

# **Documento de proyecto.**

## **Visualización de figuras triangulares en 2D y 3D.**

**Fecha** [10/01/2021]

**Versión** [1.5]

**Nombre del grupo** Grupo N°4.

**Integrantes** Benjamín Quiroz.  
**del grupo** Rodrigo Lobos.  
Pedro Cisternas.

## Historia del Documento.

<b>Versión</b>	<b>Fecha</b>	<b>Explicación del cambio</b>	<b>Autor</b>
0.1	12/10/2020	Edición de portada e índice.	Rodrigo Lobos
0.2	12/10/2020	Edición de introducción.	Benjamín Quiroz
0.3	13/10/2020	Edición de marco teórico y referencia.	Benjamín Quiroz
0.4	14/10/2020	Agregado de imágenes y figuras.	Rodrigo Lobos
0.5	15/10/2020	Revisión de fe de erratas.	Rodrigo Lobos
0.6	15/10/2020	Revisión de redacción y orden.	Pedro Cisternas
0.7	03/12/2020	Edición de informe para Hito 2.	Rodrigo Lobos/ Benjamín Quiroz
0.8	09/12/2020	Revisión final de redacción y orden Hito 2	Benjamín Quiroz
0.9	09/12/2020	Agregar información en la descripción de programa.	Rodrigo Lobos
1.0	11/12/2020	Actualización de capturas de funcionamiento y números de páginas.	Pedro Cisternas
1.1	06/11/2021	Edición de errores en función de rúbrica hito 2.	Benjamín Quiroz
1.2	07/11/2021	Edición de errores en función de rubrica hito 2.	Benjamín Quiroz
1.3	09/11/2021	Actualización de capturas de funcionamiento.	Pedro Cisternas
1.4	10/11/2021	Revisión final de informe hito 3.	Rodrigo Lobos
1.5	10/01/2021	Edición de páginas de informe	Rodrigo Lobos

# Índice de Materias.

<b>Portada</b>	<b>i</b>
<b>Historia del Documento</b>	<b>ii</b>
<b>1 Resumen Ejecutivo</b>	<b>1</b>
1.1 Introducción	1
<b>2 Planificación</b>	<b>2</b>
2.1 Objetivo general	2
2.2 Objetivos específicos	2
2.3 Actividades y Milestones	3
2.4 Cronograma	4
<b>3 Diseño de Solución</b>	<b>5</b>
3.1 Explicación de componentes creadas o utilizadas	5
3.2 Diagramas de las componentes y sus interacciones	5
<b>4 Resultados</b>	<b>10</b>
4.1 Descripción de funcionamiento	10
4.2 Resultados obtenidos, explicación y capturas	10
<b>5 Anexos</b>	<b>13</b>
5.1 Marco teórico	13
<b>6 Referencias</b>	<b>14</b>

## **Lista de Imágenes.**

Figura 1: Diagrama de flujo de datos del programa.	6
Figura 2: Diagrama de flujo de datos de equilátero.	7
Figura 3: Diagrama de flujo de datos de isósceles.	8
Figura 4: Diagrama de flujo de datos de escaleno.	9
Imagen 1: Prueba de programa N°1 (ingreso de datos por terminal).	11
Imagen 2: Prueba de programa N°2 (creación de triángulo y modelo 3D de pirámide).	11
Imagen 3: Prueba de programa N°3 (rotación de pirámide en torno a sus ejes).	12

# 1 Resumen Ejecutivo.

## 1.1 Introducción.

Este proyecto surge para facilitar a universitarios, estudiantes y profesores del área de matemáticas, una herramienta didáctica, en la cual se puedan visualizar triángulos en 2D y pirámides en 3D a partir de cierta información entregada por el usuario a la plataforma.

Pensamos que este proyecto es útil puesto que las figuras dentro de la aplicación se pueden medir de una manera mucho más exacta que al graficarla a mano. Sabemos lo difícil que es ilustrar una figura dentro de un papel y o imaginarla, por lo que el proyecto permitirá realizar esta función de una manera mucho más sencilla, precisa y visual.

Los triángulos tienen una gran importancia en la geometría, pues todo polígono puede ser descompuesto o formado por triángulos, además de ser una figura que proporciona gran resistencia y estabilidad.

Cuando se utilizan materiales de construcción para formar un triángulo, el diseño tiene una gran base y el pináculo de la parte superior es capaz de administrar el peso porque la energía se distribuye a través de todo el triángulo.

