



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

AUTOMATIZACIÓN DEL PROCESO DE RECONSTRUCCIÓN DE
MODELOS 3D A PARTIR DE IMÁGENES O VIDEOS ALMACENADOS
EN PLATAFORMA WEB PARA EXPOSICIONES EN MUSEO VIRTUAL

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

Alfredo Alejandro Castillo Gutiérrez

PROFESOR GUÍA:
Nelson Baloian Tataryan

MIEMBROS DE LA COMISIÓN:
Juan Álvarez Rubio
Nancy Hitschfeld Kahler

SANTIAGO DE CHILE
2023

Resumen

Las tecnologías y la digitalización han tenido un gran crecimiento durante los años, viéndose reflejado en la presencia de cámaras en casi todos los dispositivos de las personas, o también en el aumento de la producción y utilización de la información en línea. Este crecimiento ha impulsado al desarrollo y avance de tecnologías, donde una de ellas es la reconstrucción 3D, ya sea para la reconstrucción de terrenos que utilizan la fotogrametría con la ayuda de drones, la impresión 3D de objetos para distinto uso, o bien para exposiciones virtuales.

Dado este contexto, se conoce además la existencia de un museo virtual que trabaja con un solo tipo de objeto, por lo que surge la idea de extender este museo para que acepte cualquier tipo de objeto mediante herramientas de reconstrucción 3D, y aumentar su catálogo, mediante colaboración colectiva, para su posterior uso en ámbitos pedagógicos o de entretenimiento. Sin embargo, si bien existen estas tecnologías de reconstrucción, actualmente se necesitan de conocimientos no tan básicos para poder ejecutar estas herramientas, por lo que un usuario promedio no sería capaz de generar su propio modelo 3D.

Con esta problemática, en este trabajo se diseñó e implementó un flujo de trabajo automatizado que abarca desde la subida de un video por parte de un usuario a una plataforma web, hasta la reconstrucción 3D de objetos para su uso en exposiciones en el museo virtual existente, pero en un escenario totalmente nuevo donde el usuario pueda ver un catálogo con los objetos reconstruidos por las personas, seleccionar un objeto, y colocarlo en el escenario. El hecho de ser automatizado, quiere decir que por parte del usuario requiera solamente subir un video a una plataforma web, lo más simple y fácil de utilizar, y que obtenga como resultado final su objeto dentro del museo.

Para esto, se realiza un trabajo de ingeniería de software, guiándose por una metodología de desarrollo, planteando requisitos de usuario y de software para satisfacer los requerimientos de cada pilar y finalmente del proyecto completo, el cual fue dividido en tres pilares fundamentales: página web, reconstrucción 3D y museo virtual.

Finalmente, se muestran los resultados obtenidos en cada uno de los requisitos y se indica si se cumplieron o no, realizando luego validaciones a la página web, donde se obtienen resultados positivos y satisfactorios, además de sugerencias por parte de usuarios encuestados, con las cuales se hacen arreglos post-feedback para obtener un trabajo final óptimo.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que me han ayudado en la realización de esta tesis. En primer lugar, a mi familia, especialmente a mis padres, Alfredo y Nayadet, por su amor, comprensión y apoyo en todo momento. Han sido mi fuente de motivación para nunca rendirme, dándome consejos de vida y demostrándome con hechos y palabras que siempre se puede y todo pasa por algo. A mi hermano, Pedro, por su apoyo y por compartir conmigo sus experiencias y conocimientos para que yo pudiese seguir adelante. Has sido mi inspiración para alcanzar mis metas y un ejemplo de persona a seguir. Finalmente, a mi pareja, Camila, por apoyarme, entenderme y nunca dejar de creer en mí durante todo este proceso. Estoy orgulloso de tenerte a mi lado aconsejándome y presionándome también para que siga luchando.

Agradezco a mis amigos, Sebastián e Ignacio, por su apoyo y ánimo durante todo el proceso. Fueron importantes en mi motivación al esperar grandes cosas de mí e incitarme a seguir adelante pasase lo que pasase.

No puedo dejar de mencionar a todas las personas que participaron en mi trabajo, especialmente a Gabriel, por su tiempo y disposición para responder mis preguntas y compartir sus conocimientos y experiencias. También agradezco a la universidad, por darme la formación que tengo actualmente y por las experiencias de vida que tuve, como muchos ramos deportivos que tomé, las clases de baile, las conversaciones con mis compañeros, y los lazos que generé con muchas personas durante mi paso por la universidad.

Por último, al profesor Nelson, por su dedicación, orientación y paciencia durante todo el proceso de investigación y desarrollo del trabajo. Gracias por su apoyo, sus consejos, por transmitirme su pasión por este tema, y por permitirme participar en uno de sus proyectos.

Muchas gracias a todos.

Tabla de contenido

1. Capítulo I: Introducción	1
1.1. Contexto y problema	1
1.2. Objetivos	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Específicos	2
1.3. Contenido del documento	3
2. Capítulo II: Estado del Arte	4
2.1. Generación de objetos 3D a través de Fotogrametría	4
2.2. Técnicas de Deep Learning	5
2.3. Colaboración en Museos Virtuales	5
2.4. Crowdsourcing de objetos culturales 3D para Museos Virtuales	6
2.5. Ingeniería de Software	7
2.5.1. Metodologías de desarrollo de software	7
2.6. Desarrollo web	8
2.6.1. Frontend	9
2.6.2. Backend	9
3. Capítulo III: Problema	11
3.1. Actualidad: Museo virtual	11
3.2. Herramientas de reconstrucción 3D	12
3.3. Requisitos para una solución	12
3.3.1. Requisitos de usuario	13
3.3.2. Requisitos de software	13
4. Capítulo IV: Diseño	15
4.1. Diseño del pipeline	15
4.1.1. Captura video del objeto	16
4.1.2. Página web	17
4.1.3. Extracción de frames	17
4.1.4. Reconstrucción modelo 3D	18
4.1.5. Limpieza modelo 3D	18
4.1.6. Subida a bundles	18
4.1.7. Museo virtual	19
5. Capítulo V: Implementación	19
5.1. Página web	20
5.1.1. Secciones	21
5.1.2. Frameworks y Librerías	23
5.2. Reconstrucción 3D	24
5.2.1. Opciones de métodos de captura de video	24
5.2.2. Preprocesamiento 1: Extracción de frames	24
5.2.3. Meshroom (Preprocesamiento 1)	25
5.2.4. OpenDroneMap (Preprocesamiento 1)	26
5.2.5. Preprocesamiento 2: Rembg	27
5.2.6. Meshroom y ODM (Preprocesamiento 2)	28

5.2.7. Instant Neural Graphics Primitives	28
5.2.8. Conclusiones acerca de las posibilidades de reconstrucción	29
5.3. Museo	29
5.4. Solución de requisitos	30
6. Capítulo VI: Validación	36
6.1. Metodologías de evaluación	36
6.1.1. Parámetros a evaluar	37
6.1.2. Resultados	37
6.2. Conclusiones e Implementación Post-tests	39
7. Capítulo VII: Conclusiones y Trabajo futuro	40
Bibliografía	42

1. Capítulo I: Introducción

1.1. Contexto y problema

En los últimos años, la digitalización ha tenido un crecimiento importante debido a diversos factores. Uno de los motivos principales es la creciente cantidad de información que se produce y se utiliza en línea, además de los avances tecnológicos, como la presencia de cámaras en casi todos los dispositivos que tienen las personas, y el aumento en la calidad de captura de imágenes y videos. Todo esto, con ayuda de las nuevas tecnologías disponibles, y junto a la necesidad e interés de preservación del conocimiento y la cultura, han impulsado el surgimiento de nuevas ideas, como son los museos virtuales, que se han convertido en una alternativa cada vez más popular, permitiendo que la gente pueda explorar exposiciones que se encuentran en ubicaciones remotas sin tener que gastar dinero en viajar. Además, la pandemia de los últimos años ha llevado a muchas personas a buscar formas de explorar el mundo sin tener que salir de sus hogares, y las exposiciones virtuales han sido una opción atractiva, ya que ofrecen entre muchas cosas la posibilidad de ahorrar tiempo al permitir a los usuarios explorar diferentes museos en un solo lugar.

En este contexto, surge la posibilidad de que se creen los museos virtuales, los que permiten la participación activa de los usuarios en la creación de las exposiciones. Esto significa que las personas pueden aportar su propio material o incluso crear sus propias exposiciones utilizando los recursos disponibles. De esta manera, el museo virtual se convierte en un espacio colaborativo donde el conocimiento y la creatividad de las personas se unen para crear nuevas y emocionantes exposiciones. Además, el museo virtual ofrece una ventaja sobre los museos tradicionales, ya que no limita la cantidad de información que se puede recopilar. En un museo clásico, los propietarios son los únicos responsables de proporcionar el material para las exposiciones, lo que puede limitar la cantidad de información disponible. En cambio, el museo virtual puede preservar el conocimiento cultural al permitir que las personas de todo el mundo contribuyan con material para las exposiciones. En general, los museos virtuales ofrecen una manera emocionante y accesible de explorar la cultura y la historia sin tener que salir de casa, además de permitir la participación activa de los usuarios, generando en un espacio de colaboración y creatividad que fomenta el intercambio de conocimientos y la preservación de la cultura a nivel global.

La idea ante esta posibilidad es permitir que las personas capturen objetos que les resulten interesantes en video, para luego subir estos archivos a través de una plataforma web y, finalmente, obtener un modelo 3D del objeto en cuestión. Esta tecnología también podría ser utilizada en el ámbito educativo, por ejemplo, en una clase de arte para crear una exposición de objetos cotidianos o estatuas. Para lograrlo, se grabaría un video de cada objeto y, después de generar los modelos 3D de cada uno, se podría crear una exposición virtual en un escenario, como por ejemplo en la misma sala de clases pero representada de manera virtual, para compartirla con los compañeros. En resumen, esta tecnología permitiría a las personas capturar y compartir objetos en 3D de una manera innovadora y también ofrece una nueva forma de crear y compartir exposiciones virtuales.

En la actualidad, la *fotogrametría* es una técnica que permite reconstruir objetos en modelos 3D a partir de imágenes o cuadros de videos, pero su complejidad hace que sea difícil para una persona sin conocimientos técnicos adecuados llevar a cabo este proceso, además

de ser necesarios procedimientos que requieren de softwares con alto costo, en ciertas ocasiones, económico o de hardware. Es importante tener en cuenta que el software que genera el modelo 3D puede incluir información no deseada, como el fondo del objeto proveniente de un video, lo que puede dificultar la eliminación de elementos no deseados en la reconstrucción. Además de la fotogrametría, existen otras técnicas que pueden mejorar la experiencia de crear modelos 3D, como la reconstrucción mediante el uso de datos de Deep Learning [18], pero que requiere de un mayor costo de hardware. En resumen, aunque la fotogrametría es una técnica útil para la reconstrucción de modelos 3D, su complejidad y la necesidad de habilidades técnicas pueden limitar su uso, así como pueden ser necesarias técnicas adicionales para mejorar la calidad y facilidad del proceso.

Por lo tanto, el presente trabajo de título consiste crear un flujo de trabajo, al cual denominaremos *pipeline*, que aborde los problemas y consideraciones mencionados anteriormente. Este pipeline debe solucionar los problemas identificados y se evaluará el grado de automatización que se pueda lograr. Para esto, se estudiará la posibilidad de obtener un modelo perfecto, viendo si es posible procesar los archivos de manera inmediata o si se requiere un paso previo, y la flexibilidad del sistema para trabajar en diferentes escenarios de exposición. Todo esto, con el objetivo de determinar cuál es el grado máximo de automatización posible dado el estado del arte actual en la reconstrucción 3D de objetos, y también con el fin de permitir que tanto un usuario experto como no experto logre vivir tales situaciones expuestas (u otras) de la forma más sencilla posible.

1.2. Objetivos

1.2.1. Objetivo General

En este documento se presenta el trabajo de título cuyo objetivo general es desarrollar un pipeline lo mayormente automatizado posible, dado el estado de arte actual, que abarque la reconstrucción 3D de objetos cotidianos mediante colaboración colectiva, para su uso en exposiciones virtuales, con escenarios personalizados, por usuarios de todo tipo (expertos y no expertos).

1.2.2. Objetivos Específicos

Para lograr el objetivo general expuesto anteriormente, se establecen los siguientes objetivos específicos:

- Evaluar y seleccionar algoritmos existentes de fotogrametría y reconstrucción 3D de objetos.
- Crear una plataforma web que permita al usuario subir archivos y videos de objetos.
- Obtener reconstrucción 3D de objetos con archivos subidos a plataforma web.
- Visualizar en el museo virtual existente el modelo 3D obtenido por la reconstrucción.
- Requerir de máximo 1 o 2 interacciones humanas (no usuario) en el pipeline que abarca los puntos anteriores.
- Permitir al usuario personalizar ciertos escenarios de forma simple con artículos de decoración básicos.

1.3. Contenido del documento

Previo a la solución del problema, en el capítulo 3 se presentará un trabajo de ingeniería de software que divide las etapas y pilares fundamentales del proyecto en requisitos de usuario y de software. En el caso de los requisitos de usuario, se explicará de forma detallada cuál es el requisito, en qué consiste, cuál es su prioridad, y cómo el usuario interactúa directamente con el proyecto. Luego, para los requisitos de software, también se realizará una explicación de cada uno. Este trabajo seguirá una metodología de desarrollo de software que se justificará previamente en el estado del arte.

Posteriormente, en los capítulos 4 y 5 se explicará de forma muy detallada cada arista del proyecto, ya sean las que interactúan directamente con el usuario, o bien los métodos internos de funcionamiento para que el pipeline diseñado se ejecute de forma correcta. Para esto, primero se explicará el diseño del pipeline creado, explicando cada una de sus etapas, luego se presentará la implementación del proyecto, detallando los tres pilares fundamentales en los que se divide el trabajo, y finalmente se repasarán los requisitos de software, indicando su estado final de resolución.

Luego, en el capítulo 6 se llevará a cabo una evaluación exhaustiva de la solución desarrollada. Para ello, se reclutará un grupo de personas con perfiles de usuario variados y se aplicarán metodologías de evaluación previamente seleccionadas. Durante el proceso de evaluación, se llevará un registro detallado de los resultados obtenidos y se realizará un análisis exhaustivo de los mismos. Una vez concluida la evaluación, se presentarán los resultados y se extraerán conclusiones importantes que permitan mejorar la solución desarrollada. Además, se indicarán algunas implementaciones post-tests que se hayan realizado, con el fin de garantizar la máxima calidad en la solución final.

Finalmente, en el capítulo 7 se mostrarán las conclusiones sobre el trabajo realizado, con el recuento de los objetivos alcanzados y no alcanzados, y para terminar habrá una sección de trabajo futuro en el cual se plantearán desafíos que el presente trabajo de título propondrá a realizar, indicando cómo se relaciona con lo ya realizado, y cómo se podría mejorar el mismo.

2. Capítulo II: Estado del Arte

Durante los años y gracias a los avances tecnológicos que se van presentando en cada momento, se han dado a conocer nuevas ideas para permitirle a las personas tener la posibilidad de obtener información de todo tipo mediante internet. Luego, en el ámbito cultural, ha surgido el concepto de los museos virtuales, las cuales son plataformas digitales que permiten a las personas explorar y experimentar exposiciones y colecciones de arte y patrimonio de manera virtual. A través de tecnologías como la visualización 3D, la realidad aumentada o incluso la realidad virtual, los museos virtuales ofrecen una experiencia inmersiva que simula la visita a un museo físico. Los visitantes pueden recorrer galerías, acercarse a las obras de arte, obtener información detallada y enriquecer su conocimiento cultural. Los museos virtuales amplían el acceso a la cultura y el arte, superando las barreras geográficas y temporales, y brindando una alternativa interactiva y educativa para la apreciación del patrimonio cultural.

