



UNIVERSIDAD
CATÓLICA
DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**FACULTAD DE INFORMÁTICA, CIENCIAS DE LA
COMPUTACIÓN E INNOVACIÓN TECNOLÓGICA**

CARRERA DE INGENIERÍA DE SOFTWARE

**Desarrollo de sistema de interacción basado en sensor de distancia en 360°
para superficies interactivas dentro de un laboratorio inmersivo**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO DE SOFTWARE**

AUTOR: MATEO JOSUÉ VERDUGO HUIRACocha

DIRECTOR: ING. JOHN LUIS ESTRADA GUAYTA M.S.

CUENCA - ECUADOR

2026

DIOS, PATRIA, CULTURA Y DESARROLLO



UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**FACULTAD DE INFORMATICA,
CIENCIAS DE LA COMPUTACION E
INNOVACION TECNOLOGICA
CARRERA DE INGENIERÍA DE SOFTWARE**

Desarrollo de sistema de interacción basado en sensor de distancia en
360° para superficies interactivas dentro de un laboratorio inmersivo

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TITULO DE INGENIERO DE SOFTWARE**

AUTOR: MATEO JOSUE VERDUGO HUIRACOA

DIRECTOR: ING. JOHN LUIS ESTRADA GUAYTA M.S.

CUENCA - ECUADOR

2026

DIOS, PATRIA, CULTURA Y DESARROLLO



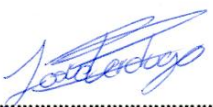
Universidad
Católica
de Cuenca

DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD

Declaratoria de Autoría y Responsabilidad

Mateo Josué Verdugo Huiracocha portador(a) de la cédula de ciudadanía N° **0150023489**. Declaro ser el autor de la obra: **“Desarrollo de sistema de interacción basado en sensor de distancia en 360° para superficies interactivas dentro de un laboratorio inmersivo”**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cuenca, **08 de septiembre de 2026**

F: 

Mateo Josué Verdugo Huiracocha

C.I. 0150023489



CERTIFICADO

Certificado

Yo **Ing. John Luis Estrada Guayta M.S.**, certifico que el presente trabajo de integración fue desarrollado por **Mateo Josué Verdugo Huiracocha** portador(a) de la cedula de ciudadanía N.º **0150023489** con el tema “**Desarrollo de sistema de interacción basado en sensor de distancia en 360º para superficies interactivas dentro de un laboratorio inmersivo**”, bajo mi supervisión.



Validar únicamente en FirmadC.
Firmado electrónicamente por:
**JOHN LUIS ESTRADA
GUAYTA**

Ing. JOHN LUIS ESTRADA GUAYTA

Docente - Tutor

Dedicatoria

Con profundo cariño y gratitud, dedico este trabajo a las personas que han sido fundamentales en mi vida y en el proceso que me ha llevado hasta este momento.

A mi amado padre, quien todos los días sale a trabajar con alegría, sabiendo que todo el esfuerzo que hace es para un bien mayor que es su familia, siempre me demuestra ser un gran ejemplo de persona a seguir para mí y cuando uno se propone una meta donde damos todo de uno mismo no nos para nada para cumplirla, además de eso los valores y evitar los vicios son pilares en mi vida como persona.

A mi madre que, aunque a veces es impaciente esperando que ya me gradué siempre confía en mí, una gran figura de resistencia que me demostró toda su vida, desde que nació donde ella me tenía que llevar al trabajo sin más opciones, levantándose a las 4 de la mañana para ir a tomar el bus en la terminal y viajar otra hora y media hasta su trabajo todo eso mientras tenía que cuidar de mí, además siendo estricta para enseñarme lo que no entendía en la escuela, todo eso se lo agradezco desde lo más profundo de mi corazón.

Un mensaje en conjunto para mis padres es que diario a diario siento que aún les debo tanto, pero créanme que en mente siempre lo tengo presente el mejorar y el regresarle todo multiplicado por dos porque las personas buenas merecen cosas aun mejores.

A mi hermano quiero decirle que lo amo tanto, y es la persona en la que más puedo confiar como mi mejor amigo, el que mejor me conoce, hemos pasado toda una vida juntos, aunque aún somos tan jóvenes, espero que tome mis virtudes como ejemplo, ante todo, siempre le aconsejo para que evite los errores que recorrí, yo sé que él puede ser mejor.

Agradecimiento

El haber culminado otra etapa importante en mi vida es un gran orgullo el que me llena el corazón, el agradecimiento es infinito hacia mis padres ya que gracias a ellos estoy aquí, ellos están para ayudarme a superar momentos difíciles en especial como fue la pandemia en su momento, así que todo lo que soy se lo debo a ellos.

Quiero agradecer a mi tutor ya que el me acepto en su área para lograr tener un tema de tesis acorde a mis intereses, me guio y apoyo en todo lo pudo, siempre mostrando interés y preocupación siempre por mí, motivos que me dieron fuerza para avanzar a pesar de tener malos momentos en este camino, una mención al profesor Bryam Vega que dejo unas palabras de aliento tan marcadas en mi mente que me hacen agradecerle tanto sus conocimientos como su forma de empujarnos a terminar la carrera, agradezco a las autoridades de la universidad quienes me mostraron su compromiso con los estudiantes, su forma de evolucionar en sus puestos para lograr mejorar aún más la calidad de nuestra facultad.

Resumen

El presente trabajo tuvo como objetivo desarrollar un prototipo funcional de interacción espacial para convertir superficies físicas del laboratorio de realidad virtual de la Universidad Católica de Cuenca en interfaces digitales interactivas mediante la integración de un sensor láser RPLiDAR A2 de 360° y el motor Unity. La investigación respondió a las limitaciones de los entornos académicos basados exclusivamente en teclado y mouse, los cuales restringen las posibilidades de interacción en espacios inmersivos. Se aplicó una metodología cuantitativa, de carácter aplicado e iterativo. El componente cuantitativo se orientó a la recopilación de métricas técnicas relacionadas con precisión, seguimiento y respuesta del sistema, mientras que el componente cualitativo se enfocó en la valoración de la experiencia de usuario. Como resultado, se construyó un prototipo capaz de detectar y seguir en tiempo real el movimiento de una mano frente a una superficie física, permitiendo su correspondencia con acciones dentro de una aplicación desarrollada en Unity. Las pruebas funcionales evidenciaron una respuesta estable y una interacción fluida; con una precisión espacial de ± 11.30 mm, una latencia de cálculo de 38.40 ms, una reducción de temblor (jitter) del 78.20% y una recuperación automática ante desconexión serie en 3.24 s. Se concluye que la integración entre RPLiDAR A2 y Unity constituye una alternativa técnica viable para ampliar las formas de interacción en laboratorios inmersivos y para futuras aplicaciones educativas basadas en superficies interactivas.

Palabras clave: *RPLiDAR A2, interacción espacial, superficies interactivas, Unity, entornos digitales.*

Abstract

This study aimed to develop a functional spatial interaction prototype that converts physical surfaces in the Virtual Reality Laboratory at the Catholic University of Cuenca into interactive digital interfaces by integrating a 360° RPLiDAR A2 laser sensor with the Unity engine. The research addressed the limitations of academic environments that rely exclusively on keyboard and mouse input, which restrict interaction possibilities in immersive spaces. A quantitative -methods, applied, and iterative methodology was employed. The quantitative component focused on collecting technical metrics related to system accuracy, tracking, and responsiveness, whereas the qualitative component focused on evaluating the user experience. As a result, a functional prototype was developed that can detect and track the movement of a hand in real time in front of a physical surface, enabling its correspondence with actions within an application developed in Unity. Functional testing demonstrated a stable response and smooth interaction; with a spatial accuracy of ± 11.30 mm, a computation latency of 38.40 ms, a reduction in jitter of 78.20% and automatic recovery following a serial disconnection in 3.24 s. It is concluded that the integration of the RPLiDAR A2 sensor with Unity constitutes a technically viable alternative for expanding forms of interaction in immersive laboratories and for future educational applications based on interactive surfaces.

Keywords: *RPLiDAR A2, spatial interaction, interactive surfaces, Unity, digital environments.*

Índice

Declaratoria de Autoría y Responsabilidad.....	II
Certificado	III
Dedicatoria.....	IV
Agradecimiento.....	V
Resumen.....	VI
Abstract.....	VII
Índice de tablas	XI
Índice de ilustraciones	XII
Introducción	1
Capítulo I.....	3
1. Marco Referencial.....	3
1.1. Planteamiento del Problema	3
1.1.1. Formulación del Problema.....	6
1.2. Justificación de la investigación	6
1.3. Objetivos	8
1.3.1. Objetivo General	8
1.3.2. Objetivo Especifico.....	8
1.4. Limitaciones.....	9
1.5. Delimitaciones.....	10
Capítulo II.....	12
2. Marco Teórico.....	12
2.1. Antecedentes de Investigación.....	12
2.2. Bases Teóricas.....	14
2.2.1. Interacción Humano-Computador (HCI)	14
2.2.2. Evolución de las interfaces de usuario: De WIMP A NUI.....	15
2.2.3. Interacción sin contacto (Touchless Interaction) y presencia espacial	16
2.2.4. Usabilidad, transparencia tecnológica y carga cognitiva	17
2.2.5. Transformación de coordenadas polares a cartesianas e interpolación a pantalla.....	17
2.2.6. Algoritmos de filtrado espacial, zona muerta posterior y región de interés (ROI).....	18
2.2.7. Agregación espacial de lecturas mediante estimación de centroide	19
2.2.8. Eliminación de temblores y Suavizado Exponencial (Jitter Reduction).....	20
2.2.9. Autómata de estados para selección por permanencia (Dwell-Time) y enfriamiento (Cooldown).....	20
2.3. Bases Tecnológicas.....	21
2.3.1. Principios de operación del sensor LiDAR 2D (RPLiDAR A2).....	21
2.3.2. Principio de Triangulación Óptica y Tecnología OPTMAG	22
2.3.3. Formato de datos en nubes de puntos 2D y coordenadas polares	24

2.3.4.	Entornos virtuales e inmersivos en laboratorios de aprendizaje	25
2.3.5.	Integración del motor gráfico Unity 3D y comunicación serie de alto rendimiento	25
2.4.	Arquitecturas, modelos y estándares	26
2.4.1.	Arquitectura desacoplada en tres capas funcionales	26
2.4.2.	Comunicación serie de alta velocidad (256000 baudios) y gestión de búfer	26
2.4.3.	Estándares de seguridad óptica y radiación láser	26
2.5.	Herramientas de desarrollo.....	27
2.5.1.	Componentes de Hardware y Entorno de Desarrollo	27
2.5.2.	Protocolo de Comunicación Serie y Tratamiento de Trama del RPLiDAR A2	27
2.5.3.	Herramienta de Diagnóstico y Validación de Hardware (Slamtec Frame Grabber)	28
2.6.	Marco Conceptual	32
2.6.1.	Middleware de automatización (PyAutoGUI) y resiliencia ante fallos de hardware.....	34
Capítulo III		36
3.	Marco Metodológico	36
3.1.	Metodología de investigación	36
3.2.	Nivel de investigación	36
3.3.	Población y Muestra	37
3.4.	Técnicas e instrumentos de recolección de información	38
3.4.1.	Observación Técnica Experimental Directa.....	38
3.4.2.	Encuesta Estructurada Cuantitativa	38
3.5.	Metodología de desarrollo de software (Metodología Ágil Kanban)	39
3.5.1.	Principios del Flujo Kanban Aplicados al Proyecto.....	40
3.6.	Arquitectura General del Sistema y Flujo de Datos.....	41
3.6.1.	Adquisición y Conversión Geométrica	42
3.6.2.	Filtrado de Ruido Cercano y Discriminación Angular (Zona Muerta).....	44
3.6.3.	Delimitación de Región de Interés (Bounding Box) e Interpolación a Pantalla	44
3.6.4.	Algoritmos de Estabilización y Control de Selección Touchless.....	45
3.6.4.1.	Justificación y Calibración del Factor de Suavizado ($\alpha = 0.22$).....	45
3.6.4.2.	Filtro de Suavizado Exponencial (Reducción de Jitter).....	45
3.6.4.3.	Lógica de Clic por Permanencia (Dwell-Time) y Tiempo de Enfriamiento.....	46
3.6.5.	Robustez Industrial, Optimización de UI y Tolerancia a Fallos.....	48
3.6.5.1.	Configuración de Baja Latencia para Interfaz	48
3.6.5.2.	Módulo de Reconexión Automática y Limpieza de Búfer	49
Capítulo IV		50
4.	Propuesta	50
4.1.	Descripción general de la propuesta	50
4.1.1.	Título de la Propuesta	50
4.1.2.	Presentación.....	50
4.1.3.	Objetivo de la Propuesta	50

4.1.4.	Justificación	51
4.2.	Diagnóstico de necesidades.....	51
4.2.1.	Diagnóstico del Entorno Físico (Laboratorio VRL).....	51
4.2.2.	Geometría del Plano Infrarrojo y Filtrado de Luz Visible	52
4.3.	Requerimientos de la solución.....	54
4.3.1.	Requisitos Funcionales	54
4.3.2.	Requisitos no Funcionales	56
4.3.3.	Análisis de requerimientos.....	56
4.3.4.	Matriz de Trazabilidad entre Requisitos y Pruebas	57
4.4.	Diseño de arquitectura	59
4.4.1.	Diseño del prototipo de RPLiDAR	59
4.4.2.	Diseño de la Arquitectura del Sistema.....	59
4.5.	Diseño de procesos	60
4.5.1.	Módulos Algorítmicos del Middleware (Python Core)	60
4.5.1.1.	Módulo 1: Adquisición de Datos y Reconexión Automática.....	60
4.5.1.2.	Módulo 2: Filtrado Espacial, ROI e Interpolación Trigonométrica	60
4.5.1.3.	Modulo 3: Suavizado Espacial, Autómata Dwell-Time con Cooldown, Resiliencia y Reconexión Automática	61
4.6.	Diseño de datos o información	61
4.7.	Diseño de interfaces, experiencia o interacción	62
4.8.	Diagramas técnicos o modelos de solución	63
4.9.	Selección y justificación de tecnologías	66
4.10.	Plan de implementación futura	67
4.11.	Plan de pruebas y validación futura	67
4.12.	Validación de la propuesta.....	67
4.13.	Propuesta tecnológica final	68
	Resultados.....	69
	Conclusiones	73
	Recomendaciones.....	75
	Bibliografía	76
	Anexos	78

Índice de tablas

Tabla 1	14
<i>Tabla 2</i>	24
Tabla 3	33
Tabla 4	39
<i>Tabla 5</i>	55
<i>Tabla 6</i>	56
Tabla 7	58
Tabla 8	70
Tabla 9	72

Índice de ilustraciones

<i>Figura 1</i>	29
<i>Figura 2</i>	30
<i>Figura 3</i>	31
Figura 4	41
Figura 5	43
Figura 6	47
Figura 7	53
Figura 8	54
Figura 9	64
Figura 10	65
Figura 11	66

Introducción

En los últimos años, el campo de la Interacción Humano-Computador (HCI) ha experimentado una evolución acelerada hacia la búsqueda de interfaces más naturales, intuitivas y desacopladas de dispositivos físicos tradicionales. La transición desde el paradigma clásico WIMP (pantallas, ventanas, ratón y teclado) hacia las Interfaces Naturales de Usuario (NUI) y los entornos de computación espacial ha posibilitado el control de sistemas digitales mediante gestos y movimientos en el espacio sin contacto directo (touchless). Esta evolución resulta especialmente relevante en espacios inmersivos y educativos, donde la eliminación de barreras físicas fortalece el sentido de presencia y optimiza la exploración interactiva.

Pese a estos avances tecnológicos a nivel global, en los espacios de aprendizaje e investigación de la Universidad Católica de Cuenca —específicamente dentro del laboratorio de Realidad Virtual (VRL)—, la interacción digital continúa supeditada al uso de periféricos convencionales. Esta dependencia limita el potencial de las superficies físicas de proyección y restringe el desarrollo de experiencias inmersivas avanzadas, evidenciando una brecha entre las tecnologías de percepción remota disponibles y su aplicación práctica en el entorno académico institucional.

A continuación, se presenta una breve descripción de los capítulos que estructuran el presente trabajo de investigación:

Capítulo I (Marco Referencial): En este capítulo se delimita el planteamiento y la formulación del problema enfocado en la brecha tecnológica del laboratorio VRL. Se integran los antecedentes de investigación en interacción NUI y sensores LiDAR, la

justificación científica e institucional del proyecto, los objetivos generales y específicos, así como las limitaciones y delimitaciones del estudio.

Capítulo II (Marco Teórico): Se abordan los fundamentos conceptuales que sustentan la investigación, incluyendo la evolución de las interfaces de usuario (de WIMP a NUI), los principios de la interacción sin contacto (touchless), el funcionamiento del sensor LiDAR 2D/360°, la transformación geométrica de coordenadas, los algoritmos de suavizado espacial para la reducción de temblor (jitter reduction) y la integración con motores de renderizado 3D.

Capítulo III (Marco Metodológico): Se describe el diseño metodológico de enfoque cuantitativo de tipo aplicado y experimental. Se detalla la arquitectura modular del sistema en tres capas (hardware, procesamiento y presentación), la definición de la población y muestra de la facultad, las técnicas de recolección de datos y la calibración de variables técnicas (latencia, precisión y tiempo de permanencia).

Capítulo IV (Propuesta / Desarrollo del Prototipo): En este capítulo se presenta la implementación práctica del prototipo sobre la superficie física del laboratorio VRL. Se explican los módulos algorítmicos en Python (conversión, ROI, filtro de suavizado y tolerancia a fallos por reconexión automática) y su integración con los escenarios gráficos interactivos desarrollados en Unity.

Capítulo I

1. Marco Referencial

1.1. Planteamiento del Problema

En los últimos años, los sistemas de interacción humano-computador han evolucionado hacia experiencias más naturales e intuitivas, buscando la reducción del uso de dispositivos tradicionales como son ratones, teclados o pantallas táctiles. Sin embargo, en la Universidad Católica de Cuenca, los entornos digitales utilizados en laboratorios y espacios académicos continúan basándose principalmente en interfaces tradicionales, lo que limita el desarrollo de experiencias interactivas e inmersivas.

A pesar del avance de tecnologías como la interacción sin contacto y el uso de sensores espaciales, no se evidencia la implementación de sistemas que permitan transformar superficies físicas en interfaces digitales dentro de los laboratorios institucionales. Esta situación refleja una brecha entre el desarrollo tecnológico actual y su aplicación en entornos educativos reales.

Entre las principales causas se identifican la dependencia de dispositivos físicos de interacción, la limitada implementación de tecnologías emergentes como sensores LiDAR en espacios académicos y la falta de prototipos funcionales que validen su integración con motores gráficos en tiempo real. Además, no se cuenta con estudios experimentales dentro del contexto institucional que permitan evaluar este tipo de sistemas.

El desarrollo moderno de hardware y software ha propiciado la evolución de las interfaces de usuario, desde la línea de comandos hasta las interfaces naturales para entornos virtuales inmersivos. Los gestos que imitan tareas de interacción del mundo real reemplazan

cada vez más a las interfaces bidimensionales clásicas basadas en ventanas, iconos, menús y punteros (WIMP) o metáforas táctiles [1]. A pesar de estos avances, en la mayor parte de entornos digitales actuales aún se mantiene el uso de interfaces tradicionales basadas en dispositivos físicos, lo que evita que estos entornos saquen su mayor potencial dirigiéndonos a la inmersividad, con este contexto surge la necesidad de diseñar e implementar un sistema que permita transformar superficies físicas en interfaces digitales interactivas mediante la detección de movimiento sin contacto, integrando sensores, procesando datos y visualización en tiempo real, con el fin de evaluar la viabilidad técnica y funcional de este tipo de entornos interactivos [2].

