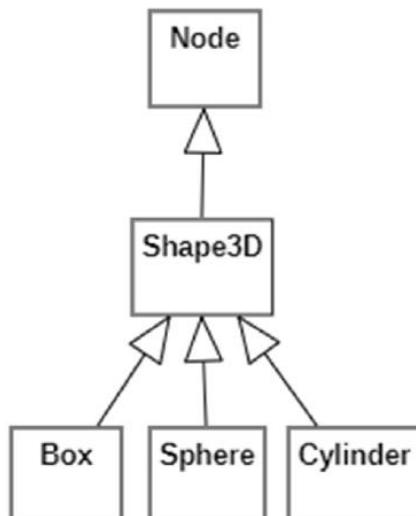


## Figuras 3D

Cualquier forma dibujada que tenga tres dimensiones (largo, ancho y profundidad) se conoce como forma 3D. Algunos ejemplos de figuras 3D son los cubos, las esferas y las pirámides.

La figura siguiente muestra un diagrama de clases que representa las formas 3D en JavaFX. Las clases de formas 3D están en el paquete `javafx.scene.shape`. Las clases `Box`, `Sphere` y `Cylinder` representan tres formas 3D predefinidas que posee JavaFX y son subclases de `Shape3D`.

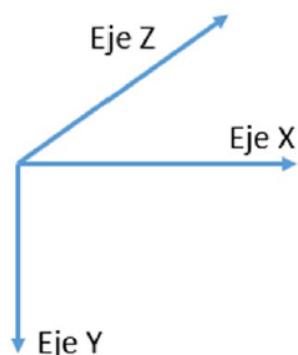


Jerarquía de clases de figuras 3D

En la tabla, se presentan los principales métodos de estas clases para crear figuras 3D.

Clase	Método	Descripción
<code>Shape3D</code>	<code>void setTranslateX(double valor)</code>	Define la coordenada <i>x</i> de la forma.
	<code>void setTranslateY(double valor)</code>	Define la coordenada <i>y</i> de la forma.
	<code>void setTranslateZ(double valor)</code>	Define la coordenada <i>z</i> de la forma.
<code>Box</code>	<code>Box(double largo, double ancho, double profundidad)</code>	Crea un cubo especificando su tamaño.
<code>Sphere</code>	<code>Sphere(double radio, int divisiones)</code>	Crea una esfera y su resolución.
<code>Cylinder</code>	<code>Cylinder(double radio, double altura, int divisiones)</code>	Crea una esfera especificando su tamaño y resolución.

Un objeto 3D tiene tres dimensiones: *x*, *y*, *z*. La siguiente figura muestra el sistema de coordenadas 3D utilizado en JavaFX. La dirección positiva del eje *x* apunta a la derecha desde el origen; la dirección positiva del eje *y* apunta hacia abajo; y la dirección positiva del eje *z* apunta a la pantalla. Las direcciones negativas en los ejes (no se muestran), se extienden en direcciones opuestas al origen.



El material de textura de una figura 3D se especifica con la clase *PhongMaterial*. Esta clase tiene métodos como *setDiffuseColor* que especifica el color del material. Con el método *setSpecularColor* se define el color especular del material (brillo que produce una fuente de luz aplicada a la figura). Luego, se invoca al método *setMaterial* de la figura pasándole como parámetro el objeto *PhongMaterial*.

Con el método *setDrawMode* se establece como los polígonos que conforman la textura de la figura 3D serán dibujados.

La visualización en 3D en JavaFX se logra utilizando luces (clase *PointLight*) y cámaras (clase *PerspectiveCamera*). Las luces y las cámaras también son nodos que se agregan a la escena, la escena se enciende con luces y se ve con una cámara. Las posiciones de luces y cámaras en el espacio determinan las áreas iluminadas y visibles de la escena.

## Objetivos de aprendizaje

Al finalizar este ejercicio, serás capaz de:

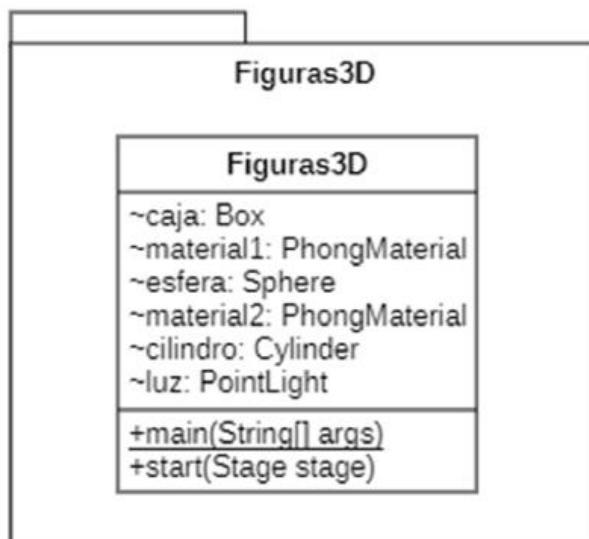
- Entender las clases que representan figuras 3D en JavaFX.
- Dibujar figuras 3D predefinidas.
- Utilizar fuentes de luz aplicadas a figuras 3D en JavaFX.

## Enunciado: Figuras 3D

Desarrollar una ventana gráfica con las siguientes figuras 3D:

- Un *Box* de tamaño 100 en sus 3 dimensiones de color amarillo en la posición: x=150, y=0; z=400.
- Una esfera de radio 60 de color naranja con 40 divisiones en su textura, en la posición: x =300, y=-5, z=400.
- Un cilindro de radio 75 y altura 100 de color difuso y especular naranja en la posición: x=500, y=5, z=500.
- El punto de luz de las figuras está ubicado en la posición: x=350, y=100, z=300 y la cámara está ubicada en x=100, y=-50, z=300..

