

**UNIVERSIDAD NACIONAL MAYOR DE SAN
MARCOS**

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA



CURSO : COMPUTACIÓN GRÁFICA Y VISUAL

TEMA : PROYECTO FINAL

PROFESOR : HERMINIO PAUCAR

CICLO : VII

ALUMNOS	: Barrozo Figueroa Osti Katriel	17200320
	Chavez Chavez, Fernando	17200101
	Chavez Malca, Emerzon Jose	18200079
	Martínez Bravo, Marín	18200090
	Pajuelo Cieza, Junior Alexis	17200080
	Pumacarhua Alvarez, Jefferson	17200123

Contenido

1. Caratula
2. Introducción:
3. Marco Teórico (Explicación teórica de: algoritmos, librerías, y otras herramientas que usaron para su proyecto)
4. Metodología de Desarrollo (Explicar la forma como abordaron el problema y explicar su procedimiento paso a paso con sus propias palabras)
 1. Incluir Diagrama de Clases
 2. Diagrama de Secuencia
 3. Prototipos Mockups de la aplicación
5. Aporte (Explicar cuál fue su aporte a la aplicación)
6. Conclusiones y Recomendaciones.
7. Referencias en formato APA.

1. Introducción:

El constante avance de los gráficos y aplicaciones 3D, obligan a los próximos profesionales a adquirir conocimientos elementales del funcionamiento interno de lo que se conoce como “Computación Gráfica”, de hecho, todavía impera en todos la misma idea o noción general del uso que se le da a la computación gráfica (Desarrollo de videojuegos, imágenes, películas 3D virtuales), sin embargo, la computación gráfica abarca ámbitos aún más complejos e importantes pues está presente y es parte importante en la medicina (Visualización interna de un organismo, placas, ecografías, etc), aeronáutica, topología, etc, básicamente en todo lo que necesite una visualización simulada de su entorno.

Debido a la importancia de la misma, los profesionales que construirán el camino hacia el mañana requieren potenciar sus habilidades tecnológicas.

Teniendo esto en mente. El proyecto presentado cumple con la función de generar conocimiento teórico práctico sobre la creación de imágenes 3D mediante uso de texturas, de manera que los implicados conozcan los fundamentos de creación básica de lo que ya es parte de nuestra vida diaria; para variar, estas representaciones serán la simulación de de figuras de revolución a fin de reducir la complejidad de su creación.

2. Marco Teórico (Explicación teórica de: algoritmos, librerías, y otras herramientas que usaron para su proyecto)

2.1. Librerías utilizadas:

2.1.1. GLM: Librería de carácter matemático empleada para el desarrollo de software gráfico, en el presente proyecto fue utilizado por el grupo de trabajo para la realización de transformaciones de escala y rotación de los sólidos.

2.1.2. GLFW: Esta librería permite crear ventanas, contextos y superficies, el equipo de trabajo empleó esta librería debido a que OpenGL por sí mismo no proporciona herramientas para el control de ventanas.

2.1.3. GLEW: Librería utilizada por el equipo de trabajo para la carga y manejo de extensiones de OpenGL.

2.2. Herramientas adicionales

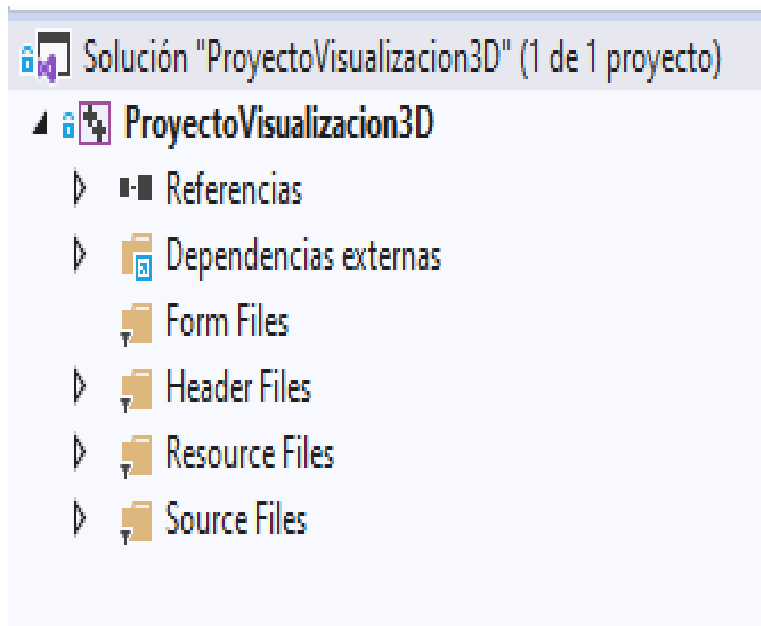
2.2.1. QT: **Qt** tiene una extensa librería con clases y herramientas para la creación de aplicaciones GUI, utilizamos la extensión de QT en visual studio para la creación de la interfaz gráfica

2.2.2. Visual Studio: Un IDE grandioso con una infinidad de complementos, el equipo se apoyó de este IDE para la creación del proyecto ya que permite su incorporación con OpenGL y QT.

