

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Desarrollo de una versión mejorada de un sistema LIDAR para el
escaneo tridimensional georreferenciado de calles

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero/a en Ciencias de la Computación

Presentado por:

Barreiro Palacios Doménica Valeria

Moreno Carrillo Robert Denilson

GUAYAQUIL - ECUADOR

Año: 2022-2023

DEDICATORIA

El presente proyecto se lo dedico a mi mamá, que es mi fuerza, ejemplo y motivación más grande. A mi familia y amigos que constantemente depositan su confianza y apoyo en mí.

A la Dra. Carmen Vaca quien me inspiró a ser parte de algo más grande, al Dr. Daniel Ochoa quien me enseñó a siempre considerar las leyes de Murphy.

Doménica Barreiro.

Dedico este proyecto a mi familia, a mis profesores, y a mis compañeros del club TAWS.

Robert Moreno.

AGRADECIMIENTOS

Agradezco a ESPOL por otorgarme el conocimiento y las herramientas necesarias para poder convertirme en una profesional.

A mis amigos quienes se volvieron parte integral de mi vida universitaria y espero conservarlos más allá de ella.

A mi familia que sin su apoyo no sería quien soy hoy.

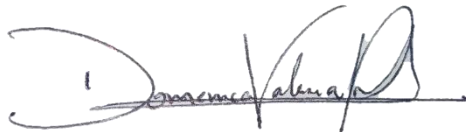
Doménica Barreiro.

Agradezco a ESPOL, por darme la oportunidad de desarrollarme como ingeniero en Computación. Agradezco a mi familia y amigos, por confiar en mí y brindarme su apoyo incondicional. Y agradezco especialmente a la Dra. Carmen Vaca por inspirarme a formar parte de esta hermosa carrera. Acompañándome durante todo este camino con su sabiduría y consejos.

Robert Moreno.

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Doménica Valeria Barreiro Palacios, Robert Denilson Moreno Carrillo y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



Doménica Valeria Barreiro
Palacios



Robert Denilson Moreno
Carrillo

EVALUADORES

Ing. Ronald Criollo

PROFESOR DE LA MATERIA

Ph.D. Daniel Ochoa

PROFESOR TUTOR

RESUMEN

La tecnología LIDAR es utilizada para mediciones y modelado geoespacial, teniendo múltiples aplicaciones en ingeniería y arquitectura. El CTD ha desarrollado un prototipo de un sistema de escaneo tridimensional basado en la tecnología LIDAR, para el cual se diseña una serie de pruebas y mejoras que permiten generar una nube de puntos georreferenciada de alta precisión.

Para el desarrollo de este proyecto se contó con un láser LIDAR, un sensor GPS, una Unidad de Medición Inercial, y una Raspberry Pi 3 como sistema embebido. Se utilizó la técnica de Navegación Cinemática en Tiempo Real para alcanzar mayor precisión de lecturas geográficas. Además, se realizaron múltiples operaciones matriciales para corregir las nubes de puntos geométrica y geográficamente.

Se logró independizar las frecuencias de captura de datos de los diferentes sensores que conforman el sistema de escaneo. También, se comprobó experimentalmente las operaciones geométricas necesarias para representar una nube de puntos en un espacio tridimensional. Y, finalmente, se logró visualizar de forma exitosa un escaneo tridimensional georreferenciado de obstáculos y objetos colocados en un trayecto.

Los resultados demuestran la factibilidad del uso del sistema de escaneo desarrollado para el escaneo tridimensional y georreferenciación. Sirviendo como base para el desarrollo de sistemas que permitan el mapeo y georreferenciación de obstáculos y deterioros en calles y carreteras del Ecuador.

Palabras Clave: LIDAR, calles, georreferenciación, nube de puntos, ROS

ABSTRACT

LIDAR technology is used to measure and model geospatially, having multiple applications in engineering and architecture. The CTD has developed a prototype of a three-dimensional scanning system based on LIDAR technology, for which a series of tests and improvements are designed to generate a high-precision georeferenced point cloud.

For the development of this project, we had a LIDAR laser, a GPS sensor, an Inertial Measurement Unit, and a Raspberry Pi 3 as an embedded system. The Real Time Kinematic Navigation technique was used to achieve greater precision of geographic readings. In addition, multiple matrix operations were performed to correct the point clouds geometrically and geographically.

It was possible to make the data capture frequencies of the different sensors that make up the scanning system independent. The geometry necessary to represent a point cloud in a three-dimensional space were also verified experimentally. Finally, it was possible to successfully visualize a georeferenced three-dimensional scan of obstacles and objects placed on a path.

The results show the feasibility of using the developed scanning system for three-dimensional scanning and georeferencing. Serving as a basis for the development of systems that allow the mapping and georeferencing of obstacles and deterioration in streets and highways of Ecuador.

Keywords: *LIDAR, Street mapping, georeferencing, point clouds, ROS.*

SIMBOLOGÍA

CVR	Centro de Visión y Robótica
CTD	Centro para la Transformación Digital Industrial
GPS	Sistema de Posicionamiento Global
GNSS	Sistema Global de Navegación por Satélite
RTK	Real Time Kinematic (Navegación Cinética Satelital en Tiempo Real)
IMU	Unidad de Medida Inercial
LIDAR	Laser Imaging Detection and Ranging (Detección y Rango de Imágenes Láser)

ÍNDICE

RESUMEN.....	I
<i>ABSTRACT</i>	II
SIMBOLOGÍA	III
ÍNDICE.....	IV
ÍNDICE DE FIGURAS.....	VI
ÍNDICE DE TABLAS	VIII
CAPÍTULO 1	1
1. Introducción	1
1.1 Descripción del problema	1
1.2 Justificación del problema.....	1
1.3 Objetivos.....	2
1.3.1 Objetivo General	2
1.3.2 Objetivos Específicos	2
1.4 Marco teórico	2
1.4.1 Sistema de escaneo tridimensional del CTD.....	2
1.4.2 Navegación Cinética Satelital en Tiempo Real	3
1.4.3 ROS	4
1.4.4 Nube de Puntos.....	5
CAPÍTULO 2.....	6
2. Metodología	6
2.1 Análisis	6
2.1.1 Requerimientos	6
2.1.2 Alcance y limitaciones de la solución	7
2.1.3 Riesgos y beneficios de la solución.....	7

2.1.4	Usuarios de la solución	9
2.2	Prototipado	9
2.3	Diseño de la Solución	11
2.3.1	Componentes de Hardware	11
2.3.2	Software	11
2.3.3	Arquitectura del sistema.....	11
2.3.4	Diagrama de flujo de solución	14
2.3.5	Frecuencias independientes de captura de datos	18
CAPÍTULO 3	19
3.	Resultados	19
3.1	Frecuencia efectiva de los sensores.....	19
3.2	Funcionamiento de los sensores a frecuencias independientes.....	21
3.3	Sincronización de datos.....	22
3.4	Generación de nube de puntos con datos sincronizados	24
3.4.1	Proyección Matricial	24
3.4.2	Proyección Tridimensional. Corrección geométrica.	25
3.4.3	Georreferenciación. Corrección geográfica.....	26
CAPÍTULO 4	36
4.	CONCLUSIONES Y RECOMENDACIONES	36
4.1	Conclusiones	36
4.2	Recomendaciones	37
BIBLIOGRAFÍA	38

ÍNDICE DE FIGURAS

Figura 1.1: Diagrama de funcionamiento de la técnica Real-Time Kinematics [Autoría propia].....	4
Figura 1.2: Nube de puntos de alta resolución [10].....	5
Figura 2.1: Prototipo del sistema de escaneo durante una prueba de campo, ESPOL [Autoría propia]	9
Figura 2.2: Acercamiento del prototipo del sistema de escaneo [Autoría propia]	10
Figura 2.3: Diagrama de la arquitectura del Sistema de Escaneo Tridimensional del CTD [3]	13
Figura 2.4: Diagrama de la arquitectura del Sistema de Escaneo Tridimensional propuesto [Autoría propia]	13
Figura 2.5: Diagrama de flujo de módulo de captura de datos [Autoría propia]	15
Figura 2.6: Diagrama de flujo del módulo de procesamiento [Autoría propia]	17
Figura 3.1: Comparación de frecuencia teórica vs frecuencia máxima efectiva de la IMU [Autoría propia].....	19
Figura 3.2: Comparación de frecuencia teórica vs frecuencia máxima efectiva del láser [Autoría propia]	20
Figura 3.3: Comparación de frecuencia teórica vs frecuencia máxima efectiva del RTK [Autoría propia]	20
Figura 3.4: Frecuencia efectiva de los 3 sensores [Autoría propia]	21
Figura 3.5: Extracto de un dataset generado por el sensor LiDAR [Autoría propia]...	22
Figura 3.6: Extracto de un dataset generado por el sensor IMU [Autoría propia]	22
Figura 3.7: Extracto de un dataset generado por el sensor GPS-RTK [Autoría propia]	23
Figura 3.8: Dimensiones de los dataset ejemplo previo a la sincronización de los datos [Autoría propia]	23
Figura 3.9: Dimensiones de los dataset ejemplo luego de la sincronización de los datos [Autoría propia]	23
Figura 3.10: Ejemplo de nube de puntos proyectada de forma matricial [Autoría propia]	24
Figura 3.11: Ejemplo de nube de puntos proyectada en el espacio tridimensional relativo al láser [Autoría propia]	25

Figura 3.12: Ejemplo de nube de puntos georreferenciada [Autoría propia].....	26
Figura 3.13: Foto del sitio de prueba de escaneo [Autoría propia]	27
Figura 3.14: Nube de puntos desde la perspectiva correspondiente a la <i>Figura 3.13</i> [Autoría propia]	27
Figura 3.15: Diagrama de los ejes de rotación del prototipo (cuerpo) alineados con las coordenadas Norte, Este y Abajo [Autoría propia]	28
Figura 3.16: Diagrama del cuerpo en estado <i>alpha</i> [Autoría propia].....	29
Figura 3.17: Diagrama del cuerpo en estado <i>beta</i> [Autoría propia]	29
Figura 3.18: Diagrama del cuerpo en estado <i>gamma</i> [Autoría propia].....	30
Figura 3.19: Diagrama del cuerpo en estado <i>alterado</i> , con todas sus variaciones angulares [Autoría propia].....	30
Figura 3.20: Nube de puntos escaneada con el cuerpo en estado <i>alpha</i> con un ángulo de 20 grados. Izquierda: Sin corregir. Derecha: Corregido [Autoría propia]	31
Figura 3.21: Nube de puntos escaneada con el cuerpo en estado <i>beta</i> con un ángulo de 15 grados, sin corregir [Autoría propia].....	31
Figura 3.22: Nube de puntos escaneada con el cuerpo en estado <i>beta</i> con un ángulo de 15 grados, corregida. [Autoría propia].....	32
Figura 3.23: Nube de puntos escaneada con el cuerpo en estado <i>gamma</i> con un ángulo de 90 grados, sin corrección [Autoría propia]	33
Figura 3.24: Nube de puntos escaneada con el cuerpo en estado <i>gamma</i> con un ángulo de 90 grados, corregida [Autoría propia].....	33
Figura 3.25: Nube de puntos escaneada con el cuerpo en estado <i>alterado</i> , corregida [Autoría propia]	34
Figura 3.26: Nube de puntos escaneada con el cuerpo en estado <i>alterado</i> , acercamiento a los objetos [Autoría propia].....	34
Figura 3.27: Izquierda: Foto del sitio de la prueba con objetos a escanear. Derecha: Perspectiva de la foto en la nube de puntos de la <i>Figura 3.25</i> [Autoría propia].....	35

ÍNDICE DE TABLAS

Tabla 2.1: Tabla de requerimientos	6
Tabla 2.2: Componentes de Hardware	11
Tabla 2.3: Frecuencia máxima de captura de datos de los sensores utilizados	18
Tabla 3.1: Frecuencia Teórica vs Frecuencia Efectiva	21

CAPÍTULO 1

1. INTRODUCCIÓN

1.1 Descripción del problema

Los departamentos de Obras Públicas de los GADs municipales son responsables de los espacios públicos, infraestructura y seguridad vial. Parte de las funciones de estos departamentos son: realizar la apertura, conservación y mantenimiento de las vías públicas; identificar deterioro de las vías urbanas e interurbanas, mapeo y georreferenciación de bienes públicos [1].

Para cumplir con la identificación y rectificación del deterioro de vías, los departamentos de obras públicas reciben solicitudes a través de redes sociales que son asignadas a técnicos que realizan la verificación, georreferenciación e inspección del defecto. Sin embargo, la mayoría de los defectos no se reportan y las calles se van deteriorando paulatinamente, al punto que se debe repavimentar toda la calle.

En los últimos años, han surgido tecnologías LIDAR (Light Detection and Ranging) para el mapeo y escaneo de calles y reconocimiento de obstáculos. No obstante, esta tecnología ha sido mayormente enfocada en la conducción autónoma [2] y no ha sido aprovechada para el mantenimiento y conservación de las vías públicas.

1.2 Justificación del problema

Una tecnología basada en un escáner LIDAR capaz de digitalizar y mapear las vías públicas del Ecuador sería fundamental para la detección automática de obstáculos, baches y grietas en el pavimento. Los datos LIDAR georreferenciados permitirán la implementación de sistemas de análisis de deterioro y daños en vías públicas, ahorrando tiempo y dinero a los gobiernos municipales.

El CTD (Centro para la Transformación Digital) cuenta con un sistema de escaneo tridimensional basado en LIDAR [3]. En nuestro proyecto, vamos a reutilizar componentes de software y hardware existentes, y crear nuevos componentes que

permitan el mapeo de vías públicas con alta precisión. Esto permitirá la futura implementación y evaluación de un prototipo de sistema de detección de obstáculos y defectos en calles.

1.3 Objetivos

1.3.1 Objetivo General

Mejorar la implementación del sistema de escaneo tridimensional LIDAR para el mapeo y georreferenciación de calles.

1.3.2 Objetivos Específicos

- Diseñar un plan de pruebas y mejoras a partir del análisis del estado actual del escáner LIDAR.
- Recolectar de forma robusta datos de sensores LIDAR, IMU y GPS con frecuencias de muestreo variable.
- Implementar mejoras en el algoritmo de generación de nube de puntos georreferenciados.

1.4 Marco teórico

1.4.1 Sistema de escaneo tridimensional del CTD

En [3] se desarrolló un prototipo de un sistema de escaneo tridimensional y georreferenciador de calles basado en un escáner LIDAR para el CVR (Centro de Visión y Robótica, ahora CTD). Este cuenta con un láser LIDAR, un receptor GPS (Global Positioning System) y un sensor IMU (Inertial Measurement Unit).

