



ESCUELA DE INGENIERÍA EN COMPUTADORES

CE-5302: PROYECTO DE DISEÑO INGENIERÍA EN COMPUTADORES

Plan de Proyecto

Generación de nubes de puntos a partir de *stacking* sin información de pose de la cámara

PROFESOR

Luis Diego Noguera Mena

AUTOR

José Julián Camacho Hernández
2019201459

ASESOR

Luis Alberto Chavarría Zamora

1 de marzo de 2024

Tabla de contenidos

1	Contexto del proyecto	1
1.1	Nombre del proyecto	1
1.2	Nombre de la institución	1
1.3	Requerimientos de confidencialidad y propiedad intelectual	1
1.4	Descripción del problema	1
1.4.1	Contexto	1
1.4.2	Antecedentes	2
1.4.3	Planteamiento del problema	3
1.4.4	Restricciones de diseño	4
1.4.5	Usuarios	4
1.4.6	Otros aspectos relevantes	4
1.5	Objetivo general	4
1.6	Objetivos específicos	4
1.7	Interesados	5
2	Descripción del trabajo	6
2.1	Descripción de la solución	6
2.2	Entregables y criterios de aceptación	8
2.3	Actividades y propuesta de esfuerzo	10
2.4	Análisis de riesgos	11
2.5	Cronograma	11
3	Evidencia de aprobación del documento	13
4	Referencias	14

1. Contexto del proyecto

1.1. Nombre del proyecto

Generación de nubes de puntos a partir de *stacking* sin información de pose de la cámara.

1.2. Nombre de la institución

- **Institución:** Instituto Tecnológico de Costa Rica.
- **Departamento:** Escuela de Ingeniería en Computadores.

1.3. Requerimientos de confidencialidad y propiedad intelectual

- **Requerimientos de confidencialidad:** No se cuenta con requerimientos de confidencialidad en el desarrollo del presente proyecto.
- **Propiedad patrimonial:** La propiedad patrimonial del producto por desarrollar le corresponde al Instituto Tecnológico de Costa Rica, según el artículo 7, Capítulo 3 del Reglamento para la Protección de la Propiedad Intelectual del Instituto Tecnológico de Costa Rica. [1]
- **Propiedad intelectual y moral:** La propiedad intelectual y moral del producto por desarrollar le corresponde a los autores de este mismo.
- **Uso de *software* de código abierto:** Este proyecto será desarrollado y distribuido como código abierto, por lo que todos los aspectos del código fuente estarán disponibles públicamente. El producto será desarrollado bajo la licencia MIT para garantizar que los usuarios tengan el derecho de acceder, modificar y redistribuir el código de acuerdo con los principios de la comunidad de código abierto.

1.4. Descripción del problema

A continuación se presentan detalles respecto al problema que soluciona el producto por desarrollar.

1.4.1. Contexto

En la actualidad, la captura y procesamiento de datos tridimensionales desempeñan un papel fundamental en diversas aplicaciones, desde la robótica hasta reconstrucción en 3D. Este proyecto se enmarca en el contexto de la generación de mapas tridimensionales a partir de la captura de vídeos mediante vehículos no tripulados, en el cual surge un desafío crítico relacionado con la combinación eficiente de las nubes de puntos individuales. Este entorno implica la utilización de drones como herramienta de exploración y recolección de datos espaciales, donde la generación precisa de mapas tridimensionales es esencial.

El mapeo con drones es una tecnología del área de la teledetección, que tiene como fin crear mapas en 2D y 3D de un área utilizando datos de sensores situados en un dron o un vehículo aéreo no tripulado (VANT). Esta técnica busca generar mapas geoespaciales que contienen coordenadas de ubicación específicas del mundo real para cada punto de datos. Los datos de mapas embebidos permiten realizar mediciones del mundo real para aplicaciones de construcción, agricultura, planificación urbana y estudios topográficos. [2]

El proyecto se ubica en el ambiente de la fotogrametría, que es una de las técnicas más comunes de mapeos tridimensionales con drones, junto con el LiDAR (*Light Detection and Ranging*). Esta segunda consiste en un proceso de mapeo con mejores resultados independientemente de la visibilidad del entorno, pero utiliza sensores que son costosos y pesados.