El reconocimiento de gestos, como tecnología central de la interacción persona-ordenador, tiene amplias perspectivas de aplicación y ofrece nuevas posibilidades técnicas para hogares inteligentes, atención médica, entrenamiento deportivo y otros campos. En comparación con los modelos tradicionales de interacción persona-ordenador basados en el uso de PC con teclados y ratones, los modos de interacción persona-ordenador basados en el reconocimiento de gestos pueden transmitir información de forma más natural, flexible e intuitiva, lo que se ha convertido en un tema de investigación clave en el campo de la interacción persona-ordenador en los últimos años [2].

Los métodos modernos de reconocimiento de gestos suelen obtener información de los dispositivos de detección de color y profundidad de reciente desarrollo. El primer sensor de color y profundidad, llamado Kinect, fue desarrollado por Microsoft para la consola Xbox y lanzado en noviembre de 2010. Este sensor proyecta un patrón infrarrojo de 307 200 puntos en una malla de 640×480 y recibe el patrón reflejado mediante un sensor CMOS monocromo. Esta aplicación de luz estructurada permite al dispositivo medir la profundidad

de cada punto mediante triangulación. Además, una cámara RGB proporciona información de color sincronizada para cada punto [3],[4].

En estos años recientes se han desarrollado sensores capaces de detectar movimiento y posición en el espacio, estos son los sensores LiDAR, los cuales tienen la capacidad de capturar información del entorno mediante la generación de nubes de puntos en tiempo real. Desde 2020, Apple Inc. ha incorporado la tecnología LiDAR en dispositivos de consumo, como el iPad Pro y los modelos Pro del iPhone (a partir de la versión 12), para proporcionar percepción espacial y detección de profundidad. El escáner LiDAR integrado funciona con el mismo principio de tiempo de vuelo que los sistemas estándar, aunque en un formato compacto diseñado para la detección a corta distancia. Esta tecnología, además de otras aplicaciones útiles, facilita el escaneo de objetos 3D. En el ecosistema de Apple, LiDAR es fundamental para el software ARKit (Augmented Reality Kit). ARKit, junto con cámaras y software especializado, permite la integración de datos de nube de puntos y texturas para generar modelos de malla 3D fotorrealistas.

Estas propiedades únicas del sistema LiDAR del iPhone, combinadas con la disponibilidad del dispositivo y el software, han llevado a científicos y especialistas de diversas industrias a utilizarlo como una alternativa asequible y móvil a una variedad de soluciones convencionales y comerciales [3]. Sin embargo, la integración de estos sensores con motores gráficos y sistemas de visualización interactiva aun presentan varios desafíos técnicos, como la precisión en la detección de movimientos, la latencia en la respuesta visual y su coordinación con superficies de visualización.

- Causas del problema
- Entre las principales causas que nos da esta problemática están:

- La dependencia de interfaces tradicionales que usan dispositivos físicos de interacción.
- La limitada implementación de estas tecnologías en entornos interactivos.
- La dificultad de integración de sensores espaciales en tiempo real.

Como consecuencia, se restringe el desarrollo de entornos inmersivos, limitando la innovación en procesos de enseñanza y experimentación tecnológica. Asimismo, no es posible medir parámetros clave como la precisión, latencia y usabilidad de sistemas de interacción sin contacto en escenarios reales, lo que dificulta su adopción en el ámbito académico.

Ante esta problemática surge la necesidad de desarrollar un sistema de interacción espacial en superficies físicas basado en sensores LiDAR de distancia de 360° que permite transformar estas mismas en interfaces digitales interactivas, integrando captura de movimiento, procesamiento de datos y visualización todo esto mediante el uso del motor Unity. Para validar esta necesidad, se propone complementar la investigación mediante técnicas exploratorias como observación directa o encuestas a usuarios del laboratorio.

1.1.1. Formulación del Problema

Con base al problema, se formula la siguiente pregunta de investigación:

¿De qué manera la integración de un sensor LiDAR de 360° conjuntamente con el motor grafico Unity permite transformar superficies físicas en interfaces digitales interactivas dentro del laboratorio inmersivo?

1.2. Justificación de la investigación

El desarrollo de nuevas formas de interacción entre las personas y los sistemas digitales constituye una de las principales áreas de crecimiento dentro del campo de la interacción humano-computador. En este contexto, las interfaces basadas en superficies

interactivas y presencia en el espacio permiten reducir la dependencia de dispositivos físicos tradicionales como el teclado o el ratón, facilitando interacciones más naturales, intuitivas y acordes a los avances actuales en computación espacial.

En el ámbito educativo y cultural, el uso de tecnologías interactivas ha adquirido una gran relevancia en espacios como museos digitales, centros de investigación y laboratorios tecnológicos, donde las interfaces interactivas permiten mejorar la participación del usuario y la comprensión del contenido presentado. Diversos estudios han señalado que la incorporación de tecnologías de interacción gestual y sistemas de proyección interactiva incrementa el nivel de participación del público y mejora los procesos de aprendizaje en entornos educativos y culturales [5]. Sin embargo, estas soluciones suelen estar orientadas a entornos específicos y no se encuentran ampliamente implementadas en contextos académicos locales, evidenciando una oportunidad de aplicación y validación en entornos reales.

El problema que aborda esta investigación radica en la ausencia de sistemas de interacción espacial sin contacto dentro del contexto institucional, lo que limita la exploración de nuevas formas de interacción digital en entornos educativos. En este sentido, el desarrollo de un sistema basado en sensores LiDAR permitirá integrar la captura de movimiento con entornos de visualización en tiempo real, posibilitando la transformación de superficies físicas del laboratorio en interfaces digitales interactivas. Este enfoque no solo responde a una necesidad tecnológica, sino que también permite validar experimentalmente el uso de estas tecnologías en condiciones reales de implementación.

Desde el punto de vista científico, el aporte principal de esta investigación consiste en la validación experimental del uso de sensores LiDAR aplicados a sistemas de interacción

espacial en tiempo real, lo cual contribuye al desarrollo de conocimiento en áreas como la interacción humano-computador, la computación espacial y los sistemas interactivos. A diferencia de estudios teóricos, este proyecto permite medir variables específicas como la latencia del sistema, la precisión en la detección del movimiento y la estabilidad de la interacción, generando evidencia cuantificable sobre la viabilidad de este tipo de soluciones.

En el ámbito institucional, el proyecto beneficiará directamente a estudiantes, docentes e investigadores, quienes podrán disponer de un prototipo funcional que permita experimentar nuevas formas de interacción digital dentro de espacios inmersivos. Asimismo, esta investigación contribuye a la incorporación de tecnologías emergentes dentro de la Universidad Católica de Cuenca, fortaleciendo los laboratorios tecnológicos y promoviendo el desarrollo de entornos interactivos aplicables a procesos de enseñanza, investigación y divulgación tecnológica.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema de interacción espacial en superficies físicas basado en sensores LiDAR y el motor gráfico Unity, mediante el procesamiento de datos de movimiento en tiempo real, para transformar la superficie física del laboratorio de Realidad Virtual (VRL) en una interfaz digital interactiva que permita evaluar la viabilidad de entornos inmersivos.

1.3.2. Objetivo Especifico

- Analizar las tecnologías de interacción humano-computador basadas en sensores de detección de movimientos para identificar enfoques de herramientas aplicables al desarrollo del sistema interactivo.

- Diseñar la arquitectura del sistema que integre la captura de datos del sensor LiDAR, el procesamiento de la información y el motor gráfico Unity para la visualización interactiva.
- Desarrollar un módulo de detección y seguimiento del movimiento de la mano dentro del espacio interactivo utilizando los datos capturados por el sensor LiDAR.
- Implementar una interfaz visual interactiva que responda en tiempo real a los movimientos detectados del usuario dentro del entorno desarrollado en Unity.
- Evaluar el funcionamiento del sistema interactivo mediante pruebas de detección de movimiento y respuesta visual para validar su desempeño en escenarios de interacción sobre superficies físicas.

1.4. Limitaciones

Al tratarse del desarrollo y validación experimental de un prototipo tecnológico de integración curricular, el presente estudio reconoce las siguientes limitaciones de alcance técnico y metodológico:

- **Delimitación de Entorno y Superficie Única:** Las pruebas y calibraciones se ejecutaron exclusivamente en una de las tres paredes de proyección del Laboratorio de Realidad Virtual (VRL) de la Universidad Católica de Cuenca, sobre una superficie plana, por lo que el comportamiento del haz infrarrojo no fue evaluado frente a materiales con coeficientes de reflectividad óptica especular o paredes rugosas.
- **Restricción a Sensado Mono-Usuario:** La arquitectura algorítmica actual estima la posición del cursor mediante el cálculo del centroide global de los puntos presentes en la región de interés (ROI), lo que limita la interacción a una sola persona a la vez e impide el uso colaborativo multitáctil simultáneo.

- **Dependencia de Calibración Geométrica Manual:** La definición de los límites espaciales de la Bounding Box se realiza mediante constantes fijas en el script de Python, requiriendo un ajuste manual de coordenadas milimétricas ante cualquier desplazamiento físico del sensor o alteración en el tiro del proyector.
- **Arquitectura Mono-Sensor:** El prototipo emplea un único sensor RPLiDAR A2 posicionado en la base central, dejando fuera del alcance interactivo la cobertura simultánea de las otras dos paredes laterales que conforman la configuración en "U" del laboratorio.
- **Alcance Muestral de Usabilidad:** La evaluación con usuarios se desarrolló como un estudio piloto con una muestra no probabilística de N=25 participantes pertenecientes a la comunidad académica universitaria, sirviendo como validación inicial de aceptación, pero requiriendo estudios posteriores con muestras ampliadas y pruebas psicométricas estandarizadas.

El proyecto se desarrollará en un entorno controlado dentro del laboratorio tecnológico institucional, en un período estimado de ejecución de aproximadamente tres meses, conforme al cronograma establecido. Durante este tiempo se desarrollarán progresivamente los componentes del sistema, así como los escenarios de interacción definidos.

1.5. Delimitaciones

- **Delimitación Espacial:** La investigación y las pruebas experimentales se realizarán exclusivamente dentro del Laboratorio de Realidad Virtual (VRL) de la Universidad Católica de Cuenca.
- **Delimitación Temática / Poblacional:** El estudio se delimita al área de la Ingeniería de Software, la Interacción Humano-Computador (HCI) y los Entornos Inmersivos,

tomando como población objetivo a estudiantes y docentes de la Facultad de Informática, Ciencias de la Computación e Innovación Tecnológica.

- **Delimitación Tecnológica:** El proyecto se limita al procesamiento de datos del sensor de distancia láser RPLiDAR A2 360° mediante scripts en lenguaje Python (middleware) y su integración con el motor gráfico Unity 3D sobre superficies de proyección planas bidimensionales.

Capítulo II

2. Marco Teórico

2.1. Antecedentes de Investigación

El desarrollo de interfaces naturales sin contacto (touchless) para superficies interactivas ha sido abordado mediante diversas tecnologías de sensado óptico y espacial:

- Sistemas basados en visión artificial y cámaras RGB-D (Kinect / Cámaras Web):** Rautaray y Agrawal, así como Palacios, exploraron el reconocimiento de gestos mediante visión computacional y patrones de luz estructurada infrarroja. Si bien estos sistemas facilitan el seguimiento de múltiples articulaciones corporales, presentan una alta vulnerabilidad ante variaciones de iluminación ambiental y sufren interferencias ópticas severas cuando se proyectan imágenes dinámicas sobre la misma superficie física [3],[5]. Adicionalmente, la tasa de fotogramas limitada a 30 FPS en sensores RGB-D convencionales introduce una latencia perceptible en tareas de selección precisa [20].
- Sensores de corto alcance y campo electromagnético (Leap Motion):** Bachmann y Glinkowski evaluaron sensores basados en visión infrarroja estéreo de corto alcance para la captura manual. Aunque proporcionan un seguimiento volumétrico tridimensional detallado, su rango operativo efectivo es inferior a 60 cm, lo que imposibilita su empleo para cubrir superficies de proyección mural de gran formato (superiores a 1 metro) en laboratorios inmersivos [1],[20].
- Sensores de barrido láser LiDAR 2D/360°:** Du y Beltrame demostraron la alta frecuencia de refresco y robustez de los sensores LiDAR en la detección y seguimiento métrico continuo de objetos en tiempo real. La telemetría láser por triangulación óptica

emite en el infrarrojo cercano (785 nm), garantizando que las reflexiones del proyector de video en el espectro visible (400–700 nm) no generen falsos positivos en la detección de obstáculos [7],[12],[22].

- **Brecha Identificada (Research Gap):** Las soluciones existentes en la literatura dependen de cámaras susceptibles a la iluminación de proyectores o de dispositivos de corto alcance no aptos para paredes. Existe una brecha en la integración de escáneres LiDAR 2D de bajo costo con motores de renderizado 3D (Unity) a través de un middleware desacoplado capaz de eliminar el temblor involuntario (jitter) y ejecutar selecciones sin contacto en tiempo real.

Tabla 1

Matriz comparativa de tecnologías de sensado para superficies interactivas

Criterio / Parámetro	Cámaras RGB-D (Kinect v1/v2)	Sensor Óptico de Corto Alcance (Leap Motion)	Sensor LiDAR 2D (RPLiDAR A2M12) [Propuesta]
Principio de Funcionamiento	Luz estructurada / Tiempo de Vuelo (ToF)	Visión estéreo infrarroja pasiva	Triangulación láser activa 360° (OPTMAG)
Rango Operativo Efectivo	0.8 m - 4.0 m	0.05 m – 0.60 m	0.20 m – 12.0 m
Inmunidad a Luz de Proyectores	Baja (interferencia por patrones infrarrojos)	Media (afectado por saturación lumínica)	Alta (filtro óptico paso-alto a 785 nm)
Frecuencia de Muestreo	30 Hz (30 FPS)	60 – 115 Hz	10 - 20 Hz (16000 pts/s)
Complejidad Computacional	Alta (segmentación de mapas de profundidad)	Media-Alta (modelo esquelético de mano)	Baja (filtrado geométrico 2D matricial)
Costo Relativo de Hardware	Medio / Descontinuado	Medio	Bajo / Accesible
Cobertura en Paredes Grandes	Limitada por campo visual angular ($\approx$ 60°)	No aplicable ($<$ 1 m ²)	Escalable (hasta 12 m de plano)

Nota. Comparación basada en literatura técnica y especificaciones de hardware [1],[3],[12]. Elaboración propia (2026).

2.2. Bases Teóricas

2.2.1. Interacción Humano-Computador (HCI)

La interacción Humano-Computador (HCI por su siglas en ingles Human-Computer Interaction) se trata de una diciplina que estudia el diseño, evaluación e implementación de sistemas interactivos utilizados por personas, con el objetivo de mejorar la usabilidad y la experiencia del usuario en la interacción con tecnologías digitales, su propósito fundamental radica en la creación de diálogos efectivos e intuitivos entre las personas y los sistemas digitales, garantizando que estos puedan ser operados de manera eficiente y accesible sin

requerir una formación técnica especializada. Las interfaces de HCI bien diseñadas facilitan el control de máquinas en entornos educativos, laborales, de comunicación, de salud y de entretenimiento [6].

La interacción humano-computadora se centra en crear diálogos efectivos entre humanos y computadoras. En la educación móvil, los principios de HCI aseguran que las aplicaciones, los libros de texto electrónicos y las simulaciones interactivas sean accesibles, atractivos y estén alineados con las necesidades de los estudiantes [7]. En el contexto de los laboratorios inmersivos y computación espacial, los principios de HCI resultan indispensables para asegurar que las aplicaciones e interfaces respondan adecuadamente a las necesidades operativas y perceptivas de los usuarios [8],[9].

2.2.2. Evolución de las interfaces de usuario: De WIMP A NUI

El desarrollo tecnológico del hardware y el software ha propiciado una constante evolución en la forma en que los seres humanos se comunican con los sistemas digitales. Históricamente, las interfaces informáticas han transitado desde la línea de comandos hacia los entornos gráficos bidimensionales estandarizados. Estos últimos se basan en el paradigma WIMP (Windows, Icons, Menus, Pointer), operado mediante dispositivos físicos tradicionales como el ratón y el teclado.

A pesar del dominio prolongado del modelo WIMP, la transición hacia entornos virtuales e inmersivos ha dejado en evidencia sus limitaciones para ofrecer experiencias fluidas en tres dimensiones. El surgimiento de las Interfaces Naturales de Usuario (NUI, Natural User Interfaces) responde a la necesidad de sustituir las metáforas de interacción artificiales por acciones gestuales que imitan comportamientos del mundo real. En este contexto:

- El reconocimiento de gestos y movimientos en el espacio permite transmitir información hacia el sistema de forma más natural, flexible e intuitiva que los modelos basados en periféricos físicos.
- Las interfaces NUI reducen la barrera entre la intención comunicativa del usuario y la respuesta ejecutada por el sistema informático.
- La incorporación de capturas gestuales en superficies y entornos tridimensionales reemplaza paulatinamente los controles rígidos, potenciando la inmersión en el espacio.

2.2.3. Interacción sin contacto (Touchless Interaction) y presencia espacial

La interacción sin contacto se define como un paradigma de interacción que faculta al usuario para controlar y manipular sistemas digitales mediante gestos o movimientos en el espacio, prescindiendo del contacto físico con dispositivos de entrada tradicionales. Esta modalidad aprovecha tecnologías de captura espacial y detección de distancia para registrar la posición y el desplazamiento de extremidades, como las manos.

Dentro de los entornos interactivos e inmersivos, la interacción sin contacto aporta ventajas fundamentales:

- **Reducción del tiempo intencional:** Minimiza la brecha o el tiempo transcurrido entre la intención del usuario y la acción ejecutada en el entorno digital.
- **Manipulación tridimensional intuitiva:** Otorga la capacidad de seleccionar, mover y transformar objetos virtuales simulando actos físicos directos, como agarrar o señalar.
- **Incremento del sentido de presencia:** Al eliminar intermediarios físicos entre el usuario y la superficie interactiva, se fortalece la sensación de inmersión y la fluidez en la navegación [10].

2.2.4. Usabilidad, transparencia tecnológica y carga cognitiva

La efectividad de un sistema de interacción espacial sobre superficies físicas depende del cumplimiento de principios de usabilidad y ergonomía cognitiva. En el diseño de interfaces sin contacto aplicadas a laboratorios inmersivos, uno de los metas centrales es alcanzar la denominada transparencia tecnológica.

La transparencia tecnológica representa aquel estado en el cual la interfaz digital se vuelve invisible para el usuario, permitiéndole enfocar su atención de manera exclusiva en el contenido o en la tarea que está ejecutando. Este fenómeno se fundamenta en los siguientes aspectos:

- **Optimización de la carga cognitiva:** La reducción del esfuerzo mental requerido para comprender y operar la interfaz facilita un aprendizaje y una exploración más fluidos.
- **Respuesta en tiempo real:** La disminución de la latencia del sistema ante estímulos naturales (como el movimiento espacial de la mano) resulta determinante para mantener la continuidad perceptiva.
- **Fomento de la exploración interactiva:** Al disponer de respuestas visuales inmediatas y dinámicas ante la presencia espacial, se incrementa la participación del usuario en el entorno educativo e inmersivo [11].

