



FACULTAD DE INGENIERÍA ELECTRÓNICA Y  
TELECOMUNICACIONES

Trabajo de grado para optar por el título de  
Ingeniero en Automática Industrial

SISTEMA DE GENERACIÓN DE  
TRAYECTORIAS PARA SUTURA  
DE TIPO CONTINUO

**Andres Alirio Morillo Montenegro**

**Juan Jose Romero Maygual**

Popayán, Septiembre, 2023

Nota de Aceptación

---

---

---

---

---

---

Director: \_\_\_\_\_

Hermes Fabian Vargas Rosero

\_\_\_\_\_

Firma del jurado

\_\_\_\_\_

Firma del jurado

Popayán, Septiembre de 2023

# Agradecimientos

## **Andres Alirio Morillo Montenegro**

Quiero agradecer primero a Dios por haberme permitido llegar hasta este punto de mi vida, agradecerle por todas las bendiciones que ha derramado sobre mí y también darle gracias a la santísima virgen del Carmen por guiarme y protegerme en cada paso de mi vida.

Infinitas gracias a mis padres que son el pilar fundamental en mi vida, gracias por apoyarme, creer en mí, permitirme soñar e impulsarme a cumplir cada una de mis metas, a mis hermanas que son el motor de mi vida, gracias por su apoyo, sus enseñanzas y toda la ayuda que me han brindado.

Quiero también honrar la memoria de mis abuelitos Clemencia Rosero y Vicente Morillo, quienes me dejaron inculcando los valores necesarios para ser una gran persona, gracias por sus enseñanzas, sus consejos, su amor y su cariño.

Agradecer al ingeniero Hermes Fabián Vargas por creer en mis capacidades, por transmitir todo su conocimiento y hacer posible este trabajo de grado.

Agradecer a mi compañero Juan José Romero por haber realizado este trabajo de investigación a mi lado, por su paciencia y dedicación.

Extiendo mis sinceros agradecimientos a la Universidad del Cauca y a los maestros que me acompañaron a lo largo de mi formación como ingeniero, son parte fundamental de mi crecimiento como ingeniero, gracias por las enseñanzas y todo el conocimiento transmitido.

**Juan José Romero Maygual**

Dedico este trabajo a todos aquellos que han sido parte integral de mi vida académica y personal. A mi familia, cuyo apoyo incondicional han sido mi fortaleza. A mis amigos y seres queridos, quienes han compartido risas, desafíos y triunfos a lo largo de este camino. Agradezco sinceramente a los mentores y colegas que han dejado una huella en mi crecimiento, brindándome su orientación y perspectivas valiosas.

Este logro no habría sido posible sin el esfuerzo y la dedicación de mis profesores, quienes no solo compartieron su conocimiento, sino que también me inspiraron a alcanzar nuevas metas. A todos aquellos que han contribuido a mi formación, les expreso mi gratitud. Este trabajo es un reflejo de la influencia positiva que cada uno de ustedes ha tenido en mi desarrollo personal y académico.

Que este trabajo sea un pequeño tributo a todos aquellos que han contribuido a mi formación y crecimiento, dejando una huella imborrable en mi vida.

# Resumen

El proyecto descrito en este documento ilustra un sistema generador de trayectorias para sutura de punto simple continuo, empleando dos robots UR3 de Universal Robot. El sistema tiene como objetivo, crear las trayectorias necesarias para llevar a cabo las puntadas de una sutura continua, las cuales serán enviadas a los robots y posteriormente ejecutadas en un entorno simulado.

Se desarrolló un entorno de simulación en Unity, el cual está conectado mediante el sistema operativo ROS a MATLAB. Desde MATLAB se generan las trayectorias, las cuales son transmitidas al entorno simulado a través de ROS. Esto permite obtener los movimientos necesarios para llevar a cabo una sutura. Como resultado, se ha obtenido un simulador donde se pueden visualizar los puntos de entrada y salida de la aguja quirúrgica, guiada por una pinza EndoWrist conectada a un robot UR3.

# Abstract

The project described in this document illustrates a trajectory generating system for continuous single stitch suturing, using two UR3 robots from Universal Robot. The system aims to create the necessary trajectories to carry out the stitches of a continuous suture, which will be sent to the robots and subsequently executed in a simulated environment.

A simulation environment was developed in Unity, which is connected through the ROS operating system to MATLAB. The trajectories are generated from MATLAB, which are transmitted to the simulated environment through ROS. This allows the movements necessary to carry out a suture to be obtained. As a result, a simulator has been obtained where the entry and exit points of the surgical needle can be visualized, guided by an EndoWrist clamp connected to a UR3 robot.

# Índice general

<b>1. Estado del arte</b>	<b>1</b>
<b>2. Introducción y generalidades</b>	<b>21</b>
2.1. Introducción . . . . .	21
2.2. Organización del documento . . . . .	24
<b>3. Marco teórico</b>	<b>25</b>
3.1. Introducción . . . . .	25
3.2. Heridas . . . . .	25
3.3. Tipos de heridas . . . . .	26
3.3.1. A. Heridas abiertas . . . . .	27
3.3.2. B. Heridas contusas . . . . .	27
3.3.3. C. Heridas incisas . . . . .	27
3.3.4. D. Penetrantes . . . . .	28
3.4. Instrumental usado para sutura . . . . .	28
3.4.1. Agujas quirúrgicas . . . . .	28
3.4.2. Características de las agujas quirúrgicas . . . . .	29
3.4.3. El hilo de sutura . . . . .	30
3.4.4. Pinzas quirúrgicas de laparoscopia . . . . .	31
3.5. Técnicas de sutura . . . . .	34
3.5.1. A. Sutura con puntos sueltos o discontinuos . . . . .	35
3.5.2. B. Sutura continua . . . . .	36
3.5.3. C. Sutura continua intradérmica . . . . .	37
3.5.4. D. Sutura de esquina . . . . .	38
3.5.5. E. Sutura de punto simple . . . . .	39
3.5.6. F. Sutura punto de colchonero . . . . .	39
3.5.7. G. Sutura punto en U o colchonero horizontal . . . . .	40
3.5.8. H. Sutura punto de Algöwer o punto Suizo . . . . .	40

3.6.	Planificador de trayectorias . . . . .	41
3.6.1.	Generación y seguimiento de trayectorias con Polinomios, B-splines, y Perfiles de Velocidad Trapezoidal . . . . .	42
3.6.2.	Diagrama de Voronoi . . . . .	43
3.6.3.	Roadmap Probabilístico (PRM) . . . . .	44
3.6.4.	Modelado del Espacio Libre . . . . .	45
3.6.5.	Campos Potenciales . . . . .	46
3.6.6.	Rapidly-exploring Random Trees (RRT) . . . . .	47
<b>4.</b>	<b>Desarrollo</b>	<b>49</b>
4.1.	Introducción . . . . .	49
4.2.	Herramientas de desarrollo . . . . .	49
4.2.1.	MATLAB . . . . .	50
4.2.2.	ROS (Robot Operating System) . . . . .	51
4.2.3.	UNITY . . . . .	52
4.2.4.	Robot UR3 . . . . .	53
4.2.5.	Pinza EndoWrist . . . . .	54
4.3.	Arquitectura . . . . .	55
4.3.1.	Módulo de desarrollo - MATLAB . . . . .	55
4.3.2.	Módulo de desarrollo - ROS . . . . .	57
4.3.3.	Módulo de desarrollo - Unity . . . . .	57
4.4.	Implementación en MATLAB . . . . .	58
4.4.1.	Conectividad MATLAB - ROS. . . . .	58
4.4.2.	Carga y configuración del modelo del brazo robot UR3 . . . . .	59
4.4.3.	Configuración del objeto para cinemática inversa . . . . .	60
4.4.4.	Definición de posiciones y transformaciones . . . . .	61
4.4.5.	Función de trayectoria de inserción y extracción de aguja . . . . .	62
4.4.6.	Planificador con algoritmo RRT . . . . .	64
4.4.7.	Funciones implementadas . . . . .	67
<b>5.</b>	<b>Resultados</b>	<b>72</b>
5.1.	Trayectoria . . . . .	72
5.1.1.	Trayectorias brazo robot UR3_1 . . . . .	75
5.1.2.	Trayectorias brazo robot UR3_2 . . . . .	77
5.2.	Análisis de trayectoria . . . . .	79
5.2.1.	Error de posición y orientación . . . . .	83

---

5.3. Entorno de simulación . . . . .	85
5.3.1. Descripción del entorno de simulación . . . . .	85
5.3.2. Simulación . . . . .	88
<b>6. Conclusiones</b>	<b>99</b>
<b>A. Anexos</b>	<b>100</b>
A.1. Introducción . . . . .	100
A.2. Instalación Ubuntu 18.04 . . . . .	100
A.3. Instalación ROS MELODIC . . . . .	105
A.4. Instalación MATLAB . . . . .	106
A.5. Instalación Unity . . . . .	109
A.6. Guía de ejecución del simulador . . . . .	110
A.6.1. Inicio del simulador . . . . .	110
A.6.2. Inicio de MATLAB . . . . .	112
A.6.3. Inicializar la comunicación por medio de ROS . . . . .	113
A.6.4. Consultar IP . . . . .	115
A.6.5. Ejecución del simulador . . . . .	116
A.6.6. Inicio de Simulación . . . . .	116
<b>Bibliografía</b>	<b>117</b>

# Índice de figuras

1.1. Sistema robótico para evaluación remota. . . . .	2
1.2. Brazo del robot con marcos asignados unidos a cada articulación. . . . .	2
1.3. Sistema de adquisición de una operación de sutura. . . . .	3
1.4. Automatización de la cirugía de tejidos blandos. . . . .	4
1.5. (a) Hardware STAR LWR de 7 DOF montado con un Endo 360 modificado. (b) Aguja circular del Endo 360 accionada por el STAR. . . . .	5
1.6. (a) Sistema de sutura laparoscópico robótico, (b) Endoscopio de imágenes 3D. . . . .	6
1.7. a) Sistema de cámara dual, b) Imagen de la cámara NIR de los marcadores en una muestra de lengua porcina, c) Nube de puntos de la muestra de prueba obtenida de la cámara RGBD, y d) Posiciones superpuestas de los marcadores en la nube de puntos 3D para el robot de planificación de rutas. . . . .	8
2.1. Pinza laparoscópica [26]. . . . .	22
2.2. Aguja de sutura sujeta de pinza laparoscópica [27]. . . . .	23
2.3. Trayectoria de aguja e hilo para sutura de herida . . . . .	23
3.1. Herida [31]. . . . .	26
3.2. Tipos de heridas [31]. . . . .	27
3.3. Aguja quirúrgicas [35]. . . . .	29
3.4. Partes de una aguja quirúrgica 1/2 [36]. . . . .	30
3.5. Sutura de herida [36]. . . . .	31
3.6. Pinzas laparoscópicas [34]. . . . .	32
3.7. Tipos de sutura [37]. . . . .	35
4.1. MATLAB [Fuente propia]. . . . .	50
4.2. ROS(Robot Operating System) [Fuente propia]. . . . .	51
4.3. Logo Unity. [43] . . . . .	52
4.4. Robot UR3. [44] . . . . .	53

---

4.5. Pinza EndoWrist. [46]	54
4.6. Arquitectura [Fuente propia].	55
4.7. Conectividad Matlab - Ros [Fuente propia].	58
4.8. Modelo brazo robot UR3 [Fuente propia].	59
4.9. Función 'loadrobot' [Fuente propia].	59
4.10. Articulaciones UR3e	60
4.11. Tabla de parámetros denavit hartenberg del robot UR3. [48]	60
4.12. Trayectoria de inserción y extracción de aguja[Fuente propia].	62
4.13. Matriz de Posiciones y Orientaciones.	63
4.14. Desplazamiento de pinza con aguja quirúrgica [Fuente propia].	63
4.15. Posición de trayectorias en sus ejes [Fuente propia].	64
4.16. Inicialización de robot UR3 y objeto de colisión [Fuente propia].	65
4.17. Trayectoria con evasión de obstáculo [Fuente propia].	65
4.18. Trayectoria generada para evadir obstáculo [Fuente propia].	66
4.19. Trayectoria generada para evadir obstáculo - Angulo 2 [Fuente propia].	66
4.20. Trayectoria generada sin objeto de colisión [Fuente propia].	67
5.1. Diagrama de Estados [Fuente propia].	73
5.2. Áreas de operación UR3_1, UR3_2 [Fuente propia].	74
5.3. Trayectorias robot UR3 y pinza EndoWrist 1 [Fuente propia].	75
5.4. Trayectorias UR3_2 y pinza EndoWrist 2 [Fuente propia].	77
5.5. Parámetros de sutura [Fuente propia].	79
5.6. Curva de inserción [Fuente propia].	80
5.7. Trayectoria de inserción de aguja [Fuente propia].	80
5.8. Curva de extracción [Fuente propia].	81
5.9. Trayectoria de extracción de aguja [Fuente propia].	81
5.10. Script puntos de sutura [Fuente propia].	82
5.11. Puntos de sutura [Fuente propia].	82
5.12. Movimientos brazo robot UR3 [Fuente propia].	83
5.13. Error de Posición [Fuente propia].	84
5.14. Error de Orientación [Fuente propia].	84
5.15. Entorno quirúrgico,Vista lateral [Fuente propia].	86
5.16. Entorno quirúrgico, Vista Simulador [Fuente propia].	86
5.17. Pestaña 'PANEL DE CONTROL' [Fuente propia].	87
5.18. Panel de control [Fuente propia].	87
5.19. Opción Conexión [Fuente propia].	88

5.20. Opción Herida [Fuente propia]. . . . .	88
5.21. Posición inicial UR3_1, UR3_2 [Fuente propia]. . . . .	89
5.22. Posición inicial- Desplazamiento UR3_1 [Fuente propia]. . . . .	89
5.23. Posición final - Desplazamiento UR3_1 [Fuente propia]. . . . .	90
5.24. Posición inicial- Inserción de Aguja [Fuente propia]. . . . .	90
5.25. Posición final - Inserción de Aguja [Fuente propia]. . . . .	91
5.26. Posición - Desplazamiento UR3_2 [Fuente propia]. . . . .	91
5.27. Posición final - Desplazamiento UR3_2 [Fuente propia]. . . . .	92
5.28. Posición inicial - Desplazamiento UR3_1 [Fuente propia]. . . . .	92
5.29. Posición final - Desplazamiento UR3_1 [Fuente propia]. . . . .	92
5.30. Posición inicial- Extracción de Aguja [Fuente propia]. . . . .	93
5.31. Posición final - Extracción de Aguja [Fuente propia]. . . . .	93
5.32. Posición inicial - Desplazamiento UR3_1 [Fuente propia]. . . . .	94
5.33. Posición final - Desplazamiento UR3_1 [Fuente propia]. . . . .	94
5.34. Posición inicial - Desplazamiento UR3_2 [Fuente propia]. . . . .	95
5.35. Posición final - Desplazamiento UR3_2 [Fuente propia]. . . . .	95
5.36. Posición inicial - Desplazamiento UR3_1 [Fuente propia]. . . . .	96
5.37. Posición final - Desplazamiento UR3_1 [Fuente propia]. . . . .	96
5.38. Posición inicial - Desplazamiento UR3_1 [Fuente propia]. . . . .	97
5.39. Posición final - Desplazamiento UR3_1 [Fuente propia]. . . . .	97
5.40. Posición inicial - Desplazamiento UR3_2 [Fuente propia]. . . . .	97
5.41. Posición final - Desplazamiento UR3_2 [Fuente propia]. . . . .	98
A.1. Descarga Ubuntu . . . . .	101
A.2. Descarga Rufus . . . . .	101
A.3. Imagen ISO de Ubuntu . . . . .	102
A.4. Opción administrador de discos . . . . .	103
A.5. Opción de partición . . . . .	104
A.6. Escritorio de Ubuntu . . . . .	104
A.7. Comandos de instalación ROS . . . . .	105
A.8. Ventana principal MathWorks . . . . .	106
A.9. Usuario y contraseña de MathWorks . . . . .	106
A.10. MATLAB - Linux . . . . .	107
A.11. Instalación de MATLAB desde terminal . . . . .	107
A.12. Pasos de instalación . . . . .	108
A.13. Pagina principal Unity . . . . .	109

---

A.14.Instrucciones de descarga Unity . . . . .	109
A.15.Comandos de instalación . . . . .	110
A.16.Ejecutable simulador . . . . .	111
A.17.Simulador . . . . .	111
A.18.Comando ‘matlab’ . . . . .	112
A.19.MATLAB, script ‘caso_tg-10’ . . . . .	112
A.20.Comando de configuración ROS . . . . .	113
A.21.Ejecución del comando de configuración ROS . . . . .	114
A.22.Comando de configuración del entorno ROS . . . . .	114
A.23.Ejecución comando de configuración del entorno ROS . . . . .	115
A.24.Comando ‘ifconfig’ . . . . .	115

# Índice de tablas

1.1. Artículos . . . . .	20
4.1. Funciones implementadas - Parte 1 . . . . .	68
4.2. Funciones implementadas - Parte 2 . . . . .	69
4.3. Funciones implementadas - Parte 3 . . . . .	70
4.4. Funciones implementadas - Parte 4 . . . . .	71

# 1. Estado del arte

La cirugía robótica ha transformado el ámbito médico a nivel mundial, ofreciendo nuevas perspectivas para mejorar los procedimientos quirúrgicos. En este contexto, la búsqueda exhaustiva de conocimiento se convierte en un paso crucial. Por ello, se llevó a cabo una investigación exhaustiva en la base de datos Web of Science, abarcando el período comprendido entre 2018 y 2024, con el propósito de identificar estudios relevantes que aborden temas como la planificación de rutas, la sutura automática y la sutura autónoma en cirugía robótica.

La sutura es un procedimiento médico que requiere ser abordado con rapidez y eficacia, ya que esto minimiza la pérdida de sangre del paciente y aumenta las posibilidades de supervivencia ante un accidente. En Estados Unidos, el 29% de las muertes por traumatismos prehospitalarios se deben a hemorragias no controladas. En muchas ocasiones, estas heridas se localizan en áreas de difícil acceso, donde la ejecución del procedimiento y su eficacia pueden influir en la cicatrización y la recuperación del paciente. A continuación, se presentarán algunos de los trabajos más relevantes sobre esta temática en particular [16].

Dentro de la investigación realizada por B. Mathur y otros [17], se propone un sistema robótico semi-autónomo para el análisis de lesiones mediante imágenes 2D y 3D obtenidas a través de una ecografía especializada en traumatismos conocida como FAST (Focused Assessment with Sonography for Trauma). El sistema está compuesto por un módulo de localización y posicionamiento autónomo para el escaneo, como se muestra en la figura 1.1. Este sistema es entrenado utilizando imágenes de tomografía para detectar la posición del ombligo, a partir de la cual se estima el ancho, la longitud y la altura del torso mediante un modelo 3D reconstruido. Posteriormente, el sistema calcula las posiciones de escaneo en el torso con base en esta información.

El radiólogo utiliza una interfaz gráfica de usuario GUI diseñada específicamente para este propósito, permitiéndole supervisar remotamente el sistema y guiar de manera autónoma al robot hacia las ubicaciones FAST estimadas previamente. La GUI proporciona información sobre la cantidad de fuerza ejercida sobre el paciente, lo que permite

realizar un escaneo rápido y efectivo de todas las lesiones posibles en la sección del torso. Esto facilita el tratamiento de las lesiones y aumenta la probabilidad de supervivencia del paciente.

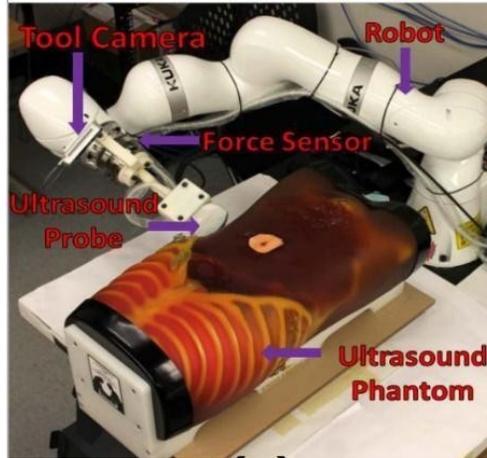


Figura 1.1.: Sistema robótico para evaluación remota.

Hossein Dehghani y otros [2], proponen una investigación que se basa en la optimización de las trayectorias de sutura. Para esto se hace uso de un algoritmo con el cual se controla ligeramente la ruta de la aguja buscando lograr una trayectoria de sutura ideal en un robot de cuatro grados de libertad, esto con el fin de compensar suturas en espacios reducidos y con poca visibilidad. El algoritmo fue evaluado únicamente para puntos de entrada y salida. La figura 1.2 muestra el dispositivo en cuestión.

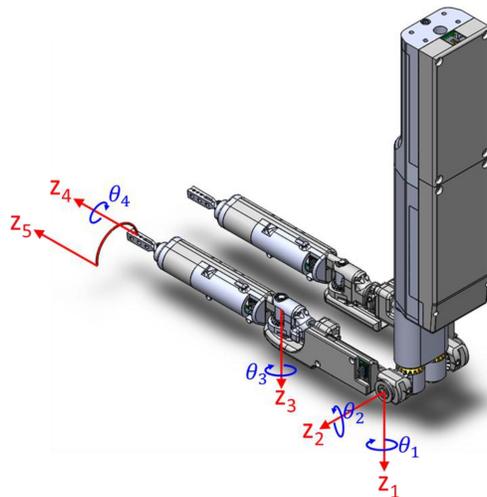


Figura 1.2.: Brazo del robot con marcos asignados unidos a cada articulación.

Dewei Yang y otros [18], seleccionaron la sutura de tejido superficial como objeto de modelado, con el propósito de mejorar el aprendizaje de este proceso y ampliar su capacidad de adaptabilidad frente a diferentes escenarios (Ver figura 1.3). Esta iniciativa busca mejorar tanto el procedimiento para la cirugía automática asistida por robots como la evaluación de las habilidades de sutura por parte de los médicos. Se empleó el método de primitivas de movimiento dinámico para lograr este objetivo, permitiendo obtener procesos dinámicos a medida que se realizan cambios dentro de la escena de sutura. Esto proporciona adaptabilidad a los múltiples escenarios que pueden presentarse al suturar tejidos blandos.

Inicialmente, se estableció un sistema de captura de movimiento dinámico utilizando 7 tiempos ópticos para medir y seguir el proceso de sutura, recopilando la posición 3D de los puntos de marcador en las pinzas quirúrgicas y la trayectoria de movimiento continuo en tiempo real. Se definieron modelos paramétricos para el estado inicial y final de la acción de sutura, mientras que los subprocesos dinámicos de la acción de sutura se modelaron utilizando el método DMP (Algoritmo de Aprendizaje Adaptable para una Solución Rápida en el Espacio Dimensional de los Robots).

El proceso de sutura dinámica se representó mediante escenas de simulación virtual, donde los modelos de habilidad de sutura propuestos pueden ser utilizados para la planificación de la operación, la verificación de la misma o la realización de suturas autónomas asistidas por robot.

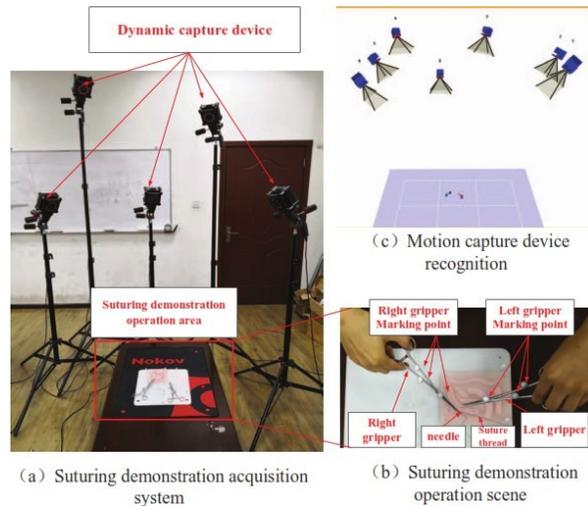


Figura 1.3.: Sistema de adquisición de una operación de sutura.

Azad Shademan y otros [19], demostraron que los procedimientos quirúrgicos autónomos ofrecen resultados superiores en comparación con las técnicas RAS convencionales.

Utilizaron el robot autónomo de tejido inteligente (STAR), al que se le implementó un sistema de reconstrucción tridimensional (3D) plenóptico y un sistema de seguimiento visual 3D (NIRF) (ver figura 1.4). Las imágenes plenópticas calculan un punto 3D para cada píxel en una imagen, triangulando la posición de las características observadas en múltiples imágenes de microlentes mediante un software. La combinación de la tecnología NIRF y las imágenes plenópticas cuantitativas en 3D aborda problemas como la oclusión y el reconocimiento de objetivos de tejido, utilizando marcadores NIRF luminiscentes. Estos marcadores guían el software de automatización de sutura para ajustar una línea según mediciones de longitud del corte y grosor del tejido, obtenidas mediante sensores o introducidas por el cirujano a través de una interfaz gráfica.

El software genera un plan de sutura optimizado geoméricamente, que incluye una secuencia de coordenadas para la inserción de la aguja, espaciado entre suturas, profundidades de mordida, fuerzas de tensión de tracción y maniobras de la herramienta de sutura antes de perforar. Estos parámetros pueden actualizarse en tiempo real para adaptarse al movimiento y la deformación del tejido. Esta metodología permitió realizar diversas intervenciones quirúrgicas en tejidos blandos con resultados superiores a los métodos manuales convencionales.

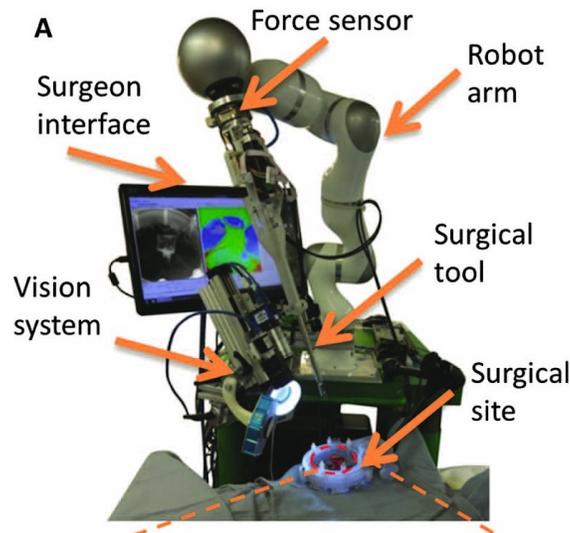


Figura 1.4.: Automatización de la cirugía de tejidos blandos.

Simón Leonard y otros [20], demostraron mediante el uso del robot de anastomosis de tejido inteligente (STAR) que su eficiencia y precisión superan a las de los cirujanos que emplean el sistema quirúrgico da Vinci o herramientas laparoscópicas manuales. Además, se comprobó que el sistema obtiene mejores resultados cuando realiza la sutura

---

de manera autónoma en comparación con la colocación manual de los puntos de sutura. Estos resultados se obtuvieron a partir de pruebas realizadas en almohadillas de sutura plana.

El robot STAR utiliza un robot ligero (LWR) de siete grados de libertad (DOF), al que se le acopla una herramienta especializada de sutura llamada Endo 360 (Ver figura 1.5). Este sistema permite al cirujano seleccionar manualmente los puntos de sutura de forma individual, mientras que en el modo automático se le permite delinear el contorno de la incisión. Una vez determinado el contorno de la incisión en la imagen, este se transforma en un marco 3D y se distribuyen uniformemente los puntos de sutura interpolando linealmente las coordenadas a lo largo de la ruta. Dado que este estudio se enfoca en incisiones planas, se realiza una transformación de las coordenadas de la imagen a coordenadas tridimensionales, asumiendo que la incisión es plana. El mapeo entre las coordenadas de la imagen de una puntada y sus coordenadas tridimensionales no restringe la incisión, ya que solo requiere rotaciones sobre los ejes X e Y del robot.”

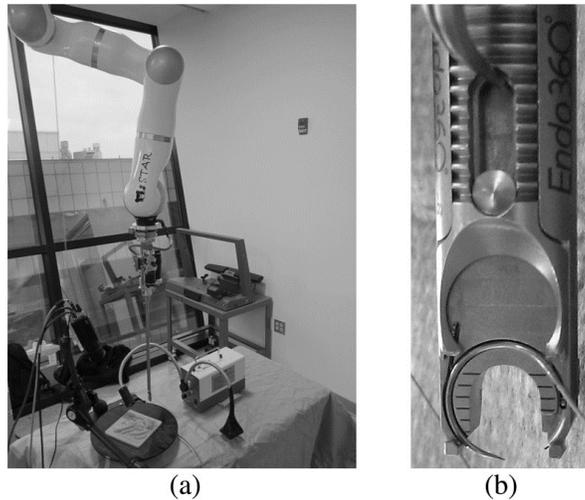


Figura 1.5.: (a) Hardware STAR LWR de 7 DOF montado con un Endo 360 modificado.  
(b) Aguja circular del Endo 360 accionada por el STAR.

H. Saeidi y otros [21], desarrollaron un sistema de sutura robótica laparoscópica autónoma, que incorpora un innovador endoscopio de imágenes 3D. Este endoscopio genera una representación en forma de nube de puntos, facilitando una planificación más precisa, como se ilustra en la figura 1.6. El sistema cuenta con una herramienta de sutura laparoscópica con dos grados de libertad en la punta, además de implementar una estrategia autónoma de planificación de suturas basada en la nube de puntos. Esta estrategia se compone de cinco pasos:

- Se delimita manualmente un polígono que define el contorno de la región de interés en la nube de puntos, permitiendo un análisis más focalizado.
- Se submuestra la nube de puntos a lo largo de una dirección que resalta la ranura de corte, determinando el espacio entre cada sección como el espacio de sutura planificado.
- Se identifica el surco de corte encontrando el mínimo prominente a lo largo de la sección de la nube de puntos.
- Se seleccionan dos regiones a ambos lados de la ranura de corte para localizar el borde de corte.
- Se calculan las coordenadas 3D de los puntos de sutura y se transforman en coordenadas globales para la manipulación robótica.

El objetivo de este sistema es eventualmente reemplazar la sutura manual, con el fin de lograr procedimientos más rápidos y precisos en el futuro.

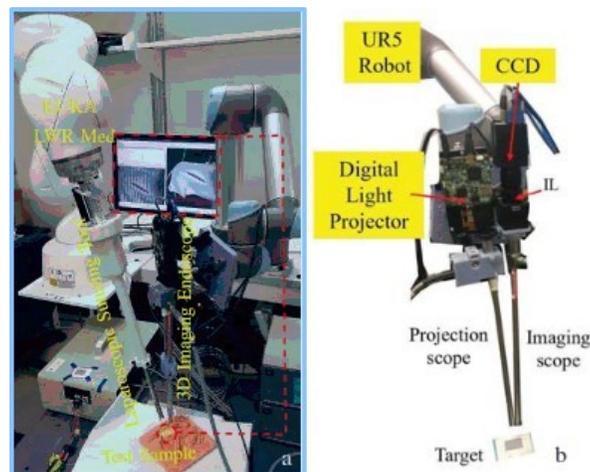


Figura 1.6.: (a) Sistema de sutura laparoscópico robótico, (b) Endoscopio de imágenes 3D.

H. Saeidi y otros [22] se propusieron mejorar los resultados de la sutura, aprovechando la repetitividad del proceso y reduciendo los errores humanos. Para ello, desarrollaron una estrategia de control, filtrado y planificación de rutas 3D autónomas supervisadas para su robot autónomo de tejido inteligente (STAR), permitiendo incisiones precisas y consistentes en tejidos blandos.

---

Esta estrategia emplea un método de marcado mediante infrarrojo cercano (NIR). Para el control, filtrado y planificación de rutas 3D autónoma supervisada para STAR, se implementó un sistema de cámara dual RGBD para obtener información de la superficie del tejido en 3D y una cámara NIR para detectar los marcadores NIR. Este sistema de imágenes extrae en tiempo real la posición 3D de los marcadores NIR mediante el trazado de rayos de sus posiciones (Ver figura 1.7).

Se aplica una técnica de compensación para proyectar las posiciones de los marcadores NIR hacia fuera en la nube de puntos, lo que permite la planificación de una ruta de incisión con márgenes específicos alrededor de los marcadores NIR. Además, se emplea un filtro para eliminar el ruido dinámico entre fotogramas, lo que facilita su uso en los controladores del robot.

Los puntos de referencia resultantes y las posiciones del robot en tiempo real se utilizan para obtener trayectorias suaves basadas en el tiempo a través de bibliotecas de movimiento. Las trayectorias del espacio de tareas del robot se convierten en trayectorias del espacio articular utilizando la Biblioteca de cinemática y dinámica (KDL) de Open Robot Control Software (OROCOS), que resuelve el problema cinemático inverso.

Finalmente, se implementan controladores de robot KUKA para que el robot siga las trayectorias deseadas en el espacio articular. Un operador o cirujano selecciona los puntos clave (marcadores NIR) que describen el tumor a través de una GUI y valida la ruta de electrocauterio para el robot antes del inicio de la autonomía.

Este proceso se verificó mediante experimentos en muestras de lengua porcina, y se comparó con una estrategia de teleoperación a través de da Vinci realizada por un cirujano experto en cabeza y cuello. La estrategia propuesta redujo el error de incisión superficial en un 40.03 % y el error de incisión profunda en un 51.5 %.

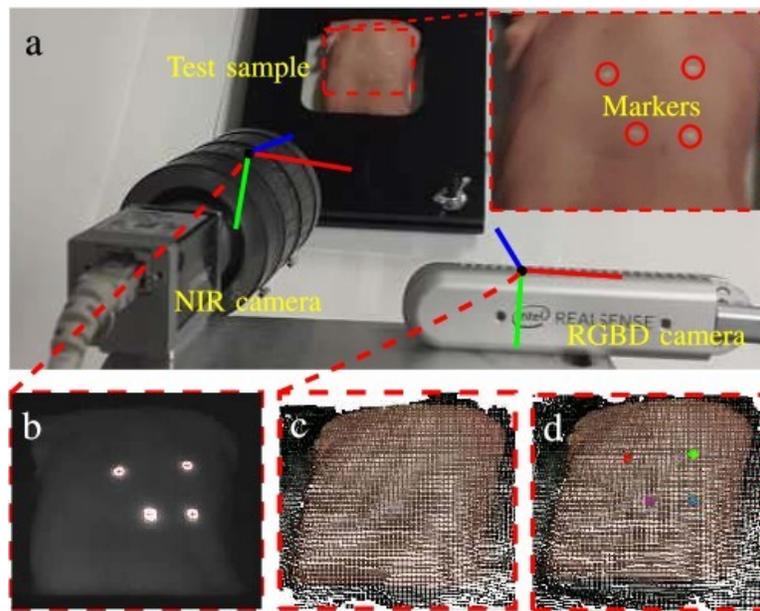


Figura 1.7.: a) Sistema de cámara dual, b) Imagen de la cámara NIR de los marcadores en una muestra de lengua porcina, c) Nube de puntos de la muestra de prueba obtenida de la cámara RGBD, y d) Posiciones superpuestas de los marcadores en la nube de puntos 3D para el robot de planificación de rutas.