Por su lado, la fotogrametría es un método más sencillo y económico debido a la gran disponibilidad de cámaras. Sin embargo, depende en gran medida de cuán visibles son las características de los datos capturados con cámaras. La visibilidad se ve afectada por la altura a la que navegue el dron y aspectos ambientales tales como oscuridad, nubes y niebla. [2]

Por su relación con la fotogrametría, el problema de este proyecto se presenta en el campo de la visión por computadora y procesamiento de imágenes, específicamente en la generación de nubes de puntos tridimensionales. Esta tarea resulta fundamental para procesar y apilar conjuntos de coordenadas a fin de crear mapas en 3D de áreas definidas.

1.4.2. Antecedentes

En el contexto presentado, donde para llevar a cabo los mapeos es necesario capturar objetos en movimiento en 3D desde múltiples ángulos o a lo largo del tiempo, se han desarrollado diversos algoritmos temporales.

Existen algunos algoritmos basados en representaciones en el formato de mallas. Sin embargo, estos métodos sufren de dos limitaciones principales: los datos proporcionados por los sensores 3D rara vez se presentan en formato de malla, y estos asumen un mapa de profundidad en continuo que no logra mantener correspondencias, las cuales son fundamentales para apilar la información. [3]

Otros estudios presentan arquitecturas de redes neurales que aprovechan representaciones de nubes de puntos para generar mapas globales. Un ejemplo de este tipo de tecnologías es el *SyNoRiM*, producto que genera resultados positivos, pero tiene dificultades para generalizar a nuevos dominios y cuenta con la limitación de que todos los conjuntos de datos utilizados para entrenamiento tienen un cambio de vista relativamente pequeño. [3]

El algoritmo *Iterative Closest Point* (ICP) es otro de los métodos clásicos para el registro rígido, que es el problema que se aborda en este proyecto: alinear un conjunto de puntos fuente con un conjunto de puntos objetivo. [4] Este algoritmo alterna entre consultas de puntos más cercanos en el conjunto objetivo y la minimización de la distancia entre puntos correspondientes, garantizando converger a un alineamiento óptimo local. No obstante, el ICP clásico puede experimentar una convergencia lenta debido a su tasa de convergencia lineal. [5]

Se han desarrollado otros métodos de registro con convergencia más rápida. Por ejemplo, algunos realizan la alineación técnica de minimización de la distancia de punto a plano, mien-

tras que en otros se minimiza una aproximación cuadrática local de la función de distancia al cuadrado. [5]

J. Zhang, Y. Yao y B. Deng presentan un método robusto que acelera su convergencia utilizando la aceleración de Anderson (AA). Esta es una técnica numérica establecida, que demuestra ser eficaz para una variedad de problemas de optimización en gráficos por computadora. Esta solución tiene la limitación de que depende de una buena alineación inicial. [5]

Otro de los estudios presentados sobre el tema fue realizado por Chen y Medioni, quienes publicaron un esquema de ICP similar utilizando un procedimiento de emparejamiento diferente basado en el vector normal de la superficie. Esta formulación solo es aplicable a puntos en superficies. [6]

Existen variantes del método de ICP como el denominado *Trimmed ICP* (TrICP). Este se basa en el uso consistente del enfoque de Mínimos Cuadrados Ajustados (LTS, por sus siglas en inglés) en todas las fases de la operación. [7]

El trabajo previo que aborda el propósito del presente proyecto de manera más completa es el llamado *PCStacking*. Este aprovecha al máximo las soluciones iterativas tanto en la estimación de la posición de la cámara como en los parámetros de calibración interna obtenidos durante el ajuste. El principio básico del algoritmo de apilamiento es el siguiente: calcula la mediana de las coordenadas Z de cada punto para múltiples modelos fotogramétricos, dando como resultado una nube de puntos con una precisión mayor que cualquiera de las nubes de puntos individuales. [8]

1.4.3. Planteamiento del problema

El proyecto se centra en abordar la problemática crucial de la generación de mapas tridimensionales a partir de nubes de puntos individuales. El problema incluye la imposibilidad de utilizar información sobre la pose de la cámara en cada instancia de captura para implementar la solución. Esta carencia de datos precisos sobre la orientación y posición de la cámara, presenta desafíos significativos, como el procesamiento, fusión y alineación correcta de las diferentes nubes de puntos, ya que estas provienen de diversas imágenes capturadas desde ángulos y alturas distintas.

Adicionalmente, en el contexto del proyecto del cual es parte el presente producto, esta dificultad se intensifica, ya que los drones deben funcionar en escenarios donde la variabilidad en las condiciones de iluminación, la presencia de obstáculos y los cambios en la topografía del terreno añaden complejidad al proceso de generación de mapas tridimensionales.

