

**DISEÑO Y FABRICACIÓN DE UN SISTEMA BASADO EN
ROBÓTICA E INTELIGENCIA ARTIFICIAL PARA LA
ASISTENCIA EN INSTRUMENTACIÓN QUIRÚRGICA**



Julian Alejandro Burbano Diaz
Rolando Mendez Sarmiento
Ingeniería Mecatrónica

**DISEÑO Y FABRICACIÓN DE UN SISTEMA BASADO EN
ROBÓTICA E INTELIGENCIA ARTIFICIAL PARA LA
ASISTENCIA EN INSTRUMENTACIÓN QUIRÚRGICA**

**JULIAN ALEJANDRO BURBANO DIAZ
COD. 1085946926**

**ROLANDO MENDEZ SARMIENTO
COD. 1057413276**



**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO MMI
INGENIERÍA MECATRÓNICA
PAMPLONA
2022**

**DISEÑO Y FABRICACIÓN DE UN SISTEMA BASADO EN
ROBÓTICA E INTELIGENCIA ARTIFICIAL PARA LA
ASISTENCIA EN INSTRUMENTACIÓN QUIRÚRGICA**

**JULIAN ALEJANDRO BURBANO DIAZ
ROLANDO MENDEZ SARMIENTO**

**TRABAJO DE GRADO COMO REQUISITO PARA OPTAR AL
TÍTULO DE INGENIERO EN MECATRÓNICA**

**Director
CRISTHIAN IVAN RIAÑO JAIMES
PH. D En sistemas mecatrónicos**

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO MMI
INGENIERÍA MECATRÓNICA
PAMPLONA
2022**

DEDICATORIA

”No es el conocimiento, sino el acto de aprendizaje; y no es la posesión, sino el acto de llegar a ella, lo que concede mayor disfrute.”

Carl Friedrich Gauss

En el transcurso de mi vida he conocido a muchas personas de gran valor que han aportado de forma significativa a mi aprendizaje y desarrollo del conocimiento, pero hubo una que me ha heredado el gusto por el trabajo y el valor del estudio, fue mi profesor y amigo, quien me brindo sus saberes y habilidad con los números y la ingeniería, por ello me es grato hacerle esta dedicatoria al docente Flavio Parra que por muchos años laboro y dejo su huella en la Institución educativa Sucre de la ciudad de Ipiales Nariño Colombia.

Julián Alejandro Burbano Diaz

“El futuro pertenece a quienes creen en la belleza de sus sueños.”

Eleanor Roosevelt

Dedico este trabajo a mi madre Claudia Patricia Sarmiento y a mi padre Ramiro Méndez Mahecha que a través de sus enseñanzas impulsan este anhelo de superación personal, con el esfuerzo constante a lo largo de los años alientan el deseo de seguir avanzando en el mundo del conocimiento; han sido unos padres ejemplares que día tras día enfrentan las arduas tareas que se realizan en el campo Colombiano, que sin importar la condiciones adversas jamás se han rendido, quisiera dedicarles a ellos este logro por la valiosa labor que han desempeñado.

Rolando Méndez Sarmiento

AGRADECIMIENTOS

Agradezco a mi familia por brindarme su apoyo y darme la oportunidad para seguir con mis estudios. Le doy gracias a mi madre por siempre estar para mí alentándome en las situaciones difíciles y también celebrando conmigo mis triunfos, a mi padre por sus consejos en los momentos que los he necesitado y a mis hermanos por la confianza que han depositado en mí.

Así mismo agradezco a Alejandra Chalpartar por ser la persona que ha estado junto a mí en toda esta etapa de mi vida, y que ha colaborado de forma directa e indirecta para la culminación exitosa de este proyecto

Julián Alejandro Burbano Diaz

Le agradezco a mis padres Claudia y Ramiro por el apoyo y la motivación de seguir adelante, por creer que la educación es el camino correcto hacia las puertas del éxito y permitirme avanzar hacia una nueva etapa de la vida, gracias a la valentía y disciplina demostrada se merecen mi mayor reconocimiento y admiración.

Agradezco a la Universidad de Pamplona y a su excelente equipo docente por guiarme en el proceso de aprendizaje, es un orgullo pertenecer a esta importante Institución.

Agradezco a Julián que es un extraordinario compañero en el ámbito académico y personal, formando un equipo perfecto de trabajo.

Rolando Méndez Sarmiento

Glosario de términos

artroplastia Extracción de las articulaciones dañadas y su sustitución por articulaciones artificiales.. 10

dataset Conjunto de datos tabulados en cualquier sistema de almacenamiento de datos estructurados.. 17

deep learning Conjunto de algoritmos de aprendizaje automático.. 1

demux circuito combinatorial que tiene una entrada de información de datos d y n entradas de control que sirven para seleccionar una de las 2^n salidas.. 50

endoscópio Dispositivo médico con una luz incluida que se utiliza para mirar dentro de una cavidad u órgano corporal.. 11

machine learning Subcampo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan.. 21

oblito Cuerpo extraño olvidado en el interior de un paciente durante una intervención quirúrgica.. vii

TensorFlow Biblioteca de código abierto para aprendizaje automático a través de un rango de tareas.. 21

Abreviaturas

2D Dos dimensiones. 5

3D Tres dimensiones. x

CAD Diseño Asistido por Computadora. ix

CNN Convolutional Neural Network. 1

CUDA Compute Unified Device Architecture. 3

EDT Estructura de descomposición del trabajo. 52

GDL Grados de libertad. x

ILSVRC ImageNet Large Scale Visual Recognition Challenge. 4

IOT Internet of things. 99

OMS Organización Mundial de la Salud. 22

PDP Proceso de Desarrollo de Productos. 25

PWM Modulación por ancho de pulso. 6

QFD Despliegue de la función de calidad. 35

Introducción

Resumen

La combinación de la medicina con la tecnología y la robótica evidencia un aumento en la calidad y fiabilidad de los procedimientos médicos, la instrumentación quirúrgica destaca su importancia para manipular los dispositivos, equipos y el instrumental en condiciones de máxima higiene a través de la conservación y protección del campo estéril y entregar los instrumentos de una manera segura; el diseño y fabricación de un sistema robótico con inteligencia artificial para la asistencia en instrumentación quirúrgica se realizara mediante un brazo articulado y el reconocimiento de objetos del instrumental, además con la capacidad de percibir y ejecutar comandos de voz específicos, por medio del cual se reconozca el instrumento que requiere el cirujano, brindando autonomía al proceso a través de la programación basada en inteligencia artificial con la finalidad de adaptarse a entornos complejos y exigentes para lograr una mayor eficiencia de tal forma que se eviten errores humanos.

Justificación

Los oblitos se presentan cuando se olvidan elementos o instrumentos de material quirúrgico en el interior del paciente, como consecuencia pone en riesgo la salud; además prolonga la recuperación, porque desde el momento en que se comienzan a presentar los primeros síntomas hasta que se hace el diagnóstico y se realiza una

nueva intervención quirúrgica puede pasar mucho tiempo incluso años, también puede provocar efectos secundarios permanentes en la vida del paciente; es por esta razón que se desarrolla un sistema robótico para la asistencia en instrumentación quirúrgica capaz de efectuar un proceso repetitivo idóneo para eliminar las causas que están vinculadas a los oblitos como las cirugías de urgencia, cambio inesperado del plan quirúrgico, obesidad, múltiples equipos quirúrgicos, excesiva pérdida de sangre, cambio de personal durante una cirugía, cansancio del equipo de cirugía o múltiples procedimientos en el mismo acto quirúrgico [1]. Por consiguiente, es necesario un sistema capaz de trabajar en colaboración con el instrumentador y el equipo de cirugía, ayudando al monitoreo constante del instrumental quirúrgico, lo cual reduce los márgenes de error humano. La aplicación de un sistema robótico facilita la labor de la instrumentación, conduciéndola a un nivel más profesional y de mejor calidad, proporcionando seguridad a los pacientes que depositan toda su confianza y esperanza en las manos de los expertos de la salud.

Planteamiento del problema

La medicina en el mundo actual es una de las principales necesidades para todo ser humano, las intervenciones quirúrgicas son unas de las opciones más efectivas en el tratamiento de enfermedades, las instalaciones y los instrumentos para llevar a cabo una intervención quirúrgica influye de manera significativa en el éxito y el bienestar de la salud del paciente, durante el procedimiento médico los márgenes de error se deben reducir al máximo, los casos donde se olvida extraer instrumentos o material quirúrgico utilizados durante el procedimiento como el caso de “una paciente de 27 años de edad, con un cuerpo extraño retenido en la cavidad abdominal, olvidado durante una cesárea de urgencia. El diagnóstico y tratamiento conllevaron a una

reintervención quirúrgica para extraer el cuerpo extraño” [2]. Otro caso similar de “una paciente femenina de 23 años de edad que se le había practicado hacía 5 meses una cesárea de urgencia, se le diagnosticó un tumor palpable en fosa iliaca derecha. Se le realizó TAC abdominal simple donde se evidenció extensa lesión hiperdensa de masa que mide 10,7x5,5 cm con pared bien definida y artefacto de alta densidad en su interior” [3]; casos que reflejan un error humano donde la persona responsable es el instrumentador quirúrgico cuya función es suministrar los instrumentos necesarios para la operación, por ello se realiza un conteo antes y después del procedimiento, las funciones del instrumentador quirúrgico son muy amplias y complejas ahora añadiendo la presión y la responsabilidad que se lleva dentro del quirófano hace vulnerable al profesional a cometer este tipo errores al momento de asociar el nombre a cada instrumento o diferenciarlos entre ellos dada la gran cantidad de instrumentos necesarios para realizar una operación médica, además la actividad repetitiva de suministrar y recibir los instrumentos quirúrgicos es extenuante cuando al mismo tiempo se tiene que estar realizando más tareas adicionales.

Objetivo general

Diseñar y fabricar un sistema robótico con inteligencia artificial para la asistencia en instrumentación quirúrgica.

Objetivos específicos

1. Diseñar un sistema robótico tridimensional utilizando un software de Diseño Asistido por Computadora (**CAD**) para la manipulación eficiente de la

instrumentación quirúrgica.

2. Fabricar las piezas del sistema robótico y ensamblar los componentes mecánicos y electrónicos con el fin de obtener un modelo a escala.
3. Aplicar algoritmos de inteligencia artificial para la identificación de los instrumentos quirúrgicos.
4. Validar el rendimiento del sistema robótico y de los algoritmos de inteligencia artificial implementados.

Alcance

El desarrollo del sistema robótico en asistencia quirúrgica está compuesto por dos partes esenciales; la primera, consta de un robot antropomórfico de 5 Grados de libertad (**GDL**), la segunda parte hace referencia al procesamiento de la información de las variables principales como el reconocimiento de objetos y la ejecución de comandos de voz específicos. El diseño del robot contempla la capacidad de los actuadores cuyo torque máximo es de 11 Kg cm ; y el proceso de fabricación de las piezas, para este caso es la impresión en Tres dimensiones (**3D**) la cual tiene un volumen de impresión de $220 \times 220 \times 260\text{ mm}^3$; para el reconocimiento de imágenes se emplea algoritmos de inteligencia artificial que tiene un alto grado de generalización comparado con el tratamiento de imágenes y segmentación tradicional aunque tiene como requisito la utilización de un potente hardware para su entrenamiento, y también la obtención de un dataset adecuado puede generar alteraciones en el resultado, de tal manera que solo se trabaja con las herramientas quirúrgicas básicas que son necesarias en cualquier tipo de intervención. Para los

comandos de voz se hace la comunicación entre la interfaz gráfica y un asistente de voz. La aplicación de un sistema inteligente en asistencia quirúrgica se enfoca hacia el desarrollo experimental donde solo se verificará aspectos funcionales desde el ámbito académico con el fin de hacer un aporte al área de medicina, sin abarcar la normativa de salubridad, sentando las bases para poder en un futuro integrar el sistema al equipo de trabajo médico, con la capacidad de interactuar y asistir en intervenciones quirúrgicas reales.

Índice general

Glosario de términos	v
Abreviaturas	vi
Lista de figuras	xv
Lista de tablas	xx
1 Marco teórico y estado del arte	1
1.1 Marco de referencia	1
1.1.1 Redes neuronales convolucionales	1
1.1.2 Diseño y Fabricación de Brazo Robótico	5
1.2 Estado del arte	8
1.2.1 Trabajos internacionales	8
1.2.2 Trabajos nacionales	20
2 Metodología y plan de trabajo	22
2.1 Análisis de la idea	22
2.1.1 Rutina quirúrgica sin implementación del sistema robótico .	22
2.1.2 Implementación del sistema robótico en el procedimiento quirúrgico	25
2.2 Proceso en el desarrollo del producto	25
2.3 Identificación de los requerimientos del sistema	28
2.4 Aplicación de técnicas de desenvolvimiento de producto	31

2.4.1	Metodología Mudge	31
2.4.2	Análisis de Pareto para Metodología Mudge	32
2.4.3	Despliegue de la función de calidad	34
2.4.4	Análisis de Pareto para Metodología QFD	42
2.4.5	Conclusiones obtenidas del desenvolvimiento del producto	43
2.5	Identificación de los recursos necesarios	48
2.6	Cronograma	52
3	Diseño electrónico	56
3.1	Características relevantes de los componentes electrónicos	56
3.2	Planteamiento de conexión	58
3.3	Diseño CAD para ensamble electrónico	63
4	Diseño mecánico	67
5	Ensamble y calibración	82
5.1	Manufactura por impresión 3D	82
5.2	Ensamble de componentes	85
5.3	Modelado cinemático	88
5.3.1	Cinemática directa	89
5.3.2	Cinemática inversa	92
5.4	Implementación de cinemática en software de simulación	96
5.5	Calibración del modelo simulado con la estructura física	98
6	Diseño de software	101
6.1	Integración del modelo cinemático al prototipo	101
6.2	Detectores de objetos y modelos CNN para la identificación de he- rramientas quirúrgicas	107
6.2.1	Obtención de datasets	111
6.2.2	Entrenamiento y validación con SSD	115
6.2.3	Entrenamiento y validación con YOLO v3	122

6.2.4	Punto medio de la herramienta y ángulo de orientación . . .	131
6.3	Implementación de comandos de voz	143
6.4	Diseño de la interfaz de unión entre los sistemas	144
7	Puesta en marcha	147
8	Análisis de resultados	152
	Bibliografía	155

Lista de figuras

1.1	Representación gráfica de la estructura en una CNN.	2
1.2	Modelo gráfico de un algoritmo de CNN.	3
1.3	Robot Puma 200.	9
1.4	Reemplazo de cadera humana realizada por ROBODOC.	10
1.5	Colecistectomía asistida por el robot aesop (1993).	11
1.6	Robot EndoAssist.	13
1.7	Sistema ZEUS.	14
1.8	Componentes del sistema da Vinci.	15
1.9	Imágenes de seguimiento automático a gasas quirúrgicas.	16
1.10	Matriz de confusión en el seguimiento de gasas.	17
1.11	Porcentaje de asertividad de cada CNN en diferentes condiciones.	18
1.12	Seguimiento de herramientas quirúrgicas mediante CNN	18
1.13	Seguimiento implementando CNN en cascada.	19
1.14	Diseño CAD de robot enfocado al análisis bacteriológico.	20
2.1	Área física del quirófano	23
2.2	Lista de verificación de la seguridad en cirugía.	24
2.3	Diagrama de flujo del proceso en el desarrollo del producto.	27
2.4	Diagrama de Pareto para la metodología de Mudge.	34
2.5	Despliegue de la función de calidad.	36

2.6	Requerimientos del usuario en el QFD.	37
2.7	Requerimientos del sistema en el QFD.	38
2.8	Análisis competitivo en el QFD.	39
2.9	Relación entre requerimientos del usuario y del sistema en el QFD.	40
2.10	Cuantificación de los requisitos del sistema en el QFD.	41
2.11	Diagrama de Pareto para QFD.	43
2.12	Estructura de descomposición del trabajo.	53
3.1	Diagrama de suministro eléctrica.	59
3.2	Diagrama de control electrónico para servomotores.	60
3.3	Diagrama de control electrónico para herramientas de visualización.	61
3.4	Diseño CAD de base para ensamble electrónico.	64
3.5	CAD del diseño electrónico explosionado.	65
3.6	CAD del diseño electrónico ensamblado.	66
4.1	CAD de vista explosionada para ensamble del 1.er gdl.	68
4.2	Ensamble CAD del 1.er gdl	70
4.3	CAD de vista explosionada para ensamble del 2.do,3.er y 4.to gdl.	71
4.4	Ensamble CAD del sistema con 4 gdl.	73
4.5	CAD de vista explosionada para ensamble del 5.to gdl.	74
4.6	Explosionado del efector final mecánico.	76
4.7	Funcionamiento del efector final mecánico	76
4.8	Explosionado del efector final de tipo magnético.	78
4.9	CAD del funcionamiento del efector magnético.	79
4.10	Dimensionamiento del prototipo robótico.	80
4.11	CAD de vista explosionada entre los gdl y base de electrónica.	81
4.12	CAD de prototipo robótico.	81
5.1	Piezas importadas en formato STL a software Cura	83

5.2	Ensamble físico de la base del robot en una vista posterior.	85
5.3	Ensamble físico de la base del robot en una vista frontal.	86
5.4	Cuerpo y base del robot antropomórfico.	87
5.5	Robot antropomórfico completamente ensamblado.	88
5.6	Estructura del robot antropomórfico simplificada.	89
5.7	Matrices de Denavit-Hartenberg para robot antropomórfico.. . . .	91
5.8	Punto de la muñeca en la estructura simplificada del robot.	92
5.9	Desacoplo cinemático en la estructura simplificada.	93
5.10	Sistemas de referencia para solución de cinemática inversa.	95
5.11	Prototipo de robot antropomórfico en software MATLAB.	96
5.12	Planeación de trayectorias mediante interpolador cúbico.	97
5.13	Posición articular de cada GDL en trayectoria simulada.	98
5.14	Conexión entre brazo robótico y Blynk.	99
5.15	Diseño de aplicación en Blynk para robot antropomórfico.	100
6.1	Protocolo TCP/IP entre prototipo físico y MATLAB.	102
6.2	Flujo de información y modelo cinemático en el sistema.	104
6.3	Sistema robótico físico con modelo cinemático implementado.	105
6.4	Espacio de trabajo del sistema robótico.	106
6.5	Algoritmo para la transferencia de aprendizaje.	108
6.6	Intersección y unión en etiquetado de imágenes.	109
6.7	Gráfica de precisión con respecto al recuerdo	110
6.8	Interfaz para la toma de imágenes.	111
6.9	Muestra de dataset compuesto por 3000 imágenes.	112
6.10	Muestra de dataset compuesto por 20000 imágenes.	112
6.11	Muestra de dataset con más de un instrumento enfocado.	113
6.12	Etiquetado de imágenes mediante imageLabeler.	114
6.13	Estructura de detección SSD.	116

6.14	Descenso del error y precisión de SSD con 3000 imágenes.	118
6.15	Precisión con respecto al recuerdo en SSD con 3000 imágenes. . . .	119
6.16	Funcionamiento de detector SSD con 3000 imágenes.	120
6.17	Descenso del error y precisión de SSD con 20000 imágenes.	120
6.18	Precisión con respecto al recuerdo en SSD con 20000 imágenes. . . .	121
6.19	Funcionamiento de detector SSD con 20000 imágenes y 4 clases. . .	122
6.20	Estructura de detección YOLO.	123
6.21	Estructura de un fire en Squeezenet.	125
6.22	Descenso del error y tasa de aprendizaje de YOLO V3 enfocando un objeto en la imagen	127
6.23	Precisión con respecto al recuerdo en YOLO V3 enfocando un solo objeto.	128
6.24	Funcionamiento de detector YOLO V3 con 20000 imágenes y 4 clases.	128
6.25	Descenso del error y tasa de aprendizaje de YOLO V3 enfocando más de un objeto en la imagen	129
6.26	Precisión con respecto al recuerdo en YOLO V3 enfocando más de un objeto en la imagen.	130
6.27	Funcionamiento de detector YOLO V3 enfocando más de un objeto en una imagen.	130
6.28	Posición del instrumento quirúrgico respecto al robot en el área de trabajo.	131
6.29	Funcionamiento en la detección del objeto y su extremo.	132
6.30	Variables para la obtención de la posición y orientación del instru- mento en la imagen.	133
6.31	Posición y orientación del instrumento en la imagen.	134
6.32	Precisión con respecto al recuerdo en la detección del extremo de los instrumentos.	135

6.33	Posición del objeto en la imagen según la orientación de la cámara.	136
6.34	Obtención de posición del instrumento en el área de trabajo con cámara movable.	137
6.35	Coordenadas del centro del instrumento respecto al sistema robótico.	139
6.36	Ángulo de ajuste en la orientación del objeto.	141
6.37	Funcionamiento del sistema de detección, posición y orientación del objeto en la imagen.	142
6.38	Implementación de Google assistant al sistema.	143
6.39	Interfaz de unión en el tiempo uno.	145
6.40	Interfaz de unión en el tiempo dos.	146
6.41	Interfaz de unión en el tiempo tres.	146
7.1	Conexión para TCP/IP automática mediante firebase.	148
7.2	Software para manipular el sistema robótico.	149
7.3	Interfaces de manipulación del sistema robótico.	150
7.4	Funcionamiento del sistema robótico y de visión artificial.	151

Lista de tablas

2.1	Relación entre usuarios y ciclo de vida.	28
2.2	Necesidades de los usuarios y ciclo de vida.	30
2.3	Requisitos del sistema robótico.	31
2.4	Diagrama de Mudge.	32
2.5	Análisis de Pareto para la metodología de Mudge.	33
2.6	Análisis de Pareto para QFD.	42
2.7	Importancia de los requerimientos del usuario.	44
2.8	Especificaciones meta del sistema robótico.	48
2.9	Componentes físicos del sistema.	49
2.10	Herramientas para ensamble de prototipo.	51
2.11	Plan de trabajo en diagrama de Gant.	55
3.1	Características de los componentes electrónicos.	57
3.2	Conexiones para tarjeta esp8266.	61
3.3	Conexiones para tarjeta esp32.	62
3.4	Piezas de la base del sistema robótico.	64
4.1	Piezas que componen el 1.er GDL.	69
4.2	Partes que componen el 2.do, 3.ro y 4.to GDL.	72
4.3	Piezas que componen el 5.to GDL.	75

4.4	Partes que componen el efector final mecánico.	77
4.5	Piezas que componen el efector magnético.	79
5.1	Tiempo y material de manufactura por impresión 3D.	84
5.2	Parámetros de Denavit-Hartenberg.	90
6.1	Clases de detectores de objeto según su estructura.	108
6.2	Modelo de datos exportado de imageLabeler.	115
6.3	Arquitectura de Resnet50.	117
6.4	Arquitectura de Squeezenet.	125
6.5	Comandos de voz definidos mediante IFTTT.	144

Capítulo 1

Marco teórico y estado del arte

1.1. Marco de referencia

1.1.1. Redes neuronales convolucionales

Es una arquitectura de red para deep learning que aprende directamente de los datos, sin necesidad de extraer características manualmente. Entre las utilidades de este tipo de estructura está el reconocimiento de objetos, caras y escenas, aunque también se emplean para la clasificación de datos como audio, series temporales y señales. Para lograr la identificación, las Convolutional Neural Network (**CNN**) contienen una serie de capas especializadas con una jerarquía entre sí, las primeras capas son más simples y son capaces de reconocer líneas o curvas, pero las que detectan las formas más complejas como el instrumental quirúrgico son las capas posteriores. El funcionamiento de las CNN, al igual que las redes neuronales tradicionales y simples como el adaline, está compuesta por una capa de entrada, una de salida y muchas en medio ocultas. Estas capas son las encargadas de hacer

las operaciones necesarias para alterar los datos con el objetivo de aprender características específicas, las capas más utilizadas son la de convolución, activación o ReLu, fully connected layer, y pooling.

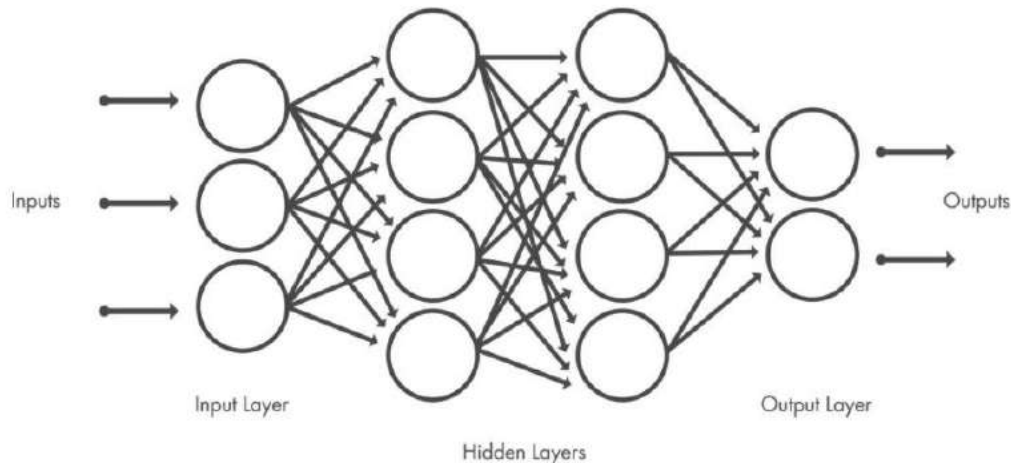


Figura 1.1: Representación gráfica de la estructura en una CNN. **Fuente:** [4]

Las capas de convolución son la parte más importante de las CNN, puesto que se encarga de la mayor carga computacional, en estas capas se aplican una serie de filtros que se encargan de modificar parámetros y aprender ciertas características de las imágenes de entrada. Este proceso se realiza tomando grupos de píxeles cercanos de la imagen de entrada e ir realizando operaciones con una matriz más pequeña llamada Kernel, una vez el kernel a pasado por toda la imagen, genera una nueva matriz de salida que será la entrada a la siguiente capa de la red. Si la imagen es a color, el filtro utilizado está compuesto por tres matrices de iguales dimensiones, después de este proceso los tres canales RED, GREEN Y BLUE se promedian para conseguir una salida en escala de grises. Para obtener varias características de una imagen es necesario un kernel especializado por cada característica; es decir, que en una capa de convolución no se necesita un solo kernel sino un conjunto de ellos. Es por esto que entre más filtros tenga será más confiable y viable la red neuronal

para la utilidad que se la desee. La unidad lineal rectificadora (ReLU) asigna en los lugares de la matriz donde hay un negativo, un cero y mantiene los positivos permitiendo un entrenamiento más rápido y eficaz. A esta capa también se le denomina capa de activación, puesto que únicamente los elementos activos pasan a la siguiente capa. La capa pooling por su parte, se encarga de reducir la carga computacional, generando que exclusivamente prevalezcan las características más importantes detectadas por la capa de convolución, aunque hay varios tipos, la más comúnmente usada es la max-pooling. La capa de fully connected layer cumple la misma función que las redes neuronales habituales, ubica y etiqueta por clases la entrada según las activaciones recibidas de las capas anteriores [4].