Artículo	Año	Análisis
<p><b>Semi-autonomous Robotic Anastomoses of Vaginal Cuffs Using Marker Enhanced 3D Imaging and Path Planning</b></p>	<p><b>2019</b></p>	<p>El algoritmo presenta un algoritmo destinado a la planificación de trayectorias 3D para el robot Smart Tissue Autonomous (STAR). El propósito principal es facilitar anastomosis robóticas semi-autónomas en tejidos deformables. Este algoritmo incluye la utilización de marcadores NIR 3D colocados manualmente en el tejido antes del procedimiento, la creación de un plan uniforme y consistente para la colocación de suturas mediante métodos de planificación de trayectorias 3D basados en la ubicación de los marcadores NIR, y la actualización del plan de sutura restante después de cada punto completado. Esto se logra mediante una técnica de registro no rígido que tiene en cuenta la deformación del tejido durante la anastomosis. Los resultados de las pruebas indican que, al utilizar el método propuesto, STAR logra una consistencia 2,6 veces mayor en el espaciado de las suturas y 2,4 veces mayor en el tamaño de la mordida de la sutura en comparación con la anastomosis manual.</p> <p>El algoritmo de planificación de trayectorias 3D mejora la precisión y consistencia en la colocación de suturas en tejidos deformables. Esto, a su vez, mejora la calidad de las anastomosis realizadas por robots en comparación con las realizadas de forma manual, siendo particularmente relevante en cirugías delicadas como la anastomosis de cuellos vaginales. [1]</p>

<p><b>An Optimization-Based Algorithm for Trajectory Planning of an Under-Actuated Robotic Arm to Perform Autonomous Suturing</b></p>	<p><b>2020</b></p>	<p>El texto presenta un algoritmo de optimización diseñado para la planificación de trayectorias de sutura autónoma en cirugías de acceso único utilizando un brazo robótico subactuado. Este algoritmo busca aproximar la trayectoria ideal de sutura al realizar pequeñas reorientaciones de la aguja, con el objetivo de minimizar la desviación respecto a la trayectoria ideal y garantizar plena libertad de grado durante el procedimiento. La precisión del algoritmo fue evaluada en diversas ubicaciones iniciales de sutura dentro del espacio de trabajo y para diferentes tamaños de aguja.</p> <p>Los resultados indican que en el 13 por ciento del espacio de trabajo examinado, la desviación acumulada fue inferior a 10 mm, mientras que en el resto del espacio de trabajo, la desviación acumulada fue menor a 30 mm. También se observó una disminución en la desviación acumulada al utilizar agujas más pequeñas. El algoritmo demostró ser eficaz para mejorar la precisión de un robot de cuatro grados de libertad al llevar a cabo trayectorias restringidas, y se destacó la relación entre la precisión y el espacio de trabajo en la realización de suturas automáticas. [2]</p>
---------------------------------------------------------------------------------------------------------------------------------------	--------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p><b>Toward Autonomous Minimally Invasive Surgery: A Hybrid Framework Combining Task-Motion Planning and Dynamic Behavior Trees</b></p>	<p><b>2023</b></p>	<p>El texto presenta un enfoque novedoso que combina la Planificación de Tareas y Movimientos (TMP) con los Árboles de Comportamiento Dinámicos (ABD) para enfrentar los desafíos de coordinar múltiples herramientas quirúrgicas en un espacio de trabajo limitado durante cirugías mínimamente invasivas. Este método mejora las tasas de éxito, reduce los tiempos de ejecución y minimiza las necesidades de reposicionamiento en comparación con enfoques individuales. Además, demuestra ser resistente incluso en condiciones de ruido aumentado. Al aplicar este marco a una tarea compleja de sutura multi-lanzada, se destaca su capacidad para manejar tareas de sutura completas, lo que sugiere su potencial aplicación en la robótica quirúrgica del mundo real y su relación con la sutura automática. Este enfoque híbrido se presenta como una solución efectiva y robusta para la coordinación de herramientas quirúrgicas en espacios de trabajo limitados, particularmente en cirugías mínimamente invasivas. [3]</p>
------------------------------------------------------------------------------------------------------------------------------------------	--------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p><b>Optimization-Based Constrained Trajectory Generation for Robot-Assisted Stitching in Endonasal Surgery</b></p>	<p>2021</p>	<p>El texto aborda el desafío de realizar tareas quirúrgicas complejas, como la sutura, en la cirugía endoscópica endonasal (EES), donde el espacio limitado dificulta estimar distancias y ángulos para realizar movimientos de sutura precisos. Aunque los sistemas quirúrgicos robóticos han mostrado promesa en cirugía laparoscópica, la sutura asistida por robots sigue siendo un reto.</p> <p>Por ello, se propone un método de sutura asistida por robots que se basa en generar en línea trayectorias óptimas para suturar con agujas curvas. Además, se implementa un marco de planificación de movimiento con restricciones para garantizar un movimiento seguro del instrumento quirúrgico. La trayectoria de la aguja se crea en línea mediante un algoritmo de optimización convexa secuencial, sujeto a restricciones cinemáticas de sutura.</p> <p>El planificador de movimiento con restricciones busca reducir daños alrededor de la cavidad nasal estableciendo un centro de movimiento remoto sobre la nariz. Se propone un solucionador de cinemática inversa dual y concurrente para lograr una solución convergente y ejecución óptima del tiempo. Se utilizan simultáneamente dos métodos de cinemática inversa con restricciones: uno basado en prioridad de tarea y otro basado en optimización no lineal.</p> <p>Los resultados muestran una mejora significativa en la tasa de éxito de la sutura en el modo asistido por robot y un tiempo de finalización más corto en el modo autónomo. Además, se logra una reducción significativa en la interacción de fuerza con el tejido al utilizar el sistema robótico, este artículo presenta un método de sutura asistida por robots para la cirugía endoscópica endonasal que mejora la precisión y reduce los daños a los tejidos mediante la generación de trayectorias óptimas y la planificación de movimientos seguros y eficientes. [4]</p>
----------------------------------------------------------------------------------------------------------------------	-------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p><b>Incorporating Uncertainty Into Path Planning for Minimally Invasive Robotic Neurosurgery</b></p>	<p><b>2021</b></p>	<p>El texto se centra en abordar la incertidumbre en la planificación de trayectorias para la neurocirugía robótica mínimamente invasiva. Los autores proponen utilizar modelos más precisos de incertidumbre en lugar de simplemente agregar un margen de seguridad arbitrario. Exploran dos fuentes de incertidumbre en la planificación de trayectorias: la incertidumbre en la segmentación y la incertidumbre causada por el desplazamiento cerebral. Se introduce un método para calcular la incertidumbre del desplazamiento cerebral y se demuestra cómo combinar estas incertidumbres en un único volumen de riesgo. Este enfoque puede resultar especialmente útil en el contexto de la sutura automática, ya que la planificación precisa de la trayectoria es crucial para evitar daños a estructuras críticas y lograr una sutura exitosa. [5]</p>
--------------------------------------------------------------------------------------------------------	--------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p><b>Safe and Fast Path Planner for Minimally Invasive Surgery</b></p>	<p><b>2021</b></p>	<p>Se emplea un método típico de planificación de trayectorias robóticas que describe los movimientos posibles de la herramienta a través del espacio <math>C</math>. Sin embargo, a diferencia de la planificación de trayectorias robóticas convencional, la descripción del espacio <math>C</math> no ofrece una solución cerrada debido al elevado grado de libertad asociado con la herramienta manejada por manos humanas. Por lo tanto, se propone un enfoque basado en intervalos para describir el espacio <math>C</math>. Este método tiene la capacidad de dividir el espacio en intervalos factibles y no factibles de varios tamaños, lo que ayuda a reducir la zona de búsqueda y a abordar los obstáculos de manera más detallada. El artículo presenta el cálculo de trayectorias que son libres de colisiones y rápidas entre cualquier par de puntos en un entorno quirúrgico 2D con obstáculos para un robot de herramientas quirúrgicas. [6]</p>
-------------------------------------------------------------------------	--------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>Automated Extraction of Surgical Needles from Tissue Phantoms</b>	<b>2019</b>	El artículo aborda el desafío de realizar suturas precisas y consistentes en cirugía ocular, especialmente considerando la delicada naturaleza del tejido corneal. Para mejorar la precisión y consistencia de las suturas, se propone un sistema que utiliza un enfoque robótico. Este sistema se beneficia de la integración de sensores de fuerza y visión, lo que le permite adaptarse a variaciones en el tejido y en el tamaño de la sutura. Esta capacidad mejorada aumenta la versatilidad y eficacia del sistema, siendo especialmente relevante en el contexto de la sutura automática, donde la precisión y la consistencia son factores críticos para lograr resultados exitosos en cirugía ocular. [7]
----------------------------------------------------------------------	-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p><b>Path planning of flexible ureteroscope based on CT Image</b></p>	<p><b>2019</b></p>	<p>El artículo propone un método para planificar la trayectoria de un ureteroscopio flexible utilizando imágenes de tomografía computarizada (CT). Inicialmente, se mejora la calidad de la imagen CT mediante un preprocesamiento que incluye filtrado y segmentación. Luego, se realiza una reconstrucción en 3D para visualizar la imagen. Después de estos pasos, se emplea un algoritmo de distancia al borde para extraer puntos voxel, los cuales se conectan para formar la línea central. Sin embargo, la línea resultante puede ser muy tortuosa debido a la gran cantidad de puntos voxel y cambios abruptos de dirección. Por lo tanto, se propone suavizar la línea central mediante la selección de puntos voxel y restricciones de ángulo.</p> <p>Además, para abordar el problema de la limitada flexión cuando el ángulo de giro del ureteroscopio es alto, se utiliza la curvatura para restringir la línea central. Esto permite que el ureteroscopio siga el camino planificado. Los resultados obtenidos en modelos y cuerpos humanos reales demuestran que este enfoque cumple plenamente con los requisitos de un ureteroscopio flexible. [8]</p>
------------------------------------------------------------------------	--------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p><b>Hybrid Path Planning of Robots Through Optimal Control and PSO Algorithm</b></p>	<p><b>2019</b></p>	<p>El artículo presenta un algoritmo híbrido para planificar las trayectorias de robots, utilizando una combinación de control óptimo y PSO (Optimización por Enjambre de Partículas). Esta fusión permite obtener soluciones iniciales y luego optimizar la trayectoria del robot mediante la mejora de la solución inicial en relación con una función de costo. El método se aplica a un sistema robótico quirúrgico de 4 grados de libertad (DOF) y demuestra que la función de costo no está vinculada a la suposición inicial, lo que conduce a una optimización global. [9]</p>
----------------------------------------------------------------------------------------	--------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p><b>Autonomous task planning and situation awareness in robotic surgery</b></p>	<p><b>2020</b></p>	<p>El escrito se centra en resolver el desafío de automatizar tareas quirúrgicas estructuradas que implican la toma de decisiones en entornos complejos. Aunque los robots han mejorado la calidad de procedimientos quirúrgicos mínimamente invasivos, las tareas estructuradas aún dependen de la intervención humana. El artículo propone un marco para automatizar estas tareas quirúrgicas.</p> <p>Este marco consta de tres módulos principales: un módulo de razonamiento a nivel de tareas basado en programación de conjuntos de respuestas, un módulo de planificación de movimiento a bajo nivel basado en primitivas de movimiento dinámicas y un módulo de conciencia situacional. La programación de conjuntos de respuestas facilita la generación de planes comprensibles y la recuperación de condiciones de error, identificadas y explicadas por el módulo de conciencia situacional. Las primitivas de movimiento dinámicas permiten imitar la destreza de los cirujanos y adaptarse a obstáculos y cambios en el entorno. La efectividad de este marco se valida mediante diferentes versiones de la tarea estándar de entrenamiento quirúrgico del pasador y el anillo. [10]</p>
-----------------------------------------------------------------------------------	--------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

<p><b>Trajectory planning for a 6-DoF manipulator used for orthopaedic surgery</b></p>	<p><b>2020</b></p>	<p>En el ámbito de la cirugía ortopédica, los robots con múltiples grados de libertad (DoFs) se utilizan cada vez más para mejorar la precisión y estabilidad de los procedimientos quirúrgicos. Este artículo introduce un manipulador recientemente diseñado con 6 grados de libertad (6-DoF) específicamente para la cirugía escapula mínimamente invasiva. Dada la complejidad del quirófano, especialmente en la cirugía ortopédica, se propone un método de planificación de trayectorias seccionadas para evitar colisiones con articulaciones lesionadas. Los resultados de simulación indican que este enfoque es eficaz para generar trayectorias suaves y está listo para ser sometido a ensayos clínicos. [11]</p>
----------------------------------------------------------------------------------------	--------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p><b>Autonomous path planning for robot-assisted pelvic fracture closed reduction with collision avoidance</b></p>	<p><b>2023</b></p>	<p>El escrito se enfoca en resolver el desafío de planificación de trayectorias en la cirugía RPFRCR. Se propone una estrategia que combina un método de detección de colisiones con una planificación de orientación acoplada en el algoritmo A* para optimizar la trayectoria de reducción y aumentar la seguridad quirúrgica. Los resultados indican mejoras tanto en la longitud de la trayectoria como en la seguridad durante la cirugía. Este enfoque tiene relevancia en la sutura automática, ya que ambos comparten el uso de robots quirúrgicos y la optimización de trayectorias para mejorar el procedimiento quirúrgico. [12]</p>
---------------------------------------------------------------------------------------------------------------------	--------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p><b>Multi-objective path planning for lung biopsy surgery</b></p>	<p><b>2022</b></p>	<p>Aborda el desafío de planificación previa en la biopsia de pulmón guiada por imágenes de CT, necesario para determinar los puntos óptimos de inserción de la aguja en la piel del paciente. El enfoque tradicional depende de la experiencia del cirujano y puede afectar el éxito de la cirugía debido a la selección inadecuada del camino. Por lo tanto, se propuso un método de planificación quirúrgica inteligente para la biopsia de pulmón, que resuelve un problema de optimización multi-objetivo utilizando el concepto de optimización de Pareto.</p> <p>Se realizó una segmentación de órganos clave del pecho basándose en imágenes de CT y se analizaron cuantitativamente varios criterios clínicos. Como resultado, se proporcionaron algunos caminos quirúrgicos óptimos para los cirujanos. La evaluación retrospectiva de datos clínicos demostró que el método propuesto podría ser aplicado con éxito en la biopsia de pulmón. [13]</p>
---------------------------------------------------------------------	--------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

<p><b>Autonomous Robot for Removing Superficial Traumatic Blood</b></p>	<p><b>2021</b></p>	<p>Se introduce un sistema robótico diseñado para llevar a cabo la tarea quirúrgica de eliminación de sangre. Este sistema incluye un par de cámaras duales, un brazo robótico con 6 grados de libertad, un aspirador cuyo mango está conectado al brazo robótico y una bomba vinculada al aspirador. Además, se ha desarrollado un algoritmo de planificación de rutas para determinar la trayectoria que la punta del aspirador debe seguir para eliminar la sangre de manera efectiva.</p> <p>Los resultados obtenidos muestran que el robot BRR logra limpiar completamente la sangre en experimentos simulados de sangrado en tejido porcino ex vivo. Este estudio representa un avance al presentar el primer diseño de un robot médico autónomo especializado en la eliminación de sangre, demostrando que puede llevar a cabo la tarea quirúrgica de manera eficiente. Esta tecnología tiene aplicaciones potenciales en cirugías que requieren la sutura de heridas y la eliminación precisa de sangre, lo que podría mejorar la calidad de las suturas y reducir el riesgo de complicaciones postoperatorias. [14]</p>
-------------------------------------------------------------------------	--------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p><b>An Automatic Path Planning Method of Pedicle Screw Placement Based on Preoperative CT Images</b></p>	<p><b>2022</b></p>	<p>El artículo presenta un método automático para planificar la colocación de tornillos pediculares basado en imágenes preoperatorias de tomografía computarizada (TC). El método utiliza un modelo de red de aprendizaje profundo previamente entrenado para segmentar la columna vertebral del paciente y clasificar cada vértebra individualmente. A continuación, se establece un sistema de coordenadas local en cada vértebra en función de sus características anatómicas.</p> <p>Después de reconocer los puntos de referencia en las imágenes, se identifican la región pedunculada y la ruta óptima para colocar el tornillo, lo que automatiza por completo el plan quirúrgico. Los resultados experimentales indican que el 93 por ciento de los planes generados pueden aplicarse directamente en las cirugías después de la evaluación del cirujano. Además de mejorar la precisión, este proceso totalmente automático reduce significativamente el tiempo necesario para la planificación quirúrgica, eliminando la necesidad de asistencia humana. [15]</p>
------------------------------------------------------------------------------------------------------------	--------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 1.1.: Artículos

## 2. Introducción y generalidades

### 2.1. Introducción

Actualmente, la robótica se ha convertido en una de las tecnologías más sofisticadas y avanzadas utilizadas por personas, empresas, industrias y maquinarias, ya que son autómatas creados para hacer, apoyar o solucionar problemáticas de cualquier índole. [23]

La robótica ha fusionado sus avances tecnológicos en varios campos, como la medicina, que se enfoca en prevenir, diagnosticar y tratar enfermedades en seres humanos. Así, surge la robótica médica, una especialidad que tiene como objetivo crear máquinas capaces de apoyar a los cirujanos durante cualquier tipo de cirugía, minimizando riesgos en procedimientos extenuantes, mejorando la precisión, agilizando procedimientos, incluso llegar donde al ser humano se le dificulta considerablemente. Esto ha generado una gran aceptación en el campo médico, debido a que ofrece varios beneficios en particular para los pacientes, tales como disminuir el riesgo de una mala ejecución de sutura, de tal manera evitando una cicatrización inadecuada.

Este documento presenta un sistema encargado de dirigir una pinza quirúrgica, de manera que realice un procedimiento de sutura, integrando áreas como la robótica y la medicina. En particular, se requiere generar una trayectoria adecuada para una pinza en el acceso de una aguja con el fin de unir los tejidos y cerrar la herida para su recuperación, haciendo uso de la trayectoria más adecuada para la pinza, se logrará un cierre limpio de la herida.

La sutura es el proceso mediante el cual el cirujano desplaza una aguja con hilo a través de una herida con el fin de unir sus tejidos para sellar y permitir su recuperación. Convencionalmente, este proceso es realizado directamente, ya sea en sutura externa o laparoscópica. Sin embargo, en la actualidad este tipo de procedimientos también se efectúan con robots quirúrgicos como el robot Da Vinci que ejecuta procedimientos mediante un conjunto de brazos e instrumental especializado. Utiliza una interfaz tipo joystick que le permite al cirujano desplazar el robot de una manera teleoperada. En particular, la sutura robótica resulta ser demandante y compleja para la mayoría de los cirujanos [24]. Lo que causa esta dificultad es que carece de realimentación de

fuerzas, además se pierde la percepción de profundidad [25]. Una posible evolución de las tareas teleoperadas en robótica quirúrgica es la ejecución de procedimientos semi automatizados, procurando mejorar los resultados para el paciente y beneficiar al cirujano a cargo. En el presente trabajo se planea el diseño de un planificador de trayectoria para sutura semiautomática. Para acotar el problema se propone utilizar una pinza laparoscópica, que tienen un eje o aguja extendida para poder maniobrar a través de incisiones que generalmente no superan los 5 mm durante los procedimientos mínimamente invasivos, permiten una rotación de 360 grados de la punta de agarre a través de una rueda o perilla ubicada en el mango. Esto generalmente se coloca en la parte superior del instrumento, donde el mango se une al eje, de modo que puede manipularse con el pulgar que se muestra en la figura 2.1 [26].

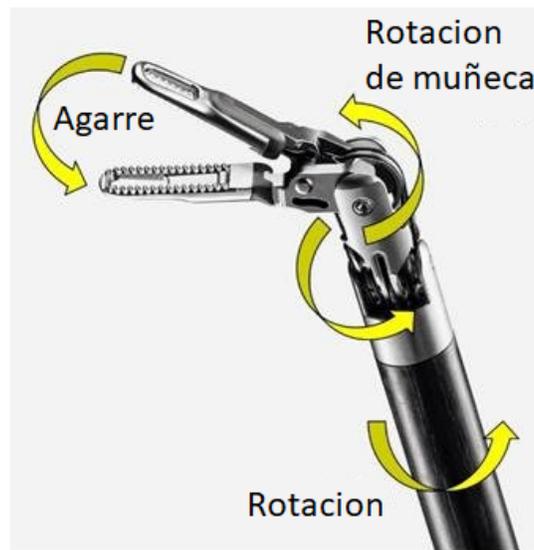


Figura 2.1.: Pinza laparoscópica [26].

Esta pinza accede a una serie de movimientos como lo son su mango más largo con una rotación de 360 grados, una rotación de su muñeca de 180 grados, la cual lleva consigo la pinza de agarre con una apertura de alrededor de 45 grados aproximadamente.

La pinza laparoscópica permite la manipulación de una aguja especial de sutura, la cual es curva para obtener una mejor manipulación y de esta manera hacer una puntada perfecta y un cierre correcto de la herida, en la figura 2.2, se puede observar cómo se realiza el agarre de la aguja con la pinza, la cual es la encargada de realizar la puntada uniendo los dos lados de la herida [27].

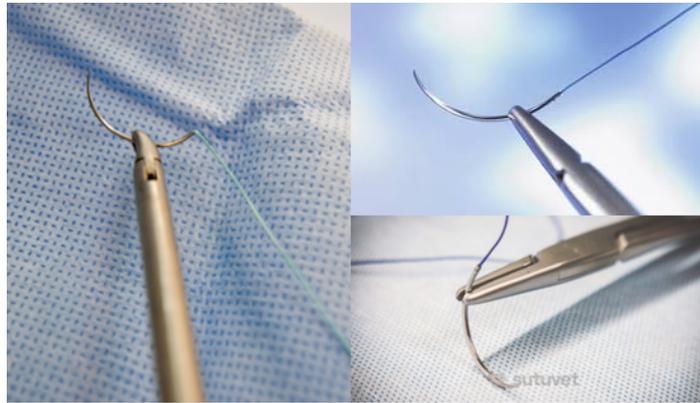


Figura 2.2.: Aguja de sutura sujeta de pinza laparoscópica [27].

La planificación de una trayectoria para un instrumento quirúrgico consiste en definir la mejor ruta a seguir para cumplir con una determinada tarea. En lo posible considerando áreas restringidas u obstáculos [28]. Un planificador de trayectorias que utilice la pinza laparoscópica para realizar una sutura ha de considerar las diferentes etapas de este procedimiento. La tarea de suturar consiste en dirigir la pinza para que ingrese la aguja en el borde de la herida y haga circular el hilo como se representa en la siguiente figura [29].

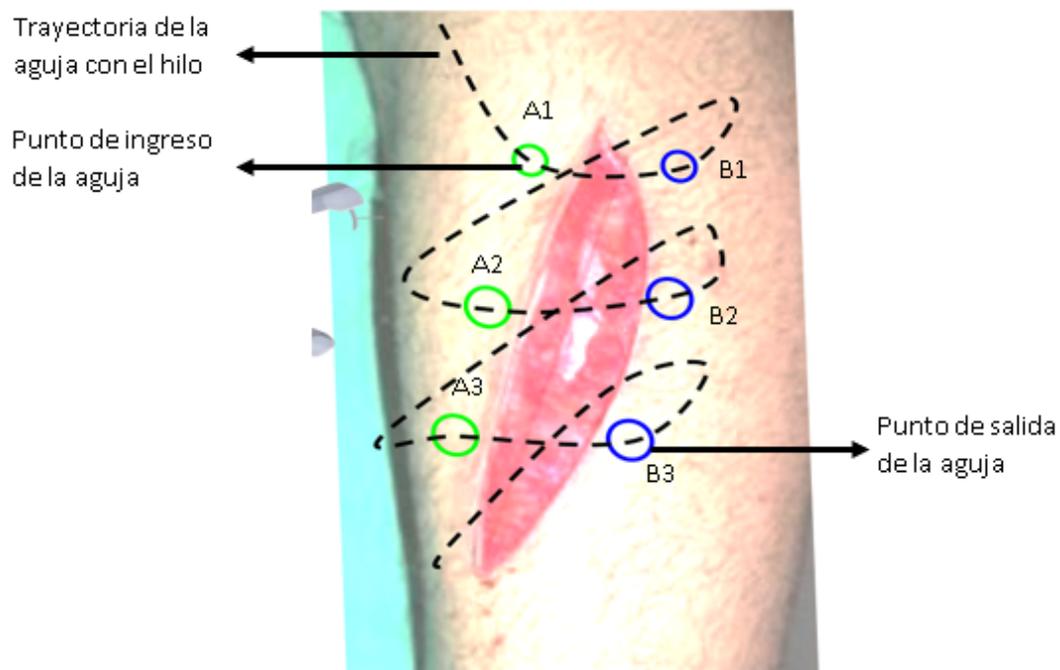


Figura 2.3.: Trayectoria de aguja e hilo para sutura de herida

En la figura 2.3 se observa una herida de tipo incisa en el brazo y el recorrido que la aguja debe realizar iniciando por penetrar el punto A1 atravesando el borde interno de la herida, paso seguido se desplaza y se orienta la pinza para hacer ingresar la aguja por el punto B1.

A continuación, la pinza se debe alejar de la herida para hacer pasar el hilo. El siguiente paso es dirigirse hasta el punto B2 e ingresar la aguja para sacarla por punto B2. Los pasos descritos se repiten hasta hacer circular el hilo por toda la herida, completando una sutura simple continua. Es importante resaltar que para la ejecución del procedimiento mencionado se requiere desplazar la aguja, hacer rotaciones en los diferentes ejes, y establecer el momento para cerrar la pinza. Por lo descrito, el planificador ha de considerar la ruta y el accionamiento de la pinza [30].

El presente trabajo investigativo tiene como propósito desarrollar un sistema de generación de trayectorias para realizar una sutura de punto continuo, teniendo en cuenta la herida, se calcula la trayectoria y se la envía a los robots para ejecutarla, siendo dos robots los encargados del paso de la aguja a través de la herida, para lo cual fue necesario hacer uso de MATLAB para generar las trayectorias, el sistema operativo Ros para enlazar el simulado con MATLAB y Unity para el desarrollo del ambiente de simulación donde se observan las trayectorias ejecutadas por los robots, con el objetivo de realizar esta tarea de forma segura para el paciente y menos compleja para el cirujano. El aporte a realizar por medio del presente documento es proveer de un sistema de generación de trayectorias que permita la ejecución de una sutura de punto simple continuo.

## **2.2. Organización del documento**

En el primer capítulo, se proporciona un resumen completo que incluye la introducción, los objetivos, el planteamiento del problema, los aportes y la metodología a seguir. El segundo capítulo abarca la recopilación de información relacionada con heridas, instrumentación para suturas y diversas técnicas de sutura, en cumplimiento del primer objetivo específico previamente mencionado. El tercer capítulo detalla el proceso de construcción del generador de trayectorias y la comunicación entre los entornos de programación y los entornos de simulación utilizados a lo largo del proyecto. Finalmente, en el cuarto capítulo, se presentan los resultados obtenidos del generador de trayectorias, que fueron evaluados en un ambiente de simulación quirúrgico, incluyendo gráficos e imágenes de todo el proceso.

## **3. Marco teórico**

### **3.1. Introducción**

Las heridas son lesiones que pueden afectar la integridad de los tejidos y en muchos casos, requieren de un procedimiento de sutura para lograr su cierre y promover una adecuada cicatrización. En este capítulo, se explora el mundo de las heridas, su clasificación en diferentes tipos y los instrumentos utilizados en los procedimientos de sutura para abordar cada una de ellas.

Los distintos tipos de heridas, van desde las abrasiones y laceraciones hasta las heridas penetrantes y contusas. Cada tipo de herida presenta características únicas que requieren enfoques específicos de sutura y el uso adecuado de instrumentos especializados.

Luego, se presentan los instrumentos esenciales para llevar a cabo los procedimientos de sutura. Las pinzas y las agujas desempeñan un papel fundamental en el proceso de sutura, y se estudian en detalle sus características y usos específicos. Desde las pinzas de disección hasta las pinzas hemostáticas, y desde las agujas rectas hasta las curvas, se detalla cada instrumento y cómo pueden facilitar el proceso de sutura.

Además, se analiza las diferentes técnicas de sutura empleadas por los profesionales de la salud. Desde la sutura continua hasta la sutura en puntos separados, pasando por la sutura en Z y la sutura sub dérmica, se analiza las ventajas y consideraciones de cada técnica según el tipo de herida y los objetivos de cicatrización.

### **3.2. Heridas**

Las heridas son el resultado de una rotura de la superficie de la piel, causada muchas veces por un objeto cortante. Las lesiones tisulares se reparan mediante el recambio del tejido lesionado. Si la pérdida de tejido es poco importante, suele bastar con aproximar los bordes de la herida, pero si la pérdida de tejido es considerable, la curación se prolonga durante mucho más tiempo. Las heridas son causadas por cortes, desgarros, quemaduras y aplastamientos de la piel. También pueden ser consecuencia del calor y

del enfriamiento, como se observan algunos ejemplos en la figura 3.2 [31].



Figura 3.1.: Herida [31].

### 3.3. Tipos de heridas

Las heridas se clasifican según extensión, localización, profundidad, gravedad, pronóstico y agente causal. Las heridas superficiales afectan solo la piel, mientras que las profundas dañan tejidos subyacentes. Las heridas pueden ocurrir en cualquier parte del cuerpo y su gravedad puede variar desde menores hasta potencialmente mortales. El pronóstico depende de varios factores, incluyendo el tipo de herida y la salud del individuo. Las heridas pueden ser causadas por varios agentes, incluyendo objetos físicos, calor o frío extremo, electricidad, productos químicos y radiación. Están las heridas abiertas, contusas, entre otras, a continuación se describen en detalle algunas de ellas observadas en la figura 3.2 [31].

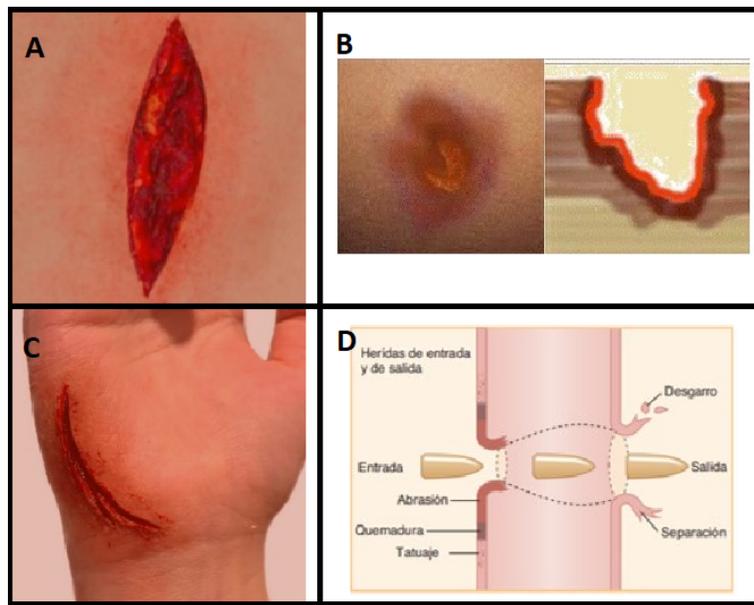


Figura 3.2.: Tipos de heridas [31].

### 3.3.1. A. Heridas abiertas

Las heridas abiertas son lesiones en la piel que resultan de un traumatismo, una incisión quirúrgica o una quemadura, y se caracterizan por la exposición de tejido subyacente, como músculos, huesos o estructuras internas. Estas heridas pueden variar en profundidad y gravedad, y su tratamiento adecuado es esencial para prevenir infecciones y promover una recuperación exitosa [16].

### 3.3.2. B. Heridas contusas

Las heridas contusas son lesiones de la piel y tejidos subyacentes causadas por un impacto o un trauma contundente, como golpes, caídas o choques. A diferencia de las heridas abiertas, no hay una ruptura visible en la piel, pero pueden ser igualmente dolorosas y con potenciales complicaciones [32].

### 3.3.3. C. Heridas incisas

Las heridas incisas son lesiones que se caracterizan por cortes limpios y profundos en la piel y los tejidos subyacentes. Estas heridas se producen comúnmente debido a objetos afilados o cortantes y requieren un tratamiento específico para prevenir complicaciones [33].

Las heridas incisivas son causadas por situaciones en las que se produce un corte nítido y profundo, como: objetos afilados, accidentes en el hogar, autolesiones.

### **3.3.4. D. Penetrantes**

Las heridas penetrantes son lesiones que se caracterizan por la perforación de la piel y los tejidos subyacentes por un objeto puntiagudo, como una bala, un cuchillo o un clavo. Estas heridas son profundas y pueden ser potencialmente graves, requiriendo una atención inmediata y específica [33].

## **3.4. Instrumental usado para sutura**

La sutura de heridas es un procedimiento común en la práctica médica para cerrar y unir los bordes de una herida con el fin de facilitar su cicatrización. Para llevar a cabo este procedimiento, se requiere de un conjunto de instrumentos especializados diseñados para realizar suturas de manera segura y efectiva [33].

Uno de los instrumentos más básicos y esenciales es la aguja de sutura. Estas agujas suelen ser de acero inoxidable y están disponibles en diferentes tamaños, son curvas, para adaptarse a las diferentes áreas del cuerpo y tipos de heridas. Las agujas de sutura se utilizan para pasar el hilo de sutura a través de los tejidos, permitiendo su unión [34].

### **3.4.1. Aguja quirúrgica**

Las agujas utilizadas en procedimientos de sutura desempeñan un papel fundamental al permitir el paso del hilo de sutura a través de los tejidos, facilitando su unión de manera eficiente. Estas agujas, confeccionadas en acero inoxidable, pueden presentar distintas formas, ya sean curvas o rectas, y su superficie puede tener una sección triangular o cilíndrica, aplanada, lo que influye en su capacidad para atravesar los tejidos de manera efectiva. Además, estas herramientas deben cumplir con una serie de características cruciales para garantizar un proceso de sutura exitoso. Por un lado, el orificio que generan en el tejido debe ser mínimo, permitiendo únicamente el paso del hilo necesario. Así mismo, es crucial que las agujas no introduzcan partículas extrañas en el organismo y que no debiliten la estructura del tejido. Por último, su diseño y construcción deben ser apropiados, garantizando un suturado exacto y preciso, lo que se traduce en una cicatrización óptima y una recuperación más efectiva para el paciente.

Estas características son esenciales para asegurar el éxito en procedimientos quirúrgicos y médicos donde se requiere sutura de tejidos [35].

En la figura 3.3, se observa los tipos de agujas, la forma de su cuerpo y punta, con la información suministrada se determinó hacer uso de una aguja quirúrgica de  $\frac{1}{2}$  para efectuar el proceso de sutura con un tamaño de 48 mm.

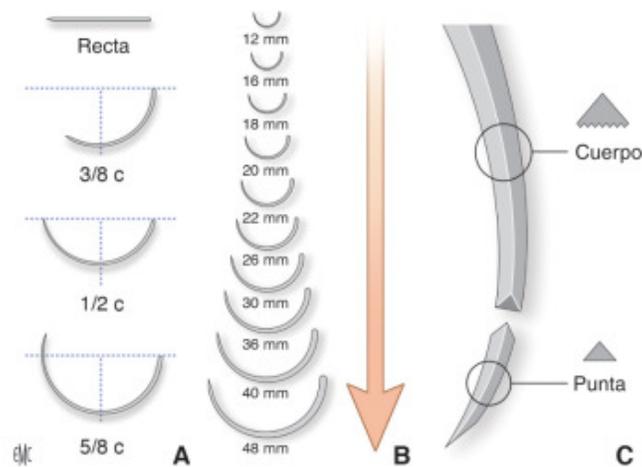


Figura 3.3.: Aguja quirúrgicas [35].

### 3.4.2. Características de las agujas quirúrgicas

Las agujas utilizadas en procedimientos de sutura presentan diversas características cruciales que determinan su eficacia en la perforación y unión de tejidos. En primer lugar, el extremo de la aguja se divide en diferentes tipos de puntas, cada una diseñada para un propósito específico. La punta cónica es ideal para tejidos blandos que requieren una fácil penetración, minimizando el traumatismo en el tejido, como en el caso de suturas intestinales. La punta roma, por otro lado, no corta el tejido y se utiliza en cirugías hepáticas y renales. La punta triangular cuenta con tres aristas cortantes y se emplea en tejidos de alta resistencia, como la piel. El tipo tapercut (PTC) combina características de la punta cónica y triangular y se usa en tejidos resistentes. Por último, la punta espatulada presenta dos aristas superiores y es adecuada para suturas en la córnea.

Además de la variedad de puntas, las agujas también se componen de un mandril en el extremo opuesto a la punta y un cuerpo que conecta ambos extremos. El uso de agujas rectas es común para suturar la piel y pueden manipularse con pinzas de disección o con los dedos. En cambio, las agujas curvas se manipulan con un portaagujas.

Las agujas más cerradas ( $1/2$  círculo) son apropiadas para suturas en áreas profundas, como cirugías gastrointestinales u oftálmicas, mientras que las más abiertas ( $1/4$  de círculo) se utilizan principalmente en cirugía oftálmica. Además de estos aspectos, los parámetros clave para definir una aguja incluyen su longitud, cuerda, radio, calibre y el arco comprendido, cada uno de los cuales desempeña un papel crucial en la elección adecuada de la aguja para procedimientos quirúrgicos y médicos específicos [36].

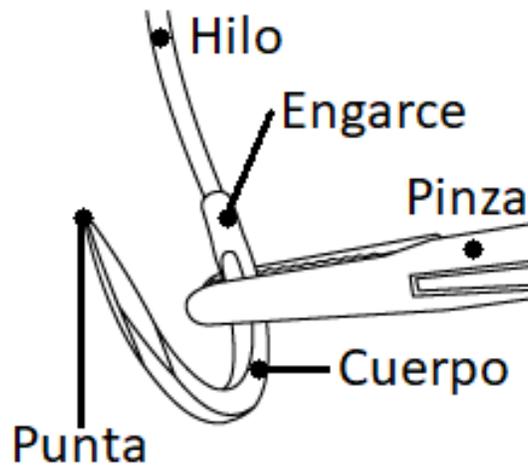


Figura 3.4.: Partes de una aguja quirúrgica  $1/2$  [36].