- Sensor LIDAR: El láser LIDAR es un dispositivo que está diseñado para producir y recibir luz, emite energía en forma de pulsos que son enviados a través de la atmósfera, y se reflejan a través de una superficie detectada [4].
- Receptor GPS: Los sensores de geoposicionamiento son capaces de detectar señales de satélites en un sistema de coordenadas geográficas y las transforma en ubicaciones. En un sistema de escaneo tridimensional permite la

georreferenciación de los puntos generados por el láser LIDAR, haciendo posible la estimación de distancias y alturas, generando un mapeo espacial a escala.

- Sensor IMU: Este sensor captura la aceleración e inclinación en un sistema de coordenadas propio, y mide la tasa de cambio respecto a una posición. Es un sensor imprescindible para sistemas de escaneo tridimensional móviles, ya que la información del movimiento se usa para conocer la orientación espacial del sensor LIDAR.

1.4.2 Navegación Cinética Satelital en Tiempo Real

Los sistemas GNSS se usan para ubicar con relativa exactitud a un receptor terrestre en el globo por medio de la comunicación entre una red de satélites, también llamadas constelaciones, que orbitan alrededor del planeta. Estos sistemas se utilizan primordialmente en navegación, georreferenciación y cronometría [5]. A lo largo de las últimas décadas varios países han creado sus propios sistemas GNSS: el GPS de Estados Unidos, GLONASS de Rusia, Galileo de la Unión Europea y el más reciente Beidou de China [6].

La incertidumbre promedio de las mediciones de un receptor GNSS es de 2 o 3 metros [6]. RTK (Real-Time Kinematic) es una técnica de corrección en tiempo real cuyo objetivo es mejorar la precisión de señales GNSS [7].

RTK explota el desfase de las señales satelitales buscando corregir por medio de métodos estadísticos las posiciones GNSS. Utilizando una estación base fija, que es un receptor GNSS con posición conocida, y una estación móvil (también llamada "rover"), que es un receptor GNSS con posición desconocida; se busca la comunicación entre estos receptores donde la base envía mensajes de corrección a la estación móvil para ser procesados y obtener una alineación en el desfase del posicionamiento [7] como se puede ver en la *Figura 1.1*.

La alineación pasa por tres fases: adquisición, modo de corrección "Float" y modo de corrección "Fixed". El modo de corrección "Float" tiene una ambigüedad flotante

que implica que el algoritmo aún no resuelve el desfase correcto entre los receptores y solo los infiere, en este modo la alineación tiene una precisión de 20 a 75 cm. El modo de corrección “Fixed” tiene una ambigüedad fija, es decir, que el algoritmo ha calculado la solución del desfase correctamente las mediciones alcanzan precisiones de 1 a 3 cm [7].

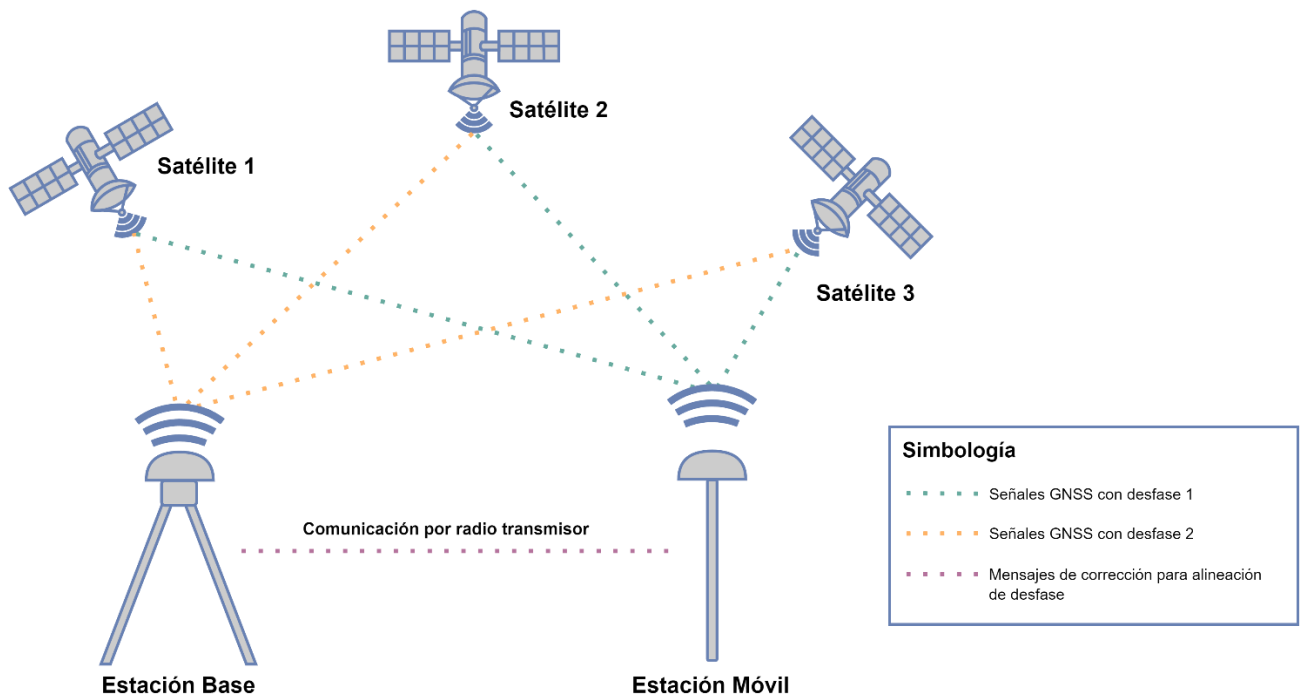


Figura 1.1: Diagrama de funcionamiento de la técnica Real-Time Kinematics [Autoría propia]

1.4.3 ROS

Robot Operating System (ROS) es un middleware de código abierto diseñado para el desarrollo de sistemas robóticos complejos. Actualmente es considerado el principal marco de desarrollo para aplicaciones de robótica [8]. En el sistema de escaneo tridimensional del CTD [3], se utiliza ROS por su flexibilidad y capacidad de manejo de archivos y fácil comunicación con dispositivos a bajo nivel. Los sistemas creados con ROS utilizan procesos independientes, llamados nodos, se comunican usando el paradigma publicador-suscriptor [9]. ROS es independiente del lenguaje de programación, permite que los nodos sean construidos en: C++, Python, Octave y LISP [9].

1.4.4 Nube de Puntos

Una nube de puntos es una representación de información tridimensional obtenida a través de fotogrametría digital, o por sistemas de escaneo tridimensional. La información provista por estos sistemas permite estimar características tridimensionales del espacio y/o objeto de interés. Características como tamaño, ubicación, inclinación, entre otras. Las nubes de puntos tienen diversas aplicaciones en la industria: control e inspección de instalaciones industriales, levantamiento topográfico y de infraestructuras entre otros.

La resolución de una nube de puntos depende de la frecuencia y campo de visión de escaneo del sensor LIDAR. En un sistema de escaneo móvil la densidad se ve afectada por la velocidad del vehículo. Los escáneres de alta resolución reportan precisión de centímetros y en algunos casos de milímetros. En la *Figura 1.2* se muestra una nube de puntos de alta resolución.



Figura 1.2: Nube de puntos de alta resolución [10]

CAPÍTULO 2

2. METODOLOGÍA

2.1 Análisis

2.1.1 Requerimientos

Tabla 2.1: Tabla de requerimientos

Requerimientos Funcionales	
Identificador	Requerimiento
F001	El sistema debe permitir que los sensores recolecten datos con frecuencias independientes.
F002	El sistema debe buscar la correspondencia temporal de los datos recopilados por los sensores.
F003	El sistema debe implementar algoritmos de reducción de ruido de los datos de los sensores.
F004	El sistema debe generar una nube de puntos georreferenciada utilizando los datos de los sensores.
Requerimientos No Funcionales	
Identificador	Requerimiento
NF001	El sistema debe almacenar los datos en formato BAG.
NF002	El sistema debe permitir la generación de Nubes de puntos en formato LAS 1.8.
NF003	El sistema debe incorporar nuevos módulos desarrollados en C++
NF004	El sistema de escaneo se debe instalar en una plataforma móvil.

2.1.2 Alcance y limitaciones de la solución

El alcance de la solución propuesta es obtener una implementación funcional y mejorada del sistema de escaneo tridimensional LIDAR disponible en el CTD; en el cual se puedan integrar diferentes sensores que capturen datos a frecuencias variables. Además, de la inclusión de los filtros de Kalman y otros algoritmos para el filtrado y procesamiento de los datos obtenidos por el sistema LIDAR, de tal forma que estos datos puedan ser utilizados para la creación de nubes de puntos georreferenciadas en formato LAS.

Dentro de las limitaciones identificadas de la solución se encuentran:

- El sistema LIDAR previo fue desarrollado con los componentes disponibles por el CTD, y las mejoras planteadas serán mejoras de software y no de hardware.
- Actualmente se cuenta con un láser LIDAR cuyo rango es de 270° y una frecuencia de escaneo de 40 Hz. Este es un láser pequeño cuyas características son aceptables, pero existen mejores alternativas en el mercado que brindan una mayor precisión y frecuencia de escaneo.
- El computador embebido utilizado para alojar y hacer funcionar el sistema LIDAR actual es una Raspberry PI 3, que cuenta con 1 Gb de RAM y una memoria externa SD con 16 Gb. Una vez realizada la independización y aumento de las frecuencias de recolección, será necesario evaluar si las capacidades descritas son suficientes para hacer funcionar el sistema por tiempos prolongados sin ningún problema.

2.1.3 Riesgos y beneficios de la solución

Dentro de los riesgos inmediatos en el desarrollo y uso de la solución tenemos:

- **Tiempo de expedición del Hardware:** Lo clasificamos como un riesgo técnico conocido. Ya sea por constante manipulación, desgaste natural, depreciación del modelo o incluso algún accidente, los componentes de hardware (particularmente los sensores) deberán pasar por un proceso de mantenimiento, de recalibración o ser por completo reemplazados, creando un riesgo en el uso

de la solución en caso de los dos primeros y en el último caso, un riesgo en el desarrollo de la solución. Esto último, se debe a que la parametrización de ciertas variables y el proceso de comunicación con los sensores dependen del fabricante y modelo de estos, generando la necesidad de alterar el código existente para que sea compatible con el nuevo sensor (en caso de no ser el mismo modelo o versión).

Es casi inevitable que este riesgo ocurra, y la mejor manera de manejarlo es procurar extender el tiempo de vida de los componentes con: un buen manejo del sistema, revisiones regulares del estado de los sensores y mantenimientos preventivos; todo esto por medio de manuales técnicos y de uso que permitan a los desarrolladores y usuarios ser conscientes de este riesgo.

- **Complejidad tecnológica:** Lo clasificamos como un riesgo predecible del proyecto. Al momento de iniciar el proyecto, como desarrolladores pudimos reconocer que el desconocimiento de la tecnología base y la inexperiencia de trabajar en proyectos con alto nivel de complejidad técnica nos pone en desventaja y representa un riesgo para el proyecto. Por ello, nuestra principal actitud fue aceptar nuestras limitaciones en conocimientos y, sobre todo, compensar estas debilidades con una planificación dedicada enteramente a la investigación de la tecnología, los aspectos técnicos y familiarización con proyectos ya existentes que manejen el mismo nivel de complejidad con la misma tecnología base.

Dentro de los beneficios, identificamos una demanda local en el uso de este tipo de tecnologías para el levantamiento geográfico con diferentes fines y aplicaciones, que puede ser satisfecha con nuestra propuesta. La introducción de tecnología LIDAR en sistemas de mapeo de calles y terrenos, identificación de baches y demás, es imperativa, por lo que tener una opción de bajo costo y por ende más accesible es una gran oportunidad para futuras iniciativas.

2.1.4 Usuarios de la solución

- Profesionales de las áreas de Ingeniería Civil y Arqueología que se encuentran involucrados en tareas de levantamiento geográfico y mapeo 3D.
- Autoridades municipales encargadas del mantenimiento de la infraestructura pública, y de la seguridad vial.
- Centros de investigación que deseen tomar esta tecnología como base para la creación de nuevos proyectos, con aplicaciones más específicas.

2.2 Prototipado

Para facilidad de transporte y movilidad al escanear, se montó el sistema LIDAR con sus sensores y demás componentes en un carro de carga soportado por una estructura de aluminio, como se muestra en las *Figura 2.1* y *Figura 2.2*, que eleva el láser LIDAR y a las antenas GPS del sensor RTK, para permitir el escaneo de la superficie en la que se traslada. Además, se cuenta con un trípode que soporta la estación base del sensor RTK.



**Figura 2.1: Prototipo del sistema de escaneo durante una prueba de campo, ESPOL
[Autoría propia]**



Figura 2.2: Acercamiento del prototipo del sistema de escaneo [Autoría propia]

2.3 Diseño de la Solución

2.3.1 Componentes de Hardware

Tabla 2.2: Componentes de Hardware

Nombre dispositivo	Descripción	Documentación
Raspberry Pi 3	Computador embebido que permite la ejecución de los scripts del sistema para la recolección de datos.	[11]
Adafruit IMU BNO055	Sensor inercial absoluto, incorpora un acelerómetro y magnetómetro que proporcionan datos del movimiento y trayectorias.	[12]
SwiftNav Piksi v2.0	Antena receptora de señales GPS que permite la incorporación de técnicas RTK para la georreferenciación de los puntos.	[13]
XBEE Pro-S2C	Antena programable para comunicación de radio. Trabaja en conjunto con las antenas GPS para realizar la corrección geográfica de las coordenadas.	[14]
Láser Hokuyo UST-10LX	Láser LIDAR, tecnología que irradia un haz de luz sobre una superficie y capta información de la distancia e intensidad a la que la superficie refleja dicha luz.	[15]

2.3.2 Software

- Ubuntu: Sistema operativo de código libre. Es requerido en la versión 16.04 LTS por compatibilidad con el marco de trabajo.
- ROS: Marco de trabajo utilizado. El proyecto fue desarrollado utilizando la versión Kinetic.
- C++: Lenguaje de programación elegido para la construcción de los nodos y sus funcionalidades bajo el marco de trabajo.