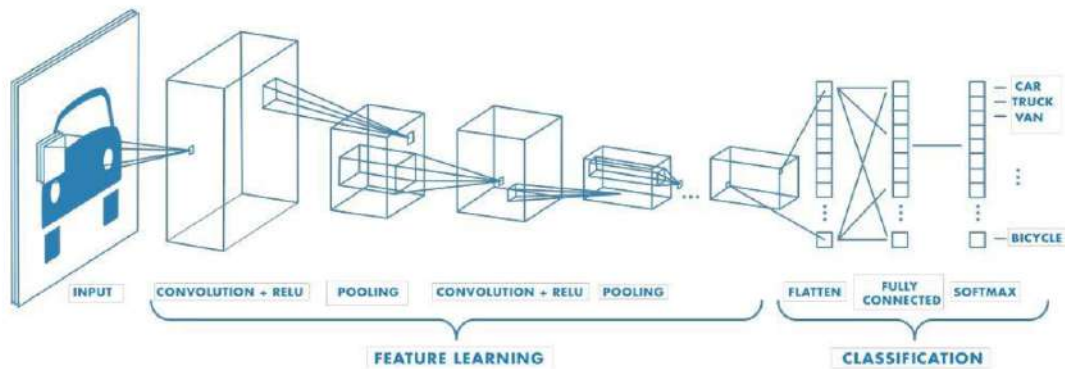


Figura 1.2: Modelo gráfico de un algoritmo de CNN. **Fuente:** [4]

AlexNet

Fue la primera red neuronal convolucional en popularizarse, ya que en el año 2012 consiguió un error del 10% menor que el segundo puesto en cuanto a la tasa de error del top 5, esta arquitectura fue creada por Alex Krizhevsky, Liya Sutskever y Geoff Hinton quienes le dieron una estructura tipo Compute Unified Device Architecture (**CUDA**). Es más sencilla en su arquitectura con respecto a

la red ResNet, pues consta de 8 capas, cinco convolucionales y tres fully connected. El sistema inicial dispone de un dataset de un millón de fotografías y capaz de clasificar más de mil categorías de objetos.

GoogLeNet

Fue la red neuronal convolucional presentada por Google ganadora del ImageNet Large Scale Visual Recognition Challenge (**ILSVRC**) en el año 2014, consiguiendo un error top 5 de 6.67 %. Esta red neuronal está compuesta por 27 capas, incluyendo las de pooling, una innovación de esta red está en que sustituye las capas de fully connected al final por average pooling que tiene un efecto beneficioso en la carga computacional y en la mejora de la precisión elevándolo en un 0,6 %. En un inicio fue diseñada para que se pudiera implementar en los teléfonos móviles, por esta razón incluye una capa de convolución de dimensiones 1x1 en el medio, con esto consigue poder incluir más capas. En un comienzo fue diseñada para trabajar con un banco de 6000 imágenes de 10 clases llamado CIFAR-10. Para poder entrenarla con un dataset diferente es necesario modificar las últimas capas al igual que Alexnet y Resnet.

ResNet-50

Al igual que Google, Microsoft también introduciría su CNN llamada Residual Network que ganaría en el año 2015 la competición de ILSVRC. Lo especial de ResNet es la capacidad de entrenar cientos o miles de capas, convirtiéndose en la red más profunda hasta la actualidad, esto lo logro tras solucionar el error del gradiente, que consiste en que el gradiente tiende a disminuir hasta llegar a cero a medida que aumentan el número de capas consiguiendo que la red dejara de

aprender. Para eludir este error, Microsoft introdujo el concepto de skip connections, que sería un salto de una o un par de capas conectadas directamente con la salida.

1.1.2. Diseño y Fabricación de Brazo Robótico

Diseño Mecánico Mediante Software CAD

Solidworks nos ofrece varias soluciones para el modelado de piezas y ensamblajes en 3D y planos en Dos dimensiones (**2D**), además de crear y diseñar, nos permite simular nuestro producto y visualizarlo sin que se realice físicamente. También podemos calcular parámetros como la masa y el volumen. Una gran ventaja de este software es que nos brinda una herramienta en la cual podemos simular las cargas que se le va a aplicar a la pieza para así prever la cinemática, dinámica, estrés, deflexión, vibración, temperatura o flujo en nuestro diseño también por este método podemos asegurarnos del factor de seguridad que le asignamos a nuestro diseño. En el mercado existen varios softwares como Autocad Inventor, Solid Edge y SolidWorks. La mayoría de estos sistemas son muy similares entre sí. Aunque sin duda uno de los más intuitivos es SolidWorks, lo que permite un aprendizaje en menor tiempo [5].

Diseño Electrónico

En el diseño electrónico existen varios componentes de gran importancia para el diseño de un brazo robótico, entre ellos son los microcontroladores, servomotores, motor paso a paso, y convertidores dc/dc. Entre los microcontroladores más populares están los de la serie Arduino, siendo uno de los más básicos el Arduino

Uno, aunque existen gran variedad de ellos, basados en el chip ATMEL, el cual se escribe mediante un software de la misma marca con el nombre de Arduino IDE. También existen placas embebidas como es el caso del ESP32 creado por la empresa Espressif Systems haciendo una analogía con el ESP8266, incluye wi-fi y bluetooth entre otras características. Una parte importante de los productos industriales, igual que en los comerciales, es el servomotor y su campo de aplicación es muy amplia, en la robótica usualmente se utiliza uno en cada junta por su precisión en el momento de realizar un giro a un ángulo deseado, su control se realiza por medio de la Modulación por ancho de pulso (**PWM**), la electrónica del servomotor responde al ancho de la señal modulada. Para un diseño electrónico donde existen elementos o dispositivos en los que varía el rango de voltaje para un adecuado funcionamiento, se hace indispensable elevar o reducir el voltaje según sea conveniente con el fin de dar solución a estos inconvenientes. Existen en el mercado gran variedad de convertidores dc/dc, ya sea elevador de voltaje (tipo boost) o reductor (tipo buck) [6].

Modelo cinemático de brazo robótico

Cinemática Directa El problema matemático de la cinemática directa consiste en encontrar la dirección y posición final del manipulador del robot, en función de los valores traslacionales y rotativos de las articulaciones. Existen varios métodos para dar solución a este problema, pero el más utilizado hace uso de las matrices de transformación homogéneas, junto con la representación de Denavit Hartenberg. Esta forma de desarrollar tiene una ventaja con respecto a la forma geométrica y es que en el proceso se conoce la posición de cada una de las articulaciones con

respecto a un sistema de coordenadas fijo.

$${}^{i-1}A_i = \begin{bmatrix} \cos(\theta_1) & -\cos(\alpha_1)\sin(\theta_1) & \sin(\alpha_1)\sin(\theta_1) & a_1\cos(\theta_1) \\ \sin(\theta_1) & \cos(\alpha_1)\cos(\theta_1) & -\sin(\alpha_1)\cos(\theta_1) & a_1\sin(\theta_1) \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.1)$$

Para esto se encuentran los parámetros necesarios que serán reemplazados en la matriz de Denavit Hartenberg tantas veces como grados de libertad tenga.

$$T = {}^0A_n = {}^0A_1{}^1A_2{}^2A_3\dots{}^{n-1}A_n \quad (1.2)$$

al final las matrices resultantes se multiplicarán, como se indica en la ecuación anterior, con el fin de dar como resultado la matriz de transformación homogénea, que brinda información sobre la posición y orientación del efector final visto desde el sistema de referencia cero.

Cinemática inversa La cinemática inversa denota el proceso contrario a la cinemática directa, donde tenemos la posición y dirección del efector final Px, Py, Pz y basado en estos parámetros necesitamos calcular los valores articulares necesarios para situar el manipulador, existen diversas formas de solucionar el problema, pero lo que usualmente se propone es un conjunto de ecuaciones cerradas y a través de relaciones matemáticas determinar los puntos adecuados para seguir la trayectoria [7].

Manufactura por medio de Impresión 3D

La impresión 3D es una tecnología capaz de crear un objeto tridimensional mediante el apilamiento de capas de plástico partiendo de un modelo digital. Los primeros equipos y materiales de construcción de una impresora 3D se diseñaron en 1976, aunque hasta 1984 varios proyectos fueron patentados basados en el proceso de estereolitografía, en este año también se estandarizó un sistema para generar objetos en tercera dimensión mediante la implementación de un patrón del objeto a formar, que dio lugar al formato STL. Entre los materiales más utilizados en con esta tecnología están el PLA, Laywoo-D3, ABS, HIPS, PET y el TPE [8].

1.2. Estado del arte

1.2.1. Trabajos internacionales

Puma 200

Es un brazo robótico programable de manipulación universal, que en sus inicios fue diseñado y desarrollado entre los años 1960 y 1970 para General Motors a cargo de Victor Scheinman, en la empresa robots Unimaton, inspirado en investigaciones y proyectos generados por él, mientras se encontraba en la Universidad de Stanfors.

Debido al éxito generado por el diseño, la empresa Unimation fabrico robots PUMA hasta que finalmente la empresa fue adquirida por Westinghouse en 1980, aunque en la década de 1980 el fabricante Nokia Robotics saco alrededor de 1500 ejemplares al mercado y vendió su división de robótica en 1990. En el año 2002 el prototipo original del robot PUMA llego a manos del Museo Nacional de Historia

Americana de la Institución de Smithsonian tras una donación de Robotics and Welding.



Figura 1.3: Robot Puma 200. **Fuente:** Gokce Anil, Robotic Thoracic Surgery, 2020

Tres modelos son los que más se rigen al modelo inicial, estos son los de las series 200, 500 y 700. Diferenciándose principalmente entre ellos por su tamaño, siendo el de la serie 700 el de mayores dimensiones, capaz de implementar en trabajos como pintura o soldadura, la serie 500 es la serie más reconocida y adquirida, pues su tamaño, con aproximadamente 2 metros de altura, permitía una gran variedad de aplicaciones, por último, la serie 200 es una serie de mesa [9].

Robodoc

Es un innovador dispositivo desarrollado con el fin de realizar cirugías de reemplazo de cadera y rodilla que comenzó en la Universidad de California y en el

Centro de investigación Tomás Watson de IBM en 1986 con la intención de crear una máquina más precisa para procesos como el reemplazo de articulaciones. Se logró la culminación de este proyecto en 1992, convirtiéndose en el primer robot quirúrgico de Estados Unidos.



Figura 1.4: Reemplazo de cadera humana realizada por ROBODOC. **Fuente:** [10]

ROBODOC revoluciono la artroplastia, ya que al realizarlo a mano no siempre es preciso y existe la posibilidad de errores como que se atrofie un hueso. El sistema de Robodoc ayuda a prevenir estas complicaciones a través de imágenes tridimensionales que facilitan una planificación antes de la cirugía. Robodoc ha hecho, aproximadamente, 28000 intervenciones quirúrgicas desde que fue aprobado en el año de 1998 por la Administración de Alimentos Y medicamentos de Estados Unidos. Actualmente, se encuentra un prototipo de 1989 en el Museo Nacional de Historia Estadounidense del Smithsonian y es una pieza relevante para la colección de Ciencia y Medicina del museo [10].

Aesop

El sistema endoscópico automatizado para posición fue desarrollado en 1993, año en el que Yulun Wang fundó Computer Motion, el diseño consta de un brazo robótico guiado por voz y equipado con un endoscopio, a este proyecto se le denominó AESOP.



Figura 1.5: Colecistectomía asistida por el robot aesop (1993). **Fuente:** Ramos Adrián, Robotic surgery, 2003

En la cirugía laparoscópica tradicional el cirujano pierde gran parte de campo visual. Ya que la cámara que transmite la imagen es manipulada por un asistente, la solución fue devolverle al cirujano el control de la lente por medio de un brazo robótico dirigido por pedales o manualmente. Proyecto denominado AESOP 1000, Este sistema fue aprobado en 1994 por la Administración de Drogas y Alimentos, dos años después de aprobación del primer AESOP salió al mercado la referencia

del AESOP 2000. En este diseño se introdujo el control por voz, dándole al cirujano más comodidad. En el diseño posterior, AESOP 3000, se le adicionó una articulación más al soporte de video aumentado los grados de libertad; finalmente, se lanzó una cuarta referencia de este diseño llamado el AESOP HR implementado varias funciones adicionales, como la posición de la mesa de operación y el control del sistema de iluminación de la sala de operación. El modelo AESOP requiere de registro previo de la voz, tiene una capacidad de 23 órdenes diferentes que se reciben y transmiten por medio de un micrófono usado por el cirujano durante el procedimiento quirúrgico.

El objetivo de este sistema es mejorar la capacidad de visión del cirujano, reducir personal médico en el quirófano y evitar errores humanos.

Endoassist

Es un dispositivo que se ajusta al puerto donde recibirá el laparoscopio a través de dos luces láser que se encuentran en el brazo, y debe ser ubicado junto a la mesa donde se realizará dicha cirugía. Sin embargo, hay que tener en cuenta que al ubicarlo se deberá activar el freno y estar sobre ruedas. Es importante que esté en un punto central, ya que debe estar coordinado con los mismos movimientos de la mesa. El robot EndoAssist está controlado por los movimientos de la cabeza del médico a través de un dispositivo infrarrojo, donde se transmite a un receptor colocándose al frente en la línea de visión que guía la cirugía. Es así como los movimientos serán activados siempre y cuando se presione un pedal para activar el dispositivo y evitar que no se efectúen movimientos involuntarios. De esa manera su brazo estará siempre sujeto al laparoscopio que se puede desmontar que se encuentra esterilizado y atornillado al dispositivo [11].



Figura 1.6: Robot EndoAssist. **Fuente:** [12]

Zeus

Es un proyecto desarrollado por Computer Motion para mejorar la calidad del proceso quirúrgico que lleva a cabo el cirujano, el sistema se encuentra integrado físicamente por dos subsistemas, el primero consta de una consola de control conectada a tres brazos robóticos unidos a la mesa de operaciones, formando un robot teleoperado similar a un sistema AESOP.

El brazo que soporta el endoscopio está controlado por voz, los otros dos brazos cuentan con cuatro grados de libertad y son los encargados de sostener una gran variedad de instrumentos, que son maniobrados por el cirujano por medio de joysticks desde la consola.

Para enlazar los dos subsistemas se emplea un software informático, el cual puede filtrar errores como son los pequeños temblores del cirujano.

La imagen obtenida por medio del endoscopio se muestra en una imagen bidi-

mensional o a través de anteojos polarizados, donde se visualiza la imagen tridimensional en una pantalla equipada con una matriz polarizadora activa.

Se mostró las ventajas de Zeus después de realizar numerosas pruebas con cadáveres y en animales. Finalmente, en 1999 se hizo una experiencia clínica exitosa en los procedimientos cardiovasculares y posteriormente el sistema se empleó para diversos tipos de cirugías, y en 2001 se haría la primera colecistectomía laparoscópica transcontinental donde un cirujano ubicado desde su consola en New York opero a un paciente ubicado en Estrasburgo.



Figura 1.7: Sistema ZEUS. **Fuente:** [13]

Da Vinci

En 1995 se consolidó el desarrollo de un sistema capaz de reproducir los movimientos de la mano del cirujano, junto con el sistema de retroalimentación de fuerza y visión 3D del prototipo da Vinci, tras la modificación que realizó la empresa Intuitive Surgical del sistema quirúrgico teleoperado desarrollado por Phil Green.

Himpens y Cardiere en 1997 ejecutó la primera colecistectomía asistida con el

sistema resultante de la modificación denominado MONA en un paciente de 72 años. El efecto de visión 3D le da al cirujano la impresión de estar inmerso en el campo operatorio.



Figura 1.8: Componentes del sistema da Vinci. **Fuente:** [13]

Un estudio realizado por Nissen, en el que se comparó el proceso quirúrgico de laparoscopia desarrollado por da Vinci y el método tradicional, confirmó la viabilidad del sistema da Vinci, pero también mostró las mejoras que se deberían hacer al prototipo puesto, que el tiempo que empleaba era más prolongado que de manera tradicional, debido a esto el próximo modelo de da Vinci incorporaría soluciones como un sistema informático para el control de la consola quirúrgica y mejora la transición entre los modos de control.

Similar al sistema Zeus, da Vinci en sus primeros diseños en el campo operatorio del paciente se ubicaban tres brazos robóticos, uno de ellos sosteniendo la cámara y los dos restantes el instrumental. En los modelos más modernos como da Vinci S y da Vinci Si utilizan 4 brazos robóticos, tres de ellos con la instrumental imitando los movimientos de la muñeca del cirujano por medio de 7 grados de libertad en el extremo. Para la transmisión de movimiento se implementaron dos joysticks que

controlan la posición de la lente y de los instrumentos, este sistema es capaz de filtrar el temblor en las manos del cirujano [12].

Seguimiento de Gasas Quirúrgicas

Francisco Javier Santos [14] de la universidad de Valladolid, en su trabajo de grado de septiembre de 2020, publica un sistema de seguimiento automático de gasas para operaciones de cirugía laparoscópica con el fin de evitar errores como el oblito quirúrgico, el cual trata de una retención involuntaria de material quirúrgico, en el interior del cuerpo del paciente, aunque es un tipo de accidente poco frecuente puede tener como mayor consecuencia la muerte del paciente.

El algoritmo desarrollado por Francisco Santos tiene tres lineamientos importantes por cumplir siendo la precisión, sensibilidad y tiempo de procesamiento. Para esto realiza diferentes pruebas haciendo uso de tres tipos de estructuras de redes convolucionales ya establecidas como son Alexnet, Resnet y GoogleNet.

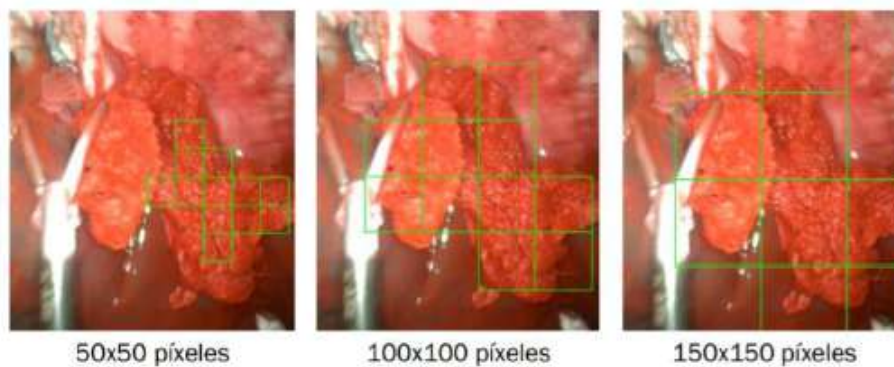


Figura 1.9: Imágenes de seguimiento automático a gasas quirúrgicas. **Fuente:** [14]

Para el desarrollo del sistema utilizó el software MATLAB. En el cual se introduce un vídeo donde es separado en fotogramas, una vez obtenidos los fotogramas

se fragmentan a un tamaño de 100 x 100 píxeles y se guardan en la memoria del hardware y se clasifican uno por uno, en seguida se detectan los fotogramas en los que se ha encontrado gasas y se recuadran de color verde. Una vez finalizada la clasificación se unen los fotogramas y se crea el vídeo de seguimiento de gasas.

Para el dataset empleó imágenes captadas de vídeos en los cuales hace uso de un endoscopio en un entorno simulado con viseras de animal y gasas quirúrgicas.

En cuanto a los resultados, la matriz de confusión para el trabajo de Francisco fue la siguiente:

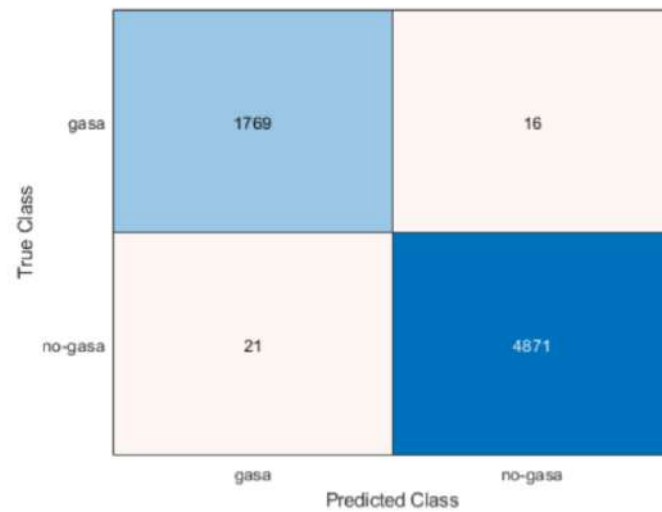


Figura 1.10: Matriz de confusión en el seguimiento de gasas. **Fuente:** [14]

Demostrando con lo obtenido la viabilidad del sistema y también el rango de precisión del mismo. Clasificando la precisión según el modelo de estructura de CNN y clases de gasa (sin sangre, parcialmente manchadas, con sangre) obtuvo:

	Alexnet	ResNet-50	GoogLeNet
Gasas sin sangre	99,44%	99,72%	99,58%
Gasas manchadas de sangre	98,87%	99,53%	98,40%
Fondo	99,57%	99,69%	99,97%

Figura 1.11: Porcentaje de asertividad de cada CNN en diferentes condiciones.
Fuente: [14]

Detección de instrumentos quirúrgicos para cirugía laparoscópica

En mayo del 2021, Daniel Sanz Gobernado [15], publicó su trabajo de grado titulado Detección de instrumentos quirúrgicos en imágenes para cirugía laparoscópica robotizada, para la universidad de Valladolid, donde hizo uso de algoritmos como YOLOv2 y YOLOv3 con el fin de detectar material quirúrgico en el interior del paciente y asistir al cirujano durante la operación.

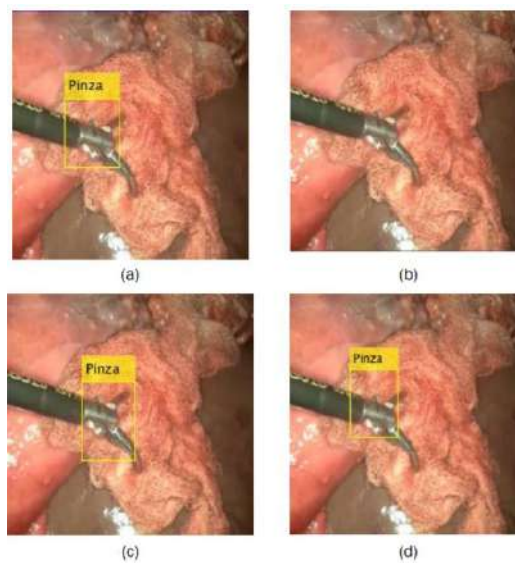


Figura 1.12: Seguimiento de herramientas quirúrgicas mediante CNN. **Fuente:** [15]

Sanz llega a la conclusión de que, aunque ResNet tiene un buen resultado

para la detección de pinzas como para gasas, en una implementación de quirófano se vería prudente utilizar YOLO, puesto que es capaz de detectar objetos más pequeños con gran precisión.

CNN en cascada para la detección de instrumentos quirúrgicos

En la revista de investigación IET (The institution of Engineering and Technology), la universidad de Shandong de la República popular de china en 2015 propuso a través de Zijijian, Tongbiao, Faliang y Xiaolin la utilización de redes convolucionales en cascada para la detección de instrumentos quirúrgicos [16], dado que los sistemas de aprendizaje actuales se limitan a la detección de una sola herramienta y sufren de baja velocidad en la clasificación. Aplican una red de reloj de arena y una red VGG modificada para predecir conjuntamente la localización, la primera CNN genera mapas de calor de detección que representan la ubicación de las áreas de información sobre la presencia de herramientas, y la segunda genera un recuadro limitador para las áreas de información.



Figura 1.13: Seguimiento de herramientas quirúrgicas implementando CNN en cascada. **Fuente:** [16]

Con la red neuronal generadora de mapas de calor aborda una ubicación aproximada del instrumento y simplifica el problema en una clasificación binaria de cada mapa de calor. Este resultado se convierte en la entrada de la red neuronal delimitadora que aplica una estructura VGG-16 modificada y extendida usada originalmente para la clasificación. Esta red contiene seis bloques convolucionales, seis capas de agrupación y tres capas completamente conectadas.

1.2.2. Trabajos nacionales

Implementación de brazo robótico en el análisis bacteriológico

En el año 2012 se publicó un artículo el cual presenta un brazo robótico de 5 grados de libertad para la automatización del proceso de análisis bacteriológico en laboratorios, dicho proyecto se plantea con un enfoque formativo, y una de sus prioridades es el manejo de un bajo coste en el producto final.

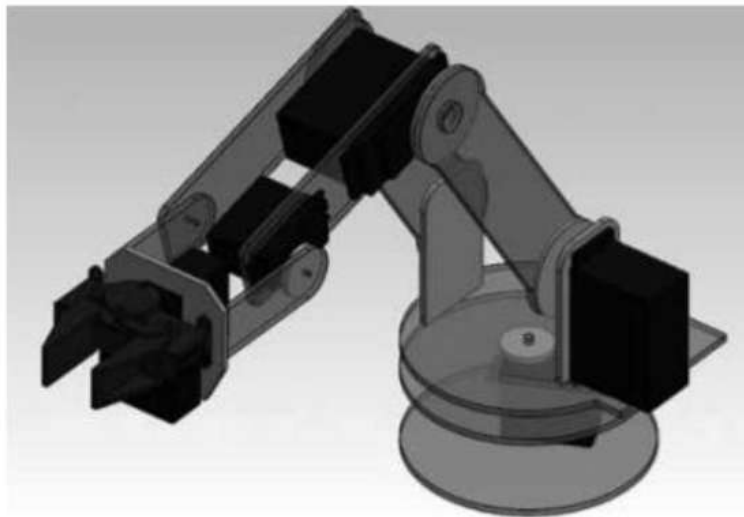


Figura 1.14: Diseño CAD de robot enfocado al análisis bacteriológico. **Fuente:** [17]

Para el desarrollo del robot antropomórfico incorporan una interfaz desarrollada mediante el software MATLAB, y tiene la finalidad de facilitar la entrada de información. En segunda instancia hacen utilidad de un sistema embebido como lo es Arduino, con el cual se permite la comunicación entre la plataforma y el brazo robótico, para así facilitar la manipulación de los objetos.