La relevancia de este trabajo radica en la necesidad de superar las restricciones en la creación de representaciones tridimensionales coherentes y precisas de entornos, especialmente en escenarios donde la información de la pose de la cámara no está disponible.

El propósito fundamental de este proyecto es desarrollar un algoritmo de *stacking* que brinde solución al problema planteado considerando sus limitaciones. La aplicación directa del producto, se traduce en la mejora significativa de la calidad y utilización de los mapas tridimensionales generados, facilitando el uso para personas interesadas y su aplicación en áreas como la exploración de terrenos remotos, la cartografía detallada y la planificación urbana.

Al abordar esta problemática, el producto no solo aspira a resolver un desafío técnico específico, sino también a proporcionar una solución práctica y económica para la generación de mapas tridimensionales en el contexto del proyecto grande del cual es pieza fundamental.

1.4.4. Restricciones de diseño

Entre las restricciones de diseño se encuentran los recursos computacionales para la ejecución del algoritmo, que deben ser suficientes para lograr los objetivos. Asimismo, otro aspecto a considerar es la disponibilidad de tecnologías necesarias para el manejo y procesamiento de conjuntos de datos tridimensionales, que no requieran la utilización de información relacionada con la pose de la cámara o propiedades de la misma. Dichas tecnologías deben ser suficientes para cumplir con el objetivo del proyecto y deben ser compatibles en el lenguaje de programación *Python*.

1.4.5. Usuarios

Los usuarios destinados a utilizar este producto son personas especializadas en el área de ingeniería en computadores o con sólidos conocimientos en programación y en conjuntos de datos tridimensionales.

1.4.6. Otros aspectos relevantes

No aplica.

1.5. Objetivo general

Desarrollar un algoritmo de *stacking* de nubes de puntos sin información de pose de cámara, por medio del uso de técnicas de visión por computadora y fotogrametría en un lenguaje de programación de alto nivel, con el propósito de crear mapas tridimensionales a partir de nubes de puntos individuales que carezcan de datos relacionados con la cámara utilizada para su generación.

1.6. Objetivos específicos

- Diseñar un algoritmo de *stacking* sin la utilización de información de la pose de la cámara, mediante el estudio y la aplicación de la teoría relacionada con nubes de puntos, con el fin de proporcionar una base conceptual sólida para su implementación práctica en lenguajes de programación de alto nivel.
- Implementar la propuesta diseñada de algoritmo de *stacking*, por medio del uso de técnicas avanzadas de procesamiento de nubes de puntos en el lenguaje de programación *Python*, para lograr una fusión coherente y precisa de conjuntos de datos tridimensionales.

- Evaluar el funcionamiento del producto de *software* mediante el uso de heurísticas y del criterio experto, con el fin de garantizar el correcto desempeño del producto en la combinación de nubes de puntos.

1.7. Interesados

A continuación, se presentan los involucrados y usuarios del producto identificados.

- ● Escuela de Ingeniería en Computadores: es la entidad educativa que respalda y supervisa el proyecto. Proporciona el marco institucional para el desarrollo del proyecto, asegurando que se alinee con los objetivos académicos y los estándares de calidad establecidos por la escuela.

El interés principal de la Escuela de Ingeniería en Computadores radica en la contribución del proyecto al desarrollo académico y la investigación. Además, busca impulsar la formación práctica del estudiante involucrado en el proyecto.

- Profesor Asesor: desempeña un papel crucial proporcionando orientación técnica y académica al desarrollador del proyecto. Supervisa y evalúa el progreso, asegurando que el proyecto cumpla con los estándares académicos y de calidad establecidos.

El profesor asesor tiene un interés particular en el éxito académico y técnico del proyecto. Busca fomentar el aprendizaje práctico, la aplicación de conocimientos teóricos y la contribución significativa al campo de estudio relacionado con la estimación de fondo monocular.

- Desarrollador: es la persona encargada de la implementación técnica del proyecto, desde el diseño hasta la entrega final del producto.

El interés del desarrollador se centra en la implementación exitosa del proyecto, la aplicación práctica de habilidades de programación y el logro de los objetivos definidos. Adicionalmente, busca aprender y mejorar sus habilidades a través de la experiencia práctica en un entorno académico y de investigación.

- Área de Investigación Estimación de Fondo Monocular: este campo de investigación proporciona el contexto y la base teórica para el desarrollo del proyecto. Sus avances y conocimientos influyen en la dirección y enfoque del proyecto, ya que se basa en los principios y descubrimientos del área de estudio.