### 3.4.3. El hilo de sutura

Es otro componente importante. Hay varios tipos de hilos de sutura disponibles, como el hilo de sutura de nylon, polipropileno, seda y catgut, entre otros. Cada tipo de hilo tiene características específicas, como su resistencia, biocompatibilidad y tiempo de absorción, que lo hacen adecuado para diferentes tipos de heridas y situaciones clínicas. Además de la aguja y el hilo de sutura, se utilizan otros instrumentos para facilitar el procedimiento de sutura. Estos instrumentos incluyen las pinzas de disección, que se utilizan para manipular los tejidos y sostener la aguja durante la sutura. Las tijeras de sutura, que son pequeñas y tienen una punta roma, se emplean para cortar el hilo de sutura una vez que se ha completado la sutura [36].

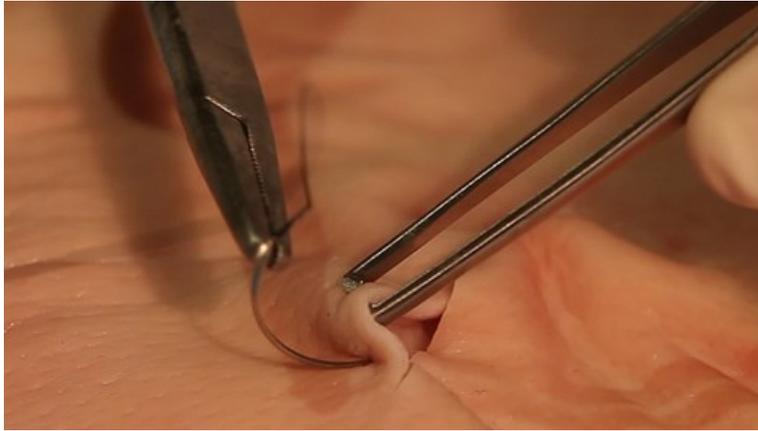


Figura 3.5.: Sutura de herida [36].

En la figura 3.5 se observa como es usada una aguja quirúrgica para un procedimiento de sutura, llevando la aguja con la pinza desde un extremo de la herida hasta el otro extremo de la herida, de esta manera el paso del hilo hace que la herida se cierre y posteriormente con el tiempo esta pueda cicatrizar adecuadamente.

#### **3.4.4. Pinzas quirúrgicas de laparoscopia**

Las pinzas son elementos esenciales en el ámbito quirúrgico y se componen de tres partes principales: mangos, vainas e insertos. Estas pinzas son desmontables y se utilizan para una variedad de propósitos en procedimientos médicos y quirúrgicos. En cuanto a las tijeras, se encuentran disponibles varios tipos, cada uno diseñado para tareas específicas. Entre ellas, se incluyen las tijeras de gancho, las tijeras Metzenbaum rectas y curvas, así como las microtijeras curvas y rectas. Estas tijeras desempeñan un papel fundamental en la realización de cortes precisos y controlados durante los procedimientos quirúrgicos. Es importante destacar que las pinzas pueden dividirse en dos categorías principales: las de disección y agarre, que pueden ser traumáticas para el tejido, y las pinzas de disección atraumáticas, que minimizan el daño a los tejidos. Entre las pinzas de disección y agarre, se encuentran las conocidas pinzas de Reddick-Olsen y las pinzas click-line, cada una con características y aplicaciones específicas en cirugía, como se observan en la figura 3.6 [34].



Figura 3.6.: Pinzas laparoscópicas [34].

La cirugía laparoscópica, también conocida como cirugía mínimamente invasiva, ha revolucionado la forma en que se realizan procedimientos quirúrgicos. Las pinzas quirúrgicas utilizadas en la laparoscopia son herramientas fundamentales que permiten a los cirujanos llevar a cabo operaciones a través de pequeñas incisiones en lugar de efectuar cortes más grandes en el cuerpo. Estas pinzas están diseñadas para proporcionar precisión y destreza en un espacio reducido. Aquí, se describen algunas de las pinzas quirúrgicas más comunes utilizadas en cirugía laparoscópica.

#### 3.4.4.1. La pinza EndoWrist

Es una innovación revolucionaria en el campo de la cirugía mínimamente invasiva, específicamente en el ámbito de la cirugía robótica. Diseñada y desarrollada por la empresa Intuitive Surgical, la pinza EndoWrist es una parte esencial de su sistema quirúrgico robótico conocido como el Sistema da Vinci.

La pinza EndoWrist es un componente fundamental que permite a los cirujanos realizar una amplia variedad de movimientos precisos dentro del cuerpo humano a través de pequeñas incisiones. La característica distintiva de la pinza EndoWrist es su capacidad para replicar con gran precisión los movimientos de la mano humana en

---

un espacio reducido, lo que la convierte en una herramienta valiosa para abordar una amplia gama de procedimientos quirúrgicos con mayor precisión y control.

Las características de la pinza EndoWrist son:

**Movilidad multidireccional:** La pinza EndoWrist es capaz de moverse en múltiples direcciones, lo que permite a los cirujanos realizar cortes, suturas y manipulaciones de tejido con una destreza y precisión asombrosas. Esta capacidad de movimiento es crucial para abordar áreas delicadas y de difícil acceso en el cuerpo.

**Ergonomía avanzada:** La pinza EndoWrist está diseñada ergonómicamente para adaptarse a la mano del cirujano, lo que reduce la fatiga y facilita un control preciso durante procedimientos largos y complejos.

**Articulación rotativa:** La pinza EndoWrist presenta una articulación rotativa que permite efectuar movimientos de muñeca tridimensionales, lo que amplía aún más las capacidades de precisión y maniobrabilidad en la cirugía.

**Amplia variedad de puntas intercambiables:** Para adaptarse a diferentes tipos de procedimientos, la pinza EndoWrist se puede equipar con una variedad de puntas intercambiables, como tijeras, pinzas, grapadoras y otros instrumentos especializados.

#### **3.4.4.2. Pinzas de agarre o pinzas de Kocher**

Estas pinzas son similares a las utilizadas en cirugía abierta, pero están diseñadas para ser más delgadas y largas. Se utilizan para agarrar y sostener tejidos o estructuras dentro del cuerpo durante la cirugía. Algunas de estas pinzas tienen dientes en sus extremos para proporcionar un agarre más seguro.

#### **3.4.4.3. Pinzas de disección o pinzas de Maryland**

Estas pinzas son delicadas y cuentan con una mandíbula más ancha en un extremo y una mandíbula más delgada en el otro. Son ideales para separar tejidos y despejar el campo de visión. Pueden utilizarse para diseccionar estructuras anatómicas con precisión.

#### **3.4.4.4. Pinzas de corte y sutura**

Estas pinzas tienen un extremo para cortar y otro para suturar. Son útiles para realizar cortes precisos en tejidos, así como para suturar vasos sanguíneos y tejidos durante la cirugía. Pueden ahorrar tiempo y simplificar el proceso quirúrgico.

#### **3.4.4.5. Pinzas de coagulación**

También conocidas como pinzas de electrocauterio, estas pinzas permiten la coagulación y el corte de tejidos a través de energía eléctrica. Son especialmente útiles para detener el sangrado durante la cirugía y para realizar cortes controlados en tejidos.

#### **3.4.4.6. Pinzas de endograpadoras**

Estas pinzas tienen la capacidad de cortar y grapar tejidos en un solo paso. Son fundamentales para realizar anastomosis (conexiones) en el tracto gastrointestinal, como la unión del intestino delgado o el colon.

#### **3.4.4.7. Pinzas de succión e irrigación**

Estas pinzas permiten la aspiración y la irrigación de fluidos dentro de la cavidad abdominal durante la cirugía. Mantienen un campo quirúrgico claro y ayudan a eliminar los residuos y la sangre.

#### **3.4.4.8. Pinzas de retención o fórceps**

Estas pinzas están diseñadas para mantener tejidos en su lugar durante la cirugía. Pueden ser útiles para estabilizar estructuras y permitir un trabajo más preciso.

Es importante destacar que el instrumental utilizado para suturar heridas debe ser esterilizado adecuadamente antes de su uso, para prevenir infecciones y asegurar la seguridad del paciente. Los profesionales de la salud que realizan suturas deben tener un conocimiento sólido sobre el empleo correcto de estos instrumentos, así como sobre las técnicas de sutura adecuadas para lograr resultados óptimos.

### **3.5. Técnicas de sutura**

Para cerrar las heridas cutáneas, es conveniente tener en cuenta la orientación de las líneas de menor tensión o líneas de Langer. Normalmente, se corresponden con las arrugas y son perpendiculares a la contracción de los músculos de la región [37].

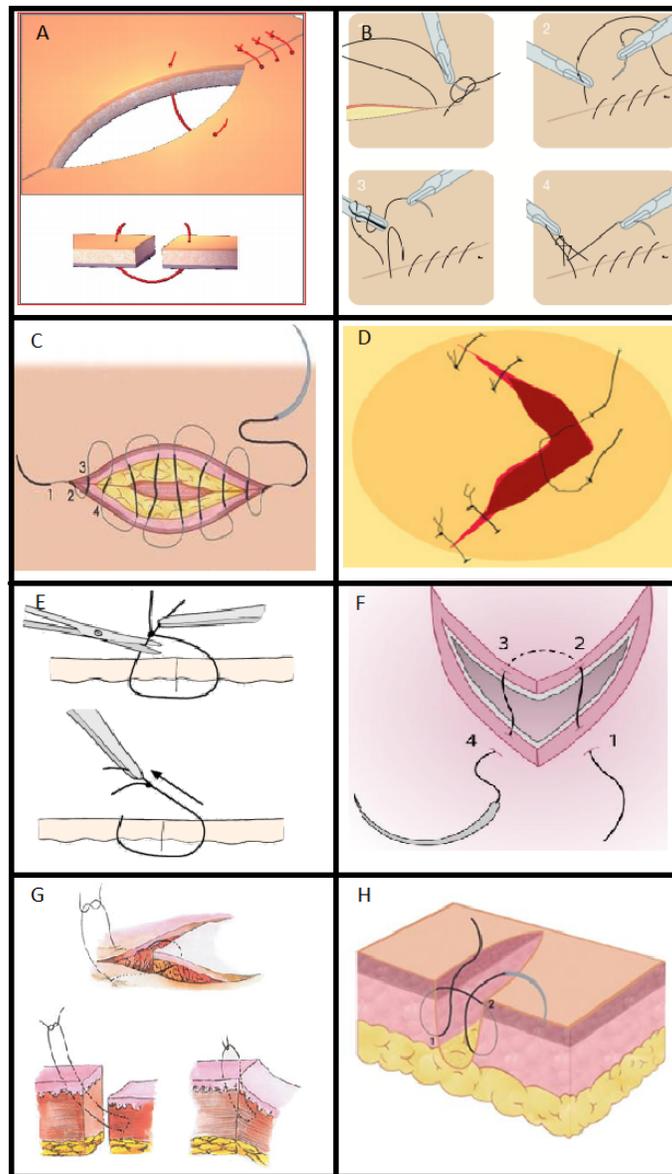


Figura 3.7.: Tipos de sutura [37].

### 3.5.1. A. Sutura con puntos sueltos o discontinuos

Con las pinzas de disección se eleva uno de los bordes de la herida, mientras que con el porta-agujas se introduce la aguja desde el exterior hacia el interior (de dermis a hipodermis). Se tira del extremo del hilo con la aguja hasta dejar un cabo distal corto. En el otro borde se realiza la misma operación, pero pasando el hilo desde el interior al exterior. Los puntos deben ponerse a unos 4 o 5 milímetros del borde de la herida y se deben espaciar entre ellos unos 6 a 8 milímetros, como se puede ver en la figura 3.7 [37].

Las suturas con puntos sueltos, también conocidas como suturas no absorbibles, se utilizan en diversas aplicaciones médicas y quirúrgicas donde es necesario un cierre temporal o una unión firme de los tejidos que no se disuelva con el tiempo. Algunas de las aplicaciones más comunes de las suturas con puntos sueltos incluyen:

**Cirugía general:** En procedimientos quirúrgicos, las suturas con puntos sueltos se utilizan para cerrar heridas profundas, como incisiones en el abdomen o tórax. También se emplean para unir capas de tejido, como músculos o fascia, y proporcionar soporte durante la recuperación.

**Cirugía vascular:** En cirugía vascular, son esenciales para unir vasos sanguíneos, arterias y venas, lo que permite la reparación de aneurismas o la restauración del flujo sanguíneo en procedimientos de bypass.

**Cirugía ortopédica:** En cirugía ortopédica, se utilizan para unir tejidos como tendones o ligamentos. Esto es especialmente importante en reparaciones de lesiones deportivas o en la fijación de fracturas.

### **3.5.2. B. Sutura continua**

La sutura continua es una técnica de sutura en la que se utiliza un solo hilo largo para coser toda la herida. Comienza con un primer punto de sutura que se anuda en el exterior y se corta solo el cabo distal. Luego, se cose toda la herida, manteniendo la aguja en el tejido subcutáneo y formando un ángulo de  $45^{\circ}$  con el eje de la herida, se finaliza realizando un segundo nudo en el exterior, de esta manera la sutura de tipo continua se realiza con un nudo de inicio, las puntadas de sutura y un nudo de final, para su correcta cicatrización. Es importante que la aguja salga a través de la piel del lado opuesto de donde entró para asegurar una distribución uniforme de la tensión a lo largo de la herida. Mantener una tensión constante en cada pase ayuda a cerrar la herida de manera uniforme y reduce el riesgo de formación de espacios vacíos o “pliegues” en la herida. Para terminar la sutura, se hace un nudo sobre el propio cabo distal. Este método es eficiente y rápido, ya que no requiere que se hagan nudos individuales después de cada pase de la aguja. Sin embargo, una desventaja es que si el hilo se rompe en cualquier punto, toda la sutura puede deshacerse. Esta técnica es comúnmente utilizada en cirugías donde se requiere una cicatrización rápida y eficiente, como en las cirugías abdominales. Sin embargo, no es adecuada para todas las heridas y su uso debe ser evaluado cuidadosamente por el cirujano. La sutura se describe en en la figura 3.7 [37].

La sutura continua, también conocida como sutura en punto continuo o sutura en racimo, es una técnica quirúrgica en la que se utiliza un solo hilo o material de sutura

---

para unir tejidos en una línea continua sin cortes intermedios. Esta técnica se emplea en diversas aplicaciones médicas y quirúrgicas debido a sus ventajas en términos de rapidez y estanqueidad. Algunas de las aplicaciones más comunes de la sutura continua:

**Cierre de heridas cutáneas:** La sutura continua es eficaz para cerrar heridas cutáneas superficiales y profundas, ya que permite un cierre uniforme y estanco. Se utiliza en cirugía menor, como la reparación de laceraciones o incisiones quirúrgicas.

**Cirugía de órganos internos:** En procedimientos quirúrgicos internos, como la cirugía gastrointestinal o ginecológica, se utiliza para unir capas de tejido, como el intestino o el útero, de manera segura y estanca. Esto ayuda a prevenir fugas y complicaciones postoperatorias.

**Cirugía vascular:** En cirugía vascular, se emplea para reparar arterias y venas, especialmente en procedimientos de bypass o en la corrección de aneurismas. Esta técnica garantiza la integridad de la unión vascular.

**Reparación de estructuras anatómicas:** Es útil en cirugía ortopédica y cirugía plástica para unir tendones, ligamentos o músculos. Proporciona una distribución uniforme de la tensión y promueve la curación adecuada.

**Cirugía maxilofacial:** Para cerrar incisiones en la boca o en la mandíbula, asegurando un cierre estanco y una apariencia estética.

**Cierre de órganos huecos:** En procedimientos como la gastrectomía o la colectomía, se utiliza para cerrar el estómago o el colon después de la extirpación de una parte del órgano.

**Cierre de la pared abdominal:** Después de una laparotomía, donde se realiza una incisión en la pared abdominal para acceder a la cavidad abdominal, se utiliza para cerrar las capas de la pared abdominal y proporcionar una unión segura.

**Cirugía oftálmica:** En procedimientos oculares, como la cirugía de cataratas, se emplea para cerrar incisiones corneales y garantizar la integridad del ojo.

**Cierre de incisiones uterinas en cesáreas:** Durante una cesárea, se utiliza para cerrar la incisión uterina, asegurando una reparación segura y un parto futuro más seguro.

### **3.5.3. C. Sutura continua intradérmica**

Se trata de unir la piel (hipodermis), sin sacar el hilo al exterior. Desde la profundidad de la herida, se introduce la aguja para que salga por la hipodermis, debajo de la superficie cutánea. Se vuelve a introducir por el otro labio de la herida, pero en esta ocasión desde arriba hacia abajo. Si se usa hilo reabsorbible, se anuda dejando los nudos por debajo de la hipodermis. Cuando se utiliza hilo reabsorbible (monofilamento 2/0 o

3/0), no hace falta anudar; los extremos se mantienen tensos fijando los extremos a la piel con esparadrapo. Cuando se vaya a retirar la sutura, se corta uno de los cabos a nivel de la piel y se saca tirando del otro extremo, como lo muestra la figura 3.7 [38].

La sutura continua intradérmica, también conocida como sutura sub cuticular, es una técnica de sutura que se utiliza en aplicaciones específicas para lograr un cierre estético y una mínima visibilidad de las incisiones. En esta técnica, se utiliza un hilo de sutura absorbible que se coloca debajo de la piel, por lo que no se ve en la superficie. Algunas de las aplicaciones comunes de la sutura continua intradérmica incluyen:

**Cirugía plástica:** Como la cirugía de mama, la abdominoplastia o la cirugía de contorno corporal, se utiliza la sutura continua intradérmica para cerrar incisiones y minimizar las cicatrices visibles. Esto contribuye a resultados estéticos satisfactorios.

**Cirugía dermatológica:** Como la extirpación de lunares o tumores cutáneos, se emplea para cerrar incisiones en la piel, de manera que las cicatrices sean menos notorias y los resultados sean más estéticos.

**Cirugía facial:** Como la blefaroplastia (cirugía de párpados), es ideal para cerrar incisiones en áreas visibles del rostro, permitiendo una recuperación estética y reduciendo la visibilidad de las cicatrices.

#### **3.5.4. D. Sutura de esquina**

Se introduce la aguja a través de la dermis por el lado contrario al colgajo, a unos 0,5 cm de la esquina de la herida. La aguja se lleva a la punta del colgajo por la hipodermis y se atraviesa la herida hasta salir por la dermis del lado opuesto al punto de entrada. Así, los dos cabos salen al exterior de la herida por la zona opuesta al colgajo, y es aquí donde se realiza el nudo. En el resto de la herida se usan los puntos discontinuos habituales, como lo indica la figura 3.7 [38].

Es una técnica de sutura que se utiliza para cerrar heridas en las esquinas de incisiones o laceraciones. Esta técnica se aplica específicamente en situaciones donde se necesita un cierre más preciso y estético en las esquinas de una herida. Algunas de las aplicaciones comunes de la sutura de esquina incluyen:

**Cirugía dermatológica:** Como la extirpación de lunares o tumores cutáneos, la sutura de esquina se emplea para cerrar las esquinas de las incisiones, de manera que las cicatrices sean menos notorias y los resultados sean más estéticos.

**Cirugía de la cabeza y cuello:** Como la tiroidectomía o la cirugía de paratiroides, se utiliza para cerrar las esquinas de las incisiones en zonas delicadas, donde las cicatrices visibles pueden ser problemáticas.

---

### **3.5.5. E. Sutura de punto simple**

Se pasa la aguja de un lado a otro de la herida, procurando mantener la misma distancia en los bordes y la misma profundidad, como se puede observar en la figura 3.7 [39].

La sutura de punto simple, también conocido como punto de sutura simple o punto de sutura en simple, es una técnica de sutura básica que implica la unión de tejidos con una sola pasada del hilo de sutura. Aunque es una técnica simple, tiene varias aplicaciones en la práctica médica. A continuación algunas de las aplicaciones comunes de la sutura de punto simple:

**Cierre de heridas cutáneas menores:** Se utiliza para cerrar heridas cutáneas superficiales y menores, como raspaduras, cortes pequeños y laceraciones superficiales. Proporciona un cierre rápido y efectivo.

**Suturas en cirugía pediátrica:** En procedimientos quirúrgicos en niños, como la corrección de hernias umbilicales, la sutura de punto simple se utiliza para cerrar incisiones y heridas menores.

### **3.5.6. F. Sutura punto de colchonero**

Se pasa la aguja por la herida, de un extremo al otro, a unos 0,5 cm del borde. A otros 0,5 cm del punto de salida, se vuelve a introducir la aguja para pasar de nuevo a través de toda la herida hasta el punto origen, pero de forma más profunda. Se mantiene la misma dirección en los cuatro puntos, así lo demuestra la figura 3.7 [39].

La sutura en punto de colchonero, también conocido como punto de colchón o punto de Lembert, es una técnica de sutura que se utiliza en aplicaciones específicas en cirugía y sutura médica. Se caracteriza por su forma de anudado, que proporciona un cierre seguro y estanco. Algunas de las aplicaciones comunes de la sutura punto de colchonero son:

**Cierre de vísceras huecas:** Se utiliza para unir órganos huecos, como el intestino delgado o el colon, en cirugía abdominal. Esta técnica proporciona un cierre hermético que ayuda a prevenir fugas y complicaciones, como las dehiscencias intestinales.

**Cirugía gastrointestinal:** Como la resección de una porción del intestino o la anastomosis de dos segmentos intestinales, se emplea para unir las capas de la pared intestinal de manera segura y estanca.

### 3.5.7. G. Sutura punto en U o colchonero horizontal

Cada uno de los puntos pasa de uno a otro borde de la herida realizando un trayecto intratisular en U. Los dos extremos del hilo quedan en el mismo lado de la herida, donde se anudan. Se pueden proteger con parches de Dacron (tejido sintético de poliéster) para que no desgarre el tejido en heridas friables, vasculares o cardíacas, como lo muestra la figura 3.7 [39].

La sutura punto en U, también conocido como punto de colchonero en U o punto en U de Halstead, es una técnica de sutura que implica crear un patrón en forma de “U” en la herida. Esta técnica se utiliza en diversas aplicaciones médicas y quirúrgicas para lograr un cierre seguro y estético. Algunas de las aplicaciones comunes se describen a continuación:

Cirugía maxilofacial: Se emplea para cerrar incisiones en la boca o la mandíbula, proporcionando un cierre seguro y estético en áreas delicadas.

Cirugía oral: Como la extirpación de quistes o tumores orales, se utiliza para cerrar incisiones en la boca, de manera que las cicatrices sean menos notorias y se logre un cierre seguro.

### 3.5.8. H. Sutura punto de Algöwer o punto Suizo

Que tiene la ventaja que la entrada y salida del hilo los hace por un solo lado de la herida. Como se observa en la figura 3.7 [39].

El punto de Algöwer, es una técnica de sutura utilizada en aplicaciones específicas en cirugía y sutura médica. Esta técnica implica un tipo de anudado que proporciona un cierre seguro y estanco. Algunas de las aplicaciones comunes de la sutura punto de Algöwer son:

Cierre de heridas cutáneas: Se utiliza para cerrar heridas cutáneas superficiales y laceraciones.

Cirugía dermatológica: Como la extirpación de lunares o tumores cutáneos, el punto de Algöwer se utiliza para cerrar incisiones de manera que las cicatrices sean menos visibles y los resultados sean más estéticos.

Después de realizar una exhaustiva recopilación de información sobre las herramientas quirúrgicas, como pinzas, agujas y suturas, se logró alcanzar exitosamente el primer objetivo específico planteado. A través de la diferente información recolectada, se determinó el enfoque para el generador de trayectorias, el cual se implementará mediante el uso de una pinza laparoscópica, una aguja quirúrgica curva, teniendo como método, la sutura de tipo continua en heridas de tipo incisas.

---

## 3.6. Planificador de trayectorias

Los métodos convencionales para la planificación de trayectorias en robots quirúrgicos incluyen una variedad de técnicas, que se centran en garantizar la seguridad, eficiencia y precisión durante los procedimientos quirúrgicos. Algunos de los métodos más destacados incluyen:

- Generación y seguimiento de trayectorias con Polinomios, B-splines, y Perfiles de Velocidad Trapezoidal: Estas técnicas utilizan ecuaciones matemáticas para generar trayectorias para robots manipuladores, permitiendo la interpolación entre matrices de rotación y transformaciones homogéneas.
- Diagrama de Voronoi: Amplía la distancia entre el robot y los obstáculos, creando un espacio libre de obstáculos mediante el uso de segmentos rectos y parábolas basados en la proximidad a los obstáculos.
- Roadmap Probabilístico (PRM): Genera un conjunto de configuraciones libres de colisión de forma aleatoria y uniforme a lo largo del área de trabajo. Luego, conecta los nodos con sus vecinos más cercanos y aplica algoritmos para encontrar la ruta óptima.
- Modelado del Espacio Libre: Utiliza cilindros rectilíneos generalizados (CRG) para representar los obstáculos como polígonos y planificar la trayectoria manteniendo al robot alejado de los obstáculos.
- Descomposición en celdas: Este método divide el espacio en celdas, utilizando un grafo de conectividad para definir la ruta. Un ejemplo es la descomposición trapezoidal, que se basa en construir rectas paralelas a partir de los vértices de los elementos del entorno.
- Campos Potenciales: Basado en técnicas reactivas para entornos desconocidos, considera al robot como una partícula en un campo potencial artificial, compuesto por fuerzas de atracción hacia el destino y repulsión de los obstáculos.
- Rapidly-exploring Random Trees (RRT): El RRT es un algoritmo ampliamente utilizado en la robótica para la planificación de trayectorias en espacios de alta dimensionalidad. Funciona extendiendo ramas desde un nodo inicial hacia el espacio de configuración hasta alcanzar el objetivo o cubrir el área de interés, ideal para entornos con obstáculos.

### 3.6.1. Generación y seguimiento de trayectorias con Polinomios, B-splines, y Perfiles de Velocidad Trapezoidal

La generación y seguimiento de trayectorias utilizando Polinomios, B-splines, y Perfiles de Velocidad Trapezoidal son métodos matemáticos y computacionales utilizados en la robótica, incluida la robótica quirúrgica, para planificar y controlar el movimiento de robots de manera precisa y eficiente.

- **Polinomios:** Estos son utilizados para generar trayectorias suaves que un robot puede seguir. Se definen matemáticamente y pueden ser ajustados para cumplir con condiciones iniciales y finales específicas, como posición, velocidad, y aceleración. Los polinomios de tercer orden (cúbicos) y de quinto orden (quinticos) son comúnmente usados para este propósito.
- **B-splines:** Son una generalización de los polinomios que permiten una mayor flexibilidad y control sobre la forma de la trayectoria. Los B-splines se construyen como una serie de segmentos polinomiales que se unen de manera suave. Esto permite definir trayectorias complejas que son fácilmente ajustables modificando los puntos de control sin alterar el resto de la trayectoria.
- **Perfiles de Velocidad Trapezoidal:** Este método se utiliza para generar trayectorias que cumplen con restricciones específicas de velocidad y aceleración. La velocidad del robot sigue un perfil trapezoidal, lo que significa que aumenta linealmente hasta alcanzar una velocidad máxima, se mantiene constante durante un período, y luego disminuye linealmente hasta detenerse. Esto asegura un movimiento suave y controlado, minimizando los impactos debido a cambios bruscos de velocidad.

Estos métodos tienen amplias aplicaciones en robótica, desde la manipulación industrial hasta la robótica quirúrgica. Permiten a los robots realizar tareas complejas con alta precisión, como ensamblajes delicados, pintura, soldadura, y procedimientos quirúrgicos donde es crucial el control preciso del movimiento.

En el contexto del movimiento autónomo de un robot quirúrgico, estos métodos permiten planificar trayectorias que el robot puede seguir para alcanzar y operar en un punto objetivo dentro del cuerpo del paciente con precisión milimétrica. Por ejemplo, al realizar una cirugía mínimamente invasiva, el robot debe ser capaz de navegar a través de espacios estrechos evitando tejidos sensibles y órganos vitales. Utilizando estos métodos, se pueden generar trayectorias optimizadas que minimizan el riesgo de daño a los tejidos circundantes y mejoran la eficacia del procedimiento quirúrgico. Estos métodos también

---

son esenciales para adaptar el movimiento del robot en tiempo real a los cambios en el entorno quirúrgico o a la respuesta del paciente. La capacidad de ajustar rápidamente las trayectorias en respuesta a la retroalimentación sensorial es clave para la seguridad y el éxito de las cirugías asistidas por robots.

### 3.6.2. Diagrama de Voronoi

El Diagrama de Voronoi es un método matemático utilizado en la planificación de trayectorias y el análisis espacial que divide un espacio en regiones basadas en la distancia a un conjunto de puntos específicos, conocidos como sitios de Voronoi. Cada región (o celda de Voronoi) contiene todos los puntos que están más cerca de un sitio de Voronoi específico que a cualquier otro. Las fronteras entre estas regiones son equidistantes de los dos sitios de Voronoi más cercanos y forman lo que se conoce como aristas de Voronoi. La construcción de un Diagrama de Voronoi comienza con la identificación de los sitios de Voronoi en el espacio. Luego, se trazan líneas o superficies equidistantes entre cada par de sitios adyacentes. Estas líneas o superficies forman las aristas de las celdas de Voronoi, separando cada sitio de sus vecinos más cercanos. El resultado es una partición del espacio que puede ser utilizada para varias aplicaciones de análisis y optimización.

Los Diagramas de Voronoi tienen una amplia gama de aplicaciones en diversos campos, tales como:

- Geografía y planificación urbana: Para modelar el área de influencia de centros urbanos o servicios, como hospitales, escuelas y estaciones de policía.
- Meteorología: Para interpolar datos de estaciones meteorológicas y predecir condiciones climáticas en áreas no medidas.
- Biología: En el modelado del crecimiento de células y tejidos, así como en el estudio de patrones de distribución de especies.
- Robótica y automatización: En la planificación de trayectorias para evitar obstáculos y optimizar rutas de navegación.

En el contexto de la robótica, especialmente en robots autónomos, el Diagrama de Voronoi se utiliza para planificar trayectorias seguras y eficientes en entornos con obstáculos. Al representar los obstáculos en el espacio, el Diagrama de Voronoi puede generar un camino que maximiza la distancia del robot a los obstáculos más cercanos, lo que resulta en una ruta segura que evita colisiones. Por ejemplo, en la robótica quirúrgica, donde el espacio de trabajo puede estar lleno de órganos y tejidos sensibles,

el Diagrama de Voronoi puede ayudar a planificar una trayectoria que minimice el riesgo de dañar tejidos circundantes al maximizar la distancia del instrumento quirúrgico a las estructuras vulnerables. En la navegación autónoma de vehículos o drones, los Diagramas de Voronoi pueden ser utilizados para planificar rutas en entornos urbanos o naturales, identificando caminos óptimos que eviten obstáculos y otros vehículos, mejorando así la seguridad y eficiencia del movimiento autónomo. La utilización del Diagrama de Voronoi en la planificación de trayectorias permite a los robots autónomos navegar de manera más segura y eficiente, adaptándose dinámicamente a los cambios en el entorno y optimizando sus rutas en tiempo real para cumplir con los objetivos de la tarea mientras minimizan los riesgos y maximizan la seguridad.

### 3.6.3. Roadmap Probabilístico (PRM)

El método de Roadmap Probabilístico (PRM, por sus siglas en inglés) es una técnica popular para la planificación de trayectorias en robótica, especialmente útil en entornos complejos y de alta dimensión. Esta técnica es particularmente adecuada para problemas de planificación de trayectorias donde se requiere navegar entre obstáculos en un espacio de configuración.

El PRM se basa en la construcción de un grafo (o "mapa") que captura las posibles rutas que un robot podría tomar en un espacio de configuración. El proceso se realiza en dos fases principales:

**Construcción del Grafo:** Se generan aleatoriamente una serie de puntos libres de obstáculos en el espacio de configuración del robot. Luego, estos puntos se conectan con otros puntos cercanos si el camino entre ellos también está libre de obstáculos. Estas conexiones forman las aristas del grafo. La generación de puntos y su conexión se hacen considerando las restricciones físicas y de movimiento del robot. **Búsqueda de Trayectoria:** Una vez construido el grafo, se utiliza un algoritmo de búsqueda (como para encontrar la ruta más corta o óptima desde el punto de inicio hasta el punto objetivo en el grafo. Esta ruta representa la trayectoria que el robot debe seguir.

El PRM se usa ampliamente en diversas aplicaciones robóticas, incluyendo:

**Robótica móvil:** Para la navegación de robots en entornos con obstáculos.

**Robótica manipuladora:** En la planificación de movimientos de brazos robóticos en entornos industriales o quirúrgicos.

**Robots autónomos:** En la exploración de terrenos desconocidos o para realizar tareas específicas en entornos complicados.

En el contexto del movimiento autónomo de un robot, el PRM permite al robot

---

planificar su trayectoria en entornos desconocidos o dinámicos de manera eficiente. Al generar un grafo de posibles rutas libres de obstáculos, el robot puede adaptar su planificación de trayectoria a cambios en el entorno, como la aparición de nuevos obstáculos o la modificación de los existentes. Por ejemplo, en la robótica quirúrgica, el PRM podría utilizarse para planificar la trayectoria de un instrumento quirúrgico que debe navegar a través de un espacio complejo evitando tejidos sensibles. En la robótica móvil, puede ayudar a un vehículo autónomo a encontrar el mejor camino a través de un entorno urbano complicado. El PRM es particularmente útil cuando el entorno es complejo y no se puede determinar a priori una trayectoria clara, permitiendo una planificación flexible y robusta frente a incertidumbres. La capacidad de generar rápidamente un mapa de rutas posibles y seleccionar la óptima basada en las condiciones actuales del entorno hace que el PRM sea una herramienta poderosa para la navegación autónoma de robots en una amplia gama de aplicaciones.

### **3.6.4. Modelado del Espacio Libre**

El método de Modelado del Espacio Libre es una técnica utilizada en la planificación de trayectorias para robots, que se enfoca en identificar y modelar las áreas navegables dentro de un entorno, excluyendo los obstáculos. Este enfoque permite a los robots determinar caminos seguros y eficientes para sus movimientos.

El Modelado del Espacio Libre implica la representación del entorno de trabajo del robot de manera que se destaquen las áreas sin obstáculos. Esto puede lograrse a través de varias técnicas, como:

Descomposición en celdas: El entorno se divide en una cuadrícula o celdas, y cada celda se clasifica como libre o bloqueada según si contiene o no obstáculos.

Descomposición exacta: El espacio libre se modela utilizando formas geométricas simples (como rectángulos o polígonos) que se ajustan alrededor de los espacios navegables, excluyendo los obstáculos.

Representación basada en sensores: Utilizando datos de sensores, el robot crea un mapa del entorno que refleja las áreas libres y ocupadas en tiempo real.

Este método tiene aplicaciones en diversas áreas de la robótica, como: Robótica móvil: Para la navegación autónoma de vehículos en entornos tanto estructurados como no estructurados. Robótica de servicio: En aplicaciones como la limpieza de pisos, donde el robot debe navegar eficientemente evitando muebles y otros obstáculos. Exploración y mapeo: En la exploración de entornos desconocidos, como cuevas o edificios derrumbados, donde es vital identificar áreas seguras para la navegación.

En el contexto del movimiento autónomo, el Modelado del Espacio Libre es fundamental para permitir que los robots tomen decisiones informadas sobre su navegación. Al tener un modelo claro de las áreas libres de obstáculos, un robot puede planificar su trayectoria para alcanzar un destino de manera eficiente, ajustando su camino en respuesta a cambios dinámicos en el entorno o a la detección de nuevos obstáculos. Por ejemplo, un robot autónomo que realiza entregas en un entorno urbano utilizaría el Modelado del Espacio Libre para planificar rutas óptimas a través de las aceras y cruces peatonales, evitando obstáculos como peatones, vehículos estacionados y construcciones. Este enfoque garantiza que el robot pueda moverse de manera segura y eficiente, adaptándose a las condiciones cambiantes del entorno en tiempo real. El Modelado del Espacio Libre es, por lo tanto, una herramienta crucial en la caja de herramientas de la robótica autónoma, proporcionando la base para la planificación de trayectorias seguras y eficientes en una amplia gama de aplicaciones.

### 3.6.5. Campos Potenciales

Es una técnica ampliamente utilizada en robótica para la planificación de trayectorias y la navegación autónoma de robots. Este enfoque se basa en la creación de campos de fuerzas artificiales que guían al robot hacia su objetivo mientras evitan obstáculos.

