

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

Sistema para la digitalización tridimensional de objetos y generación de  
modelos virtuales

**PROYECTO INTEGRADOR**

Previo la obtención del Título de:

**Ingeniero en Computación**

Presentado por:

Steven Sebastian Araujo Moran

Kenny Ariel Camba Torres

GUAYAQUIL - ECUADOR

Año: 2021

## DEDICATORIA

Le dedico este proyecto a las personas que creyeron en mí y me apoyaron en todo, al centro de investigación CIDIS con quienes incluso nació la idea del proyecto y me permitieron ampliar mis conocimientos.

**Steven Araujo**

## **DEDICATORIA**

Le dedico este trabajo a mi familia y amigos, también a todos los profesores que me acompañaron a lo largo de la carrera, al club Célula Microsoft ESPOL y a la institución que me permitió iniciar mis sueños, la ESPOL.

**Kenny Camba**

## **AGRADECIMIENTOS**

Mis más sinceros agradecimientos a mi mamá, a mi papá, a mi hermano, mi abuelito y toda mi familia que siempre me apoyaron y confiaron en mí, a mis amigos con quienes compartí y nos apoyamos mutuamente para cumplir con nuestros objetivos y a los excelentes profesores que he tenido y conocido los cuales fueron una excelente guía en este proceso.

**Steven Araujo**

## **AGRADECIMIENTOS**

Agradezco a Dios por haber estado presente desde el principio cuando parecía imposible que obtuviera un cupo para el preuniversitario en ESPOL y por ser mi principal apoyo en mi vida.

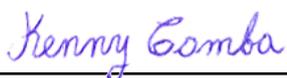
Agradezco a mis padres y hermano por siempre estar presentes y brindarme sus consejos cada vez que los necesite, en general agradezco a toda mi familia por creer en mí.

Agradezco muy especialmente a mis amigos, a quienes conocí en todo mi ciclo como estudiante y quienes se convirtieron en mi segunda familia, con ellos aprendí a superar mis miedos, a compartir, a ser yo mismo, porque gracias a ellos por primera vez en mi vida pude estar seguro de quién era y a dónde quería llegar, GDAG y Célula siempre los llevaré en mi corazón.

**Kenny Camba**

## DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Kenny Ariel Camba Torres* y *Steven Sebastian Araujo Moran*, damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



Kenny Ariel Camba  
Torres



Steven Sebastian Araujo  
Moran

# EVALUADORES

.....  
**Phd. Boris Vintimilla**

PROFESOR DE LA MATERIA

.....  
**Phd. Boris Vintimilla**

PROFESOR TUTOR

## RESUMEN

Varias industrias han incorporado tecnologías que utilizan datos 3D, como una de sus principales fuentes de producción. El uso de herramientas capaces de digitalizar objetos para su representación en entornos virtuales es de gran utilidad para industrias como videojuegos, diseño de interiores, robótica, quienes tienen en común el uso de modelos 3D, ya sea para crear un escenario, replicar o imprimir un componente.

El desarrollo del proyecto siguió la metodología ágil Scrum y utilizó herramientas de hardware para adquisición de datos tridimensionales como la cámara RealSense D435i, la lógica del software fue escrita en el lenguaje Python y además se utilizó librerías para acceder a la cámara (realsense2), manejar datos 3D (Open3D) y crear interfaces gráficas (Tkinter y Pyglet). Los recursos utilizados y el costo humano dieron un total estimado para el desarrollo de \$2739.

El software desarrollado cumplió con las evaluaciones, permitiendo determinar cuáles son las mejores condiciones y procedimientos a los que un objeto se debe someter para poder ser modelado con éxito. Además, al ser exportados los modelos en formatos comerciales, se comprobó que estos pueden ser utilizados en otros softwares disponibles en el mercado. El cliente del proyecto quedó satisfecho por cumplir con los requerimientos establecidos.

El proyecto es de gran utilidad para las industrias que utilizan información 3D. El hardware y las herramientas de software utilizados, en conjunto con los documentos técnicos y evaluaciones realizadas permiten tener un sistema completo para digitalizar objetos y que además es escalable y adaptable a otros campos de aplicación.

**Palabras Clave:** Adquisición de datos 3D, modelamiento, modelo 3D, software, RealSense D435i.

## **ABSTRACT**

*Several industries have incorporated technologies that use 3D data as one of their main sources of production. The use of tools capable of digitizing objects for their representation in virtual environments is very useful for industries such as video games, interior design, robotics, which have in common the use of 3D models, whether to create a scenario, replicate or print a component.*

*The development of the project implemented the agile Scrum methodology and used hardware tools for three-dimensional data acquisition such as the RealSense D435i camera, the software logic was written in Python language and also libraries were used to access the camera (realsense2), manage data 3D (Open3D) and create graphical interfaces (Tkinter and Pyglet). The resources used and the human cost gave an estimated total for the development of \$2739.*

*The developed software complied with the evaluations carried out, allowing to determine the best conditions and procedures to which an object must be subjected in order to be modeled successfully. In addition, when the models are exported in commercial formats, it was found that they can be used in other software available on the market. The client of the project was satisfied to comply with the established requirements.*

*The project is very useful for industries that use 3D information. The hardware and software tools used, together with the technical documents and evaluations carried out, allow us to have a complete system for digitizing objects that is also scalable and adaptable to other fields of application.*

*Keywords: 3D data acquisition, modeling, 3D model, software, RealSense D435i.*

# ÍNDICE GENERAL

EVALUADORES	7
RESUMEN	I
<i>ABSTRACT</i>	II
ÍNDICE GENERAL	III
ABREVIATURAS	VI
SIMBOLOGÍA	VII
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	XI
CAPÍTULO 1	1
1. Introducción	1
1.1 Descripción del problema	2
1.2 Justificación del problema	2
1.3 Objetivos	3
1.3.1 Objetivo general	3
1.3.2 Objetivos específicos	3
1.4 Módulos del proyecto	3
1.4.1 Adquisición de datos 3D	3
1.4.2 Generación de modelos virtuales tridimensionales	4
1.4.3 Visualización de modelos virtuales	4
1.4.4 Exportación a formatos comerciales	5
1.5 Marco teórico	5
1.5.1 Adquisición de datos 3D	5
1.5.2 Modelado de datos 3D	6
1.5.3 Visualización de modelos 3D	7

1.5.4	Exportación a formato comerciales	8
1.5.5	Soluciones existentes	9
CAPÍTULO 2		12
2.	Metodología	12
2.1	Análisis de los datos	12
2.2	Confiabilidad y compatibilidad de soluciones existentes	15
2.3	Alcance del proyecto	18
2.3.1	Requerimientos	18
2.3.2	Recursos a utilizar	19
2.3.3	Diseño conceptual	20
2.3.4	Bocetos y flujos de pantalla	24
2.4	Plan de implementación del proyecto	30
CAPÍTULO 3		32
3.	Resultados y análisis	32
3.1	Prototipo	32
3.1.1	Módulo de adquisición de datos 3D	33
3.1.2	Módulo de modelamiento de datos 3D	36
3.1.3	Módulo de visualización de modelos virtuales	38
3.1.4	Módulo de exportación de modelos virtuales	42
3.2	Evaluaciones	43
3.3	Análisis de costo	53
CAPÍTULO 4		54
4.	Conclusiones y recomendaciones	54
4.1	Conclusiones	54
4.2	Recomendaciones	55

4.3	Futuros trabajos	56
4.3.1	Adquisición con diferentes cámaras tridimensionales	56
4.3.2	ROI en adquisición	57
4.3.3	Modificación de la malla triangular para eliminar ruido	58
4.3.4	Capturas frames a frames del video streaming	59
4.3.5	Extrapolable a escenarios	59
	BIBLIOGRAFÍA	61
	APÉNDICES	64

## ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
3DF	3D Data Fusion
RF	Requerimientos funcionales
RNF	Requerimientos no funcionales
ROI	<i>Region of interest</i>
CIDIS	Centro de investigación, desarrollo e innovación de sistemas computacionales

## SIMBOLOGÍA

m	Metro
cm	Centímetro
mm	Milímetro

## ÍNDICE DE FIGURAS

Figura 1.1 Esquema general del problema.	2
Figura 1.2 Módulo de adquisición de datos 3D.	4
Figura 1.3 Módulo de generación de modelos virtuales tridimensionales.	4
Figura 1.4 Módulo de visualización de modelos virtuales.	5
Figura 1.5 Módulo de exportación a formatos comerciales.	5
Figura 2.1 Captura de <i>PointCloud with OpenCV</i> desde una perspectiva lateral.	13
Figura 2.2 Captura de <i>PointCloud with OpenCV</i> desde una perspectiva frontal.	13
Figura 2.3 Captura de <i>Align &amp; Background Removal</i> .	13
Figura 2.4 Resultado PLY de <i>RGBD-Integration-2020</i> con 133 frames.	14
Figura 2.5 Resultado PLY de <i>RGBD-Integration-2020</i> con 14 frames.	14
Figura 2.6 Fusión de datos 3D con el proyecto <i>RGBD-Integration-2020</i> [19].	16
Figura 2.7 Diagrama de componentes del <i>Backend</i> del proyecto.	21
Figura 2.8 Diagrama de componentes del <i>Frontend</i> del proyecto.	21
Figura 2.9 Diagrama de actividades del módulo de adquisición de datos 3D.	22
Figura 2.10 Diagrama de actividades del módulo de generación de modelos virtuales tridimensionales.	23
Figura 2.11 Diagrama de actividades de los módulos de visualización de modelos virtuales y de exportación a formatos comerciales.	24
Figura 2.12 Pantalla principal.	25
Figura 2.13 Configuración de cámara.	25
Figura 2.14 Visualización streaming.	26
Figura 2.15 Configuración para modelado.	27
Figura 2.16 Visualización de modelado.	28
Figura 2.17 Opciones de visualización.	28
Figura 2.18 Edición de modelado.	29
Figura 2.19 Exportación de modelado.	29
Figura 3.1 Ingreso al sistema.	32
Figura 3.2 Resoluciones y <i>frame rates</i> de la cámara Intel RealSense D435i.	33

Figura 3.3 Configuración de la cámara y captura de imágenes.	33
Figura 3.4 Directorio con capturas de un objeto.	34
Figura 3.5 Visualización principal en interfaz de adquisición.	35
Figura 3.6 Interfaz de adquisición con el color 2D y 3D desactivado.	35
Figura 3.7 Interfaz de adquisición con densidad de puntos en L3.	36
Figura 3.8 Selección de imágenes a modelar.	37
Figura 3.9 Procesando modelamiento.	37
Figura 3.10 Modelamiento finalizado.	38
Figura 3.11 Selección de archivo PLY.	38
Figura 3.12 Visualización del objeto.	39
Figura 3.13 Visualización de opciones.	40
Figura 3.14 Objeto rotado y trasladado.	40
Figura 3.15 Ventana para ingresar cantidad de triángulos.	41
Figura 3.16 Objeto nuevamente renderizado.	41
Figura 3.17 Elección de formato.	42
Figura 3.18 Formatos de exportación permitidos.	42
Figura 3.19 Escoger ubicación de archivo a exportar.	43
Figura 3.20 Modelo obtenido con pérdida de información debido a problemas de iluminación.	44
Figura 3.21 Ejemplo del movimiento entre un par de frames.	44
Figura 3.22 Histograma de tiempo vs cantidad de frames con resolución 1280x720.	45
Figura 3.23 Histograma de tiempo vs cantidad de frames con resolución 1280x720.	45
Figura 3.24 Relación entre la cantidad de triángulos en el modelo final y el número de frames a una resolución de 1280x720.	48
Figura 3.25 Relación entre la cantidad de triángulos en el modelo final y el número de frames a una resolución de 640x480.	48
Figura 3.26 Modelos virtuales capturados a resoluciones de 1280x720, 640x480 y 424x240 respectivamente.	49
Figura 3.27 Comparación entre el promedio del tamaño de archivo de los frames de un mismo objeto capturado a diferentes resoluciones y su relación con la cantidad de triángulos en el modelo final.	50

Figura 3.28 Frames a diferentes resoluciones y tamaños de archivo.	51
Figura 3.29 Modelo exportado en diferentes formatos visualizado en softwares disponibles con acceso libre.	52
Figura 4.1 Resultados obtenidos a partir de la cámara Lidar 515.	57
Figura 4.2 Selección de ROI en adquisición.	57
Figura 4.3 Selección de ROI en adquisición 2.	58
Figura 4.4 Selección de ROI en visualización.	58
Figura 4.5 Opción de grabar video.	59
Figura 4.6 Modelamiento con imperfecciones.	60
Figura 4.7 Modelamiento sin imperfecciones.	60

## ÍNDICE DE TABLAS

Tabla 1.1 Distancia mínima por resolución [5].	6
Tabla 1.2 Componentes de Tkinter [13].	7
Tabla 2.1 Comparación proyectos de adquisición.	16
Tabla 2.2 Comparación de proyectos de modelado 3D.	16
Tabla 2.3 Comparación de proyectos de visualización de modelos 3D.	17
Tabla 2.4 Comparación de proyectos de exportación de modelos 3D.	17
Tabla 2.5 Requerimientos funcionales.	18
Tabla 2.6 Requerimientos no funcionales.	19
Tabla 2.7 Plan de implementación.	30
Tabla 3.1 Opciones de visualización.	39
Tabla 3.2 Condiciones y configuración.	43
Tabla 3.3 Triángulos generados por modelos.	46

# CAPÍTULO 1

## 1. INTRODUCCIÓN

Las industrias a nivel mundial continúan en la búsqueda por mejorar su producción y atención al cliente a través del uso de la tecnología. La virtualización tridimensional de objetos es una de las principales tecnologías con variadas aplicaciones, desde impresiones en 3D para la producción de componentes hasta la visualización de escenas u objetos en diferentes perspectivas. Según el informe *Global 3D Printing Report 2019* [1] en el 2019 se ha dado un incremento de un 65% en el uso de impresiones en 3D con respecto al año 2016.

La adquisición de datos tridimensionales se realiza por medio de hardware especializado para este fin, la empresa de tecnología Intel pone a disposición la cámara RealSense D435i con sensores que son capaces de detectar la rotación de un objeto y obtener información de su profundidad [2]. La fusión de múltiples vistas 3D de un objeto permite modelar su geometría y obtener una versión virtual del mismo, conservando sus características y propiedades medibles. Existen algoritmos basados en principios matemáticos y en inteligencia artificial que ayudan en la tarea de fusionar vistas tridimensionales, los cuales han sido implementados en lenguajes de programación como Python y C++, a través de librerías como Open3D, que además exponen funcionalidades que ayudan al desarrollo de software que hacen uso de datos 3D [3].

La comercialización o distribución de modelos virtuales tridimensionales se realiza por medio de formatos comunes (PLY, STL, OBJ, DAE, 3MF, OFF) que son reconocidos por aplicaciones de visualización, procesamiento o impresión en 3D [4].

Este capítulo tiene como objetivo describir el problema a resolver y como las necesidades de las industrias contribuyen a su justificación, además se presentan los objetivos del proyecto tanto general como específicos, que a su vez constituyen los módulos a tratar a lo largo de este documento. Finalmente se realiza una revisión de la literatura para mostrar a modo de marco teórico los principales conceptos y definiciones que contextualizan el proyecto.

## 1.1 Descripción del problema

La adquisición en tiempo real de distintas vistas, mediante la cámara Intel RealSense D435i, permite obtener información parcial de un objeto o escena, requiriendo así de múltiples vistas (bajo un mismo sistema de referencia) para modelar su geometría (Figura 1.1). Existen diferentes metodologías para la fusión de múltiples vistas en 3D, las cuales permiten modelar un objeto del mundo real (Figura 1.1), para su posterior visualización y análisis dentro de un entorno virtual, por medio de la extracción de diferentes características (Figura 1.1). El presente proyecto propone usar estas metodologías para construir modelos tridimensionales virtuales que permitan obtener diferentes perspectivas y métricas de un objeto de tal forma que ayude, por medio de la exportación a formatos comerciales (como: PLY, OBJ, STL, OFF y GLTF) (Figura 1.1) a automatizar procesos dentro de industrias que involucran tomar medidas, realizar impresiones en 3D o visualizaciones virtuales muy cercanas a la realidad. Que además involucren aplicaciones como: cumplir con estándares de calidad en el mercado, creación de componentes físicos para ensamblaje o maquetación, interacción directa de usuarios con entornos virtuales como videojuegos o previsualización de objetos o escenarios.

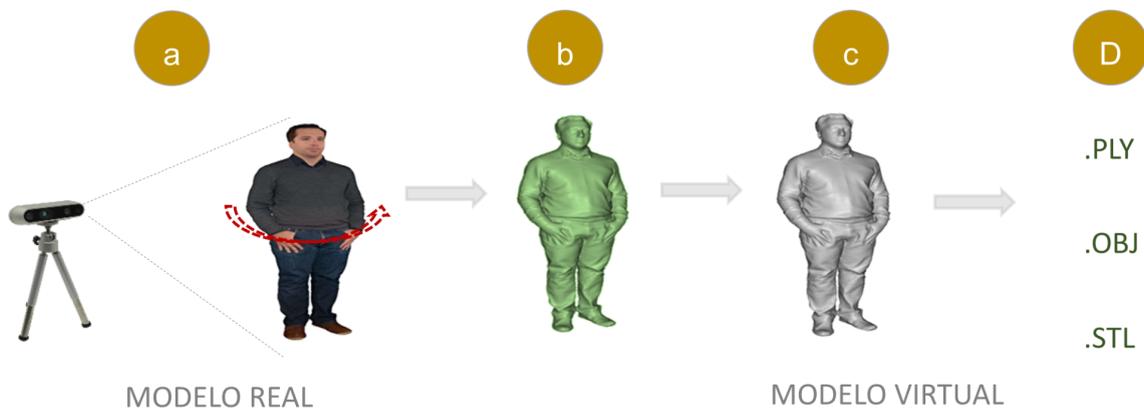


Figura 1.1 Esquema general del problema.

## 1.2 Justificación del problema

La generación de modelos virtuales tridimensionales es un proceso que comienza con la adquisición de datos 3D, continúa con la fusión de múltiples vistas 3D obtenidas a partir de objetos reales y finaliza con la visualización y exportación de los modelos generados. Industrias como la automotriz, robótica, metalmeccánica

utilizan este tipo de tecnologías para modelar sus artículos y mejorar la experiencia de sus clientes.

En la industria automotriz y en la robótica o automatización industrial la impresión en 3D ayuda a producir componentes que pueden ser comercializados como repuestos o utilizados para hacer prototipados rápidos. Industrias como metalmecánica y diseño de interiores, ofrecen a sus clientes la posibilidad de visualizar de manera virtual sus productos brindando una experiencia cercana a la realidad. En la industria de los videojuegos, el modelado 3D permite construir objetos virtuales que pueden formar parte de escenarios o interactuar con los personajes. En la agricultura industrial automatiza el análisis de calidad de los cultivos para el cumplimiento de estándares de exportación.

### **1.3 Objetivos**

#### **1.3.1 Objetivo general**

Desarrollar un software para la generación de modelos virtuales tridimensionales de objetos y sus correspondientes representaciones en formatos comerciales.

#### **1.3.2 Objetivos específicos**

- a) Desarrollar un módulo para la adquisición de datos 3D.
- b) Desarrollar un módulo para la generación de modelos digitales tridimensionales de objetos a partir de los datos adquiridos.
- c) Implementar un visualizador para los modelos 3D generados.
- d) Exportar los modelos 3D en formatos comerciales.

### **1.4 Módulos del proyecto**

#### **1.4.1 Adquisición de datos 3D**

El módulo tiene como objetivo realizar la conexión y configuración de la cámara, capturar los datos y su respectiva visualización en tiempo real. De este modo, la entrada del módulo consiste en la captura de los datos generados por la cámara Intel RealSense D435i y la salida es el resultado de la adquisición o nube de puntos (Figura 1.2).



**Figura 1.2 Módulo de adquisición de datos 3D.**

#### **1.4.2 Generación de modelos virtuales tridimensionales**

Este módulo se enfoca en analizar y fusionar los datos adquiridos, procesando y modelando objetos. Con las nubes de puntos obtenidas en el módulo de adquisición a partir de las diferentes vistas del objeto, se procede al modelado tridimensional del mismo para obtener como resultado la fusión de la información, que además son la entrada a los módulos de visualización de modelos virtuales y exportación a formatos comerciales (Figura 1.3).



**Figura 1.3 Módulo de generación de modelos virtuales tridimensionales.**

#### **1.4.3 Visualización de modelos virtuales**

Este módulo consiste en la implementación de las pantallas necesarias para la visualización y la manipulación tanto en traslación, como en rotación del modelo virtual 3D, obtenido como resultado de la fusión. Para lo cual es necesario recibir como entrada el objeto debidamente modelado (Figura 1.4).



**Figura 1.4 Módulo de visualización de modelos virtuales.**

#### 1.4.4 Exportación a formatos comerciales

Este módulo trata sobre la exportación de modelos tridimensionales a distintos formatos de uso comercial como, por ejemplo, PLY, OBJ, STL, OFF y GLTF. Requiriendo como entrada el objeto debidamente modelado, el cual es resultado del módulo de generación de modelos virtuales tridimensionales (Figura 1.5).



**Figura 1.5 Módulo de exportación a formatos comerciales.**

### 1.5 Marco teórico

En esta sección se exponen los conceptos y definiciones obtenidos a partir de la revisión literaria correspondiente a cada uno de los módulos expuestos, así como también sus soluciones existentes en base a la exploración de sus antecedentes.

#### 1.5.1 Adquisición de datos 3D

Un componente de hardware importante para la adquisición de datos 3D son las cámaras tridimensionales como por ejemplo la Intel RealSense D435i. Entre las especificaciones de esta cámara se tienen un tamaño de 90 mm x 25 mm x 25 mm, con un sensor RGB de 2 MP de 1920 × 1080, con tecnología de profundidad estereoscópica que le permite capturar dos imágenes tridimensionales diferentes y generar una de salida, además cuenta con un alcance mínimo dependiente de la resolución como se muestra en la Tabla 1.1.

**Tabla 1.1 Distancia mínima por resolución [5].**

Resolución	Distancia de profundidad mínima(m)
1280x720	0.280
848x480	0.195
640x480	0.175
640x360	0.150
480x270	0.120
424x240	0.105

En la adquisición de datos 3D Intel RealSense posee una documentación amplia y un SDK gratuito que soporta los lenguajes C, C++, Python, Node.js, entre otros. Presenta la ventaja de permitir cambiar entre dispositivos persistiendo la facilidad de desarrollo [6].

En el caso de Python, el lenguaje cuenta con una librería para el desarrollo correspondiente a la cámara Intel RealSense llamada pyrealsense2. Esta librería tiene una documentación amplia sobre las configuraciones necesarias para ejecutar la cámara y realizar la adquisición de frames en diferentes modos tanto 2D como 3D (depth).

El resultado de la adquisición 3D es una nube de puntos, la cual es una forma de representar la información tridimensional como un conjunto de puntos en el espacio, para la comprensión de los modelos virtuales [7]. Esta forma de representación es muy usada para mediciones y las inserciones en softwares gracias a su precisión milimétrica. En este caso, la nube de puntos corresponde a 2 matrices: una de vectores en el plano (x,y,z) y una de proyección (u,v), que permite proyectar la matriz de colores RGB al vector correspondiente en el plano (x,y,z) [8]. Toda esta información es dependiente de las propiedades intrínsecas de la cámara, las cuales son obtenidas mediante la librería pyrealsense2 [9].