El interés de esta área radica en la aplicación práctica de los conceptos teóricos y métodos desarrollados en el ámbito de la estimación de fondo monocular. Busca contribuir a la evolución y mejora de las técnicas utilizadas en proyectos con aplicaciones prácticas y significativas.

- **Usuarios:**

Los usuarios serán, en principio, el profesor asesor Luis Alberto Chavarría Zamora, que es el director del proyecto “Desarrollo de Aplicaciones de Procesamiento de Vídeo e Imágenes Orientadas hacia Vehículos Autónomos bajo condiciones de visión no ideales

para la Realidad Costarricense”, del cual es parte el presente producto; así como quien utilice dicho proyecto.

Adicionalmente, se identifica como usuario cualquier persona con interés en generar nubes de puntos globales a partir de nubes individuales por medio del producto, ya sea para fines académicos, en la industria, entre otros.

2. Descripción del trabajo

En la presente sección, se describen detalles relacionados al proceso de desarrollo del proyecto.

2.1. Descripción de la solución

La solución al problema planteado consiste en el desarrollo de un Generador de Nubes de Puntos. Dicho producto implementa un algoritmo de apilamiento de nubes de puntos sin utilizar información de la pose de la cámara, al realizar la combinación de conjuntos de puntos de entrada para producir una nube de puntos global de salida.

El Generador se integra en un sistema de vehículos autónomos para mapeos de áreas en tres dimensiones, que es parte del proyecto titulado “Desarrollo de Aplicaciones de Procesamiento de Vídeo e Imágenes Orientadas hacia Vehículos Autónomos bajo condiciones de visión no ideales para la Realidad Costarricense”. En esta colaboración, el sistema cuenta con módulos ya desarrollados y actualmente funcionales con los que captura vídeos y estima fondos monoculares, generando mapas de profundidad y nubes de puntos individuales por imagen. El Generador por desarrollar en el presente proyecto, se encarga únicamente de combinar estas nubes individuales en una representación tridimensional global.

En la figura 1, se visualizan los componentes del sistema general, así como sus diferentes interfaces y relaciones. En color gris se representan los módulos que ya se encuentran desarrollados. El Generador de Nubes de Puntos se presenta en color verde, y se identifican tanto sus fuentes de datos de entrada como el destino de su salida.

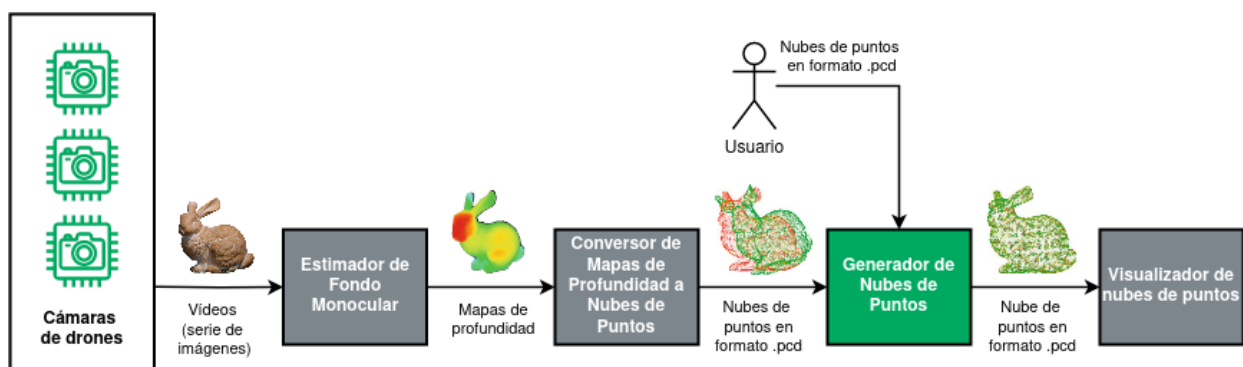


Figura 1: Componentes del sistema de vehículos autónomos para mapeos de áreas en tres dimensiones.

Esta modularidad posibilita al Generador adaptarse a otros sistemas que requieran la combinación de nubes de puntos tridimensionales, manteniendo claridad en las responsabilidades y asegurando una integración eficaz en el proyecto general.

En la figura 2, se presenta la arquitectura de la solución. En esta imagen, se destacan los diferentes componentes del producto de *software* en color verde. Asimismo, se detalla la relación que existe con los módulos del sistema de vehículos autónomos descrito previamente, que se visualizan de color gris, y con otro actor externo que corresponde al usuario.

