



Implementación de un sistema para la reconstrucción de escenas en 3 dimensiones a partir de imágenes en 2 dimensiones mediante dispositivos móviles

Modalidad: Desarrollo Tecnológico

Samuel Raúl Alemán Neira  
CC: 1097182445

**UNIDADES TECNOLÓGICAS DE SANTANDER**  
**Facultad de Ciencias Naturales e Ingenierías**  
Tecnólogo en Desarrollo de Sistemas Informáticos  
**Bucaramanga 7-abril.2026**



Implementación de un sistema para la reconstrucción de escenas en 3 dimensiones a partir de imágenes en 2 dimensiones mediante dispositivos móviles

Modalidad: Desarrollo Tecnológico

Samuel Raúl Alemán Neira  
CC: 1097182445

**Trabajo de Grado para optar al título de  
Tecnólogo en Desarrollo de Sistemas Informáticos**

**DIRECTOR**

Ph.D. Hernán Gonzalez Acuña

Grupo de Investigación en Ingeniería del Software – GRIIS

**UNIDADES TECNOLÓGICAS DE SANTANDER  
Facultad de Ciencias Naturales e Ingenierías  
Tecnólogo en Desarrollo de Sistemas Informáticos  
Bucaramanga 7-abril.2026**

Nota de Aceptación

Aprobado en cumplimiento de requisitos por las  
Unidades Tecnológicas de Santander - UTS,  
para optar al título de Tecnólogo en Desarrollo  
de Sistemas informáticos, según acta de comité  
de trabajos de grado No 7 del 07 de abril de 2026  
Docente evaluador: M.Sc. Kevin Jair Castellanos  
Gonzalez. Docente director: Ph.D. Hernan  
Gonzalez Acuña



---

Firma del Evaluador



---

Firma del Director

## DEDICATORIA

A mi madre, Johana Milena Neira Mendez. En más de un momento, cuando avanzar costaba más de lo esperado, fue su voz la que sostuvo el paso. No siempre con grandes palabras, a veces solo con su manera de estar, incluso cuando yo mismo no tenía del todo claro si iba a lograrlo.

No puedo dejar de mencionar a las Unidades Tecnológicas de Santander (UTS). Fue allí donde esta idea dejó de ser algo suelto y empezó a tomar forma, entre intentos, errores y ajustes que Durante la implementación se presentaron dificultades iniciales. Por otra parte, a quienes estuvieron cerca amigos, compañeros.

Y, al final, también a mí mismo. Por insistir. Por esas noches largas donde avanzar no era tan sencillo, y aun así seguir. No hay nada extraordinario en eso, salvo la decisión repetida de no abandonar el camino hasta llegar aquí.

## AGRADECIMIENTOS

A las Unidades Tecnológicas de Santander (UTS). Más que un lugar de estudio, fue el punto donde todo esto empezó a tomar sentido, incluso en esos días en los que nada parecía avanzar como uno esperaba.

No puedo dejar de mencionar al director del proyecto; en varios momentos, cuando las ideas no encajaban o simplemente no había claridad, su orientación ayudó a retomar el rumbo sin tanto rodeo.

Y, por otra parte, a quienes estuvieron cerca. Algunos con palabras, otros con pequeños gestos, otros simplemente estando ahí. No siempre se dice, pero todo eso suma.

## TABLA DE CONTENIDO

<b><u>RESUMEN EJECUTIVO</u></b> .....	<b><u>10</u></b>
<b><u>INTRODUCCIÓN</u></b> .....	<b><u>11</u></b>
<b><u>1. DESCRIPCIÓN DEL TRABAJO DE INVESTIGACIÓN</u></b> .....	<b><u>13</u></b>
1.1. PLANTEAMIENTO DEL PROBLEMA .....	13
1.2. JUSTIFICACIÓN .....	15
1.3. OBJETIVOS .....	17
1.3.1. OBJETIVO GENERAL .....	17
1.3.2. OBJETIVOS ESPECÍFICOS .....	17
1.4. ESTADO DEL ARTE .....	17
<b><u>2. MARCO REFERENCIAL</u></b> .....	<b><u>20</u></b>
2.1. MARCO TEÓRICO .....	20
2.2. MARCO CONCEPTUAL .....	21
2.3. MARCO LEGAL .....	24
2.4. MARCO AMBIENTAL .....	24
<b><u>3. DISEÑO DE LA INVESTIGACIÓN</u></b> .....	<b><u>25</u></b>
3.1. FLUJO DEL SISTEMA .....	26
3.2. FASE 1: ANÁLISIS DE REQUERIMIENTOS .....	27
3.3. FASE 2: DISEÑO DEL SISTEMA .....	28
3.4. FASE 3: DESARROLLO E IMPLEMENTACIÓN .....	30
3.5. FASE 4: PRUEBAS Y VALIDACIÓN .....	31
3.6. FASE 5: ANÁLISIS DE RESULTADOS .....	32
<b><u>4. DESARROLLO DEL TRABAJO DE GRADO</u></b> .....	<b><u>33</u></b>
4.1. DISEÑO DEL SISTEMA .....	33
4.2. DISEÑO DE LA APLICACIÓN MÓVIL .....	34
4.3. DISEÑO DEL SISTEMA DE PROCESAMIENTO .....	34
4.4. IMPLEMENTACIÓN DEL SISTEMA .....	35
4.5. PRUEBAS DEL SISTEMA .....	36
4.6. LIMITACIONES DEL SISTEMA .....	38

<b>5.</b>	<b><u>RESULTADOS</u></b> .....	<b>39</b>
5.1.	RESULTADOS GENERALES DE LAS PRUEBAS EXPERIMENTALES.....	39
5.2.	ANÁLISIS SEGÚN NÚMERO DE IMÁGENES .....	41
5.3.	ANÁLISIS SEGÚN CONDICIONES DE CAPTURA .....	42
5.4.	ANÁLISIS DEL TIEMPO DE PROCESAMIENTO.....	42
5.5.	CAUSAS TÉCNICAS DE FALLO.....	43
5.6.	DISCUSIÓN.....	44
5.7.	DE RESULTADOS .....	44
5.8.	CUMPLIMIENTO DE LOS OBJETIVOS .....	44
5.9.	EVIDENCIA VISUAL DEL SISTEMA .....	46
<b>6.</b>	<b><u>CONCLUSIONES</u></b> .....	<b>55</b>
<b>7.</b>	<b><u>RECOMENDACIONES</u></b> .....	<b>56</b>
<b>8.</b>	<b><u>REFERENCIAS BIBLIOGRÁFICAS</u></b> .....	<b>58</b>
<b>9.</b>	<b><u>ANEXOS</u></b> .....	<b>60</b>
9.1.	HISTORIAL DE ERRORES Y FALLOS DEL SISTEMA DE PROCESAMIENTO .....	62
9.2.	ANÁLISIS GENERAL .....	65
9.3.	MANUAL DE USUARIO.....	66

## LISTA DE FIGURAS

<b>FIGURA 3.1 - Arquitectura del sistema de reconstrucción 3D .....</b>	<b>26</b>
<b>FIGURA 5.1 - Captura del objeto desde diferentes perspectivas .....</b>	<b>46</b>
<b>FIGURA 5.2 - Interfaz de captura guiada con control de ángulo y nivel .....</b>	<b>48</b>
<b>FIGURA 5.3 – Validación del lote de imágenes y cobertura alcanzada .....</b>	<b>49</b>
<b>FIGURA 5.4 – Gestión de proyectos y estado de capturas en la aplicación.....</b>	<b>50</b>
<b>FIGURA 5.5 - Configuración de conexión con <i>backend</i> local .....</b>	<b>51</b>
<b>FIGURA 5.6 - Registro de error de conexión durante el procesamiento.....</b>	<b>52</b>
<b>FIGURA 5.7 - Ejecución exitosa del pipeline y generación del modelo .....</b>	<b>53</b>
<b>FIGURA 5.8 - Visualización del modelo 3D generado en formato GLB.....</b>	<b>54</b>
<b>Figura A 10.1 - Flujo de navegación del sistema.....</b>	<b>60</b>
<b>Figura B 10.1 - Arquitectura del sistema de reconstrucción 3D.....</b>	<b>60</b>
<b>Figura C 10.1 - Estructura funcional del sistema .....</b>	<b>61</b>
<b>FIGURA M-U 1</b>	<b>67</b>
<b>FIGURA M-U 2</b>	<b>68</b>
<b>FIGURA M-U 3</b>	<b>69</b>
<b>FIGURA M-U 4</b>	<b>70</b>
<b>FIGURA M-U 5</b>	<b>71</b>

## LISTA DE TABLAS

<b>Tabla 1 - Resultados de las pruebas experimentales .....</b>	<b>40</b>
<b>Tabla 2 - Resumen estadístico de resultados.....</b>	<b>41</b>
<b>Tabla 3 - Resultados del sistema según número de imágenes .....</b>	<b>41</b>
<b>Tabla 4 - Distribución de fallos del sistema .....</b>	<b>43</b>

## RESUMEN EJECUTIVO

Llevar imágenes planas a un entorno tridimensional —ese fue el núcleo del proyecto—, pero no desde un laboratorio ideal, sino usando lo que se tiene a mano: un dispositivo móvil. El objetivo principal fue implementar un sistema de reconstrucción 3D accesible, buscando democratizar una tecnología que, por sus altos costos, suele estar fuera del alcance en contextos académicos de bajos recursos. No se trata solo de capturar datos, sino de ofrecer una alternativa real a los escáneres tradicionales.

La metodología se planteó desde un enfoque de desarrollo tecnológico por fases. Primero, “se inició el desarrollo desde cero: entender los requerimientos y luego conectar las piezas. Se logró articular una aplicación en Flutter para la captura de fotos y un backend robusto basado en COLMAP para procesar la visión por computador. Aunque en teoría el proceso es directo, en la ejecución, integrar el flujo de datos entre el móvil y el servidor requirió ajustes constantes para estabilizar la reconstrucción multi-vista.

Como resultado, se obtuvo un prototipo funcional que ejecuta el flujo completo: desde la adquisición de imágenes hasta la generación del modelo. Sin embargo, el hardware fue el límite. En escenas con poca textura o luz no controlada, la calidad bajaba notablemente. No fue un fallo, sino una constante que obligó a entender mejor el comportamiento real del sistema bajo presión computacional.

Se concluye que, aunque no es una solución perfecta —todavía no—, el camino es viable. El sistema valida que es posible acercar la reconstrucción 3D a entornos cotidianos sin depender de equipos especializados, evidenciando que el potencial de los dispositivos móviles en la ingeniería está apenas comenzando a explotarse.

**PALABRAS CLAVE:** Reconstrucción 3D, Flutter, COLMAP, Fotogrametría, Visión-Artificial.

## INTRODUCCIÓN

La reconstrucción tridimensional a partir de imágenes bidimensionales no es una idea nueva. Durante años ha encontrado aplicación en campos como la ingeniería, la medicina y la educación. Aun así, llevarla a la práctica sigue siendo un reto en muchos contextos, principalmente por temas de acceso. Los escáneres 3D, aunque ofrecen resultados de alta precisión, implican costos elevados y requieren condiciones técnicas que no siempre están disponibles en entornos académicos con recursos limitados (Luhmann et al., 2019).

Frente a esta limitación, han ido ganando terreno alternativas basadas en el uso de imágenes convencionales. Métodos como Structure from Motion (SfM) y Multi-View Stereo (MVS) permiten reconstruir geometría tridimensional a partir de múltiples fotografías tomadas desde distintos ángulos (Ma et al., 2012). En teoría, estos enfoques están bien sustentados; en la práctica, su implementación suele enfrentarse a dificultades relacionadas con la calidad de las imágenes, el nivel de traslape entre capturas y las condiciones de iluminación.

Más recientemente, el avance tecnológico ha dado paso a propuestas como los campos de radiancia neuronal (NeRF), capaces de generar representaciones con un alto nivel de realismo (Tewari et al., 2021). Sin embargo, este tipo de soluciones demanda una capacidad de procesamiento considerable, lo que limita su uso en escenarios donde el hardware es reducido o poco especializado.

En medio de este panorama, los dispositivos móviles empiezan a jugar un papel interesante. Diversos trabajos han mostrado que los teléfonos inteligentes pueden utilizarse como herramientas de captura en procesos de fotogrametría, logrando resultados aceptables frente a sistemas más robustos, aunque con ciertas limitaciones en precisión y estabilidad (Seifert et al., 2024; Karabiber et al., 2025). Esto abre una posibilidad relevante: trasladar el problema desde equipos especializados hacia tecnologías más accesibles.

A pesar de estos avances, siguen existiendo barreras importantes. La complejidad del proceso y el costo computacional no desaparecen con el cambio de dispositivo. Generar un modelo tridimensional va mucho más allá de tomar fotografías; implica una serie de etapas intensivas en procesamiento que no siempre pueden ejecutarse de manera eficiente en contextos académicos.

A partir de esta problemática, el presente trabajo se orienta al desarrollo de un sistema pensado para condiciones reales de uso. Se implementó una solución que integra una aplicación móvil para la captura de imágenes y un backend encargado del procesamiento, apoyado en técnicas de visión por computador. El objetivo fue ejecutar el flujo completo de reconstrucción, desde la adquisición de los datos hasta la generación del modelo tridimensional.

