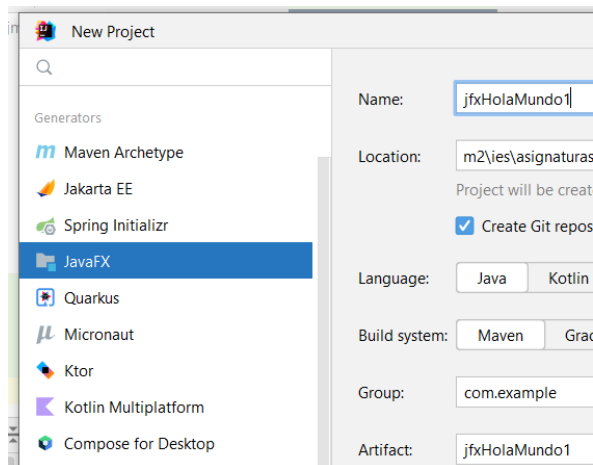


JavaFX

<https://openjfx.io/>

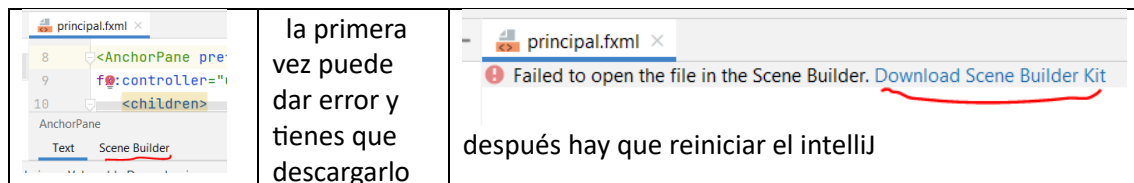
Usar javafx en intelliJ

<https://www.jetbrains.com/help/idea/javafx.html>



No añadas librerías adicionales y crea el proyecto

Los FXML, en IntelliJ se pueden crear con el menú de creación de ficheros, se colocan en la carpeta resources. En los FXML hay 2 partes: como colocar las cosas en pantalla y qué ejecutar como acciones (controllers) para los que se deben crear clases.java. El scene builder está integrado en IntelliJ y cuando abres un FXML tienes una pestaña, abajo, para abrirlo.



Si al crear el fxml tienes una etiqueta como `fx:controller="Controlador"` entonces debes crear la clase Controlador, normalmente en el paquete UI

Librería y videos para hacer juegos con javafx <https://www.youtube.com/@AlmasB0>

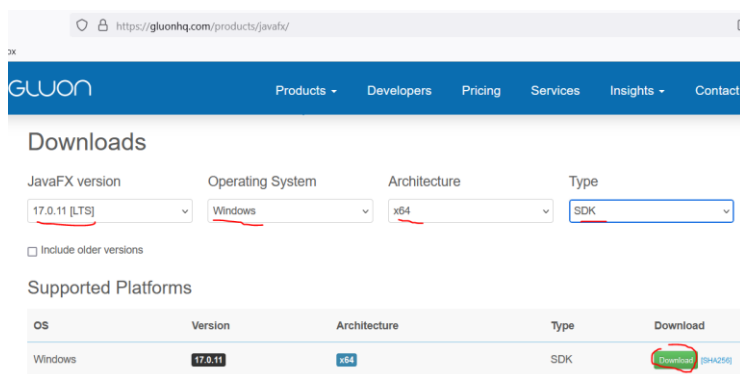
tutorial javafx https://www.tutorialspoint.com/javafx/javafx_application.htm pero implica hacerlo a mano con, stages escenas y elementos, pero lo habitual ahora es hacerlo con XML

Tutorial FXML <https://jenkov.com/tutorials/javafx/index.html> .

Lo normal es hacer el fxml con una herramienta como <https://gluonhq.com/products/scen-builder/> que hay que descargarse e instalar.

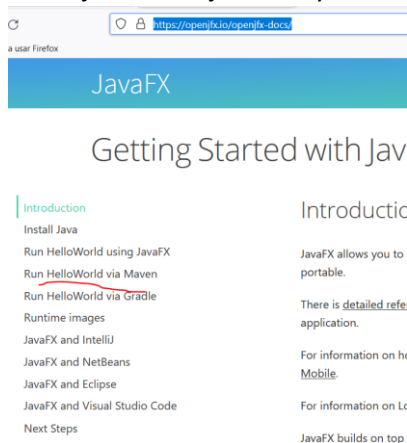
Desde la versión 9 de javafx no está en java

Desde el jdk 8 el jdk oficial de Oracle es de pago y surgieron otras versiones gratuitas (liberika, corretto,...). Aunque el intellij ya te deja ejecutar si lo falla puedes necesitar, para ejecutar javafx necesitas descargar una versión de jdk que lo incluya. Puedes elegir en <https://gluonhq.com/products/javafx/>



