



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Desarrollo de un simulador
neonatal para el entrenamiento
de habilidades en cirugía
pediátrica**

TESINA

Que para obtener el título de
Ingeniera Mecatrónica

P R E S E N T A

Daniela Alvarado Rivera

DIRECTOR DE TESINA

Dr. Fernando Pérez Escamirosa



Ciudad Universitaria, Cd. Mx., 2025



**PROTESTA UNIVERSITARIA DE INTEGRIDAD Y
HONESTIDAD ACADÉMICA Y PROFESIONAL
(Titulación con trabajo escrito)**



De conformidad con lo dispuesto en los artículos 87, fracción V, del Estatuto General, 68, primer párrafo, del Reglamento General de Estudios Universitarios y 26, fracción I, y 35 del Reglamento General de Exámenes, me comprometo en todo tiempo a honrar a la institución y a cumplir con los principios establecidos en el Código de Ética de la Universidad Nacional Autónoma de México, especialmente con los de integridad y honestidad académica.

De acuerdo con lo anterior, manifiesto que el trabajo escrito titulado DESARROLLO DE UN SIMULADOR NEONATAL PARA EL ENTRENAMIENTO DE HABILIDADES EN CIRUGIA PEDIATRICA que presenté para obtener el título de INGENIERO MECATRÓNICO es original, de mi autoría y lo realicé con el rigor metodológico exigido por mi Entidad Académica, citando las fuentes de ideas, textos, imágenes, gráficos u otro tipo de obras empleadas para su desarrollo.

En consecuencia, acepto que la falta de cumplimiento de las disposiciones reglamentarias y normativas de la Universidad, en particular las ya referidas en el Código de Ética, llevará a la nulidad de los actos de carácter académico administrativo del proceso de titulación.

DANIELA ALVARADO RIVERA
Número de cuenta: 316188465

*“Camina siempre en compañía,
verás que mil kilómetros se vuelven 100”*

-꼭씩 속았수다

Agradecimientos Institucionales

Universidad Nacional Autónoma de México, UNAM.

Facultad de Ingeniería

Instituto de Ciencias Aplicadas y Tecnología, ICAT.

Laboratorio de Sistemas Inteligentes del Departamento de Tecnologías de la Información y Procesos Educativos.

Investigación realizada gracias al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) de la UNAM IT201924. Agradezco a la DGAPA-UNAM la beca recibida

Asesor

Mi más sincero agradecimiento al Dr. Fernando Pérez Escamirosa por su apoyo, paciencia y valiosas enseñanzas a lo largo de este proyecto y durante mi estancia en el instituto. Me considero afortunada de haber encontrado en usted no solo a un excelente investigador con una profunda vocación por su área, sino también a un académico comprometido con el crecimiento de sus estudiantes. Su disposición para guiarnos y mostrarnos el mejor camino ha dejado una huella significativa en cada uno de nosotros.

Sinodales

Dr. Adrián Espinoza Bautista, Dr. Miguel Ángel Bañuelos Saucedo, M.I. Serafín Castañeda Cedeño y M.C. Luis Ochoa Bautista por todo su tiempo y aportaciones a este trabajo.

Agradecimientos Personales

En primer lugar, quiero agradecer a mis padres, quienes me han dado la oportunidad de vivir experiencias tan significativas como lo ha sido mi formación académica. Me brindaron confianza, un apoyo invaluable y su compañía en todas esas madrugadas tan complicadas. A mi madre, Guadalupe Rivera, y a mi padre, Daniel Alvarado: por absolutamente todo, este logro también es de ustedes. A mi hermana, la persona por la cual, aun sin saberlo, me he esforzado cada día desde que llegó a mi vida.

Agradezco a mis amigos Daniel M., Joshua G., Michelle O. y Héctor G. Su compañía a lo largo de mi estancia en la universidad fue invaluable. Me enseñaron lo que significa seguir, sin miedo, aquello que me apasiona; reencontrar el amor por lo que hago y dar siempre lo mejor de mí. Ustedes son la prueba de que, en el camino, siempre encontramos personas con quienes compartir sueños y que nos inspiran constantemente.

Gracias al Laboratorio de Sistemas Inteligentes: Margareli, Mariano, Esther y Guillermo. Por sus enseñanzas, por los días de risas y por las conversaciones que aligeraron la carga en los momentos más difíciles de este proyecto. En ustedes no solo encontré un gran equipo en el que se puede confiar, sino también una familia.

Mi agradecimiento también al Dr. Javier Solís O., por su apoyo constante y sus valiosas enseñanzas. Es un académico apasionado por la ingeniería, que contagia el deseo de seguir aprendiendo y descubrir todo lo que esta disciplina puede ofrecer. Sin su guía y entusiasmo, este proyecto no sería lo que es hoy.

Agradezco a la señora Eva E. por todos sus consejos y su apoyo incondicional en esta última etapa. Me ayudó a ver el valor de lo que hacía y de lo que quería hacer, incluso cuando yo no podía verlo. A Miguel S., por motivarme a seguir creciendo profesionalmente y brindarme apoyo en ese proceso.

Un especial agradecimiento a Jessica Alcántar, con quien compartí este último tramo. Aprendí y encontré en ti a una gran amiga. Sin ti, esta experiencia no habría sido igual ni tan buena.

A todos, gracias por siempre.

Contenido

Capítulo 1. Introducción	6
Problemática.....	7
Objetivo	8
Hipótesis.....	8
Antecedentes	9
La cirugía laparoscópica	9
Instrumental en la cirugía laparoscópica	10
Simuladores para cirugía laparoscópica	11
Capítulo 2. Estado del arte	13
Simulador laparoscópico de código abierto.....	13
Simulador de Cirugía Laparoscópica Pediátrica para el Análisis de Movimiento.....	14
Simulador de cirugía laparoscópica pediátrica de puerto único.....	15
Entrenador quirúrgico toracoscópico instrumentado	16
Requisitos del simulador neonatal laparoscópico.....	18
Capítulo 3. Metodología	19
Diseño electromecánico	21
Diseño de las articulaciones.....	21
Integración y conexión del sistema.....	27
Tratamiento de datos	28
Matriz de Rotación	28
Ángulos de Euler	29
Interfaz gráfica de usuario	32
Capítulo 4. Resultados	38
Capítulo 5. Conclusiones	42
Conclusiones	42
Trabajo a futuro	43
Referencias	44
Anexo A. Código de Interfaz	45
Anexo B. Código de gráficas en MATLAB	58

Capítulo 1. Introducción

De acuerdo con el *Colegio y Sociedad Mexicana de Cirugía Pediátrica*, (2024), un cirujano pediatra es quien únicamente puede tratar a niños y adolescentes, y quien cuenta con la experiencia necesaria para no solamente diagnosticar, sino también para encargarse del cuidado operatorio y postoperatorio de los pacientes con enfermedades que les impiden tener una adecuada calidad de vida a menos que sean tratados quirúrgicamente.

En el campo de la cirugía pediátrica, la capacitación y el desarrollo de habilidades quirúrgicas son fundamentales para garantizar la seguridad y el bienestar de los pacientes pediátricos. Particularmente, la cirugía neonatal laparoscópica, considerada dentro de la alta especialidad de pediatría en México, presenta desafíos únicos debido a las características de los pacientes y la complejidad de los procedimientos.

Actualmente, existe una creciente demanda de entrenamiento y evaluación objetiva de las habilidades psicomotrices y destrezas quirúrgicas en el área de la cirugía laparoscópica, impulsada por las ventajas que ofrece en comparación con la cirugía abierta. Sin embargo, la capacitación se ve limitada por factores como la falta de oportunidades para los estudiantes y residentes de la especialidad, la baja frecuencia de los procedimientos laparoscópicos, la escasez de equipos y las implicaciones éticas y legales al practicar en pacientes reales. Teniendo como resultado que algunas herramientas como los simuladores sean una solución valiosa para el entrenamiento de habilidades quirúrgicas, los cuales proporcionan un entorno seguro y controlado para el aprendizaje y la práctica de los aprendices.

En diversos estudios sobre la curva de aprendizaje de los cirujanos dentro del campo de la laparoscopia como el caso presentado por Peláez et al., (2021), se muestra que, aunque la curva de aprendizaje no cuenta con un modelo uniforme para la adquisición de habilidades quirúrgicas, el uso de los simuladores quirúrgicos para el entrenamiento de los residentes demostró una reducción en el tiempo requerido para la ejecución de tareas y ejercicios al igual que una mejora en las habilidades y destrezas laparoscópicas entre el 30-40% en los estudiantes.

Por lo anterior, el objetivo de este trabajo es presentar el desarrollo de un simulador para cirugía neonatal pediátrica, que permitirá a los residentes de esta especialidad desarrollar y mantener sus habilidades quirúrgicas y destrezas laparoscópicas durante su formación.

Problemática

De acuerdo con distintos autores, existe un constante aumento en la demanda de entrenamiento y evaluación objetiva de las habilidades psicomotrices y destrezas técnicas en cirugía laparoscópica, esto debido a las ventajas que tiene en el paciente respecto a una cirugía abierta. Dado que las habilidades requeridas para este tipo de procedimientos quirúrgicos son diferentes a las utilizadas en una cirugía abierta, ya sea por las diferencias en el equipo e instrumental utilizado o por el conocimiento previo y habilidad para aplicar ciertas técnicas, se requiere otro tipo de entrenamiento para el residente; entrenamiento que en su mayoría se ve limitado por factores externos como poca oportunidad del aprendiz para participar en este tipo de procedimientos (por falta de habilidad/experiencia), “poca” frecuencia de los procedimientos, no contar con el equipo necesario para la práctica, implicaciones legales y éticas por el entrenamiento de cirujanos con pacientes y la necesidad de ser evaluados por la observación subjetiva de cirujanos expertos.

Tomando en cuenta lo anterior, el entrenamiento quirúrgico, basado en el uso de simuladores médicos, representa una gran ventaja dado que no pone en riesgo a los pacientes pediátricos y reduce o elimina el uso de recursos sanitarios para la práctica; por lo tanto, el desarrollo de tecnología que permita el entrenamiento y evaluación de las habilidades y destrezas requeridas en cirugía laparoscópica tiene gran importancia en este campo de la medicina.

Aunque en el presente ya existen simuladores para este tipo de procedimientos laparoscópicos, estos tienen limitaciones como trabajar solamente con ciertos materiales y tareas específicas (creando la necesidad de requerir más simuladores para cubrir la demanda del entrenamiento faltante en los residentes), incluyendo que son de difícil acceso para muchos hospitales y centros de salud en México por su alto costo. En el caso de simuladores de cirugía neonatal, estos son igual o mayormente limitados, principalmente por el uso de instrumental especializado para el paciente neonato y el equipo requerido para el desarrollo del procedimiento.

De acuerdo con Gavrilovic et al., (2018), a diferencia de la cirugía abierta, el movimiento de los instrumentos en la cirugía laparoscópica sólo puede realizarse en cuatro grados de libertad (GDL): uno de traslación (avance) y tres de rotación (*roll*, *pitch* y *yaw*); si a esto añadimos el caso de la cirugía neonatal, donde el paciente es de dimensiones en extremo menores respecto a las de un paciente pediátrico habitual y cuya anatomía también es diferente, es notoria la necesidad de una mayor exigencia en la habilidad y entrenamiento quirúrgico del cirujano para poder realizar un procedimiento de este tipo, el cual tendrá mayores exigencias y limitaciones.

Objetivo

El objetivo de esta tesina es el diseño y construcción de un simulador de cirugía laparoscópica neonatal que permita la adquisición y el entrenamiento de habilidades y destrezas en cirugía pediátrica, a través de la captura y registro del movimiento de los instrumentos laparoscópicos en tareas básicas.

Hipótesis

El simulador de cirugía laparoscópica neonatal permitirá el entrenamiento, desarrollo y mantenimiento de las habilidades laparoscópicas y destrezas técnicas de los cirujanos, en un ambiente controlado para realizar distintas tareas y ejercicios. Adicionalmente, el desarrollo del simulador neonatal laparoscópico contribuirá dentro de la especialidad en cirugía pediátrica como una herramienta de aprendizaje y entrenamiento, que permita aumentar el nivel de experiencia quirúrgica de los cirujanos y residentes, con el fin de mejorar sus habilidades mínimamente invasivas, sin requerir de tecnologías de alto costo o de un aprendizaje directo con pacientes reales.

Antecedentes

La cirugía laparoscópica

Los autores Schlottmann & Sadava, (2023) señalan que los primeros registros de la cirugía de mínima invasión se remontan entre los años 460-375 A.C. en la Escuela Hipocrática y no fue sino hasta la década de 1980 que comenzó a tener un completo auge y mayor desarrollo dentro de la cirugía general hasta llegar a lo que hoy conocemos como cirugía laparoscópica. En comparación con la cirugía abierta, la cirugía laparoscópica cuenta con múltiples beneficios como los siguientes:

1. Menor pérdida hemática.
2. Reducción en el dolor posoperatorio.
3. Menor tiempo requerido en estadía hospitalaria.
4. Menor tasa de eventraciones.
5. Menor tasa de sangrado.
6. Posibilidad de exploración casi completa de la cavidad abdominal.

Siendo esta última uno de los factores destacados ya que en la práctica permite aumento de la visión de 3-5 puntos, al igual que el uso de endocámaras con diferentes ángulos que mejoran la visualización del campo quirúrgico (Figura 1).

Además de las ventajas mencionadas anteriormente, existen complicaciones, aunque menores, dentro de la cirugía laparoscópica, de las cuales cabe mencionar las siguientes:

- a) Por ingreso a la cavidad: Vasculares, lesión gastrointestinal, lesión vesical y eventración de sitio de trocar (siendo la complicación vascular y la lesión gastrointestinal la segunda y tercera causa de muerte dentro de la cirugía laparoscópica, estando en primer lugar las complicaciones anestésicas).
- b) Relacionadas con el neumoperitoneo: Embolia gaseosa (por CO₂), enfisema subcutáneo y mediastínico, hipercapnia, neumotórax, arritmias, omalgia, síndrome compartimental abdominal.
- c) Por la disección: Lesión gastrointestinal, lesión del tracto urinario.
- d) Metástasis en sitios de trocar

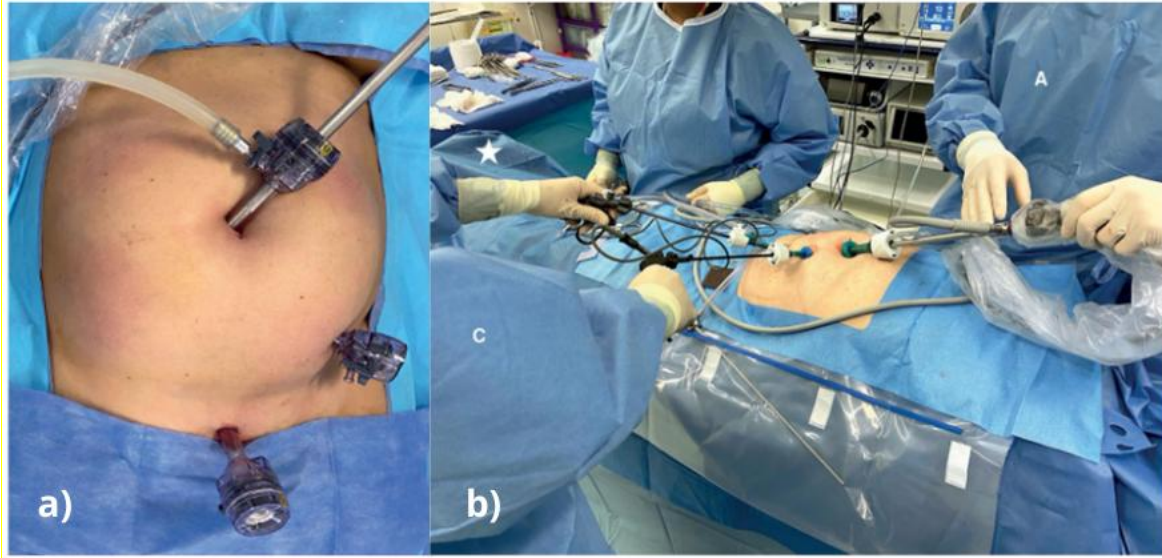


Figura 1 (a) Triangulación entre el trocar principal (umbilical) y los trocares accesorios para lograr una adecuada ergonomía. (b) Posición del equipo quirúrgico: cirujano principal (**A**), contralateral a la hernia, y del ayudante (**C**), homolateral a la hernia. Monitores ubicados hacia caudal del paciente (★). Schlottmann & Sadava (2023).

