



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

ESCOPE: MEDICIÓN DE DESEMPEÑO POR SEGMENTOS PARA
SIMULADOR DE CIRUGÍA LAPAROSCÓPICA

TESIS
QUE PARA OPTAR POR EL GRADO DE MAESTRÍA EN
CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

PRESENTA:

LUIS ANGEL JUÁREZ VILLALOBOS

TUTORES:

DRA. NIDIYARE HEVIA MONTIEL
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN
SISTEMAS

DR. JORGE LUIS PÉREZ GONZÁLEZ
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN
SISTEMAS

MÉRIDA, YUCATÁN, ABRIL 2022



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Vo. Bo.



Ing. Luis Ángel Juárez Villalobos
Alumno



Dra. Nidivate Hevia Montiel
Directora de tesis



Dr. Jorge Luis Pérez González
Co-Director de tesis

Agradecimiento

Investigación realizada gracias al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) de la UNAM IA102920 y IT100220. Agradezco a la DGAPA-UNAM la beca recibida. Agradezco al CONACYT por la beca de maestría recibida.

Resumen

La cirugía laparoscópica es un procedimiento quirúrgico que en contraste con la cirugía convencional, es menos invasiva, traumática y por lo tanto con una estancia postoperatoria más corta, sin embargo, requiere una gran destreza en el manejo del instrumental y una gran experiencia previa.

Durante el entrenamiento quirúrgico, es importante que el cirujano desarrolle buenas habilidades motoras a lo largo de su entrenamiento. Por esta razón, se han desarrollado varios sistemas de entrenamiento quirúrgico para mejorar estas habilidades. Sin embargo, uno de los grandes retos en estos sistemas de entrenamiento es poder medir objetivamente la capacidad y desempeño de las principales tareas quirúrgicas, donde actualmente solo se obtiene una medición global una vez finalizada la tarea.

Es por esta razón que en este trabajo se muestra el desarrollo del sistema mecánico, electrónico y de realidad virtual del sistema de entrenamiento de cirugía laparoscópica **eScope**. Por otro lado, se propone un nuevo esquema para la evaluación temporal, es decir, una evaluación del desempeño quirúrgico local en diferentes intervalos de tiempo en el entrenamiento de tareas típicas (orientación espacial y traslado de objetos), además, se realizó la comparación con otras tareas típicas (anudado, paso de agujas y sutura).

Se clasificó automáticamente a los cirujanos expertos y no expertos según su desempeño durante el entrenamiento, en base a tres clasificadores: k-vecinos más cercanos (KNN), Bosques aleatorios (RF) y Máquina de soporte vectorial (SVM). A diferencia de otros métodos reportados anteriormente, este trabajo propone un nuevo esquema de evaluación basado en segmentos o intervalos de tiempo, que puede ser un indicador del desempeño local del cirujano dado un conjunto de parámetros de movimiento. El rendimiento de clasificación de los resultados obtenidos fue en precisión 83 % a 100 %, 88 % a 100 % de AUC-ROC, y 88 % a 100 % de F1-Score en la prueba final entre expertos y cirujanos no expertos, donde las máquinas de soporte vectorial presentaron el mejor desempeño. Estos resultados sugieren que este método propuesto por intervalos de tiempo podría usarse en varios entrenadores quirúrgicos para evaluar el desempeño local de un cirujano durante el entrenamiento. Los resultados obtenidos por los algoritmos de clasificación son comparables con lo reportado en el estado del arte, incluso en algunos casos superiores. Por último se concluyó que el sistema **eScope** es una propuesta que puede ser utilizada para fortalecer la mejora continua en el aprendizaje de procedimientos quirúrgicos.

Índice general

Resumen	3
1 Introducción	9
§1.1 Estado del arte	11
§1.2 Simuladores para cirugía laparoscópica	11
§1.2.1 Entrenadores de caja	11
§1.2.2 Entrenadores de realidad virtual	12
§1.3 Medición del desempeño en simuladores de cirugía laparoscópica	15
§1.4 Comparativa entre sistemas de realidad virtual y sistemas de caja	17
§1.5 Planteamiento del problema	18
§1.6 Justificación	18
§1.7 Contribución	18
§1.8 Objetivos	18
§1.8.1 Objetivo General	18
§1.8.2 Objetivos Específicos	18
2 Metodología	20
§2.1 Sistema eScope	20
§2.1.1 Diseño mecánico	21
§2.1.2 Diseño electrónico	23
§2.1.3 Diseño de interfaz gráfica	28
§2.1.4 Diseño de las tareas para el sistema eScope	28
§2.1.5 Programación del simulador laparoscópico	29
§2.1.6 Programación de las tareas del simulador con Unity	30
§2.1.7 Traslado de objetos	33
§2.2 Bases de datos	34
§2.2.1 Base de datos JIGSAWS	34
§2.2.2 Base de datos eScope	35
§2.3 Extracción de características	35
§2.3.1 Variables a partir de los sensores.	36
§2.3.2 Extracción de segmentos	36
§2.3.3 Parámetros de análisis de movimiento	37
§2.4 Clasificación	39
§2.4.1 Selección de características	39
§2.4.2 Métodos de clasificación	39

§2.4.3 Validación del clasificador	41
3 Resultados y discusión	42
§3.1 Sistema eScope	42
§3.1.1 Diseño mecánico	42
§3.1.2 Diseño electrónico	42
§3.1.3 Diseño de interfaz gráfica	43
§3.2 Extracción de características	47
§3.3 Clasificación	47
§3.3.1 Selección de características	47
§3.3.2 Métodos de clasificación	50
§3.3.3 Validación del algoritmo.	53
4 Conclusiones	56
Apéndice I: Dibujos técnicos del diseño mecánico	60
Apéndice II: Programas del sistema eScope	64
§.1 Lectura de sensores en Arduino	64
§.2 Menú principal	66
§.3 Registro del usuario	67
§.4 Calibración del sistema	68
§.5 Configuración del sistema	70
§.6 Movimientos básicos	72
§.7 Tarea de orientación espacial	75
§.8 Tarea de traslado de objetos	77

Índice de figuras

1.1	El sistema de entrenamiento todo en uno FLS incorpora todas las características del sistema de entrenamiento FLS [1].	12
1.2	lapx-X box Pro: Entrenador de caja innovador con posibilidades de medición de métricas y registro de rendimiento [2].	13
1.3	lap-X box: Funciona como un entrenador de cajas tradicional con un paquete de software de grabación, que incluye una base de datos para registrar diferentes usuarios [2].	13
1.4	Lap-X VR: es un simulador laparoscópico de realidad virtual, que incluye módulos básicos, intermedios, avanzados y de procedimientos como apendicectomía, colecistectomía, nefrectomía, etc [2].	14
1.5	Lap-X hibrid: Es un simulador laparoscópico único que combina la funcionalidad de un entrenador de realidad virtual y un entrenador de caja [2].	15
2.1	Diagrama metodológico para el desarrollo e implementación del sistema eScope	20
2.2	Se muestra el dibujo explosionado del sistema eScope , en donde se pueden observar las diferente partes que lo componen.	21
2.3	Mecanismo que se encarga de generar los movimientos para el sistema eScope : A) Vista explosionada, B) Vista ensamblada.	22
2.4	Mecanismo de desplazamiento lineal del sistema eScope	22
2.5	Diseño exterior de la pinza del sistema eScope	23
2.6	Diseño del interior de la pinza para el montaje del circuito electrónico.	23
2.7	Ensamble mecánico general del sistema eScope	23
2.8	Diseño electrónico utilizado en el sistema eScope	24
2.9	Circuito electrónico de la tarjeta de adquisición de datos del sistema eScope	26
2.10	Circuito electrónico de las pinzas laparoscópicas del sistema eScope	26
2.11	Diagrama del diseño de la interfaz gráfica	28
2.12	Diagrama de flujo del programa implementado en Arduino para la lectura y transmisión de datos con la PC (Computadora personal).	29
2.13	Diagrama de flujo principal del sistema eScope en Unity.	30
2.14	Diagrama de flujo de los movimientos básicos de la pinza en Unity.	31
2.15	Diagrama de flujo de tarea de orientación espacial.	32
2.16	Diagrama de flujo de tarea de traslado de objetos.	33
2.17	Tareas quirúrgicas en la base JIGSAWS (de izquierda a derecha): suturar, atar nudos, pasar agujas.	34
2.18	Tareas de orientación espacial (izquierda) y traslado de objetos (derecha).	36

2.19	Un ejemplo representativo de la división en segmentos de una variable cinemática (desplazamientos en $x[n]$, $y[n]$ y $z[n]$), adquirido durante el entrenamiento.	37
2.20	Imágenes representativas de los algoritmos de clasificación: KNN (A), RF (B) y SVM (C).	39
3.1	Ensamble mecánico: componentes que conforman el sistema eScope ensamblado de forma física.	43
3.2	Grados de libertad de la articulación mecánica: A) arriba / abajo, B) izquierda / derecha y C) adentro / afuera.	44
3.3	Tarjeta de adquisición de datos del sistema eScope	45
3.4	Placa electrónica de la pinza del sistema eScope	45
3.5	Flujo principal de la interfaz gráfica: ventana principal (1), registro (2), configuración (3) y calibración (4A-4E).	46
3.6	Interfaz gráfica de la tarea de orientación espacial	46
3.7	Interfaz gráfica de la tarea de traslado de objetos	46
3.8	Ejemplos representativos de la trayectoria del instrumento de cirujanos expertos y no expertos en tareas de (A) anudado, (B) paso de agujas y (C) sutura, de la base de datos JIGSAWS.	47
3.9	Ejemplos representativos de la trayectoria del instrumento de cirujanos expertos y no expertos en tareas de (A) Orientación espacial y (B) Traslado de objetos, de la base de datos eScope	48
3.10	Se muestran los resultados de la clasificación de los segmentos de dos cirujanos, uno experto y otro no experto que realizaron la tarea de atado de nudos de la base de datos JIGSAWS ordenados de forma temporal, se observa la predicción de clase (experto=1, no experto=0) en color azul y la probabilidad de pertenecer a la clase experto en color naranja. La figura 1A, 2A y 3A muestran los resultados de la clasificación de un cirujano no experto que realiza la tarea de atado de nudos, utilizando los clasificadores KNN, RF y SVM respectivamente. La figura 1B, 2B y 3B muestran los resultados de la clasificación de un cirujano experto que realiza la tarea de atado de nudos, utilizando los clasificadores KNN, RF y SVM respectivamente.	51
3.11	Se muestran los resultados de la clasificación de los segmentos de dos cirujanos, uno experto y otro no experto que realizaron la tarea orientación espacial de la base de datos eScope ordenados de forma temporal, se observa la predicción de clase (experto=1, no experto=0) en color azul y la probabilidad de pertenecer a la clase experto en color naranja. La figura 1A, 2A y 3A muestran los resultados de la clasificación de un cirujano no experto que realiza la tarea de atado de nudos, utilizando los clasificadores KNN, RF y SVM respectivamente. La figura 1B, 2B y 3B muestran los resultados de la clasificación de un cirujano experto que realiza la tarea de atado de nudos, utilizando los clasificadores KNN, RF y SVM respectivamente.	52

Índice de tablas

2.1	Características de los sensores utilizados en el sistema electrónico.	25
2.2	Resumen de parámetros de análisis de movimiento (MAP), utilizadas como características en el sistema de medición de desempeño.	38
3.1	Prueba de Mann whitney entre grupos de expertos y no expertos para la base de datos JIGSAWS (p-value)	49
3.2	Prueba de Mann Whitney entre grupos de expertos y no expertos para la base de datos eScope (p-value)	49
3.3	Resultados de rendimiento de los clasificadores para las tareas de la base de datos JIGSAWS (media y desviación estándar).	55
3.4	Resultados de rendimiento de los clasificadores para las tareas de la base de datos eScope (media y desviación estándar).	55

1

Introducción

La Sociedad Americana de Cirujanos de Colon y Recto [3], indican que la cirugía laparoscópica o mínimamente invasiva es una técnica especializada para realizar cirugía, que utiliza varias incisiones de 0.5-1 cm en la pared abdominal llamadas puertos, en cada puerto se inserta un instrumento tubular conocido como trocar y a través de los trocares se pasan instrumentos especializados y una cámara especial conocida durante el procedimiento. Al comienzo del procedimiento, el abdomen se infla con gas de dióxido de carbono para proporcionar un espacio de trabajo y visualización para el cirujano. El laparoscopia transmite imágenes de la cavidad abdominal a monitores de video de alta resolución en la sala de operaciones. Durante la operación, el cirujano observa imágenes detalladas del abdomen en el monitor. Este sistema permite al cirujano realizar las mismas operaciones que la cirugía tradicional pero con incisiones más pequeñas.

Cuesta et al. [4] enuncia que, las ventajas de la cirugía laparoscópica se derivan del hecho de evitar total o parcialmente la existencia de una herida quirúrgica en la pared abdominal, a esto se añade una manipulación visceral más cuidadosa durante el acto quirúrgico, una menor pérdida de sangre y una menor manipulación intestinal. Todos estos factores comportan menor dolor postoperatorio, una más rápida recuperación del tránsito intestinal, reducen la estancia hospitalaria y permiten una rápida reincorporación a las actividades normales y un mejor aspecto estético, proporcionando una menor incidencia y gravedad de las complicaciones de la herida como la infección, la aparición de adherencias y eventración tardía. De forma complementaria, Sietses et al. [5] enuncia que derivado del menor traumatismo quirúrgico, se acompaña una menor inmunodepresión postoperatoria en comparación con la cirugía convencional.

Además, Singla et al. [6] menciona que la cirugía laparoscópica ha experimentado un crecimiento significativo desde principios de la década de 1990 y fue considerado el estándar de atención para muchos procedimientos como por ejemplo la colecistectomía y las intervenciones laparoscópicas se convirtieron en tecnologías más establecidas y representaron más de la mitad de todos los procedimientos de cirugía electiva realizados en pacientes con cáncer de colon en el año 2009. Por otro lado, Laudicella et al. [7] concluye que es probable que la difusión de la laparoscopia haya tenido un impacto en los costos ya que un número creciente de cirujanos logran una mayor experiencia en la realización del nuevo procedimiento, reduciendo así el tiempo de operación y la duración de la estadía del paciente, por este motivo la cirugía laparoscópica es una alternativa que ahorra cos-

tos a la cirugía abierta en pacientes con cáncer de colon y que la adopción de la cirugía laparoscópica puede reducir la estadía en el hospital, la morbilidad y la mortalidad en el tratamiento del cáncer de colon, lo que se traduce en ahorros de costos para el sistema de salud.

Respecto al panorama en México, Chavez et al. [8], en su estudio realizado en 2015, indica que encontró 55 diferentes procedimientos de cirugía general, ginecología y urología realizados a través de abordajes laparoscópicos. Se realizaron un total de 30,174 procedimientos laparoscópicos; de los cuales el 79.7% en mujeres y el 20.3% en hombres. La edad más común en que fueron realizados osciló entre los 25 y los 29 años. Los diez primeros procedimientos fueron colecistectomía, apendicectomía, histerectomía total abdominal laparoscópica, procedimientos para creación de competencia esfinteriana esofagogástrica, salpingooforectomía unilateral, colecistectomía parcial, escisión local o destrucción de ovario, plastia umbilical, apendicectomía incidental y ooforectomía unilateral. La colecistectomía laparoscópica representa el 75.9%. En conjunto estos diez procedimientos corresponden al 94.6% del total de los procedimientos encontrados. El 96% tuvieron estancia normal y el 4% fueron de corta estancia; el 79.35% oscilaron entre los 0 y 3 días de estancia. Por lo tanto, el 20.65% permaneció más de 3 días en hospitalización postoperatoria.

En el mismo estudio realizado por Chavez et al. [8], se menciona que el procedimiento más común ofrecido como corta estancia fue la colecistectomía, seguido de la plastia inguinal y de los procedimientos para creación de competencia esfinteriana esofagogástrica. El 98% fueron egresados por mejoría o curación. Los Estados con mayor número registrado fueron Ciudad de México, Estado de México, Jalisco, Guanajuato y Sonora. El análisis de los resultados del estudio de Gerardo et al. [8] muestra que aún existe poca diversificación en el uso de los abordajes laparoscópicos entre los 55 diferentes procedimientos registrados: el 94.6% se usan en solo diez procedimientos. Es probable que la explicación se deba a múltiples factores: falta de entrenamiento y experiencia en los diferentes procedimientos, falta de disponibilidad de equipo, patologías y procedimientos menos frecuentes, etc. Es un punto crítico que requiere acciones correctivas dentro de un plan estratégico más amplio. se encontró que los procedimientos laparoscópicos en México, aunque seguros, efectivos y aceptados, aún enfrentan retos relacionados con la disponibilidad de elementos organizativos, equipo, infraestructura y entrenamiento, aunque existen diferentes alternativas para vencerlos.

De acuerdo a Chavez et al. [8], una opción viable y demostrada para mejorar el entrenamiento de los cirujanos, ginecólogos y urólogos es el uso de simuladores (de fábrica y auto-fabricados) y de modelos animales. Es por este motivo que existe una evidencia clara que para impulsar el desarrollo de la cirugía laparoscópica en México, se hace necesario el uso de entrenadores de laparoscopia.

1.1. Estado del arte

En esta sección se describen los diversos sistemas de entrenamiento en cirugía laparoscópica, así como las características de cada uno de los sistemas, desarrollados con fines comerciales o para investigación científica, además, se exponen otros sistemas de evaluación de desempeño.

1.2. Simuladores para cirugía laparoscópica

La preparación de un cirujano laparoscopista, ya sea aprendiz o experto presenta obstáculos en su proceso de formación así como el entrenamiento, por ejemplo, la superación del inicio del aprendizaje, presiones económicas, tiempo, presión en cuanto a la menor cantidad de errores para evitar poner en riesgo a los pacientes en una cirugía real. Por lo tanto, es necesario el desarrollo de simuladores con reproducción artificial que tomen en cuenta la visualización, manipulación del instrumento, orientación espacial y en algunos casos simulación de procedimientos quirúrgicos completos y métodos de evaluación. Los simuladores quirúrgicos se pueden clasificar dependiendo de los procedimientos que se vayan a realizar en el entrenador, por ejemplo simuladores con tareas básicas que implican el traslado de objetos de diferente geometría y tamaño, hasta simuladores más especializados donde se pueden realizar procedimientos quirúrgicos completos para el entrenamiento de las habilidades y destrezas del cirujano. Existen diferentes tipos de simuladores y entrenadores laparoscópicos. Cada uno de ellos tienen características particulares, que generalmente se pueden clasificar en entrenadores de caja y entrenadores de realidad virtual, estos simuladores son descritos a continuación.

1.2.1. Entrenadores de caja

Son entrenadores que simulan la cavidad abdominal. La característica principal de este tipo de simuladores es que utilizan componentes básicos, como una videocámara, una fuente de luz y poseen conexión a una computadora para mostrar la imagen en pantalla. Utilizan los mismos instrumentos que en la cirugía real, por que proporcionan una sensación natural. Las ventajas de este tipo de entrenadores es que se puede ocupar cualquier tipo de instrumental laparoscópico con una mejor háptica natural de los mismos. Además, ofrecen una solución económica y portátil que permite a los cirujanos practicar sus habilidades en un ambiente seguro, sin presiones, sin complicaciones y sin límite de tiempo. En contraste, la manera de evaluación a los cirujanos en estos entrenadores se basa en la observación y supervisión de un cirujano experto, donde la calificación es entre el profesor y el alumno. A continuación se describen algunos de estos simuladores

Sistema de entrenamiento *Fundamentals of Laparoscopic Surgery* FLS

El sistema de entrenamiento todo en uno FLS incorpora todas las características del sistema de entrenamiento FLS y también incluye lazos de ligadura, suturas y un kit de instrumentos aprobado por FLS. Este producto proporciona un centro de capacitación con todo lo que necesita para ejecutar un programa de capacitación de FLS, figura 1.1.