Durante el desarrollo se hicieron evidentes varias limitaciones. Las restricciones de hardware influyeron directamente en el desempeño del sistema, y en distintos casos las reconstrucciones no alcanzaron los resultados esperados. Factores como la baja calidad de las imágenes o la falta de características visuales suficientes afectaron el proceso. Estas situaciones no se interpretan como fallos aislados, sino como parte del comportamiento del sistema cuando se enfrenta a condiciones no controladas.

El enfoque adoptado permitió no solo construir una solución funcional, sino también comprender mejor sus alcances y limitaciones. A lo largo del desarrollo se emplearon herramientas de asistencia basadas en inteligencia artificial como apoyo en tareas de depuración y optimización del código, siempre bajo validación manual. Más que presentar un sistema ideal, el trabajo busca mostrar hasta qué punto este tipo de soluciones puede ser viable en contextos con restricciones. En las secciones siguientes se describe el proceso de desarrollo, los resultados obtenidos y los principales aprendizajes derivados de la implementación.

## 1. DESCRIPCIÓN DEL TRABAJO DE INVESTIGACIÓN

### 1.1. PLANTEAMIENTO DEL PROBLEMA

La reconstrucción 3D representa una solución atractiva. En la práctica, no siempre es tan fácil. Aunque hoy los modelos tridimensionales aparecen en áreas tan distintas como la educación, la medicina, la ingeniería o el diseño, su generación sigue estando muy atada a equipos especializados que no cualquiera puede comprar ni operar con facilidad. Los escáneres 3D, por ejemplo, ofrecen buenos resultados, pero su costo y las exigencias técnicas que traen detrás los convierten en una opción lejana para buena parte de la población (Heredia-Lidón et al., 2025).

En el ámbito académico el choque se siente más. Estudiantes, docentes e investigadores se encuentran con una barrera doble: por un lado, la falta de acceso a este tipo de tecnología; por otro, la dificultad de sostener proyectos que además exigen equipos con buena capacidad de procesamiento. La realidad es que no todos cuentan con estaciones de trabajo potentes ni con presupuesto para herramientas de alto costo. Y fuera de la universidad pasa algo parecido: emprendedores y desarrolladores independientes también terminan frenados por esas mismas trabas económicas y técnicas.

El problema no nace de una sola causa. Pesa el precio de los dispositivos de escaneo, claro, pero también la necesidad de hardware capaz de soportar el procesamiento y la complejidad de los algoritmos que intervienen en la reconstrucción 3D. A eso se suma otra dificultad menos visible, aunque igual de seria: la mayoría de soluciones pensadas para este tipo de tareas no están hechas para el usuario común. Muchas requieren experiencia previa, configuraciones poco intuitivas o conocimientos que, en escenarios de formación tecnológica, no siempre están disponibles desde el inicio.

¿El resultado? Menos acceso. Menos experimentación. Menos proyectos que realmente puedan aterrizar en contextos educativos. Cuando una tecnología depende de equipos

caros, software complejo y recursos que no están al alcance de todos, termina convirtiéndose en algo distante, casi reservado para laboratorios muy bien dotados o instituciones con mayores capacidades. En lugares donde los recursos son limitados, como ocurre con frecuencia en escenarios académicos, eso golpea directamente la posibilidad de aprender haciendo.

Y aunque existe software profesional para generar modelos 3D, ahí aparece otra dificultad. No basta con tener la herramienta; hace falta saber usarla, configurarla y sostener el procesamiento que exige. Para usuarios sin experiencia técnica, la curva de entrada suele ser alta. Demasiado alta.

Por eso la necesidad no apunta simplemente a crear otra aplicación, sino a aterrizar una solución que acerque esta tecnología a dispositivos que ya están en manos de la mayoría, como los teléfonos móviles. La apuesta, entonces, es clara: aprovechar la captura de imágenes desde equipos accesibles y apoyarse en técnicas de visión por computador para democratizar, al menos en parte, el acceso a la reconstrucción 3D en escenarios reales y con limitaciones reales.

## 1.2. JUSTIFICACIÓN

El principal obstáculo se encuentra en el acceso. La digitalización de objetos en tres dimensiones sigue siendo, en muchos casos, una tarea limitada cuando depende de equipos especializados como los escáneres 3D, cuyo costo y exigencias técnicas dificultan su incorporación en entornos académicos con recursos restringidos (Heredia-Lidón et al., 2025). Esto genera una brecha clara, sobre todo en contextos donde la infraestructura tecnológica no es suficiente para sostener este tipo de procesos.

En el caso de las Unidades Tecnológicas de Santander (UTS), el interés por trabajar con reconstrucción tridimensional va más allá de lo teórico y apunta a su aplicación práctica. Sin embargo, esta intención se enfrenta a limitaciones concretas, como la disponibilidad de hardware adecuado y el uso de herramientas que demandan mayor capacidad de procesamiento. Esto implica que el desarrollo de una solución no depende únicamente del planteamiento conceptual, sino también de su capacidad de ajustarse a las condiciones reales del entorno.

Desde el punto de vista técnico, el proyecto se orienta a llevar técnicas de visión por computador hacia un escenario más accesible, reduciendo la dependencia de equipos especializados. En este contexto, no se plantea una solución perfecta, sino funcional, capaz de integrar la captura de imágenes mediante dispositivos móviles, su procesamiento y la generación de modelos tridimensionales. Durante el proceso se hicieron evidentes varias dificultades, relacionadas con la calidad de los resultados, el ajuste de parámetros y la necesidad de repetir pruebas hasta lograr configuraciones aceptables.

En términos económicos, la propuesta representa una alternativa relevante al sustituir equipos de alto costo por dispositivos ampliamente disponibles, como los teléfonos móviles. Aunque este cambio no elimina por completo las limitaciones técnicas, sí reduce de manera significativa la barrera de acceso, facilitando el uso de estas tecnologías en entornos académicos y de emprendimiento. Esto amplía el alcance de la reconstrucción 3D hacia escenarios donde antes no era viable.

Desde una perspectiva formativa, el desarrollo del proyecto aporta un valor importante al proceso de aprendizaje. Implementar, evaluar y ajustar una solución en condiciones reales permite comprender con mayor profundidad el comportamiento de los algoritmos y las limitaciones del sistema. Este enfoque práctico fortalece la capacidad de análisis y resolución de problemas, competencias clave en la formación en ingeniería de sistemas.

A nivel investigativo, trabajar con técnicas de fotogrametría y reconstrucción por múltiples vistas en condiciones no controladas permite evidenciar cómo factores como el hardware, la iluminación y la calidad de las imágenes influyen directamente en los resultados. Este tipo de escenarios ofrece una visión más realista del comportamiento del sistema, alejándose de condiciones ideales o controladas.

El sistema desarrollado logró ejecutar el flujo completo de reconstrucción, desde la captura de imágenes hasta la generación del modelo tridimensional. Aun así, los resultados mostraron variaciones en su calidad, especialmente en escenas con poca información visual o en condiciones desfavorables. Este comportamiento no se interpreta como fallos aislados, sino como una consecuencia de las limitaciones propias del entorno en el que se implementa la solución.

Para las Unidades Tecnológicas de Santander, este tipo de iniciativas se alinea con el enfoque institucional orientado al desarrollo de soluciones aplicadas con impacto real. Más que obtener resultados ideales, se valora la capacidad de trabajar con restricciones, adaptar tecnologías y proponer soluciones viables dentro de contextos reales.

En síntesis, el proyecto no pretende presentar una solución definitiva, sino mostrar la viabilidad de alternativas accesibles para la reconstrucción tridimensional. La reducción de las barreras tecnológicas, junto con el aprendizaje obtenido durante el proceso de implementación, constituye el principal aporte de este trabajo.

### **1.3. OBJETIVOS**

#### **1.3.1. OBJETIVO GENERAL**

Implementar una aplicación móvil que genere una escena en 3D a partir de imágenes en 2D capturadas con dispositivos móviles, aplicando técnicas de visión por computador.

#### **1.3.2. OBJETIVOS ESPECÍFICOS**

- Determinar los requerimientos técnicos y limitaciones para la captura, procesamiento y visualización de imágenes para la reconstrucción de una escena en 3D.
- Diseñar la interfaz de la aplicación móvil que permita la adquisición de las imágenes para una reconstrucción tridimensional.
- Implementar el algoritmo para la reconstrucción en 3D que procese las imágenes s mediante la técnica de visión por computador seleccionada
- Evaluar la precisión y eficiencia del sistema mediante pruebas experimentales con distintos objetos y condiciones de iluminación controladas.

### **1.4. ESTADO DEL ARTE**

La reconstrucción 3D a partir de imágenes no es un tema nuevo. Lleva años evolucionando. Y, aun así, sigue dejando preguntas abiertas. La geometría sola ya no basta. Hoy se mezclan enfoques clásicos con modelos más recientes que intentan resolver lo mismo: cómo pasar de múltiples imágenes planas a una representación tridimensional coherente.

Dentro de ese panorama, Structure from Motion (SfM) sigue siendo uno de los puntos de partida más utilizados. Lo que Cuervo et al. (2020) demostraron —trabajando con cámaras no métricas— es que incluso con recursos limitados es posible reconstruir estructuras reales, como el monumento La Rebeca. Los resultados fueron aceptables. Pero hay un detalle: la cobertura de imágenes sigue siendo crítica. Si faltan vistas, el modelo lo paga.

Aquí es donde el panorama cambia. Los métodos basados en aprendizaje profundo comenzaron a empujar los límites. El salto técnico vino con Müller et al. (2022), quienes introdujeron una codificación hash multiresolución que acelera el entrenamiento de modelos tipo NeRF. Menos tiempo de cómputo, resultados de alta calidad. Suena ideal. Pero el costo se traslada a otro lado: la capacidad de procesamiento. Li et al. (2023) siguieron una línea similar, combinando velocidad y precisión en reconstrucción a partir de datos RGB. Funciona. En laboratorio, sobre todo.

¿Y qué pasa fuera de ese escenario controlado? Ahí la historia cambia. Cuando se baja a dispositivos móviles, la reconstrucción 3D deja de ser un problema puramente algorítmico y se convierte en un choque técnico constante. Poór y Fang (2023) lograron reconstruir túneles utilizando múltiples smartphones, con errores mínimos. Es un avance claro. Pero depende de condiciones muy específicas. Lo mismo ocurre con Quispe-Enríquez et al. (2023), quienes obtuvieron modelos craneofaciales precisos usando teléfonos móviles, siempre que se respetaran protocolos estrictos de captura. El detalle está ahí: "siempre que".

Porque en la práctica, esas condiciones no siempre se cumplen. Seifert et al. (2024) lo dejan claro: los smartphones pueden generar modelos aceptables, pero no alcanzan la precisión de sistemas profesionales. Los datos dicen lo contrario a lo que uno quisiera. Y Patonis (2024) añade otro problema: la inestabilidad en los parámetros de las cámaras móviles. Pequeñas variaciones. Grandes impactos en el resultado final.

Más recientemente, trabajos como los de Wang et al. (2024) y Feng et al. (2025) muestran hacia dónde va todo esto: menos dependencia de la geometría clásica y más peso en modelos basados en aprendizaje profundo. Mayor robustez, mejor adaptación a escenas complejas. Pero, otra vez, aparece el mismo obstáculo el hardware.

Al final, el panorama es claro, aunque no del todo cómodo. Hay avances importantes. Sí. Los métodos funcionan. También. Pero la brecha entre lo que se logra en entornos controlados y lo que realmente puede ejecutar un usuario común sigue siendo evidente.

F-DC-125

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE PROYECTO  
DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO, MONOGRAFÍA,  
EMPREDIMIENTO Y SEMINARIO

VERSIÓN: 2.0

Accesibilidad, complejidad de uso y capacidad de procesamiento continúan marcando el límite.

**ELABORADO POR:**  
Docencia

**REVISADO POR:**  
Sistema Integrado de Gestión

**APROBADO POR:** Líder del Sistema Integrado de Gestión  
**FECHA APROBACIÓN:** Octubre de 2023

## 2. MARCO REFERENCIAL

### 2.1. Marco teórico

La reconstrucción tridimensional se fundamenta en los principios de la visión por computador, disciplina orientada al análisis de imágenes digitales mediante la identificación de características, patrones y relaciones espaciales. Desde esta perspectiva, una imagen no se limita a una matriz de píxeles, sino que se interpreta como una fuente de información geométrica susceptible de ser transformada en representaciones tridimensionales. Los planteamientos descritos por Szeliski (2022) establecen la base conceptual de este enfoque. En escenarios reales, su aplicación implica enfrentar condiciones no controladas, entre ellas la presencia de ruido en las imágenes, variaciones en la iluminación y limitaciones en la calidad de los datos.

Dentro de este campo, Structure from Motion (SfM) se posiciona como uno de los enfoques más utilizados. Este método permite estimar de forma conjunta la posición de las cámaras y la estructura tridimensional de una escena a partir de múltiples imágenes. El proceso se apoya en la identificación de correspondencias entre puntos clave presentes en distintas vistas, lo que conduce a la generación de una nube de puntos dispersa que describe la geometría inicial del objeto. Aunque el procedimiento se encuentra ampliamente definido a nivel teórico (Ma et al., 2012), durante las pruebas se observó que su desempeño depende de factores como la calidad de las imágenes, el nivel de traslape entre capturas y la presencia de características visuales distintivas.

Sobre esta reconstrucción inicial se integra la técnica Multi-View Stereo (MVS), orientada a densificar la nube de puntos mediante la estimación de profundidad a partir de múltiples vistas. Este proceso incrementa el nivel de detalle del modelo tridimensional, aunque implica una mayor demanda computacional. En entornos con recursos limitados, como los contextos académicos evaluados, esta etapa representa una de las principales restricciones para la ejecución completa del pipeline.