El estudio de los triángulos es muy amplio, tanto que ha generado en sí misma una rama de la Geometría y de las Matemáticas: la Trigonometría, que estudia las relaciones entre los lados y los ángulos de un triángulo.

Aunque no nos demos cuenta, los triángulos están presentes continuamente en nuestra vida cotidiana ya que en cualquier actividad que realicemos, utilizamos o vemos algo que fue construido con ayuda de un triángulo. La estructura compuesta por triángulos, más conocida es la Torre Eiffel, en la que vemos la aplicación de múltiples triángulos en su estructura, para dar la rigidez y estabilidad a esta gran torre de 300 m de altura.

Ya hablando del programa, al iniciarlo se le pedirá al usuario una serie de datos los cuales se irán recolectando para luego en base a estos, formar la figura deseada y mostrarla por una pantalla externa (funcionando de la misma manera para las figuras en 2D y 3D). Para la creación y visualización de las figuras se utilizará el IDE multiplataforma Qt Creator y dos de sus librerías incluidas, llamadas Qpainter y QtOpenGL. Para el uso de nuestro sistema, se requerirá al usuario el programa de visualización nombrado recientemente, y también un computador capaz de procesar el sistema. El programa solo se limitará a mostrar triángulos válidos, este mismo se encargará de notificar al usuario si con los datos ingresados es posible o no. Para el formato en 3D, las bases sólo podrán ser triangulares equiláteras o cuadradas.

Este informe estará compuesto de cinco secciones, la primera trata sobre los objetivos y planificación del proyecto. La segunda sobre los componentes y funcionamiento del programa. La siguiente sección trata sobre los resultados al hacer funcionar el programa, mostrando capturas y dando una conclusión. La cuarta sección es el marco teórico utilizado para la creación del programa y finalmente se hace mención a las referencias utilizadas para el proyecto.

## 2 Planificación.

### 2.1 Objetivo General.

El objetivo principal de nuestro proyecto es crear una interfaz la cual sea capaz de mostrarle al usuario por pantalla la figura que este desee ver, dentro de una serie de posibilidades.

### 2.2 Objetivos específicos.

- Crear un programa que guarde como variables todos los datos necesarios para la creación de la posterior figura.
- Crear un sistema que grafique la figura deseada utilizando el IDE Qt Creator.
- Desarrollar una interfaz para mostrar de manera organizada el programa funcionando en su totalidad.

### 2.3 Actividades y Milestones.

#### - Hito número 1:

En este hito se construyó el código base, el cual trata de pedirle al usuario todos los Datos necesarios para confeccionar la futura figura que este desee.

#### Actividades:

- configuración del programa inicial.
- edición de README.
- edición informe.
- edición y prueba de programa.
- aprender sobre el uso de GitLab y sus comandos.

#### - Hito número 2:

En este hito se concretó la visualización de las figuras utilizando Qt Creator y las variables obtenidas en el código base.

#### Actividades:

- estudio sobre el uso de qtcreator.
- edición de README.
- creación del programa para visualizar las figuras en Qtcreator.
- actualización de informe.
- estudio sobre la construcción de interfaces.

- **Hito número 3:**

En este hito se formalizó el programa de visualización de figuras mediante la creación de una interfaz para este.

**Actividades:**

- edición de README.
- actualización de informe.
- estudio sobre la construcción de interfaces.
- creación de la presentación final.

## 2.4 Cronograma.





## 3 Diseño de solución.

### 3.1 Explicación de componentes creados o utilizados.

Como pediremos por pantalla datos sobre el triángulo (y su respectiva altura en caso de haber pedido una pirámide), tendremos que crear variables para cada uno de estos datos. Luego, serán estas mismas variables las que utilizaremos para visibilizar las figuras al utilizar Qt Creator.

Para reconocer la figura (2D/3D), tipo de triángulo, lado o forma, utilizaremos condicionales que van a depender netamente de lo que se le pida al programa.

Crearemos una variable ‘visualizar’ que puede ser igual a los números 1 o 2, correspondiente a ‘triángulo’ o ‘pirámide’ respectivamente.

Si esta variable es igual a triángulo crearemos otra variable ‘tipo’, la cual podrá ser igual a los números 1, 2 o 3, correspondiente a ‘escaleno’, ‘isósceles’ o ‘equilátero’ respectivamente.