2.3.3 Arquitectura del sistema

El diagrama de arquitectura de sistema de escaneo tridimensional del CTD desarrollado en [3] se visualiza en la *Figura 2.3* **Error! Reference source not**

found.. Este sistema posteriormente fue actualizado por el CTD quitando componentes como la cámara y el detector GCP.

El diagrama de la arquitectura del sistema que proponemos se muestra en la *Figura 2.4*. La solución se planteó bajo un esquema de 2 módulos. Un módulo de captura de datos que se ejecuta dentro de la Raspberry y que interactúe con los sensores. Y un módulo de procesamiento que puede ejecutarse fuera del sistema embebido.

El módulo de captura de datos está compuesto por los siguientes nodos:

- *ControlSistema*: Nodo principal del sistema, encargado de inicializar y de apagar los otros nodos.
- *TimeServer*: Nodo utilizado para la sincronización del tiempo de inicio de captura de datos de los demás sensores.
- *RTK*: El Nodo RTK se encarga de la captura de datos de información geolocalizada provista por el sensor GPS-RTK a su propia frecuencia, y de publicar estos datos.
- *IMU*: El Nodo IMU se encarga de la captura de datos del sensor IMU a su propia frecuencia, y de publicar estos datos.
- *Láser*: El Nodo de láser se encarga de la captura de datos del láser LIDAR Hokuyo a su propia frecuencia, y de publicar estos datos.
- *Record*: El Nodo Record se encarga de encolar los datos publicados por los sensores RTK, IMU y Láser, y de guardarlos en archivos bag provistos por una librería del marco de trabajo llamada ROSBAG.

El módulo de procesamiento cuenta con un solo nodo.

- *Processing*: Este nodo se encarga de extraer los datos del archivo bag, filtrarlos y prepararlos para generar la nube de puntos en formato LAS.

Los archivos resultantes del nodo de procesamiento son 3.

- *Cloud_raw*: Nube de puntos con datos sin procesar.
- *Cloud_raw2*: Nube de puntos únicamente con datos del láser, y con corrección geométrica.
- *Cloud*: Nube de puntos con corrección geométrica, espacial y georreferenciación.

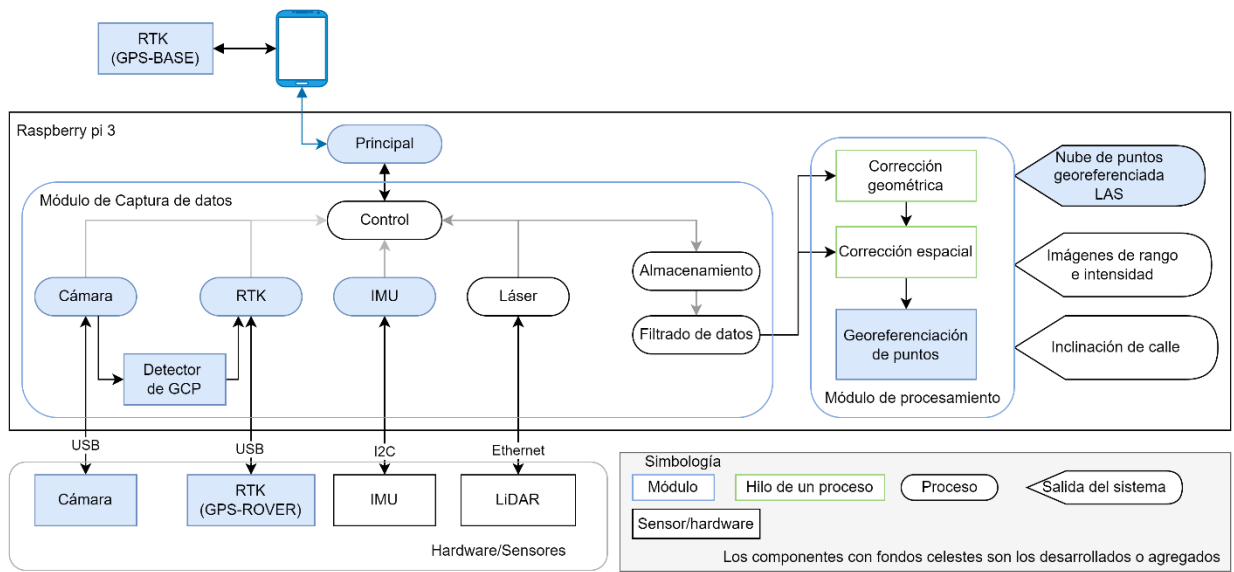


Figura 2.3: Diagrama de la arquitectura del Sistema de Escaneo Tridimensional del CTD [3]

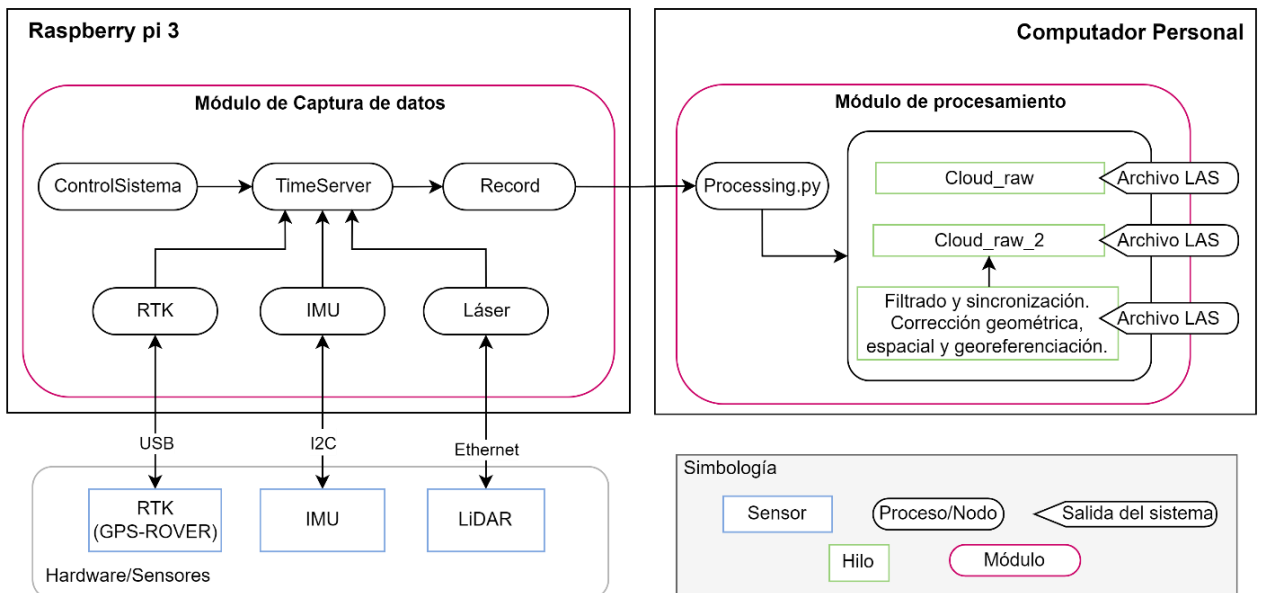


Figura 2.4: Diagrama de la arquitectura del Sistema de Escaneo Tridimensional propuesto [Autoría propia]

2.3.4 Diagrama de flujo de solución

Módulo de captura de datos

Tal y como se muestra en la *Figura 2.5*, el flujo del módulo de captura y almacenamiento de datos comienza con la inicialización del nodo *ControlSistema*, que cumple el rol de nodo maestro, y se encarga de inicializar los demás nodos. Los nodos son inicializados en el orden: IMU, RTK, Láser, Record y TimeServer.

Una vez el nodo *TimeServer* es inicializado, los nodos *IMU*, *RTK* y *Láser* comienzan con la captura de datos, cada uno publicando datos a su propia frecuencia, mientras que el nodo *Record* se encarga de encolar estos datos y almacenarlos en archivos diferentes (1 para cada sensor) dentro de un solo archivo serializado BAG ACTIVE.

Este proceso de captura y recolección se mantiene activo hasta que se mande una señal para finalizarlo, lo que cesa la captura y recolección de datos, apaga los nodos y finalmente convierte los archivos BAG ACTIVE en archivos BAG, que es el formato final en el que se deben almacenar los datos para posteriormente recuperar los archivos serializados y poder exportarlos en formatos TXT o CSV.

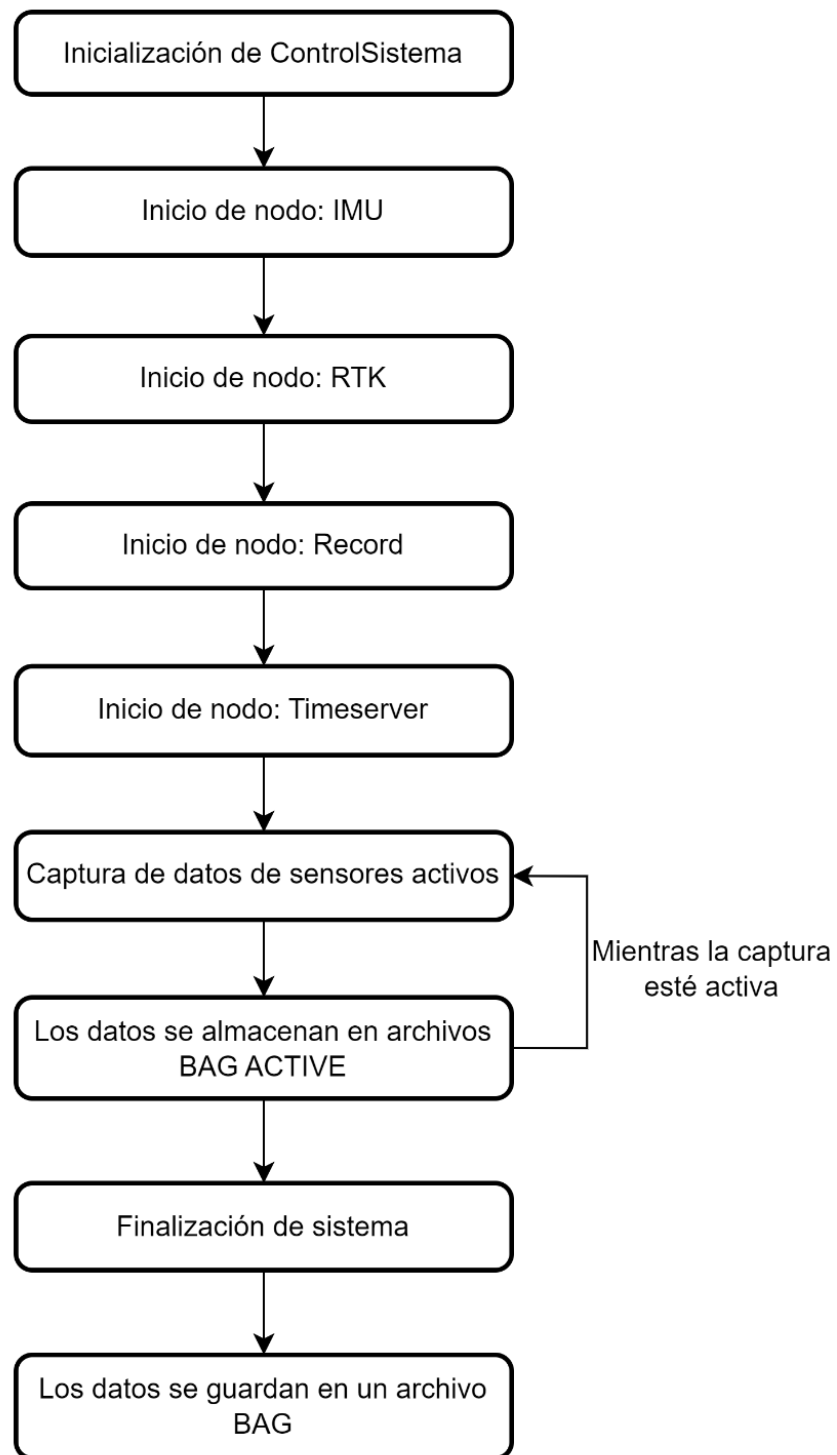


Figura 2.5: Diagrama de flujo de módulo de captura de datos [Autoría propia]

Módulo de procesamiento

En la *Figura 2.6* se presenta el flujo de trabajo del nodo de procesamiento que va a tener nuestra solución. Como primer paso, tenemos la inicialización del sistema por medio del nodo *Control/Sistema* que a su vez se encarga de encender el nodo de procesamiento. A continuación, se ejecuta la primera fase del procesamiento: la extracción de los datos en formato CSV, se eligió este formato en lugar del TXT que se usaba en el sistema anterior, debido a lo fácil que es de manejar y visualizar en otras herramientas por lo estándar que es. Se extraen los datos obtenidos por cada sensor en un archivo independiente para luego ser debidamente procesados.

Se procede a hacer la verificación y filtrado de los datos, lo cual se realiza por medio de la validación de los valores esperados por cada sensor y si estos se encuentran en un rango determinado por las especificaciones del hardware, si no se llega a cumplir estas condiciones se ignoran esos datos en la siguiente etapa de procesamiento.

Una vez que se obtengan todos los datos válidos se procede a realizar la sincronización temporal de los mismos. En el sistema anterior, la sincronización era simplemente verificar las marcas de tiempo de los datos y mantener los datos en común que era bastante sencillo ya que todos los sensores capturaban datos a una misma frecuencia. Sin embargo, en nuestra solución la independencia de las frecuencias provoca que ciertos sensores capturen mayor cantidad de datos por segundo que otros. Esta diferencia de cantidad de datos nos empuja a realizar una agregación de tal forma que al final de la sincronización todos los archivos tengan la misma cantidad de datos y, por conveniencia, la misma marca temporal. Para realizar este proceso, se proponen dos métricas de agregación: medidas de tendencia central (media, mediana) y un algoritmo de estimación llamado Filtro de Kalman. Después de la agregación de los datos, se procede a realizar la corrección espacial, corrección geométrica y georreferenciación de cada punto utilizando los datos del LIDAR, la IMU y el GPS-RTK respectivamente.

Finalmente, se genera la nube de puntos en el formato LAS especificado en los requerimientos (véase *Tabla 2.1*: Tabla de requerimientos).