Figura 1.1: El sistema de entrenamiento todo en uno FLS incorpora todas las características del sistema de entrenamiento FLS [1].

Lap-X Box Pro

LAP-X Box pro es un entrenador de cajas innovador con posibilidades de medición de métricas y registro de rendimiento. El software LAP-X Box pro puede registrar el desempeño de los usuarios, medir y proporcionar retroalimentaciones instantáneas (puntajes) después del desempeño de los usuarios. Todos los ejercicios de caja validados son compatibles con el LAP-X. El instructor tiene la funcionalidad de pregrabar el ejercicio de ejemplo. Posibilidad de registrar el rendimiento del usuario y tener la evaluación de los instructores en línea. El LAP-X Box pro es actualizable a LAP-X Hybrid, figura 1.2.

Lap-X Box

LAP-X Box funciona como un entrenador de cajas tradicional con un paquete de software de grabación, que incluye una base de datos para registrar diferentes usuarios. El rendimiento del usuario se puede grabar para que el instructor lo evalúe más adelante. Todos los ejercicios de caja validados son compatibles con el LAP-X, figura 4.

1.2.2. Entrenadores de realidad virtual

Existen múltiples entrenadores de realidad virtual, en el campo de la investigación académica, Hong *et al.* [9] describe un sistema de guía háptica desarrollado para ayudar en el entrenamiento basado en modelos de procedimientos laparoscópicos mínimamente invasivos. En el documento, se presenta el escenario de tareas intercambiables y el controlador de lógica difusa con generador de referencia. El sistema, aunque está diseñado para uso fuera de línea, en el laboratorio, tiene un excelente potencial para funciones de asistencia en tiempo real en la sala de operaciones.

Rozenblit *et al.* [10] describen un sistema desarrollado para ayudar en la capacitación basada en modelos de procedimientos laparoscópicos mínimamente invasivos. El concepto de



Figura 1.2: lapx-X box Pro: Entrenador de caja innovador con posibilidades de medición de métricas y registro de rendimiento [2].



Figura 1.3: lapx-X box: Funciona como un entrenador de cajas tradicional con un paquete de software de grabación, que incluye una base de datos para registrar diferentes usuarios [2].



Figura 1.4: Lap-X VR: es un simulador laparoscópico de realidad virtual, que incluye módulos básicos, intermedios, avanzados y de procedimientos como apendicectomía, colecistectomía, nefrectomía, etc [2].

diseño y la arquitectura se presentan con sus elementos principales que facilitan la ejecución guiada (tanto háptica como visual) de las tareas, la evaluación del rendimiento y el análisis comparativo de los resultados. Se dan modelos e implementaciones de software y hardware. Kalavakonda *et al.* [11] proporciona un método para desarrollar un simulador basado en realidad virtual con software de código abierto y un motor de juego. Otra característica incluida en el simulador es el uso de modelos de órganos específicos del paciente. Esto es ventajoso ya que es seguro y menos costoso en comparación con otros simuladores, y también es más flexible debido al uso del motor del juego. Para una simulación realista, se realiza un modelo de interacción herramienta-tejido entre la herramienta y un tumor externo utilizando el método descrito. El modelo de órgano geométrico es reconstruido, después de ser mallado, se integra en el sistema de simulación quirúrgica laparoscópica que consiste en un dispositivo de interfaz háptica y una pantalla gráfica.

Analizando las 2 versiones de simuladores de caja, de la marca Lap-X [2], uno de los simuladores más buscados del mercado, se tienen las siguientes características:

Lap-X VR

LAP-X VR [2] es un simulador laparoscópico de realidad virtual. LAP-X VR proporciona un paquete validado de 5 planes de estudio de dificultad creciente en un entorno de realidad virtual para alcanzar niveles de habilidad de competencia. Los usuarios pueden capacitarse en los siguientes módulos curriculares: módulos básicos, intermedios, avanzados y de procedimientos como apendicectomía, colecistectomía, nefrectomía, etc. Se muestra en la figura 1.4.

Lap-X Hybrid

LAP-X Hybrid es un simulador laparoscópico único que combina la funcionalidad de un entrenador de realidad virtual y un entrenador de caja. Los usuarios pueden entrenarse en el



Figura 1.5: Lap-X hibrid: Es un simulador laparoscópico único que combina la funcionalidad de un entrenador de realidad virtual y un entrenador de caja [2].

modo de realidad virtual en diferentes módulos curriculares: módulos básicos, intermedios, avanzados y de procedimientos como apendicectomía, colecistectomía, nefrectomía, etc. El aprendiz puede cambiar fácilmente al modo entrenador de caja en cualquier momento. Como entrenador de cajas, el LAP-X Hybrid tiene la posibilidad de medir métricas y registrar el rendimiento. Todos los ejercicios de caja validados son compatibles con el LAP-X, proporcionando al usuario una capacitación real de retroalimentación háptica. El instructor tiene la funcionalidad de pregrabar el ejercicio de ejemplo. El usuario puede grabar su propio desempeño y enviarlo al instructor para que lo evalúe en línea [2].

1.3. Medición del desempeño en simuladores de cirugía laparoscópica

Evans *et al.* [12], utiliza sensores de inercia inalámbricos, para la evaluación automatizada de las habilidades quirúrgicas en un simulador de cirugía física laparoscópica. Las métricas de rendimiento se aplicaron a los datos de inercia de sujetos categorizados por experiencia laparoscópica previa: no-expertos (menos de 2 meses); intermedios (menos de 2 años) y expertos (6 a 10 años de experiencia operativa laparoscópica activa). Además, Evans *et al.* [12] muestra que las métricas propuestas basadas en inercia, distinguen entre los 3 grupos lo que demuestra la validez del sistema. Además el método se evaluó contra un sistema de seguimiento de instrumentos ópticos y se comparó favorablemente, lo que demuestra la validez concurrente con un método existente.

Por otro lado, Pérez *et al.* [13], realizaron tres tareas en el sistema de entrenamiento EndoViS. Se capturaron datos de movimiento de los instrumentos. con un sistema de seguimiento de video integrado en el simulador EndoViS, utilizando 13 parámetros de análisis de movimiento (MAP). Se utilizaron redes de funciones de base radial (RBFNets), K-star (K *) y bosques aleatorios (RF) para clasificar a los cirujanos en las puntuaciones de los MAP de todos los participantes. Los tres métodos propuestos demostraron un alto

rendimiento en la clasificación de los cirujanos laparoscópicos, según su nivel de habilidades psicomotoras. Junto con el análisis de movimiento y tres tareas laparoscópicas del programa fundamental de cirugía laparoscópica, estos clasificadores proporcionan un medio para clasificar objetivamente la competencia quirúrgica de los cirujanos para entrenadores de caja laparoscópica existentes.

De forma análoga en el trabajo propuesto por Siyar *et al.* [14], veintitrés neurocirujanos y residentes de neurocirugía senior que comprenden los grupos de expertos y 92 residentes de neurocirugía junior y estudiantes de medicina del grupo "novato". La tarea involucró la eliminación de una serie de tumores cerebrales virtuales sin causar daño al tejido circundante. Se extrajeron más de 100 características y se seleccionaron 68 mediante análisis de prueba T. Estas características fueron proporcionado a 4 clasificadores: KNN, ventana de Parzen, SVM y KNN difuso. La tasa de clasificación se utilizó para evaluar el rendimiento del clasificador, obteniendo el mejor desempeño de tasa de clasificación del 95 %.

Lavachy *et al.* [15] propone un método de aprendizaje automático de tres etapas para automatizar la evaluación de habilidades quirúrgicas en videos de colecistectomía laparoscópica: primero, se entrenó una red neuronal convolucional para identificar y localizar instrumentos quirúrgicos. En segundo lugar, las características de movimiento se extrajeron de las localizaciones de los instrumentos detectados a lo largo del tiempo. En tercer lugar, se entrenó un modelo de regresión lineal basado en las características de movimiento extraídas para predecir las habilidades quirúrgicas. Este enfoque de modelado de tres etapas logró una precisión del $87 \pm 0,2\%$ para distinguir la habilidad quirúrgica buena de la errónea.

Dentro de los trabajos relacionados con el análisis de subtareas está el de Reiley *et al.* [16]. En este artículo se compararon modelos ocultos de Markov discretos para clasificar desempeños de habilidad quirúrgicos a nivel de tarea y gestos (subtareas). Se Aplicaron estas técnicas a 57 conjuntos de datos etiquetados del sistema quirúrgico da Vinci y se lograron niveles de precisión del 100 % en los datos de entrenamiento, 95 % en datos de prueba a nivel de tarea, y 100 % con los modelos de nivel de gestos en la identificación correcta del nivel de habilidad.

En Hassan *et al.* [17], se diseñó una red neuronal convolucional (CNN) para evaluar las habilidades del cirujano extrayendo patrones en los movimientos cinemáticos en cirugía robótica. Se utilizó un conjunto de datos *The JHU-ISI Gesture and Skill Assessment Working Set* (JIGSAWS) y logró resultados muy competitivos con una precisión del 100 % en la sutura y el paso de agujas. En este trabajo, se logró mitigar su efecto de caja negra usando el mapa de activación de clases. Esta característica permite que nuestro método resalte automáticamente qué partes de la tarea quirúrgica influyeron en la predicción de la habilidad y se pueden utilizar para explicar la clasificación y proporcionar retroalimentación personalizada al aprendiz.

En Vedula *et al.* [18] se utilizan modelos de ecuaciones de estimación logística generalizada para clasificar el nivel de habilidad (experto frente a novato). Para este análisis, estudiaron la tarea de incisión cercana dividida en las siguientes maniobras: Sutura, nudo de 2 vueltas, nudo de 1 vuelta y varios gestos necesarios para completar la tarea. Sin embargo, solo clasifican entre cirujanos expertos y no-expertos a nivel de tarea, maniobras y gestos.

1.4. Comparativa entre sistemas de realidad virtual y sistemas de caja

De acuerdo a las recomendaciones de Steigerwald *et al.* [19], es difícil recomendar el uso de un simulador de realidad virtual para la evaluación de habilidades laparoscópicas sin mayor investigación, desarrollo y evaluación, especialmente cuando se toman en cuenta costos. En un nivel práctico y en términos de implementación amplia, la mayoría de los programas de capacitación puede continuar recibiendo un mejor servicio utilizando formas de simulación bien validadas, de menor costo y más factibles, como FLS para entrenamiento y evaluación de habilidades laparoscópicas.

Sin embargo, en el estudio realizado por Guaqueta *et al.* [20] en el que participaron 20 residentes, con edad promedio de 27.5 años. En promedio, los participantes habían realizado previamente 69 laparoscopias, con un mínimo de 0 y un máximo de 210, dato relacionado directamente con el nivel de residencia, con 0 procedimientos previos en los residentes en el nivel 1, y por encima de 155 para los residentes en niveles 3 y superiores. En relación con la lateralidad, 19 de los 20 residentes eran de lateralidad diestra. Existe en la muestra un predominio del sexo masculino, con 14 hombres, y 4 mujeres. El total de la muestra realizó una prueba previa al entrenamiento y una prueba posterior, la cual fue aplicada en 17 residentes para el simulador de caja y 18 para simulador virtual ya que no fue posible que el total de la muestra culminara el estudio. Este estudio indica que el entrenamiento mediante entrenadores de cirugía laparoscópica mejora sustancialmente las habilidades quirúrgicas, sin embargo, los resultados muestran un mayor número de cambios significativos en las destrezas evaluadas en el simulador virtual comparadas con las del simulador de caja.

1.5. Planteamiento del problema

La manera tradicional en que los cirujanos adquieren habilidades quirúrgicas, es entrenando directamente en los pacientes, resultando en un aprendizaje de alto riesgo. Por esta razón, existen actualmente simuladores de laparoscopia comerciales para el aprendizaje y el mejoramiento de las técnicas quirúrgicas.

Uno de los problemas fundamentales en este tipo de simuladores, principalmente en simuladores tradicionales, es que no proporcionan una retroalimentación cuantitativa del desempeño, y requiere del análisis de un experto o en la mayoría de los casos se lleva a cabo una autoevaluación cualitativa. Por otro lado, los simuladores virtuales solo proporcionan una evaluación cuantitativa de manera global. Es por esto, que en este trabajo se plantea la necesidad de crear un sistema que proporcione una medida cuantitativa del desempeño por segmentos durante el entrenamiento.

1.6. Justificación

Los simuladores de cirugía laparoscópica han mostrado un gran desarrollo en los últimos años, sin embargo, se ha observado que todos los simuladores existentes otorgan una medida de desempeño global, es por esto que la principal aportación de este trabajo de investigación es el de otorgar una medida de desempeño local, para que el usuario obtenga una retroalimentación inmediata de sus acciones.

1.7. Contribución

La principal contribución del presente trabajo fue diseñar y construir el sistema de entrenamiento de realidad virtual llamado **eScope**, el cual está constituido de componentes electrónicos, mecánicos, de realidad virtual y aprendizaje computacional. A diferencia de otros simuladores que otorgan medidas de desempeño globales, este sistema proporciona una retroalimentación del desempeño por segmentos temporales en tareas de orientación espacial y traslado de objetos.

1.8. Objetivos

1.8.1. Objetivo General

Diseñar y construir un sistema de entrenamiento virtual de cirugía laparoscópica, que simule tareas de orientación espacial y traslado de objetos, y además proporcione una retroalimentación del desempeño por segmentos mediante algoritmos de aprendizaje computacional.

1.8.2. Objetivos Específicos

Diseño del sistema virtual de cirugía laparoscópica

- Diseñar y construir una pinza acoplada a un sistema mecánico, que permita recrear el entrenamiento de un cirujano para la ejecución de tareas de orientación espacial y traslado de objetos.
- Diseñar e implementar un sistema electrónico robusto, que adquiera y transmita las señales derivadas de los movimientos durante el entrenamiento.
- Desarrollar un sistema de realidad virtual que permita el entrenamiento de tareas de orientación espacial y traslado de objetos.

Estimación y validación automática del desempeño por segmentos

- Extraer características a partir del movimiento de los instrumentos, utilizando los sensores del sistema.
- Entrenar el sistema utilizando técnicas de aprendizaje automático supervisado a partir de las características extraídas.
- Validar el sistema eScope con el conjunto de datos adquiridos y base de datos externa.

2

Metodología

La metodología consta de 4 etapas fundamentales que tienen la finalidad de cumplir con el objetivo general y los objetivos específicos para el desarrollo del sistema de medición y entrenamiento **eScope**, las cuales son: 1) Diseño y construcción del sistema **eScope**; 2) Conformación de las bases de datos; 3) Extracción de características de movimiento; y 4) Clasificación y validación del algoritmo para la estimación del desempeño. En la figura 2.1 se muestra un diagrama detallado de la metodología.

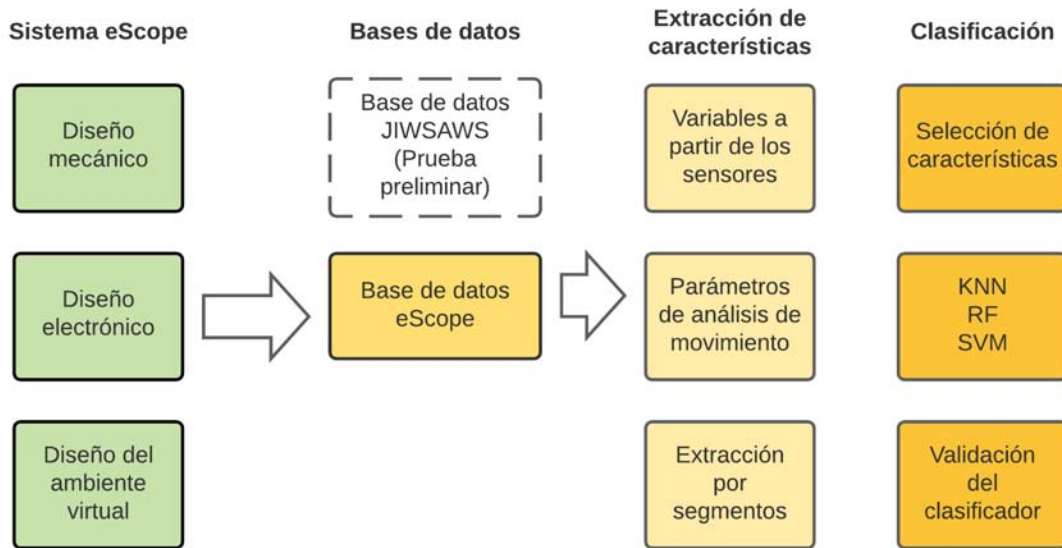


Figura 2.1: Diagrama metodológico para el desarrollo e implementación del sistema **eScope**.

2.1. Sistema eScope

El prototipo del sistema **eScope**, realizado en este trabajo se diseñó con el objetivo de simular virtualmente los movimientos y la sensación de un instrumento de laparoscopia,

permitiendo 4 grados de libertad, así como la apertura y cierre del instrumento. En cuanto al tamaño, el simulador diseñado es portátil, fácil de conectar a computadoras personales para la interacción con la interfaz gráfica, y de fácil calibración, lo que podría proporcionar una ventaja competitiva frente a otros simuladores.

En esta sección se muestra las etapas para el desarrollo del simulador de realidad virtual: diseño mecánico (partes físicas que conforman al simulador laparoscópico, montaje de los sensores y circuitos), diseño electrónico (instrumentación electrónica, sensores, así como la transferencia de información entre el sistema **eScope** y una computadora) y diseño de interfaz gráfica (ambiente virtual e interacción de objetos 3D).

2.1.1. Diseño mecánico

El diseño mecánico se muestra en la figura 2.2 mediante una vista explosionada, donde puede observarse el montaje de cada grupo de piezas que conforman el simulador laparoscópico.

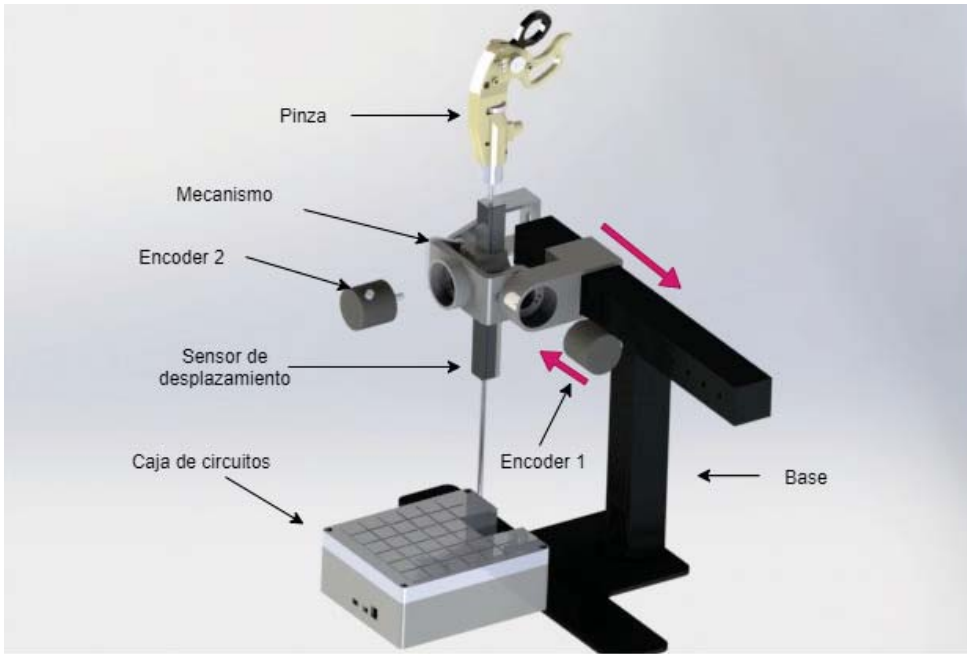


Figura 2.2: Se muestra el dibujo explosionado del sistema **eScope**, en donde se pueden observar las diferentes partes que lo componen.