El método de Campos Potenciales involucra dos componentes principales:

**Campo Potencial Atractivo:** Este campo es generado por el objetivo o punto de destino del robot. Funciona como una fuerza que atrae al robot hacia el objetivo. Matemáticamente, este campo se modela de manera que su intensidad disminuye a medida que el robot se acerca al objetivo, incentivando un movimiento suave y dirigido hacia el punto deseado.

**Campo Potencial Repulsivo:** Este campo es generado por los obstáculos presentes en el entorno. Actúa como una fuerza que repele al robot, evitando colisiones. La intensidad de este campo aumenta a medida que el robot se acerca a un obstáculo, creando una 'zona de repulsión' alrededor de él.

La trayectoria del robot se determina entonces por la superposición de estos campos, buscando minimizar el potencial total en cada punto del espacio. El robot se mueve en la dirección del gradiente negativo del campo potencial, lo que le permite avanzar hacia el objetivo mientras esquiva obstáculos.

El método de Campos Potenciales es utilizado en una variedad de aplicaciones robóticas, como: Robótica móvil: Para la navegación de robots en entornos dinámicos con obstáculos móviles y estáticos.

---

Robótica manipuladora: En la planificación de movimientos para brazos robóticos que deben evitar colisiones con objetos en su entorno.

Vehículos autónomos: Para la planificación de rutas en entornos urbanos o naturales con múltiples obstáculos.

En el movimiento autónomo del robot, el método de Campos Potenciales es particularmente útil para situaciones que requieren reacción en tiempo real a cambios en el entorno. Por ejemplo, un robot que navega a través de un entorno poblado de personas y otros robots puede utilizar campos potenciales para ajustar continuamente su trayectoria y evitar colisiones, manteniendo al mismo tiempo un curso hacia su destino. Este enfoque permite una navegación fluida y adaptable, especialmente en entornos donde los obstáculos pueden cambiar de posición o aparecer de manera inesperada. Además, la implementación de Campos Potenciales es relativamente sencilla y computacionalmente eficiente, lo que lo hace adecuado para sistemas robóticos con recursos limitados. Sin embargo, uno de los desafíos del método de Campos Potenciales es la posibilidad de que el robot quede atrapado en mínimos locales, situaciones donde el robot se encuentra en un punto que es localmente óptimo pero no globalmente, impidiendo que alcance su destino. Diversas estrategias se han propuesto para mitigar este problema, como la introducción de técnicas de escape o la combinación de Campos Potenciales con otros métodos de planificación de trayectorias.

### **3.6.6. Rapidly-exploring Random Trees (RRT)**

Es un algoritmo eficaz para la planificación de trayectorias en espacios con obstáculos y dimensiones altas, especialmente útil en robótica y simulaciones de sistemas dinámicos.

El algoritmo RRT funciona generando de manera incremental un árbol que explora el espacio de búsqueda, comenzando desde el estado inicial del robot. En cada iteración, el algoritmo selecciona un punto aleatorio en el espacio de búsqueda y encuentra el nodo más cercano en el árbol ya construido hacia ese punto. Luego, intenta extender el árbol desde ese nodo hacia el punto aleatorio, dentro de un margen definido, asegurándose de que esta nueva extensión esté libre de obstáculos. Este proceso se repite hasta que el árbol alcanza el estado objetivo o se cumple una condición de terminación, como un número máximo de iteraciones.

El método RRT tiene una amplia gama de aplicaciones, como: Robótica móvil: Para la navegación autónoma en entornos desconocidos o complejos con obstáculos.

Robótica manipuladora: En la planificación de caminos para brazos robóticos que necesitan alcanzar un punto final evitando colisiones.

Simulación de sistemas dinámicos: Para planificar trayectorias en simulaciones que involucren física compleja o vehículos autónomos.

Juegos y efectos visuales: Para generar movimientos realistas de entidades en entornos virtuales.

En el contexto del movimiento autónomo, el RRT permite al robot planificar su trayectoria en tiempo real, adaptándose a entornos desconocidos o que cambian dinámicamente. Al generar un árbol de exploración que se extiende rápidamente hacia áreas no exploradas, el RRT es capaz de encontrar rutas viables de manera eficiente, incluso en espacios de configuración complejos. Por ejemplo, en un vehículo autónomo que navega por una ciudad, el RRT puede utilizarse para planificar rutas seguras alrededor de obstáculos temporales, como obras en la carretera o tráfico denso, ajustando la trayectoria a medida que se obtiene nueva información del entorno. De manera similar, en robótica móvil para exploración espacial o de desastres, el RRT puede guiar a los robots a través de terrenos desconocidos, optimizando la exploración y minimizando el riesgo de quedar atrapados o dañados. La ventaja principal del RRT es su capacidad para manejar espacios de alta dimensionalidad y obstáculos complejos, haciendo que sea especialmente útil para robots que operan en entornos desafiantes. Sin embargo, es importante considerar que, aunque el RRT es eficiente en la exploración del espacio y la generación de trayectorias factibles, las rutas generadas pueden no ser óptimas en términos de longitud o suavidad. Para abordar esto, existen variantes del RRT, como el RRT\*, que buscan mejorar la calidad de las trayectorias encontrando caminos más cortos o más suaves a lo largo del tiempo.

## 4. Desarrollo

### 4.1. Introducción

En la robótica quirúrgica, la generación de trayectorias precisas y suaves es crucial para la ejecución de procedimientos de alta precisión y la reducción de riesgos. Específicamente, la navegación precisa de una aguja quirúrgica a lo largo de una ruta determinada exige algoritmos y herramientas avanzadas.

La generación de trayectorias implica planificar una secuencia de puntos en el espacio tridimensional por los cuales un dispositivo, como una aguja o un instrumento quirúrgico, debe desplazarse. Esta trayectoria puede ser recta, curva o compleja, según las necesidades del procedimiento.

Aplicaciones específicas, como la sutura robótica, presentan desafíos adicionales. En este caso, se requiere un movimiento meticuloso de la aguja para garantizar una colocación adecuada de las suturas y la unión precisa de los tejidos. Se deben considerar variables como la orientación de la aguja, la profundidad de la sutura y la curvatura del tejido.

Para lograrlo, se determina una ecuación que define la trayectoria de la aguja, generando secuencias de matrices de transformación 3D que representan su movimiento, incluyendo posición y orientación en cada paso. El generador de trayectorias también permite rotaciones de la aguja basadas en ángulos definidos por el usuario, lo que facilita suturar en distintos planos y ángulos.

La generación de trayectorias para la sutura robótica es intrincada, y su desarrollo ha impulsado significativamente la precisión y seguridad en cirugías asistidas por robots.

### 4.2. Herramientas de desarrollo

En esta sección se proporcionará una breve descripción de las principales herramientas empleadas en el desarrollo de esta investigación: la plataforma MATLAB, el sistema operativo ROS, la plataforma Unity, el robot UR3 y la pinza EndoWrist. El objetivo es

ofrecer una comprensión clara de cómo cada una de ellas contribuye con un conjunto único de capacidades y funcionalidades en los ámbitos de simulación, manipulación y desarrollo de sistemas robóticos. A lo largo de este segmento, se detalla el papel crucial que desempeña cada una de estas herramientas en el cumplimiento de los objetivos y el éxito de la presente investigación.

### 4.2.1. MATLAB

MATLAB (Laboratorio de Matrices) es una plataforma de programación diseñada específicamente para ingenieros y científicos. Se destaca por su eficiencia en el análisis y diseño de sistemas, así como en la realización de cálculos numéricos y manipulaciones de matrices.

Una de las principales fortalezas de esta plataforma es su interfaz, caracterizada por su sencillez y facilidad de navegación. Sobresale en la presentación de datos, ya sea en formato textual o gráfico, proporcionando una experiencia de usuario óptima.

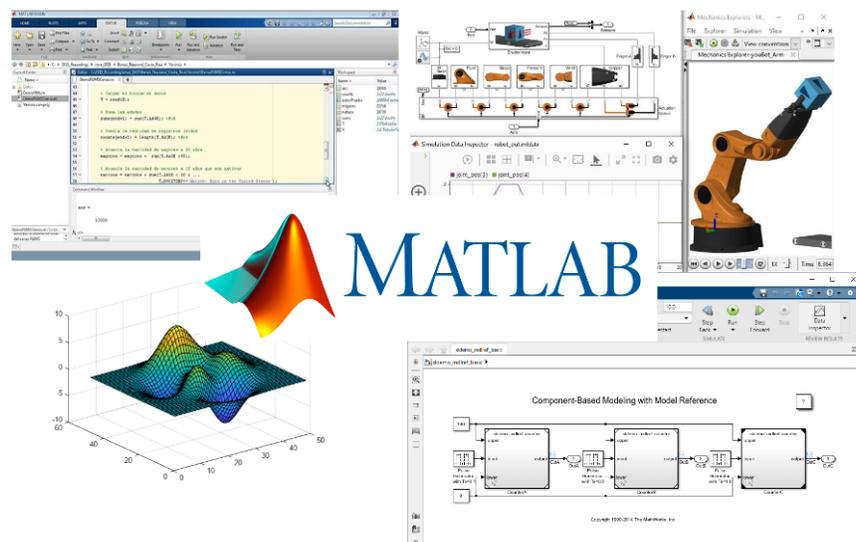


Figura 4.1.: MATLAB [Fuente propia].

Adicionalmente, se complementa con una amplia selección de librerías y comandos diseñados para satisfacer una diversidad de aplicaciones. Entre las funcionalidades más destacadas se encuentran la manipulación de matrices, el manejo de operaciones de álgebra lineal, así como la potente capacidad de visualización y creación de gráficos.

MATLAB tiene su propio lenguaje, fundamentado en el manejo de matrices, lo que permite a los usuarios crear sus propias funciones y algoritmos, ofreciendo una amplia

---

variedad de aplicaciones en múltiples campos, como procesamiento de señales, control y automatización, análisis numérico, métodos iterativos, inteligencia artificial y aprendizaje automático [40].

Teniendo en cuenta la amplia gama de herramientas que ofrece MATLAB y su integración con ROS y UR3, se decide utilizar esta plataforma en el desarrollo del sistema. Esto permitirá crear funciones para establecer los puntos de inserción de la aguja, calcular la trayectoria de la sutura y determinar la cinemática inversa del robot para guiar la aguja en el entorno virtual. Además, se establecen las funciones encargadas de enviar los datos mediante una comunicación a través de ROS. Como se mencionó anteriormente, gracias a la capacidad de MATLAB, este proceso puede abordarse de manera mucho más sencilla en comparación con otras plataformas.

#### 4.2.2. ROS (Robot Operating System)

El Robot Operating System (ROS, por sus siglas en inglés) es una plataforma compuesta por varios frameworks diseñados para el desarrollo de software destinado a robots. Posee la capacidad de proporcionar servicios similares a un sistema operativo convencional, incluyendo la abstracción del hardware, el control de dispositivos de bajo nivel, la implementación de funcionalidades comunes, la comunicación de mensajes entre procesos y la gestión de paquetes.

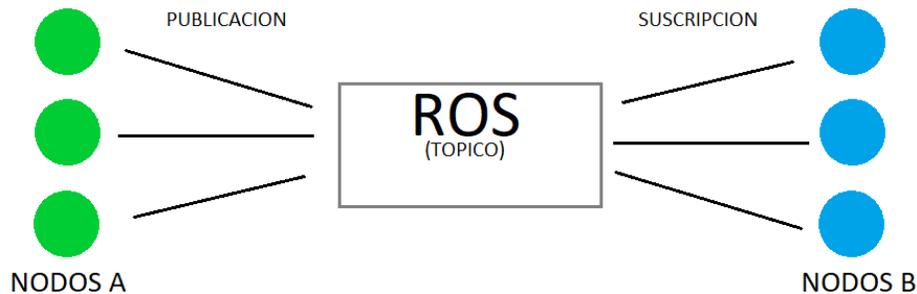


Figura 4.2.: ROS(Robot Operating System) [Fuente propia].

Este sistema se fundamenta en una arquitectura de grafo, donde el procesamiento se realiza en nodos capaces de recibir, enviar y multiplexar mensajes provenientes de sensores, controladores, estados, planificaciones y actuadores, tal como se muestra en la figura anterior [41].

Dadas las funcionalidades ofrecidas por ROS, se establece como el puente de comunicación entre nuestra plataforma de desarrollo (MATLAB) y nuestro simulador

(Unity). Entre estas funcionalidades, se destaca la comunicación mediante tópicos de tipo Cliente/Servidor, la cual será empleada para facilitar la transmisión de posiciones y orientaciones desde los nodos a (MATLAB), hacia los robots en el entorno de simulación representados por los nodos b (ver Figura 4.2).

### 4.2.3. UNITY

Unity es una plataforma de desarrollo centrada principalmente en la creación de videojuegos. Sin embargo, ha demostrado su versatilidad al ser utilizada para desarrollar experiencias interactivas de Realidad Virtual en una variedad de sectores, incluyendo el cine, la animación, la salud, la industria automotriz y la construcción, entre otros.

Una de las principales fortalezas de Unity reside en su capacidad para desarrollar juegos utilizando el lenguaje de programación C++, y adaptarlos a múltiples dispositivos sin necesidad de cambiar de plataforma. Estas características facilitan tanto el desarrollo como la exportación de cualquier proyecto de una manera relativamente sencilla.

Esta plataforma no se limita únicamente a motores para el renderizado de imágenes, físicas en 2D/3D, audio y animaciones. También incluye herramientas de networking para juegos multijugador y herramientas de navegación NavMesh para la implementación de Inteligencia Artificial, así como soporte para Realidad Virtual.

La interfaz de Unity se distingue por su alta intuitividad, lo que facilita su manejo y mejora significativamente la experiencia del usuario. Dentro de la interfaz, se encuentran varios paneles que pueden ser ajustados y reorganizados según las necesidades del usuario [42].



Figura 4.3.: Logo Unity. [43]

Siendo Unity una interfaz altamente intuitiva, utilizada por millones de personas en el desarrollo de diversas aplicaciones, posibilita un aprendizaje rápido gracias a las múltiples herramientas disponibles en la red, como tutoriales, foros, blogs, entre otros. Esto facilita la creación de un simulador que nos permita observar los resultados de la investigación.

---

#### 4.2.4. Robot UR3

El UR3 es un robot industrial colaborativo, ultraligero y compacto, ideal para ensamblajes de precisión en mesas de trabajo. Se trata del modelo más pequeño dentro de la línea de robots UR, con 6 articulaciones giratorias. Universal Robots ha puesto un énfasis particular en la reducción de sus dimensiones y en el diseño de un mecanismo intuitivo, lo que implica que no se necesitan conocimientos previos para su operación.

Entre las principales características de este modelo se encuentran:

- Peso: 11 kg
- Capacidad de carga útil: 3 kg
- Capacidades de rotación de  $\pm 360$  grados en todas las articulaciones
- Rotación infinita en el último eje



Figura 4.4.: Robot UR3. [44]

El UR3 se destaca notablemente en diversas industrias y tipos de trabajos. Algunas de las principales funciones que puede desempeñar incluyen empaquetado y paletizado, lijado y pulido, soldadura continua y atornillado [45].

Considerando el entorno y las actividades previstas para el brazo robótico dentro de este proyecto, donde se necesitan movimientos altamente precisos para manipular cargas inferiores a 1 kg en un entorno con movilidad limitada, las características del brazo cumplen plenamente todos los criterios establecidos para el proyecto.

### 4.2.5. Pinza EndoWrist

La pinza EndoWrist es una herramienta empleada en cirugía mínimamente invasiva, destacándose principalmente en procedimientos asistidos por robots.



Figura 4.5.: Pinza EndoWrist. [46]

Su meticuloso diseño tiene como objetivo emular con alta precisión y destreza los movimientos de la mano del cirujano dentro del cuerpo del paciente. Entre sus características y funcionalidades más destacadas, se encuentran:

- Ofrece siete grados de libertad, lo que significa que puede ejecutar movimientos en siete direcciones distintas, permitiendo al cirujano imitar con gran fidelidad los gestos que haría con su propia mano.
- Puede estar equipada con una punta diseñada para suturar o cortar tejidos, dependiendo del tipo específico de pinza EndoWrist.
- Es completamente compatible con sistemas quirúrgicos robóticos, como el conocido sistema da Vinci, que permite al cirujano llevar a cabo procedimientos de manera remota y con una precisión incrementada.

Esta sofisticada herramienta se aplica en una amplia variedad de procedimientos quirúrgicos, especialmente en el ámbito de la cirugía laparoscópica y la asistida por robot. Es imperativo subrayar que la pinza EndoWrist es un instrumento altamente especializado cuyo manejo requiere capacitación y experiencia por parte del cirujano [47].

### 4.3. Arquitectura

En esta sección, se aborda la arquitectura implementada dentro del proyecto, esta se encuentra compuesta por tres componentes fundamentales: MATLAB, ROS y Unity. Esta configuración se erige como un sistema integrado destinado a la planificación y ejecución de una sutura quirúrgica. A continuación, se detallarán cada uno de estos elementos y su contribución esencial en el proceso.

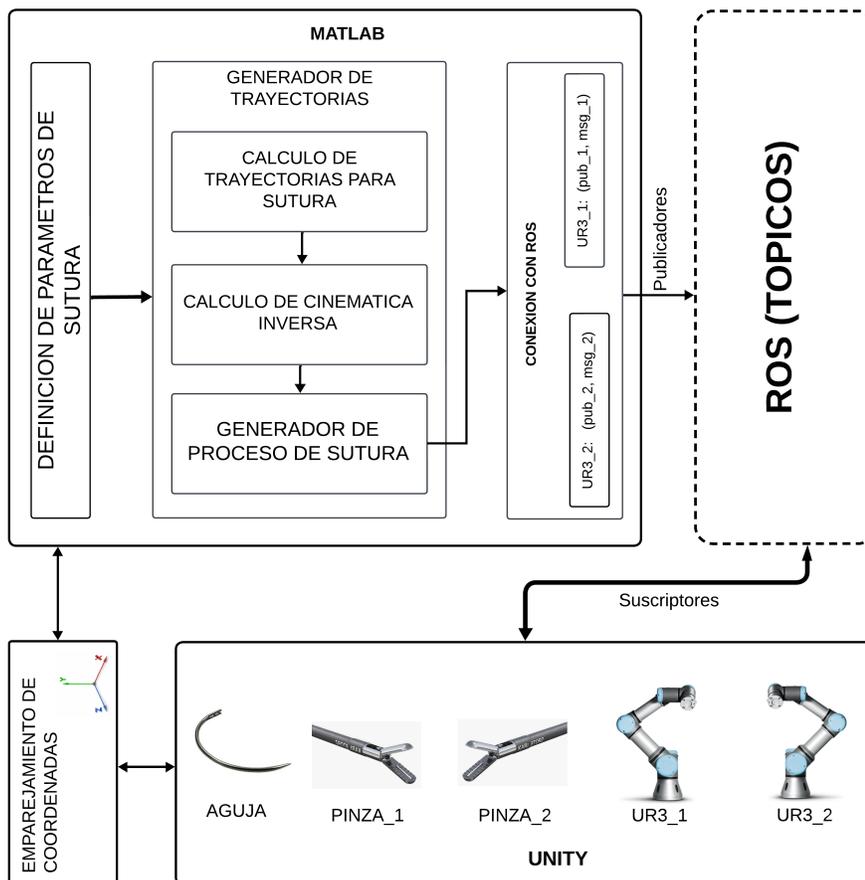


Figura 4.6.: Arquitectura [Fuente propia].

#### 4.3.1. Módulo de desarrollo - MATLAB

Dentro de la figura 4.6 se visualiza el módulo de desarrollo MATLAB, el cual se encuentra dividido en cinco submódulos distintos: definición de parámetros de sutura, cálculo de trayectorias para sutura, cálculo de cinemática inversa, generador de proceso

de sutura y por ultimo, conexión con ROS. Cada uno de los submódulos mencionados, constituye, en conjunto, el sistema generador de trayectorias para sutura. A continuación, se proporcionará una breve descripción de qué consta cada uno de ellos y cómo se interrelacionan.

#### **4.3.1.1. Definición de parámetros de sutura**

El submódulo de definición de parámetros de sutura permite al usuario establecer los parámetros necesarios para llevar a cabo la sutura. Estos incluyen el número de puntadas a realizar, la definición de los puntos de sutura y la dimensión de la aguja.

- Numero de puntadas: Define la cantidad de iteraciones de puntada que se realiza durante el proceso de sutura.
- Definición de puntos de sutura: Establece las coordenadas de los puntos de sutura en sus 3 ejes, para cada uno de los puntos de sutura requeridos.
- - Dimensión de la aguja: Define la dimensión de la aguja con la que se va a desarrollar el proceso.

Todos los parámetros mencionados se definen en relación con el espacio de trabajo del robot, lo que asegura su capacidad para realizar el proceso de sutura dentro de su rango de movimiento. Una vez definidos los parámetros de sutura, se puede iniciar el generador de trayectorias, lo que implica un flujo de información desde este módulo hacia el generador de trayectorias.

#### **4.3.1.2. Generador de trayectorias**

Dentro de este submódulo se tienen tres secciones encargadas de realizar todos los cálculos necesarios que establecen el proceso de sutura. Dentro de ellas se encuentran el cálculo de trayectorias para sutura, el cálculo de cinemática inversa y, por último, el generador del proceso de sutura. A continuación, se explicará el proceso que se desarrolla en cada una de estas secciones.

- Calculo de trayectorias para sutura  
A partir de los puntos de sutura y del radio de la aguja ingresados, se calculan las diversas trayectorias que el sistema debe seguir para llevar a cabo este proceso. Estas trayectorias comprenden los desplazamientos de un punto a otro realizados

---

por los brazos robóticos, así como las acciones relacionadas con la inserción y extracción de la aguja.

- **Cálculo de cinemática inversa**

A partir de las trayectorias obtenidas, se calcula la cinemática inversa con el fin de determinar las configuraciones necesarias para el movimiento de los robots. Finalmente, estos datos se almacenan dentro de las variables asociadas a cada uno de ellos. Todos los datos generados dependen completamente de la posición de los puntos, razón por la cual existe una relación directa entre este apartado y el anterior.

- **Generador del proceso de sutura**

En esta sección, se sistematizan las diversas fases del procedimiento de sutura, que incluyen acciones como la inserción, extracción y desplazamientos. Esto facilita la coordinación de cada uno de los brazos robóticos a lo largo del proceso, evitando interferencias entre ellos y asegurando una ejecución fluida a medida que avanza el procedimiento. Se garantiza el cumplimiento de cada tarea hasta la finalización de la sutura.

### **4.3.2. Módulo de desarrollo - ROS**

El módulo de ROS (Robot Operating System) sirve como un puente para facilitar la comunicación entre MATLAB y Unity. Gracias a ROS, se establece un flujo constante de datos que incluye posiciones, orientaciones e interacciones para los distintos objetos dentro de Unity. Cada uno de estos componentes publica en un tópico específico, como se muestra en la figura 4.6, dirigiendo toda la información que fluye a través de los diferentes canales a los suscriptores asociados.

### **4.3.3. Módulo de desarrollo - Unity**

Por último, el módulo de Unity desempeña un papel vital como receptor y procesador de la información generada por MATLAB. En Unity, se ha desarrollado un entorno de simulación que recrea un escenario quirúrgico, donde los principales elementos son los dos robots UR3e, las dos pinzas EndoWrist y sus respectivas agujas. Unity es responsable de llevar a cabo la simulación de las trayectorias generadas por MATLAB y de visualizar el proceso de sutura en tiempo real, proporcionando así una simulación completa y fluida dentro del entorno.

## 4.4. Implementación en MATLAB

### 4.4.1. Conectividad MATLAB - ROS.

En esta sección, se aborda uno de los principales componentes del desarrollo de este proyecto: la conexión entre MATLAB y Unity a través de ROS. Esta conexión posibilita la implementación efectiva de los componentes desarrollados en el entorno de simulación, lo que permite manipular cada uno de ellos con precisión. A continuación, se presenta una gráfica y se describe el proceso de creación y configuración de esta conexión.

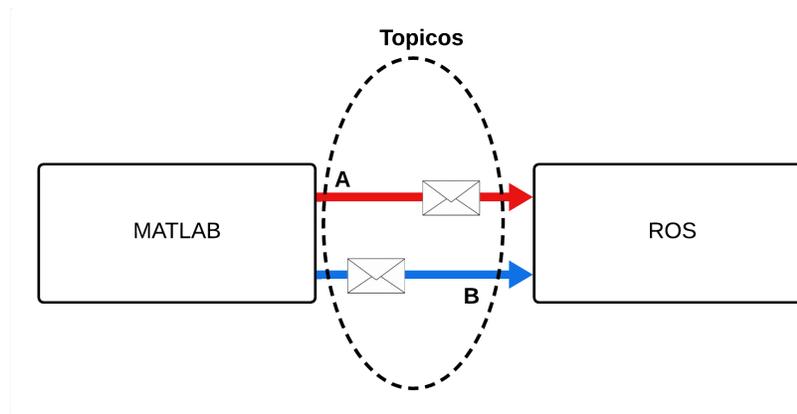


Figura 4.7.: Conectividad Matlab - Ros [Fuente propia].

- Creación del Canal de Comunicación

Para establecer una comunicación efectiva entre MATLAB y ROS, es indispensable crear canales de comunicación que faciliten el flujo de datos en tiempo real para distintos componentes, como el estado de las articulaciones del robot u otros aspectos de simulación. Cada canal posee un identificador único que permite la dirección precisa y rápida de los mensajes que se deseen enviar.

- Preparación del Mensaje

Dado que ROS se basa en la comunicación a través de mensajes, es esencial elaborar mensajes que contengan la información vital para cada uno de los componentes dentro del entorno de simulación. Por lo tanto, estos mensajes se estructuran de manera que puedan incluir una marca de tiempo, un marco de referencia y una lista detallada de identificadores para las articulaciones, posiciones, velocidades o propiedades de los objetos dentro del entorno. Esto asegura que la información relevante sea transmitida de manera efectiva a través del mensaje.

#### 4.4.2. Carga y configuración del modelo del brazo robot UR3

El modelo del robot UR3 desempeña un papel crucial en el proceso de simulación y manipulación del sistema robótico dentro de esta investigación. A continuación, se explicará el procedimiento empleado para su integración y configuración.

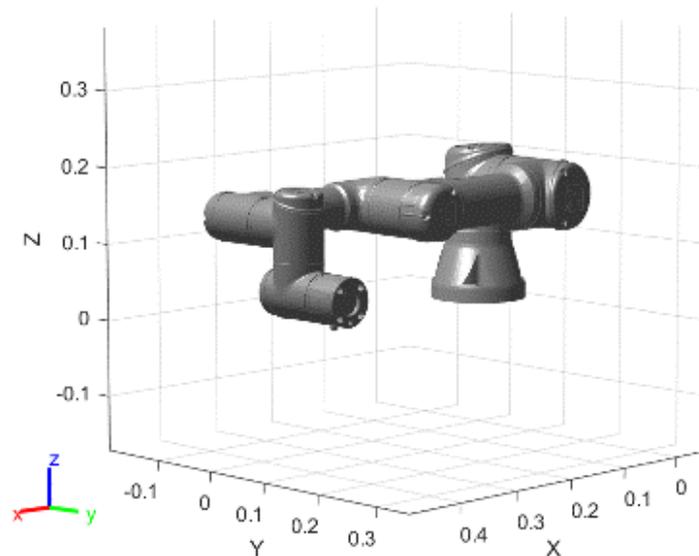


Figura 4.8.: Modelo brazo robot UR3 [Fuente propia].

##### ■ Carga del modelo del robot UR3

Este modelo es esencial para simular y manipular el comportamiento del robot. Por tanto, resulta indispensable cargar el modelo del robot UR3 desde MATLAB. Este modelo constituye una representación virtual del sistema físico. Para lograrlo, se utiliza la función 'loadrobot', la cual permite cargar el modelo del robot UR3 desde la biblioteca de MATLAB. Esto proporciona información detallada sobre el modelo, incluyendo configuraciones de las articulaciones, límites de movimiento, masas de los eslabones, entre otros aspectos. Además, se pueden especificar opciones adicionales, como la gravedad.

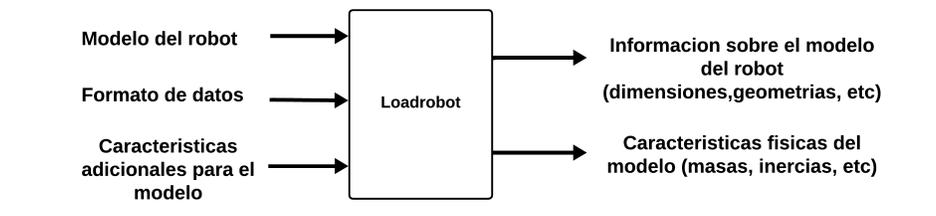


Figura 4.9.: Función 'loadrobot' [Fuente propia].

- Configuración de la transformación espacial

Con el fin de asegurar el comportamiento correcto del robot en su entorno de trabajo, se aplica una transformación espacial a la representación del robot UR3 para modificar su orientación en el espacio y así establecer una transformación fija en el componente del robot correspondiente a su base.

#### 4.4.3. Configuración del objeto para cinemática inversa

La cinemática inversa (IK) es esencial para calcular los ángulos de las articulaciones de un brazo robótico, necesarios para alcanzar una posición y orientación específica en el extremo de agarre, conocido como efector final. Este proceso se logra mediante el uso de ecuaciones cinemáticas.

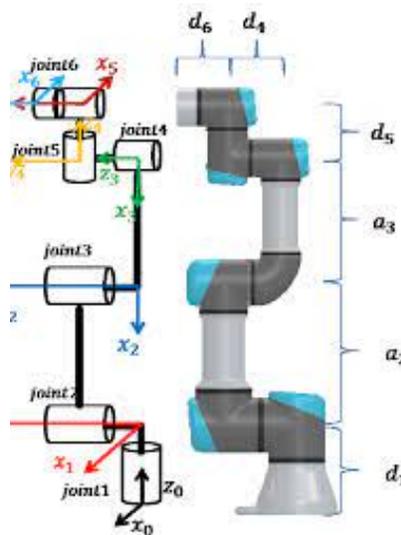


Figura 4.10.: Articulaciones UR3e

UR3e							
Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]	Dynamics	Mass [kg]	Center of Mass [m]
Joint 1	0	0	0.15185	$\pi/2$	Link 1	1.98	[0, -0.02, 0]
Joint 2	0	-0.24355	0	0	Link 2	3.4445	[0.13, 0, 0.1157]
Joint 3	0	-0.2132	0	0	Link 3	1.437	[0.05, 0, 0.0238]
Joint 4	0	0	0.13105	$\pi/2$	Link 4	0.871	[0, 0, 0.01]
Joint 5	0	0	0.08535	$-\pi/2$	Link 5	0.805	[0, 0, 0.01]
Joint 6	0	0	0.0921	0	Link 6	0.261	[0, 0, -0.02]

Figura 4.11.: Tabla de parámetros denavit hartenberg del robot UR3. [48]

---

La configuración del robot se refiere a una serie de posiciones articulares que se encuentran dentro de los límites de posición del modelo del robot y no violan ninguna restricción impuesta por el mismo. La cinemática inversa determina una configuración articular adecuada para que el efector final se mueva a la posición objetivo. A continuación, se detalla el procedimiento utilizado para esta configuración:

- Creación del objeto para cinemática inversa

Para la creación de este objeto, se utiliza la función ‘inverseKinematics’ la cual crea un solucionador cinemático inverso que permite encontrar una configuración conjunta que logra la pose específica del efector final. Para esto, es necesario proporcionar el modelo del robot UR3, donde se almacenan sus características.

A partir del objeto mencionado anteriormente, es posible realizar operaciones de cinemática inversa en un sistema robótico UR3. Para ello, se requiere especificar el punto de referencia en el extremo del robot, la matriz de transformación que define la posición y orientación deseada, los pesos asociados a cada componente, y la configuración inicial o posición de inicio del robot antes de realizar el cálculo de la cinemática inversa. De esta manera, se logra obtener una matriz que contiene los ángulos de cada articulación que logran la configuración deseada del extremo efectivo. Dado que el robot UR3 tiene 6 grados de libertad, la dimensión de esta matriz será de  $1 \times 6$ , lo que permite posicionar el efector del brazo del robot sobre el punto indicado.

#### **4.4.4. Definición de posiciones y transformaciones**

A continuación, se describe el procedimiento para establecer las configuraciones iniciales del brazo robot, con el objetivo de definir el estado y la orientación inicial del robot antes de su operación.

- Definición de posiciones iniciales

Con el propósito de determinar la posición inicial de cada brazo robótico UR3, es necesario crear un vector que defina los ángulos de las articulaciones en grados para cada uno de los brazos robóticos. Simultáneamente, se establecen los valores de esfuerzo en las posiciones iniciales de los robots, asegurando así la seguridad, eficiencia y precisión operativa.

Todos los datos se asignan a un mensaje, que una vez configurado, se transmitirá a través del canal preestablecido, lo que permitirá llevar a cada robot a su posición inicial.

- Obtención de la transformación inicial

Teniendo en cuenta las posiciones iniciales, se obtiene la descripción de la configuración del robot, representada como una matriz de transformación, a partir de esta se extraen las componentes de posición para representar la configuración inicial y final en el espacio de trabajo.

#### 4.4.5. Función de trayectoria de inserción y extracción de aguja

La función para generar una trayectoria circular se basa en una combinación de cálculos geométricos, conversiones de coordenadas y técnicas de interpolación lineal para establecer una trayectoria en forma de círculo para el movimiento de la aguja. A partir de los puntos de inicio (p1), finalización de puntada (p2) y el radio de la sutura, se calcula un tercer punto en el eje z (p3), con los cuales se crea la trayectoria de puntada y se determina la profundidad adecuada para la aguja. Estos cálculos aseguran que la aguja se posicione en el punto óptimo para realizar la sutura de manera efectiva.

De esta manera la función genera una circunferencia en un espacio 3D como se observa en la figura 4.12.

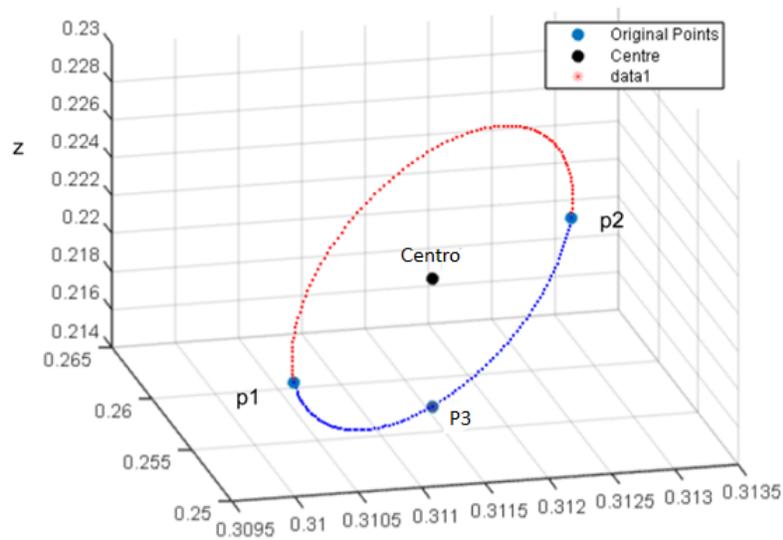


Figura 4.12.: Trayectoria de inserción y extracción de aguja[Fuente propia].

#### 4.4.5.1. Desplazamiento de aguja quirúrgica en 3D

A partir de la mencionada trayectoria previamente, se origina un recorrido tridimensional en forma de arco que define el desplazamiento planeado de la aguja quirúrgica; este arco se materializa como una secuencia de matrices de transformación tridimensional de cuatro por cuatro, las cuales encapsulan tanto la posición como la orientación de la aguja en cada punto a lo largo de la trayectoria.

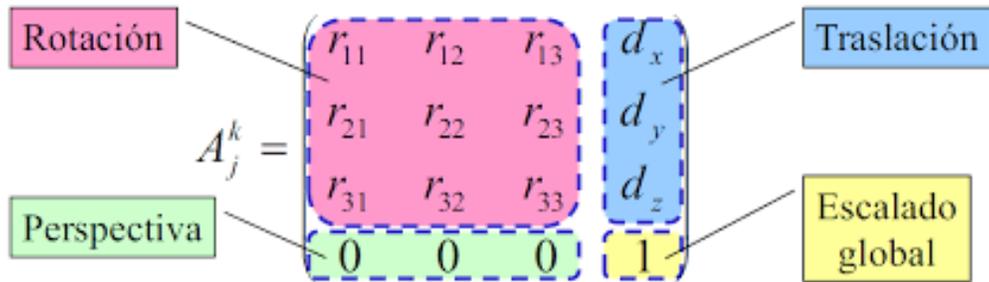


Figura 4.13.: Matriz de Posiciones y Orientaciones.