Dependiendo de a que sea igual la variable ‘tipo’, pediremos y guardaremos en distintas variables los lados del triángulo, 3 variables distintas en el caso que ‘tipo’ sea ‘escaleno’, 2 variables distintas en el caso que ‘tipo’ sea ‘isósceles’, y solo una variable en el caso que ‘tipo’ es igual a ‘equilátero’. Cabe recalcar que si faltan algunas de estas variables de los lados en triángulos como son el isósceles o escaleno, se va a pedir (ya sea en una o dos variables) el ángulo que conforman con uno o dos lados, y utilizando el teorema del seno y coseno se podrá deducir el lado restante y así poder entregar de igual manera los datos por pantalla.

Si la variable ‘visualizar’ es igual a pirámide crearemos otra variable ‘base’, la cual podrá ser igual a los strings ‘cuadrada’ o ‘triangular’, con el fin de saber de qué tipo será la base de la pirámide (triangular equilátera o cuadrada).

Para cualquier caso de la variable ‘base’, pediremos después la medida del largo de uno de sus lados basales, la cual guardaremos en una nueva variable ‘pirámide’. Ahora solo nos faltará pedirle al usuario la medida de la altura la cual guardaremos en una variable ‘H’. De este modo tendremos todas las variables necesarias para poder graficar y visualizar la figura deseada.

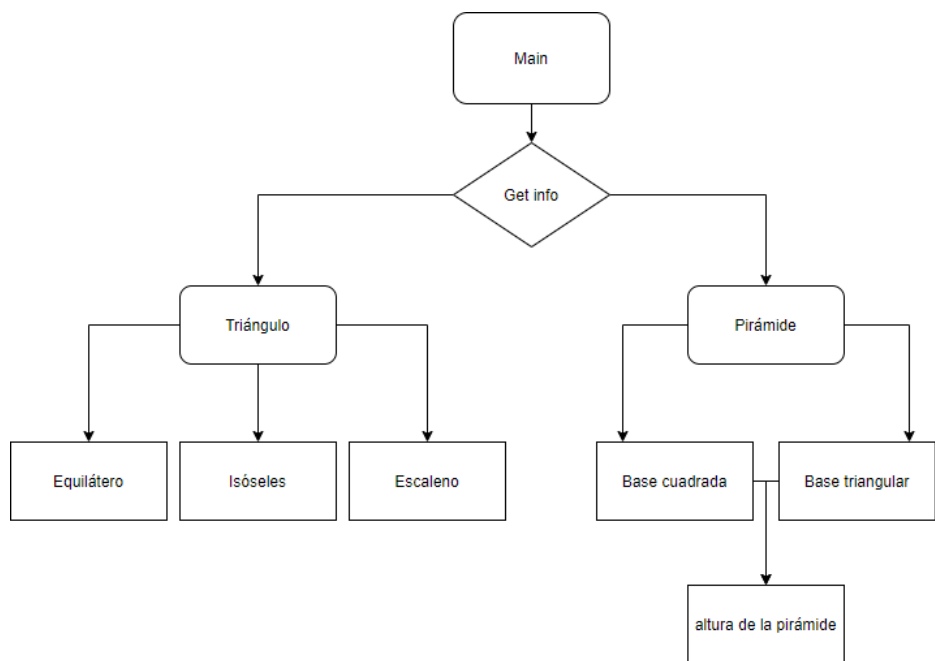
Luego de tener todas las variables necesarias ya guardadas, será momento de graficar la figura. Para esto utilizando el IDE de QtCreator, crearemos la función getPuntos(), la cual nos ayudará a poder graficar a través de coordenadas (x, y), ( para las figuras en 2D) al obtener los vértices de la figura, usaremos la biblioteca QPainter, la cual nos ayudará a visualizarlo en una ventana externa.

Para las figuras en 3D, usaremos la biblioteca QtOpenGL, específicamente la función glVertex3f(), la cual nos creará los vértices de la figura, en forma matricial, para finalmente con las funciones de la biblioteca, poder mostrar la pirámide por ventana.