Para el diseño CAD de este robot se usó el empaquetado de SolidWorks en el que de manera sencilla se pueden obtener diferentes características antes de la fabricación, como por ejemplo el máximo alcance del brazo y la forma en cada una de las articulaciones. Posteriormente, para la fabricación el material base usado fue acrílico [17].

CNN para el diagnóstico de neumonía en imágenes radiológicas

Se desarrolló por medio del lenguaje de programación Python un modelo de machine learning para identificar mediante imágenes radiológicas si un paciente posee neumonía, para la construcción del conjunto de datos de entrenamiento o dataset se parte de una base de datos ya existente proporcionada por el Instituto de la Salud de Estados Unidos.

Para disminuir el poder computacional requerido se implementa un resize a las imágenes de entrada disminuyendo el tamaño a 256x256 píxeles, esto a su vez permite también aumentar la cantidad de imágenes con las que se va a entrenar la red neuronal.

En el modelado de la CNN se utiliza una herramienta de uso libre llamada TensorFlow que como resultado genera una exactitud superior a 70 % y una exactitud de entrenamiento igual al 96 %, aunque su precisión no es tan alta, se comprueba el funcionamiento del modelo [18].

Capítulo 2

Metodología y plan de trabajo

2.1. Análisis de la idea

2.1.1. Rutina quirúrgica sin implementación del sistema robótico

El procedimiento quirúrgico inicia con el ingreso del equipo médico al área negra; donde se encuentran los vestidores en los que el personal deja accesorios personales y se viste con el uniforme quirúrgico. Para enseguida dirigirse al área gris, en el área gris se realiza el correcto lavado de las manos con la respectiva técnica. Luego los circulantes ingresan al área blanca, área en la cual se sitúa el quirófano, ahí se verifica el estado de la sala de operación, sus equipos y la funcionalidad de las herramientas eléctricas. Realizado la verificación se procede al ingreso del paciente y se realiza la supervisión de entrada acorde con la lista de seguridad quirúrgica de la Organización Mundial de la Salud (OMS).

Tras la adecuada verificación, se ubica al paciente en la mesa de operación y se le suministra anestesia. Mientras que el instrumentador quirúrgico alista la mesa de mayo y la mesa de riñón, sin embargo, antes de la entrada del cirujano y de los ayudantes, los circulantes limpian con técnicas específicas el área a intervenir en el paciente.

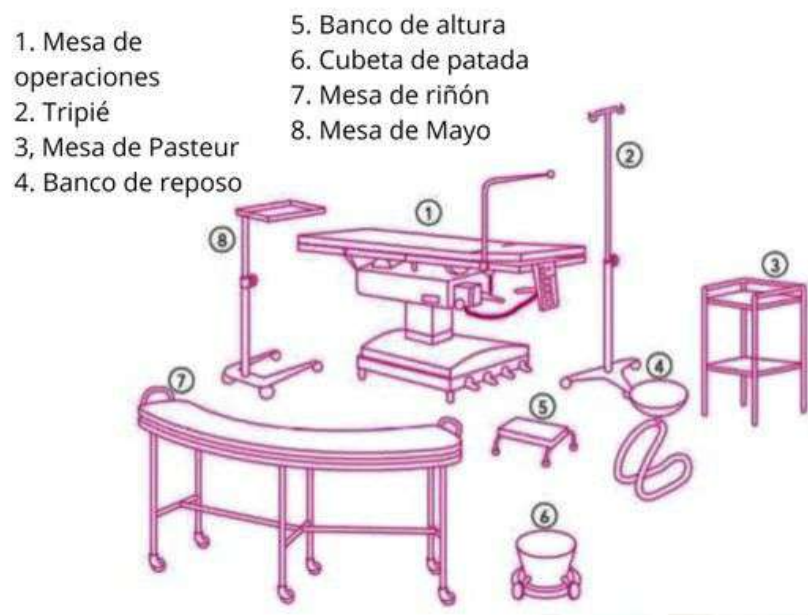


Figura 2.1: Características y clasificación del área en el quirófano. **Fuente:** Yoa-molaenfermeriablog.com, Características de quirófano, 2018

El instrumentista quirúrgico realiza el conteo del material sobre la mesa de riñón de acuerdo a los lineamientos de cirugía segura de la OMS. Y lo ordena por secciones según su función y tiempo en el que se le da uso. También divide esta mesa para ubicar el material contaminado por el procedimiento.

Los ayudantes delimitan la zona del cuerpo la cual se va a intervenir y se ubican en su respectiva posición para dar inicio a la operación, el instrumentista localiza la mesa de mayo sobre el paciente y cierra colocando la mesa de riñón al pie de la

Posteriormente, se sitúan algunos instrumentos según su utilidad en la mesa de mayo y se da inicio al procedimiento. Durante la cirugía, el cirujano o los ayudantes solicitan de forma verbal cada material. Al finalizar la intervención, el instrumentista hace un conteo de todas las herramientas quirúrgicas para evitar inconvenientes por descuidos como los oblitos y se realiza la supervisión de salida según lo indica la lista de seguridad quirúrgica de la OMS.

[illegible]

24

2.1.2. Implementación del sistema robótico en el procedimiento quirúrgico

Para lograr un producto que como objetivo principal es la asistencia en la instrumentación quirúrgica y su justificación está dada en la prevención de oblitos además de liberar carga ocupacional para el personal médico nos enfocamos en el momento en que suelen suceder estos inconvenientes, es decir en el transcurso de la intervención quirúrgica, es por ello que se plantea para ser parte del sistema un robot antropomórfico con la habilidad de sujetar el instrumental médico de la mesa de mayo y facilitarlo a quien lo solicite, además de un sistema de visión artificial para identificar cada una de las herramientas.

El sistema en el transcurso de su funcionamiento debe ser capaz de guardar en un historial, datos importantes relacionados con el procedimiento quirúrgico, dentro de estos datos están los indicados por la lista de seguridad quirúrgica de la OMS, también información adicional como los tiempos de la cirugía y fecha de la misma. Con el fin de aumentar la confiabilidad en el seguimiento de la reseña clínica y librar al personal de estas tareas.

2.2. Proceso en el desarrollo del producto

Dado que, al momento de ejecutar un proyecto, este implica muchas actividades a realizar, el Proceso de Desarrollo de Productos (**PDP**) se basa en el flujo de información ordenada entre las diferentes áreas que comprenden el desarrollo de un producto, y con ello, lograr comprender los vínculos entre cada una de ellas.

En los casos empresariales se hablaría del flujo de información entre empresa, mercado, proveedores y fuentes de información, en ámbitos académicos hablaremos

del flujo de información que conecta a los usuarios, encargados del desarrollo del proyecto, proveedores de los recursos, conocedores del área de enfoque e información obtenida de la investigación.

El PDP integra actividades que van desde la planificación de un producto hasta su retiro, considerando todo su ciclo de vida, buscando transformar el lenguaje en el cual se solicitan unas necesidades para convertirlos en requisitos de un proyecto, el cual debe de ser capaz de solventar esa necesidad. el PDP ayuda a disminuir tiempos, problemas técnicos e inconvenientes con el planteamiento de diseño. Puede anticipar problemas que de no dar solución pueden incurrir en gastos innecesarios y retrasos.

En este proyecto se plantea un modelo de PDP el cual seguirá la secuencia presentada en la figura 2.3 la que inicia con la entrada de la idea en la que se plantea el enfoque principal del producto.

En el desarrollo del sistema se pretende comenzar por dar claridad en el área conceptual, identificando las necesidades y convirtiéndolas en requerimientos del proyecto, en seguida aplicar técnicas de desenvolvimiento para el análisis de datos con los cuales se pueda tomar decisiones técnicas y de diseño.

Con una propuesta de diseño sólida podemos elaborar un esquema el cual nos indique los recursos necesarios para la construcción del sistema y con el que también podamos evaluar el costo que conllevara la realización de dicho proyecto.

En este punto es necesario la elaboración de un cronograma específico y técnico para el desarrollo del prototipo.

Al finalizar el proyecto y después de analizar los resultados obtendremos como salida las conclusiones sobre la funcionalidad y elaboración del prototipo tangible.

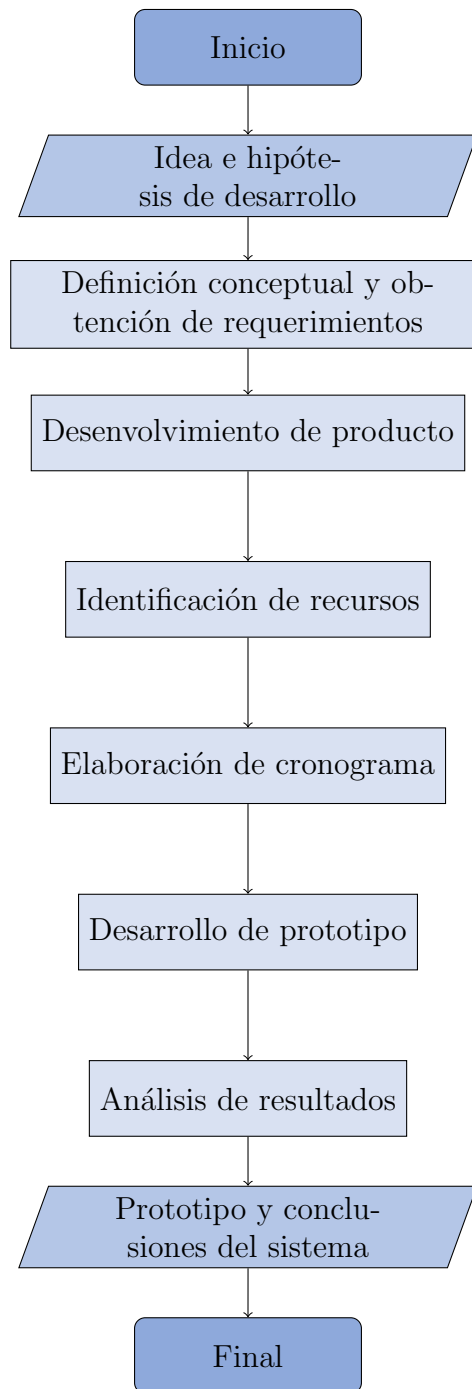


Figura 2.3: Diagrama de flujo del proceso en el desarrollo del producto. **Fuente:** Elaboración propia

2.3. Identificación de los requerimientos del sistema

A fin de lograr una correcta identificación de los requerimientos del sistema, se deben determinar los usuarios que intervienen en cada una de las etapas por las que pasa el proyecto (ciclo de vida) desde el surgimiento de la idea hasta que deja de ser compromiso de los encargados del producto. Cada una de las etapas propuestas en el ciclo de vida se caracterizan por ser secuenciales y jerárquicas, luego de esto se organizan y se caracterizan las necesidades de cada usuario de las que se desglosara una lista de requerimientos que debe cumplir el proyecto.

Ciclo de vida	
Etapas	Usuarios
Planteamiento e investigación	Investigadores responsables del proyecto
	Profesionales de la salud consultados
	Estudiantes y docentes
Coordinación de logística para ejecución de proyecto	Proveedores de equipo y herramientas de manufactura
	Investigadores encargados de fabricación
	Proveedores de material
Montaje	Investigadores responsables del proyecto
	Proveedores de componentes
	Estudiantes y docentes
Uso de robot	Investigadores responsables del proyecto
	Congresos y equipos de investigación
	Estudiantes y docentes
Mantenimiento y actualización	Estudiantes y docentes
	Equipo de mantenimiento
	Investigadores responsables del proyecto
Reciclaje	Equipo de reciclaje
	Estudiantes y docentes

Cuadro 2.1: Relación entre usuarios y ciclo de vida del sistema. **Fuente:** Elaboración propia

En la tabla 2.1 se listan las etapas del ciclo de vida y los usuarios correspon-

dientes a cada una de ellas en primer lugar tenemos la etapa de planteamiento e investigación en la cual se le da forma conceptual y técnica al proyecto, se realiza una investigación del estado del arte y técnicas utilizadas por productos que estén en la línea de enfoque de nuestro objetivo, se construye un prototipo virtual con los requerimientos obtenidos por el desenvolvimiento con ayuda de los profesionales de la salud consultados, se establece una ruta de acción para la manufactura del proyecto y se enumeran de forma detallada los recursos que serán requeridos en el proceso.

En segundo lugar, se tiene la coordinación necesaria para la ejecución del proyecto, es decir, se evalúan costos y disponibilidad de los recursos planteados anteriormente, de la mano de los proveedores.

A continuación, se plantea el montaje del sistema, en el cual se genera una estructura tangible del modelo virtual, dándole vida al proyecto y preparándolo para la etapa más importante que sería en la que se le da uso. En el transcurso de su empleo se requerirá un adecuado mantenimiento y actualización antes de culminar su ciclo de vida y seguir con la última fase de reciclaje.

La identificación de los usuarios en cada etapa del ciclo de vida nos permite tener un mayor enfoque de a quien va dirigido el producto y así designar una estrategia para conocer las necesidades de cada usuario según su forma de interacción con el proyecto, estas necesidades se expresan de forma lingüística y se pueden obtener mediante diferentes métodos estos pueden ser observación directa, entrevistas, grupos focales entre otros, en nuestro caso se logró conseguir estas necesidades mediante observación directa y entrevistas.

En la tabla 2.2 se muestra el resultado de la comprensión e interacción con los diferentes usuarios de cada etapa del ciclo de vida, registrando sus necesidades y aspiraciones con respecto al proyecto.

Necesidades asociadas al sistema acorde con el ciclo de vida	
Etapas	Necesidades
Planteamiento e investigación	Espacio de trabajo adecuado al interior del quirófano
	Bajo costo en fabricación
	Material resistente
	Fácil de manejar
	Incorporar modelos que detecten las herramientas
	Almacenamiento de historial con datos importantes
	Reconocimiento de comandos de voz para solicitud de material
Coordinación de logística para ejecución de proyecto	Componentes comercialmente asequibles
	Métodos de manufactura accesibles
Montaje	Fácil de ensamblar
	Fácil de transportar
	Adaptable para ejecutar diferentes tareas
Uso de robot	Estructura con alta precisión y estabilidad
	Bajo consumo de energía
	Visualmente agradable
	Alta precisión en modelos de visión artificial
	Alta precisión en reconocimiento de comandos de voz
Mantenimiento y actualización	componentes fáciles de intercambiar
	Fácil de desmontar
	Cómodo de reprogramar
Reciclaje	Estructura mayormente biodegradable

Cuadro 2.2: Relación entre las necesidades de los usuarios y ciclo de vida del sistema. **Fuente:** Elaboración propia

El principal fin de determinar las necesidades de los usuarios es encontrar los requerimientos del sistema que agradan y sorprenden de manera favorable, para así poder generar beneficios que el usuario no esperaba o no pudo expresarlas con claridad.

Con el procedimiento anterior se inició la primera fase para obtener el criterio de los usuarios del sistema, pero para pasar a un requerimiento se precisa pasar estas necesidades en un lenguaje que sea de fácil entendimiento para los diseñadores, es decir, se debe pasar esta información en términos de ingeniería técnica como se muestra en la tabla 2.3. En este punto se toman las primeras decisiones físicas sobre el proyecto que se va a realizar.

Funciones	Requerimientos del sistema
1	Fácil mantenimiento
2	Estructura resistente
3	Fácil fabricación
4	Entorno agradable
5	Visualmente agradable
6	Fácil manejo
7	Movible y fácil de transportar
8	Herramienta intercambiable
9	Alta precisión en su posición
10	Fabricación de bajo costo
11	Alta precisión en modelos de visión artificial
12	Alto reconocimiento de los comandos de voz
13	Generar base de datos con historial

Cuadro 2.3: Requisitos del sistema robótico. **Fuente:** Elaboración propia

2.4. Aplicación de técnicas de desenvolvimiento de producto

2.4.1. Metodología Mudge

Si bien anteriormente se logró identificar los requerimientos del sistema se hace necesario una optimización y jerarquización de los mismos teniendo en cuenta su importancia, para ello hacemos uso de una herramienta como lo es el diagrama de Mudge, mediante este instrumento se compara cada uno de los requisitos con los otros planteando dos preguntas entre ellos ¿Qué requisito es más importante? Y ¿cuán más importante es?

En la tabla 2.4 se hace la representación del diagrama de Mudge para el sistema

robótico a implementar, en la parte tanto superior como derecha se encuentran los requerimientos del proyecto y en la intersección entre ellos por medio del valor numérico se indica cuál prevalece en importancia y en valor alfabético muestra cuan más importante es, en la parte superior derecha se encuentra la tabla de eficiencia donde nos indica el rango y valor numérico de cada variable siendo de la letra (A) a la letra (E) y del número 5 al 1 respectivamente.

Funciones													Tabla de eficiencia A=5 Max B=4 C=3 D=2 E=1 Min		
Fácil mantenimiento	Estructura resistente	Fácil Fabricación	Entorno agradable	Visualmente agradable	Fácil manejo	Movible y fácil de transportar	Herramienta intercambiable	Alta precisión en su posición	Fabricación de bajo costo	Alta precisión en modelos de visión artificial	Alto reconocimiento de los comandos de voz	Generar base de datos con historial médico			
1	2	3	4	5	6	7	8	9	10	11	12	13	Sum	%	Requerimientos del sistema
1	1E	3C	4B	1B	6D	7E	1C	9A	1D	11A	12A	13C	10	3.71	Fácil mantenimiento
	2	2D	4C	2D	6C	7E	2C	9A	2C	11A	12A	2E	11	4.09	Estructura resistente
		3	4B	5B	6B	7D	8E	9A	10D	11A	12A	13E	3	1.11	Fácil Fabricación
			4	4C	6E	4E	4C	9A	4E	11A	12A	4C	22	8.17	Entorno agradable
				5	6B	7C	8E	9A	5D	11A	12A	13B	6	2.23	Visualmente agradable
					6	6E	6B	9A	6C	11A	12A	6C	25	9.29	Fácil manejo
						7	8D	9A	7D	11A	12A	13B	9	3.34	Movible y fácil de transportar
							8	9A	8C	11A	12A	13B	7	2.6	Herramienta intercambiable
								9	9A	9D	9C	9A	55	20.44	Alta precisión en su posición
									10	11A	12A	13D	2	0.74	Fabricación de bajo costo
										11	11C	11B	52	19.33	Alta precisión en modelos de visión artificial
											12	12B	49	18.21	Alto reconocimiento de los comandos de voz
												13	18	6.69	Generar base de datos con historial médico
Total													269	100	

Cuadro 2.4: Diagrama de Mudge del sistema robótico. **Fuente:** Elaboración propia

Al final, el diagrama de Mugde nos proporciona un resultado expresado en puntuación y otro en porcentaje que representa la importancia de cada ítem.

2.4.2. Análisis de Pareto para Metodología Mudge

Con el conteo de datos y porcentajes obtenidos por medio del diagrama de Mugde contamos con los recursos necesarios para la construcción de un diagrama

de Pareto, el cual es una herramienta potente al momento de ayudarnos a identificar las acciones más relevantes o vitales para nuestro proyecto para así estimar recursos, esfuerzo y tiempo en estos requerimientos, dejando un poco de lado las acciones que resulten más triviales para el producto final.

Funciones	Requerimientos del sistema	%	Acumulado	Suma
9	Alta precisión en su posición	20.44	20.44	55
11	Alta precisión en modelos de visión artificial	19.33	39.67	52
12	Alto reconocimiento de los comandos de voz	18.21	57.88	49
6	Fácil manejo	9.29	67.17	25
4	Entorno agradable	8.17	75.34	22
1	Generar base de datos con historial médico	6.69	82.03	18
8	Estructura resistente	4.09	86.12	11
2	Fácil mantenimiento	3.71	89.83	10
7	Movible y fácil de transportar	3.34	93.17	9
3	Herramienta intercambiable	2.6	95.77	7
5	Visualmente agradable	2.23	98	6
10	Fácil fabricación	1.11	99.11	3
13	Fabricación de bajo costo	0.74	100	2

Cuadro 2.5: Análisis de Pareto para sistema robótico. **Fuente:** Elaboración propia

En la tabla 2.5 se realiza el diseño necesario para el análisis de Pareto, en primer lugar se ordenan los requerimientos de mayor a menor puntuación o porcentaje y se realiza una suma acumulativa de ellos mediante las fórmulas:

$$\% = \frac{Suma}{Total} * 100 \quad (2.1)$$

$$Acumulado_i = \% + Acumulado_{i-1} \quad (2.2)$$

En el diagrama de la figura 2.6 se muestra por medio de barras y de color azul el resultado de la suma en puntuación obtenidos del diagrama de Mudge y de color rojo el porcentaje de forma acumulada obtenido por medio del análisis de Pareto.

La razón fundamental en la implementación de este esquema radica en que

con tan solo un vistazo podemos hacer afirmaciones como por ejemplo que si nos enfocamos en los primeros siete requerimientos ya tendríamos el proyecto en un avance de más del 80 %.

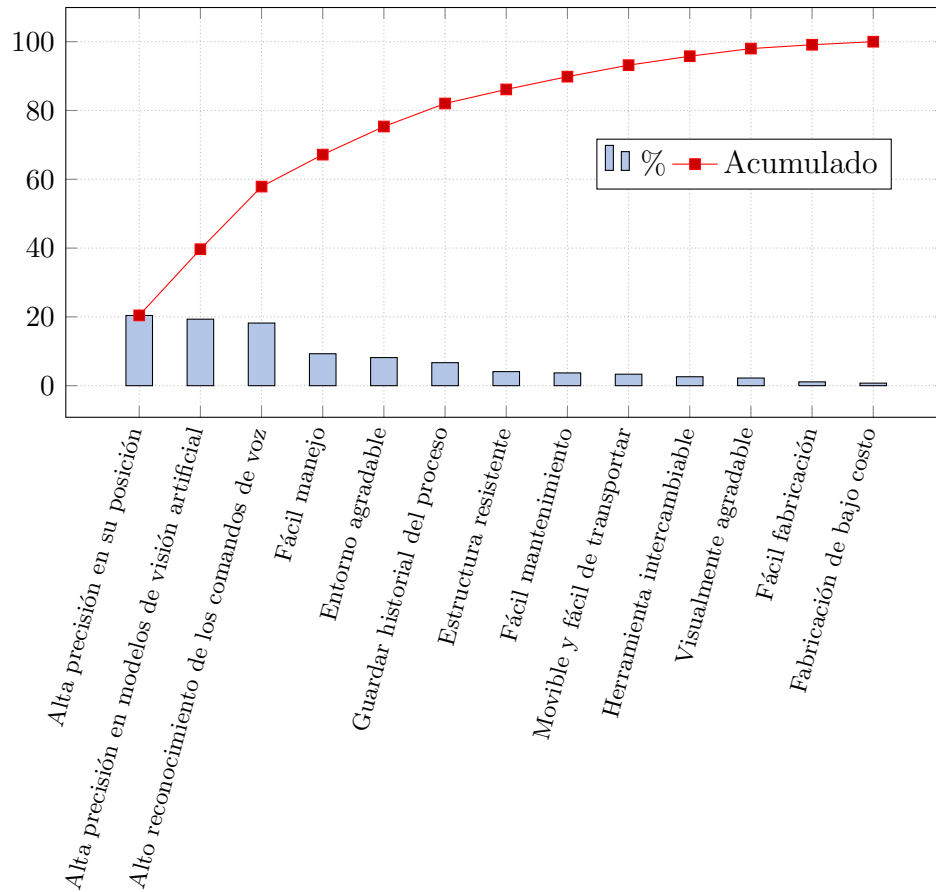


Figura 2.4: Diagrama de Pareto con resultados obtenidos del diagrama de Mudge.
Fuente: Elaboración propia

2.4.3. Despliegue de la función de calidad

A los parámetros cuantitativos y medibles que debe tener el sistema se les denomina especificaciones objeto o meta y una de las técnicas más utilizadas para

definirlas, se le denomina Despliegue de la función de calidad (**QFD**) o matriz de la casa de la calidad, aunque también este método nos despliega una serie de información de tipo cualitativa valiosa para el correcto desarrollo del producto.

Un motivo importante para la implementación de esta metodología está en que nos ofrece una comparativa con respecto a proyectos similares, para así, establecer diferencias o similitudes entre ellos y fortalecer aspectos que lo requieran.

Lo fundamental del despliegue de la función de calidad es obtener las especificaciones meta, pues, son estas las características que va a poseer nuestro sistema y con ello ya podemos construir una lista de recursos técnicos y precisos para realizar un balance de precios y tiempo que conllevara la ejecución del proyecto, es decir ya podemos realizar un cronograma de actividades y organizar el tiempo en el cual se va a dar la implementación.

Existen varias versiones de QFD, entre las más conocidas son las propuestas por Akao, King y ASI en este proyecto utilizaremos el modelo ASI debido a su tipicidad.

En la figura 2.5 se presenta la matriz de la casa de la calidad para el sistema robótico en asistencia quirúrgica con sus diferentes segmentos.

El primer segmento es donde se refleja los requerimientos de los usuarios o denominados ¿Qué? En el cual se hace una evaluación de los puntajes obtenidos en el diagrama de Mudge y se obtiene a la vez un porcentaje de importancia. El segundo segmento corresponde a los como y se refiere a las funcionalidades del sistema y la relacionan que existe entre ellas, en la parte media de la gráfica se hace la relación entre los ¿Qué? Y los como de forma cuantitativa en la cual 1 significa una relación débil, 3 una relación media y 9 una relación fuerte. Con estos valores y la importancia en porcentaje de los ¿Qué? Se obtiene una cuantización de los

requerimientos del sistema, refiriéndose a la importancia o peso de cada uno con respecto a los demás. En la izquierda de la figura se hace una comparación con los proyectos similares que existen en el mercado teniendo en cuenta los requerimientos de los usuarios.