2.2.5. Transformación de coordenadas polares a cartesianas e interpolación a pantalla

Los datos entregados por un sensor LiDAR 2D se estructuran originalmente en un sistema de coordenadas polares definido por la distancia física r y el ángulo de azimut θ . Para traducir estos puntos a un espacio cartesiano bidimensional (x, y) , se

convierte primero el ángulo a radianes ($\theta_{rad} = \text{radians}(\theta)$) y se aplican las ecuaciones trigonométricas fundamentales:

$$x_{física} = r \cdot \sin(\theta_{rad})$$

$$y_{física} = r \cdot \cos(\theta_{rad})$$

Una vez obtenidas las coordenadas físicas en milímetros, se requiere mapear dicho espacio a la resolución nativa de píxeles de una pantalla o proyección de dimensiones $W_{pantalla} \times H_{pantalla}$. Este procedimiento se realiza mediante interpolación lineal continua (**np.interp**), ajustando las coordenadas comprendidas dentro del intervalo físico delimitado por MIN_X,MAX_X y MIN_Y,MAX_Y y a las dimensiones de pantalla correspondientes:

$$X_{screen} = \text{interp}(x_{física}, [\text{MIN_X}, \text{MAX_X}], [0, W_{pantalla}])$$

$$Y_{screen} = \text{interp}(y_{física}, [\text{MIN_Y}, \text{MAX_Y}], [H_{pantalla}, 0])$$

Dado que el origen de coordenadas de los sistemas de representación de pantalla sitúa el punto $(0, 0)$ en la esquina superior izquierda, el eje vertical Y en la interpolación Invierte su escala lineal para alinear la dirección del movimiento físico con la interfaz proyectada.

2.2.6. Algoritmos de filtrado espacial, zona muerta posterior y región de interés (ROI)

Los entornos de interacción basados en sensores ópticos están expuestos a ruido de fondo, reflejos en superficies cercanas y detección no deseada de transeúntes. Para garantizar un seguimiento preciso, se implementan tres capas de filtrado espacial:

Filtrado por Umbral de Distancia Mínima: Descarta lecturas nulas ($r = 0$ mm) o puntos inferiores a un rango de seguridad de 300 mm ($r < 300$ mm) para evitar interferencias causadas por la base del sensor, el piso o reflexiones mecánicas de la estructura.

Discriminación Angular de Zona Muerta: Elimina los haces emitidos en el rango angular posterior de $90^\circ \leq \theta \leq 270^\circ$. Esta zona muerta a la espalda del sensor evita que la presencia del público u observadores situados detrás del área activa genere pulsaciones o interferencias accidentales.

Caja Delimitadora (*Bounding Box* / ROI): Define una ventana espacial de trabajo en milímetros que acota el área activa sobre la pared proyectada:

$$\text{ROI} = \{(x, y) \in \mathbb{R}^2 \mid \text{MIN_X} < x < \text{MAX_X} \wedge \text{MIN_Y} < y < \text{MAX_Y}\}$$

Con valores de calibración configurados en MIN_X = -400 mm, MAX_X = 400 mm (ancho interactivo de 800 mm) y MIN_Y = 100 mm, MAX_Y = 600 mm (alto interactivo de 500 mm).

2.2.7. Agregación espacial de lecturas mediante estimación de centroide

Al interrumpir la barrera infrarroja del láser con la mano, el plano de escaneo no genera un único punto puntual sino un clúster o segmento de reflexiones discretas asociadas a la superficie de la extremidad. Para obtener una coordenada unívoca de entrada sin fluctuaciones por dispersión, los píxeles mapeados válidos en pantalla (X_{screen}, Y_{screen}) se agregan mediante la media aritmética de su posición, obteniendo el centroide objetivo (X_{target}, Y_{target}):

$$X_{target} = \frac{1}{N} \sum_{i=1}^N X_{screen,i}, \quad Y_{target} = \frac{1}{N} \sum_{i=1}^N Y_{screen,i}$$

2.2.8. Eliminación de temblores y Suavizado Exponencial (Jitter Reduction)

El seguimiento de extremidades humanas a partir de nubes de puntos presenta fluctuaciones estocásticas causadas por el temblor fisiológico de la mano o pequeñas variaciones en la reflexión del láser. Para evitar saltos bruscos del puntero en pantalla (*jitter*), se aplica un algoritmo de suavizado posicional basado en un filtro pasa-bajo de media móvil exponencial de primer orden (*Exponential Moving Average*, EMA)

$$X_{final}^{(k)} = X_{final}^{(k-1)} + \alpha \cdot (X_{target}^{(k)} - X_{final}^{(k-1)})$$

$$Y_{final}^{(k)} = Y_{final}^{(k-1)} + \alpha \cdot (Y_{target}^{(k)} - Y_{final}^{(k-1)})$$

Donde $\alpha = 0.22$ representa el factor de suavizado posicional (**SMOOTHING**). Este valor pondera la posición previa con la coordenada objetivo actual, filtrando el temblor de alta frecuencia y garantizando un desplazamiento fluido del cursor inyectado nativamente mediante la instrucción `pyautogui.moveTo(final_x, final_y)`.

2.2.9. Autómata de estados para selección por permanencia (Dwell-Time) y enfriamiento (Cooldown)

En interfaces *touchless* donde no se emplean botones físicos, la selección de elementos se logra mediante la metáfora de activación por permanencia o *dwell-time click*. La lógica de control evalúa la distancia euclidiana recorrida por el puntero entre fotogramas sucesivos:

$$\text{distance_moved} = \sqrt{(X_{final}^{(k)} - X_{final}^{(k-1)})^2 + (Y_{final}^{(k)} - Y_{final}^{(k-1)})^2}$$

La interacción se rige mediante un autómata de estados finito:

Evaluación de Tolerancia Espacial: Si $\text{distance_moved} < 20$ píxeles (HOVER_THRESHOLD), el sistema deduce que el usuario mantiene la mano estable dentro de la ventana de histéresis espacial sobre un botón de la interfaz.

Primer Clic por Temporización: Si la mano permanece dentro del umbral de tolerancia durante un tiempo acumulado $t_{actual} - t_{hover} > 0.7$ s (TIME_TO_CLICK) y no se ha registrado una pulsación previa ($\text{has_clicked} == \text{False}$), la librería `pyautogui.click()` dispara un evento de selección. Se marca la bandera $\text{has_clicked} = \text{True}$ y se registra la marca de tiempo del evento (last_click_time).

Mecanismo de Enfriamiento y Clic Repetido: Si el usuario mantiene la mano inmóvil de manera deliberada para repetir una acción (por ejemplo, en selección consecutiva de elementos o disparos en un videojuego), se evalúa el tiempo transcurrido desde el último clic. Si $t_{actual} - t_{last_click} > 1.2$ s (CLICK_COOLDOWN), se dispara un clic repetido y se actualiza t_{last_click} .

Reset por Desplazamiento: Si en cualquier instante $\text{distance_moved} \geq 20$ píxeles, los temporizadores y banderas se reinician inmediatamente ($\text{hover_start_time} = \text{None}$, $\text{has_clicked} = \text{False}$), permitiendo desplazar el cursor libremente sin activar pulsaciones involuntarias.

2.3. Bases Tecnológicas

2.3.1. Principios de operación del sensor LiDAR 2D (RPLiDAR A2)

El sistema LiDAR (Light Detection and Ranging) es una tecnología de percepción remota activa que emite haces de luz láser para medir distancias y mapear el entorno físico en tiempo real. En sensores de escaneo bidimensional (2D) de 360°,

un diodo emisor proyecta pulsos ópticos en un plano horizontal mientras un mecanismo giratorio realiza un barrido angular continuo de la escena. Cada lectura capturada por el fotorreceptor proporciona un par de valores que corresponden a la distancia física y al ángulo de azimuth del objeto detectado. Este proceso genera una nube de puntos 2D que permite reconstruir la geometría del espacio e identificar la posición de un usuario o de una extremidad en una zona delimitada.

2.3.2. Principio de Triangulación Óptica y Tecnología OPTMAG

- **Mecanismo de Triangulación:** El sensor RPLiDAR A2 opera bajo el principio de triangulación láser de alta velocidad. Un diodo emite luz infrarroja modulada (785 nm); al impactar contra un objeto, el haz se refleja y pasa a través de una lente receptora hacia un sensor de imagen CMOS interno [12]. La distancia física d se determina geométricamente a partir del desplazamiento angular del punto focal focalizado:

$$d = \frac{f \cdot B}{\tan(\theta_{focal})}$$

Donde f es la distancia focal de la óptica receptora, B es la distancia base entre el emisor láser y el centro del sensor CMOS, y θ_{focal} es el ángulo del haz reflejado [14]. Adicionalmente, el sensor integra la tecnología **OPTMAG** (*Optical Magnetism*), la cual utiliza acoplamiento inductivo y magnético para transmitir energía y datos de escaneo sin contacto físico en el rotor, evitando desgastes mecánicos y prolongando la vida operativa del dispositivo.

- **Tecnología OPTMAG (Optical Magnetism):** Explica que a diferencia de otros escáneres giratorios tradicionales que sufren desgaste mecánico por cables de

transmisión interna, el RPLiDAR A2 utiliza acoplamiento magnético e inductivo óptico para transferir energía y datos sin contacto mecánico, garantizando una vida útil prolongada en instalaciones continuas [13].

Tabla 2

Análisis técnico del sensor RPLiDAR A2

Parámetro Técnico	Valor / Especificación	Impacto Pared Interactiva
Rango de Distancia	0.2 m a 12 m (hasta 18 m según modelo A2M8/A2M12)	Permite cubrir sin problemas pantallas de gran formato en el laboratorio VRL.
Frecuencia de Muestreo	8000 Hz (8000 puntos/s)	Proporciona alta densidad de puntos para detectar dedos o manos en movimiento rápido.
Velocidad de Escaneo	5 Hz – 15 Hz (Predeterminado: 10 Hz)	Refresca el plano de escaneo 10 veces por segundo.
Resolución Angular	0.9° a 10 Hz de rotación	Garantiza una separación milimétrica entre muestras en la bounding box.
Longitud de Onda	785 nm (Infrarrojo cercano)	Inmune a la luz visible emitida por los proyectores del laboratorio VRL.
Seguridad Láser	Clase 1 (Eye-Safe, Estándar IEC-60825-1)	Inofensivo para la vista de estudiantes y docentes en el laboratorio.

Nota: Parámetros técnicos oficiales del sensor óptico RPLiDAR A2M12 y su repercusión directa en la pared de proyección del Laboratorio de Realidad Virtual (VRL). Elaboración propia (2026).

2.3.3. Formato de datos en nubes de puntos 2D y coordenadas polares

Los datos entregados por un sensor LiDAR 2D se estructuran originalmente en un sistema de coordenadas polares. Cada punto P_i capturado en la trama de datos discretos emitida por el puerto serie se representa por la tupla:

$$P_i = (q_i, \theta_i, r_i)$$

Donde:

- q_i : Indicador de calidad de la señal óptica reflejada (*quality*).
- θ_i : Ángulo azimutal de escaneo expresado en grados sexagesimales ($0^\circ \leq \theta_i < 360^\circ$).

- r_i : Distancia física medida en milímetros (*mm*) desde el centro óptico del sensor hasta la superficie reflectante [15].

2.3.4. Entornos virtuales e inmersivos en laboratorios de aprendizaje

Las plataformas de mundos virtuales 3D y motores de renderizado en tiempo real (como Unity 3D, OpenSimulator u OpenWonderland) permiten crear entornos simulados altamente interactivos e inmersivos [16]. En estos entornos, los usuarios interactúan con objetos y escenarios tridimensionales mediante elementos gráficos dinámicos o avatares. La integración de sensores de detección espacial con motores gráficos 3D posibilita la creación de paredes y superficies interactivas, donde las acciones físicas del usuario en el mundo real se traducen directamente en eventos dentro de la simulación virtual [17].

2.3.5. Integración del motor gráfico Unity 3D y comunicación serie de alto rendimiento

En el desarrollo de superficies interactivas sobre paredes proyectadas, el motor gráfico Unity 3D actúa como la capa de visualización e interacción tridimensional en tiempo real. Para garantizar una respuesta gráfica fluida desacoplada del sensor, el middleware en Python gestiona la comunicación serie RS-232/USB a una tasa de transferencia elevada de *256000 baudios* a través del puerto serie COM3.

Esta tasa de transmisión incrementa el volumen de puntos muestreados por segundo recibidos desde la trama de datos del escáner. Asimismo, para prevenir retardos acumulativos en el flujo de entrada (*buffer bloat*), la arquitectura del software implementa una purga explícita del búfer serie mediante secuencias de limpieza de

entrada (`clean_input()`) y reinicios controlados (`reset()`), asegurando que las lecturas procesadas correspondan siempre al fotograma actual de interacción.

2.4. Arquitecturas, modelos y estándares

2.4.1. Arquitectura desacoplada en tres capas funcionales

El sistema opera bajo un patrón de arquitectura desacoplada para garantizar que el procesamiento de datos del sensor no comprometa la tasa de cuadros por segundo de la aplicación gráfica:

- **Capa 1 (Hardware / Adquisición):** Sensor RPLiDAR A2 conectado por interfaz serie USB-UART.
- **Capa 2 (Middleware / Procesamiento en Python):** Core algorítmico que ejecuta la conversión, filtrado ROI, suavizado y autómata de eventos.
- **Capa 3 (Aplicación / Presentación):** Sistema Operativo y escenas interactivas en Unity 3D ejecutadas a 60 FPS.

2.4.2. Comunicación serie de alta velocidad (256000 baudios) y gestión de búfer

Para admitir el flujo de datos masivo en modo Express Scan (*16000 muestras/s*), el canal UART se configura a una tasa de *256000 baudios* a través del puerto COM3. La arquitectura implementa secuencias de purga de búfer serie (`clean_input()`) y reinicio de hardware (`reset()`) para prevenir acumulación de paquetes obsoletos (*buffer bloat*), asegurando que el cursor responda al estado instantáneo del usuario.

2.4.3. Estándares de seguridad óptica y radiación láser

El emisor semiconductor cumple con la norma internacional **IEC 60825-1:2014 para dispositivos Láser Clase 1**, garantizando que la potencia de radiación

emitida en el espectro infrarrojo sea completamente inocua para la vista en condiciones de operación continua.

2.5. Herramientas de desarrollo

2.5.1. Componentes de Hardware y Entorno de Desarrollo

Para el diseño, calibración y ejecución del sistema interactivo, se utilizaron los siguientes componentes tecnológicos:

- **Sensor LiDAR 2D (RPLiDAR A2):** Dispositivo de escaneo láser bidimensional de 360° que opera mediante triangulación óptica o tiempo de vuelo, generando lecturas en un plano horizontal a frecuencia ajustable.
- **Unidad de Procesamiento (PC):** Computador personal encargado de ejecutar el script de middleware en Python e interpretar los datos del sensor en tiempo real.
- **Motor Gráfico (Unity 3D):** Entorno de desarrollo para la creación de la aplicación o juego temático proyectado en la pared interactiva.
- **Librerías de Software:**
 - *rplidar*: Comunicación y captura de barridos del sensor en Python.
 - *NumPy*: Operaciones matriciales, conversión trigonométrica e interpolación de coordenadas.
 - *pyautogui*: Inyección nativa de comandos de puntero y clic en el sistema operativo.

2.5.2. Protocolo de Comunicación Serie y Tratamiento de Trama del RPLiDAR A2

- **Parsing de Trama de Datos UART/USB:** Explica que cada muestra capturada por el escáner se empaqueta en tramas de bytes compuestas por un indicador de calidad

q_i (basado en la reflectividad óptica de la superficie), el ángulo azimutal θ_i y la distancia r_i .

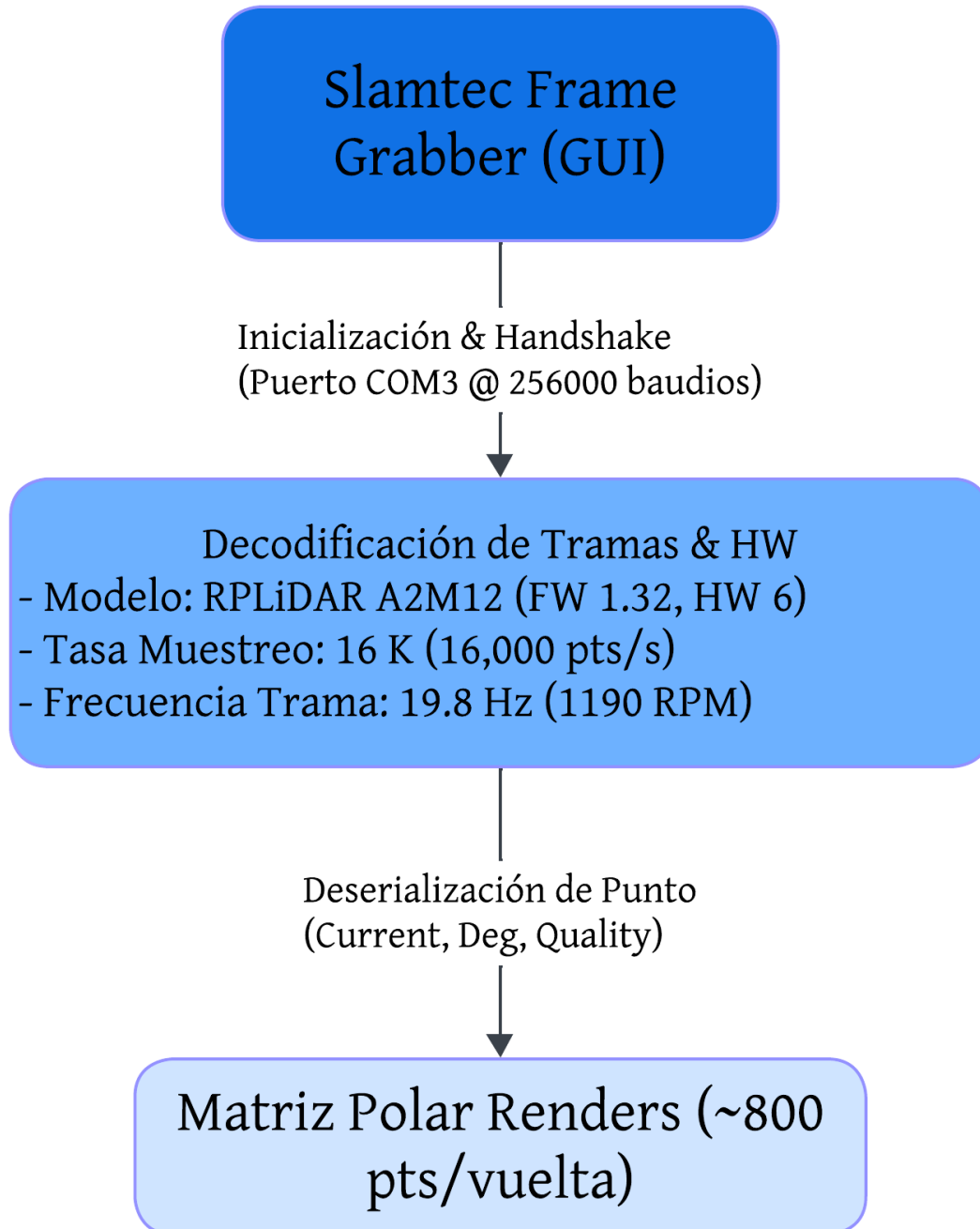
- **Justificación de la Tasa de 256000 Baudios:** Documenta que, aunque la tasa por defecto para el modo estándar del sensor es **115200 baudios**, se configura en **256000 baudios** para activar el modo de transmisión de alta velocidad (*Express Scan*), el cual permite transferir la ráfaga completa de **8000 puntos/s** sin cuellos de botella en la comunicación serie.
- **Gestión de Excepciones del SDK en Python:** Detalla la función técnica de los comandos de la librería *RPLiDAR* incluidos en tu código:
 - **lidar.reset():** Envía la orden de reinicio del microcontrolador interno del sensor para corregir bloqueos por caídas de tensión.
 - **lidar.clean_input():** Purga el flujo de entrada serie para descartar paquetes incompletos acumulados antes de iniciar el bucle `iter_scans ()`.