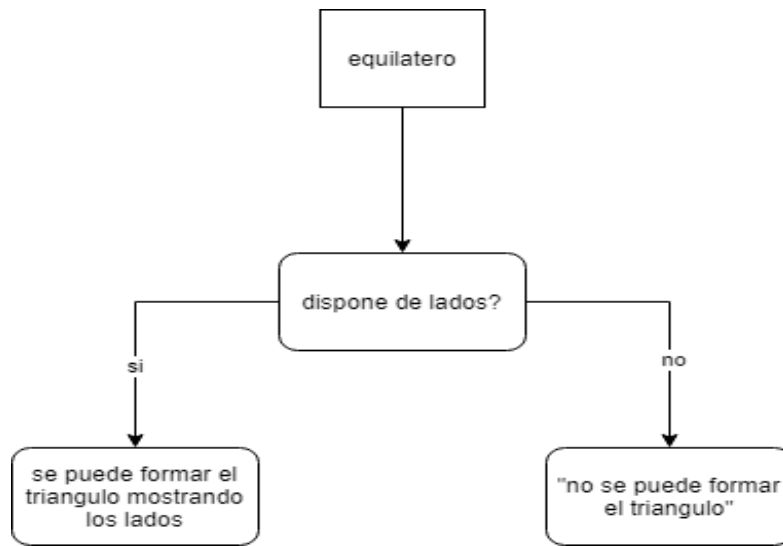
### 3.2 Diagramas de los componentes y sus interacciones.

Para poder expresar de manera más dinámica el funcionamiento e interacciones del programa, presentamos el siguiente diagrama de flujos:

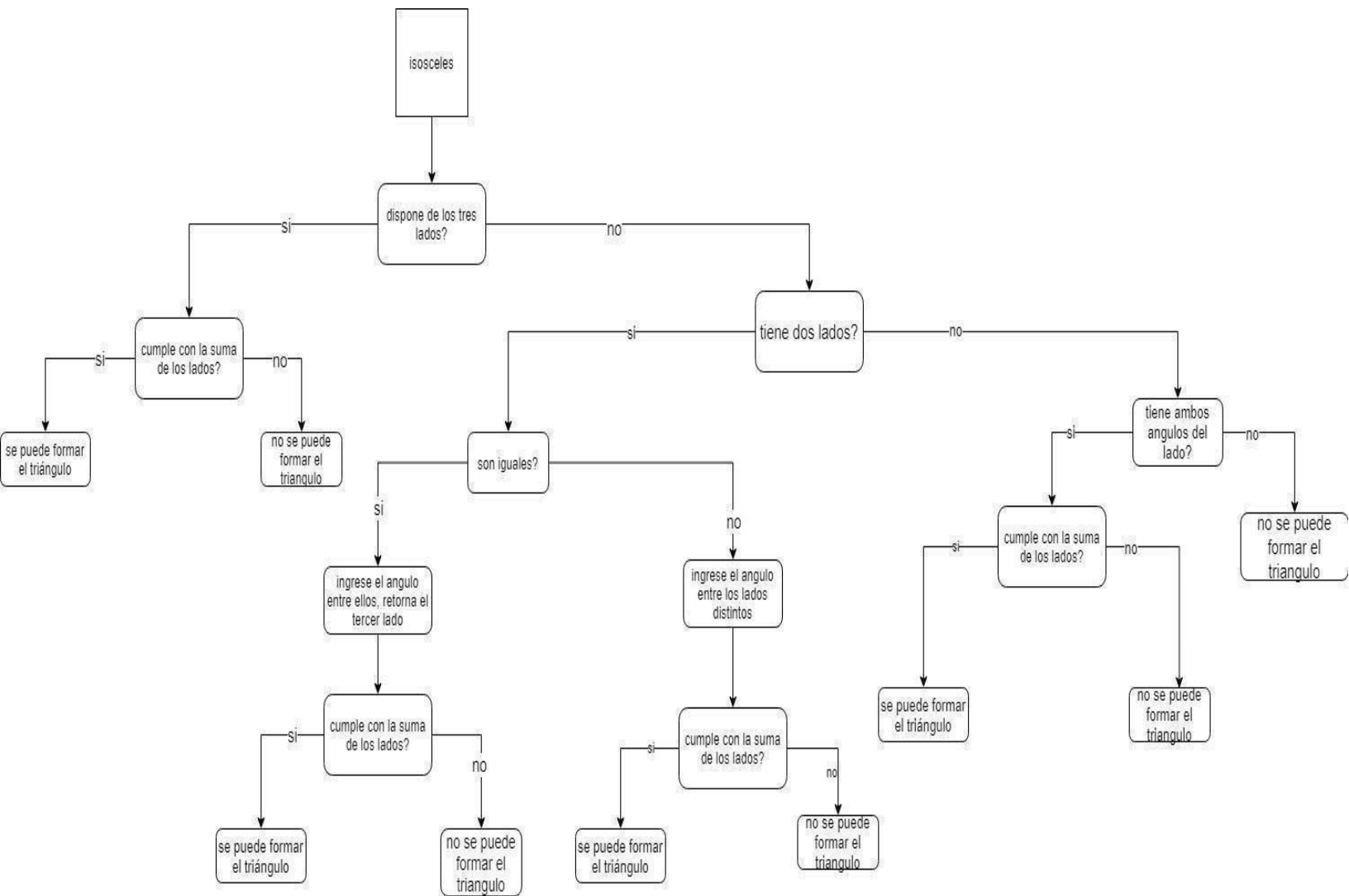




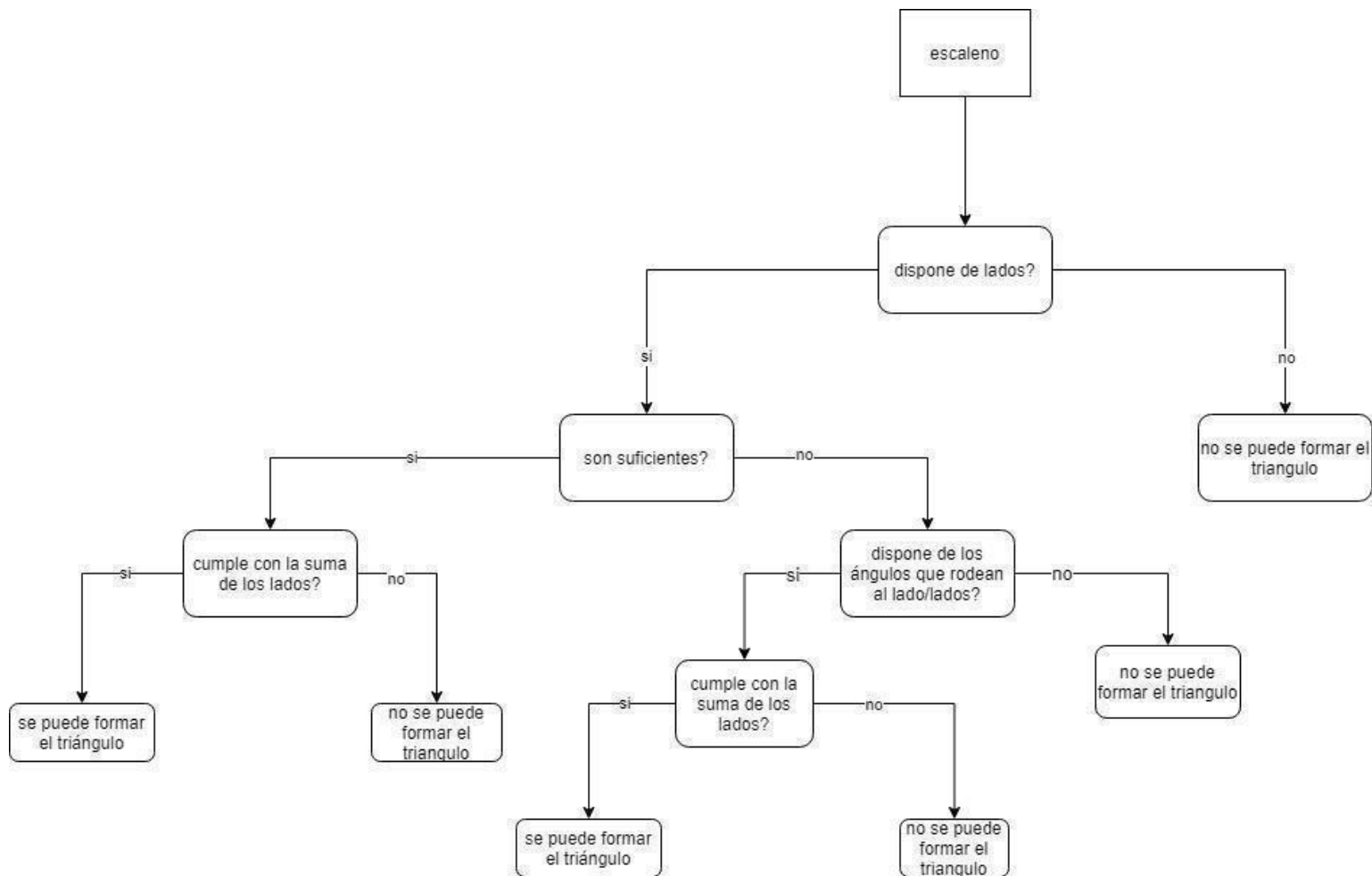
*Figura 1: Diagrama de flujo, datos del programa.*



*Figura 2: Diagrama de flujo de datos de equilátero*



*Figura 3: Diagrama de flujo de datos de isósceles.*



*Figura 4: Diagrama de flujo de datos de escaleno.*

## 4 Resultados.

### 4.1 Descripción de funcionamiento.

Para una mejor comprensión del funcionamiento la explicaremos dando ejemplos de cómo respondería el programa al querer visualizar un triángulo escaleno el cual sabemos las medidas de los lados y una pirámide triangular.