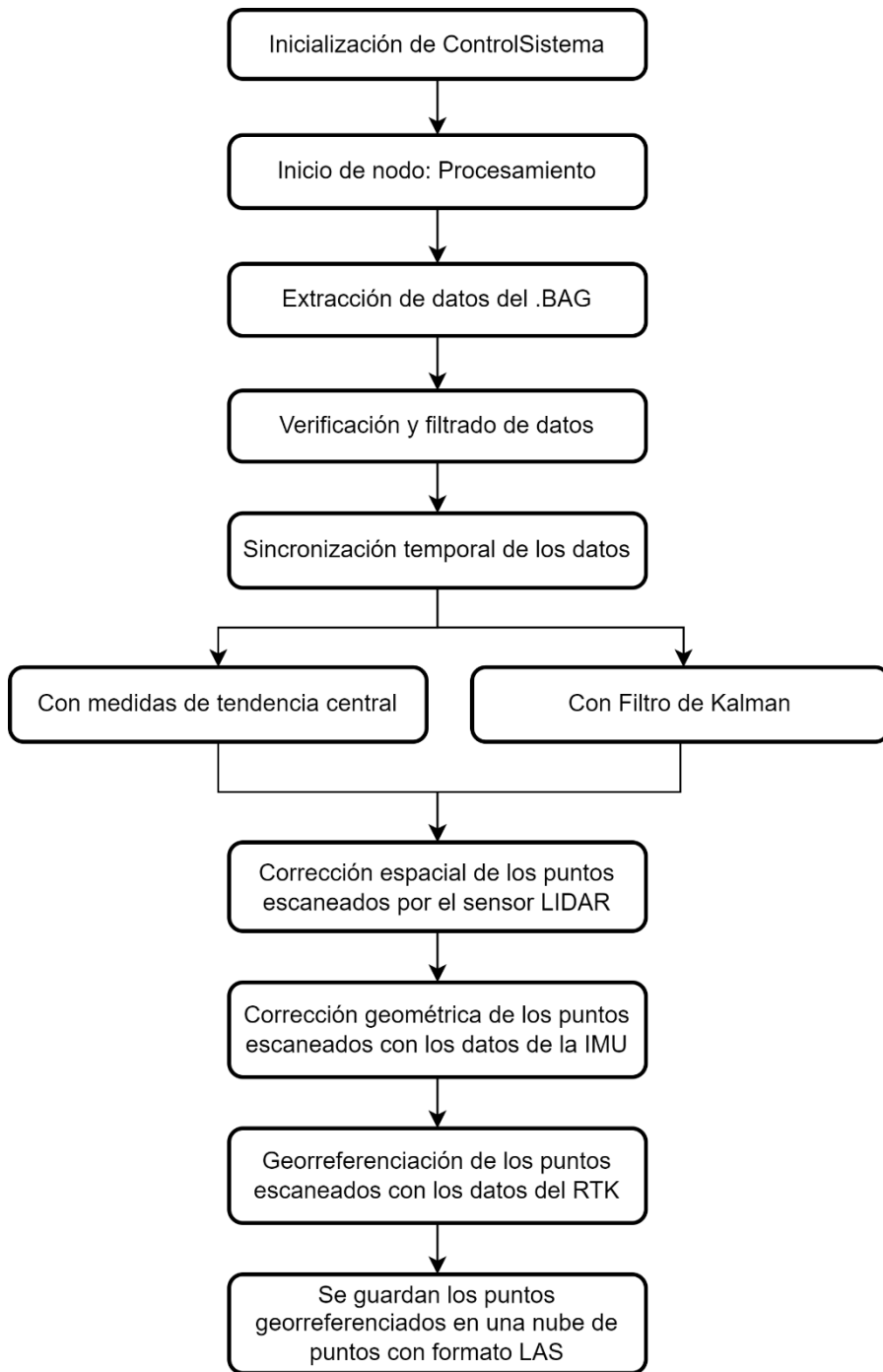


Figura 2.6: Diagrama de flujo del módulo de procesamiento [Autoría propia]

2.3.5 Frecuencias independientes de captura de datos

Cada sensor maneja frecuencias diferentes, el enfoque de manejar una sola frecuencia de captura para todo el sistema implicaría trabajar a la frecuencia del sensor con menor frecuencia de captura, y esto limitaría la capacidad de captura de datos de los demás sensores que trabajan a una frecuencia mayor.

Por ello se optó por un enfoque de independencia de frecuencias, en donde cada sensor trabaje con una frecuencia parametrizable independiente. Permitiendo así configurar de forma sencilla la frecuencia de recolección de cada uno de los sensores, haciendo que cada sensor pueda ser configurado para trabajar a su frecuencia máxima de captura de datos. Al obtener la mayor cantidad de datos posibles en los sensores que presentan alta incertidumbre, tenemos la capacidad de reducir el ruido por medio de técnicas de procesamiento de datos y así mejorar la precisión de los puntos al generar la nube que es la salida del sistema.

Las frecuencias máximas de captura de datos dependen de las especificaciones de cada sensor. En la *Tabla 2.3* se muestra la frecuencia máxima de recolección de los sensores utilizados.

Tabla 2.3: Frecuencia máxima de captura de datos de los sensores utilizados

Nombre	Frecuencia máxima de captura de datos (Hz)
SwiftNav Piksi v2.0	30
Adafruit IMU BNO055	100
Láser Hokuyo UST-10LX	40

CAPÍTULO 3

3. RESULTADOS

3.1 Frecuencia efectiva de los sensores

La arquitectura propuesta tiene como objetivo que cada sensor trabaje en su máxima frecuencia, si bien el fabricante de cada uno de los sensores nos indica una frecuencia máxima a la que este trabaja, en la práctica la frecuencia puede no cumplirse con exactitud, ya sea por limitaciones de hardware o de software. Por ello evaluamos la *frecuencia efectiva* que es la frecuencia promedio obtenida de cada uno de los sensores al ser evaluado independientemente a diferentes frecuencias.

En la *Figura 3.1*: Comparación de frecuencia teórica vs frecuencia máxima efectiva de la IMU nos muestra que a pesar de que el fabricante de la IMU nos indica que el sensor trabaja a una frecuencia máxima de 100Hz, en las pruebas se encontró que en promedio trabaja a una frecuencia máxima de 90 Hz, es decir que se guardan en promedio 90 registros de la IMU por segundo.

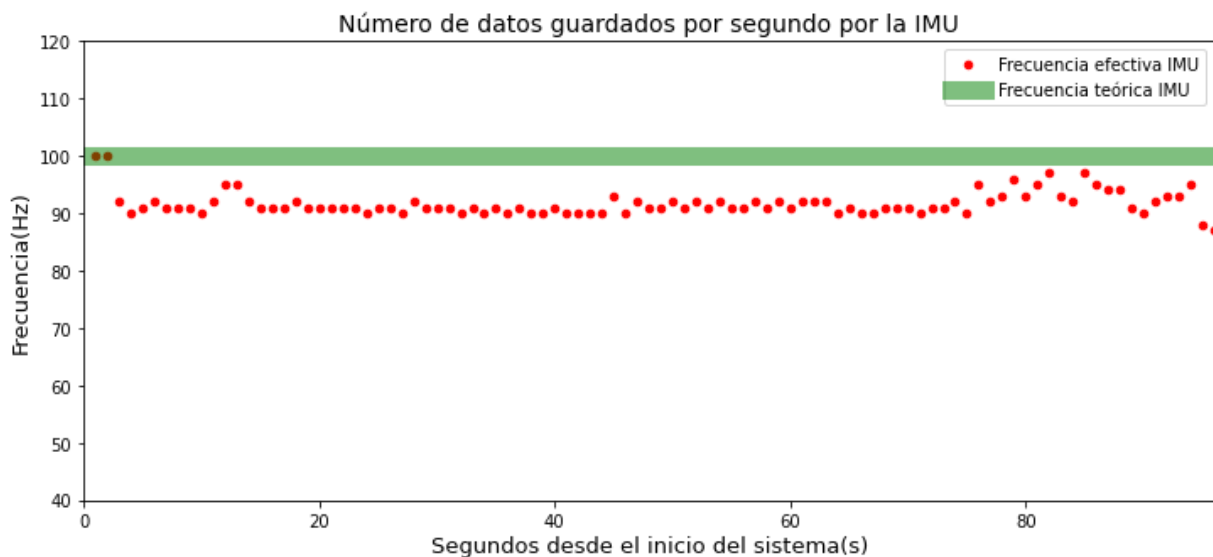


Figura 3.1: Comparación de frecuencia teórica vs frecuencia máxima efectiva de la IMU
[Autoría propia]

Mientras que en las pruebas realizadas del láser y del RTK, la frecuencia efectiva de estos es igual a la frecuencia teórica del fabricante. Tal y como se observa en *Figura 3.2* y *Figura 3.3*.

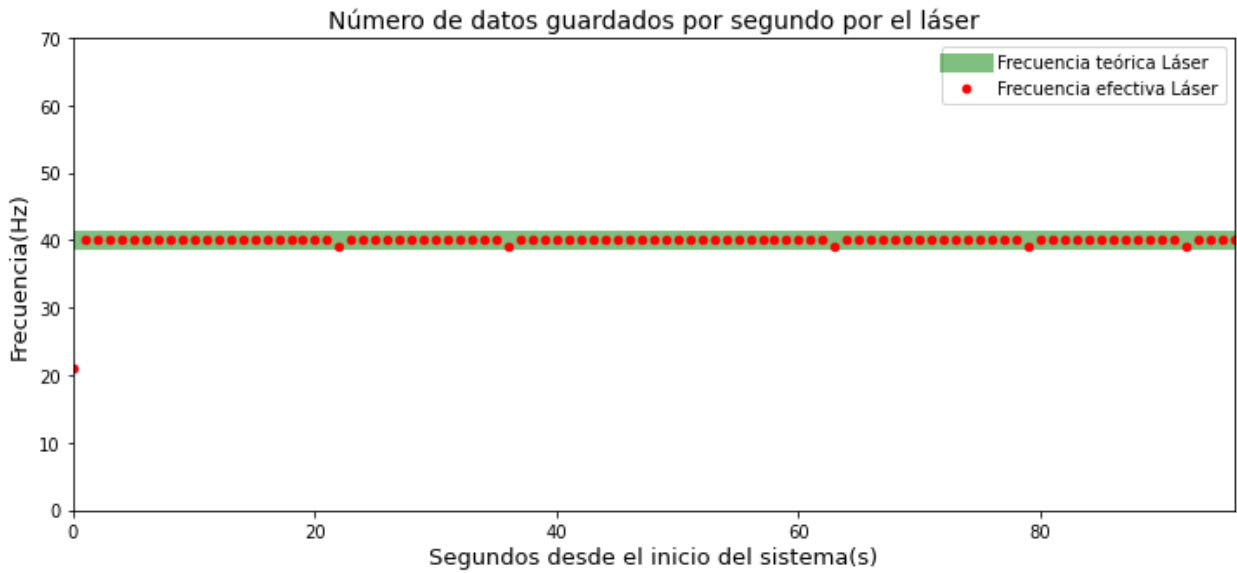


Figura 3.2: Comparación de frecuencia teórica vs frecuencia máxima efectiva del láser
[Autoría propia]

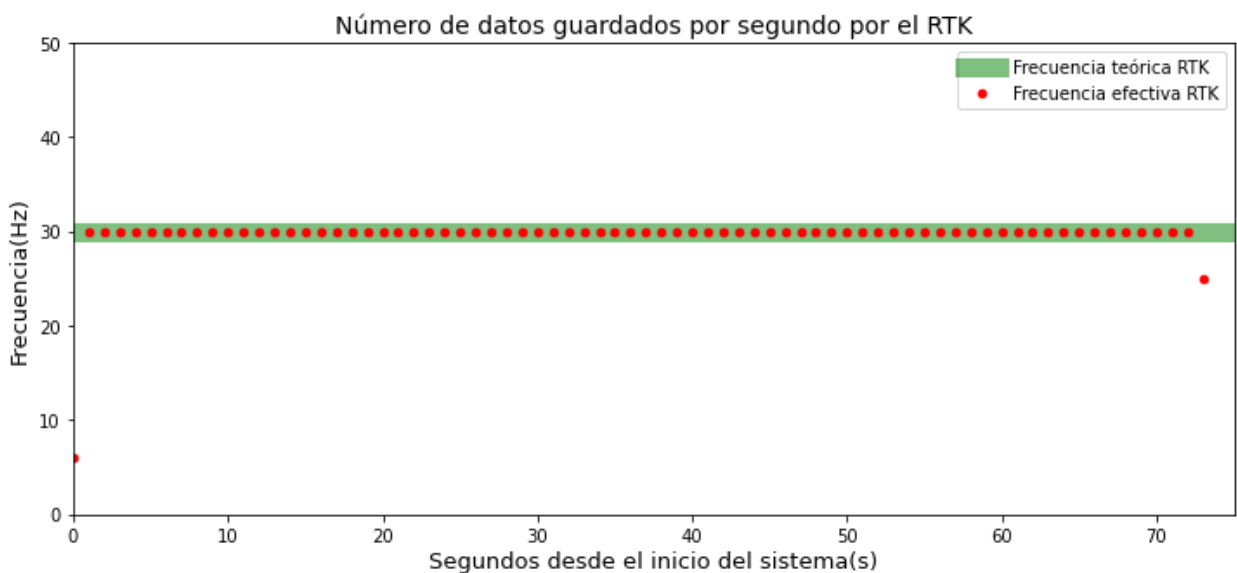


Figura 3.3: Comparación de frecuencia teórica vs frecuencia máxima efectiva del RTK
[Autoría propia]

Es así como se establecieron los valores de frecuencia en la que trabajará cada sensor independientemente de acuerdo con la *Tabla 3.1: Frecuencia Teórica vs Frecuencia Efectiva*. Tomamos la frecuencia efectiva máxima para configurar cada uno de los nodos correspondientes a cada sensor.

Tabla 3.1: Frecuencia Teórica vs Frecuencia Efectiva

Nombre	Frecuencia Máxima Teórica (Hz)	Frecuencia efectiva (Hz)
SwiftNav Piksi v2.0	30	30
Adafruit IMU BNO055	100	90
Láser Hokuyo UST-10LX	40	40

3.2 Funcionamiento de los sensores a frecuencias independientes

Realizando los cambios de arquitectura propuestos en el Capítulo 2, cada sensor captura y guarda datos a su propia frecuencia. Una vez configurada la frecuencia máxima efectiva en cada sensor, se logró observar en las pruebas que con éxito cada uno de los sensores guarda registros acordes a su frecuencia máxima efectiva, tal y como se muestra en la *Figura 3.4: Frecuencia efectiva de los 3 sensores*.