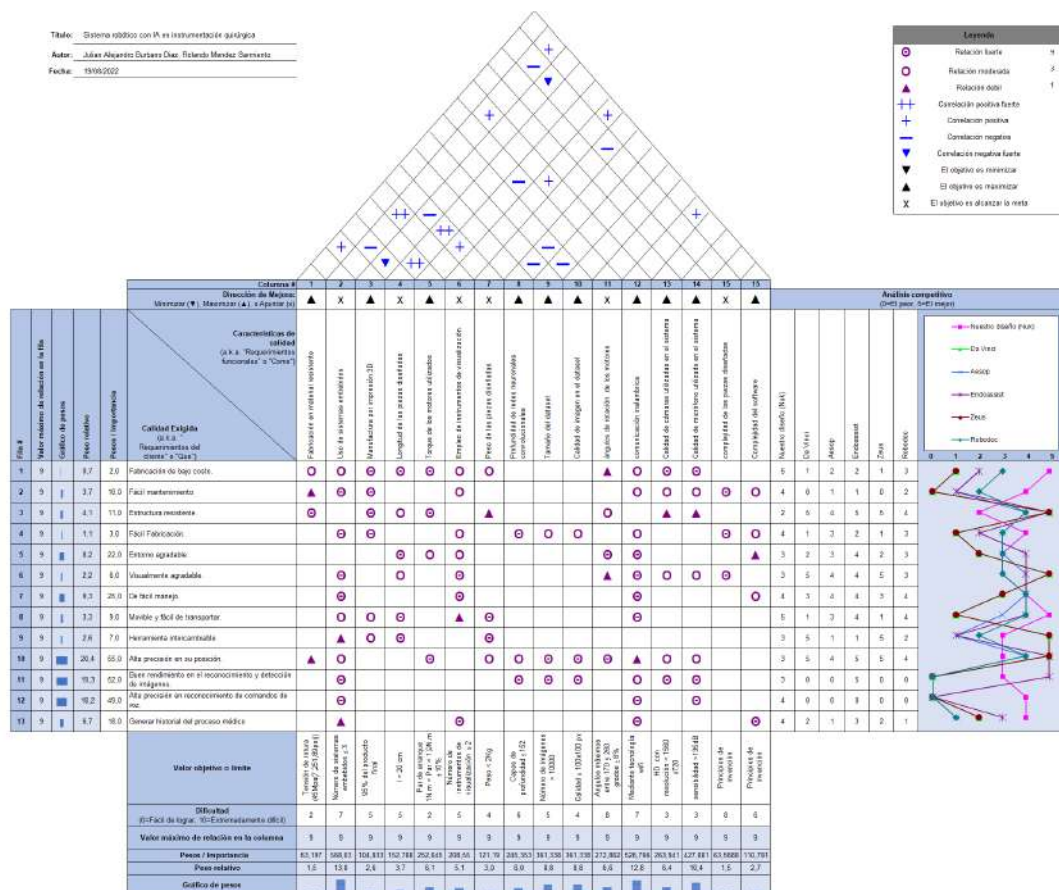


Figura 2.5: Despliegue de la función de calidad para sistema robótico. **Fuente:** Elaboración propia

En la figura 2.6 se presenta el primer segmento y dentro de él en la columna nombrada como calidad exigida de forma cualitativa se nombran los 13 requerimientos del usuario, también ingresamos la relación de importancia que fue obtenida a través del diagrama de Mudge y al igual que este, la casa de calidad nos

proporciona sus porcentajes de forma numérica y gráfica.

Fila #	Valor máximo de relación en la fila	Gráfico de pesos	Peso relativo	Pesos / Importancia	Calidad Exigida (a.k.a. "Requerimientos del cliente" o "Que")
1	9		0,7	2,0	Fabricación de bajo costo.
2	9		3,7	10,0	Fácil mantenimiento.
3	9		4,1	11,0	Estructura resistente.
4	9		1,1	3,0	Fácil Fabricación.
5	9		8,2	22,0	Entorno agradable.
6	9		2,2	6,0	Visualmente agradable.
7	9		9,3	25,0	De fácil manejo.
8	9		3,3	9,0	Movible y fácil de transportar.
9	9		2,6	7,0	Herramienta intercambiable.
10	9		20,4	55,0	Alta precisión en su posición.
11	9		19,3	52,0	Buen rendimiento en el reconocimiento y detección de imágenes.
12	9		18,2	49,0	Alta precisión en reconocimiento de comandos de voz.
13	9		6,7	18,0	Generar historial del proceso médico

Figura 2.6: Matriz de los requerimientos del usuario en el QFD. **Fuente:** Elaboración propia

En la sección donde se enumeran las características del sistema también se hace la correlación entre ellas de manera cualitativa y se clasifican como correlación positiva fuerte, correlación positiva, correlación negativa, correlación negativa fuerte, o sin correlación, como se muestra en la figura 2.7. Por ejemplo, la fabricación en un material resistente está directamente relacionada con el método de impresión

3D, pues entre más porcentaje del producto se encuentre formado por un material resistente, más se implementará este proceso de manufactura, a ello se conoce como correlación positiva.

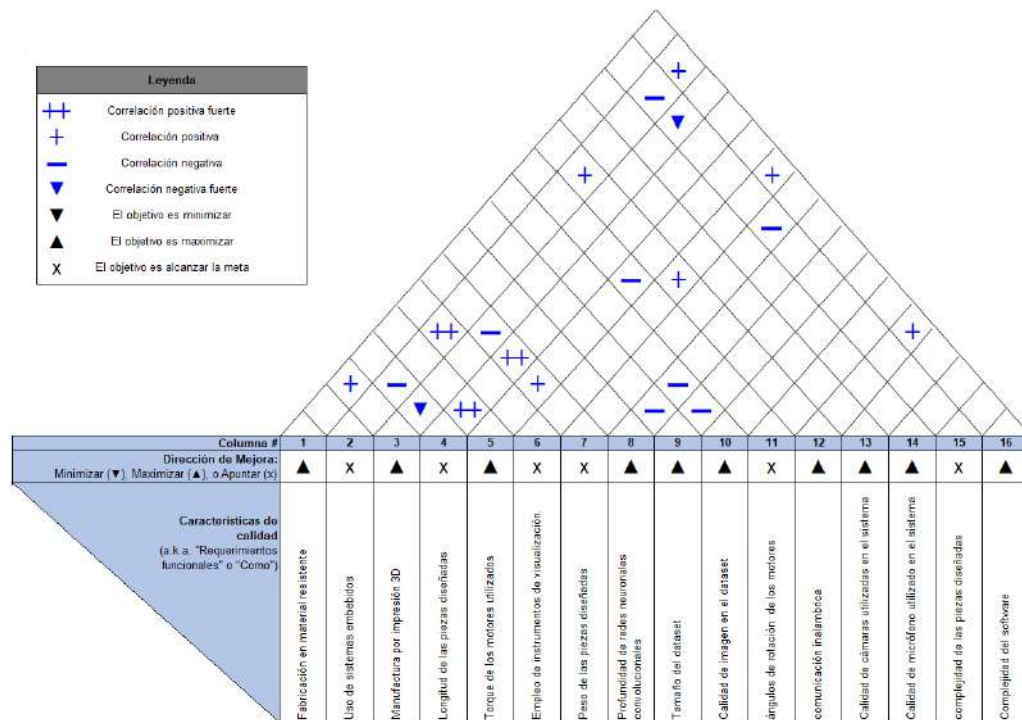


Figura 2.7: Matriz de los requerimientos del sistema en el QFD. **Fuente:** Elaboración propia

En la parte superior de la descripción de las funcionalidades del sistema se hace una caracterización de ellas mediante símbolos que clasifican cada requerimiento según su dirección de mejora, ya sea el objetivo maximizar, minimizar o dejar en una meta específica.

Observando la gráfica 2.8 se pueden generar conclusiones relevantes para el diseño del prototipo, ya que la casa de calidad en este segmento nos ofrece una vista amplia con respecto a productos ya existentes en el mercado, como lo son el

sistema Da Vinci, Aesop, Endoassist, Zeus y Robodoc. La diferenciación se realiza con las características exigidas por el usuario y puntuaciones que se le asigna a cada producto en los diferentes aspectos.

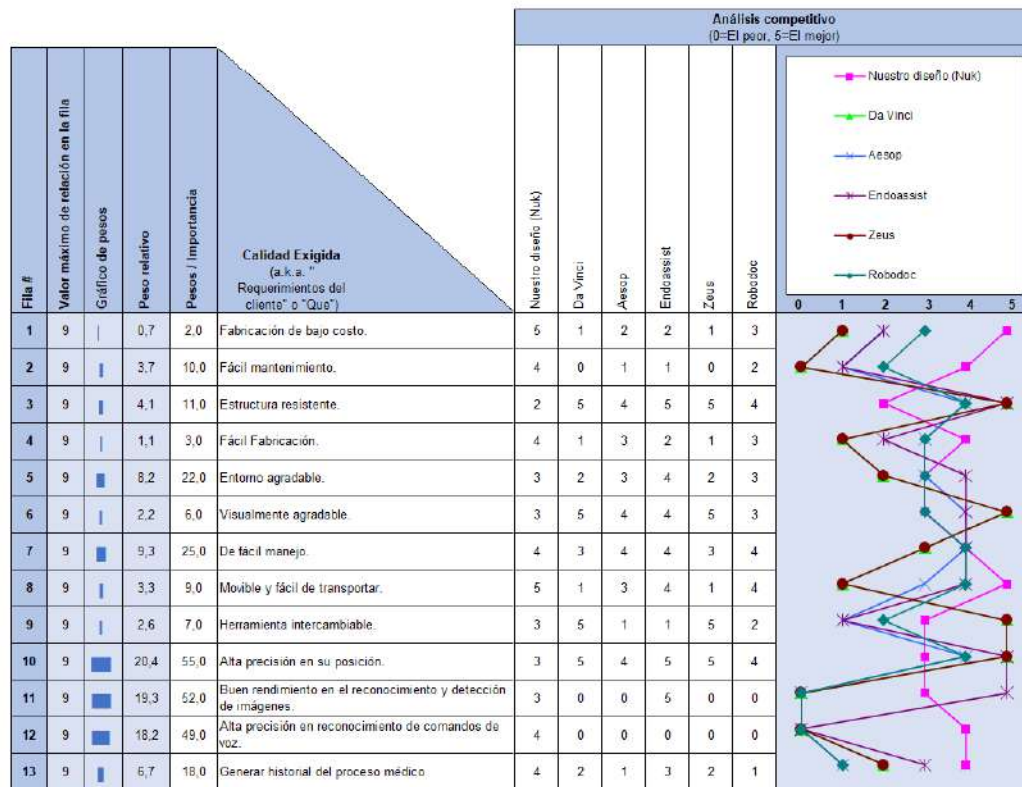


Figura 2.8: Matriz del análisis competitivo en el QFD. **Fuente:** Elaboración propia

Para hacer una relación entre la calidad exigida por los usuarios y las características de calidad del sistema se utiliza la simbología que se indica en la parte superior de la figura 2.9 donde se muestra la equivalencia de una relación fuerte moderada y débil con los números 9, 3 y 1 respectivamente, la identificación de correspondencia entre ques y comos que se realiza en la misma figura es una de las partes más importantes del QFD, ya que de esta comparación y de la importancia en porcentajes de la figura 2.6 se obtiene la cuantificación de importancia de los

requerimientos del sistema.

Legenda		
⊖	Relación fuerte	9
○	Relación moderada	3
▲	Relación debil	1

Calidad Exigida (a.k.a. "Requerimientos del cliente" o "Que")	Características de calidad (a.k.a. "Requerimientos funcionales" o "Como")															
	Fabricación en material resistente	Uso de sistemas embebidos	Manufactura por impresión 3D	Longitud de las piezas diseñadas	Torque de los motores utilizados	Empleo de instrumentos de visualización.	Peso de las piezas diseñadas	Profundidad de redes neuronales convolucionales	Tamaño del dataset	Calidad de imagen en el dataset	ángulos de rotación de los motores	comunicación inalámbrica	Calidad de cámaras utilizadas en el sistema	Calidad de micrófono utilizado en el sistema	complejidad de las piezas diseñadas	Complejidad del software
Fabricación de bajo costo.	○	○	○	○	○	○	○				▲	○	○	○		
Fácil mantenimiento.	▲	○	○			○						○	○	○	○	○
Estructura resistente.	○		○	○	○		▲				○		▲	▲		
Fácil Fabricación.		○	○			○		○	○	○		○			○	○
Entorno agradable.				○	○	○					○	○				▲
Visualmente agradable.		○		○		○					▲	○	○	○	○	
De fácil manejo.		○				○						○				○
Movible y fácil de transportar.		○	○	○		▲	○					○				
Herramienta intercambiable.		▲	○	○			○									
Alta precisión en su posición.	▲	○			○		○	○	○	○	○	▲	○	○		
Buen rendimiento en el reconocimiento y detección de imágenes.		○						○	○	○		○	○	○		
Alta precisión en reconocimiento de comandos de voz.		○										○		○		
Generar historial del proceso médico		▲				○						○				○

Figura 2.9: Matriz de relación entre requerimientos del usuario y del sistema en el QFD. **Fuente:** Elaboración propia

En la sección de cuantificación de los requerimientos del sistema indicada en la figura 2.10 tenemos 6 filas en la primera se encuentran las especificaciones meta del producto, en segundo lugar le asignamos en nivel de complejidad de cada característica en un rango de 0 a 10, siendo 0 un objetivo fácil de lograr y 10 extremadamente difícil en tercer lugar tenemos la fila de valor máximo por columna correspondiente a la comparación entre requerimientos del cliente y del sistema, en cuarto lugar tenemos la importancia dada en puntuación de cada funcionalidad

estos datos son cruciales en el momento de toma de decisiones y administración de tiempo al igual que el peso relativo que corresponde a la relevancia de cada función expresada en porcentaje con respecto a las otras especificaciones.

Valor objetivo o límite	Tensión de rotura (45Mpa/(251.59psi))	Número de sistemas embebidos ≤ 3	95% del producto final	$l < 20$ cm	Par de arranque 1N.m < Par < 1.9N.m $\pm 10\%$	Número de instrumentos de visualización ≤ 2	Peso < 2Kg	Capas de profundidad ≤ 152	Número de imágenes > 10000	Calidad $\geq 100 \times 100$ px	Ángulos máximos entre 170 y 200 grados $\pm 0\%$	Mediante tecnología wifi	HD con resolución < 1500 x720	sensibilidad > 150dB	Principios de invención	Principios de invención
Dificultad (0=Fácil de lograr, 10=Extremadamente difícil)	2	7	5	5	2	5	4	6	5	4	8	7	3	3	8	6
Valor máximo de relación en la columna	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
Pesos / Importancia	63,197	568,03	104,833	152,788	252,045	208,55	121,19	245,353	361,338	361,338	272,862	526,766	263,941	427,881	63,5688	110,781
Peso relativo	1.5	13.8	2.6	3.7	6.1	5.1	3.0	6.0	8.8	8.8	6.6	12.8	6.4	10.4	1.5	2.7
Gráfico de pesos																

Figura 2.10: Matriz de cuantificación de los requisitos del sistema en el QFD.
Fuente: Elaboración propia

El algoritmo matemático para la obtención de la cuantización de los requerimientos del sistema se evalúa por columnas y está dado por la sumatoria indicada a continuación.

$$P/I = \sum_{i=1}^n Pr_i * Rs_i = Pr_1 * Rs_1 + Pr_2 * Rs_2 + ... Pr_n * Rs_n \quad (2.3)$$

Donde:

- i = Número de fila correspondiente al requerimiento del usuario.
- n = Número total de requerimientos del usuario.
- Pr = Peso relativo en la fila i .
- Rs = Relación existente entre requerimiento del sistema y del usuario.

- p/i= Peso / Importancia.

2.4.4. Análisis de Pareto para Metodología QFD

Funciones	Requerimientos del sistema	%	Acumulado	Suma
2	Uso de sistemas embebidos	13.84	13.84	568.03
12	Comunicación inalámbrica	12.83	26.57	526.77
14	Calidad de micrófono	10.42	37.1	427.88
10	Calidad de imagen en el dataset	8.8	45.9	361.34
9	Tamaño del dataset	8.8	54.71	361.34
11	Ángulos de rotación de motores	6.65	61.35	272.86
13	Calidad de cámara	6.43	67.78	263.94
5	Torque de los motores	6.14	73.92	252.04
8	Profundidad de CNN	5.98	79.9	245.35
6	Instrumentos de visualización	5.08	84.98	208.55
4	Longitud de las piezas diseñadas	3.72	88.71	152.78
7	Peso de las piezas diseñadas	2.95	91.66	121.19
16	Complejidad del software	2.7	94.35	110.78
3	Manufactura por impresión 3D	2.55	96.91	104.83
15	Complejidad de las piezas diseñadas	1.55	98.46	63.57
1	Fabricación en material resistente	1.54	100	63.19

Cuadro 2.6: Análisis de Pareto para la importancia de requerimientos del sistema obtenidos por QFD. **Fuente:** Elaboración propia

Al igual que con los datos obtenidos a través del diagrama de Mugde se hizo el análisis de Pareto, podemos hacerlo también con la cuantización que se obtuvo de los requerimientos del sistema en el despliegue de la función de calidad como se muestra en la tabla 2.6, esto con el fin de priorizar recursos y esfuerzo al tiempo que se hace posible ver desde una perspectiva más general el proyecto siendo una herramienta de ayuda en la construcción de un cronograma eficaz.

A diferencia del esquema obtenido con los datos del diagrama de Mugde de la figura 2.6, el que se realizó con los resultados de la casa de la calidad de la figura 2.11 tiene una curva más suave en sus inicios y, por lo tanto, se requiere

enfocarse en más actividades para lograr un avance significativo en el desarrollo del producto. Para lograr un progreso del 80 % es necesario tener como prioridad las 9 primeras actividades, es decir, la mitad de los requerimientos del sistema.

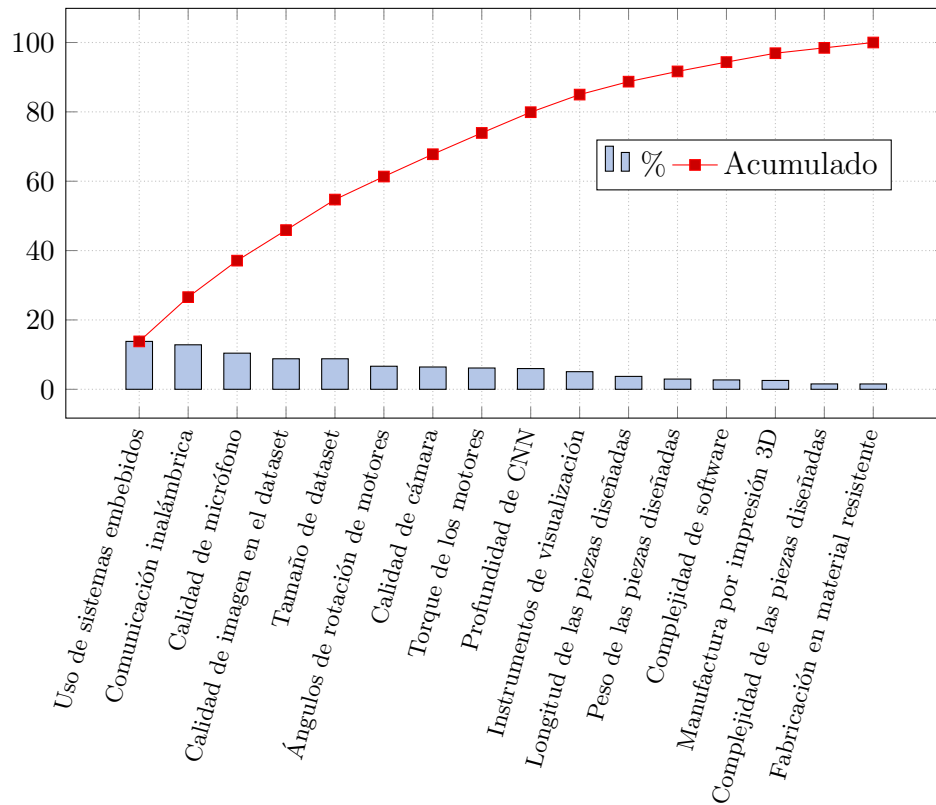


Figura 2.11: Diagrama de Pareto del proceso en el desarrollo del producto para QFD. **Fuente:** Elaboración propia

2.4.5. Conclusiones obtenidas del desenvolvimiento del producto

Acorde con el diagrama de Mudge podemos dividir los requerimientos del usuario en tres tipos según su importancia, el primer grupo estaría dado por los que su valor en puntuación sea mayor a dos tercios del valor máximo, es decir 13.626,

el segundo por los valores superiores a un tercio del valor máximo y menor a dos tercios, finalmente el tercer grupo estaría conformado por los requerimientos que su valor sea menor a un tercio.

Requerimientos del usuario e importancia	
Peso/Importancia	Requerimientos del usuario
Importancia alta	Alta precisión en su posición
	Alta precisión en modelos de visión artificial
	Alto reconocimiento de los comandos de voz
Importancia media	Fácil manejo
	Entorno agradable
	Generar base de datos con historial médico
Importancia baja	Estructura resistente
	Fácil mantenimiento
	Movible y fácil de transportar
	Herramienta intercambiable
	Visualmente agradable
	Fácil fabricación
	Fabricación de bajo costo

Cuadro 2.7: Clasificación por importancia de los requerimientos del usuario. **Fuente:** Elaboración propia

En los requerimientos de mayor importancia se encuentra la alta precisión en la posición del efector final, lo cual es consecuente, pues, en el proceso es de vital importancia la confiabilidad en el momento de obtener y entregar correctamente las herramientas quirúrgicas, en seguida se encuentra la precisión en el sistema de visión artificial, esto sería de gran ayuda en el momento de escoger el instrumento correcto y llevarlo a la posición final adecuada, por último a este grupo lo conforma el reconocimiento confiable de comandos de voz, ya que es por este medio que se asignara el material a entregar.

En el diagrama de Pareto de la figura 2.4 se visualiza el gran impacto que tienen los tres requerimientos que hacen parte del primer grupo, dado que aproxi-

madamente son el 60 % del avance del proyecto, aunque para la ejecución de estas características por su forma no tienen gran ingreso de calidad económica, si de tiempo, es por ello que se objeta darle una relevancia considerable en el momento del diseño del cronograma.

Para el desarrollo de estas tres actividades se proponen varias soluciones estipuladas en la casa de calidad de la figura 2.5 en la que se encuentra la elección de servomotores adecuados para el sistema y una de las características importantes que se plantea, es su torque y se calcula para que sea consecuente con una altura media del robot antropomórfico que favorezca al entorno agradable, su par ideal o meta está entre 1Nm y 1.9Nm podría darse, según pruebas en un modelo virtual que servomotores más cercanos al límite superior de este intervalo se encuentren en los tres primeros grados de libertad, ya que son los que más van a requerir esfuerzo.

Como se mencionó anteriormente, realizar un modelo virtual del brazo robótico y simular su funcionamiento favorece y es ideal para evitar errores en la puesta en marcha al igual que para garantizar su precisión.

Con respecto a los sistemas de visión artificial se propone el uso de las CNN, ya que su especialidad es la clasificación y detección de objetos dentro de una imagen, el software seleccionado para su implementación es MATLAB, pues, nos permite integrar de forma ágil las simulaciones con la puesta en marcha, también nos brinda herramientas de gran ayuda para el entrenamiento de las redes neuronales como para la obtención del dataset.

Las herramientas que nos ofrece MATLAB en cuanto al training de las redes neuronales son estructuras de redes profundas ya definidas como las nombradas en el capítulo uno, y con respecto al etiquetado de objetos nos brinda la app imageLabeler.

MATLAB también nos permite diseñar una app por medio de un guide que nos ayude a la toma de un dataset adecuado.

En relación con el reconocimiento de voz se plantea como método efectivo implementar un asistente virtual ya diseñado porque su gran calidad es admirable con respecto a uno que se vaya a diseñar desde sus inicios, además los sistemas más reconocidos han sido creados por empresas como Google y Microsoft. En este punto se tienen dos posibilidades, ya sea Google assistant o Alexa, pero dado que para este proyecto y acorde con la casa de calidad es relevante la incorporación de sistemas embebidos, se elige Google assistant, pues tiene mayor información y facilidades de conectividad con estos instrumentos.

En el segundo grupo tenemos los requerimientos de importancia media y en ellos está el fácil manejo del sistema, el entorno agradable y la utilización de una base de datos para salvar información importante del procedimiento. Estas funcionalidades requeridas por los usuarios, albergan alrededor de un 20 % en el avance del proyecto, como se puede observar en la figura 2.4.

Para acondicionar el sistema robótico a estas características se debe tener en cuenta las funcionalidades para lograr ese objetivo, por ejemplo para que el sistema sea de fácil manejo se debe implementar instrumentos de visualización que ayuden en la interacción persona-máquina, dentro de estos instrumentos se propone la utilización de una pantalla integrada al robot así como un led indicador, la forma de control y de establecimiento de mando se incorporara desde un ordenador el cual se conectara de forma inalámbrica con el robot antropomórfico así se lograra crear una interfaz mediante un guide para facilitar el ingreso y guardado de datos.

El aspecto de entorno agradable se refiere a la comodidad de interacción dentro del quirófano entre el personal médico con la máquina, para aportar significativamente a este campo se plantea una altura total del prototipo menor a los 90 cm

para no sobredimensionar el proyecto y hacerlo lo más cómodo posible, también los grados de libertad deben ser los suficientes para un movimiento ágil con la posibilidad de que sus trayectorias no incomode a los presentes en el procedimiento, y se ha elegido que el robot sea de cinco grados de libertad para que no se desfavorezca la facilidad de fabricación y aporte a esta característica teniendo varias posibilidades de trayectoria para la obtención de una posición específica.

Los requerimientos de menor importancia están compuestos por la mitad del total de características exigidas por los usuarios y corresponden aproximadamente al último 20 % del avance del proyecto, el primer requerimiento de este grupo es que la estructura física del sistema debe ser resistente, tal como se plasmó en las especificaciones meta de la casa de calidad el material propuesto es el PLA, ya que es un plástico resistente, y a su vez aporta a otro aspecto que se encuentra en este grupo como lo es la fácil fabricación, pues el PLA puede ser impreso en 3D, inclusive otra de las ventajas de su utilización es su amigabilidad con el medio ambiente al ser biodegradable.