Descomprimir el zip descargado

Para ejecutar un javafx hay 3 formas <https://openjfx.io/openjfx-docs/>



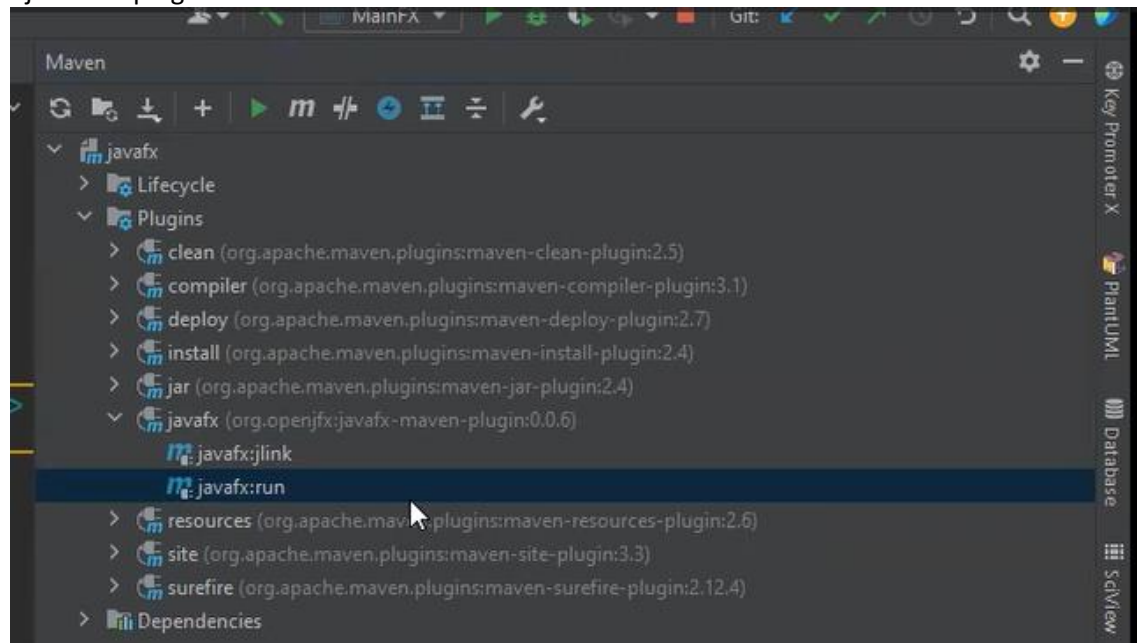
1-Hacer el **proyecto modular creando un fichero module-info.java** justo debajo del directorio java, pero tienen problemas porque puede haber librerías que no son modulares. En el module-info hay que añadir los módulos que necesitas, normalmente intellij te los pondrá automáticamente, y los exports para que sean visibles tus elementos desde otros módulos. Por eso, por ejem., se escribe exports ui; en el module-info, para hacer visible el paquete en el que está el main.



2-Si el proyecto no es modular y tengo Maven puedes añadir un plugin e indicar cuál es la clase principal en la etiqueta mainClass <https://openjfx.io/openjfx-docs/#maven>

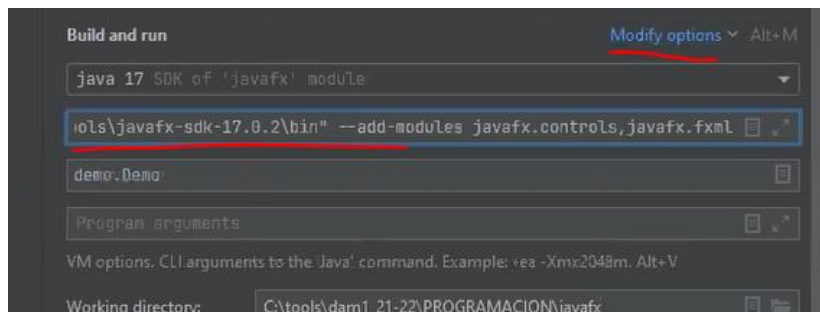
```
<build>
<plugins>
  <plugin>
    <groupId>org.openjfx</groupId>
    <artifactId>javafx-maven-plugin</artifactId>
    <version>0.0.8</version>
    <configuration>
      <mainClass>HelloFX</mainClass>
    </configuration>
  </plugin>
</plugins>
</build>
```

Ejecuta el plugin de Maven desde el intelliJ



3.- Añadir opciones de ejecución al intelliJ para que encuentre el jdk con javafx, colocando la línea

--module-path "c:\tools\javafx-sdk-17.9.2\bin" --add-modules javafx.controls,javafx.fxml
Puedes sustituir la ruta entre comillas por una variable de entorno que hayas creado en tu ordenador. Estas opciones modificadas están guardadas en el fichero workspace.xml que no se sube a github porque está en el gitignore



File ->Settings->Plugins

Instalo el plugin JavaFX runtime for plugins

Rearranca intelliJ

File- new Project – javafx

Marca bootstrapFX y controlFX y luego Create

Después acepta las opciones sobre Maven