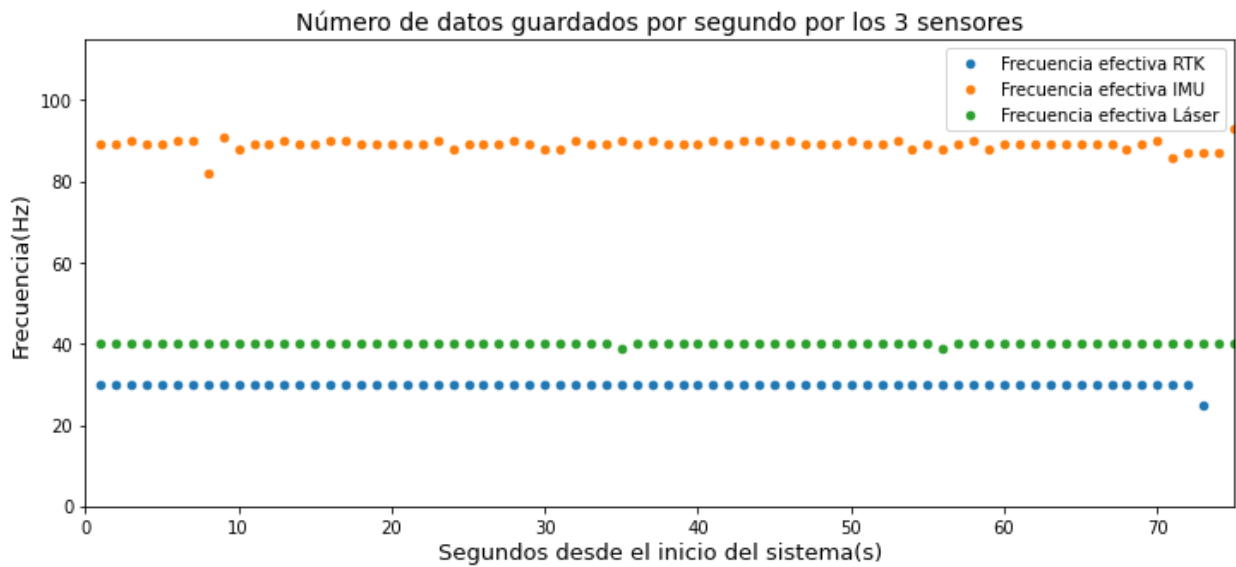


Figura 3.4: Frecuencia efectiva de los 3 sensores [Autoría propia]

3.3 Sincronización de datos

Un paso previo a la generación de la nube de puntos es la sincronización de los datos. Ya que los datos se encuentran a diferentes frecuencias es importante generar una correspondencia temporal entre los datos de los diferentes sensores.

Para ello se decidió trabajar con la menor frecuencia de todos los sensores que es de 30Hz correspondientes al RTK. Para ello los datos del láser y de la IMU fueron agrupados y promediados para que todos sean llevados a 30 Hz.

Esta parte del procesamiento se realizó en el lenguaje de programación Python, con las librerías Pandas y Numpy. En las *Figura 3.5*, *Figura 3.6* y *Figura 3.7* se presentan extractos de los dataset generados por cada sensor.

```
In [4]: laser.head()
```

```
Out[4]:
```

	%time	field.header.seq	field.header.stamp	field.header.frame_id	field.angle_min	field.angle_max	field.angle_increment	field.time_increment
0	1671722208445927192	152	1671722208396072244	laser	-2.356194	2.356194	0.004363	0.000017
1	1671722208470759536	153	1671722208432668338	laser	-2.356194	2.356194	0.004363	0.000017
2	1671722208496619536	154	1671722208462354119	laser	-2.356194	2.356194	0.004363	0.000017
3	1671722208520645630	155	1671722208495671723	laser	-2.356194	2.356194	0.004363	0.000017
4	1671722208545711255	156	1671722208495671723	laser	-2.356194	2.356194	0.004363	0.000017

5 rows × 1092 columns

Figura 3.5: Extracto de un dataset generado por el sensor LiDAR [Autoría propia]

```
In [8]: imu.tail()
```

```
Out[8]:
```

	%time	field.header.seq	field.header.stamp	field.header.frame_id	field.roll	field.pitch	field.yaw
4140	1671722253238493477	4144	1671722252295543738	/laser	-5.75	7.375	0.1875
4141	1671722253250389571	4145	1671722252295543738	/laser	-5.75	7.375	0.1875
4142	1671722253262183894	4146	1671722252295543738	/laser	-5.75	7.375	0.1875
4143	1671722253274107071	4147	1671722252295543738	/laser	-5.75	7.375	0.1875
4144	1671722253283949310	4148	1671722252295543738	/laser	-5.75	7.375	0.1875

Figura 3.6: Extracto de un dataset generado por el sensor IMU [Autoría propia]

```
In [5]: rtk.head()
```

```
Out[5]:
```

	%time	field.header.seq	field.header.stamp	field.header.frame_id	field.tow	field.north	field.east	field.down	field.lat	field.lot	field
0	1671722210997328962	4	1671722210994702660	NaN	400638700	-14.671	-17.203	1.401	-2.145452	-79.966437	83.40
1	1671722211030625004	5	1671722211028026722	NaN	400638700	-14.671	-17.203	1.401	-2.145452	-79.966437	83.40
2	1671722211064009483	6	1671722211061366827	NaN	400638800	-14.671	-17.203	1.401	-2.145434	-79.966310	127.73
3	1671722211097469587	7	1671722211094687920	NaN	400638800	-14.673	-17.204	1.401	-2.145452	-79.966437	83.40
4	1671722211130372504	8	1671722211128024014	NaN	400638800	-14.673	-17.204	1.401	-2.145452	-79.966437	83.40

Figura 3.7: Extracto de un dataset generado por el sensor GPS-RTK [Autoría propia]

Una vez que se genera la correspondencia temporal entre los dataset y se filtran y promedian los datos correspondientes, todos los dataset deben tener la misma cantidad de filas, esta diferencia se puede ver en las *Figura 3.8* y *Figura 3.9*.

```
In [9]: print(f"Dimensiones de datos Laser (antes de sync): ", laser.shape[0], ", ", laser.shape[1])
print(f"Dimensiones de datos RTK (antes de sync): ", rtk.shape[0], ", ", rtk.shape[1])
print(f"Dimensiones de datos IMU (antes de sync): ", imu.shape[0], ", ", imu.shape[1])
```

```
Dimensiones de datos Laser (antes de sync): 1726 , 1092
Dimensiones de datos RTK (antes de sync): 1230 , 14
Dimensiones de datos IMU (antes de sync): 4145 , 7
```

Figura 3.8: Dimensiones de los dataset ejemplo previo a la sincronización de los datos
[Autoría propia]

```
In [12]: print(f"Dimensiones de datos Laser (después de sync): ", new_laser.shape[0], ", ", new_laser.shape[1])
print(f"Dimensiones de datos RTK (después de sync): ", new_rtk.shape[0], ", ", new_rtk.shape[1])
print(f"Dimensiones de datos IMU (después de sync): ", new_imu.shape[0], ", ", new_imu.shape[1])
```

```
Dimensiones de datos Laser (después de sync): 1229 , 1091
Dimensiones de datos RTK (después de sync): 1229 , 14
Dimensiones de datos IMU (después de sync): 1229 , 6
```

Figura 3.9: Dimensiones de los dataset ejemplo luego de la sincronización de los datos
[Autoría propia]

3.4 Generación de nube de puntos con datos sincronizados

Una vez que se obtienen los datos sincronizados de todos los sensores, se utiliza el módulo de procesamiento indicado en la *Figura 2.4*: Diagrama de la arquitectura del Sistema de Escaneo Tridimensional propuesto para generar las nubes de puntos deseadas. Para facilitar el entendimiento de las operaciones geométricas necesarias al proyectar los puntos en un espacio tridimensional georreferenciado se realizan dos proyecciones menos complejas, las cuales pueden resaltar algún error en los cálculos o en los datos.

3.4.1 Proyección Matricial

La primera nube que se genera realiza una proyección matricial de los puntos bajo la asunción de que el escaneo se realizó a velocidad constante en una dirección fija. Esta representación oculta muchos de los detalles que pueden ser encontrados en el escaneo, entre esos la altura; sin embargo, nos permite ver la distribución de los puntos y el relieve o “sombra” de ciertos objetos en un plano, ver *Figura 3.10*.

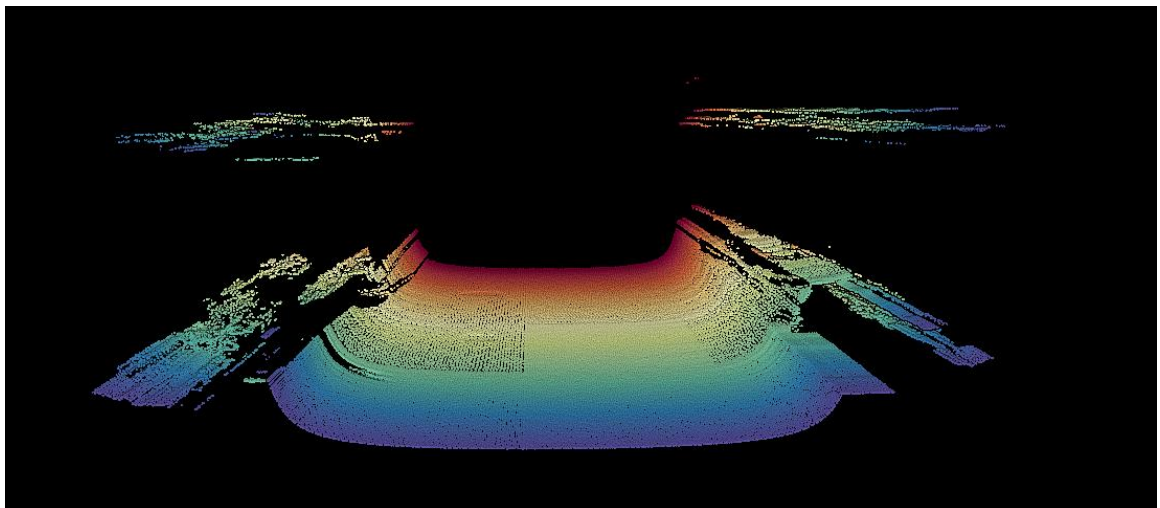


Figura 3.10: Ejemplo de nube de puntos proyectada de forma matricial [Autoría propia]

3.4.2 Proyección Tridimensional. Corrección geométrica.

La segunda nube que se genera utiliza los datos del láser LIDAR para proyectar las distancias que el sensor capta (rangos) en un plano con coordenadas relativas al cabezal del láser. Si el plano relativo L_i de un escaneo i es generado por los ejes (X_i, Y_i) , un punto P perteneciente a ese escaneo tendría coordenadas (x_p, y_p) . Este procedimiento es posible hacerlo para cada escaneo del sensor; sin embargo, un escaneo por sí solo es una proyección bidimensional.

Cuando se quiere generar una nube de puntos, los planos generados por los escaneos consecutivos generan un nuevo eje Z, que es el producto cruz entre los ejes coordenados X y Y de estos. Este eje es el que permite la visualización tridimensional de la nube, como se puede apreciar en la *Figura 3.11*.

En la proyección tridimensional el eje Z es generado bajo la misma asunción de la proyección matricial (el escaneo se realiza a velocidad constante y en una sola dirección Z positivo), por tanto, se generan valores consecutivos positivos para que los escaneos proyectados en el plano relativo al láser tengan un desplazamiento constante y se produzca el espacio tridimensional relativo.

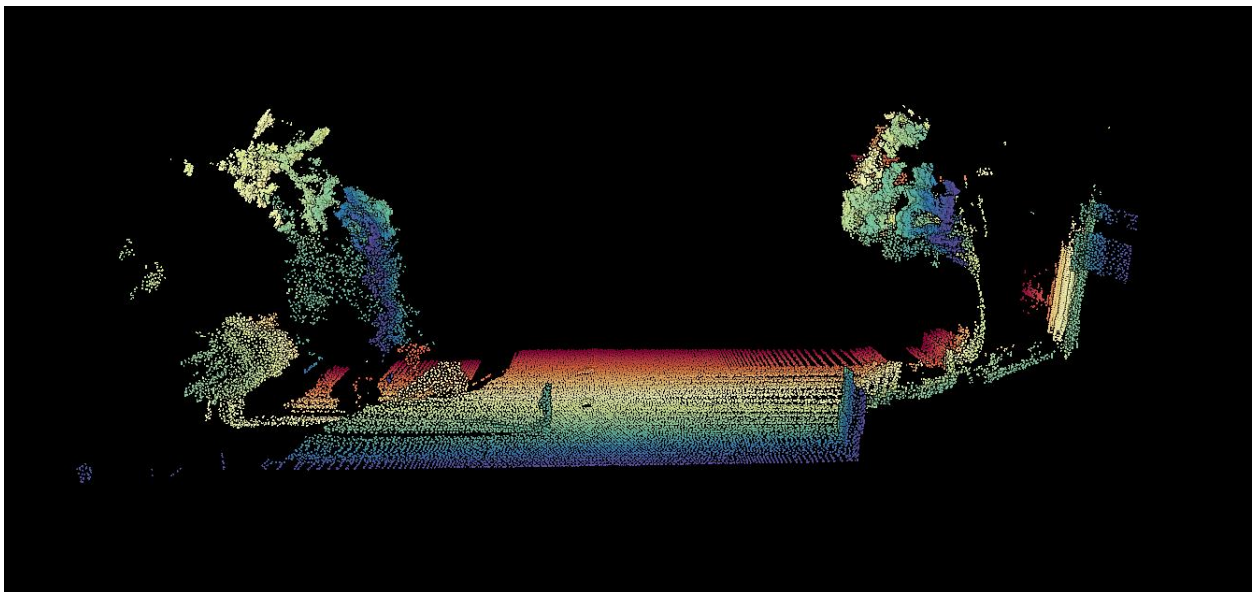


Figura 3.11: Ejemplo de nube de puntos proyectada en el espacio tridimensional relativo al láser [Autoría propia]

3.4.3 Georreferenciación. Corrección geográfica.

Una vez que se consigue la representación geométrica correcta de los puntos en el espacio relativo al láser, se procede a utilizar los datos de los otros sensores: el GPS-RTK y los ángulos de rotación de la IMU, para poder transformar la proyección de los puntos de un espacio vectorial tridimensional relativo a uno absoluto con coordenadas geográficas correctas. A este proceso lo denominamos georreferenciación.

