



3. Componentes gráficos

3.1. Stage, Scene y Nodos

En una aplicación JavaFX con una IGU se diferencian tres grandes componentes:

- **Stage (ventana)**: es la ventana principal de la aplicación y viene determinada por el sistema operativo donde estemos ejecutando el programa. Una ventana del mismo programa se verá de forma diferente en cada S. O. sin que el programador pueda modificarlo.
- **Scene (escena)**: es lo que se muestra dentro de la aplicación. La escena describe todo lo que hay dentro de una ventana en una aplicación JavaFX. Está definida completamente por el programador, que puede diseñarla y cambiarla a su gusto.
- **Nodos**: son todos aquellos componentes gráficos que conforman la escena. Estos nodos se almacenan en el fichero FXML en forma de árbol. Puedes repasar este concepto en el módulo de Lenguajes de marcas.

Por tanto, acabamos de ver que un stage corresponde al contenedor de nivel superior y consta, como mínimo, de una escena, que a su vez es contenedora de otros componentes gráficos.

En la siguiente imagen puedes observar las diferentes partes de una ventana con JavaFX:

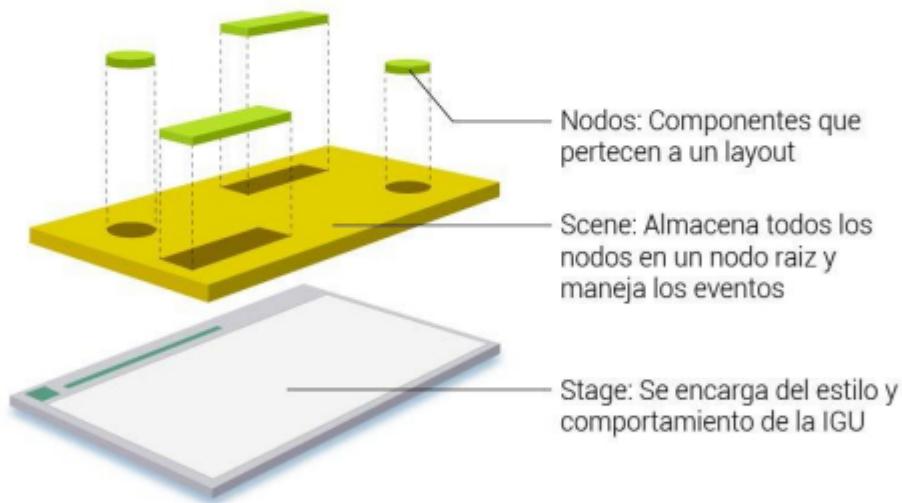


Fig. 13.7 Componentes de una ventana JavaFX

Si nuestro programa es una aplicación de escritorio, el stage será la ventana, con su barra de título y sus botones de maximizar, minimizar o cerrar. En el caso de que se ejecutase en un navegador web, nuestro stage estará embebido dentro del navegador.

Los nodos son elementos individuales que forman una escena, como, por ejemplo, botones, cuadros de texto, layouts, etc. En Java FX, los nodos pueden tener nodos hijos, que, a su vez, tienen otros nodos hijos, como si de las ramas de un árbol se tratase.



El conjunto de todos los nodos que forman una escena es lo que llamamos Scene Graph o **grafo de la escena**.

Existe un nodo especial llamado root node –o nodo raíz/padre–, del cual se extienden todos los demás nodos. Este es el único que no tiene un nodo padre, pues el resto de nodos sí lo tienen y son así descendientes –ya sea directa o indirectamente– del nodo raíz.

3.2. Contenedores y Layouts

Después de construir todos los nodos necesarios en una escena, generalmente los organizaremos en una disposición adecuada al tipo de aplicación que estemos diseñando. Crearemos contenedores dentro del contenedor principal y le indicaremos la disposición de los componentes dentro del contenedor (**disposición del contenedor o layout**). También podemos decir que seguimos un diseño, ya que incluye colocar todos los componentes en una posición particular dentro del contenedor.

JavaFX proporciona varios diseños predefinidos, como HBox, VBox, Border Pane, Stack Pane, Text Flow, Anchor Pane, Title Pane, Grid Pane, Flow Panel, etc.

Cada uno de los diseños mencionados anteriormente está representado por una clase, y todas estas clases pertenecen al paquete `javafx.layout`. La clase llamada *Pane* es la clase base de todos los diseños en JavaFX.