Por otro lado, el crowdsourcing, que en español podría ser traducido como el concepto de “colaboración abierta”, es una herramienta de colaboración masiva que genera la inclusión de personas externas a un proyecto, con el fin de que ayuden a generar contenido para éste. Con las tecnologías existentes, esta herramienta está siendo cada vez más utilizada para diferentes fines, donde uno de ellos es la preservación de objetos culturales. Esto va directamente relacionado con los museos o exposiciones virtuales que permiten a los usuarios interactuar con objetos de todo el mundo, los cuales en ocasiones ya no existen, desde un computador u otro dispositivo.

Con estos dos conceptos, en este capítulo se presentan trabajos relacionados al crowdsourcing, los museos virtuales y la fusión de ambos, contextualizando así el resto de capítulos contenidos en la memoria.

2.1. Generación de objetos 3D a través de Fotogrametría

La fotogrametría [3] es una técnica que permite crear modelos 3D a partir de una serie de fotografías (o cuadros de video) de un objeto, tomadas desde diferentes ángulos. El proceso esencialmente consiste en la obtención de descriptores en las imágenes, el emparejamiento de estos entre diferentes fotografías y finalmente el cálculo del movimiento que tuvo la cámara entre estas. Con esto, se puede generar una nube de puntos que, al aplicarle algún algoritmo de generación de malla poligonal 3D, puede convertirse en un objeto 3D.

El desarrollo de la computación y la fotografía digital ha permitido explotar el potencial de estas técnicas, utilizándose en áreas como: adquisición de expresiones humanas, creación de escenarios para videojuegos, reconstrucción digital de objetos simétricos, geología, entre otros. Existen varios casos de estudio documentados en estas áreas, por ejemplo, el uso de drones para hacer fotogrametría de edificios históricos [15] y terrenos [6], el uso de cámaras de teléfonos móviles para el escaneo de fachadas históricas [10] y extracción de prótesis faciales [8] o reconstrucción de estructuras antiguas colapsadas a través de imágenes obtenidas por crowdsourcing [5]. También existen trabajos que buscan guiar a una buena fotogrametría, como Ch'ng [14] quien propone diversos tipos de prácticas dependiendo de lo que se quiere escanear y su ubicación (interior, exterior, elevado, etc).

Existe una gran variedad de software open source (COLMAP, VisualSFM, MicMacm, entre otros) y no open source (Agisoft Metashape, AutoDesk ReCap, WebODM, entre otros) que permiten realizar *Structure From Motion* (SFM) o parte de ella, tanto con fotografías como video. A raíz de esto, el trabajo propuesto en este documento es el de utilizar y adaptar el que más apto para cualquier objeto, con fines prácticos o incluso educativos.

2.2. Técnicas de Deep Learning

El deep learning, es un subconjunto de machine learning que se basa en algoritmos y modelos inspirados en la estructura y función del cerebro humano, conocida como red neuronal artificial. Estos modelos están compuestos por múltiples capas de unidades interconectadas, y son capaces de aprender y extraer características relevantes de los datos de entrada para realizar tareas específicas, como clasificación, reconocimiento de objetos, generación de texto, entre otras.

El uso de Deep Learning en la tarea de reconstrucción 3D de objetos es tan reciente que la primera encuesta centrada en la reconstrucción utilizando Deep Learning fue publicada en 2019 [18]. Sin embargo, estos métodos han mostrado obtener resultados prometedores, y han sido aplicados recientemente. Groueix et al., por ejemplo, han demostrado obtener buenos resultados usando datos posiblemente ruidosos [11], y muchos otros han mostrado gran mejora.

En los últimos años, ha tomado fuerza el uso de técnicas que hacen uso de redes neuronales, como la reconstrucción 3D para retratos, escenas dinámicas, mejorar la iluminación de objetos, entre otras. Tal como indica Frank Dellaert [16], esta “explosión” fue provocada en el año 2020 por el artículo NeRF [17] de Mildenhall et al.

El uso de estas técnicas en la generación de objetos 3D ha mostrado resultados prometedores en diversas aplicaciones. Estos enfoques permiten generar modelos 3D detallados y realistas, incluso a partir de datos de entrada limitados o ruidosos. Sin embargo, el campo sigue siendo objeto de investigación activa y se están explorando constantemente nuevas técnicas y arquitecturas para mejorar la calidad y la eficiencia de la generación de objetos 3D con Deep Learning.

Con este contexto, se motiva el estudio y la evaluación de técnicas de Deep Learning para su uso en el presente trabajo de título, puesto que podría ayudar en la automatización del pipeline a realizar, específicamente en el apartado de la reconstrucción haciendo uso de imágenes que contengan más elementos aparte del objeto principal que se quiera reconstruir.

2.3. Colaboración en Museos Virtuales

El agregar al usuario como un factor más en la creación o modificación de exposiciones virtuales es algo que se ha estado estudiando activamente desde inicio de siglo. Dos de los primeros casos de este tipo de interacciones fueron Kirner [1] en 2001 y Hall [2] en 2002. El primero desarrolló un ambiente colaborativo para propósitos educativos, donde alumnos de primaria podían crear, individual o grupalmente a través de una página web, sus propias exhibiciones 3D de objetos culturales y compartirlas, ya sea imprimiéndolas o a través del mismo sistema con otros estudiantes. El segundo implementó un sistema de realidad virtual

que permitía a visitantes de varios museos europeos poder cambiar las épocas temporales de las exhibiciones, dejando así que el flujo de la exposición esté controlado por los usuarios. Si bien estos casos de estudio tuvieron éxito, tenían ciertas restricciones que la tecnología de hoy ha eliminado, como por ejemplo, el poder ingresar al museo desde cualquier lugar del mundo y no solo donde está físicamente el sistema. Algunos conocidos museos como el *Lowre Museum* o el *The National Museum of Computing* tienen versiones virtuales gratuitas de parte de sus exposiciones, pero solo con la opción de verlas y no cambiarlas.

Baloian [9] [12] propone un sistema de museo virtual 3D en donde los visitantes puedan ser parte de las exposiciones, modificándolas y agregando información extra de los objetos presentes. El proyecto está implementado en un museo 3D de khachkars, un tipo de piedra conmemorativa del arte armenio, donde los usuarios pueden agregar, ver información, mover, orientar o eliminar piedras de una escena, agregar puntos de interés o simplemente ver una escena creada por otras personas. En base a esto, actualmente el museo tiene 5 escenas predefinidas, donde cada una es un paisaje distinto con piedras (khachkars) distribuidas de diferente forma. Lo único modificable en la escena por el usuario son las piedras en sí: su posición, orientación y agregar o quitar más piedras. El proceso de agregar más piedras a una escena está optimizado para que el proyecto no tenga que almacenar todas las piedras posibles, sino que las descargue convenientemente *on demand*, es decir, cuando se requiera. Para esto, se tienen varios *bundles*¹ de piedras hosteados en el mismo servidor de donde está corriendo el proyecto. Toda esta gestión es realizada por Gabriel Azócar. Además, Cristián Llul [21] se encuentra buscando un pipeline de reconstrucción 3D con Deep Learning, para mejorar la experiencia de reconstrucción, obteniéndose como resultados modelos con más detalles y profundidad debido al uso de varios cuadros de video, por ejemplo.

Sin embargo, en ninguno de estos casos anteriormente mencionados los usuarios tienen la posibilidad de agregar nuevos elementos al catálogo de objetos ya existentes, lo que aporta a la motivación de este trabajo.

2.4. Crowdsourcing de objetos culturales 3D para Museos Virtuales

En 2001, Barbieri [23] ya había descrito la importancia y gran potencial que tiene la cooperación en los museos virtuales, mencionando algunas de las plataformas que en ese entonces existían, como WebTalk, Net2Gether y Microsoft Virtual Worlds. Una década más tarde, Oomen y Arroyo vuelven a recalcar lo mismo, pero haciendo hincapié que en la época moderna compartir información es mucho más fácil que hace diez años atrás. YouTube, Wikipedia o Flickr son ejemplos de sitios que reciben un tráfico abismante de información por hora [4]. En 2015, Stathopoulou [5] tomó estas ideas y desarrolló una plataforma de crowdsourcing para lograr hacer un modelo 3D de un puente en Grecia que había colapsado. Con el uso de un sitio web, las personas podían subir imágenes, videos y metadata del ya desaparecido puente, lo que desembocó en alrededor de 500 archivos con los cuales se pudo hacer una reconstrucción digital completa del puente, a través de fotogrametría. Biella [7] presenta una visión general de crowdsourcing y co-curation (creación conjunta de una exposición) en museos virtuales permitiendo a personas seleccionadas (no profesionales del ámbito) y especialistas, crear exhibiciones o tours tanto en 2D como en 3D. Para este propósito propone ViMCOX (Virtual Museum and Cultural Object Exchange Format), un formato de metadata estándar

¹Paquete de elementos que se ofrecen como una unidad, y que cumplen un objetivo común.

basado en LIDO (Lightweight Information Describing Objects) que busca hacer la colaboración más estructurada. Para el proceso, identifica tres etapas en el proceso: Creación online, proposiciones y publicaciones de tours en la plataforma y discusiones online del contenido. Para averiguar el interés de la gente en la plataforma creada utilizaron encuestas, con escala del 1 al 5, cuyas preguntas se basaban principalmente en consultas sobre funcionalidad de la plataforma (si los audios se podían escuchar, si el movimiento en la exposición era fluido, etc) y sobre satisfacción con la plataforma (si gustó la exposición, si gustó el hardware utilizado para interactuar, etc). Finalmente, mencionan que tienen un proyecto asociado al descrito en [9] y [12] llamado DiKEViMA, el cual consiste en una página web donde las personas pueden aportar con khachkars procesados por ellos mismos. Sin embargo, para poder aportar, las personas deben tener conocimientos previos de digitalización de objetos o bien aprender de los tutoriales que la misma página sugiere, lo cual implica inversión de tiempo y puede hacer que los interesados pierdan el interés en colaborar. En contraste con lo anterior, Dhonju [13] propone un sistema automático de crowdsourcing de objetos 3D, reconstruidos a través del software comercial Agisoft PhotoScan. Este elimina la necesidad de algún grado de conocimiento experto del usuario al crear una plataforma móvil y web que permite subir una serie de fotos que serán reconstruidas automáticamente. Esta plataforma permite visualizar los objetos subidos, localizar en un mapa dónde se encuentran y además ver las imágenes originales que ayudaron a crear el modelo 3D.

Con este contexto, son estos fines prácticos de automatización, entre otros, los que se quieren abordar en el presente documento de trabajo de título.

2.5. Ingeniería de Software

La ingeniería de software es una disciplina que aplica principios y técnicas de ingeniería al proceso de desarrollo de software con el fin de producir productos de alta calidad de manera eficiente. Según Roger S. Pressman[24], la ingeniería de software es una disciplina que abarca todos los aspectos de la producción de software, desde la concepción del sistema hasta su entrega y mantenimiento. Por lo tanto, se trata de un campo en constante evolución y desarrollo, con un impacto cada vez más significativo en la industria del software y en la sociedad en general.

Una forma común de abordar un proyecto mediante ingeniería de software es mediante el uso de una metodología de desarrollo de software. En la siguiente sección, se explicarán algunas de estas metodologías.

2.5.1. Metodologías de desarrollo de software

Una metodología de desarrollo de software es un conjunto de técnicas y prácticas que se utilizan para gestionar el proceso de creación de un software. En general, las metodologías de desarrollo de software buscan mejorar la eficiencia, calidad y fiabilidad del proceso de desarrollo de software.

Algunos ejemplos comunes de metodologías de desarrollo de software son:

1. **Modelo en Cascada:** El modelo en cascada es una metodología de desarrollo de software que se basa en un enfoque secuencial en el que las etapas del proceso de

desarrollo se llevan a cabo en orden, una tras otra. Este enfoque se divide en fases secuenciales, como la planificación, el análisis, el diseño, la implementación, las pruebas y el mantenimiento. Cada fase debe completarse antes de avanzar a la siguiente y el proceso de desarrollo se lleva a cabo de manera secuencial. Este modelo es útil para proyectos con requisitos claramente definidos y cambiantes mínimos durante el ciclo de vida del proyecto.

2. **Desarrollo iterativo:** El enfoque de desarrollo iterativo implica la división del proceso de desarrollo de software en iteraciones o ciclos. Cada ciclo implica la realización de actividades de diseño, implementación y pruebas. Después de cada iteración, el equipo de desarrollo evalúa el software y los requisitos del cliente para ajustar el plan de desarrollo. Este enfoque es útil en proyectos donde los requisitos del cliente no están completamente definidos al inicio del proyecto y se espera que cambien a lo largo del ciclo de vida del proyecto.
3. **Desarrollo ágil:** El enfoque de desarrollo ágil es un enfoque iterativo e incremental que se enfoca en la entrega temprana y continua de software funcional. El equipo de desarrollo trabaja en colaboración con el cliente para identificar y entregar características importantes en ciclos cortos. El enfoque ágil se centra en la flexibilidad, la retroalimentación continua del cliente y la adaptación a los cambios en los requisitos del cliente. Los marcos ágiles comunes incluyen Scrum, Kanban y Extreme Programming (XP).
4. **Modelo V:** El modelo V es una variante del modelo en cascada que se centra en la validación y verificación del software. Cada etapa del desarrollo se asocia con una etapa de pruebas correspondiente. Este modelo implica el desarrollo de pruebas tempranas y la integración temprana de componentes. Las pruebas se realizan en el nivel del sistema y se aseguran de que el software cumpla con los requisitos del cliente. El modelo V se utiliza en proyectos donde la calidad del software es una prioridad.
5. **Desarrollo en espiral:** El enfoque de desarrollo en espiral es un enfoque iterativo que se enfoca en la gestión de riesgos. El proceso de desarrollo se divide en ciclos, cada uno de los cuales incluye una evaluación de riesgos y una revisión de las decisiones de diseño. El equipo de desarrollo trabaja en colaboración con el cliente para identificar los riesgos del proyecto y desarrollar un plan para gestionarlos. Este enfoque es útil en proyectos grandes y complejos donde la gestión de riesgos es crítica.

Para el proyecto se tiene como primera opción tomar el Modelo en Cascada, pero considerando que el proyecto se debe realizar en un periodo corto de tiempo, también se tomarían metodologías de desarrollo ágil, puesto que se necesita tener un feedback constante durante el desarrollo. Es por esto, que se opta finalmente para el proyecto por el Modelo V, ya que la etapa de pruebas por etapa de desarrollo que propone el modelo se adapta a los objetivos del presente trabajo.

2.6. Desarrollo web

El desarrollo web es el proceso de crear aplicaciones, sitios web y otras plataformas de software que se ejecutan en un navegador web o aplicación móvil. Esto implica una variedad de tareas técnicas, como la creación de interfaces de usuario, la conexión con bases de datos, entre otras. El desarrollo web también puede implicar la creación de contenido digital, como gráficos y contenido multimedia.

Este concepto a menudo se divide en dos categorías principales: el desarrollo del lado del cliente, o frontend, y el desarrollo del lado del servidor, más conocido como backend. El frontend se refiere a todo lo que el usuario final ve y experimenta en un sitio web, como el diseño y la funcionalidad de la interfaz de usuario. El backend, por otro lado, se refiere a la lógica y la funcionalidad que se ejecutan en el servidor para realizar tareas, como procesar datos o interactuar con bases de datos. Finalmente, ambos trabajan juntos para proporcionar una experiencia completa al usuario en la aplicación web.