Para el caso del triángulo, se nos preguntará que tipo de triángulo queremos visualizar a lo que responderemos 'escaleno'.

Posteriormente tendremos que responder si poseemos el largo de los lados del triángulo y las medidas de estos.

Para estas preguntas responderemos 'si' y luego en tres variables distintas se guardarán las medidas que ingresemos por pantalla para cada uno de los lados, estos podrán ser tanto enteros como decimales.

En este punto el sistema ya estará listo para poder graficar la figura.

Para el caso de la pirámide, se podrá visualizar y girar a través de sus ejes en la ventana del programa funcionando.

En este punto el sistema ya estará listo para poder graficar las figuras.

### 4.2 Resultados obtenidos, capturas y conclusión.

Tras haber terminado el programa, los resultados fueron satisfactorios, puesto que se cumplen todas las restricciones y condicionales de manera correcta, almacena las variables que se estuvieron pidiendo por pantalla, funcionan a la perfección las funciones del seno/coseno y se lograron los principales objetivos del hito, los cuales eran la visualización de las figuras 2D y el código para posteriormente poder plasmar las figuras 3D.

A continuación, podemos observar las capturas del programa mostrando su correcto funcionamiento.

```
Terminal

||| ;Bienvenido al visualizador de triangulos 2D y piramides 3D! |||

Introduzca el tipo de triangulo:
(1.- Equilatero / 2.- Isosceles / 3.- Escaleno): 1

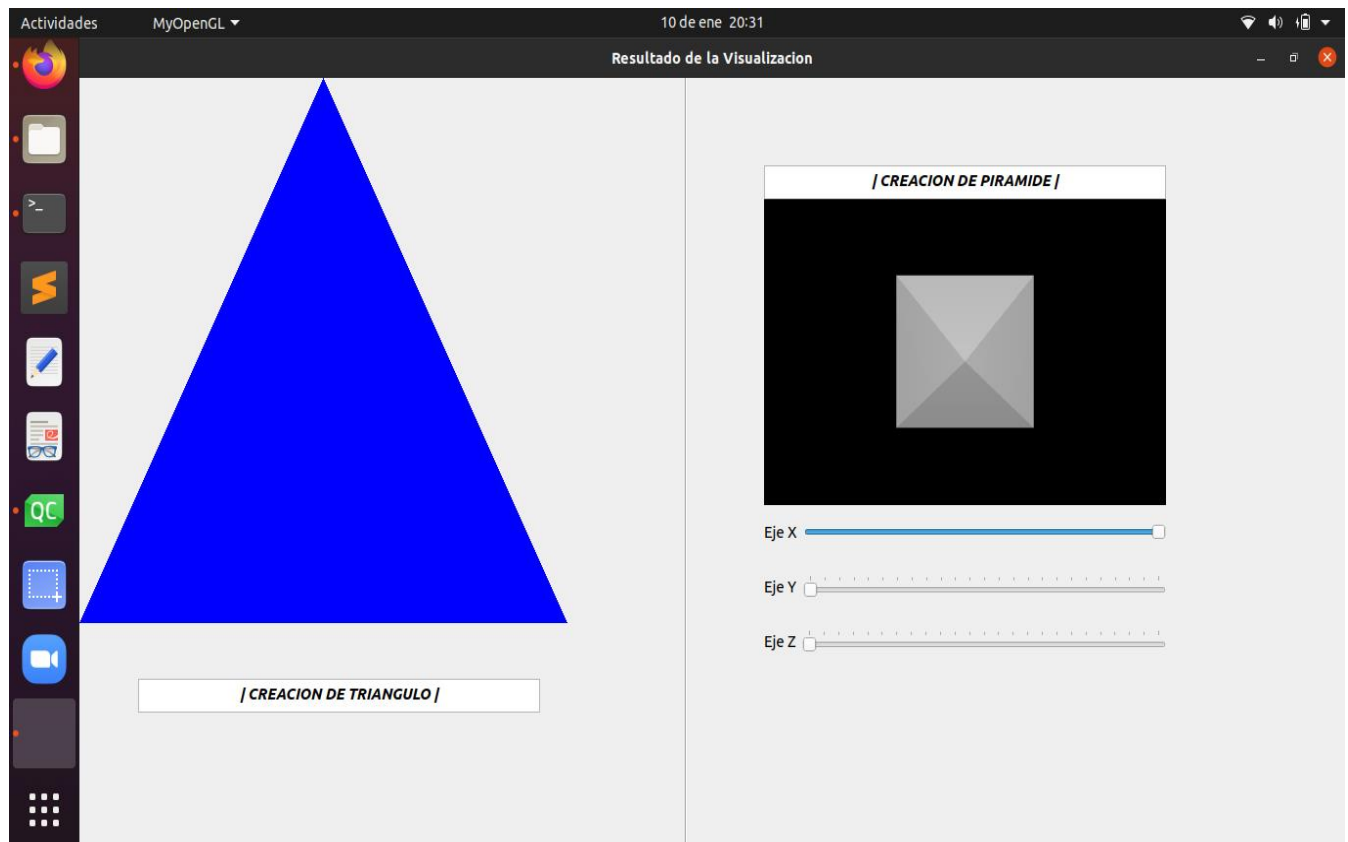
¿Dispone de la medida del lado del triangulo? (1-Si/2-No): 1

Introduzca la medida de los lados del triangulo: 499

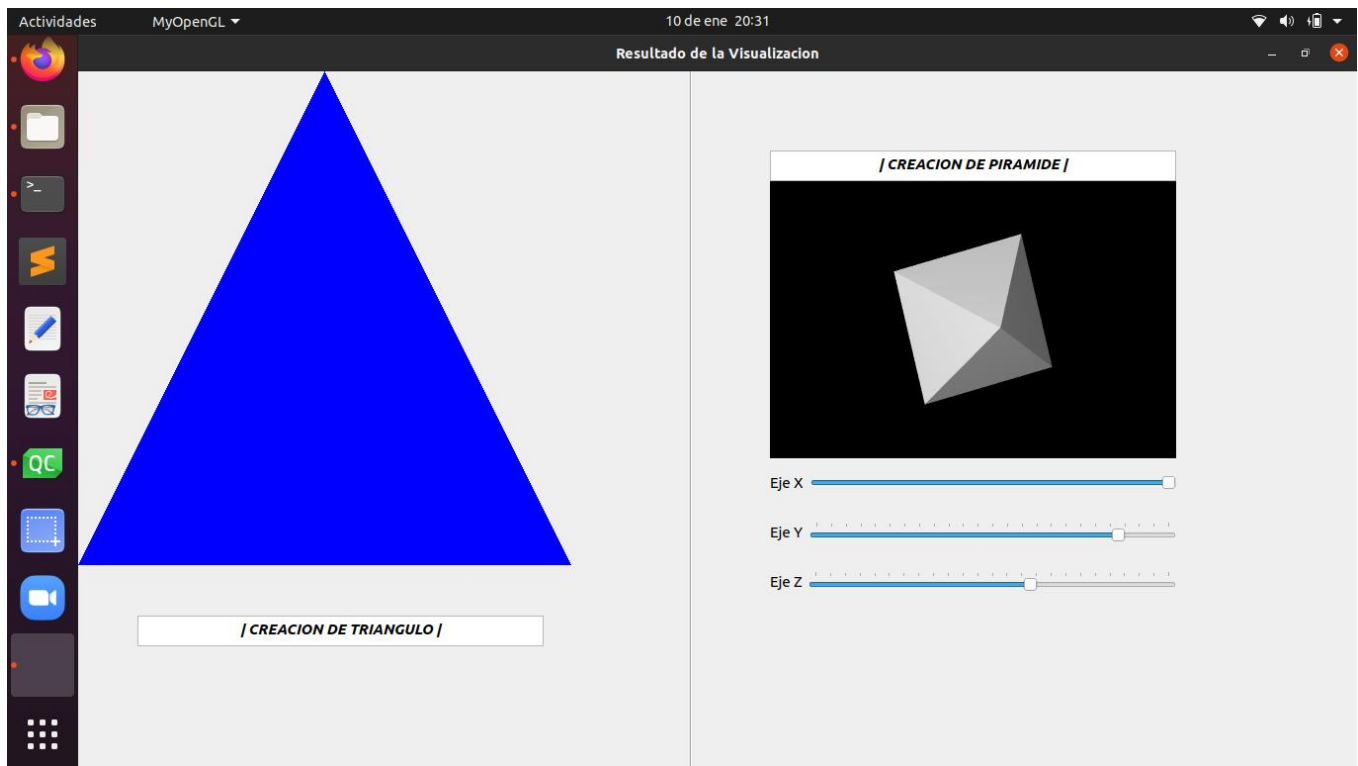
Las medidas de los lados del triangulo a graficar son:
Lado 1: 499
Lado 2: 499
Lado 3: 499

Creando triangulo...
Creando piramide...
```