En cuanto a las características de fácil mantenimiento, movable, fácil de transportar y de fácil fabricación, se ajustan a estrategias nombradas anteriormente, como el uso de sistemas embebidos, pues hacen que el producto sea liviano y en caso de fallo de uno de estos componentes, su reparación se limitaría al cambio del mismo, además la mayoría de estos componentes son de bastante accesibilidad en el mercado y a un coste relativamente económico. Con relación a que nuestro producto sea de bajo costo, visualmente agradable y de herramienta intercambiable, es clave el método de manufactura, por ello se empleara la impresión 3D porque permite pasar de un modelo virtual a una estructura física de buena calidad por un precio asequible.

Especificaciones Meta	
Requerimientos del sistema	Especificaciones
Fabricación en material resistente	Tensión de rotura (45Mpa(7251,89psi))
Uso de sistemas embebidos	Número de sistemas embebidos ≥ 3
Manufactura por impresión 3D	95 % del producto final
Longitud de las piezas diseñadas	$l < 20$ cm
Torque de los motores utilizados	Par de arranque ($1N.m < Par < 1,9N.m$) $\pm 10\%$
Empleo de instrumentos de visualización.	Número de instrumentos de visualización ≥ 2
Peso de las piezas diseñadas	Peso $< 2Kg$
Profundidad de redes neuronales convolucionales	Capas de profundidad ≥ 152
Tamaño del dataset	Número de imágenes > 10000
Calidad de imagen en el dataset	Calidad $\geq 100 \times 100$ px
ángulos de rotación de los motores	Ángulos máximos entre 170 y 260 grados $\pm 6\%$
comunicación inalámbrica	Mediante tecnología wifi
Calidad de cámaras utilizadas en el sistema	HD con resolución $< 1560 \times 720$
Calidad de micrófono utilizado en el sistema	sensibilidad $> 135dB$
complejidad de las piezas diseñadas	Principios de invención
Complejidad del software	Principios de invención

Cuadro 2.8: Especificaciones meta del sistema robótico. **Fuente:** Elaboración propia

2.5. Identificación de los recursos necesarios

Establecidas las especificaciones del producto y sus lineamientos, podemos adentrarnos en la selección de materiales necesarios para la construcción física del mismo. Existen dos tipos de instrumentos requeridos, los primeros son los componentes que están dentro y que hacen parte del sistema y el segundo tipo son los utilizados solo en el armado o ensamble del proyecto.

La tabla 2.9 lista todos los recursos físicos necesarios que comprende la primera clase de instrumentos requeridos, es decir, los que hacen parte del sistema, entre ellos tenemos una fuente de poder encargada de energizar la parte electrónica del proyecto. También encontramos dos microcontroladores implementados en tarjetas de desarrollo capaces de hacer cálculos, controlar las salidas y facilitar la comunicación en el sistema, como se plasmó en el desenvolvimiento del producto, se necesita un flujo de información de forma inalámbrica, por esto se hace una se-

lección entre la familia de sistemas embebidos ESP, ya que cuentan con tecnología wifi y en el caso del esp32 también con conexión mediante bluetooth.

Componentes del sistema			
Número	Descripción	objetivo	Cantidad
1	Fuente de poder 5A 12v	Energizar el sistema	1
2	Microcontrolador ESP32	Operaciones de control	1
3	Microcontrolador ESP8266	Operaciones de control	1
4	Modulo Lm2596	Regulador de voltaje	3
5	Modulo pca9685	Demux para servomotores	1
6	Modulo rele de un canal	Switch magnetico	1
7	Led RGB	Indicador	2
8	Pantalla ili9341	Visualizador	1
9	Ventilador 5v	Refrigerador	1
10	Servomotor mg996r	Generador de torque	3
11	Servomotor lewanSoul LD-20MG	Generador de torque	2
12	Servomotor sg90	Generador de torque	1
13	Filamento de PLA	Material de construcción	2 Kg
14	Tornillo acero Inoxidable M3	Sujetador	20 u/n
15	Tornillo acero Inoxidable M4	Sujetador	20 u/n
16	kit de sutura	Herramienta quirúrgica	1
17	Guantes de látex	Herramienta quirúrgica	1
18	Bata quirúrgica	Herramienta quirúrgica	1
19	Rodamiento de bolas uxccl	Reductor de fricción	1
20	Cámara	Obtención de datos	2
21	Asistente de voz	Obtención de datos	1
22	Ordenador	Control	1
23	Madera tipo MDF	Material de construcción	1
24	Cable de conexión	Conectores	1 m
25	Extensiones para servomotor	Conectores	4
26	Mini switch	Interruptor mecánico	1
27	Jack Hembra para alimentación	Acople	1

Cuadro 2.9: Lista de componentes físicos del sistema robótico. **Fuente:** Elaboración propia

Puesto que el voltaje de entrada de los sistemas embebidos no es el mismo de la fuente, se hace necesaria la implementación de reguladores como lo son los de la serie Lm2596 y por el hecho de que su limitante está en el valor máximo de corriente que puede entregar se requiere varios de este tipo, para el movimiento se

emplea generadores de torque, en este caso servomotores, cada servomotor posee tres conexiones la primera es GND o tierra la segunda alimentación o V+ y la tercera es una señal PWM indicadora del ángulo en el que se debe posicionar. Por el número de servomotores que se utiliza se necesita introducir un elemento que actúe como demux entre las tarjetas de desarrollo y los servomotores, este componente es de la serie pca9685 la cual incluye hasta 16 canales de salida.

En los requerimientos del producto se plantea la importancia de los elementos de visualización que facilita la interacción entre hombre-máquina, para satisfacer este requerimiento se plantea la utilización de una pantalla ili9341 seleccionada por sus dimensiones y gran capacidad para la visualización de los gráficos haciendo más dinámico el uso del sistema, con el mismo fin hacemos uso de leds RGB que además de mejorar la comunicación entre el usuario y el sistema también hace que el proyecto sea visualmente agradable.

Para la obtención de datos a los cuales nuestro sistema reaccionará utilizamos cámaras y un asistente de voz. Por la complejidad del tratamiento de estos datos con respecto a la visión artificial no es práctico hacerlo mediante sistemas embebidos, necesitamos un hardware más avanzado y requerimos de un ordenador que actúe como centro de control y que además nos facilite la creación de una base de datos, puesto que es un requerimiento de los usuarios el guardar información del proceso en forma de un historial.

Al momento de ensamblar nuestro sistema y hacer de este una estructura sólida y resistente, recurrimos a sujetadores como tornillos que por su seguridad y dimensiones se selecciona de tipo M3 Y M4.

Además de implementar el PLA como material principal de construcción de las piezas mecánicas, hacemos uso de MDF para generarle una base a nuestro

producto, la cual tiene como finalidad darle solidez y hacer de este un sistema cómodo en su uso. Con respecto al empalme eléctrico utilizamos cable de conexión por metro y extensiones de servomotor de venta en el mercado.

Herramientas para armado de prototipo		
Número	Descripción	objetivo
1	Impresora 3D	Manufactura de piezas
2	Desarmadores	Atornillar
3	Multímetro	Comprobar conexiones
4	cautín	Aplicar soldadura
5	Estaño	Soldadura
6	Crema de soldadura	Mejorar soldadura
7	Termo encogible	Proteger conexiones
8	Pegamento para base de impresión	mejorar adherencia
9	Hoja de lija	Acabados
10	Pintura negra	Acabados
11	Pintura naranja	Acabados
12	Pintura cromada	Acabados
13	Taladro	Soltar soporte
14	pinzas	Soltar soporte

Cuadro 2.10: Herramientas para armado de prototipo. **Fuente:** Elaboración propia

En el entrenamiento de redes neuronales, en las pruebas y en la puesta en marcha del prototipo se hace necesario la creación de un ambiente quirúrgico, por esto se debe incluir instrumental básico de sutura, guantes de látex y bata quirúrgica.

En cuanto a acabados se refiere, necesitamos herramientas que no son propias del sistema, pero que se requieren para dar forma a su estructura física, entre ellas tenemos lijas, pintura, pinzas, termo-encogible, taladro, pegamento para base de impresión 3D y crema de soldadura. Además de las herramientas de uso indispensable como impresora 3D, multímetro, desarmadores, estaño y cautín.

2.6. Cronograma

Con claridad de los recursos necesarios que necesitara nuestro proyecto y con un lineamiento obtenido a través de técnicas de desenvolvimiento de producto, nos enfocamos en el planteamiento de un cronograma específico donde se plasme el tiempo que conllevara la realización de este, para esto se hace uso de una Estructura de descomposición del trabajo (**EDT**) con el fin de prever las acciones necesarias en cada área que respecta a la creación del sistema robótico.

Mediante la figura 2.12 se pretende desglosar nuestro proyecto en subsistemas más simples para comprender las tareas que conllevara en cada uno de ellos y así poder definir tiempo de trabajo.

La realización de un sistema robótico basado en inteligencia artificial para la asistencia quirúrgica podemos subdividirlo en un diseño electrónico y mecánico del mismo, a la vez que en un proceso de ensamble y calibrado, el cual necesita integrarse con un software de visión artificial y reconocimiento de comandos, al finalizar la implementación estructural del proyecto se lo somete a una puesta en marcha de la cual se documenta los resultados y se concluye sobre ellos.

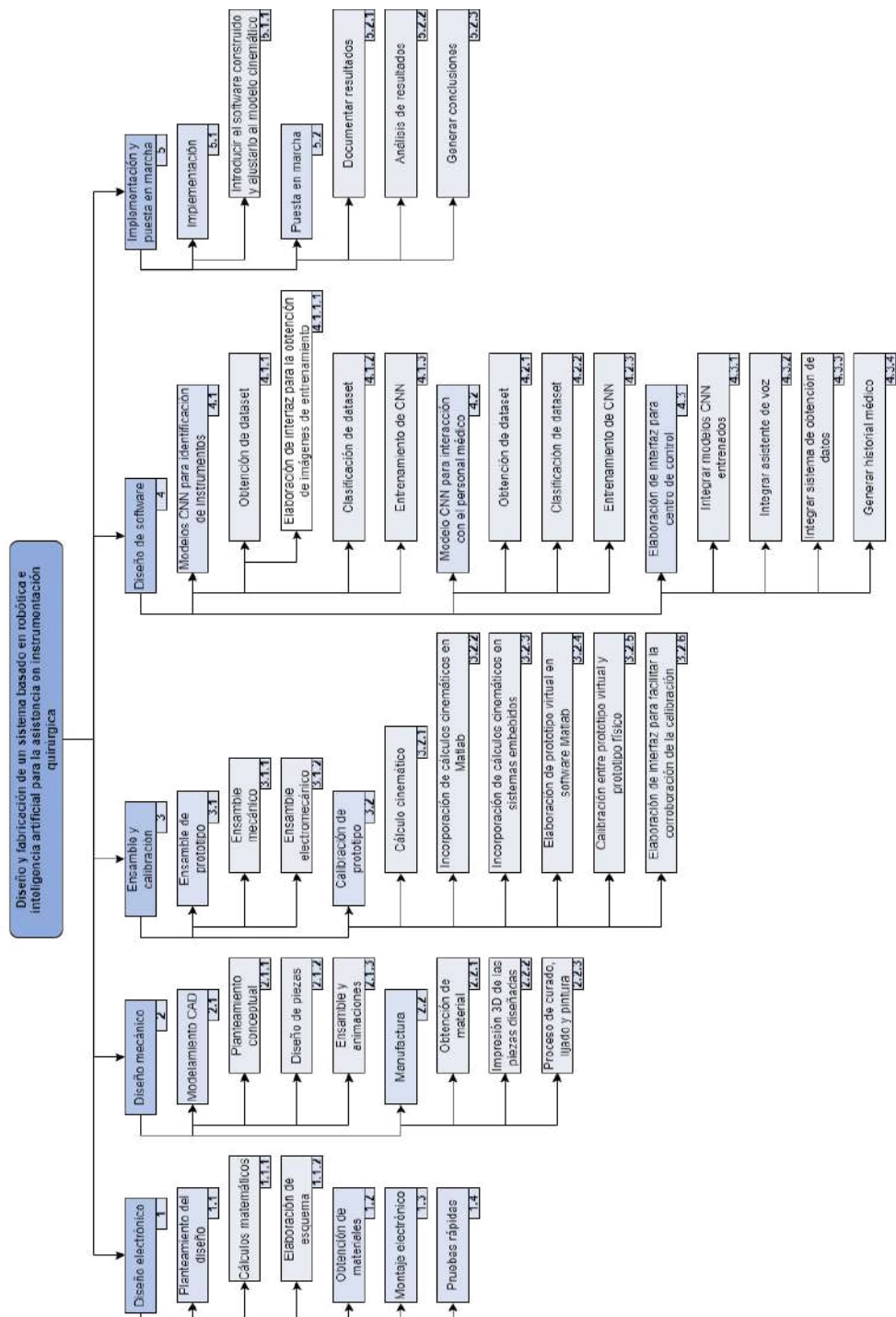
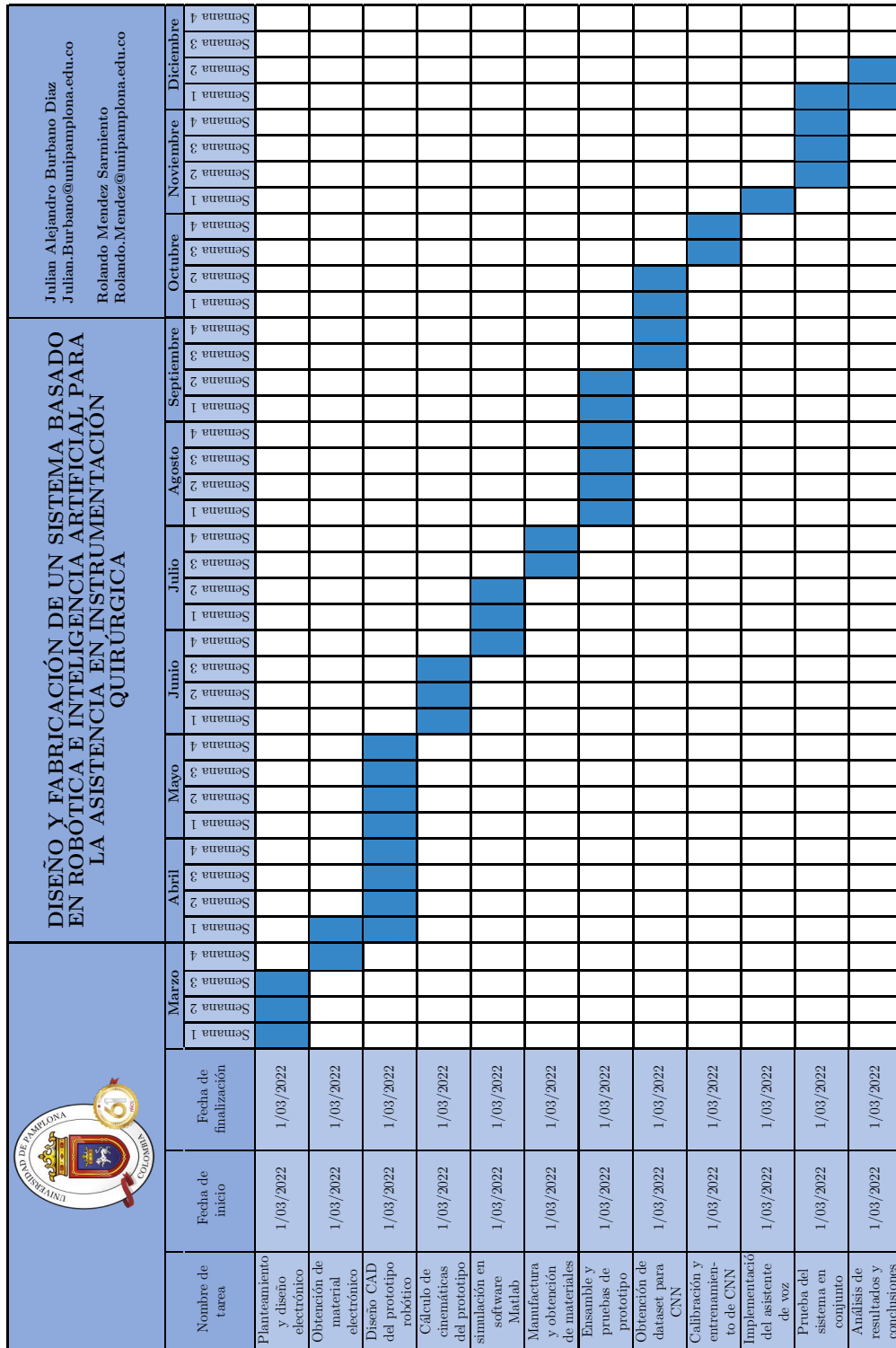


Figura 2.12: Estructura de descomposición del trabajo para sistema robótico. **Fuente:** Elaboración propia

Una forma eficiente y ordenada de hacer una planificación en cuanto a tiempos se refiere podemos hacerlo mediante un diagrama de Gant el cual es ilustrado en la figura 2.11, en él se agrupan de una manera más generalizada las actividades obtenidas mediante la estructura de descomposición de trabajo y se estima el tiempo que conllevara cada una en su ejecución, al finalizar obtendremos una fecha estimada de finalización del proyecto.

Nuestro proyecto se plantea en el transcurso de 10 meses, el cual empieza el primero de marzo del año 2022 y finaliza el nueve de diciembre del mismo año.



Cuadro 2.11: Diagrama de Gant con las actividades a realizar en el plan de trabajo. **Fuente:** Elaboración propia

Capítulo 3

Diseño electrónico

3.1. Características relevantes de los componentes electrónicos

Para el desarrollo de un modelo virtual del prototipo, empezamos por obtener las características de cada uno de los elementos electrónicos, los cuales, nos van a ayudar en el momento de hacer las respectivas conexiones, por ejemplo, nos ayudara al suministro de energía, pues, obteniendo su voltaje de operación se podría realizar un acople de ser necesario, así evitaríamos averiar los instrumentos.

También con el peso de cada elemento podemos al finalizar el prototipado dar un valor aproximado bastante acertado del peso en conjunto del sistema robótico, de la misma forma, un aspecto importante a tener en cuenta en cada una de las tarjetas electrónicas, es la corriente necesaria para su funcionamiento y en el caso de los reguladores es fundamental el amperaje que son capaces de suministrar.

Características de los componentes electrónicos	
Componente electrónico	Características
Regulador Lm2596	Rango de voltaje de entrada < 40V
	Rango de voltaje de salida < 37V
	Oscilador interno de 150KHz
	Corriente máxima de salida 3A
Controlador Pca9586	Voltaje de entrada = 5V
	Máxima corriente por pin = 25mA
	Frecuencia de operación: 40 Hz a 1000 Hz
	Número de canales = 16
Servomotor Mg966r	Voltaje de operación: 4.8V a 7.2V
	Velocidad de operación: 0.14s/60° a 0.17s/60°
	Torque: 9.4 kgf*cm a 11 kgf*cm
	Peso: 55 g
	Largo de cable: 31 cm
	Consumo de Corriente: 500-900mA
Servomotor Lewansoul ld-27mg	Voltaje de operación: 6V a 7.4V
	Velocidad de operación: 0.2s/60°
	Torque: 20 kgf*cm
	Peso: 66 g
	Largo de cable: 30 cm
Microcontrolador Esp32	Consumo de Corriente : 2.4A(carga máxima)
	Voltaje de operación: 3V a 3.6V
	Consumo de Corriente: 80mA
	Oscilador interno de 40MHz
	Rango de frecuencia wifi: 2.4Ghz-2.5Ghz
	Protocolo de bluetooth: bluetooth v4.2 BR/EDR
Microcontrolador Esp8266	Modulo de interfaz: UART, SPI, PWM, GPIO
	Voltaje de operación: 2.5V a 3.6V
	Consumo de Corriente: 80mA
	Velocidad de reloj: 80MHz
	Rango de frecuencia wifi: 2.4Ghz-2.5Ghz
Extractor ventilador	Modulo de interfaz: UART, SPI, PWM, GPIO
	Voltaje de operación: 5v
	Consumo de Corriente: 110mA
Módulo relé 1 canal	Dimensiones en milímetros: 40x40x10
	Voltaje de operación: 5V
	Máximo voltaje de conmutación: 30V DC Y 220V AC
Leds RGB	Dimensiones en milímetros: 44x24x17
	Voltaje de operación (RGB): (2.0, 3.2, 3.2)V
	Tipo: Ánodo y cátodo común
	Número de terminales: 4

Cuadro 3.1: Características importantes de cada uno de los componentes electrónicos. **Fuente:** Elaboración propia

En cuanto al uso de servomotores, es de gran importancia tener en cuenta el torque de los mismos, pues tienen que ser capaces de generar un momento de fuerza suficiente para un movimiento estable sin afectar la vida útil del dispositivo.

La velocidad de procesamiento en los microcontroladores insertos en las tarjetas de desarrollo utilizadas debe ser óptima, además estos sistemas embebidos deben

contar con las salidas necesarias para el tipo de comunicaciones que se requiera, como lo es la i^2c para el control de servos y SPI en el flujo de información con la pantalla de visualización.

Otros aspectos de interés son el rango de frecuencia de las conexiones a wifi de las tarjetas, pues debido a este valor se puede establecer la calidad de la conexión, ya que en este caso tanto el esp32 como el esp8266 son de aproximadamente 2.4Ghz tienen como ventajas ser accesibles desde largas distancias y además compatibles con la mayoría de dispositivos.

Los leds rgb conseguidos en el mercado usualmente son de dos tipos, el primero de cátodo común en el cual tres de los cuatro pines que posee corresponden a los colores rojo, azul y verde, el cuarto pin se debe conectar a GND; de manera contraria se conecta el cuarto pin en los leds de tipo ánodo común donde este iría a un voltaje de entrada positivo.

Con el fin de dar un orden y hacer una caracterización adecuada de cada uno de los elementos electrónicos se hace necesario construir un listado el cual podemos observar mediante la tabla 3.1, en ella encontraremos información como la anteriormente nombrada.

3.2. Planteamiento de conexión

En la figura 3.1 podemos observar la forma en que se planteó la conexión electrónica partiendo de una fuente de 12 V conectada a tres reguladores, dos de ellos con voltaje de salida de 5V y uno de 6V, la razón por la cual es importante hacer dos líneas de un mismo voltaje se debe a la capacidad de suministro de corriente de la tarjeta Lm2596, y el suministro de una línea de voltaje superior a los

5V para la alimentación de los motores tal como lo indica la tabla de características anterior.

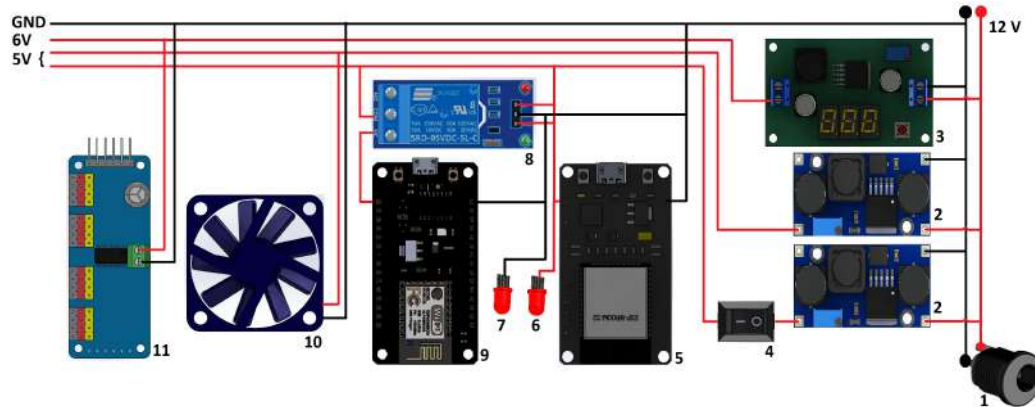


Figura 3.1: Diagrama de suministro eléctrica. **Fuente:** Elaboración propia

Conectado la línea de 6V y tierra a los pines v+ y GND del dispositivo pca 9586 se energiza los servomotores, de las líneas de 5V se alimentan las tarjetas esp32, esp8266, el módulo relé de un canal, y los leds rgb.

El motivo por la cual se incorporan dos tarjetas de desarrollo al sistema, se debe a la complejidad de programación, puesto que al hacer uso de un lenguaje secuencial, como lo es Arduino, que se basa en c++, en un solo microcontrolador se hace difícil la sincronización de lo mostrado en los elementos de visualización mientras el prototipo está en movimiento, ya que el dispositivo realiza la lectura de un comando a la vez y no es posible ejecutar comandos en paralelo.

El módulo relé actúa como un switch detector de voltaje de la fuente de alimentación de 12V así cuando esta no esté conectada o se corte la energía, el circuito se abre y desconecta la línea de voltaje que va hacia el esp8266 de esta manera solucionamos el inconveniente de que cuando se ingrese una fuente independiente a

este dispositivo puede ser para reprogramarlo no se energicen los demás elementos conectados a la misma línea.

Usamos un ventilador conectado a una línea de 5V con el fin de refrigerar los componentes, pues, debido a las conexiones, es usual que se genere algo de calor en los microcontroladores y si se utiliza la electrónica en un entorno cerrado esto puede generar daños.

En la salida de una de las líneas de 5V se introduce un switch mecánico, esta línea es la que suministra energía a las tarjetas controladoras de la visualización y movimiento al abrir la línea funciona como un paro de emergencia o también como un modo de programación donde es seguro suministrar energía al esp8266 o esp32 de forma independiente sin temor a generar corriente inversa en los reguladores.