En resumen, el desarrollo web es un proceso técnico que implica la creación de aplicaciones y sitios web para usuarios en línea y requiere una combinación de habilidades de diseño, programación y análisis de datos.

2.6.1. Frontend

Como se introdujo anteriormente, el frontend es la parte de la aplicación que se ejecuta en el navegador web del usuario. Para que esto funcione, se tiene una interfaz de usuario para la cual existen una variedad de frameworks² que ayudan a su creación. A continuación, se presentarán dos frameworks popularmente conocidos de los muchos existentes.

- **BootstrapVue**[19]: Es un framework de front-end basado en Bootstrap 4 y Vue.js. Permite la creación de interfaces de usuario responsivas y atractivas con una variedad de componentes preconstruidos. BootstrapVue es ampliamente utilizado en el desarrollo web y ha sido utilizado en proyectos de alto perfil como Uber y NASA.
- **React**[25]: Es un framework de JavaScript de código abierto desarrollado por Facebook que permite la creación de interfaces de usuario dinámicas y escalables. React utiliza un enfoque basado en componentes para la creación de aplicaciones web, lo que permite una mayor modularidad y reutilización de código. React es ampliamente utilizado en el desarrollo web y ha sido utilizado en proyectos de alto perfil como Netflix y Airbnb.

Como se puede ver, ambos frameworks ofrecen una experiencia similar a la hora de desarrollar una aplicación web por el lado de interfaz de usuario, por lo que para el presente caso se optará por el framework BootstrapVue, puesto que se cuenta con una experiencia previa en su uso, y también se considera que React se utiliza para proyectos mucho más complejos o para aplicaciones móviles.

2.6.2. Backend

Por otro lado, el backend se refiere a la parte de un sistema o aplicación que se encarga de procesar y almacenar los datos. En el contexto de la programación web, el backend se refiere específicamente al lado del servidor, que procesa las solicitudes de los clientes y devuelve las respuestas correspondientes. Al igual que para el frontend, existen variados frameworks que ayudan en la creación e implementación del backend. A continuación, se presentarán dos frameworks conocidos por el autor del presente trabajo, el cual será el encargado de llevar a cabo un desarrollo web, por lo que se mencionarán las siguientes opciones a considerar para trabajar en el proyecto.

²Conjunto de herramientas, librerías, convenciones y patrones de diseño que se utilizan para facilitar la creación de aplicaciones de software.

- **Django**[20]: Es un framework web de alto nivel basado en Python que sigue el patrón de diseño Modelo-Vista-Controlador (MVC). Proporciona un conjunto de herramientas y bibliotecas para crear aplicaciones web rápidas y seguras. Django es ampliamente utilizado en el desarrollo web empresarial y ha sido utilizado en proyectos de alto perfil como Instagram y Mozilla.
- **Nuxtjs**[22]: Es un framework de JavaScript basado en Vue.js que permite la creación de aplicaciones web rápidas y escalables. Nuxt.js es especialmente útil para la creación de aplicaciones de una sola página (Single Page Applications) y aplicaciones web universales (Universal Web Applications), que ofrecen una experiencia de usuario más fluida y rápida.

Como se puede notar, ambos frameworks ofrecen una experiencia similar a la hora de desarrollar una aplicación web por el lado del backend, pero para el presente caso donde se requiere de una experiencia fluida, rápida y sencilla para el usuario, se optará por el framework Nuxtjs para este proyecto, además que se cuenta con una experiencia previa en su uso y, adicionalmente, Nuxtjs se complementa de buena forma con BootstrapVue debido a la relación con Vue.js.

3. Capítulo III: Problema

Para abordar la descripción del problema, se presentarán tres elementos principales: el museo virtual, las herramientas de reconstrucción 3D, y el planteamiento de requisitos. En primer lugar, se abordará la actualidad del museo virtual ya existente, su actual funcionamiento y cómo es necesario contar con una solución que lo involucre. Luego, se introducirá la reconstrucción 3D, para la cual se deben investigar diferentes tecnologías y herramientas de reconstrucción a partir de imágenes o cuadros de video, ya que su resultado es el objeto a utilizar por los otros elementos de la solución a construir. Por último, se presentarán los requisitos para una solución a construir, dividiéndose en requisitos de usuario y software. Para estos tres elementos mencionados, se explicará el por qué forman parte del problema, cuánto se tiene desarrollado o investigado actualmente, y finalmente se especificará en todo detalle los requisitos de la solución a construir.

3.1. Actualidad: Museo virtual

Actualmente, el museo virtual ya creado trabaja solamente con khachkars, para los cuales se utilizan bundles, paquetes de elementos que se ofrecen juntos como una unidad, que almacenan los objetos y la información de cada uno. Estos bundles se deben crear de forma manual, y luego ordenar las carpetas y archivos que lo componen. Es por esto, que es necesario el abrirle nuevas oportunidades al museo y extenderlo de tal forma que pueda manejar todo tipo de objetos, incluyendo los actuales khachkars, con el fin de poder ampliar la funcionalidad del museo, pudiéndose utilizar en lugares como colegios para fines pedagógicos de variados contenidos, donde un usuario (alumno/profesor) pueda hacer por ejemplo una exposición en el ramo de Artes, o poder usarlo para exposiciones relacionadas a investigaciones arqueológicas, o bien como hobby, donde un usuario aficionado a los museos y exposiciones artísticas pueda recolectar objetos y crear la suya propia. Además, cabe destacar un concepto clave con el que se debe trabajar, el *crowdsourcing*, ya que esto permitiría que el catálogo del museo se extienda de una forma mucho más rápida, y así agrandar el abanico de posibilidades para crear exposiciones.

A continuación, se podrá ver el museo virtual existente, con su menú principal, donde es posible escoger uno de los 5 escenarios predefinidos que posee, y que al hacerle clic ingresa a la escena que contiene los khachkars ubicados a decisión del usuario, pudiéndose también ver la información de cada uno de ellos, y además poder insertar nuevos objetos presentes en el catálogo de khachkars disponible, o rotar y mover los ya colocados. Todos los cambios que se realizan en cada escenario pueden ser guardados para su posterior visita.

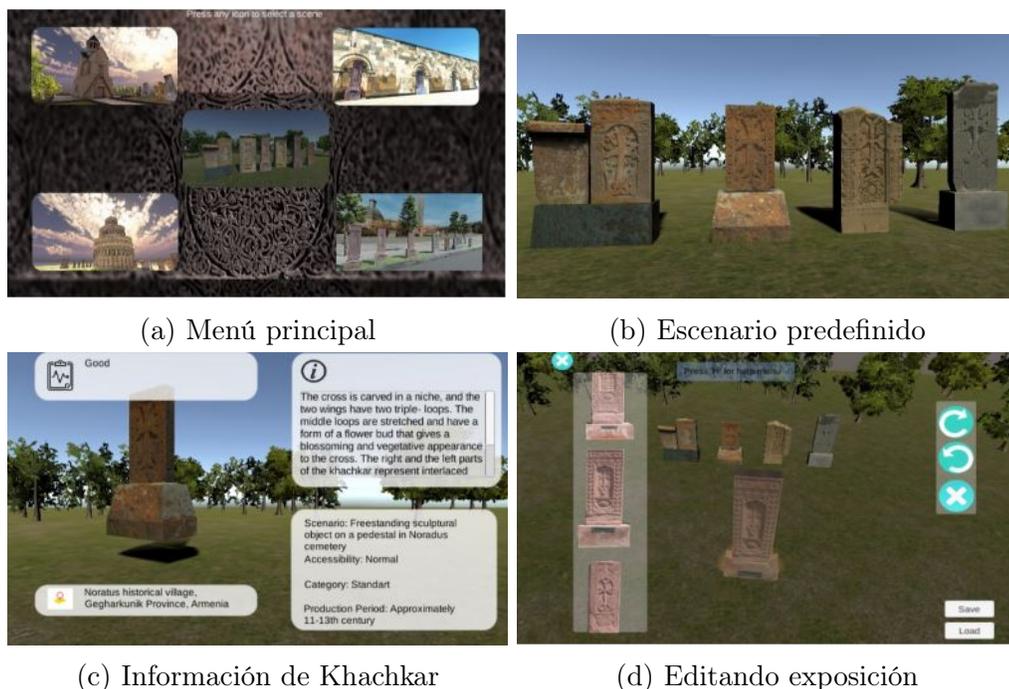


Figura 1: Museo de Khachkars existente

3.2. Herramientas de reconstrucción 3D

Como se explicó anteriormente, para subir un nuevo objeto al museo virtual se necesita crear de forma manual el bundle junto a los archivos que lo componen, entre ellos el *.obj* que contiene el objeto en su forma tridimensional. Luego, para crear este objeto se necesita de una herramienta de reconstrucción 3D, muchas de las cuales se encuentran en repositorios y como proyectos open-source, con las respectivas instrucciones para su montaje y funcionamiento. Sin embargo, para un usuario no experto en estos temas se le haría muy dificultoso seguir las instrucciones para ejecutar de forma correcta el algoritmo de reconstrucción plasmado en cada proyecto, por lo que es muy relevante contar con una solución para esto, es decir, crear un método que utilice alguna herramienta de reconstrucción 3D de objetos a partir de videos de forma automática, y que el usuario que desee utilizarla deba realizar la mínima intervención humana posible, dado el estado actual de la tecnología, facilitando así el proceso general.

3.3. Requisitos para una solución

Si se mira desde una perspectiva mucho más general, existe una serie de problemáticas las cuales promueven la idea de una solución en común, la cual es automatizar el proceso de permitirle a un usuario que pueda utilizar el museo ya existente, creando exposiciones con objetos que el mismo usuario decida, ya sean objetos propios o exportados del catálogo de algún museo virtual existente.

Para esto, es relevante crear un flujo de trabajo que comience con la grabación de un video a un objeto, subirlo a un servidor mediante una plataforma web, y que termine con el objeto reconstruido en el museo; flujo el cual interactuará con el usuario solamente mediante

la plataforma web y el museo, eliminando así la necesidad de la utilización de la herramienta de reconstrucción. Partimos de la idea de que el perfil esperado del usuario que usará esta plataforma web es más bien la de experto en arte, historia y/o arqueología y no un experto en computación, por lo cual el diseño de ella debe ser amigable y sencilla de aprender.

A continuación, se presentan de forma detallada una serie de requisitos de usuario, con su respectiva prioridad, los cuales buscan describir la problemática definida, y también requisitos de software, quienes estarán asociados a uno o más requisitos de usuario.

3.3.1. Requisitos de usuario

En este caso, los requisitos se centran en las necesidades, expectativas y comportamientos de los usuarios finales. Estos, describen las funcionalidades y características que los usuarios desean que el software proporcione para satisfacer sus objetivos y resolver sus problemas. A continuación, se presentan los siguientes requisitos de usuario:

1. **Tutorial para uso de plataforma y museo (urgente):** Es necesario que el usuario, sobre todo el no experto, cuente con indicaciones de cómo grabar el video, cómo subir el video y su metadata, y cómo observar finalmente el objeto en el museo. Esto se hará mediante imágenes que detallen el paso a paso del proceso donde el usuario deba interactuar.
2. **Permitir subir un video (urgente):** El usuario debe ser capaz de subir un video usando la página web, indicándose cuando el proceso sea exitoso.
3. **Poder agregar nombre al objeto que se subirá (urgente):** Al momento de completar el formulario, el usuario debe poder agregarle un nombre al objeto del cual grabó el video.
4. **Poder agregar descripción al objeto que se subirá (normal):** Al momento de completar el formulario, deberá ser posible también agregarle una descripción al objeto, la cual será visible en el museo al inspeccionar el objeto.
5. **Poder agregar medidas al objeto que se subirá (normal):** Al momento de completar el formulario, el usuario podrá elegir si quiere agregarle las dimensiones del objeto en caso de conocerlas.
6. **Permitir ver el objeto subido en museo (urgente):** Luego de un tiempo después de subir el video, el usuario debe poder ver el objeto reconstruido en 3D dentro del museo virtual.
7. **Colocar objeto en un escenario dentro del museo (normal):** El usuario podrá agregar a su escenario objetos del catálogo del museo, tanto propios como objetos subidos por otros usuarios.

3.3.2. Requisitos de software

Por otro lado, estos requisitos se enfocan en las características y funcionalidades que el software debe tener para cumplir con los requisitos de usuario, por lo que son más específicos y técnicos, definiendo las restricciones y los detalles de diseño del sistema. Dicho esto, se establecen los siguientes requisitos de software:

1. **Página web sección Tutorial (urgente):** En la página web debe haber una sección de tutorial donde se indique con imágenes el paso a paso para completar el pipeline por parte del usuario, es decir, cómo debe grabar el video, cómo debe subir el video en la misma página, y cómo revisar en el museo el objeto resultante.
2. **Formulario subida de video (urgente):** Debe existir un formulario donde el usuario pueda rellenar con el nombre del objeto a subir, su descripción para ser mostrada en el museo, agregar dimensiones (largo, ancho, alto) de forma opcional, y poder subir un video. Realizar validaciones para el formulario.
3. **Página web sección Inicio (normal):** Sección de la página web donde se da la bienvenida al usuario y se le invita a visitar las otras secciones de la página.
4. **Página web sección About (normal):** Sección donde se dan detalles sobre el proyecto con el cual se asocia la página web.
5. **Almacenamiento del video y metadata (urgente):** Luego de subir el video exitosamente, se debe almacenar de forma local al proyecto, junto a la metadata correspondiente.
6. **Convertir video en carpeta con imágenes (urgente):** Se deberá utilizar un método para extraer frames por segundo del video almacenado y guardar las imágenes en una carpeta aparte.
7. **Remove fondo de las imágenes (urgente):** Debe existir un método para extraer el fondo de las imágenes creadas a partir del video.
8. **Herramienta de reconstrucción 3D (urgente):** Se debe investigar, seleccionar e implementar una herramienta de reconstrucción 3D de objetos a partir de imágenes, creando un método el cual permita automatizar el proceso de configuración y ejecución de la herramienta, y solo necesite ser llamado junto al nombre del objeto a reconstruir.
9. **Guardar archivos para exportar (urgente):** En el caso de obtener una reconstrucción 3D exitosa, se deben guardar en una carpeta aparte el objeto 3D texturizado (.obj) y un archivo con la metadata del mismo.
10. **Subida de archivos finales al servidor del museo (urgente):** Se deben subir los archivos necesarios al servidor donde se almacena el museo virtual, respetando la jerarquía de las carpetas correspondiente.

Con esto, se da paso a la solución realizada para lograr satisfacer cada uno de los requisitos planteados anteriormente, en la cual se dará mayor detalle de cada una de las etapas internas que componen la completitud del pipeline a diseñar.

4. Capítulo IV: Diseño

La solución construida contempla tres elementos principales: diseño del pipeline, implementación y solución de requisitos. Por un lado, el diseño de pipeline creado junto a todas sus etapas que componen la solución completa del proyecto. Por otro lado, la implementación la cual divide el proyecto en tres pilares fundamentales: página web, reconstrucción 3D y museo. Finalmente, la última sección contiene la solución de cada uno de los requisitos planteados en el capítulo anterior.