### **1.5.2 Modelado de datos 3D**

En la fusión de datos 3D la información se encuentra distribuida en varias imágenes, no hay perfección en la información al combinarla, es posible

encontrar ambigüedad, incertidumbre o incompletitud. La fusión tiene como objetivo tratar todos estos tipos de defectos, considerando principalmente los factores de redundancia parcial de datos y complementariedad entre datos [10]. Existen librerías de código abierto tales como Open3D que permiten el desarrollo de software que hace uso de datos tridimensionales. La biblioteca, en la parte del backend, se encuentra desarrollada en el lenguaje C++, manteniendo un alto nivel de optimización y admite paralelización. En su *frontend* Open3D proporciona estructuras de datos para representaciones de nubes de puntos, mallas triangulares e imágenes RGB-D, además posee algoritmos para muestreo, visualización y conversión de datos, estando disponible para los lenguajes de programación C++ y Python [3].

Open3D proporciona la estructura de mallas triangulares para el modelado de objetos. Las mallas triangulares son conjuntos de múltiples triángulos, representados por los índices de sus vértices, que permiten aproximar la superficie de un objeto o escena [11].

### 1.5.3 Visualización de modelos 3D

Para la visualización de los modelos tridimensionales Python incluye librerías que permiten el desarrollo de interfaces gráficas de usuario (GUI), una de las más populares es Tkinter que cuenta con una documentación amplia y de gratis acceso. Este paquete permite visualizar en tiempo real el módulo de adquisición y el resultado del procesamiento de fusión obtenido con Open3D para manipularlo en rotación y traslación para su completa visualización [12]. La Tabla 1.2 muestra los principales componentes de Tkinter.

**Tabla 1.2 Componentes de Tkinter [13].**

Componente	Descripción
Tk	Permite crear una ventana nueva, se puede asignar un nombre de título, tamaño.
Label	Agregar una etiqueta nueva, con una propiedad para asignar la ubicación en la ventana.
Button	Agregar un nuevo botón, con una propiedad para asignar la ubicación en la ventana, propiedad para el manejo de evento clic.

Frame	Permite ubicar los componentes en partes específicas, por ejemplo: izquierda, derecha y centro.
Grid	Permite ubicar los componentes a manera de grillas.
Entry	Componente para agregar un nuevo cuadro de texto, con propiedad para modificar el ancho y su ubicación en la ventana.
Combobox	Permite añadir el widget de combobox, el cual permite agregar valores a las opciones y su ubicación en la ventana.
Canvas	Permite agregar imágenes para visualizar y actualizar cada milisegundo a manera de video.

El método para la visualización de modelos virtuales se realiza por medio de frames, las visualizaciones siempre ocurren en 2D, mostrando así una cara del modelo virtual que puede estar afectada por un ángulo de inclinación. Adicionalmente es posible realizar rotaciones y traslaciones utilizando un sistema de coordenadas  $X, Y$ , realizando cálculos para derivar la posición y el ángulo de visión del modelo virtual con respecto al movimiento del mouse, del cual Tkinter ofrece acceso.

#### 1.5.4 Exportación a formatos comerciales

La representación de datos tridimensionales tiene varias aplicaciones en diferentes campos como: la medicina, ingeniería estructural, industria del automóvil, entre otros. Existen formatos estándares para la distribución de modelo 3D, entre ellos se tienen: PLY, STL, OBJ, DAE, 3MF [14].

- **PLY** (*Polygon File Format*): Formato de archivo que permite almacenar objetos gráficos descritos como una colección de polígonos, agregando otras propiedades como el color, la dirección normal de cada elemento, coordenadas de texturas, transparencia y otras propiedades vinculadas a un polígono [15].
- **STL** (*Stereo Lithography*): Formato para representación triangular de superficies tridimensionales. Se tiene una lista de triángulos conocidos como facetas y cada uno es identificado por una unidad normal y por tres vértices.

Cada triángulo o faceta debe compartir dos vértices con otros dos triángulos adyacentes [16].

- **OBJ** (*Object*): Formato de archivo que define a un objeto por su geometría y otras propiedades. Admiten objetos poligonales (puntos, líneas y caras) y objetos de forma libre (curvas y superficies) [15].
- **DAE**: Formato de intercambio de paquetes 3D basado en XML [15].
- **3MF** (*3D Manufacturing Format*): Formato para impresión 3D, basado en XML con la capacidad de almacenar un modelo 3D, conteniendo un nivel amplio de detalle [17].
- **OFF** (*Object File Format*): Formato de archivo para la representación de una colección de polígonos planos que posiblemente comparten vértices. Además, se permiten polígonos cóncavos [15].

### 1.5.5 Soluciones existentes

Existen proyectos que implementan soluciones de software para cada uno de los módulos descritos en el apartado Módulos del proyecto.

En el módulo de **adquisición de datos 3D** se han desarrollado proyectos como “*PointCloud with OpenCV*” [18] que incorpora un visualizador tridimensional por medio de la librería *OpenCV*. Este proyecto es parte de la sección de ejemplos proporcionados por el SDK de Intel RealSense y se trata de una transmisión en streaming de una nube de puntos obtenida a partir de la cámara, además se muestra cómo realizar rotaciones y traslaciones por medio de cálculos matemáticos junto con el poder de procesamiento de matrices de Numpy.

Otro proyecto vinculado a los ejemplos del SDK de Intel RealSense es el denominado “*Align & Background Removal*” [18] el cual permite capturar información tridimensional, considerando una profundidad máxima definida, para posteriormente realizar un filtrado de la nube de puntos resultante, de tal forma que se elimina información innecesaria y se enfoca en un objeto a una distancia definida.

En el módulo de **generación de modelos virtuales tridimensionales** o fusión de datos 3D, se encuentra el proyecto denominado “*RGBD-Integration-2020*” [19], el cual permite integrar fotogramas de color (RGB) y profundidad (Depth) obtenidos por medio de la cámara RealSense. Este proyecto utiliza la librería Open3D para leer un conjunto de imágenes 2D con información de color RGB y profundidad, para posteriormente fusionarlas, obtenido así una nube de puntos, finalmente se extrae una malla triangular del modelo generado, que es utilizada para visualización y generación de un archivo en formato PLY, que contiene al objeto tridimensional.

También existen proyectos de investigación como el denominado “Localización y Mapeo Simultáneo con Imágenes 3D a Color” [20], el cual utiliza fusión de datos tridimensionales, con la finalidad de solucionar el problema de localización y mapeo en el ambiente por donde se moviliza un robot. Para esto se utiliza el algoritmo ICP, que permite converger imágenes tridimensionales por medio de un proceso iterativo que toma en cuenta cada punto en una nube de puntos, de esta forma es posible construir un escenario para que el robot pueda realizar un mapeo del lugar y localizar su ubicación.

Existe otro proyecto como “*Monocular 3D human body reconstruction towards depth augmentation of television sequences*” [21] en el cual a partir de un video 2D se reconstruye el modelo tridimensional de un cuerpo, segmentándolo por medio de una imagen con un algoritmo basado en el diagrama de Voronoi, para así definir la postura del objetivo y por último proyectar a la construcción del modelo tridimensional, el cuerpo segmentado con mayor similitud que se presenta en el video. Este proyecto parte con la base de modelar cada una de las partes del cuerpo usando “*Superquadrics*” que sirve para componer objetos con diferentes formas, como esferas, cilindros, paralelepípedos, etc.

En el módulo de **visualización de modelos virtuales** se encuentra el proyecto “*3D-Rendering-Desktop-App*” [22], es un software de visualización de objetos 3D que utiliza librerías como Numpy y Tkinter, permitiendo únicamente abrir

archivos en formato OBJ. El proyecto utiliza Numpy para realizar cálculos sobre matrices para hacer transformaciones y obtener una matriz de rotación. Tkinter es utilizada para crear la interfaz de usuario, en donde se visualiza el objeto 3D y existen controles para Zoom, rotación en los ejes X, Y, Z y traslación en las direcciones arriba, abajo, izquierda, derecha.

El SDK de Intel RealSense también proporciona ejemplos para visualizar modelos virtuales 3D, uno de estos es “*PyGlet PointCloud Viewer*” [18]. Que utiliza la librería multimedia PyGlet para renderizar el modelo 3D y permite acceder a eventos como presionar teclas o al desplazamiento del mouse, para realizar la manipulación del objeto.

En el módulo de **exportación a formatos comerciales**, el artículo “Guía de 5 pasos para generar mallas tridimensionales utilizando nubes de puntos” [23] describe cómo utilizar la librería Numpy para cargar los datos tridimensionales (nube de puntos), para posteriormente convertir esta información en un objeto de Open3D, con el cual se pueda realizar el procesamiento con mallas triangulares. Finalmente se realiza la exportación de la malla por medio de la misma librería Open3D, especificando el formato como extensión en el nombre del archivo resultante.

En los ejemplos proporcionados por el SDK de Intel RealSense existe un proyecto conocido como “*Export Point Cloud to PLY*” [18] en el cual se proporciona un algoritmo para exportar los modelos en formato PLY directamente con la librería proporcionada por el SDK.

El siguiente capítulo define la metodología a seguir en la implementación de este proyecto y tiene como objetivo analizar los datos pertinentes basados en proyectos existentes. Además, se busca determinar la confiabilidad de los datos y cuáles son las mejores alternativas para desarrollar una propuesta de solución a los módulos expuestos en este capítulo. Esta propuesta representa el alcance del proyecto que abarca un diseño conceptual, requerimientos y recursos a utilizar, para finalmente elaborar un plan de implementación.

# CAPÍTULO 2

## 2. METODOLOGÍA

El presente capítulo describe la metodología que se utilizó para el desarrollo del proyecto, partiendo de las soluciones existentes tratadas en el capítulo anterior, para realizar un análisis con respecto a cómo estas soluciones utilizan los datos de entrada (información tridimensional) para producir salidas mediante un procesamiento interno. Posteriormente se realiza otro análisis para determinar la confiabilidad y cómo pueden ser adaptados estos proyectos existentes a la implementación del software desarrollado.

El alcance del proyecto busca determinar las funcionalidades finales que son requeridas por parte del cliente y duración necesaria para la implementación del proyecto, para esto se seleccionó Scrum como metodología de desarrollo de software, estableciendo así los *sprints* necesarios en relación con los requerimientos funcionales y no funcionales. Así mismo se determinaron los recursos que iban a ser utilizados para el desarrollo de cada módulo, se hizo un diseño conceptual para visualizar mediante diagramas UML la estructura y comportamiento del software y finalmente se crearon bocetos de las pantallas y el flujo que deben de seguir en la aplicación.

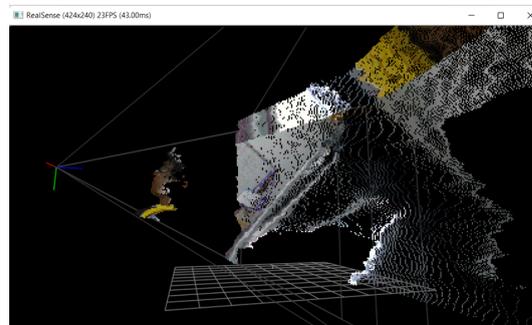
La última sección de este capítulo muestra el cronograma establecido por cada *sprint*, así como las fechas planificadas para la implementación del proyecto.

### 2.1 Análisis de los datos

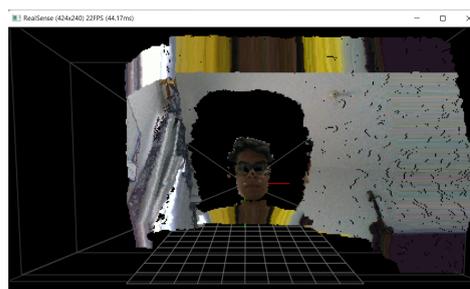
De acuerdo con los datos que se obtuvieron a partir de las Soluciones existentes con respecto al módulo de Adquisición de datos 3D, se observó que la nube de puntos presenta una proyección con respecto a la captura de imágenes en 2D. Esto demostró que, debido a la captura estereoscópica, no existe información uno a uno en cuanto a los puntos en el plano (X, Y, Z) y la matriz de píxeles RGB correspondiente a la imagen capturada, por lo cual la información de proyección (U, V) es la encargada de determinar a qué punto en el plano (X, Y, Z) pertenece la información de color.

La información obtenida desde la cámara es dependiente de las configuraciones establecidas, ya que la captura de datos 2D y 3D se realizan de manera aislada.

Con el proyecto de “*PointCloud with OpenCV*” [18] se puede apreciar que la información obtenida con distancias de profundidades altas es decir a partir de 2.5m posee un mayor grado de pérdida de información (Figura 2.1 y Figura 2.2).

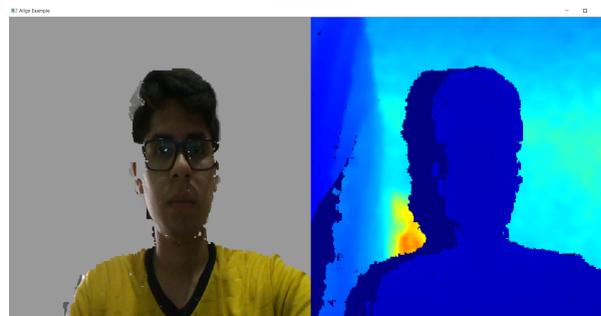


**Figura 2.1** Captura de *PointCloud with OpenCV* desde una perspectiva lateral.



**Figura 2.2** Captura de *PointCloud with OpenCV* desde una perspectiva frontal.

Con el proyecto “*Align & Background Removal*” [18] se realiza un enfoque de la información obtenida por medio de la cámara, mostrando solo la información del objeto a una distancia específica, eliminando así información considerada innecesaria. La distancia específica del siguiente ejemplo es de 1m (Figura 2.3).



**Figura 2.3** Captura de *Align & Background Removal*.

Para la **generación de los modelos virtuales tridimensionales** se usó tanto la información 2D como 3D, obtenida por la cámara Intel RealSense D435i. Para el modelamiento del objeto se necesita múltiples vistas desde diferentes ángulos, en “*RGBD-Integration-2020*” [19] usan 133 capturas diferentes para la construcción del modelo virtual de una persona, su salida está representada en el formato PLY.



**Figura 2.4 Resultado PLY de *RGBD-Integration-2020* con 133 frames.**

Su resultado es fiel a la realidad y se puede apreciar que las imperfecciones son mínimas en cuanto al resultado esperado demostrando así un acabado suave y fino (Figura 2.4).



**Figura 2.5 Resultado PLY de *RGBD-Integration-2020* con 14 frames.**

El mismo proyecto con una cantidad menor de frames se puede apreciar que el resultado posee una gran cantidad de imperfecciones demostrando así un acabado grueso (Figura 2.5) en comparación con el ejemplo anterior (Figura 2.4). Los diferentes resultados poseen ventajas de acuerdo con sus objetivos de uso. Cuando se registra menos información, se genera un acabado grueso que es usado en tareas que requieran menor tiempo de procesamiento o no necesiten un alto nivel de detalle como, por ejemplo, en la previsualización del resultado final en el caso del diseño de interiores que tiene como objetivo darle una idea al usuario de cómo se vería el interior a diseñar, más no mostrarle un nivel de detalle alto. A diferencia de otras aplicaciones como robótica, el desarrollo de ciudades

inteligentes, la estimación de medidas para cumplimiento de calidad o video juegos, en los cuales el nivel de detalle requerido es superior, buscando un acabado fino que requiere mayor tiempo de procesamiento [24].

Los proyectos “Localización y Mapeo Simultáneo con Imágenes 3D a Color” [20] y “*Monocular 3D human body reconstruction towards depth augmentation of television sequences*” [21] no están considerados dentro del análisis de esta sección debido a que son de carácter privado.

Los datos analizados proporcionaron información valiosa que permitieron tomar decisiones con respecto a los tipos de objetos que deben ser usados para el desarrollo del proyecto, es importante que estos sean rígidos, es decir que no sea flexible o que pueda deformarse, debido a que la carencia de esta propiedad puede causar ruido o pérdida de información. Por ejemplo, este proyecto no debería ser utilizado para digitalizar un ser humano.

## 2.2 Confiabilidad y compatibilidad de soluciones existentes

Las soluciones existentes poseen características que las hacen confiables para su uso y por lo tanto pueden ser compatibles con el presente proyecto. Esta sección realiza un análisis de cada una de estas soluciones para determinar cuáles de sus propiedades son aptas para contribuir en el desarrollo del proyecto. El análisis es comparativo y se realiza por cada uno de los módulos establecidos.

Evaluando los resultados de soluciones existentes en el módulo de **adquisición de datos 3D** se observan resultados acordes a la realidad con una presencia de ruido mínima en cuanto a la información 3D, que varía de acuerdo con la distancia del objeto y se puede ver afectada por la intensidad de luz que presente la escena, es recomendable aplicar una distancia que se encuentra entre 0.3m y 2.5m.

La Tabla 2.1 demuestra que la mejor alternativa es realizar una eliminación del fondo de manera similar al proyecto “*Align & Background Removal*” [18]. Reducir la cantidad de información para disminuir el tiempo de procesamiento, mejorar la precisión y enfocarse en el objeto objetivo.

**Tabla 2.1 Comparación proyectos de adquisición.**

Proyecto	Cantidad de información	Presencia de ruido	Permite Enfoque en objeto
PointCloud with OpenCV	Alta	Sí	No
Align & Background Removal	Media	Sí	Sí

La Tabla 2.2 muestra una comparación entre tres proyectos que realizan **modelado 3D**, de aquí resaltan las características de la solución “*RGBD-Integration-2020*” [19], principalmente porque es de código abierto y además utiliza estructuras para el manejo de datos 3D (Nube de puntos y Mallas triangulares) que se encuentran en la librería Open3D. La Figura 2.6 muestra el resultado de la fusión de imágenes 2D para generar el modelo virtual de una persona.



**Figura 2.6 Fusión de datos 3D con el proyecto RGBD-Integration-2020 [19].**

**Tabla 2.2 Comparación de proyectos de modelado 3D.**

Proyecto	Fusión 3D	Fusión 2D	Representación	Mallas triangulares	Open Source
RGBD-Integration-2020	Sí	Sí	Nube de puntos	Sí	Sí
Localización y Mapeo de Simultáneo con imágenes 3D a Color	Sí	No	Nube de puntos	No	No

Monocular 3D human body reconstruction towards depth augmentation of television sequence	No	Sí	Algoritmo de Voronoi	Sí	No
--	----	----	----------------------	----	----

La Tabla 2.3 muestra una comparación de dos proyectos de **visualización de modelos 3D**, ambos son de código abierto y permiten interactuar con el modelo virtual por medio de zoom, rotaciones y traslaciones. Ambas soluciones proporcionan utilidades que son adaptables al presente proyecto: “3D-Rendering-Desktop-App” [22] utiliza la librería Tkinter para realizar la visualización, *PyGlet PointCloud Viewer* utiliza archivos PLY, que es el formato que proporciona la cámara Intel RealSense.

**Tabla 2.3 Comparación de proyectos de visualización de modelos 3D.**

Proyecto	Formatos admitidos	Librería gráfica	Controles admitidos	Tipo de control	Open Source
3D-Rendering-Desktop-App	OBJ	Tkinter	Zoom, Rotación, Desplazamiento	Botones en interfaz	Sí
PyGlet PointCloud Viewer	PLY	PyGlet	Zoom, Rotación, Desplazamiento	Botones del mouse	Sí

La Tabla 2.4 compara dos proyectos de **exportación de modelos 3D a formatos comerciales**. Ambos son de código abierto, el primero de ellos es de mayor utilidad, porque utiliza la librería Open3D y además proporciona más formatos de salida en comparación con el proyecto *Export Point Cloud to PLY*.

**Tabla 2.4 Comparación de proyectos de exportación de modelos 3D.**

Proyecto	Librería	Formatos de salida	Open Source
Guía de 5 pasos para generar mallas tridimensionales utilizando nubes de puntos	Open3D	PLY, STL, OBJ, OFF, GLTF	Sí
Export Point Cloud to PLY	pyrealsense2	PLY	Sí

## 2.3 Alcance del proyecto

Para el presente proyecto se opta por la utilización de una metodología de desarrollo de software ágil, seleccionando así la metodología SCRUM, por su versatilidad y por contar con un tiempo limitado para él desarrollo. El proyecto será llevado a cabo en 3 sprints, cada sprint tiene una duración de 3 semanas (de acuerdo con el Plan de implementación) y finalizan con una reunión con el cliente (Product Owner). El equipo de desarrollo mantendrá reuniones diarias con el fin de revisar los avances del día anterior y acordar los que se harán ese día.

### 2.3.1 Requerimientos

Las necesidades expuestas por el cliente por medio del CIDIS (Centro de Investigación, desarrollo e innovación de sistemas computacionales) son:

- a. Adquisición de datos tridimensionales por medio de la cámara Intel RealSense D435i.
- b. Modelar objetos de forma virtual y tridimensional, a partir de múltiples vistas 3D.
- c. Software para visualizar objetos tridimensionales y manipularlos a través de rotaciones y traslaciones.
- d. Exportar modelos tridimensionales a formatos comerciales.

Cabe mencionar que el módulo de modelamiento no es totalmente dependiente del módulo de adquisición es decir que es posible realiza el modelamiento del objeto a partir de las imágenes obtenidas en la etapa de adquisición o usar un archivo tridimensional PLY para su posterior visualización y exportación.

De acuerdo con estas necesidades, los requerimientos funcionales (Tabla 2.5) y no funcionales (Tabla 2.6) del software son:

**Tabla 2.5 Requerimientos funcionales.**

ID	Descripción
RF01	Integración y configuración de cámara Intel RealSense D435i
RF02	Adquisición de información mediante cámara Intel RealSense D435i
RF03	Visualización streaming de la cámara Intel RealSense D435i

RF04	Almacenamiento de la información obtenida (capturas desde diferentes vistas)
RF05	Carga de información obtenida (capturas realizadas) o de modelos ya creados
RF06	Procesamiento para el modelado del objeto
RF07	Visualización de resultado de modelamiento
RF08	Manipulación de resultado (rotación y traslación)
RF09	Edición de mallas triangulares de los modelos
RF10	Exportación de modelo a diferentes formatos

**Tabla 2.6 Requerimientos no funcionales.**

ID	Descripción
RNF01	Rendimiento: el software tiene que soportar el visualizar, capturar y procesar la información necesaria.
RNF 02	Usabilidad: el software tiene que ser fácil e intuitiva de usar.
RNF 03	Escalabilidad: el software debe desarrollarse para permitir incorporar otras cámaras tridimensionales a futuro.
RNF 04	Confiableabilidad: el software debe ser capaz de procesar toda la información
RNF 05	Mantenibilidad: el software debe ser fácil de mantener para realizar los cambios necesarios a futuro, como dar soporte a diferentes cámaras tridimensionales.