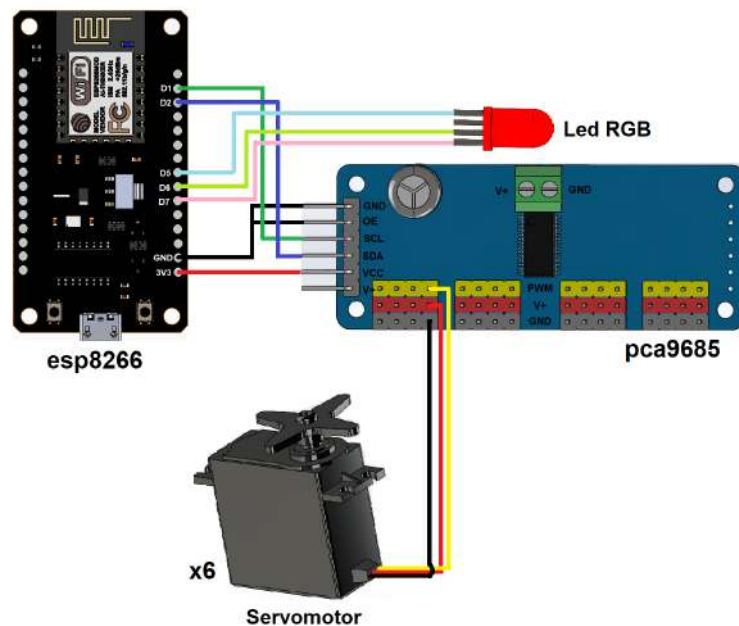


Figura 3.2: Diagrama de control electrónico para servomotores. **Fuente:** Elaboración propia

Conexiones para tarjeta esp8266		
Pin de salida	Dispositivo de llegada	Pin de llegada
D1	pca9685	SCL
D2	pca9685	SDA
D5	led RGB	Azul
D6	led RGB	Verde
D7	led RGB	Rojo
GND	pca9685	GND
3V3	pca9685	VCC

Cuadro 3.2: Conexiones para tarjeta esp8266. **Fuente:** Elaboración propia

Mediante la figura 3.2 se muestra la forma de conexión que integraría la placa esp8266 la cual sería responsable del movimiento de los servomotores de nuestro prototipo, esta tarjeta tendría un tipo de comunicación i2c con el controlador pca9685 el cual actúa como un demux, este dispositivo recibe dos parámetros, el primero es el número de canal de salida y el segundo es el duty o señal PWM que este debe aplicar en el canal indicado.

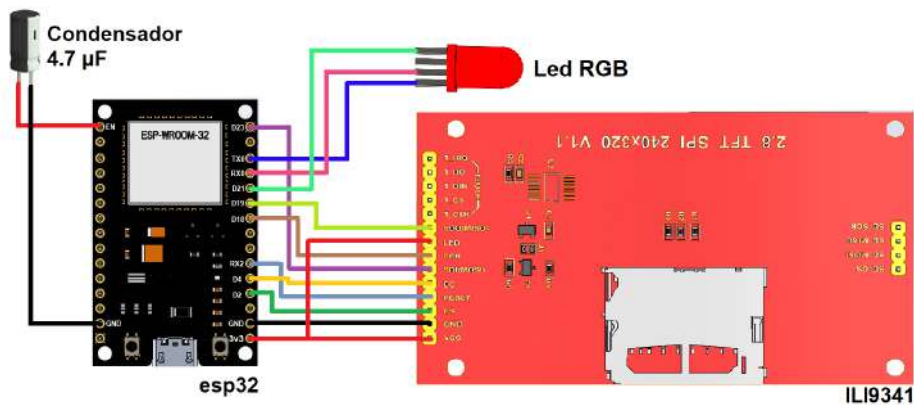


Figura 3.3: Diagrama de control electrónico para herramientas de visualización. **Fuente:** Elaboración propia

Uno de los dos leds RGB también es controlado por el microcontrolador esp8266 conectando los tres pines restantes como se muestra en la tabla 3.2 recordemos que ya se conectó el común como se lo indico en la figura 3.1, cada uno de los tres pines corresponde a un color y mediante un duty de un rango entre 0 y 255 podemos controlar la intensidad de cada uno.

Conexiones para tarjeta esp32		
Pin de salida	Dispositivo de llegada	Pin de llegada
D2	ili9341	CS
D4	ili9341	DC
RX2	ili9341	RESET
D18	ili9341	SCR
D19	ili9341	MISO
D21	led RGB	Verde
RX0	led RGB	Red
TX0	led RGB	Azul
D23	ili9341	MOSI
GND	ili9341	GND
3V3	ili9341	VCC
GND	Condensador	cátodo
EN	Condensador	ánodo

Cuadro 3.3: Conexiones para tarjeta esp32. **Fuente:** Elaboración propia

Las conexiones que se le realizan con la placa esp32 se indica de forma gráfica en la figura 3.3, y se listan tanto los pines de salida como de entrada mediante la tabla 3.3, una de las desventajas de la comunicación SPI que se implementa con la pantalla ili9341 es la cantidad de puertos que requiere para establecerla, aunque su aplicación mediante código de Arduino es intuitiva y de fácil manejo.

Así mismo como el esp8266, el esp32 también controla un led rgb, de manera similar como se lo explico anteriormente, una conexión extra que es necesaria con

el sistema embebido esp32 es la de un capacitor entre Gnd y Enable simulando un retraso que comúnmente se hace presionando el botón boot presente en la tarjeta para cargar un programa, pero, ya que en el momento de tener el prototipado completo no se tendrá la disponibilidad de pulsarlo ni tampoco será eficiente se incorpora el condensador electrolítico.

Cabe resaltar que en el momento de definir los puertos en el código no se realiza mediante el número de la leyenda que este impreso en la placa, sino por el GPIO que le corresponde a cada uno.

3.3. Diseño CAD para ensamble electrónico

Para definir los aspectos exteriores del diseño de la base tenemos que tener en cuenta los elementos con los cuales se interactuara en las etapas de calibración, uso y mantenimiento del dispositivo, pues, además de integrar el sistema de visualización, también se hace uso de un switch de paro de emergencia o modo de programación, un conector hembra correspondiente a la fuente de voltaje y los conectores para la reprogramación de los sistemas embebidos. identificado estos instrumentos se establece el lugar donde se encontraran ubicados para hacer un prototipo visiblemente atractivo. Una salida del sistema que se debe considerar son las conexiones de los servomotores que estarán ubicados en la parte superior de ella.

En la figura 3.4 se muestra una vista explosionada de lo que sería el diseño CAD sin la parte electrónica, este consta de 5 piezas enumeradas de las cuales se expone su funcionalidad y el material en el que se plantea su manufactura mediante la tabla 3.4.

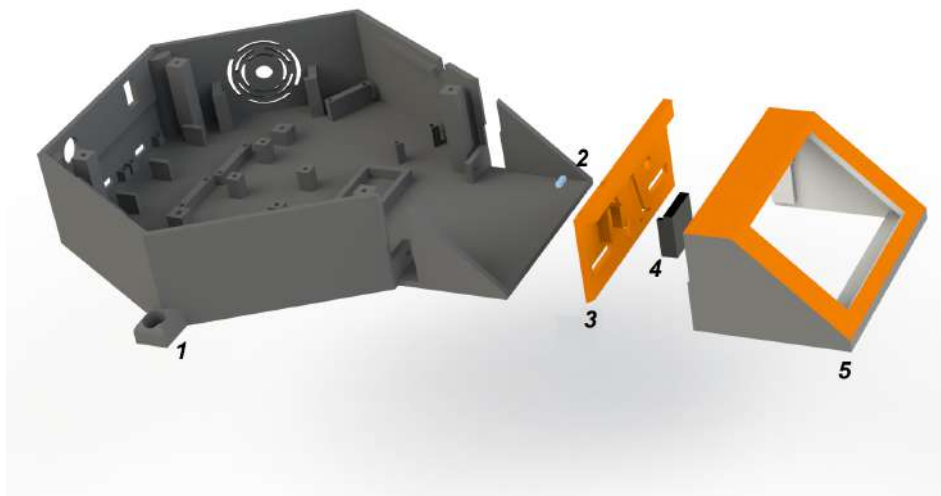


Figura 3.4: Diseño CAD de la base para ensamble electrónico. **Fuente:** Elaboración propia

Piezas de la base del sistema robótico.		
Número	Manufactura	Función
1	Impresión 3D	Servir de alojamiento al diseño electrónico para ser ajustado mediante tornillos tipo M3.
2	Acrílico	Difuminar la luz proveniente del led rgb para mejorar la visualización
3	Impresión 3D	Separa la electrónica general de la pantalla de visualización, ayuda a ordenar los cables provenientes de la comunicación SPI y sirve de acople para el primer led rgb
4	Impresión 3D	Actua como reflectante de la luz del led rgb para que se muestre de mejor manera en el acrílico.
5	Impresión 3D	Pieza en la cual se ensamblara la pantalla ili9341.

Cuadro 3.4: Función y manufactura de las partes de la base del sistema robótico. **Fuente:** Elaboración propia

En el diseño del interior de la base debe estar designado el lugar en que se ubicaran las tarjetas embebidas, además de proporcionar una sujeción para cada

una de estas.

La base, además de ser el lugar de ensamble de la electrónica, también se utilizará como soporte para el cuerpo del robot, por este motivo su forma debe de integrar una solución para facilitar su acople.

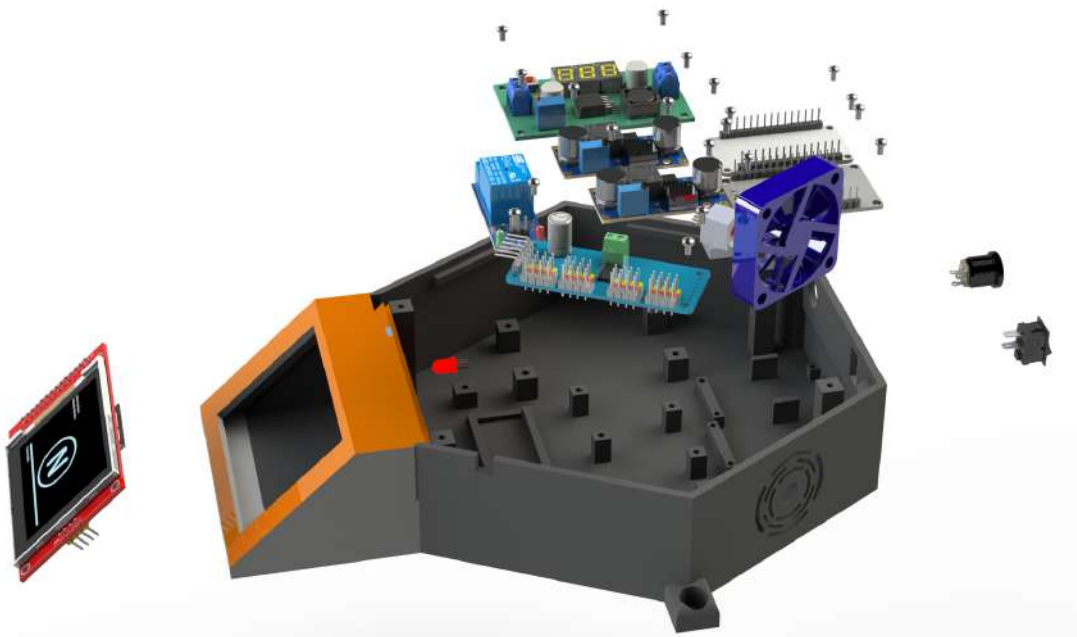


Figura 3.5: CAD del diseño electrónico explosionado. **Fuente:** Elaboración propia

En la vista explosionada de la figura 3.5 se puede apreciar el armado de la base con la forma en que se dispone a incorporar la electrónica, a sí mismo, mediante la figura 3.6 vemos una imagen renderizada de lo que sería el ensamble electrónico terminado, el cual nos brinda una idea bastante cercana de nuestro modelo después de construido físicamente.



Figura 3.6: CAD del diseño electrónico ensamblado. **Fuente:** Elaboración propia

Capítulo 4

Diseño mecánico

En este capítulo mostraremos la forma en que se realizó el diseño CAD de los cinco grados de libertad del sistema robótico, cada uno de sus componentes y su funcionalidad.

Además, se obtendrá una imagen renderizada en la que se muestre la estructura muy similar de como se visualizara en nuestro prototipo ya construido.

En la figura 4.1 se muestra el primer grado de libertad de forma explosionada para darnos una idea más acertada sobre el acople de esta sección de nuestro robot, en este ensamble se hace uso de piezas para ser fabricadas en PLA, y partes obtenidas de forma comercial.

Dos aspectos relevantes que se tuvo en cuenta en este diseño son la disminución máxima posible de fricción para generar un movimiento fluido sin desgastar el servomotor o las piezas, y una buena estabilidad, pues, si no cumple con este último requisito, más adelante en la calibración de ángulos en pro de obtener una posición final certera se podría incurrir en errores.

Para disminuir el rozamiento y generar un punto de apoyo más estable para el cuerpo del robot se hace uso de un rodamiento inserto en la pieza número 2 y mediante un pasador se traslada el movimiento a la parte superior, de esta manera se le da fijeza y estabilidad al primer grado de libertad.

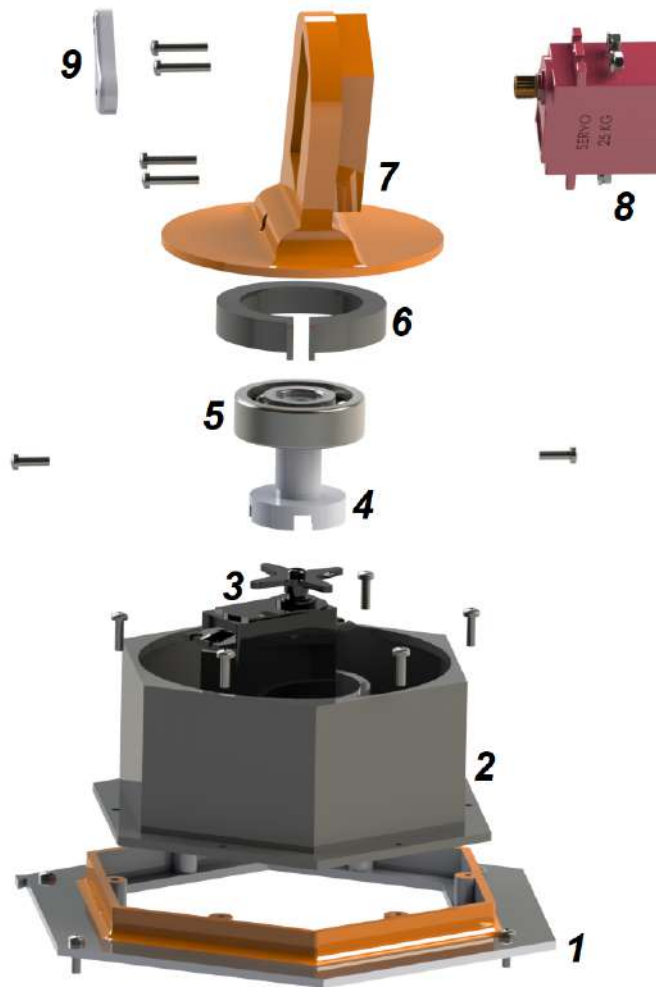


Figura 4.1: CAD de vista explosionada para ensamble del 1.er gdl. **Fuente:** Elaboración propia

A través de la tabla 4.1 se describe de una manera detallada cada uno de los componentes que hacen parte del diseño CAD del primer grado de libertad. La

forma de sujeción empleada para unir cada pieza con otra requiere tornillos de tipo M3 también mostrados mediante la figura 4.1.

Partes que componen el primer grado de libertad.		
Número	Manufactura	Función
1	Impresión 3D	Tapa de acople entre la caja contenedora de la electrónica y el cuerpo del robot.
2	Impresión 3D	Pieza de ensamble para el primer servomotor y el rodamiento.
3	Obtención comercial	Servomotor generador de torque para 1.er gdl.
4	Impresión 3D	Pasador que permite transmitir el movimiento a la parte superior del robot.
5	Obtención comercial	Rodamiento que disminuye la fricción en el movimiento y genera un soporte a los diseños superiores.
6	Impresión 3D	Abrazadera de fijación generadora de estabilidad al sistema.
7	Impresión 3D	Pieza de soporte para el servomotor generador de torque en el 2.do gdl.
8	Obtención comercial	Servomotor generador de torque para 2.do gdl.
9	Obtención comercial	Accesorio de servomotor que facilita el acople con el siguiente eslabón.

Cuadro 4.1: Función y manufactura de las partes que componen el 1.er GDL.
Fuente: Elaboración propia

En la figura 4.2 se muestra una imagen renderizada en la que se aprecia el diseño CAD del primer grado de libertad ya ensamblado y se visualiza como lo obtendremos al momento de realizar el montaje físico de esta sección del cuerpo del robot antropomórfico, también se indica el color con el cual hemos proyectado los acabados de dichas partes para que además de obtener una estructura funcional igualmente sea visiblemente agradable para el usuario.

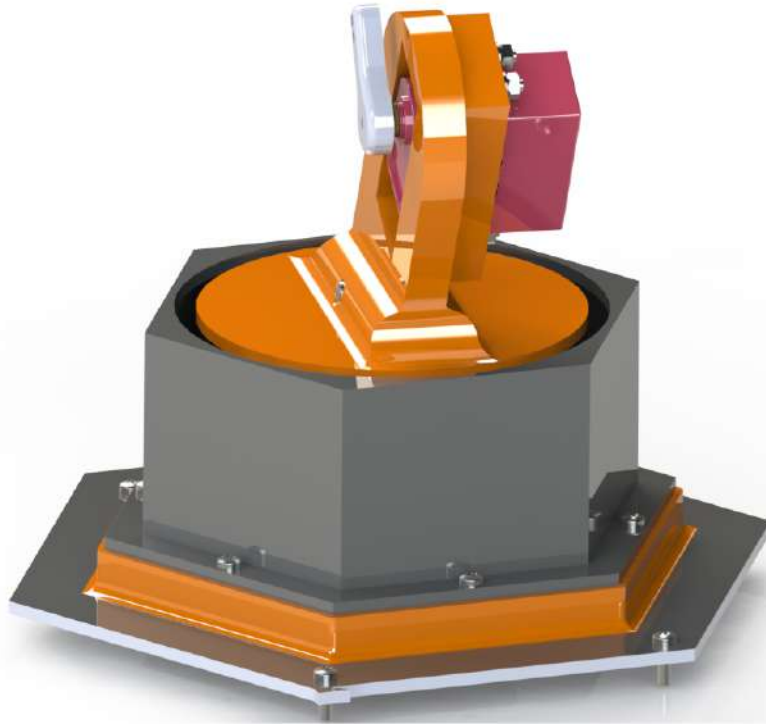


Figura 4.2: Ensamble CAD del 1.er gdl **Fuente:** Elaboración propia

En la imagen de la figura 4.3 se presenta el CAD del segundo, tercero y cuarto grado de libertad de manera explosionada, para dar una idea más clara de las partes que conforman este ensamble, así mismo muestra el primer grado de libertad ya ensamblado, con el fin de seguir una secuencia en la construcción del diseño total del cuerpo del robot e indicar de mejor manera los acoples que se van implementando en el transcurso de este.

De forma general se puede decir que para crear el diseño de esta sección se tuvo en cuenta lo obtenido a través del desenvolvimiento del producto, procedimiento plasmado en el capítulo dos del presente libro. Parámetros como la fácil manufactura, el fácil ensamble, el tamaño de las piezas y la complejidad de cada una, son características fundamentales en esta etapa, así como también la estabilidad y

compacidad que más adelante desarrollan un papel crucial, pues, si nuestro robot careciera de estas cualidades obtendríamos problemas en cuanto a la exactitud en el momento de ajustar su cinemática o ingresar un punto de llegada del efector final en el espacio.

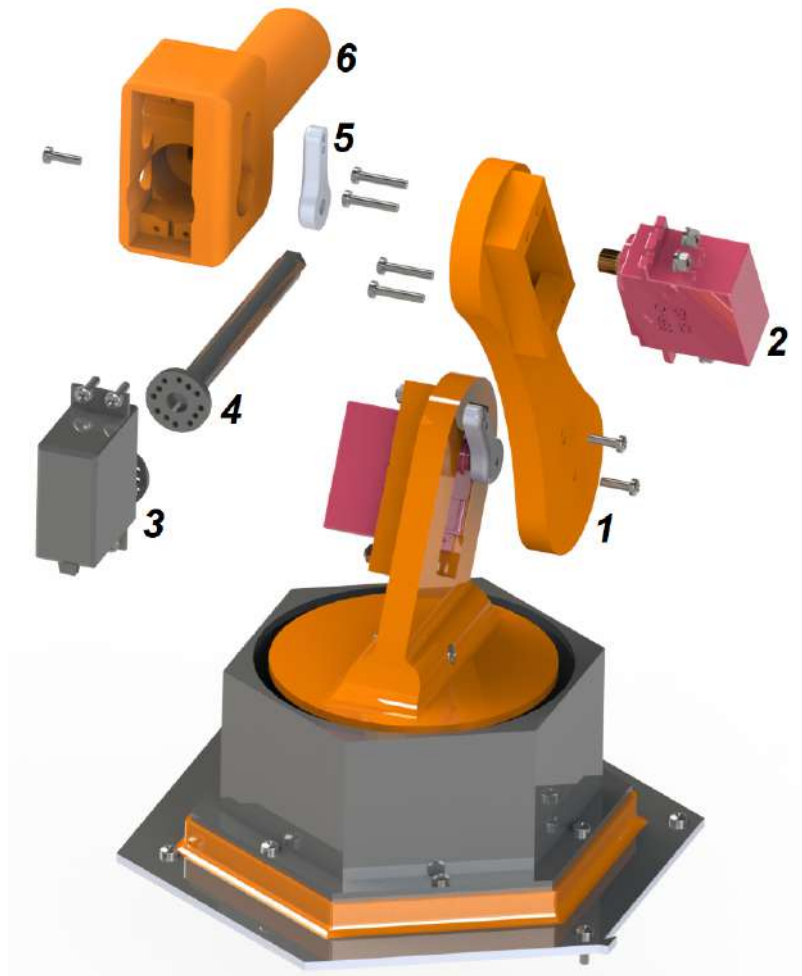


Figura 4.3: CAD de vista explosionada para ensamble del 2.do,3.er y 4.to gdl.
Fuente: Elaboración propia

En cuanto al dimensionamiento del prototipo, también se considera que no debe generar alguna incomodidad al momento de ingresarlo en espacios como una

sala quirúrgica donde habrá personal médico e instrumentos como los indicados en la figura 2.1

Mediante la tabla 4.2 y siguiendo la numeración indicada en la figura 4.3 se describe detalladamente cada uno de los componentes que conforman desde el segundo hasta el cuarto grado de libertad.

Partes que componen el segundo, tercero y cuarto gdl		
Número	Manufactura	Función
1	Impresión 3D	Eslabón para el segundo gdl y apoyo para servomotor del tercer gdl.
2	Obtención comercial	Servomotor generador de torque para 3.er gdl.
3	Obtención comercial	Servomotor generador de torque para 4.to gdl.
4	Impresión 3D	Pasador transmisor de movimiento.
5	Obtención comercial	Accesorio que facilita el acople entre el servomotor 2 y el eslabón 6.
6	Impresión 3D	Eslabón para el tercer grado de libertad.

Cuadro 4.2: Función y manufactura de las partes que componen el 2.do, 3.ro y 4.to GDL. **Fuente:** Elaboración propia

Con el fin de darnos una imagen más acorde a como se verá nuestro diseño físicamente, la figura 4.4 nos exhibe una imagen renderizada del ensamble desde el primer grado de libertad hasta el cuarto, así mismo como la figura 4.2, se puede observar aspectos de acabados como el color.

El método que se utiliza para la unión de las diferentes piezas es realizado mediante tornillos tipo M3,y ya que entre medio del eslabón enumerado en la figura 4.3 con un valor de seis se ensambla el pasador número cuatro, como se puede notar de mejor manera en la figura renderizada, se plantea hacer uso de algún tipo de lubricante disponible en el mercado con el objetivo de disminuir la

fricción y lograr un movimiento fluido sin necesidad de sobre esforzar el servomotor generador de torque para el cuarto grado de libertad



Figura 4.4: Ensamble CAD del sistema con 4 gdl. **Fuente:** Elaboración propia

La vista explosionada de la figura 4.5 expone de buena manera tanto los componentes que conforman el quinto grado de libertad como sus posiciones dentro del ensamble.

Un aspecto primordial a tener en cuenta en este diseño fue el requerimiento obtenido a través de los usuarios de una herramienta intercambiable, puesto que,

al ser el último grado de libertad propuesto, es en este dónde se crea un acople entre nuestro efector final y el cuerpo del robot en sí.

Al igual que en los anteriores grados de libertad, este diseño también tiene que ser compacto para así garantizar uno de los requisitos más importantes referidos en el análisis de Mugde como lo es el alto grado de presión del sistema robótico en su posición.



Figura 4.5: CAD de vista explosionada para ensamble del 5.to gdl. **Fuente:** Elaboración propia

Para lograr una descripción detallada de los diferentes elementos que hacen parte del ensamble del quinto grado de libertad, se enumeran cada uno, como se observa en la figura 4.5 y se listan en la tabla 4.3, describiendo el método de manufactura a emplear y su fusión dentro del sistema robótico. Cabe resaltar que para lograr una sujeción efectiva entre ellos se hace uso de tornillos M3 y M4 con

su respectiva tuerca

Partes que componen el quinto grado de libertad.		
Número	Manufactura	Función
1	Impresión 3D	Pieza receptora del movimiento generado por el servomotor del cuarto grado de libertad y transmitido por el pasador enumerado en la figura 4.3 con un valor de cuatro.
2	Impresión 3D	Pasador transmisor de movimiento hacia la pieza cinco para generar el quinto grado de libertad.
3	Obtención comercial	Servomotor generador de torque para quinto grado de libertad.
4	Impresión 3D	Hexagonal de acople entre la pieza uno y pasador transmisor de movimiento del cuarto grado de libertad.
5	Impresión 3D	Pieza diseñada para la articulación del sistema con el efector final.
6	Impresión 3D	Efector final encargado de sujetar el material quirúrgico.
7	Obtención comercial	Servomotor generador de torque para proporcionar movimiento en el efector final.

Cuadro 4.3: Función y manufactura de las partes que componen el 5.to GDL.

Fuente: Elaboración propia

Una vista que permite analizar más minuciosamente la herramienta o gripper es la mostrada en la figura 4.6, en ella se enumeran sus componentes, además de brindarnos de manera intuitiva el lugar en el que se ensambla cada pieza, apoyándonos igualmente en la figura 4.7 donde puede apreciar el efector final ya ensamblado y la funcionalidad del diseño. Mediante la tabla 4.4 se describe la fusión y el método de manufactura de cada una de las partes citadas en la figura 4.6, pero en términos generales para la creación de este prototipo virtual de gripper se tuvo en cuenta la capacidad de agarre, así como también el dimensionamiento adecuado

para obtener un correcto desempeño del robot antropomórfico y no generar carga innecesaria en los diferentes grados de libertad ocasionando mayor consumo de energía.