De forma paralela, se han propuesto enfoques basados en aprendizaje profundo, entre ellos Neural Radiance Fields (NeRF), que permiten modelar tanto la geometría como la apariencia de una escena con altos niveles de detalle (Tewari et al., 2021). A pesar de sus ventajas, estos métodos requieren capacidades de cómputo elevadas, lo que limita su aplicación en escenarios donde no se dispone de hardware especializado.

Para la implementación práctica de estos procesos, herramientas como COLMAP integran las distintas etapas del pipeline de reconstrucción en un flujo de trabajo estructurado. Este tipo de soluciones automatiza tareas como la extracción de características, el emparejamiento de imágenes y la estimación de cámaras. Durante los ensayos realizados se identificó que, aun con este nivel de automatización, el comportamiento del sistema continúa condicionado por factores externos, entre ellos la calidad de las imágenes, el número de capturas y la capacidad de procesamiento disponible.

En el contexto del presente trabajo, estos fundamentos se aplican en la implementación de un sistema de reconstrucción 3D basado en imágenes capturadas mediante dispositivos móviles. Las pruebas realizadas muestran que las limitaciones de hardware y las condiciones de captura influyen de manera directa en el desempeño del pipeline.

A partir de estos fundamentos teóricos, resulta necesario precisar los conceptos técnicos que permiten interpretar el funcionamiento del sistema implementado y los resultados obtenidos durante las pruebas experimentales.

## 2.2. Marco conceptual

Para interpretar el funcionamiento del sistema desarrollado y analizar los resultados obtenidos, se requiere delimitar los conceptos técnicos que intervienen en el proceso de reconstrucción 3D. Estos elementos permiten comprender con mayor precisión el comportamiento del pipeline, las condiciones de captura y las salidas generadas durante las pruebas experimentales.

**Structure from Motion (SfM).**

Método que permite estimar la posición y orientación de las cámaras, junto con la estructura tridimensional de una escena, a partir de múltiples imágenes capturadas desde distintos puntos de vista.

**Multi-View Stereo (MVS).**

Técnica orientada a la generación de una reconstrucción densa mediante la estimación de profundidad a partir de varias vistas de la escena.

**Reconstrucción sparse.**

Representación inicial de la escena en forma de nube de puntos de baja densidad, utilizada como base para el proceso de reconstrucción tridimensional.

**Reconstrucción densa.**

Proceso que permite obtener una nube de puntos con mayor densidad y nivel de detalle, lo que mejora la calidad del modelo tridimensional final.

**Traslape (overlap).**

Porcentaje de información visual compartida entre imágenes consecutivas. Durante las pruebas se identificó como un factor necesario para establecer correspondencias válidas entre vistas.

**Características (features).**

Puntos distintivos detectados en las imágenes que sirven como referencia para identificar coincidencias entre distintas capturas.

**SIFT (Scale-Invariant Feature Transform).**

Algoritmo empleado para la detección y descripción de características locales, con capacidad de mantener su desempeño frente a variaciones de escala, rotación e iluminación.

**Matching.**

Proceso mediante el cual se establecen correspondencias entre las características detectadas en diferentes imágenes.

**Correspondencias (matches).**

Relaciones establecidas entre puntos clave de distintas imágenes. Durante los ensayos se observó que su ausencia impide la reconstrucción 3D al no poder definir relaciones geométricas entre vistas.

#### **Verificación geométrica.**

Etapas del pipeline que valida las correspondencias mediante restricciones geométricas, con el fin de eliminar coincidencias incorrectas.

#### **Mapper.**

Componente encargado de la reconstrucción incremental, responsable de registrar cámaras y generar la estructura tridimensional inicial a partir de correspondencias válidas.

#### **Par inicial (Initial Pair).**

Primer conjunto de imágenes que permite iniciar el proceso de reconstrucción en SfM. La calidad de este par condiciona la continuidad del pipeline.

#### **Bundle Adjustment.**

Proceso de optimización que ajusta de manera conjunta las posiciones de las cámaras y los puntos tridimensionales, con el objetivo de reducir el error de reproyección.

#### **Nube de puntos.**

Conjunto de puntos en el espacio tridimensional que representa la estructura de la escena reconstruida.

#### **Fallback sparse.**

Mecanismo alternativo que permite generar una reconstrucción básica cuando la reconstrucción densa no puede ejecutarse, situación identificada en algunos ensayos del sistema.

#### **Etapas de fallo.**

Fase específica del pipeline en la que se produce un error que interrumpe la continuidad del proceso de reconstrucción.

La definición de estos conceptos permite interpretar los resultados experimentales obtenidos, en particular el comportamiento del pipeline de reconstrucción, la generación de

modelos tridimensionales y las causas técnicas asociadas a los fallos identificados en el sistema.

### **2.3. Marco legal**

El desarrollo del proyecto involucra el tratamiento de información visual, lo que lo sitúa dentro del marco normativo colombiano en materia de protección de datos personales, particularmente lo dispuesto en la Ley 1581 de 2012. Esta regulación establece las condiciones para el uso, almacenamiento y tratamiento de datos personales, con el fin de garantizar los derechos de los titulares de la información.

En el desarrollo de las pruebas se identificó la necesidad de gestionar las imágenes capturadas bajo criterios de uso responsable. Esta consideración adquiere mayor relevancia cuando las capturas pueden incluir información que permita la identificación de personas. En estos casos, el uso de los datos requiere consentimiento previo y debe limitarse a los fines definidos dentro del alcance del sistema, evitando cualquier utilización no autorizada.

El proyecto mantiene coherencia con los lineamientos institucionales de las Unidades Tecnológicas de Santander en relación con el uso ético de la información. Se tuvieron en cuenta aspectos asociados al respeto por los derechos de autor y al uso adecuado de herramientas digitales, incluidas aquellas que incorporan tecnologías basadas en inteligencia artificial.

### **2.4. Marco ambiental**

El desarrollo del proyecto no genera un impacto ambiental directo asociado a residuos físicos o consumo de materiales, ya que su ejecución se realiza en un entorno completamente digital. A pesar de ello, durante las pruebas se identificó un consumo indirecto de recursos energéticos vinculado al procesamiento de imágenes y a la ejecución del pipeline de reconstrucción 3D.

Este impacto se relaciona principalmente con el uso de hardware requerido para el procesamiento de datos, en especial en tareas que demandan alta capacidad de cómputo. En los ensayos realizados se observó que las etapas más exigentes del pipeline incrementan el consumo energético, lo que establece una relación directa entre la complejidad del procesamiento y el uso de recursos.

Frente a esta condición, se adoptaron estrategias orientadas a optimizar el uso de recursos computacionales. Se priorizó la reducción de ejecuciones innecesarias, se ajustaron configuraciones del sistema y se favoreció el uso de alternativas que permitieran obtener resultados funcionales sin requerir un procesamiento intensivo.

Aunque el impacto ambiental no se manifiesta de forma convencional, el uso eficiente de los recursos computacionales adquiere relevancia dentro del desarrollo tecnológico. Esta consideración resulta especialmente importante en contextos donde la disponibilidad de hardware es limitada y donde la eficiencia del sistema influye directamente en su viabilidad operativa.

### **3. DISEÑO DE LA INVESTIGACIÓN**

Este trabajo no se concibió como un ejercicio meramente teórico. Desde el comienzo, se orientó a la construcción de un sistema funcional, probado en condiciones reales de uso. Bajo esta premisa, se abordó como una investigación aplicada con enfoque cuantitativo, lo que permitió analizar el desempeño del sistema a partir de su ejecución directa.

El estudio se desarrolló bajo un enfoque descriptivo, centrado en observar cómo se comporta el sistema durante las pruebas experimentales. El análisis se dirigió a identificar patrones de funcionamiento, detectar fallos y comprender las limitaciones presentes,

muchas de ellas relacionadas con las capacidades del hardware disponible en el entorno de desarrollo.

Para la validación del sistema, se llevaron a cabo pruebas funcionales y experimentales que cubrieron todo el flujo de reconstrucción, desde la captura de imágenes hasta la generación del modelo tridimensional. Se buscó evaluar la respuesta del sistema en condiciones no ideales, más cercanas a su uso en escenarios reales.

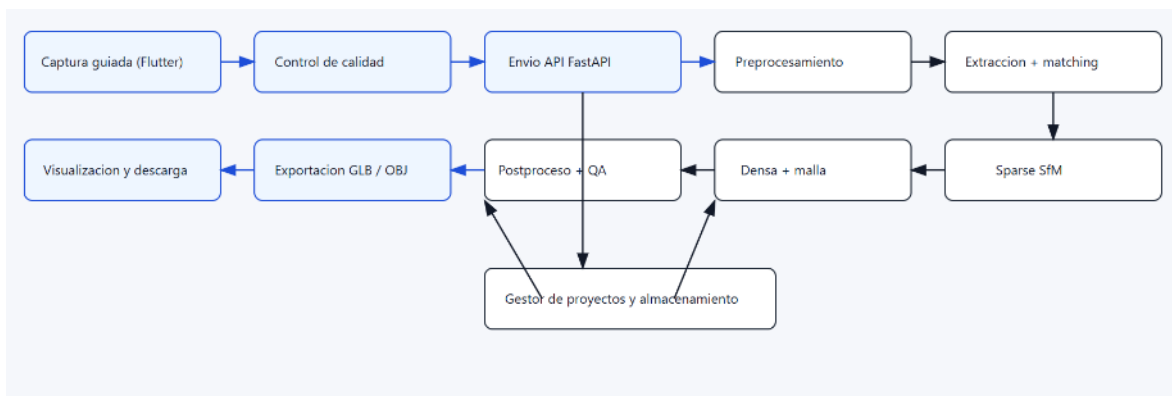
El desarrollo se estructuró por fases, lo que permitió organizar el proceso y analizar con mayor claridad cada una de sus etapas.

### 3.1. Flujo del sistema

Antes de entrar en el detalle de cada fase, resulta necesario entender cómo funciona el sistema en su conjunto.

Para ello, se presenta el diagrama general de la arquitectura del sistema de reconstrucción tridimensional, el cual permite visualizar de forma clara la interacción entre sus componentes principales.

**FIGURA 3.1 - Arquitectura del sistema de reconstrucción 3D**



Fuente: Elaboración propia

La Figura 3.1 sintetiza la arquitectura completa del sistema propuesto, abarcando desde la captura de imágenes hasta la obtención del modelo tridimensional. El proceso comienza con la captura guiada desde la aplicación móvil, donde se controla la calidad de las imágenes a partir de criterios como el traslape, la cobertura y la estabilidad durante la toma.

Una vez obtenidas, las imágenes se envían al backend, donde se realiza el preprocesamiento y se ejecuta el pipeline de reconstrucción basado en COLMAP. En esta fase se llevan a cabo tareas como la extracción de características, el emparejamiento entre imágenes, la estimación de la posición de las cámaras y la generación de la nube de puntos.

Al finalizar la reconstrucción, el sistema verifica la coherencia de la geometría generada y procede a exportar el modelo tridimensional en formatos compatibles, como GLB u OBJ, lo que permite su visualización directamente en la aplicación.

Tal como se aprecia en la Figura 3.1, el sistema se apoya en una arquitectura cliente-servidor: el dispositivo móvil asume la captura de datos, mientras que el backend se encarga del procesamiento intensivo requerido para la reconstrucción tridimensional.

### **3.2. Fase 1: Análisis de requerimientos**

La viabilidad de ejecutar un pipeline de reconstrucción 3D directamente en un entorno móvil fue analizada en las primeras fases del proyecto. La evaluación se centró en el comportamiento de las etapas más exigentes del flujo basado en SfM, particularmente en la generación de la nube de puntos densa. Este proceso implica un alto coste computacional, asociado al volumen de datos y a la complejidad de los algoritmos de optimización y densificación.

El análisis evidenció restricciones propias de los dispositivos móviles que condicionan este tipo de procesamiento. La arquitectura ARM, junto con las limitaciones de memoria volátil y capacidad de cómputo, introduce cuellos de botella que afectan tanto los tiempos de

ejecución como la estabilidad del sistema. En escenarios con múltiples imágenes y alto traslape, el estrés de hardware se incrementa de forma considerable, generando riesgos de fallos o interrupciones en el flujo.

La ejecución local del proceso completo fue descartada. No resultaba sostenible. En su lugar, se adoptó un modelo cliente-servidor que permite desacoplar la captura del procesamiento intensivo. Bajo este esquema, el dispositivo móvil se limita a la adquisición de imágenes y a la gestión de la interacción con el usuario, mientras que el backend asume la ejecución del pipeline utilizando herramientas especializadas y recursos computacionales más robustos.

La arquitectura elegida introduce una distribución más eficiente de la carga. El procesamiento crítico se traslada a un entorno controlado, donde es posible garantizar mayor estabilidad durante etapas como el emparejamiento, la estimación de cámaras y la reconstrucción densa. El cliente permanece liviano. La experiencia de uso se mantiene fluida.

La separación de responsabilidades también impacta en la evolución del sistema. El componente de procesamiento puede ser optimizado, escalado o incluso migrado a infraestructuras de mayor capacidad sin alterar la lógica implementada en la aplicación móvil. Esta flexibilidad resulta clave en contextos donde el volumen de datos o la complejidad del pipeline tienden a incrementarse con el tiempo.