Dicho esto, en el presente capítulo se presentará el diseño de pipeline construido para la solución, explicando cada una de sus etapas que lo conforman, e indicando su aporte individual en la construcción del proyecto desarrollado. Luego, en el siguiente capítulo, se describirá la implementación del proyecto realizado, mostrando detalladamente cada pilar que lo compone, y se revisarán los requisitos establecidos en el capítulo 3, presentando su estado actual y detallando cuáles fueron implementados o proporcionando una explicación de por qué no se resolvieron.

4.1. Diseño del pipeline

Como dicho anteriormente, se realiza un boceto del pipeline diseñado, con sus respectivas etapas y mostrando su flujo completo, y luego se detalla una a una explicando sus características.



Figura 2: Diseño pipeline: flujo completo

Todo este diseño, se realiza con la finalidad de implementarse en un museo virtual 3D de Khachkars³ ya existente. Esta aplicación (incluido en el pipeline) implementada en Unity⁴,

³tipo de piedra conmemorativa del arte armenio

⁴motor gráfico multiplataforma para el desarrollo de videojuegos

actualmente ofrece a los usuarios la posibilidad de generar sus propias exhibiciones virtuales mediante varias funcionalidades. La básica y primera necesaria para construir una nueva exhibición es elegir uno entre 5 escenarios predefinidos que se muestran al ingresar a la aplicación. Estos escenarios representan lugares típicos donde se pueden encontrar estos objetos. En seguida, una vez inmerso en escenario, el usuario puede elegir los khachkars que habrá en la exhibición. Esto lo hace eligiendo las piedras de un menú gráfico de tipo "pull-down" que consta de íconos que representan a los modelos de khachkar que hay disponibles en el sistema. Otras funcionalidades son las de moverse a través de la sala, obtener información de los Khachkars presentes, y poder modificar la sala agregando más Khachkars, cambiándolos de lugar, eliminándolos de la exhibición, agregando más información acerca del khachkar, si cuenta con ella, entre otras. (Ver Figura 1)

El diseño del pipeline consiste en un total de 7 etapas, debidamente ordenadas para implementar la inclusión de un nuevo objeto en un museo 3D desde la subida del video hasta la aparición de este objeto en el menú de incorporación de nuevos objetos al ambiente elegido, donde cada una tiene sus propios procesos y características.

A continuación, se muestran los detalles de cada uno de estos procesos, y además se abordará la descripción del proceso de guardado de archivos y/o carpetas en el servidor donde se almacenará el proyecto.

En primer lugar, el proyecto se almacenará en un servidor destinado para este proyecto, el cual está alojado dentro de la red de computadores del Departamento de Ciencias de la Computación de la Universidad de Chile, dentro del cual se almacenan los paquetes con los objetos del museo de *Khachkars* explicado anteriormente. Luego, los archivos generados en el proceso implementado por el pipeline diseñado, se guardarán dentro de una carpeta llamada *src* ubicada dentro del directorio del proyecto. Dentro de ésta, estarán las carpetas principales *360videos* y *object*, en las cuales se almacenarán temporalmente los videos de los objetos que se suban y los archivos del objeto listo para su subida al museo, y una carpeta temporal con el nombre de un nuevo objeto a crear. Esta carpeta temporal estará estructurada según los requerimientos de la tecnología de reconstrucción a utilizar.

Cabe destacar, que el servidor en el cual se trabajará no posee GPU⁵, algo a considerar para la investigación próxima que se realizará.

4.1.1. Captura video del objeto

El inicio del proceso debe ser hecha por el usuario que desea que un objeto que capturó por medio de un video y que quiere obtener su representación 3D computacional siguiendo las indicaciones del tutorial.

El video se debe grabar con el objetivo de capturar todos los ángulos del objeto, ya sea con el objeto estático y el dispositivo rodeando al objeto, o bien manteniendo la cámara fija y rotando el objeto sobre su eje, obteniendo así una filmación en 360 grados del objeto deseado. Se debe intentar mantener una velocidad constante de rotación del objeto o traslación de la cámara para que haya una cantidad equilibrada de imágenes por ángulo del objeto ya que esto permite una mejor reconstrucción. Idealmente se debe evitar que hayan objetos o partes

⁵Unidad de Procesamiento Gráfico, también llamada tarjeta gráfica o de video

humanas (p. ej. manos) que obstruyan la captura de del objeto por la cámara durante la grabación. Además, la duración del video debe ser entre 10 y 30 segundos, para que sean mínimo 100 fotos, ya que en las pruebas que se ejecutaron durante el desarrollo del proyecto advertimos que tener menos de 80 imágenes es perjudicial para el proceso de reconstrucción 3D del objeto.

Finalmente, se hará un tutorial con todas estas indicaciones, el cual se podrá ver en la página web creada, para que los usuarios sepan cómo grabar correctamente y aprovechar la herramienta de la mejor forma.

4.1.2. Página web

Se creará una página web básica para la subida de archivos, con cuatro secciones principalmente:

- Inicio: bienvenida con imágenes del museo virtual y un mensaje que invita al usuario a navegar la página, indicando un breve resumen de qué es lo que encontrará en cada sección.
- Tutorial: se enseña cómo grabar el video, con imágenes y/o GIFs, y explicado a detalle.
- Subida: formulario para subir el video del objeto que se desea modelar (Nombre, Descripción, Subir Archivos, Opciones avanzadas).
- About: contextualización y explicación del proyecto por el cuál se creó la página web.

Para esta etapa, se debe lograr automatizar la validación del video subido, obteniendo su duración para chequear que esté dentro del rango permitido, y chequeando el formato para corroborar que corresponde a un video. Con esto, se debería entregar un mensaje en pantalla que diga si el archivo fue subido exitosamente, o si fue rechazado, indicándole el error al usuario para que intente el procedimiento nuevamente. Internamente, se debe subir el archivo al repositorio, y si se logra de forma satisfactoria, proceder con el flujo hacia la siguiente etapa.

4.1.3. Extracción de frames

Para esto, se evaluará la metodología para extraer frames de un video de 10 a 30 segundos, buscando una herramienta/software que sea posible de incluir dentro del proyecto, evitando usar un servicio externo (página o servicio web de terceros) del cual no podamos depender con plena seguridad de su disponibilidad a futuro. Cabe destacar, que el número máximo de fotos esperadas a obtener y por ende la duración máxima del video señalada, están sujetos a cambios, puesto que dependen de la calidad obtenida por la herramienta de reconstrucción, y también a las capacidades del hardware que manejará el proceso.

Una vez que el video subido al sistema por medio de la página web, se debe automatizar la extracción de frames y el guardado de estas imágenes en una subcarpeta estructurada correctamente según los requerimientos que tenga la herramienta de reconstrucción, la cual será la encargada de utilizar esta nueva carpeta con imágenes. Esto puede hacerse por medio de un archivo de comandos.

4.1.4. Reconstrucción modelo 3D

Para esta etapa, se debe evaluar el mejor software para la reconstrucción de modelos 3D, considerando:

- Menos ruido en el resultado inicial (directo del software), es decir, la menor presencia de pequeñas fluctuaciones o variaciones en los datos de la reconstrucción que no están relacionadas con la estructura real del objeto (fondo del video, mesa de apoyo, etc).
- Mayor flexibilidad del software, es decir, que sea fácil de implementar mediante línea de comandos, librerías, importable, etc.
- Mayor grado de automatización, lo cual abarca los puntos anteriores, ya que mientras menos ruido tenga el primer modelo resultante, y mayor sea la flexibilidad de la herramienta, será mucho más sencillo automatizar el proceso.

Este proceso es muy importante dentro del pipeline, ya que el tener un buen software de reconstrucción, el cual deje muy poco ruido en el objeto resultante, puede aliviar la carga de la siguiente etapa, la limpieza. Lo ideal sería que no se necesitara realizar el proceso de limpieza del modelo 3D y de esta forma lograr una total automatización del proceso sin ser necesario la intervención humana en el proceso. Lamentablemente, el estado del arte del proceso de reconstrucción 3D de un objeto a partir de un video está aún bastante lejos de lograr esto y la mayoría de las veces se requiere que un ser humano elimine las partes que aparecen en el modelo reconstruido pero que no pertenecen en realidad al objeto real. Por lo tanto, lo automatizable (a priori) de esta etapa es la ejecución del proceso de reconstrucción del objeto utilizando las imágenes guardadas en su carpeta correspondiente, y luego nuevamente el guardado del objeto resultante en otra subcarpeta de la carpeta principal.

4.1.5. Limpieza modelo 3D

Como se dijo anteriormente, este proceso es necesario pues en la reconstrucción de objetos los algoritmos actuales suelen introducir pedazos que no son realmente del objeto original y que son necesario retirarlos. Este proceso es el más complicado con respecto al intentar automatizarlo, puesto que la eliminación del ruido de un modelo 3D reconstruido es complicado de programar para obtener un modelo completamente limpio, sin eliminar partes del objeto original o dejar pedazos de ruido incorporados en el modelo. Por esto, inicialmente se considera que debe realizarse de forma manual, utilizando un editor de objetos 3D, para que se obtengan modelos fieles a la forma del objeto. Por esto mismo, se deben evaluar softwares para la limpieza de los modelos 3D, considerando:

- Facilidad de aprendizaje, para que lo pueda realizar una persona no necesariamente experta.
- Limpieza rápida, para optimizar el tiempo que tarde la limpieza del objeto, y seguir con el pipeline lo más rápido posible.

4.1.6. Subida a bundles

Un modelo 3D se representa usando más de un archivo que contienen distintos atributos del modelo (como por ejemplo los puntos y aristas de la malla que representa el volumen y la textura) y estos están definidos en archivos separados. Como se dijo anteriormente, el

conjunto de archivos que representa un modelo 3D generalmente se conoce como bundle (paquete). Al tener el modelo listo para su uso, se debe subir el bundle al servidor que los aloje y los pueda mostrar en un ambiente adecuado. Aquí se deben tomar decisiones acerca de cómo etiquetarlos para guardarlos de modo de que su recuperación por parte del usuario que desea verlos sea más efectiva (es decir, el usuario encuentra lo que busca) y eficiente (que su búsqueda sea más rápida dentro del museo).

La estructura y organización de los archivos en los bundles es clave para la optimización de búsqueda de los objetos a mediano/largo plazo, ya que se espera tener una gran cantidad de objetos que vaya aumentando gracias al *crowdsourcing*. De esto, se espera automatizar la organización mencionada gracias a los nombres de las carpetas de cada objeto.

4.1.7. Museo virtual

Finalmente, los objetos subidos en los bundles deberán ser importados al museo virtual para su visualización. Estos podrían ser buscados idealmente por categorías, para que a largo plazo sea más eficiente su búsqueda.

Esta etapa debería ser la menos tediosa, puesto que el museo actualmente ya está creado, y solo faltaría extenderle funcionalidades, agregándole un nuevo escenario donde el usuario pueda ser capaz de ver el objeto que subió reconstruido en su forma 3D, y colocarlo a su gusto dentro de la sala. Todo esto, dentro del mismo ambiente *Unity* ya creado.

5. Capítulo V: Implementación

El proyecto se dividió en tres pilares fundamentales: La página web, un pilar que tendrá interacción directa con el usuario, puesto que será el enlace entre este y el museo virtual; Reconstrucción 3D, en la cual se investigaron diferentes herramientas y se seleccionará la más indicada; Museo, el paso final del pipeline en el cual se almacenarán los objetos reconstruidos, y donde el usuario pueda ser capaz de ver el resultado de la reconstrucción. Dentro de cada pilar se desarrollan las distintas etapas del pipeline diseñado, y luego en su conjunto constituirán el flujo completo requerido.

En el siguiente diagrama se mostrarán de forma general cada uno de los servicios y funciones internas que hacen posible el funcionamiento del proyecto, y posteriormente se explicará la implementación en mayor detalle.

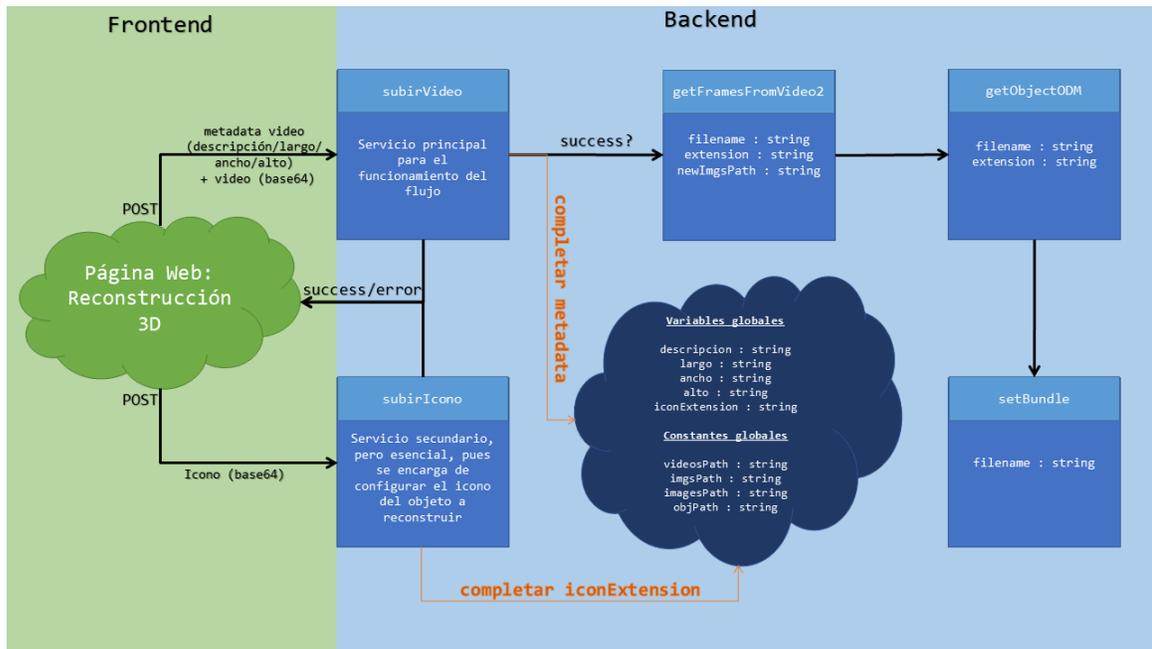


Figura 3: Diagrama de interacción

Como se aprecia, el frontend y backend se comunican principalmente mediante dos servicios, *subirVideo* y *subirIcono*, los cuales mediante una llamada POST envían al servidor el video y el icono en formato base64, y la metadata del objeto (descripción, alto, ancho, largo). Posteriormente, si estos servicios responden con éxito, se modifican las *variables globales* con estos nuevos datos, y se sigue con el pipeline, dándole paso en primer lugar a *getFramesFromVideo2*, donde se capturan imágenes del video y se guardan en una carpeta aparte, luego a *getObjectODM*, donde se llama a la tecnología de reconstrucción 3D OpenDroneMap (ODM) y se genera el objeto texturizado (con ruido), y finalmente *setBundle*, en el cual se prepara el bundle, carpeta en la cual se almacenarán los archivos necesarios para importar el objeto desde el museo virtual, como el archivo .obj, las imágenes de la textura, un archivo XML con su metadata, el icono para el catálogo, entre otros archivos de configuración (generados por ODM). Además, existen las *constantes globales*, las cuales corresponden a las rutas a las distintas carpetas del servidor donde se almacenarán los archivos generados durante la ejecución.

5.1. Página web

Para que el usuario pudiese interactuar de forma directa con el museo, pero sin tener que configurar la herramienta de reconstrucción, se desarrolló una página web que funcione como enlace entre el usuario y el museo. Esta, de diseño simple y directo, ya que se tiene como objetivo minimizar las complicaciones que pueda llegar a tener un usuario promedio. A continuación, se mostrarán las partes que componen la página y con las cuales interactúa el usuario, los detalles técnicos que maneja la misma de forma interna, y cómo ésta se comunica con el siguiente pilar que es la reconstrucción 3D del objeto.