El diseño de este sistema se realizó en Solid Works 2018, las especificaciones técnicas, así como las dimensiones del diseño propuesto, se encuentran en el Anexo I.

El material utilizado para construir la base es perfil tubular rectangular (PTR) de acero, unidos mediante soldadura. El PTR se fabrica con lámina rollada en caliente por lo que cuenta con una mayor solidez y resistencia para usos estructurales.

Para el diseño del sistema **eScope**, se adicionó una caja de circuitos construida mediante manufactura aditiva, el material de impresión fue ácido poliláctico.

El mecanismo de desplazamiento de la pinza consiste en 3 grados de libertad (DOF): 1) arriba/abajo (guiñada), 2) derecha/izquierda (alabeo) y 3) adelante/atrás (penetración).

El primer DOF se ubica en la base sensado mediante el *encoder* no. 1. El segundo y tercer DOF del movimiento se ejercen mediante el *encoder* no. 2 y el sensor de desplazamiento respectivamente.

En la figura 2.3 se muestra la relación entre los *encoders* (no. 1 y no.2), los soportes (1, 3 y 4), que son estrictamente necesarios para generar el movimiento para el primero y segundo DOF, y el soporte 2 que funciona como refuerzo estructural evitando el deterioro del sistema. En la figura 2.4 se observa la incorporación del sensor de desplazamiento lineal, que permite la medición del tercer DOF.

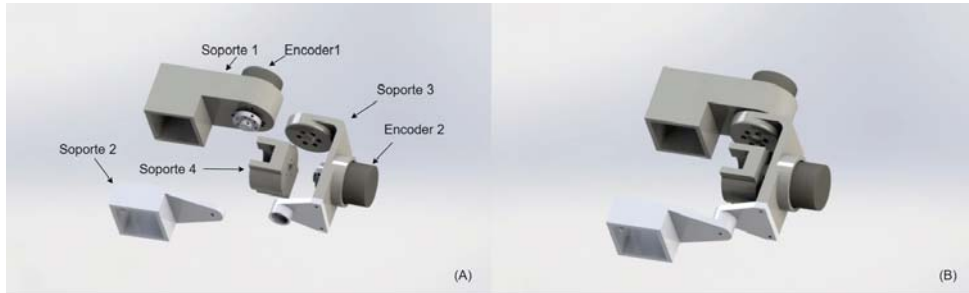


Figura 2.3: Mecanismo que se encarga de generar los movimientos para el sistema **eScope**: A) Vista explosionada, B) Vista ensamblada.



Figura 2.4: Mecanismo de desplazamiento lineal del sistema **eScope**.

La pinza es considerada un componente fundamental en este trabajo porque es la interfaz entre el usuario y el sistema de realidad virtual. La fabricación de este componente se propone mediante manufactura aditiva con un diseño genérico, que incorpore un potenciómetro para medir la abertura del efector final, un sensor de desplazamiento para medir la profundidad de penetración y *encoders* (no. 3 y no. 4) para medir el movimiento del efector final con respecto a la pinza (figura 2.5).

La figura 2.6 muestra el circuito electrónico integrado, necesario para la instrumentación de estos componentes. Es importante mencionar que la pinza se diseñó de tal forma que los circuitos electrónicos se encuentran ensamblados dentro del armazón contemplado dentro del diseño mecánico.

La figura 2.7 muestra el ensamble general desarrollado para el sistema **eScope**, donde pueden observarse la estructura, el mecanismo y la pinza.



Figura 2.5: Diseño exterior de la pinza del sistema eScope.

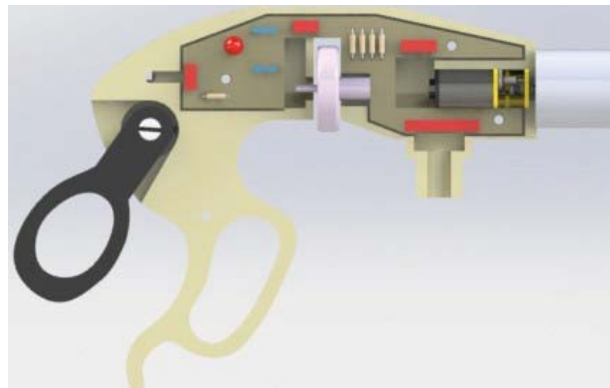


Figura 2.6: Diseño del interior de la pinza para el montaje del circuito electrónico.



Figura 2.7: Ensamble mecánico general del sistema eScope.

2.1.2. Diseño electrónico

La etapa del diseño electrónico, tiene como objetivo tener una forma confiable de poder medir los movimientos que se realizan en el dispositivo **eScope** y transmitirlos a una PC. El diseño electrónico, consta de sensores y un dispositivo de adquisición y procesamiento de datos (Arduino Due). Los sensores son los encargados de medir las variables cinemáticas del dispositivo **eScope** y transformarlas en señales eléctricas, mientras que el Arduino realiza la lectura de las señales de los sensores y permite el envío de información a una PC

utilizando el protocolo USB.

En la figura 2.8, se muestra el esquema electrónico, donde se observan los puertos para la conexión de los *encoders* del mecanismo principal del dispositivo **eScope**, los *encoders* la pinza, el potenciómetro de la pinza y el sensor de desplazamiento.

Es importante mencionar que se agregaron puertos duplicados, de tal forma que el dispositivo sea fácilmente escalable para incorporar un mayor número de sensores que los utilizados en el presente trabajo.

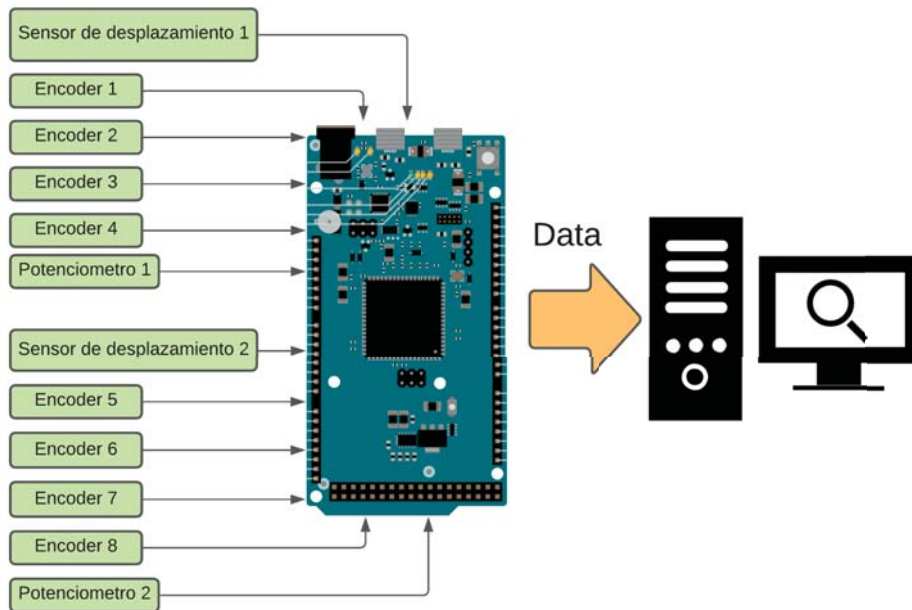








Figura 2.8: Diseño electrónico utilizado en el sistema **eScope**.

En la tabla 2.1 se enlistan los sensores utilizados en el diseño electrónico, además se agregan las variables físicas que son capaces de medir y las características de la señal de salida.

Con el objetivo de poder recrear el comportamiento deseado de la figura 2.8 y tomando en cuenta las características mostradas en la tabla 2.1, se diseñó el circuito de la figura 2.9 mediante el software Eagle CAD. En esta figura se puede observar el Arduino due, usado para procesamiento y la transmisión de datos; las terminales de conexión que permiten el fácil acoplamiento entre el circuito y los sensores, y un convertidor electrónico de potencia de tipo *buck* que se encarga de mantener un voltaje estable en el circuito.

Tabla 2.1: Características de los sensores utilizados en el sistema electrónico.

Dispositivo	Señal de entrada	Características	Imagen
<i>Encoder 1,2,4,5</i>	Desplazamiento Angular	<ul style="list-style-type: none"> ▪ <i>Encoder</i> rotatorio óptico de dos fases. ▪ 400 pulsos/rev. ▪ Salida de colector abierto NPN. ▪ Voltaje de funcionamiento: 3-24 V. 	
<i>Encoder 3,6</i>	Desplazamiento angular	<ul style="list-style-type: none"> ▪ <i>Encoder</i> rotatorio mecánico de dos fases. ▪ 20 pulsos/rev. ▪ Voltaje de funcionamiento: 3-5 V. 	
Sensor de desplazamiento 1 y 2	Desplazamiento lineal	<ul style="list-style-type: none"> ▪ Longitud de medición: 125 mm ▪ Resistencia de 1 $k\Omega$. ▪ Voltaje de funcionamiento: 3-5 V. 	
Micromotor con <i>encoder</i>	Desplazamiento angular	<ul style="list-style-type: none"> ▪ <i>Encoder</i> magnético de dos fases. ▪ 120 pulsos por revolución. ▪ Salida de colector abierto NPN. ▪ Voltaje de funcionamiento 3-5V. 	
Potenciómetro	Desplazamiento angular	<ul style="list-style-type: none"> ▪ Resistencia de 10 $k\Omega$ 	
Arduino Due	Señales eléctricas analógicas y digitales	<ul style="list-style-type: none"> ▪ Microcontrolador: AT91SAM3X8E. ▪ Voltaje de operación: 3.3V. ▪ Voltaje de entrada: 7-12V. ▪ Pines de Entrada/salida: 54. ▪ Entradas analógicas: 12. ▪ Frecuencia de reloj: 84 MHz 	

Por otro lado, se realizó otro circuito para acondicionar las señales de las pinzas. El circuito realizado para las pinzas del sistema **eScope**, es el mostrado en la figura 2.10. En esta figura se puede observar los conectores que permiten la conexión de los sensores, así como la conexión con la placa principal, además de 2 circuitos RC (resistivo capacitivo), que sirven como filtros para poder eliminar las componentes de alta frecuencia de los *encoders*.

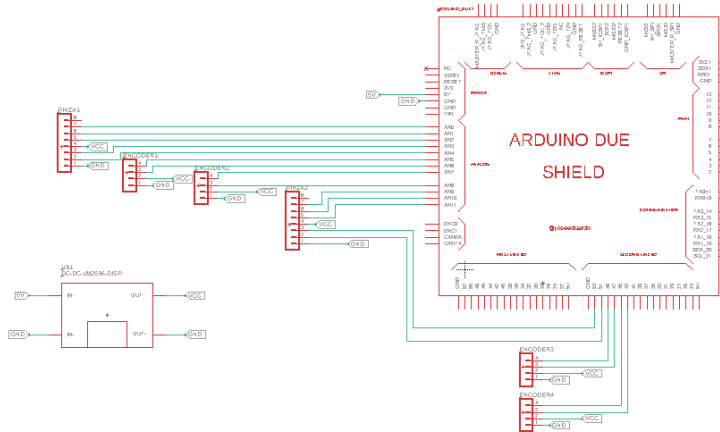


Figura 2.9: Circuito electrónico de la tarjeta de adquisición de datos del sistema **eScope**.

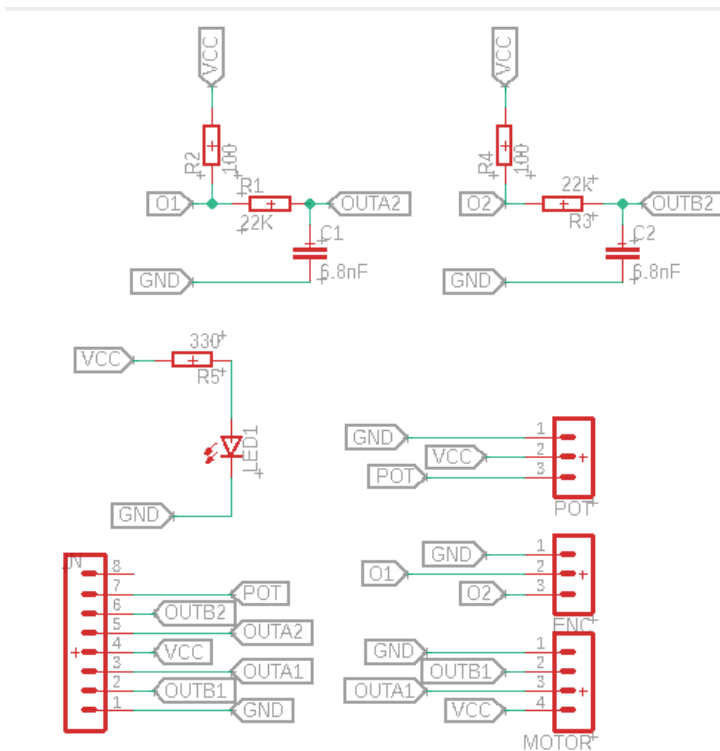


Figura 2.10: Circuito electrónico de las pinzas laparoscópicas del sistema **eScope**.

En la sección de resultados se mostrarán la implementación de la placa de circuito

impreso de los circuitos electrónicos, lo que proporciona suficientemente rigidez para mantener los componentes siempre conectados, disipan bien el calor y permite tener pocas pérdidas de energía eléctrica debido a sus propiedades aislantes.

2.1.3. Diseño de interfaz gráfica

El diseño de la interfaz gráfica en el sistema **eScope**, es la parte del sistema que permite que los movimientos del hardware, sean visualizados mediante un entorno virtual.

En la figura 2.11 se muestra la metodología para el diseño de la interfaz gráfica.

Esta etapa incluye el diseño tridimensional utilizando el software SolidWorks, con el cual se efectuó: el diseño de las piezas del ambiente virtual, el ensamble de estas piezas para formar conjuntos llamados ensamblajes, conformar las relaciones cinemáticas que existen entre cada una de las piezas, además de generar jerarquización y dependencia entre ensamblajes y piezas.

Posteriormente, se aprovecharon las capacidades del programa 3DS MAX para mejorar las características de texturas, los colores y la iluminación de los objetos tridimensionales.

Por último, se empleó el motor de videojuegos Unity para la comunicación con el hardware, y crear las interacciones del modelo tridimensional creado con SolidWorks y 3DS MAX para la obtención del entorno virtual.

El modelo virtual utilizado en esta etapa, es el creado en la sección de diseño mecánico, esto se realizó de esta forma, para mantener una relación directa entre el ambiente real y el ambiente virtual.

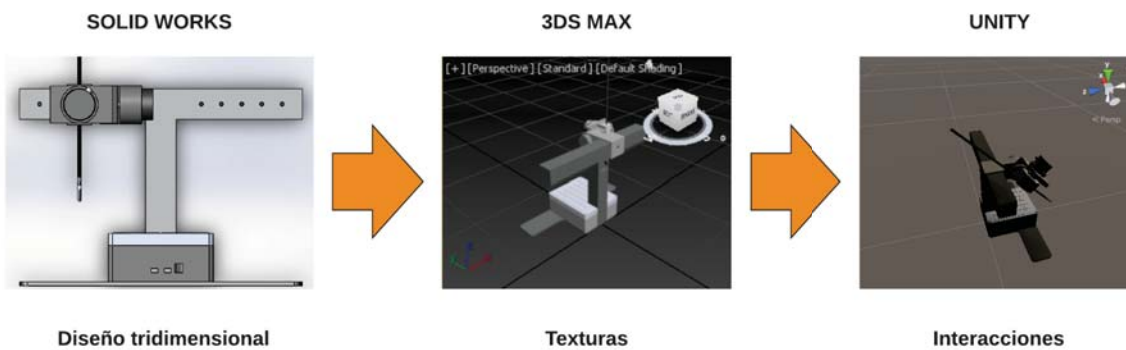


Figura 2.11: Diagrama del diseño de la interfaz gráfica

2.1.4. Diseño de las tareas para el sistema eScope

Las tareas que se programaron para el sistema **eScope**, fueron el ejercicio de orientación espacial y traslado de objetos. El ejercicio de orientación espacial, consiste en, tocar un objeto colocado en una posición espacial con el instrumento laparoscópico virtual para que desaparezca del área de trabajo, enseguida es colocado un nuevo objeto en una nueva posición espacial, para repetir el procedimiento, hasta que se cumplan 10 iteraciones.

La segunda tarea que se programó, fue el ejercicio de traslado de objetos. Esta tarea consiste en trasladar un aro, originalmente situado en una posición espacial, a un poste situado en otra posición espacial, tratando de generar la menor cantidad de movimientos posibles y de no tirar ninguno de los aros fuera del área de trabajo. Una vez situado el

aro en el poste, desaparece el poste y aparece un nuevo poste en otra posición espacial, el procedimiento es repetido 4 veces, en la última iteración, el ejercicio es finalizado.

2.1.5. Programación del simulador laparoscópico

Programación de sensores en la tarjeta arduino

La lectura de los sensores (*encoders* ópticos y potenciómetro) se realizó mediante la programación y configuración en el IDE de Arduino. En la figura 2.12 se muestra el diagrama de flujo del programa realizado en Arduino. En este diagrama se ven declaradas las entradas de cada uno de los sensores, la lectura de los sensores y el envío de datos a la interfaz serial. Es importante observar que el programa en Arduino, se trato de hacer lo mas simple y genérico posible, de tal manera que no sean necesarias posibles configuraciones, y todos los cambios y actualizaciones futuras, sean accesibles en el programa realizado en Unity.

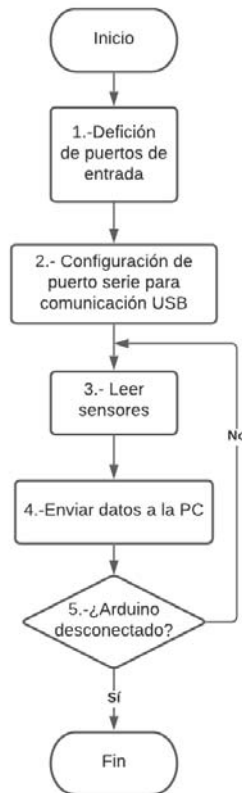


Figura 2.12: Diagrama de flujo del programa implementado en Arduino para la lectura y transmisión de datos con la PC (Computadora personal).

2.1.6. Programación de las tareas del simulador con Unity

La programación de las tareas en Unity se crearon mediante diferentes programas en C#. La figura 2.13 muestra el diagrama de flujo del funcionamiento de todo el programa realizado en Unity. El primer paso es el registro del participante, que se encarga de obtener la información del usuario (Nombre y número de horas de experiencia). Posteriormente, se selecciona el puerto serie para la conexión con el dispositivo, este paso depende del puerto USB donde el dispositivo físico fue conectado. El tercer paso es la calibración del dispositivo, que es un procedimiento para que los movimientos del ambiente virtual correspondan a los movimientos realizados en el dispositivo físico. Una vez realizados los tres pasos (Registro, selección de puerto serie y calibración), las configuraciones están listas para realizar la ejecución de la tarea.

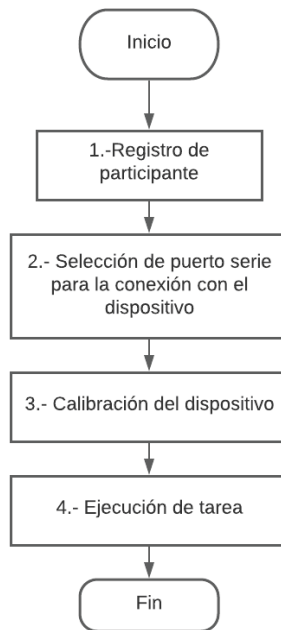


Figura 2.13: Diagrama de flujo principal del sistema **eScope** en Unity.