En este proceso se utilizan las coordenadas geográficas UTM: Este, Norte y Altura; provistas por los datos del sensor GPS para identificar la posición absoluta de los puntos de un escaneo con respecto a la Tierra. A diferencia de la generación automática del eje Z de la corrección geométrica, este eje se convierte en la coordenada geográfica absoluta Norte del sistema UTM, que gracias a los datos del GPS puede ser representada sin problemas y con una incertidumbre mínima en las pruebas realizadas, ver *Figura 3.12*.

Cabe recalcar que, las pruebas de campo realizadas para analizar, corregir y verificar el proceso de georreferenciación se hicieron en condiciones controladas, evitando así, el ruido de ciertos datos como el del sensor IMU, que es bastante volátil, y poder conseguir una representación precisa del escaneo. En las *Figura 3.13* y *Figura 3.14* se muestran el sitio de una prueba de escaneo y el resultado correspondiente.

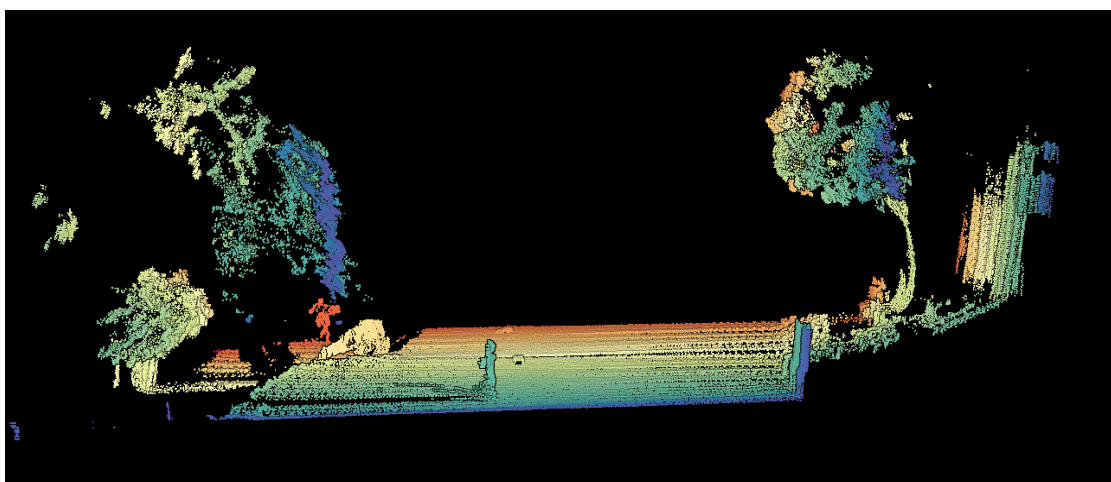


Figura 3.12: Ejemplo de nube de puntos georreferenciada [Autoría propia]



Figura 3.13: Foto del sitio de prueba de escaneo [Autoría propia]

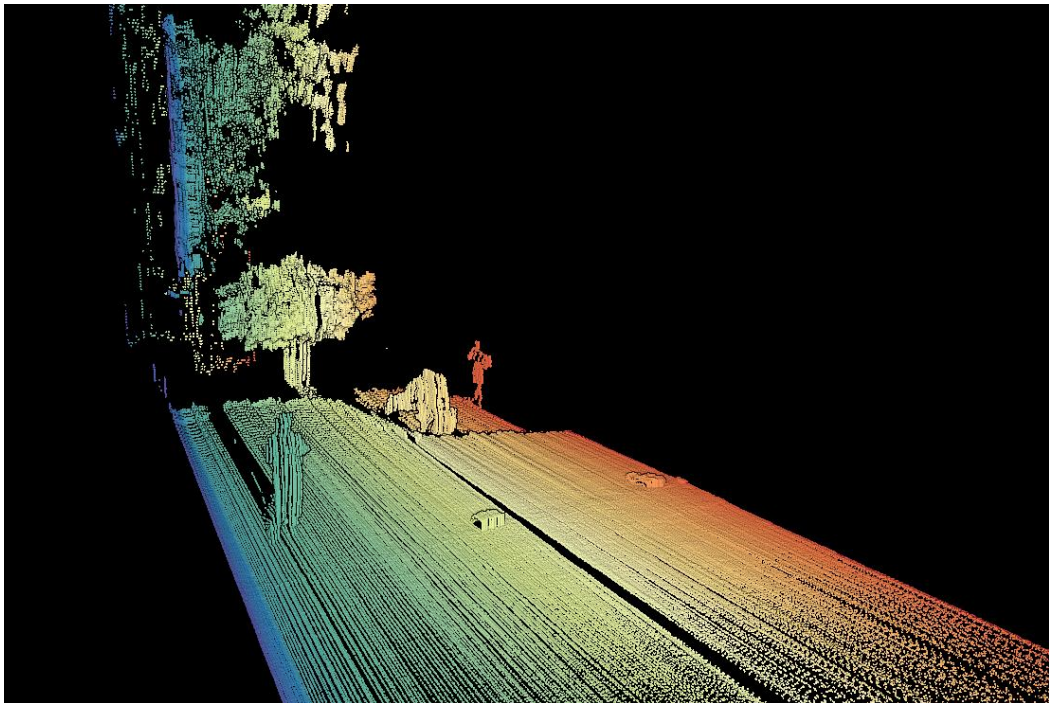


Figura 3.14: Nube de puntos desde la perspectiva correspondiente a la *Figura 3.13* [Autoría propia]

Hasta este punto, las nubes de puntos resultantes cumplen satisfactoriamente con el proceso de georreferenciación; sin embargo, se planearon y realizaron pruebas de campo con diferentes condiciones, de forma que se puedan simular los datos provistos por la IMU: los ángulos formados en los ejes de rotación pitch, roll y yaw del cuerpo, donde roll se alinea con la coordenada absoluta Norte, pitch con la coordenada absoluta Este y yaw se alinea con el eje resultante del producto cruz apuntando hacia Abajo, en la *Figura 3.15* se muestra el diagrama respectivo.

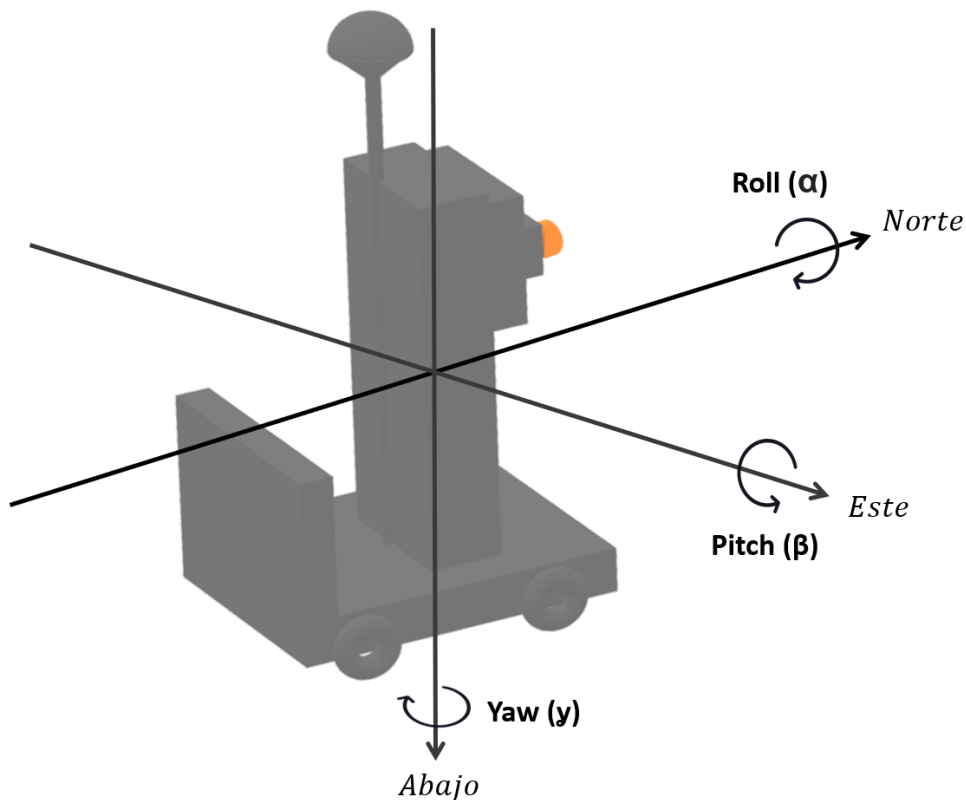


Figura 3.15: Diagrama de los ejes de rotación del prototipo (cuerpo) alineados con las coordenadas Norte, Este y Abajo [Autoría propia]

Para realizar estas simulaciones, se plantean los siguientes estados de prueba de escaneo:

1. *Estado neutral o encerado*: El prototipo, de aquí en adelante referido como *el cuerpo*, no presenta variaciones angulares en ninguno de sus ejes de rotación. Las pruebas de escaneo anteriores se realizaron en este estado, ver *Figura 3.15*.

2. *Estado alpha*: Donde el cuerpo presenta una variación en el eje de rotación roll, el ángulo resultante de la variación será denominado *alpha* (α), ver *Figura 3.16*.

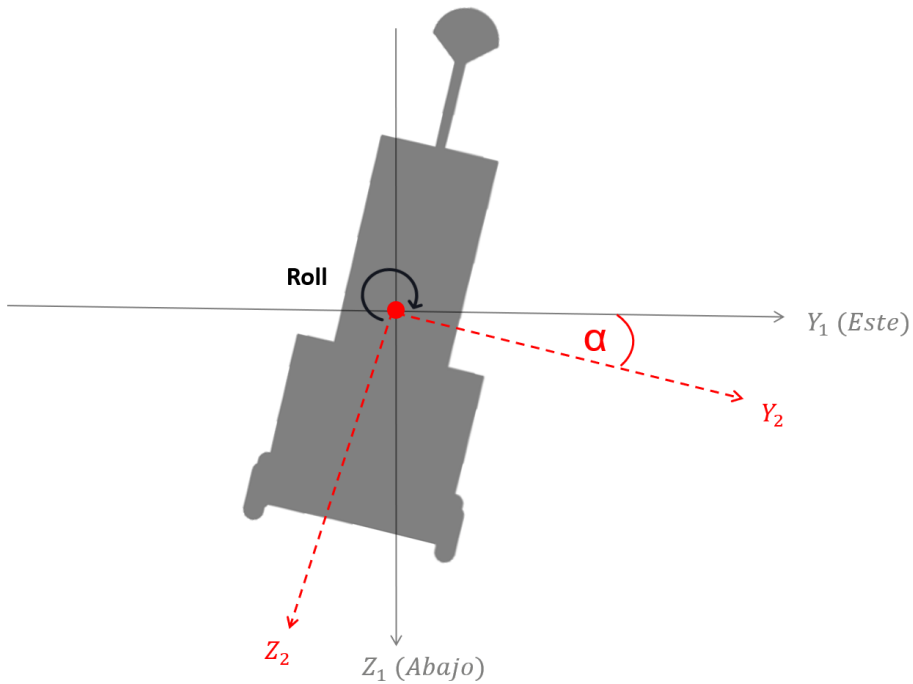


Figura 3.16: Diagrama del cuerpo en estado *alpha* [Autoría propia]

3. *Estado beta*: Donde el cuerpo presenta una variación en el eje de rotación pitch, el ángulo resultante de la variación será denominado *beta* (β), ver *Figura 3.17*.

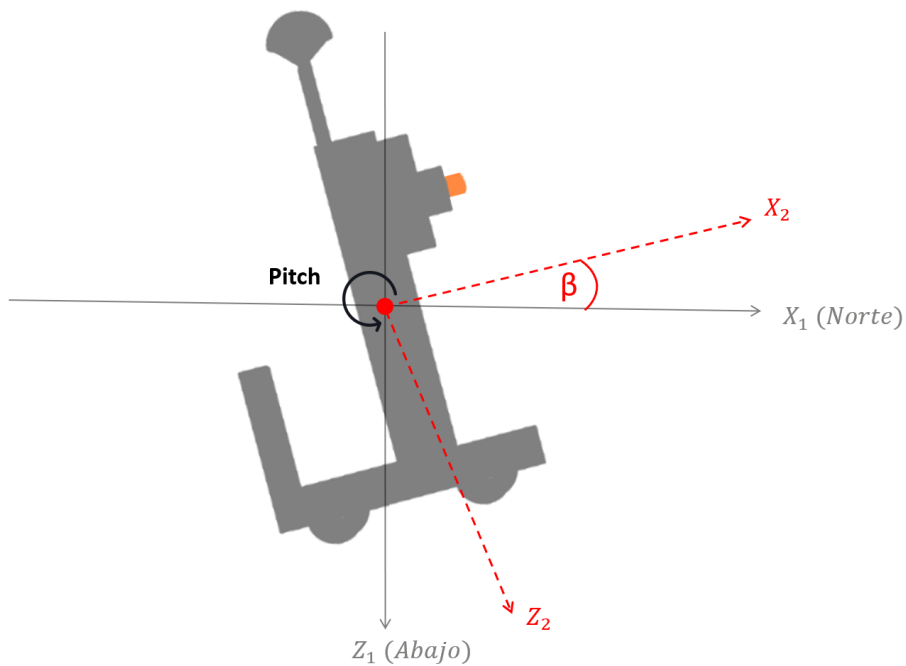


Figura 3.17: Diagrama del cuerpo en estado *beta* [Autoría propia]

4. *Estado gamma*: Donde el cuerpo presenta una variación en el eje de rotación yaw, el ángulo resultante de la variación será denominado *gamma* (γ), ver *Figura 3.18*.