La decisión arquitectónica no respondió únicamente a criterios de implementación, sino a las restricciones observadas durante la ejecución real del sistema. La arquitectura cliente-servidor se consolidó como la alternativa más adecuada al equilibrar rendimiento, estabilidad y capacidad de crecimiento dentro del contexto del proyecto.

### **3.3. Fase 2: Diseño del sistema**

La definición de las especificaciones técnicas se abordó considerando dos componentes claramente diferenciados: la aplicación móvil como punto de adquisición de datos y el backend como entorno de procesamiento del pipeline de reconstrucción. Cada uno impone restricciones distintas. Cada uno exige decisiones propias.

La aplicación móvil se diseñó bajo un criterio de abstracción de la complejidad para el usuario final, sin comprometer los requisitos mínimos del proceso fotogramétrico. Se estableció un protocolo de adquisición de datos volumétricos basado en la captura de múltiples imágenes desde diferentes perspectivas, asegurando traslape suficiente entre vistas y cobertura adecuada del objeto. La organización de las fotografías en proyectos también fue contemplada como parte del flujo, no solo por razones de usabilidad, sino por su impacto directo en la consistencia del procesamiento posterior. La interfaz debía ser simple. El dato, no.

El componente de procesamiento planteó un escenario distinto. Se requería un pipeline completo basado en SfM y MVS, capaz de operar con conjuntos de imágenes capturadas en condiciones no controladas. La evaluación de herramientas incluyó alternativas como OpenMVG combinado con OpenMVS, VisualSFM y Meshroom (AliceVision). Cada una ofrecía capacidades relevantes, pero también presentaba limitaciones en términos de integración, automatización o estabilidad bajo cargas variables. OpenMVG y OpenMVS, por ejemplo, proporcionan modularidad, pero implican una mayor complejidad en la orquestación del flujo. VisualSFM, aunque funcional, mostró restricciones en su mantenimiento y adaptación a entornos modernos. Meshroom, por su parte, facilita el uso mediante interfaces gráficas, pero introduce dependencias que dificultan su integración en arquitecturas automatizadas.

La elección fue pragmática. COLMAP resultó ser el núcleo idóneo.

Su principal ventaja radica en la integración coherente de todas las etapas del pipeline dentro de un mismo entorno: extracción de características, emparejamiento, estimación de poses de cámara, reconstrucción dispersa y densificación. Esta continuidad reduce la

fricción entre etapas y permite una ejecución más controlada del proceso. A nivel interno, COLMAP implementa un enfoque híbrido basado en SfM para la estimación estructural inicial y MVS para la generación de una nube de puntos densa, lo que se traduce en resultados consistentes incluso con variaciones en la calidad de entrada.

La arquitectura del backend se cimentó sobre esta decisión. La incorporación de COLMAP implicó diseñar un entorno capaz de ejecutar procesos intensivos, gestionar archivos intermedios y mantener la integridad del flujo de reconstrucción. La aplicación móvil quedó relegada a su rol natural: captura y envío de datos. El procesamiento, donde realmente ocurre la complejidad, se centralizó en el servidor.

La selección de la herramienta no fue un aspecto aislado. Condicionó la arquitectura, el flujo de datos y la forma en que el sistema responde ante diferentes escenarios de uso. COLMAP no solo se integró como una herramienta más; se convirtió en el eje sobre el cual se estructuró toda la solución de reconstrucción tridimensional.

### **3.4. Fase 3: Desarrollo e implementación**

En esta fase se desarrolló. Se construyó la aplicación móvil en Flutter, buscando que la captura fuera lo más fluida posible. Del lado del procesamiento, se integró COLMAP para encargarse de la reconstrucción 3D usando técnicas de visión por computador.

Durante la implementación se presentaron dificultades iniciales. Se presentaron múltiples errores durante el proceso. En esta etapa, herramientas como ChatGPT, Gemini y Claude se volvieron aliadas para destrabar problemas específicos, probar enfoques y depurar código. Pero nada se dejó al azar. Cada solución pasó por revisión, ajuste y validación manual. Era necesario entender qué estaba pasando, no solo copiar y pegar.

### 3.5. Fase 4: Pruebas y validación

El comportamiento del sistema se evaluó bajo distintos escenarios de captura, variando deliberadamente condiciones como la iluminación, el número de imágenes y la textura de las superficies. El interés no se centró únicamente en verificar la generación de modelos tridimensionales, sino en identificar los límites operativos del pipeline de reconstrucción basado en Structure from Motion (SfM) y reconstrucción densa.

El desempeño más consistente se registró en conjuntos de entre 20 y 40 imágenes, siempre que existiera un traslape suficiente entre tomas y condiciones de iluminación relativamente estables. En estos casos, los registros del sistema evidenciaron una alta densidad de correspondencias válidas durante la etapa de emparejamiento, lo que permitió una estimación robusta de las poses de cámara. El resultado fue una nube de puntos dispersa con estructura geométrica coherente, seguida de una reconstrucción densa que preservó adecuadamente la forma general del objeto. La calidad no fue perfecta, pero sí estable. Eso es clave.

Cuando las condiciones de captura se degradaron, el comportamiento del sistema cambió de forma notable. La iluminación irregular introdujo variaciones en la intensidad de los píxeles que afectaron directamente la detección de características locales, reduciendo la repetibilidad de descriptores como SIFT. La consecuencia fue clara: el algoritmo de emparejamiento experimentó una caída en el número de correspondencias geoméricamente consistentes. En superficies con baja textura, el problema se intensificó. Sin suficientes puntos distintivos, el proceso de alineación de vistas perdió estabilidad, derivando en reconstrucciones parciales o, en algunos casos, en la interrupción completa del pipeline. No fue posible generar una estructura geométrica válida.

Los registros generados durante la ejecución permitieron rastrear fallos específicos. Se documentaron eventos de lectura incompleta de imágenes y múltiples instancias donde el sistema no logró establecer correspondencias entre pares de vistas. Este patrón no fue

aleatorio. Está directamente relacionado con una baja relación señal-ruido en los datos de entrada, lo que limita la capacidad del sistema para identificar patrones consistentes entre imágenes. En términos prácticos, el sistema no falla por sí mismo; falla porque la información visual disponible no es suficiente para sostener el modelo matemático subyacente.

El tiempo de procesamiento presentó una variabilidad considerable. A medida que aumentaba el número de imágenes, también lo hacía la carga computacional, particularmente en las etapas de ajuste global (bundle adjustment) y reconstrucción densa. En equipos con recursos limitados, se observaron incrementos significativos en los tiempos de ejecución, acompañados en algunos casos por interrupciones en etapas críticas. Esto confirma que el rendimiento no depende exclusivamente del algoritmo, sino también de la infraestructura disponible. Este comportamiento responde directamente a la carga computacional: más datos implican más costo computacional.

En conjunto, los resultados permiten afirmar que el sistema cumple con su propósito funcional dentro de un rango de condiciones bien definidas. Sin embargo, su desempeño está fuertemente condicionado por la calidad de las imágenes de entrada y la capacidad de procesamiento del entorno. La reconstrucción 3D, en este contexto, no es un proceso completamente robusto frente a variaciones adversas; es, más bien, un equilibrio delicado entre captura, procesamiento y recursos.

### **3.6. Fase 5: Análisis de resultados**

Con los datos en mano, se realizó un análisis objetivo de los resultados. El sistema funciona. Sí. Pero con restricciones claras. Se identificaron fallos recurrentes, limitaciones en el procesamiento y dependencia directa de la calidad de entrada. Este análisis no buscaba demostrar perfección, sino entender hasta dónde llega realmente la solución en condiciones reales.

## 4. DESARROLLO DEL TRABAJO DE GRADO

Este capítulo no trata de una idea en papel. Aquí es donde todo se aterrizó. El objetivo fue claro desde el inicio: lograr que un conjunto de imágenes 2D terminara convertido en un modelo 3D funcional. No perfecto. Funcional.

Para eso, optamos por integrar dos mundos: una aplicación móvil que captura y un sistema de procesamiento que hace el trabajo pesado. Desde una perspectiva conceptual. En la práctica, hubo varios puntos donde el sistema simplemente no respondía como se esperaba.

El código fuente del sistema desarrollado se encuentra disponible en los siguientes repositorios:

- **Aplicación móvil (Flutter):**  
[https://github.com/AlemanSamu/proyecto\\_3d](https://github.com/AlemanSamu/proyecto_3d)
- **Backend de procesamiento (reconstrucción 3D):**  
<https://github.com/AlemanSamu/PROCESAMIENTO>

### 4.1. Diseño del sistema

Se decidió separar responsabilidades desde el inicio. Era necesario. No tenía sentido intentar hacer todo en el móvil.

Se optó por una arquitectura cliente-servidor con dos piezas claras:

- Aplicación móvil: encargada de capturar las imágenes desde distintos ángulos.
- Sistema de procesamiento (backend): donde realmente ocurre la reconstrucción.

La clave estuvo en la separación de responsabilidades. La captura debía ser ligera. El procesamiento, inevitablemente pesado. Si todo se dejaba en el dispositivo, simplemente

no iba a funcionar en equipos de gama media —que es justo lo que se tiene en muchos casos en las UTS.

#### 4.2. Diseño de la aplicación móvil

Para la app, decidí trabajar con Flutter. No fue casualidad. La idea era evitar que la aplicación se volviera lenta o inestable al momento de capturar imágenes.

El enfoque fue sencillo: permitir al usuario tomar múltiples fotos desde diferentes ángulos sin complicarse demasiado. Pero había un detalle importante. Las imágenes tenían que servir. No cualquier foto funciona en reconstrucción 3D.

Por eso, la aplicación guía la captura —aunque de forma básica— buscando que haya variación de ángulos y cobertura suficiente del objeto. También permite organizar proyectos y enviar las imágenes al backend.

En la práctica, aquí ya se notaban problemas. Si el usuario no capturaba bien, el sistema después fallaba. Y eso no se puede evitar del todo.

#### 4.3. Diseño del sistema de procesamiento

Aquí es donde el sistema realmente se pone a prueba. El backend recibe las imágenes y ejecuta el pipeline de reconstrucción usando COLMAP. Integrarlo no fue tan directo como parecía. Presenta configuraciones complejas. Configuraciones que si no se ajustan bien, simplemente no funcionan.

El flujo que sigue el sistema es este:

- Extracción de características de las imágenes
- Emparejamiento de puntos entre imágenes

- Estimación de la posición de las cámaras
- Generación de nube de puntos dispersa (Sparse Reconstruction)
- Generación de nube de puntos densa (Dense Reconstruction)
- Exportación del modelo 3D

Cada etapa depende de la anterior. Si algo falla al inicio, todo lo demás se ve afectado. Y Esta situación se presentó en múltiples ocasiones.

#### 4.4. Implementación del sistema

Aquí fue donde todo se conectó. La aplicación móvil con el backend. El flujo completo funcionando.

El sistema sigue el siguiente flujo de procesamiento: El cumplimiento de este objetivo se abordó mediante la integración de un flujo de procesamiento sustentado en técnicas de visión por computador, utilizando COLMAP como núcleo para la ejecución del pipeline de reconstrucción basado en SfM. La solución se estructuró bajo una arquitectura cliente-servidor, donde la aplicación móvil se encargó de la captura y envío de imágenes, mientras que el procesamiento se realizó en el backend. Esta separación permitió gestionar de forma más eficiente las cargas computacionales asociadas a la reconstrucción.

El flujo se inició con la recepción de las imágenes y su organización dentro del entorno de procesamiento. Esta etapa, aunque operativa, resultó determinante para garantizar la consistencia de los datos de entrada. A continuación, se ejecutó la extracción de características locales en cada fotografía, proceso mediante el cual se identificaron puntos distintivos susceptibles de ser comparados entre diferentes vistas. Sobre esta base, el sistema llevó a cabo el emparejamiento de imágenes, estableciendo correspondencias a partir del traslape existente y construyendo relaciones geométricas iniciales entre las capturas.

Con las coincidencias definidas, se procedió a la estimación de la posición y orientación de las cámaras, etapa fundamental dentro del enfoque SfM. Este cálculo permitió generar una nube de puntos dispersa que representa la estructura básica de la escena. A partir de esta reconstrucción inicial, se ejecutó la fase de densificación, orientada a incrementar la cantidad de puntos y mejorar el nivel de detalle del modelo. Este paso implicó un mayor costo computacional, pero resultó necesario para obtener una representación más completa del entorno procesado.

El resultado final fue exportado en un formato compatible con el sistema desarrollado, permitiendo su visualización e integración dentro del prototipo. Más allá de la obtención del modelo, se buscó garantizar que el flujo de salida fuera coherente con los requerimientos de la aplicación, evitando incompatibilidades en etapas posteriores.

Las pruebas realizadas evidenciaron que el pipeline implementado logra ejecutar el proceso completo de reconstrucción dentro del entorno propuesto. No obstante, se observaron variaciones en la calidad de los resultados asociadas principalmente a las condiciones de captura y a la calidad de las imágenes de entrada. A pesar de estas fluctuaciones, el comportamiento general del sistema permitió validar que la técnica basada en SfM fue incorporada de manera funcional, cumpliendo con el objetivo planteado en el desarrollo del prototipo.