### 2.3.2 Recursos a utilizar

Los recursos a utilizar se dividen en hardware y software y son puestos a disposición por parte del centro de investigación CIDIS:

- Cámara Intel RealSense D435i
- Workstation para desarrollo del modelo
- Asesoramiento del grupo de visión por computadora

Cada módulo descrito en este proyecto necesita recursos tanto de software como de hardware, como se especifica a continuación:

#### **Módulo de adquisición**

- **Software:** Lenguaje de programación Python, SDK de Intel RealSense (librería pyrealsense2) para configuración y administración de la cámara Intel

RealSense D435i, OpenCv con operaciones bit a bit y segmentación de objeto, sección de código del proyecto “*Align & Background Removal*” [18] para eliminación del fondo de las imágenes adquiridas estableciendo un máximo de distancia profundidad.

- **Hardware:** Cámara Intel RealSense D435i, Computadora.

#### **Módulo de modelado de datos 3D**

- **Software:** Lenguaje de programación Python, librería de procesamiento de datos 3D Open3D. Sección de código para la fusión de imágenes del proyecto “*RGBD-Integration-2020*” [19].
- **Hardware:** Computadora.

#### **Módulo de visualización de modelos virtuales**

- **Software:** Lenguaje de programación Python, librería para creación de interfaces gráficas Tkinter. Inspiración para el diseño de la interfaz gráfica, tomando en cuenta el proyecto “*3D-Rendering-Desktop-App*” [22], librería para el desarrollo de aplicaciones multimedia PyGlet, sección de código del proyecto “*PyGlet PointCloud Viewer*” [18].
- **Hardware:** Computadora.

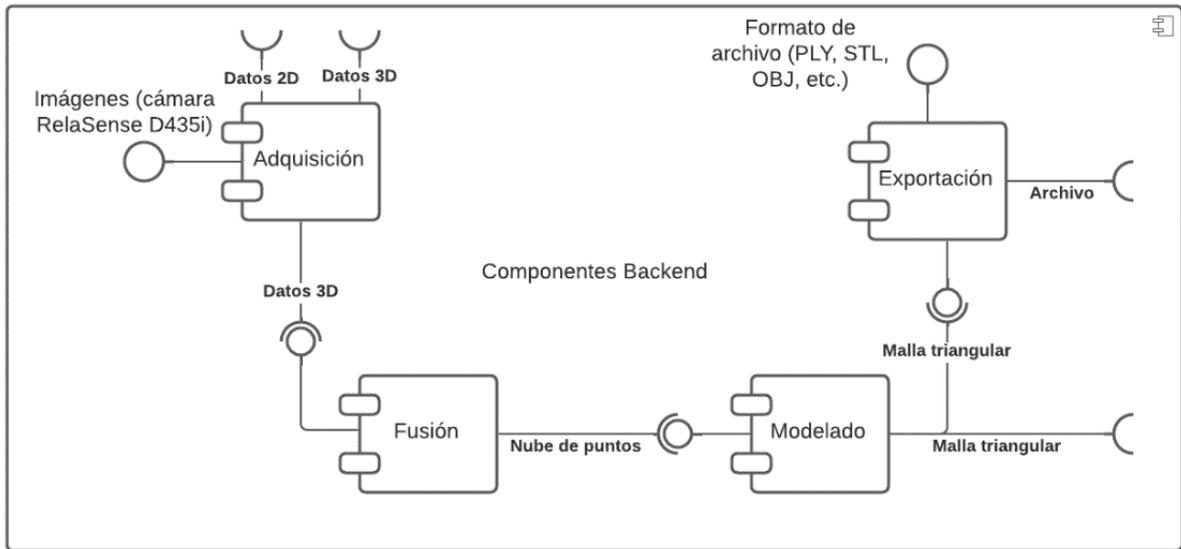
#### **Módulo de exportación a formatos comerciales**

- **Software:** Lenguaje de programación Python, librería de procesamiento de datos 3D Open3D. Sección de código para exportar modelos virtuales 3D proporcionado en la “Guía de 5 pasos para generar mallas tridimensionales utilizando nubes de puntos” [23].
- **Hardware:** Computadora.

### **2.3.3 Diseño conceptual**

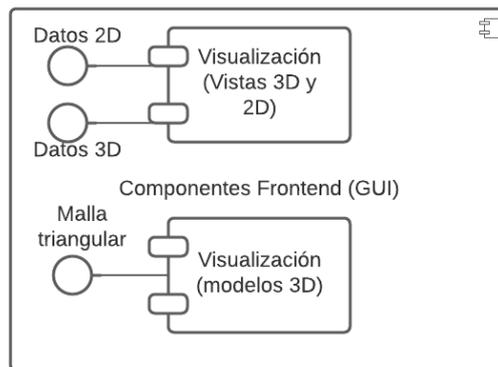
El proyecto está compuesto por una arquitectura que se compone de dos capas: *Backend* y *Frontend*. El *Backend* permite manejar toda la lógica interna del sistema, como lo es la comunicación con la cámara, el procesamiento y fusión de datos 3D, para obtener una representación de la información en mallas

triangulares, que es la estructura utilizada para visualización y exportación, como se muestra en el diagrama de componentes de la Figura 2.7.



**Figura 2.7 Diagrama de componentes del *Backend* del proyecto.**

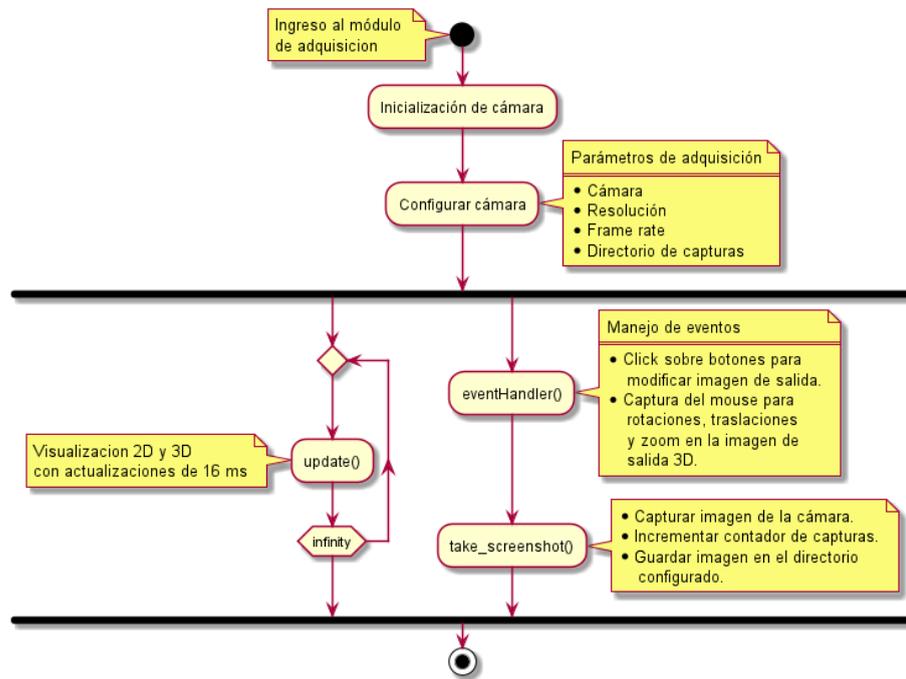
El *Frontend* está destinado a manejar las interacciones del usuario y a mostrar la información procesada por el *Backend*. El diagrama de la Figura 2.8 muestra dos componentes que interactúan con el *Backend* (Figura 2.7) para recibir información que luego puede ser visualizada en la interfaz de usuario.



**Figura 2.8 Diagrama de componentes del *Frontend* del proyecto.**

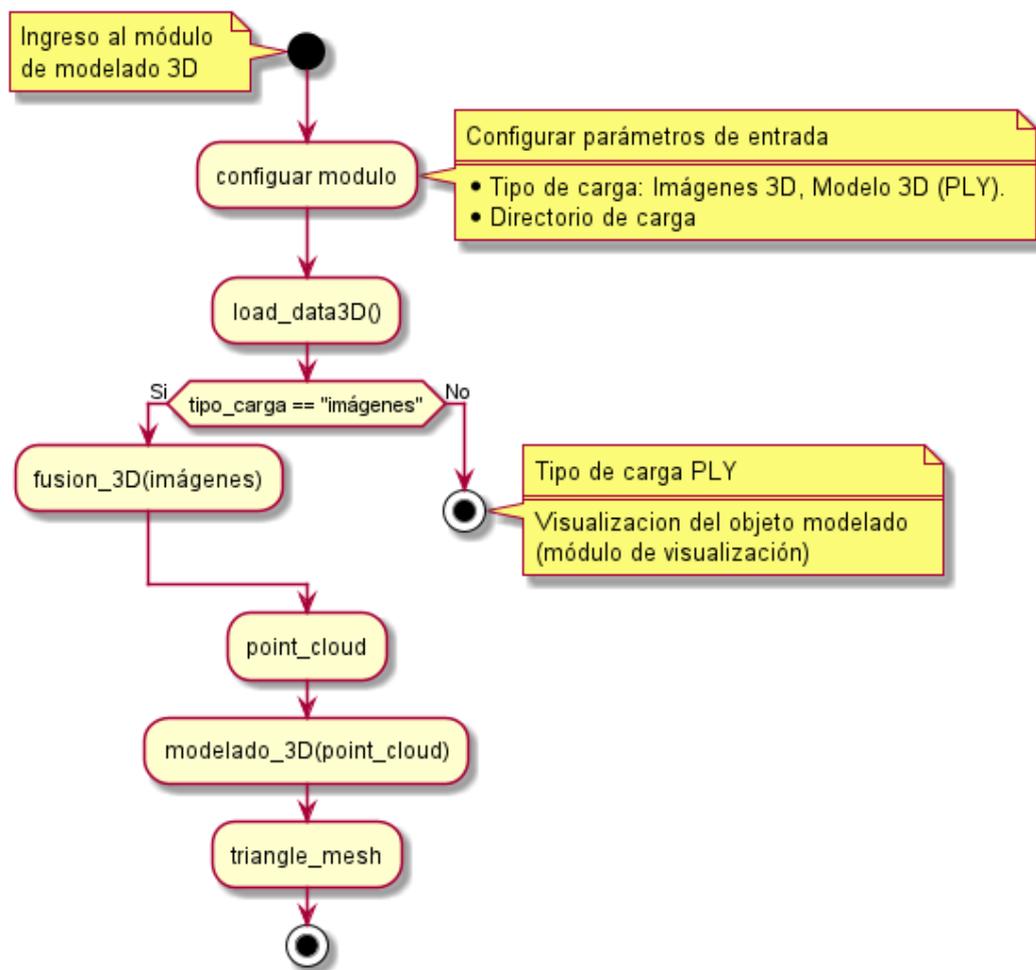
El comportamiento del sistema puede ser esquematizado por medio de diagramas de actividades. La Figura 2.9 describe las actividades que son realizadas por el módulo de adquisición de datos 3D, comenzando con la iniciación y configuración de la cámara, hasta la visualización del streaming de

los datos obtenidos por la cámara y el manejo de eventos e interacciones del usuario.



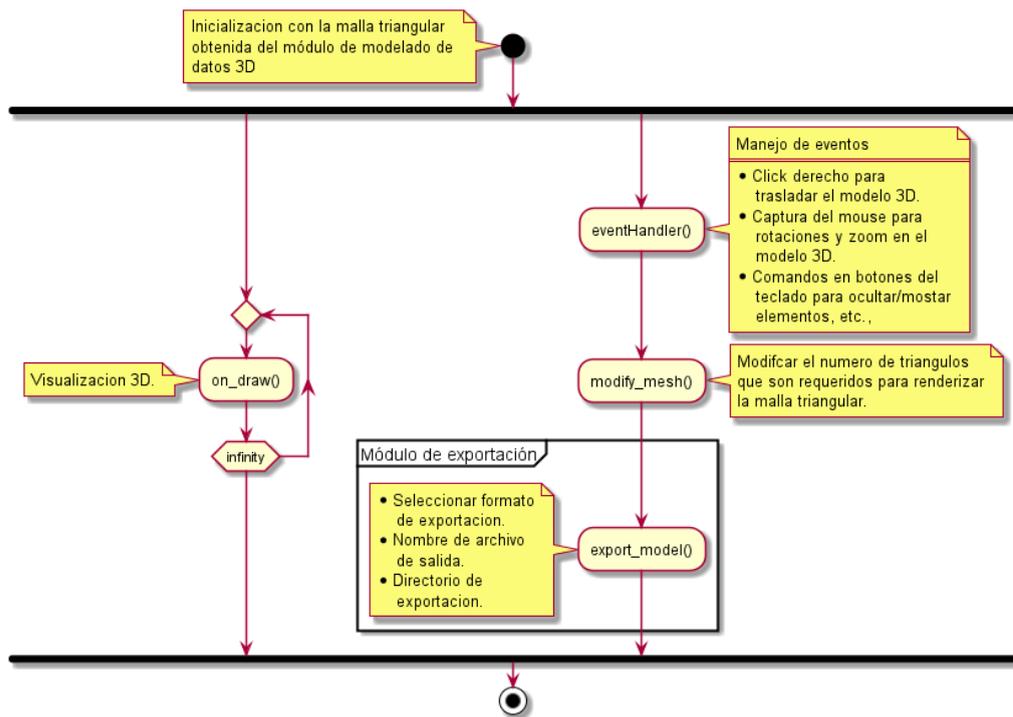
**Figura 2.9 Diagrama de actividades del módulo de adquisición de datos 3D.**

El diagrama de la Figura 2.10 muestra las actividades del módulo de generación de modelos virtuales, el cual es independiente y permite configurar el tipo de carga (Imágenes y Modelo PLY) y el directorio, para según estos parámetros generar una nube de puntos que posteriormente será modelada para finalmente obtener una representación en mallas triangulares, que es la entrada del módulo de visualización. Cuando se escoge como tipo de carga Modelo PLY se da apertura directamente al módulo de visualización.



**Figura 2.10 Diagrama de actividades del módulo de generación de modelos virtuales tridimensionales.**

Los módulos de visualización de modelos virtuales y de exportación a formatos comerciales están vinculados, la Figura 2.11 muestra las actividades de ambos módulos, describiendo las interacciones del usuario y las actualizaciones de la interfaz de usuario.

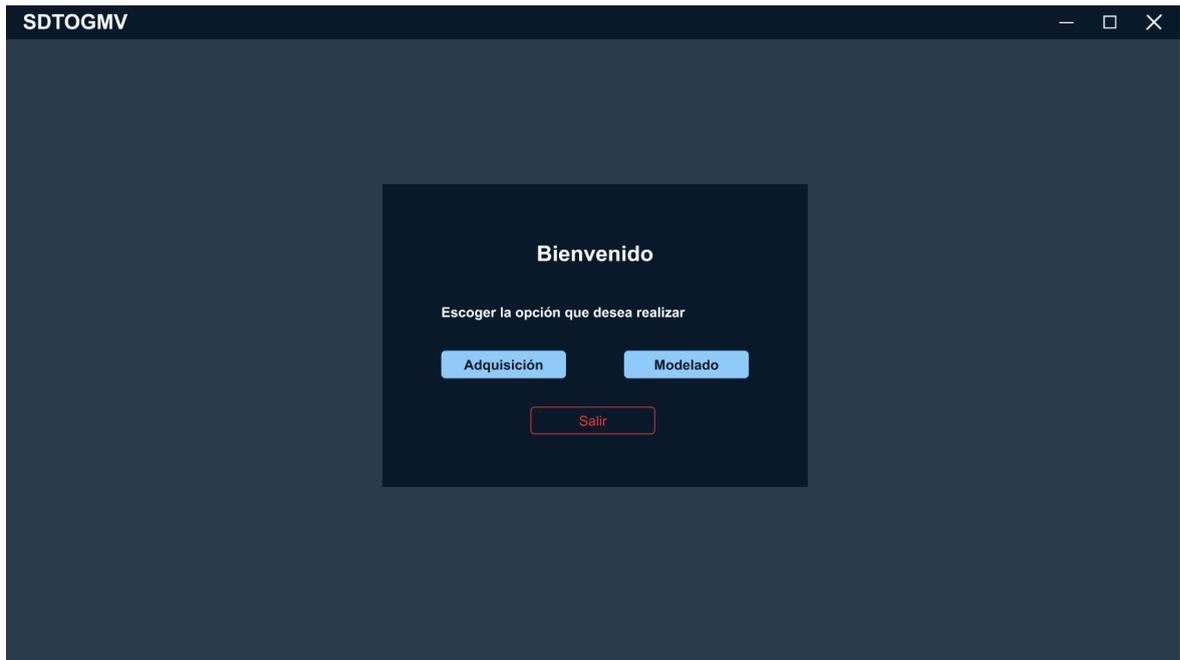


**Figura 2.11 Diagrama de actividades de los módulos de visualización de modelos virtuales y de exportación a formatos comerciales.**

### 2.3.4 Bocetos y flujos de pantalla

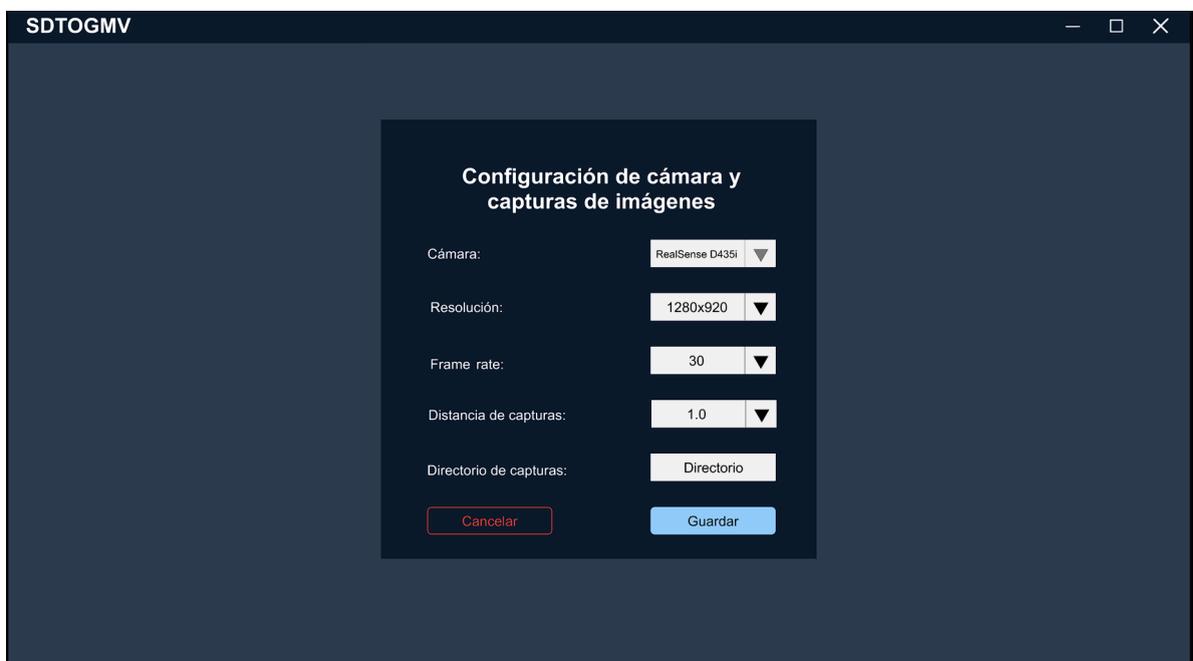
Enlace para acceder al flujo y diseño de pantallas: <https://bit.ly/2ZycuY9>.

**Vista 1 Ingreso a la aplicación:** Permite al usuario escoger entre capturar o adquirir imágenes 3D y modelar un objeto 3D, además se muestra la opción salir de la aplicación (Figura 2.12).



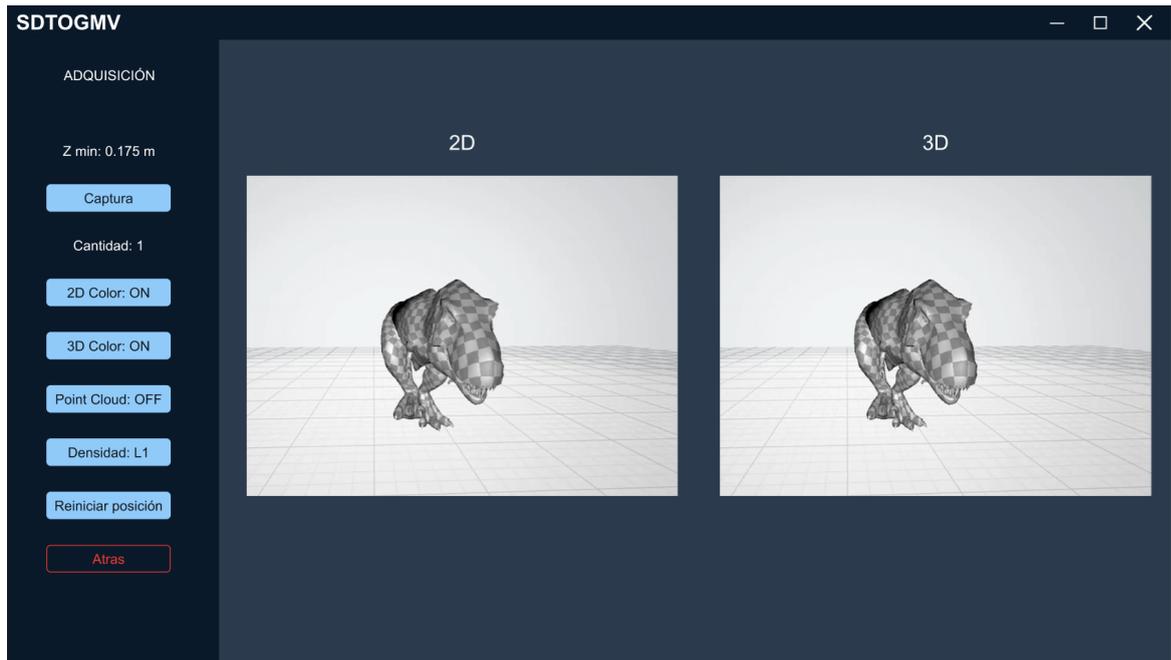
**Figura 2.12 Pantalla principal.**

**Vista 2 Módulo de adquisición:** Antes de ingresar al módulo se solicita al usuario configurar la resolución de la cámara, el *frame rate*, la distancia de captura y el directorio donde se almacenarán las imágenes capturadas (Figura 2.13).