Instrumental en la cirugía laparoscópica

Dentro de la cirugía laparoscópica es necesario el uso de equipo especializado e indispensable para su desarrollo, el cual, además del requerimiento de torres de laparoscopia, monitores, neumoinsufladores, cámaras de video y fuentes de luz adecuadas para el procedimiento, utiliza instrumental quirúrgico como el que se menciona a continuación:

- Instrumental de ingreso a la cavidad abdominal: Aguja de veress, puertos de trabajo (con punta disectora, con cuchilla retráctil y con balón de anclaje)
- Instrumental de exposición: pinzas de exposición (exposición y disección, coagulación)
- Instrumental de corte: Tijeras (con punta roma, rectas, curvas, etc.)
- Instrumental de disección y energía: Requieren conectarse a energía (Hook, pinzas de disección, etc.), no requieren conectarse a energía (pinzas con hisopos).
- Instrumental de hemostasia: Clips, elementos de disección.
- Sutura: Manuales (utilizando pinzas portaagujas) y mecánicas (lineales o circulares)
- Instrumentos de fijación de materiales protésicos. Mallas
- Instrumentos de aspiración-irrigación: Cánulas.

- Separadores viscerales: Separador de Nathanson, separador de abanico, etc.
- Dispositivos de extracción: Endo catch.

La Figura 2 muestra un ejemplo del instrumental utilizado para el procedimiento quirúrgico en cirugía laparoscópica.

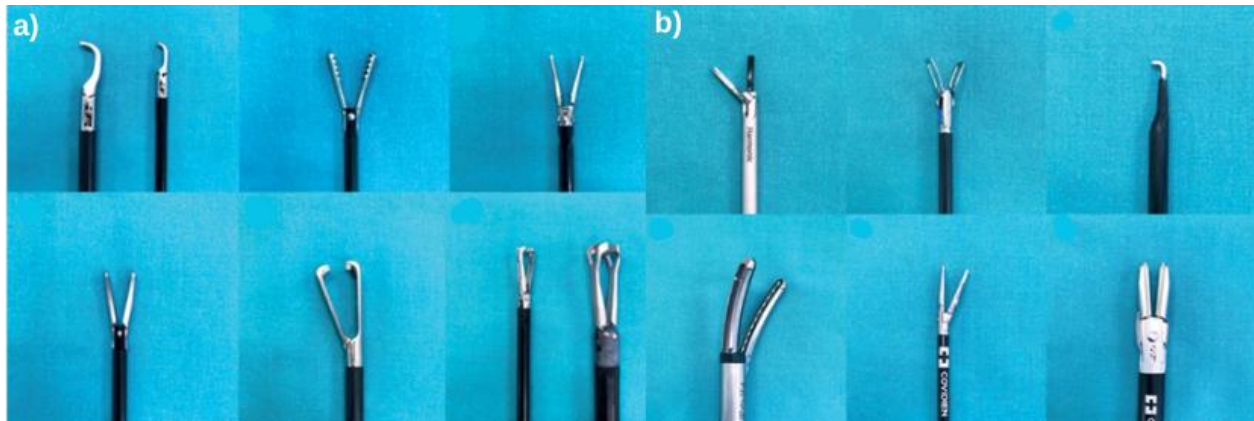


Figura 2 Resumen de instrumentos que se utilizan en la cirugía laparoscópica. a) Instrumental de disección y energía. b) Instrumental de hemostasia. Schlottmann & Sadava, (2023).

Cabe mencionar que, el instrumental laparoscópico, tendrá una variación en sus medidas de acuerdo con el procedimiento a realizar, teniendo de 34-38 [cm] de longitud para un procedimiento estándar, un rango menor para pacientes pediátricos de 20-22 [cm] y alrededor de 48[cm] de longitud para pacientes con obesidad o panículo adiposo alto, con diámetros que van desde 10[mm], 5[mm], 3[mm] hasta 2.7[mm], incluyendo que este instrumental representa un breve resumen de lo que se requiere dentro de un procedimiento laparoscópico y su resultado dependerá de la destreza o habilidad del cirujano, al igual que su nivel de experiencia.

Simuladores para cirugía laparoscópica

De acuerdo con León et al., (2015), el nivel de experiencia de los cirujanos se ve afectado por diversos factores como entrenamiento quirúrgico limitado debido a implicaciones legales, la exigencia de dominar nuevas técnicas complejas dentro de la cirugía y la falta de un ambiente completamente seguro y controlado. Dado lo anterior, el uso de la simulación permite el entrenamiento de estas habilidades y generar la experiencia suficiente para que sean llevadas a quirófano, disminuyendo las curvas de aprendizaje. En la actualidad, existen distintos tipos de simuladores, de los cuales se pueden destacar los siguientes:

- a) Caja de simulación laparoscópica (*Endotrainers*): Simuladores de bajo costo que consisten de “cajas” sencillas que permiten realizar distintos ejercicios con el instrumental quirúrgico, utilizando material quirúrgico como gasas, suturas, tubos de drenaje de látex, etc.
- b) Modelos cadavéricos: Siendo de origen humano o animal (dependiendo del procedimiento), permiten simular el procedimiento en su totalidad, aunque con las diferencias esperadas ante un tejido vivo, son de difícil acceso y de un alto costo de mantenimiento.
- c) Simuladores virtuales: Simuladores de alto costo que permiten realizar procedimientos completos, los cuales brindan una retroalimentación del trabajo realizado con la finalidad de corregir errores en los siguientes entrenamientos.
- d) Modelos vivos: Generalmente utilizan animales vivos como el conejo o cerdos para simular procedimientos completos y de difícil reproducción en otro tipo de simuladores como el *Endotrainer*. Se requiere de equipo e insumos completos (por ejemplo, anestesia) para el procedimiento, este tipo de simulación tiene costos elevados.

Cada uno de estos simuladores, con sus ventajas y desventajas, muestran su capacidad para cubrir necesidades específicas en el entrenamiento de los cirujanos y mejorar sus habilidades.

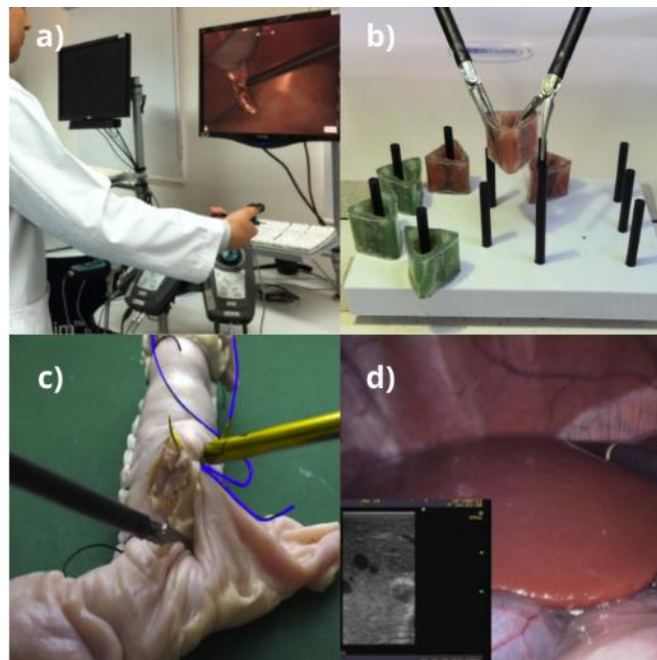


Figura 3 Procedimiento de cirugía laparoscópica en **a)** Simulador virtual, **b)** Endotrainer, **c)** Modelo cadavérico, **d)** Modelo vivo. León et al., (2015).

Capítulo 2. Estado del arte

En la actualidad, existen diferentes simuladores quirúrgicos especializados de acuerdo con la actividad a realizar. Para el campo de la cirugía laparoscópica pediátrica, se pueden destacar los siguientes dispositivos:

Simulador laparoscópico de código abierto

Los autores Gavrilovic et al., (2018), pertenecientes a la *Division of General and Thoracic Surgery* del *Hospital for Sick Children* en Canadá, desarrollaron un simulador de bajo costo y de código abierto de tamaño pediátrico encargado de medir la fuerza y el movimiento, además mantiene la capacidad de realizar una evaluación completa de habilidades y destrezas laparoscópicas como tiempo para completar una tarea y precisión al realizarla (ver Figura 5).



Figura 4 Simulador laparoscópico de caja completo de bajo costo y código abierto con ambos soportes ópticos instrumentados, completamente insertados en la caja. Gavrilovic et al., (2018).

Este simulador, el cual inicialmente era un simulador de caja convencional, se le integraron los productos comerciales como:

1. Sensores ópticos ADNS-9800 LaserStream para la medición del movimiento Roll y la traslación y unidades de medición inercial (IMU) LSM6DS33 para medir los movimientos de izquierda y derecha (pitch) y arriba hacia abajo (yaw) de los instrumentos laparoscópicos (Figura 5)

2. Soportes ópticos, fabricados por manufactura aditiva, que funcionan como puertos de entrada de los instrumentos y para resguardar los sensores ópticos junto con las IMU's
3. Un microcontrolador Teensy 3.2 para interconectar las IMU's y los sensores de los lados izquierdo y derecho utilizando los protocolos I2C y SPI respectivamente, contando con un micro-USB que permite la comunicación con la PC y dar la alimentación a 5V
4. Un mouse 3D DX-700034 SpaceNavigator encargado de medir la fuerza traslacional y rotacional aplicada en los ejes X, Y y Z.
5. Una cámara digital de alta definición Microsoft LifeCam HD-6000 para la grabación y transmisión del video de la actividad de entrenamiento en tiempo real.

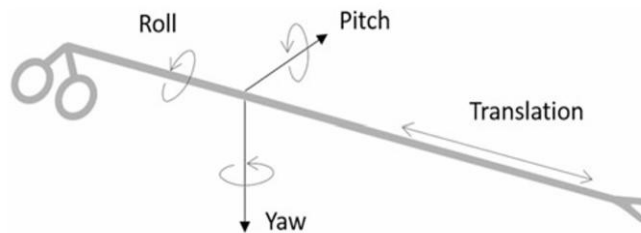


Figura 5 Grados de libertad (GDL) medidos para los instrumentos laparoscópicos. Gavrilovic et al., (2018).

Todo lo mencionado anteriormente y en conjunto, a través de un software desarrollado en Java para operar el proyecto, permiten la lectura y registro de los datos de movimiento durante los procedimientos realizados teniendo una alta precisión y permitiendo así su uso en entornos con recursos abundantes o limitados, permitiendo al usuario personalizar diversas características como las pertenecientes a la cámara durante el entrenamiento.

Simulador de Cirugía Laparoscópica Pediátrica para el Análisis de Movimiento

El simulador presentado por Nasr et al., (2014), tuvo como objetivo principal su uso en el entrenamiento de habilidades laparoscópicas pediátricas. Para ello, fue diseñado a partir de un simulador de caja convencional con dimensiones adaptadas para un caso pediátrico, llegando a un volumen de 1620 [cm³] y teniendo la capacidad de realizar el seguimiento de movimiento de los instrumentos. La caja fue equipada con una cámara web que permite el uso de cualquier computadora como pantalla para visualizar el procedimiento, un sensor tipo mouse ADNS-7530 Avago Technologies y un acelerómetro de 3 ejes BMA-180 Bosch Sensortec GmbH, montados en un circuito impreso previamente diseñado y finalmente conectados a un microprocesador

ATmega168 Atmel, permitiendo sincronizar los datos adquiridos con los instrumentos utilizados y mostrándolos a través de una interfaz.

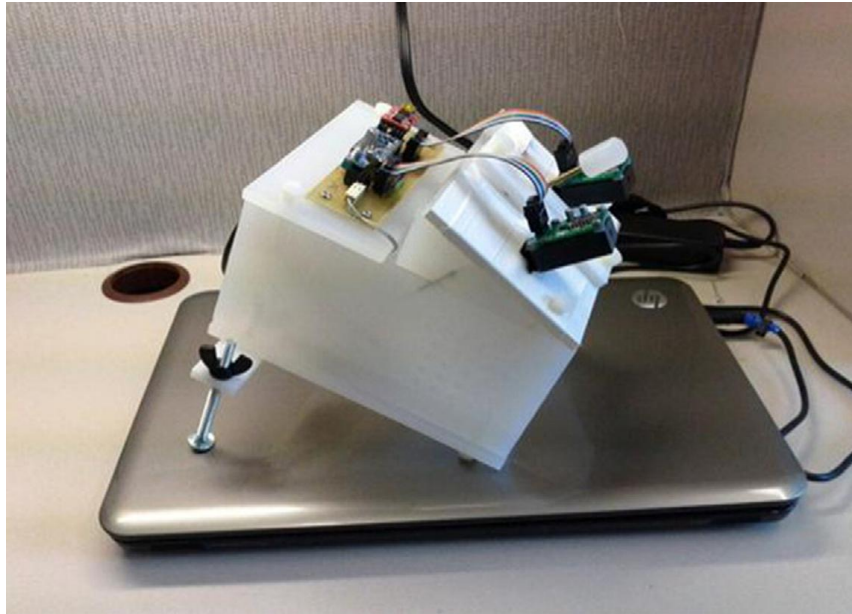


Figura 6 Simulador de cirugía laparoscópica pediátrica equipado con hardware de seguimiento de movimiento personalizado. Nasr et al., (2014).

Para las pruebas de validación de este simulador fueron requeridos 75 cirujanos pediatras con distintos niveles de experiencia laparoscópica, clasificados como novatos (menos de 10 procedimientos de cirugía laparoscópica realizados), intermedios (entre 10-50 procedimientos realizados) y expertos (más de 50 procedimientos realizados). Todos los participantes fueron sometidos a realizar una tarea de sutura donde, gracias al sistema fabricado, se evaluaron 3 características en sus procedimientos: velocidad, aceleración y rango de las distancias recorridas, obteniendo así una retroalimentación y comparación de sus resultados.

Simulador de cirugía laparoscópica pediátrica de puerto único

En este caso, se trató de un simulador de cirugía laparoscópica pediátrica de múltiples puertos adaptado a uno de puerto único para determinar la posibilidad de su uso en capacitaciones y evaluación de habilidades quirúrgicas realizado por Herbert et al., (2015). Para ello, se requirieron 41 participantes clasificados de la siguiente manera: novatos (menos de 10 procedimientos de cirugía de puerto múltiple y menos de 10 casos de cirugía de un solo puerto, intermedios (más de 10 procedimientos de puerto múltiple y menos de 10 procedimientos de un solo puerto) y

expertos(más de 10 procedimientos de puerto múltiple y más de 10 procedimientos de un solo puerto) que, a partir de la realización de 4 tareas de cirugía utilizando el simulador, fueron evaluados con aspectos como tiempo de finalización y rendimiento.

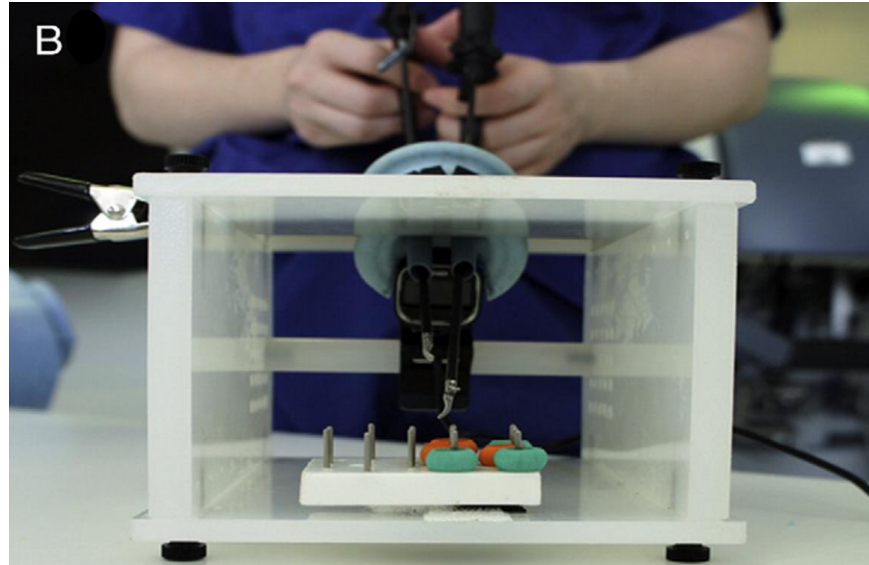


Figura 7 Vista en ángulo inverso dentro del simulador PLS modificado, que muestra la disposición del puerto de entrada única, los instrumentos y la cámara web. Imagen de Herbert et al., (2015).

La construcción del simulador requirió un puerto SILS Covidien para simular una ubicación umbilical en un paciente pediátrico y poder contar con los puertos de ingreso de los instrumentos laterales a utilizar, al igual que una adaptación para permitir el ingreso de una cámara web de alta definición que permitiera tener un registro de la actividad real por cada uno de los participantes; una vez obtenidos los resultados se realizó un análisis de los mismos para evaluar la diferencia entre los niveles de experiencia entre cada participante y las áreas a reforzar. A diferencia de los anteriores simuladores laparoscópicos pediátricos, este no mantiene tecnología para realizar la captura y registro del movimiento y la fuerza de los instrumentos.