#### **4.5. Pruebas del sistema**

Evaluación del comportamiento del sistema mediante pruebas experimentales. La evaluación del sistema se llevó a cabo mediante un conjunto de pruebas experimentales orientadas a analizar su desempeño bajo diferentes condiciones de captura. Para ello, se variaron tanto los parámetros asociados a la adquisición de imágenes como las características visuales de los objetos procesados. Este enfoque permitió aproximarse a escenarios de uso reales, donde la calidad de los datos de entrada no siempre es homogénea y puede afectar directamente el comportamiento del pipeline de reconstrucción basado en SfM.

El análisis de los resultados evidenció que el sistema presenta un desempeño más estable cuando las imágenes cumplen ciertas condiciones básicas: iluminación adecuada, suficiente traslape entre capturas y presencia de textura en las superficies. Bajo este tipo de configuraciones, la nube de puntos generada mostró una mayor densidad y coherencia estructural, lo que se tradujo en una representación más completa de la escena. La continuidad geométrica fue notablemente superior en comparación con otros escenarios evaluados.

Cuando estas condiciones no se cumplen, el comportamiento del sistema cambia de forma significativa. La presencia de iluminación deficiente, variaciones bruscas en los ángulos de captura o superficies con escasa información visual afecta la detección de características y reduce la cantidad de correspondencias válidas. Como consecuencia, se observó una degradación en la calidad de la reconstrucción, manifestada en vacíos dentro de la nube de puntos o en inconsistencias geométricas en el modelo generado.

El rendimiento computacional también mostró variaciones relevantes durante las pruebas. El tiempo de ejecución del pipeline se incrementó en función del número de imágenes procesadas y de la capacidad del hardware disponible en la arquitectura cliente-servidor. En entornos con recursos limitados, ciertas etapas críticas, como la reconstrucción densa, presentaron mayores tiempos de respuesta e incluso inestabilidad en algunos casos. Este comportamiento pone en evidencia que la eficiencia del sistema no está determinada exclusivamente por el algoritmo, sino por la interacción entre el volumen de datos y la infraestructura de procesamiento.

Los resultados obtenidos permiten afirmar que el sistema ejecuta de manera funcional el proceso de reconstrucción tridimensional dentro del entorno propuesto. Sin embargo, la calidad del modelo generado y la estabilidad del flujo dependen de factores externos al software, particularmente de las condiciones de captura y de los recursos computacionales disponibles. La evaluación experimental, en este sentido, no solo validó el funcionamiento del prototipo, sino que también delimitó sus principales restricciones operativas.

#### **4.6. Limitaciones del sistema**

Las limitaciones no fueron sorpresa. Se sintieron desde el inicio.

- El hardware fue el límite.
- El procesamiento es pesado.
- La calidad de las imágenes manda.

En escenarios con pocas texturas o malas condiciones de captura, el sistema tiene dificultades para registrar correctamente las imágenes. Lo cual afecta directamente la calidad del resultado final.

También hay que decirlo: el tiempo de procesamiento en equipos de bajo rendimiento es alto. No es inmediato. Hay que esperar.

Aun así, el sistema logra completar el flujo. No siempre con la mejor calidad, pero sí demostrando que es posible trabajar reconstrucción 3D en condiciones más accesibles.

## 5. RESULTADOS

### 5.1. Resultados generales de las pruebas experimentales

La evaluación del sistema se realizó a partir de un conjunto de pruebas ejecutadas en condiciones reales de uso. Más que buscar un entorno controlado, el interés estuvo en observar cómo respondía el sistema cuando se enfrentaba a escenarios cercanos a su aplicación práctica.

Las pruebas se plantearon variando aspectos clave durante la captura, como el número de imágenes, la iluminación y las características de los objetos. Este enfoque permitió identificar con mayor claridad qué factores afectan el comportamiento del pipeline de reconstrucción.

La información obtenida proviene directamente de los registros generados por el sistema, incluyendo archivos de metadatos (meta.json) y logs técnicos del procesamiento en COLMAP. A partir de estos datos fue posible analizar el estado de cada ejecución, los tiempos de procesamiento, la cantidad de imágenes registradas y los puntos tridimensionales generados.

Los resultados detallados de cada prueba se presentan en la Tabla 1.

**Tabla 1 - Resultados de las pruebas experimentales**

Prueba	Objeto	Nº imágenes	Tiempo (s)	Resultado	Imágenes registradas	Puntos 3D	Observación
1	E2E remoto	3	1.020	Fallido	0	0	Sin correspondencias
2	E2E remoto	3	0.203	Fallido	0	0	Error CLI
3	Objeto simple	3	1.212	Fallido	0	0	Sin correspondencias
4	Objeto simple	5	23.866	Fallido	0	0	Sin par inicial
5	Objeto simple	7	33.468	Fallido	0	0	Sin par inicial
6	Audífono	24	100.012	Exitoso parcial	10	632	Fallback sparse
7	Cargador	48	100.031	Exitoso parcial	4	32	Fallback sparse
8	Objeto técnico	24	49.759	Exitoso parcial	24	777	Fallback sparse
9	Hilo	24	75.500	Fallido	0	0	Error BA
10	E2E remoto	3	1.915	Fallido	0	0	Sin correspondencias

Al revisar los datos, se observa que de las 10 pruebas realizadas, únicamente 3 lograron completarse, mientras que 7 terminaron en fallo. Esto corresponde a una tasa de éxito global del 30%.

El comportamiento del tiempo de procesamiento también deja ver diferencias claras. Las ejecuciones exitosas tomaron en promedio 83.267 segundos, mientras que las fallidas apenas alcanzaron 19.598 segundos. Esto sugiere que los fallos ocurren en etapas tempranas del proceso, antes de que el sistema avance hacia fases más exigentes.

El resumen de estos resultados se presenta en la Tabla 2, donde se consolidan los principales indicadores.

**Tabla 2 - Resumen estadístico de resultados**

Indicador	Valor
Total de pruebas	10
Pruebas exitosas	3
Pruebas fallidas	7
Tasa de éxito global	30.0%
Promedio tiempo (éxito)	83.267 s
Promedio tiempo (fallo)	19.598 s

Un aspecto importante es que, incluso en los casos exitosos, no se logró completar la reconstrucción densa. En su lugar, el sistema generó resultados utilizando un mecanismo de fallback sparse, debido a limitaciones en el uso de GPU. Aun así, se obtuvieron modelos funcionales, lo que permite afirmar que el sistema cumple con su propósito dentro de las condiciones evaluadas.

La cantidad de pruebas realizadas permitió reconocer comportamientos recurrentes del sistema frente a distintas condiciones de captura. Si bien no se trata de una validación estadística de carácter exhaustivo, los resultados obtenidos resultan suficientes para describir el funcionamiento operativo del prototipo.

## 5.2. Análisis según número de imágenes

Para analizar el efecto de la cantidad de imágenes en el proceso de reconstrucción, las pruebas se organizaron según el número de capturas utilizadas en cada ejecución.

Los resultados se presentan en la Tabla 3.

**Tabla 3 - Resultados del sistema según número de imágenes**

Rango de imágenes	Total, de pruebas	Pruebas exitosas	Pruebas fallidas	Tasa de éxito
≤ 7 imágenes	6	0	6	0%
≥ 24 imágenes	4	3	1	75%

El comportamiento del sistema cambia de manera evidente según la cantidad de imágenes utilizadas. En los conjuntos con siete imágenes o menos no se logró completar ninguna

reconstrucción, lo que sugiere que no se generan suficientes correspondencias entre las capturas.

Cuando se trabaja con conjuntos de 24 imágenes o más, el desempeño mejora de forma notable. La tasa de éxito alcanza el 75%, lo que pone en evidencia que contar con mayor información visual favorece directamente el avance del proceso.

Este resultado coincide con lo que se espera en métodos basados en Structure from Motion, donde el traslape entre imágenes cumple un papel clave. Sin suficientes puntos en común entre las vistas, resulta difícil establecer relaciones geométricas que permitan reconstruir la escena.

### **5.3. Análisis según condiciones de captura**

Las condiciones de captura tuvieron un impacto directo en los resultados. Durante las pruebas se hizo evidente que no basta con tener varias imágenes; la calidad de esas imágenes es igual de importante.

Cuando las capturas se realizaron con buena iluminación, suficiente traslape y objetos con textura, el sistema logró registrar más imágenes y generar nubes de puntos más completas.

En cambio, en condiciones menos favorables, como iluminación deficiente o superficies con poca textura, el comportamiento del sistema cambió. La detección de características disminuyó, las correspondencias se redujeron y el proceso terminó fallando en etapas tempranas.

Esto deja claro que el sistema depende en gran medida de la calidad de los datos de entrada. No es solo un tema de algoritmo, sino de cómo se capturan las imágenes.

### **5.4. Análisis del tiempo de procesamiento**

El tiempo de procesamiento no fue constante entre pruebas, y esa variación está directamente relacionada con el avance del sistema dentro del pipeline.

Las ejecuciones exitosas tomaron más tiempo porque lograron recorrer más etapas del proceso. En estos casos, el sistema alcanzó fases como la estimación de cámaras y la generación de la nube de puntos.

Por el contrario, las ejecuciones fallidas finalizaron rápidamente. Esto ocurre cuando el sistema no logra superar las etapas iniciales, como la extracción de características o el emparejamiento de imágenes.

En este sentido, el tiempo de procesamiento puede interpretarse como una señal del estado del sistema: cuanto más avanza, mayor es el tiempo requerido.

### 5.5. Causas técnicas de fallo

El análisis de los registros del sistema permitió identificar patrones claros en los fallos.

La distribución de estos errores se presenta en la Tabla 4.

**Tabla 4** - Distribución de fallos del sistema

Tipo de fallo	Frecuencia
Sin correspondencias entre imágenes	3
Sin par inicial válido	2
Error en ajuste global (BA)	1
Error de configuración (CLI)	1

La mayoría de los fallos se concentran en las primeras etapas del pipeline. En particular, destacan dos situaciones: la ausencia de correspondencias entre imágenes y la imposibilidad de establecer un par inicial válido.

Estos problemas están directamente relacionados con la calidad de las capturas. Cuando no hay suficiente traslape o no existen características distintivas en las imágenes, el sistema no puede establecer relaciones entre ellas.

En estos casos, el fallo no proviene del algoritmo en sí, sino de la falta de información necesaria para que el proceso funcione.

## 5.6. Discusión

### 5.7. de resultados

Los resultados muestran que el sistema es capaz de ejecutar el flujo completo de reconstrucción 3D bajo condiciones adecuadas. Es decir, el proceso funciona.

Sin embargo, ese funcionamiento no es uniforme. El desempeño varía según factores externos como la calidad de las imágenes, el número de capturas y los recursos disponibles para el procesamiento.

Si se compara con entornos controlados reportados en la literatura, donde la reconstrucción densa suele completarse sin mayores inconvenientes, aquí se observan limitaciones claras. La ausencia de GPU y las condiciones variables de captura influyen directamente en los resultados.

La evaluación realizada se centra en el comportamiento operativo del sistema. Se consideraron indicadores como el estado de ejecución, el número de imágenes registradas y la cantidad de puntos generados.

Es importante tener en cuenta que este análisis no corresponde a una validación basada en métricas geométricas de precisión, ni a una comparación directa con modelos de referencia. El objetivo fue entender cómo responde el sistema en condiciones reales.

En ese sentido, más que presentar una solución perfecta, los resultados muestran una implementación funcional con limitaciones bien definidas.

### 5.8. Cumplimiento de los objetivos

Los resultados obtenidos permiten evidenciar el cumplimiento de los objetivos planteados en el proyecto, a partir del análisis experimental desarrollado en el capítulo 5.

En cuanto a la identificación de los requerimientos técnicos, estos se reflejan directamente en los resultados presentados en las Tablas 1, 2 y 4. Allí se puede observar cómo factores como la iluminación, el traslape entre imágenes y la capacidad de procesamiento influyen de forma clara en el comportamiento del sistema, especialmente en la generación de correspondencias y en la estabilidad del pipeline de reconstrucción.

El diseño de la aplicación móvil se valida en la práctica mediante la captura organizada de imágenes, lo que permitió construir conjuntos de datos adecuados para el proceso de reconstrucción 3D. Esto se evidencia en las ejecuciones exitosas del sistema, donde la calidad de las capturas fue suficiente para que el flujo de procesamiento se desarrollara sin interrupciones.

Por otro lado, la implementación del algoritmo se demuestra a partir de la ejecución del pipeline completo utilizando COLMAP. Esta integración se refleja en las pruebas experimentales presentadas en la Tabla 1, donde el sistema logró avanzar por las etapas clave del proceso, desde la extracción de características hasta la generación de la nube de puntos, confirmando que la técnica fue incorporada de manera funcional.

En relación con la evaluación del sistema, esta se sustenta en el análisis de los resultados experimentales, especialmente en las Tablas 2 y 3. A partir de estos datos se puede observar la tasa de éxito del sistema y cómo el número de imágenes influye directamente en la estabilidad del proceso, permitiendo establecer condiciones mínimas para obtener resultados válidos.

En conjunto, la evidencia obtenida permite afirmar que el sistema implementado logra ejecutar el proceso de reconstrucción 3D de extremo a extremo utilizando imágenes capturadas con dispositivos móviles.

Aunque el desempeño del sistema depende de factores externos como la calidad de las imágenes y la capacidad de procesamiento disponible, los resultados obtenidos muestran

que la solución es técnicamente viable en escenarios reales, incluso cuando las condiciones no son ideales.

### 5.9. Evidencia visual del sistema

Se incorpora evidencia visual del sistema desarrollado, la cual abarca las etapas de captura de imágenes, validación, procesamiento en el backend y visualización del modelo tridimensional. Este conjunto de elementos permite respaldar los resultados experimentales obtenidos y aportar claridad sobre el funcionamiento del sistema en condiciones reales.