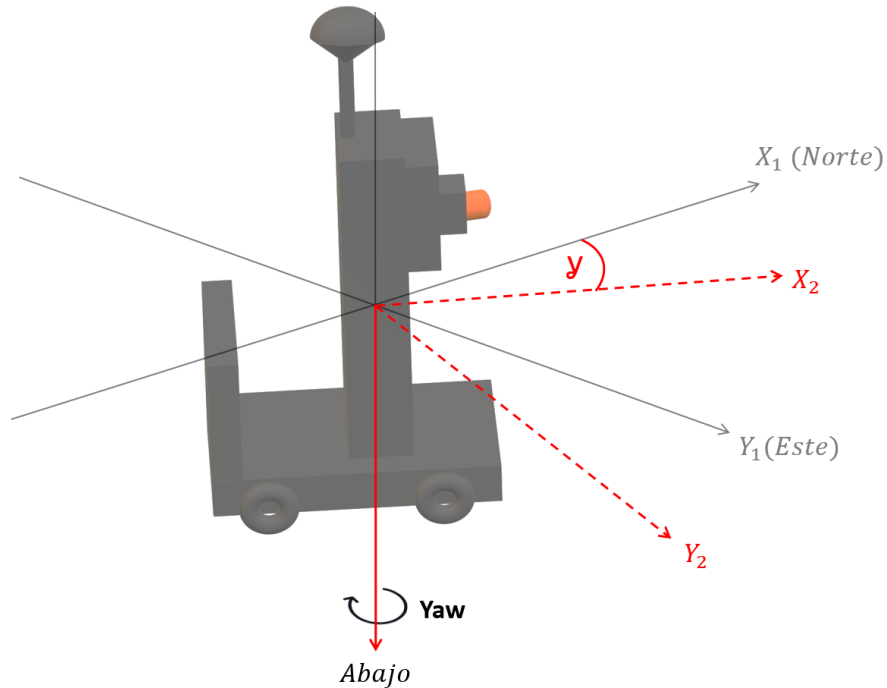


Figura 3.18: Diagrama del cuerpo en estado *gamma* [Autoría propia]

5. *Estado alterado*: Donde el cuerpo presenta variaciones angulares en todos sus ejes de rotación, ver *Figura 3.19*.

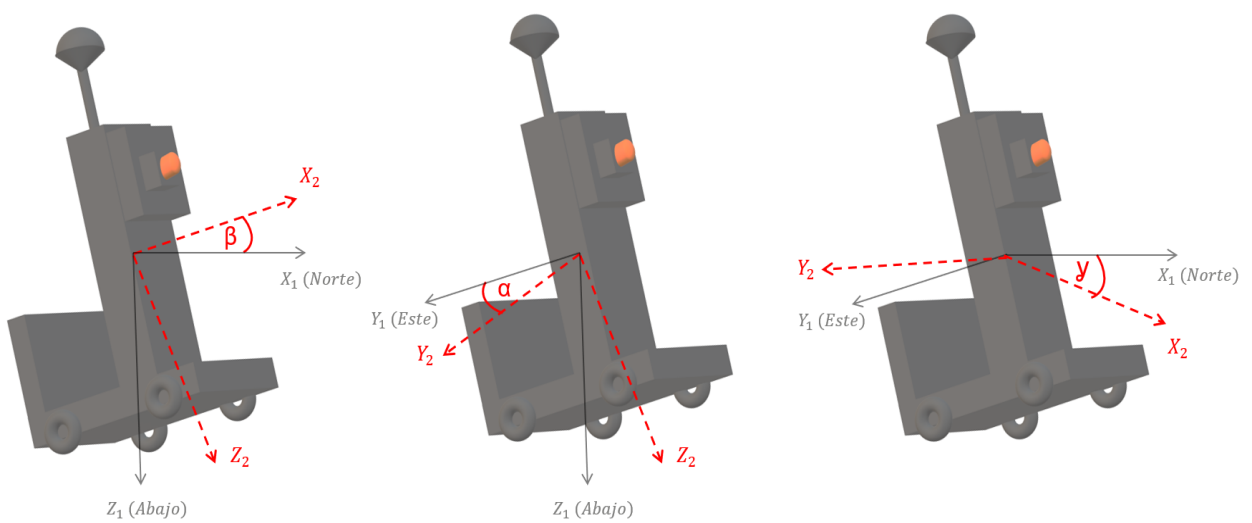


Figura 3.19: Diagrama del cuerpo en estado *alterado*, con todas sus variaciones angulares [Autoría propia]

Las pruebas necesarias para la corroboración de los cálculos involucrados para representar las nubes de puntos correctamente cuando el cuerpo se encuentra en los estados *alpha* y *beta* se realizaron in situ en el CTD, ya que no requerían de corrección geográfica, sino solo de corrección geométrica para ver sus efectos. Los resultados de estas pruebas se aprecian en las *Figura 3.20*, *Figura 3.21* y *Figura 3.22*.

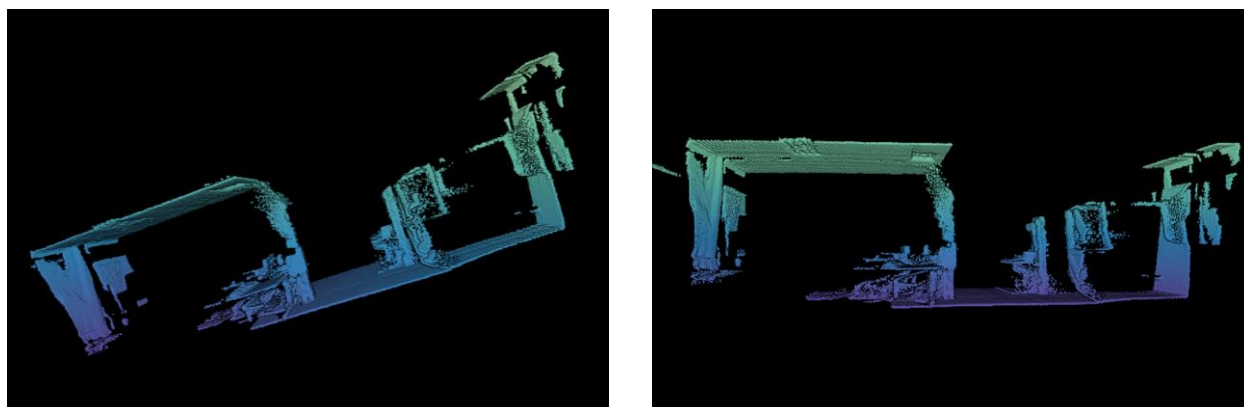


Figura 3.20: Nube de puntos escaneada con el cuerpo en estado *alpha* con un ángulo de 20 grados. Izquierda: Sin corregir. Derecha: Corregido [Autoría propia]

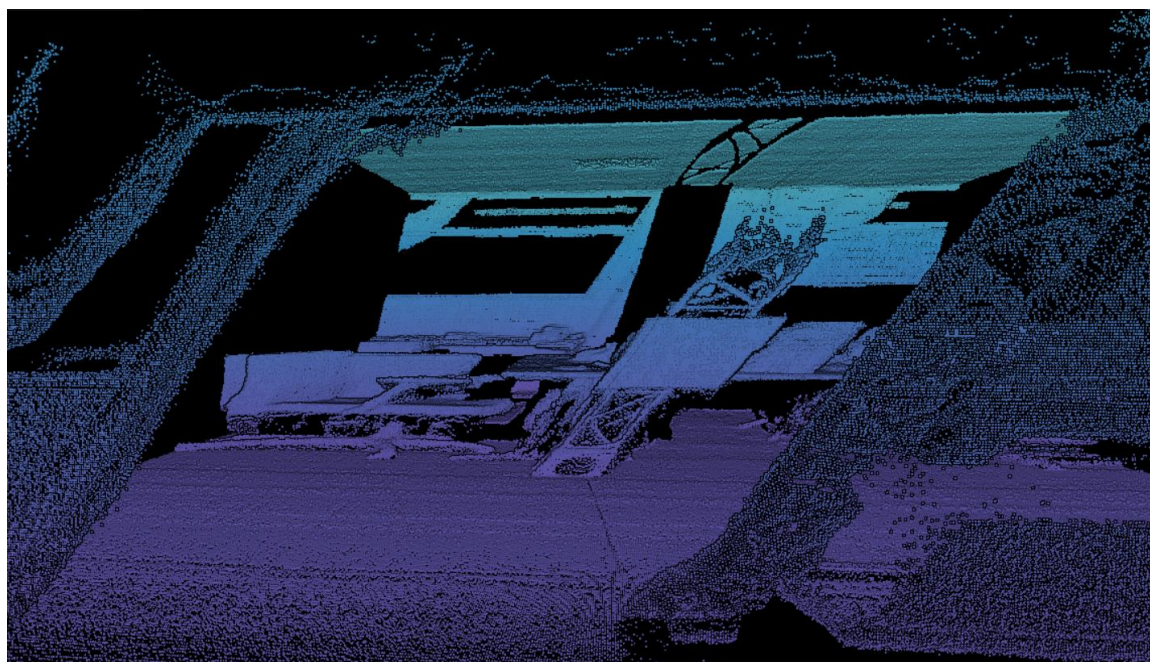


Figura 3.21: Nube de puntos escaneada con el cuerpo en estado *beta* con un ángulo de 15 grados, sin corregir [Autoría propia]

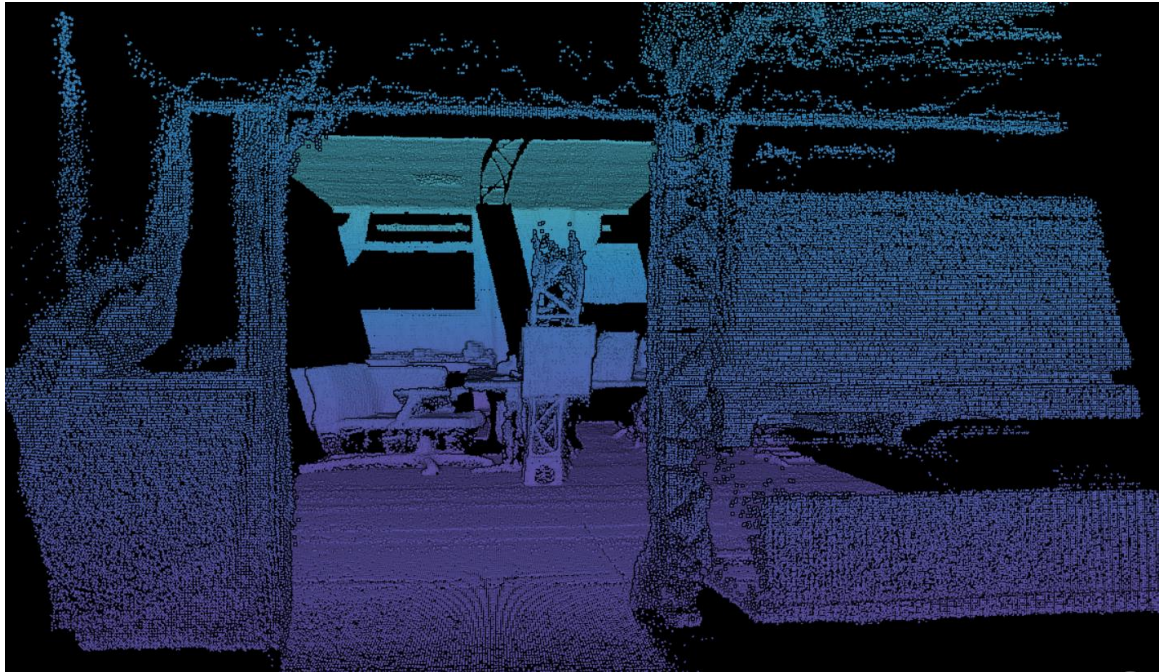


Figura 3.22: Nube de puntos escaneada con el cuerpo en estado *beta* con un ángulo de 15 grados, corregida. [Autoría propia]

Para apreciar los efectos de los estados *gamma* y *alterado* se requiere de realizar la corrección geográfica; por tanto, las pruebas se realizaron en la plazoleta de FIEC con varios objetos con forma identificable para apreciar el efecto de la corrección de estos ángulos.

El estado *gamma* del cuerpo representa una variación en el eje de rotación yaw y este a su vez indica la dirección absoluta que el cuerpo toma con respecto al norte magnético de la Tierra. En las *Figura 3.23* y *Figura 3.24* se experimentó con un ángulo *gamma* de 90 grados, lo cual implica que el cuerpo estaba moviéndose en dirección Este.

Sin las correcciones geométricas y geográficas adecuadas, la nube de puntos puede verse reflejada en la coordenada Este, compactarse de forma que no es fiel al espacio escaneado o presentar alguna otra deformación, como se puede apreciar en la *Figura 3.23*.

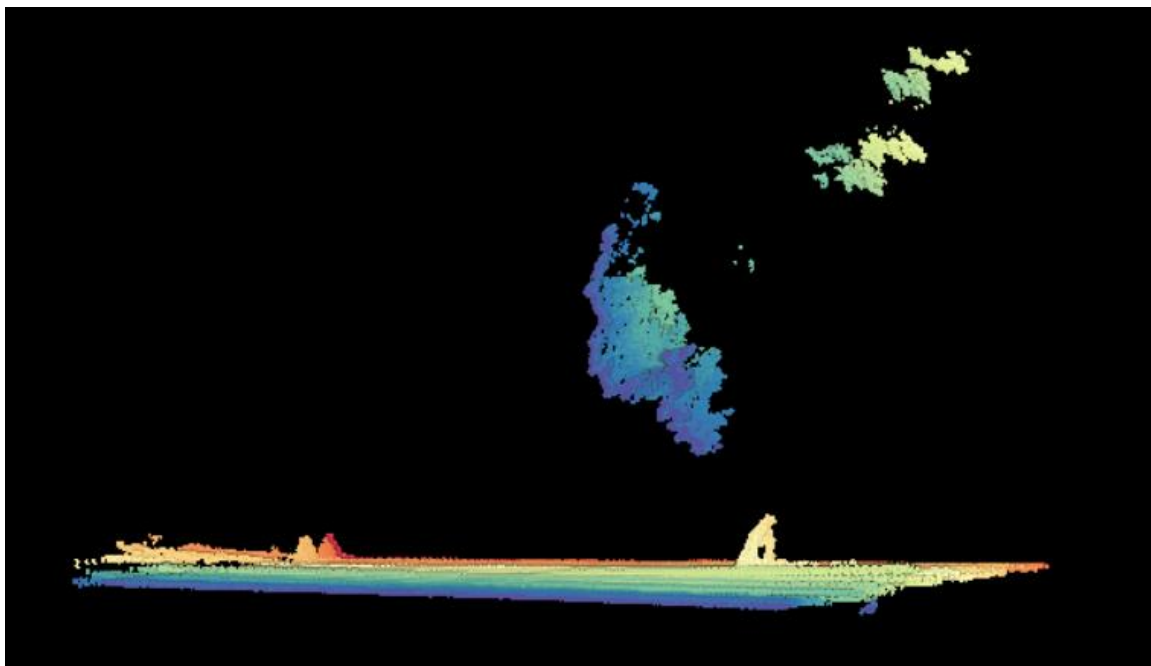


Figura 3.23: Nube de puntos escaneada con el cuerpo en estado *gamma* con un ángulo de 90 grados, sin corrección [Autoría propia]