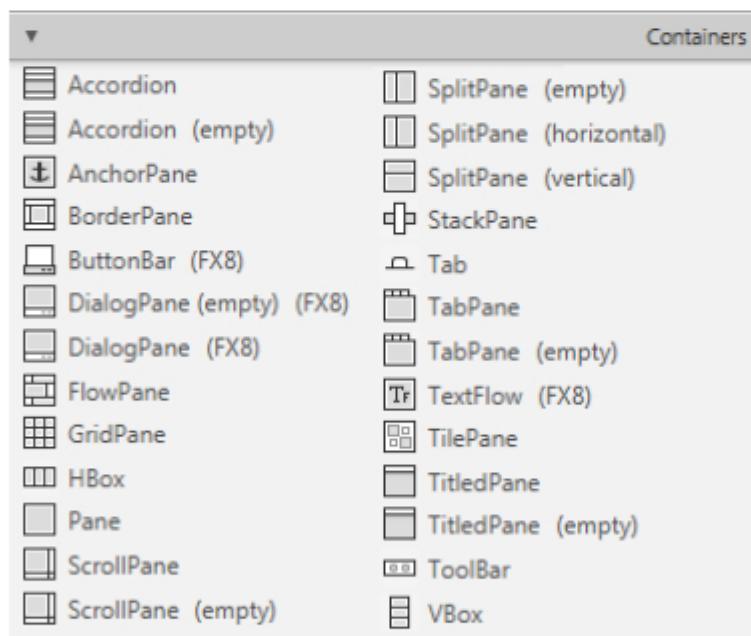


Fig. 13.8 Contenedores JavaFX que nos ofrece SceneBuilder



Los paneles más utilizados de JavaFX son:

Panel	Descripción
VBox	Organiza de una forma sencilla una serie de nodos en una sola columna.
HBox	Organiza de una forma sencilla una serie de nodos en una sola fila.
TilePane	Un panel de mosaico es similar a un panel de flujo. El panel de diseño de TilePane coloca todos los nodos en una cuadrícula en la que cada celda o mosaico tiene el mismo tamaño. Los nodos se pueden colocar horizontalmente (en filas) o verticalmente (en columnas).
GridPane	Permite crear una cuadrícula flexible de filas y columnas en la que distribuir los nodos. Los nodos se pueden colocar en cualquier celda de la cuadrícula y pueden abarcar celdas según sea necesario. Esto es útil para crear formularios o para elaborar cualquier diseño que esté organizado en filas y columnas.
BorderPane	Proporciona cinco regiones en las que colocar los nodos: superior, inferior, izquierda, derecha y central. Las regiones pueden ser de cualquier tamaño. Si su aplicación no necesita una de las regiones, no es necesario que la defina, y de este modo no se le asignará espacio.
FlowPane	Los nodos se distribuyen consecutivamente y se ajustan al límite establecido para el panel. Los nodos pueden fluir verticalmente (en columnas) u horizontalmente (en filas).
StackPane	Organiza todos los nodos dentro de una sola pila con cada nodo nuevo agregado encima del nodo anterior. Este modelo de diseño proporciona una manera fácil de superponer texto en una forma o imagen, o de superponer formas comunes para crear una forma compleja.
AnchorPane	Permite anclar nodos en la parte superior, inferior, izquierda, derecha o centro del panel. A medida que se cambia el tamaño de la ventana, los nodos mantienen su posición en relación con su punto de anclaje.

Tabla. 13.1 . Paneles JavaFX

A continuación, se muestra una imagen para que puedas entender mejor los conceptos explicados en la tabla anterior:

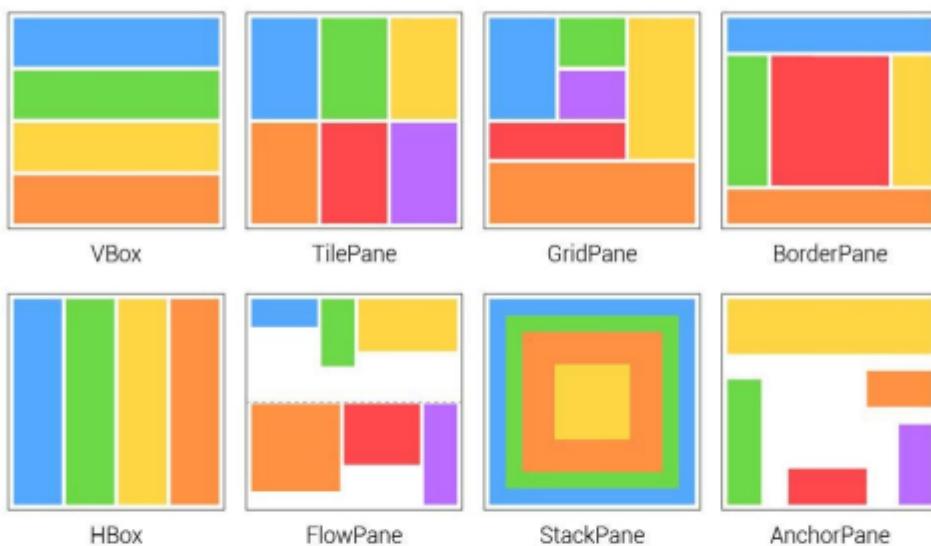


Fig. 13.9 Estructura gráfica de los layouts de JavaFX

3.3. Componentes

	Controls
<input checked="" type="checkbox"/> Button	<input checked="" type="radio"/> RadioButton
<input checked="" type="checkbox"/> CheckBox	<input type="checkbox"/> ScrollBar (horizontal)
<input checked="" type="checkbox"/> ChoiceBox	<input checked="" type="checkbox"/> ScrollBar (vertical)
<input checked="" type="checkbox"/> ColorPicker	<input type="checkbox"/> Separator (horizontal)
<input checked="" type="checkbox"/> ComboBox	<input type="checkbox"/> Separator (vertical)
<input checked="" type="checkbox"/> DatePicker (FX8)	<input checked="" type="checkbox"/> Slider (horizontal)
<input checked="" type="checkbox"/> HTMLEditor	<input checked="" type="checkbox"/> Slider (vertical)
<input checked="" type="checkbox"/> Hyperlink	<input checked="" type="checkbox"/> Spinner (FX8)
<input checked="" type="checkbox"/> ImageView	<input checked="" type="checkbox"/> SplitMenuItem
<input checked="" type="checkbox"/> Label	<input type="checkbox"/> TableColumn
<input checked="" type="checkbox"/> ListView	<input type="checkbox"/> TableView
<input checked="" type="checkbox"/> MediaView	<input type="checkbox"/> TextArea
<input checked="" type="checkbox"/> MenuBar	<input type="checkbox"/> TextField
<input checked="" type="checkbox"/> MenuButton	<input type="checkbox"/> ToggleButton
<input checked="" type="checkbox"/> Pagination	<input type="checkbox"/> TreeTableColumn (FX8)
<input checked="" type="checkbox"/> PasswordField	<input type="checkbox"/> TreeTableView (FX8)
<input checked="" type="checkbox"/> ProgressBar	<input type="checkbox"/> TreeView
<input checked="" type="checkbox"/> ProgressIndicator	<input type="checkbox"/> WebView

Fig. 13.10 Componentes gráficos JavaFX que nos ofrece SceneBuilder



Los componentes más utilizados de JavaFX son:

Componente	Descripción
Button	Permite que una aplicación JavaFX ejecute alguna acción cuando el usuario de la aplicación hace clic en el botón.
CheckBox	Es un componente gráfico que puede estar en estado activado (verdadero) o desactivado (falso).
ComboBox	Permite a los usuarios elegir una opción de una lista predefinida de opciones, o escribir otro valor si ninguna de las opciones predefinidas coincide con lo que el usuario desea seleccionar.
ImageView	Permite mostrar una imagen dentro de una IGU de JavaFX.
Label	Muestra una etiqueta de texto dentro de una IGU de JavaFX.
RadioButton	Un RadioButton de JavaFX es un botón que se puede seleccionar o no. Pero no se puede «deseleccionar» una vez seleccionado. Si pertenece a un grupo de RadioButtons, cuando se marque otro, el primero se «deseleccionará» automáticamente.
TextArea	Permite a los usuarios de una aplicación JavaFX introducir texto que abarque varias líneas, y que luego la aplicación puede leer.
TextField	Permite a los usuarios de una aplicación JavaFX introducir texto que luego puede ser leído por la aplicación.