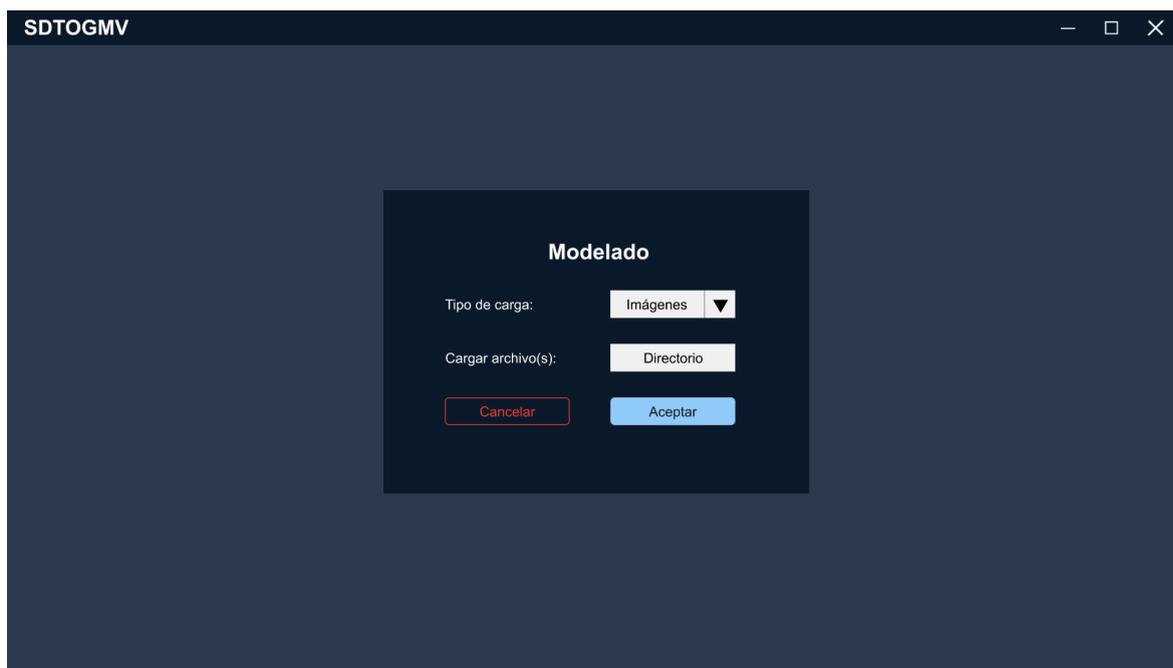
**Figura 2.13 Configuración de cámara.**

A continuación, se muestra la pantalla que permite ir capturando cada una de las vistas del objeto (Figura 2.14).



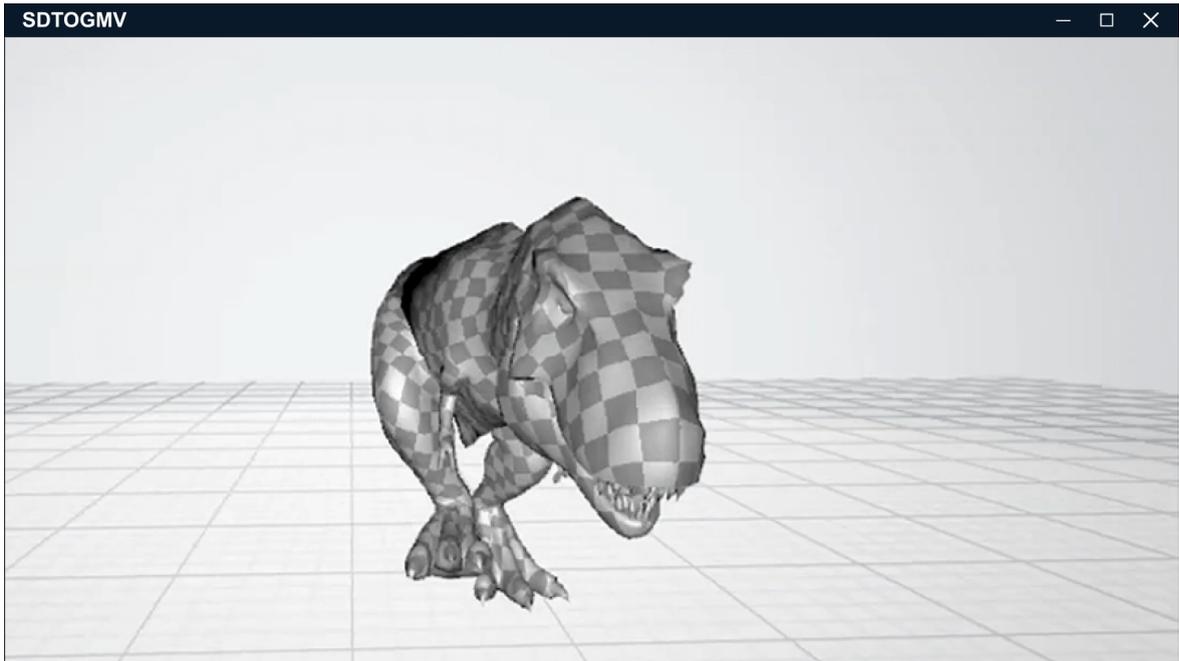
**Figura 2.14 Visualización streaming.**

**Vista 3 Módulo de modelado de datos 3D:** Antes de ingresar al módulo se solicita al usuario que escoja que tipo de carga desea modelar, donde puede seleccionar entre imágenes (obtenidas en el módulo de adquisición) o modelo 3D en formato PLY (Un objeto 3D previamente modelado y que se desea modificar) (Figura 2.15).



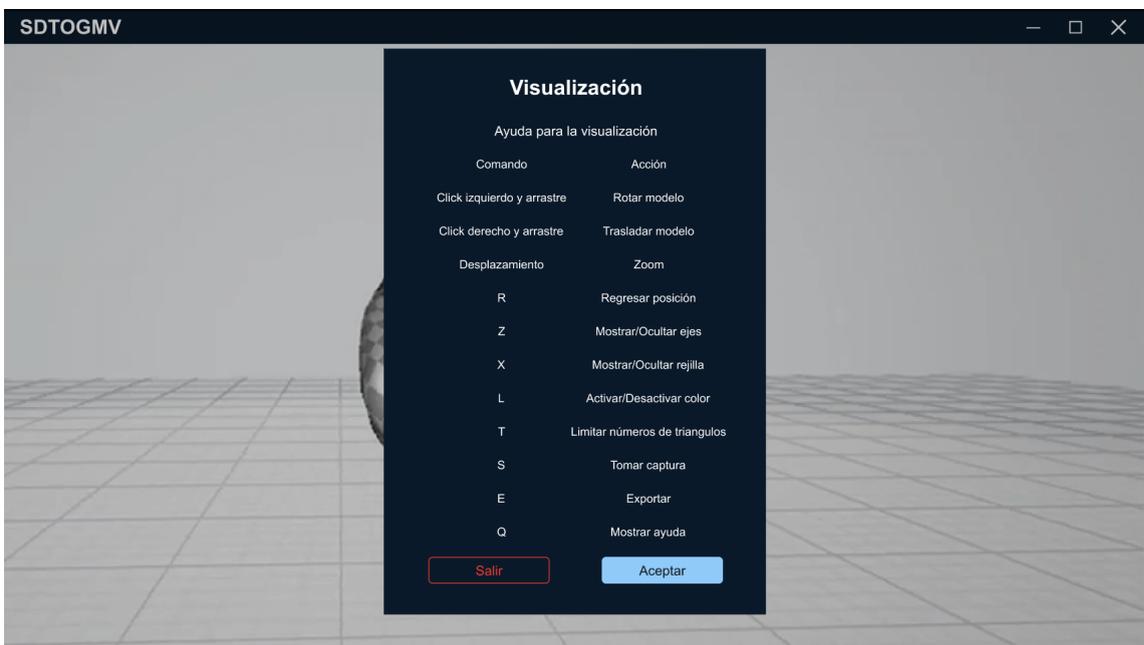
**Figura 2.15 Configuración para modelado.**

Posteriormente se muestra el modelo 3D generado (Figura 2.16), con opciones para modificar el número de triángulos que se muestran en la malla triangular que lo modela (Figura 2.18), trasladar el modelo en las direcciones arriba, abajo, izquierda y derecha. Además, es posible rotar el objeto 3D por medio del mouse.

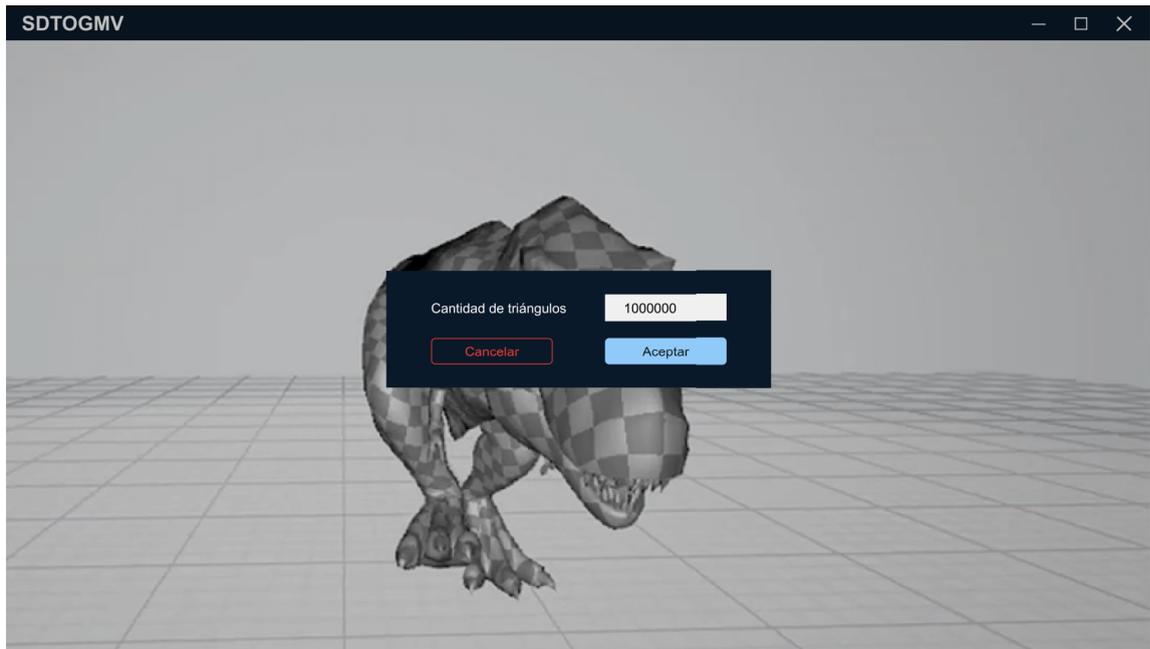


**Figura 2.16 Visualización de modelado.**

Como método de ayuda para la visualización se muestra un cuadro de diálogo con todos los comandos y opciones disponibles (Figura 2.17).

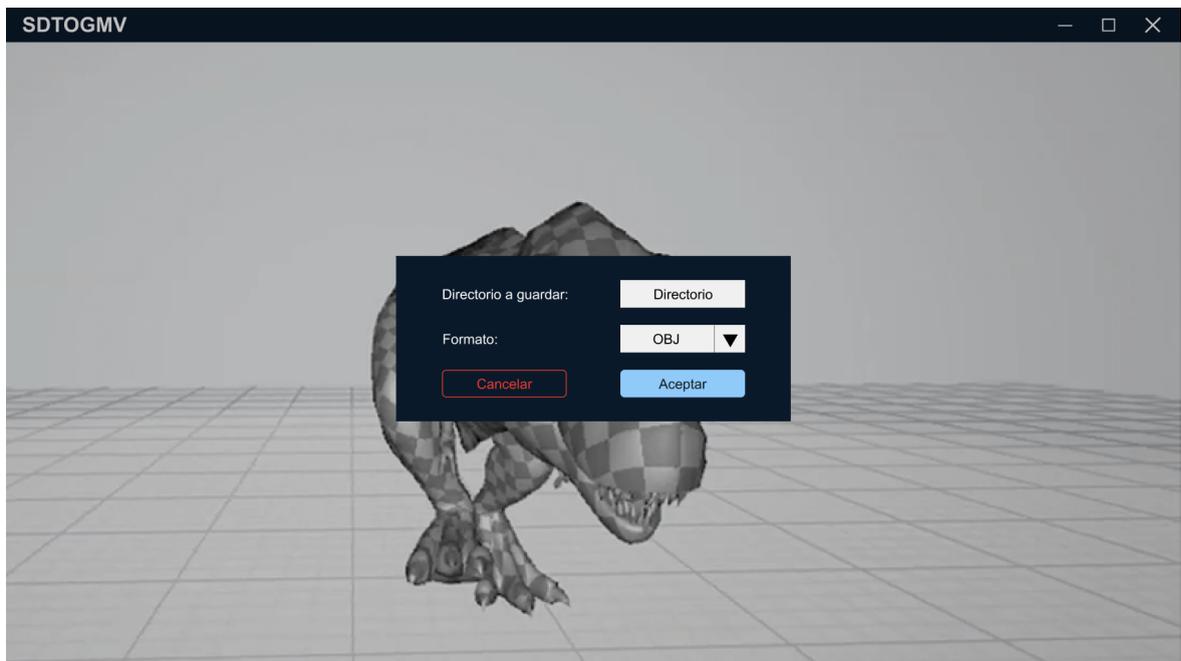


**Figura 2.17 Opciones de visualización.**



**Figura 2.18 Edición de modelado.**

**Vista 4 Módulo de exportación:** El acceso a esta pantalla se encuentra dentro del módulo de modelado de datos 3D, donde al seleccionar el botón de exportar muestra un diálogo para seleccionar el directorio de exportación del archivo y el formato (Figura 2.19).



**Figura 2.19 Exportación de modelado.**

## 2.4 Plan de implementación del proyecto

El plan de implementación está planificado para desarrollarse desde el 5 de noviembre hasta el 6 de enero con un total de 9 semanas. La planificación corresponde al desarrollo de las historias de usuarios y se describe en la Tabla 2.7:

**Tabla 2.7 Plan de implementación.**

Semanas	Fecha inicio	Fecha fin	Actividades	Sprint
1	26/11/21	02/12/21	Desarrollo de adquisición <i>backend</i> : conexión y configuración de cámara, captura de datos. Desarrollo <i>frontend</i> : diseño de interfaz principal para visualización streaming	1
2	03/12/21	09/12/21	Almacenamiento y carga de información para el modelado, modelamiento de los datos adquiridos. Procesamiento y modelamiento de objetos con fusión de datos y visualización de resultados.	1
3	10/12/21	16/12/21	<i>Frontend</i> de pantallas necesarias para la visualización y manipulación tanto rotación como traslación de los resultados de la fusión. Edición de mallas triangulares de los modelos, desarrollo <i>frontend</i> exportación de modelos en distintos formatos.	2
4	17/12/21	23/12/21	Desarrollo backend exportación de modelos en distintos formatos. Evaluación de la adquisición, procesamiento y modelado en diferentes condiciones para conocer o definir su robustez y visualización de resultados.	2
5	24/12/21	30/12/21	Evaluación de la adquisición, procesamiento y modelado en diferentes condiciones para conocer o definir su robustez y visualización de resultados. Evaluación de interfaces, pruebas de sistemas.	3

El siguiente capítulo describe los resultados y el análisis del software desarrollado, mostrando la implementación final del prototipo, la evaluación de todas las funcionalidades creadas, por medio de pruebas de integración y de sistema. Finalmente se realiza un análisis de los costos que fueron necesarios para desarrollar la solución.

# CAPÍTULO 3

## 3. RESULTADOS Y ANÁLISIS

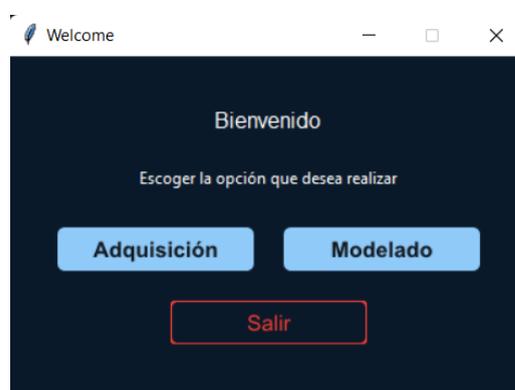
El presente capítulo aborda todo lo concerniente a la solución de software que se desarrolló como resultado de la metodología expuesta en el capítulo anterior. Al principio del capítulo se describe cada una de las funcionalidades del prototipo creado, sus interacciones y las diferentes opciones que ofrece.

Posteriormente se realizan evaluaciones por cada uno de los módulos del prototipo, para conocer cómo estos se comportan en diferentes situaciones. Continuando con una evaluación completa del prototipo para comprobar que exista una integración correcta de cada uno de los módulos. Entre las evaluaciones realizadas se analizó el rendimiento y los resultados del modelamiento, estableciendo así las variables que afectan directamente a la cantidad de poder computacional necesario o la resolución de capturas que debe ser escogida.

Finalmente, se demuestra cuáles son los costos necesarios para el desarrollo del prototipo, incluyendo valores de hardware, software y recurso humano.

### 3.1 Prototipo

Esta sección describe el prototipo desarrollado y los resultados obtenidos por cada uno de los módulos del proyecto. La Figura 3.1 muestra la interfaz principal la cual es la encargada de dar apertura al módulo de adquisición y al de modelado.



**Figura 3.1 Ingreso al sistema.**

### 3.1.1 Módulo de adquisición de datos 3D

- Conexión y configuración de la cámara: el sistema desarrollado funciona con la cámara Intel RealSense D435i, la cual proporciona varias resoluciones con su respectivo *frame rate* (Figura 3.2).

```
'1280x720': ('6', '15', '30'),  
'848x480': ('6', '15', '30', '60', '90'),  
'640x480': ('6', '15', '30', '60', '90'),  
'640x360': ('6', '15', '30', '60', '90'),  
'480x270': ('6', '15', '30', '60', '90'),  
'424x240': ('6', '15', '30', '60', '90'),
```

Figura 3.2 Resoluciones y *frame rates* de la cámara Intel RealSense D435i.

Además de los parámetros de la cámara, es posible configurar la distancia a la que se ubicará el objeto a capturar y el directorio donde se almacenará la información tridimensional (Figura 3.3).



Configuración de cámara y capturas de imágenes

Cámara: Intel RealSense D435i

Resolución: 1280x720

Frame Rate: 30

Distancia de capturas: 1.0

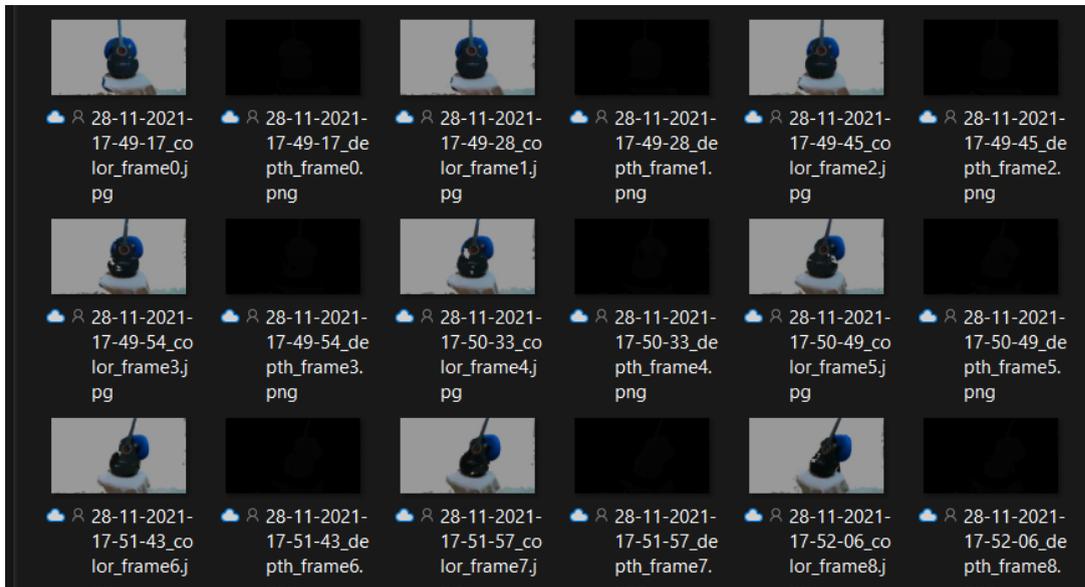
Directorio de capturas: C:/Users/kcamb/Documents/projects/python

Salir Guardar

Figura 3.3 Configuración de la cámara y captura de imágenes.

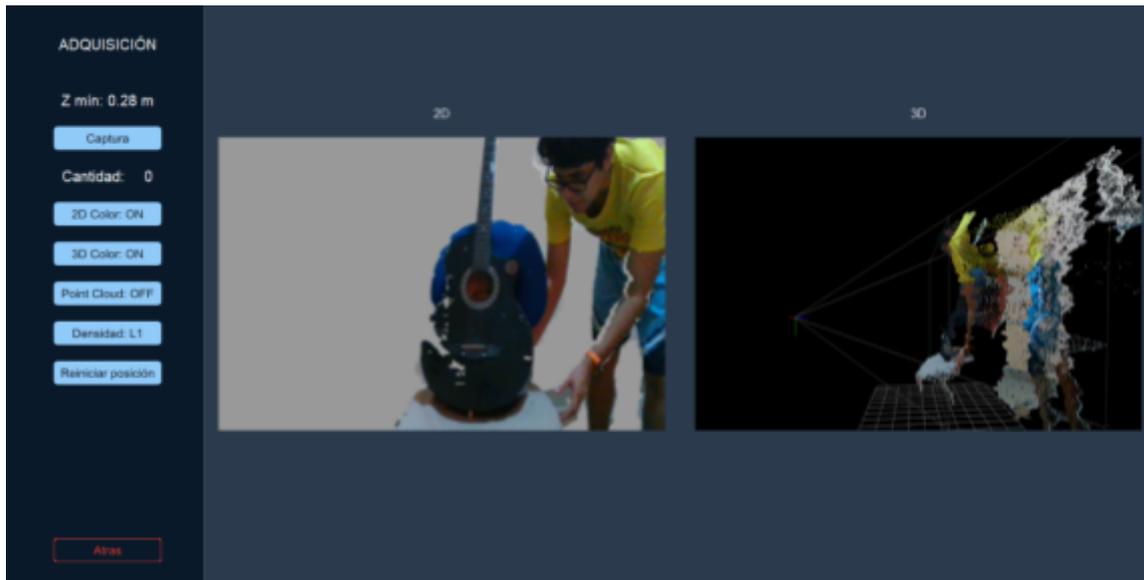
Las imágenes capturadas son almacenadas en el directorio seleccionado, obteniendo la información de color RGB en imágenes JPG y la información de profundidad en imágenes PNG. El nombre de los archivos está compuesto por la fecha y hora de captura, el tipo de imagen (color, profundidad) y el número de frame capturado (Figura 3.4). Además, en este mismo directorio se

almacenan las propiedades intrínsecas de la cámara utilizada, en un archivo de texto con el nombre *intrinsics* seguido del nombre de la cámara.