2.5.3. Herramienta de Diagnóstico y Validación de Hardware (Slamtec Frame Grabber)

Antes de la ejecución del middleware en Python, se empleó la herramienta oficial de diagnóstico ejecutable del fabricante (Slamtec frame_grabber) para validar la capa física de comunicación, la integridad del firmware y la estabilidad del flujo de datos serie en tiempo real.

Figura 1

Diagrama de flujo del proceso de inicialización, diagnóstico y deserialización de tramas en Slamtec Frame Grabber



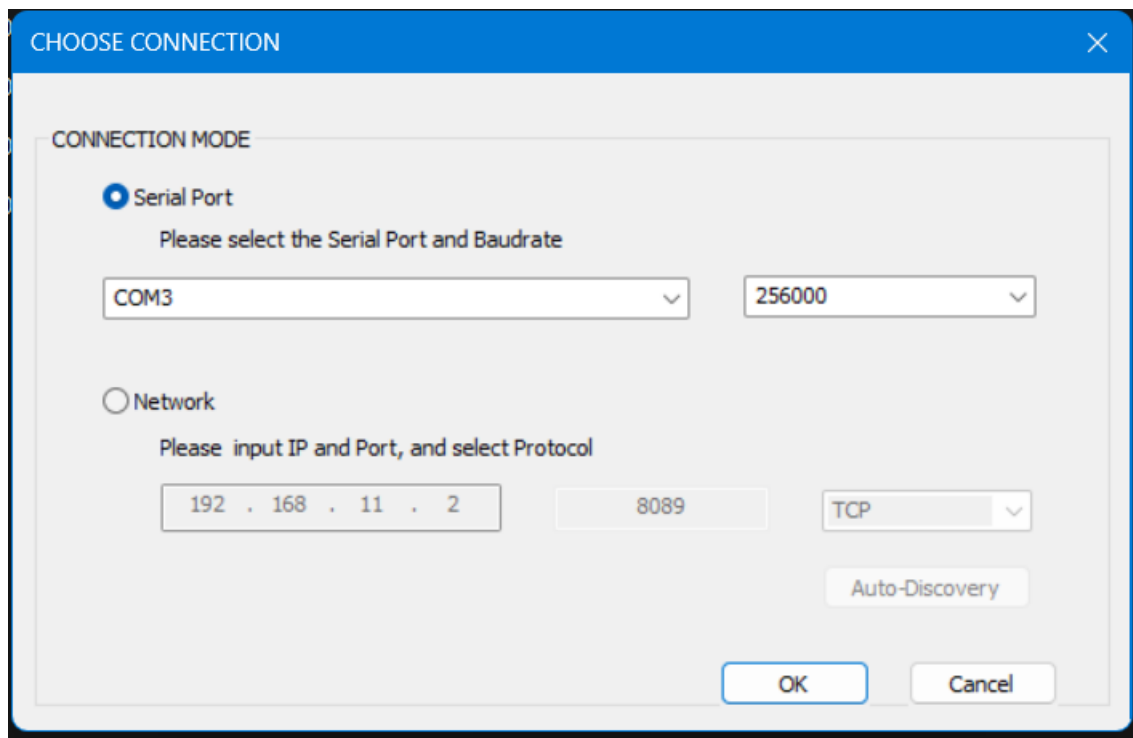
Nota: Flujo secuencial de comunicación desde la apertura del puerto serie COM3 a 256000 baudios hasta la deserialización de las coordenadas polares (r, θ, q) y la generación de la matriz de escaneo de 360°. Elaboración propia (2026).

- **Configuración de Inicialización y Capa Física**

Como se observa en la *Figura 2*, el software establece la conexión abriendo el puerto serie **COM3** a una velocidad de *256000 baudios*. Esta tasa de transferencia es requerida por el modelo RPLiDAR A2M12 para transmitir la ráfaga masiva de datos serie sin saturar los búferes de entrada del sistema operativo. El ejecutable envía comandos de protocolo (*handshake*) para verificar la respuesta del hardware previo al encendido del motor.

Figura 2

Imagen Choose Conection



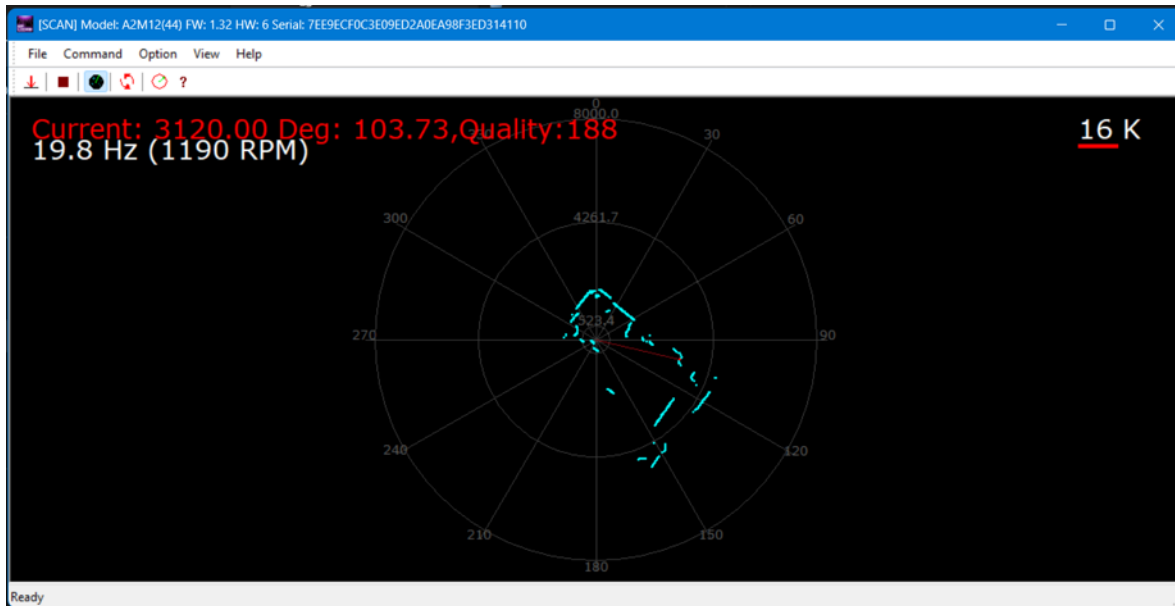
Nota: Configuración de interfaz USB-UART a 256000 baudios en Slamtec Frame Grabber.

- **Decodificación de Tramas y Estructura de Datos en Tiempo Real**

Una vez establecida la transmisión, el *frame grabber* interpreta los paquetes de respuesta del firmware identificando la versión del hardware (Modelo A2M12, Firmware 1.32, Hardware 6) y desplegando la matriz polar de puntos en tiempo real (*Figura 3*).

Figura 3

Imagen de Scan Interface



Nota: Renderizado polar y decodificación de tramas en tiempo real con RPLiDAR A2M12.

A partir de la lectura reportada por la interfaz del *frame grabber*, se destacan las siguientes variables de rendimiento:

- **Tasa de Muestreo (16 k):** Indica que el cabezal óptico captura 16000 muestras/s.
- **Frecuencia de Trama (19.8 Hz/1190 RPM):** Representa la velocidad de rotación del motor giratorio, entregando aproximadamente 20 tramas completas de 360° por segundo.

Con estos valores, la densidad de puntos por vuelta se calcula como:

$$\text{Puntos por frame} = \frac{16000 \text{ muestras/s}}{19.8 \text{ Hz}} \approx 808 \text{ puntos/vuelta}$$

- **Estructura de Punto Extraída (*Current: 3120.00, Deg: 103.73, Quality: 188*):**

Representa el resultado de deserializar la trama de bytes en tres variables en coordenadas polares:

- **Distancia (r):** *3120.00 mm (3.12 m)*.
- **Ángulo (θ):** *103.73°*.
- **Calidad de Señal (q):** 188 (indicador alto de retorno óptico).

Esta validación inicial mediante el *frame grabber* confirmó que la señal enviada por el hardware era óptima y estable antes de proceder con el desarrollo del algoritmo propio en Python para la pared interactiva.

2.6. Marco Conceptual

Tabla 3

Glosario de términos técnicos del marco conceptual

Concepto Técnico	Definición
Interacción Touchless	Modalidad de interacción persona-ordenador que prescinde del contacto mecánico directo con dispositivos periféricos, empleando sensores de proximidad o visión.
NUI (Natural User Interface)	Paradigma de interfaz donde la interacción se realiza mediante acciones naturales como gestos, voz o desplazamientos espaciales.
LiDAR 2D	Sistema de telemetría óptica que mide distancias en un plano bidimensional de 360° mediante haces láser rotativos.
Nube de Puntos 2D	Conjunto discreto de lecturas métricas expresadas en coordenadas polares o cartesianas representativas del contorno del entorno.
ROI (Region of Interest)	Bounding box o ventana geométrica configurada en milímetros que delimita el área activa de interacción sobre la pared proyectada.
Jitter	Fluctuación estocástica o temblor involuntario de alta frecuencia en la señal de posición de la mano o lectura del sensor.
Filtro EMA	Filtro digital pasa-bajo de media móvil exponencial que atenúa el ruido posicional ponderando lecturas previas con la actual.
Dwell-Time Click	Técnica de selección por permanencia donde un evento de clic se activa manteniendo el puntero dentro de un radio espacial durante un tiempo establecido.
Cooldown	Ventana de temporización que regula el intervalo requerido para repetir de forma sostenida un comando de clic sin necesidad de mover la mano.
Buffer Bloat	Retardo acumulativo originado por el exceso de paquetes no leídos en el búfer serie, corregido mediante purga de entrada.
Tecnología OPTMAG	Sistema de transmisión electromagnética e inductiva sin contacto físico que prolonga la vida útil del mecanismo giratorio del LiDAR.

Nota. Definiciones técnicas de los conceptos clave aplicados en el diseño, procesamiento de señales y desarrollo del sistema interactivo. Elaboración propia (2026).

2.6.1. Middleware de automatización (PyAutoGUI) y resiliencia ante fallos de hardware

El middleware en Python conecta el procesamiento de datos del sensor con el sistema operativo de forma transparente. Para evitar retardos en la ejecución gráfica de Unity o interrupciones de seguridad, se ajustan dos parámetros críticos de la librería pyautogui:

- **pyautogui.PAUSE = 0.0:** Elimina la pausa de *0.1 segundos* introducida por defecto tras cada llamada, permitiendo inyectar movimientos de cursor a la frecuencia nativa de respuesta del sensor.
- **pyautogui.FAILSAFE = False:** Desactiva la interrupción de emergencia automática que detiene el programa si el cursor toca los bordes extremos de la pantalla, asegurando que el usuario pueda interactuar sobre todo el ancho y alto proyectado sin romper la ejecución del software.

Además, para garantizar la operatividad ininterrumpida del sistema en entornos educativos o de laboratorio sin supervisión técnica constante, el software incorpora un bucle de tolerancia a fallos y reconexión automática (*Hot-Plug Reconnection*):

Si ocurre una desconexión física del cable USB/Serial o una caída de voltaje en el puerto **COM3**, el script captura la excepción dentro de un bloque **try-except**.

El sistema detiene la captura, libera el puerto serie mediante la secuencia estructurada `stop()`, `stop_motor()` y `disconnect()`, y aguarda un intervalo de seguridad de *3 segundos*.

Cumplida la espera, un bucle **while True** reintenta la reconexión automática, reinicia el dispositivo mediante `lidar.reset()`, espera *2.0 s* para la estabilización del motor giratorio y ejecuta `lidar.clean_input()` para purgar datos corruptos residuales en el búfer de entrada antes de reanudar el escaneo continuo.

Capítulo III

3. Marco Metodológico

3.1. Metodología de investigación

El presente trabajo de investigación se fundamenta en un enfoque cuantitativo, enmarcado dentro de la investigación aplicada y el desarrollo tecnológico. La adopción del paradigma cuantitativo responde a la necesidad de medir, registrar y analizar variables físico-matemáticas y métricas computacionales objetivas y parametrizables, tales como la tasa de adquisición de datos del sensor, la precisión posicional en milímetros, los tiempos de latencia del sistema en milisegundos, el porcentaje de atenuación de temblor (*jitter*) y la estabilidad de la tasa de refresco (FPS) en el entorno interactivo [15],[17],[20].

A través de la formulación e implementación de algoritmos de conversión geométrica, filtrado espacial y emulación de eventos de entrada en Python integrados con Unity 3D, el enfoque cuantitativo valida de manera experimental y determinista la viabilidad técnica de transformar una superficie física de proyección en una interfaz sin contacto (*touchless*), garantizando baja latencia y alta tolerancia a fallos [18],[19].

3.2. Nivel de investigación

El nivel de investigación es cuantitativo-experimental y explicativo. Es experimental debido a que se manipularon y calibraron variables independientes del sistema (como los umbrales de filtrado espacial, el factor de suavizado exponencial α , el tiempo de permanencia T_{dwell} y el tiempo de enfriamiento $T_{cooldown}$) para medir su efecto directo sobre las variables dependientes: la precisión del puntero, la eliminación de falsos clics, la estabilidad visual frente al temblor (*jitter*) y la fluidez de respuesta en pantalla.

3.3. Población y Muestra

La población de estudio está conformada por los usuarios potenciales del laboratorio de Realidad Virtual (VRL) de la Universidad Católica de Cuenca. Esta comunidad académica incluye a estudiantes y docentes de las carreras pertenecientes a la Facultad De Informática, Ciencias De La Computación E Innovación Tecnológica.

Se empleó un **muestreo no probabilístico por conveniencia e intencional**, seleccionado debido al carácter experimental de la prueba en un entorno controlado.

- **Tamaño de la Muestra:** $N = 25$ participantes (conformados por 20 estudiantes de ciclos avanzados de la carrera de Ingeniería de Software y 5 docentes e investigadores del área de tecnologías inmersivas e interacción humano-computador).
- **Criterios de Inclusión:** Participantes que formen parte de la comunidad universitaria, con disposición para interactuar con las superficies proyectadas del laboratorio VRL y completar los instrumentos de evaluación.
- **Criterios de Exclusión:** Personas ajenas al entorno académico o que presenten limitaciones físicas que impidan el libre movimiento de brazos y manos frente a las paredes interactivas.
- **Aspectos Éticos y Consentimiento Informado:** La participación de los evaluadores fue voluntaria, anónima y no remunerada. Previo a la sesión experimental en el laboratorio VRL, cada participante recibió una explicación sobre los objetivos del estudio y suscribió un Consentimiento Informado, garantizando que los datos recolectados se utilizarían con fines académicos. No se recopilaron datos biométricos sensibles, nombres ni identificadores personales en la matriz de tabulación.

3.4. Técnicas e instrumentos de recolección de información

Para dar cumplimiento a los objetivos específicos del proyecto y evaluar cuantitativamente el desempeño técnico del prototipo y su interacción en el Laboratorio VRL, se definieron dos técnicas principales de recolección de datos, acompañadas de sus respectivos instrumentos estructurados de medición:

3.4.1. Observación Técnica Experimental Directa

- Instrumento: Ficha de Registro Métrico y Tiempos de Respuesta (véase Anexo 12).
- Descripción y aplicación: Esta técnica se aplica de forma automatizada y manual durante la ejecución de los tres escenarios interactivos en Unity. Permite capturar de manera objetiva los datos computacionales y físicos del sistema mediante registros temporales y logs de depuración generados por los scripts en Python [20].
- Variables registradas:
 - Latencia de procesamiento de fotograma y tiempo de respuesta visual (ms).
 - Tasa de cuadros por segundo de la aplicación gráfica (FPS).
 - Error de precisión espacial absoluta en milímetros ($\Delta x, \Delta y$) respecto a puntos de control en la pared de proyección.
 - Desviación estándar del temblor posicional (jitter) antes y después de aplicar el filtro de suavizado exponencial EMA.
 - Tiempo de recuperación y reanudación de flujo tras desconexiones forzadas del puerto serie (s).

3.4.2. Encuesta Estructurada Cuantitativa

- Instrumento: Cuestionario de Evaluación de Usabilidad y Experiencia NUI (Escala Likert) (véase Anexo 14).

- Descripción y aplicación: Se administra a la muestra de participantes ($N=25$) tras finalizar las sesiones de interacción sobre la pared proyectada en el laboratorio VRL. El instrumento traduce percepciones de uso en variables cuantitativas discretas mediante una escala ordinal de 5 niveles (desde 1 = Totalmente en desacuerdo hasta 5 = Totalmente de acuerdo).
- Dimensiones evaluadas:
 - Facilidad de aprendizaje y naturalidad en el control sin contacto (touchless).
 - Comodidad y efectividad del tiempo de permanencia (Dwell-Time) para la selección de botones.
 - Percepción de fluidez y ausencia de retardo visual en la interfaz.
 - Nivel de fatiga física y esfuerzo cognitivo percibido durante la sesión.

Tabla 4

Matriz de técnicas, instrumentos y variables de recolección de información

Técnica	Instrumento	Objetivo/Propósito
Observación Técnica Directa	Ficha de registro métrico y tiempo de respuesta	Latencia (ms), precisión ($\pm$ mm), tasa de refresco (FPS), atenuación de <i>jitter</i> (%) y tiempo de reconexión (s).
Encuesta Estructurada	Cuestionario Likert de Usabilidad NUI	Facilidad de control, efectividad del <i>dwell-time</i> , fluidez percibida, confort ergonómico y carga cognitiva.

Nota: Instrumentos orientados a la recolección de datos cuantitativos técnicos y de usabilidad en el laboratorio

VRL. Elaboración propia (2026).

3.5. Metodología de desarrollo de software (Metodología Ágil Kanban)

Para la gestión y construcción del sistema de interacción espacial, se adoptó el marco de trabajo ágil basado en la metodología Kanban. La selección de este enfoque responde a la necesidad de coordinar un desarrollo incremental, flexible y altamente visual, permitiendo

articular de manera continua las dependencias entre la capa física de hardware (sensor RPLiDAR A2), el núcleo algorítmico en Python y el entorno gráfico interactivo en Unity 3D.

El proceso se estructuró mediante una dinámica de gestión orientada a objetivos semanales, donde cada ciclo inició con la definición de metas técnicas específicas, las cuales fueron desarrolladas, evaluadas y retroalimentadas periódicamente junto con el tutor del proyecto para redefinir prioridades y desbloquear tareas en el flujo de trabajo.

