Explicación del proyecto

# Presentación

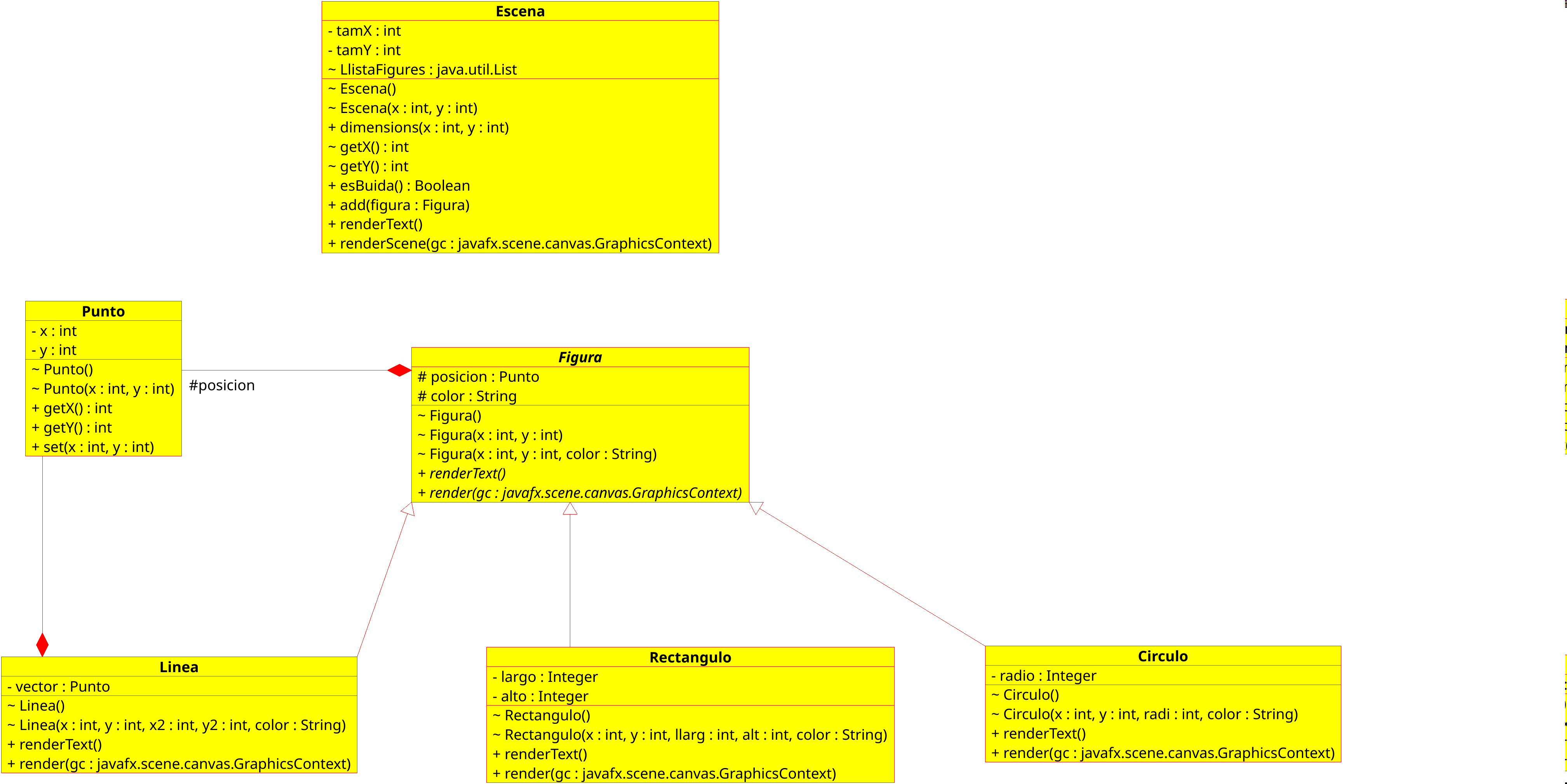
En este proyecto se va a crear un programa para la realización de dibujos, mediante la introducción de ciertas figuras. Este proyecto se pasa parcialmente iniciado, y el alumno deberá de ir completando aquellas partes que se requieran. Se facilita un proyecto creado con gradle, para la gestión de las dependencias teniendo ya, añadidas las librerías de javafx, para la representación gráficas de las mismas. La parte de la representación gráfica está totalmente implementada y se invita al alumno a su revisión y análisis.