En la arquitectura propuesta para lograr el objetivo, en primer lugar, el Generador debe utilizar los módulos de lectura según se presente alguno de los siguientes casos. Si los datos provienen del Conversor de Mapas de Profundidad a Nubes de Puntos, se leen los archivos en formato *pcd* que contienen los conjuntos de puntos. En caso contrario, si es el usuario externo quien utiliza el producto, se realiza la lectura de un archivo de configuración que contendrá información de las rutas donde se encuentran las nubes de puntos de entrada y de salida, así como valores para ajustar parámetros del algoritmo.

Una vez la lectura fue exitosa, los datos se pasan al componente de preprocesamiento, que tiene la función de manejar las nubes de puntos de entrada con el fin de eliminar ruido, hacer un submuestreo, normalizar los datos, entre otras técnicas para ajustar las nubes de puntos y así generar mejores resultados.

Seguidamente, el Generador lleva a cabo el algoritmo de apilamiento con las nubes de puntos preprocesadas. En dicho algoritmo se aplican técnicas de visión por computadora y fotogrametría para conocer la correspondencia entre puntos en el espacio, y de esa manera, alinear y combinar múltiples nubes de puntos con el objetivo de crear una representación tridimensional unificada del entorno.

Para finalizar el proceso, el producto cuenta con un módulo que se encarga de generar un archivo *pcd*, en el cual se almacena la información relacionada con la nube de puntos global. Dicho archivo, como se presenta en la figura, puede ser utilizado posteriormente para su análisis en el componente denominado Visualizador de Nubes de Puntos del sistema de vehículos autónomos, o en un ambiente independiente.

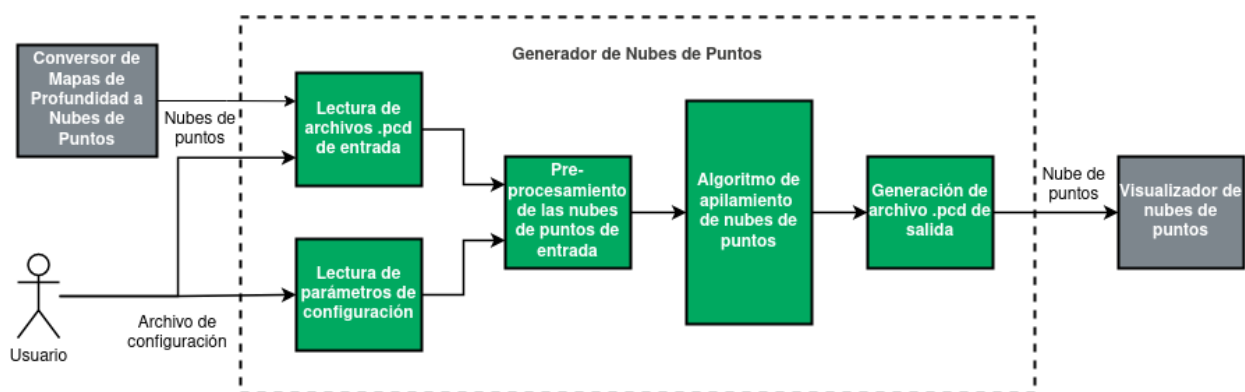


Figura 2: Arquitectura del producto de *software*.

2.2. Entregables y criterios de aceptación

En el cuadro 1, se presentan los entregables del proyecto, así como sus criterios de aceptación y los objetivos específicos a los que corresponden.

Id.	Entregable	Criterios de aceptación	Objetivos específicos relacionados
E-1	Documento de diseño del algoritmo de <i>stacking</i>	1. Se presenta un documento de diseño detallado y el mismo es aprobado tanto por el profesor del curso como la persona supervisora según lo establecido en la especificación. [9] 2. El algoritmo es diseñado para ser implementado en un lenguaje de programación de alto nivel.	1
E-2	Código en <i>Python</i> que implemente el algoritmo de <i>stacking</i> de nubes de puntos sin información de la pose de la cámara	1. El programa aplica el algoritmo de <i>stacking</i> sobre las nubes de puntos entrada y guardar la nube de salida. 2. El programa utiliza las direcciones de los archivos de entrada y salida indicados en el archivo de configuración. 3. El programa utiliza como insumo nubes de puntos en formato <i>pcd</i> . 4. El programa utiliza las configuraciones del algoritmo indicadas en el archivo de configuración. 5. El programa utiliza valores por defecto si el formato de las configuraciones especificadas no es el correcto. 6. El programa muestra un mensaje error y terminar su ejecución si: - El formato de los archivos de entrada no es el correcto. - Las nubes de puntos de entrada no tienen suficientes coincidencias. 7. El programa muestra un mensaje de éxito y de inicio del procesamiento cuando se validen los contenidos del archivo de configuración. 8. El programa muestra un mensaje de éxito en la finalización cuando se haya obtenido el archivo de salida al concluir el algoritmo.	2