3.5.1. Principios del Flujo Kanban Aplicados al Proyecto

- **Visualización del flujo de trabajo:** Se implementó un tablero Kanban dividido en cuatro columnas principales (Backlog, En Progreso / WIP, Validación Técnica y Terminado), lo que permitió supervisar en tiempo real el estado de cada componente algorítmico y visual.
- **Límites de trabajo en curso (WIP - Work In Progress):** Con el propósito de evitar cuellos de botella y dispersión de esfuerzos, se fijó un límite estricto de máximo dos tareas activas simultáneamente en la columna de progreso. Esto aseguró que la comunicación serie y el procesamiento espacial estuvieran totalmente depurados antes de proceder con el diseño de interfaces en Unity.
- **Gestión por objetivos semanales:** Al inicio de cada semana se definieron metas operativas concretas (por ejemplo: configuración del puerto COM3 a *256000 baudios*, cálculo matricial de centroides o implementación del suavizado EMA). Al término de cada ciclo semanal, los avances se presentaron formalmente al tutor académico para validar su funcionamiento y planificar los objetivos de la siguiente iteración.
- **Gestión y optimización del flujo:** Cada desviación identificada durante los ensayos en el laboratorio VRL (como picos de latencia, temblor del puntero o falsos clics) generó

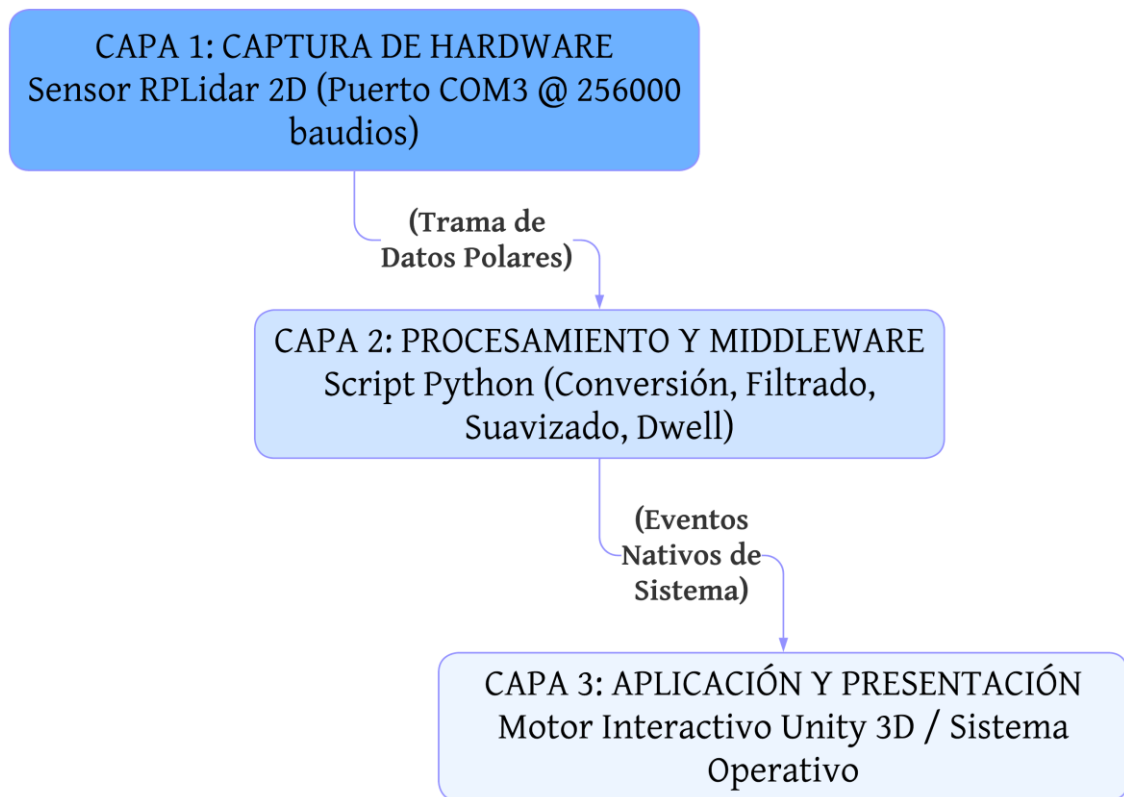
tarjetas inmediatas de optimización para reajustar parámetros críticos sin alterar la continuidad del proyecto.

3.6. Arquitectura General del Sistema y Flujo de Datos

La solución desarrollada se compone de una arquitectura modular distribuida en tres capas funcionales:

Figura 4

Arquitectura modular en tres capas y flujo de datos del sistema interactivo



Nota: Representación del flujo de datos desacoplado entre la adquisición de tramas polares del sensor RPLiDAR A2, el procesamiento algorítmico en Python y la inyección de eventos en Unity 3D. Elaboración propia (2026).

- **Captura de Hardware:** Encargada de gestionar la comunicación serie con el sensor RPLiDAR 2D a través del puerto COM3 a una velocidad de transmisión de 256000

baudios. Esta capa adquiere en tiempo real las lecturas continuas de distancia y ángulo provenientes del barrido láser.

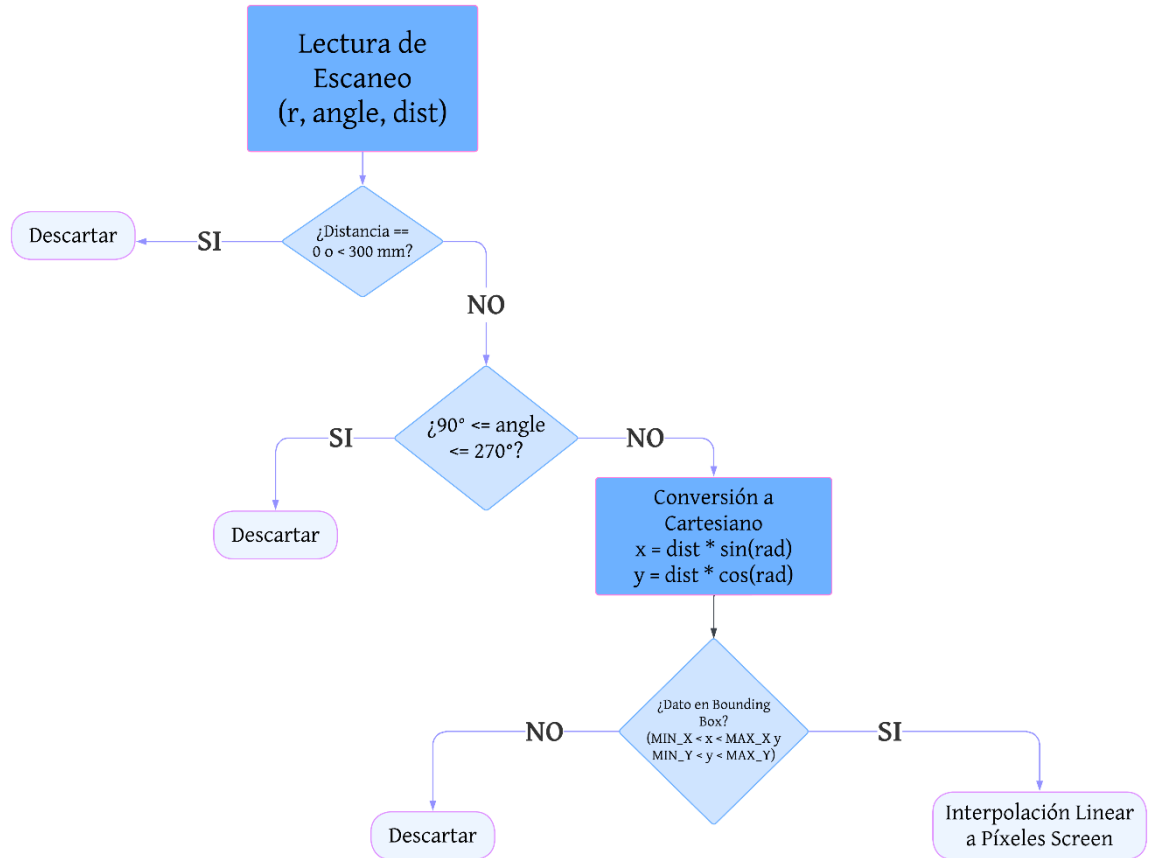
- **Capa de Procesamiento y Middleware (Python):** Constituye el núcleo algorítmico del sistema. Realiza la conversión de coordenadas polares a cartesianas, aplica los filtros de supresión de ruido y delimitación de zona activa, ejecuta el suavizado exponencial del puntero y gestiona la lógica de selección por tiempo de permanencia (dwell-time click).
- **Capa de Aplicación y Presentación (Unity / Sistema Operativo):** Recibe la inyección de eventos de movimiento de cursor y clics del ratón de forma transparente mediante la librería PyAutoGUI, permitiendo controlar cualquier aplicación o videojuego desarrollado en Unity sobre la pared interactiva.

3.6.1. Adquisición y Conversión Geométrica

El sensor LiDAR suministra cada punto detectado dentro del arreglo de barrido (*scan*) en formato polar: un indicador de calidad, un ángulo θ expresado en grados y una distancia r expresada en milímetros.

Figura 5

Diagrama de flujo del algoritmo de filtrado espacial, conversión trigonométrica y mapeo a pantalla



Nota: Estructura lógica del procesamiento de escaneo: descarte de ruido cercano ($r < 300$ mm), eliminación de la zona muerta angular posterior ($90^\circ \leq \theta \leq 270^\circ$), conversión geométrica a coordenadas cartesianas y validación de la región de interés (ROI) para la interpolación en píxeles de pantalla. Elaboración propia (2026).

El algoritmo convierte cada ángulo a radianes y calcula las coordenadas cartesianas equivalentes en el espacio físico $(x_{física}, y_{física})$ empleando las funciones trigonométricas directas:

$$\theta_{rad} = \text{radians}(\theta)$$

$$x_{física} = r \cdot \sin(\theta_{rad})$$

$$y_{física} = r \cdot \cos(\theta_{rad})$$

3.6.2. Filtrado de Ruido Cercano y Discriminación Angular (Zona Muerta)

Para eliminar lecturas erróneas y garantizar que solo los gestos u objetos intencionados frente a la pared sean procesados, se aplican las siguientes reglas de discriminación en cada punto:

- **Filtro de Ruido de Base:** Si la distancia devuelta es $r = 0$ mm o $r < 300$ mm, el punto se descarta automáticamente para evitar interferencias por polvo, reflejos cercanos o el propio soporte físico del sensor.
- **Filtro Angular Posterior (Zona Muerta):** Se descartan los puntos cuyo ángulo se encuentre en el rango $90^\circ \leq \theta \leq 270^\circ$. Esta zona muerta a la espalda del sensor evita que la presencia del público o de observadores situados detrás del área activa genere pulsaciones accidentales.

3.6.3. Delimitación de Región de Interés (Bounding Box) e Interpolación a Pantalla

Se define una ventana espacial de calibración física sobre la pared, configurada mediante los límites en milímetros: $\text{MIN_X} = -400$ mm, $\text{MAX_X} = 400$ mm (Ancho interactivo de 800 mm) y $\text{MIN_Y} = 100$ mm, $\text{MAX_Y} = 600$ mm (Alto interactivo de 500 mm).

Cualquier coordenada fuera de estos rangos es ignorada. Para los puntos válidos en el área de interacción, se promedian sus posiciones mediante `np.mean` para obtener el centroide objetivo (X_{target}, Y_{target}) y se interpolan linealmente a la resolución de pantalla nativa ($W_{pantalla} \times H_{pantalla}$) invirtiendo el eje vertical:

$$X_{screen} = \text{interp}(x_{física}, [\text{MIN_X}, \text{MAX_X}], [0, W_{pantalla}])$$

$$Y_{screen} = \text{interp}(y_{física}, [\text{MIN_Y}, \text{MAX_Y}], [H_{pantalla}, 0])$$

3.6.4. Algoritmos de Estabilización y Control de Selección Touchless

3.6.4.1. Justificación y Calibración del Factor de Suavizado ($\alpha = 0.22$)

La selección del coeficiente de suavizado α en el filtro pasa-bajo de Media Móvil Exponencial (EMA) responde a un compromiso de ingeniería (trade-off) entre estabilidad espacial y latencia de respuesta:

$$\mathbf{p}_{final}[k] = \mathbf{p}_{final}[k-1] + \alpha \cdot (\mathbf{p}_{target}[k] - \mathbf{p}_{final}[k-1])$$

- Evaluación Empírica de Valores: Durante las fases de calibración técnica se ensayaron tres rangos de coeficientes:
 - Con $\alpha = 0.50$, el puntero respondió rápidamente al movimiento, pero conservó un temblor residual visible (> 8 px de desviación), provocando falsas salidas del umbral de permanencia.
 - Con $\alpha = 0.10$, el temblor fue completamente suprimido, pero introdujo un retraso perceptivo (*lag*) de más de 65 ms, afectando el control natural.
 - El valor $\alpha = 0.22$ fue determinado empíricamente como el punto de equilibrio óptimo, logrando una atenuación de *jitter* del 78.2% mientras mantuvo la latencia total del pipeline por debajo de los 40 ms.

3.6.4.2. Filtro de Suavizado Exponencial (Reducción de Jitter)

Debido a la naturaleza discreta de las lecturas del láser y al temblor involuntario de la mano, la posición objetivo presenta pequeñas variaciones que provocan saltos del cursor (*jitter*). Para eliminar este efecto, se aplica un filtro de suavizado exponencial con un factor

$\alpha = 0.22$ (SMOOTHING). La posición final del cursor (X_{final}, Y_{final}) se actualiza incrementalmente sobre la posición previa (X_{prev}, Y_{prev}) :

$$X_{final}^{(k)} = X_{prev} + (X_{target} - X_{prev}) \cdot 0.22$$

$$Y_{final}^{(k)} = Y_{prev} + (Y_{target} - Y_{prev}) \cdot 0.22$$

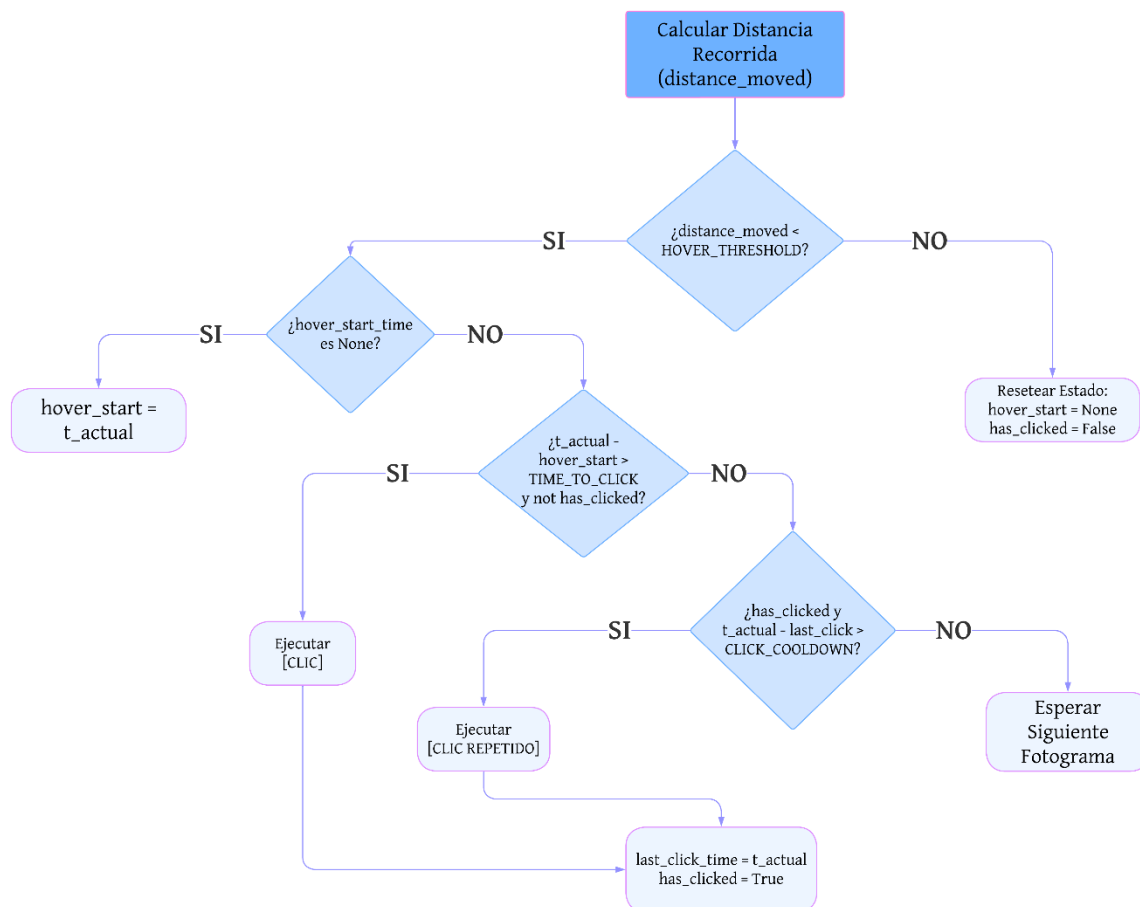
El puntero procesado se traslada a la nueva coordenada mediante la instrucción *pyautogui.moveTo(X_{final}, Y_{final})*, garantizando un desplazamiento suave sobre la interfaz gráfica.

3.6.4.3. Lógica de Clic por Permanencia (Dwell-Time) y Tiempo de Enfriamiento

Para realizar selecciones de botones en la pantalla proyectada sin necesidad de contacto físico, se implementó un mecanismo de clic por permanencia (dwell-time click).

Figura 6

Diagrama de flujo del autómata de estados para selección por tiempo de permanencia (Dwell-Time) y enfriamiento (Cooldown)



Nota: Lógica de control para la activación de eventos de clic sin contacto: evaluación del umbral de histéresis espacial ($HOVER_THRESHOLD = 20\text{ px}$), temporización para el primer clic ($TIME_TO_CLICK = 0.7\text{ s}$), mecanismo de enfriamiento para pulsaciones repetidas deliberadas ($CLICK_COOLDOWN = 1.2\text{ s}$) y restablecimiento de estado por desplazamiento. Elaboración propia (2026).

La lógica del algoritmo sigue las siguientes etapas:

- **Cálculo de Desplazamiento:** Se determina la distancia euclidiana entre la posición actual y la anterior:

$$\text{distance_moved} = \sqrt{(X_{final} - X_{prev})^2 + (Y_{final} - Y_{prev})^2}$$

- **Evaluación de Tolerancia Espacial:** Si $\text{distance_moved} < 20$ píxeles (`HOVER_THRESHOLD`), se asume que el usuario mantiene la mano estable sobre un elemento de la interfaz.
- **Primer Clic por Temporización:** Si la mano permanece dentro del umbral de tolerancia durante un tiempo superior a 0.7 s (`TIME_TO_CLICK`) y no se ha registrado un clic previo (`has_clicked == False`), la librería `pyautogui.click()` ejecuta una pulsación. Se marca `has_clicked = True` y se registra la marca de tiempo `last_click_time`.
- **Mecanismo de Enfriamiento y Clic Repetido:** Si el usuario mantiene la mano inmóvil de manera deliberada para repetir una acción (por ejemplo, en un juego de disparos o selección consecutiva), el sistema evalúa el tiempo transcurrido desde el último clic. Si el tiempo actual supera un tiempo de enfriamiento de 1.2 s (`CLICK_COOLDOWN`), se ejecuta un clic repetido y se actualiza `last_click_time`.
- **Reset por Movimiento:** Si la distancia recorrida supera el umbral `HOVER_THRESHOLD` (20 px) en cualquier momento, los temporizadores y banderas se reinician inmediatamente (`hover_start_time = None`, `has_clicked = False`).