De forma concurrente a cada tarea, se ejecuta el programa de movimientos básicos, el cual se encarga de realizar los movimientos del instrumento laparoscópico. En la figura 2.14, se muestra el diagrama de flujo, que se sigue para generar estos movimientos.

El primer paso es la configuración de la conexión con Arduino, en la cual se toman los datos obtenidos en el segundo paso del flujo principal y se crea una conexión de tipo serie.

Posteriormente, se abre el puerto, se leen los datos y se muestran en consola (la apertura del puerto se puede repetir en caso de error).

En el quinto paso, se mapean las señales de los sensores en las variables cinemáticas que describen el comportamiento del dispositivo en el ambiente virtual, de tal forma que exista una coherencia entre los movimientos del ambiente real y el ambiente virtual.

Si el ejercicio no ha terminado, se repite el procedimiento hasta el paso 3, en caso contrario, se finaliza el programa.

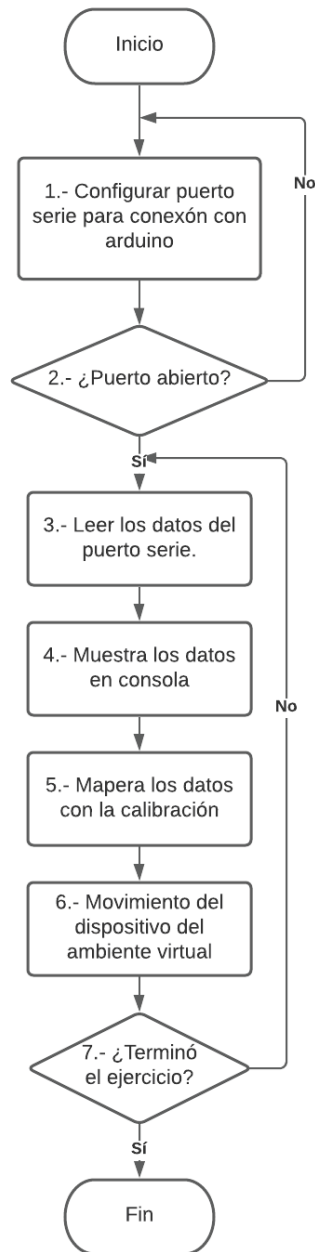


Figura 2.14: Diagrama de flujo de los movimientos básicos de la pinza en Unity.

Orientación espacial

Para el caso de la tarea 1 orientación espacial se programó un *script* basado en *colliders*, para desaparecer cada una de las figuras una vez que son tocadas con la punta de la pinza laparoscópica virtual. El diagrama de flujo del programa de esta tarea se muestra en la figura 2.15.

El primer paso es la creación del objeto (esfera) en una posición *i*-ésima. Posteriormente es ejecutada una condición, para saber si el objeto fue tocado, en caso de que el objeto no haya sido tocado aún, se vuelve a ejecutar la condición. Por otro lado, si el objeto, ya fue tocado (o entró en colisión) por el instrumento laparoscópico, el objeto se desaparece y se vuelve a repetir el procedimiento 10 veces.

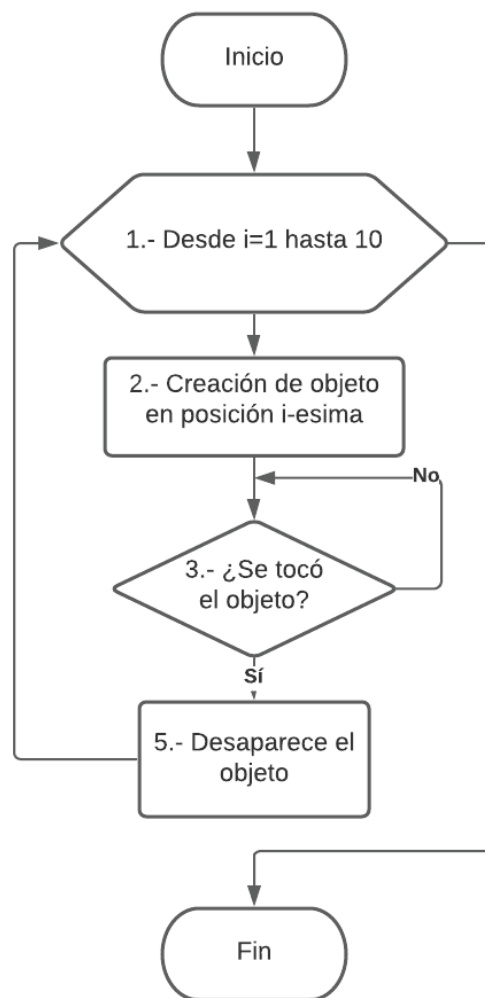


Figura 2.15: Diagrama de flujo de tarea de orientación espacial.

2.1.7. Traslado de objetos

Para la tarea traslado de objetos, se programó un script que permite mover e interactuar con los aros virtuales y la pinza laparoscópica virtual para la transferencia de los mismos de una posición espacial a otra. El diagrama de flujo de esta tarea se muestra en la figura 2.16. El primer paso en la tarea de traslado de objetos, es declarar como cuerpos rígidos a la pinza laparoscópica y al aro, lo que permite una interacción realista entre los dos objetos.

Posteriormente, se ejecuta una condición, la cual pregunta si el aro es sujetado, en caso de que la condición sea verdadera, el padre del aro es NULL, lo cual hace que el movimiento del aro se rija por las interacciones físicas (gravedad, colisiones, etc.). Si el aro está siendo sujetado, la pinza se declara como padre del aro, lo cual hace que el objeto pueda ser cambiado de posición con el movimiento del instrumento laparoscópico. Como tercer paso, se ejecuta una condición para saber si se colocó el aro en la ubicación espacial requerida. El procedimiento se repite 4 veces.

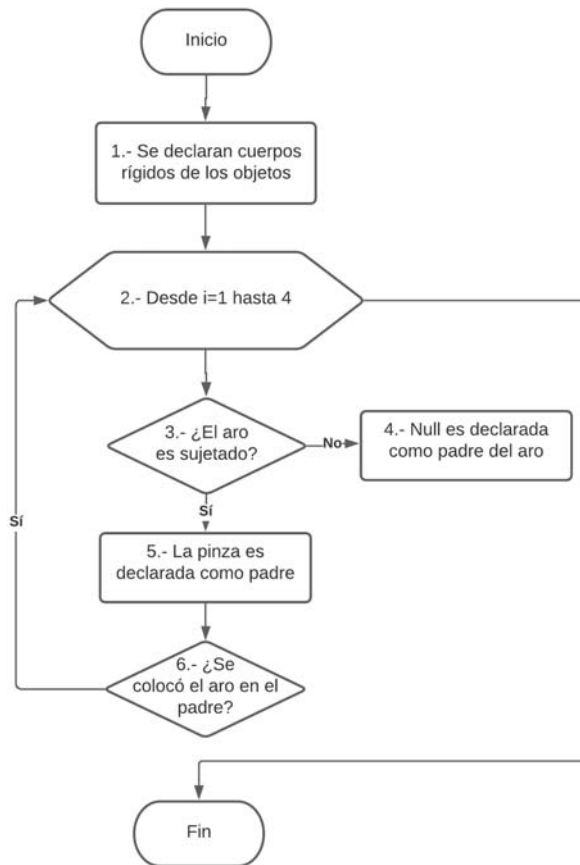


Figura 2.16: Diagrama de flujo de tarea de traslado de objetos.

2.2. Bases de datos

2.2.1. Base de datos JIGSAWS

Para efectuar pruebas preliminares, se utilizó la base de datos *The JHU-ISI Gesture and Skill Assessment Working Set* (JIGSAWS) [21], para posteriormente realizar la validación final con la base de datos adquirida mediante el sistema **eScope**.

La base de datos JIGSAWS, comprende ejercicios de sutura, anudado y paso de agujas con el robot da Vinci. Aunque ya existe un trabajo interesante utilizando esta base de datos, realizado por Narges et al. [22], el objetivo de este trabajo de investigación no es el reconocimiento de gestos, sino la medición del desempeño por segmentos. El conjunto de datos se capturó utilizando el sistema quirúrgico da Vinci y consta de cinemática de 4 cirujanos expertos (mayor a 100 horas de experiencia) y 4 cirujanos no expertos (menor a 10 horas de experiencia) que realizan cinco repeticiones de tres tareas quirúrgicas elementales en un modelo de mesa, además incluye el nivel de desempeño basado en la experiencia.

Tareas

Las tareas, que incluyen suturar, atar nudos y pasar agujas, son componentes estándar de la mayoría de los planes de estudio de capacitación en habilidades quirúrgicas, las tareas se pueden observar en la figura 2.17 y son descritas por el trabajo realizado por Gao et al.[21] de la siguiente manera:



Figura 2.17: Tareas quirúrgicas en la base JIGSAWS (de izquierda a derecha): suturar, atar nudos, pasar agujas.

Sutura

El sujeto toma la aguja, procede a la incisión (designada como una línea vertical en el modelo de mesa) y pasa la aguja a través del tejido, entrando en el punto marcado en un lado de la incisión y saliendo por el punto correspondiente marcado en el otro lado de la incisión. Después del primer paso de la aguja, el sujeto extrae la aguja del tejido, lo pasa a la mano derecha y repite el pase de la aguja tres veces más.

Atar nudos

El sujeto toma un extremo de una sutura atada a un tubo flexible unido por sus extremos a la superficie del modelo de sobremesa, y ata un nudo simple.

Pasar aguja

El sujeto levanta la aguja y la pasa a través de cuatro pequeños aros de metal de derecha a izquierda. Los aros están sujetos a una pequeña altura sobre la superficie del modelo de sobremesa.

2.2.2. Base de datos eScope

Para propósitos de este trabajo, se nombrará base de datos **eScope** al conjunto de datos adquirido utilizando el sistema **eScope** mencionado en este trabajo.

El conjunto de datos **eScope** y consta de cinemática de 5 cirujanos clasificados como expertos y 5 cirujanos no expertos de la localidad de Salina Cruz, Oaxaca. Los cirujanos fueron clasificados de acuerdo al nivel de desempeño basado en la experiencia (menor a 10 horas, es considerado no experto y mayor a 100 hrs, es considerado experto), con 2 repeticiones cada uno por ejercicio, teniendo un total de 20 trayectorias para cada ejercicio (10 para cirujanos expertos y 10 para no expertos).

Las tareas que incluye la base de datos **eScope**, fueron las tareas de orientación espacial y traslado de objetos, las cuales se describen a continuación.

Orientación espacial

En el ejercicio de orientación espacial, participaron 5 cirujanos expertos y 5 cirujanos no expertos, con un total de 2 repeticiones por participante. Este ejercicio consistió en tocar un objeto colocado en una posición espacial, con la punta del instrumento laparoscópico virtual con el propósito de observar la percepción de profundidad y coordinación ojo-mano del usuario. Se capturaron los datos cinemáticos, utilizando el sistema **eScope**.

Traslado de objetos

En el ejercicio de orientación espacial, participaron 5 cirujanos expertos y 5 cirujanos no expertos, con un total de 2 repeticiones por participante. Esta tarea consistió en trasladar los aros de un poste al otro tratando de generar la menor cantidad de movimientos posibles y de no tirar ninguno de los aros fuera del área de trabajo. Se capturaron los datos cinemáticos, utilizando el sistema **eScope**.

2.3. Extracción de características

Para la clasificación local de desempeño, es necesario obtener características de las series de tiempo que nos proporcionen la información para realizar las inferencias y proporcionar una medida de desempeño por segmentos. Esta sección se dividió en 3 etapas, las cuales

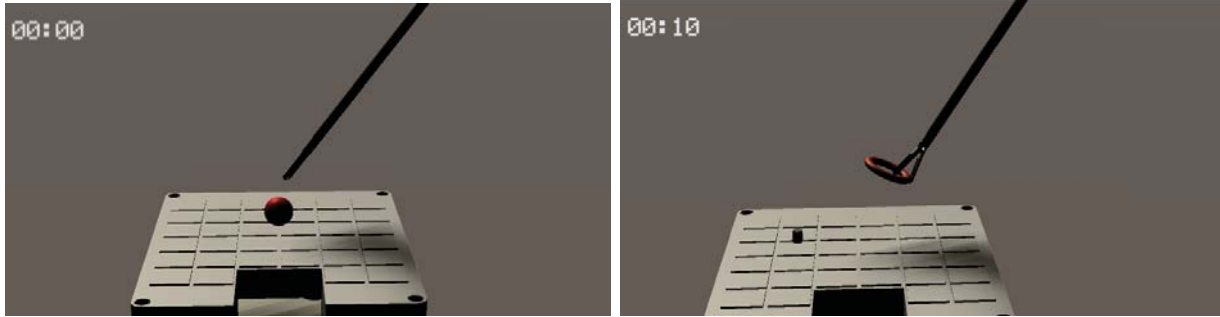


Figura 2.18: Tareas de orientación espacial (izquierda) y traslado de objetos (derecha).

consistieron en la obtención de variables a partir de los sensores, la extracción de segmentos y la obtención de parámetros de análisis de movimiento.

2.3.1. Variables a partir de los sensores.

De forma general, las variables obtenidas de la base de datos **eScope** y la base de datos JIGSAWS, son de naturaleza cinemática y representan las posiciones de la punta del instrumento laparoscópico en coordenadas rectangulares $(x[n], y[n]$ y $z[n])$. La base de datos JIGSAWS ya proporciona estas variables en este espacio de coordenadas; en contraste, para la base de datos **eScope** es necesario realizar una serie de pasos, los cuales se describen a continuación.

Base de datos eScope

El proceso de obtención de las variables de posición en esta base de datos se obtuvo colocando un objeto invisible en la punta del instrumento laparoscópico en la interfaz virtual y colocando la pinza como elemento padre en Unity. De modo que cuando la pinza genera un movimiento, la posición del elemento invisible es la misma que la punta del instrumento. Además, se programó un script en lenguaje *C* para guardar los datos en un archivo *.CSV* compatible con la mayoría de los lenguajes de programación y que además permite el almacenamiento de datos de forma temporal. El periodo de muestreo de los datos obtenidos a partir del sistema **eScope** es de 10 *mS*.

2.3.2. Extracción de segmentos

La base de datos JIGSAWS [21] y la base de datos **eScope**, contienen los datos de movimiento registrados de los instrumentos laparoscópicos.

El conjunto de datos contiene los datos cinemáticos registrados de los instrumentos quirúrgicos para cada tarea. A partir de los datos cinemáticos, los desplazamientos de los ejes x , y y z se obtienen durante el entrenamiento. Para obtener una métrica de rendimiento local, se propone dividir cada tarea quirúrgica realizada (anudado, paso de agujas y sutura) en segmentos temporales superpuestos, utilizando cambios temporales de muestras entre el segmento actual y el siguiente. La figura 2.19 muestra un ejemplo representativo de extracción de segmentos de una variable cinemática.

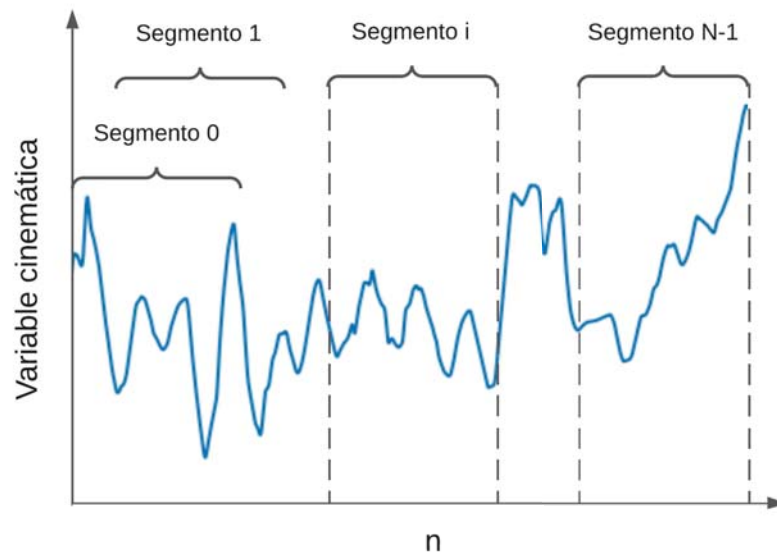


Figura 2.19: Un ejemplo representativo de la división en segmentos de una variable cinemática (desplazamientos en $x[n]$, $y[n]$ y $z[n]$), adquirido durante el entrenamiento.

2.3.3. Parámetros de análisis de movimiento

En este trabajo, se considera que la elección de características informativas, discriminatorias e independientes es un elemento crucial de los algoritmos efectivos en el reconocimiento, clasificación y regresión de patrones.

Es por este motivo que los datos cinemáticos de los segmentos, se analizaron utilizando 10 parámetros de análisis de movimiento (MAP) [13] (tabla 2.2). Estos parámetros se calcularon a partir de los segmentos que contienen información de la posición $x[n]$, $y[n]$, $z[n]$ de los instrumentos y se calcularon mediante software escrito en Python 3.7 tal como se muestra en la tabla ???. La longitud de cada uno de los segmentos es grande comparada con el número de segmentos disponibles, es por esto que se obtuvieron los MAP, con el objetivo de tener un número reducido de características que representen la información de los segmentos de la mejor forma posible. A partir de este momento se le conocerá como MAP a las características extraídas a partir de las variables cinemáticas.

Tabla 2.2: Resumen de parámetros de análisis de movimiento (MAP), utilizadas como características en el sistema de medición de desempeño.

Métrica	Definición	Ecuación
Posición media eje X (AX)	Posición media de la punta del instrumento en el eje X ($\frac{mm}{s^2}$)	$\frac{\int_{t=0}^T x(t)dt}{T}$
Posición media eje Y (AY)	Posición media de la punta del instrumento en el eje Y ($\frac{mm}{s^2}$)	$\frac{\int_{t=0}^T y(t)dt}{T}$
Posición media eje Z (AZ)	Posición media de la punta del instrumento en el eje Z ($\frac{mm}{s^2}$)	$\frac{\int_{t=0}^T z(t)dt}{T}$
Velocidad eje X (VX)	Velocidad de la punta del instrumento en el eje X ($\frac{mm}{s^2}$)	$\frac{dx}{dt}$
Velocidad eje Y (VY)	Velocidad de la punta del instrumento en el eje Y ($\frac{mm}{s^2}$)	$\frac{dy}{dt}$
Velocidad eje Z (VZ)	Velocidad de la punta del instrumento en el eje Z ($\frac{mm}{s^2}$)	$\frac{dz}{dt}$
Longitud de camino (PL)	Ruta total seguida por la punta del instrumento mientras realiza la tarea (mm)	$\int_{t=0}^T \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2} dt$
Percepción de profundidad (DP)	Distancia total recorrida por el instrumento a lo largo del eje (mm)	$\int_{t=0}^T \sqrt{\left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2} dt$
Suavidad de movimiento (MS)	Cambios abruptos en la aceleración que provocan movimientos del instrumento ($\frac{mm}{s^3}$)	$\sqrt{\frac{T^5}{2PL^2} \int_{t=0}^T \left(\left(\frac{d^3x}{dt^3}\right)^2 + \left(\frac{d^3y}{dt^3}\right)^2 + \left(\frac{d^3z}{dt^3}\right)^2 \right)}$
Velocidad media (AV)	Tasa de cambio de la posición del instrumento $\frac{mm}{s}$	$\frac{1}{T} \int_{t=0}^T \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2}$
Aceleración media (AA)	Tasa de cambio de la velocidad del instrumento $\frac{mm}{s^2}$	$\frac{1}{T} \int_{t=0}^T \sqrt{\left(\frac{d^2x}{dt^2}\right)^2 + \left(\frac{d^2y}{dt^2}\right)^2 + \left(\frac{d^2z}{dt^2}\right)^2}$

2.4. Clasificación

Para otorgar las medidas de desempeño local, en este trabajo, se propuso realizar procedimientos de clasificación, con el objetivo de tener la capacidad para poder diferenciar entre segmentos realizados por cirujanos expertos (segmentos con un alto nivel de desempeño) y cirujanos no expertos (bajo nivel de desempeño). Es por esto, que en esta sección se muestra el procedimiento seguido para realizar la clasificación a partir de las MAPS. Este procedimiento se basa en 3 etapas que se mencionan a continuación:

2.4.1. Selección de características

Con el fin de determinar qué MAP se podrían utilizar para entrenar y mejorar el desempeño de los tres clasificadores, se utilizó la prueba U de Mann-Whitney para identificar diferencias estadísticamente significativas en rendimiento entre los experimentados y los no experimentados cirujanos. Se consideró estadísticamente significativa una probabilidad de ($p \leq 0.05$). El análisis estadístico se realizó con Python 3.7.