Dichas matrices de transformación albergan información detallada sobre la configuración espacial de la aguja, lo que permite obtener la trayectoria de color azul mostrada en la figura 4.14,. Este proceso se aplica tanto para la inserción como para la extracción de la aguja. Sin embargo, estos procesos difieren tanto en el punto de inicio y finalización de la trayectoria como en el punto de sujeción de la aguja.

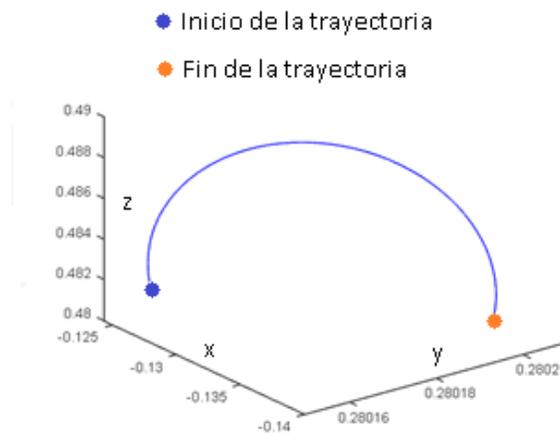


Figura 4.14.: Desplazamiento de pinza con aguja quirúrgica [Fuente propia].

Debido a que la trayectoria desarrollada no solo implica un cambio de posición, sino también una variación en la orientación, la figura 4.15, ilustra la trayectoria seguida por la aguja. En esta representación, se destacan los diversos cambios en posición y orientación que ocurren a lo largo de sus ejes, dando lugar a una combinación de movimientos. Estos movimientos se configuran como un arco, necesario para que la pinza pueda introducir y extraer la aguja desde distintos puntos de sutura. Dentro del marco del algoritmo implementado, se genera una lista compuesta por 100 matrices homogéneas de dimensiones 4x4. Estas matrices describen de manera eficiente la posición y orientación en cada instante, lo que permite ejecutar de manera precisa el movimiento requerido para llevar a cabo el proceso de sutura.

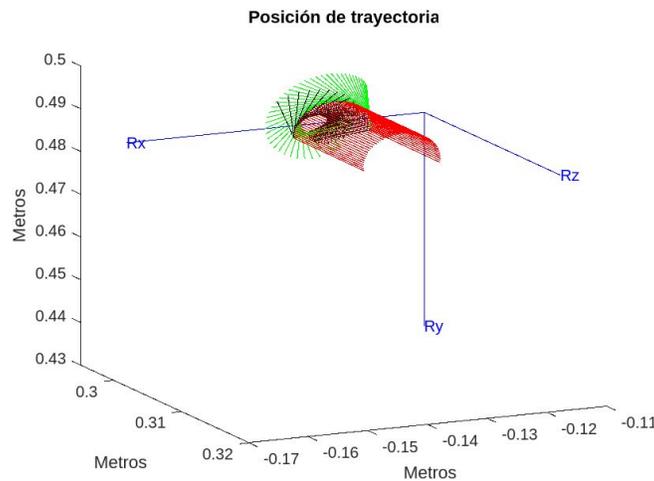


Figura 4.15.: Posición de trayectorias en sus ejes [Fuente propia].

#### 4.4.6. Planificador con algoritmo RRT

El planificador desarrollado es capaz de generar una trayectoria para el desplazamiento desde un punto A hasta un punto B. Durante este desplazamiento, si se identifica un obstáculo, la trayectoria se ajustará para permitir que el robot lo evite. Para lograr esto, se llevan a cabo una serie de pasos, que incluyen cargar el robot, cargar el objeto (en este caso, la pinza), definir la colisión y el objeto de colisión, que en este caso se representa como una esfera, como se ilustra en la figura 4.16.

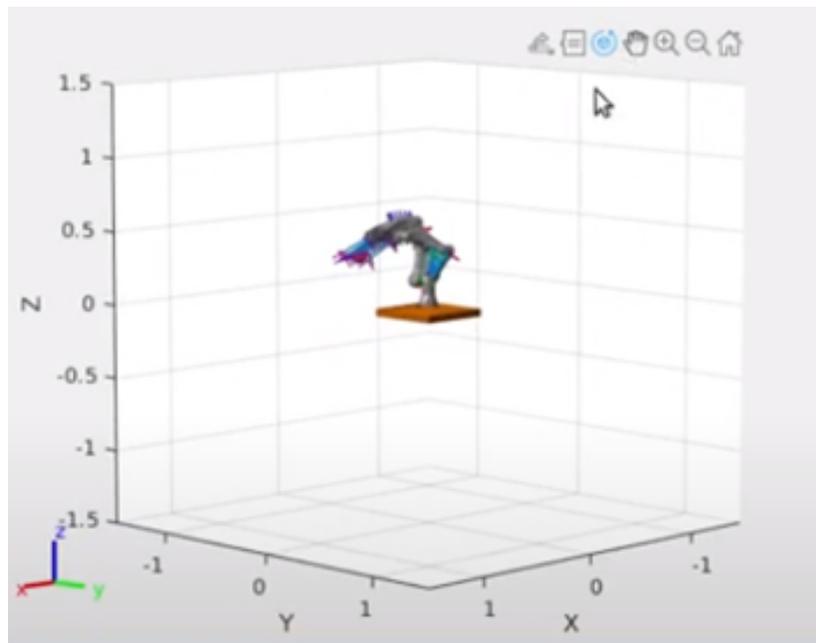


Figura 4.16.: Inicialización de robot UR3 y objeto de colisión [Fuente propia].

En la siguiente figura 4.17 se observa desde dos ángulos diferentes la trayectoria que realiza el robot, evadiendo el objeto de colisión.

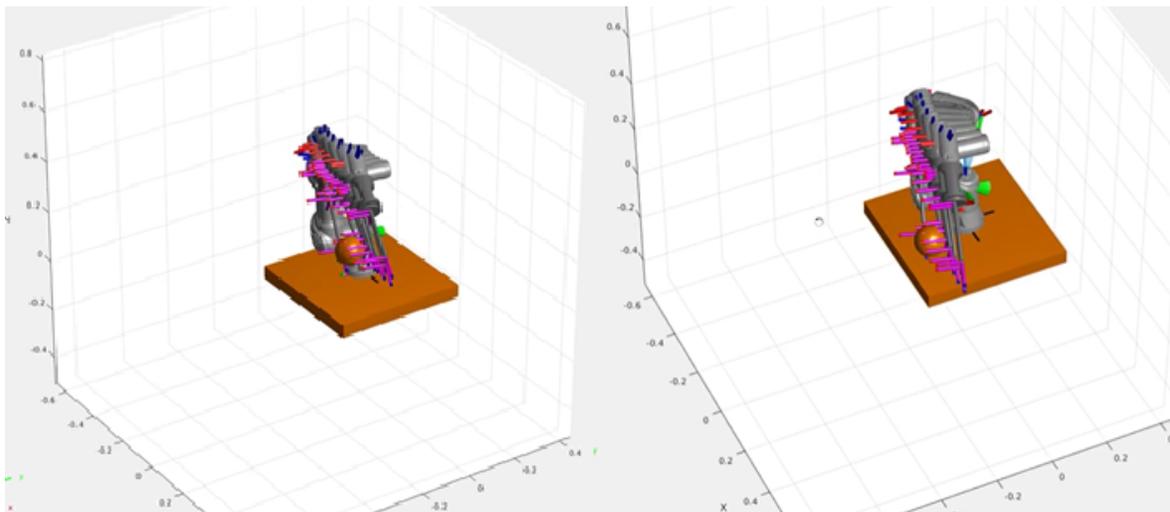


Figura 4.17.: Trayectoria con evasión de obstáculo [Fuente propia].

En la figura 4.18 se observa de manera más clara la trayectoria que sigue el brazo

robótico al desplazarse desde el punto A hasta el punto B, evitando el objeto de colisión.

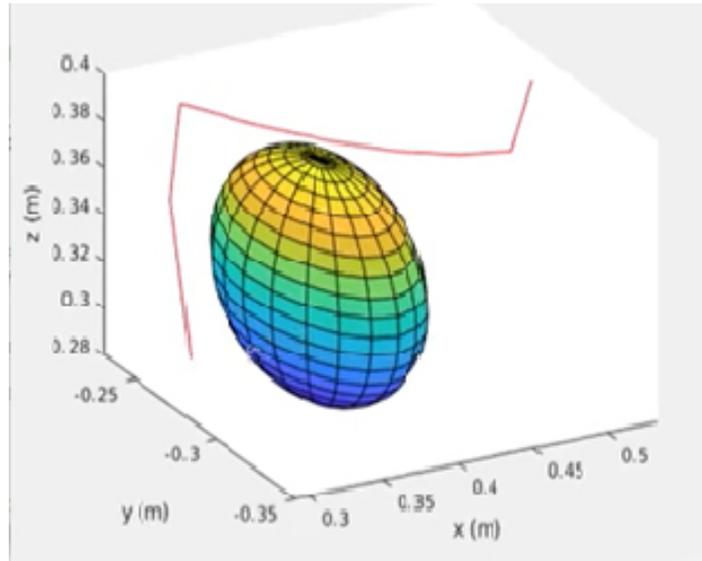


Figura 4.18.: Trayectoria generada para evadir obstáculo [Fuente propia].

De esta manera, se puede observar cómo el planificador es capaz de generar una trayectoria que, al mismo tiempo, evita obstáculos.

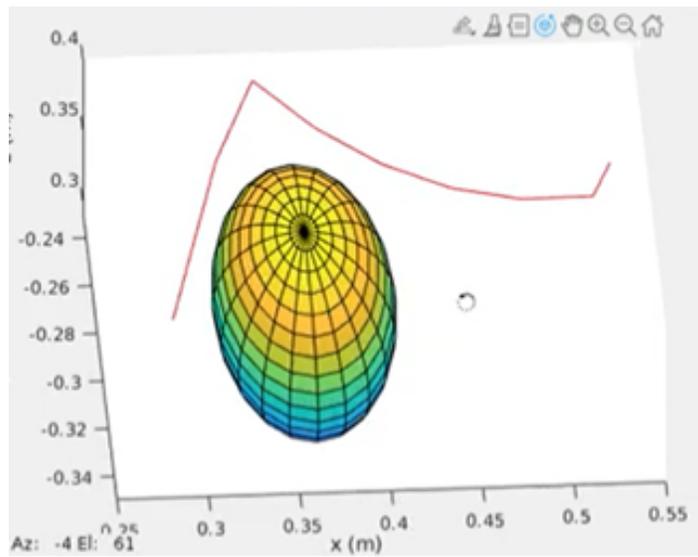


Figura 4.19.: Trayectoria generada para evadir obstáculo - Angulo 2 [Fuente propia].

En la siguiente figura se observa cómo el planificador genera una trayectoria desde

---

un punto hasta otro punto, donde el objeto de colisión se encuentra fuera de dicha trayectoria.

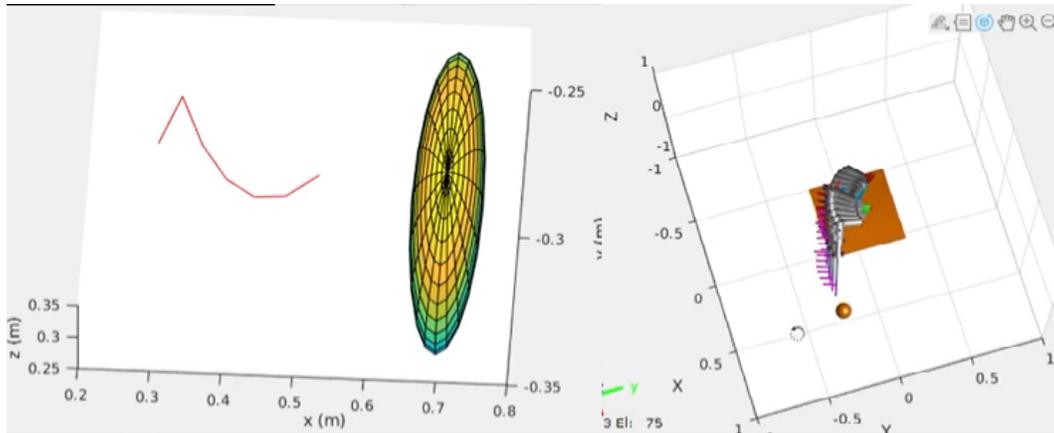


Figura 4.20.: Trayectoria generada sin objeto de colisión [Fuente propia].

#### 4.4.7. Funciones implementadas

Esta sección ofrece una breve explicación de las diversas funciones empleadas en la implementación del sistema generador de trayectorias. Se abordan los distintos componentes y parámetros que podrían conformar estas funciones, así como su aplicación en un contexto general. Así mismo, se explica cómo se integraron en el sistema para asegurar un rendimiento óptimo y coherente en la generación de trayectorias.

CLASE	PARAMETROS	DESCRIPCIÓN
<b>rosinit</b>	<p><b>Opcional:</b></p> <ul style="list-style-type: none"> <li>• (hostname)</li> <li>• (hostname, port)</li> <li>• (URI)</li> </ul>	<p>Permite inicializar y establecer la conexión entre el entorno de ROS y MATLAB, además, configura los parámetros y ajustes esenciales para establecer una comunicación eficaz con los nodos ROS en la red. [49]</p> <p><b>Usado:</b></p> <ul style="list-style-type: none"> <li>• rosinit</li> </ul>
<b>rospublisher</b>	<p><b>Opcional:</b></p> <ul style="list-style-type: none"> <li>• (topicname)</li> <li>• (topicname, msgtype)</li> <li>• (___, Name, Value)</li> <li>• (___, “DataFormat”, “struct”)</li> <li>• (node, topicname)</li> <li>• (node, topicname,type)</li> <li>• (___, “IsLatching”, value)</li> <li>• (___, “DataFormat”, “struct”)</li> </ul>	<p>Se usa para establecer una comunicación bidireccional con el sistema de comunicaciones de ROS. Permite crear un editor de ROS para enviar mensajes a través de una red ROS. [50]</p> <p><b>Usado:</b></p> <ul style="list-style-type: none"> <li>• rospublisher ( ' \joint_states', 'sensor_msgs \JointState');</li> </ul> <p><b>joint_states:</b> Es el nombre del tópico en el que se publicaran las posiciones articulares de cada robot</p> <p><b>sensor_msgs\JointState:</b> Define el tipo de mensaje que se publicará “sensor_msgs” y en que tópico “joint_states”</p>

Tabla 4.1.: Funciones implementadas - Parte 1

CLASE	PARAMETROS	DESCRIPCIÓN
<b>rosmessage</b>	<p><b>Opcional:</b></p> <ul style="list-style-type: none"> <li>• (messagetype)</li> <li>• (pub)</li> <li>• (sub)</li> <li>• (client)</li> <li>• (server)</li> <li>•(---, "DataFormat", "struct")</li> </ul>	<p>Permite la creación y configuración de mensajes ROS, que son las unidades de datos utilizadas para la transmisión de información entre nodos en un entorno ROS. Al utilizar rosmessage, los usuarios pueden definir y manipular estructuras de datos específicas del sistema ROS, lo que facilita la interacción y coordinación entre los diferentes componentes de un Sistema. [51]</p> <p><b>Usado:</b></p> <ul style="list-style-type: none"> <li>• rosmessage(pub)</li> </ul> <p><b>pub:</b> Establece al editor de ROS como un publicador.</p>
<b>loadrobot</b>	<p><b>Opcional:</b></p> <ul style="list-style-type: none"> <li>• (robotname)</li> <li>•(robotname,Name, Value)</li> </ul>	<p>Permite carga el modelo cinemático del robot junto con sus características físicas y geométricas, ya sean modelos predefinidos de la biblioteca de MATLAB o personalizados. [52]</p> <p><b>Usado:</b></p> <ul style="list-style-type: none"> <li>• loadrobot('universal UR3', 'DataFormat','row', 'Gravity',[0 0 -9.81]);</li> </ul> <p><b>universalUR3:</b> Es el nombre del modelo que se desea cargar.</p> <p><b>DataFormat,row:</b> Especifica que los datos de configuración del robot se almacenen en una fila en lugar de una columna.</p> <p><b>Gravity, [0 0 -9.81]:</b> Establece la gravedad aplicada al robot para que sea <math>-9,81 \text{ m/s}^2</math> en la dirección del eje <math>z</math>.</p>

Tabla 4.2.: Funciones implementadas - Parte 2

CLASE	PARAMETROS	DESCRIPCIÓN
<b>Inverse Kinematics (Creacion)</b>	<p><b>Opcional:</b></p> <ul style="list-style-type: none"> <li>• (Name, Value)</li> </ul>	<p>Se utiliza para configurar un objeto que puede calcular soluciones de cinemática inversa para un robot, entregando finalmente las configuraciones de las articulaciones necesarias para colocar el efector final en una posición y orientación específicas. [53]</p> <p><b>Usado:</b></p> <pre>inverseKinematics('RigidBodyTree', UR3);</pre> <p><b>RigidBodyTree:</b> Es una representación de la conectividad de cuerpos rígidos con articulaciones.</p> <p><b>UR3:</b> Es el objeto del modelo del robot UR3, el cual contiene la descripción del robot, sus enlaces, articulaciones y cómo están conectados.</p>
<b>Inverse Kinematics (Uso)</b>	<p><b>Opcional:</b></p> <ul style="list-style-type: none"> <li>• (endeffector, pose, weights, initialguess)</li> </ul>	<p>Se usa para calcular la configuración de las articulaciones de un robot donde el efector final encuentre las posiciones y orientaciones específicas, almacenando los resultados para su posterior uso. [53]</p> <p><b>Usado:</b></p> <pre>ik.UR3('tool0', Tp, weights, Home);</pre> <p><b>tool0:</b> Se utiliza para definir como referencia la herramienta en el extremo del robot.</p> <p><b>Tp:</b> Representa la matriz de transformación del efector final del robot con una posición y orientación actualizadas.</p> <p><b>Weights:</b> Establece los valores de tolerancia que serán utilizados, los cuales determinan qué tan flexible o estricto será el sistema.</p> <p><b>Home:</b> Es una lista de configuraciones de articulaciones del robot expresadas en radianes.</p>

Tabla 4.3.: Funciones implementadas - Parte 3

---

CLASE	PARAMETROS	DESCRIPCIÓN
<b>getTransform</b>	<p><b>Opcional:</b></p> <ul style="list-style-type: none"> <li>•(robot,configuration, bodyname)</li> <li>•(robot,configuration, source-body,targetbody)</li> </ul>	<p>Se utiliza para obtener la transformación entre dos marcos de referencia en el espacio tridimensional, obteniendo la posición y orientación relativa entre dos objetos o sistemas. [54]</p> <p><b>Usado:</b></p> <ul style="list-style-type: none"> <li>• getTransform(UR3,Home,"tool0");</li> </ul> <p><b>UR3:</b> Este es el objeto del modelo del robot UR3 que se proporciona a la función getTransform.</p> <p><b>Home:</b> Corresponde a la configuración específica para el modelo.</p> <p><b>tool0:</b> Se utiliza para definir como referencia la herramienta ensamblada en el extremo del robot.</p>

Tabla 4.4.: Funciones implementadas - Parte 4

## 5. Resultados

En este capítulo, se exponen los resultados derivados del desarrollo de las diversas actividades detalladas en el capítulo anterior, las cuales fueron implementadas con el objetivo de concebir un sistema de trayectorias destinadas a la ejecución de suturas de tipo continuo.

Además, se llevó a cabo un análisis de cada una de las trayectorias ejecutadas por los diferentes componentes involucrados en el proceso de sutura. Este análisis se realiza con el propósito de evaluar la capacidad de adaptación del sistema al ejecutar suturas, considerando que las heridas pueden presentar diversas formas y ubicaciones en relación con el cuerpo humano. Este enfoque técnico busca proporcionar una comprensión detallada de la aptitud del sistema para abordar situaciones heterogéneas en el contexto de procedimientos de sutura.

### 5.1. Trayectoria

El procedimiento de sutura implica la aplicación de una serie de puntos que se adaptan meticulosamente a la configuración de la herida. Para llevar a cabo este proceso, se utilizan dos robots UR3e que, al trabajar en conjunto, garantizan un resultado óptimo en cuanto a eficacia y precisión.

Con el fin de proporcionar una comprensión clara del proceso, se ha desarrollado un diagrama de estados que ilustra los pasos esenciales que deben seguirse para llevar a cabo exitosamente la trayectoria establecida. Este diagrama no solo sirve como una guía visual, sino que también constituye un recurso útil para entender la secuencia de operaciones y la interrelación entre las diferentes etapas del procedimiento de sutura.

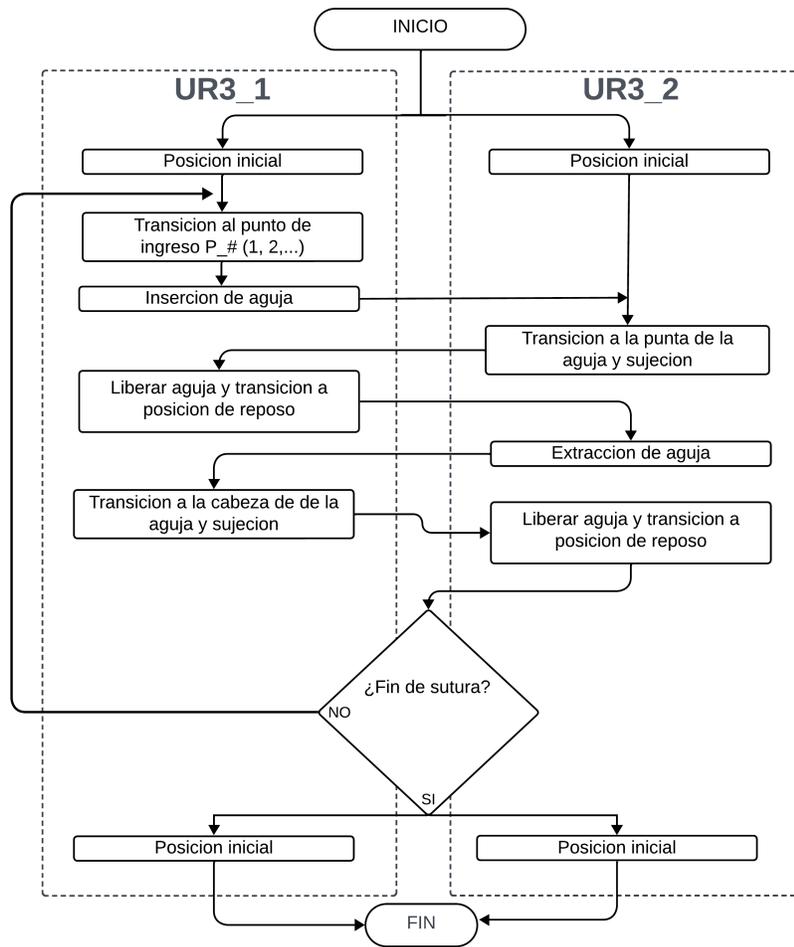


Figura 5.1.: Diagrama de Estados [Fuente propia].

Como se puede observar en la figura 5.1, la sutura es un proceso complejo que requiere la coordinación precisa de los dos brazos robóticos UR3. Al iniciar, ambos se posicionan en su lugar inicial. El brazo robot UR3\_1, junto con la aguja, se desplaza hacia la ubicación del primer punto de sutura, donde realiza la inserción de la aguja. Una vez completada esta etapa, el brazo robot UR3\_2 se dirige a la posición de la punta de la aguja y la sujeta. Una vez sujeta, el brazo robot UR3\_1 la libera y vuelve a su posición de reposo, permitiendo que UR3\_2 realice la extracción de la aguja. Una vez extraída, UR3\_1 se desplaza a la posición de la cabeza de la aguja y la sujeta nuevamente, para que UR3\_2 la libere y regrese a su posición de reposo, completando así la primera puntada, que corresponde al primer par de puntos ingresados.

Como se mencionó anteriormente, dentro de los parámetros de la sutura se establece la cantidad de puntadas a realizar, por lo que el proceso se repetirá hasta alcanzar la

cantidad establecida. Una vez completadas, los brazos robóticos UR3\_1 y UR3\_2 regresan a su posición inicial, finalizando así el proceso de sutura.

Dado el conjunto de acciones desempeñadas por cada brazo robótico y sus interacciones durante el proceso, fue imperativo definir zonas específicas de operación para cada uno, con el fin de mitigar posibles colisiones o interferencias durante la ejecución de las tareas asignadas. A continuación, se detallan las áreas de operación individuales de cada robot.

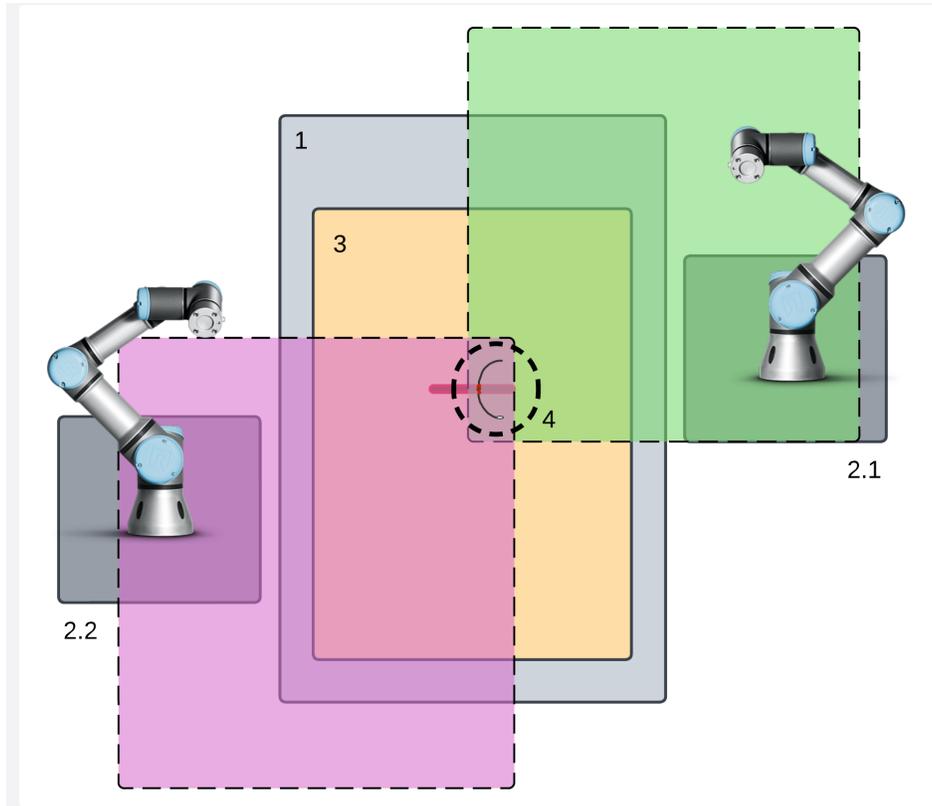


Figura 5.2.: Áreas de operación UR3\_1, UR3\_2 [Fuente propia].

Dentro de la figura 5.2 encontramos el numeral 1 el cual corresponde al área donde se ubicaría un paciente (camilla), el numeral 2.1 y 2.2 representan el posicionamiento de los brazos robot UR3\_1 y UR3\_2 respectivamente, el numeral 3 corresponde al área donde podríamos encontrar una herida y por último tenemos el numeral 4 que corresponde al área circular punteada, la cual se definirá como el área de operación.

Teniendo en cuenta las diferentes áreas mencionadas, se plantean 2 áreas más, una de color morado y otra de color verde, las cuales se encuentran en direcciones opuestas, estas áreas corresponden a la segmentación que se ha realizado para los diferentes

movimientos de cada uno de los brazos robot, con lo cual se buscó evitar colisiones entre los mismos, además de esto se planteó una segmentación más, la cual corresponde al numeral 4, zona en la que no se ejecuta ningún proceso de inserción o extracción de la aguja mientras estén los dos brazos robot en ésta, por lo que siempre se retira a uno de los brazos robot fuera de esta área, para que así el otro pueda ejecutar su actividad.

### 5.1.1. Trayectorias brazo robot UR3\_1

En esta sección se expone el conjunto de trayectorias que encapsula las acciones empleadas por el brazo robótico UR3\_1 y su pinza correspondiente, ubicada en la última articulación del mismo. Es importante destacar que las distintas trayectorias presentadas en la figura 5.3 no son ejecutadas de manera continua, sino que se despliegan de forma secuencial en concordancia con las ejecutadas por el brazo robótico UR3\_2.

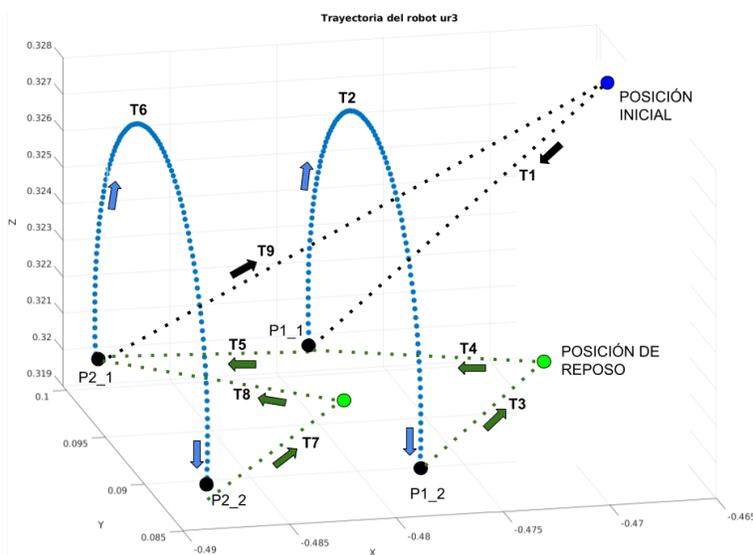


Figura 5.3.: Trayectorias robot UR3 y pinza EndoWrist 1 [Fuente propia].

A continuación se presenta una descripción de cada una de las trayectorias mostradas en la figura 5.3, en concordancia con las acciones que ejecutan dentro del proceso de sutura.

- **Trayectoria T1: Posicionamiento inicial**

La trayectoria T1 del brazo robótico UR3\_1 describe la translación desde la posición inicial hasta el primer punto de sutura P1\_1.

- **Trayectoria T2: Inserción de aguja – Primera puntada**

Una vez que el brazo robótico UR3\_1 se ha posicionado sobre el punto de sutura P1\_1, ejecuta la trayectoria T2, la cual representa una ruta semicircular. Esta trayectoria resulta en la inserción de la aguja, durante este proceso, la aguja penetra la superficie de la herida desde el extremo de la herida P1\_2, y a medida que la inserción se completa, la punta de la aguja emerge por el extremo opuesto P1\_1. Tras la inserción, el brazo robótico UR3\_1 se sitúa en el punto P1\_2 en espera, para que el brazo robótico UR3\_2 sujete la aguja.

- **Trayectoria T3: Posición de reposo – Primera puntada**

Una vez la aguja ha sido sujeta por el brazo robot UR3\_2, el brazo robot UR3\_1 ejecuta la trayectoria T3 con el fin de apartarse del área de operación para poder realizar la extracción de la aguja.

- **Trayectoria T4: Retorno para sujeción de aguja – Primera puntada**

Al concluir el proceso de extracción de la aguja, el brazo robótico UR3\_1 sigue la trayectoria T4, diseñada para retomar la aguja y avanzar en el proceso de sutura. Durante este desplazamiento, el brazo se mueve desde el punto de reposo hacia la ubicación de la cabeza de la aguja, donde la recupera.

- **Trayectoria T5: Posicionamiento - Segunda puntada**

Una vez la aguja es sujeta, se ejecuta la trayectoria T5, donde se dirige el brazo robótico UR3\_1 desde el punto de sujeción hacia el segundo punto de sutura P2\_1.

- **Trayectoria T6: Inserción de aguja – Segunda puntada**

Al estar ubicado sobre el segundo punto de sutura P2\_1, el brazo robótico UR3\_1 repite el proceso de inserción de aguja mediante la trayectoria T6. De manera análoga a la puntada anterior, la aguja ingresa por el punto P2\_2 y al completar la trayectoria, la punta emerge por el punto P2\_1. La posición final del brazo robótico tras esta acción es el punto P2\_2, donde nuevamente libera la aguja para que el brazo robótico UR3\_2 la sujete.

- **Trayectoria T7: Posición de reposo – Segunda puntada**

Una vez la aguja ha sido sujeta por el brazo robot UR3.2, el brazo robot UR3.1 ejecuta la trayectoria T7, para nuevamente apartarse del área de operación y ejecutar el proceso de extracción de la aguja.

- **Trayectoria T8: Retorno para sujeción de aguja – Segunda puntada**

Al concluir el proceso de extracción de la aguja, el brazo robot UR3.1 ejecuta la trayectoria T8, donde el brazo robot se desplaza desde la posición P2.2 hacia la cabeza de la aguja para posteriormente sujetarla.

- **Trayectoria T9: Retorno a la posición inicial.**

Considerando que el proceso de sutura desarrollado en este caso particular, cuenta con dos puntadas y dado que el robot UR3.1 ha cumplido a cabalidad con la ejecución de sus tareas, este regresa a partir del punto P2.2 a su posición inicial, ejecutando la trayectoria T9.

### 5.1.2. Trayectorias brazo robot UR3\_2

Las distintas operaciones realizadas por el brazo robótico UR3.2, junto con su herramienta de pinza, se visualizan como trayectorias en la figura 5.4, como se ha mencionado previamente, estas operaciones se ejecutan de manera coordinada con las trayectorias llevadas a cabo por el brazo robótico UR3.1, tal como se muestra a continuación.

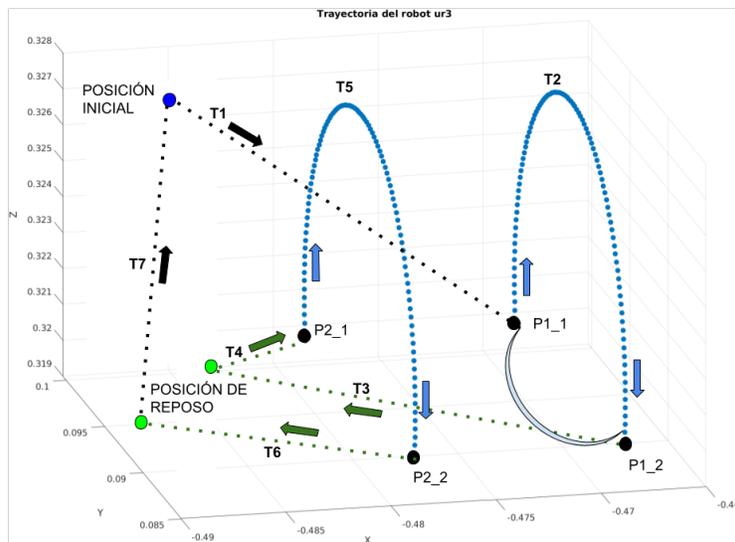


Figura 5.4.: Trayectorias UR3.2 y pinza EndoWrist 2 [Fuente propia].

- **Trayectoria T1: Posicionamiento inicial**

La trayectoria T1 del brazo robot UR3.2 describe la translación desde la posición inicial hasta el punto de sutura P1\_1, donde sujeta la aguja desde la punta.

- **Trayectoria T2: Extracción de aguja – Primera puntada**

Una vez la aguja ha sido sujeta, se ejecuta la trayectoria de extracción T2, con el fin de extraer completamente la aguja de la piel del paciente, una vez finalizado este proceso, el brazo robot UR3.1 sujeta la aguja desde la cabeza.

- **Trayectoria T3: Posición de reposo - Primera puntada**

Posteriormente, el brazo robot UR3.2 libera la aguja y ejecuta la trayectoria T3 con el fin de evitar colisiones en el proceso de sutura, hasta llegar al punto de reposo

- **Trayectoria T4: Posicionamiento - Segunda puntada**

Una vez completado el proceso de inserción de la segunda puntada, el brazo robot UR3.2 ejecuta la trayectoria T4, desde su posición de reposo hasta el punto P2\_1, donde retoma la aguja.

- **Trayectoria T5: Extracción de aguja - Segunda puntada**

Al retomar la aguja el brazo robot UR3.1 se retira del área de operación y se ejecuta la trayectoria extracción T5, para así extraer completamente la aguja, una vez finalizado este proceso, el brazo robot UR3.1 sujeta nuevamente la aguja desde la cabeza.