Entrenador quirúrgico toracoscópico instrumentado

El desarrollo de este simulador tuvo el enfoque de medir la fuerza y los movimientos aplicados de manera confiable y tal que, al finalizar su uso, se obtuviera una retroalimentación cuantitativa de la habilidad técnica del cirujano Moorhead et al., (2020).

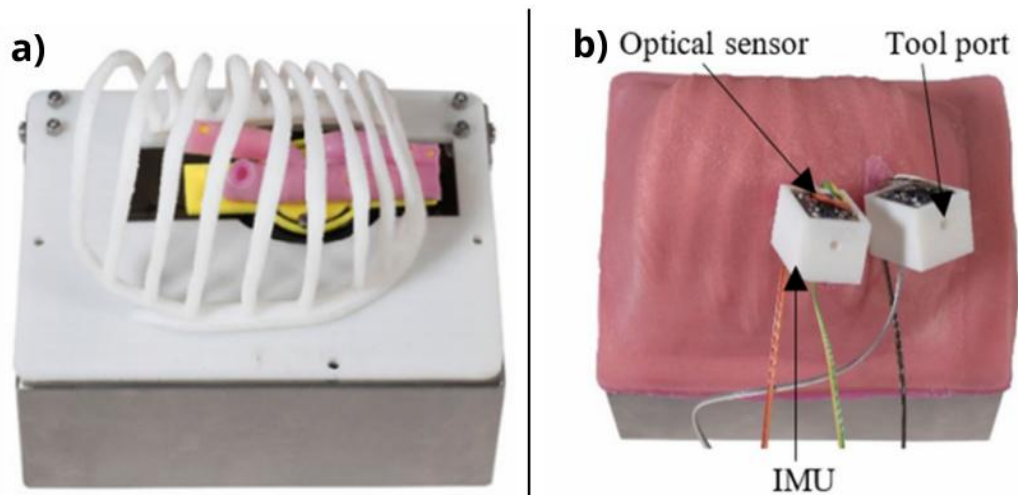


Figura 8 a) Simulador con la tapa de bisagras cerrada y la piel retirada para mostrar el modelo montado en el mouse 3D y la base de la caja torácica recortada, permitiendo el movimiento libre del modelo. b) Simulador que muestra la piel de silicona cubriendo el simulador, con alojamientos de sensores montados a través de los puertos de la piel, cada uno conteniendo un sensor óptico y una IMU. Moorhead et al., (2020).

Para su realización fue requerido el uso de la manufactura aditiva con el fin de reproducir las características de un paciente neonatal que es sometido a una reparación toracoscópica de la atresia de esófago. Para ello, se incluyeron impresiones 3D de costillas, moldes para tejido como piel y órganos, soportes para placas de circuito, entre otros. A la reproducción realizada le fue añadido un sistema instrumentado electrónicamente para el seguimiento del movimiento y medición de la fuerza, siendo utilizados los siguientes materiales:

1. Mouse 3D (3DConnexion, 3DX-700,043 SpaceMouse Wireless): Utiliza un sistema de resortes e interferómetros ópticos encargados de medir el desplazamiento, rotación, fuerza y torque en 3 dimensiones de un sistema de coordenadas cartesianas. Se comunicaba de manera inalámbrica a través de un dongle USB conectado a la PC utilizada.
2. Unidad de Medición Inercial (Bosch BNO055): Utilizado para medir el ángulo de las herramientas para torascopía
3. Sensor óptico (PMW3360, Pixart): Para este sensor fue fabricada una placa de circuito que permitía su conexión con el instrumental completo, permitiendo enviar las mediciones realizadas.

4. Microcontrolador (Arduino Uno): Encargado de ejecutar el firmware personalizado y encargado de sondear los sensores ópticos y el IMU para ser recibidos por medio de una interfaz de python que mostraría el video en tiempo real al usuario y, finalmente, guardar la información procesada para dar la retroalimentación sobre el procedimiento realizado por el cirujano.

Requisitos del simulador neonatal laparoscópico

Para el diseño y desarrollo de este simulador neonatal laparoscópico, se partió de los siguientes requerimientos:

- a) Contar con la capacidad de realizar la captura de datos del movimiento de la punta de los instrumentos utilizados en cirugía laparoscópica neonatal de 22[cm] de longitud y 3[mm] de diámetro.
- b) Mantener el uso de los 4 grados de libertad de los instrumentos, sin afectar o restringir el movimiento durante los procedimientos de entrenamiento a realizar.
- c) Permitir el análisis del movimiento realizado por cada instrumento con sus respectivos ejes y obtener los gráficos correspondientes, a partir de los datos obtenidos.
- d) Programar una interfaz gráfica de usuario que permita registrar el procedimiento a realizar, así como quien lo realiza, hora, fecha y obtener los archivos correspondientes con los datos obtenidos.
- e) Adaptar el tamaño de un simulador laparoscópico de caja común a un simulador neonatal con los componentes necesarios para realizar lo mencionado anteriormente.
- f) Obtener y almacenar los datos obtenidos durante el entrenamiento laparoscópico para su análisis final dentro de MATLAB.

Capítulo 3. Metodología

Simulador neonatal laparoscópico

Con el fin de cubrir los requerimientos mencionados anteriormente, se realizó un análisis de los posibles materiales a utilizar para la instrumentación electrónica y la fabricación de la forma física del simulador. Para iniciar el diseño del nuevo simulador neonatal, se obtuvieron las dimensiones usuales de un simulador de caja convencional. Este cuenta con dos piezas, caja y tapa. La caja está adaptada a una puerta para el ingreso y retiro de las tareas correspondientes como transferencia, corte y sutura, además de algunos otros procedimientos de cirugía utilizados en el entrenamiento del cirujano. Por otra parte, la tapa, cuenta con una curvatura y perforaciones que permitían el uso de un endoscopio y dos pinzas para realizar los procedimientos quirúrgicos.

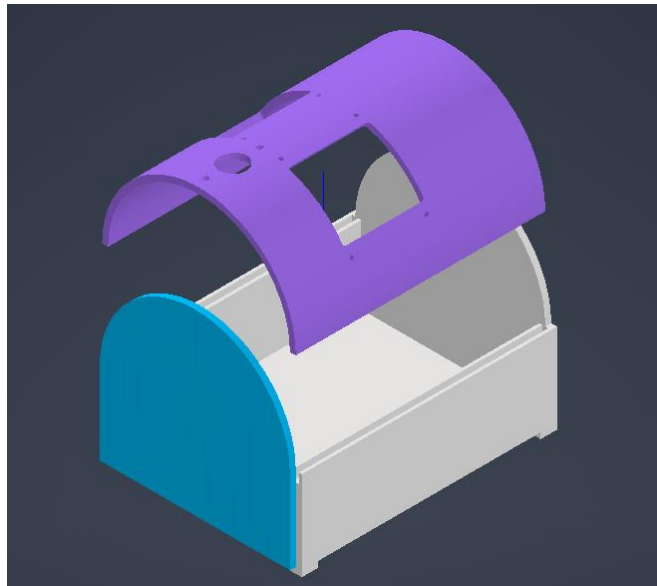


Figura 9 Partes que componen al simulador de cirugía laparoscópica pediátrica con puerta, para ingreso y retiro de tareas quirúrgicas (azul), y tapa para ingreso de pinzas de laparoscopia y endoscopio (morado).

Con estas características, se inició con la búsqueda de los sensores necesarios que, para los fines del proyecto, serían para capturar y registrar los 4 grados de movimiento de los instrumentos laparoscópicos pediátricos de 3 [mm] de diámetro y 22 [cm] de longitud. Entre los sensores existentes en el mercado en ese momento, se tomaron en consideración los siguientes:

Tabla 1 Tabla de comparación de los sensores

Sensor	Tecnología	Resolución [bit]	Rango	Comunicación	Nivel de Integración	Dimensiones [mm]
AS5600	Magnético	12	360°	I^2C	Circuito Integrado	6x5x1.75
MPU6050	Acelerómetro y giroscopio	Ac: 16 Gi: 16	Ac: ± 16 [g] Gi: ± 2000 dsp	I^2C	Módulo	21x17
BMI160	Acelerómetro y giroscopio	Ac: 16 Gi: 16	Ac: ± 16 [g] Gi: ± 2000 dsp	I^2C , SPI	Módulo	13x18
MPU 9250	Acelerómetro, giroscopio, magnetómetro	Ac: 16 Gi: 16 Mag: 16	Ac: ± 16 [g] Gi: ± 2000 [dsp] Mag: ± 4800 [uT]	I^2C , SPI	Módulo	26x15.5x2.6

Estos sensores fueron seleccionados debido a las ventajas que presentan para el proyecto como dimensiones pequeñas, costo, tiempo de entrega, materiales adicionales que requieren y espacio de trabajo necesario para su funcionamiento. Con la comparación anterior, el sensor AS5600 (Figura 10) resultó ser el más idóneo debido a que, además de contar con lo necesario para la instrumentación electrónica, mostraba una mayor ventaja debido a su facilidad de adquisición junto con los imanes requeridos para su funcionamiento, lo que permitía un ahorro de recursos.

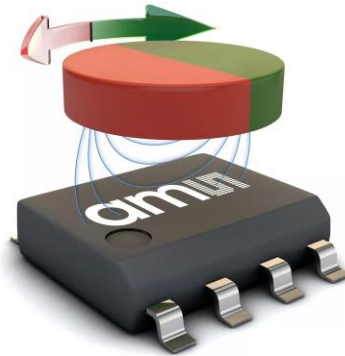


Figura 10 Sensor AS5600 de AMS OSRAM Position Sensor.

Los sensores de efecto hall rotativos AS5600 utilizados tienen las siguientes características:

1. Capacidad de medir un ángulo de 0° a 360° utilizando un imán radial (norte-sur) de 6 [mm] de diámetro con 2.5 [mm] de espesor
2. Cuenta con una resolución de 12 [bits] o 0.087°
3. Puede utilizar comunicación I^2C o PWM

4. Dirección I²C 0x36

La información para medir los ángulos y calcular los desplazamientos presentes en los 4 GDL necesarios para los instrumentos laparoscópicos se realiza por medio del protocolo de comunicación I²C. Dadas las características del simulador, los grados de libertad, y el número de puertos para el ingreso del instrumental en el simulador, se requirió el uso de 8 sensores, cuyas tarjetas electrónicas estaban adaptadas de acuerdo con el módulo al que estarían fijas para medir la señal de movimiento y las dimensiones del imán radial, requiriendo componentes como los siguientes:

- Resistencias pull-up
- Conectores FPC

Diseño electromecánico

Diseño de las articulaciones

Las articulaciones realizadas contaron con un diseño que permitía dar seguimiento con los 4 GDL requeridos para el instrumental laparoscópico pediátrico. A partir de ello, fueron adaptadas al simulador de caja inicial, considerando sus dimensiones y de los instrumentos a utilizar, el sensor seleccionado para la captura de datos, así como un aproximado de las dimensiones requeridas para los circuitos y el espacio disponible.

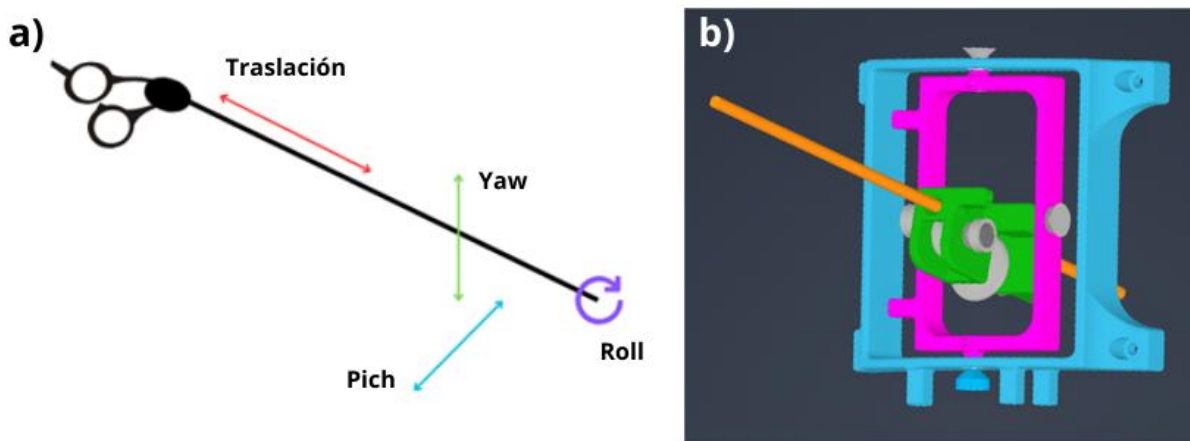


Figura 11 (a) Grados de libertad de la pinza, (b) Mecanismo completo con representación del ingreso del instrumento (naranja) y módulos: primario (verde), secundario (magenta), terciario (azul).

Como se observa en la Figura 11b, la articulación cuenta con 3 módulos primario (P), secundario (S) y terciario (T), cada uno encargado de cada grado de libertad del instrumento laparoscópico

pediátrico, a excepción del módulo P, el cual cubre 2 grados de libertad como la traslación y el giro sobre su propio eje.

Módulo P

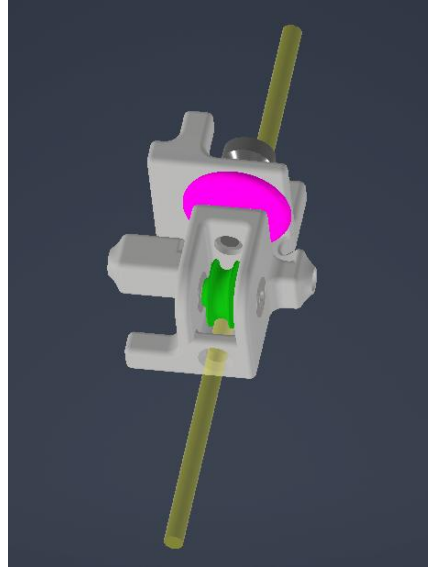


Figura 12 Módulo P y grados de libertad Traslación (verde) y Roll (morado) que representa junto con zona de ingreso de la pinza (amarillo).

Este módulo fue diseñado con el fin de registrar la entrada y salida del instrumento por medio de su avance (GDL 1) y el giro sobre su propio eje (GDL 2), Figura 12. Para registrar estos datos de acuerdo con el giro de los imanes, se diseñaron dos tipos de rueda adaptadas a las dimensiones y forma tubular del instrumento: una para el ingreso (a) y otra para el giro (b) mostrados en la Figura 13, cada una contando con un eje conectado al imán para registrar el giro de acuerdo con el movimiento realizado.

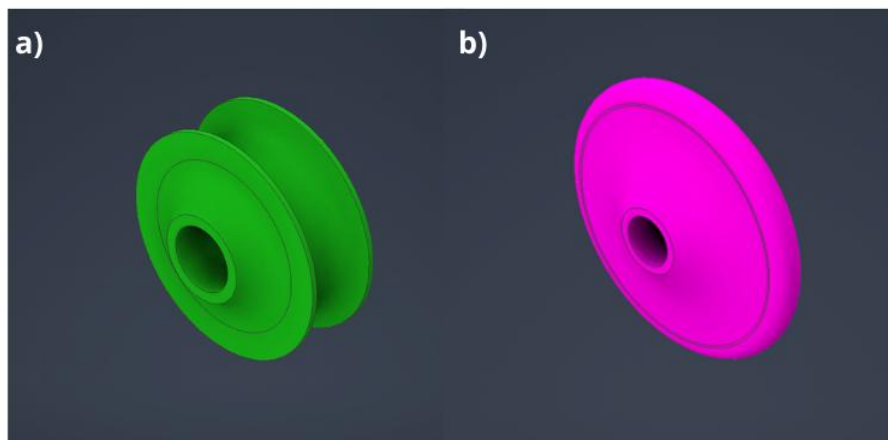


Figura 13 Ruedas del módulo primario.

Módulos S y T

En el caso de los módulos S y T, tienen un diseño en forma de anillos con el fin de permitir una libertad de movimiento durante su uso en la toma de datos de las herramientas quirúrgicas. Solamente, se destinaron a los dos grados de libertad restantes (GDL 3 y GDL 4) junto con los ejes conectados y necesarios para los imanes de los sensores AS5600, Figura 14.