2.4.2. Métodos de clasificación

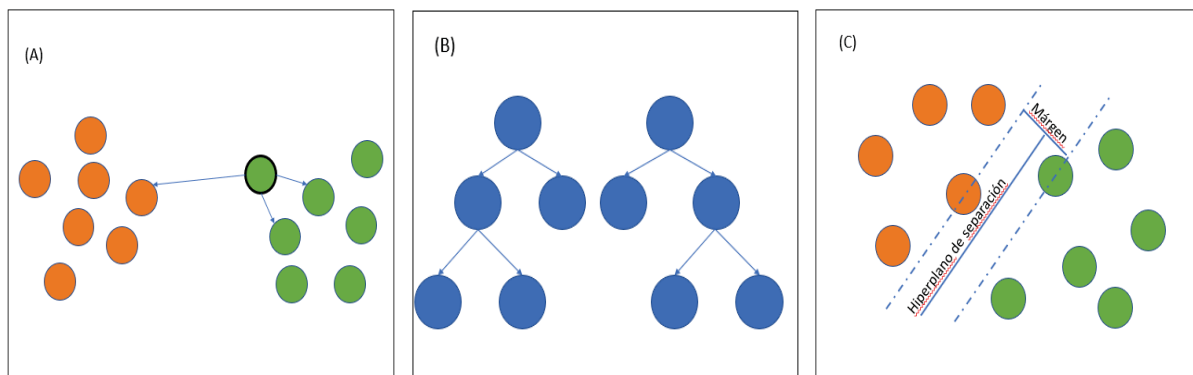


Figura 2.20: Imágenes representativas de los algoritmos de clasificación: KNN (A), RF (B) y SVM (C).

El algoritmo propuesto se basa en la clasificación automática de expertos y no expertos durante las tareas de atado de nudos, paso de agujas y sutura, para el caso de la base de datos JIGSAWS; traslado de objetos y orientación espacial para la base de datos **eScope**. En ambas bases de datos se utilizaron solamente las MAP relevantes obtenidas a partir de la selección de características. Cada base de datos proporciona las tareas etiquetadas si las realiza un cirujano experto o un cirujano no experto, que se utilizó para inferir el rendimiento a lo largo del tiempo (para cada segmento analizado). De esta forma, se asignó una etiqueta binaria a cada segmento extraído, según el nivel de experiencia (etiqueta igual a -1 para las clases de no expertos y +1 para la clase de expertos). Se propuso utilizar tres métodos de clasificación, basados en enfoques diferentes, tal como se muestra en la figura 2.20. K-vecinos mas cercanos (KNN) basado en instancias, bosques aleatorios (RF)

basado en margen de clasificación máximo. A continuación se describen los métodos de clasificación mencionados:

- K-vecinos mas cercanos (KNN) [23]: es un método de clasificación supervisada que sirve para estimar la función de densidad de probabilidad. Este es un método de clasificación no paramétrico, que estima el valor de la función de densidad de probabilidad o directamente la probabilidad a *posteriori* de que un elemento x pertenezca a la clase C_j a partir de la información proporcionada por el conjunto de prototipos. En el proceso de aprendizaje no se hace ninguna suposición acerca de la distribución de las variables predictoras. En el reconocimiento de patrones, el algoritmo KNN es usado como método de clasificación de elementos basado en un entrenamiento mediante ejemplos cercanos en el espacio de los elementos. KNN es un tipo de aprendizaje, donde la función se aproxima solo localmente y todo el cómputo es diferido a la clasificación. Para el entrenamiento de este método de clasificación se utilizaron $n = 3$, donde n es el número de vecinos para hacer la predicción y distancia Euclidiana.
- Bosques aleatorios (RF) [24]: son una combinación de árboles predictores, tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Una vez que se generan los árboles, la predicción de clase para cada entrada se toma como un voto mayoritario de los árboles en el bosque, mejorando así el rendimiento de los árboles de decisión individuales que son propensos al sobreajuste. Para el entrenamiento de este método se utilizaron 100 árboles de decisión, con una profundidad de árbol de 3 niveles y se utilizó el criterio de Gini como métrica de impureza.
- Máquinas de soporte vectorial (SVM) [25]: Es un algoritmo de aprendizaje supervisado que se utiliza en muchos problemas de clasificación y regresión. El objetivo del algoritmo SVM es encontrar un hiperplano que separe de la mejor forma posible dos clases diferentes de puntos de datos. El margen más amplio entre las dos clases como se muestra en la figura 2.20 (A). El margen se define como la anchura máxima de la región paralela al hiperplano que no tiene puntos de datos interiores. El algoritmo solo puede encontrar este hiperplano en problemas que permiten separación lineal; en la mayoría de los problemas prácticos, el algoritmo maximiza el margen flexible permitiendo un pequeño número de clasificaciones erróneas. Para el entrenamiento de este método se utilizó un parámetro de regularización de $C = 1$ con kernel lineal.

Se seleccionaron los métodos de clasificación anteriores debido a que cada uno de estos, está basado en diferentes tipos de enfoque, además que resulta en un particular interés, explorar su desempeño en esta área debido a la complejidad de los conjuntos de datos en este estudio. Como se mencionó anteriormente, los tres métodos de clasificación fueron entrenados para clasificar el desempeño de los participantes en cada segmento de tiempo en las tareas de las bases de datos JIGSAWS y base de datos **eScope**, en función de su desempeño por separado en cada tarea, como cirujano experto o cirujano no experto. Todos los métodos de clasificación se configuraron con sklearn, una colección de software de código abierto de algoritmos de aprendizaje computacional escritos en código Python 3.7. El software sklearn contiene varias herramientas de preprocesamiento, visualización, clasificación, y regresión.

2.4.3. Validación del clasificador

Con el objetivo de evaluar el desempeño de los clasificadores, se agruparon los segmentos de acuerdo a la tarea a la que corresponden, se calculó la moda del resultado de clasificación de cada grupo de segmentos y luego se comparó con las etiquetas globales (experto o no experto) previamente asignadas en la base de datos **eScope** y la base de datos JIGSAWS.

Debido a que la base de datos **eScope** y la base de datos JIGSAWS tienen diferentes números de ejercicios, los parámetros de validación varían ligeramente. A continuación se proporciona la información de la validación realizada para cada base de datos:

Base de datos JIGSAWS

La base de datos se dividió en 70 % de datos de entrenamiento y 30 % de datos para la prueba final. Con los datos de entrenamiento, se realizó un proceso de validación cruzada *10-fold* para las tareas analizadas: se utilizaron las métricas de Precisión (ACC), Puntaje F1 y Área Bajo la Curva de las Características Operativas del Receptor (AUC-ROC) para medir el desempeño de clasificación entre expertos y cirujanos no expertos.

Base de datos eScope

La base de datos se dividió en 70 % de datos de entrenamiento y 30 % de datos para la prueba final. Con los datos de entrenamiento, se realizó un proceso de validación cruzada *5-fold* para las tareas analizadas: se utilizaron las métricas de Precisión (ACC), Puntaje F1 y Área Bajo la Curva de las Características Operativas del Receptor (AUC-ROC) para medir el desempeño de clasificación entre expertos y cirujanos no expertos.

3

Resultados y discusión

En esta sección se muestran los resultados de la implementación y las pruebas del sistema de realidad virtual, lo cual incluye los resultados del diseño mecánico, es decir, la implementación física del dispositivo. Los resultados del diseño electrónico, que incluye la implementación de los circuitos y la instrumentación de los sensores. Los resultados del diseño de la interfaz gráfica, que permite la visualización del ambiente virtual. Además, se muestran los resultados y discusión de la implementación del algoritmo de medición del desempeño por segmentos.

3.1. Sistema eScope

3.1.1. Diseño mecánico

El diseño mecánico final del simulador laparoscópico permite 5 grados de libertad, los cuales son: derecha / izquierda (alabeo), adentro / afuera (cabeceo), arriba / abajo (guiñada) y 2 movimientos en el instrumento que son el giro de la punta sobre su propio eje, así como la apertura / cierre de la punta de la pinza. El dispositivo **eScope** ensamblado se muestra en la figura 3.1.

En la figura 3.2 se muestran los tres grados de libertad del mecanismo del sistema **eScope**, se puede ver los movimientos de arriba/abajo, izquierda/derecha y adentro/afuera. Es importante observar en la figura 3.2, que la punta de la pinza del dispositivo **eScope**, logra cubrir todo el área de trabajo, que para propósitos pragmáticos, se utilizó como el espacio comprendido por el área gris de la caja electrónica, lo que proporciona al instrumento la suficiente libertad de movimiento, para poder realizar las tareas adecuadamente.

3.1.2. Diseño electrónico

A partir del circuito de la tarjeta de adquisición de datos de la figura 2.9 y el circuito de la pinza del sistema **eScope** de la figura 2.10, se diseñaron y se manufacturaron las placas electrónica de la figura 3.3, y la figura 3.4, con el objetivo de tener una buena funcionalidad y un correcto ensamble mecánico. Es importante mencionar, que el circuito de la pinza, fue diseñado de tal forma que ensamblara correctamente con las piezas impresas en 3D. Por otro lado, la placa electrónica de adquisición de datos, se diseñó de tal forma que

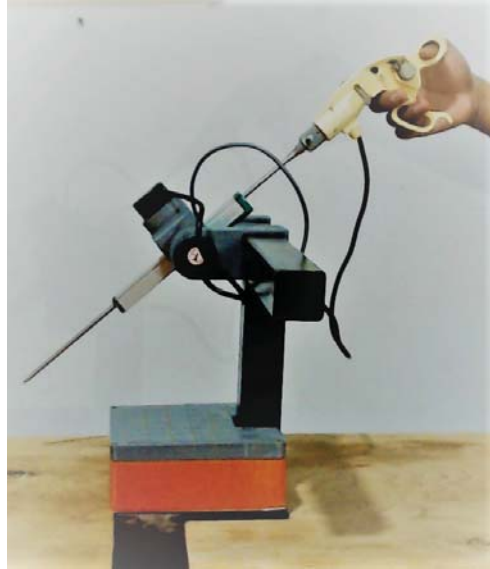


Figura 3.1: Ensamble mecánico: componentes que conforman el sistema **eScope** ensamblado de forma física.

tuviera medidas de 10 cm X 10 cm, esto debido a la optimización de costos.

Es importante mencionar que antes de realizar todas las conexiones, se realizaron pruebas unitarias, de modo que se probaron uno a uno los sensores del sistema para asegurar el correcto funcionamiento de cada uno de ellos.

3.1.3. Diseño de interfaz gráfica

El diseño de la interfaz gráfica tiene como objetivo proporcionar un entorno visual sencillo para permitir la comunicación entre el sistema **eScope** y una computadora personal. Para la interfaz gráfica se añadieron las siguientes ventanas:

Menú principal: Es la ventana con la cual inicia el sistema **eScope** y se considera la ventana principal. En esta ventana se añadieron botones, que proporcionan acceso a las ventanas de calibración, configuración, tarea de traslado de objetos y tarea de orientación espacial.

Registro del usuario: Esta ventana tiene el objetivo de obtener los datos del usuario y almacenarlos en memoria cuando estos datos se requieran. Se colocaron campos de textos para colocar el nombre y el número de horas de experiencia. Además, a la izquierda de cada campo de texto, se agregaron etiquetas que permiten la identificación de cada campo.

Configuración: Esta ventana tiene el objetivo de configurar la conexión del dispositivo **eScope**. Tiene un campo de texto que permite la colocación del puerto serie al cual está conectado el dispositivo **eScope**. Además, se incorporó un botón "conectar" que permite validar la información proporcionada en el campo de texto.

Calibración en 5 pasos: Secuencia de calibración que permite que los movimientos del dispositivo físico, coincidan con los movimientos del dispositivo virtual. Se indica mediante unos sencillos pasos los movimientos que se deben de realizar en el dispositivo **eScope** para poder calibrarlo.

Tarea de orientación espacial: En esta ventana se ejecuta la tarea de orientación espacial,

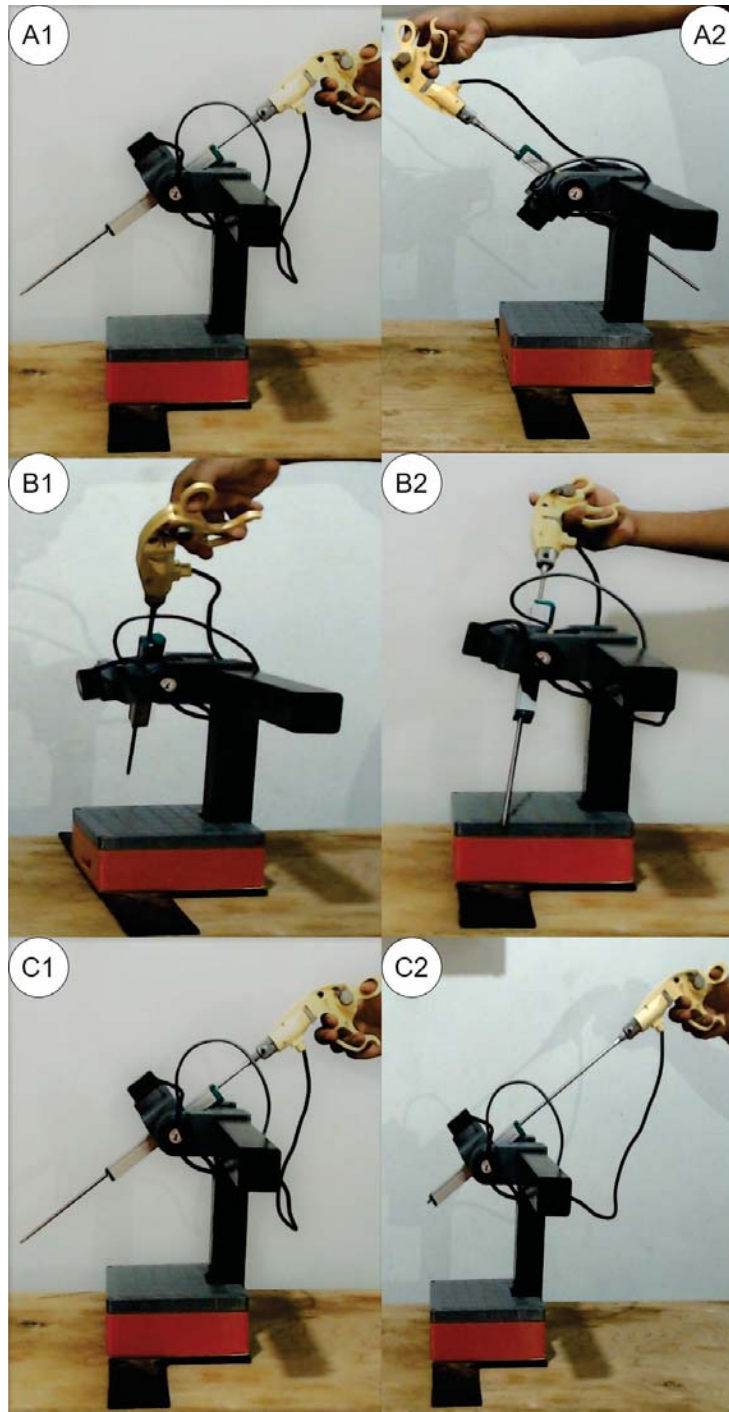


Figura 3.2: Grados de libertad de la articulación mecánica: A) arriba / abajo, B) izquierda / derecha y C) adentro / afuera.

que consistió en el diseño de una herramienta virtual y una esfera, de tal forma que mediante los movimientos del dispositivo eScope se pueda controlar la herramienta virtual, con el objetivo de tocar la esfera.

Tarea de traslado de objetos: En esta ventana se ejecuta la tarea de orientación espacial

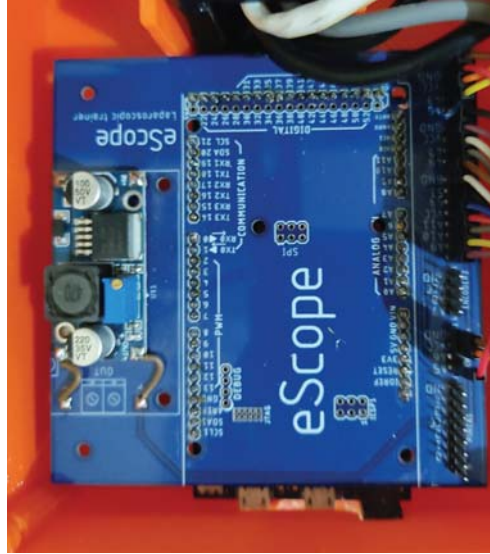


Figura 3.3: Tarjeta de adquisición de datos del sistema eScope.



Figura 3.4: Placa electrónica de la pinza del sistema eScope.

que consistió en el diseño de una herramienta virtual y un aro, de tal forma que mediante los movimientos del dispositivo eScope se pueda controlar la herramienta virtual, con el objetivo de tocar trasladar el aro de un lugar a otro.

En la figura 3.5 se muestran los resultados del diseño de la interfaz gráfica descritos anteriormente.

Por otro lado, la figura 3.6 y la figura 3.7, muestran los resultados de la interfaz gráfica de la tarea de orientación espacial y la tarea de traslado de objetos, respectivamente.

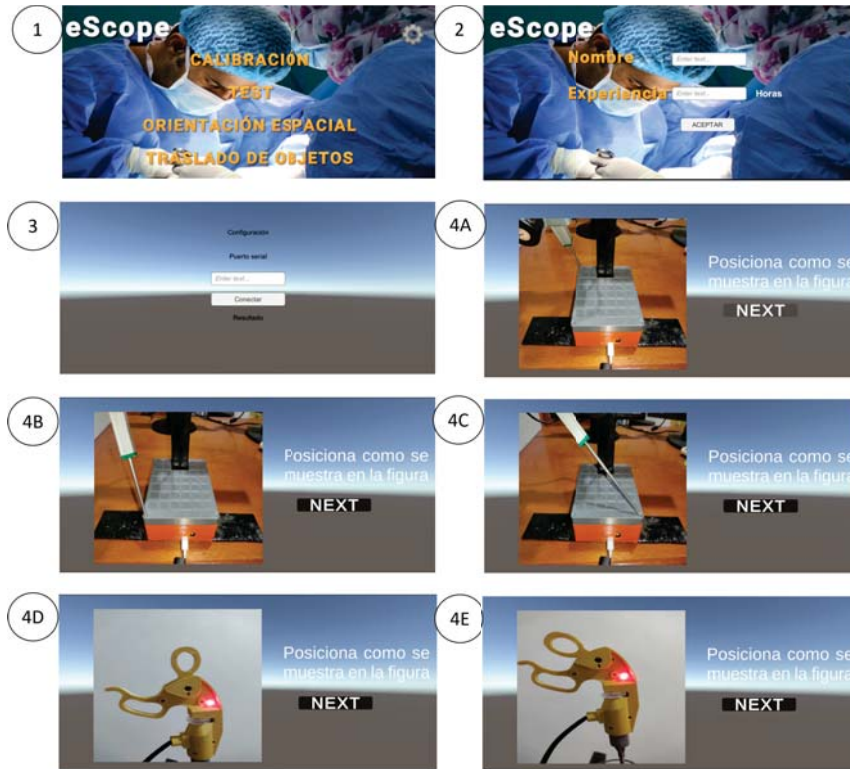


Figura 3.5: Flujo principal de la interfaz gráfica: ventana principal (1), registro (2), configuración (3) y calibración (4A-4E).