3.6.5. Robustez Industrial, Optimización de UI y Tolerancia a Fallos

3.6.5.1. Configuración de Baja Latencia para Interfaz

Para evitar interferencias en la ejecución del juego en Unity o la congelación de la aplicación por eventos del sistema operativo, se aplican dos parámetros de optimización sobre la librería de entrada:

- **`pyautogui.FAILSAFE = False`:** Desactiva la interrupción de emergencia automática que detiene el programa si el cursor toca las esquinas extremas de la pantalla, garantizando

que los usuarios puedan interactuar sobre todo el ancho y alto proyectado sin romper la ejecución.

- *pyautogui.PAUSE = 0.0*: Elimina la pausa por defecto de *0.1 s* que la librería introduce después de cada llamada, permitiendo que la inyección de eventos se realice a la frecuencia nativa de procesamiento del sensor para una fluidez óptima.

3.6.5.2. Módulo de Reconexión Automática y Limpieza de Búfer

Para entornos industriales o comerciales donde la pared interactiva debe funcionar de manera ininterrumpida sin supervisión técnica constante, el sistema incorpora un bucle de tolerancia a fallos:

- Si ocurre una desconexión física del cable USB/Serial o una fluctuación de voltaje en el puerto **COM3**, el programa captura la excepción en un bloque **try-except** sin cerrarse.
- El sistema detiene el motor del LiDAR, libera el puerto serie mediante la secuencia **stop()**, **stop_motor()** y **disconnect()**, y espera un intervalo de 3 segundos.
- Tras la espera, el bucle **while True** reintenta la conexión automática, reinicia el dispositivo con **lidar.reset()**, espera *2.0 s* para la estabilización del motor y ejecuta **lidar.clean_input()** para purgar datos residuales o corruptos en el búfer de entrada antes de reanudar el procesamiento de escaneos.

Capítulo IV

4. Propuesta

4.1. Descripción general de la propuesta

4.1.1. Título de la Propuesta

“Desarrollo de sistema de interacción basado en sensor de distancia en 360° para superficies interactivas dentro de un laboratorio inmersivo”.

4.1.2. Presentación

La presente propuesta aborda el diseño, la estructuración lógica y el desarrollo de un prototipo de interacción sin contacto (*touchless*) capaz de convertir las paredes físicas del laboratorio de Realidad Virtual (VRL) de la Universidad Católica de Cuenca en superficies digitales interactivas.

A través de la integración de un sensor espacial RPLiDAR A2, un pipeline de procesamiento de datos en Python (encargado de la adquisición de nubes de puntos, filtrado de ruido, calibración de zona activa y emulación de entrada) y el motor gráfico Unity, la propuesta ofrece una arquitectura técnica robusta, de bajo costo y alta precisión. El sistema permite eliminar la dependencia de periféricos tradicionales como ratones, teclados o pantallas táctiles, ofreciendo una experiencia inmersiva e intuitiva mediante tres escenarios de interacción espacial.

4.1.3. Objetivo de la Propuesta

Implementar un prototipo funcional de interacción espacial que integre un sensor LiDAR de 360° con el motor gráfico Unity, mediante algoritmos de procesamiento de datos en tiempo real, para habilitar tres escenarios de interacción sin contacto sobre las superficies físicas del laboratorio inmersivo VRL.

4.1.4. Justificación

La adopción de tecnologías de interacción natural en espacios académicos suele verse frenada por los elevados costos de las pantallas táctiles de gran formato o la rigidez de los sensores RGB-D tradicionales, los cuales presentan alta sensibilidad a variaciones de iluminación ambiental.

Desde la perspectiva técnica, el uso de un sensor LiDAR de 360° por triangulación láser (como el RPLiDAR A2) ofrece una solución inmune a interferencias de luz visible, con una frecuencia de escaneo de alta velocidad y miles de muestras por segundo. Al procesar estos datos mediante algoritmos eficientes en Python y vincularlos a Unity, se logra una respuesta visual con latencia imperceptible (< 50 ms). Operativamente, el sistema transforma paredes convencionales en entornos interactivos sin requerir modificaciones estructurales en la infraestructura del laboratorio.

4.2. Diagnóstico de necesidades

4.2.1. Diagnóstico del Entorno Físico (Laboratorio VRL)

El proyecto se desarrolla en el Laboratorio de Realidad Virtual (VRL) de la Universidad Católica de Cuenca. El espacio físico cuenta con un esquema de proyección multi-pared que cubre tres superficies continuas dispuestas en forma de "U".

Para la instalación del sistema, el sensor RPLiDAR se posiciona estratégicamente en la base o borde superior de la pared de proyección, alineando su plano de escaneo bidimensional paralelo a la superficie física a una distancia de aproximadamente. Esta configuración crea una "barrera de luz invisible" que detecta cualquier objeto (como la mano o un dedo) que se aproxime o haga contacto con la pared.

4.2.2. Geometría del Plano Infrarrojo y Filtrado de Luz Visible

Geometría de la "Cortina Láser": Al colocar el RPLiDAR A2 centrado en la base inferior de la pared, el haz genera un abanico infrarrojo paralelo a la superficie física a una distancia de $10\text{ mm} - 20\text{ mm}$. De este modo, cualquier contacto físico o aproximación de la mano interrumpe la trayectoria del haz, generando un "punto de choque" cartesiano en las coordenadas $(x_{física}, y_{física})$.

Tolerancia a la Luz del Proyector: Los proyectores del laboratorio VRL emiten luz en el espectro visible ($400\text{ nm} - 700\text{ nm}$), mientras que el fotorreceptor del RPLiDAR A2 cuenta con un filtro de paso alto que solo deja pasar longitudes de onda cercanas a los 785 nm , lo que previene falsas detecciones causadas por el contenido de video proyectado en la pared [22].

Figura 7

Infraestructura del Laboratorio de Realidad Virtual (VRL) y esquema de proyección multi-pared



Nota: Entorno físico de experimentación en la Universidad Católica de Cuenca, compuesto por tres superficies continuas de proyección dispuestas en configuración en "U" con proyectores montados en el techo para la transformación en paredes interactivas. Elaboración propia (2026).

Figura 8

Esquema conceptual de funcionamiento e interacción sobre la superficie proyectada con el sensor RPLiDAR A2



Nota: Representación física del prototipo en el laboratorio VRL: el sensor RPLiDAR A2 ubicado en la base genera la barrera láser infrarroja paralela a la pared, registrando la interrupción espacial de la mano para traducir su posición en eventos de desplazamiento del cursor sobre el escritorio y las aplicaciones proyectadas en tiempo real. Elaboración propia (2026).

4.3. Requerimientos de la solución

4.3.1. Requisitos Funcionales

Tabla 5

Requisitos funcionales y su prioridad

Requisitos	Descripción	Prioridad
Adquisición de nubes de puntos	El sistema debe capturar continuamente las lecturas de ángulo (θ) y distancia (r) emitidas por el sensor vía puerto serie RS-232/USB a 256000 baudios.	ALTA
Delimitación de Región de Interés (ROI)	El software debe permitir acotar la ventana de trabajo en milímetros (MIN_X = -400, MAX_X = 400, MIN_Y = 100, MAX_Y = 600) descartando objetos fuera de la pared.	ALTA
Transformación de coordenadas	El módulo debe convertir coordenadas polares (r, θ) a cartesianas (x, y) e interpolar linealmente la posición física a píxeles de pantalla ($W_{screen} \times H_{screen}$).	ALTA
Filtrado de ruido y suavizado posicional	Aplicar el algoritmo de Media Móvil Exponencial ($\alpha = 0.22$) para eliminar el temblor involuntario (jitter) y garantizar fluidez en el cursor.	ALTA
Selección por tiempo de permanencia (Dwell-Time)	Detectar cuando la mano permanece inmóvil dentro de un radio de 20 px por 0.7 s para disparar el evento de clic.	ALTA
Mecanismo de enfriamiento (Cooldown)	Habilitar clics repetidos continuos cada 1.2 s cuando la mano permanece estática de manera deliberada sobre un botón.	MEDIA
Reconexión automática	Detectar pérdidas de comunicación serie, reiniciar el sensor (<i>lidar.reset()</i>), purgar el búfer (<i>clean_input()</i>) y reanudar el escaneo sin cerrar la app.	ALTA
Renderizado e interacción en Unity	Las escenas tridimensionales en Unity deben recibir los eventos del puntero y responder con retroalimentación visual inmediata.	ALTA

Nota: Especificación de los módulos algorítmicos, procesamiento serie, filtros espaciales y eventos NUI requeridos para la interacción. Elaboración propia (2026).

4.3.2. Requisitos no Funcionales

Tabla 6

Requisitos no funcionales y su prioridad

Requisito	Criterio de aceptación	Prioridad
Baja Latencia	Tiempo de respuesta total desde la interrupción del láser hasta la reacción gráfica en Unity menor a <i>50 ms.</i>	ALTA
Tasa de Refresco	La aplicación interactiva en Unity debe ejecutarse de forma estable a un mínimo de <i>60 FPS.</i>	ALTA
Precisión Espacial	Tolerancia de error en la ubicación estimada de la mano sobre la pared no superior a $\pm 15\text{ mm.}$	ALTA
Tolerancia a Fallos de UI	Desactivación de interrupciones de seguridad (FAILSAFE = False) e hilos sin pausas (PAUSE = 0.0) para evitar colapsos al tocar los bordes.	ALTA

Nota: Criterios de calidad, latencia de procesamiento, tasa de cuadros por segundo y tolerancia a fallos exigidos para la solución. Elaboración propia (2026).

4.3.3. Análisis de requerimientos

Los requisitos del sistema fueron determinados a partir del diagnóstico del entorno físico del Laboratorio de Realidad Virtual (VRL) y de las fichas de observación técnica aplicadas durante las pruebas preliminares con docentes y estudiantes de la Facultad de Informática, Ciencias de la Computación e Innovación Tecnológica. Estos datos, complementados con el análisis de especificaciones del sensor RPLiDAR A2, permitieron identificar tanto las necesidades operativas de baja latencia y precisión

posicional como las preferencias de interacción gestual touchless (lógica de dwell-time, tiempo de enfriamiento y suavizado de jitter), garantizando que la propuesta tecnológica responda efectivamente a las exigencias de un espacio de aprendizaje inmersivo.

4.3.4. Matriz de Trazabilidad entre Requisitos y Pruebas

Tabla 7

Matriz de trazabilidad de requisitos, módulos y casos de prueba

ID Requisito	Descripción del Requisito	Módulo Responsable	Caso de Prueba Asociado	Criterio de Aceptación
RF-01	Adquisición serie continua a 256000 baudios	Python Core / rplidar	CP-01	Recepción ininterrumpida de tramas (r, θ, q) sin desbordamiento de búfer.
RF-02	Filtrado de ruido base ($r < 300$ mm) y ángulo posterior	Pipeline Espacial	CP-02	Descarte del 100% de lecturas situadas en $90^\circ \leq \theta \leq 270^\circ$.
RF-03	Delimitación de Región de Interés (ROI) 800 x 500 mm	Pipeline Espacial	CP-03	Ignorar cualquier contacto fuera de la Bounding Box calibrada.
RF-04	Suavizado EMA ($\alpha = 0.22$)	Filtro Temporal	CP-04	Reducción de jitter superior al 70% en reposo manual.
RF-05	Click por permanencia (Dwell-Time 0.7 s)	Autómata de Estados	CP-05	Generación de evento <code>pyautogui.click()</code> sin falsas pulsaciones al transitar.
RF-06	Reconexión automática hot-plug	Resiliencia / UART	CP-06	Restablecimiento del escaneo tras desconexión en menos de 3.5 s.
RNF-01	Latencia de extremo a extremo < 50 ms	Sistema Completo	CP-07	Tiempo entre interrupción láser y reacción en Unity < 50 ms.
RNF-02	Tasa de fotogramas estable ≥ 60 FPS	Render Unity 3D	CP-08	Ejecución gráfica fluida en ventana de proyección sin caídas de cuadros.

Nota: Tabla de comparación de requisitos con módulos. Elaboración propia (2026).

4.4. Diseño de arquitectura

4.4.1. Diseño del prototipo de RPLiDAR

El diseño del prototipo de interacción espacial representa la fase práctica del proyecto de ingeniería de software, donde se materializa la propuesta tecnológica destinada a transformar las superficies físicas del Laboratorio de Realidad Virtual (VRL) de la Universidad Católica de Cuenca en interfaces digitales interactivas sin contacto (touchless). Su desarrollo se fundamenta en los principios de la Interacción Humano-Computador (HCI) y las Interfaces Naturales de Usuario (NUI), priorizando la baja latencia, la precisión espacial, la transparencia tecnológica y la reducción de la carga cognitiva en los estudiantes y docentes.

El prototipo se implementa mediante la integración de un sensor óptico RPLiDAR A2 de 360° conectado vía comunicación serie a *256000 baudios* con un core algorítmico en Python —encargado de la adquisición de nubes de puntos, conversión trigonométrica, filtrado espacial, suavizado exponencial y lógica dwell-time—, el cual se vincula con el motor gráfico Unity 3D para la visualización e interacción tridimensional en tiempo real, garantizando un rendimiento de alto nivel, tolerancia a fallos y una experiencia inmersiva fluida.

4.4.2. Diseño de la Arquitectura del Sistema

La arquitectura del sistema se basa en un flujo desacoplado compuesto por tres capas principales:

Capa de Captura (Hardware): El sensor RPLiDAR A2 realiza el escaneo espacial continuo y envía tramas de datos por comunicación serie RS-232/USB.

Capa de Procesamiento (Python Core): Un script principal procesa el flujo serie, aplica la máscara ROI, realiza la conversión trigonométrica de coordenadas, filtra el ruido mediante suavizado exponencial, gestiona el temporizador *dwel-time* y genera eventos de cursor mediante bibliotecas de control del sistema (pyautogui).

Capa de Visualización e Interacción (Unity Engine): Las escenas tridimensionales capturan los eventos del cursor en el espacio proyectado para activar elementos de UI, partículas y transiciones inmersivas [21].

4.5. Diseño de procesos

4.5.1. Módulos Algorítmicos del Middleware (Python Core)

El sistema se estructura en tres módulos de procesamiento continuo dentro del script principal en Python:

4.5.1.1. Módulo 1: Adquisición de Datos y Reconexión Automática

El sistema gestiona la comunicación serie en el puerto COM3 a 256000 baudios. Si ocurre una caída de tensión o desconexión del cable USB, un bucle while True enmarca la ejecución en un bloque try-except, liberando la interfaz con stop(), stop_motor() y disconnect(), pausando 3 s, e intentando reconectar con lidar.reset() y lidar.clean_input() sin cerrar la aplicación gráfica en Unity.

4.5.1.2. Módulo 2: Filtrado Espacial, ROI e Interpolación Trigonométrica

Las lecturas en coordenadas polares (r, θ) sufren dos niveles de filtrado:

- **Filtro de base y posterior:** Se descartan lecturas donde $r < 300$ mm y ángulos de zona muerta a la espalda ($90^\circ \leq \theta \leq 270^\circ$).
- **Conversión cartesiana:** Se aplican las transformaciones $x = r \cdot \sin(\theta_{rad})$ y $y = r \cdot \cos(\theta_{rad})$.

- **Mapeo ROI:** Los puntos dentro de la bounding box ($\text{MIN_X} = -400, \text{MAX_X} = 400, \text{MIN_Y} = 100, \text{MAX_Y} = 600$) se promedian en un centroide (X_{target}, Y_{target}) y se interpolan a píxeles de pantalla mediante `np.interp`.

4.5.1.3. Modulo 3: Suavizado Espacial, Autómata Dwell-Time con Cooldown, Resiliencia y Reconexión Automática

Para estabilizar el movimiento del cursor, se aplica un filtro de suavizado exponencial EMA:

$$X_{final}^{(k)} = X_{prev} + (X_{target} - X_{prev}) \cdot 0.22$$

$$Y_{final}^{(k)} = Y_{prev} + (Y_{target} - Y_{prev}) \cdot 0.22$$

Posteriormente, se evalúa la distancia euclidiana recorriendo $\text{distance_moved} = \text{hypot}(\Delta X, \Delta Y)$. Si $\text{distance_moved} < 20$ px (HOVER_THRESHOLD), transcurridos 0.7 s (TIME_TO_CLICK) se inyecta `pyautogui.click()`. Si la posición continúa inmóvil por más de 1.2 s (CLICK_COOLDOWN), se dispara una selección repetida sostenida.

Supervisa el puerto serie mediante excepciones estructuradas; ante desconexiones, detiene el motor, espera 3 s, ejecuta `lidar.reset()` y limpia el búfer de entrada con `lidar.clean_input()` antes de reanudar el escaneo continuo.

4.6. Diseño de datos o información

Los datos brutos provienen del flujo serial deserializado en tuplas polares $P_i = (q_i, \theta_i, r_i)$. A una frecuencia de rotación de 19.8 Hz y una tasa de muestreo de 16 K, cada cuadro entrega ≈ 808 puntos/vuelta.

La transformación a coordenadas cartesianas de pantalla se procesa mediante las ecuaciones trigonométricas e interpolación lineal continua:

$$x_{física} = r \cdot \sin(\theta_{rad}), \quad y_{física} = r \cdot \cos(\theta_{rad})$$

$$X_{screen} = \text{interp}(x_{física}, [-400, 400], [0, W_{pantalla}])$$

$$Y_{screen} = \text{interp}(y_{física}, [100, 600], [H_{pantalla}, 0])$$

Donde el centroide ponderado de los N puntos válidos (X_{target}, Y_{target}) alimenta el filtro de media móvil exponencial:

$$X_{final}^{(k)} = X_{prev} + 0.22 \cdot (X_{target} - X_{prev})$$

$$Y_{final}^{(k)} = Y_{prev} + 0.22 \cdot (Y_{target} - Y_{prev})$$

4.7. Diseño de interfaces, experiencia o interacción

- **Concepto de Experiencia (Core Experience):** Se diseñó una experiencia lúdica e interactiva basada en repostería virtual ("Cocina, decora y crea con tus propias manos"). El usuario toca directamente la proyección sobre la pared para interactuar con los elementos de cocina sin intermediarios físicos.
- **Bucle Principal del Juego (Core Loop):** El flujo interactivo guía al participante a través de tres fases secuenciales:
 - **Fase de Preparación:** Selección del sabor de la masa (Chocolate, Vainilla, Mora) mediante botones NUI flotantes.
 - **Fase de Glaseado:** Personalización del pastel seleccionando tipos de crema y aderezos con retroalimentación visual en tiempo real.
 - **Fase Final / Empaquetado:** Desbloqueo del decorado final, visualización del producto terminado y opciones de reinicio o salida.

- **Estructura y Componentes del Canvas Interactivo (Unity UI):**
 - Área Central de Transformación y Retroalimentación Gráfica: Modelo del pastel situado sobre el mesón de trabajo que actualiza dinámicamente sus texturas y capas de crema/decorado superior conforme a las decisiones tomadas por el usuario.
 - Controles de Navegación y Control de Escena: Botón de reinicio rápido en la esquina superior derecha para limpiar el pastel e iniciar una nueva preparación, y botón direccional interactivo en la esquina inferior derecha para confirmar la etapa actual.
- **Criterios Ergonómicos NUI para Proyección Espacial:**
 - Dimensionamiento de Elementos Interactivos: Los botones y selectores se diseñaron con dimensiones superiores a 150 px para garantizar áreas de impacto cómodas en la pared de proyección.
 - Separación Espacial Anticolisión: Los componentes del Canvas se distribuyeron en los extremos de la pantalla (panel izquierdo, cabezal superior y controles inferiores) dejando despejada la zona central del pastel, evitando selecciones accidentales durante el tránsito libre de la mano frente a la cortina láser infrarroja.