En primera instancia, se presentan las imágenes correspondientes al objeto de estudio (cargador), capturadas desde múltiples ángulos mediante la aplicación móvil.

**FIGURA 5.1 - Captura del objeto desde diferentes perspectivas.**



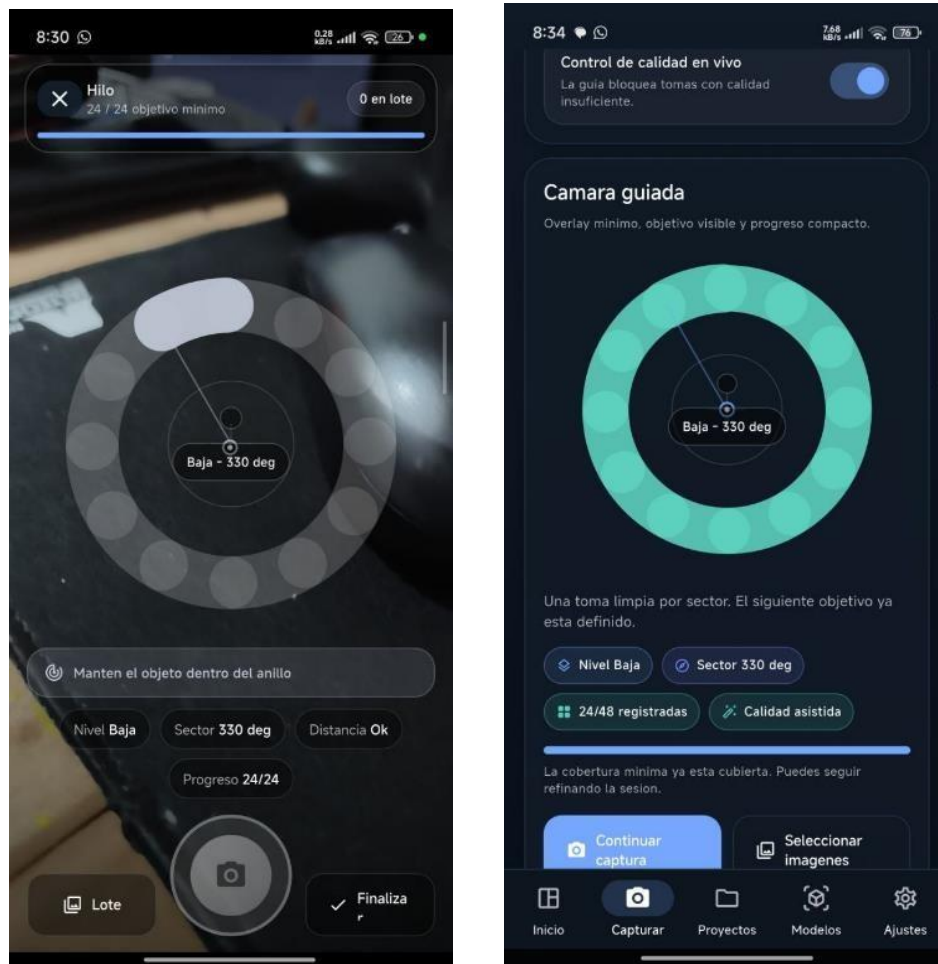


Fuente: Elaboración propia.

Las imágenes presentadas permiten evidenciar la variación de puntos de vista necesaria para llevar a cabo el proceso de reconstrucción tridimensional, aspecto clave para garantizar la calidad del modelo generado.

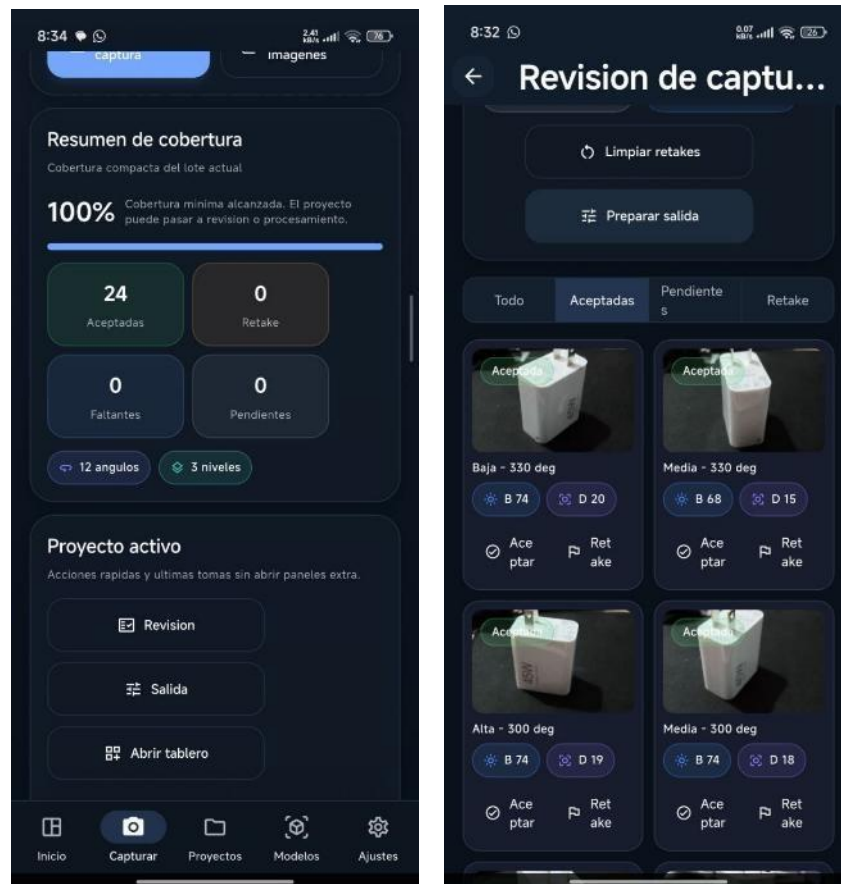
Se muestra la interfaz de captura guiada implementada en la aplicación, diseñada para orientar al usuario durante la toma de imágenes mediante un sistema de anillos y referencias angulares. Este recurso permite controlar de forma más precisa la posición, el ángulo y la cobertura de las capturas, lo que favorece la obtención de un conjunto de datos adecuado para el proceso de reconstrucción.

**FIGURA 5.2 - Interfaz de captura guiada con control de ángulo y nivel.**



Fuente: Elaboración propia.

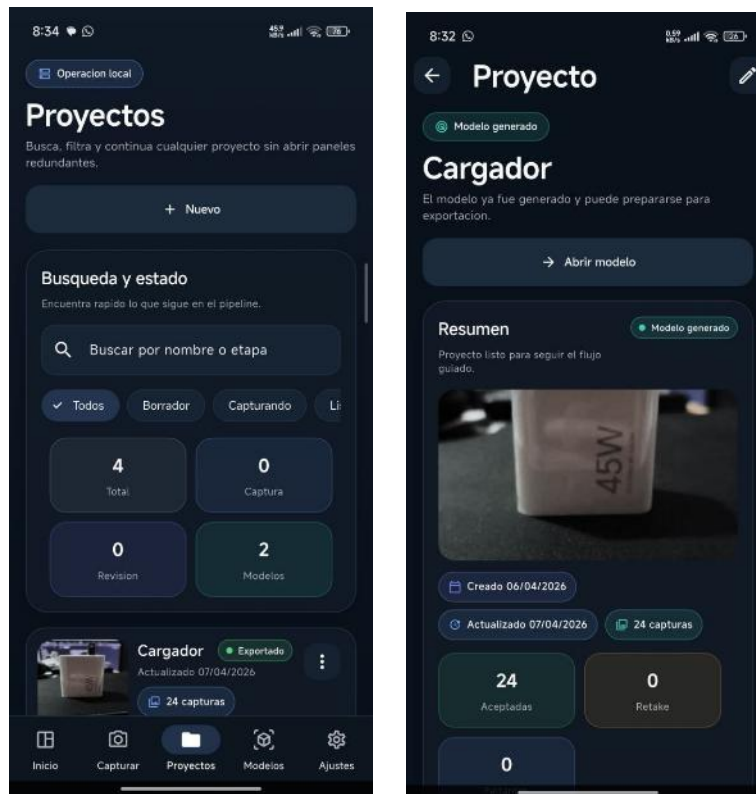
**FIGURA 5.3 – Validación del lote de imágenes y cobertura alcanzada.**



Fuente: Elaboración propia.

La Figura 5.3 muestra la interfaz de validación del conjunto de imágenes, en la cual se puede observar el nivel de cobertura alcanzado y la clasificación de las capturas según su estado. Este proceso de validación permite comprobar que se cumplen las condiciones mínimas necesarias antes de iniciar el procesamiento, lo que contribuye a garantizar la calidad de los datos de entrada.

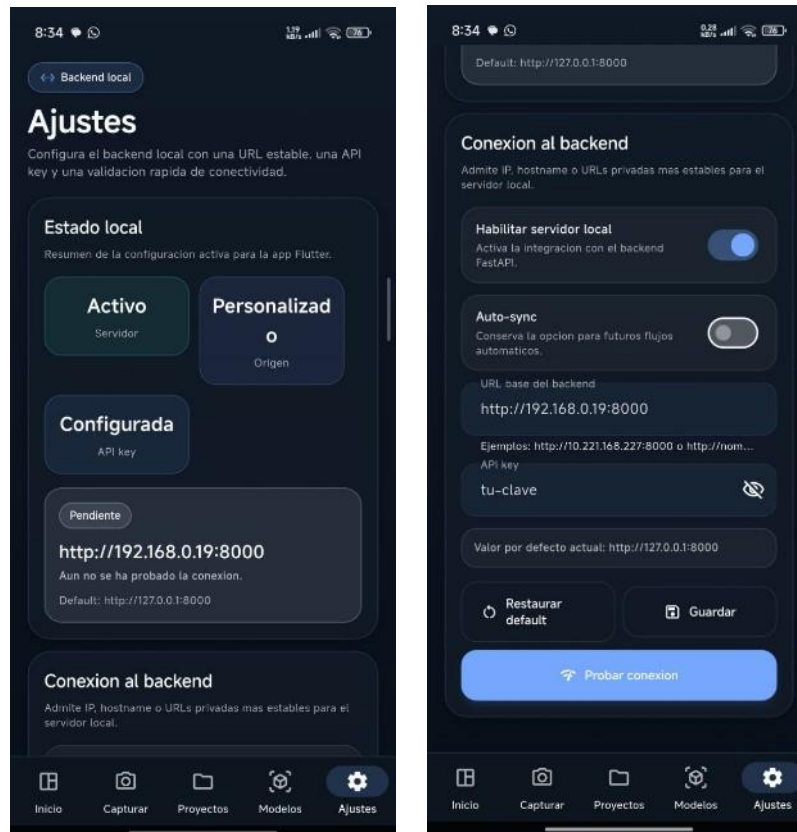
**FIGURA 5.4 – Gestión de proyectos y estado de capturas en la aplicación.**



Fuente: Elaboración propia.

El sistema integra, además, la configuración de un *backend* local (servidor de procesamiento), lo cual permite evidenciar la implementación de una arquitectura cliente-servidor orientada al tratamiento de los datos. Como se observa en la Figura 5.4, esta configuración se encuentra vinculada con la gestión de proyectos y el estado de las capturas dentro de la aplicación.

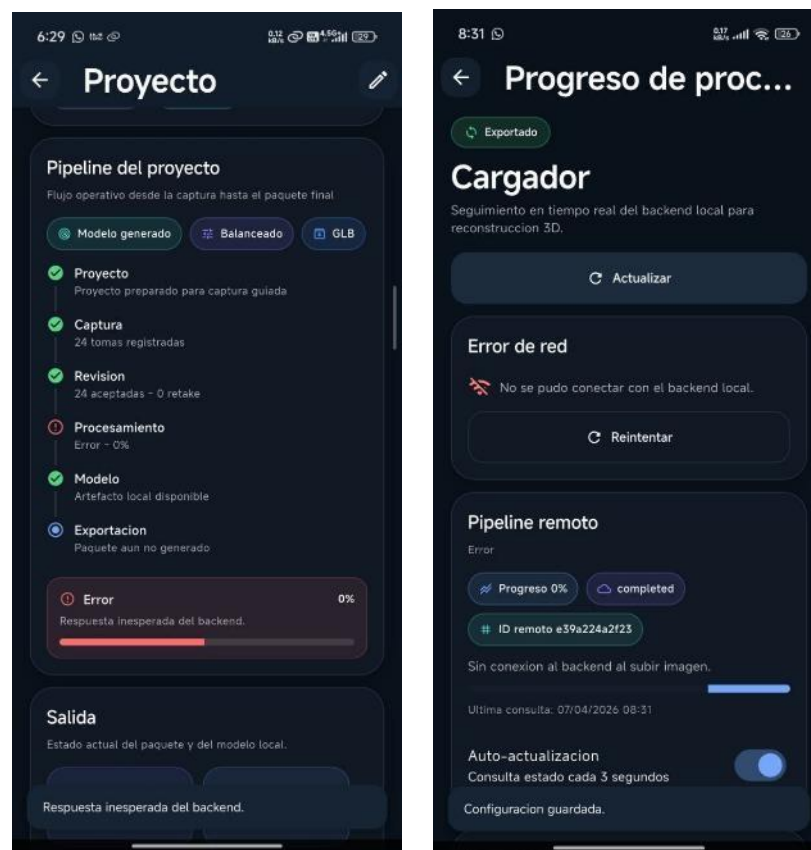
**FIGURA 5.5 - Configuración de conexión con *backend* local.**



Fuente: Elaboración propia.