Figura 3.6: Interfaz gráfica de la tarea de orientación espacial



Figura 3.7: Interfaz gráfica de la tarea de traslado de objetos

3.2. Extracción de características

Esta sección constó de 3 etapas, las cuales son la obtención de variables a partir de los sensores, los parámetros de análisis de movimiento y la extracción por segmentos. Para esto, se realizó un programa escrito en Python 3.7, el cual fue capaz de leer las bases de datos **eScope** y JIGSAWS, además, se graficaron los datos cinemáticos obtenidos de la base de datos JIGSAWS en la figura 3.8 y los datos cinemáticos de la base de datos **eScope** en la figura 3.9.

Las trayectorias de un cirujano experto y no experto para las tareas de (A) atar nudos, (B) pasar agujas y (C) suturar de la base de datos JIGSAWS se muestran en la figura 3.8. Se puede observar una clara diferencia entre las trayectorias de los cirujanos según los desplazamientos en AX , AY y AZ en las tres tareas estudiadas.

Por otro lado, las trayectorias de un cirujano experto y no experto para las tareas de (A) orientación espacial y (B) traslado de objetos de la base de datos **eScope** se muestran en la figura 3.8. Se puede observar que no existe una clara diferencia entre las trayectorias de los cirujanos según los desplazamientos en AX , AY y AZ en las 2 tareas estudiadas; sin embargo, es importante mencionar que esto es solo observando las trayectorias de forma cualitativa. Posteriormente, se calcularon los parámetros de análisis de movimiento mediante el mismo programa escrito en Python 3.7 y se realizó la extracción de segmentos, para posteriormente calcular los parámetros de análisis de movimiento de cada uno de los segmentos.

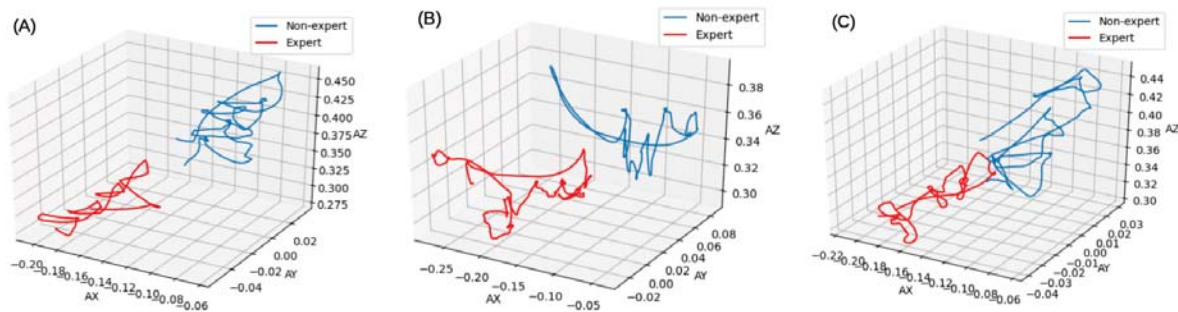


Figura 3.8: Ejemplos representativos de la trayectoria del instrumento de cirujanos expertos y no expertos en tareas de (A) anudado, (B) paso de agujas y (C) sutura, de la base de datos JIGSAWS.

3.3. Clasificación

3.3.1. Selección de características

Con el objetivo de conocer los parámetros de análisis de movimiento que son estadísticamente significativos, se realizaron pruebas de Mann-Whitney en las tareas de la base de datos JIGSAWS, obteniendo los resultados mostrados en la tabla 3.1 y la base de datos **eScope**, obteniendo los resultados mostrados en la tabla 3.2.

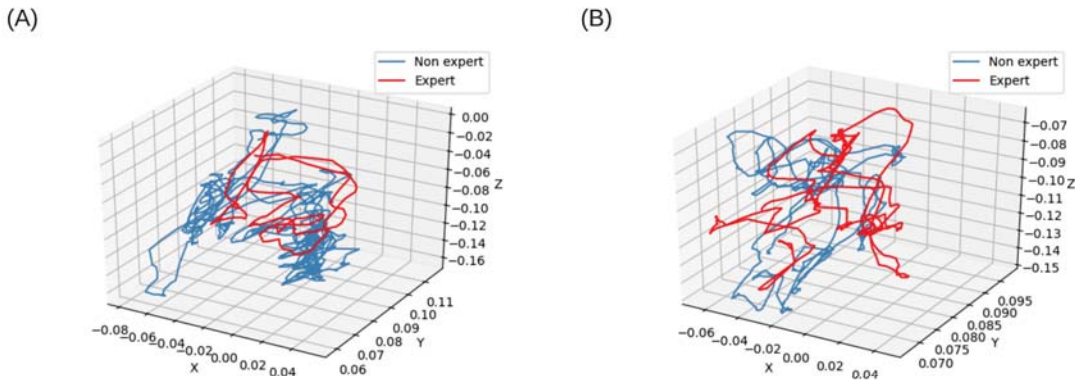


Figura 3.9: Ejemplos representativos de la trayectoria del instrumento de cirujanos expertos y no expertos en tareas de (A) Orientación espacial y (B) Traslado de objetos, de la base de datos **eScope**.

Las pruebas U de Mann-Whitney de la base de datos JIGSAWS, se encontraron diferencias estadísticamente significativas ($p \leq 0.05$) en todas las MAP analizadas para las tareas de atado de nudos, pasar agujas y sutura. Por esta razón, todos los MAP se utilizaron para el entrenamiento de clasificadores.

Para la tarea de atado de nudos en la base de datos **eScope**, las MAPS AX y ADY con valores de $p = 0.1369$ y $p = 0.18$ obtuvieron valores de $p > 0.05$, lo cual indica que las variables AX y ADY, resultaron no significativas; sin embargo, las 9 MAPs restantes, obtuvieron valores de $p \leq 0.05$, por lo cual se consideraron estadísticamente significativas y adecuadas para el entrenamiento de los clasificadores.

Para la tarea de paso de agujas en la base de datos **eScope** se obtuvieron valores de $p > 0.05$ para AX, lo que significa que AX no es estadísticamente significativa; sin embargo, se obtuvieron valores de $p \leq 0.05$ para las 10 MAPs restantes, lo que indica que estas últimas son estadísticamente significativas y adecuadas para el entrenamiento de los clasificadores.

Tabla 3.1: Prueba de Mann whitney entre grupos de expertos y no expertos para la base de datos JIGSAWS (p-value)

MAP	Atar nudos	Pasar agujas	Suturar
Posición media eje X (AX)	0	0	0
Posición media eje Y (AY)	0	0	0
Posición media eje Z (AZ)	0	0	0
Velocidad eje X (ADX)	$1.46 * 10^{-12}$	$1.46 * 10^{-6}$	$1.85 * 10^{-82}$
Velocidad eje Y (ADY)	$5.07 * 10^{-96}$	$7.15 * 10^{-68}$	$1.08 * 10^{-10}$
Velocidad eje Z (ADZ)	$3.62 * 10^{-18}$	$1.14 * 10^{-05}$	$2.61 * 10^{-7}$
Longitud de camino (PL)	0	$1.11 * 10^{-230}$	$2.37 * 10^{-272}$
Percepción de profundidad (DP)	0	$3.95 * 10^{-137}$	$1.31 * 10^{-87}$
Suavidad de movimiento (MS)	0	$7.51 * 10^{-198}$	$1.27 * 10^{-301}$
Velocidad media (AV)	0	$3.12 * 10^{-183}$	$7.32 * 10^{-144}$
Aceleración media (AA)	0	$1.70 * 10^{-38}$	$1.44 * 10^{-37}$

Tabla 3.2: Prueba de Mann Whitney entre grupos de expertos y no expertos para la base de datos **eScope** (p-value)

MAP	Orientación espacial	Traslado de objetos
Posición media eje X (AX)	0.1369	0.1202
Posición media eje Y (AY)	$1.49 * 10^{-99}$	0
Posición media eje Z (AZ)	$2.01 * 10^{-20}$	$1.92 * 10^{-237}$
Velocidad eje X (ADX)	$5.23 * 10^{-44}$	$1.09 * 10^{-42}$
Velocidad eje X (ADY)	0.18	$4.65 * 10^{-4}$
Velocidad eje X (ADZ)	$1.04 * 10^{-5}$	$6.89 * 10^{-8}$
Longitud de camino (PL)	$1.36 * 10^{-119}$	0
Percepción de profundidad (DP)	$1.36 * 10^{-149}$	0
Suavidad de movimiento (MS)	$1.35 * 10^{-299}$	0
Velocidad media (AV)	$1.08 * 10^{-140}$	0
Aceleración media (AA)	$6.49 * 10^{-239}$	0

3.3.2. Métodos de clasificación

En los resultados de la figura 3.10, se muestran de forma comparativa los desempeños de un cirujano no experto y un cirujano experto que realizaron la tarea de atado de nudos de la base de datos JIGSAWS inferidos a partir del algoritmo propuesto. Se muestra el resultado de la clasificación de los segmentos ordenados temporalmente y la probabilidad de que el segmento pertenezca a la clase experto. Es posible observar en esta figura que el clasificador KNN proporciona variaciones abruptas con respecto al tiempo en los resultados de clasificación y probabilidad estimada. Con respecto al clasificador RF, se obtuvo un resultado menos abrupto en el caso de la probabilidad estimada. Por último, el resultado obtenido por el clasificador SVM tiene un comportamiento similar con respecto al tiempo que el obtenido con el clasificador KNN.

En la figura 3.11, se muestran los resultados de clasificación y probabilidad estimada para un cirujano no experto y un cirujano experto que realizaron el ejercicio de orientación espacial de la base de datos **eScope**. Se observa que en el caso del clasificador KNN proporciona un valor de clasificación constante en el cirujano experto y no experto, así como la probabilidad estimada. En los resultados obtenidos por el clasificador RF, se observa que la clasificación de los segmentos se mantiene constante con respecto al tiempo, sin embargo la probabilidad estimada si muestra cambios. Finalmente, en el resultado obtenido por el clasificador SVM, se mantuvo constante la clasificación en el cirujano experto y no experto, sin embargo, la probabilidad estimada mostró cambios con respecto al tiempo.

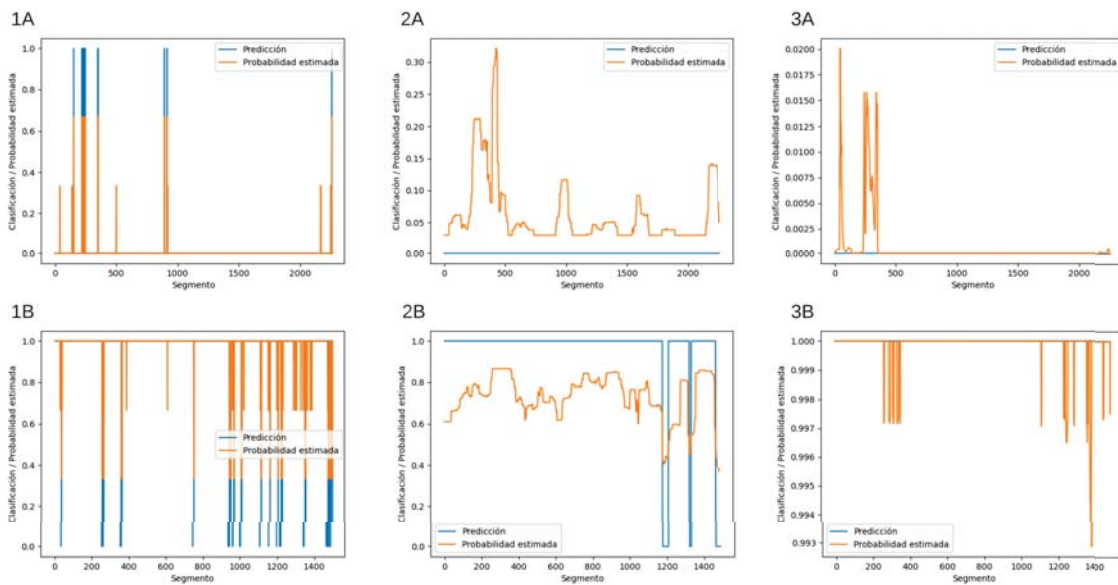


Figura 3.10: Se muestran los resultados de la clasificación de los segmentos de dos cirujanos, uno experto y otro no experto que realizaron la tarea de atado de nudos de la base de datos JIGSAWS ordenados de forma temporal, se observa la predicción de clase (experto=1, no experto=0) en color azul y la probabilidad de pertenecer a la clase experto en color naranja. La figura 1A, 2A y 3A muestran los resultados de la clasificación de un cirujano no experto que realiza la tarea de atado de nudos, utilizando los clasificadores KNN, RF y SVM respectivamente. La figura 1B, 2B y 3B muestran los resultados de la clasificación de un cirujano experto que realiza la tarea de atado de nudos, utilizando los clasificadores KNN, RF y SVM respectivamente.

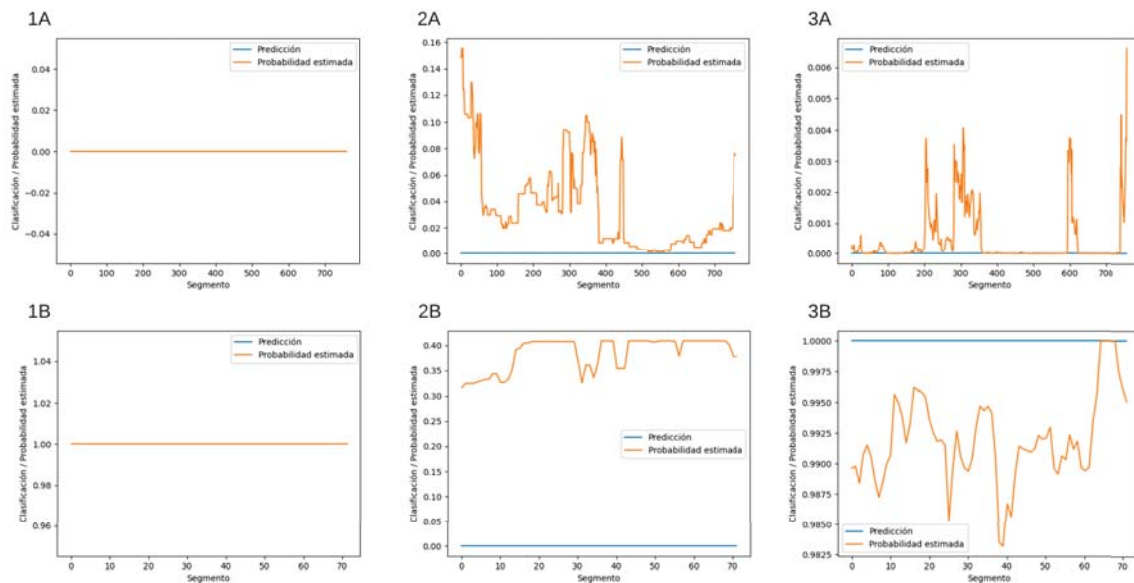


Figura 3.11: Se muestran los resultados de la clasificación de los segmentos de dos cirujanos, uno experto y otro no experto que realizaron la tarea orientación espacial de la base de datos **eScope** ordenados de forma temporal, se observa la predicción de clase (experto=1, no experto=0) en color azul y la probabilidad de pertenecer a la clase experto en color naranja. La figura 1A, 2A y 3A muestran los resultados de la clasificación de un cirujano no experto que realiza la tarea de atado de nudos, utilizando los clasificadores KNN, RF y SVM respectivamente. La figura 1B, 2B y 3B muestran los resultados de la clasificación de un cirujano experto que realiza la tarea de atado de nudos, utilizando los clasificadores KNN, RF y SVM respectivamente.

3.3.3. Validación del algoritmo.

JIGSAWS

Los resultados de la clasificación de las tareas atado de nudos, paso de agujas y sutura de la base de datos JIGSAWS, se presentan en la tabla 3.3. En la primera columna se presentan las tres tareas analizadas; la segunda columna muestra cada clasificador implementado. La primera sección corresponde a la validación cruzada y la última sección de la tabla es la prueba final. En cuanto a la tarea de anudado, se puede notar que en los tres clasificadores (KNN, RF y SVM) se obtuvo el 100 % en todas las métricas para validación cruzada y prueba final. Esto concuerda con la figura 3.8 (A), donde se observan diferencias entre la trayectoria de un cirujano experto y un cirujano no experto. En cuanto a la tarea de pasar agujas, se puede observar que todos los clasificadores obtuvieron un buen desempeño. El clasificador que mostró el mejor desempeño para esta tarea fue SVM con 100 % (para todas las métricas) para validación cruzada y prueba final, seguido de RF con 95 % de tasa de clasificación, 93 % para AUC-ROC y 86 % para puntaje F1 en la etapa de entrenamiento y 100 % en la prueba final para todas las métricas. El clasificador KNN presentó el rendimiento más bajo en la prueba final con 83 % de tasa de clasificación, 88 % para AUC-ROC y 88 % para puntaje F1. Finalmente, para la tarea de sutura, los clasificadores KNN y SVM obtuvieron 100 % en todas las evaluaciones; RF mostró un rendimiento del 97 % de tasa de clasificación, 97 % para AUC-ROC, 96 % para puntaje F1 en la validación cruzada y 100 % en la prueba final para todas las métricas. En general, los rendimientos de clasificación obtenidos muestran ser consistentes con lo mostrado en la figura 3.8, donde se muestran las diferencias visuales entre cirujanos expertos y no expertos.

eScope

Los resultados de la clasificación de las tareas de orientación espacial y traslado de objetos de la base de datos **eScope** se muestran en la tabla 3.4. La primera sección corresponde a la validación cruzada y la última sección de la tabla es la prueba final. En cuanto a la tarea de orientación espacial, se puede observar que el clasificador SVM, obtuvo la mejor clasificación con un 100 % para todas las métricas en validación cruzada y prueba final, seguido por KNN, con 65 % de tasa de clasificación, 62 % para AUC-ROC y 33 % para puntaje F1 en validación cruzada y 88 % de tasa de clasificación, AUC-ROC y puntaje F1 en prueba final y en último lugar el clasificador RF con 60 % de tasa de clasificación, 59 % de AUC-ROC y 27 % de puntaje F1 para validación cruzada y 75 % de tasa de clasificación, AUC-ROC y puntaje F1 para prueba final. En cuanto a la tarea de orientación espacial, se puede observar que el clasificador SVM, obtuvo la mejor clasificación con un 85 % para todas las métricas en validación cruzada y 75 % para todas las métricas en prueba final, seguido por KNN, con 60 % de tasa de clasificación, 58 % para AUC-ROC y 20 % para puntaje F1 en validación cruzada y 75 % para todas las métricas en prueba final y en último lugar el clasificador RF con 55 % de tasa de clasificación, 54 % de AUC-ROC y 12 % de puntaje F1 para validación cruzada y 50 % de tasa de clasificación, AUC-ROC y puntaje F1 para prueba final.

