulte el documento [Trabajar con el gráfico de escena JavaFX](#).

API públicas de Java para funciones JavaFX

La capa superior de la arquitectura JavaFX que se muestra en [la Figura 1](#) proporciona un conjunto completo de API públicas que admiten el desarrollo de aplicaciones cliente enriquecidas. Estas API brindan libertad y flexibilidad incomparables para crear aplicaciones de cliente enriquecidas. La plataforma JavaFX combina las mejores capacidades de la plataforma Java con una funcionalidad multimedia integral e inmersiva en un entorno de desarrollo integral e intuitivo. Estas nuevas API de Java para funciones JavaFX:

- Permitir el uso de potentes funciones de Java, como genéricos, anotaciones y subprocesos múltiples.
- Facilite a los desarrolladores web el uso de JavaFX desde otros lenguajes dinámicos basados en JVM, como Groovy y JavaScript.
- Permita que los desarrolladores de Java utilicen otros lenguajes del sistema, como Groovy, para escribir aplicaciones JavaFX grandes o complejas.
- Permitir el uso de vinculación que incluye soporte para la vinculación diferida de alto rendimiento, expresiones de vinculación, expresiones de secuencia vinculada y reevaluación de vinculación parcial. Los lenguajes alternativos (como Groovy) pueden usar esta biblioteca de enlace para introducir una sintaxis de enlace similar a la de JavaFX Script.
- Amplíe la biblioteca de colecciones de Java para incluir listas y mapas observables, que permiten a las aplicaciones conectar interfaces de usuario a modelos de datos, observar cambios en esos modelos de datos y actualizar el control de UI correspondiente en consecuencia.

Las API de JavaFX y el modelo de programación son una continuación de la línea de productos JavaFX 1.x. La mayoría de las API de JavaFX se han portado directamente a Java. Algunas API, como Layout y Media, junto con muchos otros detalles, se han mejorado y simplificado en función de los comentarios recibidos de

los usuarios de la versión JavaFX 1.x. JavaFX se basa más en estándares web, como CSS para controles de estilo y ARIA para especificaciones de accesibilidad. También se está revisando el uso de estándares web adicionales.

Sistema de gráficos

El sistema de gráficos JavaFX, que se muestra en azul en [la Figura 1](#), es un detalle de implementación debajo de la capa de gráfico de escena JavaFX. Admite gráficos de escenas 2D y 3D. Proporciona renderizado de software cuando el hardware de gráficos de un sistema no es suficiente para soportar el renderizado acelerado por hardware.

Se implementan dos canales de aceleración de gráficos en la plataforma JavaFX:

- **Los procesos de prisma** generan trabajos. Puede ejecutarse tanto en renderizadores de hardware como de software, incluido 3-D. Es responsable de la rasterización y renderizado de escenas JavaFX. Las siguientes rutas de renderizado múltiples son posibles según el dispositivo que se utilice:
 - DirectX 9 en Windows XP y Windows Vista
 - DirectX 11 en Windows 7
 - OpenGL en Mac, Linux, integrado
 - Java2D cuando la aceleración por hardware no es posible

La ruta totalmente acelerada por hardware se utiliza cuando es posible, pero cuando no está disponible, se utiliza la ruta de procesamiento Java2D porque la ruta de procesamiento Java2D ya está distribuida en todos los entornos de ejecución de Java (JRE). Esto es particularmente importante cuando se manejan escenas 3D. Sin embargo, el rendimiento es mejor cuando se utilizan las rutas de procesamiento del hardware.

- **Quantum Toolkit** une Prism y Glass Windowing Toolkit y los pone a disposición de la capa JavaFX que se encuentra encima de ellos en la pila. También gestiona las reglas de subprocesamiento relacionadas con la representación versus el manejo de eventos.

Kit de herramientas para ventanas de vidrio

El Glass Windowing Toolkit, que se muestra en color beige en la parte central de [la Figura 1](#), es el nivel más bajo en la pila de gráficos JavaFX. Su principal responsabilidad es proporcionar servicios operativos nativos, como la gestión de

ventanas, temporizadores y superficies. Sirve como capa dependiente de la plataforma que conecta la plataforma JavaFX con el sistema operativo nativo.