Como se aprecia en la Figura 5.5, el sistema permite configurar la conexión con el backend local mediante una dirección IP y un puerto específico. Esta configuración resulta clave para establecer la comunicación entre la aplicación móvil y el servidor encargado del procesamiento.

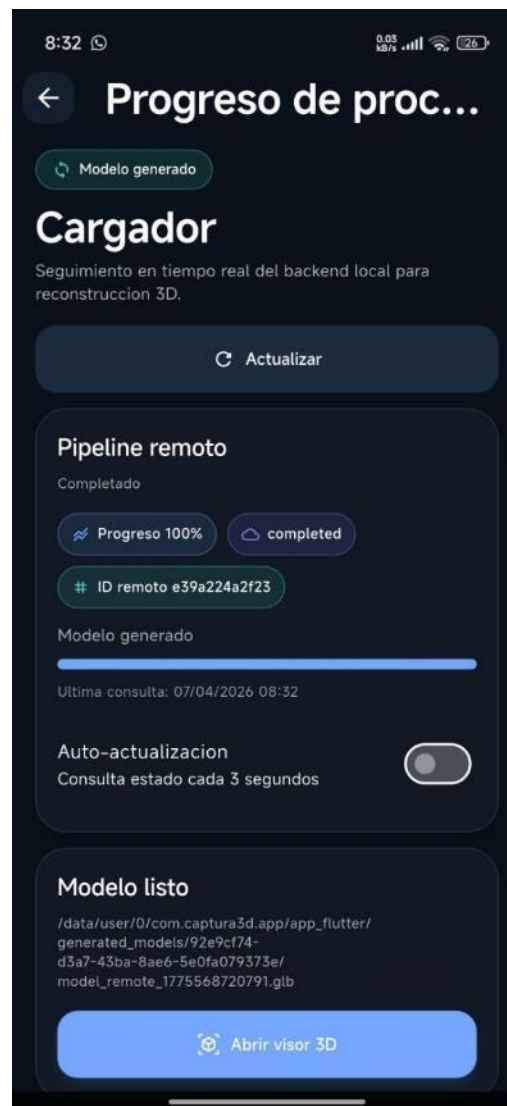
**FIGURA 5.6 - Registro de error de conexión durante el procesamiento.**



Fuente: Elaboración propia.

Durante la ejecución del pipeline, se registraron fallos relacionados con problemas de conectividad con el *backend*, tal como se observa en la Figura 5.6. Estas situaciones permiten evidenciar la dependencia del sistema en la comunicación entre la aplicación y el servidor, así como la importancia de disponer de una conexión estable para asegurar un procesamiento adecuado.

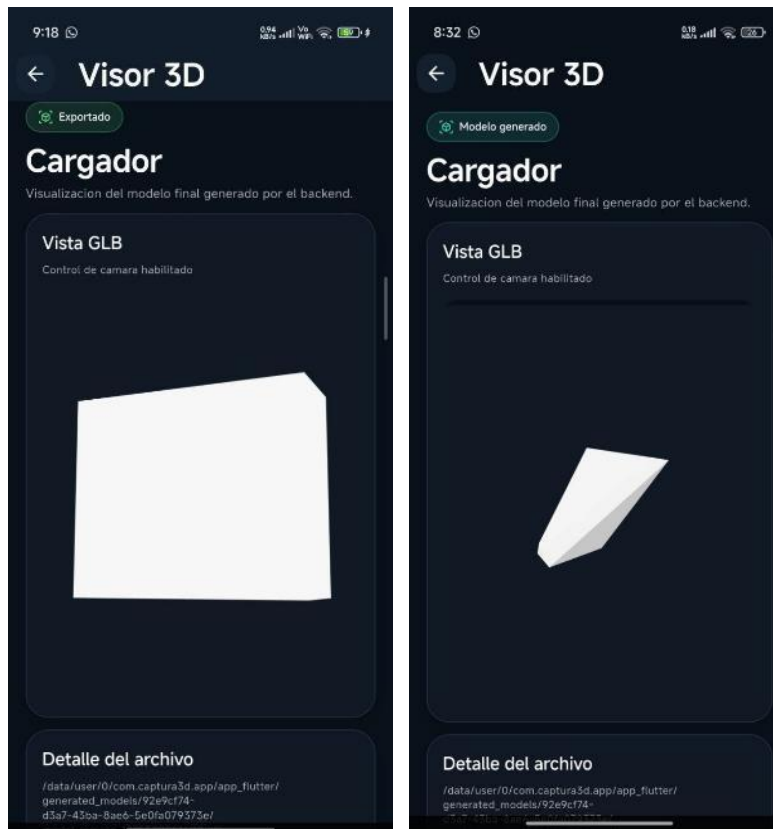
**FIGURA 5.7 - Ejecución exitosa del pipeline y generación del modelo.**



Fuente: Elaboración propia.

En la Figura 5.7 se puede observar que el sistema finaliza correctamente el proceso de reconstrucción 3D, mostrando el progreso completo y dejando disponible el modelo generado para su visualización.

**FIGURA 5.8 - Visualización del modelo 3D generado en formato GLB.**



Fuente: Elaboración propia.

En la Figura 5.8 se observa el modelo 3D generado y su visualización dentro de la aplicación. Aunque la forma obtenida no coincide completamente con el objeto original, el resultado permite evidenciar que el proceso de reconstrucción se ejecutó de principio a fin y, al mismo tiempo, deja ver algunas limitaciones del sistema bajo las condiciones de captura utilizadas.

## 6. CONCLUSIONES

El desarrollo del proyecto permitió comprobar que es posible implementar un sistema de reconstrucción 3D a partir de imágenes capturadas con dispositivos móviles, integrando una aplicación de captura con un entorno de procesamiento basado en técnicas de visión por computador.

Desde el punto de vista de la implementación, el uso de COLMAP permitió integrar un pipeline completo basado en Structure from Motion. En las ejecuciones exitosas, el sistema logró avanzar por las etapas principales del proceso, incluyendo la estimación de cámaras y la generación de la nube de puntos. No obstante, la reconstrucción densa no se completó en los ensayos realizados, debido principalmente a limitaciones en el hardware disponible, lo que llevó al uso de alternativas como el fallback sparse.

A lo largo de los ensayos realizados, el sistema logró ejecutar el flujo completo de reconstrucción, desde la captura de imágenes hasta la generación de modelos tridimensionales funcionales. Los resultados obtenidos corresponden a la ejecución real del sistema, sino que corresponde a lo observado directamente durante la ejecución del sistema.

En particular, se identificó que conjuntos con menos de siete imágenes no permiten generar reconstrucciones válidas, mientras que a partir de 24 imágenes el sistema comienza a mostrar un comportamiento más estable, alcanzando una tasa de éxito del 75%. Esto confirma que la disponibilidad de información visual suficiente es clave para sostener el proceso de reconstrucción.

El análisis de los fallos mostró que la mayoría se presentan en etapas iniciales del pipeline, especialmente cuando no existen suficientes correspondencias entre imágenes o no se logra establecer un par inicial válido. Esto refuerza la idea de que la calidad de los datos de entrada es determinante en el resultado final.

## 7. RECOMENDACIONES

En términos generales, los resultados permiten concluir que la reconstrucción 3D a partir de dispositivos móviles es viable en contextos académicos, siempre que se cumplan ciertas condiciones mínimas de captura y se cuente con un entorno de procesamiento adecuado.

A partir de los resultados obtenidos, queda claro que uno de los factores más determinantes en el desempeño del sistema es la calidad de la captura. Por esta razón, una de las principales mejoras debe enfocarse en la aplicación móvil.

Sería recomendable incorporar mecanismos más claros de guía para el usuario durante la captura de imágenes. Indicaciones sobre ángulos, cantidad de fotografías, distancia y condiciones de iluminación pueden ayudar a reducir errores desde el inicio del proceso. En la práctica, gran parte de los fallos observados se originan en esta etapa.

Otro punto clave está relacionado con la capacidad de procesamiento. El sistema actual funciona, pero se encuentra limitado por el hardware disponible. La incorporación de equipos con GPU dedicada permitiría mejorar significativamente tanto los tiempos de ejecución como la calidad de los modelos generados, especialmente en la etapa de reconstrucción densa.

En esa misma línea, una alternativa viable es migrar el procesamiento a entornos en la nube. Esto permitiría escalar el sistema y manejar cargas de trabajo más exigentes sin depender directamente de la infraestructura local. Además, facilitaría el acceso a recursos más robustos sin afectar la experiencia del usuario en el dispositivo móvil.

Desde el punto de vista técnico, también resulta pertinente explorar enfoques más recientes de reconstrucción, como aquellos basados en aprendizaje profundo. Métodos como NeRF ofrecen resultados de alta calidad, aunque implican mayores requerimientos

computacionales. Su evaluación podría abrir nuevas posibilidades para mejorar el sistema en futuras versiones.

A nivel del sistema, aún hay espacio para optimizar el pipeline de procesamiento. La automatización de ciertas etapas, la mejora en la gestión de errores y la optimización de parámetros podrían contribuir a una mayor estabilidad del proceso.

Finalmente, desde una perspectiva institucional, se recomienda fortalecer los recursos disponibles para este tipo de proyectos. El acceso a estaciones de trabajo más robustas y herramientas especializadas permitiría ampliar el alcance de desarrollos similares y mejorar los resultados obtenidos.

En conjunto, estas recomendaciones no buscan replantear el sistema desde cero, sino aprovechar la base construida para continuar su evolución, ajustándolo progresivamente a condiciones más exigentes y escenarios de uso más amplios.

## 8. REFERENCIAS BIBLIOGRÁFICAS

Bai, H., Chen, Z., & Wang, Y. (2024). Hi-NeuS: High-Fidelity Mask-free Neural Surface Reconstruction for Virtual Reality. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). <https://doi.org/10.48550/arXiv.2402.18008>

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., & Leonard, J. J. (2016). Past, present, and future of SLAM: Towards the robust-perception age. IEEE Transactions on Robotics, 32(6), 1309–1332.

Feng, S., Li, Y., Zhang, H., & Chen, J. (2025). A survey of multi-view stereo 3D reconstruction algorithms. Expert Systems with Applications, 269, Artículo 124505.

Karabiber, O., Yılmaz, B., & Yıldırım, S. (2025). Are different photogrammetry applications on smartphones sufficiently reliable? Measurement, 239, Artículo 114573.

Luhmann, T., Robson, S., Kyle, S., & Boehm, J. (2019). Close-range photogrammetry and 3D imaging (3.<sup>a</sup> ed.). De Gruyter.

Ma, Y., Soatto, S., Košecká, J., & Sastry, S. (2012). An invitation to 3D vision: From images to geometric models. Springer.

Müller, T., Evans, A., Schied, C., & Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics, 41(4), 1–15. <https://doi.org/10.1145/3528223.3530127>

Patonis, P. (2024a). A comparative study on the use of smartphone cameras in photogrammetry applications. Remote Sensing Letters, 15(6), 598–610.

F-DC-125

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE PROYECTO  
DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO, MONOGRAFÍA,  
EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 2.0

Patonis, P. (2024b). Comparative evaluation of the performance of a mobile device camera and a full-frame mirrorless camera in close-range photogrammetry applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 208, 34–49.

Poór, P., & Fang, Y. (2023). 3D tunnel reconstruction using multi-smartphone photogrammetry systems. *Measurement*, 222, Artículo 113511. <https://doi.org/10.1016/j.measurement.2023.113511>

Quispe-Enríquez, O. C., Pacheco, F., Morales, R., & Ureta, C. (2023). Craniofacial 3D morphometric analysis with smartphone-based photogrammetry. *3D Body Processing*. National Center for Biotechnology Information (NCBI). <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10781299>

Seifert, L. B., Blum, A., Berneburg, M., & Heppt, M. V. (2024). Comparative accuracy of stationary and smartphone-based facial scan systems. *Journal of Clinical Medicine*, 13(22), 6678. <https://doi.org/10.3390/jcm13226678>

Szeliski, R. (2022). *Computer vision: Algorithms and applications* (2.<sup>a</sup> ed.). Springer.

Tewari, A., Fried, O., Thies, J., Sitzmann, V., Lombardi, S., Sunkavalli, K., Ricardo, M., Nießner, M., Saragih, S., & Theobalt, C. (2021). Neural rendering: From learning-based methods to applications. *Foundations and Trends in Computer Graphics and Vision*, 12(2-3), 155–351.

Wang, F., Yan, K., Xie, W., Chen, J., Li, Y., Zhu, S., & Zhang, H. (2024). Learning-based multi-view stereo: A survey. *arXiv preprint arXiv:2408.15235*.

Wu, J., Wyman, O., Tang, Y., Pasini, D., & Wang, W. (2024). Multi-view 3D reconstruction based on deep learning: A survey and comparison of methods. *Neurocomputing*, 582(3), 127553.