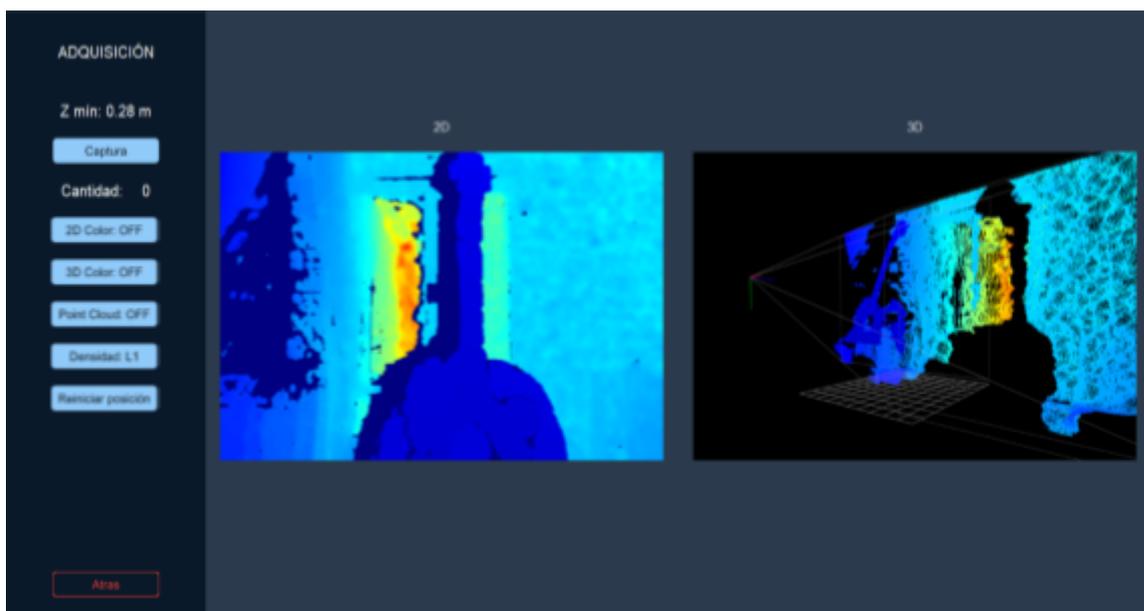
**Figura 3.4 Directorio con capturas de un objeto.**

- Interfaz de usuario: para la adquisición se creó una pantalla que permite visualizar el *stream* obtenido desde la cámara, de acuerdo con los parámetros de configuración. En esta pantalla se muestran dos salidas de video, la primera es una visualización 2D de la escena, aquí se ha eliminado el background, tomando en cuenta la distancia configurada, de tal forma que solo se enfoque el objeto y se omita detalles de la escena que pueden causar ruido. La segunda salida de video es una proyección 3D de la primera aquí es posible manipular la escena con rotaciones y traslaciones para tener una mejor percepción al ubicar el objeto a digitalizar (Figura 3.5).



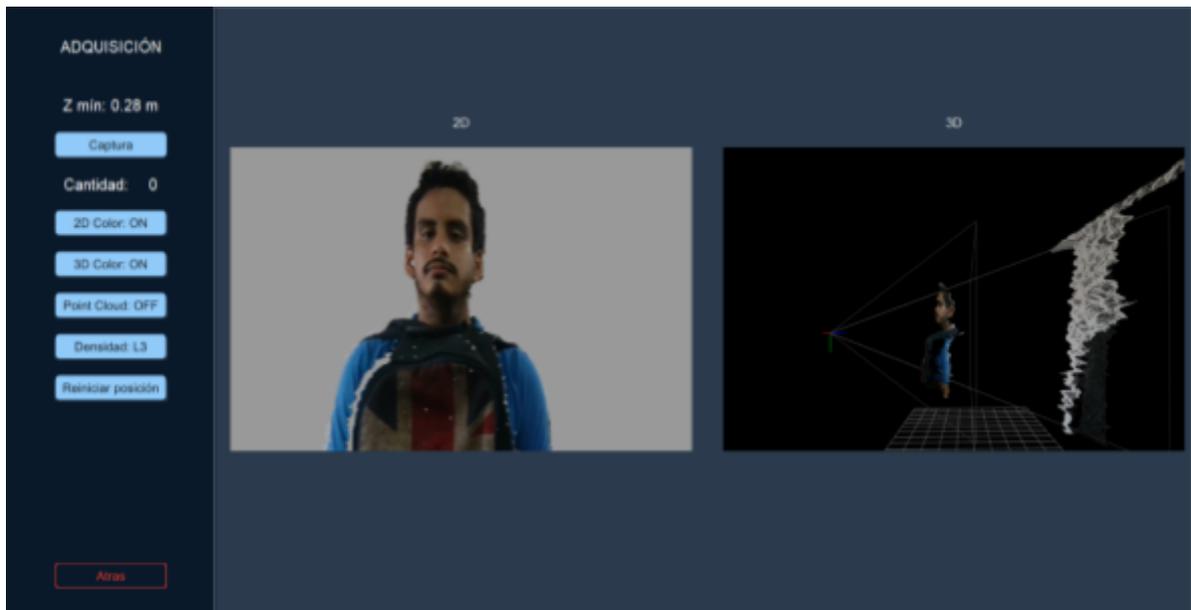
**Figura 3.5 Visualización principal en interfaz de adquisición.**

La interfaz de adquisición además cuenta con otros modos para visualizar el objeto que se va a digitalizar, permitiendo cambiar el modo de color tanto en la visualización 2D como en la 3D, de tal forma que se pueda observar el objeto más cercano en un tono azul y a medida que aumenta la distancia de profundidad los detalles de la escena cambian de tonos llegando a representar las distancias más grandes en colores rojizos (Figura 3.6).



**Figura 3.6 Interfaz de adquisición con el color 2D y 3D desactivado.**

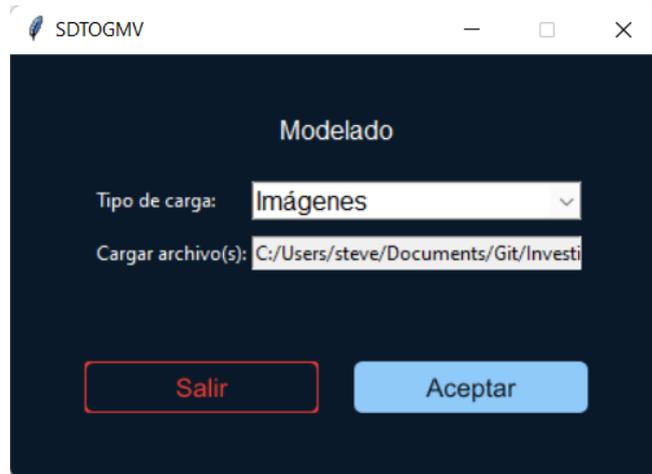
Existen otros parámetros que pueden ser configurados en esta interfaz y permiten realizar cambios sobre la visualización 3D, como son: Activar o desactivar nube de puntos, cambiar la densidad de puntos en la imagen de salida y regresar el objeto a la posición original (Figura 3.7).



**Figura 3.7** Interfaz de adquisición con densidad de puntos en L3.

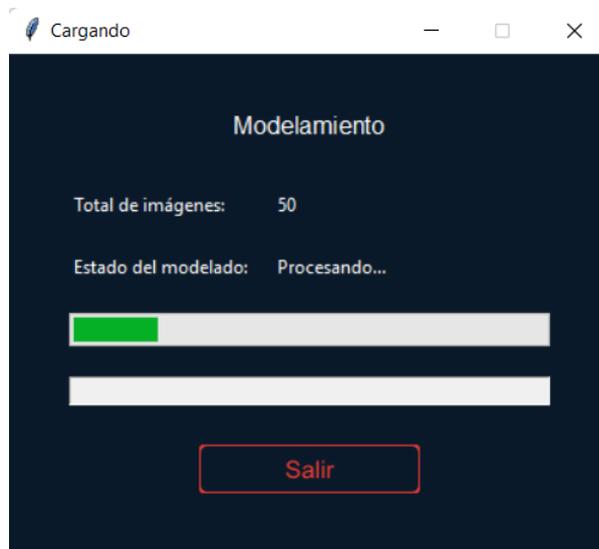
### **3.1.2 Módulo de modelamiento de datos 3D**

- Seleccionar imágenes capturadas: para lo cual es importante que la estructura del nombre de los archivos de las imágenes sea exactamente la ilustrada en la Figura 3.4. El modelamiento mantiene una secuencia frame a frame que utiliza el nombre de los archivos para obtener el orden de los frames que es requerido (Figura 3.8). La fusión de la información 3D necesita de las propiedades intrínsecas de la cámara las cuales se encuentran almacenadas en el mismo directorio que las imágenes.

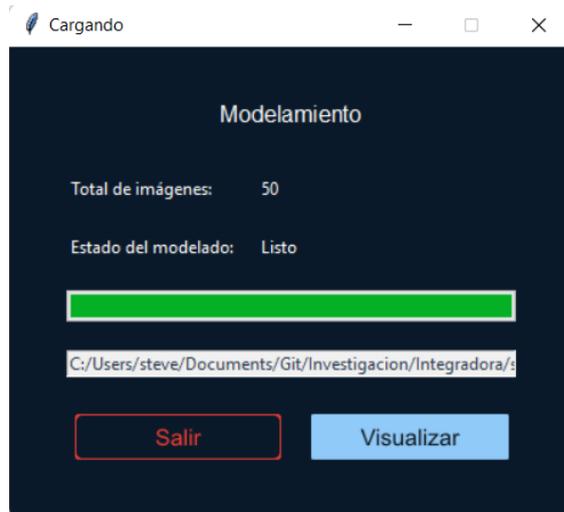


**Figura 3.8 Selección de imágenes a modelar.**

- Procesamiento de fusión y modelamiento: La información tridimensional es fusionada recorriendo cada frame capturado para encontrar el movimiento de la cámara entre dos pares de imágenes RGBD consecutivas, con lo cual se obtiene y registra la trayectoria (Figura 3.9). Luego se integra el volumen mediante TSDF (Figura 3.10). Las imágenes RGBD son creadas a partir de la información a color y profundidad.



**Figura 3.9 Procesando modelamiento.**



**Figura 3.10 Modelamiento finalizado.**

### **3.1.3 Módulo de visualización de modelos virtuales**

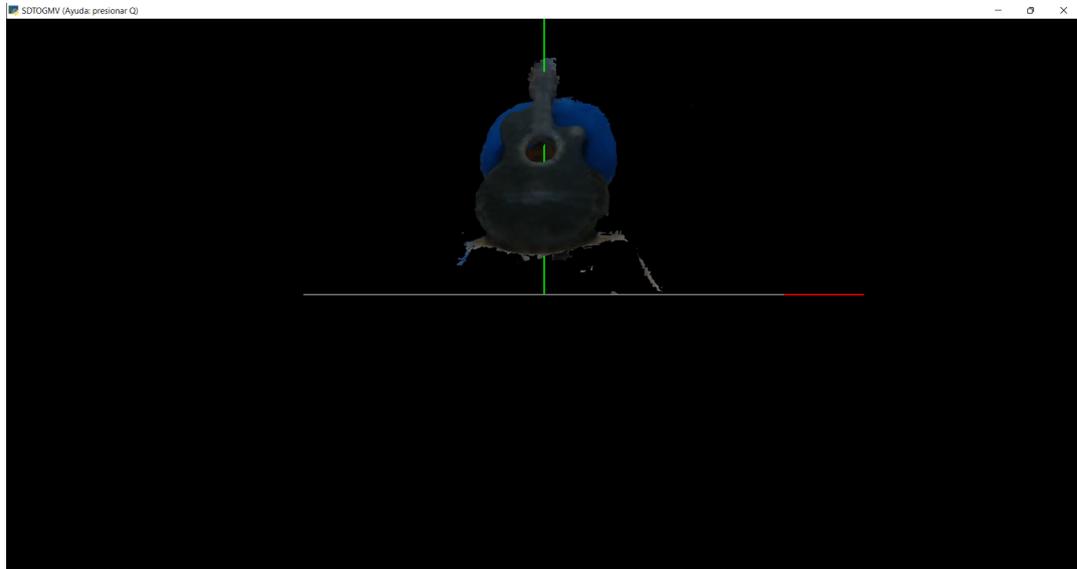
Visualización de archivos PLY: para la visualización, el software permite cargar un archivo PLY que contiene un modelo tridimensional previamente fusionado, así mismo este archivo puede contener una pequeña porción de información tridimensional, como puede ser una sección de un objeto (Figura 3.11).



**Figura 3.11 Selección de archivo PLY.**

Visualización del objeto modelado o archivo PLY: una vez finalizado el modelamiento del objeto o seleccionado el archivo PLY, se ilustrará el objeto modelado por medio de la extracción de una malla triangular. Esta pantalla

cuenta con un sistema de referencia (ejes x, y, z) y una base (rejilla) para mejorar la forma como se hacen los desplazamientos y traslaciones del modelo (Figura 3.12).



**Figura 3.12 Visualización del objeto.**

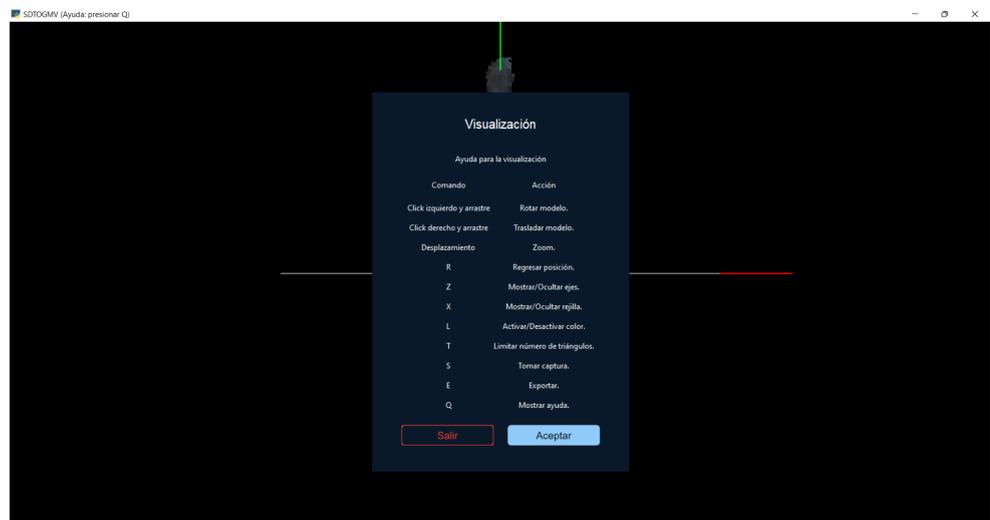
- Opciones en la pantalla de visualización: la visualización es controlada mediante los botones del mouse y algunas teclas del teclado (Tabla 3.1), que tienen como objetivo facilitar y personalizar la visualización (Figura 3.14). Para obtener una ayuda de cómo usar estos controles se debe presionar la tecla “Q” y se presentará una ventana con los comandos y acciones disponibles (Figura 3.13).

Los comando y acciones son:

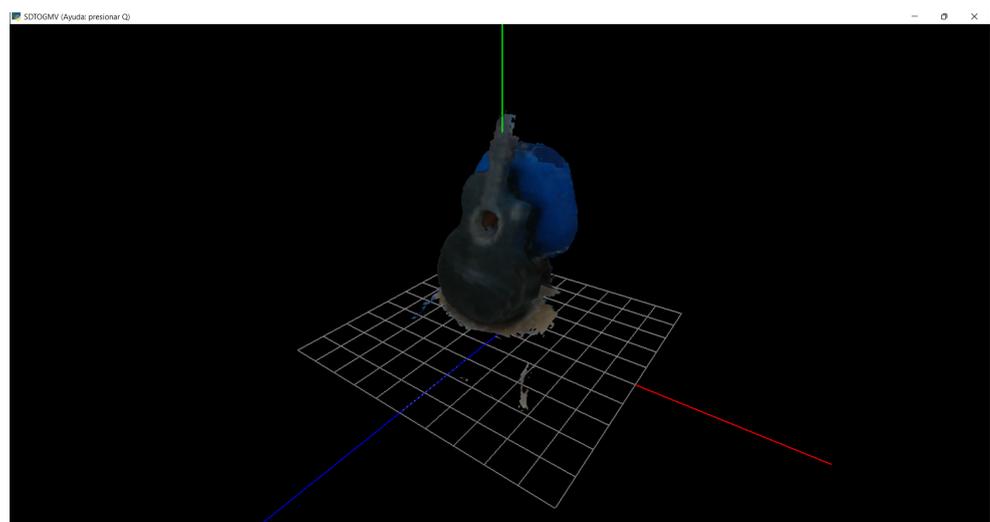
**Tabla 3.1 Opciones de visualización.**

Comando	Acción
Click izquierdo y arrastre	Rotar modelo.
Click derecho y arrastre	Trasladar modelo.
Desplazamiento	Zoom.
R	Regresar posición.
Z	Mostrar/Ocultar ejes.

X	Mostrar/Ocultar rejilla.
L	Activar/Desactivar color.
T	Limitar número de triángulos.
S	Tomar captura.
E	Exportar.
Q	Mostrar ayuda.



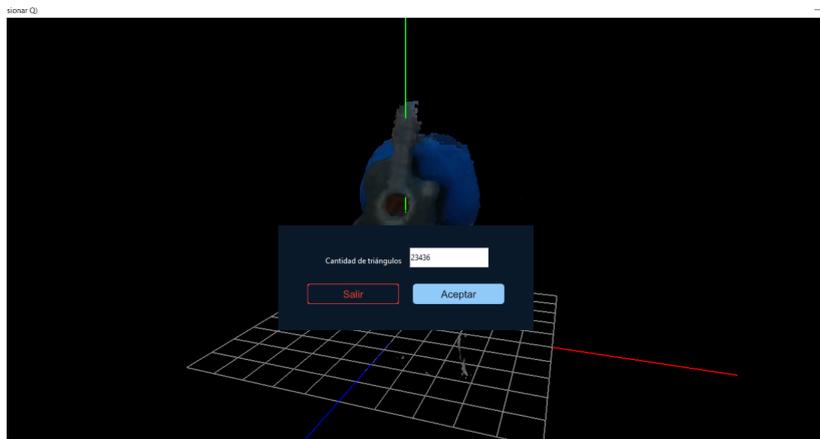
**Figura 3.13 Visualización de opciones.**



**Figura 3.14 Objeto rotado y trasladado.**

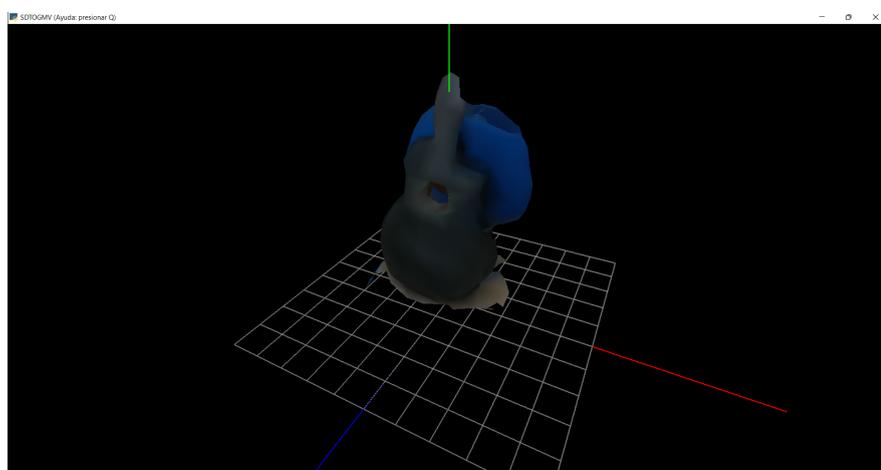
- Edición de mallas triangulares: limitar el número de triángulos es una de las opciones que permite la visualización, de este modo cambiar el acabado del objeto entre fino con mayor cantidad de triángulos y grueso con menor cantidad de triángulos. Esto de acuerdo con el uso que se le vaya a dar al modelo como se menciona en el capítulo 2 (Metodología, Análisis de los datos).

Al presionar la tecla “T” se muestra una ventana en donde se visualizan la cantidad actual de triángulos y además se permite editar este valor (Figura 3.15).



**Figura 3.15 Ventana para ingresar cantidad de triángulos.**

Al aceptar se renderizará nuevamente el objeto con la cantidad de triángulos seleccionados (Figura 3.16).



**Figura 3.16 Objeto nuevamente renderizado.**

### 3.1.4 Módulo de exportación de modelos virtuales

- Opción de exportación: en la pantalla de visualización, al presionar la tecla “E” se presenta un cuadro de diálogo con los diferentes formatos disponibles para la exportación. (Figura 3.17).

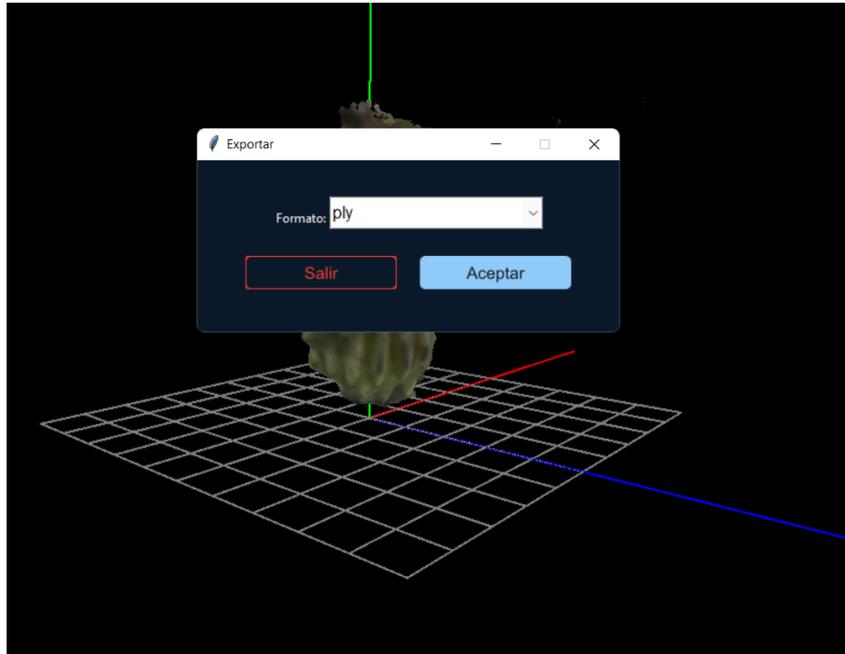


Figura 3.17 Elección de formato.

- Selección de formato a exportar: los formatos permitidos son “PLY”, “STL”, “OBJ”, “OFF” y “GLTF” (Figura 3.18).

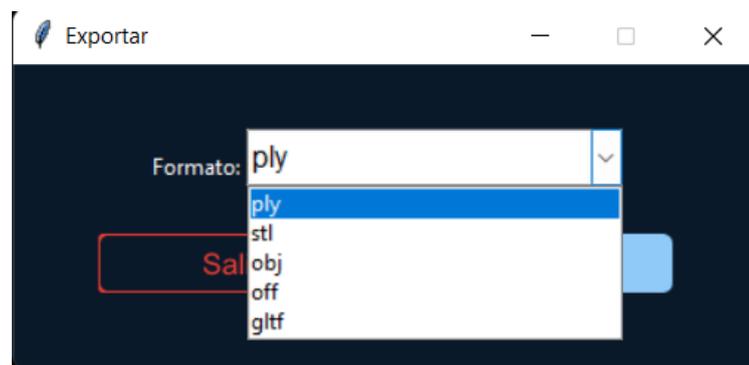
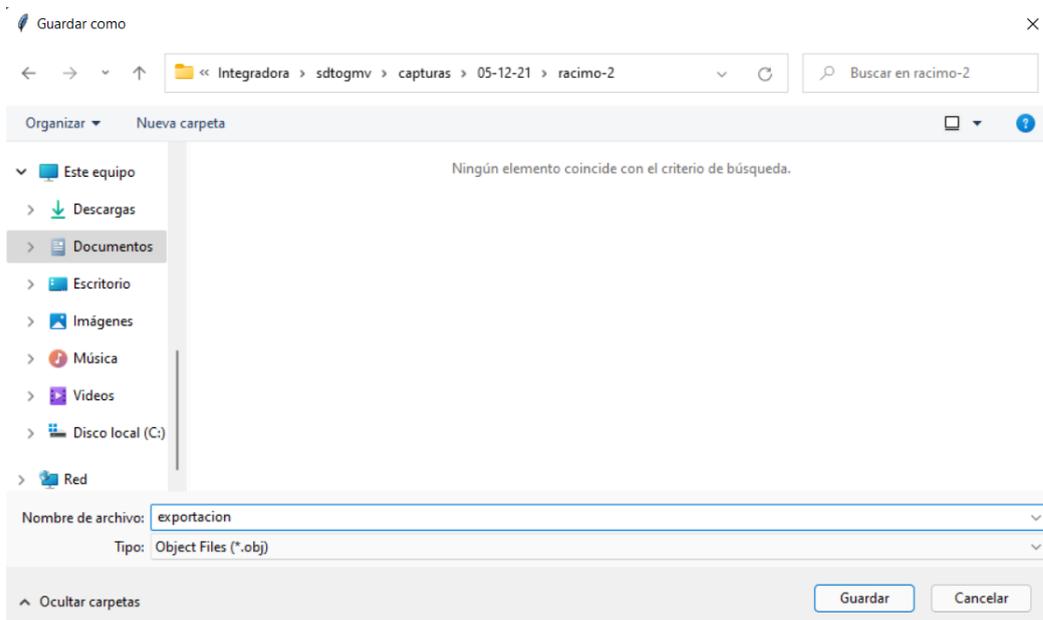


Figura 3.18 Formatos de exportación permitidos.