5.1.1. Secciones

En primer lugar, se tiene la sección de *Inicio*, en la cual se le da la bienvenida al usuario a la página, y además se explica qué hace cada una de las otras secciones. Esto, con el fin de que el sujeto al llegar a la pantalla inicial, ya tenga una idea de dónde está y cómo proceder.



Figura 4: Página web, Inicio

Luego, se tiene la sección *Tutorial*, la segunda más importante, ya que ayuda al usuario a entender cómo funciona el proyecto, y cómo debe proceder para realizar la reconstrucción del objeto 3D de forma correcta. Esto, contextualizando al usuario sobre su aporte a realizar dentro del pipeline, y explicando mediante GIFs el paso a paso, que abarca desde la grabación del video hasta la utilización del objeto en el museo en caso de que el proceso de reconstrucción 3D haya sido exitoso.



Figura 5: Página web, Tutorial

Después, se tiene *Subida*, la sección más importante y que tiene directa conexión con el llamado a los métodos internos del proyecto, puesto que contiene el formulario en el cual el usuario deberá subir el video, su ícono para que pueda verse en el catálogo del museo, y colocar la metadata asociada al objeto el cual grabó.



Figura 6: Página web, Subida

Como se puede ver, la metadata asociada corresponde al nombre que el objeto tendrá dentro del museo, su descripción, y existe un checkbox⁶ el cual despliega una serie de casilleros para colocarle medidas al objeto a subir. Esto se implementó con el fin de ayudar a dimensionar y escalar el objeto de forma correcta dentro del museo, y que no tenga medidas desproporcionadas respecto a otros objetos que puedan haber en la escena.

Finalmente, se tiene la sección *About*, en la cual se hablará sobre el proyecto, y detalles como la motivación, finalidad y objetivos presentes en el trabajo de título que abarca el proyecto.



Figura 7: Página web, About

⁶Casilla de verificación que, en este caso, activa opciones avanzadas en el formulario

Cabe destacar que, mediante pequeños detalles estéticos y funcionales dentro de la página, ya sea botones de ayuda, textos explicativos, entre otros, se intenta obtener una interacción fluida por parte del usuario, en donde pueda lograr su objetivo de subir un video en poco tiempo.

5.1.2. Frameworks y Librerías

Para la implementación del proyecto se consideraron 2 opciones de frameworks para el backend y 2 para el frontend. Por un lado, para el backend se tomaron como opciones Django y Nuxtjs, y por otro lado, para el frontend se debatió entre usar BootstrapVue o React. Todos los frameworks ofrecen soluciones con funcionalidades similares, pero finalmente para el backend se optó por utilizar *Nuxtjs*, el framework de Vuejs, ya que permite hacer un uso más fácil de cualquier librería para interfaces de usuario, además de tener una estructura de archivos mucho más organizada a la hora de utilizar layouts y crear las distintas secciones de la página web. Sobre la posibilidad de trabajar con ReactJS para el lado del frontend, debido a la previa experiencia se puede sacar mejor provecho de ésta en el trabajo con aplicaciones móviles, y por lo tanto se eligió finalmente *BootstrapVue* para esta parte.

Luego, se decidió completar el esqueleto del proyecto con el entorno de Nodejs, para trabajar con lenguaje javascript en el backend del proyecto, y así poder utilizar la infraestructura Express, la cual es compatible con este entorno y permite crear una API con una comunicación mucho más rápida y sencilla gracias a su manejo de llamadas HTTP.

A continuación, se detallarán otras librerías y módulos utilizados dentro del proyecto, y se describirá su aporte dentro del mismo.

- **FileSystem**: es un módulo de node que permite trabajar con el sistema de archivos del computador, por lo que es posible leer, crear, actualizar, borrar o renombrar archivos.
- **Axios**: es un cliente basado en promesas que permite hacer llamadas HTTP, para comunicarse, por ejemplo, desde los métodos de la página web, hacia el backend de la aplicación.
- **ffmpeg**: es un módulo de node que provee funcionalidades para el manejo de audios y videos.
- **fluent-ffmpeg**: una librería que utiliza *ffmpeg* y lo simplifica para usar el módulo de nodejs. Este, para funcionar requiere que se encuentre instalado el módulo ffmpeg, y con este es posible realizar de forma más directa la captura de imágenes por segundo del video que se sube.
- **child_process**: es un módulo que permite ejecutar llamadas en la consola del computador, por lo que con esto se maneja el uso de líneas de comando para llamar, por ejemplo, a una herramienta de reconstrucción, o para eliminar archivos puntuales.
- **rembg**: es una herramienta para remover el fondo de las imágenes. Esta, se instala mediante pip, es decir, requiere python en el sistema para poder ser utilizado.
- **get-video-duration**: corresponde a un módulo que sirve para obtener la duración de un video en segundos, por lo que es muy útil para crear el arreglo de marcas de tiempo que se utiliza para capturar imágenes del video posteriormente.
- **on-finished**: es un módulo que ejecuta un callback cuando una solicitud HTTP cierra, finaliza o lanza error. Esto, servirá para saber cuándo termina el servicio llamado desde

la página web, y poder empezar a realizar el trabajo interno del proyecto luego de almacenar el video.

Como se puede ver, estos módulos y librerías trabajan en conjunto para poder manejar los archivos que se van almacenando en el servidor luego de subir un video desde la página web. Luego, el último módulo presentado *on-finished*, avisa a la API que regresó del servicio y que ya se subió el video exitosamente, por lo que de forma interna se termina el servicio que se encarga de la comunicación entre back y frontend, y avanza en el pipeline, siguiendo con el corte del video en imágenes, la reconstrucción 3D del objeto en base a tales imágenes, y finalmente la preparación del bundle para el museo. Todo esto de forma interna y sin llamar a servicios mediante solicitudes HTTP.

5.2. Reconstrucción 3D

Para este proceso, se investigaron diversas opciones de tecnologías para la reconstrucción 3D. Esto, mediante el uso de dos ejemplos de captura de videos, para los cuales se mostrarán y analizarán los resultados con cada tecnología investigada, concluyendo luego cuál sería la mejor opción dadas las condiciones actuales

5.2.1. Opciones de métodos de captura de video

Primero, se preparó el material con el cual se realizarían las pruebas, utilizándose dos videos: el primero, donde un objeto gira sobre su propio eje, ya que está sobre una plataforma giratoria; y el segundo, donde el dispositivo rodea al objeto, es decir, el usuario se mueve mientras graba al objeto estático.



(a) Video 1, Cámara estática



(b) Video 2, Objeto estático

Figura 8: Videos de prueba, Khachkars

5.2.2. Preprocesamiento 1: Extracción de frames

Luego, se utilizó *fluent-ffmpeg*⁷, una librería que utiliza *ffmpeg* y lo simplifica para usar el módulo de nodejs. Esta herramienta, permite ingresarle la ruta al video el cual se quiere procesar, y luego toma screenshots de éste según un arreglo de marcas de tiempo que utiliza como parámetro. En este caso, se crea un arreglo con timemarks con una frecuencia de 10 FPS⁸. Luego, estas capturas se almacenan en una carpeta con el nombre del objeto. Con este preprocesamiento, se da paso a testear cada conjunto de imágenes.

⁷<https://github.com/fluent-ffmpeg/node-fluent-ffmpeg>

⁸Frames Per Second o imágenes por segundo.

5.2.3. Meshroom (Preprocesamiento 1)

En primer lugar, se hicieron pruebas con el software *Meshroom*⁹, programa con interfaz sencilla de utilizar, y que permite generar modelos 3D en base al input de imágenes que se le entregue. Posee una sección donde se colocan las fotografías, otra donde sale una visualización de una imagen seleccionada, el visor del modelo 3D, un grafo donde se muestra cada nodo por donde pasa el proceso de reconstrucción, junto a las configuraciones de cada uno de ellos a su derecha, entre otras opciones.

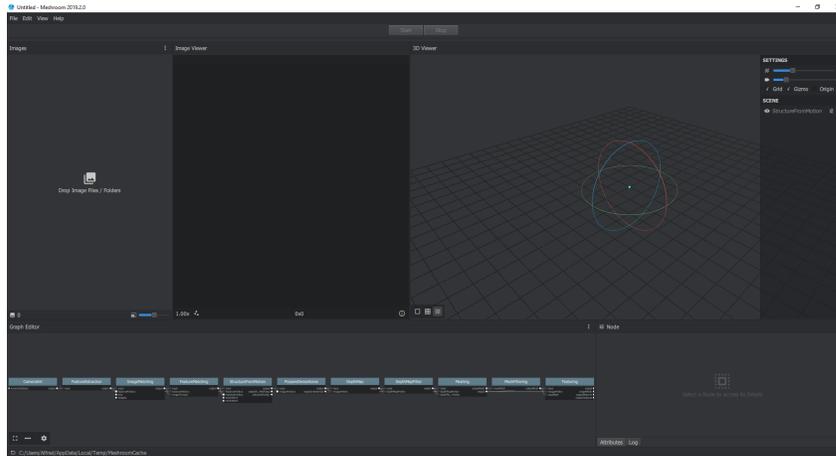


Figura 9: Meshroom, vista principal

Como resultado para el primer video, se obtuvo una gran cantidad de ruido en el modelado, el cual puede ser debido al tipo de grabación, ya que al mantenerse la cámara estática, el fondo se mantiene, por lo que el software reconoce este patrón como algo que debe reconstruir, generando esta confusión y, además, quitándole la parte trasera del khachkar, posiblemente por insuficiencia de imágenes de ese lado del objeto.

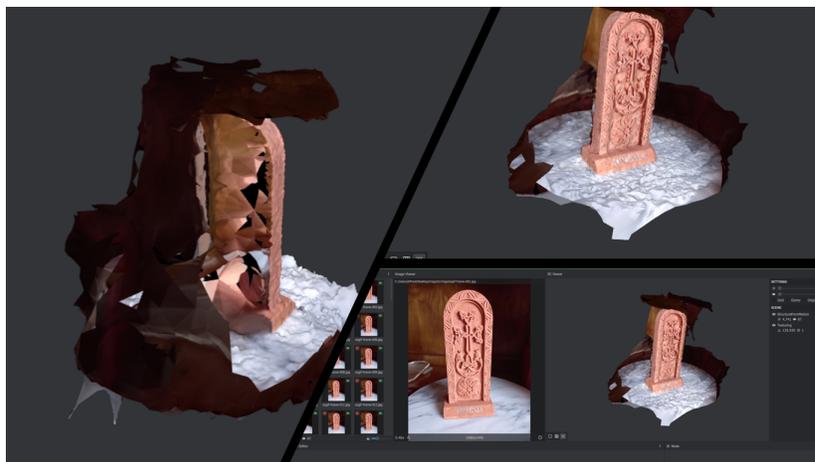


Figura 10: Meshroom, resultado video 1

Luego, para el segundo video, se obtuvo aún mucho más ruido que el primer video. Esto, puede ser visto como el otro extremo de tipo de grabación, ya que al variar tanto el fondo, el

⁹<https://meshroom-manual.readthedocs.io/en/latest/>

software supuso que tendría que reconstruir todo el ambiente que se mostraba en las imágenes, como se ve a continuación.

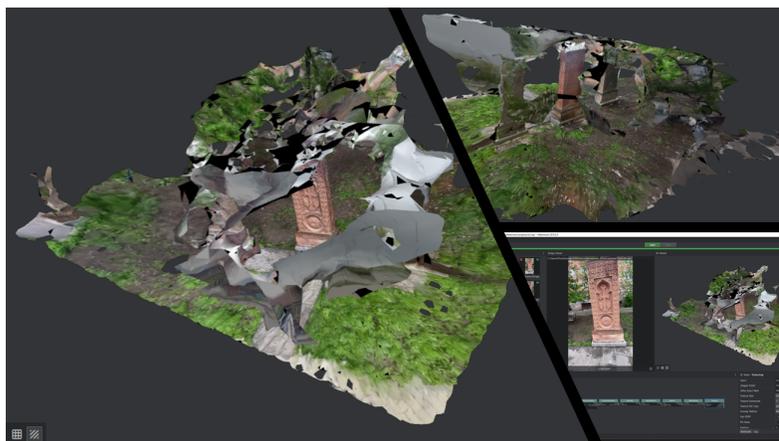


Figura 11: Meshroom, resultado video 2

5.2.4. OpenDroneMap (Preprocesamiento 1)

En segundo lugar, se utiliza el software de fotogrametría OpenDroneMap (ODM)¹⁰ el cual, en su versión Open Source, es posible desplegar un docker con esta herramienta. Para lograr ejecutar el proceso de reconstrucción, se deben ingresar las imágenes en una carpeta organizada de forma específica, entregando así diferentes tipos de objetos distribuidos en carpetas, como: malla 3D, objeto texturizado, entre otros. Toda estas instrucciones están muy explícitas en la documentación de ODM en su repositorio en GitHub, por lo que es fácil de seguir el paso a paso.

Respecto al primer video, se obtuvo un modelo con mucho menos ruido que con *Meshroom*, pero nuevamente no se logró generar la parte trasera del khachkar, reafirmando el punto de que faltaron imágenes en esos ángulos del objeto.

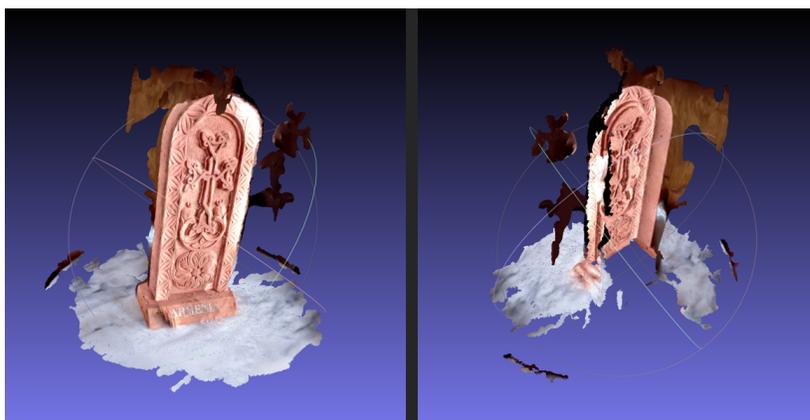


Figura 12: ODM, resultado video 1

¹⁰<https://github.com/OpenDroneMap/ODM>

Mientras que, para el segundo video, se obtuvo un resultado con un poco más de ruido que el primer video, pero mucho menos ruido que la reconstrucción con el software *Meshroom*, por lo que tomó ventaja la grabación con el objeto estático y el dispositivo rodeándolo.