- **Trayectoria T6: Posición de reposo - Segunda puntada**

A continuación, el brazo robot UR3.2 libera la aguja y ejecuta la trayectoria T6, saliendo así del área de operación hasta llegar al punto de reposo.

- **Trayectoria T7: Posicionamiento inicial**

Finalmente, el brazo robot UR3.2, ejecuta la trayectoria T7, volviendo así a su posición de origen y dando por terminado el proceso de sutura, después de realizar dos puntadas.

---

## 5.2. Análisis de trayectoria

Dentro de esta sección, se exponen los resultados derivados de las curvas generadas para el proceso de inserción y extracción de la aguja, comparándolos con los movimientos obtenidos dentro del simulador. El propósito es discernir de manera meticulosa el comportamiento del sistema. Además, se determinó el rendimiento general del sistema en relación con las posiciones y orientaciones deseadas, contrastándolas con las posiciones y orientaciones obtenidas.

En el script de parámetros, se definió la dimensión de la aguja, el número de puntadas y las coordenadas de los puntos de sutura. Se buscó que estos tuvieran variaciones en todos sus ejes, como se observa en la figura 5.5.

```
1 %%INICIALIZACION
2
3 %rosinit
4 clearvars
5 clear all
6 close all
7 clc
8
9 %% PARAMETROS
10
11 % Radio de la aguja
12
13 R_aguja= 0.0048;
14
15 % Numero de Puntadas
16
17 N=1;
18
19 % Puntos de inicio desutura(Cm)
20
21 A11 = [0; 1.8; 0.10];      B11 = [0.5; 0.9; 0.17];
22
```

Figura 5.5.: Parámetros de sutura [Fuente propia].

A partir de los parámetros definidos, se ejecutó el sistema, calculando tanto el proceso de inserción como el de extracción. Para el proceso de inserción, se generó la curva observada en la figura.5.6 y a su vez el movimiento obtenido por ella se observa en la figura 5.7.

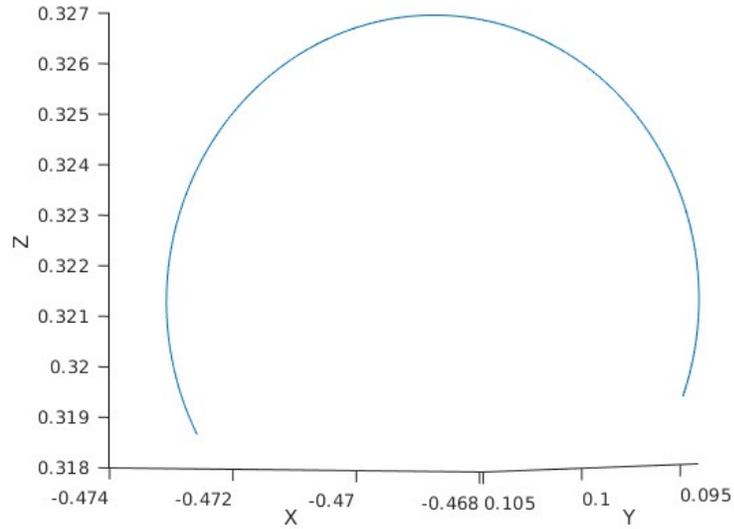


Figura 5.6.: Curva de inserción [Fuente propia].

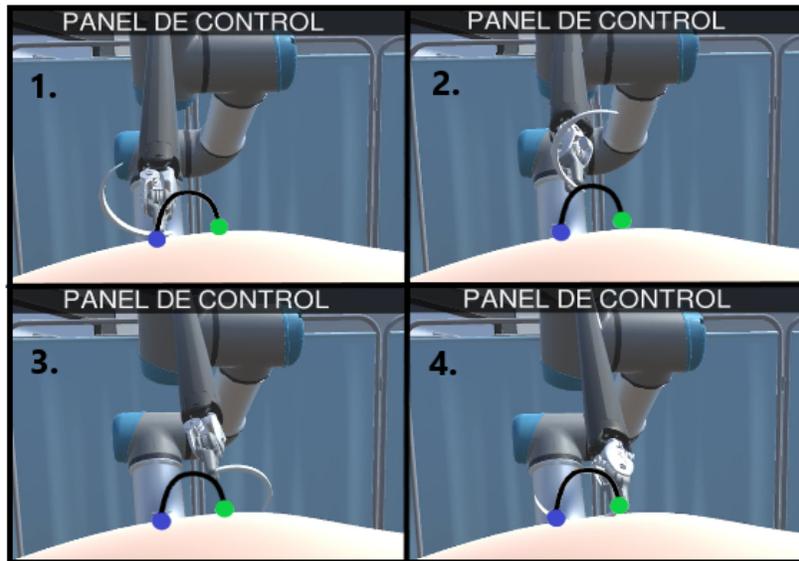


Figura 5.7.: Trayectoria de inserción de aguja [Fuente propia].

A partir de la figura 5.6 y 5.7 se logra visibilizar cómo la trayectoria calculada permite efectuar el proceso de inserción de la aguja, siguiendo fielmente dicha trayectoria y llevando a cabo de forma correcta y eficaz el proceso.

Siguiendo con el análisis, para el proceso de extracción se obtuvo como trayectoria la curva observada en la figura 5.8, generando el movimiento observado en la figura 5.9, la

---

cual se ejecuta posteriormente al proceso de inserción.

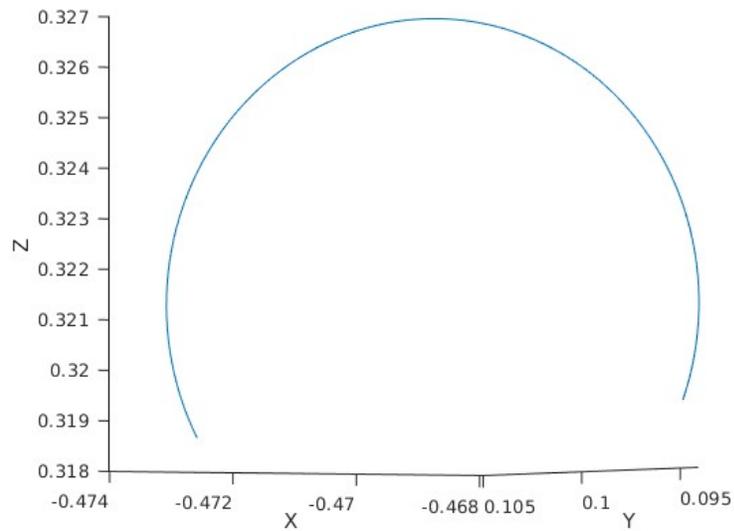


Figura 5.8.: Curva de extracción [Fuente propia].

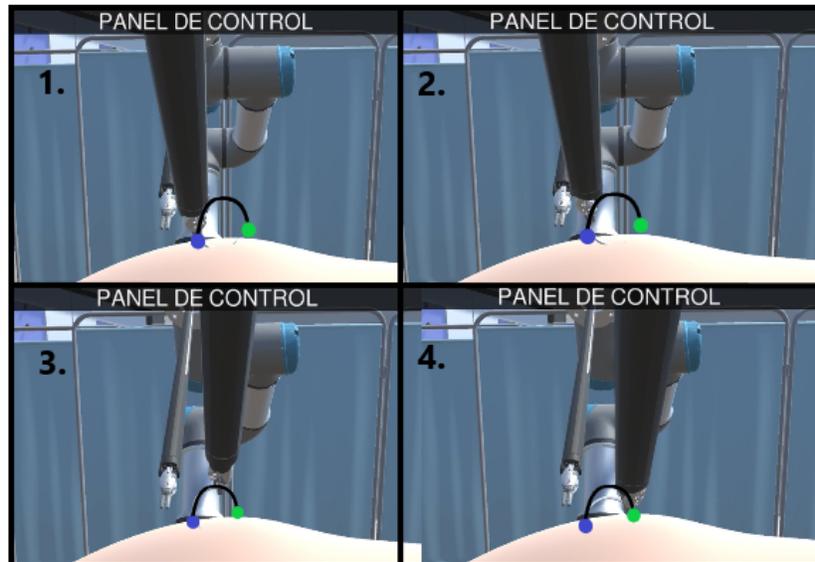


Figura 5.9.: Trayectoria de extracción de aguja [Fuente propia].

Al considerar las gráficas 5.8 y 5.9 , se evidencia cómo la trayectoria generada culmina en el proceso de extracción de la aguja, lo que proporciona claridad sobre la realización del seguimiento y el cumplimiento de dicha tarea.

Es importante señalar que, en este caso, se optó por observar el funcionamiento de la inserción y extracción de la aguja mediante una sola puntada. No obstante, el sistema es capaz de calcular 'n' puntadas siempre y cuando se definan las coordenadas de sutura (Ver figura 5.11).

```

% Puntos de inicio desutura(Cm)
A11 = [0; 2.4; 0.14];      B11 = [0; 1.3; 0.12];

% Puntos de desutura (Cm)
A12 = [1; 0.2; 0];        B12 = [1; 0.2; 0];
A13 = [0.5; 0.5; 0];      B13 = [0.8; 0.5; 0.01];
A14 = [1; -0.5; 0];       B14 = [1; -0.5; 0];
A15 = [1; -0.2; 0.02];    B15 = [1; -0.2; 0.02];

```

Figura 5.10.: Script puntos de sutura [Fuente propia].

Como se puede observar en la figura anterior, se establecieron cinco puntadas, presentando variaciones para cada punto en cualquiera de sus tres ejes. Esto posibilita al usuario definir diversas formas, que incluyen inclinaciones (Ver figura 5.11 ). Posteriormente, se calculan sus respectivas trayectorias y son ejecutadas por los dos brazos robóticos UR3.

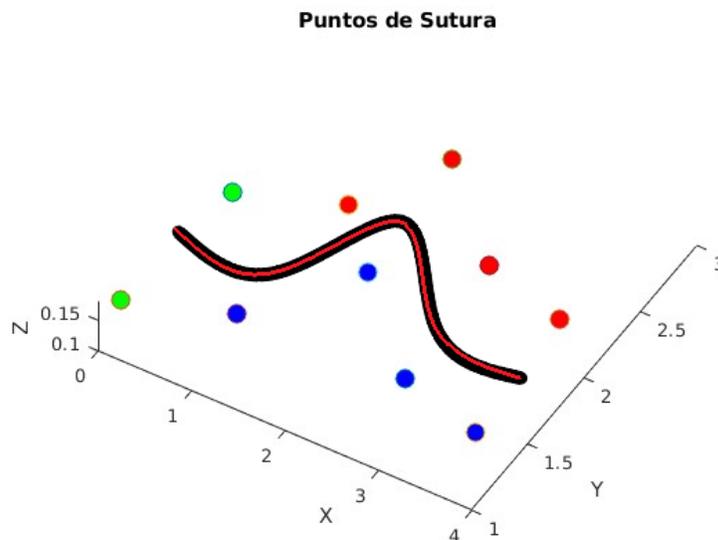


Figura 5.11.: Puntos de sutura [Fuente propia].

### 5.2.1. Error de posición y orientación

Para llevar a cabo las diversas trayectorias mencionadas anteriormente, el brazo robot UR3 ejecuta una serie de movimientos en sus diferentes articulaciones, lo que permite el cumplimiento del proceso de sutura. Dentro de la figura 5.12 se proporciona una visualización detallada del involucramiento de las diversas articulaciones, demostrando cómo siguen tanto los puntos como las orientaciones calculadas para llevar a cabo la trayectoria. Con el propósito de evaluar la precisión en este proceso, se determina la discrepancia entre las posiciones y orientaciones deseadas y las posiciones y orientaciones efectivamente alcanzadas.

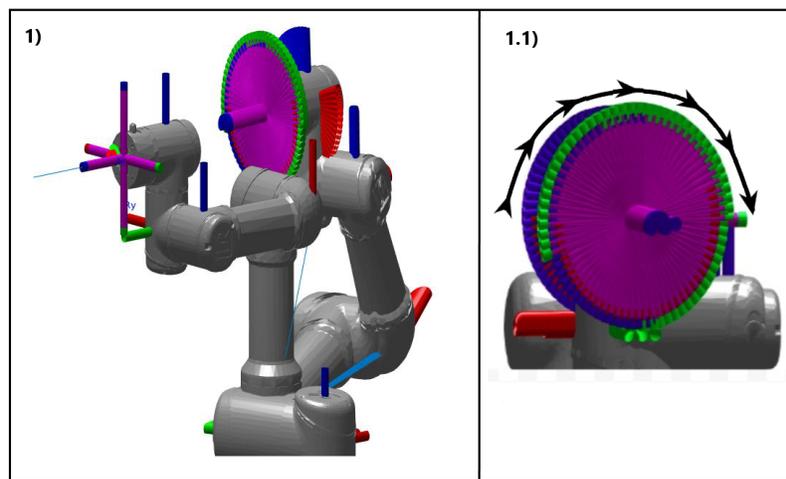


Figura 5.12.: Movimientos brazo robot UR3 [Fuente propia].

La figura 5.13, que representa el error de posición en las coordenadas x, y, z, derivado de los movimientos del brazo robótico UR3e, el cual presenta una gran precisión con la que los ejecuta. Los valores de error registrados no superan el orden de  $8 \times 10^{-9}$ , lo que subraya la notoria exactitud con la que el robot ejecuta sus movimientos, esta pequeña discrepancia en la posición, no representa ningún inconveniente en el desarrollo del proceso; resaltando la fiabilidad y la capacidad de este sistema para llevar a cabo las diferentes tareas con una alta precisión espacial.

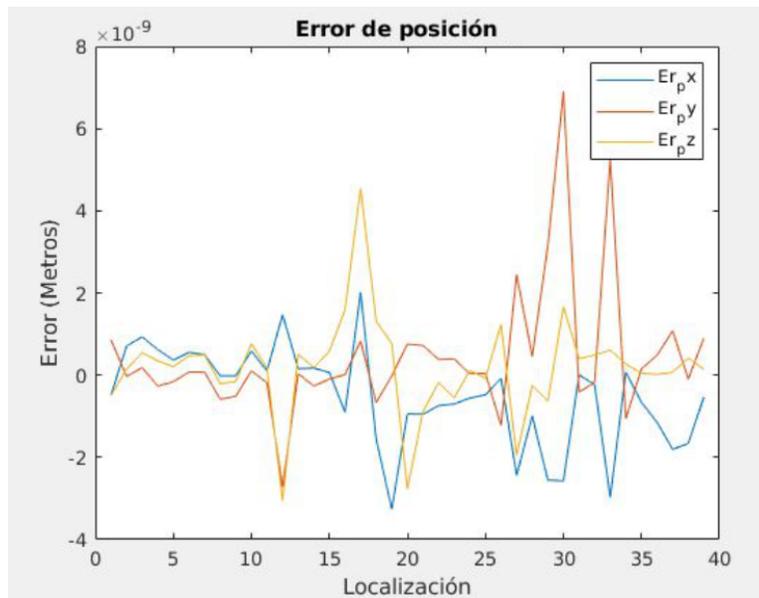


Figura 5.13.: Error de Posición [Fuente propia].

Por otro lado, la figura 5.14 que ilustra el error de orientación en las coordenadas x, y, z, también revela un desempeño notable, los valores de error para este caso se mantienen con un error máximo en el orden de  $1 \times 10^{-7}$ , denotando la habilidad del brazo robótico UR3e para mantener la orientación deseada, esta precisión en la orientación, respaldando la confiabilidad y la eficacia de este sistema con una alineación precisa.

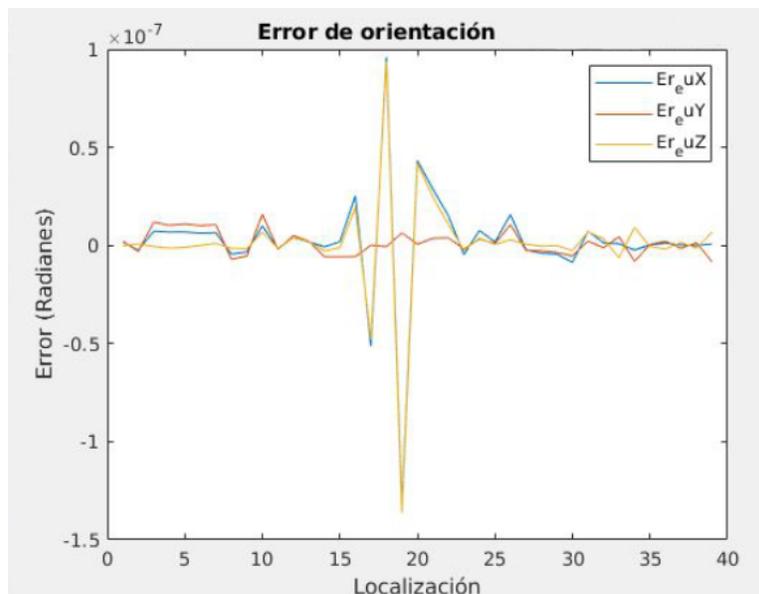


Figura 5.14.: Error de Orientación [Fuente propia].

---

Los resultados obtenidos respecto al error muestran una excepcional precisión y confiabilidad del brazo robótico UR3e en la ejecución de movimientos y en la correcta preservación de la orientación deseada durante la realización del proceso de sutura.

### **5.3. Entorno de simulación**

Teniendo en cuenta el problema abordado, resulto imperativo establecer un entorno de simulación, por lo que, se optó por aprovechar un simulador previamente desarrollado en Unity que ya incluía un brazo robótico UR3 y funcionalidades de comunicación mediante ROS, cuya base se originó dentro del trabajo de grado “Plataforma software para la manipulación de objetos 3D para un robot UR” de los estudiantes en Automática Industrial Juan David Ruiz Florez y Juan Sebastian Montenegro Bravo, no obstante, para alinear el simulador con los objetivos específicos de esta tesis, se requirió una adaptación detallada del simulador.

Esta adaptación comprendió la incorporación de los elementos esenciales para el desarrollo de este trabajo, incluyendo dos brazos robóticos UR3, dos pinzas EndoWrist y dos agujas, así mismo se creó un entorno similar al encontrado en un quirófano, esto con el fin de dar más realismo al proceso y de simular el ambiente al que debe adaptarse el procedimiento, por último, se modificó la interfaz con la que se interactúa creando una interfaz que estuviera alineada con los requerimientos específicos de esta investigación.

#### **5.3.1. Descripción del entorno de simulación**

El escenario comprende una amplia gama de objetos, entre los cuales destacan los robots UR3e, las pinzas EndoWrist y las agujas, que constituyen los principales elementos de simulación, además, se encuentran presentes objetos como mesas, pantallas, camilla quirúrgica, porta suero, entre otros elementos relacionados con el entorno, estos últimos se han incorporado con el propósito de contextualizar el escenario y conferirle un ambiente más realista, es importante destacar que estos elementos ambientales no exhiben movimiento ni interacción con los objetos principales. A continuación, se presentará el escenario desarrollado desde diversas perspectivas para proporcionar una visión completa del mismo.

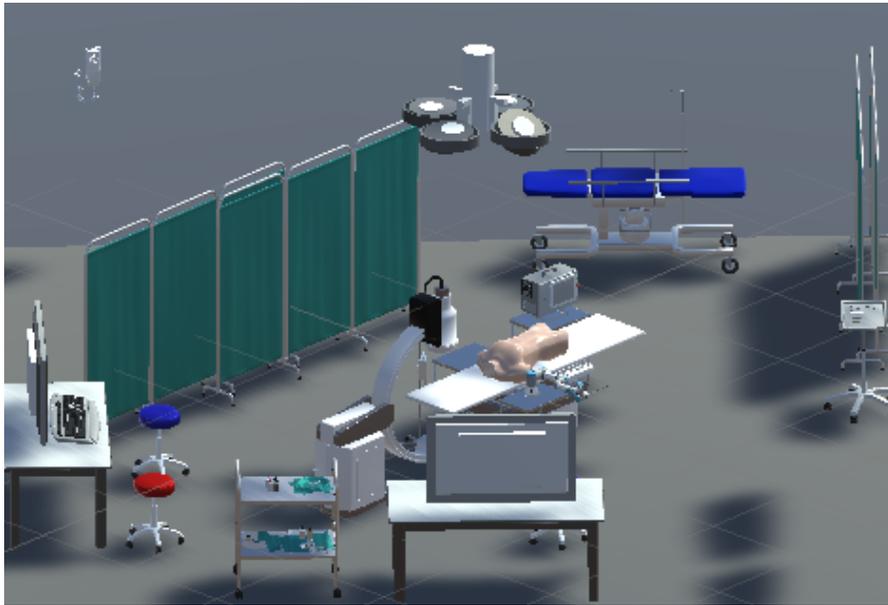


Figura 5.15.: Entorno quirúrgico, Vista lateral [Fuente propia].

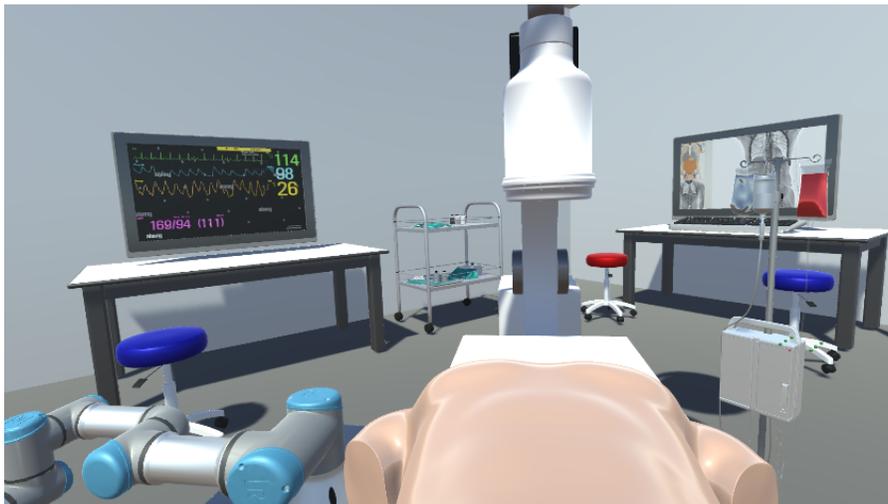


Figura 5.16.: Entorno quirúrgico, Vista Simulador [Fuente propia].

En la figura 5.16, la vista proporcionada se centra en el área donde se llevará a cabo el proceso de sutura. Esto se realiza con el propósito de visualizar con precisión los movimientos ejecutados por cada uno de los brazos robóticos en conjunto con sus pinzas. Es importante destacar que, a pesar del enfoque principal, es posible obtener otro ángulo de visión mediante el desplazamiento de la cámara con las teclas de flechas o su rotación utilizando las teclas W, A, S, D. Estas funciones permiten obtener una

---

visibilidad completa del entorno y del proceso de sutura.

Dentro de la figura 5.17, se logra observar la parte superior de la pantalla, una pestaña denominada ‘PANEL DE CONTROL’, esta pestaña posibilita expandir y contraer un menú en el cual se encuentran las opciones de ‘Conexión’ y ‘Herida’



Figura 5.17.: Pestaña ‘PANEL DE CONTROL’ [Fuente propia].



Figura 5.18.: Panel de control [Fuente propia].

La funcionalidad ‘Conexión’ posibilita el establecer la conexión entre MATLAB y Unity mediante el uso de ROS. Para llevar a cabo este procedimiento, resulta imperativo ingresar la dirección IP del equipo en la sección correspondiente y activar el botón ‘Conectar’, en caso de requerir la interrupción de la conexión, simplemente se debe seleccionar el botón ”Desconectar” deteniendo así la conexión.



Figura 5.19.: Opción Conexión [Fuente propia].

La funcionalidad ‘Herida’ fue concebida con el propósito de obtener una perspectiva adicional del proceso de sutura, sin requerir ajustes en la posición de la cámara principal, esta característica brinda dos puntos de vista en tiempo real, aportando claridad y una comprensión más completa del procedimiento en curso.



Figura 5.20.: Opción Herida [Fuente propia].

### 5.3.2. Simulación

Dentro de esta sección, se detalla el proceso de sutura tal como se observa dentro del entorno de simulación. Al dar inicio a la simulación, los brazos robot obtienen sus respectivas coordenadas iniciales y se posicionan como se muestra en la siguiente imagen, donde el brazo robot UR3\_1 contara en su pinza EndoWrist\_1 con la aguja que realizará

la sutura.

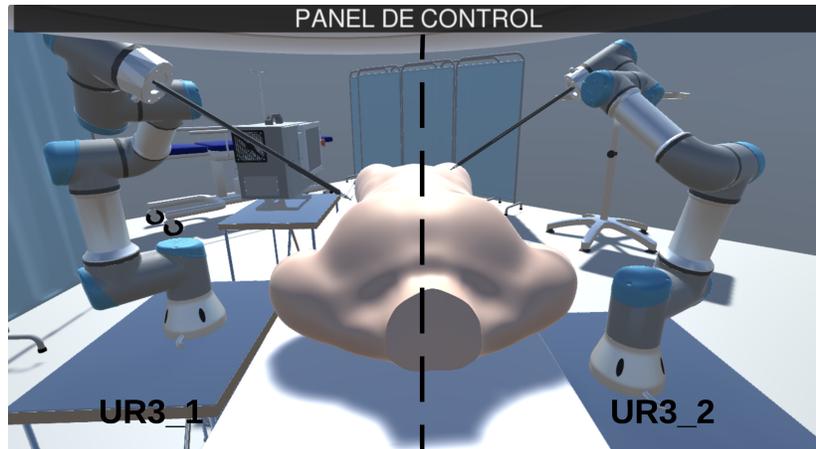


Figura 5.21.: Posición inicial UR3\_1, UR3\_2 [Fuente propia].

Al encontrarse el brazo robot UR3\_1 en su posición inicial (Ver figura 5.22), se realiza el primer movimiento por parte del brazo robot UR3\_1, el cual se desplaza hacia las coordenadas que corresponden al primer punto de sutura (Ver figura 5.23).

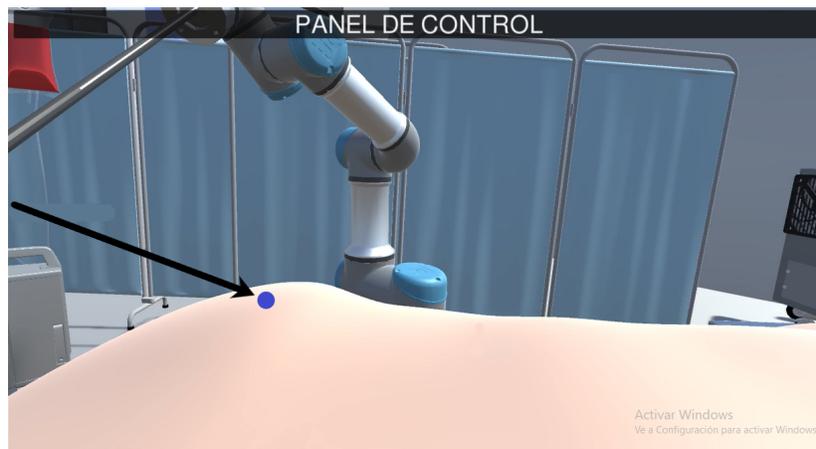


Figura 5.22.: Posición inicial- Desplazamiento UR3\_1 [Fuente propia].

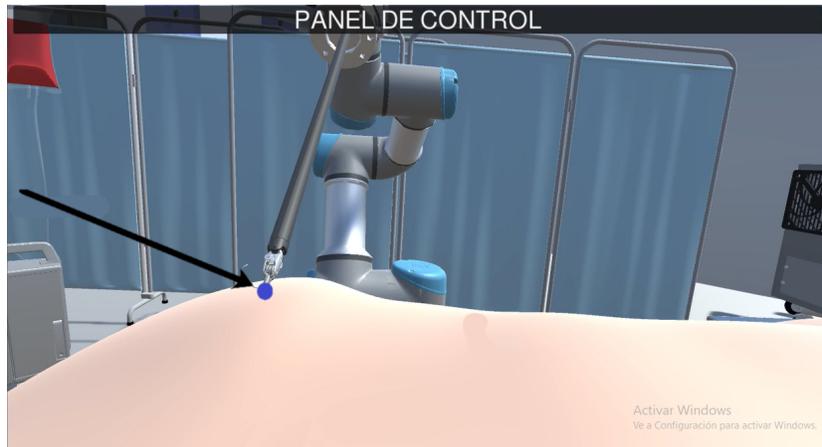


Figura 5.23.: Posición final - Desplazamiento UR3\_1 [Fuente propia].

Una vez posicionado sobre los puntos de sutura, el brazo robot UR3\_1 inicia el proceso de inserción de la aguja. Este proceso implica una traslación semicircular en conjunto con la rotación de su última articulación, la cual tiene sujeta la pinza EndoWrist\_1, generando un movimiento sobre la aguja. Esta ingresa por uno de los extremos de la herida y sale por el otro extremo de la misma (Ver figura 5.24 y 5.25).

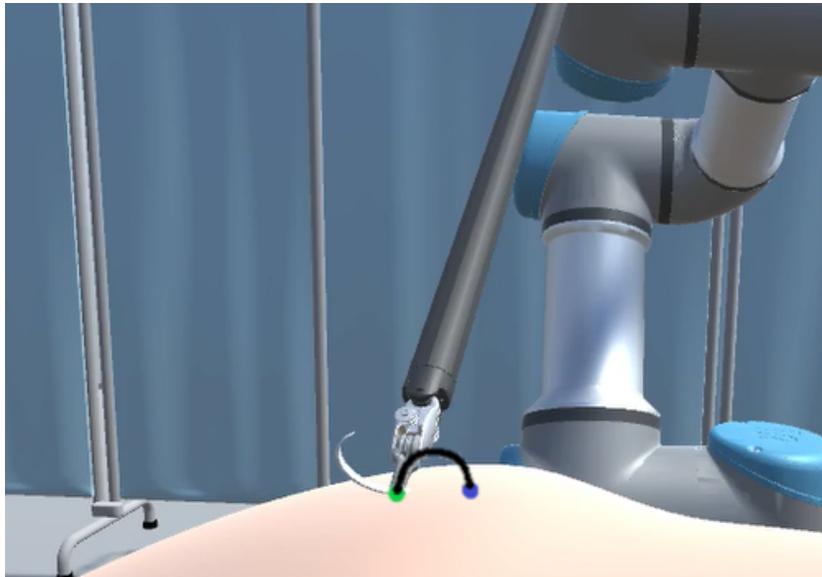


Figura 5.24.: Posición inicial- Inserción de Aguja [Fuente propia].

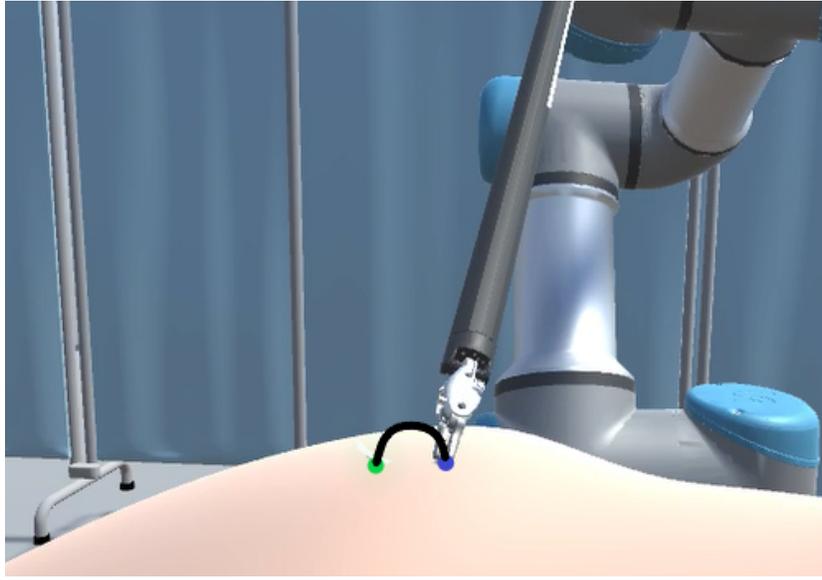


Figura 5.25.: Posición final - Inserción de Aguja [Fuente propia].

Al completarse el proceso de inserción de la aguja, el brazo robot UR3.2 en conjunto con la pinza EndoWrist.2 ubicada en su última articulación, se desplaza hacia la posición de la punta de la aguja (Ver figura 5.26 y 5.27). Una vez finaliza la trayectoria, el brazo robot UR3.2 sujeta la aguja, para que así el brazo robot UR3.1 la libere.

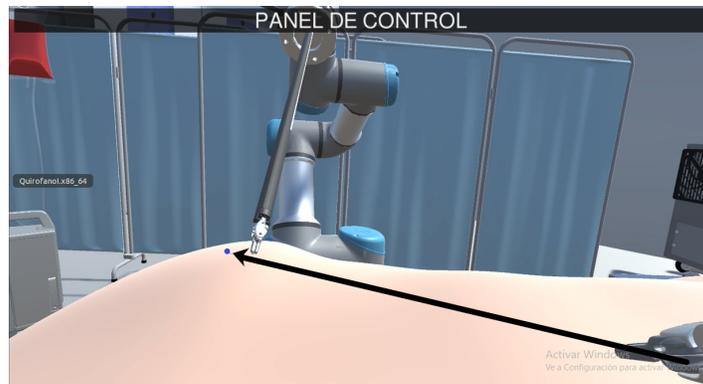


Figura 5.26.: Posición - Desplazamiento UR3.2 [Fuente propia].

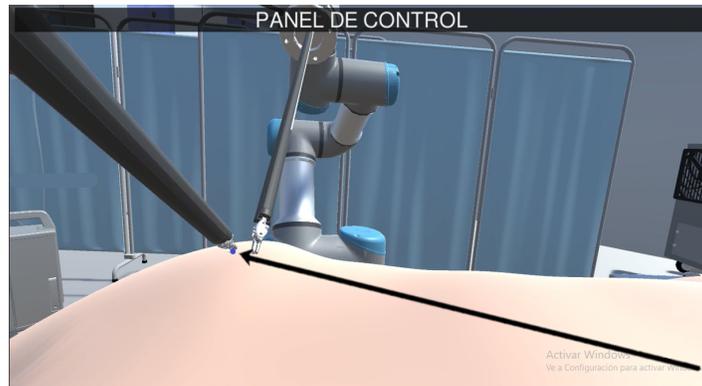


Figura 5.27.: Posición final - Desplazamiento UR3.2 [Fuente propia].

Al ser liberada la aguja por el brazo robot UR3.1, este se desplaza a su posición de reposo, saliendo del área de operación (Ver figura 5.28 y 5.29) permitiéndole al brazo UR3.2 iniciar con el proceso de extracción de aguja.

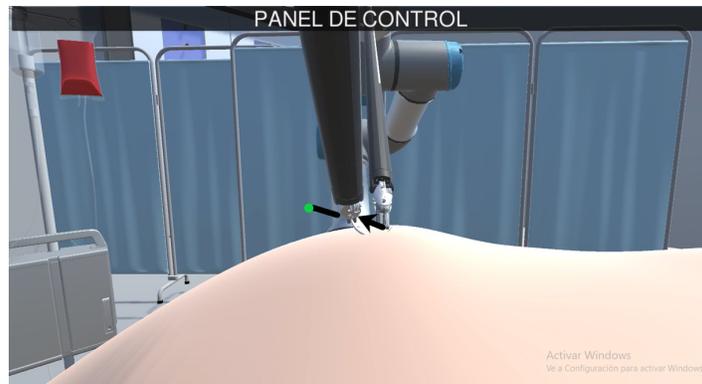


Figura 5.28.: Posición inicial - Desplazamiento UR3.1 [Fuente propia].

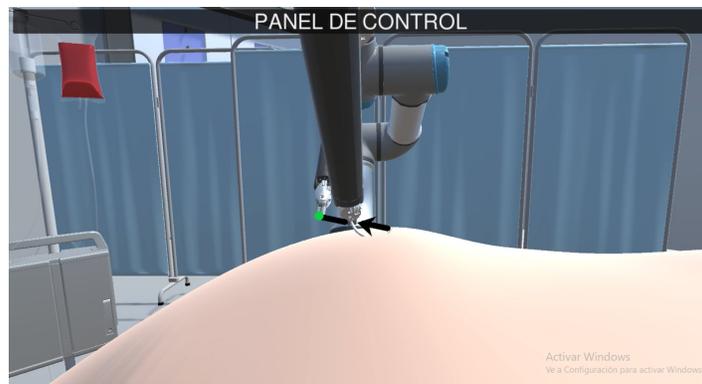


Figura 5.29.: Posición final - Desplazamiento UR3.1 [Fuente propia].