- Luego al dar “aceptar”, será necesario escoger la ubicación para guardar archivo con el formato correspondiente (Figura 3.19).



**Figura 3.19 Escoger ubicación de archivo a exportar.**

### 3.2 Evaluaciones

- Las pruebas de adquisiciones permitieron definir los parámetros de configuración de la cámara Intel RealSense D435i de acuerdo con el ecosistema necesario para obtener los mejores resultados.

Estos parámetros y condiciones se describen en la Tabla 3.2:

**Tabla 3.2 Condiciones y configuración.**

Condición	Descripción
Luz	Es necesario cuidar la intensidad de la luz ya que este puede ocasionar perdida de información con la cámara Intel RealSense D435i (Figura 3.20). Evitar que el objeto presente colores brillantes en escenarios muy iluminados o aplicar luz difusa.
Cantidad de capturas	Aumentar la cantidad de capturas y disminuir la distancia entre cada intervalo, ayuda a mejorar el modelamiento ya que es necesario calcular la trayectoria de la cámara con el movimiento (Figura 3.20). Se recomienda al menos 15 capturas si se recorren los 360 grados del objeto (Figura 3.21). Cabe mencionar que mientras más capturas más poder computacional será necesario.

Resolución	Este parámetro de configuración es necesario configurarlo de acuerdo con el tamaño del objeto ya que objetos pequeños requieren encontrarse a menor distancia de la cámara, por lo cual es necesario disminuir la resolución permitiendo así disminuir el z mínimo. Para objetos grandes es necesario aumentar la resolución para realizar un modelado completo.
------------	--



**Figura 3.20 Modelo obtenido con pérdida de información debido a problemas de iluminación.**

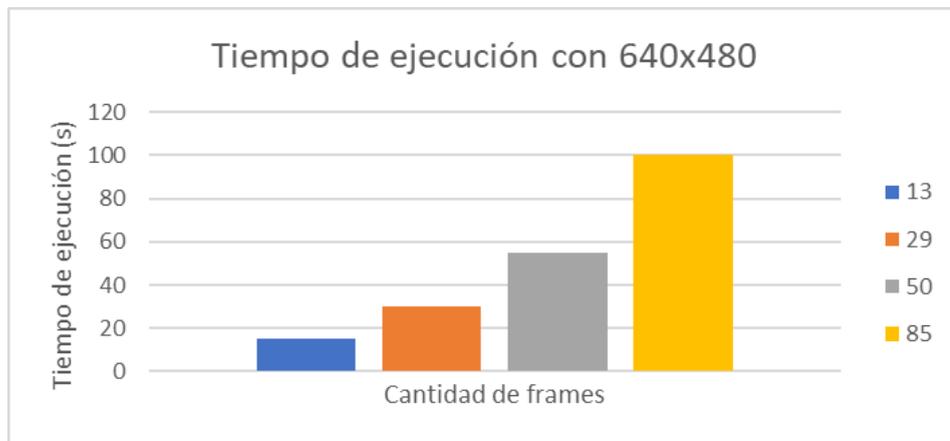


**Figura 3.21 Ejemplo del movimiento entre un par de frames.**

- Las pruebas de modelamiento se realizaron en base al tiempo que le toma completar el trabajo. Estas pruebas se hicieron en un computador con las siguientes características: Intel Core i7 9th Gen, 16GB de RAM y una GPU NVIDIA GeForce GTX 1650.

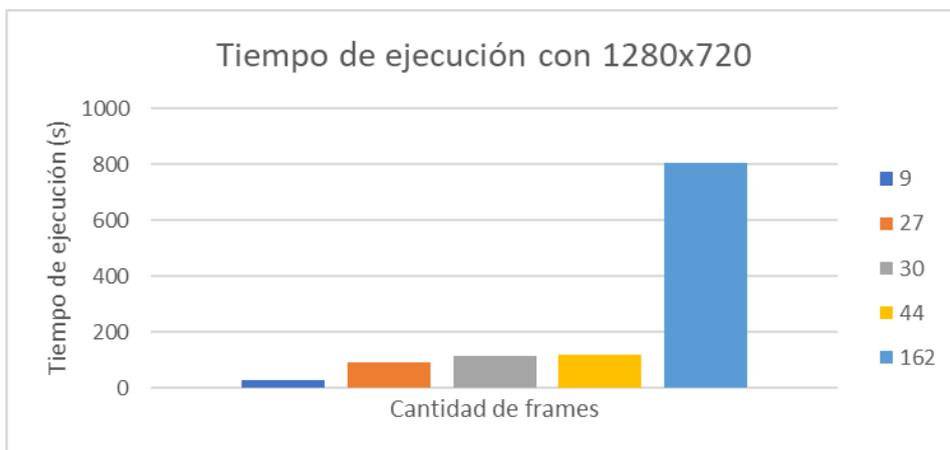
Los tiempos de ejecución varían con respecto a la cantidad de información o tamaño de frames y la cantidad de capturas que se utilicen para el modelamiento como se puede observar en la Figura 3.22 y Figura 3.23.

Se observa que con la resolución de 640x480 los tiempos no superaron los 100 segundos para realizar el modelamiento (Figura 3.22). Esta resolución es ideal para objetos pequeños y requiere menor poder computacional.



**Figura 3.22 Histograma de tiempo vs cantidad de frames con resolución 1280x720.**

Se observa que con una resolución mayor como 1280x720, el modelamiento supera los 100 segundos con tan solo 30 frames por la cantidad de información que posee esta resolución (Figura 3.23).



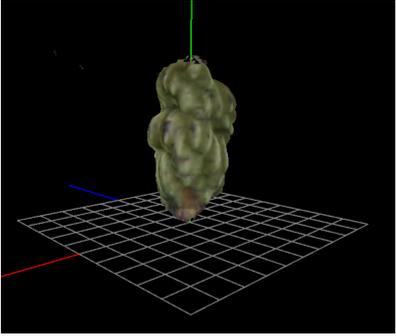
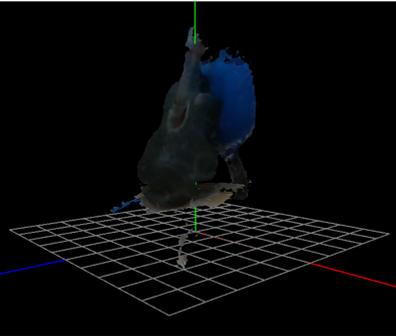
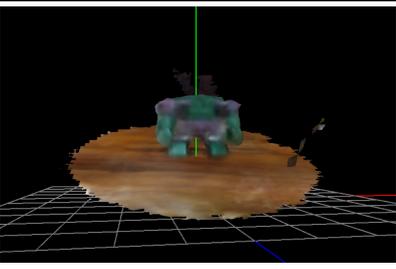
**Figura 3.23 Histograma de tiempo vs cantidad de frames con resolución 1280x720.**

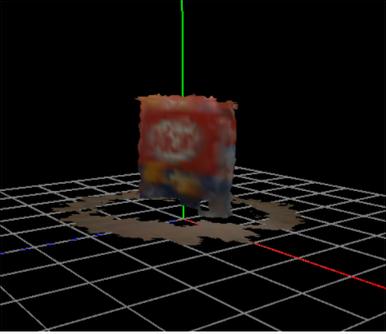
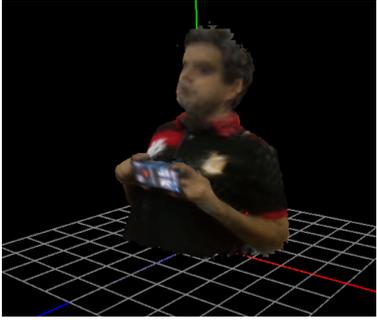
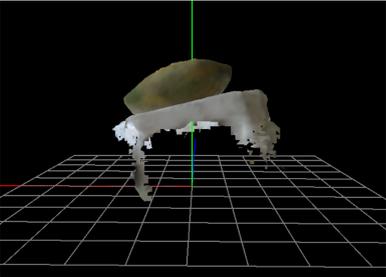
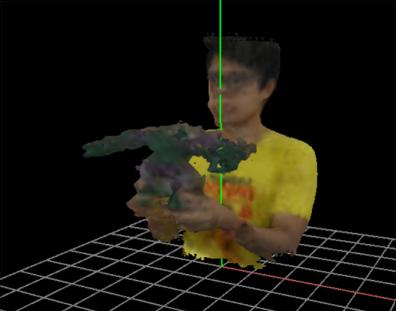
- Para las pruebas de modelamiento se usaron diferentes sujetos, con la finalidad de observar cómo se comportaba el sistema al modelar objetos con diferentes características. Las pruebas se realizaron con los siguientes sujetos: en frutos como racimo de banano o bodega, en objetos como guitarra, muñeco de juguete, caja de galletas y en personas.

- Pruebas de visualización con los diferentes objetos modelos:

La cantidad de triángulos resultantes en los modelos finales depende de la resolución con la cual se realizaron las capturas y no tanto del número de capturas hechas, como se puede observar en la Tabla 3.3.

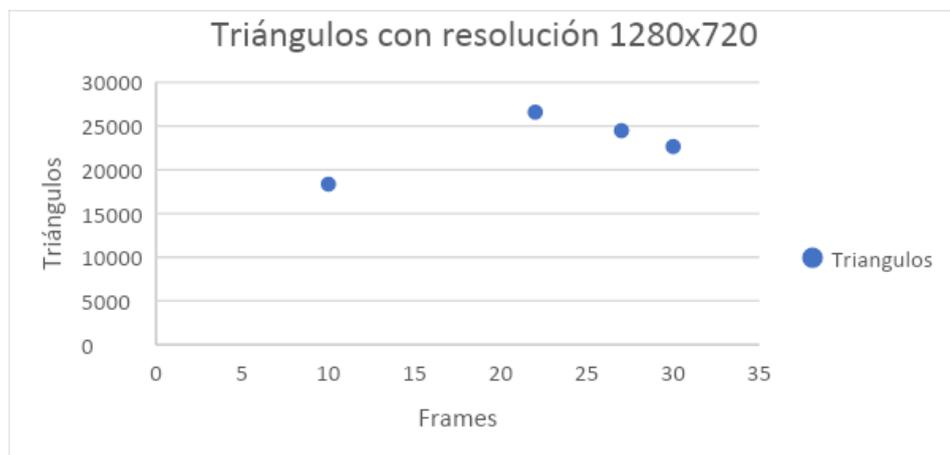
**Tabla 3.3 Triángulos generados por modelos.**

Modelo	Triángulos	Frames	Resolución
	18118	50	640x480
	24461	27	1280x720
	12571	102	640x480

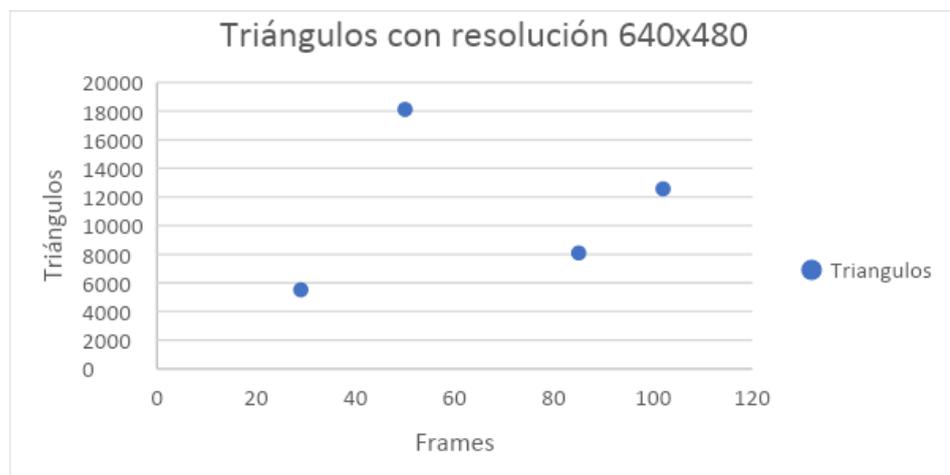
	5515	29	640x480
	22640	30	1208x720
	8086	85	640x480
	18342	10	1280x720
	26574	22	1280x720

- Evaluación de mallas triangulares:

El número de triángulos que se generan en el modelo final depende del volumen del objeto capturado, mientras más volumen hay un mayor número de triángulos, por esto se recomienda usar resoluciones más altas cuando se intenta modelar un objeto de mayores dimensiones para así obtener más información. La cantidad de frames capturados no afectan directamente al número de triángulos en el modelo final, los gráficos de la Figura 3.24 y Figura 3.25 muestran cómo es posible tener menor número de triángulos con muchas más capturas y esto es debido a que en el proceso de modelado el algoritmo elimina información y solo toma lo que necesita para construir el objeto.



**Figura 3.24** Relación entre la cantidad de triángulos en el modelo final y el número de frames a una resolución de 1280x720.



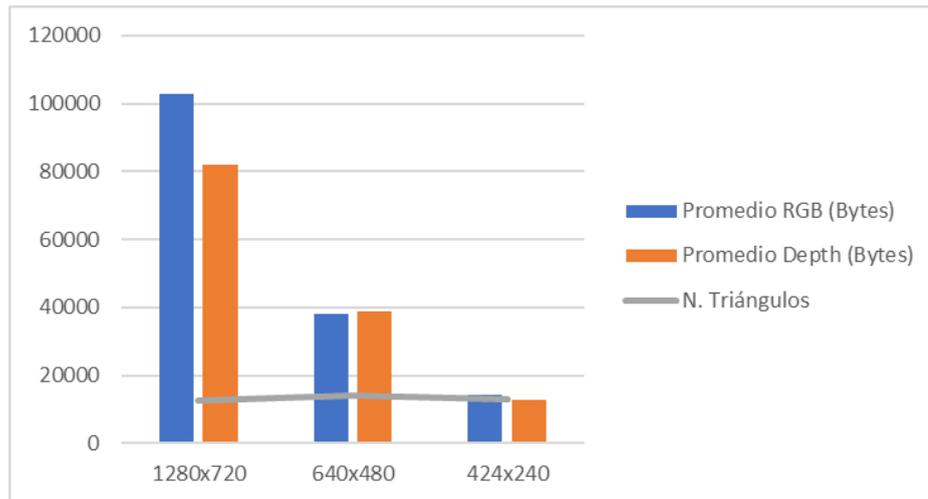
**Figura 3.25** Relación entre la cantidad de triángulos en el modelo final y el número de frames a una resolución de 640x480.

- Comparación de un mismo objeto a diferentes resoluciones:

Un mismo objeto que es capturado con diferentes resoluciones en la etapa de adquisición presenta una diferencia considerable, considerando los tamaños de los archivos generados con la información de color y de profundidad (Figura 3.27), a mayor resolución se captura mucha más información. Después de realizado el modelamiento, la cantidad de triángulos que conforman la malla triangular del objeto son relativamente iguales, con una diferencia poco significativa (Figura 3.27), lo que indica que los modelos resultantes son muy similares, como se observa en la Figura 3.26. Es importante mencionar que al ser un objeto grande el que se modeló, usar resoluciones pequeñas afecta a la exactitud de la fusión, debido a que se captura menos información provocando resultados con errores muy notorios (Figura 3.26, modelo de resolución 424 x 240).



**Figura 3.26 Modelos virtuales capturados a resoluciones de 1280x720, 640x480 y 424x240 respectivamente.**

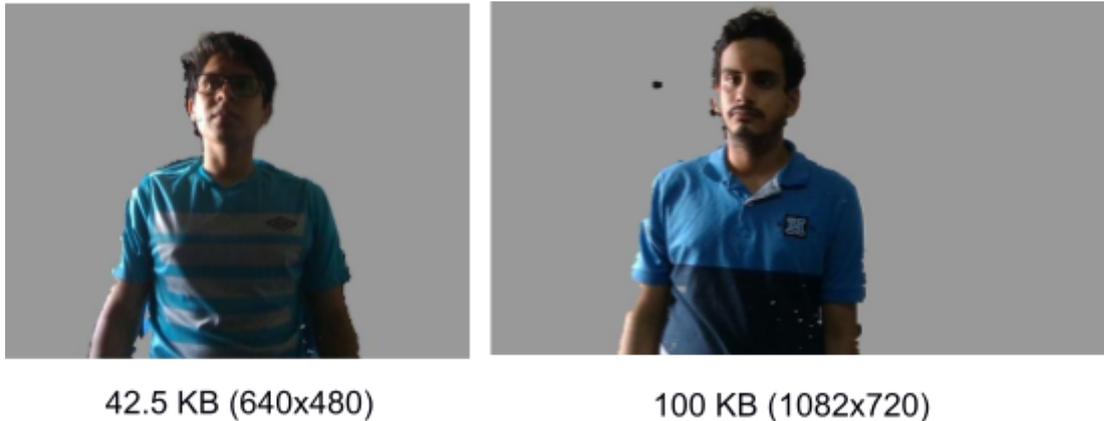


**Figura 3.27 Comparación entre el promedio del tamaño de archivo de los frames de un mismo objeto capturado a diferentes resoluciones y su relación con la cantidad de triángulos en el modelo final.**

- Pruebas de interfaz

Después de evaluar el sistema por cada uno de los módulos y analizar los resultados obtenidos, ahora es momento de observar el comportamiento del software cuando todos los módulos que lo componen interactúan entre sí. Es importante recordar que cada componente contiene sus propias configuraciones, por lo que se debe asegurar que el software no tendrá un comportamiento inesperado al variar alguno de estos parámetros. Para lograr este objetivo es necesario hacer pruebas exhaustivas de todo el sistema.

El software responde adecuadamente al realizar capturas consecutivas de un objeto, se ha observado que realizar cambios en la densidad de puntos provoca un mayor uso de recurso computacional en la toma de las capturas, debido a que es necesario hacer un procesamiento extra para mostrar la nueva densidad. Modificar los parámetros dentro de la interfaz de adquisición no afecta las capturas finales obtenidas, porque siempre se trabaja sobre los datos originales (como se muestra originalmente el streaming). Los frames finales obtenidos pueden variar si se modifica la resolución con la que se hacen las capturas lo cual repercute en el tamaño final del archivo (Figura 3.28).

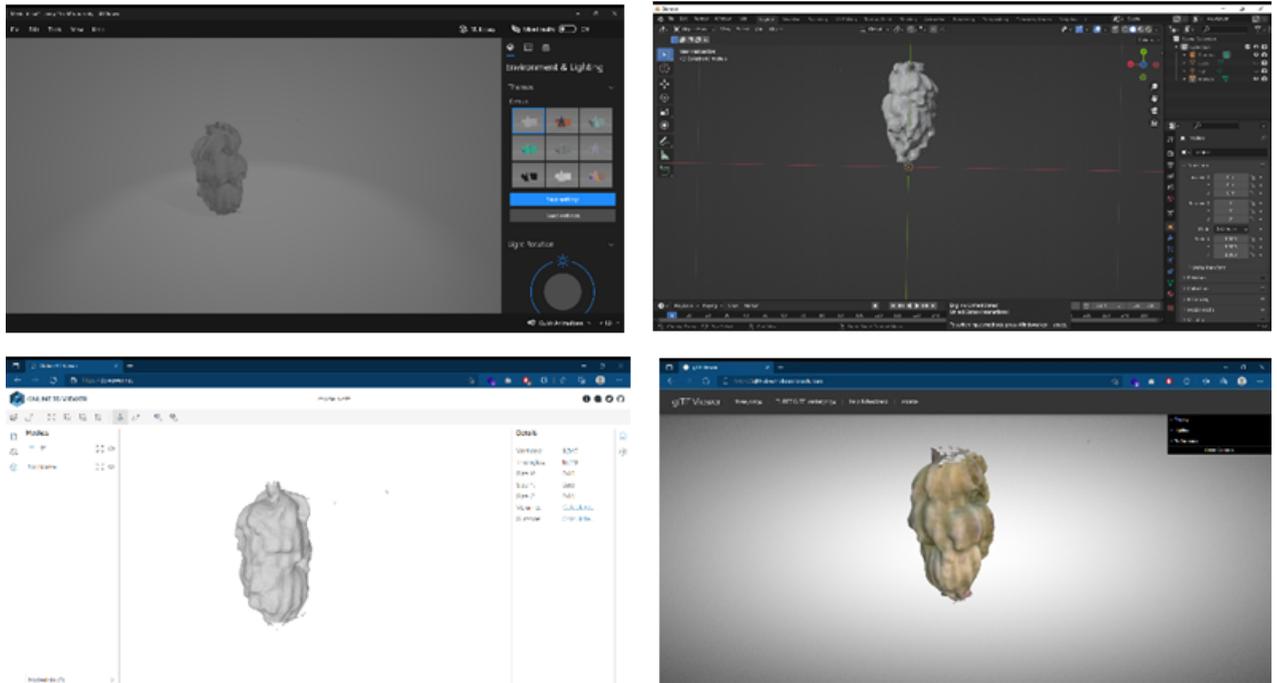


**Figura 3.28 Frames a diferentes resoluciones y tamaños de archivo.**

El software sigue su proceso funcional tomando las capturas obtenidas anteriormente en conjunto con el archivo generado que contiene las propiedades intrínsecas de la cámara. A partir de aquí se procede con el modelamiento del objeto. Todas las pruebas indicaron que mientras no se haya afectado el archivo de las propiedades intrínsecas y las capturas mantengan el formato establecido de nomenclatura, el software es capaz de terminar el modelamiento con éxito. Al finalizar el proceso es generado un nuevo archivo, dentro del mismo directorio donde están las capturas el cual es el modelo 3D en formato PLY.

El archivo PLY generado en el módulo de generación de modelos virtuales es el utilizado para la visualización. El visualizador del software carga el modelo leyendo la malla triangular del mismo, por lo que si el modelo está compuesto por muchos triángulos la pantalla se mostrará en blanco hasta que el procesamiento termine. Una vez cargado el modelo, la visualización está optimizada para permitir una experiencia agradable y natural al usuario.