Id.	Entregable	Criterios de aceptación	Objetivos específicos relacionados
E-3	Archivo que ejecuta la solución	1. El archivo permite la ejecución del programa de la solución por medio de la terminal o un <i>Jupyter Notebook</i> . 2. El archivo funciona como interfaz para la utilización por parte del usuario.	2
E-4	Plan de pruebas del funcionamiento del producto y reporte de resultados	1. El producto incluye un informe que detalle las pruebas por ejecutar. 2. Para cada prueba, el informe incluye lo siguiente: <ul style="list-style-type: none"> - Pasos e insumos para realizar la prueba. - Resultado esperado. - Resultados observables destacados. - Resultado de la prueba. 3. El informe incluye un resumen general de los resultados de las pruebas aplicadas, indicando si el producto aprueba las verificaciones.	3
E-5	Manual de usuario y archivo de ayuda	1. El producto incluye un archivo <i>README</i> con instrucciones claras sobre: <ul style="list-style-type: none"> - El uso del programa. - Método de ejecución. - Requisitos del sistema. 2. El producto incluye un manual de usabilidad en formato <i>pdf</i> para los usuarios finales. 3. La persona supervisora y usuario principal del producto valida que el propósito de los documentos se cumple.	2

Cuadro 1: Entregables, criterios de aceptación y objetivos relacionados.

2.3. Actividades y propuesta de esfuerzo

En la presente sección, en el cuadro 2, se detallan las actividades necesarias por realizar para construir la solución al problema y desarrollar cada uno de los entregables. Asimismo, se presenta el presupuesto de horas de ingeniería requeridas para cada actividad.

Identificador	Descripción	Horas requeridas
AC-1	Redactar documento de requerimientos	8
AC-2	Redactar documento de plan de proyecto	8
AC-3	Redactar documento de diseño	8
AC-4	Estudiar algoritmos de <i>stacking</i> , así como implementaciones existentes en <i>Python</i>	8
AC-5	Buscar conjuntos de datos de nubes de puntos que funcionen para desarrollar el programa	6
AC-6	Preparar el entorno de desarrollo, así como las bibliotecas necesarias	2
AC-7	Implementar código que lee y procesa el archivo de configuración	4
AC-8	Implementar código de preprocesamiento de las nubes de puntos de entrada	8
AC-9	Redactar informe de evaluación y presentación intermedia	6
AC-10	Desarrollar algoritmo de <i>stacking</i> de nubes de puntos sin información de pose de la cámara	16
AC-11	Implementar código que genere el archivo de nube de puntos de salida	8
AC-12	Optimizar el algoritmo de <i>stacking</i> para mejorar los resultados	10
AC-13	Crear el plan de pruebas para verificar el funcionamiento del producto	8
AC-14	Aplicar las pruebas unitarias para verificar los requerimientos que las necesiten	6
AC-15	Aplicar las pruebas funcionales de acuerdo con el plan de pruebas	8
AC-16	Redactar el reporte de los resultados de las pruebas funcionales	4
AC-17	Redactar el manual de usuario y archivo de ayuda	4
AC-18	Redactar informe y presentación final	8
Total		130

Cuadro 2: Actividades necesarias y presupuesto de horas de ingeniería requeridas.

2.4. Análisis de riesgos

A continuación, en el cuadro 3, se presenta el análisis de riesgos del proyecto.

No.	Descripción del riesgo	Categoría	Probabilidad	Impacto en el esfuerzo [horas]
1	Limitación de recursos computacionales: Se cuenta con el riesgo de no contar con suficientes recursos físicos para ejecutar el algoritmo, principalmente con grandes cantidades de datos de entrada.	Herramientas	15 %	6
2	Insuficientes datos de prueba: Debido a que el proyecto es parte de uno más grande, se cuenta con el riesgo de no tener datos de prueba suficientes para verificar la integración del sistema completo.	Insumos	10 %	10
3	Afectaciones en la salud del desarrollador: Existe la posibilidad de que el desarrollador del proyecto enfrente problemas de salud que impliquen tiempo de reposo o disminución de la productividad.	Personal	10 %	8
4	Falta de experiencia: Puede ocurrir que la implementación del algoritmo de apilamiento se vea afectada por la inexperiencia del desarrollador.	Procesos / Métodos	35 %	12
5	Cambios en los requerimientos por parte de la persona asesora: Se cuenta con el riesgo de que algún requerimiento sufra una modificación, lo que implica un cambio en el planteamiento del proyecto.	Procesos / Métodos	3 %	8

Cuadro 3: Análisis de riesgos.