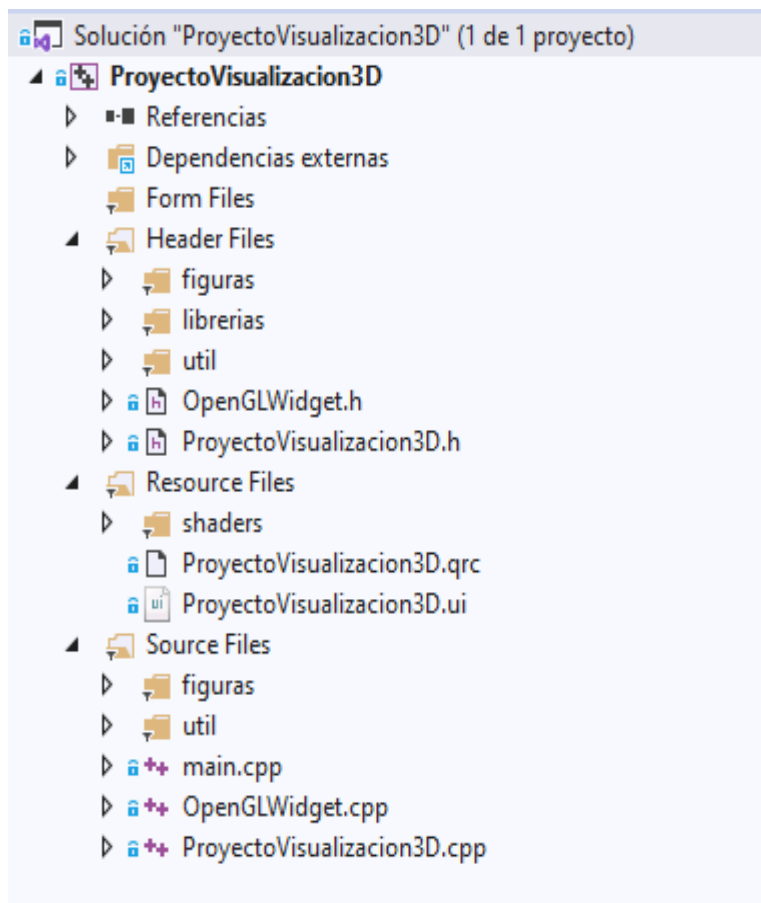
3. Metodología de Desarrollo (Explicar la forma como abordaron el problema y explicar su procedimiento paso a paso con sus propias palabras)