*Imagen 1: Prueba de programa N° 1.*



*Imagen 2: Prueba de programa N° 2.*



*Imagen 3: Prueba de programa N°3.*

Como compañeros y equipo de trabajo damos como conclusión que este proyecto nos hizo ver De una manera mucho más ‘tangible’ la dificultad y compromiso necesario que requiere la creación de Cualquier tipo de sistema. Nos aporta una gran motivación para seguir estudiando e investigando Sobre programación ya que, sabemos que recién estamos empezando nuestro camino como ingenieros Y que la información que controlamos es mínima. Aprendimos sobre el uso de herramientas muy actuales como lo son GitLab, Qt Creator u OpenGL.



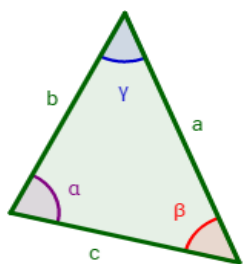
## 5 Anexos

### 5.1 Marco teórico.

Para confeccionar un triángulo es necesario considerar algunas restricciones, en caso de que se quiera formar un triángulo escaleno o isósceles, tenemos que la suma de sus lados siempre tiene que ser mayor al tercer lado, y a su vez, la resta en valor absoluto de dos lados tiene que ser menor al tercer lado.

Teorema del seno y coseno:

El teorema del coseno son resultados que establecen las relaciones entre los ángulos interiores de cualquier triángulo con el coseno de los lados opuestos a los ángulos, su aplicación permite conocer los ángulos o los lados del triángulo sin conocerlos todos:



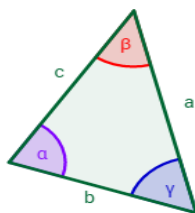
$$a^2 = b^2 + c^2 - 2 \cdot b \cdot c \cdot \cos(\alpha)$$

$$b^2 = a^2 + c^2 - 2 \cdot a \cdot c \cdot \cos(\beta)$$

$$c^2 = a^2 + b^2 - 2 \cdot a \cdot b \cdot \cos(\gamma)$$

El teorema del seno (o teorema de los senos) es un resultado de trigonometría que establece la relación de proporcionalidad existente entre las longitudes de lados de un triángulo cualquiera con los senos de sus ángulos interiores opuestos. Esta relación fue descubierta en el siglo X

Sea un triángulo cualquiera con lados a, b y c y con ángulos interiores α, β y γ (son los ángulos opuestos a los lados, respectivamente):



$$\frac{a}{\sin(\alpha)} = \frac{b}{\sin(\beta)} = \frac{c}{\sin(\gamma)}$$

Una pirámide, es un poliedro constituido por un polígono simple llamado base (en nuestro caso Cuadrada o triangular equilátera) y triángulos que tienen un único lado que coincide con uno del polígono base. Para poder formar esta figura con una base triangular, es necesario respetar todas las restricciones recién vistas sobre la formación de triángulos y para el caso de una pirámide de base cuadrada no existirá ninguna restricción basal. Además, para cualquiera de estos casos la altura de la pirámide no tendrá ninguna limitación.

## 6 Referencias.

[1] Restricción de creación de triángulos:

*sistemas.fciencias.unam.mx. (2018). Facultad de ciencias UNAM, pág 2[online]. disponible en:*

*[http://sistemas.fciencias.unam.mx/~erhc/geometria\\_analitica\\_1\\_2018\\_2/Desigualdad\\_triangular.pdf](http://sistemas.fciencias.unam.mx/~erhc/geometria_analitica_1_2018_2/Desigualdad_triangular.pdf) / [accedido 08 dic. 2020]*

[2] Teorema del seno y coseno:

*geogebra.org. (2017). Teorema del seno y del coseno [online]. Disponible en:*

*<https://www.geogebra.org/m/NrJ3UDKf> / [accedido 08 dic. 2020]*

[3] Integración de OpenGL en QT:

*Universidad Carlos III de Madrid. Departamento de ING. SISTEMAS Y AUTOMÁTICA (2011). Integración de OpenGL en QT [online]. Disponible en: <https://core.ac.uk/download/pdf/30045438.pdf> [accedido 08 dic. 2020]*