# Programa Principal

El programa principal ya viene con código ejecutable. Al lanzarlo, aparece un prompt con la palabra #Figura: que admite las siguientes opciones:

* dimensiones ancho alto: Admite dos valores numéricos, para establecer las dimensiones del lienzo
* circulo x y radio #color: Añade un círculo a la escena, admitiendo tres parámetros, con la posición y el radio, y un color en formato #RRGGBB. (Siendo cada pareja de valores un valor hexadecimal con la cantidad de Rojo, Verde y Azul.
* rectangulo x1 y1 largo alto #color: Que añade una figura rectangulo, con el vértice superior izquierdo en (x1, y1)  las dimensiones del mismo en el color indicado.
* linea x1 y1 x2 y2 #color: Que genera una línea, de grosor 3 pixeles (por defecto), entre el punto (x1,y1) y el (x2,y2), del color indicado.
* draw: Que abre una ventana con JavaFX i dibuja la escena creada. Al cerrar la ventana cerrará la aplicación

# El diagrama de clases



### Estructura del proyecto

De los ficheros de la carpeta raíz, nos interesa el fichero build.gradle, que contiene la información sobre la construcción del proyecto. Este ya incluye las librerías para la gestión de los gráficos. En un futuro habrá que añadir alguna librería.

Lo que nos interesa, es todo aquello ubicado dentro del paquete figuras. Aquí tenemos:

• App.java: Se trata del punto de entrada de la aplicación, y se encarga de mostrar el prompt de la escena. En principio no hay que modificar nada en esta clase.

• Figura.java: La superclase de la cual heredaran el resto de figuras. Almacena la posición (con un objeto de tipo Punto) y el color de la figura. En esta clase no habrá que hacer grandes cambios, pero sí que tendrá que definir como métodos abstractos, los que vayamos añadiendo al resto de figuras. Además, habrá que importar algunas librerías, para la serialización y la manipulación de ficheros XML y JSON.

• Circulo.java: Especialización de figura que dibuja un círculo. Además de los atributos heredados de figura, incorpora un nuevo atributo que representa el radio. Tendrá que implementar algunos métodos adicionales para obtener una representación de la figura en diferentes formatos (texto, XML, JSON), e incorporar las librerías pertinentes.

• Linea.java: Especialización de figura que dibuja una línea. Además de los atributos heredados de figura, incorpora un nuevo atributo de la clase punto (llamado vector), que dice el punto final de la línea. Tendrá que implementar algunos métodos adicionales para obtener una representación de la figura en diferentes formatos (texto, XML, JSON), e incorporar las librerías pertinentes.

• Rectángulo.java Especialización de la figura que dibuja un rectángulo. Incorpora  nuevos atributo que representan el largo y ancho. Tendrá que implementar algunos métodos adicionales para obtener una representación de la figura en diferentes formatos (texto, XML, JSON), e incorporar las librerías pertinentes.

• Punto.java: Clase auxiliar. No hereda de nadie ni tiene que añadir ningún método nuevo.

• HexColorValidator.java: Clase auxiliar que verifica el formato RGB de los colores correcto. No se tiene que modificar.

• Escena.java: Representa el escenario. Tiene los atributos de las dimensiones del escenario y una lista de figuras, que tendremos que recorrer para guardarla en disco.

• FileManager.java: Clase que se encargará de la gestión del almacenamiento. Será la que tendremos que implementar, y por eso le dedicaremos un apartado.