---

El proceso de extracción de la aguja se realiza ejecutando una trayectoria semicircular en conjunto con la rotación de su última articulación, la cual tiene sujeta la pinza EndoWrist\_2, logrando que se extraiga en su totalidad la aguja de la herida (Ver figura 5.30 y 5.31).

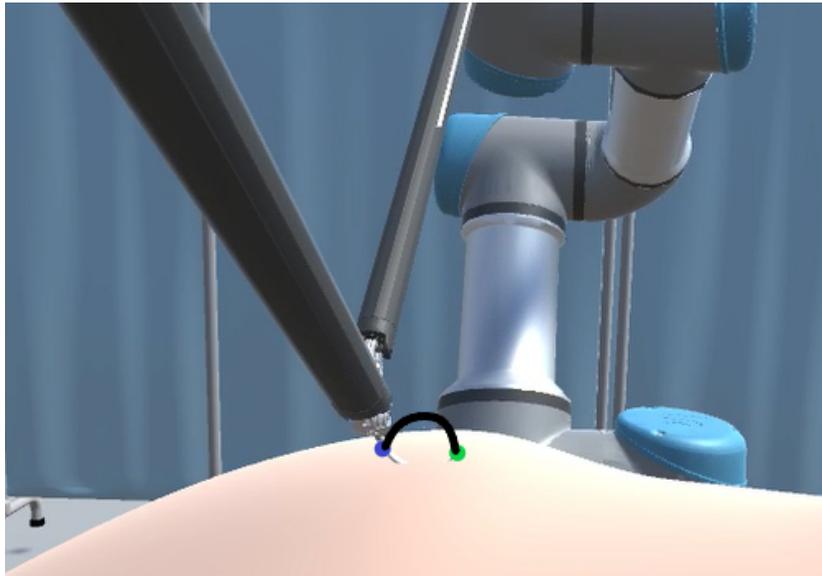


Figura 5.30.: Posición inicial- Extracción de Aguja [Fuente propia].

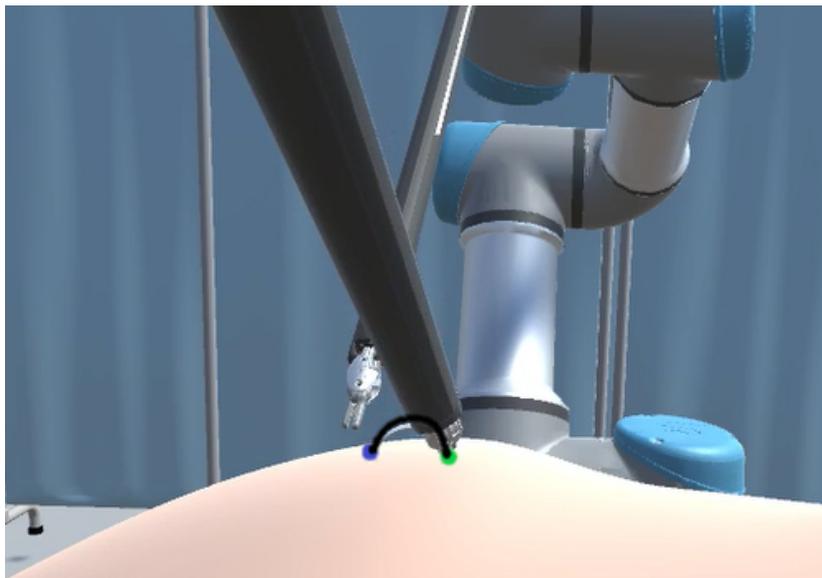


Figura 5.31.: Posición final - Extracción de Aguja [Fuente propia].

Posteriormente, el brazo robot UR3.1 se desplaza desde su posición de reposo hacia la

posición de la cabeza de la aguja, en este punto la pinza EndoWrist\_1, sujeta nuevamente la aguja (Ver figura 5.32 y 5.33).

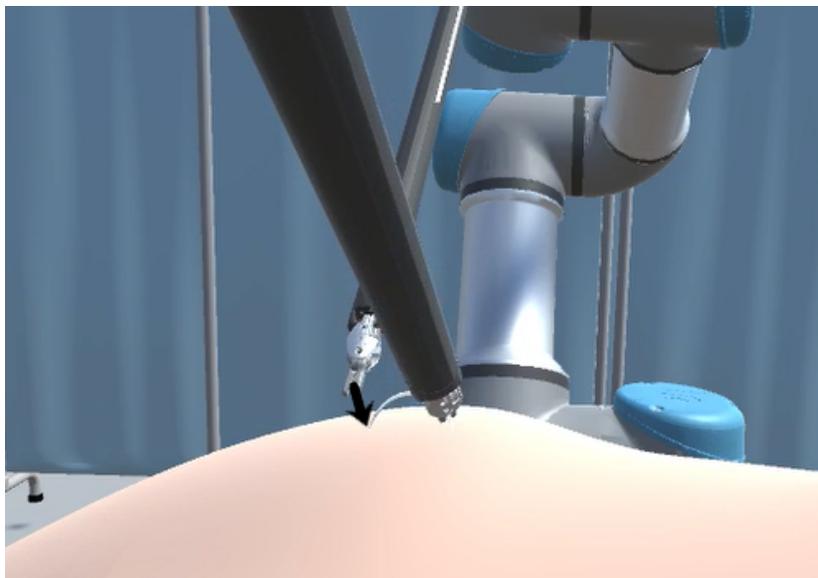


Figura 5.32.: Posición inicial - Desplazamiento UR3\_1 [Fuente propia].

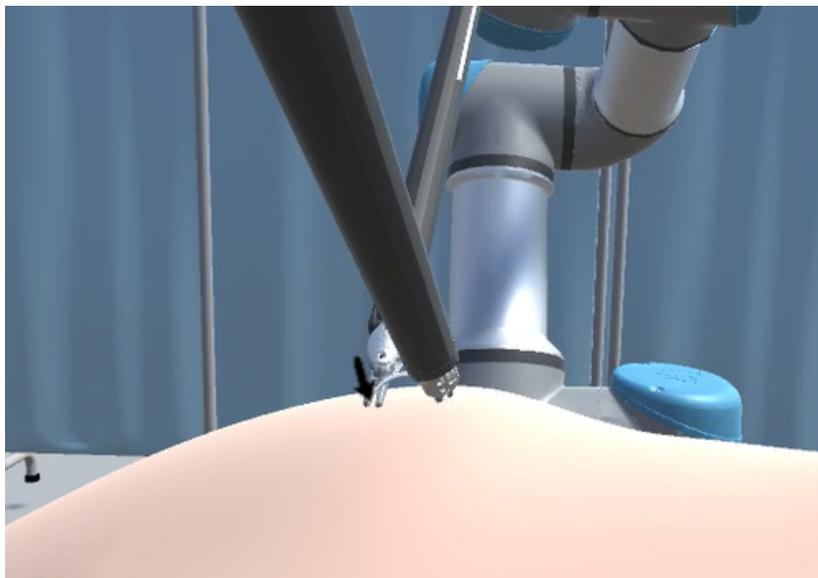


Figura 5.33.: Posición final - Desplazamiento UR3\_1 [Fuente propia].

Una vez está asegurada la aguja, el brazo robot UR3.2 se desplaza a su posición de reposo (Ver figura 5.34 y 5.35) despejando así la zona de operación.

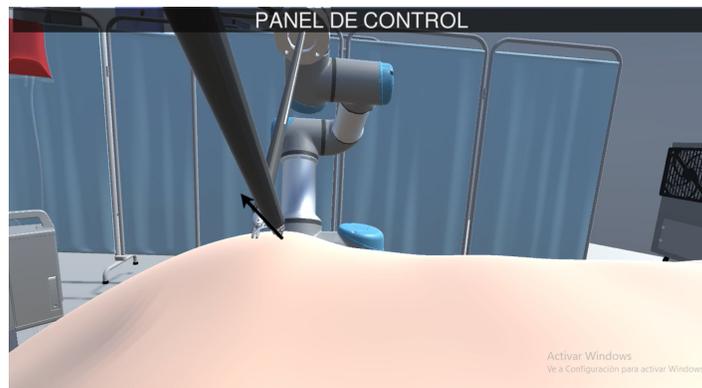


Figura 5.34.: Posición inicial - Desplazamiento UR3\_2 [Fuente propia].

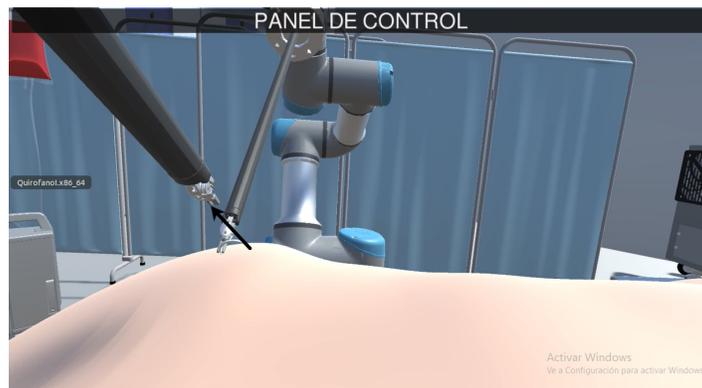


Figura 5.35.: Posición final - Desplazamiento UR3\_2 [Fuente propia].

Dando así por terminada la puntada, en este punto, el sistema determina por medio del número de puntas definidas, si se continúa con otra o se da por finalizado el proceso de sutura. En el caso de que se continúe con otra punta, el brazo robot UR3.1 se desplazará al siguiente punto de sutura (Ver figura 5.36 y 5.37), donde se repetirá nuevamente el proceso de la puntada, este ciclo se repite hasta realizar el total de puntadas definidas.

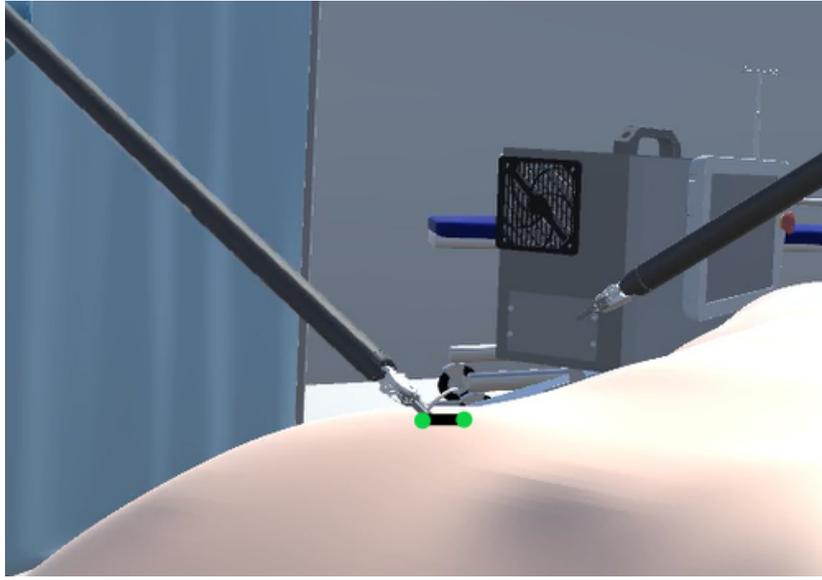


Figura 5.36.: Posición inicial - Desplazamiento UR3\_1 [Fuente propia].

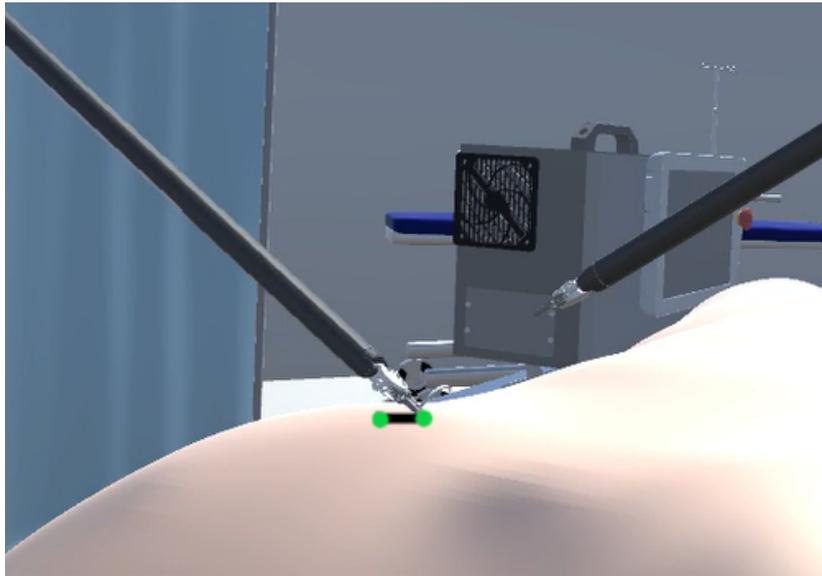


Figura 5.37.: Posición final - Desplazamiento UR3\_1 [Fuente propia].

Por otro lado, si se da por finalizado el proceso de puntadas, los brazos robot UR3\_1 y UR3\_2 ejecutan de forma ordenada sus respectivas trayectorias para llegar a su posición inicial, dando así por completado el procedimiento de sutura, siguiendo el orden respectivo mostrado en las siguientes figuras 5.38, 5.39, 5.40 y 5.41.

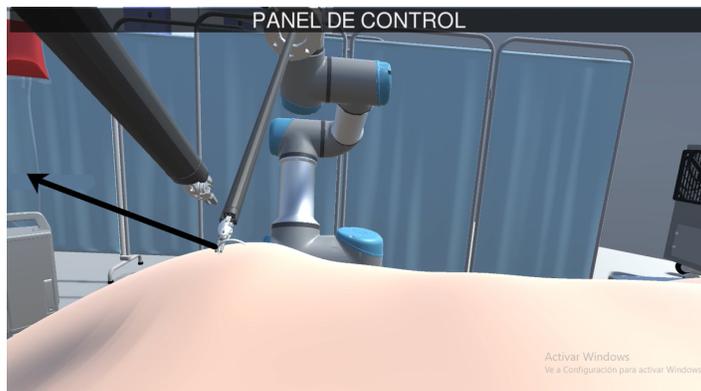


Figura 5.38.: Posición inicial - Desplazamiento UR3\_1 [Fuente propia].

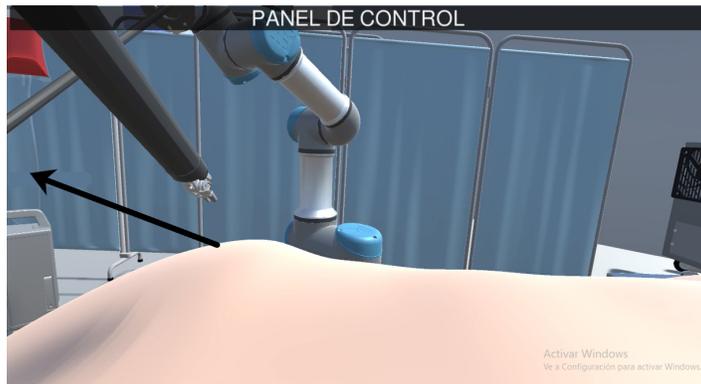


Figura 5.39.: Posición final - Desplazamiento UR3\_1 [Fuente propia].

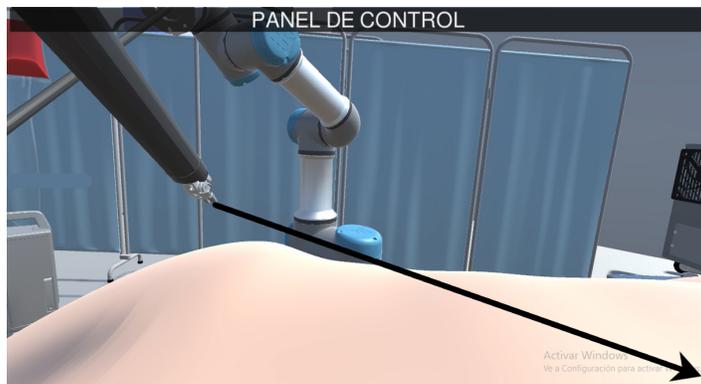


Figura 5.40.: Posición inicial - Desplazamiento UR3\_2 [Fuente propia].

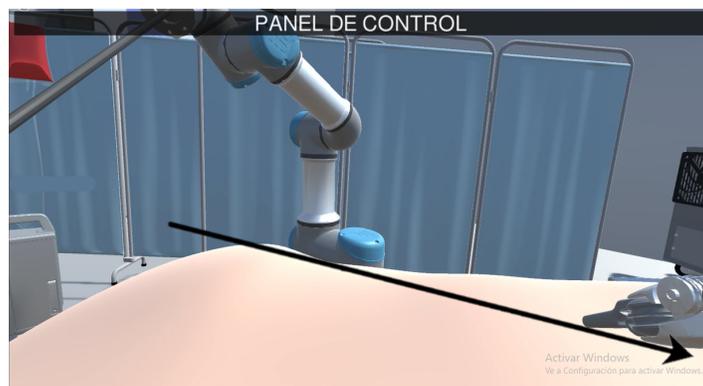


Figura 5.41.: Posición final - Desplazamiento UR3.2 [Fuente propia].

## 6. Conclusiones

- En el marco de este estudio, se pudo obtener un sistema efectivo, capaz de determinar de manera precisa, la trayectoria tridimensional de una sutura compuesta por tres puntos sobre la superficie de una herida simulada.
- Se logró la creación de un sistema versátil capaz de calcular tanto trayectorias lineales como circulares. Este sistema se traduce en la precisa ubicación del robot, desempeñando funciones cruciales como la inserción y extracción de la aguja en el proceso de sutura.
- Se ha logrado desarrollar un entorno simulado que posibilita la evaluación del sistema en condiciones análogas a la realidad, contribuyendo así a una mejor comprensión del proceso y de las características con las que se enfrenta.
- Se logró exitosamente establecer una conexión eficiente entre Matlab y Unity mediante el uso de ROS (Robot Operating System), asegurando un flujo de información en tiempo real, permitiendo la integración de ambas plataformas, y facilitando la transferencia instantánea y precisa de datos entre las dos plataformas.

# **A. Anexos**

## **A.1. Introducción**

El desarrollo de este trabajo de grado implica la aplicación de una variedad de recursos, programas y aplicaciones que desempeñan el papel fundamental en la investigación y la producción del proyecto. Estos elementos tecnológicos no solo facilitan el proceso de recopilación y análisis de datos, sino que también contribuyen significativamente a la presentación y representación visual de los resultados obtenidos. En el contexto de este trabajo de grado, se han empleado diversos recursos que abarcan desde herramientas de investigación en línea hasta software especializado. La combinación de estas herramientas ha permitido abordar de manera integral los objetivos planteados en la investigación, asegurando la calidad y la fiabilidad de los resultados obtenidos. Entre los recursos utilizados se encuentran plataformas como lo son Matlab, Unity, ROS y una partición en disco duro en la cual se hizo uso del sistema operativo LINUX. A continuación, se especifican cada una de plataformas y recursos que fueron usados a lo largo del desarrollo del presente trabajo.

## **A.2. Instalación Ubuntu 18.04**

El primer paso será descargar Ubuntu 18.04, donde se selecciona la opción Desktop imagen como se muestra en la figura-A.1.

## Ubuntu 18.04.6 LTS (Bionic Beaver)

### Select an image

Ubuntu is distributed on three types of images described below.



A red arrow points to the 'Desktop image' option. The page shows two main sections: 'Desktop image' and 'Server install image'. Each section has a title, a description, and a link to the download page.

<b>Desktop image</b> The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 1024MiB of RAM to install from this image.	<a href="#">64-bit PC (AMD64) desktop image</a> Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.
<b>Server install image</b>	<a href="#">64-bit PC (AMD64) server install image</a> Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

### Anexo A.1.: Descarga Ubuntu

Posteriormente se descarga Rufus, donde se selecciona la versión más reciente, como se observa en la figura-A.2



The screenshot shows the Rufus website with a dark background. It includes a description of the tool, its uses, and a list of download links. A red arrow points to the 'Rufus 3.21' link in the list.

Rufus es una utilidad que le ayuda a formatear y crear soportes USB de arranque, como «pendrives», tarjetas de memoria, etcétera.

Es especialmente útil en casos donde:

- necesite crear medios de instalación USB a partir de ISOs arrancables (Windows, Linux, UEFI, etc.)
- necesite trabajar en un equipo que no tenga un sistema operativo instalado
- necesite actualizar el firmware o BIOS de un ordenador desde DOS
- quiera ejecutar una utilidad de bajo nivel

A pesar de su pequeño tamaño, Rufus ofrece todo lo que necesita.

Ah, y Rufus es **rápido**. Por ejemplo, es casi dos veces más rápido que UNetbootin, Universal USB Installer o la herramienta de descarga a USB de Windows 7, en la creación de un instalador USB de Windows 7 desde una ISO. También es ligeramente más rápido en la creación de USB de arranque de Linux a partir de ISO. <sup>(1)</sup>

Más abajo hay una lista no exhaustiva de ISO admitidas por Rufus. <sup>(2)</sup>

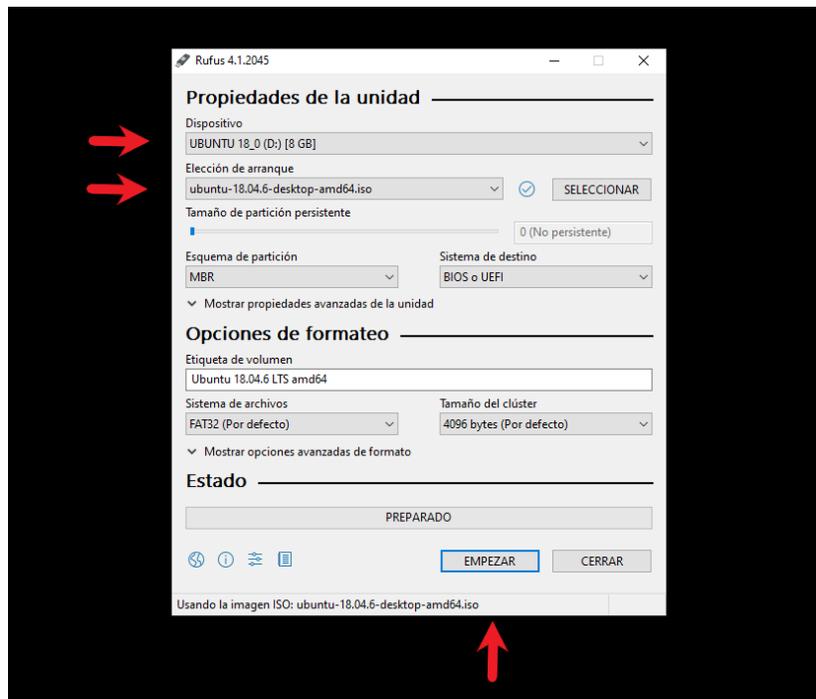
**Descargar**

Actualizado el 2022.11.28:

- **Rufus 3.21** (1.3 MB)
- Rufus 3.21 Portable (1.3 MB)
- Otras versiones (GitHub)
- Otras versiones (FossHub)

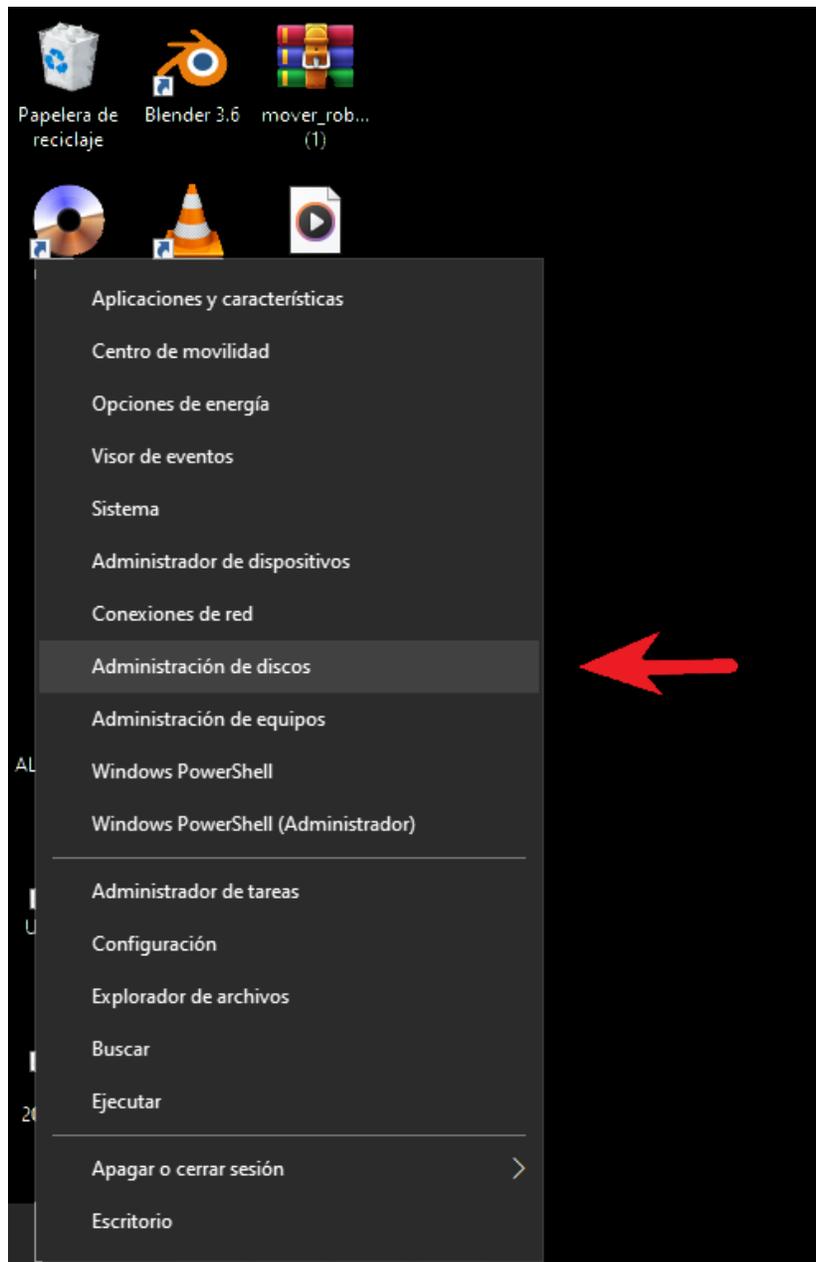
### Anexo A.2.: Descarga Rufus

Para continuar con la instalación es necesario insertar una memoria USB, la cual no contenga ningún tipo de información o que pueda ser formateada, una vez insertada se abre rufus, aquí se debe seleccionar el dispositivo insertado y la imagen ISO de Ubuntu, como se muestra a continuación.



Anexo A.3.: Imagen ISO de Ubuntu

Se debe tener en cuenta que para realizar la instalación es necesario que se disponga como mínimo de 8 Gb de memoria RAM y de alrededor de 300 GB de espacio en nuestro disco duro. Teniendo en cuenta las recomendaciones anteriores, se crea una nueva partición en el disco duro, para esto se presiona clic derecho sobre el icono de inicio y seleccionar la opción de administrador de discos.



Anexo A.4.: Opción administrador de discos

En el administrador de Discos dar clic derecho sobre la última partición y se modifica el volumen de tal manera que nos quede un nuevo disco de 300GB.



---

## A.3. Instalación ROS MELODIC

La instalación de Ros Melodic se realiza directamente desde la página (<https://wiki.ros.org/melodic/Installation/Ubuntu>) siguiente a esto en una terminal se debe ejecutar los siguientes comandos de manera consecutiva como se muestra a continuación.

### Configuración de lista de fuentes

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" >
/etc/apt/sources.list.d/ros-latest.list'
```

### Configuración de llaves

```
sudo apt install curl # if you haven't already installed curl

curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key
add -
```

### Instalación

```
sudo apt update
```

### instalación completa de escritorio

```
sudo apt install ros-melodic-desktop-full
```

### Configuración del entorno

```
echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

### Dependencias para construir paquetes

```
sudo apt install python-rosdep python-rosinstall python-rosinstall-generator python-
wstool build-essential
```

### Inicializar Rosdep

```
sudo apt install python-rosdep

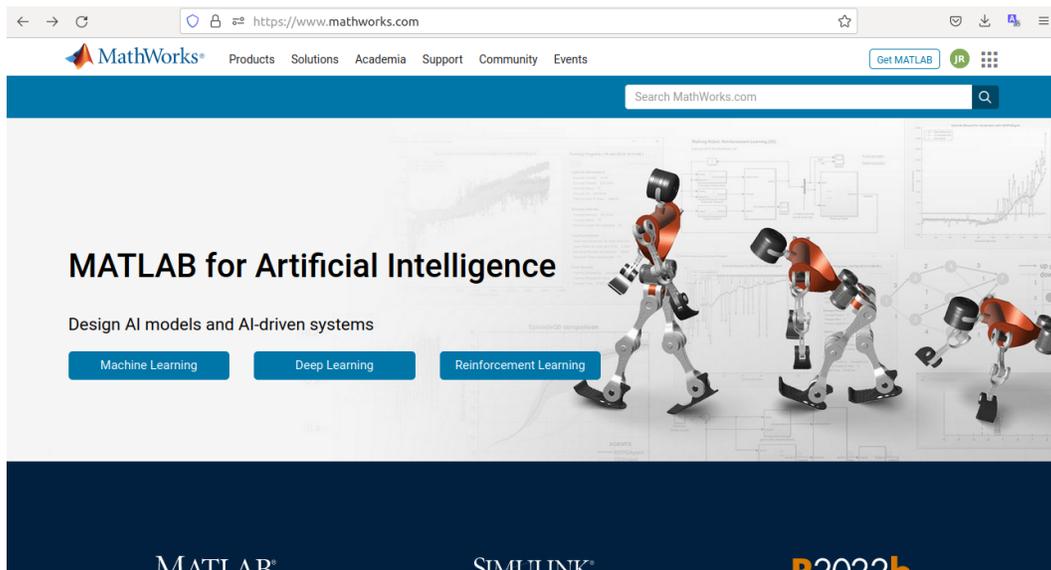
sudo rosdep init

rosdep update
```

Anexo A.7.: Comandos de instalación ROS

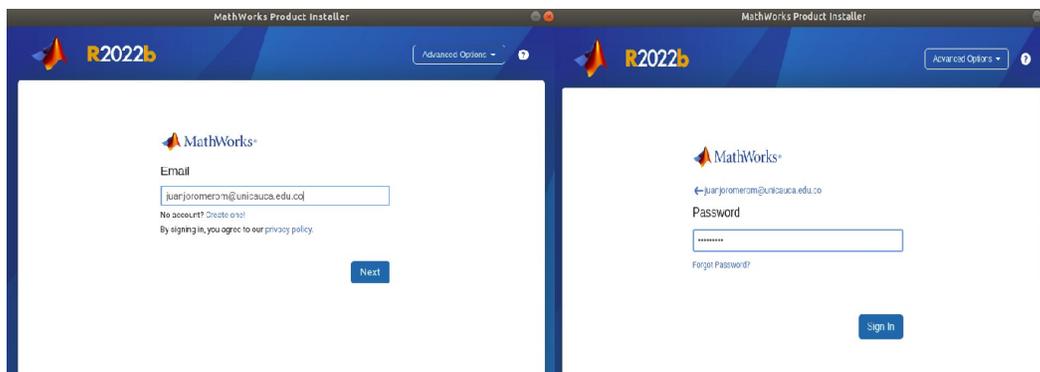
## A.4. Instalación MATLAB

Para la instalación de Matlab nos dirigimos al enlace (<https://www.mathworks.com/products/matlab/student.html>). Siendo esta la ventana principal de MATHWORKS como se observa en la siguiente imagen.



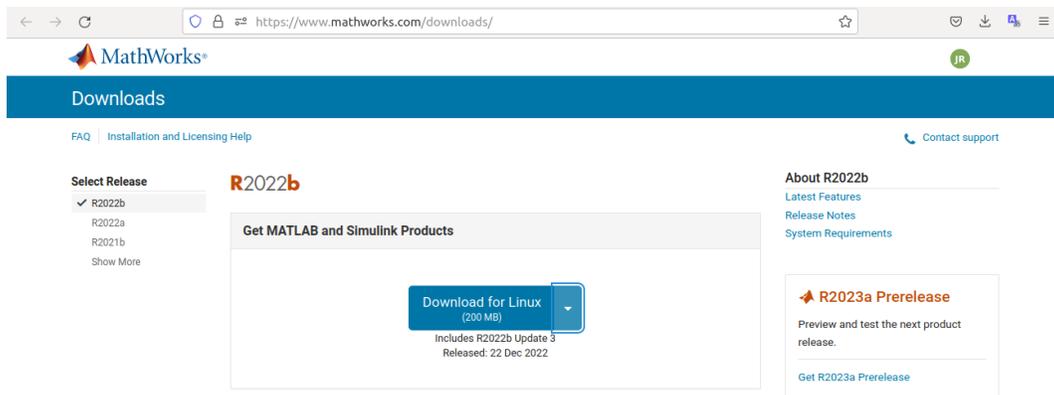
Anexo A.8.: Ventana principal MathWorks

Se ingresa con usuario y contraseña para poder realizar la descarga de Matlab, como se muestra a continuación.



Anexo A.9.: Usuario y contraseña de MathWorks

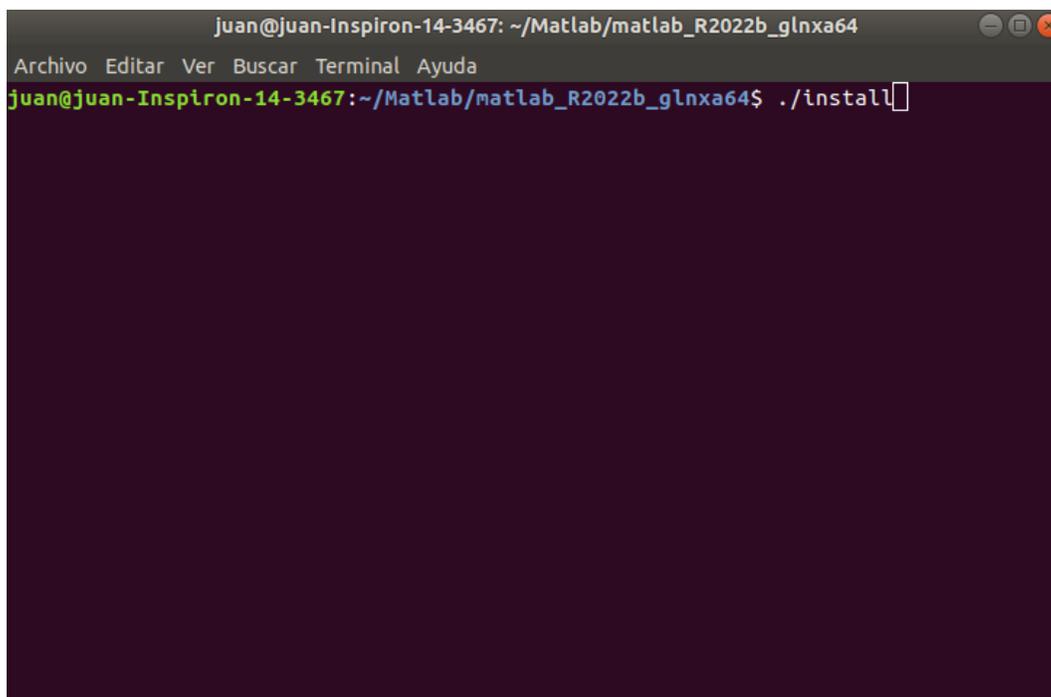
Se debe seleccionar la versión de matlab según el sistema operativo, en este caso se selecciona la versión R2022b para linux, ya que se utiliza Ubuntu para realizar esta acción.