3.1. Incluir Diagrama de Clases

3.1.1. Estructura general:

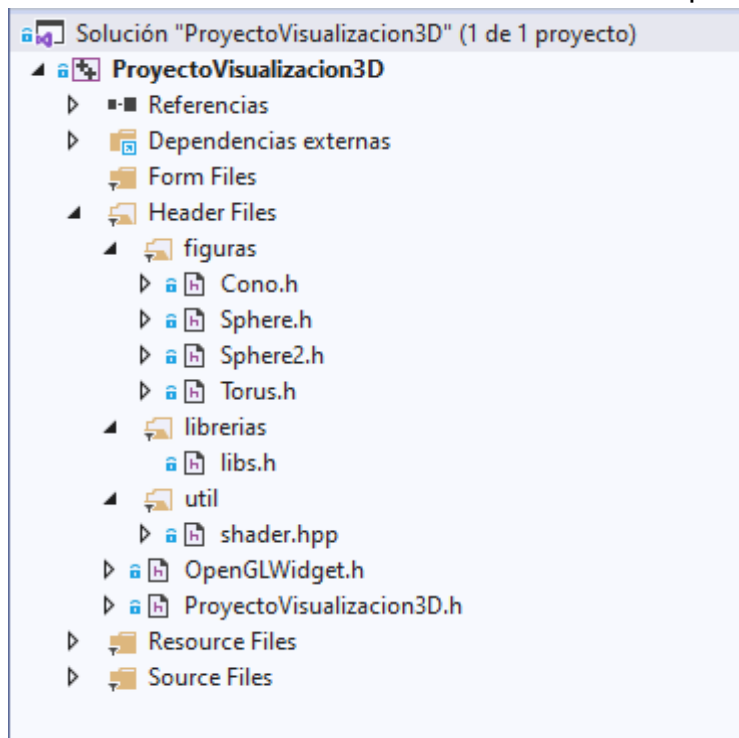


3.1.2. Estructura interna 1:

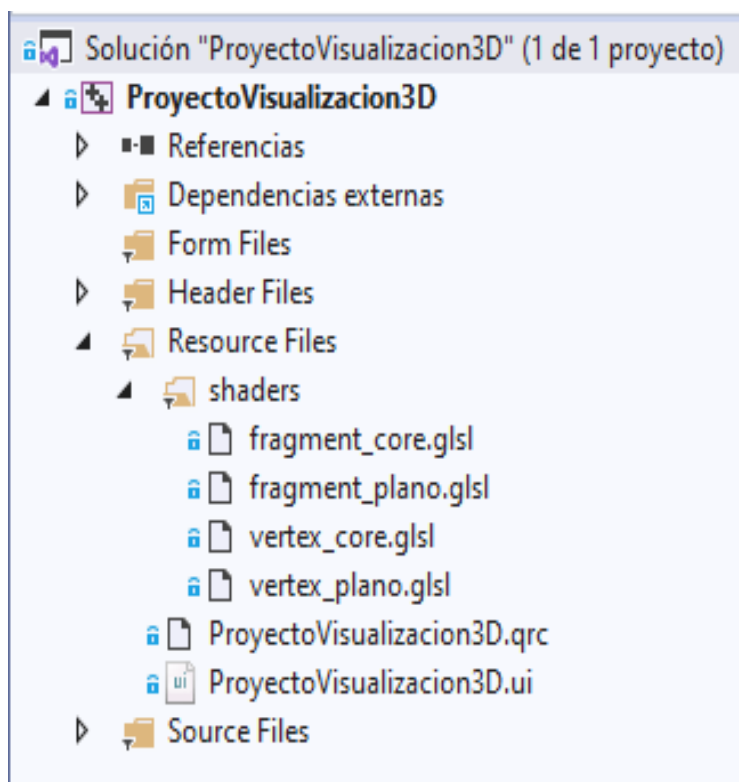


3.1.3. Estructura interna 2:

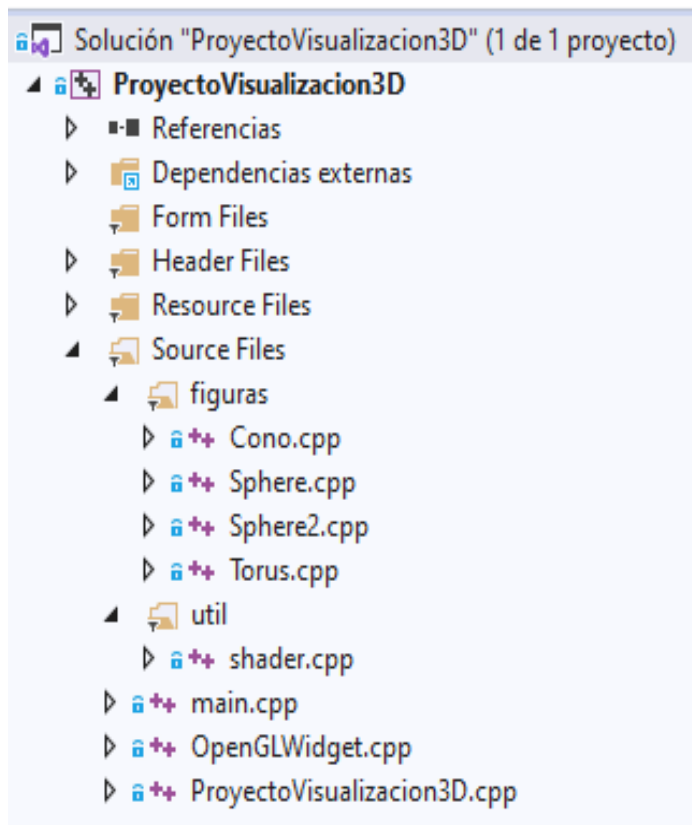
3.1.3.1. Estructura de la carpeta Header Files



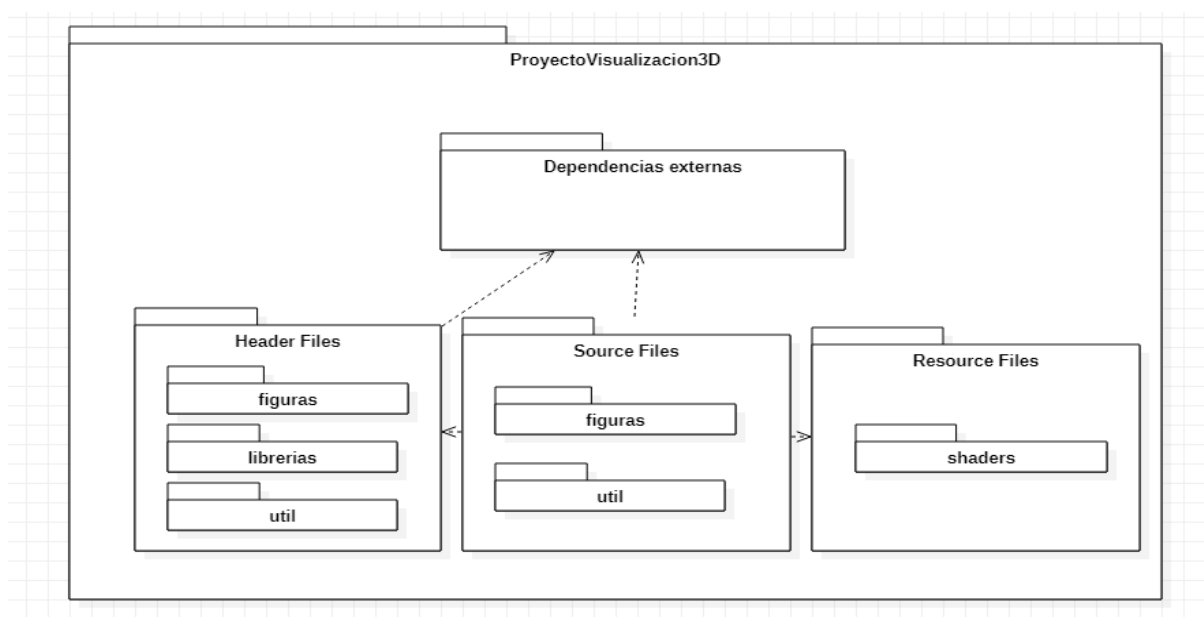
3.1.3.2. Estructura de la carpeta Resource Files:



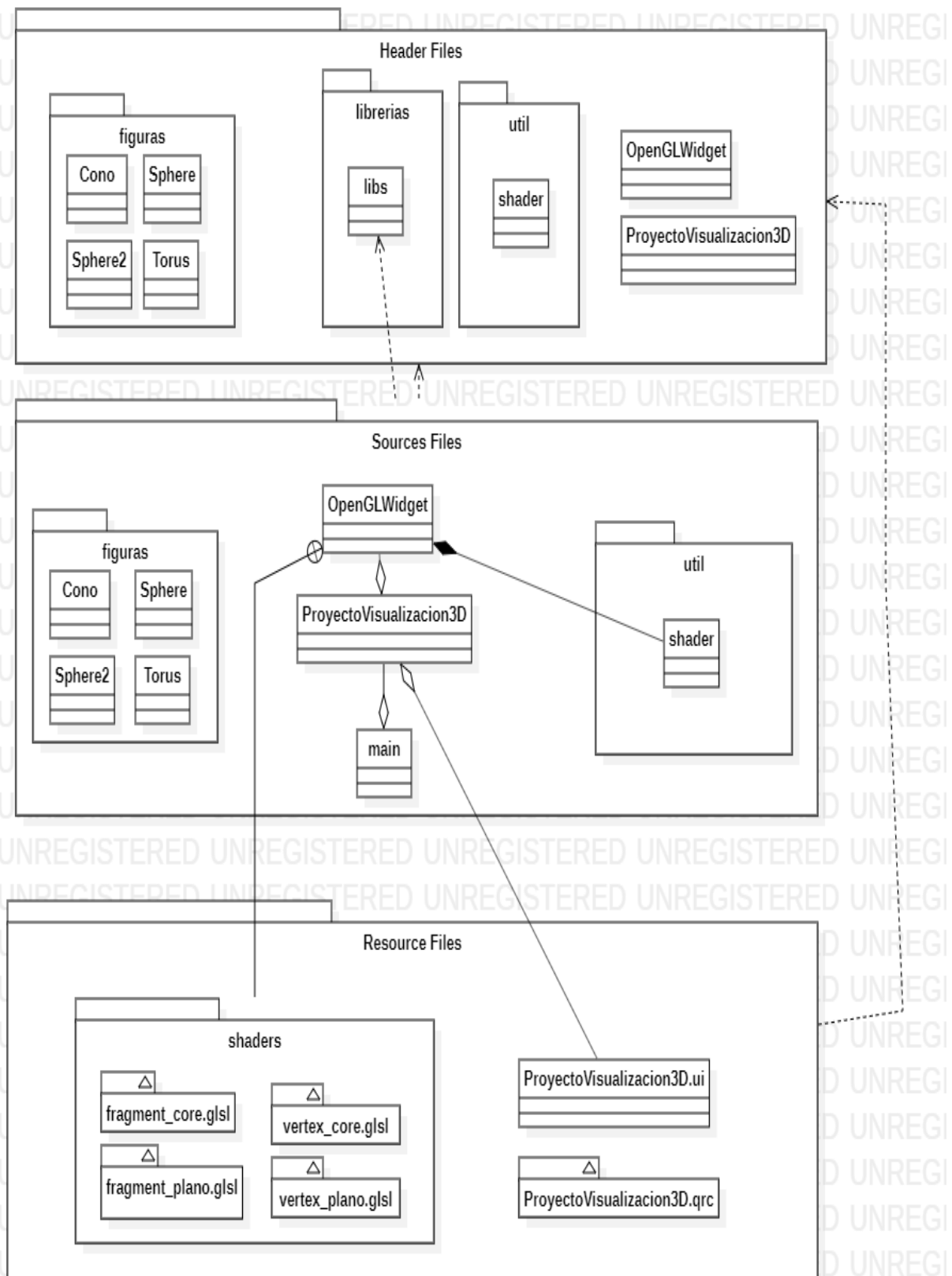
3.1.3.2. Estructura de la carpeta Source Files:



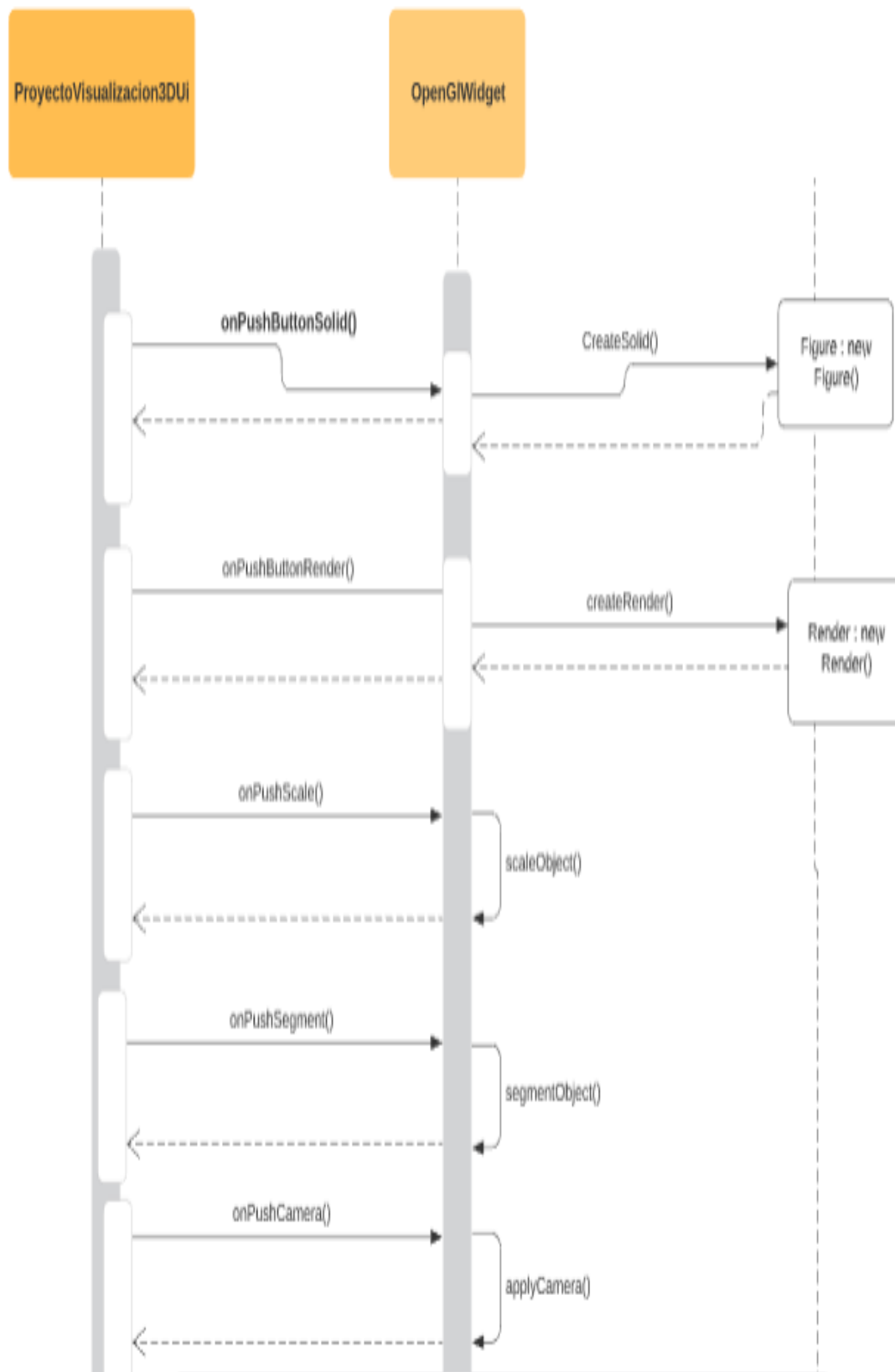
3.1.2. Diagrama de Paquetes:



3.1.2. Diagrama de Clases:

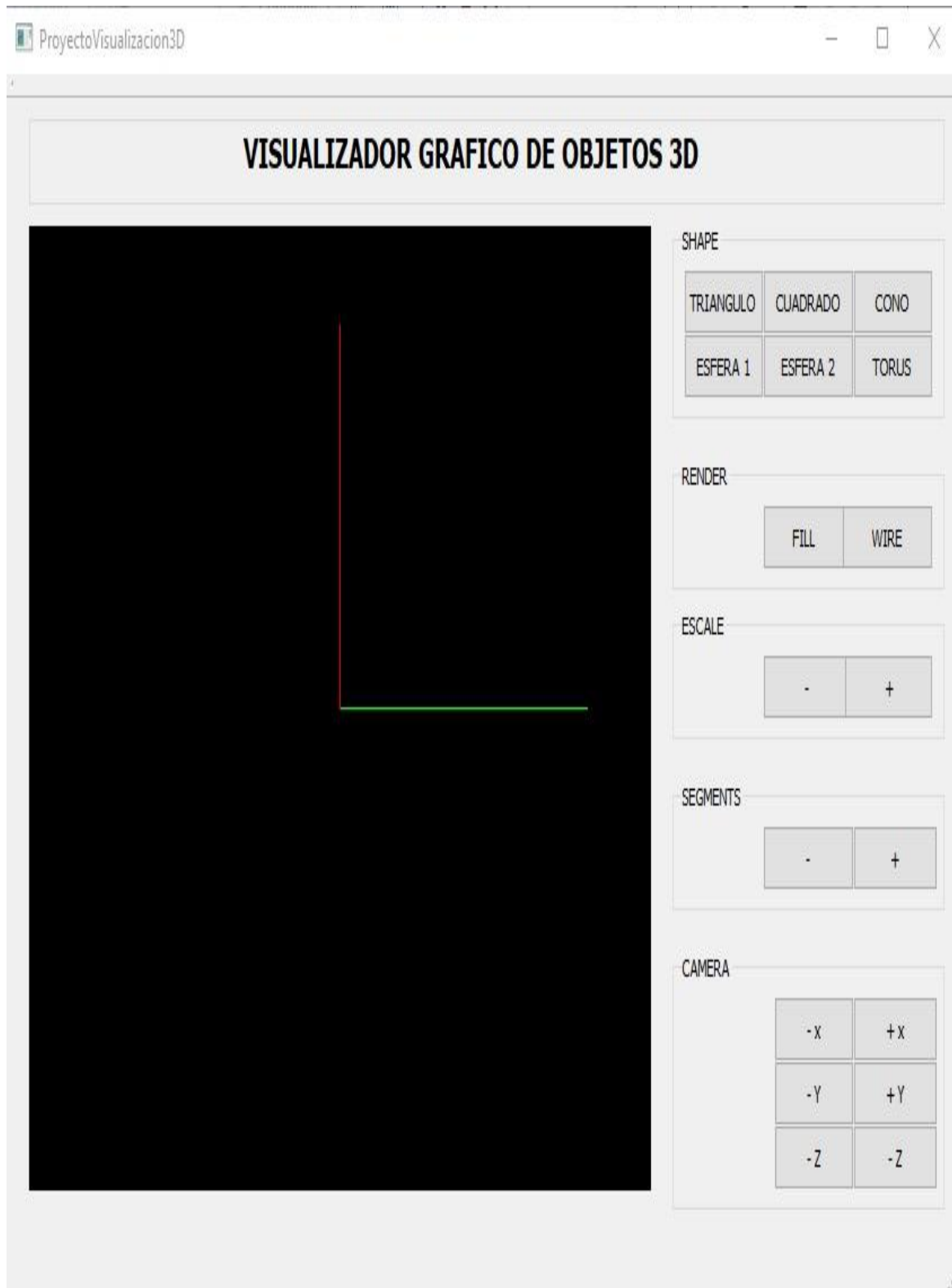


3.2. Diagrama de Secuencia

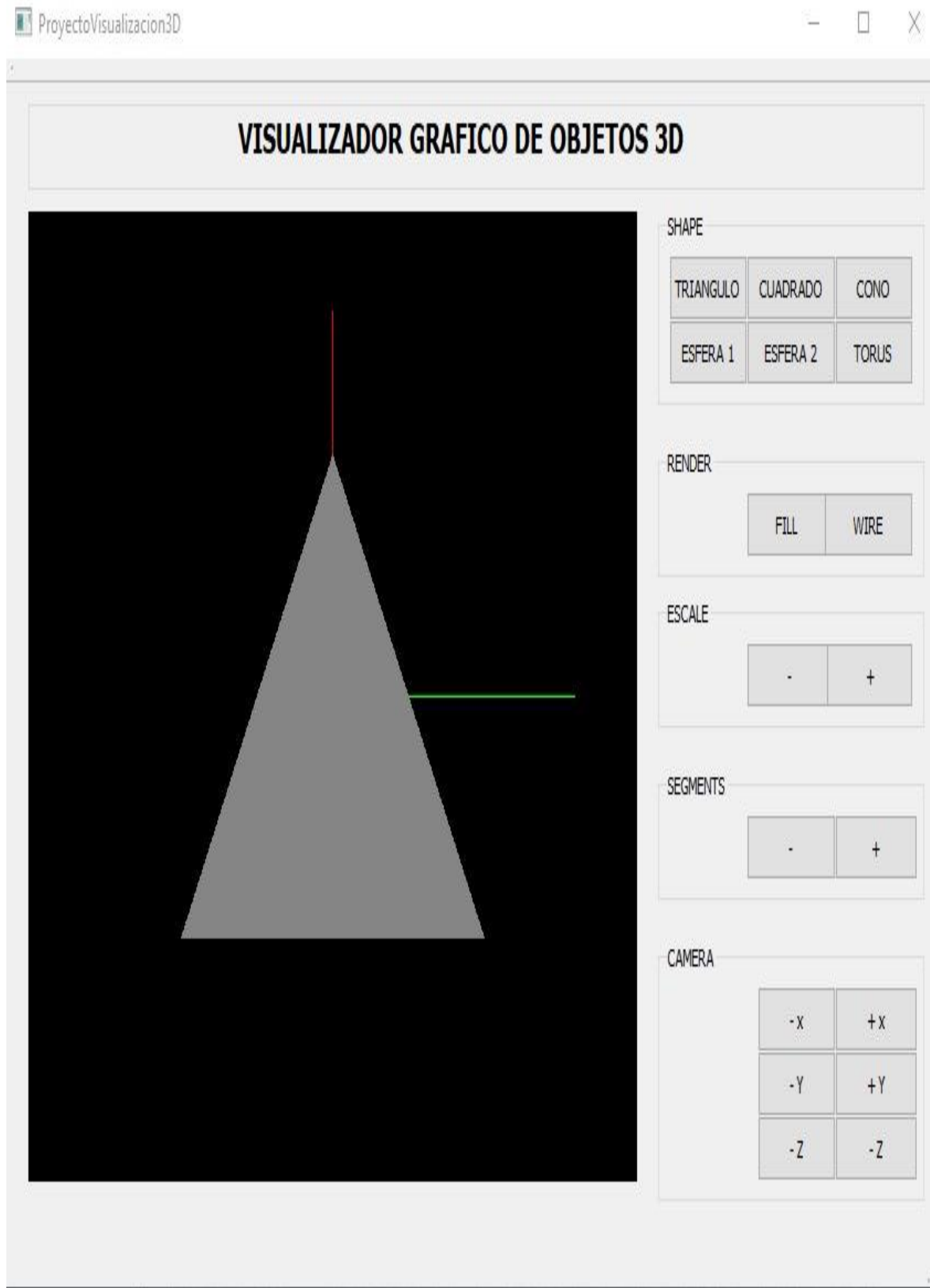


3.3. Prototipos Mockups de la aplicación