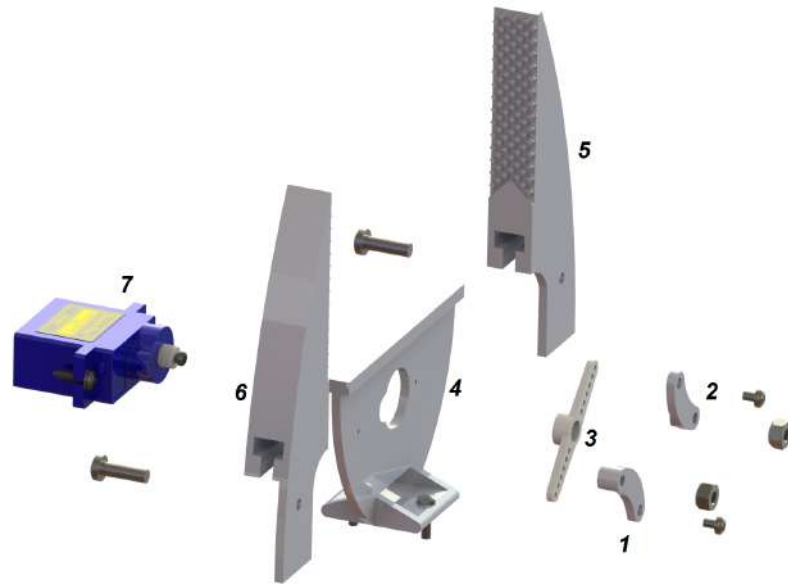


Figura 4.6: Explosionado del efector final mecánico. **Fuente:** Elaboración propia



Figura 4.7: Funcionamiento del efector final mecánico **Fuente:** Elaboración propia

Partes que componen el efector final basado en pinzas.		
Número	Manufactura	Función
1	Impresión 3D	Arco de acople y transmisor de movimiento hacia la pinza número 6.
2	Impresión 3D	Arco de acople y transmisor de movimiento hacia la pinza número 5.
3	Obtención comercial	Accesorio utilizado para acoplar el servomotor y los arcos.
4	Impresión 3D	Pieza de encaje para servomotor y soporte tipo corredera para pinzas.
5	Impresión 3D	Pinza encargada sé la sujeción del material quirúrgico.
6	Impresión 3D	Pinza encargada sé la sujeción del material quirúrgico.
7	Obtención comercial	Servomotor generador de torque.

Cuadro 4.4: Función y manufactura de las partes que componen el efector final mecánico. **Fuente:** Elaboración propia

De acuerdo con lo expresado, en el capítulo dos del presente libro, uno de los requerimientos que se debe de tener presente en el diseño del prototipo, es la característica de una herramienta intercambiable, de esta manera se pueden manipular diferentes tipos de instrumentos según su forma y sus propiedades.

Además de la pinza presentada anteriormente como efector final, se propone una herramienta de tipo magnética que cuente con la capacidad de interactuar con los instrumentos quirúrgicos de acero inoxidable. Ya que dicho material se clasifica como ferromagnético y por ello es posible imantarlo.

Un atributo que destaca en el momento del diseño e implementación de un efector intercambiable, es también la posibilidad de la integración de un último grado de libertad. Este podría ser de gran utilidad en diferentes ocasiones, como

por ejemplo en las cuales además de depositar el instrumento en una posición, también queremos indicar su orientación.

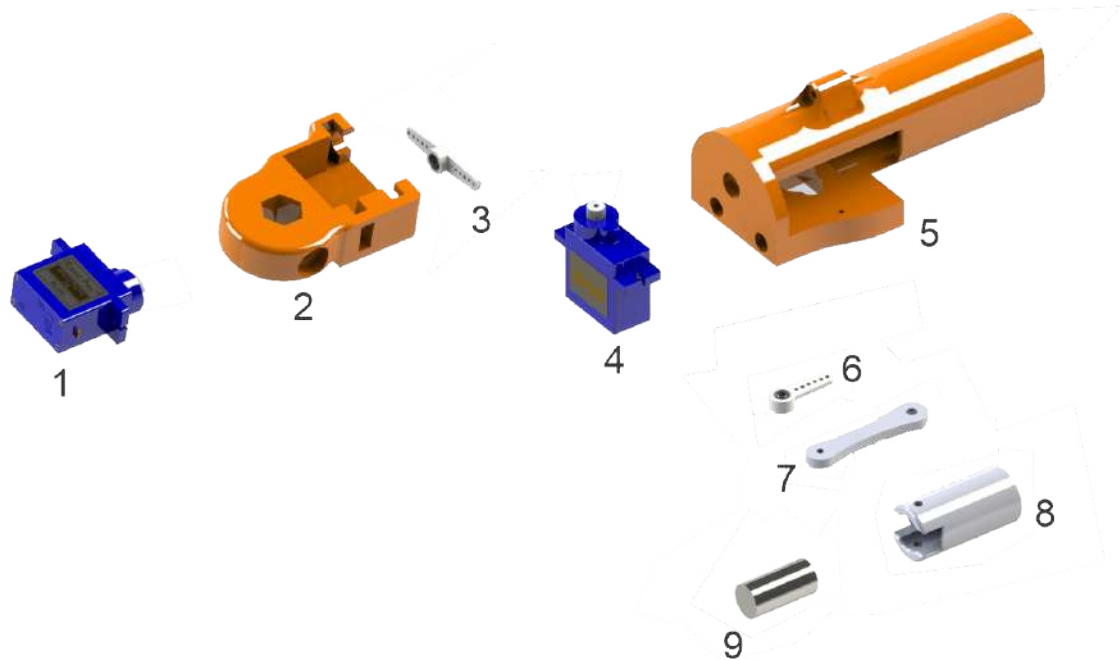


Figura 4.8: Explosionado del efector final de tipo magnético. **Fuente:** Elaboración propia

A través de la figura 4.8 se indica los diferentes componentes que hacen parte del diseño de la gripper de tipo magnética. Como elemento principal de su funcionamiento se tiene un imán de neodimio que mediante un sistema biela manivela adquiere la capacidad de sujetar, transportar y depositar el instrumental, en el lugar que se indique según la necesidad que requiera el proceso de asistencia en instrumentación quirúrgica.

Con el fin de ilustrar de una mejor manera el funcionamiento de este sistema biela manivela se presenta la imagen 8.9, que por intermedio de la tabla 4.5 y de la numeración indicada en la figura de los componentes de la herramienta magnética se define la funcionalidad de cada uno de estos; su manufactura o forma de

obtención.

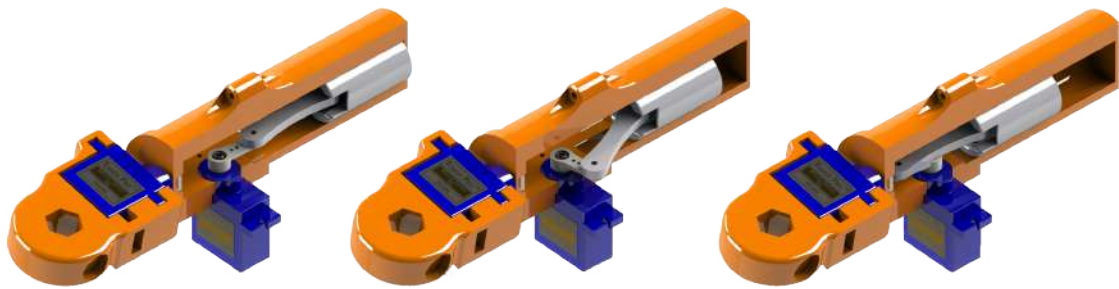


Figura 4.9: CAD del funcionamiento del efector magnético. **Fuente:** Elaboración propia

Partes que componen el efector final magnético.		
Número	Manufactura	Función
1	Obtención comercial	Servomotor generador de torque para la adición del gdl.
2	Impresión 3D	Pieza de ensamble para el servomotor número 1
3	Obtención comercial	Accesorio usado para acoplar el servomotor número 1 con la guía o pieza número 5
4	Obtención comercial	Servomotor generador de torque para movimiento biela manivela.
5	Impresión 3D	Guía del sistema biela manivela y soporte para el servomotor número 4
6	Obtención comercial	Accesorio de servomotor utilizado como manivela.
7	Impresión 3D	Pieza que actúa como biela.
8	Impresión 3D	Émbolo y cubierta del imán de neodimio.
9	Obtención comercial	Imán de neodimio.

Cuadro 4.5: Función y manufactura de las partes que componen el efector magnético. **Fuente:** Elaboración propia

Para finalizar con el ensamble del prototipo virtual, unimos el diseño CAD de la parte electrónica con el de los diferentes grados de libertad, tal como se logra apreciar en la figura número 4.11, de esta manera culminamos el diseño mediante el software SolidWorks en el cual logramos apreciar de una forma exacta aspectos como lo puede ser el dimensionamiento del prototipo indicado en la figura 4.10, así mismo este software nos permite exportar archivos en formato STL, a partir de las piezas generadas, los cuales son necesarios para la manufactura por medio de impresión 3D.

Las medidas obtenidas a partir del modelado son de gran importancia en el cálculo cinemático del robot, el cual se estará tratando en el siguiente capítulo.

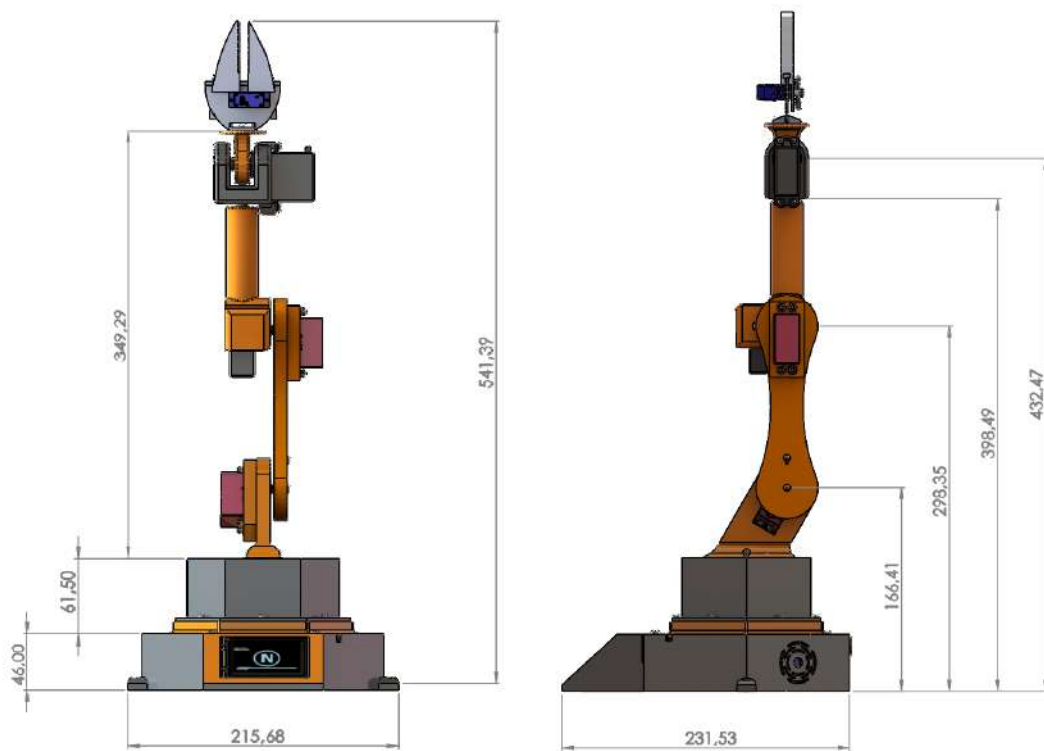


Figura 4.10: Dimensionamiento del prototipo robótico. **Fuente:** Elaboración propia



Figura 4.11: CAD de vista explosionada entre los gdl y base de electrónica. **Fuente:** Elaboración propia



Figura 4.12: CAD de prototipo robótico para la asistencia en instrumentación quirúrgica. **Fuente:** Elaboración propia

Capítulo 5

Ensamble y calibración

En este capítulo se tratará los temas relacionados con la obtención de la estructura física del robot antropomórfico, así nos adentraremos en temas como impresión 3D, ensamble de piezas del prototipo, cálculos cinemáticos, e implementación de estos en software para la simulación y análisis de trayectorias, en consecuencia, se plantea una interfaz por medio de Blynk que permita la calibración del modelo simulado con la estructura física de una forma intuitiva.

5.1. Manufactura por impresión 3D

Existen varias técnicas de modelados 3D, entre ellas tenemos: la fabricación con filamento fundido, fabricación por Estereolitografía, y el sinterizado selectivo por láser. Cada una de estas tiene sus ventajas y desventajas según la aplicación que deseemos darle, nosotros acorde a lo planteado en el capítulo 2, donde se trató el desenvolvimiento del producto, elegimos la opción de la fabricación mediante

filamento fundido, ya que son características relevantes de este método la fácil utilización, el bajo costo, y su rapidez.

El proceso inicia con la importación de las piezas diseñadas en el software CAD a un programa especializado para la configuración y el control de la impresión, en este caso utilizaremos Cura como lo muestra la figura 5.1.

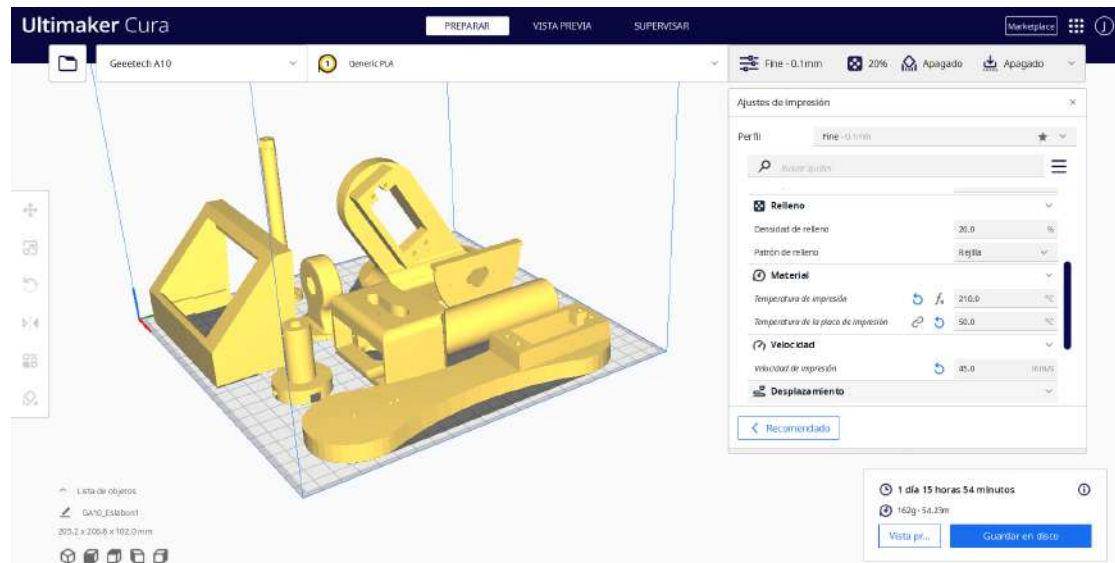


Figura 5.1: Piezas importadas de SolidWorks en formato STL a software Cura .
Fuente: Elaboración propia

Cura nos permite configurar parámetros importantes como la velocidad de impresión, la densidad de relleno de la pieza, forma y generación de soporte, temperatura de la superficie donde se deposita el material fundido, y la temperatura de fundido o extrusión. Al finalizar la configuración, el software nos indica el tiempo y material estimado para la realización del procedimiento. Cabe resaltar que cada pieza según su forma y su función necesitara configuraciones diferentes, por ello se recomienda hacer la impresión por separado, también hay que tener en cuenta la holgura o tolerancia en las juntas entre piezas, para este diseño se le dio un valor en el rango de 0.3 y 0.5 según la forma de la unión.

Tiempo y material de manufactura por impresión 3D			
Sección de ubicación	Numeración	Tiempo de impresión	Peso
Base del sistema robótico (Figura 3.4)	1	1 día 12 horas 34 minutos	216 g (72.28 m)
	3	2 horas 1 minuto	13 g (4.37 m)
	4	15 minutos	2 g (0.52 m)
	5	7 horas 37 minutos	52 g (17.52 m)
Primer grado de libertad (Figura 4.1)	1	10 horas 4 minutos	64 g (21.4 m)
	2	17 horas 46 minutos	118 g (39.43 m)
	4	1 hora 47 minutos	11 g (3.7 m)
	6	1 hora 1 minuto	8 g (2.54 m)
	7	12 horas 18 minutos	82 g (27.5 m)
Segundo, tercero y cuarto gdl (Figura 4.3)	1	9 horas 49 minutos	70 g (23.59 m)
	4	1 hora 54 minutos	9 g (2.99 m)
	6	14 horas 16 minutos	89 g (29.98 m)
Quinto grado de libertad (Figura 4.5)	1	7 horas 12 minutos	44 g (14.64 m)
	2	42 minutos	3 g (1.09 m)
	4	10 minutos	1 g (0.33 m)
	5	2 horas 44 minutos	17 g (5.67 m)
Efector final (Figura 4.6)	1	4 minutos	0.5 g (0.1 m)
	2	4 minutos	0.5 g (0.1 m)
	4	1 hora 47 minutos	9 g (2.88 m)
	5	2 horas 0 minutos	11 g (3.75 m)
	6	2 horas 0 minutos	11 g (3.75 m)
Total de tiempo y material de impresión:		5 días 12 horas 5 minutos	831 g (278.03 m)

Cuadro 5.1: Tiempo y material de manufactura por impresión 3D. **Fuente:** Elaboración propia

En la tabla 5.1 observamos el tiempo y material de impresión de cada una de las partes del prototipo robótico, enumeradas según las figuras indicadas en el recuadro de la primera columna. La impresora utilizada como herramienta de manufactura es de la serie Geeetech A10.

El paso siguiente a la impresión es el curado y acabado de las piezas, donde se extrae el soporte generado, y se pulen imperfecciones por medio de procedimientos como el lijado, estando las piezas en óptimas condiciones se les da el acabado según lo planteado en el diseño por medio de varias capas de pintura.

5.2. Ensamble de componentes

Ya con las piezas finalizadas procedemos al ensamblaje electromecánico del prototipo físico es así como comenzamos con la conexión de la parte electrónica y su instalación en la base del robot esto por medio de tornillos tipo M3, una vez terminado se realiza pruebas rápidas del correcto funcionamiento de todos los componentes, esto se logra observar de forma clara mediante las figuras 5.2 y 5.3.

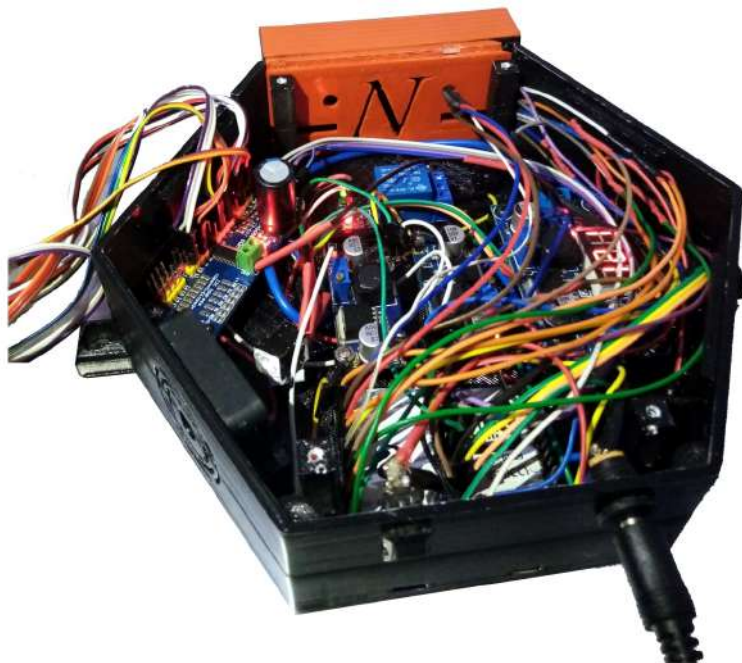


Figura 5.2: Ensamble físico de la base del robot en una vista posterior. **Fuente:** Elaboración propia

Las características con las que cuenta la base del robot son planteadas tras el análisis del desenvolvimiento del proyecto, pues cuenta con un switch que opera como un paro de emergencia y modo programación a la vez, así mismo cuenta

con dos salidas para conexión micro USB con las cuales de manera fácil se puede reprogramar el prototipo sin necesidad de desmontar alguna parte del robot. En cuanto al suministro de energía es necesario una fuente de poder con un voltaje igual a 12 V y un amperaje de 5A. Además, se integró una pantalla SPI y dos leds con el fin de una interacción agradable entre hombre máquina y embellecer la estructura.



Figura 5.3: Ensamble físico y validación del funcionamiento de la base del robot en una vista frontal. **Fuente:** Elaboración propia

Para el armado del cuerpo del robot se utilizó tornillos tipo M3 y M4 como se lo propuso en el diseño virtual mediante software CAD, se instaló los servomotores para cada articulación, además de fijar los eslabones, y al finalizar el ensamble se obtuvo una estructura estable con la cual es posible implementar un software cinemático y por consiguiente controlar la posición exacta de cada articulación y

efector final según las trayectorias a realizar.

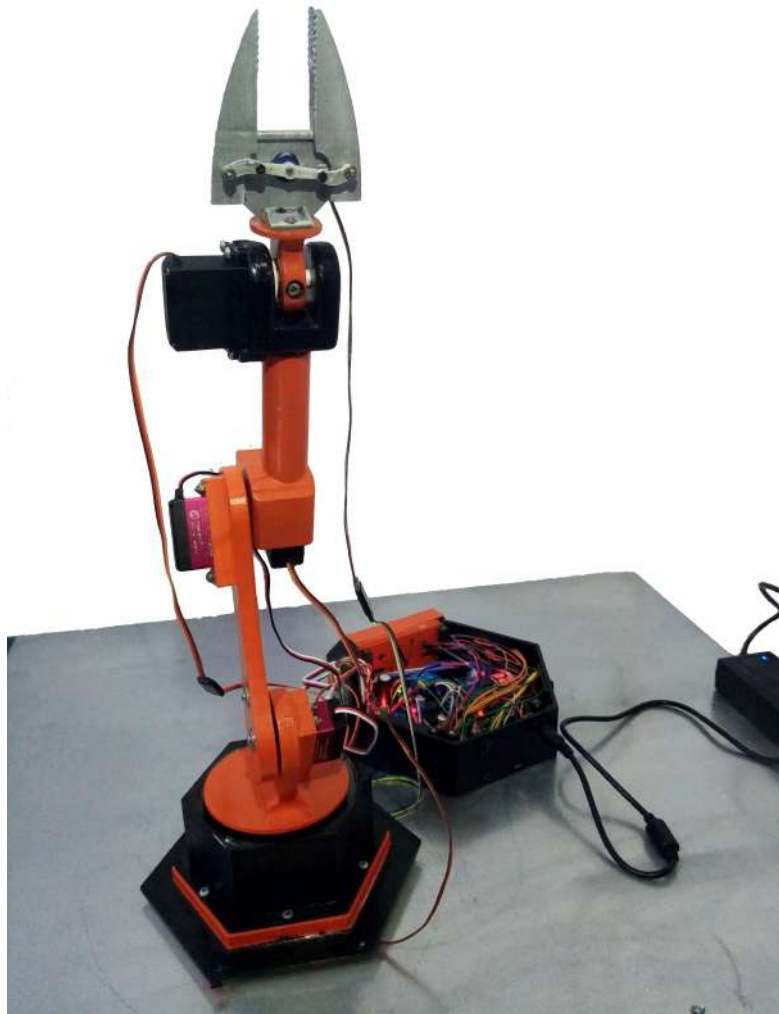


Figura 5.4: Cuerpo y base del robot antropomórfico. **Fuente:** Elaboración propia

Mediante la figura 5.4 se puede identificar tanto el cuerpo del robot y la base de forma separada en óptimas condiciones y listas para ser ensambladas y dar por terminada la construcción física del prototipo, como se indica en la figura 5.5.



Figura 5.5: Robot antropomórfico completamente ensamblado. **Fuente:** Elaboración propia

5.3. Modelado cinemático

En este apartado se da comienzo al estudio de los movimientos del robot, y se describirá de forma analítica el movimiento espacial del mismo, estableciendo relaciones entre la posición y orientación del efector final del prototipo con las coordenadas articulares.

5.3.1. Cinemática directa

Con el fin de facilitar la visualización de la estructura del robot, se crea un modelo simplificado presentado en la figura 5.6, de este se extrae las medidas necesarias y se identifica la relación existente entre las diferentes articulaciones, de manera que se obtengan los parámetros de Denavit-Hartenberg indicados en la tabla 5.2.

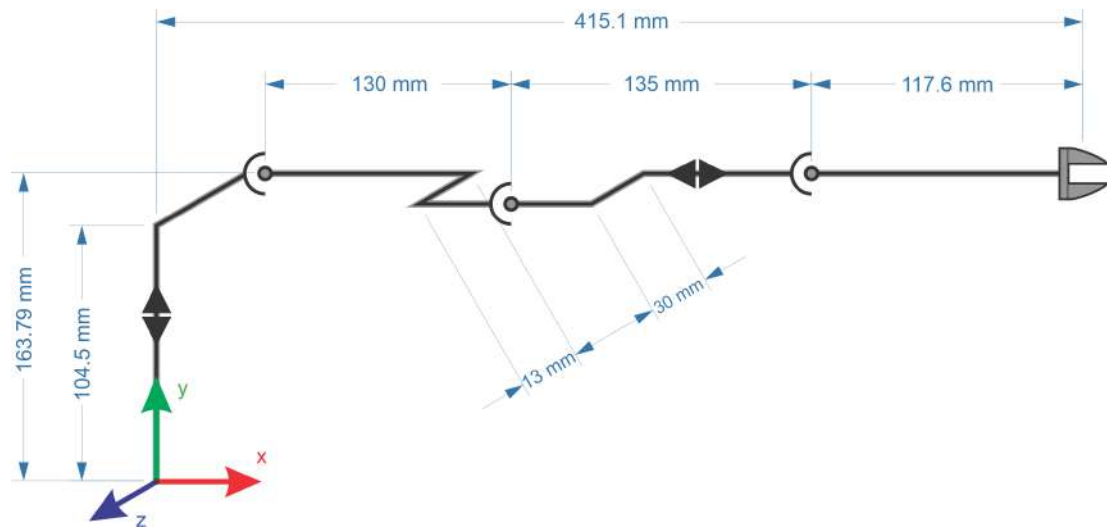


Figura 5.6: Estructura del robot antropomórfico simplificada para cálculos cinemáticos. **Fuente:** Elaboración propia

$$L1 = 163,79$$

$$L3 = 130$$

$$L2 = 30 - 13,5 = 16,5$$

$$L4 = 135$$

$$L3A = 382,6 - 415,1 = 32,5$$

$$L5 = 117,6$$

Los parámetros DH hacen relación a cuatro criterios indicados por convención para generar unión entre marcos de referencia a los eslabones de una cadena cinemática.