El kit de herramientas Glass también es responsable de gestionar la cola de eventos. A diferencia del Abstract Window Toolkit (AWT), que administra su propia cola de eventos, el kit de herramientas Glass utiliza la funcionalidad de la cola de eventos del sistema operativo nativo para programar el uso de subprocesos. Además, a diferencia de AWT, el kit de herramientas Glass se ejecuta en el mismo hilo que la aplicación JavaFX. En AWT, la mitad nativa de AWT se ejecuta en un subproceso y el nivel de Java se ejecuta en otro subproceso. Esto introduce muchos problemas, muchos de los cuales se resuelven en JavaFX utilizando el enfoque de subproceso único de la aplicación JavaFX.

Hilos

El sistema ejecuta dos o más de los siguientes subprocesos en un momento dado.

- **Hilo de aplicación JavaFX** : este es el hilo principal utilizado por los desarrolladores de aplicaciones JavaFX. Desde este hilo se debe acceder a cualquier escena “en vivo”, que es una escena que forma parte de una ventana. Un gráfico de escena se puede crear y manipular en un subproceso en segundo plano, pero cuando su nodo raíz está adjunto a cualquier objeto vivo en la escena, se debe acceder a ese gráfico de escena desde el subproceso de la aplicación JavaFX. Esto permite a los desarrolladores crear gráficos de escenas complejos en un hilo de fondo mientras mantiene las animaciones en escenas "en vivo" fluidas y rápidas. El subproceso de la aplicación JavaFX es un subproceso diferente del subproceso de envío de eventos (EDT) de Swing y AWT, por lo que se debe tener cuidado al incrustar código JavaFX en las aplicaciones Swing.
- **Hilo de renderizado de Prism** : este hilo maneja el renderizado por separado del despachador de eventos. Permite renderizar el fotograma N mientras se procesa el fotograma N +1. Esta capacidad de realizar procesamiento simultáneo es una gran ventaja, especialmente en sistemas modernos que tienen múltiples procesadores. El subproceso de renderizado de Prism también puede tener múltiples subprocesos de rasterización que ayudan a descargar el trabajo que debe realizarse en el renderizado.
- **Hilo multimedia** : este hilo se ejecuta en segundo plano y sincroniza los fotogramas más recientes a través del gráfico de escena mediante el hilo de la aplicación JavaFX.

Legumbres

Un pulso es un evento que indica al gráfico de escena JavaFX que es hora de sincronizar el estado de los elementos en el gráfico de escena con Prism. Un pulso se acelera a un máximo de 60 fotogramas por segundo (fps) y se activa cada vez que se ejecutan animaciones en el gráfico de escena. Incluso cuando la animación no se está ejecutando, se programa un pulso cuando se cambia algo en el gráfico de la escena. Por ejemplo, si se cambia la posición de un botón, se programa un pulso.

Cuando se activa un pulso, el estado de los elementos en el gráfico de escena se sincroniza con la capa de renderizado. Un pulso permite a los desarrolladores de aplicaciones manejar eventos de forma asíncrona. Esta importante característica permite al sistema procesar por lotes y ejecutar eventos en el pulso.

El diseño y el CSS también están vinculados a los eventos de pulso. Numerosos cambios en el gráfico de escena podrían dar lugar a múltiples actualizaciones de diseño o CSS, lo que podría degradar seriamente el rendimiento. El sistema realiza automáticamente un pase de CSS y diseño una vez por pulso para evitar la degradación del rendimiento. Los desarrolladores de aplicaciones también pueden activar manualmente pases de diseño según sea necesario para tomar medidas antes de un pulso.

El Glass Windowing Toolkit es responsable de ejecutar los eventos de pulso. Utiliza temporizadores nativos de alta resolución para realizar la ejecución.

Medios e imágenes

La funcionalidad multimedia JavaFX está disponible a través de las `javafx.scene.mediaAPI`. JavaFX admite medios visuales y de audio. Se proporciona soporte para archivos de audio MP3, AIFF y WAV y archivos de video FLV. La funcionalidad multimedia de JavaFX se proporciona como tres componentes separados: el objeto `Media` representa un archivo multimedia, `MediaPlayer` reproduce un archivo multimedia y `MediaView` es un nodo que muestra los medios.

El componente Media Engine, que se muestra en verde en [la Figura 1](#), se diseñó teniendo en cuenta el rendimiento y la estabilidad y proporciona un comportamiento consistente en todas las plataformas. Para obtener más información, lea el documento [Incorporación de activos multimedia en aplicaciones JavaFX](#).