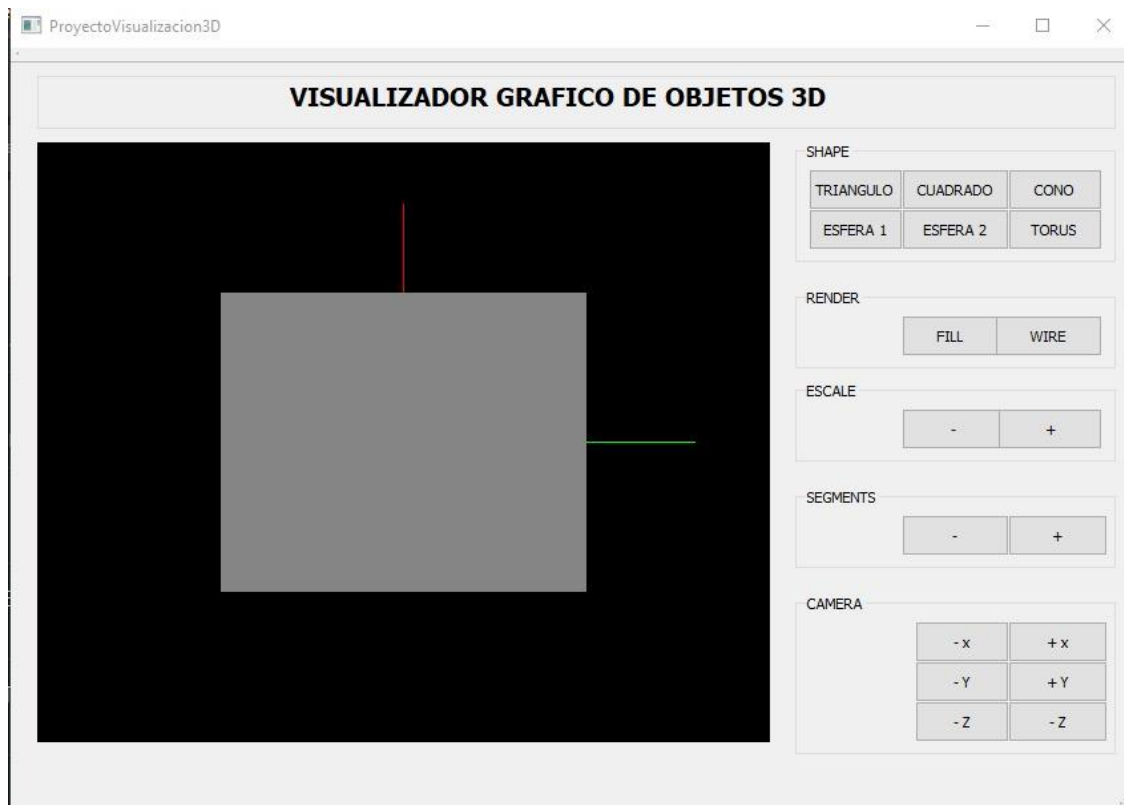
Pantalla principal del proyecto: contiene el widget de QT para openGL incorporado ademas de las distintas figuras y opciones a elegir