## 9. ANEXOS

**Figura A 9.1 - Flujo de navegación del sistema**



Fuente: Elaboración propia

**Figura B 9.1 - Arquitectura del sistema de reconstrucción 3D**



Fuente: Elaboración propia

**Figura C 9.1 - Estructura funcional del sistema**



Fuente: Elaboración propia

### 9.1. Historial de errores y fallos del sistema de procesamiento

**Figura D 9.1 – Registro de errores obtenidos desde archivos meta y logs del sistema**



Fuente: Elaboración propia

**Descripción:**

En esta etapa se llevó a cabo un análisis detallado del comportamiento real del sistema a partir de los registros generados durante su ejecución. El enfoque se centró en los logs obtenidos durante el procesamiento, lo que permitió observar cómo responde el sistema en condiciones reales de uso.

Se encontró que los errores no aparecían de manera aleatoria, sino que estaban asociados a etapas específicas del pipeline, especialmente durante la extracción de características (feature extraction) y el proceso de mapeo (mapper). El sistema registra cada fallo de forma estructurada, indicando con precisión en qué etapa ocurre, lo que facilita el acceso a información más detallada para su análisis.

A partir de estos registros, fue posible identificar las causas que impedían la correcta generación del modelo. En muchos casos, los problemas no estaban directamente relacionados con el algoritmo, sino con las condiciones en las que se realizaba la captura, como iluminación deficiente, ángulos poco adecuados o baja calidad en las imágenes.

De igual forma, las limitaciones del hardware influyeron en el comportamiento del sistema, afectando el rendimiento en algunas fases del procesamiento.

En conjunto, este análisis permitió evidenciar que los logs cumplen un papel clave en la identificación de fallos, ya que hacen posible ubicar con mayor claridad el punto del proceso en el que se presentan los errores antes de la generación del modelo tridimensional.

### Figura D 9.2 - Error en la lectura de imágenes durante la extracción de características



Fuente: Elaboración propia

#### Descripción:

La imagen presentada muestra el punto exacto en el que el proceso se detiene durante la etapa de extracción de características. En ese momento, el sistema genera el siguiente error:

Bitmap::Read: failed to read the image file format

A partir de este fallo, fue necesario revisar las imágenes de forma individual para identificar la causa del problema. Durante este proceso se encontró que algunos archivos no podían ser procesados, ya que el sistema no los reconocía como válidos para la etapa de análisis.

En las pruebas realizadas con imágenes capturadas en laboratorio se detectaron archivos con formatos incorrectos o posibles daños generados durante la transferencia. Estas inconsistencias provocaron fallos desde las primeras etapas del pipeline, impidiendo que el proceso avanzara correctamente.

Este comportamiento deja en evidencia que, si una imagen no puede ser procesada, no es posible extraer puntos de interés (SIFT). Como consecuencia, no se generan coincidencias entre imágenes, lo que bloquea completamente el avance del proceso de reconstrucción.

A partir de lo observado, se hace necesario validar previamente el estado de las imágenes antes de iniciar el procesamiento, asegurando su integridad y compatibilidad.

En este contexto, si una imagen no puede ser interpretada correctamente por el sistema, la reconstrucción tridimensional simplemente no puede llevarse a cabo.

**Figura D 9.3 - Error por ausencia de correspondencias entre imágenes**

```

1 I20260406 10:06:29.001850 3672 incremental_pipeline.cc:265] Loading database
2 I20260406 10:06:29.003597 3672 database_cache.cc:72] Loading rigs...
3 I20260406 10:06:29.003924 3672 database_cache.cc:82] 0 in 0.000s
4 I20260406 10:06:29.003971 3672 database_cache.cc:90] Loading cameras...
5 I20260406 10:06:29.004013 3672 database_cache.cc:108] 0 in 0.000s
6 I20260406 10:06:29.004040 3672 database_cache.cc:116] Loading frames...
7 I20260406 10:06:29.004077 3672 database_cache.cc:126] 0 in 0.000s
8 I20260406 10:06:29.004102 3672 database_cache.cc:134] Loading matches...
9 I20260406 10:06:29.004141 3672 database_cache.cc:139] 0 in 0.000s
10 I20260406 10:06:29.004169 3672 database_cache.cc:147] Loading images...
11 I20260406 10:06:29.004237 3672 database_cache.cc:241] 0 in 0.000s (connected 0, loaded 0)
12 I20260406 10:06:29.004269 3672 database_cache.cc:255] Loading pose priors...
13 I20260406 10:06:29.004305 3672 database_cache.cc:263] 0 in 0.000s
14 I20260406 10:06:29.004329 3672 database_cache.cc:271] Building correspondence graph...
15 I20260406 10:06:29.004904 3672 database_cache.cc:300] in 0.001s (ignored 0)
16 I20260406 10:06:29.004940 3672 timer.cc:90] Elapsed time: 0.000 [minutes]
17 W20260406 10:06:29.004968 3672 incremental_pipeline.cc:302] No images with matches
18 E20260406 10:06:29.005295 3672 sfm.cc:279] Failed to create any sparse model
  
```

Fuente: Elaboración propia

**Descripción:**

En esta etapa del proceso se presenta un bloqueo total durante la reconstrucción. Aunque el sistema logra cargar las imágenes, no consigue establecer correspondencias entre ellas.

El registro generado muestra los siguientes mensajes:

No images with matches

Failed to create any sparse model

A partir de este resultado, se identificó que las imágenes no compartían suficientes puntos en común, lo que impide establecer las relaciones geométricas necesarias para avanzar en la reconstrucción.

Durante las pruebas realizadas en laboratorio se observaron varias condiciones que afectaban directamente este proceso, entre ellas el bajo traslape entre imágenes, la presencia de superficies con poca textura, una iluminación deficiente y ángulos de captura poco adecuados.

El funcionamiento del algoritmo depende de la existencia de puntos coincidentes entre las imágenes para poder realizar la alineación. Cuando estas correspondencias no están presentes, no es posible estimar la posición de las cámaras ni reconstruir la estructura de la escena.

Lo anterior pone en evidencia la necesidad de mejorar las condiciones de captura, prestando mayor atención a la planificación de las tomas y al entorno en el que se realiza la adquisición de datos.

En estas condiciones, sin correspondencias entre imágenes, la generación del modelo tridimensional no puede llevarse a cabo.

## **9.2. Análisis general.**

Durante las pruebas realizadas en laboratorio se observó que la aplicación responde de forma adecuada en las etapas iniciales, especialmente en la captura y validación de imágenes. No obstante, al ejecutar el procesamiento mediante COLMAP y el pipeline de Structure from Motion, el comportamiento del sistema empezó a mostrar variaciones.

Estas variaciones no se explican por una única causa, sino por la interacción de varios factores. En algunos casos, los modelos generados resultaron incompletos, mientras que en otros no fue posible obtener ningún resultado.

Además, el hardware disponible tuvo un impacto directo en la ejecución del proceso, afectando tanto los tiempos de procesamiento como la estabilidad en ciertas etapas.

A partir del análisis realizado, se puede establecer que la calidad de los resultados depende en gran medida de las condiciones de entrada. Aspectos como el nivel de traslape entre imágenes, la iluminación y los ángulos de captura influyen directamente en la capacidad del sistema para generar reconstrucciones válidas.

Cuando estas condiciones no se cumplen, el sistema no cuenta con la información necesaria para ejecutar el proceso de manera adecuada.

En términos generales, el sistema logra funcionar correctamente, pero su desempeño óptimo depende de contar con condiciones de captura relativamente controladas.

Finalmente, se concluye que la reconstrucción tridimensional es altamente sensible a errores en la adquisición de los datos, lo que repercute directamente en la calidad del modelo generado.

### **9.3. Manual de Usuario.**

Este manual describe el uso básico del sistema desarrollado.

#### **Pantalla de inicio**

**FIGURA M-U 1**



Fuente: Elaboración propia

### Descripción:

Pantalla principal donde el usuario gestiona todo el flujo.

### Funcionalidades:

- Nuevo proyecto: Crear un nuevo escaneo
- Abrir captura: Iniciar proceso de captura
- Revisar proyectos: Ver proyectos existentes
- Ver modelos: Acceder a modelos generados

### Proceso de captura

**FIGURA M-U 2**



Fuente: Elaboración propia

### Descripción:

La aplicación guía al usuario para capturar imágenes alrededor del objeto.

### Instrucciones:

- Mantener el objeto dentro del anillo guía
- Seguir el ángulo indicado (ej: 330°)
- Capturar imágenes en diferentes niveles (bajo, medio, alto)
- Completar el progreso requerido (ej: 24/24)

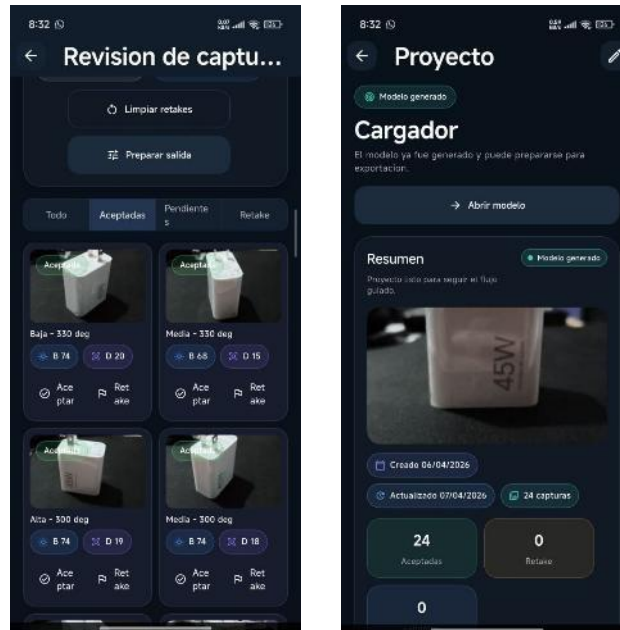
### Indicadores clave:

- Nivel de captura
- Ángulo actual
- Distancia

- Progreso total

## Revisión de capturas

FIGURA M-U 3



Fuente: Elaboración propia

### Descripción:

Permite validar las imágenes antes del procesamiento.

### Funcionalidades:

- Ver imágenes capturadas
- Aceptar o marcar para retoma (retake)
- Filtrar por estado (aceptadas, pendientes, retake)
- Indicadores:
- Brillo (B)
- Distancia (D)

- Estado de cada imagen.

## Procesamiento del modelo

FIGURA M-U 4



Fuente: Elaboración propia

- **Descripción:**

Seguimiento del procesamiento del modelo en backend.

### Funcionalidades:

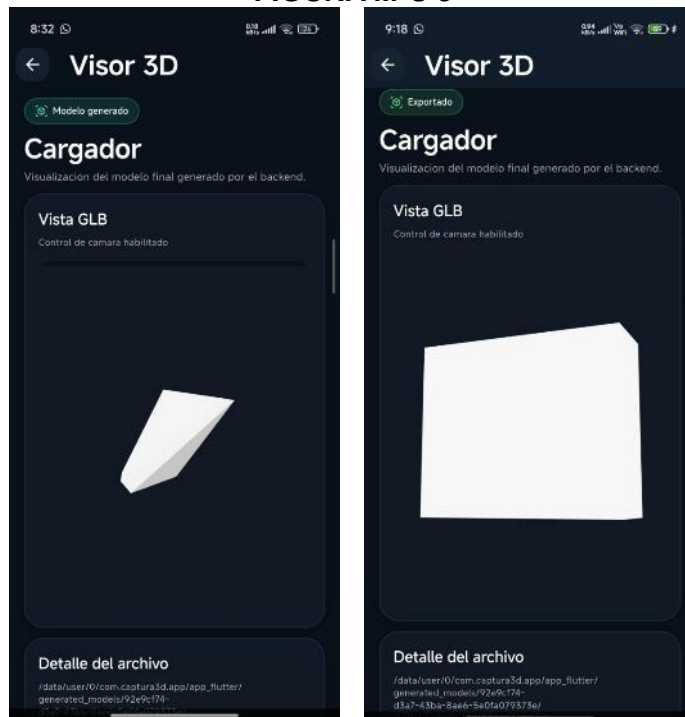
- Ver progreso del pipeline
- Identificador del proceso
- Actualización manual o automática

### Estados:

- En proceso
- Completado

## Visualización del modelo 3D

FIGURA M-U 5



Fuente: Elaboración propia

Descripción:

Permite visualizar el modelo generado.

Funcionalidades:

- Rotar el modelo
- Control de cámara
- Visualizar archivo GLB

Al final, cuando se llevaron las pruebas al laboratorio, se vio que la aplicación sí cumple en lo básico: permite capturar las imágenes y validarlas sin mayor problema. Esa parte fluye. Pero al momento de entrar al procesamiento con COLMAP y todo el pipeline de SfM, el comportamiento del sistema presenta variaciones.

Se notó que la reconstrucción 3D no siempre respondía como se esperaba. Hubo casos donde el modelo salía incompleto, otras veces directamente no se generaba nada. Y no era un fallo único, era una combinación de factores. El procesamiento en el equipo disponible también influía; no siempre daba la talla y eso alargaba los tiempos o dejaba procesos a medias.

Lo que Los resultados permiten establecer, sin darle muchas vueltas, es que la calidad del resultado depende más de la entrada que del algoritmo en sí. Si las imágenes no tienen buen traslape, si la iluminación no ayuda o si los ángulos no están bien pensados, el sistema simplemente no tiene con qué trabajar. Dicho de otra forma, el software cumple, pero exige condiciones que en la práctica no siempre se logran fácilmente en el entorno del laboratorio.

Al final, más que un problema del desarrollo, esto expone una limitación real del proceso completo: la reconstrucción 3D no es tolerante a errores en la captura. Y eso impacta directo en la calidad del modelo generado.