Componente web

El componente web es un nuevo control de interfaz de usuario de JavaFX, basado en Webkit, que proporciona un visor web y una funcionalidad de navegación completa a través de su API. Este componente de Web Engine, que se muestra en [la Figura 1](#), se basa en WebKit, que es un motor de navegador web de

código abierto que admite HTML5, CSS, JavaScript, DOM y SVG. Permite a los desarrolladores implementar las siguientes funciones en sus aplicaciones Java:

- Renderizar contenido HTML desde URL local o remota
- Admite historial y proporciona navegación hacia adelante y hacia atrás
- recargar el contenido
- Aplicar efectos al componente web.
- Editar el contenido HTML
- Ejecutar comandos de JavaScript
- Manejar eventos

Este componente de navegador integrado se compone de las siguientes clases:

- `WebEngine` proporciona capacidad básica de navegación de páginas web.
- `WebView` encapsula un objeto `WebEngine`, incorpora contenido HTML en la escena de una aplicación y proporciona campos y métodos para aplicar efectos y transformaciones. Es una extensión de una `Node`.

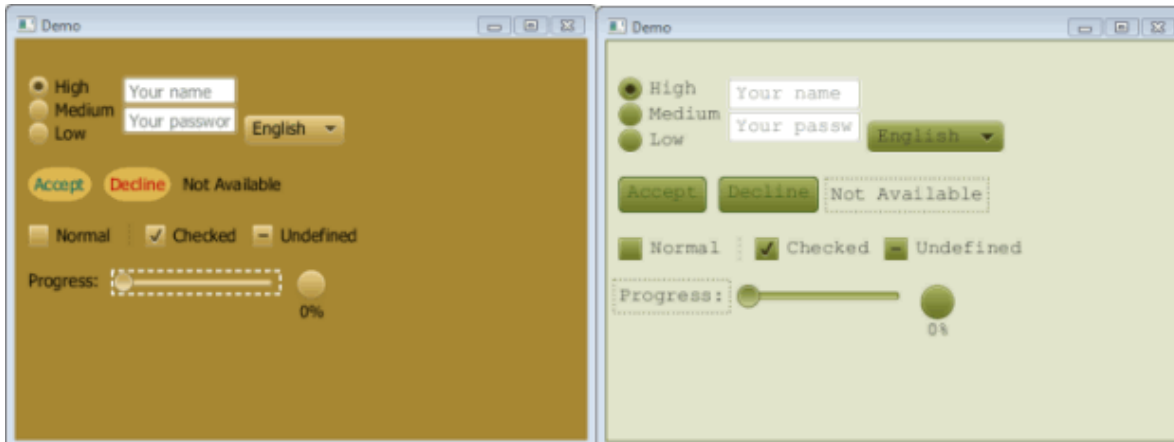
Además, las llamadas Java se pueden controlar a través de JavaScript y viceversa para permitir a los desarrolladores aprovechar al máximo ambos entornos. Para obtener una descripción general más detallada del navegador integrado JavaFX, consulte el documento [Agregar contenido HTML a aplicaciones JavaFX](#).

CSS

Las hojas de estilo en cascada (CSS) de JavaFX brindan la capacidad de aplicar estilos personalizados a la interfaz de usuario de una aplicación JavaFX sin cambiar el código fuente de esa aplicación. CSS se puede aplicar a cualquier nodo en el gráfico de escena JavaFX y se aplica a los nodos de forma asincrónica. Los estilos CSS de JavaFX también se pueden asignar fácilmente a la escena en tiempo de ejecución, lo que permite que la apariencia de una aplicación cambie dinámicamente.

[La Figura 2](#) demuestra la aplicación de dos estilos CSS diferentes al mismo conjunto de controles de UI.

Figura 2 Muestra de hoja de estilo CSS



Descripción de "Figura 2 Muestra de hoja de estilo CSS"

JavaFX CSS se basa en las especificaciones W3C CSS versión 2.1, con algunas adiciones del trabajo actual en la versión 3. El soporte y las extensiones de JavaFX CSS se han diseñado para permitir que las hojas de estilo JavaFX CSS sean analizadas limpiamente por cualquier analizador CSS compatible, incluso uno que no admite extensiones JavaFX. Esto permite mezclar estilos CSS para JavaFX y para otros fines (como páginas HTML) en una única hoja de estilo. Todos los nombres de propiedades de JavaFX tienen el prefijo "-fx-", incluidos aquellos que pueden parecer compatibles con HTML CSS estándar, porque algunos valores de JavaFX tienen una semántica ligeramente diferente.