2.5. Cronograma

Finalmente, en el cuadro 4, se presenta el detalle del cronograma del proyecto. Se incluyen las actividades por realizar, así como las relaciones entre las mismas, y las fechas de inicio y finalización planeadas.

Asimismo, en el siguiente enlace se adjunta el diagrama de *Gantt* de dicho cronograma: <https://app.instagantt.com/shared/65dd81d3d105ab5bddaa4e61>

Id.	Descripción	Fecha de inicio	Fecha de finalización	Dependencias
AC-1	Redactar documento de requerimientos	09/02	23/02	No aplica
AC-2	Redactar plan de proyecto	16/02	01/03	AC-1
AC-3	Redactar documento de diseño	23/02	08/03	AC-1 AC-2
AC-4	Estudiar algoritmos de <i>stacking</i> , así como implementaciones existentes en <i>Python</i>	26/02	12/03	No aplica
AC-5	Buscar conjuntos de datos de nubes de puntos	07/03	14/03	No aplica
AC-6	Preparar el entorno de desarrollo, así como las bibliotecas necesarias	11/03	14/03	No aplica
AC-7	Implementar código que lee y procesa el archivo de configuración	15/03	22/03	AC-6
AC-8	Implementar código de preprocesamiento de entrada	19/03	04/04	AC-5 AC-6
AC-9	Redactar informe de evaluación y presentación intermedia	04/04	12/04	Todas las previas
AC-10	Desarrollar algoritmo de <i>stacking</i> de nubes de puntos sin información de pose de la cámara	06/04	20/04	AC-4 AC-6
AC-11	Implementar código que genere el archivo de nube de puntos de salida	20/04	26/04	AC-10
AC-12	Optimizar el algoritmo de <i>stacking</i> para mejorar los resultados	27/04	04/05	AC-10
AC-13	Crear el plan de pruebas para verificar el funcionamiento correcto	01/05	08/05	AC-1
AC-14	Aplicar las pruebas unitarias para verificar los requerimientos que las necesiten	09/05	13/05	AC-13
AC-15	Aplicar las pruebas funcionales de acuerdo con el plan de pruebas	13/05	18/05	AC-13
AC-16	Redactar el reporte de resultados de las pruebas funcionales	19/05	22/05	AC-15 AC-14
AC-17	Redactar el manual de usuario y archivo de ayuda	20/05	27/05	AC-15
AC-18	Redactar informe y presentación final	24/05	31/05	Todas las previas

Cuadro 4: Cronograma de actividades del proyecto. Las fechas se expresan en formato DD/MM y corresponden al presente año 2024.

3. Evidencia de aprobación del documento

A continuación, en las figuras 3 y 4, se muestra la aprobación del presente documento por parte de la persona supervisora.

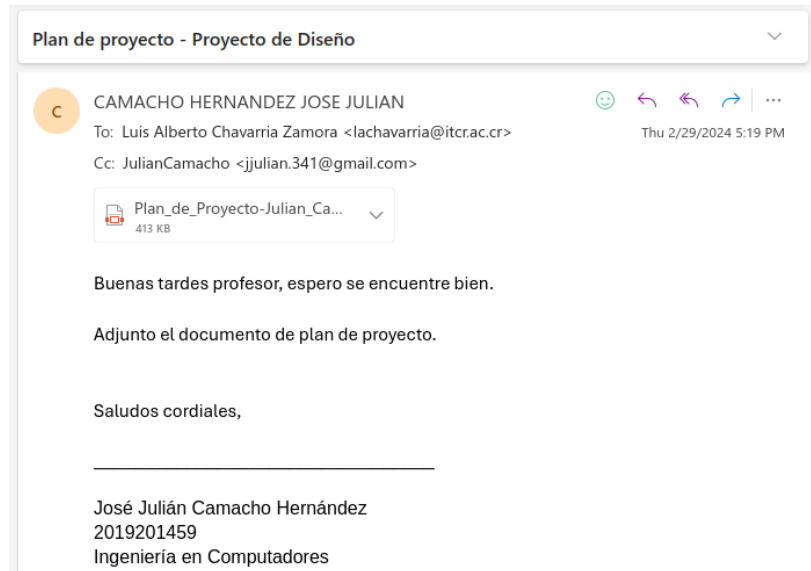


Figura 3: Evidencia de envío al asesor.