## Anexo A.10.: MATLAB - Linux

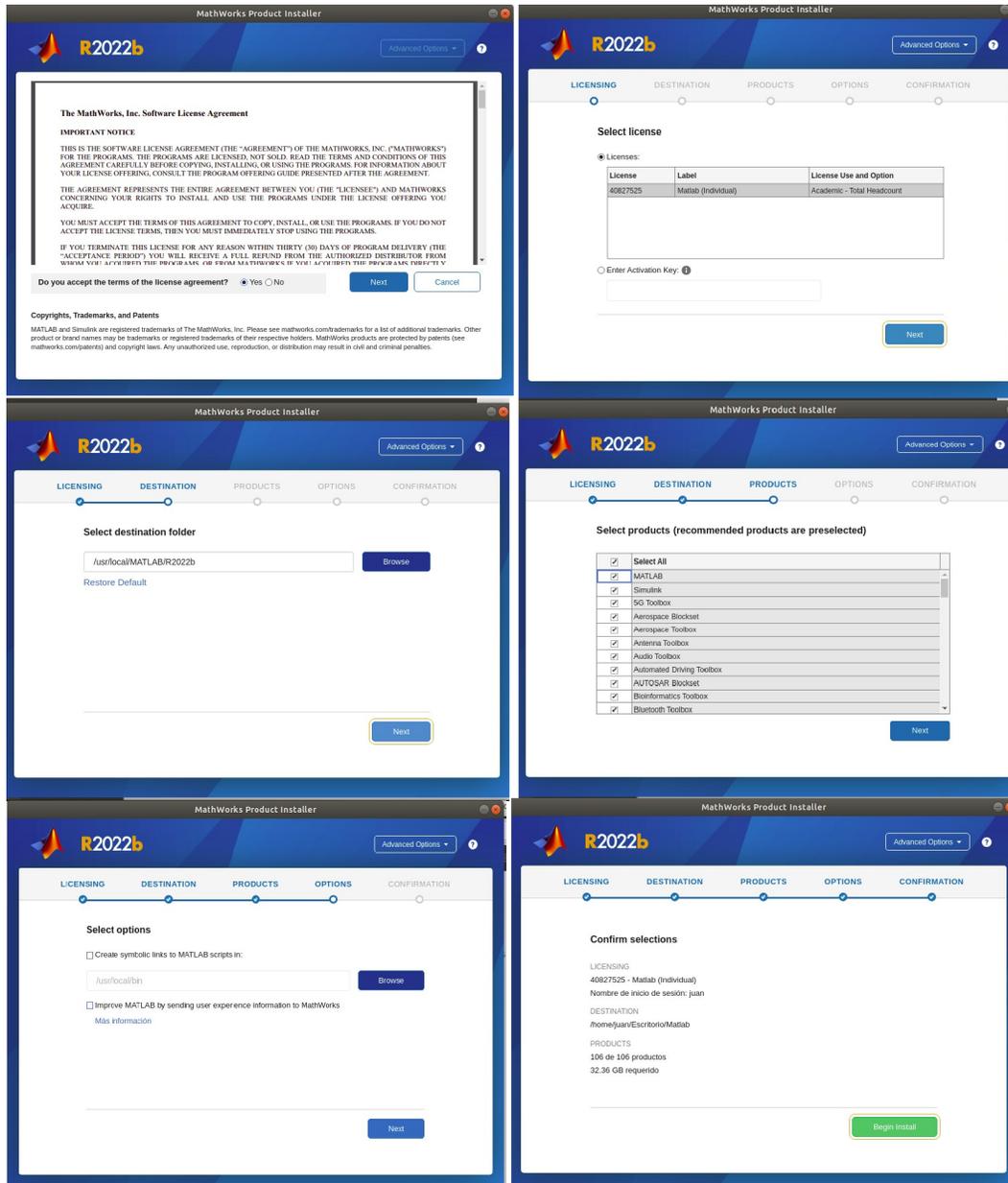
Una vez se descargue el archivo, se descomprime la carpeta, se abre la carpeta descomprimida, se hace clic derecho y seleccionar “Abrir en una terminal”.

En la terminal se escribe el comando “sudo sh install” para ejecutar el instalador como administrador.



## Anexo A.11.: Instalación de MATLAB desde terminal

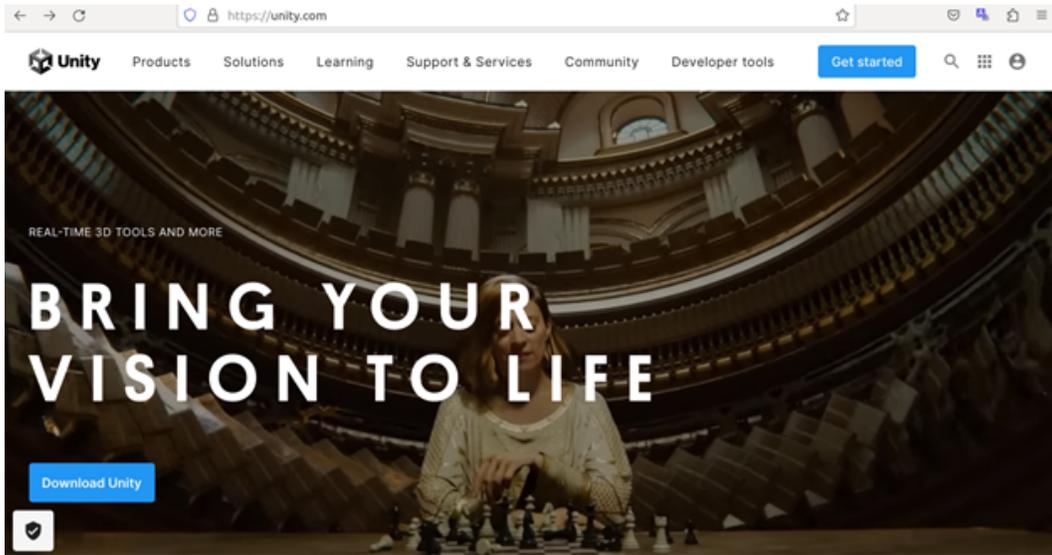
En el instalador, inicia sesión con una cuenta de MathWorks, se acepta los terminos, se selecciona la licencia, confirmar el usuario, se deja por defecto la carpeta de instalación, en este caso se marcam todos los productos, de esta forma se evita tener que descargar alguna librería en el caso de ser necesaria, seleccionar las opciones para crear el icono MATLAB en el escritorio, por ultimo se inicia la instalación como se observa en la imagen.



Anexo A.12.: Pasos de instalación

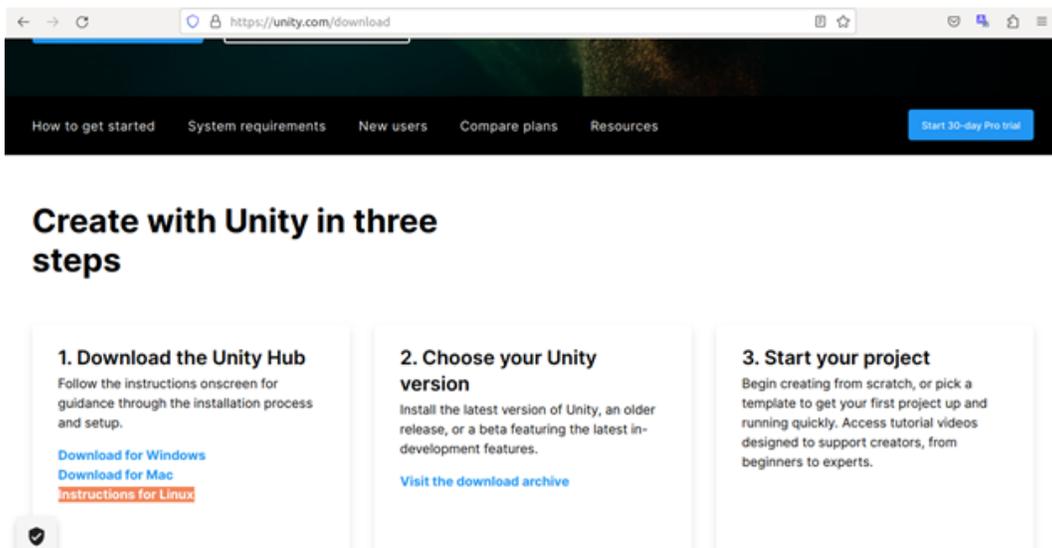
## A.5. Instalación Unity

Para la instalación de unity se dirige al siguiente enlace (<https://unity.com>) y dar clic en el boton de download Unity como se muestra en la imagen.



Anexo A.13.: Pagina principal Unity

A continuación se debe dirigir a la sección “Create with Unity in three steps” y dar click en “Instructions for Linux”.



Anexo A.14.: Instrucciones de descarga Unity

En la siguiente imagen se encuentran los comandos que se deben ejecutar secuencialmente para su correcta instalación, siendo este el paso final.

Para instalar Unity Hub en una distribución Debian o Ubuntu Linux, debe agregar el repositorio de Unity Hub Debian junto con la clave de firma pública para verificar la integridad de los paquetes.

**Para agregar la clave de firma pública, ejecute el siguiente comando:**

```
$ wget -qO - https://hub.unity3d.com/linux/keys/public | gpg --dearmor | sudo tee /usr/share/keyrings/Unity_Technologies_ApS.gpg > /dev/null
```

**Para agregar el repositorio de Unity Hub, necesita una entrada en /etc/apt/sources.list.d. Ejecute el siguiente comando para agregar el repositorio de Unity Hub:**

```
$ sudo sh -c 'echo "deb [signed-by=/usr/share/keyrings/Unity_Technologies_ApS.gpg] https://hub.unity3d.com/linux/repos/deb stable main" > /etc/apt/sources.list.d/unityhub.list'
```

**Actualice la caché del paquete e instale el paquete:**

```
$ sudo apt update
```

```
$ sudo apt-get install unityhub
```

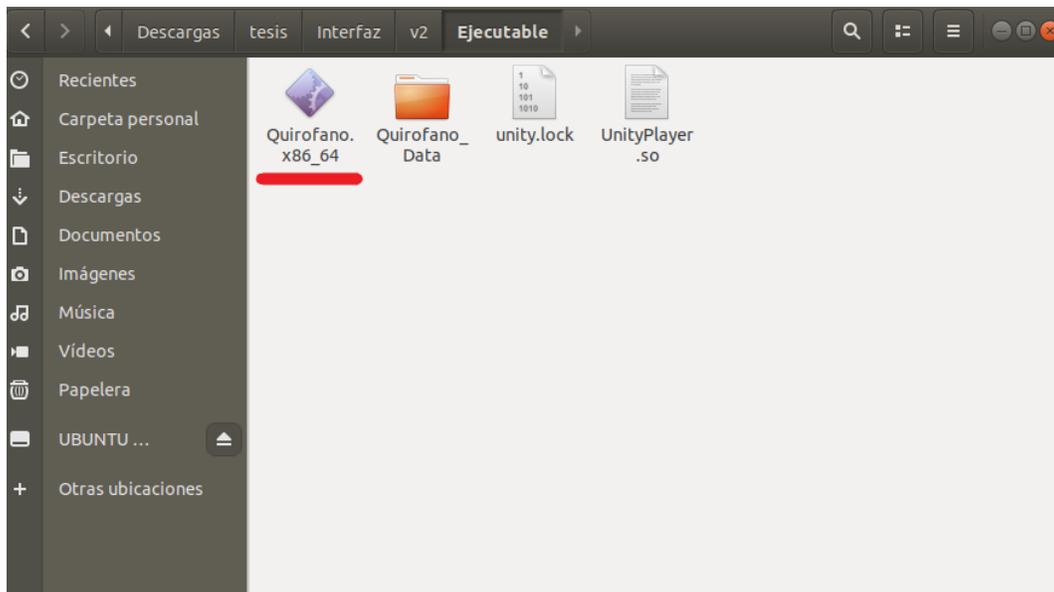
Anexo A.15.: Comandos de instalación

## A.6. Guía de ejecución del simulador

Para iniciar la ejecución del simulador, se debe tener en cuenta que el orden en el que se ejecute tanto como la primer y segunda subsección no afectan en el funcionamiento del mismo, a partir de la 3 subsección se recomienda ejecutarlas en el mismo orden que se propone a continuación.

### A.6.1. Inicio del simulador

En este caso los archivos que contienen el ejecutable del simulador se encuentran en la dirección /home/juan/Descargas/tesis/Interfaz/v2/Ejecutable, al entrar en la ubicación se encuentran los archivos mostrados en la figura A.16, donde para dar inicio al simulador, se da clic sobre el icono Quirofano x86\_64.



Anexo A.16.: Ejecutable simulador

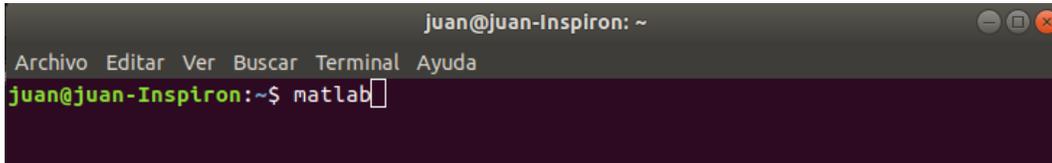
Al dar clic sobre el icono mencionado anteriormente, cargará el simulador como se muestra en la figura A.17, en este punto no se realiza ninguna acción sobre el mismo hasta que se complete los pasos de la tercera subsección.



Anexo A.17.: Simulador

## A.6.2. Inicio de MATLAB

Para dar inicio a Matlab, se ejecuta dentro de una terminal el comando ‘matlab’, como se muestra en la figura A.18, hasta que cargue completamente, para evitar ralentizar el proceso de ejecución.



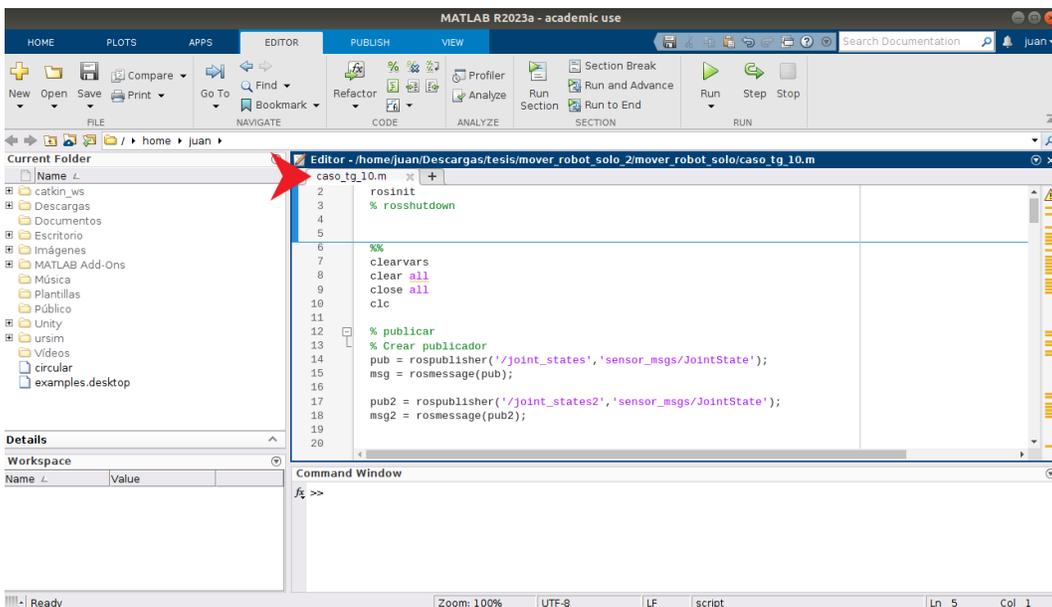
```

juan@juan-Inspiron: ~
Archivo Editar Ver Buscar Terminal Ayuda
juan@juan-Inspiron:~$ matlab

```

Anexo A.18.: Comando ‘matlab’

Una vez haya cargado completamente Matlab, se debe abrir el script con el nombre ‘caso\_tg\_10’ como se observa en la figura A.19, el cual es el encargado de ejecutar todos los procesos para generar la trayectoria de sutura.



```

MATLAB R2023a - academic use
HOME PLOTS APPS EDITOR PUBLISH VIEW
New Open Save Compare Go To Find Refactor Profiler Section Break Run and Advance Run Step Stop
FILE NAVIGATE CODE ANALYZE SECTION RUN
Current Folder: /home/juan
Name: caso_tg_10.m
2 rosinit
3 % rosshutdown
4
5
6 %%
7 clearvars
8 clear all
9 close all
10 cfc
11
12 % publicar
13 % Crear publicador
14 pub = rospublisher('/joint_states','sensor_msgs/JointState');
15 msg = rosmessage(pub);
16
17 pub2 = rospublisher('/joint_states2','sensor_msgs/JointState');
18 msg2 = rosmessage(pub2);
19
20
Command Window
>>

```

Anexo A.19.: MATLAB, script ‘caso\_tg\_10’

Este script se ejecuta una vez se tenga establecida la conexión con el simulador, razón por la cual no se realiza ninguna otra acción dentro del mismo.

---

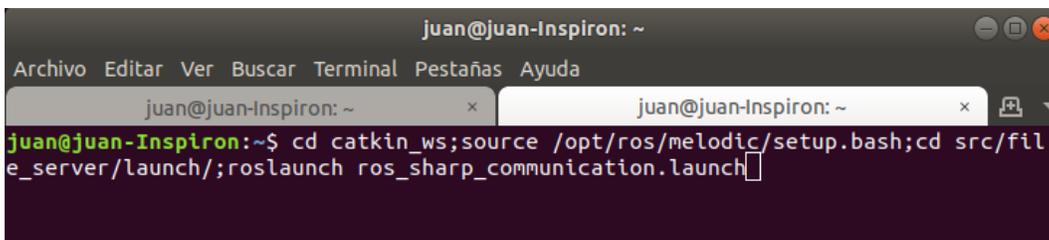
### A.6.3. Inicializar la comunicación por medio de ROS

Para iniciar la comunicación por medio de ROS es primordial configurar la misma, para esto se hará uso de los siguientes comandos dentro de una nueva terminal.

```
% cd catkin_ws  
% source /opt/ros/melodic/setup.bash  
% cd src/file_server/launch/  
% roslaunch ros_sharp_communication.launch
```

Con el fin de facilitar el proceso de inserción de todos los comandos, se puede simplificar separando cada línea por un ‘punto y coma’, como se muestra a continuación o en la figura A.20, donde se muestra su uso en la terminal.

```
% cd catkin_ws;source /opt/ros/melodic/setup.bash;cd src/file_server/launch/;roslaunch  
ros_sharp_communication.launch
```

A screenshot of a terminal window titled 'juan@juan-Inspiron: ~'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', 'Pestañas', and 'Ayuda'. Below the menu bar, there are two tabs, both labeled 'juan@juan-Inspiron: ~'. The terminal content shows the command: 'juan@juan-Inspiron:~\$ cd catkin\_ws;source /opt/ros/melodic/setup.bash;cd src/file\_server/launch/;roslaunch ros\_sharp\_communication.launch' with a cursor at the end of the line.

Anexo A.20.: Comando de configuración ROS

Una vez ejecutados los diferentes comandos, se debe obtener como respuesta al final ‘ver started at ws’, como se observa en la figura A.21

```

ros_sharp_communication.launch http://localhost:11311
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
juan@juan-Inspiron: ~ x ros_sharp_communication.launch http://lo... x
[INFO] [1700097011.824177]: Rosapi started
2023-11-15 20:10:13-0500 [-] Log opened.
2023-11-15 20:10:14-0500 [-] registered capabilities (classes):
2023-11-15 20:10:14-0500 [-] - rosbridge_library.capabilities.call_service.Call
Service
2023-11-15 20:10:14-0500 [-] - rosbridge_library.capabilities.advertise.Adverti
se
2023-11-15 20:10:14-0500 [-] - rosbridge_library.capabilities.publish.Publish
2023-11-15 20:10:14-0500 [-] - rosbridge_library.capabilities.subscribe.Subscri
be
2023-11-15 20:10:14-0500 [-] - <class 'rosbridge_library.capabilities.defragmen
tation.Defragment'>
2023-11-15 20:10:14-0500 [-] - rosbridge_library.capabilities.advertise_service
.AdvertiseService
2023-11-15 20:10:14-0500 [-] - rosbridge_library.capabilities.service_response.
ServiceResponse
2023-11-15 20:10:14-0500 [-] - rosbridge_library.capabilities.unadvertise_servi
ce.UnadvertiseService
2023-11-15 20:10:15-0500 [-] WebSocketServerFactory starting on 9090
2023-11-15 20:10:15-0500 [-] Starting factory <autobahn.twisted.websocket.WebSoc
ketServerFactory object at 0x7f0eb0badd90>
2023-11-15 20:10:15-0500 [-] [INFO] [1700097015.189578]: Rosbridge WebSocket ser
ver started at ws://0.0.0.0:9090

```

Anexo A.21.: Ejecución del comando de configuración ROS

Una vez se ha configurado la comunicación con ROS, se configura el entorno de ROS lo que facilita la comunicación entre aplicaciones externas y el sistema ROS, para esto se hará uso de los siguientes comandos en una nueva terminal.

```

%source /opt/ros/melodic/setup.bash
% roslaunch rosbridge_server rosbridge_websocket.launch

```

Teniendo en cuenta que la nueva terminal se abre como una pestaña nueva, es necesario incluir el comando 'cd', con el fin de trabajar desde el directorio de inicio, al igual que con el anterior grupo de comandos estos se simplifican de la siguiente manera y se puede observar su implementación en la figura A.22.

```

juan@juan-Inspiron: ~/catkin_ws/src/file_server/launch
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
juan@juan-Inspiron: ~ x ros_sharp_communicatio... x juan@juan-Inspiron: ~/cat... x
juan@juan-Inspiron:~/catkin_ws/src/file_server/launch$ cd;source /opt/ros/melodi
c/setup.bash;roslaunch rosbridge_server rosbridge_websocket.launch

```

Anexo A.22.: Comando de configuración del entorno ROS

Una vez ejecutados los anteriores comandos, se obtiene como respuesta final ‘Couldn’t listen on any’, como se observa en la figura A.23, la cual cambiara de estado una vez se inicie la comunicación desde unity.

```
/opt/ros/melodic/share/rosbridge_server/launch/rosbridge_websocket.launch http://localh...
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
juan@juan-Inspiron: ~ x ros_sharp_comunicatio... x /opt/ros/melodic/share/r... x
process[rosapi-2]: started with pid [10407]
2023-11-15 20:11:24-0500 [-] Log opened.
2023-11-15 20:11:24-0500 [-] registered capabilities (classes):
2023-11-15 20:11:24-0500 [-] - rosbridge_library.capabilities.call_service.Call
Service
2023-11-15 20:11:24-0500 [-] - rosbridge_library.capabilities.advertise.Adverti
se
2023-11-15 20:11:24-0500 [-] - rosbridge_library.capabilities.publish.Publish
2023-11-15 20:11:24-0500 [-] - rosbridge_library.capabilities.subscribe.Subscri
be
2023-11-15 20:11:24-0500 [-] - <class 'rosbridge_library.capabilities.defragmen
tation.Defragment'>
2023-11-15 20:11:24-0500 [-] - rosbridge_library.capabilities.advertise_service
.AdvertiseService
2023-11-15 20:11:24-0500 [-] - rosbridge_library.capabilities.service_response.
ServiceResponse
2023-11-15 20:11:24-0500 [-] - rosbridge_library.capabilities.unadvertise_servi
ce.UnadvertiseService
[INFO] [1700097084.852319]: Rosapi started
2023-11-15 20:11:24-0500 [-] [WARN] [1700097084.946614]: Unable to start server:
Couldn't listen on any:9090: [Errno 98] Address already in use. Retrying in 5s.
2023-11-15 20:11:29-0500 [-] [WARN] [1700097089.955187]: Unable to start server:
Couldn't listen on any:9090: [Errno 98] Address already in use. Retrying in 5s.
```

Anexo A.23.: Ejecución comando de configuración del entorno ROS

#### A.6.4. Consultar IP

Teniendo en cuenta que para establecer la conexión entre Matlab y Unity es necesario conocer la dirección IP de nuestro equipo, razón por la se usara el comando ‘ifconfig’, dentro de una terminal nueva, con el fin de cual obtener la dirección IP, como se muestra a continuación.

```
juan@juan-Inspiron: ~
Archivo Editar Ver Buscar Terminal Ayuda
juan@juan-Inspiron:~$ ifconfig
```

Anexo A.24.: Comando ‘ifconfig’

Al final de este comando se encontrará la dirección IP de nuestro equipo, con la que

se podrá establecer conexión.

### **A.6.5. Ejecución del simulador**

Una vez ejecutados los anteriores pasos, se vuelve nuevamente al simulador de unity, donde en la sección de conexión se ingresa la dirección IP que se obtuvo previamente, una vez ingresada se debe presionar el botón conectar, tras unos segundos dentro del simulador cambiara el estado ha ‘conectado’.

### **A.6.6. Inicio de Simulación**

Una vez el simulador se encuentra en estado de ‘conectado’, se debe ejecutar el script ‘caso\_tg\_10’ de Matlab, al iniciar la ejecución el sistema, este va calcular la trayectoria para el proceso de sutura y a su vez irá realizando el envío de las diferentes posiciones calculadas.

Dado que el sistema dará inicio al envío de las posiciones, estas serán visibles dentro del simulador, por lo que al ejecutar el script de Matlab, se debe regresar al simulador para observar los diferentes movimientos que se están ejecutando a partir de las trayectorias calculadas.

# Bibliografía

- [1] M. Kam, H. Saeidi, S. Wei, J. D. Opfermann, S. Léonard, M. H. Hsieh, J. U. Kang, and A. Krieger, “Semi-autonomous robotic anastomoses of vaginal cuffs using marker enhanced 3d imaging and path planning,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 65–73.
- [2] H. Dehghani, Y. Sun, L. Cubrich, D. Oleynikov, S. Farritor, and B. Terry, “An optimization-based algorithm for trajectory planning of an under-actuated robotic arm to perform autonomous suturing,” *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 4, pp. 1262–1272, 2020.
- [3] Y. Tian, M. Draelos, G. Tang, R. Qian, A. Kuo, J. Izatt, and K. Hauser, “Toward autonomous robotic micro-suturing using optical coherence tomography calibration and path planning,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5516–5522.
- [4] J. Colan, J. Nakanishi, T. Aoyama, and Y. Hasegawa, “Optimization-based constrained trajectory generation for robot-assisted stitching in endonasal surgery,” *Robotics*, vol. 10, no. 1, p. 27, 2021.
- [5] S. Frisken, J. Luo, N. Haouchine, S. Pieper, Y. Wang, W. M. Wells, and A. J. Golby, “Incorporating uncertainty into path planning for minimally invasive robotic neurosurgery,” *IEEE Transactions on Medical Robotics and Bionics*, vol. 4, no. 1, pp. 5–16, 2021.
- [6] S. Nema and L. Vachhani, “Safe and fast path planner for minimally invasive surgery,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2549–2554.
- [7] L. Wang, Q. Li, H. Zhao, H. Yuan, and R. Song, “Path planning of robot-assisted osteotomy in orthognathic surgery,” in *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2019, pp. 452–457.

- [8] F. Yang, Y. Dai, J. Zhang, H. Sun, L. Cui, X. Yin, X. Gao, and L. Li, "Path planning of flexible ureteroscope based on ct image," in *2019 Chinese Control Conference (CCC)*. IEEE, 2019, pp. 4667–4672.
- [9] M. T. Ramezanlou, V. Azimirad, and M. Zakeri, "Hybrid path planning of robots through optimal control and pso algorithm," in *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*. IEEE, 2019, pp. 259–264.
- [10] M. Ginesi, D. Meli, A. Roberti, N. Sansonetto, and P. Fiorini, "Autonomous task planning and situation awareness in robotic surgery," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3144–3150.
- [11] Q. Wang, Z. Wang, and M. Shuai, "Trajectory planning for a 6-dof manipulator used for orthopaedic surgery," *International Journal of Intelligent Robotics and Applications*, vol. 4, pp. 82–94, 2020.
- [12] M. Pan, Y. Chen, Z. Li, X. Liao, Y. Deng, and G.-B. Bian, "Autonomous path planning for robot-assisted pelvic fracture closed reduction with collision avoidance," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 19, no. 2, p. e2483, 2023.
- [13] N. Bao, Y. Chen, Y. Liu, and C. Chakraborty, "Multi-objective path planning for lung biopsy surgery," *Multimedia tools and applications*, pp. 1–18, 2022.
- [14] B. Su, S. Yu, X. Li, Y. Gong, H. Li, Z. Ren, Y. Xia, H. Wang, Y. Zhang, W. Yao *et al.*, "Autonomous robot for removing superficial traumatic blood," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 9, pp. 1–9, 2021.
- [15] X. Qi, J. Meng, M. Li, Y. Yang, Y. Hu, B. Li, J. Zhang, and W. Tian, "An automatic path planning method of pedicle screw placement based on preoperative ct images," *IEEE Transactions on Medical Robotics and Bionics*, vol. 4, no. 2, pp. 403–413, 2022.
- [16] R. Valero, Y. Ko, S. Chauhan, O. Schatloff, A. Sivaraman, R. Coelho, F. Ortega, K. Palmer, R. Sánchez-Salas, H. Davila *et al.*, "Cirugía robótica: Historia e impacto en la enseñanza," *Actas urológicas españolas*, vol. 35, no. 9, pp. 540–545, 2011.
- [17] B. Mathur, A. Topiwala, S. Schaffer, M. Kam, H. Saeidi, T. Fleiter, and A. Krieger, "A semi-autonomous robotic system for remote trauma assessment," in *2019 IEEE*

- 
- 19th International conference on bioinformatics and bioengineering (BIBE)*. IEEE, 2019, pp. 649–656.
- [18] D. Yang, Q. Lv, G. Liao, K. Zheng, J. Luo, and B. Wei, “Learning from demonstration: dynamical movement primitives based reusable suturing skill modelling method,” in *2018 Chinese Automation Congress (CAC)*. IEEE, 2018, pp. 4252–4257.
- [19] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. Kim, “Supervised autonomous robotic soft tissue surgery,” *Science translational medicine*, vol. 8, no. 337, pp. 337ra64–337ra64, 2016.
- [20] S. Leonard, K. L. Wu, Y. Kim, A. Krieger, and P. C. Kim, “Smart tissue anastomosis robot (star): A vision-guided robotics system for laparoscopic suturing,” *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 4, pp. 1305–1317, 2014.
- [21] H. Saeidi, H. N. Le, J. D. Opfermann, S. Léonard, A. Kim, M. H. Hsieh, J. U. Kang, and A. Krieger, “Autonomous laparoscopic robotic suturing with a novel actuated suturing tool and 3d endoscope,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1541–1547.
- [22] H. Saeidi, J. Ge, M. Kam, J. Opfermann, S. Leonard, A. Joshi, and A. Krieger, “Supervised autonomous electrosurgery via biocompatible near-infrared tissue tracking techniques,” *IEEE transactions on medical robotics and bionics*, vol. 1, no. 4, pp. 228–236, 2019.
- [23] A. Barrientos and J. del Cerro, “La robótica en medicina,” *Medicina Clínica*, vol. 152, no. 12, pp. 493–494, 2019.
- [24] S. Leonard, K. L. Wu, Y. Kim, A. Krieger, and P. C. Kim, “Smart tissue anastomosis robot (star): A vision-guided robotics system for laparoscopic suturing,” *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 4, pp. 1305–1317, 2014.
- [25] D. Yang, Q. Lv, G. Liao, K. Zheng, J. Luo, and B. Wei, “Learning from demonstration: Dynamical movement primitives based reusable suturing skill modelling method,” in *2018 Chinese Automation Congress (CAC)*, 2018, pp. 4252–4257.
- [26] K. Armas Moredo, B. A. Armas Pérez, L. Segura Pujal, J. Márquez Hernández, and K. Armas Moredo, “Materiales de sutura quirúrgico,” *Revista Archivo Médico de Camagüey*, vol. 13, no. 5, pp. 0–0, 2009.

- [27] R. Bertolo, R. Campi, T. Klatte, M. C. Kriegmair, M. C. Mir, I. Ouzaid, M. Salagierski, S. Bhayani, I. Gill, J. Kaouk *et al.*, “Suture techniques during laparoscopic and robot-assisted partial nephrectomy: a systematic review and quantitative synthesis of peri-operative outcomes,” *BJU international*, vol. 123, no. 6, pp. 923–946, 2019.
- [28] J. Rosen and T. S. Huang, *Robots in Medicine*. Ciudad de publicación: Editorial, Año de publicación.
- [29] H. Saeidi, H. N. D. Le, J. D. Opfermann, S. Leonard, A. Kim, M. H. Hsieh, J. U. Kang, and A. Krieger, “Autonomous laparoscopic robotic suturing with a novel actuated suturing tool and 3d endoscope,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1541–1547.
- [30] B. Mathur, A. Topiwala, S. Schaffer, M. Kam, H. Saeidi, T. Fleiter, and A. Krieger, “A semi-autonomous robotic system for remote trauma assessment,” in *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*, 2019, pp. 649–656.
- [31] D. E. Cucalón, “La herida quirúrgica. tipos de herida.”
- [32] T. Haidegger, S. Speidel, D. Stoyanov, and R. M. Satava, “Robot-assisted minimally invasive surgery—surgical robotics in the data age,” *Proceedings of the IEEE*, vol. 110, no. 7, pp. 835–846, 2022.
- [33] S. Léonard, K. L. Wu, Y. Kim, A. Krieger, and P. C. W. Kim, “Smart tissue anastomosis robot (star): A vision-guided robotics system for laparoscopic suturing,” *IEEE Transactions on Biomedical Engineering*, vol. 61, pp. 1305–1317, 2014.
- [34] O. L. S. Sarría, Y. G. Diez, C. M. H. Dávila, and E. D. C. de Villa, “Manual de instrumental quirúrgico,” *Medisur*, vol. 12, no. 5, pp. 781–818, 2014.
- [35] L. de la Garza Villaseñor, “Bisturíes, agujas y suturas: La evolución del material básico de la cirugía,” *Cirujano General*, vol. 30, no. 4, pp. 224–230, 2008.
- [36] N. Montejo Viamontes, L. Varela Valls, and A. Hernández González, “Materiales de sutura en cirugía: antecedentes históricos y empleo actual de los mismos,” *Rev. cuba. cir*, pp. 211–24, 1990.
- [37] M. Revol and J.-M. Servant, “Sutura,” *EMC-Cirugía Plástica Reparadora y Estética*, vol. 18, no. 2, pp. 1–8, 2010.

- 
- [38] C. Lehmann, Á. Sanabria, E. Valdivieso, and G. Gómez, “Cierre de la pared abdominal: qué técnica y qué sutura se deben usar,” *Revista Colombiana de Cirugía*, vol. 22, no. 2, pp. 102–108, 2007.
- [39] J. D. A. Adriana Marcela González, Audrey Miranda Díaz, “Principios en técnicas de suturas de piel: una guía para estudiantes,” *Bucaramanga*, 2018.
- [40] Ó. R. García, L. M. J. García, L. P. Castelló, A. G. Aparicio, and A. P. Vidal, *MATLAB: conceptos básicos y descripción gráfica*. Universidad Miguel Hernández, 2018.
- [41] A. García Cazorla *et al.*, “Ros: robot operating system,” 2013.
- [42] A. Cerón Cardona and P. A. Bedoya Herrera, “Manual básico de unity 3d como apoyo al desarrollo turístico nacional,” 2014.
- [43] <https://mmmacademy.es/wp-content/uploads/2022/04/Unity-logo.jpg>, accessed: 2023-11-12.
- [44] [https://a.storyblok.com/f/169662/1125x1500/d81c866521/png-ur3e\\_01\\_r.png/m/fit-in/343x480](https://a.storyblok.com/f/169662/1125x1500/d81c866521/png-ur3e_01_r.png/m/fit-in/343x480), accessed: 2023-11-12.
- [45] “Brazo robótico UR3e,” <https://www.universal-robots.com/es/productos/robot-ur3/>, accessed: 2023-11-03.
- [46] <https://blue-images-isi.imgix.net/en-gb/-/media/ISI/Intuitive/Card/Force-Bipolar/force-bipolar-jaws-open.jpg>, accessed: 2023-10-22.
- [47] <https://riuma.uma.es/xmlui/bitstream/handle/10630/20222/Pomalia>, accessed: 2023-11-03.
- [48] “Universal robots - DH parameters for calculations of kinematics and dynamics,” <https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/>, accessed: 2023-11-12.
- [49] “Rosinit,” <https://www.mathworks.com/help/ros/ref/rosinit.html>, accessed: 2023-11-12.
- [50] “Publish message on a topic - MATLAB,” <https://www.mathworks.com/help/ros/ref/publisher.html>, accessed: 2023-11-09.

- 
- [51] “Rosmessage,” <https://www.mathworks.com/help/ros/ref/rosmessage.html>, accessed: 2023-11-09.
- [52] “Loadrobot,” <https://www.mathworks.com/help/robotics/ref/loadrobot.html>, accessed: 2023-11-09.
- [53] “Crear un solver de cinemática inversa - MATLAB - MathWorks américa latina,” <https://la.mathworks.com/help/robotics/ref/inversekinematics-system-object.html>, accessed: 2023-11-03.
- [54] “gettransform,” <https://la.mathworks.com/help/robotics/ref/rigidbodytree.gettransform.html>, accessed: 2023-11-03.