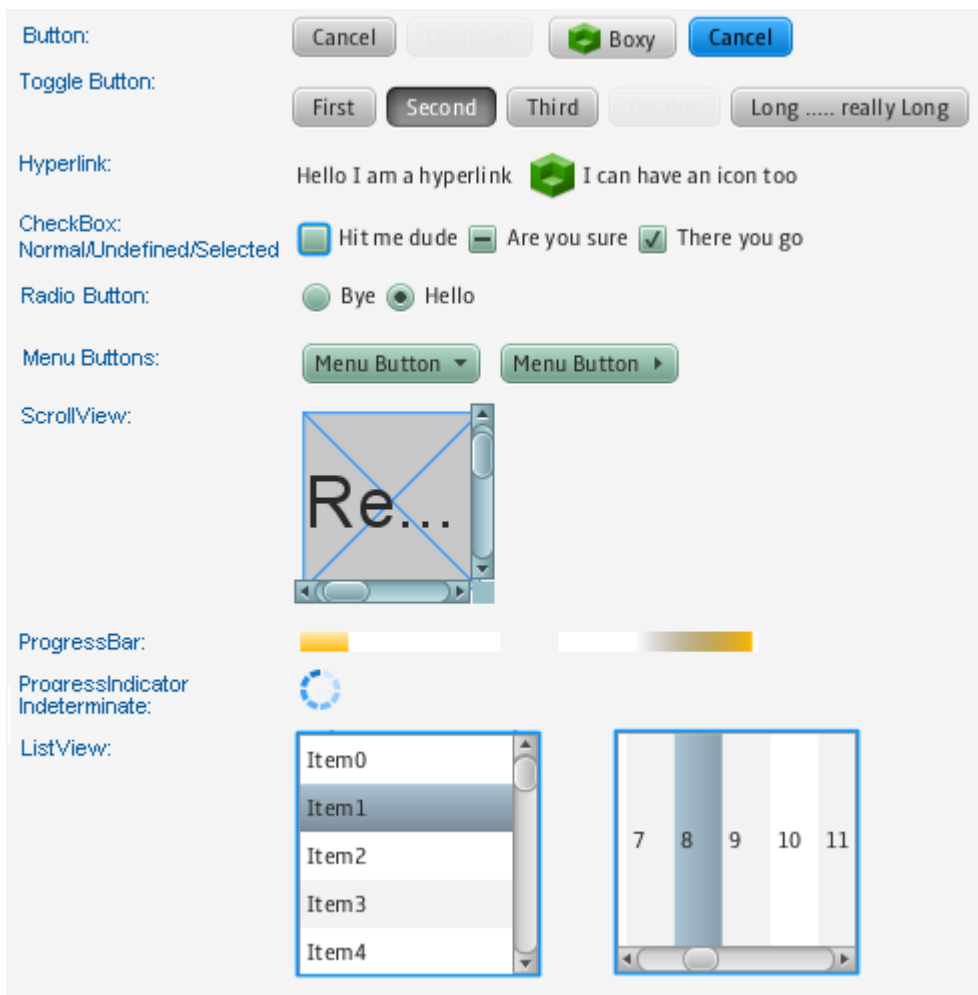
Para obtener información más detallada sobre JavaFX CSS, consulte el artículo [Diseño de aplicaciones JavaFX con CSS](#).

Controles de interfaz de usuario

Los controles de la interfaz de usuario de JavaFX disponibles a través de la API de JavaFX se crean utilizando nodos en el gráfico de escena. Pueden aprovechar al máximo las funciones visualmente ricas de la plataforma JavaFX y son portátiles a través de diferentes plataformas. JavaFX CSS permite tematizar y personalizar los controles de la interfaz de usuario.

[La Figura 3](#) muestra algunos de los controles de la interfaz de usuario que se admiten actualmente. Se han introducido nuevos controles de la interfaz de usuario de Java, como TitlePane o Accordion, con el SDK de JavaFX. Estos nuevos controles residen en el `javafx.scene.control` paquete.