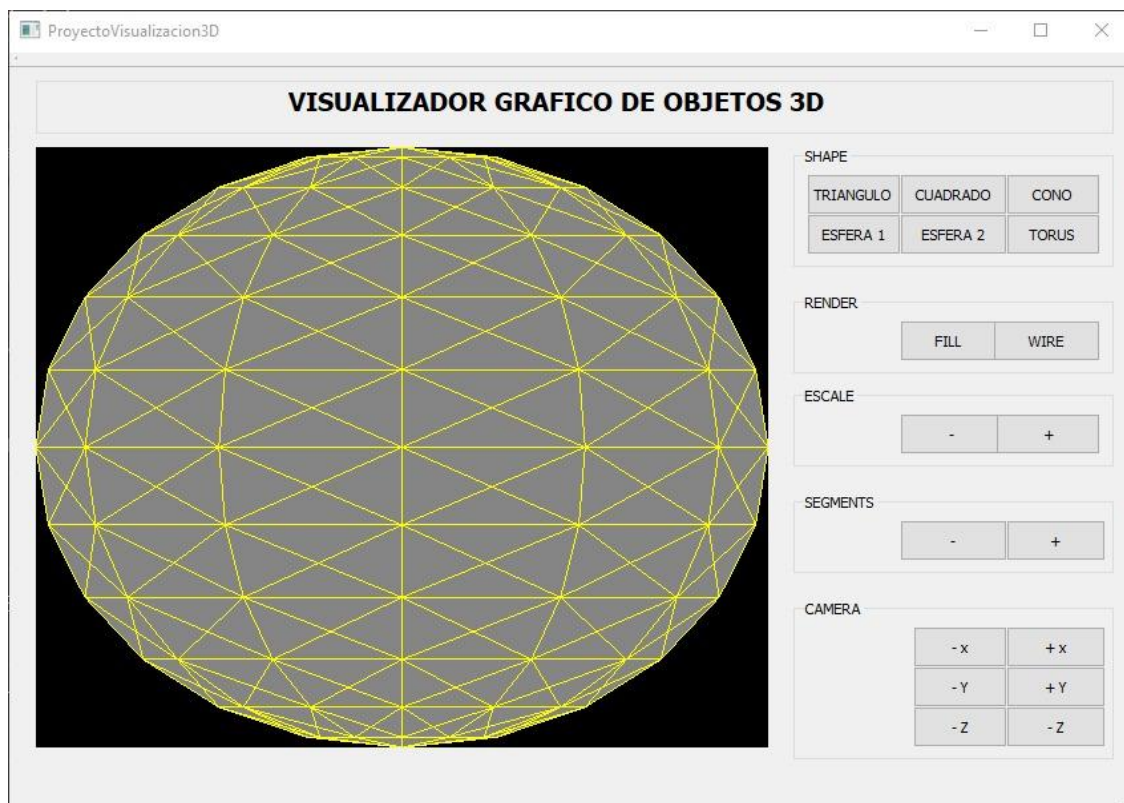
Triangulo: como se observa en la imagen al seleccionar al boton nos muestra un triangulo por defecto, dicho triangulo puede ser escalado, renderizado o segmentado segun las preferencias del usuario.