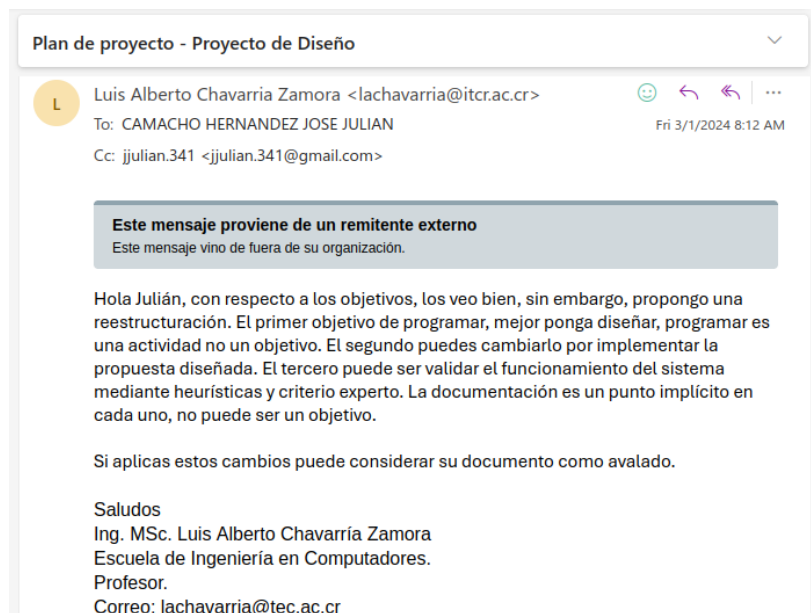


Figura 4: Evidencia de aprobación por parte del asesor.

4. Referencias

- [1] Instituto Tecnológico de Costa Rica. *Reglamento para la Protección de la Propiedad Intelectual del Instituto Tecnológico de Costa Rica*". [En línea]. Disponible en: <https://www.tec.ac.cr/reglamentos/reglamento-proteccion-propiedad-intelectual-instituto-tecnologico-cost Rica>. 2007.
- [2] MathWorks. "What Is Drone Mapping?" [En línea]. Disponible en: <https://mathworks.com/discovery/drone-mapping.html>. 2024.
- [3] J. Huang et al. «Multiway Non-rigid Point Cloud Registration via Learned Functional Map Synchronization». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.2 (2022), págs. 2038-2053. URL: <https://arxiv.org/pdf/2111.12878.pdf>.
- [4] P. J. Besl y N. D. McKay. «A Method for Registration of 3-D Shapes». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), págs. 239-256. URL: <https://ieeexplore.ieee.org/document/121791>.
- [5] J. Zhang, Y. Yao y B. Deng. «Fast and Robust Iterative Closest Point». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (2021), págs. 3450-3466. URL: <https://arxiv.org/pdf/2007.07627.pdf>.
- [6] Yang Chen y Gérard Medioni. «Object modelling by registration of multiple range images». En: *Image and vision computing* 10.3 (1992), págs. 145-155. URL: <http://graphics.stanford.edu/courses/cs348a-17-winter/Handouts/chen-medioni-align-rob91.pdf>.
- [7] Dmitry Chetverikov et al. «The trimmed iterative closest point algorithm». En: *2002 International Conference on Pattern Recognition*. Vol. 3. IEEE. 2002, págs. 545-548.
- [8] Xabier Blanch, Antonio Abellan y Marta Guinau. «Point cloud stacking: A workflow to enhance 3D monitoring capabilities using time-lapse cameras». En: *Remote Sensing* 12.8 (2020), pág. 1240. URL: <https://www.mdpi.com/2072-4292/12/8/1240>.
- [9] Escuela Ingeniería en Computadores. "Documento de Diseño". [En línea]. Disponible en: https://tecdigital.tec.ac.cr/dotlrn/classes/IDC/CE5302/S-1-2024.CA.CE5302.1/file-storage/view/Proyectos%2FCE-5302_Instrucciones_Documento_de_Dise%C3%B1o.pdf. 2024.