Tabla. 13.2. Componentes JavaFX

En resumen:

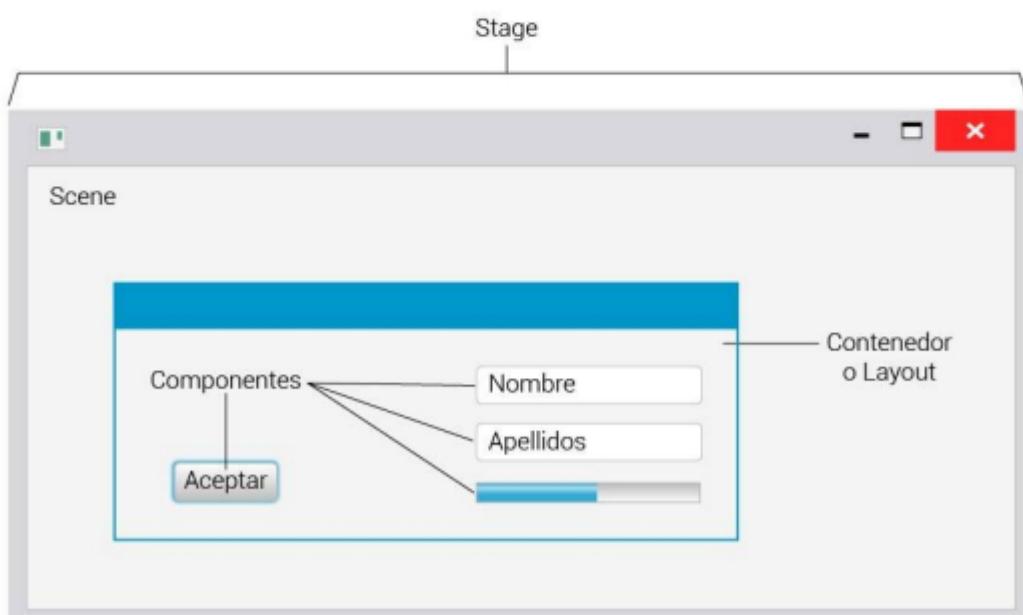


Fig. 13.11 Componentes de una ventana JavaFX



3.4. Menús

JavaFx nos ofrece una serie de componentes para crear una barra de menús con todas las opciones que necesite nuestro programa. Hemos de tener en cuenta cómo incluir los distintos elementos.

Un control **MenuBar** se coloca, tradicionalmente, en la parte superior de la interfaz de usuario, y dentro de él se encuentran los **menús**. De forma predeterminada, para cada menú agregado a la barra de menú, se representará como un botón con el valor del texto del menú mostrado. Cada uno de los distintos menús de la barra contendrá distintos elementos, es decir, distintos **menuitems**.

MenuBar > Menu > MenuItem

Podemos utilizar las siguientes clases de la API de JavaFX para crear menús en nuestra aplicación JavaFX.

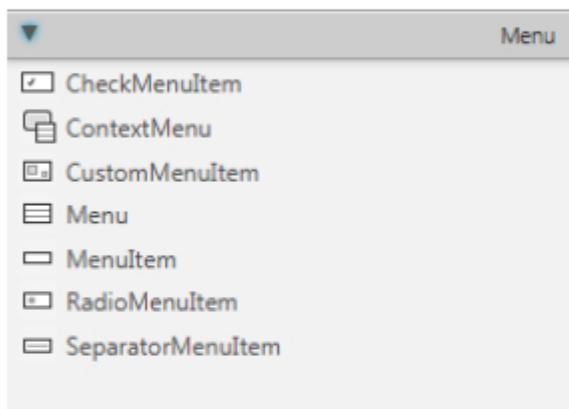


Fig. 13.12 Componentes para menús JavaFX que nos ofrece SceneBuilder

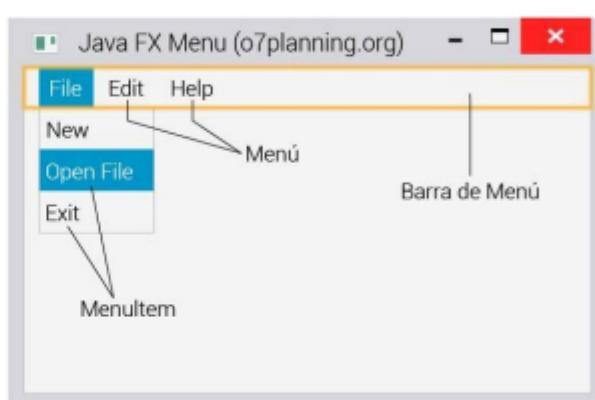


Fig. 13.13 Partes de un Menú JavaFX