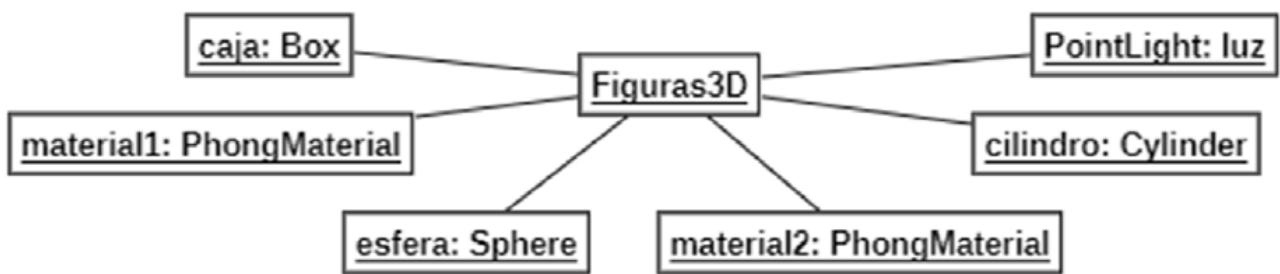
## Diagrama de clases



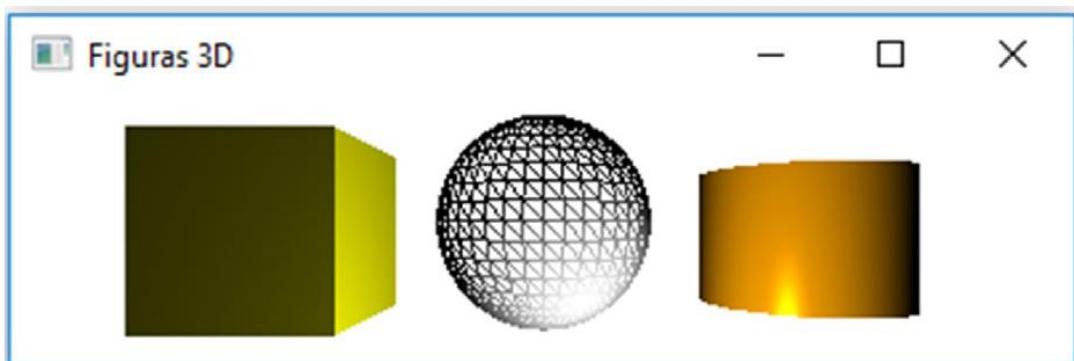
## Explicación del diagrama de clases

El diagrama de clases muestra un paquete denominado “Figuras3D”, que contiene una sola clase denominada también “Figuras3D”. Esta clase es una ventana gráfica que tiene atributos para indicar diferentes figuras tridimensionales que se presentarán en la ventana: un cubo (`Box`), una esfera (`Sphere`) y un cilindro (`Cylinder`). Los atributos tienen visibilidad de paquete indicada con el símbolo `~`. La clase cuenta con dos métodos: `start` que inicia la aplicación gráfica y el método `main` (punto de entrada a la aplicación). El método `main` es un método estático, por lo cual se representa con su texto subrayado. Los dos métodos son públicos, lo cual se indica con el símbolo `+`.

## Diagrama de objetos

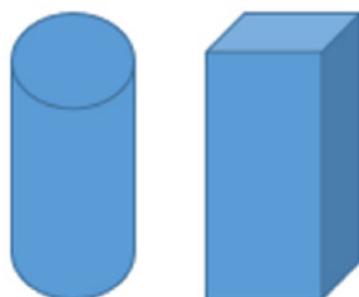


## Ejecución del programa



## Ejercicios propuestos

- Desarrollar la siguiente escena 3D:



## Instrucciones java utilizadas en el ejercicio

Clase	Método	Descripción
<i>Shape3D</i>	<code>void setMaterial(material1)</code>	Establece el material de la superficie de la figura 3D.
	<code>void setDrawMode(DrawModel valor)</code>	Establece cómo los polígonos de la textura de la figura 3D serán dibujados.
<i>Stage</i>	<code>Stage(Parent root, double anchura, double altura)</code>	Crea un objeto <i>Scene</i> para un nodo específico con un tamaño específico.
	<code>void setScene(Scene valor)</code>	Especifica la escena utilizada en este escenario.
	<code>void setTitle(String valor)</code>	Establece el título del escenario.
	<code>void sizeToScene()</code>	Establece el tamaño del escenario.
	<code>void show()</code>	Muestra la ventana.
<i>Scene</i>	<code>void setCamera(Camera valor)</code>	Establece una cámara para una escena.
<i>PhongMaterial</i>	<code>PhongMaterial()</code>	Crea un objeto <i>PhongMaterial</i> .
	<code>void setDiffuseColor(Color valor)</code>	Establece el color del material.
	<code>void setSpecularColor(Color valor)</code>	Establece el color especular (brillo que produce la fuente de luz).
<i>PointLight</i>	<code>PointLight()</code>	Crea un objeto fuente de luz.
<i>PerspectiveCamera</i>	<code>PerspectiveCamera()</code>	Especifica una cámara para renderizar una escena.
<i>Group</i>	<code>Group()</code>	Constructor de un contenedor con una lista de elementos que se presentan en orden.