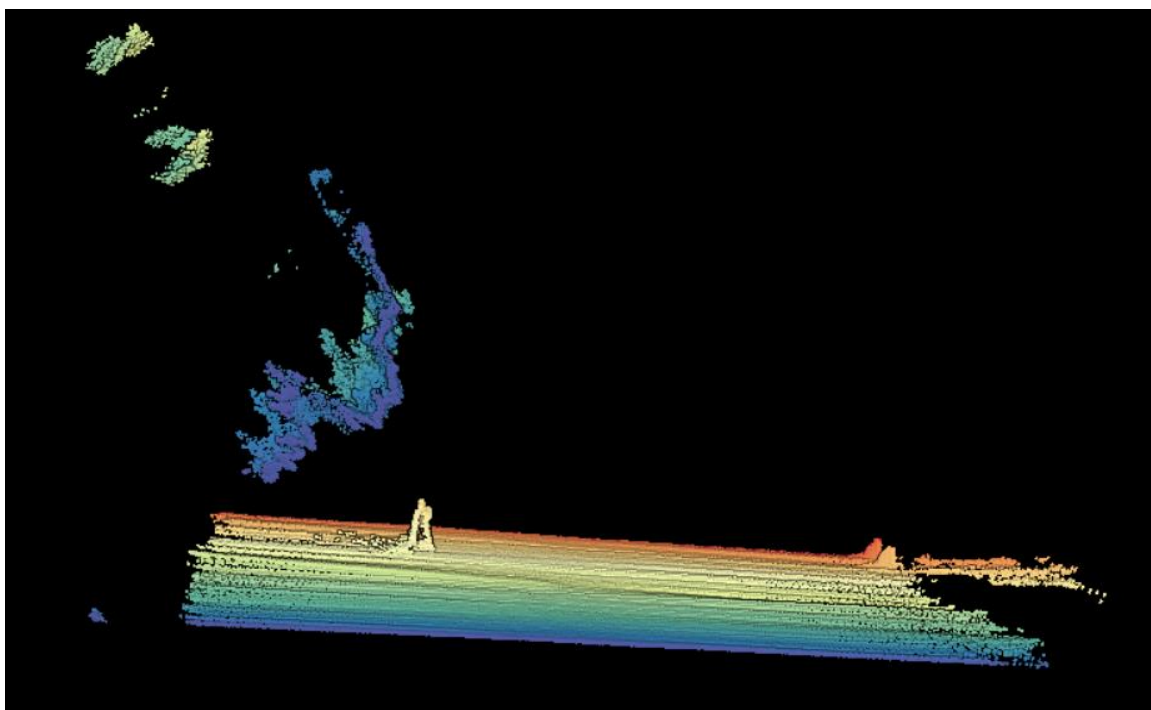


Figura 3.24: Nube de puntos escaneada con el cuerpo en estado *gamma* con un ángulo de 90 grados, corregida [Autoría propia]

Finalmente, se realizó una última prueba con el cuerpo escaneando en *estado alterado* para corroborar que la corrección geométrica se realiza para todos los ejes de rotación y que la posterior georreferenciación es correcta. Para esta prueba se consideraron los ángulos $\alpha = 25^\circ$, $\beta = 20^\circ$ y $\gamma = 15^\circ$, los resultados de pueden apreciar en las *Figura 3.25*, *Figura 3.26* y *Figura 3.27*.

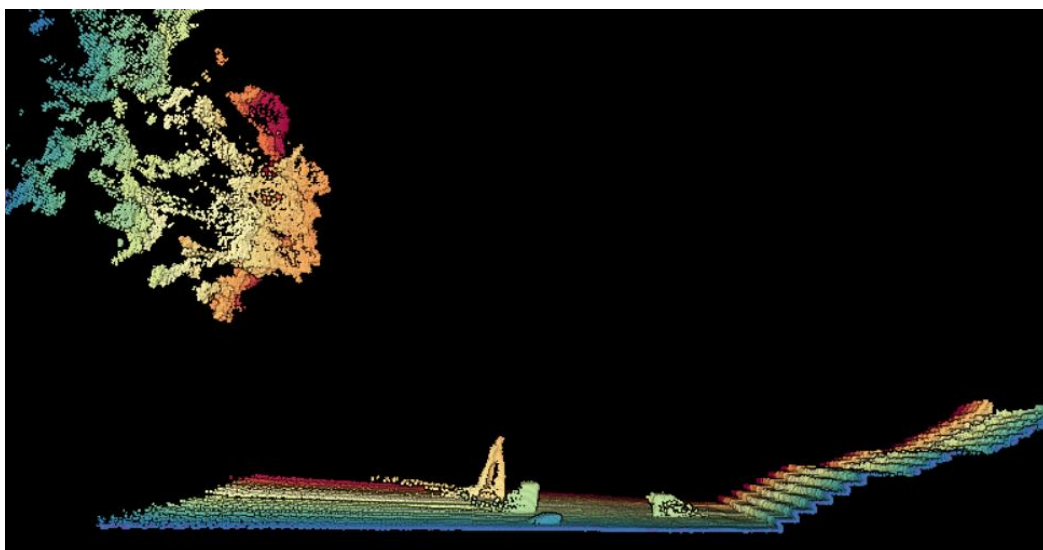


Figura 3.25: Nube de puntos escaneada con el cuerpo en estado *alterado*, corregida
[Autoría propia]

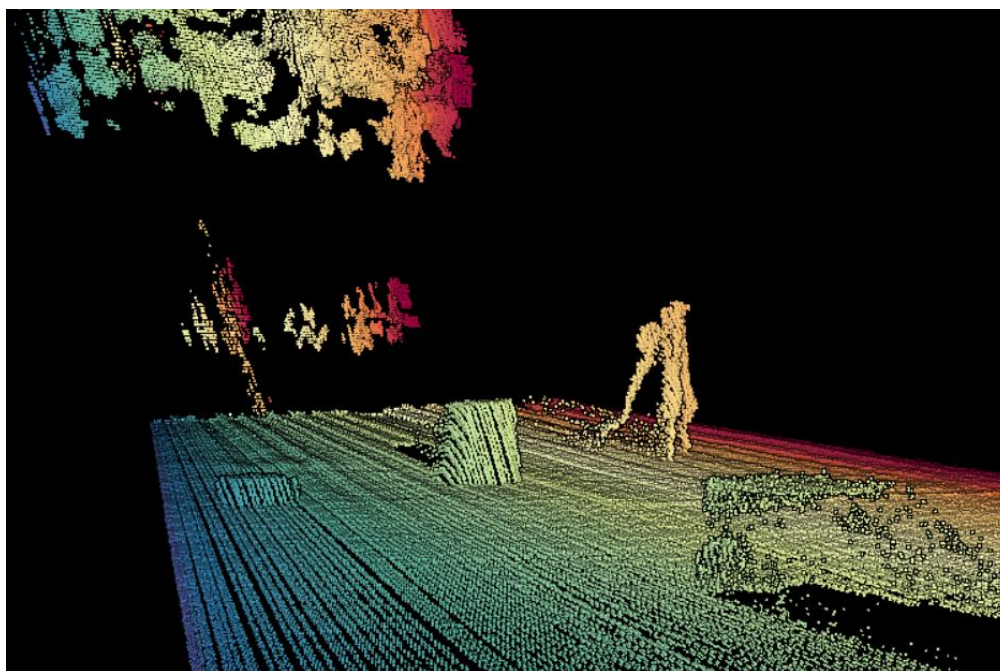


Figura 3.26: Nube de puntos escaneada con el cuerpo en estado *alterado*, acercamiento a los objetos [Autoría propia]

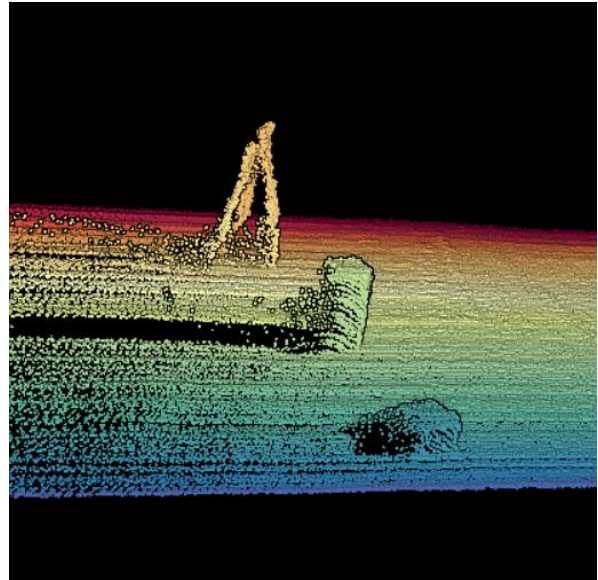


Figura 3.27: Izquierda: Foto del sitio de la prueba con objetos a escanear. Derecha: Perspectiva de la foto en la nube de puntos de la *Figura 3.25* [Autoría propia]

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Si bien los fabricantes de los sensores nos indican la frecuencia a la que estos trabajan, en la práctica puede que esta no se cumpla. Como posibles causas están las limitaciones y fallos de hardware, o limitaciones de software que no permitan aprovechar al máximo las capacidades del hardware.
- Las operaciones matriciales para la conversión de un espacio vectorial tridimensional relativo a uno absoluto evidencian que la georreferenciación de las nubes de puntos es acertada y su incertidumbre recae en la calidad de los datos de los sensores que proveen la información geográfica.
- Las nubes de puntos generadas en este proyecto logran proyectar con precisión los obstáculos colocados durante las pruebas, con esto se demuestra que es factible la utilización de esta tecnología para la detección de obstáculos y de defectos en las calles.
- La mayoría de los sistemas LIDAR que existen en el mercado son montados en vehículos, cuyos movimientos (vibraciones, rotaciones y desplazamientos) pueden ser representados por variaciones angulares en los ejes de rotación del cuerpo, datos que son provistos por el sensor IMU. Con las pruebas planteadas se realizó la verificación de las correcciones geométricas y geográficas de forma satisfactoria, generando así nubes de puntos georreferenciadas de mediana a alta resolución.

4.2 Recomendaciones

- Como metodología para sincronizar la frecuencia de los datos, utilizamos la frecuencia menor de todos los sensores, que en nuestro caso fue la del RTK. Sin embargo, esto provoca que se pierdan datos del láser. Ya que en sistemas LIDAR el láser es el sensor principal, se recomienda para trabajos futuros que los datos sean sincronizados tomando en cuenta la frecuencia del Láser. Y aquellos sensores que posean una frecuencia menor, que se le apliquen técnicas de interpolación de datos.
- Por las limitaciones generales de la tecnología GPS, se aconseja la introducción de un odómetro como fuente de datos y la implementación de algoritmos de corrección geográfica para no depender únicamente de las lecturas provistas por la técnica RTK, ampliando así el alcance del sistema e incluso el montaje de este en un vehículo.
- El proyecto se llevó a cabo utilizando como sistema embebido una Raspberry PI 3 con 1 Gb de RAM, y una memoria SD de 16 GB, que son los componentes con los que se contaba en el CTD. Sin embargo, actualmente existen mejores alternativas en el mercado que aseguran un mejor rendimiento. Permitiendo llevar a cabo el preprocesamiento de la nube de puntos dentro de la Raspberry, además de que, aumentando la capacidad de almacenamiento, se aumenta la capacidad de escanear por tiempos más prolongados.

BIBLIOGRAFÍA

[1] GAD Municipal de Guayaquil. (2022). Dirección de Obras Públicas [Online]. Disponible en: <https://www.guayaquil.gob.ec/direccion-de-obras-publicas/>

[2] Bengtsson, M., & Lindkvist, L. (2017). Mapping the Road to Future Projects: Roadmapping as a Balancing and Transformation Process. *Project Management Journal*, 48(2), 39–54. doi:10.1177/875697281704800204

[3] Leiton y Poveda "Calibración y generación de nubes de puntos georreferenciadas de calles". ESPOL, Guayaquil, Ecuador, 2020.

[4] J. Salamanca y J. Pérez, "LIDAR, una tecnología de última generación, para planeación y desarrollo urbano," *Revista Científica y Tecnológica de la Facultad de Ingeniería*, vol. 13, no. 1, pp. 67 Enero, 2008.

[5] Federal Aviation Administration (N.A.). Satellite Navigation – Global Positioning System (GPS) [Online]. Disponible en: https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps

[6] NASA. (2012, Oct 27). Global Positioning System History [Online]. Disponible en: https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS_History.html

[7] Global GPS Systems. (N.A.). What is RTK and what does it stand for? [Online]. Disponible en: <https://globalgpsystems.com/gnss/what-is-rtk-and-what-does-it-stand-for/>

[8] Anis Koubaa, "Studies in Computational Intelligence" Robot Operating System, xth ed. Warsaw, Polonia: Academy of Sciences, 2016, pp. 5

[9] Mortan Quingle, Brian Gerkey, "ROS: an open-source Robot Operating System" Robot Operating System. Stanford University, Stanford, CA.

[10] GEONIM. (2018, Nov 14). Geoinn Geoespatial Innovations: Drones Empresariales DJI, Servicios de Fotogrametría digital, imágenes satélites de alta resolución para Centroamérica y el Caribe [Online]. Disponible en: <https://geoinn.com/nube-de-puntos/>.

[11] Raspberry Pi. "Raspberry Pi Documentation". Raspberry Pi [online]. Disponible en: <https://www.raspberrypi.com/documentation/>

[12] Boch, "BNO055 Quick start guide". (2015, Mar) Application note: BNO055 Quick start guide [Online], disponible en: https://drive.google.com/drive/folders/1Fu4SrRjVeMuf_G4HeLU0s8_dvSGO6SvB.

[13] Swift Navigation. "Piksi Multi | Multi-Band, Multi-Constellation RTK GNSS Receiver. Offers Precise Positioning Solutions | GNSS" [Online]. Disponible en: <https://www.swiftnav.com/piksi-multi>.

[14] Ingeniería MCI Ltda. "Tutorial Xbee | XBee.cl - Comunicación Inalámbrica para Tus Proyectos" [Online]. Disponible en <https://xbee.cl/tutorial-xbee/>.

[15] Kamioka Y., "Scanning Laser Range Finder Smart-URG mini UST-10LX-H01 Specification". (2016, Jun 30) Guide: UST-10LX-H01: C-42-04213 [Online], disponible en: <https://drive.google.com/file/d/10tNF4CAVBjJvfSSiCTJLiebD4VucVHZx/view>.