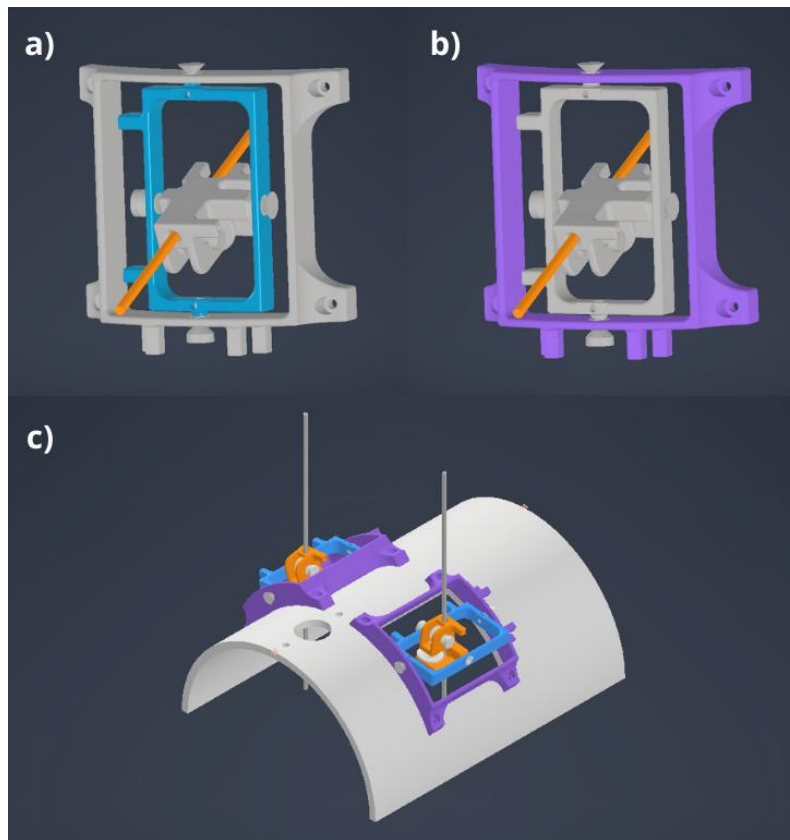


Figura 14 (a) Módulos S (azul) con GDL3, (b) Módulo T (violeta) con GDL4, (c) Ensamble completo de articulación sobre la tapa de ingreso de pinzas a la caja del simulador.

Endoscopio

Con el fin de visualizar las tareas dentro del simulador y que la imagen fuera mostrada sin retraso a través de un monitor externo para el cirujano, se realizó el diseño del siguiente endoscopio, Figura 15.

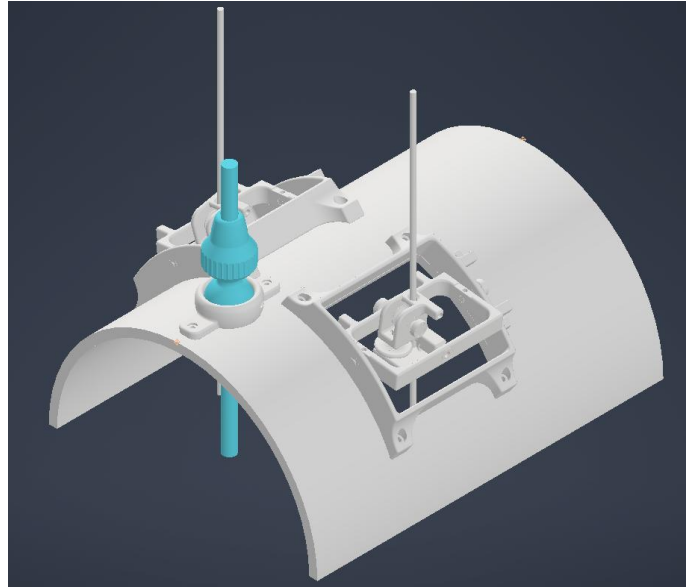


Figura 15 Endoscopio y ensamble en tapa del simulador.

A este diseño le fue adaptada una minicámara de 5 mm × 5 mm dentro del tubo de inserción (Azul), el cual permite tener una visibilidad de la zona de trabajo con un ángulo 0° de visualización. Dadas las características de la cámara, esta fue conectada a una pantalla por medio de un cable RCA, transmitiendo así la imagen en tiempo real hacia el monitor.



Figura 16 Endoscopio con cámara y conectado a un cable HDMI.

Diseño electrónico

El diseño electrónico partió del diseño de las articulaciones con el fin de lograr que se adaptara a la medida de cada módulo y sus sensores, obteniendo los siguientes en la Figura 17. Estas tarjetas mantienen la electrónica suficiente para realizar la comunicación entre los sensores con el microcontrolador seleccionado a través del protocolo I^2C , sin interferir o afectar el movimiento u acciones que realice el cirujano durante el uso del simulador, permitiendo una captura de datos adecuada. Cabe mencionar que cubren únicamente la medición de movimiento, por lo que, para aspectos del funcionamiento del microcontrolador como la comunicación con la PC, fue requerido el diseño de tarjetas adicionales, donde las principales serán mencionadas a continuación.

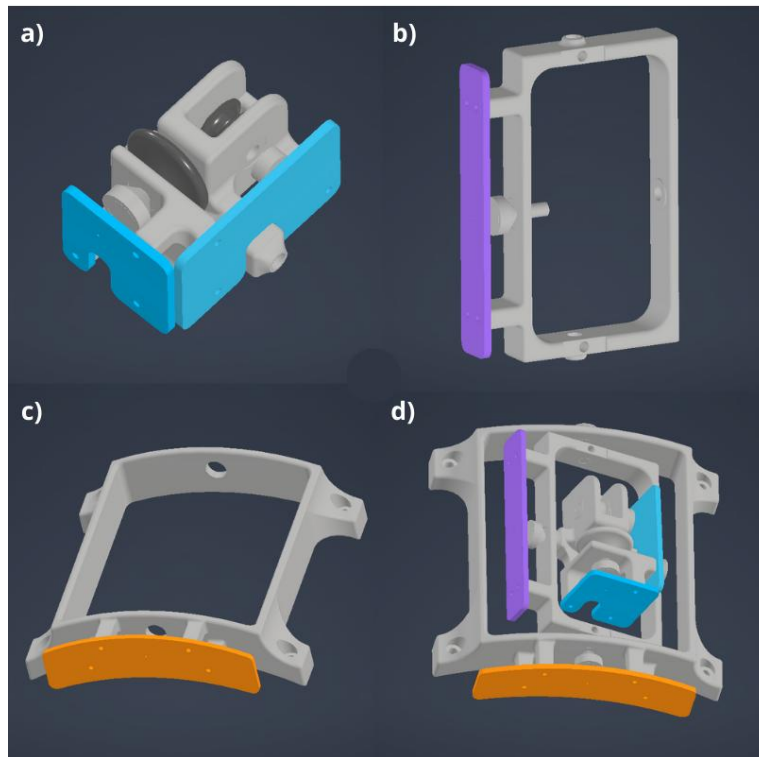


Figura 17 Diseño y visualización de dimensiones para las tarjetas electrónicas de los sensores AS5600 en cada módulo: (a) Tarjetas del módulo primario (azul), (b) Tarjeta del módulo secundario (morado), (c) Tarjeta del módulo terciario (naranja), (d) Ensamble completo con los tres módulos y ubicación prevista para las tarjetas electrónicas en el mecanismo.

Tarjeta electrónica de comunicación

Multiplexor TCA9548A

Debido a que se requieren obtener información de los 8 sensores AS5600 (4 en cada articulación mecánica) y que todos cuentan con la dirección 0x36 para el registro de las señales de movimiento, fue necesario el uso de un multiplexor con el fin de poder diferenciarlos dentro del proceso de comunicación. Para este fin, se utilizó el circuito integrado TCA9548A de Texas Instruments. Considerado esto, fueron requeridos los siguientes componentes:

- Resistencias pull-up
- conectores FPC de 10 pines y uno de 4 pines

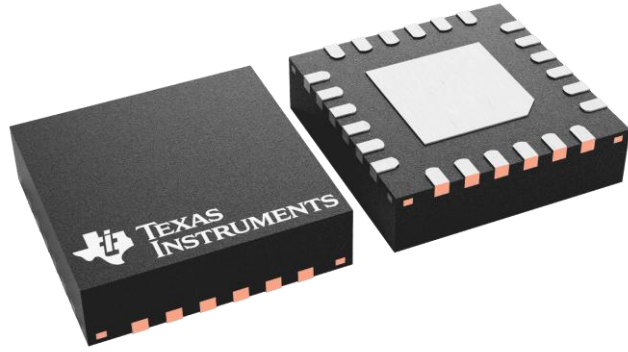


Figura 18 Circuito integrado TCA9548A de Texas Instruments. Imagen de Texas Instruments, 2012, TCA9548A Low-Voltage 8-Channel I2C Switch with Reset.

Microcontrolador PIC18F4550

Para la tarjeta encargada de la comunicación con los sensores AS5600 de rotación en las articulaciones, fue utilizado un microcontrolador PIC18F4550 de Microchip debido sus características principales: 32 pines I/O, 13 canales ADC de 10 [bit], memoria de programa Flash de 32 [kB], voltaje de operación de 4.2 [V] a 5.5 [V], un módulo USB 2.0, comunicación (I2C SPI, SPP, EUSART) entre otros, Figura 19.

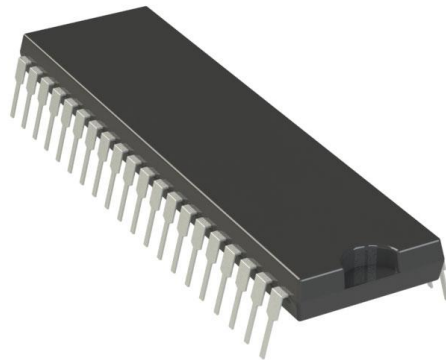


Figura 19 Microcontrolador PIC18F4550 de Microchip Technology.

Las características anteriores cumplieron algunos de los requerimientos del proyecto como el funcionamiento y comunicación del arreglo de sensores AS5600 utilizando la comunicación I2C, el uso de los canales ADC de 10 [bit] y el módulo USB para permitir la conexión del sistema con la interfaz gráfica solicitada para el registro de los datos. Además, el uso de este tipo de diseño y manufactura para los circuitos utilizados se debió al ahorro de espacio y la posibilidad de

adaptarse a una forma específica, sin afectar su funcionamiento, lo que permitió conservar la libertad de movimiento que brindan las articulaciones diseñadas.

Integración y conexión del sistema

Una vez obtenidas las tarjetas electrónicas restantes, fue requerido el diseño de una caja adicional para poder resguardarlas y permitir su comunicación dentro del simulador, logrando siguiente módulo:

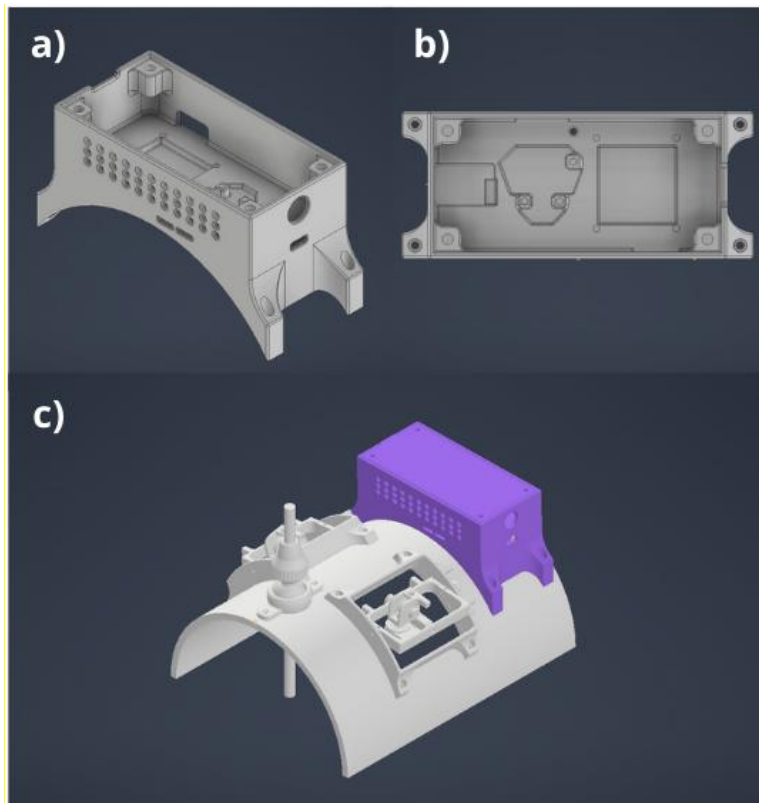


Figura 20 Caja de tarjetas electrónicas. (a) Vista isométrica, (b) Vista superior y (c) Ensamble de caja completa (violeta) con simulador

Este diseño permitió cubrir diferentes acciones necesarias para el funcionamiento y desarrollo del simulador, además de la comunicación con los sensores, ya que contaba con un puerto de entrada para la alimentación del sistema a partir de la conexión con la PC, un puerto para la programación del PIC y un switch para encendido y apagado del sistema.

Tratamiento de datos

Matriz de Rotación

Una matriz de rotación también puede describirse como el producto de una sucesión de rotaciones sobre un sistema de ejes coordenados principales tomando un orden específico Spong et al., (2006), Acosta (2020).

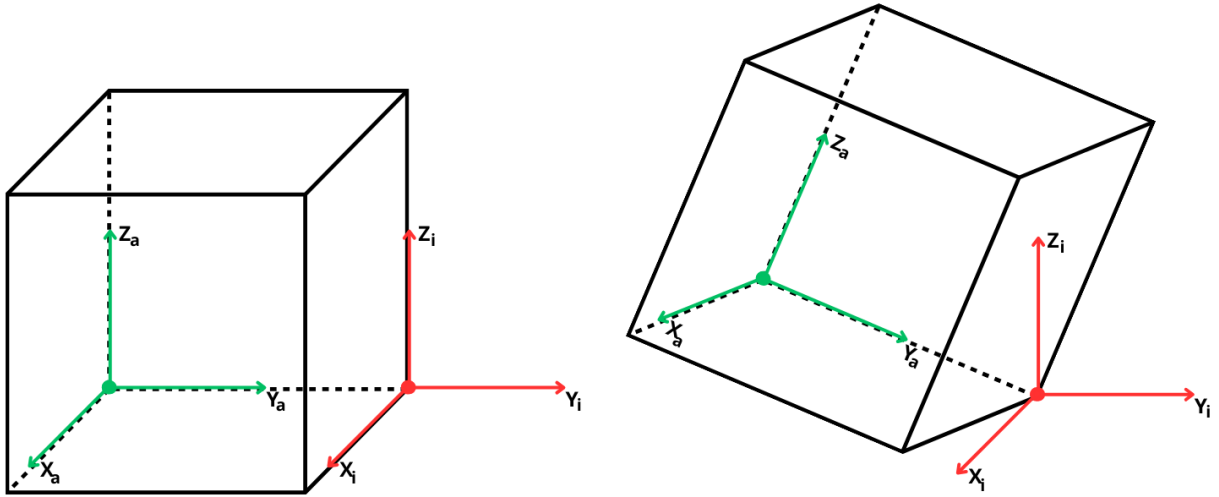


Figura 21 Sistemas de referencia “a” (verde) (fijo en el cuerpo rígido), “i” (rojo) (inercial), asociados a un cuerpo rígido.

Para Acosta (2020), en el caso de estudio de la orientación de un cuerpo rígido en el espacio es necesario el uso de dos sistemas de referencia, uno fijo ($\sum_i = \{x_i, y_i, z_i\}$) y otro ubicado en el cuerpo rígido ($\sum_a = \{x_a, y_a, z_a\}$). A partir de ello, la orientación del cuerpo rígido se establecería utilizando la relación entre los vectores unitarios de los sistemas de referencia involucrados, obteniendo lo siguiente:

donde x_a^i, y_a^i, z_a^i son las proyecciones de \sum_a sobre \sum_i .

$$x_a^i = \begin{bmatrix} x_a & x_i \\ x_a & y_i \\ x_a & z_i \end{bmatrix} \quad y_a^i = \begin{bmatrix} y_a & x_i \\ y_a & y_i \\ y_a & z_i \end{bmatrix} \quad z_a^i = \begin{bmatrix} z_a & x_i \\ z_a & y_i \\ z_a & z_i \end{bmatrix} \quad (1)$$

$$R = \begin{bmatrix} x_a \cdot x_i & y_a \cdot x_i & z_a \cdot x_i \\ x_a \cdot y_i & y_a \cdot y_i & z_a \cdot y_i \\ x_a \cdot z_i & y_a \cdot z_i & z_a \cdot z_i \end{bmatrix} \quad (2)$$

Entonces, R , además de ser una transformación lineal que expresa cualquier vector de Σ_a con respecto a Σ_i , también es una matriz de rotación, que describe la orientación de Σ_a con respecto a Σ_i .

En el caso del simulador, la matriz de rotación es necesaria debido a la necesidad del registro de movimiento de los instrumentos utilizados por los cirujanos, donde estos instrumentos, representados como un cuerpo rígido en el espacio que tendrá cambios de orientación, necesitarán un sistema de referencia adicional (fijo y representado por alguna ubicación en el simulador) para realizar las mediciones necesarias de las trayectorias obtenidas a partir de los cambios de orientación descritos.