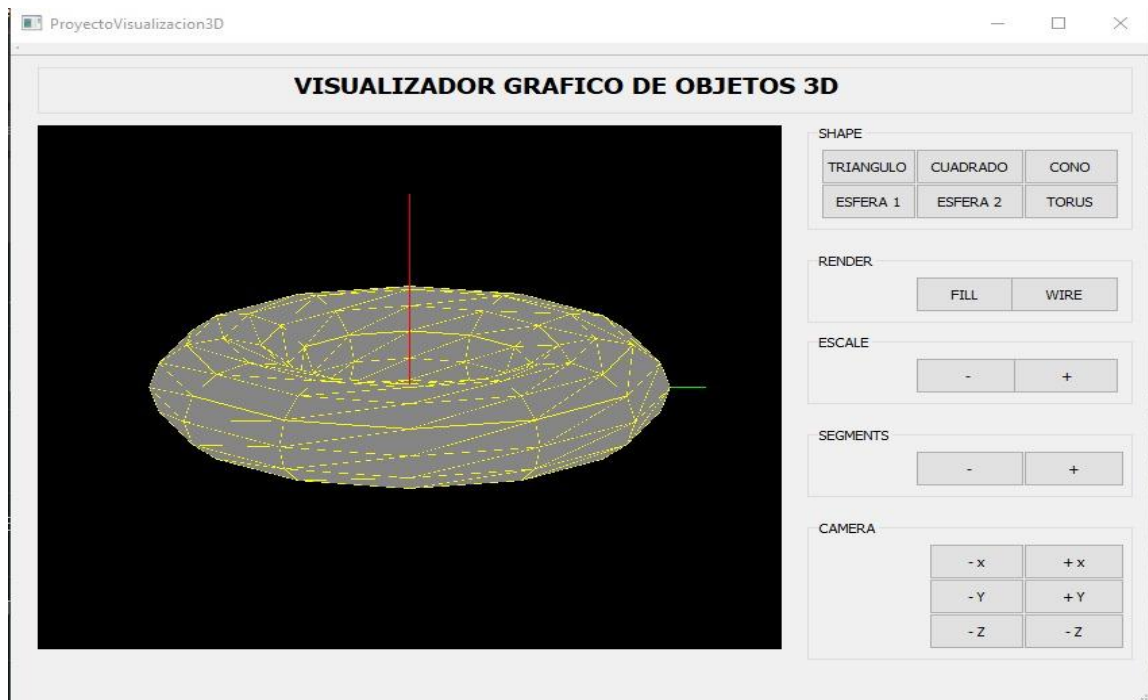
De igual manera con el cuadrado



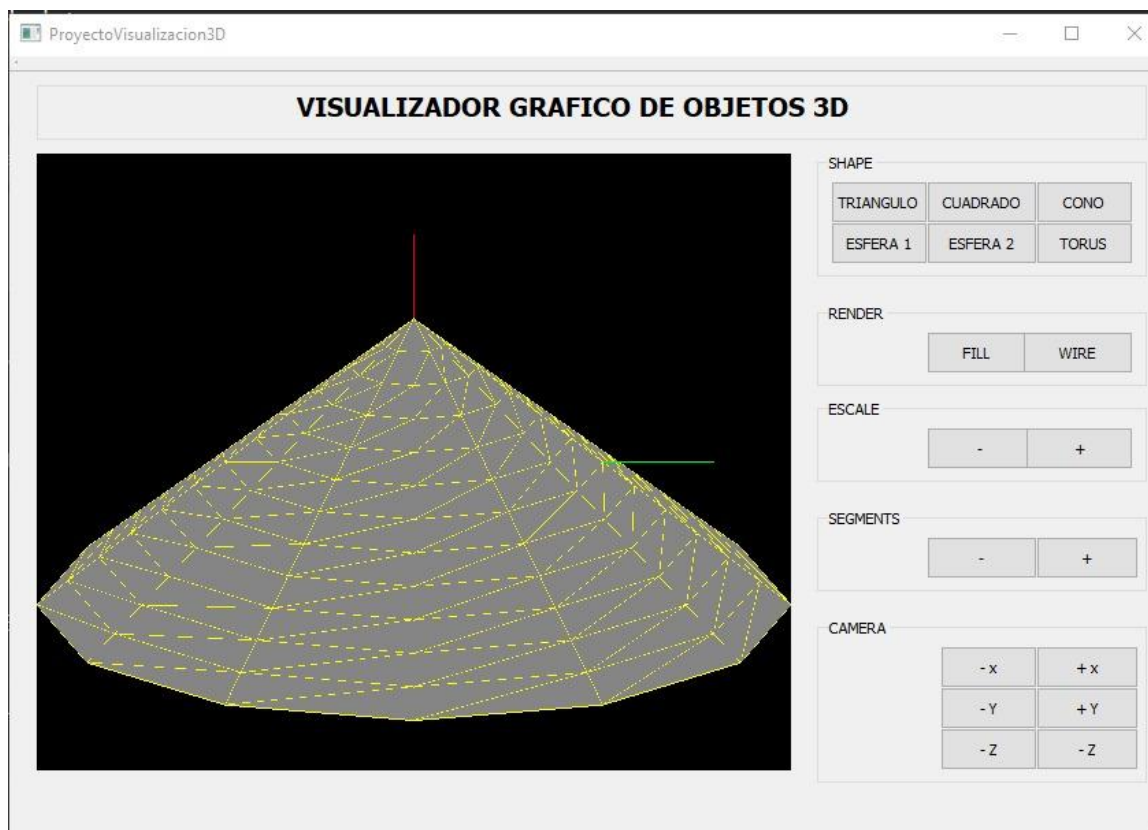
De igual manera con la esfera



De igual manera con la Torus



De igual manera con la Piramide



4. Aporte (Explicar cuál fue su aporte a la aplicación)

Los aportes de los miembros del grupo se detallarán a continuación con la consecuente justificación de la nota de participación individual.

Debido a la complejidad que representaba para algunos miembros el uso de Qt, se optó por dividir el trabajo en: Aquellos que diseñan las figuras en OpenGL, aquellos que usan esos diseños en Qt y aquellos que no pudieron participar tan activamente en el desarrollo de las aplicaciones.

Barrozo Figueroa, Osti Katriel: Se encargó de la implementación del código en Visual Studio y Qt además de su aporte a todo el proyecto.

Chavez Chavez, Fernando: Se encargó de la creación del interfaz visto en Qt, configuración e incorporación a Visual Studio además de la instalación de las librerías asociadas a OPENGL(GLEW, GLM).

Chavez Malca, Emerzon : Se encargó del diseño del cono.

Martinez Bravo, Martin: Se encargó del diseño de la esfera.

Pajuelo Cieza, Junior Alexis: Realizó su participación en la documentación

Pumacarhua Alvarez, Jefferson: Se encargó del diseño del toroide y documentación

Además de los puntos anteriores, todos los miembros realizaron su parte en la documentación del proyecto.

5. Conclusiones y Recomendaciones.

Las conclusiones obtenidas del proyecto se muestran a continuación:

1. El proyecto ha cumplido con su principal función: asegurar la comprensión de los modelos teóricos de representación en los miembros del proyecto.
2. El proyecto, si bien parecía complejo, no ha representado un esfuerzo más allá de lo imaginado; en verdad, con una semana adicional el equipo estaría preparado para una presentación modelo.
3. El desarrollo remoto, ha constituido un impedimento para el desarrollo del proyecto.
4. La gestión de tiempo para el proyecto del equipo fue deficiente, teniendo en cuenta a los miembros con conocimientos reducidos, problemas que no fueron considerados y que se convirtieron en la traba del avance del proyecto.
5. Finalmente y a pesar de todo, se concluye que el proyecto ha sido exitoso, al menos desde la perspectiva fundamental, y todos han tenido la experiencia gratificante que, observar resultados propios, conlleva.

Las recomendaciones generadas después de la implementación del proyecto son:

1. Gestionar los tiempos de entrega y encargarse de quienes presentan dificultades.
2. Valerse de medios necesarios para llevar el adecuado control de las actividades de los miembros del grupo.
3. Hacer uso de la herramienta QT creator para facilitar el diseño de interfaz y su consecuente presentación.
4. Utilizar recursos adicionales a los especificados según la conveniencia del desarrollador, ya sea un lenguaje de programación diferente o un entorno de desarrollo variado.

6. Referencias en formato APA.

- Gordon, V., Clevenger, J. (2018). Computer Graphics Programming in OpenGL with C++.
- SurajSharma. OpenGL/C++ 3D Tutorial 04 - Linking GLFW in Visual Studio (2018, enero 28)
- <https://www.youtube.com/watch?v=7cWjtE9C UM&list=PL6xSOsbVA1eYSZTKBxnoXYboy7wc4yg-Z&index=5>
- Bogoto(2020). QT5 TUTORIAL OPENGL WITH QGLWIDGET - 2020
https://www.bogotobogo.com/Qt/Qt5_OpenGL_QGLWidget.php
- QT OpenGL (2020). Cube OpenGL ES 2.0
<https://doc.qt.io/archives/qt-5.8/qtopengl-cube-example.html>
- Jaime King. 3D Computer Graphics Using OpenGL
<https://www.youtube.com/playlist?list=PLRwVmtr-pp06qT6ckboaOhnm9FxmzHpbY>