La exportación es accesible desde el visualizador, presionando la tecla “E”, luego escogiendo el formato de exportación, posteriormente se genera un archivo con el formato seleccionado. Los archivos que se obtienen como resultado de la exportación pueden ser abiertos en otros softwares que se encuentran disponibles y son de libre acceso (Figura 3.29).



**Figura 3.29 Modelo exportado en diferentes formatos visualizado en softwares disponibles con acceso libre.**

### 3.3 Análisis de costo

Para el desarrollo y ejecución del proyecto se demandó del uso de diferentes recursos de hardware y poder de procesamiento, a continuación, se detalla cada uno de estos costos:

- Cámara Intel RealSense D435i: \$329.00
- Computador Intel Core i7 9th Gen, 16GB de RAM y una GPU NVIDIA GeForce GTX 1650: \$1200
- Consumo de energía eléctrica: \$10
- Costo humano: Tomando en cuenta las fechas establecidas en el Plan de implementación del proyecto, se tienen 4 semanas de trabajo con un promedio de 6 horas al día, en total son 120 horas. Considerando dos recursos y un total de \$5 la hora, se tiene que el costo es de \$1200.
- Aplicaciones de software de terceros:
  - Open3D (open source): \$0
  - SDK de Intel RealSense (open source): \$0
  - Lenguajes de programación (open source): \$0
  - Proyectos de terceros (open source): \$0

El costo total del proyecto considerando los rubros antes expuestos es: \$2739.

Una vez hechas las evaluaciones correspondientes y analizado el funcionamiento del prototipo, es momento de recopilar toda la información generada, para en el siguiente capítulo concluir con base en los resultados obtenidos, estableciendo la importancia de este proyecto y aportando con recomendaciones para que el trabajo realizado obtenga un mayor alcance.

# CAPÍTULO 4

## 4. CONCLUSIONES Y RECOMENDACIONES

Este capítulo tiene como objetivo describir la importancia del proyecto, tomando en cuenta sus fortalezas y debilidades. Se incluyen conclusiones que resaltan lo más importante de los resultados obtenidos y recomendaciones que ayudarán a mejorar el desempeño del software realizado. El cierre del capítulo permite conocer las implicaciones en los posibles trabajos futuros que podrían ser adaptados al resultado obtenido.

### 4.1 Conclusiones

- El resultado final del proyecto fue el desarrollo de un prototipo funcional capaz de modelar objetos reales en un entorno virtual tridimensional permitiendo además la exportación a diferentes formatos comerciales. De esta forma las industrias que su actividad productiva se basa en el uso de modelos tridimensionales pueden beneficiarse de un software que permite abstraer un objeto real como un modelo 3D el cual puede ser procesado por computadora para obtener información volumétrica, conocer si cumple con estándares de calidad o reproducir el objeto por medio de impresión 3D.
- El uso de hardware especializado para capturar datos 3D como el caso de la cámara Intel RealSense D435i y los documentos técnicos que brindan la información necesaria para configurar la cámara y el sistema de adquisición permitió obtener resultados, que bajo los lineamientos descritos en este proyecto son óptimos para modelar un objeto completamente en un entorno virtual.
- El modelamiento es el núcleo de todo el sistema, es aquí donde las imágenes capturadas convergen y producen un modelo tridimensional. Este módulo es independiente del resto del software por lo que puede ser adaptado a otros sistemas de adquisiciones, siempre y cuando se proporcione el formato adecuado para los nombres de archivos de las imágenes RGB y de profundidad y se conozca las propiedades intrínsecas de la cámara.
- El módulo de visualización debido a su variedad de opciones y optimización se convierte en una excelente opción para obtener una vista previa del resultado

del modelamiento. La interacción permitida da la posibilidad de analizar si existieron desperfectos que deben ser corregidos, obtener una captura de algún área de interés del objeto o modificar la cantidad de triángulos que componen el modelo con la finalidad de obtener un acabado que se adecue a las necesidades del usuario final.

- El software desarrollado fue capaz de superar cada una de las evaluaciones que se le hicieron para comprobar su funcionamiento, cumpliendo con el comportamiento esperado por cada una de las etapas, lo que permite asegurar que el sistema puede ser usado en un ambiente de producción industrial.
- Imperfectos como exceso de luz o una mala posición de la cámara afectan directamente a la adquisición de datos y su efecto se extiende a lo largo de todo el sistema, dando como resultado objetos mal modelados con errores completamente visibles y que no son aptos para ser usados en la industria.
- Los objetos modelados y exportados pueden ser aplicados a distintas áreas de interés, estableciendo un preprocesamiento, en caso de ser necesario, debido a presencia de ruido que puede estar presente en los modelos finales generados, de tal forma que se obtenga un objeto virtual 3D que pueda ser medido, visualizado en escenarios o impreso tridimensionalmente.

## **4.2 Recomendaciones**

- El software fue construido en el lenguaje de programación Python utilizando librerías que fueron compiladas en lenguaje C/C++ (para un mejor desempeño). Python es un lenguaje muy flexible y fue de gran utilidad para la construcción del software, pero su carácter de ser interpretado hace que no esté optimizado para paralelismo, por lo que se recomienda migrar el software al lenguaje C/C++ donde también están disponibles las principales librerías usadas en el proyecto (SDK de Intel RealSense y Open3D).
- Para usar el software es recomendable tomar las capturas del objeto de forma adecuada, utilizando iluminación difusa la cual es homogénea y suave dando una luminosidad precisa, en un ambiente sin presencia de ruido, ubicándolo a la distancia configurada, estableciendo una resolución de acuerdo con el tamaño del objeto y haciendo la mayor cantidad de capturas posibles. De tal

forma que obtenga imágenes que sean adecuadas para obtener un modelamiento correcto y sin problemas.

- Un método que puede ayudar a mejorar el proceso de adquisición de datos 3D es usar una mesa rotatoria que se encuentre vinculada al sistema de tal forma que se pueda regular los giros que se le realiza al objeto, estableciendo un ángulo de giro. La adición de este implemento sería de gran utilidad para tener capturas uniformes con giros controlados, que garantizaría obtener resultados con el acabado deseado ya sea fino con ángulos más pequeños o grueso con ángulos más grandes.

### **4.3 Futuros trabajos**

El presente proyecto puede ser extrapolable a situaciones distintas a las mencionadas en este documento, con la intención de mejorar aspectos como la adquisición y el modelamiento, a continuación, se describen posibles nuevas funcionalidades y pruebas que se realizaron con la intención de demostrar la capacidad de adaptabilidad del proyecto:

#### **4.3.1 Adquisición con diferentes cámaras tridimensionales**

En el mercado hay disponibilidad de varios tipos de cámaras tridimensionales, entre las cuales destacan la familia de cámaras 3D Intel RealSense: Depth 435i (utilizada en este proyecto) y Lidar 515, con esta última se han realizado diferentes pruebas que permiten demostrar que es posible utilizar diferentes cámaras tridimensionales para el módulo de adquisición.

La cámara Intel RealSense Lidar 515, tiene un funcionamiento diferente en comparación con la Intel RealSense Depth 435i ya que trabaja con información obtenida por medio de láser para una mejor precisión, resolución y detalle en los objetos.

Las capturas 3D adquiridas con la cámara Lidar 515 para ser procesadas exigen un poder computacional superior en comparación con las capturas obtenidas por medio de la cámara Depth 435i. Las primeras pruebas realizadas con respecto al modelamiento del objeto 3D arrojaron resultados muy imprecisos, por lo que fue necesario modificar los parámetros de modelamiento, obteniendo así mejores

resultados, alcanzando un nivel cercano a la realidad y en otras ocasiones superior a los obtenidos a partir de la cámara principal utilizada en el proyecto (Figura 4.1).

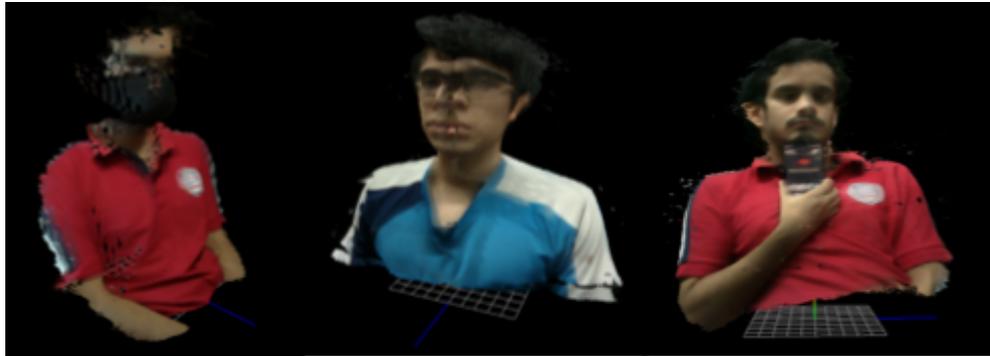


Figura 4.1 Resultados obtenidos a partir de la cámara Lidar 515.

#### 4.3.2 ROI en adquisición

Actualmente el proyecto realiza un preprocesamiento a cada captura obtenida en el módulo de adquisición, eliminando por completo el fondo, con la intención de eliminar ruido que pueda afectar el procesamiento de la imagen. Se plantea la posibilidad de incluir una opción que permita seleccionar una región de interés o en inglés *region of interest (ROI)*. Esto permitiría eliminar el ruido que no se encuentra en el fondo, sino que se encuentra dentro del rango de captura elegido previamente (Figura 4.2 y Figura 4.3).

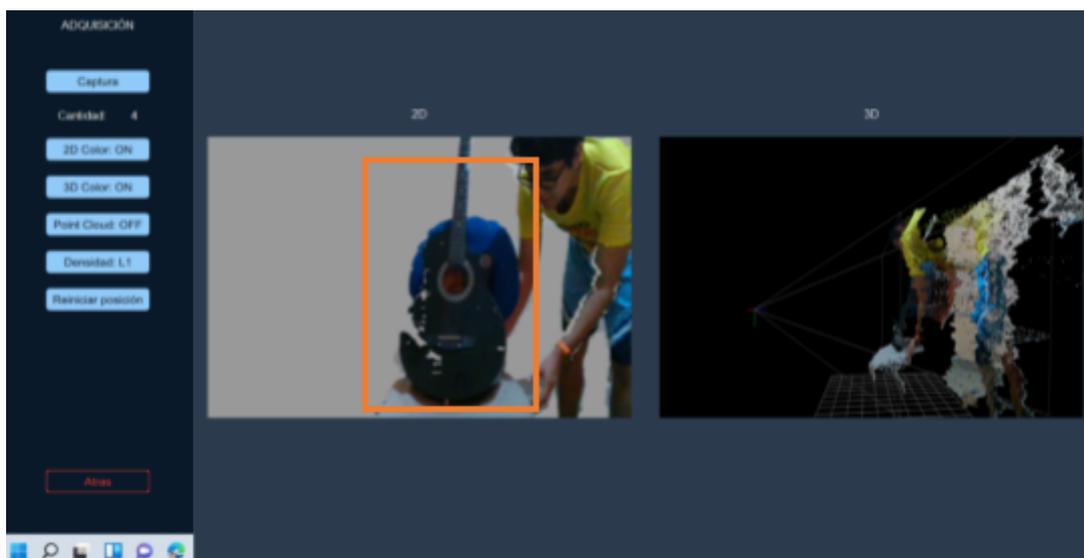


Figura 4.2 Selección de ROI en adquisición.

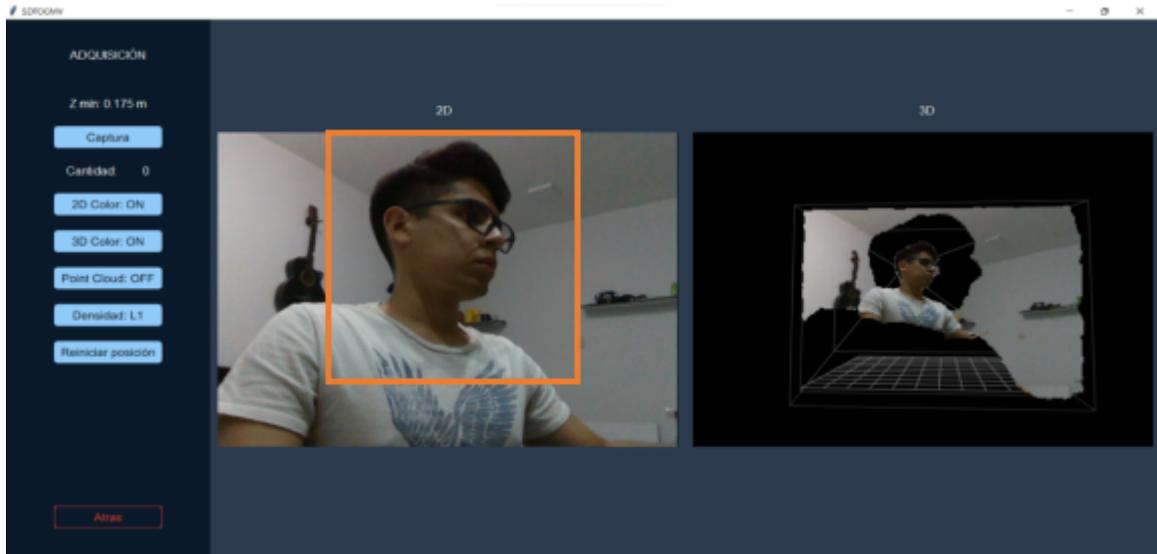


Figura 4.3 Selección de ROI en adquisición 2.

#### 4.3.3 Modificación de la malla triangular para eliminar ruido

Esta opción de forma similar que el punto anterior “ROI en adquisición”, está pensada para ayudar a eliminar el ruido presente en el modelo resultante. La propuesta consiste en escoger la región de interés, por medio del módulo de visualización para así editar las mallas triangulares resultantes (Figura 4.4).

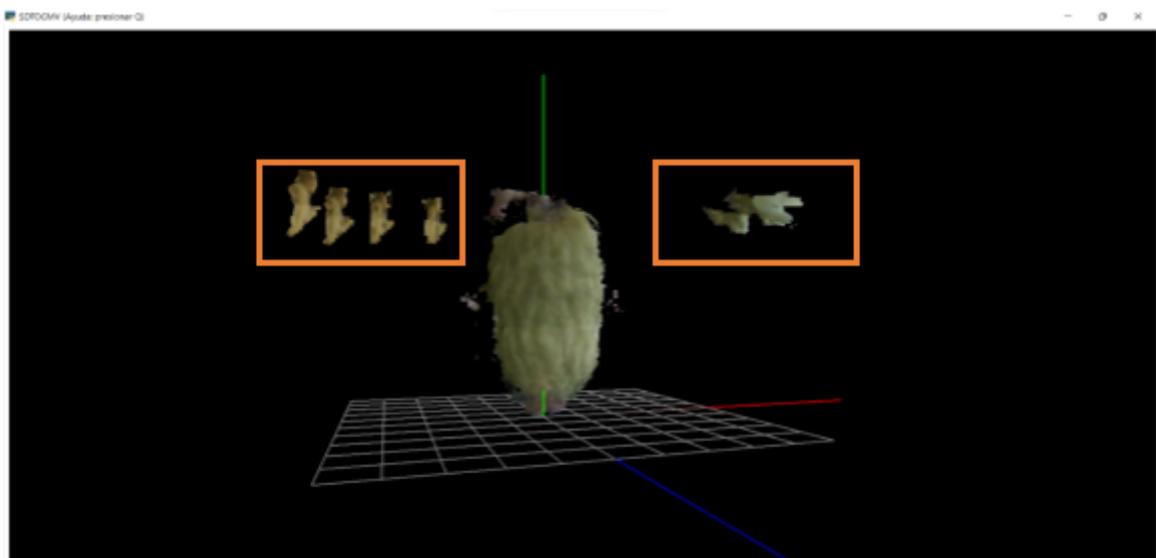


Figura 4.4 Selección de ROI en visualización.

#### 4.3.4 Capturas frames a frames del video streaming

Implementar una opción que permita almacenar capturas frames a frames conforme se visualiza el video streaming, desde un determinado momento hasta detener este proceso (Figura 4.5).

Este procedimiento mejoraría la interacción con el usuario y aumentaría la cantidad de frames almacenados aumentando así la precisión del modelamiento, pero sería necesario aumentar el poder computacional para procesar todos los frames.

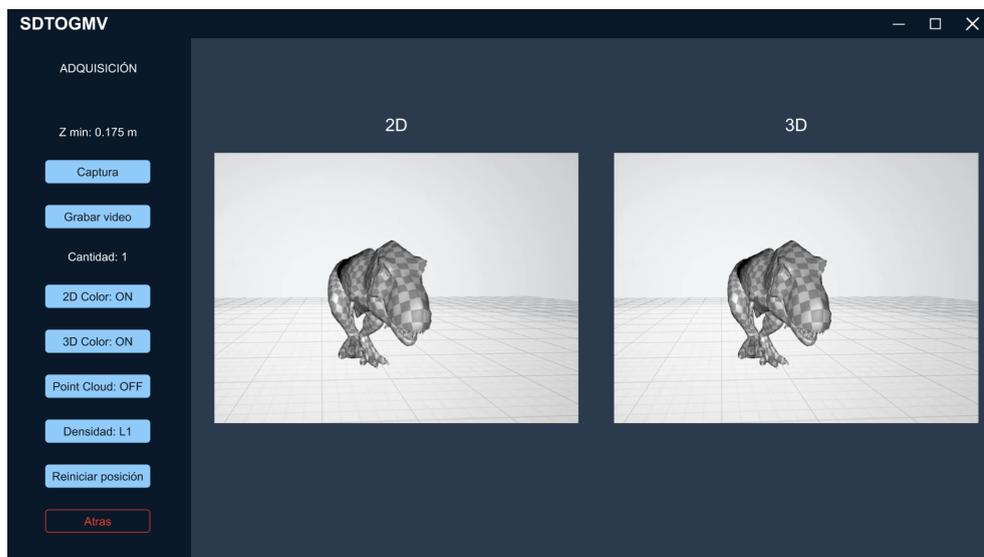
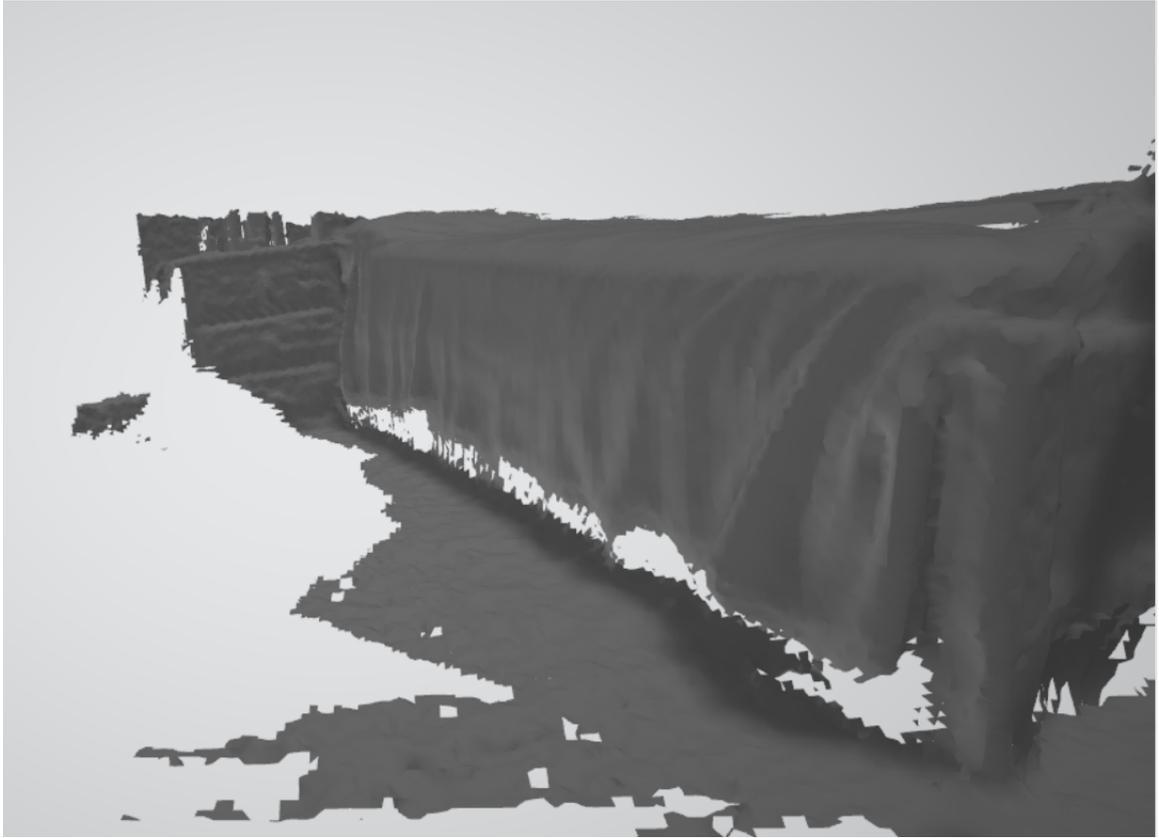


Figura 4.5 Opción de grabar video.

#### 4.3.5 Extrapolable a escenarios

El modelamiento de escenarios es una mejora posible, pero requiere un mayor poder computacional, ya que el área a capturar y modelar sería mucho mayor y por lo tanto aumenta también el tamaño de la información capturada como la cantidad de frames.

El proyecto es extrapolable a escenarios, las pruebas realizadas dieron resultados que presentan imperfecciones (Figura 4.6), las cuales pueden ser corregidas a partir de modificaciones en los parámetros de modelamiento. El proyecto "*Colored Point Cloud Registration Revisited Supplementary Material*" [25] demuestra en sus resultados que es posible modelar escenarios de forma completa (Figura 4.7).



**Figura 4.6 Modelamiento con imperfecciones.**



**Figura 4.7 Modelamiento sin imperfecciones.**

# BIBLIOGRAFÍA

- [1] L. C. Howard, "3D Natives," 4 Noviembre 2019. [Online]. Available: <https://www.3dnatives.com/es/global-3d-printing-report-2019-041120192/#!>
- [2] Corporación Intel, "Intel® RealSense™ Depth and Tracking Cameras," Intel, 2021. [Online]. Available: <https://www.intelrealsense.com/depth-camera-d435i/>. [Accessed 13 Octubre 2021].
- [3] Q.-Y. Z. a. J. P. a. V. Koltun, "Open3D: A Modern Library for 3D Data Processing," arXiv:1801.09847, 2018.
- [4] D. Trapero, "¿Qué formatos se usan en impresión 3D? - Bitfab," 7 Enero 2021. [Online]. Available: <https://bitfab.io/es/blog/formatos-impresion-3d/>. [Accessed 13 Octubre 2021].
- [5] Intel, "Intel® RealSense Product Family D400 Series," *Intel RealSense*, p. 134, 2021.
- [6] IntelRealSense, "Intel® RealSense™ SDK 2.0 and Open3D," [Online]. Available: <https://www.intelrealsense.com/open3d/>. [Accessed 17 10 2021].
- [7] Tchapmi, Lyne P and Kosaraju and Vineet and Rezatofig, Topnet: Structural point cloud decoder. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
- [8] YodaYoda, "Medium," 23 09 2020. [Online]. Available: <https://medium.com/yodayoda/from-depth-map-to-point-cloud-7473721d3f>. [Accessed 17 10 2021].
- [9] Intel, "IntelRealSense Developers," [Online]. Available: <https://www.intelrealsense.com/developers/>. [Accessed 08 19 2021].
- [10] I. BLOCH and H. MAITRE, "Data fusion in 2D and 3D image processing: an overview," in *Proceedings X Brazilian Symposium on Computer Graphics and Image Processing*, París, 1997, pp. 127-134.