Ángulos de Euler

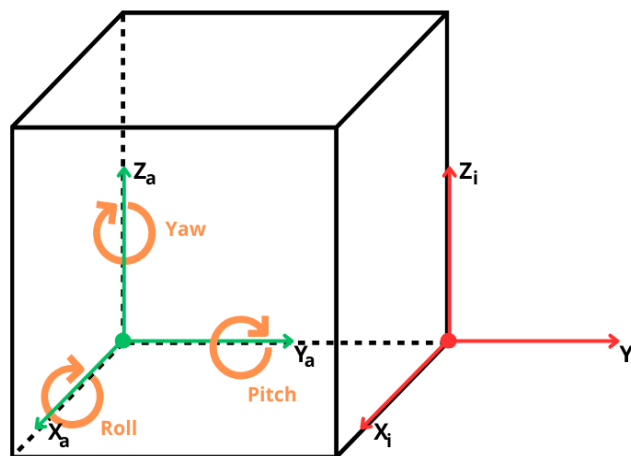


Figura 22 Sistemas de referencia "a" e "i" y ángulos respectivos a cada eje.

De acuerdo con Solís (2021), y tomando en cuenta los sistemas de referencia "a" e "i" mencionados anteriormente, la transformación de coordenadas de un punto P^a entre el sistema de referencia "a" y el sistema de referencia fijo "i" es:

$$P^i = R_a^i P^a \quad (3)$$

Donde:

$R_a^i \rightarrow$ Matriz de rotación del sistema de referencia "a" respecto al sistema de referencia "i"

La matriz R_a^i es obtenida a partir de los ángulos de Euler, teniendo giros sobre los ejes X, Y y Z, teniendo entonces:

$$R_a^i = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\text{sen}\gamma - \text{sen}\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \text{sen}\alpha\text{sen}\gamma \\ \text{sen}\alpha\cos\beta & \text{sen}\alpha\sin\beta\text{sen}\gamma + \cos\alpha\cos\gamma & \text{sen}\alpha\sin\beta\cos\gamma + \cos\alpha\text{sen}\gamma \\ -\text{sen}\beta & \cos\beta\text{sen}\gamma & \cos\beta\cos\gamma \end{bmatrix} \quad (4)$$

Siendo los giros de α , β y γ los ángulos yaw, pitch y roll respectivamente, y para poder estimarlos son utilizados los componentes de la matriz de rotación obtenida anteriormente.

$$\alpha = \tan^{-1}\left(\frac{r_{2,1}}{r_{1,1}}\right) \quad \beta = \tan^{-1}\left(\frac{-r_{3,1}}{r_{3,2}}\right) \quad \gamma = \tan^{-1}\left(\frac{r_{3,2}}{r_{3,3}}\right) \quad (5)$$

En este caso, (matriz de rotación anterior 4) se estimará a partir de las mediciones de los sensores ubicados dentro del simulador del presente proyecto. Para ello, fue necesaria la integración de sistemas de referencia como los siguientes:

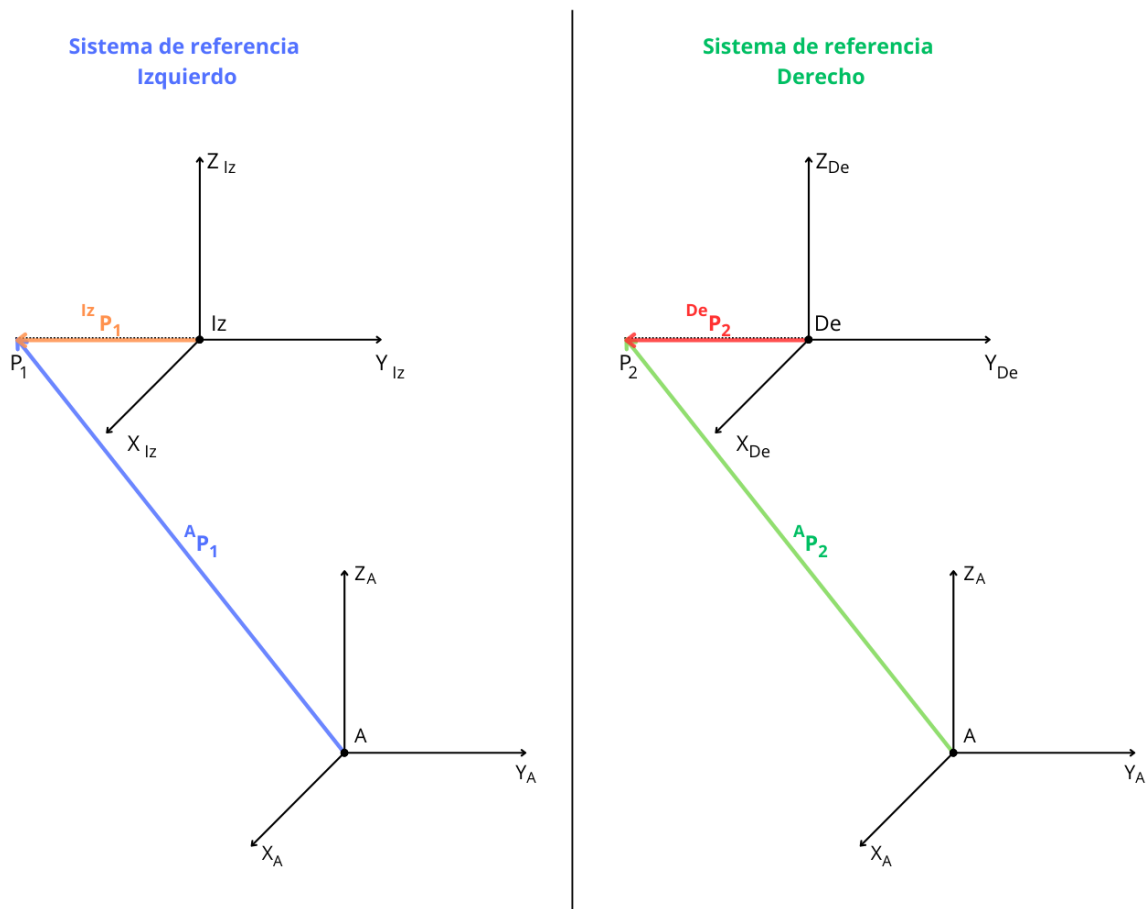


Figura 23 Sistemas de referencia de los puertos izquierdo y derecho

Un sistema de referencia fijo A se utilizó como referencia, los sistemas locales fijos Iz y De, se utilizaron para representar el puerto de entrada de los instrumentos izquierdo y derecho, respectivamente. Los puntos P1 y P2 representan las puntas de los instrumentos quirúrgicos, sus coordenadas se obtienen con las siguientes ecuaciones:

Puerto Izquierdo

$$P_1^A = P_{Iz}^A + R_{Iz}^A P_1^{Iz} \quad (6)$$

Puerto Derecho

$$P_2^A = P_{De}^A + R_{De}^A P_2^{De} \quad (7)$$

Donde:

$P_1^A \rightarrow$ Vector del sistema de referencia A a la punta del instrumento izquierdo

$P_{Iz}^A \rightarrow$ Vector del sistema de referencia A al sistema de referencia Iz

$P_2^A \rightarrow$ Vector del sistema de referencia A a la punta del instrumento derecho

$P_{De}^A \rightarrow$ Vector del sistema de referencia A al sistema de referencia De

$R_{Iz}^A \rightarrow$ Matriz de rotación lado izquierdo

$R_{De}^A \rightarrow$ Matriz de rotación lado derecho

$P_1^{Iz} \rightarrow$ Vector del sistema de referencia B a la punta del instrumento izquierdo

$P_2^{De} \rightarrow$ Vector del sistema de referencia B a la punta del instrumento derecho

$$P_1^A \rightarrow \{X_A, Y_A, Z_A\} \quad (8)$$

$$P_2^A \rightarrow \{X_A, Y_A, Z_A\} \quad (9)$$

$$P_1^{Iz} \rightarrow \{X_{Iz}, Y_{Iz}, Z_{Iz}\} = (0, Des_{Iz}, 0) \quad (10)$$

$$P_2^{De} \rightarrow \{X_{De}, Y_{De}, Z_{De}\} = (0, Des_{De}, 0) \quad (11)$$

Dado que los cirujanos cuentan con cuatro grados de libertad en cada una de las pinzas, se consideran para cada puerto las variables siguientes:

Puerto izquierdo: γ_1 (giro sobre el eje X_{Iz}), β_1 (giro sobre el eje Y_{Iz}), α_1 (giro sobre el eje Z_{Iz}), δ_1 (desplazamiento sobre el eje Y_{Iz})

Puerto derecho: γ_2 (giro sobre el eje X_{De}), β_2 (giro sobre el eje Y_{De}), α_2 (giro sobre el eje Z_{De}), δ_2 (desplazamiento sobre el eje Y_{De})

Los datos anteriores se utilizaron para construir las matrices de rotación R_{Iz}^A y R_{De}^A que al multiplicarse por los vectores P_1^{Iz} y P_2^{De} permitieron obtener las coordenadas de los puntos P_1^A y P_2^A , respectivamente.

Interfaz gráfica de usuario

El diseño de la interfaz gráfica de usuario se enfocó en que esta tuviera la capacidad de hacer un registro de cada uno de los cirujanos a partir de datos como los siguientes:

- a) Nombre (Iniciales)
- b) Prueba o tarea a realizar (Transferencia, Corte, Sutura, etc.)
- c) Fecha y hora

Utilizando esta información, permite la creación de un archivo con extensión *.txt* encargado de almacenar el registro de los movimientos realizados durante el procedimiento de entrenamiento en las tareas del simulador, así como poder identificar quién realizó la tarea, el tipo de tarea y la fecha de esta para su evaluación objetiva por un cirujano experto pediátrico.

Dadas estas funciones, se seleccionó Python como lenguaje de programación debido a sus características de código abierto, lo que permitiría su uso en diferentes medios y brindarle mejoras en el trabajo a futuro.

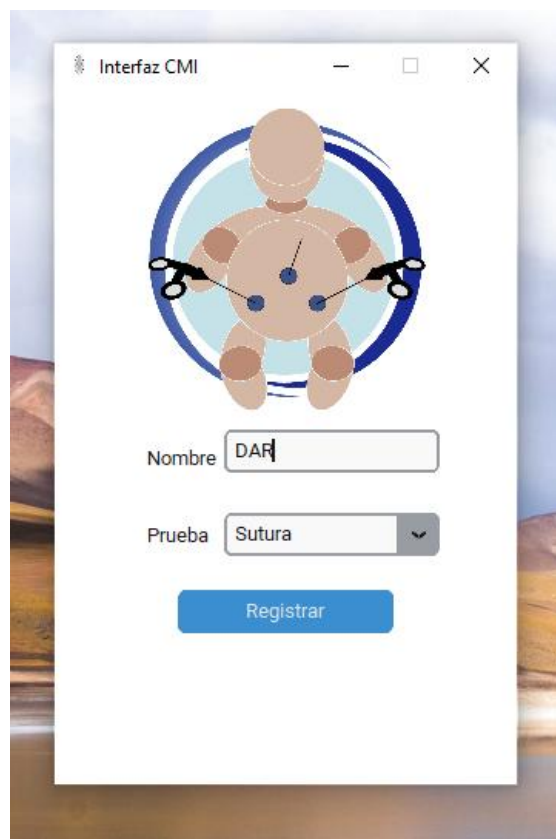


Figura 24 Ventana de inicio (Home) de la interfaz

Dentro del diseño de la interfaz, se consideró el uso de varios menús que ayudarán durante el registro y la captura de los datos de movimiento en 4 GDL con el fin de hacer su uso más amigable para el usuario. A partir de ello, se tiene una ventana de inicio Figura 24, con un menú principal que encargado de registrar los datos del cirujano y permitir seleccionar la tarea que se va a realizar.

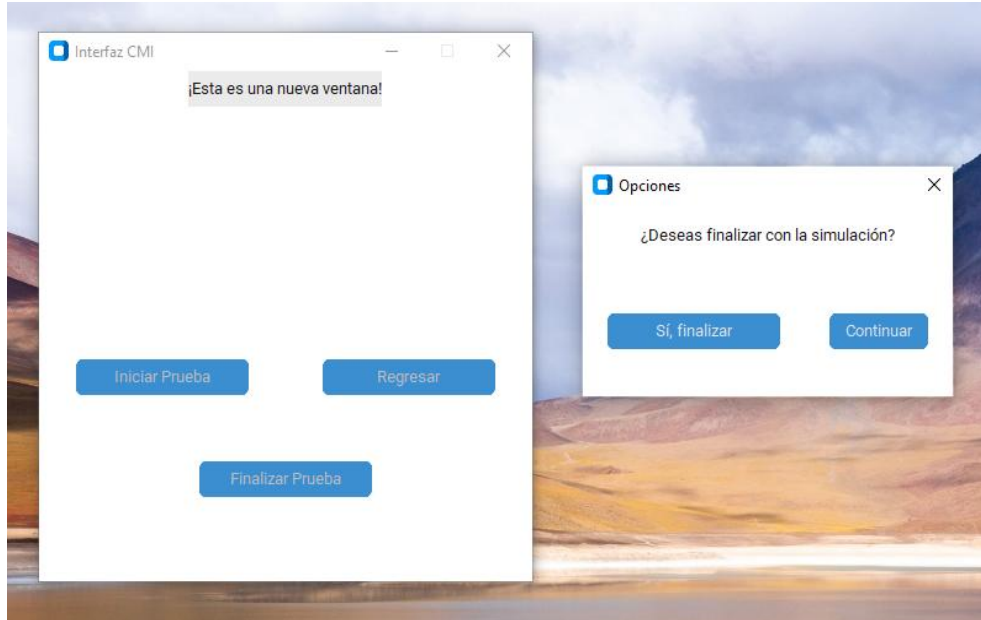


Figura 25 Ventana izquierda: Menú para seleccionar la acción a realizar (Inicio de prueba, finalización de prueba, regreso a menú principal/ventana principal) Ventana derecha: Menú para confirmación de finalización de prueba.

Una vez iniciada la prueba, la interfaz se encarga de cargar los datos obtenidos a partir del programa en C y comienza a almacenarlos en dos archivos *.txt*: 1) un archivo con los datos crudos y 2) un archivo con los datos tratados dentro de la matriz, hasta que el usuario finalice la simulación. De esta manera, automáticamente los archivos se guardan con un nombre creado a partir de los datos del cirujano pediátrico registrado en el menú principal y se guardan en una carpeta predeterminada para poder acceder a ellos en cualquier momento.

Al finalizar la prueba, la interfaz muestra las gráficas de movimiento correspondientes a cada puerto de entrada del instrumental (izquierdo y derecho) donde se observa el movimiento que realizó el cirujano con cada una de las herramientas laparoscópicas durante el procedimiento. Dichas imágenes también serán almacenadas dentro de la carpeta anterior con el mismo nombre que los archivos *.txt*.

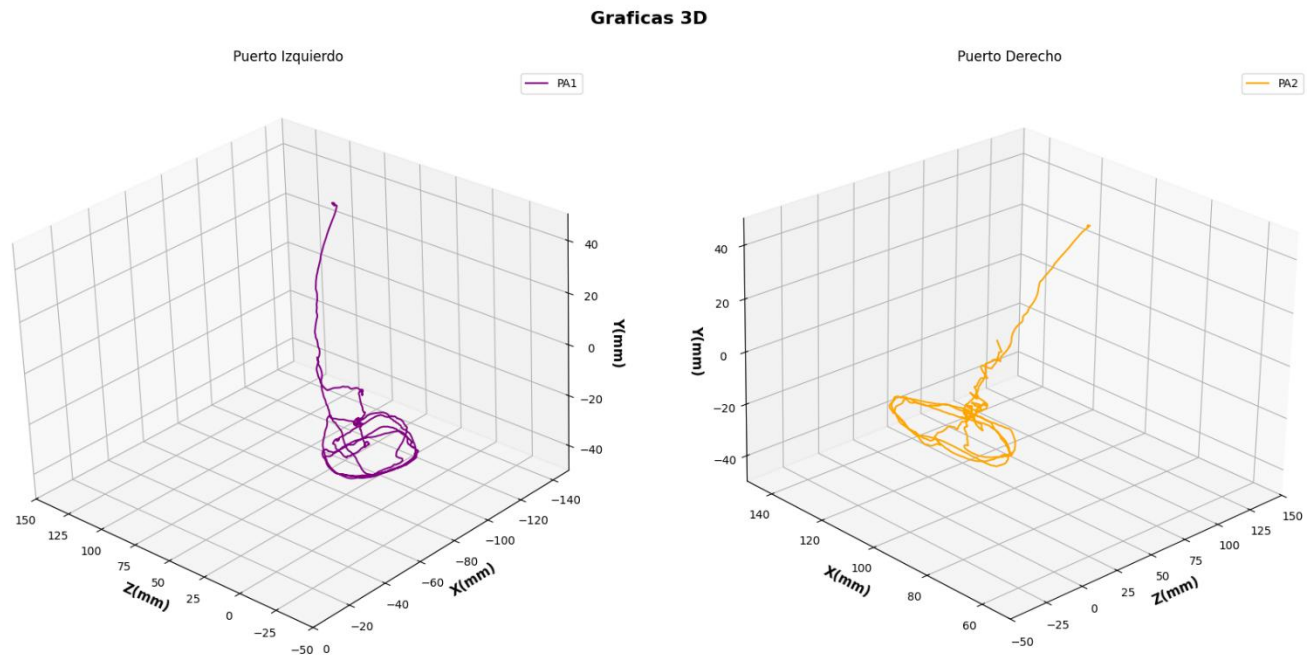


Figura 26 Gráficas mostradas por la interfaz del simulador laparoscópico pediátrico.