Los resultados obtenidos son comparables a los reportados previamente por Pérez et al. [13], quienes presentaron un enfoque para la clasificación automática entre cirujanos

expertos y no expertos durante el entrenamiento en un simulador de cirugía laparoscópica. Informan un ACC de 93,33 % para K-star, 87,58 % para RBFNet y 84,85 % para clasificadores de RF para la validación final de la prueba. Estos resultados son el desempeño promedio de tres tareas: transferencia de clavijas, corte de patrones y sutura de nudos intracorpóreos. Como puede observarse, los resultados obtenidos por el algoritmo propuesto superan a los reportados por Pérez et al. con tasa de clasificación entre 83 % y 100 % para la validación final. En contraste, a los resultados reportados por estos autores, quienes utilizaron MAP y desempeño global, se propusieron análisis locales y métricas de desempeño por segmentos. Se considera que el análisis basado en segmentos puede proporcionar un indicador del desempeño local durante el entrenamiento quirúrgico.

De igual forma, los resultados obtenidos son comparables con los mostrados por Reiley et al.[16], en donde se comparan dos métodos para identificar el nivel de habilidad de un sujeto al realizar una tarea quirúrgica. En el primer método, se construyeron modelos ocultos de Markov discretos a nivel de tarea. En el segundo método, se construyeron modelos ocultos de Markov discretos de gestos quirúrgicos y la evaluación de habilidad en este nivel. Estas técnicas lograron una tasa de clasificación del 100 % utilizando modelos de nivel de tarea y segmentación de gestos conocidos, 95 % con modelos de nivel de tarea y segmentación de gestos desconocidos, y 100 % con los modelos de nivel de gestos en la identificación correcta del nivel de habilidad.

Además, Vedula et al. [18] utiliza un vocabulario jerárquico semántico para segmentar una tarea quirúrgica de pasar agujas y atar nudos en maniobras y gestos. Se tomo en cuenta el tiempo, el camino longitud y movimientos para la tarea, maniobras y gestos utilizando datos de movimiento de la herramienta. Se utilizaron modelos de regresión logística para predecir la habilidad basada en la experiencia utilizando métricas cuantitativas. Se comparó la métrica AUC-ROC para modelos de nivel de tarea a nivel de maniobra y nivel de gesto. Las AUC fueron 79 % para modelos a nivel de tarea, 78 % para modelos de nivel de maniobra y 70 % para el nivel de gestos.

Finalmente, se realizaron comentarios al sistema **eScope** por parte de los usuarios, donde se discutió acerca de los beneficios sustanciales que tiene el contar con un dispositivo de entrenamiento de cirugía laparoscópica. Además, los participantes, solo habían tenido experiencia con simuladores de caja, debido a que los simuladores de realidad virtual son de difícil acceso, por lo que consideraron que el dispositivo podría ser una opción viable, para que los cirujanos complementen el entrenamiento de habilidades quirúrgicas. En otro sentido, los participantes consideran que una importante área de oportunidad es realizar mejoras al diseño mecánico del sistema **eScope** con el objetivo mejorar la ergonomía y comodidad. Como punto final, los cirujanos mencionaron que el contar con una medición del desempeño por segmentos, podría mejorar considerablemente el desarrollo de habilidades quirúrgicas.

Tabla 3.3: Resultados de rendimiento de los clasificadores para las tareas de la base de datos JIGSAWS (media y desviación estándar).

Tarea	Clasificador	Validación cruzada 10-Fold			Prueba final		
		Tasa de clasificación	AUC-ROC	Puntaje F1	Tasa de clasificación	AUC-ROC	Puntaje F1
Atado de nudos	KNN	100 ± 0 %	100 ± 0 %	100 ± 0 %	100 %	100 %	100 %
	RF	100 ± 0 %	100 ± 0 %	100 ± 0 %	100 %	100 %	100 %
	SVM	100 ± 0 %	100 ± 0 %	100 ± 0 %	100 %	100 %	100 %
Paso de agujas	KNN	100 ± 0 %	100 ± 0 %	100 ± 0 %	83 %	88 %	88 %
	RF	95 ± 16 %	93 ± 19 %	86 ± 38 %	100 %	100 %	100 %
	SVM	100 ± 0 %	100 ± 0 %	100 ± 0 %	100 %	100 %	100 %
Sutura	KNN	100 ± 0 %	100 ± 0 %	100 ± 0 %	100 %	100 %	100 %
	RF	97 ± 11 %	97 ± 9 %	96 ± 12 %	100 %	100 %	100 %
	SVM	100 ± 0 %	100 ± 0 %	100 ± 0 %	100 %	100 %	100 %

Tabla 3.4: Resultados de rendimiento de los clasificadores para las tareas de la base de datos **eScope** (media y desviación estándar).

Tarea	Clasificador	Validación cruzada 10-Fold			Prueba final		
		Tasa de clasificación	AUC-ROC	Puntaje F1	Tasa de clasificación	AUC-ROC	Puntaje F1
Orientación espacial	KNN	65 ± 19 %	62 ± 14 %	33 ± 38 %	88 %	88 %	88 %
	RF	60 ± 23 %	59 ± 12 %	27 ± 33 %	75 %	75 %	75 %
	SVM	100 ± 0 %	100 ± 0 %	100 ± 0 %	100 %	100 %	100 %
Traslado de objetos	KNN	60 ± 16 %	58 ± 17 %	20 ± 40 %	75 %	75 %	75 %
	RF	55 ± 10 %	54 ± 8 %	12 ± 25 %	50 %	50 %	50 %
	SVM	85 ± 19 %	85 ± 17 %	79 ± 25 %	75 %	75 %	75 %

4

Conclusiones

En este trabajo, se presentó la metodología, el diseño y la implementación de un sistema de simulación y entrenamiento de cirugía laparoscópica, mostrando las fases de diseño mecánico, diseño electrónico, diseño del ambiente virtual y evaluación del desempeño mediante aprendizaje computacional. El diseño mecánico se implementó mediante perfil tubular rectangular y manufactura aditiva (impresión 3D) con el objetivo de crear un ensamblaje que permitiera recrear de forma artificial, los movimientos característicos de una cirugía laparoscópica. Además, se logró crear una electrónica robusta que permite la obtención y el procesamiento de señales de los sensores, y transferir esta información mediante protocolo USB. Se realizó la implementación del ambiente virtual, utilizando el software SolidWorks para el modelado de cada uno de los elementos tridimensionales, 3DS MAX para la mejora de texturas y el motor de videojuegos Unity para la comunicación con el hardware, y recrear de forma artificial las tareas de orientación espacial y traslado de objetos. Por otro lado, se mostró un nuevo esquema de evaluación del desempeño de un cirujano durante un entrenamiento en una tarea quirúrgica determinada, con el que se logró la participación, prueba y obtención de datos de 5 cirujanos expertos y 5 cirujanos no expertos, que participaron en las tareas del sistema **eScope**. El 100 % de los cirujanos entrevistados solo habían tenido experiencia con sistemas de entrenamiento de caja, por lo que la propuesta de un sistema de entrenamiento de realidad virtual les pareció bastante interesante. El algoritmo propuesto se basa en un conjunto de parámetros de movimiento que alimentan a tres clasificadores: KNN, RF y SVM con el objetivo de clasificar automáticamente entre cirujanos expertos y no expertos según su desempeño durante el entrenamiento. A diferencia de los trabajos reportados previamente, en este trabajo se propuso un esquema de evaluación por segmentos que proporciona un indicador del desempeño local del cirujano. Los resultados obtenidos por los algoritmos de clasificación son comparables con lo reportado en el estado del arte, incluso en algunos casos superiores. Como áreas de oportunidad, se encuentra la mejora de la ergonomía, la comparativa realizando el entrenamiento del algoritmo con tareas etiquetada a nivel de segmentos con base en parámetros más objetivos que el número de horas y añadir mas ejercicios al sistema. Se considera que esta propuesta puede ser utilizada en simuladores médicos para mejorar la experiencia del usuario y fortalecer la mejora continua en el aprendizaje de procedimientos quirúrgicos.

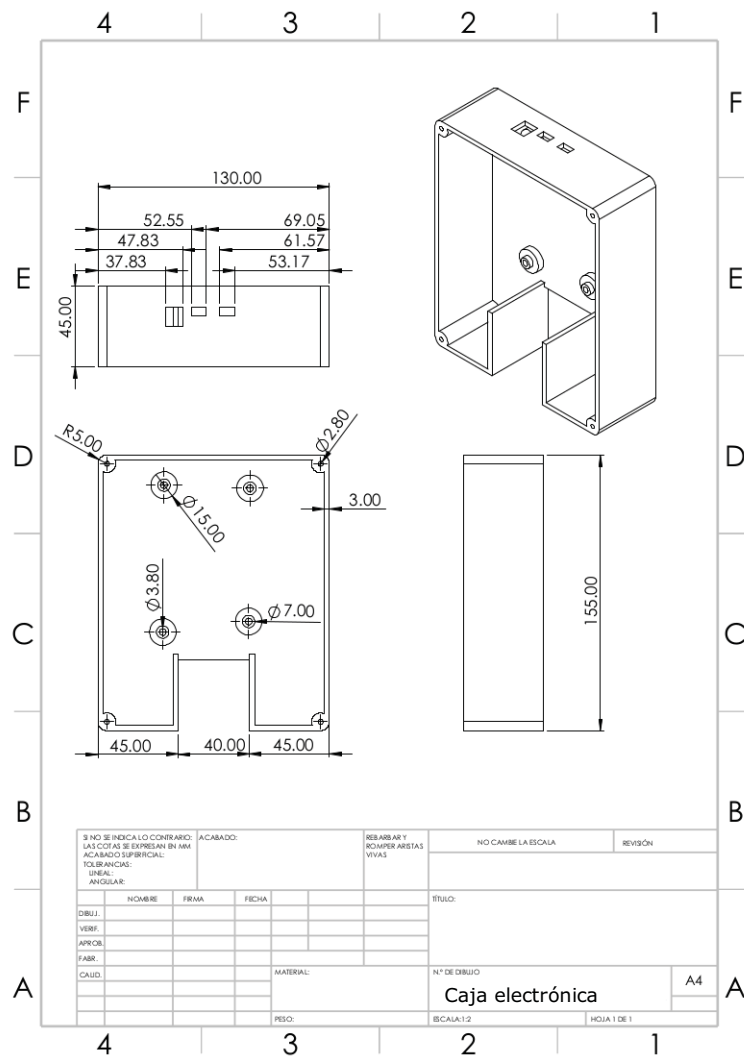
Referencias

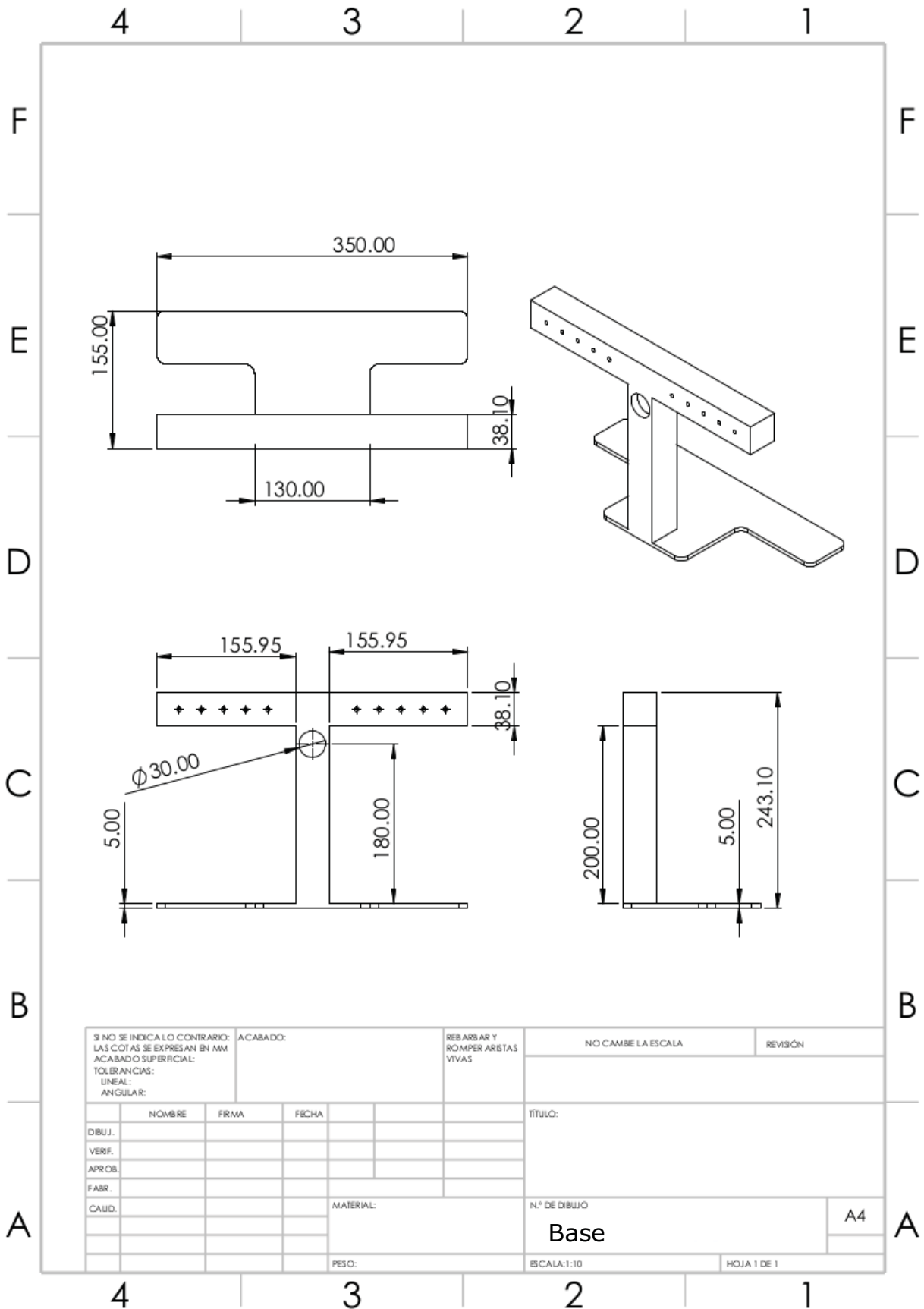
- [1] “FLS Trainer System and Accessories - Fundamentals of Laparoscopic Surgery.”
- [2] “LAP-X, Laparoscopy simulator | MEDICAL-X.”
- [3] “Laparoscopic surgery - what is it? | ascrs.” <https://fascrs.org/patients/diseases-and-conditions/a-z/laparoscopic-surgery-what-is-it>.
- [4] M. Cuesta, “Cirugía laparoscópica,” *Cirugía Española*, vol. 68-4, pp. 420–423, 2000.
- [5] C. Sietses, R. H. J. Beelen, S. Meijer, and M. A. Cuesta, “Immunological consequences of laparoscopic surgery, speculations on the cause and clinical implications,” vol. 384, pp. 250–258, June 1999.
- [6] A. Singla, Y. Li, S. C. Ng, N. G. Csikesz, J. F. Tseng, and S. A. Shah, “Is the growth in laparoscopic surgery reproducible with more complex procedures?,” vol. 146, pp. 367–374, Aug. 2009.
- [7] M. Laudicella, B. Walsh, A. Munasinghe, and O. Faiz, “Impact of laparoscopic versus open surgery on hospital costs for colon cancer: A population-based retrospective cohort study,” *BMJ Open*, vol. 6, p. e012977, nov 2016.
- [8] M.-S. G. Chávez-Saavedra G, Lara-Lona E, Hidalgo-Valadez C, Romero-Salinas N, “Experiencia en procedimientos laparoscópicos en México durante 2015: ¿dónde estamos?,” *Cirugía y Cirujanos*, vol. 87, pp. 292–298, 2019.
- [9] M. Hong and J. W. Rozenblit, “Haptic guidance with fuzzy control in simulation-based surgical training,” in *Proceedings of the Symposium on Modeling and Simulation in Medicine*, MSM '15, (San Diego, CA, USA), p. 25–30, Society for Computer Simulation International, 2015.
- [10] J. W. Rozenblit, C. Feng, M. Riojas, L. Napalkova, A. J. Hamilton, M. Hong, P. Berthet-Rayne, P. Czapiewski, G. Hwang, J. Nikodem, and et al., “The computer assisted surgical trainer: Design, models, and implementation,” in *Proceedings of the 2014 Summer Simulation Multiconference*, SummerSim '14, (San Diego, CA, USA), Society for Computer Simulation International, 2014.
- [11] N. Kalavakonda, S. Chandra, and A. Thondiyath, “Development of virtual reality based robotic surgical trainer for patient-specific deformable anatomy,” in *Proceedings*

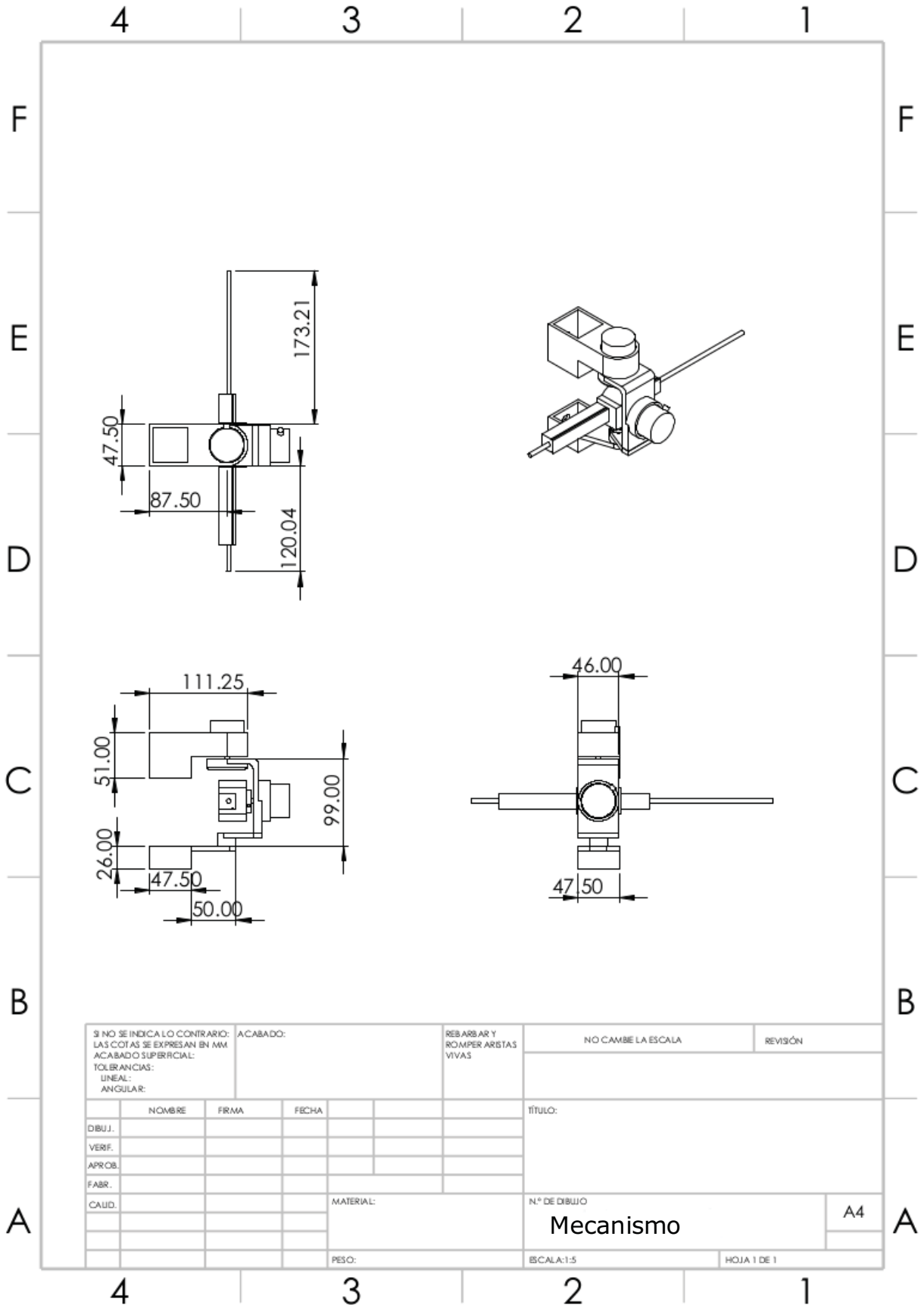
- of the 2015 Conference on Advances In Robotics, AIR '15, (New York, NY, USA), Association for Computing Machinery, 2015.
- [12] R. L. Evans, R. W. Partridge, and D. K. Arvind, "Demonstration paper: A comparative study of surgical skills assessment in a physical laparoscopy simulator using wireless inertial sensors," in *Proceedings of the Wireless Health 2014 on National Institutes of Health*, WH '14, (New York, NY, USA), p. 1–8, Association for Computing Machinery, 2014.
- [13] F. Pérez-Escamirosa, A. Alarcón-Paredes, G. A. Alonso-Silverio, I. Oropesa, O. Camacho-Nieto, D. Lorias-Espinoza, and A. Minor-Martínez, "Objective classification of psychomotor laparoscopic skills of surgeons based on three different approaches," *International Journal of Computer Assisted Radiology and Surgery*, vol. 15, pp. 27–40, jan 2020.
- [14] S. Siyar, H. Azarnoush, S. Rashidi, A. Winkler-Schwartz, V. Bissonnette, N. Ponnudurai, and R. F. Del Maestro, "Machine learning distinguishes neurosurgical skill levels in a virtual reality tumor resection task," *Medical & Biological Engineering & Computing*, pp. 1–11, apr 2020.
- [15] J. L. Lavanchy, J. Zindel, K. Kirtac, I. Twick, E. Hosgor, D. Candinas, and G. Beldi, "Automation of surgical skill assessment using a three-stage machine learning algorithm," *Scientific Reports*, vol. 11, no. 1, 2021.
- [16] C. E. Reiley and G. D. Hager, "Task versus subtask surgical skill evaluation of robotic minimally invasive surgery," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009* (G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, and C. Taylor, eds.), (Berlin, Heidelberg), pp. 435–442, Springer Berlin Heidelberg, 2009.
- [17] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. Muller, "Evaluating surgical skills from kinematic data using convolutional neural networks," *CoRR*, vol. abs/1806.02750, 2018.
- [18] S. S. Vedula, A. Malpani, N. Ahmidi, S. Khudanpur, G. Hager, and C. Chen, "Task-Level vs. Segment-Level Quantitative Metrics for Surgical Skill Assessment," *Journal of surgical education*, vol. 73, pp. 482–489, may 2016.
- [19] S. N. Steigerwald, J. Park, K. M. Hardy, L. M. Gillman, and A. S. Vergis, "Does laparoscopic simulation predict intraoperative performance? A comparison between the Fundamentals of Laparoscopic Surgery and LapVR evaluation metrics," *American Journal of Surgery*, vol. 209, pp. 34–39, jan 2015.
- [20] V. A. Beltran and D. L. Hernández, *Impacto en el entrenamiento con simuladores de laparoscopia dentro del programa de Cirugía General-Universidad del Rosario*. PhD thesis, 2013.
- [21] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Béjar, D. D. Yuh, C. C. G. Chen, R. Vidal, S. Khudanpur, and