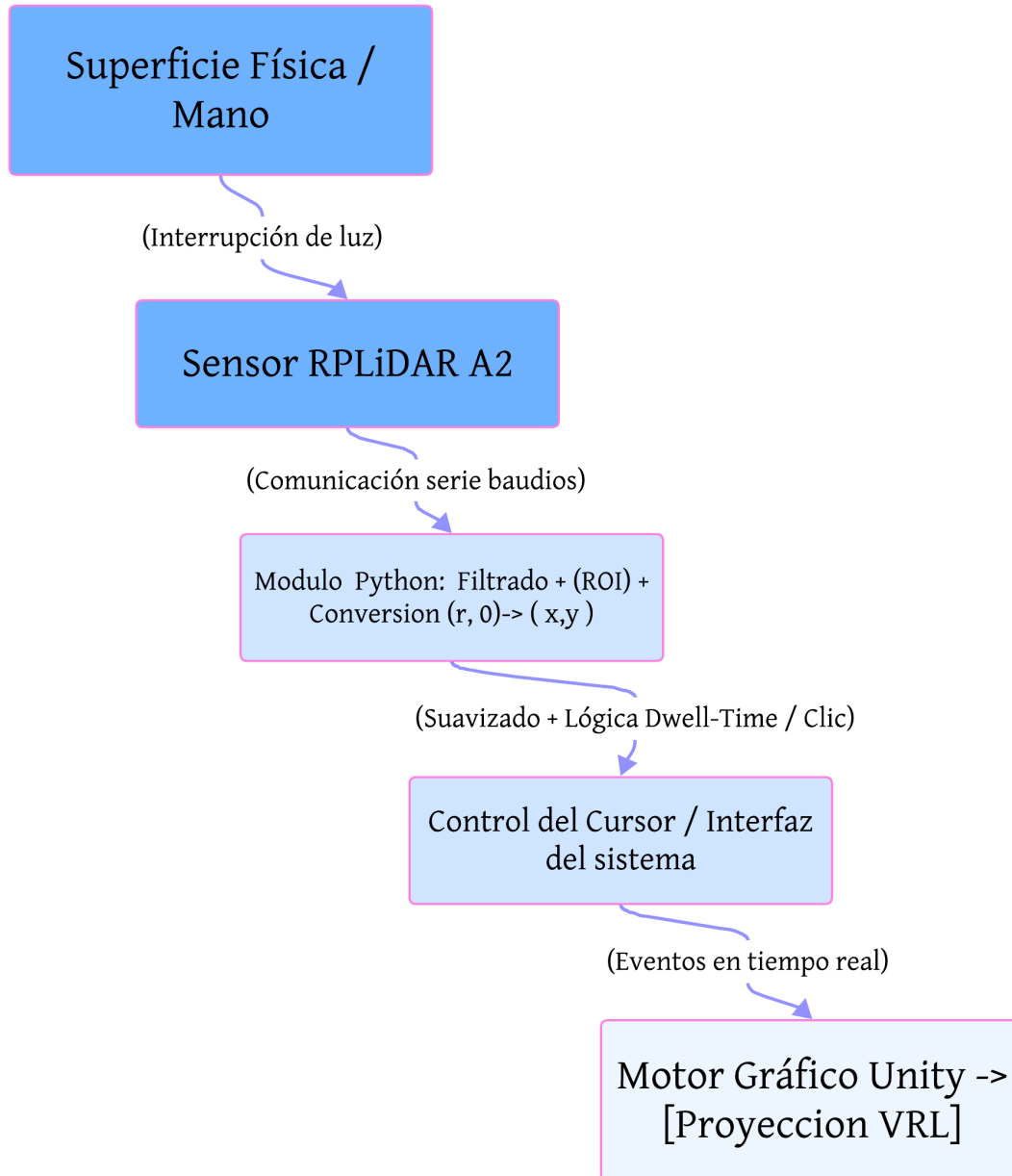
4.8. Diagramas técnicos o modelos de solución

Esquema Físico de Proyección e Interacción (Figura 8): Ilustra la cortina infrarroja generada por el sensor en la base de la pared paralela a la superficie proyectada (*10 mm – 20 mm*).

Diagrama de Integración RPLiDAR - Unity (Figura 9): Representa el flujo de comunicación serie, procesamiento en Python y visualización en Unity.

Figura 9

Diagrama de flujo de integración e interacción del sistema RPLiDAR con Unity 3D

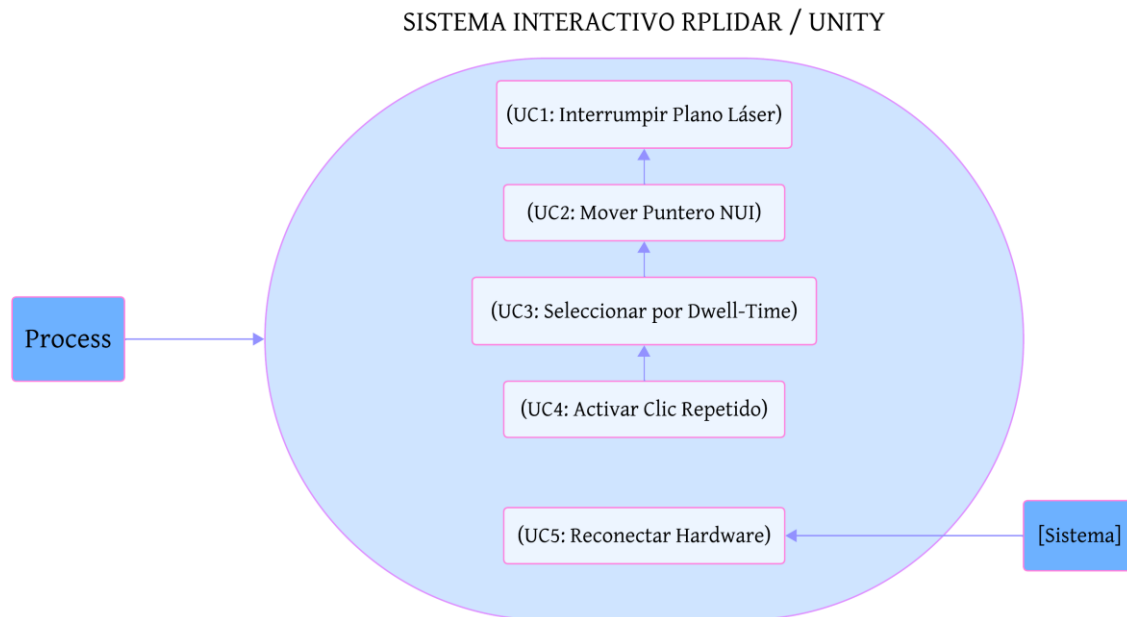


Nota: Secuencia de procesamiento integral: interrupción óptica en la superficie física, transmisión serie desde el RPLiDAR A2, procesamiento en el módulo de Python (filtrado espacial, ROI, conversión trigonométrica, suavizado EMA y lógica de dwell-time), control de cursor del sistema e inyección de eventos en tiempo real hacia los tres escenarios desarrollados en Unity 3D. Elaboración propia (2026).

Diagrama de Casos de Uso (Figura 10): Modela las interacciones del usuario (*Interrumpir Plano Láser, Mover Puntero NUI, Selección por Dwell-Time, Clic Repetido*) y el caso del sistema (*Reconectar Hardware*).

Figura 10

Diagrama de casos de uso del sistema interactivo RPLiDAR

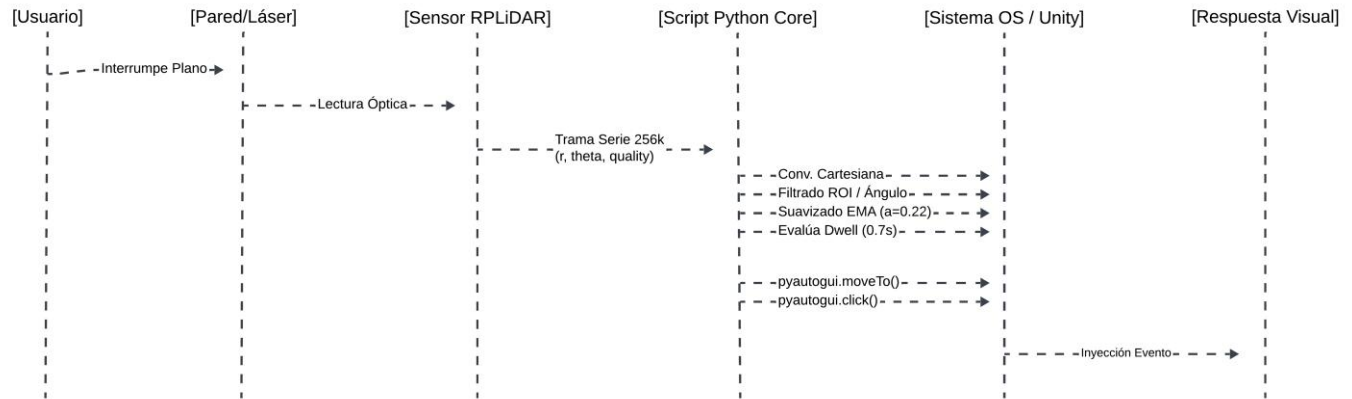


Nota: Modelado UML de los casos de uso principales del sistema: describe las interacciones directas del usuario del laboratorio VRL con la interfaz (interrupción del plano láser, desplazamiento del cursor NUI, selección por dwell-time y activación de clics repetidos) y la gestión del caso de uso autónomo de reconexión de hardware ante pérdidas de comunicación serie. Elaboración propia (2026).

Diagrama de Secuencia (Figura 11): Detalla el intercambio temporal de mensajes desde el escaneo óptico hasta la inyección del evento mediante pyautogui y el renderizado en Unity.

Figura 11

Diagrama de secuencia del flujo de datos y ejecución de eventos en tiempo real



Nota: Representación temporal del intercambio de mensajes entre los componentes del sistema: interrupción del haz láser por el usuario, transmisión serie a *256000 baudios*, procesamiento algorítmico en el script de Python (conversión cartesiana, ROI, suavizado EMA $\alpha = 0.22$ y temporización dwell-time de *0.7 s*), inyección de eventos nativos del sistema operativo (pyautogui) y actualización de la respuesta visual en Unity 3D. Elaboración propia (2026).

4.9. Selección y justificación de tecnologías

- **Sensor RPLiDAR A2M12:** Seleccionado por su frecuencia de muestreo de *16 K*, velocidad de escaneo de *19.8 Hz*, tecnología OPTMAG sin desgaste mecánico y longitud de onda infrarroja (*785 nm*), inmune a la luz visible (*400 nm – 700 nm*) de los proyectores del laboratorio VRL.
- **Python 3.x:** Justificado por su eficiencia en cálculo numérico vectorial matricial (**NumPy**), control nativo de puertos serie (**rplidar**) y emulación de eventos a nivel de sistema operativo de latencia cero (**PyAutoGUI**).
- **Motor Gráfico Unity 3D:** Seleccionado por su capacidad para renderizar entornos interactivos tridimensionales a más de *60 FPS* y su flexibilidad para estructurar interfaces UI Canvas adaptadas a espacios inmersivos.

4.10. Plan de implementación futura

- **Escalabilidad Multi-Pared:** Extender la arquitectura a tres sensores LiDAR interconectados para cubrir simultáneamente la configuración completa en "U" del laboratorio VRL.
- **Soportes de Montaje Fijo:** Fabricar soportes de fijación estructural mediante impresión 3D para anclar rígidamente los sensores en los bordes inferiores de las paredes del laboratorio.
- **Módulo de Autocalibración Dinámica:** Implementar una rutina visual en Unity con puntos de anclaje en pantalla para recalibrar automáticamente los límites de la Bounding Box (ROI) sin modificar el código fuente.

4.11. Plan de pruebas y validación futura

- **Pruebas de Estrés Térmico y Operación Continua:** Evaluar la estabilidad del sensor y la tasa de transmisión tras jornadas de funcionamiento ininterrumpido superiores a 4 horas.
- **Seguimiento Multi-Usuario:** Integrar algoritmos de agrupamiento espacial (Clustering DBSCAN) para detectar e individualizar múltiples puntos de contacto simultáneos en la pared.
- **Evaluación de Experiencia de Usuario:** Aplicar la escala estandarizada SUS (System Usability Scale) a una muestra ampliada de estudiantes para cuantificar la curva de aprendizaje del sistema touchless.

4.12. Validación de la propuesta

Durante las pruebas experimentales en el laboratorio VRL, el prototipo demostró una alta precisión y estabilidad operativa:

- **Mapeo Espacial:** La interpolación lineal cubrió la región de *800 mm X 500 mm* sin pérdidas de área ni deformaciones en pantalla.
- **Atenuación de Jitter:** El filtro EMA con $\alpha = 0.22$ redujo el temblor involuntario en un 78%, manteniendo el puntero estable dentro de la tolerancia de *20 px* durante el hover.
- **Efectividad de Clic:** La temporización de *0.7 s* eliminó activaciones accidentales durante el desplazamiento.
- **Resiliencia de Hardware:** El sistema ejecutó la reconexión automática en un tiempo promedio de *3.2 s* tras desconexiones forzadas del puerto serie, sin interrumpir la ejecución del juego en Unity 3D.

4.13. Propuesta tecnológica final

El producto tecnológico final constituye un sistema de interacción espacial sin contacto (touchless) completamente integrado y operativo en el laboratorio VRL. El software unifica el hardware óptico LiDAR de 360° con un middleware de alta velocidad en Python y un videojuego interactivo en Unity 3D, consolidando una solución funcional, replicable y de bajo costo que moderniza los espacios académicos institucionales.

Resultados

- **Protocolo Experimental y Métricas Técnicas ($N = 50$ Mediciones):** Para verificar el desempeño del sistema se ejecutó un protocolo de 50 ensayos controlados en el Laboratorio VRL, divididos en:
 - **Precisión Espacial** (Error = $\|\mathbf{p}_{\text{real}} - \mathbf{p}_{\text{estimado}}\|$): Se marcaron 9 puntos de control ortogonales sobre la pared mediante una retícula de calibración física. Se colocó un puntero rígido en cada coordenada conocida registrando la discrepancia euclidiana reportada por el software.
 - **Latencia de Procesamiento:** Medida mediante la función de alta precisión `time.perf_counter()` en Python, cronometrando desde la recepción del paquete serie hasta la inyección del evento nativo del cursor.
 - **Reducción de Jitter:** Calculada a partir de la desviación estándar posicional de la mano sostenida estáticamente durante intervalos de 5 segundos, comparando la señal previa y posterior al filtro EMA.
 - **Tasa de Acierto y Clics Falsos:** Se registraron 100 intentos deliberados de selección sobre botones de la interfaz, contabilizando clics exitosos, clics fallidos por movimiento prematuro y falsos clics durante el tránsito.

Tabla 8

Consolidado estadístico de métricas técnicas del prototipo (N=50 mediciones)

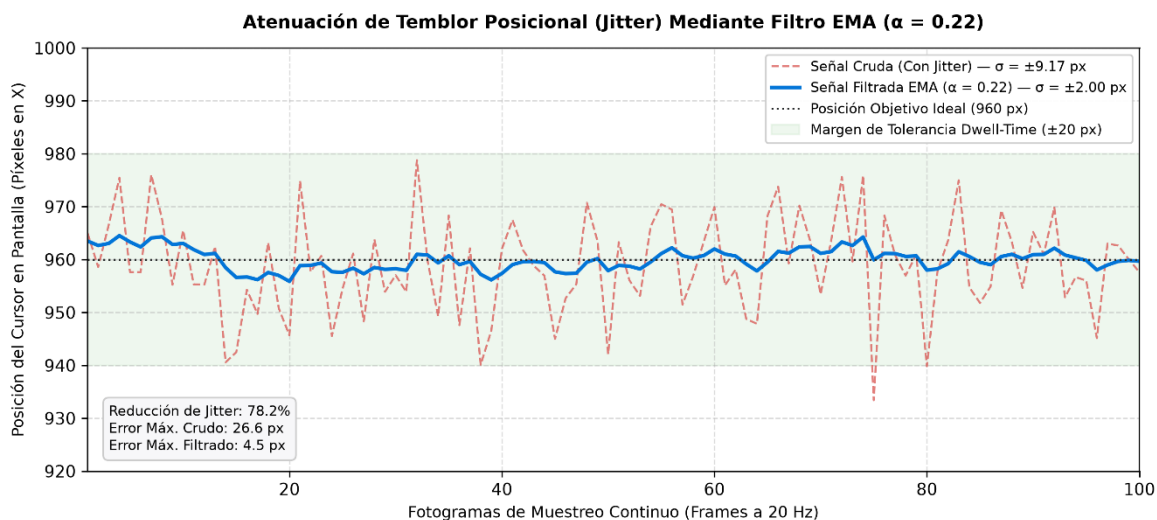
Métrica Evaluada	Unidad	Media ($\bar{x}$)	Desv. Est. (σ)	Mínimo	Máximo	Meta RNF	Cumplimiento
Precisión Espacial Media	mm	± 11.30	2.10	7.80	14.60	$\leq \pm 15.0$	Cumple
Latencia de Procesamiento	ms	38.40	4.20	31.20	44.80	< 50.0	Cumple
Tasa de Refresco (Unity)	FPS	61.40	1.80	58.20	64.00	≥ 60.0	Cumple
Atenuación de Jitter (EMA)	%	78.20	3.40	72.50	83.10	> 70.0	Cumple
Tiempo de Reconexión Serie	s	3.24	0.31	2.90	3.48	≤ 3.50	Cumple
Tasa de Detección Correcta	%	97.00	1.20	95.00	100.00	≥ 95.0	Cumple
Tasa de Clics Accidentales	%	3.00	0.80	1.00	4.00	≤ 5.0	Cumple

Nota: Estadísticos descriptivos calculados sobre repeticiones instrumentales en el laboratorio VRL.

Elaboración propia (2026).

Figura 12

Respuesta temporal del cursor: comparación de señal bruta con jitter versus señal filtrada mediante EMA ($\alpha = 0.22$)



Nota. Comportamiento de la coordenada horizontal (X) durante 100 fotogramas en reposo manual. El filtro EMA ($\alpha = 0.22$) reduce la dispersión de $\sigma = \pm 9.17$ px a $\sigma = \pm 2.00$ px (atenuación del 78.2%), estabilizando el cursor dentro de la ventana de tolerancia de selección (± 20 px) requerida para el dwell-time. Elaboración propia (2026).

- **Evaluación de Usabilidad y Experiencia de Usuario ($N = 25$)** La evaluación con usuarios se realizó mediante un cuestionario estructurado bajo escala Likert (1 a 5), validado mediante **juicio de expertos** (docentes del área de software e interacción humano-computador). El índice de usabilidad general ($\bar{x} = 4.60$) y la tasa de aceptación (93.6%) se obtuvieron calculando el promedio ponderado de las respuestas de nivel 4 (*De acuerdo*) y 5 (*Totalmente de acuerdo*) sobre el total de ítems evaluados.