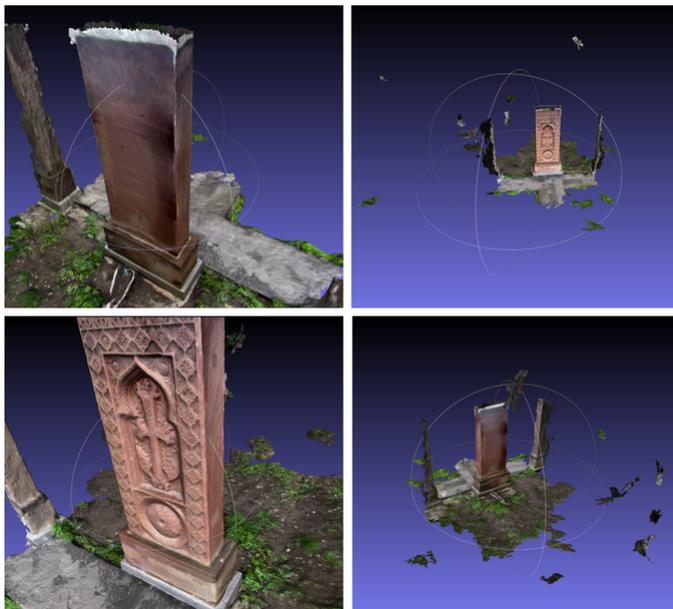
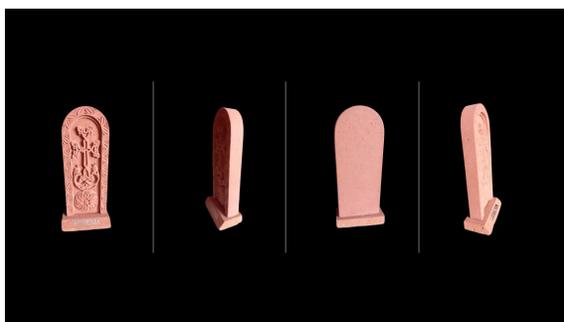


Figura 13: ODM, resultado video 2

5.2.5. Preprocesamiento 2: Rembg

Luego de estas pruebas, con el objetivo de reducir el ruido lo mayormente posible, se decidió añadir un preprocesamiento adicional, el cual consiste en eliminar el fondo de las imágenes antes de procesarlas para su reconstrucción. Esto, se realizó con *Rembg*¹¹, una herramienta implementada para su uso con python y creada exclusivamente para remover el fondo de las imágenes con un objeto centrado en cámara.



(a) Video 1 sin fondo, Cámara estática



(b) Video 2 sin fondo, Objeto estático

Figura 14: Videos de prueba sin fondo, Khachkars

Es posible notar que con ambos videos se logra una buena extracción de fondo, por lo que quizás se necesite modificar la “regla de grabación”, no determinando quién es el que se debe mover (dispositivo u objeto), sino que dejar como requisito el mantener el objeto deseado lo más centrado posible durante la grabación del video.

¹¹<https://github.com/danielgatis/rembg>

5.2.6. Meshroom y ODM (Preprocesamiento 2)

En el caso de *Meshroom*, para ambos videos se obtuvo una reconstrucción con una menor cantidad de ruido, pero aún así no logró detectar para ningún caso la parte trasera del khachkar, dejando incompleto el modelo.

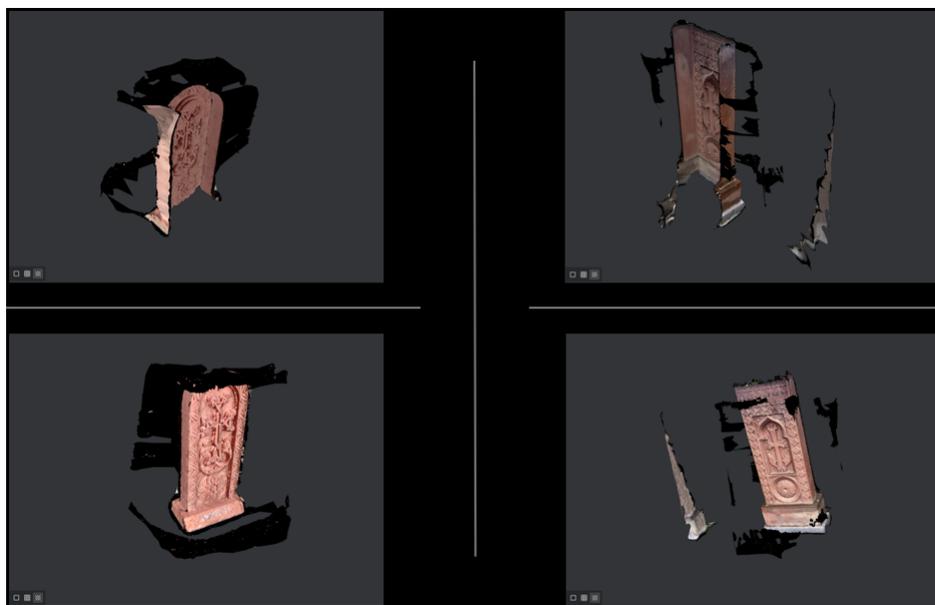


Figura 15: Meshroom, resultado video 1 (izq) y video 2 (der) sin fondo

Por otro lado, para el caso de ODM, no se obtuvo resultado para ninguno de los dos videos, pues en la carpeta donde debería estar el objeto texturizado, solo crea archivos temporales y se detiene la ejecución. Esto, podría ser debido a problemas de compatibilidad con imágenes sin fondo, confundiendo parámetros y/o no encontrando suficientes texturas como para entregar resultados correctos.

5.2.7. Instant Neural Graphics Primitives

En tercer y último lugar, se realizaron pruebas con *Instant NGP*, herramienta recomendada por el tesista Cristián Llul, con el cual se está colaborando, y quien ha utilizado ya ésta implementación. Cristián indica que NeRF¹², a grandes rasgos, funciona mediante el entrenamiento sobreajustado de una red neuronal. La información se almacena respectiva a colores y densidad de cada voxel¹³ del volumen, en la red, según el ángulo de visión del observador. Luego, *Instant NGP* lo que hace es optimizar el proceso de sobreajuste, acelerando el proceso de entrenamiento, tardando así mucho menos en procesar una reconstrucción 3D. Ésta tecnología requiere del uso de GPU.

¹²<https://www.matthewtancik.com/nerf>

¹³píxel en 3D

Los resultados vistos sobre las imágenes sin fondo utilizando esta herramienta fueron destacados, puesto que se generó un modelo 3D completo con muy poco ruido. Por lo tanto, se seguirá colaborando con Cristian para sacarle el mayor provecho posible a esta herramienta, y así lograr obtener unos modelos 3D optimizados, con el objetivo de intentar evitar la etapa de “limpieza modelo 3D” con intervención humana.

5.2.8. Conclusiones acerca de las posibilidades de reconstrucción

Con todas estas pruebas, es posible concluir que el camino a seguir probablemente sería el de la última tecnología probada, es decir, *Instant NGP*, herramienta de reconstrucción 3D mediante Deep Learning que optimiza a *NeRF*. Además, todo esto con el “preprocesamiento 2” planteado durante la investigación, el cual consiste en utilizar *RemBG* para quitarle el fondo al conjunto de imágenes asociadas al video que suba el usuario, y luego comenzar el procedimiento de reconstrucción, el cual lograría minimizar el proceso de limpieza del objeto resultante.

Luego, en el caso de inclinarse por el uso de fotogrametría, se debe reconsiderar el uso del “preprocesamiento 2”, puesto que puede haber problemas a la hora de buscar el ángulo de la cámara por parte de las herramientas de reconstrucción, ya que al trabajar con imágenes sin fondo, pueden perder la noción de espacio y presentar problemas. Sin embargo, para este caso se debe exigir al usuario mantener centrado el objeto para la grabación del video, además de que se tenga que perder grado de automatización, puesto que se necesitaría de una limpieza manual del objeto posterior a su reconstrucción 3D.

Finalmente, y considerando las condiciones actuales, es decir, el tener un servidor sin GPU, se optará por la herramienta de reconstrucción *OpenDroneMap (ODM)*, que hace uso de fotogrametría y puede trabajar tanto con GPU como con solo CPU, siendo ideal para este caso. En consecuencia, se deberán estructurar las carpetas según los requerimientos de ODM para su correcta utilización en el proyecto, y además se deberá considerar un trabajo manual de limpieza previo a dejar preparado el bundle del objeto para su posterior importación al museo.

5.3. Museo

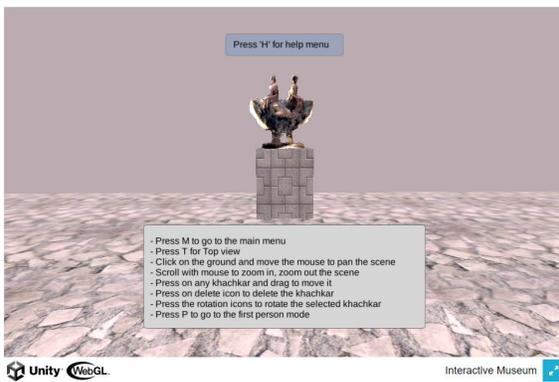
El tercer y último pilar corresponde al museo virtual 3D, el cual ya está creado, pero que actualmente trabaja solamente con khachkars. Este, ampliará su catálogo gracias a la subida de los objetos al servidor donde se encuentra almacenado, puesto que al llegar los archivos al servidor en unas carpetas específicas que ya se detallarán próximamente, estos se reorganizarán de forma interna mediante scripts o de forma manual para poder ser visualizados en el museo. Este procedimiento interno será gestionado por Gabriel Azócar, en el cual se usa la herramienta de Unity conocida como *batch mode*¹⁴.

¹⁴forma de ejecutar comandos o scripts de manera automatizada en Unity sin necesidad de tener abierto el entorno de desarrollo

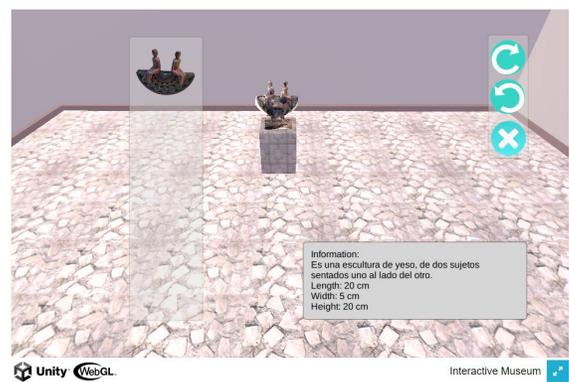
Previo a explicar el procedimiento, debemos conocer el concepto de *AssetBundle*¹⁵, los cuales son paquetes que contienen activos¹⁶ de Unity que se pueden cargar en tiempo de ejecución, tales como el modelo junto a sus texturas, los archivos de configuración, entre otros. Esto significa que se pueden crear paquetes de activos más pequeños y separarlos de la aplicación principal, lo que puede reducir el tamaño del archivo de instalación y permitir que los usuarios descarguen recursos adicionales a medida que se utiliza la aplicación.

Con la información en mente, se tiene un Script que corre una función llamada *BuildObjectAssetBundle*, creada específicamente para el pipeline. Esta función crea un *AssetBundle* vacío, toma un objeto desde una carpeta determinada y lo agrega al *AssetBundle*. Finalmente se guarda ese *AssetBundle* como un nuevo archivo. Con esta transformación podemos después en Unity, en tiempo de ejecución, instanciar el objeto guardado en ese *AssetBundle*. Esto es muy útil si pensamos en que llegarán objetos nuevos al servidor y necesitamos hacerlos “legibles” por Unity.

Finalmente, el usuario podrá interactuar con el museo, pudiendo ver el objeto que subió en su forma 3D dentro del catálogo, colocarlo a su gusto en un nuevo escenario prediseñado, es decir, siendo capaz de rotar y mover el objeto, y además ver la información del objeto que fue definida por el mismo usuario mediante la descripción del formulario de subida de la página web.



(a) Showroom: primera persona



(b) Showroom: tercera persona

Figura 16: Showroom: Nuevo escenario prediseñado para visualizar los nuevos objetos.

5.4. Solución de requisitos

A continuación, se presentará cada requisito de software planteado previamente en el capítulo 3, y luego se mostrará con la ayuda de imágenes cómo se implementó cada funcionalidad requerida.

- **Página web sección Tutorial (urgente):** En la página web existe una sección de tutorial donde se indica mediante GIFs el paso a paso para completar el pipeline por parte del usuario, es decir, cómo debe grabar el video, cómo debe subir el video en la misma página, y cómo revisar en el museo el objeto resultante.

¹⁵<https://docs.unity3d.com/Manual/AssetBundlesIntro.html>

¹⁶Elementos fundamentales que se utilizan para construir juegos en Unity y se pueden importar desde una variedad de formatos de archivo.



(a) Tutorial, Cómo grabar el video (b) Tutorial, Metadata del objeto en museo (c) Tutorial, Subida de video y link a sección subida

Figura 17: Tutorial, Detalles de la sección

- **Formulario subida de video (urgente):** Existe una sección de subida con un formulario donde el usuario puede rellenar con el nombre del objeto a subir, su descripción para ser mostrada en el museo, agregar dimensiones (largo, ancho, alto) de forma opcional, y poder subir tanto el ícono del objeto que aparecerá en el catálogo del museo, como el video. Se realizan validaciones para el formulario previo a subir los archivos y datos.

(a) Opciones avanzadas desactivadas

(b) Opciones avanzadas activadas

Figura 18: Subida: formulario

- **Página web sección Inicio (normal):** Existe una sección de la página web donde se da la bienvenida al usuario y se le invita a visitar las otras secciones de la página. Además, se agregan links dentro del párrafo explicativo hacia el museo virtual (Ver Figura 4).
- **Página web sección About (normal):** Existe una sección donde se dan detalles y se contextualiza al usuario sobre el proyecto con el cual se asocia la página web. Al igual que en la sección de *Inicio*, se añaden links al museo virtual. Además, se muestra una imagen del museo virtual existente de khachkars, el cual se busca extender a largo plazo (Ver Figura 7).

- **Almacenamiento del video y metadata (urgente):** Luego de subir el video exitosamente, se almacena de forma local al proyecto, junto a la metadata correspondiente en variables dentro del código.

```

  src
  360videos
  Minion_LoL.mp4
  10 let descripcion = "";
  11 let largo = "";
  12 let ancho = "";
  13 let alto = "";

```

Figura 19: Almacenamiento video y metadata

- **Convertir video en carpeta con imágenes (urgente):** Se utilizar un método para extraer frames por segundo del video almacenado y guardar las imágenes en una carpeta aparte. Estas, se guardan siguiendo la estructura que requiere OpenDroneMap.

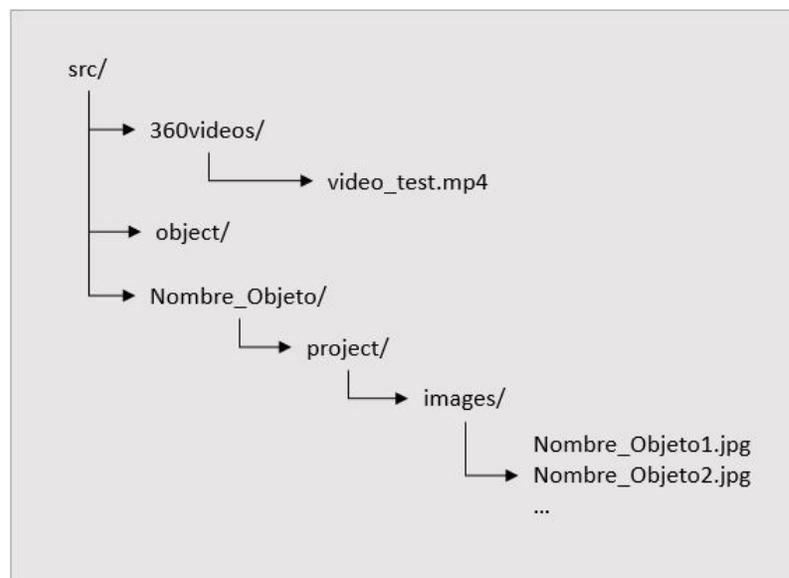


Figura 20: Carpeta con imágenes: ejemplo video_test.mp4 y Nombre_Objeto

- **Remover fondo de las imágenes (urgente):** Existe un método para extraer el fondo de las imágenes creadas a partir del video. Sin embargo, se deja comentado y sin uso, puesto que actualmente se trabaja en un servidor sin GPU, no siendo compatible utilizar fotogrametría con imágenes sin fondo.
- **Herramienta de reconstrucción 3D (urgente):** Se investiga, selecciona e implementa la herramienta de reconstrucción 3D de objetos *OpenDroneMap* que hace uso de fotogrametría. Esto, montando la herramienta en el servidor mediante docker, y luego creando un método el cual permita automatizar el proceso de configuración y ejecución de la herramienta, y solo necesite ser llamado junto al nombre del objeto a reconstruir. Se decidió utilizar esta herramienta debido a su capacidad de funcionar solo con CPU, y además porque resulta ser fácil de llamar por línea de comandos, por lo que lograba satisfacer la mayor cantidad de requisitos dadas las condiciones.