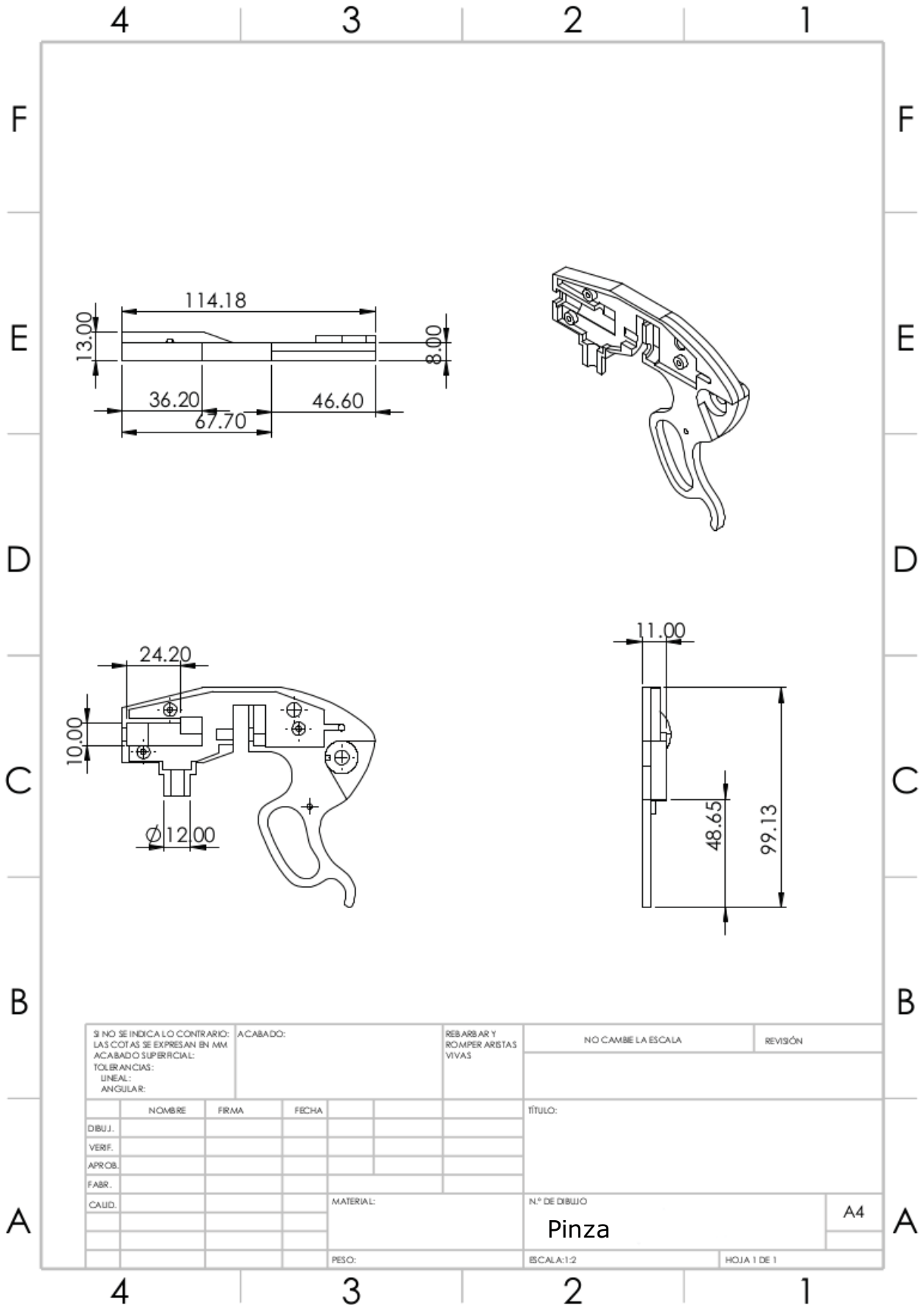
- G. Hager, “Jhu-isi gesture and skill assessment working set (jigsaws) : A surgical activity dataset for human motion modeling,” 2014.
- [22] N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B. B. Haro, L. Zappella, S. Khudanpur, R. Vidal, and G. D. Hager, “A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2025–2041, 2017.
- [23] N. S. A. (1992), “An Introduction to Kernel and Nearest-Neighbor Non-parametric Regression, *The American Statistician*, 46:3, 175-185, DOI: 10.1080/00031305.1992.10475879,”
- [24] L. Breiman, “Random Forests. *Machine Learning* 45, 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>,”
- [25] V. V. Cortes, C., “Support-vector networks. *Mach Learn* 20, 273–297 (1995). <https://doi.org/10.1007/BF00994018>,”

Apéndice I: Dibujos técnicos del diseño mecánico









Apéndice II: Programas del sistema eScope

.1. Lectura de sensores en Arduino

```
//Este programa se encarga de la lectura de los sensores  
//del sistema eScope y de enviar los datos de los  
//sensores a una PC conectada al puerto serie
```

```
#define ENCODER_OPTIMIZE_INTERRUPTS  
#define ENCODER_USE_INTERRUPTS  
#include <Encoder.h>
```

```
Encoder Encoder1(45, 43);  
Encoder Encoder2(47, 49);  
int pinPot=A0;  
int pinLineal=A6;  
// avoid using pins with LEDs attached
```

```
void setup() {  
  Serial.begin(115200);  
  //Serial.println("E1 E2 E3 E4 gripper penetracion");  
}
```

```
long oldPosition = -999;
```

```
int E1;  
int E2;  
int E3;  
int E4;  
int gripper;  
int penetracion;
```

```
void loop(){
```

```
E1=Encoder1.read();
E2=Encoder2.read();
gripper=analogRead(pinPot);
penetracion=analogRead(A5);
Serial.print(E1);
Serial.print(",");
Serial.print(E2);
Serial.print(",");
Serial.print(penetracion);
Serial.print(",");
Serial.println(gripper);
delay(100);
}
```

.2. Menú principal

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class mainMenuScript : MonoBehaviour
{
    // Start is called before the first frame update
    public void PlayCalibration()
    {
        SceneManager.LoadScene("Calibration", LoadSceneMode.Single);
    }
    public void PlayTest()
    {
        SceneManager.LoadScene("Test", LoadSceneMode.Single);
    }

    public void PlaySpacialOrientation()
    {
        SceneManager.LoadScene("SpatialOrientation", LoadSceneMode.Single);
    }
    public void PlayTransferObjects()
    {
        SceneManager.LoadScene("transferObjects", LoadSceneMode.Single);
    }
    public void back2Menu()
    {
        SceneManager.LoadScene("Menu", LoadSceneMode.Single);
    }
    public void playSettings()
    {
        SceneManager.LoadScene("settings", LoadSceneMode.Single);
    }
}
```

.3. Registro del usuario

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class registroScript : MonoBehaviour
{
    // Start is called before the first frame update
    public InputField Name;
    public InputField Expertise;
    public void getData(){
        CSVManager.setName(Name.text);
        CSVManager.setExpertise(Expertise.text);
        SceneManager.LoadScene("Menu", LoadSceneMode.Single);
    }
}
```

.4. Calibración del sistema

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
using System.IO.Ports;

public class Calibration : MonoBehaviour
{
    public SerialPort serialPort;
    private string [] datos = new string [4];
    private char [] splitters = new char []{ ' ', ' ' };
    private float [] entrada = new float [4];
    public string SerialLine;
    static public Vector3 position1 , position2 , position3 ;
    static public float pos1Gripper , pos2Gripper ;
    public RawImage Image;
    public Texture [] myTextures = new Texture [5];
    private int currentItem=0;
    // Start is called before the first frame update
    void Start ()
    {
        string serialPortString;
        Image.texture = myTextures [currentItem];
        serialPortString=PlayerPrefs.GetString ("serialPort");
        serialPort=new SerialPort (serialPortString ,115200);
        serialPort.Open ();
        serialPort.ReadTimeout=100;
    }

    void Update ()
    {
        try
        {
            if (serialPort.IsOpen)
            {
                SerialLine=serialPort.ReadLine ();
            }
        }
        catch (System.Exception ex)
        {
            ex = new System.Exception ();
        }
    }
}

```

```
    }  
  }  
  public void NextButton()  
  {  
    datos=SerialLine.Split(splitters);  
    for (int i = 0; i < datos.Length; i++)  
    {  
      float.TryParse(datos[i], out entrada[i]);  
    }  
    //Debug.Log(entrada[0].ToString()+", "+entrada[1].ToString()+", "+entrada[2].ToString());  
    if(currentItem == 0){  
      position1= new Vector3(entrada[0], entrada[1], entrada[2]);  
    }  
    if(currentItem == 1){  
      position2= new Vector3(entrada[0], entrada[1], entrada[2]);  
    }  
    if(currentItem == 2){  
      position3= new Vector3(entrada[0], entrada[1], entrada[2]);  
    }  
    if(currentItem == 3){  
      pos1Gripper = entrada[3];  
    }  
    if(currentItem == 4){  
      pos2Gripper = entrada[3];  
      SceneManager.LoadScene("Menu", LoadSceneMode.Single);  
      return;  
    }  
    currentItem++;  
    Image.texture = myTextures[currentItem];  
  }  
}
```

.5. Configuración del sistema

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.IO.Ports;
using UnityEngine.SceneManagement;

public class ConnectionScript : MonoBehaviour
{
    // Start is called before the first frame update
    private static SerialPort serialPort;
    public InputField fieldPort;
    public Text messageField;

    public void getConnection()
    {
        string line;
        string [] datos = new string [4];
        char [] splitters = new char []{ ',', ' ' };
        float [] entrada = new float [4];
        Debug.Log(fieldPort.text);
        try
        {
            serialPort =new SerialPort(fieldPort.text,115200);
            serialPort.Open();
            serialPort.ReadTimeout=100;
            if(serialPort.IsOpen)
            {
                float seconds;
                line=serialPort.ReadLine();
                seconds=Time.deltaTime;
                datos=line.Split(splitters);
                for (int i = 0; i < datos.Length; i++)
                {
                    float.TryParse(datos[i], out entrada[i]);
                }
                messageField.text="conexion_exitosa";
                PlayerPrefs.SetString("serialPort", fieldPort.text);
                SceneManager.LoadScene("Menu", LoadSceneMode.Single);
            }
        }
        else
    }
}

```



```
        {
            messageField.setText("error");
        }
    }
    catch
    {
        messageField.setText("No se pudo establecer conexion con el puerto "+f
    }
}
```

.6. Movimientos básicos

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO.Ports;

public class GimballScript : MonoBehaviour
{
    // Start is called before the first frame update
    public SerialPort serialPort;
    private string line;
    private string [] datos = new string [4];
    private char [] splitters = new char []{',',' '};
    private float [] entrada = new float [4];
    public bool enableGripper;
    public static bool isTransferObject;
    private Vector3 OutCalibration;
    public static float calibrationGripperOut;
    private GameObject Gimball, soportePinza, Pinza, Pieza1, Pieza1_1;
    static public bool enableSaveData=false;
    void Start()
    {
        string serialPortString;
        serialPortString=PlayerPrefs.GetString("serialPort");
        serialPort=new SerialPort(serialPortString,115200);
        serialPort.Open();
        serialPort.ReadTimeout=100;
    }
    // Update is called once per frame
    void Update()
    {
        try
        {
            if(serialPort.IsOpen)
            {
                float seconds;
                line=serialPort.ReadLine();
                //Debug.Log(line);
                seconds=Time.deltaTime;
                datos=line.Split(splitters);
                for (int i = 0; i < datos.Length; i++)
                {

```

```

        float.TryParse(datos[i], out entrada[i]);
    }
    Gimball = GameObject.Find("Gimball");
    soportePinza=GameObject.Find("soportePinza");
    Pinza=GameObject.Find("pinza");
    Pieza1=GameObject.Find("Pieza1");
    Pieza1_1=GameObject.Find("Pieza1_1");
    OutCalibration = CalibrationFunctions.map2calibrate(
        new Vector3(entrada[0], entrada[1], entrada[2]));
    calibrationGripperOut=CalibrationFunctions.gripperCalibrated(
        entrada[3]);
    Gimball.transform.localEulerAngles = new
        Vector3(45-OutCalibration[0]*0.225f, 0, 0);
    soportePinza.transform.localEulerAngles =
        new Vector3(0,-OutCalibration[1]*0.225f, 0);
    Pinza.transform.localPosition =
        new Vector3(0, -0.009f, -OutCalibration[2]
        *0.12f/1023.0f-0.1f);

    if(enableGripper){
        if(isTransferObject)
        {
            if(transferObjectsScript.blockGripperValue<
            calibrationGripperOut){
                Pieza1.transform.localEulerAngles=
                    new Vector3(0,-calibrationGripperOut,90);
                Pieza1_1.transform.localEulerAngles=
                    new Vector3(0,calibrationGripperOut,-90);
            }
        }
        else
        {
            Pieza1.transform.localEulerAngles=
                new Vector3(0,-calibrationGripperOut,90);
            Pieza1_1.transform.localEulerAngles=
                new Vector3(0,calibrationGripperOut,-90);
        }
    }
    else
    {
        Pieza1.transform.localEulerAngles=new Vector3(0,0,90);
        Pieza1_1.transform.localEulerAngles=new Vector3(0,0,-90);
    }
    if(enableSaveData){
        CSVManager.AppendToReport(OutCalibration,seconds);
    }

```

```
        }
    }
}
catch (System.Exception ex)
{
    Debug.LogException(ex, this);
}
}
```

.7. Tarea de orientación espacial

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class SpatialOrientationScript : MonoBehaviour
{
    public GameObject Fig1;
    static public int i=0;
    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.tag=="Fig1")
        {
            switch (i)
            {
                case 0:
                    Fig1.transform.position=new
                        Vector3(0.003f,0.095f,-0.113f);
                    GimballScript.enableSaveData=true;
                    ClockScript.enable=true;
                    break;
                case 1:
                    Fig1.transform.position=new
                        Vector3(-0.019f,0.084f,-0.12f);
                    break;
                case 2:
                    Fig1.transform.position=new
                        Vector3(0.029f,0.089f,-0.057f);
                    break;
                case 3:
                    Fig1.transform.position=new
                        Vector3(-0.031f,0.074f,-0.084f);
                    break;
                case 4:
                    Fig1.transform.position=new
                        Vector3(-0.038f,0.085f,-0.11f);
                    break;
                case 5:
                    Fig1.transform.position=new
                        Vector3(-0.045f,0.092f,-0.044f);
                    break;
                case 6:
```

```
        Fig1.transform.position=new
            Vector3(0.014f,0.078f,-0.065f);
        break;
    case 7:
        Fig1.transform.position=new
            Vector3(-0.018f,0.081f,-0.096f);
        break;
    case 8:
        Fig1.transform.position=new
            Vector3(0.057f,0.095f,-0.047f);
        break;
    case 9:
        Fig1.transform.position=new
            Vector3(0.01f,0.109f,-0.054f);
        break;
    case 10:
        i=0;
        SceneManager.LoadScene("Menu", LoadSceneMode.Single);
        break;
    }
    i++;
}
}
```

.8. Tarea de traslado de objetos

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine.SceneManagement;
using UnityEngine;

public class transferObjectsScript : MonoBehaviour
{
    // Start is called before the first frame update
    public Rigidbody rb;
    public GameObject padre;
    public GameObject palo;
    private int i;
    public static float blockGripperValue;
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        Debug.Log("T1= "+collisionPinzaScript.T1.ToString()+"",
            T2= "+collisionPinzaScript.T2.ToString());
        if(collisionPinzaScript.T1 && collisionPinzaScript.T2){
            if(rb.isKinematic==true){return;}
            rb.isKinematic=true;
            GimballScript.enableSaveData=true;
            ClockScript.enable=true;
            transform.SetParent(padre.transform);
            blockGripperValue=GimballScript.calibrationGripperOut;
            GimballScript.isTransferObject=true;
        }
        else{
            rb.isKinematic=false;
            transform.SetParent(null);
            GimballScript.isTransferObject=false;
            Vector2 pos1=
                new Vector2(palo.transform.position.x,
                    palo.transform.position.z);
            Vector2 pos2=
                new Vector2(rb.transform.position.x,
                    rb.transform.position.z);
        }
    }
}

```

```
float distancia = Vector2.Distance(pos1, pos2);
if(rb.transform.position.y < 0.07 && distancia < 0.03f)
{
    i=i+1;
    switch(i){
        case 1:
            palo.transform.position=
                new Vector3(-0.03f, 0.066f, -0.1f);
            break;
        case 2:
            palo.transform.position=
                new Vector3(0f, 0.066f, -0.08f);
            break;
        case 3:
            SceneManager.LoadScene("Menu",
                LoadSceneMode.Single);
            break;
    }
}
}
```