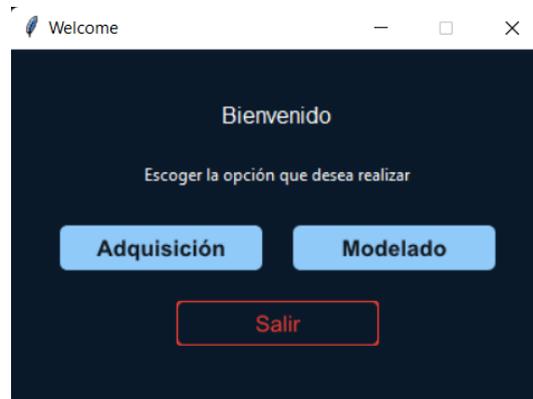
- [11] Open3D, "Open3D: A Modern Library for 3D Data Processing," 2020. [Online]. Available: <http://www.open3d.org/docs/0.12.0/index.html>. [Accessed 31 Octubre 2021].
- [12] G. Martín, "adictosaltrabajo," 30 06 2020. [Online]. Available: <https://www.adictosaltrabajo.com/2020/06/30/interfaces-graficas-en-python-con-tkinter/>. [Accessed 18 10 2021].
- [13] M. Ebrahim, "Likegeeks," 10 01 2019. [Online]. Available: <https://likegeeks.com/es/ejemplos-de-la-gui-de-python/>. [Accessed 19 10 2021].
- [14] K. McHenry and P. Bajcsy, "An Overview of 3D Data Content, File Formats and Viewers," Champaign, 2008.
- [15] P. Bourke, "Data Formats: 3D, Audio, Image," 2021. [Online]. Available: <http://paulbourke.net/dataformats/>. [Accessed 20 10 2021].
- [16] M. Burns, "El formato StL," 1989. [Online]. Available: [http://www.fabbers.com/tech/STL\\_Format](http://www.fabbers.com/tech/STL_Format). [Accessed 20 12 2021].
- [17] J. Kočí, "3MF file format and why it's great," 1 11 2019. [Online]. Available: [https://blog.prusaprinters.org/3mf-file-format-and-why-its-great\\_30986/](https://blog.prusaprinters.org/3mf-file-format-and-why-its-great_30986/). [Accessed 20 10 2021].
- [18] IntelRealSense, "GitHub," 2021. [Online]. Available: <https://github.com/IntelRealSense/librealsense/tree/development>. [Accessed 28 Octubre 2021].
- [19] Ritchizh, "Ritchizh/RGBD-Integration-2020: Applying Open3D functions to integrate experimentally measured color and depth frames into a 3D object. (github.com)," GitHub, 08 05 2020. [Online]. Available: <https://github.com/Ritchizh/RGBD-Integration-2020>. [Accessed 17 10 2021].
- [20] M. A. Realpe, B. Vintimilla, J. Gaspar and J. Santos-Victor, "Localización y Mapeo Simultáneo con Imágenes 3D a Color," *Revista Tecnológica - ESPOL*, vol. 19, no. 1, pp. 99-100, 2013.
- [21] A. Sappa, N. Aifanti, S. Malassiotis and M. G. Strintzis, "Monocular 3D human body reconstruction towards depth augmentation of television sequences," *IEEE*, vol. 3, no. 4, pp. III-325, 2003.

- [22] Rad-hi, "GitHub," 11 11 2020. [Online]. Available: <https://github.com/Rad-hi/3D-Rendering-Desktop-App>. [Accessed 24 10 2021].
- [23] F. Poux, "Genere mallas 3D desde nubes de puntos con Python | Hacia la ciencia de datos," 21 Abril 2020. [Online]. Available: <https://towardsdatascience.com/5-step-guide-to-generate-3d-meshes-from-point-clouds-with-python-36bad397d8ba>. [Accessed 27 10 2021].
- [24] X. Huang, J. Zhang, Q. Wu, L. Fan and C. Yuan, "A Coarse-to-Fine Algorithm for Registration in 3D Street-View Cross-Source Point Clouds," *IEEE*, no. 6, pp. 1--6, 2016.

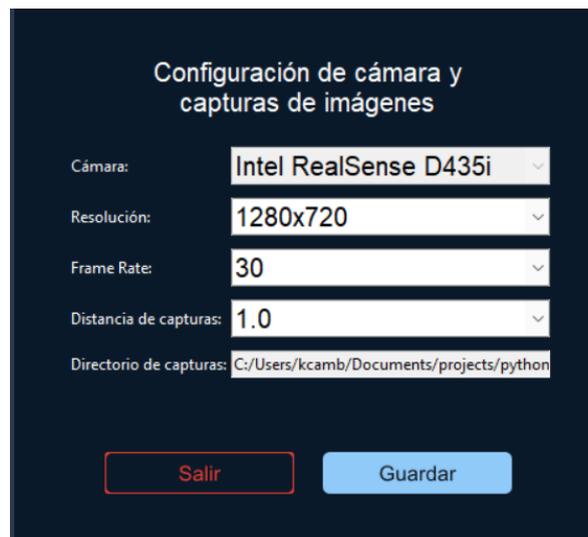
# APÉNDICES

# APÉNDICE A

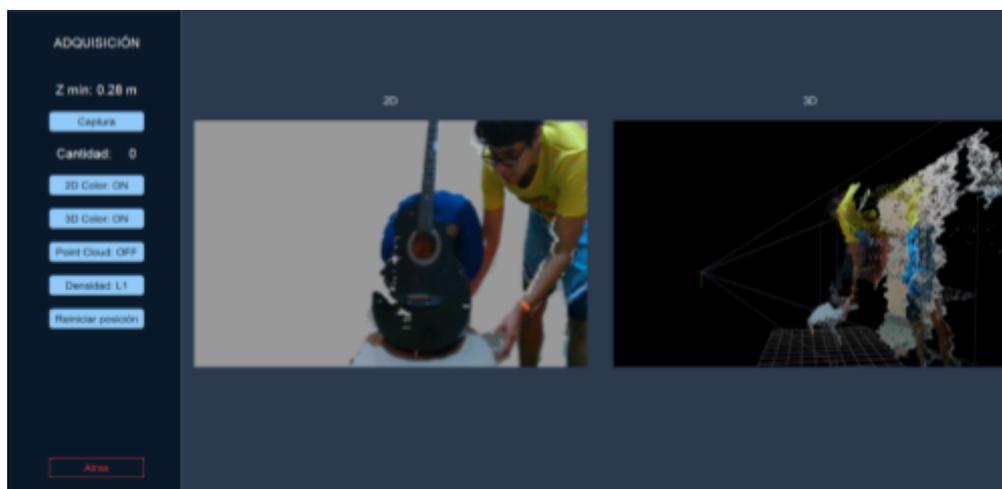
Conjunto de pantallas resultado del desarrollo del prototipo



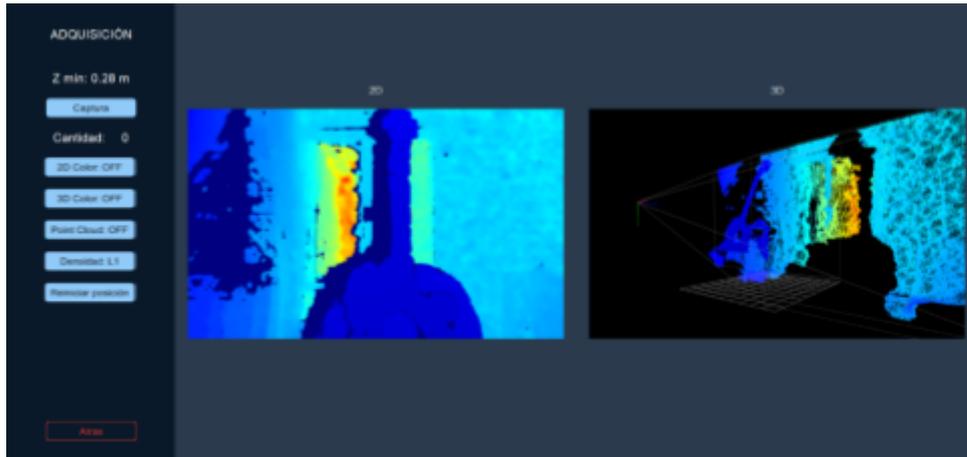
Pantalla de bienvenida



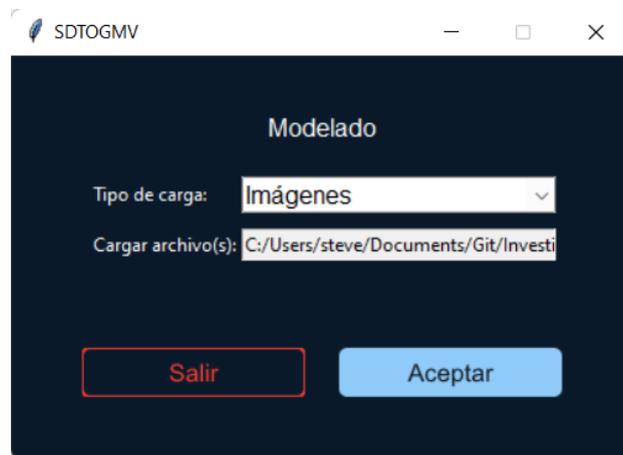
Pantalla de configuración de cámara tridimensional



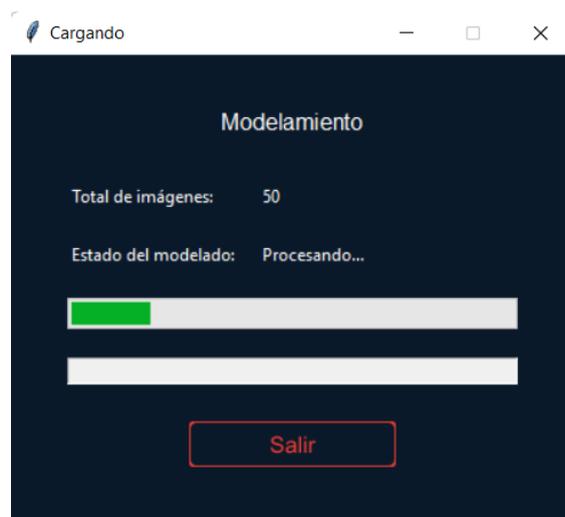
Pantalla de adquisición



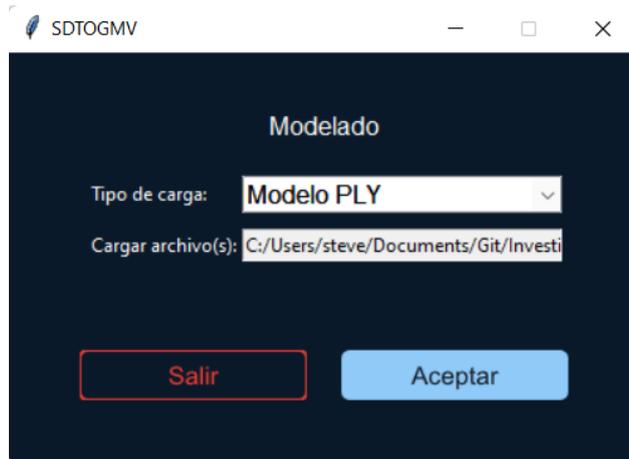
Pantalla de adquisición modificada con opciones de configuración



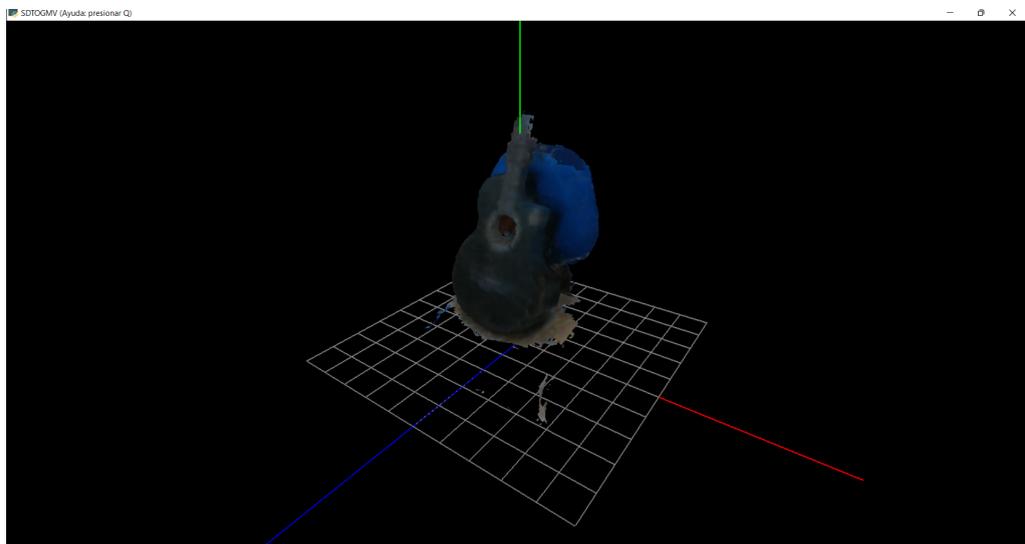
Pantalla de selección de modelado



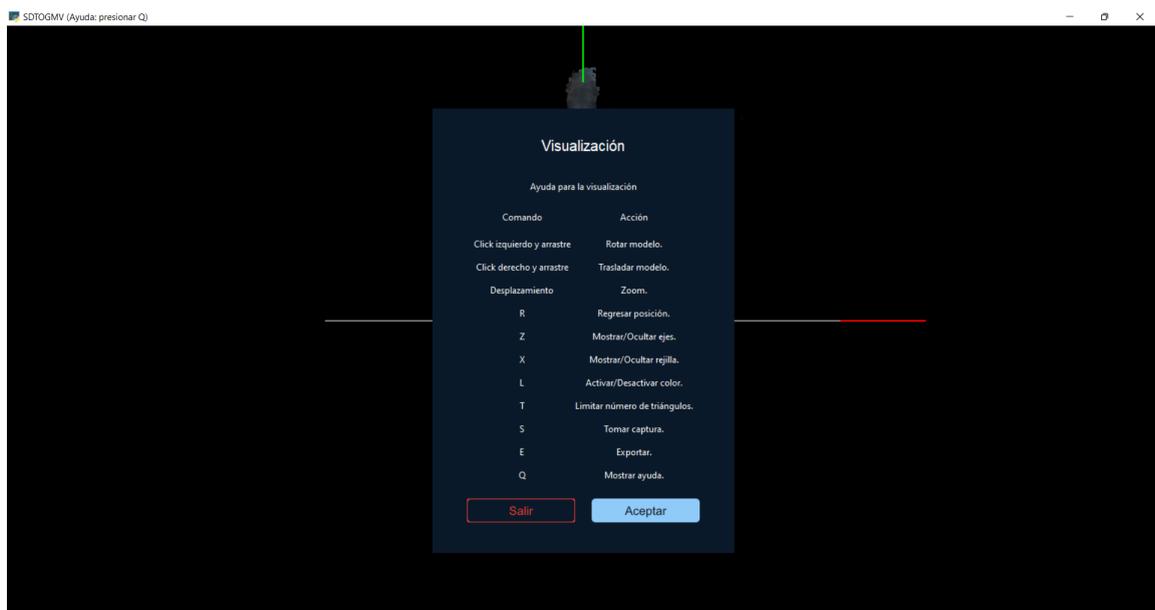
Modelamiento en proceso



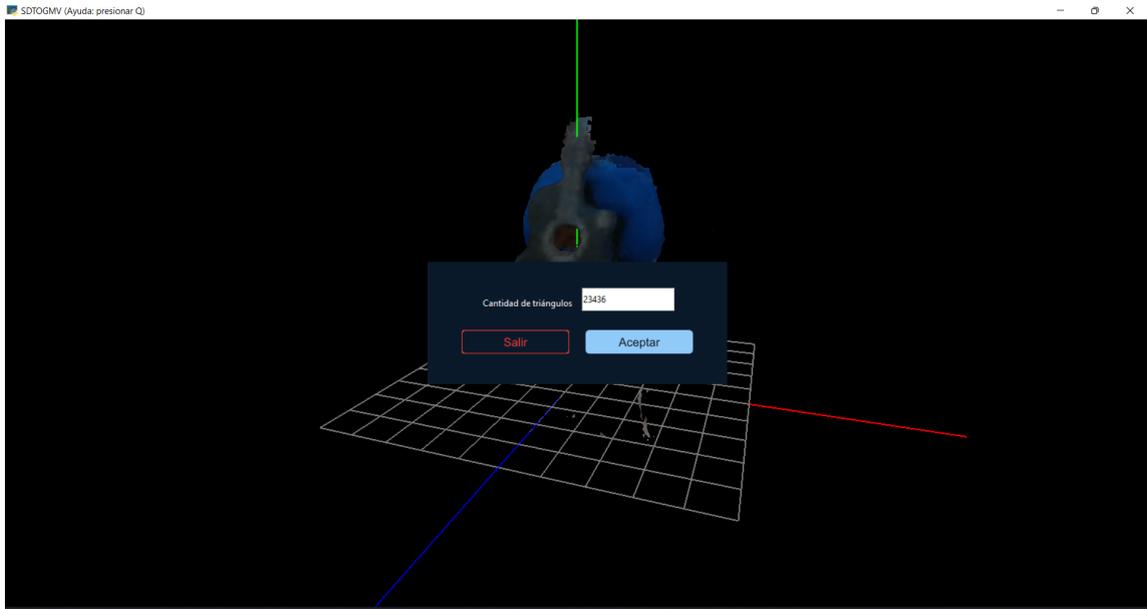
Selección de visualización



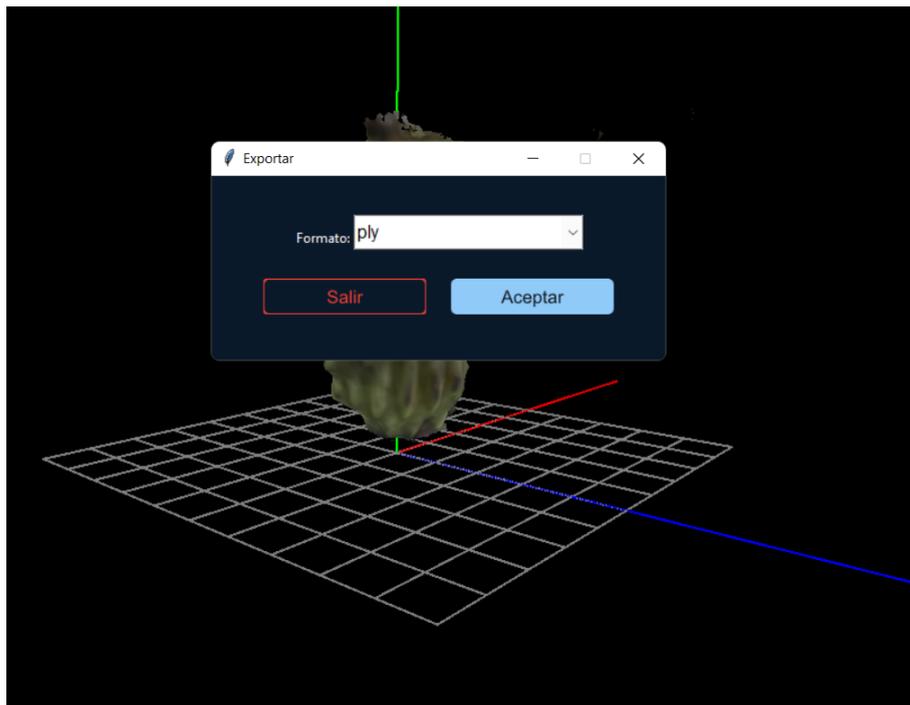
Visualización del modelo resultante



Opciones de ayuda en visualización



Modificación de mallas triangulares



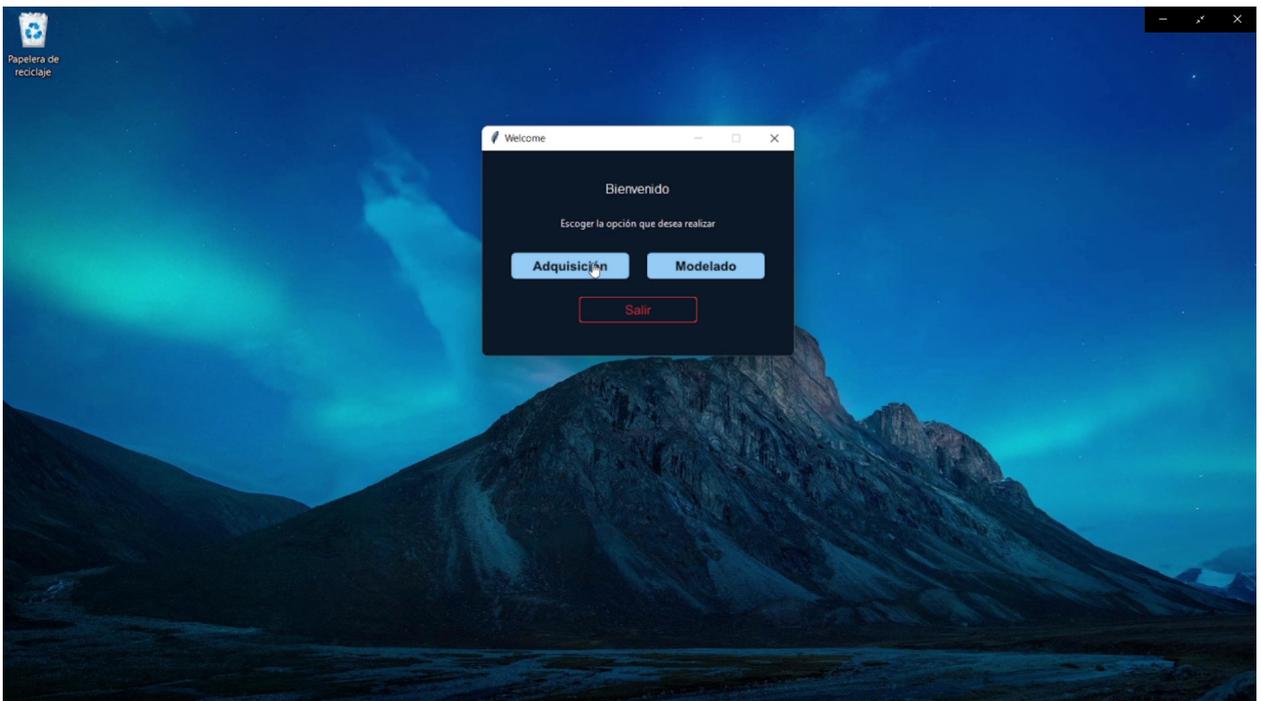
Selección de formato a exportar

# APÉNDICE B

## Videos explicativos de la solución



Tutorial de uso: [bit.ly/3deltatutorial](https://bit.ly/3deltatutorial)



Demostración rápida: [bit.ly/3deltademo](https://bit.ly/3deltademo)