MATLAB

Para un posterior análisis del movimiento, se realizó un código en MATLAB para el tratamiento de los datos utilizando la matriz de rotación realizada y, con el mismo, generar las gráficas de los ejes X, Y y Z correspondientes al comportamiento realizado en ambos puertos de entrada del simulador (puerto izquierdo y derecho). En este código, se ingresa el archivo `.txt` de los datos crudos generado con la interfaz gráfica en Python, obteniendo la siguiente Figura 27.

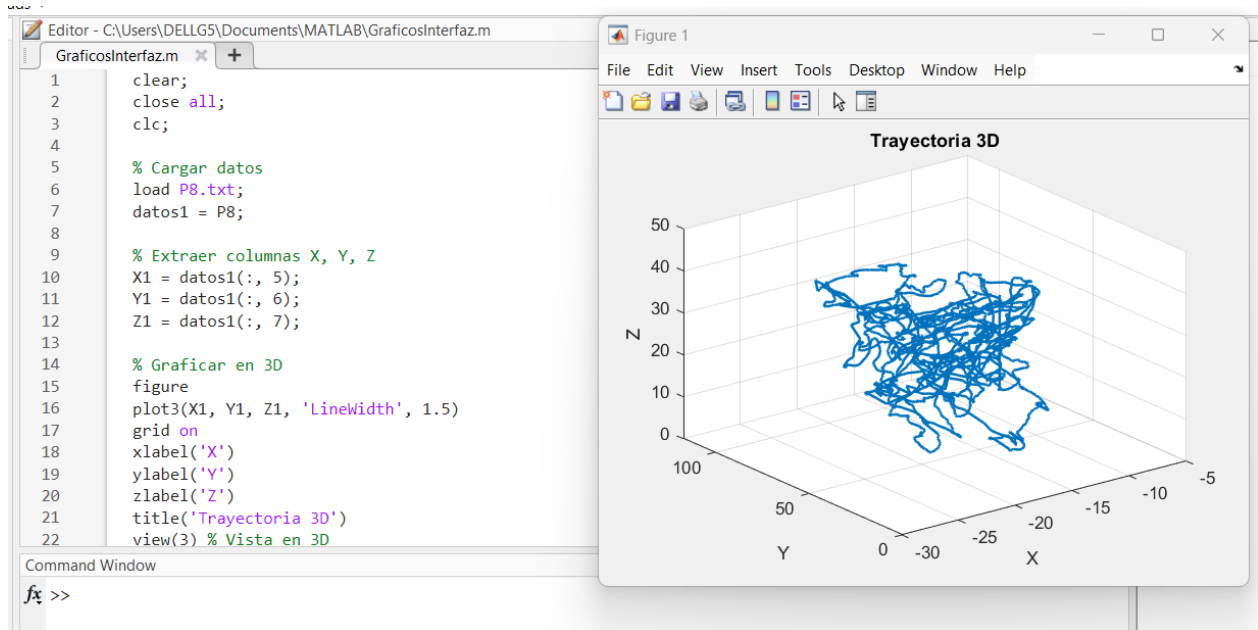


Figura 27 Muestra de código en MATLAB y gráficas generadas

Calibración

En el caso de la calibración de los sensores del sistema electrónico del dispositivo, se requirió el diseño de un calibrador adaptado a las características del simulador, Figura 28a. Este diseño permite añadir distintas placas para poder realizar la calibración de la instrumentación electrónica del simulador, como líneas rectas de 30 [mm] permitiendo seguir el movimiento en los 3 ejes (X, Y o Z).

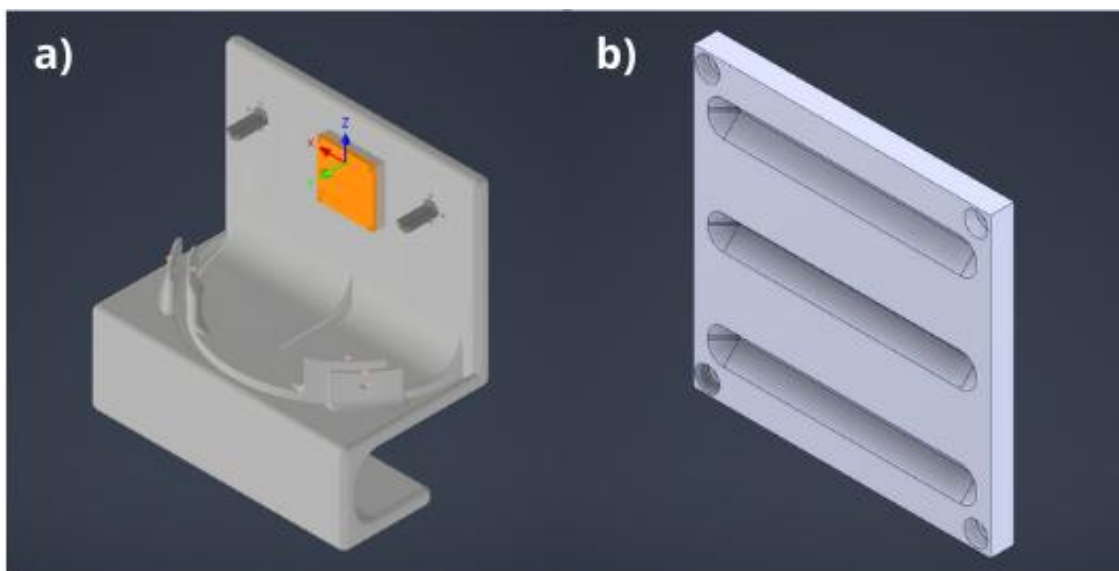


Figura 28 a) Calibrador, y b) Tarea de calibración con líneas de 30[mm].

Para la calibración, se realizaron 10 repeticiones de movimiento en cada una de las líneas de 30mm ubicadas en la tarea de calibración (Figura 28b), a partir de ello se obtuvieron los datos necesarios para el cálculo de aspectos como el error y la exactitud.

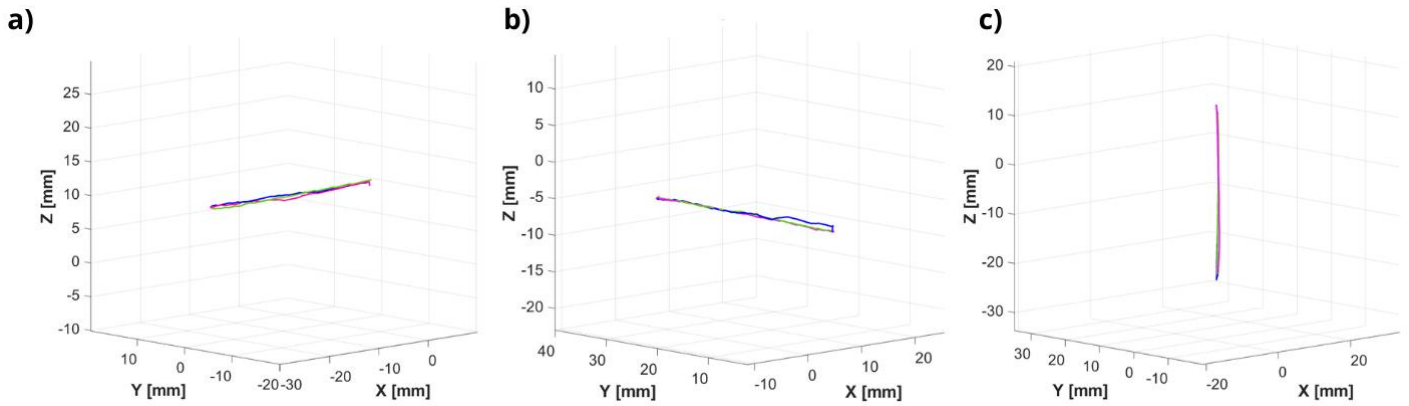


Figura 29 Resultado de las pruebas de calibración utilizando la tarea de líneas de 30 mm en cada eje. a) Trayectoria en eje X, b) Trayectoria en eje Y, c) Trayectoria en eje Z.

Simulador para cirugía laparoscópica pediátrica

Una vez realizados los requerimientos de manera secuencial y la calibración de los sensores, se inició el armado de todo el simulador (Figura 29).

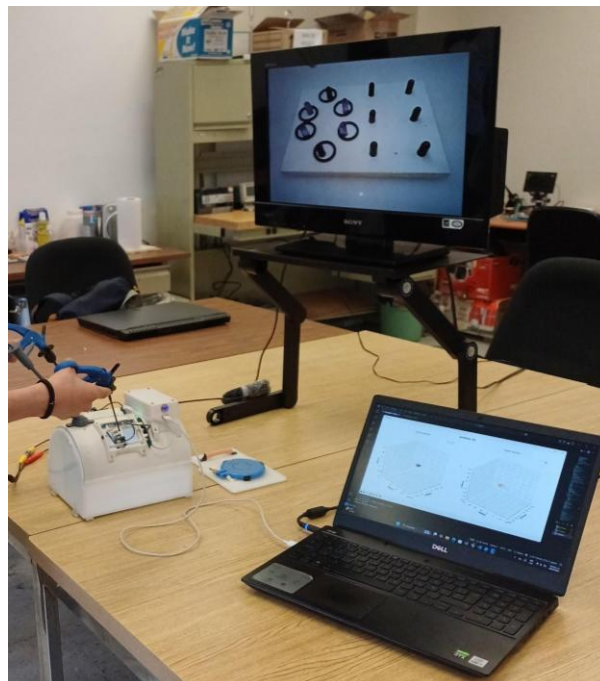


Figura 30 Simulador completo con circuitos, pinzas y conectado a la PC junto con la interfaz en funcionamiento.

Para el funcionamiento del simulador y de acuerdo con los requerimientos iniciales, se rediseñaron tareas comúnmente utilizadas en la práctica y entrenamiento de cirugía laparoscópica en adultos, adaptándolas a dimensiones relacionadas o cercanas con lo que un cirujano neonatal enfrentaría en un procedimiento real.

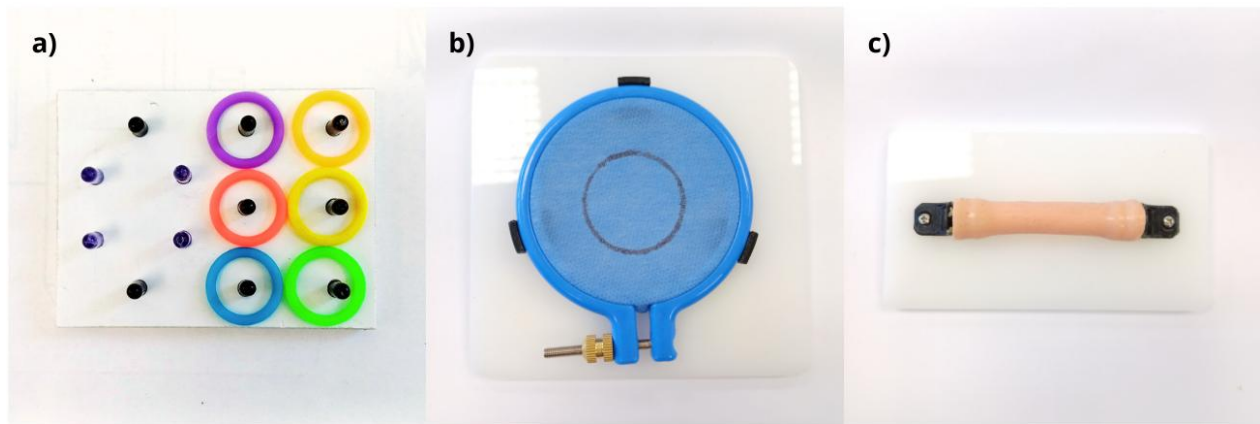


Figura 31 Tareas de cirugía laparoscópica utilizadas para el simulador: (a) Tarea de transferencia, (b) Tarea de corte circular, (c) Tarea de sutura.

Estas tareas laparoscópicas, se enfocaron en las habilidades y destrezas principales del entrenamiento pediátrico neonatal, lo que permitirá una evaluación de manera objetiva para comparar los resultados entre cirujanos expertos y residentes de la especialidad debido a que durante su ejecución en la práctica las gráficas suelen presentar una imagen de figuras sencillas como líneas que permiten visualizar con mayor facilidad la trayectoria requerida para cada tarea y si esta se realizó de manera correcta.

Capítulo 4. Resultados

Los resultados de las pruebas de calibración de los sensores se muestran en la Tabla 2, y se obtuvieron a partir del promedio de los datos de las 10 pruebas realizadas de cada tarea diseñada para la calibración, calculando la precisión, exactitud y el error.

Tabla 2 Valores promedio obtenidos en cada eje después de las 10 pruebas de medición para la calibración en mm.

Eje	Medida real de tarea	Promedio (Trayectorias)	Error promedio	Precisión	Valor máx.	Valor mín.	Error máximo
X	30	29,3381	-0,6618	0,4500	29,9874	28,7875	1,2124
Y	30	30,2097	0,2097	0,6492	30,8796	29,2046	-0,8796
Z	30	30,4828	0,4828	0,4628	29,5380	30,9608	-0,9608

El sistema electrónico para el movimiento de los instrumentos laparoscópicos pediátricos del simulador neonatal logra: 1) capturar de las señales de movimiento de la punta de las pinzas, obtenidas por los sensores AS5600, sin afectar la fluidez ni restringir los grados de libertad de las pinzas utilizadas, 2) generar los archivos de los datos crudos y tratados con la información obtenida, siendo diferenciados con los datos del cirujano que realizó la tarea a evaluar y la fecha de realización, 3) generar las gráficas de movimiento de los puertos izquierdo y derecho del simulador, logrando visualizar el movimiento que se realizó con ambas pinzas durante el procedimiento y 4) permitir el uso de distintas tareas de cirugía para el entrenamiento de habilidades quirúrgicas, específicamente para el área de cirugía neonatal.

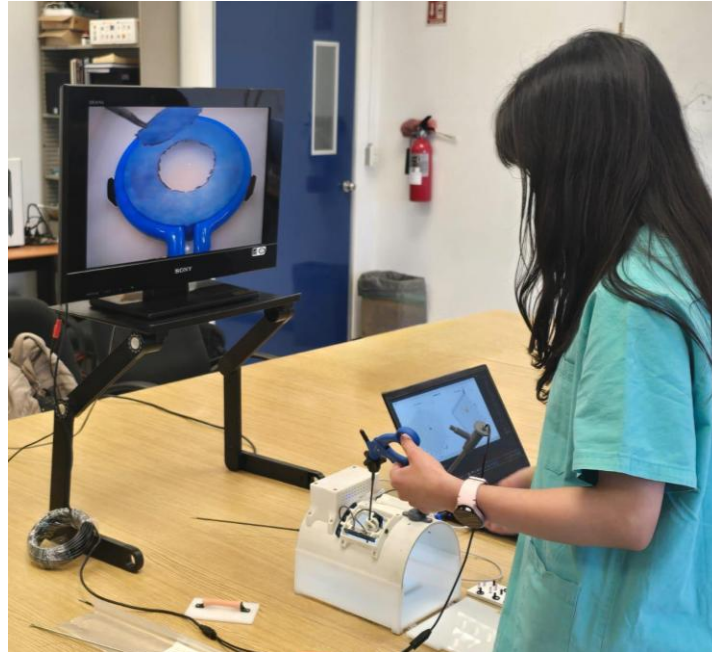


Figura 32 Simulador neonatal laparoscópico funcionando con la tarea de corte.

Adicionalmente, se realizaron las 3 tareas de entrenamiento laparoscópico utilizando el instrumental laparoscópico pediátrico de 3mm (disectoras, Maryland, tijeras, y portaagujas) y las tareas de la Figura 31 para obtener las trayectorias tridimensionales de la punta de las herramientas quirúrgicas.

Los resultados de las trayectorias registradas con el simulador neonatal de cirugía laparoscópica pediátrica para la tarea de transferencia de objetos se muestran en la Figura 33.