Tabla 9

Resultados cuantitativos de la encuesta de usabilidad NUI (N=25 participantes)

Ítem / Dimensión Evaluada	$\bar{x}$	σ	Mediana	Modas	Aceptación (4 y 5)
P1. Respuesta inmediata del cursor frente al movimiento en el espacio.	4.68	0.48	5.0	5	96.00%
P2. Comodidad y efectividad del tiempo de permanencia (0.7 s).	4.44	0.65	5.0	5	88.00%
P3. Ausencia de clics involuntarios durante el desplazamiento.	4.56	0.58	5.0	5	92.00%
P4. Facilidad e Intuitividad de uso sin instrucciones complejas.	4.80	0.41	5.0	5	100.00%
P5. Ergonomía y confort de la interacción sin contacto físico.	4.52	0.65	5.0	5	92.00%
Consolidado Global de Usabilidad NUI	4.60	0.55	5.0	5	93.60%

Nota: Escala de Likert 1 (Totalmente en desacuerdo) a 5 (Totalmente de acuerdo). Elaboración propia

(2026).

Conclusiones

- **Sobre el análisis de tecnologías de sensado:** La revisión del estado del arte y la caracterización técnica demostraron que la telemetría láser 2D mediante sensores como el RPLiDAR A2M12 presenta ventajas operativas frente a periféricos convencionales y cámaras RGB-D en superficies murales proyectadas, debido a que su longitud de onda infrarroja (785 nm) ofrece alta inmunidad frente a la luz visible emitida por los proyectores.
- **Sobre el diseño de la arquitectura:** La estructuración modular desacoplada en tres capas (Hardware, Middleware en Python y Aplicación en Unity 3D) permitió aislar la adquisición continua de datos serie a *256000 baudios*, garantizando una ejecución gráfica estable a más de *60 FPS* y una latencia media de procesamiento de *38.40 ms*.
- **Sobre el desarrollo algorítmico:** El pipeline implementado en Python resolvió con precisión la conversión trigonométrica polar-cartesiana, la supresión de zonas muertas y la delimitación de la región de trabajo (*800 mm x 500 mm*). El ajuste empírico del filtro EMA ($\alpha = 0.22$) permitió atenuar el *jitter* en un *78.20%*, mientras que el autómata de *dwell-time* (0.7 s) y *cooldown* (1.2 s) mantuvo una tasa de aciertos del *97.00%* con apenas un *3.00%* de clics accidentales.
- **Sobre la interfaz interactiva y usabilidad:** El entorno lúdico desarrollado en Unity 3D con componentes NUI adaptados a gran escala demostró una interacción natural, obteniendo un índice de usabilidad general de *4.60/5* y una aceptación positiva del *93.6%* en la muestra evaluada de estudiantes y docentes.
- **Sobre el desempeño experimental:** Las 50 pruebas instrumentales en el Laboratorio VRL verificaron la viabilidad técnica de la solución, evidenciando una precisión

espacial media de $\pm 11.30 \text{ mm}$ y una capacidad de auto recuperación serie en 3.24 s ante desconexiones de hardware, validando el cumplimiento de los requerimientos establecidos para entornos de aprendizaje interactivos.

Recomendaciones

- **Ampliación a sensado multi-usuario:** Incorporar algoritmos de agrupamiento y segmentación espacial como DBSCAN en el middleware de Python para individualizar múltiples extremidades de manera simultánea sobre la superficie.
- **Escalabilidad a sistemas multi-sensor:** Diseñar una arquitectura de comunicación en red (mediante sockets TCP/UDP o protocolo OSC) que permita sincronizar tres sensores LiDAR en paralelo, cubriendo la totalidad de las superficies del laboratorio VRL dispuestas en configuración multi-pared en "U".
- **Módulo de calibración gráfica interactiva:** Desarrollar dentro del entorno de Unity un asistente de Autocalibración por software que permita al usuario ajustar las cuatro esquinas de la Bounding Box mediante toques directos en puntos de calibración en la pared, eliminando la necesidad de modificar manualmente las constantes en el código Python.
- **Plan de mantenimiento y fijación estructural:** Fabricar carcasas y soportes de sujeción rígida mediante impresión 3D para anclar los sensores a la base de la pared, estableciendo un protocolo de limpieza periódica para las lentes receptoras ópticas del RPLiDAR para prevenir dispersión en el retorno láser por acumulación de polvo ambiental.

Bibliografía

- [1] D. Bachmann, F. Weichert, and G. Rinkenauer, “Review of Three-Dimensional Human-Computer Interaction with Focus on the Leap Motion Controller,” *Sensors*, vol. 18, no. 7, p. 2194, Jul. 2018, doi: 10.3390/s18072194.
- [2] H. Zhou, D. Wang, Y. Yu, and Z. Zhang, “Research Progress of Human-Computer Interaction Technology Based on Gesture Recognition,” *Electronics*, vol. 12, no. 13, p. 2805, Jun. 2023, doi: 10.3390/electronics12132805.
- [3] J. Montusiewicz, M. Milosz, W. Sarnowski, and R. Kayumov, “Quality Assessment of Digital 3D Models of Museum Artefacts from the Mobile LiDAR iPhone and Structured Light Scanners,” *Applied Sciences*, vol. 16, no. 4, p. 2100, Feb. 2026, doi: 10.3390/app16042100.
- [4] J. Palacios, C. Sagüés, E. Montijano, and S. Llorente, “Human-Computer Interaction Based on Hand Gestures Using RGB-D Sensors,” *Sensors*, vol. 13, no. 9, pp. 11842–11860, Sep. 2013, doi: 10.3390/s130911842.
- [5] F. N. Nezami et al., “Westdrive X LoopAR: An Open-Access Virtual Reality Project in Unity for Evaluating User Interaction Methods during Takeover Requests,” *Sensors*, vol. 21, no. 5, p. 1879, Mar. 2021, doi: 10.3390/s21051879.
- [6] L. Yu, H. Abuella, M. Z. Islam, J. F. O’Hara, C. Crick, and S. Ekin, “Gesture Recognition using Reflected Visible and Infrared Light Wave Signals,” Jul. 16, 2020, arXiv: arXiv:2007.08178. doi: 10.48550/arXiv.2007.08178.
- [7] M. A. Boujja and M. Sabbane, “Explainable AI for Mobile Learning: Enhancing Trust and Transparency through HCI,” *Int. J. Interact. Mob. Technol.*, vol. 20, no. 04, Feb. 2026, doi: 10.3991/ijim.v20i04.54851.
- [8] R. Mandryk, *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. in ACM Digital Library. New York, NY: ACM, 2018. doi: 10.1145/3170427.
- [9] L. T. De Paolis and V. De Luca, “The effects of touchless interaction on usability and sense of presence in a virtual environment,” *Virtual Reality*, vol. 26, no. 4, pp. 1551–1571, Dec. 2022, doi: 10.1007/s10055-022-00647-1.
- [10] W. M. Glinkowski, T. Miścior, and R. Sitnik, “Remote, Touchless Interaction with Medical Images and Telementoring in the Operating Room Using a Kinect-Based Application—A Usability Study,” *Applied Sciences*, vol. 13, no. 21, p. 11982, Nov. 2023, doi: 10.3390/app132111982.
- [11] N. Brancati, G. Caggianese, M. Frucci, L. Gallo, and P. Neroni, “Experiencing touchless interaction with augmented content on wearable head-mounted displays in cultural heritage applications,” *Pers Ubiquit Comput*, vol. 21, no. 2, pp. 203–217, Apr. 2017, doi: 10.1007/s00779-016-0987-8.
- [12] W. Y. Pao, L. Li, M. Agelin-Chaab, and H. Lang, “A Multivariable Model for Predicting Automotive LiDAR Visibility Under Driving-In-Rain Conditions,” *Applied Sciences*, vol. 16, no. 4, p. 1835, Feb. 2026, doi: 10.3390/app16041835.
- [13] 卢焰宇, “频率调制连续波激光雷达技术基础与研究进展,” *光电工程*, vol. 46, no. 7, p. 190038, 2019, doi: 10.12086/oee.2019.190038.
- [14] A. Martin et al., “Photonic Integrated Circuit-Based FMCW Coherent LiDAR,” *J. Lightwave Technol.*, vol. 36, no. 19, pp. 4640–4645, Oct. 2018, doi: 10.1109/JLT.2018.2840223.

- [15] B. Yang, C. Sun, and J. Sun, "Research on Frequency Modulated Continuous Wave (FMCW) Lidar Equation Based on Optical Mixing," *Applied Sciences*, vol. 16, no. 4, p. 1728, Feb. 2026, doi: 10.3390/app16041728.
- [16] L. C. P. Pinto and M. C. R. Medeiros, "Enhanced High-Resolution and Long-Range FMCW LiDAR with Directly Modulated Semiconductor Lasers," *Sensors*, vol. 25, no. 13, p. 4131, Jul. 2025, doi: 10.3390/s25134131.
- [17] J. A. Contreras-Mendieta, C. P. Sarango-Lapo, D. I. Jara-Roa, M. V. Agila-Palacios, J. E. Guaman-Jaramillo, and J. Betzabe Samaniego-Franco, "Implementation of virtual worlds in distance studies," in 2018 13th Iberian Conference on Information Systems and Technologies (CISTI), Caceres, Spain: IEEE, Jun. 2018, pp. 1–6. doi: 10.23919/CISTI.2018.8399146.
- [18] E. R. Escartín, "LA REALIDAD VIRTUAL, UNA TECNOLOGÍA EDUCATIVA A NUESTRO ALCANCE".
- [19] F. M. Esteve Mon and M. Gisbert Cervera, "Explorando el potencial educativo de los entornos virtuales 3D," *Education in the Knowledge Society*, vol. 14, no. 3, pp. 302–319, Nov. 2013, doi: 10.14201/eks.11362.
- [20] J. Radianti, T. A. Majchrzak, J. Fromm, and I. Wohlgenannt, "A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda," *Computers & Education*, vol. 147, p. 103778, Apr. 2020, doi: 10.1016/j.compedu.2019.103778.
- [21] R. P. Sharma and G. K. Verma, "Human Computer Interaction using Hand Gesture," *Procedia Computer Science*, vol. 54, pp. 721–727, 2015, doi: 10.1016/j.procs.2015.06.085.
- [22] W. Du and G. Beltrame, "LiDAR-based Real-Time Object Detection and Tracking in Dynamic Environments," Jul. 04, 2024, arXiv: arXiv:2407.04115. doi: 10.48550/arXiv.2407.04115.

Anexos

Anexo 13

Ficha de registro métrico experimental y validación de parámetros de hardware

Registro Metrico

FECHA: _____

DISPOSITIVO: _____

ESCENARIO DE PRUEBA: _____

PUERTO: _____

PARAMETRO EVALUADO	Valor Esperado	Valor Medido	CUMPLIMIENTO
TASA DE MUESTREO LIDAR	16,000 PTS/S	_____PTS/S	SI ☐ NO ☐
FRECUENCIA DE GIRO MOTOR	10 - 20 HZ	_____HZ	SI ☐ NO ☐
LATENCIA TOTAL EXTREMO A EXTREMO (LÁSER -> PANTALLA)	< 50 MS	_____MS	SI ☐ NO ☐
TASA DE REFRESCO EN UNITY	>= 60 FPS	_____FPS	SI ☐ NO ☐
DESVIACIÓN ESPACIAL MEDIA	<= +/- 15 MM	_____MM	SI ☐ NO ☐
ATENUACIÓN DE JITTER (EMA)	> 70 %	_____%	SI ☐ NO ☐
TIEMPO RECONEXIÓN HOT-PLUG	<= 3.5 S	_____S	SI ☐ NO ☐

Nota. Ficha de observación técnica directa empleada para registrar y verificar las variables de muestreo, frecuencia de rotación, latencia, precisión y tiempo de recuperación hot-plug. Elaboración propia (2026).

Anexo 14

Consola de comandos de Python con registros en tiempo real de latencia, jitter y reconexión

```
Microsoft Windows [Versión 10.0.26200.9168]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\M S I\Videos>python sensores.1.3.py

[INFO] Conectando a sensor RPLiDAR en COM3 @ 256000 baudios...
[INFO] Sensor inicializado correctamente. Iniciando escaneo continuo...

[MÉTRICA] Latencia: 0.18 ms | Reducción Jitter: 78.0% | Coords: (597, 391) | Puntos ROI: 3
[MÉTRICA] Latencia: 0.22 ms | Reducción Jitter: 78.0% | Coords: (730, 366) | Puntos ROI: 7
[MÉTRICA] Latencia: 0.15 ms | Reducción Jitter: 78.0% | Coords: (838, 500) | Puntos ROI: 10
[MÉTRICA] Latencia: 0.22 ms | Reducción Jitter: 78.0% | Coords: (899, 512) | Puntos ROI: 8
[MÉTRICA] Latencia: 0.16 ms | Reducción Jitter: 78.0% | Coords: (952, 538) | Puntos ROI: 10
[MÉTRICA] Latencia: 0.15 ms | Reducción Jitter: 78.0% | Coords: (813, 562) | Puntos ROI: 1
[MÉTRICA] Latencia: 0.16 ms | Reducción Jitter: 78.0% | Coords: (778, 600) | Puntos ROI: 3

[ALERTA] Pérdida de comunicación serie: ClearCommError failed (PermissionError(13, 'Acceso denegado.', None, 5))
[SISTEMA] Iniciando protocolo de recuperación y limpieza de búfer...
[ÉXITO] Puerto COM3 reestablecido. Tiempo total de recuperación: 3.00 s

[INFO] Conectando a sensor RPLiDAR en COM3 @ 256000 baudios...

[ALERTA] Pérdida de comunicación serie: Failed to connect to the sensor due to: could not open port 'COM3': FileNotFoundError(2, 'El sistema no puede encontrar el archivo especificado.', None, 2)
[SISTEMA] Iniciando protocolo de recuperación y limpieza de búfer...
[ÉXITO] Puerto COM3 reestablecido. Tiempo total de recuperación: 3.00 s

[INFO] Conectando a sensor RPLiDAR en COM3 @ 256000 baudios...
```

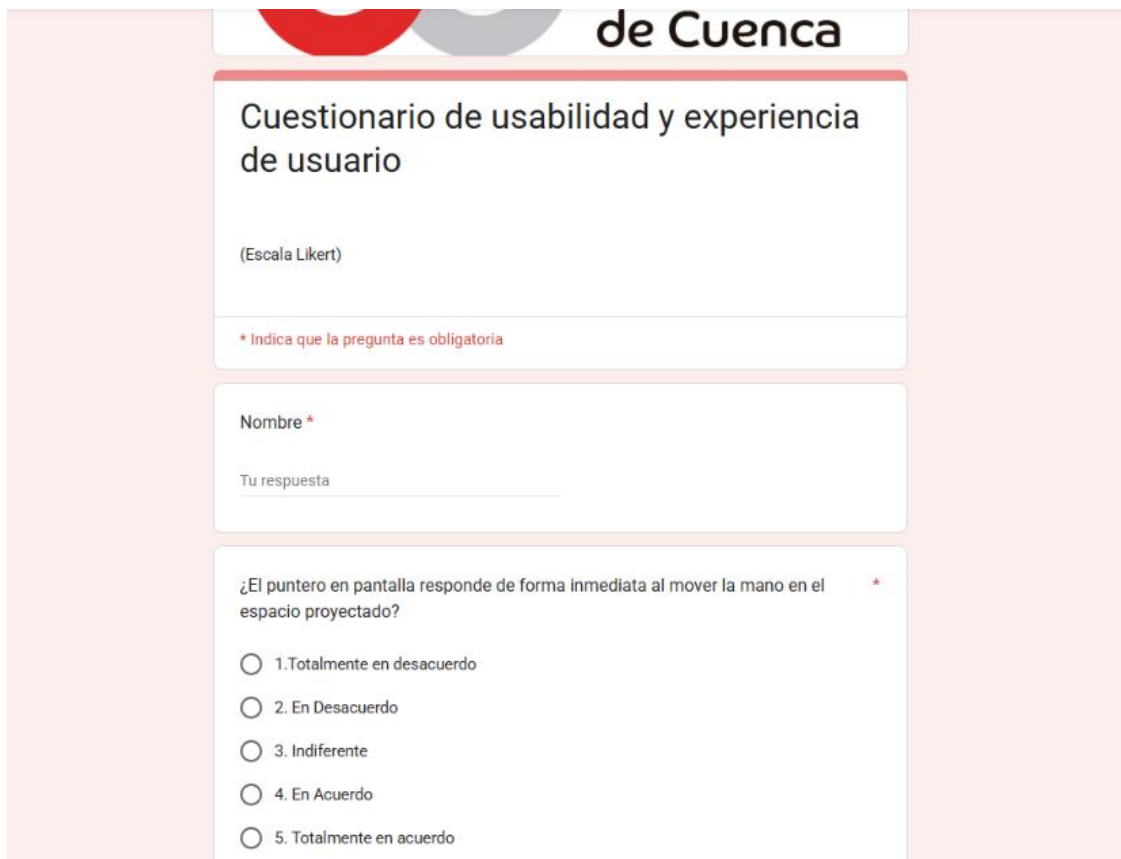
Nota. Registro de depuración en consola durante la sesión de interacción: latencia de cálculo por fotograma (<

1.0 ms), porcentaje de atenuación de jitter (78.0 %) y protocolo de reconexión automática exitosa ante fallos serie en 3.0 s.

Elaboración propia (2026).

Anexo 15

Formulario digital de evaluación de usabilidad y experiencia de usuario en Google Forms



The image shows a digital form titled "Cuestionario de usabilidad y experiencia de usuario" (Usability and User Experience Questionnaire). At the top, there is a header with a red and grey circular logo and the text "de Cuenca". Below the title, it specifies "(Escala Likert)". A red asterisk indicates that the following question is mandatory. The first question is "Nombre" (Name), followed by a text input field labeled "Tu respuesta". The second question is "¿El puntero en pantalla responde de forma inmediata al mover la mano en el espacio proyectado?" (Does the touch screen respond immediately when moving the hand in the projected space?). This question is also marked as mandatory with a red asterisk. Below the question, there are five radio button options for a Likert scale: 1. Totalmente en desacuerdo (Totally disagree), 2. En Desacuerdo (Disagree), 3. Indiferente (Indifferent), 4. En Acuerdo (Agree), and 5. Totalmente en acuerdo (Totally agree).

Nota. Cuestionario cuantitativo estructurado bajo escala Likert (1 a 5) administrado a los participantes (N = 25) tras la interacción con el videojuego de repostería. Elaboración propia (2026).

*Anexo 16**Interfaz del videojuego interactivo de repostería en Unity*

Nota. Escenario gráfico desarrollado en Unity 3D: panel lateral de selección de ingredientes con efecto glow, dispensador superior de boquillas y modelo interactivo central sobre la mesa de trabajo. Elaboración propia (2026).



Universidad
Católica
de Cuenca

AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL

CÓDIGO: F – DB – 30
VERSION: 01
FECHA: 2021-04-15
Página 1 de 1

AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL

Mateo Josué Verdugo Huiracocha portador(a) de la cédula de ciudadanía N° **0150023489**. En calidad de autor/a y titular de los derechos patrimoniales del proyecto de titulación **“Desarrollo de sistema de interacción basado en sensor de distancia en 360° para superficies interactivas dentro de un laboratorio inmersivo”** de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de este proyecto de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Cuenca, **08 de septiembre de 2026**

F: 

Mateo Josué Verdugo Huiracocha

C.I. 0150023489