- **Guardar archivos para exportar (urgente):** En el caso de obtener una reconstrucción 3D exitosa, los archivos resultantes de este proceso se guardan dentro de carpetas que forman parte de una estructura generada por ODM.

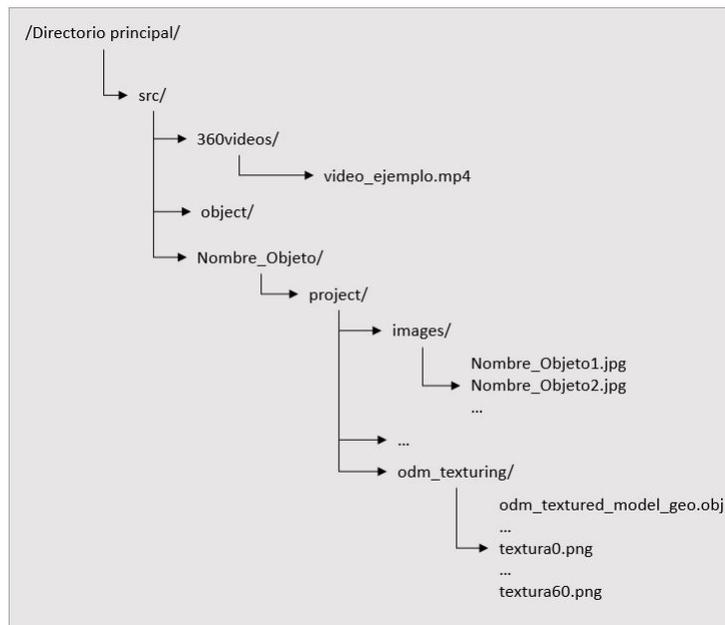


Figura 21: Estructura carpetas post reconstrucción de OpenDroneMap

Además, se crea un archivo XML que contiene la información del objeto (nombre, descripción y medidas si es el caso).

De todas estas carpetas, hay una en particular llamada *odm_texturing*, cuyo contenido debe ser desplazado de forma automática hacia una subcarpeta de la carpeta *object*, o de forma manual en caso de fallar el proceso de copia de archivos. En esta subcarpeta de nombre igual al nombre del objeto (bundle), se encuentran los archivos del objeto 3D, el archivo XML, las imágenes que componen su textura, y otros archivos de configuración.

Dado que ODM genera ruido en el modelo final 3D, se necesita de un proceso de limpieza posterior a organizar la carpeta final, pero previo a subir el modelo al museo, puesto que en caso contrario se vería contenido no deseado dentro de la exposición. Esto, es posible realizarse en poco tiempo mediante *MeshLab*¹⁷, software el cual fue elegido por su fácil instalación, sencillez de uso, y su alta compatibilidad con distintos sistemas operativos. Para realizar la limpieza, en primer lugar se debe abrir Meshlab, importar el objeto texturizado mediante *Import Mesh*, y luego ir seleccionando las partes del objeto que sean ruido con *Select Vertices* e ir eliminándolo con *Delete Selected Vertices*, como se puede ver en las siguientes imágenes.

¹⁷<https://www.meshlab.net/#download>

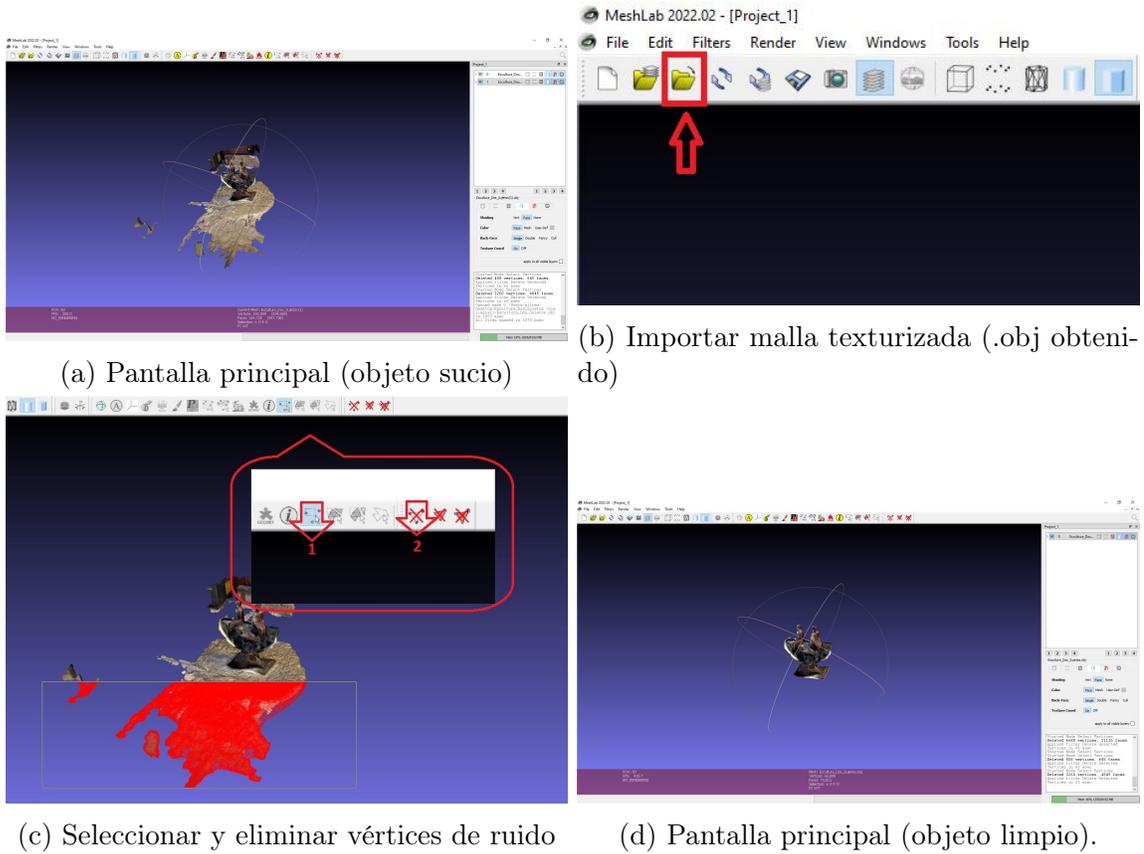


Figura 22: Limpieza del objeto. MeshLab

Finalmente, la estructura de las carpetas dentro del directorio principal del proyecto, luego de la reconstrucción 3D y posterior a la limpieza, quedará de la siguiente forma para seguir con la etapa de importación del bundle al museo.

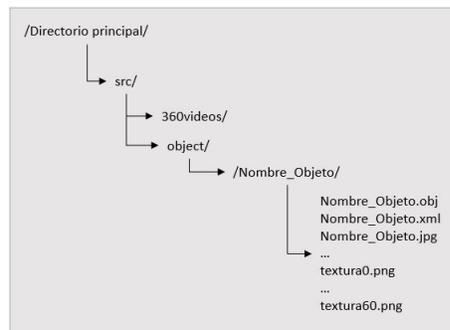


Figura 23: Estructura carpetas previo a importar bundle en museo.

- **Subida de archivos finales al servidor del museo (urgente):** Se suben los archivos necesarios al servidor donde se almacena el museo virtual, respetando la jerarquía de las carpetas correspondiente. Esto, mediante el procedimiento existente explicado anteriormente, en el cual se toma como input el bundle creado previamente, y generando así un *AssetBundle* de Unity.

El procedimiento existente se debe llevar a cabo puesto que desde Unity no se puede instanciar directamente los objetos de forma sencilla, salvo que se haga desde su versión de Escritorio, pero como no es el caso en este proyecto, se genera por cada bundle creado (explicado en requisito “*Guardar archivos para exportar*”), un nuevo *AssetBundle*, el cual se usa a través de descargas on demand, desde el museo, los cuales se instancian finalmente al momento de querer colocar el objeto dentro del museo, pudiéndose finalmente ver el objeto reconstruido en el nuevo escenario prediseñado (Ver Figura 16).

6. Capítulo VI: Validación

La evaluación del trabajo realizado se centró específicamente en la página web, cuya función es servir de enlace entre el usuario y el resto del pipeline, para lograr el objetivo final el cual es el museo virtual. Para esto, se juntó a un total de 15 personas, de distintos perfiles de usuario, los cuales fueron seleccionadas mediante un formulario de inscripción publicado públicamente, tanto en Redes Sociales como en los Foros de la Universidad de Chile, en el cual se solicitaban voluntarios(as) para evaluar una página web. En este formulario, se pidió de forma específica que detallaran su Profesión/Ocupación, teniendo como resultados los siguientes perfiles:

- Diseñador Gráfico
- Enfermería
- Estudiante Universitario (Licenciatura Física, Ingeniería Eléctrica, Ingeniería en Minas)
- Estudiante de Magister (Computación)
- Trabajador
- Ingeniero Civil Industrial
- Estudiante Animación Digital / Editor de Videos

6.1. Metodologías de evaluación

Luego de seleccionar a los distintos perfiles, se investigaron distintas metodologías para evaluar el trabajo realizado, llegando a la decisión de realizar una evaluación rápida, puesto que para realizar una evaluación rigurosa se necesita de un grupo mucho más grande de personas que evalúen el trabajo. Además, se optó por hacer una evaluación formativa, es decir, durante el proyecto y con la finalidad de mejorar el conocimiento acerca del diseño de la página, comprendiendo lo que funciona y lo que no, y así poder actualizar la página para lograr una versión mucho más óptima para el usuario. Con respecto a los parámetros de medición, la evaluación se enfocó en datos tanto cuantitativos como cualitativos, puesto que se busca que el usuario no tarde mucho tiempo en interactuar con la página, y a la vez evaluar el comportamiento y la percepción del usuario, como sus dificultades al navegar, la comprensión de los conceptos, el flujo de sus interacciones, entre otros datos.

Con todo esto en mente, se les contactó a cada uno de los voluntarios para agendar una reunión, la cual iba a ser grabada mientras ellos compartían pantalla, y al estar en ella se les daba un contexto, el cual era:

“Necesitas hacer una exposición de arte por trabajo, o quieres hacerla por hobby, dependiendo tu situación. Empiezas a buscar opciones y encuentras una página donde puedes recrear objetos mediante la grabación de un video y finalmente ver su reconstrucción 3D en un museo virtual, colocarlos en un escenario y poder montar tu exposición.”

Luego, se les indicaba el link para que entraran a la página web, y se les pedía que navegaran por ella hasta lograr su objetivo, el cual era subir un video según los requerimientos de la página. Para este punto, cabe recalcar que se le dio la opción al usuario de simular subir el video en caso de no tener las condiciones adecuadas, pero que hiciera todo el procedimiento tal como lo haría si hubiese grabado el video. Finalmente, antes de que el usuario empezase

a interactuar con la página, se les solicitó: Primero, pensar en voz alta, con el objetivo de evaluar de mejor manera el comportamiento y las reacciones del usuario a la página de todo tipo; Segundo, que al terminar de subir el video fuesen a la sección *Inicio* de la página e ingresaran a un link de una encuesta la cual debían contestar al momento.

6.1.1. Parámetros a evaluar

Para evaluar las entrevistas se tomaron en cuenta los siguientes parámetros:

- Tiempo total de interacción.
- Tiempo de interacción por sección (Inicio, Tutorial, Subida, About).
- Dificultad para navegar entre secciones.
- Comentarios y reacciones (pensamiento en voz alta) del usuario respecto a las funcionalidades (botones, textos y GIFs explicativos, etc).
- Comentarios y reacciones (pensamiento en voz alta) del usuario respecto al diseño de la página (paleta de colores, estructura página).

Luego, para añadirle un poco de rigurosidad a la evaluación, se añadió la encuesta final mencionada, donde:

- Se le pedía al usuario en primer lugar escribir cómo se graba el video, para así confirmar la eficacia del tutorial.
- Se le hizo una encuesta en *escala de Likert*¹⁸ donde debieron responder aseveraciones con las opciones *En desacuerdo, De acuerdo, Totalmente de acuerdo, etc.*
- Se les dio la opción de escribir feedback extra para la página web.

6.1.2. Resultados

Al analizar las entrevistas grabadas, se obtuvo que el tiempo total de interacción tuvo un promedio de 6 minutos, por lo que si se le añade el caso de tener que grabar el video (5 minutos aproximadamente), no superarían los 20 minutos por usuario en una interacción completa, es decir, visitando cada una de las secciones de la página.

Perfil	Edad	Sexo	Inicio [s]	Tutorial [s]	Subida [s]	About [s]	Total [min]
Diseñador Gráfico	25	Masculino	158	94	129	22	6,7
Estudiante Universitario	25	Femenino	20	90	245	35	6,5
Estudiante Magister Computación	25	Masculino	21	77	109	25	3,9
Ingeniero Civil Industrial	25	Masculino	14	90	120	30	4,2
Estudiante Ingeniería en Minas	25	Masculino	27	116	135	5	4,7
Estudiante Animación Digital	25	Masculino	29	105	210	33	6,3
Estudiante	26	Masculino	50	150	440	60	11,7
Estudiante Ingeniería Civil Eléctrica	22	Femenino	41	86	70	32	3,8
Trabajadora	38	Femenino	55	240	363	90	12,5
Estudiante	21	Masculino	24	60	235	30	5,8
Ingeniero Civil Eléctrico	33	Masculino	16	101	89	24	3,8
Estudiante Enfermería	25	Femenino	20	98	160	20	5,0
Estudiante Psicología	19	Femenino	70	143	157	21	6,5
Estudiante Ingeniería Industrial	32	Masculino	40	95	121	33	4,8
Estudiante Ingeniería Eléctrica	25	Masculino	28	120	136	10	4,9
Promedio Tiempo de Interacción			41	111	181	31	6

Figura 24: Tiempos de interacción con página. Tiempos totales por perfil, y promedios por sección.

¹⁸método de medición que tiene el objetivo de evaluar la opinión y actitudes de las personas.

Con respecto al tiempo de interacción por sección, se obtuvo un mayor tiempo de interacción en la sección de *Subida*, ya que es donde se agrega la información del video y la metadata del objeto que se subirá al museo, y por ende es una sección donde se debe colocar todo detalladamente y respetando todos los requisitos del formulario. Luego, la sección que lo sigue es el *Tutorial*, puesto que es donde el usuario debe leer y entender a la perfección el paso a paso a seguir para subir el video en la plataforma. Finalmente, las secciones menos concurridas son las de *Inicio*, y luego la de *About*, ya que son las secciones donde solamente se entrega información para que el usuario pueda saber cómo proceder en la página, en el caso de *Inicio*, o simplemente contextualizar al usuario sobre el proyecto, en el caso de *About*

En temas de dificultades a la hora de navegar entre secciones, por lo general no hubo complicaciones para pasar de una sección a otra. Sin embargo, hubo casos particulares donde el usuario estaba en el tutorial y al terminar de leerlo se pausaba su interacción por un leve tiempo, hasta que comprendió que debía subir hasta el tope superior de la página para hacer clic en la sección de Subida.