Figura 3 Ejemplo de controles de interfaz de usuario de JavaFX



Descripción de "Figura 3 Ejemplo de controles de interfaz de usuario de JavaFX"

Para obtener información más detallada sobre todos los controles de UI de JavaFX disponibles, consulte el documento [Uso de controles de UI de JavaFX](#) y la [documentación de API](#) para el `javafx.scene.control` paquete.

Disposición

Se pueden utilizar contenedores o paneles de diseño para permitir disposiciones flexibles y dinámicas de los controles de la interfaz de usuario dentro de un gráfico de escena de una aplicación JavaFX. La API de diseño JavaFX incluye las siguientes clases de contenedor que automatizan modelos de diseño comunes:

- La `BorderPane` clase dispone sus nodos de contenido en la región superior, inferior, derecha, izquierda o central.
- La `HBox` clase organiza sus nodos de contenido horizontalmente en una sola fila.

- La VBoxclase organiza sus nodos de contenido verticalmente en una sola columna.
- La StackPaneclase coloca sus nodos de contenido en una única pila de atrás hacia adelante.
- La GridPaneclase permite al desarrollador crear una cuadrícula flexible de filas y columnas en las que diseñar nodos de contenido.
- La FlowPaneclase organiza sus nodos de contenido en un “flujo” horizontal o vertical, envolviéndose en los límites de ancho (para horizontal) o alto (para vertical) especificados.
- La TilePaneclase coloca sus nodos de contenido en celdas o mosaicos de diseño de tamaño uniforme.
- La AnchorPaneclase permite a los desarrolladores crear nodos de anclaje en la parte superior, inferior, izquierda o central del diseño.

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

Para obtener más información sobre cómo trabajar con diseños, consulte el artículo [Trabajar con diseños en JavaFX](#) . Para obtener más información sobre la API de diseño JavaFX, consulte la documentación de la API del `javafx.scene.layout` paquete.

Transformaciones 2D y 3D

Cada nodo en el gráfico de escena JavaFX se puede transformar en la coordenada xy usando las siguientes `javafx.scene.transform` clases:

- `translate`– Mover un nodo de un lugar a otro a lo largo de los planos x, y, z con respecto a su posición inicial.
- `scale`– Cambiar el tamaño de un nodo para que parezca más grande o más pequeño en los planos x, y, z, según el factor de escala.
- `shear`– Gire un eje para que los ejes x e y ya no sean perpendiculares. Las coordenadas del nodo se desplazan según los multiplicadores especificados.
- `rotate`– Girar un nodo alrededor de un punto de pivote específico de la escena.
- `affine`– Realice un mapeo lineal de coordenadas 2-D/3-D a otras coordenadas 2-D/3-D conservando las propiedades 'rectas' y 'paralelas' de las líneas. Esta clase debe usarse con las

clases Translate, Scale, Rotateo Sheartransform en lugar de usarse directamente.

Para obtener más información sobre cómo trabajar con transformaciones, consulte el documento [Aplicar transformaciones en JavaFX](#) . Para obtener más información sobre las javafx.scene.transformclases de API, consulte la [documentación de API](#) .

Efectos visuales

El desarrollo de interfaces de cliente enriquecidas en el gráfico de escena JavaFX implica el uso de efectos visuales o efectos para mejorar la apariencia de las aplicaciones JavaFX en tiempo real. Los efectos JavaFX se basan principalmente en píxeles de imagen y, por lo tanto, toman el conjunto de nodos que están en el gráfico de escena, lo representan como una imagen y le aplican los efectos especificados.

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

- Drop Shadow– Representa una sombra de un contenido determinado detrás del contenido al que se aplica el efecto.
- Reflection– Representa una versión reflejada del contenido debajo del contenido real.
- Lighting– Simula una fuente de luz que brilla sobre un contenido determinado y puede darle a un objeto plano una apariencia tridimensional más realista.

Para obtener ejemplos sobre cómo utilizar algunos de los efectos visuales disponibles, consulte el documento [Creación de efectos visuales](#) . Para obtener más información sobre todas las clases de efectos visuales disponibles, consulte la [documentación API](#) del javafx.scene.effectpaquete.

Bibliografía:

(Oracle, s.f.)

Oracle. (s.f.). JavaFX Architecture and Framework.

<https://docs.oracle.com/javafx/2/architecture/jfxpub-architecture.htm>