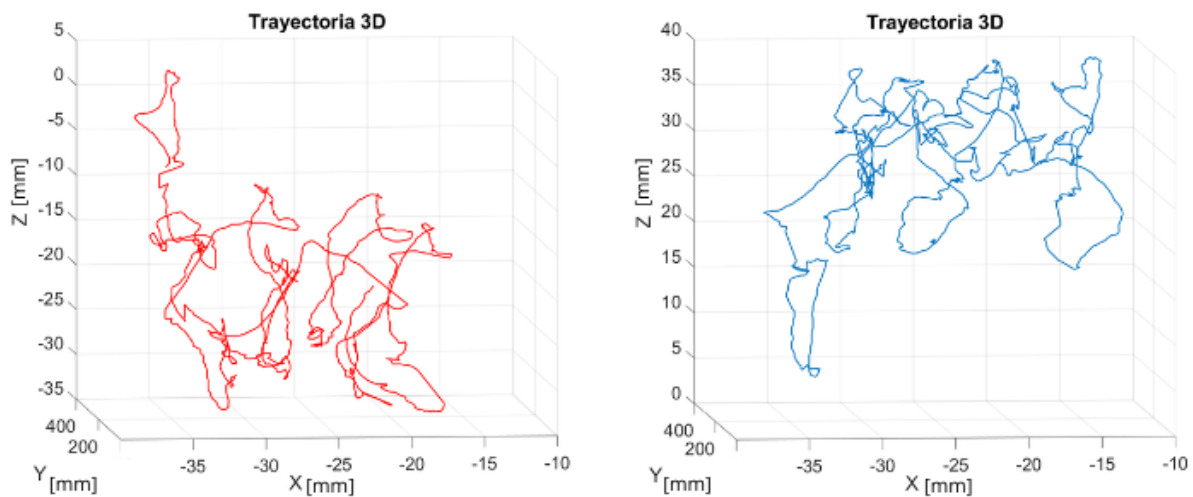


Figura 33 Trayectorias 3D de los instrumentos laparoscópicos pediátricos en la tarea de transferencia. Puerto izquierdo (rojo), Puerto derecho (azul).

Los resultados de las trayectorias registradas con el simulador neonatal de cirugía laparoscópica pediátrica para la tarea de corte circular se muestran en la Figura 34.

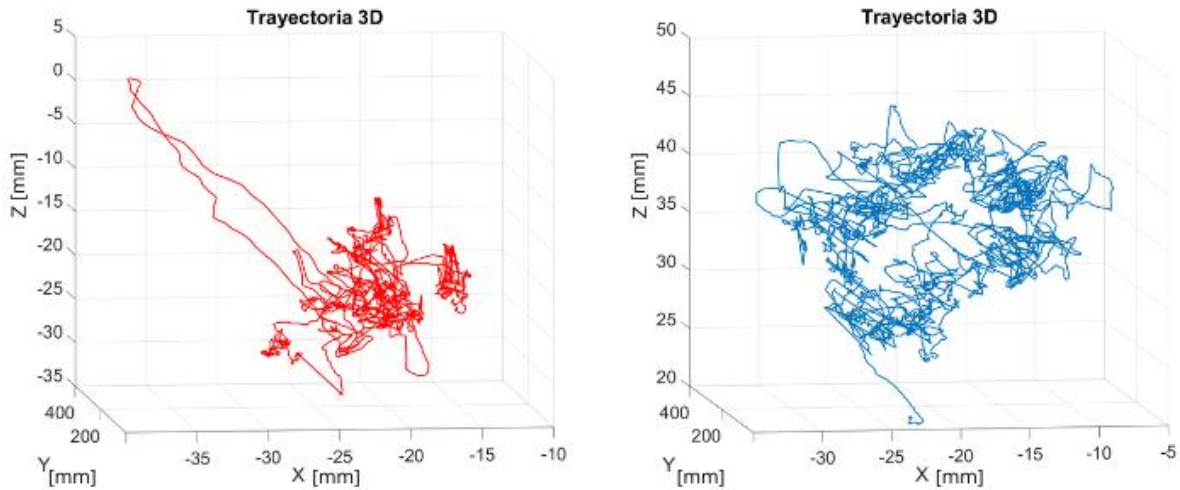


Figura 34 Trayectorias 3D de los instrumentos laparoscópicos pediátricos en la tarea de corte. Puerto izquierdo (rojo), Puerto derecho (azul)

Finalmente, los resultados de las trayectorias registradas con el simulador neonatal de cirugía laparoscópica pediátrica para la tarea de sutura se muestran en la Figura 35.

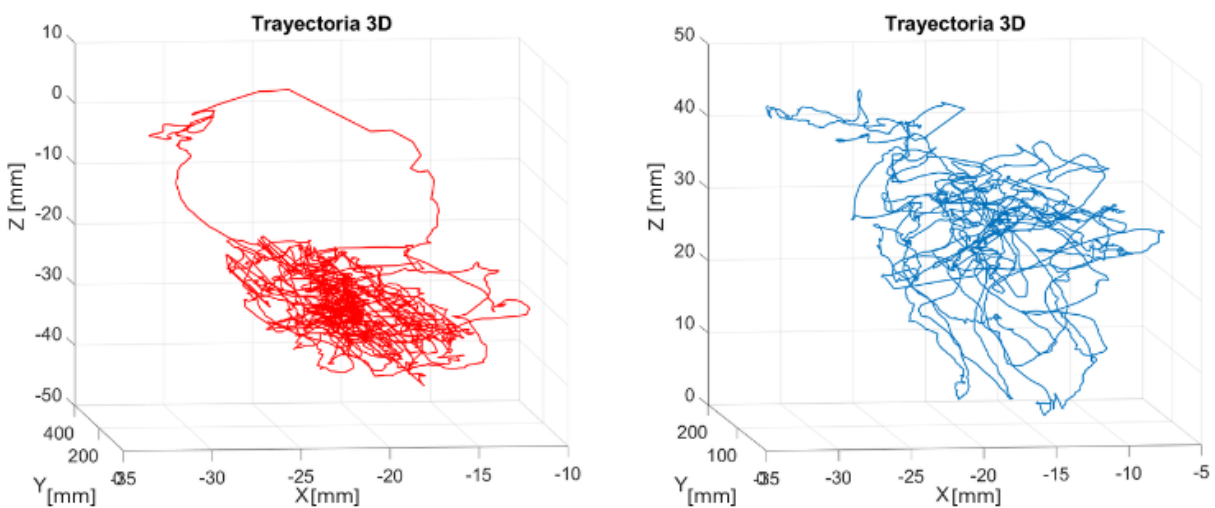


Figura 35 Trayectorias 3D de los instrumentos laparoscópicos pediátricos en la tarea de sutura. Puerto izquierdo (rojo), Puerto derecho (azul).

Las figuras 33, 34 y 35 muestran las trayectorias típicas realizadas de cada una de las tareas de cirugía, en las cuales, se puede observar la captura de movimiento de la mano derecha e izquierda. A partir de este resultado de habilidades se puede evaluar el desempeño a través de métricas o cálculos matemáticos en el área de laparoscopia y analizar los patrones de comportamiento a mayor profundidad con otras herramientas como MATLAB, redes neuronales, etc.

Capítulo 5. Conclusiones

Conclusiones

Se realizó el diseño y construcción de un simulador para el entrenamiento de residentes en cirugía laparoscópica neonatal, el cual, cumple con los requerimientos solicitados al inicio del proyecto, esto debido a los puntos que se mencionan a continuación:

Permite el entrenamiento y evaluación de habilidades y destrezas psicomotrices por medio de ejercicios estandarizados como transferencia, corte y sutura, adaptadas a dimensiones cercanas a las que se pueden presentar en un procedimiento real.

El sistema completo realiza la captura y registro de las señales de movimiento de instrumentos laparoscópicos de 3[mm] de diámetro y 22[cm] de longitud, midiendo dentro de los 4 GDL, a través de los sensores de efecto hall AS5600. Estos sensores se encuentran integrados al arreglo instrumental de articulaciones, fabricadas por manufactura aditiva utilizando materiales de PLA y Resina y en tarjetas electrónicas adaptadas a los módulos de las articulaciones del sistema, los cuales permiten el libre movimiento de los instrumentos durante el registro del entrenamiento del usuario. Los datos capturados se encuentran en coordenadas del mundo real, con una precisión de ± 0.649 [mm] y una exactitud de ± 1.623 [mm], valores que fueron obtenidos a partir del resultado de las pruebas de calibración realizados en el capítulo anterior.

Para el registro de los datos capturados, se programó una interfaz gráfica de usuario que tiene la capacidad de crear archivos *.txt* con los datos correspondientes del procedimiento completo, así como la información del cirujano que lo realizó, permitiendo la evaluación de expertos hacia los novatos al realizar distintas tareas laparoscópicas. Esta interfaz permitirá, además la visualización gráfica del movimiento realizado por los mismos cirujanos, observar en primer instancia posibles errores durante el procedimiento y en el caso de requerir una evaluación más exhaustiva brindaba un archivo *.txt* adicional con la información adaptada para ser cargada en MATLAB y que a su vez el software permitiera obtener un análisis gráfico más detallado. El simulador también cuenta con un endoscopio diseñado, el cual permite la visualización en tiempo real del procedimiento tal como sucedería en un entorno real de cirugía laparoscópica, permitiendo así el entrenamiento de perspectiva y profundidad del cirujano.

Trabajo a futuro

Con el fin de mejorar la experiencia de usuario y poder realizar una mayor cantidad de procedimientos para el entrenamiento de habilidades y destrezas laparoscópicas, se propone adaptar el simulador de manera que pueda llegar a utilizarse el instrumental completo requerido en este tipo de cirugía. Esto, además de permitir una experiencia más apegada a la realidad y permitir que el cirujano aprenda a usar de manera adecuada y segura cualquier tipo de instrumento sin llegar a dañar al paciente, permitirá aumentar la variedad de procedimientos que podrían realizarse dentro del simulador. Añadido a lo anterior, también se sugiere incluir al sistema un método para el seguimiento de la curva de aprendizaje del cirujano, basándose en la captura de datos en la nube, así como el uso de métodos de medición de aprendizaje. En cuanto al diseño del simulador, se propone incluir el uso del sistema a un modelo anatómico de un neonato con sus órganos y tejidos específicos, con la finalidad de completar el entrenamiento del residente en un procedimiento quirúrgico similar y permitir el análisis del procedimiento con el fin de mejorar sus habilidades y destrezas.

Referencias

- Acosta, J. H. (2020). *Diseño de controladores para seguimiento de trayectorias en vehículos aéreos de cuatro rotores*. cicese. https://cicese.repositorioinstitucional.mx/jspui/bitstream/1007/3216/1/Tesis_Hirata%20Acosta%20Jonathan_04_feb_2020.pdf
- ams OSRAM. (2018). *ams AS5600 Position Sensor*. ams OSRAM. <https://ams-osram.com/products/sensor-solutions/position-sensors/ams-as5600-position-sensor>
- BOSCH. (2020). *Data sheet BMI160*. BOSCH. <https://uelectronics.com/wp-content/uploads/2024/06/AR4223-BMI160-Sensor-Giroscopio-y-Acelerometro-Datasheet.pdf>
- Colegio y Sociedad Mexicana de Cirugía Pediátrica, A.C. (2024). *Colegio y Sociedad Mexicana de Cirugía Pediátrica, A.C.* socmexcirped. <https://socmexcirped.org/>
- Gavrilovic, B., Fahy, A. S., Carrillo, B., Nasr, A., Gerstle, J. T., & Azzie, G. (2018). Development of an Open-Source Laparoscopic Simulator Capable of Motion and Force Assessment: High Tech at Low Cost. *JOURNAL OF LAPAROENDOSCOPIC & ADVANCED SURGICAL TECHNIQUES*, 28(10), 1253-1260. 10.1089/lap.2018.0126
- Herbert, G. L., Cundy, T. P., Singh, P., Retrosi, G., Sodergren, M. H., Azzie, G., & Darzi, A. (2015). Validation of a pediatric single-port laparoscopic surgery simulator. *Journal of Pediatric Surgery*, 50, 1762-1766. 10.1016/j.jpedsurg.2015.03.057
- InvenSense. (2014). *MPU-9250 Product Specification Revision 1.0*. invensense. <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-9250-Datasheet.pdf>
- León, F.F., Varas, C.J., Buckel, S.E., Crovari, E.F., Pimentel, M.F., Martínez, C.J., Jarufe, C.N., Boza, W.C. (2015). Simulación en cirugía laparoscópica. *ELSEVIER*, 93(1), 4-11. 10.1016/j.ciresp.2014.02.011
- Microchip Technology. (n.d.). *PIC18F2455/2550/4455/4550 Data Sheet*. microchip. <https://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>
- Moorhead, A. A., Nair, D., Morison, C., Cook, N. J., Beasley, S. W., & Wells, J. M. (2020). Development of an instrumented thoracoscopic surgical trainer for objective evaluation of esophageal atresia/tracheoesophageal fistula repair. *Medical & Biological Engineering & Computing*, 58, 601-609. 10.1007/s11517-019-02107-6
- Nasr, A., Carrillo, B., Gerstle, J. T., & Azzie, G. (2014, Mayo). Motion analysis in the pediatric laparoscopic surgery (PLS) simulator: validation and potential use in teaching and assessing surgical skills. *Journal of Pediatric Surgery*, 49(5), 791-794. 10.1016/j.jpedsurg.2014.02.063
- Peláez Mata, D., Herrero Álvarez, S., Gómez Sánchez, A., Pérez Egado, L., Corona Bellostas, C., & de Agustín Asensio, J. C. (2021). Curvas de aprendizaje en laparoscopia. *Cir Pediatr*, 34(1), 20-27.
- Schlottmann, F., & Sadava, E. E. (2023). *Cirugía laparoscópica: Técnica quirúrgica en cirugía general* (F. Schlottmann & E. E. Sadava, Eds.). Editorial Médica Panamericana S.A.
- Solís Oviedo, C. J. (2021). *DESARROLLO INSTRUMENTAL PARA LA SIMULACIÓN DE CIRUGÍA DE MÍNIMA INVASIÓN EN RODILLA*. TESIUNAM Digital. <http://132.248.9.195/ptd2021/octubre/0817439/Index.html>
- Texas Instruments. (2012). *TCA9548A Low-Voltage 8-Channel I2C Switch with Reset*. https://www.ti.com/lit/ds/symlink/tca9548a.pdf?ts=1737853835605&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252Fes-mx%252FTCA9548A

Anexo A. Código de Interfaz

```
#Biblioteca de interfaz
import customtkinter as tk
from tkinter import PhotoImage
import os
from datetime import datetime
import time
from tkinter import messagebox
import serial
import numpy as np
import threading
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.gridspec as gridspec
import threading

#Variables de color
color1="#ccfdf9"

# Crear ventana principal
root = tk.CTk()

# Cambiar el texto de la barra de título
root.title("Interfaz CMI")

#Este indica las dimensiones de la ventana
root.geometry("300x450")

#Para ponerle un icono a la venta
root.iconbitmap("C:/Users/DELLG5/Downloads/interfaz (2)/interfaz/cirujano
(2).ico")

#Para que restringir las dimensiones de la pantalla (el usuario no puede
modificar)
#(ancho,alto)
root.resizable(0,0)

#Para cambiar el color de venta y el tipo de cursor
root.config(bg=color1,cursor="arrow")
```

```

# Cargar la imagen de perfil
imagen_perfil = PhotoImage(file="C:/Users/DELLG5/Downloads/interfaz
(2)/interfaz/cirujano (1).png")

# Crear un Canvas con forma redondeada como fondo en la primera ventana
canvas_fondo_redondeado = tk.CTkCanvas(root, width=400, height=400,
bg=color1, highlightthickness=0)

# Mantener la referencia de imagen en el Canvas
canvas_fondo_redondeado.image = imagen_perfil

# Calcular las coordenadas para centrar la imagen en el Canvas
centro_x = canvas_fondo_redondeado.winfo_reqwidth() / 2
centro_y = canvas_fondo_redondeado.winfo_reqheight() * 0.72

# Agregar la imagen de perfil al Canvas en el centro del Canvas
canvas_fondo_redondeado.create_image(centro_x, centro_y,
image=canvas_fondo_redondeado.image)

# Colocar el Canvas en la ventana y en la parte superior central
canvas_fondo_redondeado.place(relx=0.5, rely=0.1, anchor=tk.CENTER)

#Registro
nombre=tk.StringVar()
tarea=tk.StringVar()
#nombre.set("Escribe aqui tu nombre")

#Etiqueta de nombre y su posición en la ventana
#bd indica el borde y font el tipo de fuente
Enombre=tk.CTkLabel(root, text="Nombre", bg_color=color1)
Enombre.place(x=60,y=225)

# Campo de texto
entrada_Nom=tk.CTkEntry(root,textvariable=nombre,bg_color=color1)
entrada_Nom.place(x=110,y=220)

# Variables globales para la lectura de datos
leer_datos = False