Lo anterior va de la mano con los comentarios de los usuarios sobre las funcionalidades de la página que, por lo general, fueron muy positivos. Por ejemplo, sobre los GIFs y textos explicativos que fueron encontrados muy útiles y se agradeció que se mostrase un caso real de reconstrucción, pero que llegaron a una crítica común de necesitar un enlace directo desde el fin del tutorial hasta la sección Subida, también un botón de vuelta al tutorial en caso de tener una duda al estar en la sección Subida, y finalmente un indicador de que el objeto se está subiendo y qué es lo que procede luego de eso, es decir, no sabían si debían esperar, si podían salirse de la página, o debían ir a otra sección para comprobar si el objeto estaba. Además, con respecto a los comentarios sobre la estética de la página, los usuarios fueron críticos respecto a la paleta de colores utilizada, pero se obtuvieron buenos comentarios sobre la estructura de las secciones y la sencillez de la página web.

Luego, pasando al formulario realizado, se obtuvieron unos excelentes resultados en la pregunta de escribir cómo se debe grabar el video, siendo todas las respuestas acertadas e indicándose la duración recomendada del video, y enfatizándose muchas veces que el objeto debe quedar centrado, lo cual no se explica en el texto del paso a paso, pero si se puede notar en el GIF.

En lo que respecta a la encuesta en escala de Likert con notas en un rango de -2 a 2, donde -2 es Totalmente en desacuerdo, -1 En desacuerdo, 0 Ni de acuerdo ni en desacuerdo, 1 De acuerdo y 2 Totalmente de acuerdo, se obtuvieron los siguientes resultados.

RECUESTO RESPUESTAS DE ENCUESTA POR SECCIÓN Y TOTAL								
Nivel acuerdo	Inicio	Tutorial	Subida	About	Estética	Total	Total %	
Totalmente en desacuerdo (-2)	0	0	0	0	0	0	0%	
En desacuerdo (-1)	0	0	0	1	2	3	6%	
Ni de acuerdo ni en desacuerdo (0)	0	1	0	0	4	5	10%	
De acuerdo (1)	4	8	5	1	4	22	44%	
Totalmente de acuerdo (2)	6	1	5	8	0	20	40%	
Puntaje Total	16	10	15	16	2	59		

Figura 25: Resultados encuesta. Escala de Likert de -100 a 100

En primer lugar, se logra ver una aceptación por parte de los usuarios a las secciones de la página web, sobre todo en Inicio, Subida y About, donde hubo la mayor cantidad de

voto máximo. Sin embargo, la sección Tutorial tuvo una mayoría de votos en *De acuerdo*. En segundo lugar, hubo más división en la aseveración que abordaba la estética, donde se presentó indecisión y un par de votos negativos. Ambos resultados coinciden con los comentarios en voz alta de los encuestados durante la entrevista, ya que efectivamente sugerían hacer cambios en la paleta de colores y que faltaba más información en el tutorial sobre cómo ver el objeto creado y ayudas para avanzar hacia el formulario de subida.

En consecuencia, evaluando los resultados mediante la escala de Likert, se puede ver en la imagen que se obtuvo un resultado de **57 puntos**, donde el puntaje total se encuentra en un rango que varía de -100 a 100 puntos.

6.2. Conclusiones e Implementación Post-tests

Analizando los resultados se puede concluir que: por un lado, se creó una página web exitosa funcionalmente hablando, la cual permite que se complete el objetivo de subir un video de un objeto para verlo luego en un showroom del museo virtual, pero que igualmente está sujeta a mejoras; por otro lado, se implementó una página aceptable en temas estéticos, donde tiene un amplio margen de mejora la paleta de colores utilizada. En lo que respecta el puntaje obtenido de la escala de Likert, se obtuvo un puntaje total muy positivo, el cual refleja la conformidad de las personas que evaluaron la página.

Con estos resultados y conclusiones, se implementaron algunos detalles en base al feedback recibido durante la evaluación. Estos son:

- Se añadió un link directo al museo en la barra de navegación de la página.
- (Tutorial) Se menciona con más exactitud el rango de duración recomendado para el video.
- (Tutorial) Se añade imagen del museo con el objeto del tutorial reconstruido.
- (Tutorial) Se añade enlace directo a la sección Subida al final de la explicación.
- (Subida) Se agregó un texto al principio de la sección donde se insiste al usuario de pasar por el tutorial en caso de no haberlo visto antes.
- (Subida) Se agrega un icono de ayuda por si el usuario quiere volver al Tutorial.
- (Subida) Se agrega el texto “Cargando...” luego de presionar el botón “Subir Video”.
- (Subida) Luego de subirse exitosamente el video, se agregó un redireccionamiento a la página de Inicio.
- (About) Se muestra una imagen del museo de khachkars existente para dar mayor contexto al usuario del origen y motivación del proyecto.

Todos estos cambios tienen la finalidad de satisfacer gran parte de los requerimientos mencionados por los usuarios al momento de interactuar con la página web durante las entrevistas.

7. Capítulo VII: Conclusiones y Trabajo futuro

Recapitulando todo el trabajo realizado, es posible decir que se abordó la problemática principal planteada, la cual consistía en que las personas sean capaces de ver un objeto de su interés en un lugar cualquiera, grabarlo, y luego poder almacenarlo de cierta forma, como por ejemplo en un modelo 3D, pero que esto no era posible porque muchas personas no tienen los conocimientos técnicos para poder implementar de forma manual una reconstrucción y todo lo que conlleva, como instalar las librerías, limpiar el modelo luego de obtener la reconstrucción, etc.

Es por esto que, con respecto al objetivo general de la memoria, se logró desarrollar un pipeline que abarque desde la grabación de un objeto y la subida del video por parte de un usuario a una plataforma web, hasta la reconstrucción 3D de objetos para su uso en exposiciones en un museo virtual, el cual ya existe, pero en un escenario totalmente nuevo donde el usuario pueda ver un catálogo con los objetos reconstruidos por las personas (mediante crowdsourcing), seleccionar un objeto, y colocarlo en el escenario. Esto fue desarrollado lo mayormente automatizado posible dado el estado de arte actual, puesto que se seleccionó una tecnología de reconstrucción que no era la óptima, pero que correspondía a la mejor opción al considerar los recursos disponibles para el proyecto.

Siguiendo con el desarrollo del pipeline, se efectuó un trabajo de ingeniería de software, guiándose por la metodología de desarrollo de software del Modelo V, que abarcó el proyecto dividiéndolo en tres pilares fundamentales, y planteando tanto requisitos de usuario como de software para satisfacer los requerimientos de cada pilar y finalmente el proyecto completo, requisitos los cuales fueron explicados junto al problema y posteriormente resueltos en la solución del trabajo.

En primer lugar, en cuanto al objetivo *“Crear una plataforma web que permita al usuario subir archivos y videos de objetos”*, se desarrolló una página web de diseño simple y directo, que sirve como enlace entre el usuario y el resto de los pilares. Esta página se diseñó con cuatro secciones que en su conjunto permiten al usuario contextualizarse sobre el proyecto por el cual se creó la plataforma, conocer el paso a paso de cómo navegar y entender la página, y finalmente subir videos a una plataforma web.

En segundo lugar, respecto al objetivo *“Evaluar y seleccionar algoritmos existentes de fotogrametría y reconstrucción 3D de objetos”*, se investigaron, evaluaron y seleccionaron tecnologías de reconstrucción 3D a partir de videos, y se concluyó utilizar una basada en fotogrametría, ya que era la adecuada dadas las condiciones actuales, es decir, teniendo un servidor que no posee una tarjeta gráfica. Así, se logró también el objetivo *“Obtener reconstrucción 3D de objetos con archivos subidos a plataforma web”*, permitiendo que el usuario suba un video a la plataforma web anteriormente descrita, y que luego de un tiempo obtenga su reconstrucción 3D lista para su uso en el museo virtual.

En tercer lugar, respecto al objetivo *“Visualizar en el museo virtual existente el modelo 3D obtenido por la reconstrucción”*, se creó un nuevo escenario, llamado *showroom*, donde es posible ver los objetos reconstruidos en un catálogo, colocarlos, rotarlos y ver su descripción. Sin embargo, no se cumplió con el objetivo de *“Permitir al usuario personalizar ciertos escenarios de forma simple con artículos de decoración básicos”*, ya que corresponde a un es-

cenario vacío donde se pueden manipular los objetos. Luego, en relación al objetivo “*Requerir de máximo 1 o 2 interacciones humanas (no usuario) en el pipeline que abarca los puntos anteriores*”, se logró subir y guardar el objeto grabado previamente, mediante la creación automatizada de un bundle del objeto, dejando como único trabajo manual la limpieza del modelo 3D para luego exportar el objeto al showroom, por lo que actualmente se requiere de solo 1 interacción humana.

En resumen, se logró desarrollar un pipeline donde cualquier tipo de perfil de usuario que desee integrarse al mundo de las exposiciones sea capaz de subir su propio video y ver su respectiva reconstrucción en un museo virtual. Este pipeline, de la forma más automatizada posible, es decir, automatizando desde la subida del video por la página web, hasta el arreglo del bundle listo para subirse al museo, dejando solo 1 interacción humana que consista en tomar el bundle del objeto y limpiar el modelo 3D para que después sea exportado al museo virtual mediante un procedimiento existente (gestionado por Gabriel Azócar). Además, se realizó una evaluación del trabajo, con resultados finales muy positivos, obteniendo así un feedback rápido por parte de los entrevistados, y logrando hacer mejoras durante la realización del trabajo, consiguiendo finalmente una versión mucho más completa de lo que es el enlace principal entre el usuario y el museo.

Cabe concluir adicionalmente que, en el caso de querer lograr una reconstrucción totalmente automática, es decir, donde se limpie el objeto sin necesidad de interacciones humanas como se hace actualmente, se necesita de un servidor con tarjeta gráfica para poder utilizar la tecnología *Instant NGP* investigada, junto a la librería *Rembg* que permite quitar el fondo de las imágenes previo a utilizar la reconstrucción 3D, y así obtener un mejor resultado final del objeto.

Finalmente, es posible proponer posibles trabajos futuros a realizar, ya que mediante la evaluación recibida se obtuvo un feedback en el que se recibieron sugerencias. Primero, se puede mejorar la paleta de colores de la página web, usando una temática más cercana a lo que sería un museo, para así atraer la atención del usuario y lograr que permanezca más tiempo dentro de la página y más cómodo; Segundo, implementar un sistema de registro de usuario junto a una notificación por correo, para que así el individuo que realice el procedimiento de subida pueda estar atento al estado en el que se encuentra su objeto; Tercero, considerar el algoritmo de reconstrucción 3D que hace uso de Deep Learning, pero solo en el caso de contar con un nuevo servidor que tenga GPU. Esto con la finalidad de obtener una mejor reconstrucción 3D y con mínimo nivel de ruido, puesto que se podría hacer uso de la herramienta para remover el fondo de las imágenes, para así lograr automatizar la limpieza del objeto y posteriormente la exportación al museo; Cuarto, crear los escenarios personalizables, ya que así el usuario podrá crear su propio espacio y así ambientalizar a su gusto el escenario en el cual pueda realizar una exposición.

Bibliografia

- [1] T. Kirner, A. Kawamoto J. Cantão A. Pinto R. Wazlawick(2001): *Development of a collaborative virtual environment for educational applications*61–68.
- [2] T. Hall, L. Bannon M. Fraser S. Benford J. Bowers C. Green halgh S. Hellstr om S. Izadi H.Schn adelbach M. Filntham(2001): *The visitor as virtual archaeologist: explorations in mixed reality technology to enhance educational and social interaction in the museum.*
- [3] Remondino, F. / El Hakim, S.(2006): *Image-based 3d modelling: A review*, 115: .
- [4] Arroyo, J. Oomen L.(2011): *Crowdsourcing in the cultural heritage domain: Opportunities and challenges*138–149.
- [5] E. Stathopoulou, G. Panagiotopoulos D. Kaliampakos(2015): *Crowdsourcing lost cultural heritage*295–300.
- [6] Shahbazi, M. / G. Sohn 2, J. Th eau / Menardm, P.(2015): *Development and evaluation of a uav-photogrammetry system for precise 3d environmental modeling.*
- [7] D. Biella, D. Sacher B. Weyers W. Luther N. Baloian T. Schreck(2016): *Crowdsourcing and co-curation in virtual museums: A practice- driven approach*, 10: 1277–1297.
- [8] Salazar Gamarra, R. / Seelaus, R. / Silva, J. Lopes da / Silva, A. Moreira da / Dib, L. Lauria(2016): *Monoscopic photogrammetry to obtain 3d models by a mobile device: A method for making facial prostheses.*
- [9] N. Baloian, D. Biella N. Karapetyan J. Pino T. Schreck A. Ferrada N. Hitschfeld(2017): *Exploring collaboration in the realm of virtual museums.*
- [10] Dhonju, H. / Xiao, W. / Mills, J. / Sarhosis, V. / Wilkinson, S. / Wang, Z. / Thapa, L. / Panday, U. S.(2017): *Feasibility study of low- cost image-based heritage documentation in nepal.*
- [11] Thibault Groueix, Vladimir G. Kim Bryan Russell Mathieu Aubry(2018): *AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation.*
- [12] N. Baloian, J. Pino D. Biella N. Karapetyan(2018): *Collaborati- ve virtual museums design, generation and exploration using standardized metadata.*
- [13] H. Dhonju, J. Mills / Sarhosis, V.(2018): *Share our cultural heritage (soch): Worldwide 3d heritage reconstruction and visualization via web and mobile gis*360.
- [14] E. Ch’ng, T. Zhang F. Leow(2019): *Crowdsourcing 3d cultural heritage: best practice for mass photogrammetry*, 1: 24–42.
- [15] E. Karachaliou, D. Psaltis E.Stylianidis(2019): *Uav for mapping historic buildings: from 3d modelling to bim*397–402.
- [16] Dellaert, Frank(2020): *NeRF Explosion 2020.*

- [17] Ben Mildenhall, Matthew Tancik Jonathan T. Barron Ravi Ramamoorthi Ren Ng(2020): *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*.
- [18] Xian Feng Han, Hamid Laga Mohammed Bennamoun(2021): *Image-Based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era*.1578–1604.
- [19] BootstrapVue (2023): *BootstrapVue Documentation*
, Accedido el 31 de marzo de 2023.
- [20] Foundation, Django Software (2022): *The Django Book: Version 3.2.* , Django Software Foundation.
- [21] Llul, Cristian (2022): *3D Reconstruction Pipeline of Cultural Objects using a Smartphone, propuesta de trabajo de tesis DCC Universidad de Chile.* .
- [22] Nuxt.js (2023): *Nuxt.js Documentation*
, Accedido el 31 de marzo de 2023.
- [23] Paolini, T. Barbieri P. (2001): *Cooperation metaphors for virtual museums, Museums and the Web 2001: Selected Papers from an International Conference, pp. 15–26, Seattle, Washington, March 15-17.* .
- [24] Pressman, Roger S. (2010): *Software Engineering: A Practitioner’s Approach.* , McGraw-Hill Education, 7th Auflage.
- [25] React (2023): *React Documentation*
, Accedido el 31 de marzo de 2023.