```

```

thread_serial = None

# Función para crear una carpeta y guardar los archivos en ella
# Función para crear una carpeta y guardar los archivos en ella
def crear_carpeta_y_guardar(nombre_archivo):

    # Ruta predeterminada para guardar los archivos generados
    ruta_base = "C:/Users/DELLG5/Downloads/interfaz (2)/Resultados"

    # Verificación de existencia de carpeta
    if not os.path.exists(ruta_base):
        os.makedirs(ruta_base)

    # Obtener el nombre sin extensión
    nombre_base = os.path.join(ruta_base,
os.path.splitext(nombre_archivo)[0])

    # Crear una carpeta con el nombre del archivo
    if not os.path.exists(nombre_base):
        os.makedirs(nombre_base)

    # Mover el archivo txt procesado a la carpeta
    archivo_destino = os.path.join(nombre_base, nombre_archivo)
    os.rename(nombre_archivo, archivo_destino)

    # Mover el archivo crudo a la misma carpeta
    nombre_archivo_crudo = nombre_archivo.replace(".txt", "_crudo.txt")
    archivo_destino_crudo = os.path.join(nombre_base,
nombre_archivo_crudo)
    os.rename(nombre_archivo_crudo, archivo_destino_crudo)

    # Guardar las gráficas en la carpeta
    mostrar_grafica_3d_dual(archivo_destino, nombre_base)

    print(f"Archivos guardados en la carpeta: {nombre_base}")

# Función para la matriz de rotación
def rotation_matrix(gama, beta, alfa):

```



```

g = np.radians(gama)
b = np.radians(beta)
a = np.radians(alfa)

R = np.array([[np.cos(a) * np.cos(b) - np.sin(b) * np.sin(g) *
np.sin(a),
              -np.cos(g) * np.sin(a),
              np.cos(a) * np.sin(b) + np.cos(b) * np.sin(g) *
np.sin(a)],
             [np.cos(a) * np.sin(b) * np.sin(g) + np.cos(b) *
np.sin(a),
              np.cos(g) * np.cos(a),
              -np.cos(b) * np.cos(a) * np.sin(g) + np.sin(b) *
np.sin(a)],
             [-np.cos(g) * np.sin(b),
              np.sin(g),
              np.cos(b) * np.cos(g)]]

return R

def leer_y_guardar_datos_seriales(port, baudrate, nombre_archivo):
    global leer_datos
    global thread_serial
    ser = serial.Serial(port, baudrate)
    print(f"Conectado al puerto {port} con baudrate {baudrate}")

    # Inicializa el temporizador
    tiempo_inicial = None

    # Crea un archivo para los datos procesados y otro para los datos
    crudos
    nombre_archivo_crudo = nombre_archivo.replace(".txt", "_crudo.txt")

    with open(nombre_archivo, 'a') as archivo, open(nombre_archivo_crudo,
'a') as archivo_crudo:
        while leer_datos:
            try:
                # Leer una línea del puerto COM

```

```

        line = ser.readline().decode('utf-8',
errors='replace').strip()

# Verificar que la línea tenga 8 elementos separados por
comas

data = line.split(',')
if len(data) != 8:
    print(f"Datos recibidos inválidos: {line}")
    continue

# Guardar los datos crudos en el archivo adicional
archivo_crudo.write(line + "\n")

try:
    # Convertir los datos a flotantes
    data = list(map(float, data))
except ValueError as e:
    print(f"Error de conversión: {e} - Datos recibidos:
{line}")

    continue

# Establece el tiempo inicial en la primera iteración
if tiempo_inicial is None:
    tiempo_inicial = time.time()

# Calcula el tiempo transcurrido desde el inicio en
segundos

tiempo_actual = time.time()
tiempo_transcurrido = tiempo_actual - tiempo_inicial # en
segundos

# Procesamiento de datos para PA1 y PA2 usando la matriz
de rotación

gama1, beta1, alfa1, b21 = data[0], data[1], data[2],
data[3]

gama2, beta2, alfa2, b22 = data[4], data[5], data[6],
data[7]

Piz1 = [[0], [b21], [0]]
R1 = rotation_matrix(gama1, beta1, alfa1)

```

```

RP1 = np.dot(R1, Piz1)
PA1 = RP1 + [[-56.75], [68.147], [-1.75]]

Pde2 = [[0], [b22], [0]]
R2 = rotation_matrix(gama2, beta2, alfa2)
RP2 = np.dot(R2, Pde2)
PA2 = RP2 + [[56.75], [72.088], [1.75]]

# Convertir PA1 y PA2 a una cadena con valores separados
por comas

PA1_str = ','.join([f"{val[0]:.2f}" for val in PA1])
PA2_str = ','.join([f"{val[0]:.2f}" for val in PA2])

# Guarda los datos procesados en el archivo original

archivo.write(f"{PA1_str},{beta1:.2f},{PA2_str},{beta2:.2f},{tiempo_transc
urrido:.2f}\n")

# Imprimir el tiempo transcurrido junto con el vector PA1
y PA2

print(f"Tiempo: {tiempo_transcurrido:.2f} s - Vector PA1:
{PA1}, ángulo de giro: {beta1}")
print(f"Tiempo: {tiempo_transcurrido:.2f} s - Vector PA2:
{PA2}, ángulo de giro: {beta2}")

except KeyboardInterrupt:
    print("Detenido por el usuario.")
    break
except Exception as e:
    print(f"Error: {e}")
    break

ser.close()
print("Puerto cerrado.")

def registro():
    nombre_ingresado = nombre.get()
    if not nombre_ingresado:

```

```

        mensaje_error.configure(text=";Por favor ingrese su nombre!",
text_color="red")
    else:
        mensaje_error.configure(text="") # Borrar el mensaje de error
        print("Registrando a " + nombre_ingresado + "...")
        new_root()

# Crear una etiqueta para mostrar mensajes de error
mensaje_error = tk.CTkLabel(root, text="", bg_color=color1)
mensaje_error.place(x=60, y=250)

# Botón para registrar
boton1 = tk.CTkButton(root, text="Registrar", command=registro,
bg_color=color1)
boton1.place(relx=0.5, rely=0.75, anchor=tk.CENTER)

# Opciones para el Combobox
opciones = ["Lineas horizontales ", "Lineas verticales ", "Triangulo",
"Circulo", "Otro"]
combo = tk.CTkComboBox(root, values=opciones, bg_color=color1)
combo.place(relx=0.6, rely=0.67, anchor=tk.S)
combo.configure(state="readonly")
#Etiqueta de nombre y su posición en la ventana
#bd indica el borde y font el tipo de fuente
EnTarea=tk.CTkLabel(root, text="Prueba", bg_color=color1)
EnTarea.place(x=60,y=275)

# Cargar la imagen de perfil
imagen_perfil = PhotoImage(file="C:/Users/DELLG5/Downloads/interfaz
(2)/interfaz/cirujano (1).png")

def new_root():
    global nueva_ventana, btn3, btn2, btn4

    root.withdraw()
    nueva_ventana = tk.CTkToplevel(root)
    nueva_ventana.title("Interfaz CMI")
    nueva_ventana.geometry("400x400")
    nueva_ventana.iconbitmap("C:/Users/DELLG5/Downloads/interfaz
(2)/interfaz/cirujano (2).ico")

```

```

nueva_ventana.resizable(0, 0)
nueva_ventana.config(bg=color1, cursor="arrow")

# Crear un Canvas con forma redondeada como fondo en la nueva ventana
canvas_fondo_redondeado = tk.CTkCanvas(nueva_ventana, width=400,
height=400, bg=color1, highlightthickness=0)

# Mantener la referencia de imagen en el Canvas de la segunda ventana
canvas_fondo_redondeado.image = imagen_perfil

# Calcular las coordenadas para centrar la imagen en el Canvas
centro_x = canvas_fondo_redondeado.winfo_reqwidth() / 2
centro_y = canvas_fondo_redondeado.winfo_reqheight() * 0.72

# Agregar la imagen de perfil al Canvas en el centro del Canvas
canvas_fondo_redondeado.create_image(centro_x, centro_y,
image=canvas_fondo_redondeado.image)

# Colocar el Canvas en la nueva ventana y en la parte superior central
canvas_fondo_redondeado.place(relx=0.5, rely=0.1, anchor=tk.CENTER)

# Botones y otros elementos de la segunda ventana
#label = tk.CTkLabel(nueva_ventana, text="¡Esta es una nueva
ventana!")
#label.pack()

btn2 = tk.CTkButton(nueva_ventana, text="Iniciar Prueba",
command=simulacion, bg_color=color1)
btn2.place(relx=0.25, rely=0.60, anchor=tk.CENTER)

btn3 = tk.CTkButton(nueva_ventana, text="Regresar", command=regresar,
bg_color=color1)
btn3.place(relx=0.75, rely=0.60, anchor=tk.CENTER)

btn4 = tk.CTkButton(nueva_ventana, text="Finalizar Prueba",
command=finalizar, bg_color=color1)
btn4.place(relx=0.5, rely=0.80, anchor=tk.CENTER)

def cerrar_ventana():
    nueva_ventana.destroy()

```

```

        root.quit()

nueva_ventana.protocol("WM_DELETE_WINDOW", cerrar_ventana)

def regresar():
    root.deiconify()
    nueva_ventana.destroy()

def iniciar_lectura_datos(port, baudrate, nombre_archivo):
    global leer_datos
    global thread_serial
    leer_datos = True
    thread_serial = threading.Thread(target=leer_y_guardar_datos_seriales,
args=(port, baudrate, nombre_archivo))
    thread_serial.start()

def detener_lectura_datos():
    global leer_datos
    leer_datos = False
    if thread_serial is not None:
        thread_serial.join() # Espera a que el hilo termine
    print("Lectura de datos detenida.")

# Modifica la función simulacion para que incluya un temporizador antes de
iniciar la lectura de datos
def simulacion():
    global nombre_archivo_global
    seleccion = combo.get()
    nombre_ingresado = nombre.get()
    print("Iniciando simulación de " + nombre_ingresado + " con la prueba
tipo " + seleccion + "...")

    btn2.configure(state=tk.DISABLED)
    btn3.configure(state=tk.DISABLED)

    fecha_act = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
    nombre_archivo_global =
f"{nombre_ingresado}_{seleccion}_{fecha_act}.txt"

```

```

with open(nombre_archivo_global, "w") as archivo:
    archivo.write(f"Nombre: {nombre_ingresado}\n")
    archivo.write(f"Opción seleccionada: {seleccion}\n")
    archivo.write(f"Fecha de creación: {fecha_act}\n")
    archivo.write(f" Xi      Yi      Zi thetai  Xd      Yd      Zd thetad
Tiempo[s]\n")

    # Iniciar la lectura de datos después de 5 segundos usando
threading.Timer
    temporizador = threading.Timer(5, iniciar_lectura_datos, args=('COM6',
500000, nombre_archivo_global))
    temporizador.start()

# Define la frecuencia de muestreo (en Hz)
frecuencia_muestreo = 55 # 55 muestras por segundo obsevas

def mostrar_grafica_3d_dual(nombre_archivo, carpeta):
    # Crear listas para almacenar los datos
    datos_x1, datos_y1, datos_z1 = [], [], []
    datos_x2, datos_y2, datos_z2 = [], [], []

    # Leer los datos del archivo
    with open(nombre_archivo, "r") as archivo:
        lineas = archivo.readlines()

    for linea in lineas:
        valores = linea.strip().split(',')
        if len(valores) >= 7:
            try:
                # Datos para PA1
                x1, y1, z1 = float(valores[0]), float(valores[1]),
float(valores[2])
                datos_x1.append(z1)
                datos_y1.append(x1)
                datos_z1.append(y1)

                # Datos para PA2
                x2, y2, z2 = float(valores[4]), float(valores[5]),
float(valores[6])

```

```

        datos_x2.append(z2)
        datos_y2.append(x2)
        datos_z2.append(y2)
    except ValueError:
        continue

# Crear la figura con dos subplots en 3D
fig = plt.figure(figsize=(16, 7))
# Título general de la ventana de gráficos
fig.suptitle("Graficas 3D", fontsize=16, fontweight='bold')

# Primer gráfico 3D para PA1
ax1 = fig.add_subplot(121, projection='3d')
ax1.plot(datos_x1, datos_y1, datos_z1, label='PA1', color='purple')
ax1.set_xlim(150, -150)
ax1.set_ylim(50, -200)
ax1.set_zlim(-150, 150)
ax1.set_xlabel('Z (mm)', fontsize=12, fontweight='bold')
ax1.set_ylabel('X (mm)', fontsize=12, fontweight='bold')
ax1.set_zlabel('Y (mm)', fontsize=12, fontweight='bold')
ax1.set_title('Puerto Izquierdo')
ax1.legend()

# Cambia la perspectiva inicial del gráfico 3D
ax1.view_init(elev=30, azim=45)

# Segundo gráfico 3D para PA2
ax2 = fig.add_subplot(122, projection='3d')
ax2.plot(datos_x2, datos_y2, datos_z2, label='PA2', color='orange')
ax2.set_xlim(200, -200)
ax2.set_ylim(250, -100)
ax2.set_zlim(-200, 200)
ax2.set_xlabel('Z (mm)', fontsize=12, fontweight='bold')
ax2.set_ylabel('X (mm)', fontsize=12, fontweight='bold')
ax2.set_zlabel('Y (mm)', fontsize=12, fontweight='bold')
ax2.set_title('Puerto Derecho')
ax2.legend()

# Cambia la perspectiva inicial del gráfico 3D
ax2.view_init(elev=30, azim=45)

plt.tight_layout()

```



```

fig.savefig(os.path.join(carpetas, "grafica_3d_dual.png"))
plt.show()

# Llama a la nueva función en lugar de las funciones anteriores
#crear_carpetas_y_guardar(nombre_archivo_global)

def finalizar():
    # Pausa la lectura de datos
    detener_lectura_datos()

    # Deshabilita el botón de finalizar para evitar múltiples pausas
    btn4.configure(state=tk.DISABLED)

    # Crear una ventana de opciones
    ventana_opciones = tk.CTkToplevel(nueva_ventana)
    ventana_opciones.title("Opciones")
    ventana_opciones.geometry("300x150")
    ventana_opciones.config(bg=color1)

    # Configura la ventana de opciones para que aparezca en primer plano
    sobre la segunda ventana
    ventana_opciones.transient(nueva_ventana) # Hace que la ventana sea
    hija de la segunda ventana
    ventana_opciones.grab_set() # Hace que la ventana de opciones sea
    modal

    # Etiqueta de pregunta
    etiqueta = tk.CTkLabel(ventana_opciones, text="¿Deseas finalizar con
la simulación?", bg_color=color1)
    etiqueta.pack(pady=10)

    # Botón para Continuar con la simulación
    boton_continuar = tk.CTkButton(
        ventana_opciones, text="Sí, finalizar",
        command=lambda:
finalizar_completamente(ventana_opciones),bg_color=color1
    )

```

```

boton_continuar.pack(side="left", padx=20, pady=20)

# Botón para Finalizar completamente la simulación
boton_finalizar = tk.CTkButton(
    ventana_opciones, text="Continuar",
    command=lambda:
continuar_lectura_datos(ventana_opciones),bg_color=color1
)
boton_finalizar.pack(side="right", padx=20, pady=20)

def continuar_lectura_datos(ventana_opciones):
    # Cierra la ventana de opciones y reanuda la lectura de datos
    ventana_opciones.destroy()

    # Vuelve a habilitar el botón de finalizar
    btn4.configure(state=tk.NORMAL)

    global leer_datos
    leer_datos = True
    # Reinicia la lectura de datos en el puerto serial con los parámetros
    iniciales
    thread_serial = threading.Thread(target=leer_y_guardar_datos_seriales,
args=('COM6', 500000, nombre_archivo_global))
    thread_serial.start()

def finalizar_completamente(ventana_opciones):
    # Finaliza completamente la simulación
    ventana_opciones.destroy()

    # Minimiza la segunda ventana antes de mostrar las gráficas
    nueva_ventana.iconify()

    # Guarda los archivos y muestra las gráficas
    crear_carpeta_y_guardar(nombre_archivo_global)

    # Mostrar las gráficas y esperar a que se cierren antes de continuar
    plt.show(block=True)

    # Restaurar la ventana secundaria después de cerrar las gráficas

```

```
nueva_ventana.deiconify()
```

```
root.mainloop()
```

Anexo B. Código de gráficas en MATLAB

```
clear;
close all;
clc;
% Cargar datos
load P8.txt;
datos1 = P8;
% Extraer columnas X, Y, Z
X1 = datos1(:, 5);
Y1 = datos1(:, 6);
Z1 = datos1(:, 7);
% Graficar en 3D
figure
plot3(X1, Y1, Z1, 'LineWidth', 1.5)
grid on
xlabel('X')
ylabel('Y')
zlabel('Z')
title('Trayectoria 3D')
view(3) % Vista en 3D
```