- $a \rightarrow$ Longitud de la línea normal común entre los dos ejes z , que es perpendicular a ambos ejes
- $\alpha \rightarrow$ Ángulo de rotación para la normal común
- $d \rightarrow$ Desplazamiento a lo largo del eje z en la dirección normal, desde el principal a secundario
- $\theta \rightarrow$ Ángulo de rotación del eje x a lo largo del eje z anterior

Parámetros DH				
	θ	d	a	α
$0 \rightarrow 1$	$q1 + 90^\circ$	$L1$	$L3A$	90°
$1 \rightarrow 2$	$-q2$	$-L2$	$L3$	0°
$2 \rightarrow 3$	$q3 + 90^\circ$	0	0	90°
$3 \rightarrow 4$	$q4 + 180^\circ$	$L4$	0	90°
$4 \rightarrow 5$	$q5 + 90^\circ$	0	$L5$	0°

Cuadro 5.2: Parámetros de Denavit-Hartenberg para prototipo robótico. **Fuente:** Elaboración propia

Ya identificado los parámetros DH, reemplazamos estos valores en la ecuación 1.1 indicada en el capítulo uno del presente libro, con la cual obtenemos la posición y orientación del marco de referencia i con respecto al $i-1$ es por ello que el número de veces que se aplica este proceso es el mismo de los grados de libertad que posee el prototipo.

Mediante la figura 5.7 se muestra de manera detallada la matriz de Denavit-Hartenberg realizada para los cinco GDL del robot antropomórfico a partir de los parámetros DH. Las tres primeras filas de las tres primeras columnas nos indican la orientación, mientras que las tres primeras filas de la cuarta columna corresponden

a las coordenadas de la posición, con respecto a la cuarta fila tenemos la perspectiva y el escalado.

$${}^0A_1 = \begin{bmatrix} \cos(q1 + 90) & -\cos(90)\sin(q1 + 90) & \sin(90)\sin(q1 + 90) & L3A * \cos(q1 + 90) \\ \sin(q1 + 90) & \cos(90)\cos(q1 + 90) & -\sin(90)\cos(q1 + 90) & L3A * \sin(q1 + 90) \\ 0 & \sin(90) & \cos(90) & L1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1A_2 = \begin{bmatrix} \cos(-q2 + 0) & -\cos(0)\sin(-q2 + 0) & \sin(0)\sin(-q2 + 0) & L3 * \cos(-q2 + 0) \\ \sin(-q2 + 0) & \cos(0)\cos(-q2 + 0) & -\sin(0)\cos(-q2 + 0) & L3 * \sin(-q2 + 0) \\ 0 & \sin(0) & \cos(0) & -L2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2A_3 = \begin{bmatrix} \cos(q3 + 90) & -\cos(90)\sin(q3 + 90) & \sin(90)\sin(q3 + 90) & 0 * \cos(q3 + 90) \\ \sin(q3 + 90) & \cos(90)\cos(q3 + 90) & -\sin(90)\cos(q3 + 90) & 0 * \sin(q3 + 90) \\ 0 & \sin(90) & \cos(90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3A_4 = \begin{bmatrix} \cos(q4 + 180) & -\cos(90)\sin(q4 + 180) & \sin(90)\sin(q4 + 180) & 0 * \cos(q4 + 180) \\ \sin(q4 + 180) & \cos(90)\cos(q4 + 180) & -\sin(90)\cos(q4 + 180) & 0 * \sin(q4 + 180) \\ 0 & \sin(90) & \cos(90) & L4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4A_5 = \begin{bmatrix} \cos(q5 + 90) & -\cos(0)\sin(q5 + 90) & \sin(0)\sin(q5 + 90) & L5 * \cos(q5 + 90) \\ \sin(q5 + 90) & \cos(0)\cos(q5 + 90) & -\sin(0)\cos(q5 + 90) & L5 * \sin(q5 + 90) \\ 0 & \sin(0) & \cos(0) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 5.7: Matrices de Denavit-Hartenberg para robot antropomórfico.. **Fuente:** Elaboración propia

Haciendo uso de la ecuación 1.2 indicada en el capítulo uno, y por medio de la multiplicación de las matrices descritas en la figura 5.7, obtenemos la matriz de transformación homogénea que relaciona el sistema coordenado del efector final con respecto al de la base del robot, tal como lo indica la siguiente igualdad.

$${}^0A_5 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5$$

5.3.2. Cinemática inversa

Existen diferentes métodos para el cálculo de la cinemática inversa, cada uno de estos busca determinar los valores que debe de adquirir cada articulación para que el efector final se posicione en determinadas coordenadas, es decir que a partir de la matriz de transformación homogénea del efector final con respecto al sistema coordinado cero, obtener las variables de cada articulación.

Ya que nuestro robot cuenta con un número de grados de libertad superior a tres, se hace necesario un desacople cinemático, el cual consiste en identificar el punto de la muñeca como se logra observar en la imagen 5.8, y dividir de esta forma los cálculos en dos partes.

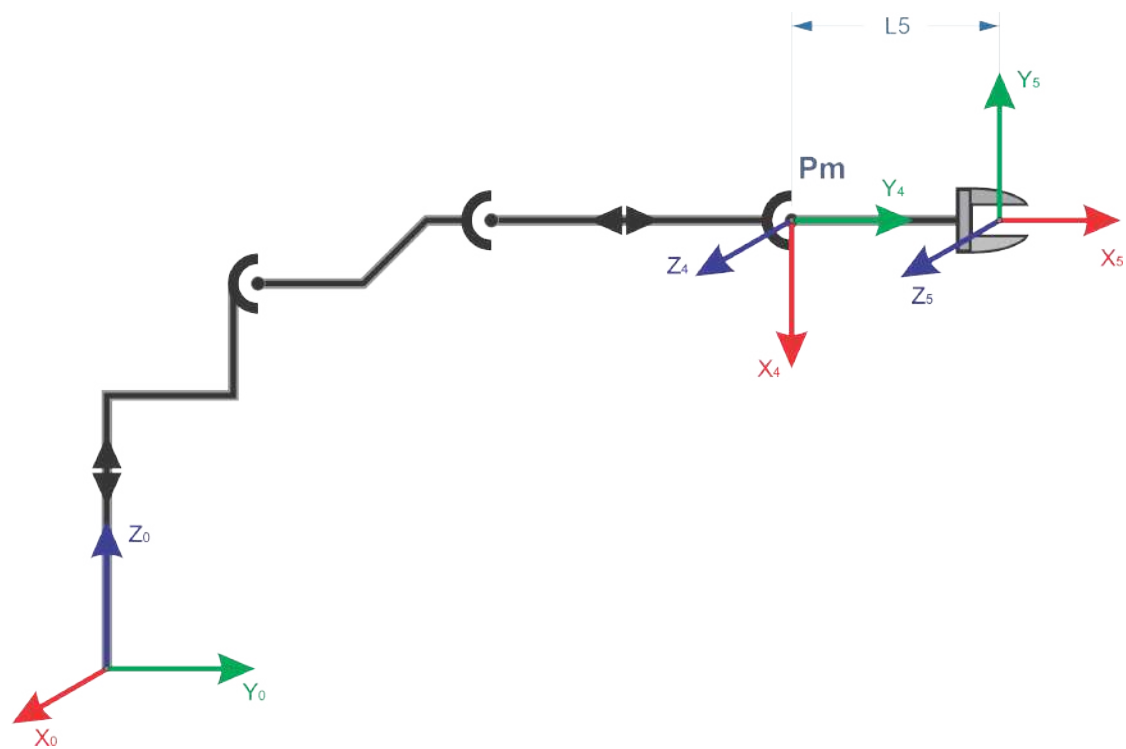


Figura 5.8: Punto de la muñeca en la estructura simplificada del robot. **Fuente:** Elaboración propia

A partir de la figura 5.8 se logra extraer de forma analítica la posición del punto de la muñeca necesario para el análisis matemático:

$${}^0\bar{P}_m = {}^0\bar{P}_5 + \bar{P}_{aux}$$

$$\bar{P}_{aux} = |L_5|(-X_5)$$

$${}^0\bar{P}_m = {}^0\bar{P}_5 + |L_5|(-X_5)$$

Referenciado el punto de la muñeca podemos realizar el desacople observado en la figura 5.9, y aplicar un modelo geométrico para resolver el problema cinemático inverso de la primera parte del robot, con el fin de encontrar el valor de q_1 , q_2 y q_3 correspondientes las primeras tres articulaciones.

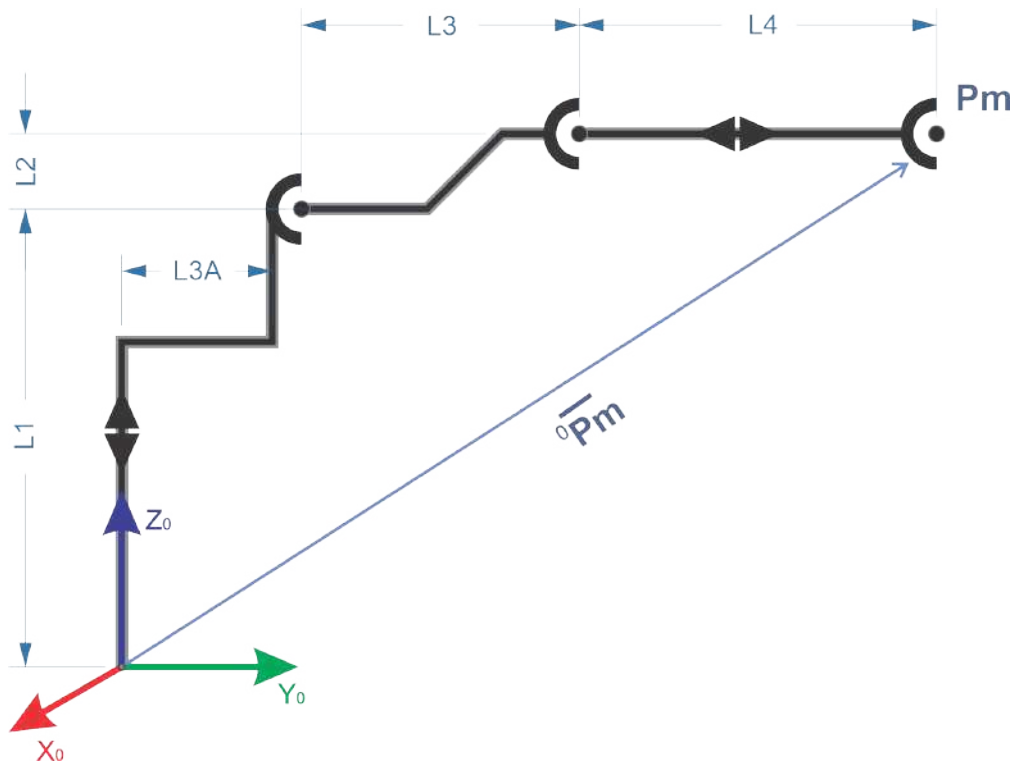


Figura 5.9: Desacople cinemático en la estructura simplificada del robot. **Fuente:** Elaboración propia

Método geométrico para la obtención de la articulación q1:

$$\begin{aligned}
 m &= \sqrt{(P_{my})^2 + (P_{mx})^2} & \beta_1 &= \text{atan2}(L_2, D) \\
 m^2 &= D^2 + (L_2)^2 & \tan(\alpha_1) &= \frac{P_{mx}}{P_{my}} \\
 D &= \sqrt{m^2 - (L_2)^2} & \alpha_1 &= \text{atan2}(-P_{mx}, P_{my}) \\
 \tan(B_1) &= \frac{L_2}{D} & q_1 &= \alpha_1 - \beta_1
 \end{aligned}$$

Método geométrico para la obtención de la articulación q3:

$$\begin{aligned}
 J^2 &= (L_3)^2 + (L_4)^2 - 2L_3L_4\cos(\theta) & \cos(q3) &= \frac{J^2 - (L_3)^2 - (L_4)^2}{2L_3L_4} \\
 \theta &= 180 - q3 & \cos(q3)^2 + \sin(q3)^2 &= 1 \\
 \cos(180 - q3) &= -\cos(q3) & \sin(q3) &= \pm\sqrt{1 - \cos(q3)^2} \\
 J^2 &= (L_3)^2 + (L_4)^2 + 2L_3L_4\cos(q3) & q3 &= \text{atan2}(\sin(q3), \cos(q3))
 \end{aligned}$$

Método geométrico para la obtención de la articulación q2:

$$\begin{aligned}
 \beta_2 &= \text{atan2}(L_4\sin(q3), L_3 + L_4\cos(q3)) \\
 \tan(\alpha_2) &= \frac{P_{mz} - L_1}{D - L_3A} \\
 \alpha_2 &= \text{atan2}(P_{mz} - L_1, D - L_3A) \\
 q2 &= \alpha_2 - \beta_2
 \end{aligned}$$

Para la obtención del valor articular correspondiente al cuarto grado de libertad lo calculamos con respecto al sistema de referencia S3 como se muestra en el

apartado a de la figura 5.10 y su solución analítica se presenta de la siguiente forma:

$$Z_4.Y_3 = |Z_4||Y_3|\cos(q_4)$$

$$\cos(90 + q_4) = -\sin(q_4)$$

$$Z_4.Y_3 = \cos(q_4)$$

$$\sin(q_4) = -Z_4.X_3$$

$$Z_4.X_3 = |Z_4||X_3|\cos(90 + q_4)$$

$$q_4 = \text{atan2}(\sin(q_4), \cos(q_4))$$

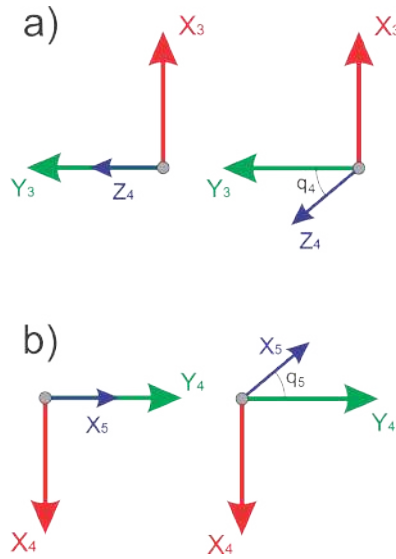


Figura 5.10: Sistemas de referencia S3, S4 Y S5, para solución de cinemática inversa. **Fuente:** Elaboración propia

En cuanto al quinto grado de libertad, realizamos el mismo algoritmo de obtención de q_4 , pero esta vez con respecto al sistema de referencia S4, como se muestra en el apartado b de la figura 5.10.

$$X_5.Y_4 = |X_5||Y_4|\cos(q_5)$$

$$\cos(90 + q_5) = -\sin(q_5)$$

$$X_5.Y_4 = \cos(q_5)$$

$$\sin(q_5) = -X_5.X_4$$

$$X_5.X_4 = |X_5||X_4|\cos(90 + q_5)$$

$$q_5 = \text{atan2}(\sin(q_5), \cos(q_5))$$

5.4. Implementación de cinemática en software de simulación

Antes de integrar el modelo cinemático descrito anteriormente, al robot físico, es importante construir una simulación de verificación donde nos aseguramos de la confiabilidad de este. Para ello se utilizó el software MATLAB, donde con ayuda de la cinemática directa calculada en el apartado anterior, se implementa de forma virtual la estructura simplificada mostrada en la figura 5.6, y a partir de ella ensamblar nuevamente nuestro prototipo, importando cada una de las piezas en formato STL desde SolidWorks.

El resultado se logra observar mediante la figura 5.11 en la que se indica el prototipo, el sistema coordenado del efector final y el valor de las coordenadas (x , y , z) del mismo.

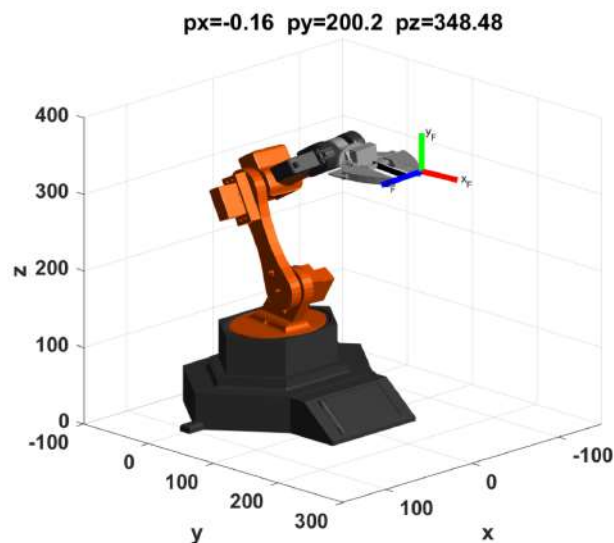


Figura 5.11: Prototipo de robot antropomórfico en software MATLAB. **Fuente:** Elaboración propia

Construido el modelo virtual se tiene como objetivo la planeación de trayectorias, pero para lograr ello se debe de reescribir los cálculos de la cinemática inversa calculada anteriormente, en el entorno de MATLAB así por medio de la matriz de transformación homogénea del efector final lograremos obtener los valores de cada articulación.

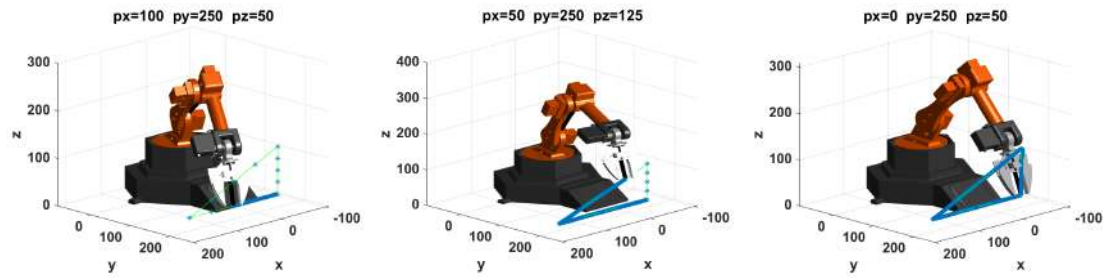


Figura 5.12: Planeación de trayectorias mediante interpolador cúbico en software MATLAB. **Fuente:** Elaboración propia

Si introducimos dos referencias finales respecto al efector final en la función de la cinemática inversa, ella nos obtendrá dos valores para cada articulación, es decir, un valor inicial y otro final, el cual podemos unir por medio de interpolación cúbica.

Mediante la figura 5.12 se describe el recorrido del prototipo virtual en una trayectoria que introduce una secuencia de puntos para la obtención de un triángulo. En la figura 5.13 se indica de una forma detallada la variación en el valor de cada articulación según el movimiento descrito.

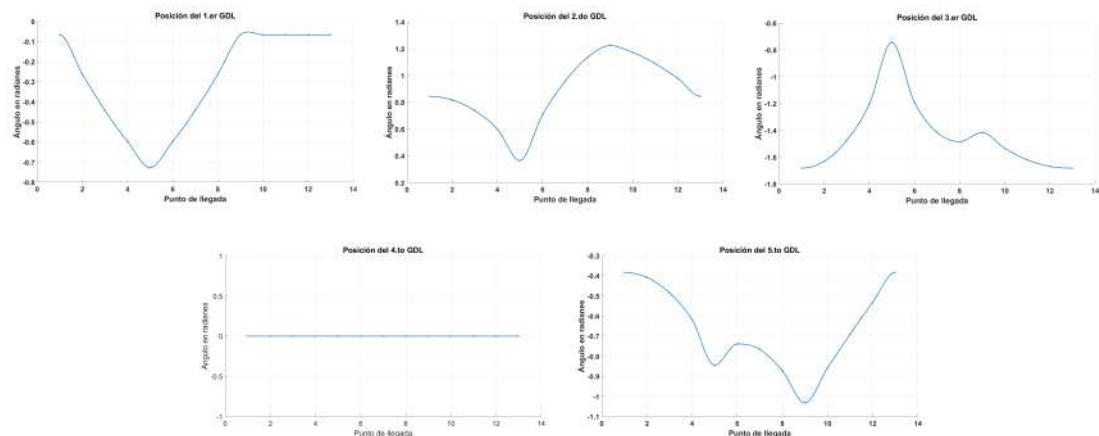


Figura 5.13: Posición articular de cada grado de libertad en trayectoria simulada.
Fuente: Elaboración propia

5.5. Calibración del modelo simulado con la estructura física

Por calibración nos referimos a la configuración para la concordancia entre el modelo simulado anteriormente y la estructura física, pues, en el momento de ensamblar la estructura física se hace difícil establecer el ángulo de inicio exacto de un servomotor además de que estos varían dependiendo de la frecuencia PWM que se les asigne, cabe aclarar que para cada servomotor suele ser ligeramente variante el número de pulsos PWM correspondiente a un ángulo, cuando estamos trabajando con la tarjeta pca 9685, puesto que está trabaja a 12 bits es decir 4095 pulsos, por ello se hace necesario en el momento de realizar el código en para el sistema embebido ESP8266, convertir de pulsos a ángulos para cada servomotor de forma individual.

Además de establecer el ángulo 0 y 180 del servomotor se debe de calibrar con el punto inicial indicado en la cinemática de la simulación, para ello se creó

una función denominada ajuste de ceros donde se establece las configuraciones y restricciones necesarias como por ejemplo el valor máximo posible y mínimo, que son 180 y 0 respectivamente. Para verificar el correcto funcionamiento antes de la implementación de trayectorias en el prototipo, se diseñó una conexión con la plataforma Blynk como se lo muestra en la figura 5.14.



Figura 5.14: Conexión entre brazo robótico y Blynk. **Fuente:** Elaboración propia

Blynk es una plataforma de Internet of things (IOT) que permite la conexión de diferentes dispositivos a la nube para controlarlos y supervisarlos por medio del diseño de aplicaciones.

En nuestro modelo se controla la posición articular del prototipo desde un ordenador o una aplicación móvil, sin necesidad de estar conectados en la misma red, esto tiene como fundamental ventaja la operabilidad desde cualquier parte del mundo, la aplicación diseñada para los dispositivos de supervisión la podemos observar en la figura 5.15 cuenta con seis sliders con los que se puede cambiar el

estado de los cinco grados de libertad y de las pinzas de agarre en el efector final, esto nos facilita junto con la simulación de la cinemática directa establecer valores articulares y comparar el punto de llegada simulado y real.

Seguros de la concordancia en el punto de llegada entre la simulación y la estructura física, nos disponemos al diseño del software del prototipo y a la implementación del modelo cinemático y de trayectorias en el mismo.

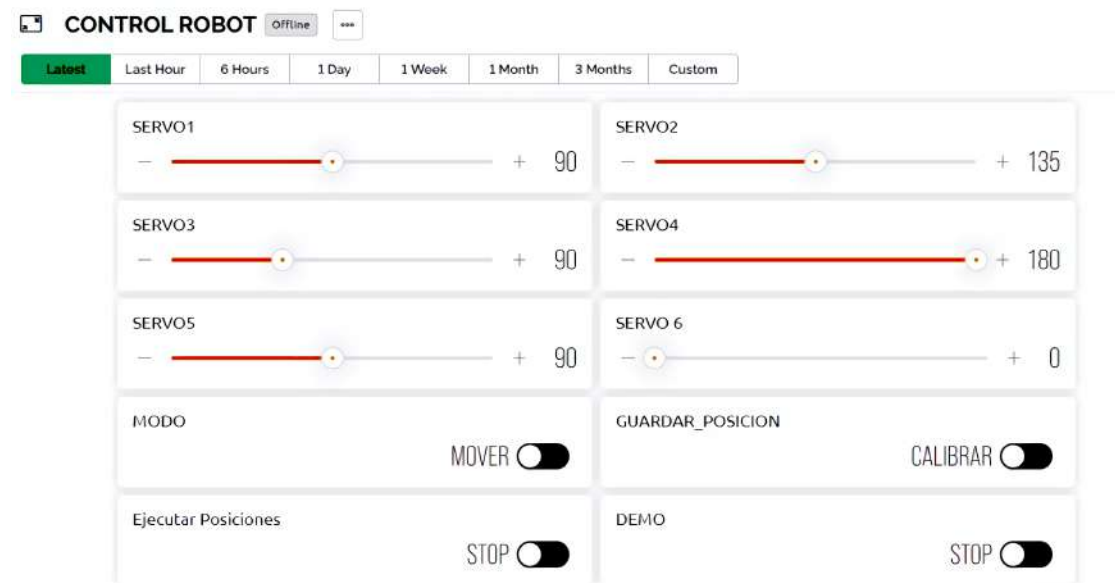


Figura 5.15: Diseño de aplicación en Blynk para el control de posición en servomotores. **Fuente:** Elaboración propia

Capítulo 6

Diseño de software

En este capítulo se indicará el paso a paso de la creación del software para dar funcionalidad al prototipo, iniciando con la implementación de los cálculos de la obtención de trayectorias en el sistema embebido, así mismo mostraremos los modelos CNN entrenados, el proceso de entrenamiento y sus resultados, finalizaremos integrando el reconocimiento de comandos de voz y creando una interfaz de unión para cada uno de estos métodos, que además sea capaz de guardar datos del procedimiento.

6.1. Integración del modelo cinemático al prototipo

Para la implementación de la cinemática simulada en el prototipo, se hizo uso del protocolo de comunicación TCP/IP, este cuenta con varias capas que permiten el flujo de información entre los sistemas embebidos y el software MATLAB.

En la primera capa llamada de aplicación se transforma el mensaje encapsulado en código binario, en la capa de transporte, se segmentan los datos a enviar según la cantidad de bytes, tanto en la capa de red como de enlace de datos se agregan encabezados de ruta tanto de origen como de destino para que el mensaje sea entregado de manera correcta, en la última capa o capa física es donde se transforma el mensaje en ondas de luz o eléctricas para ser enviadas por medio del enrutador.

El proceso anterior se repite de forma inversa en la recepción del mensaje por parte del servidor o el cliente, tal como se indica en la figura 6.1.

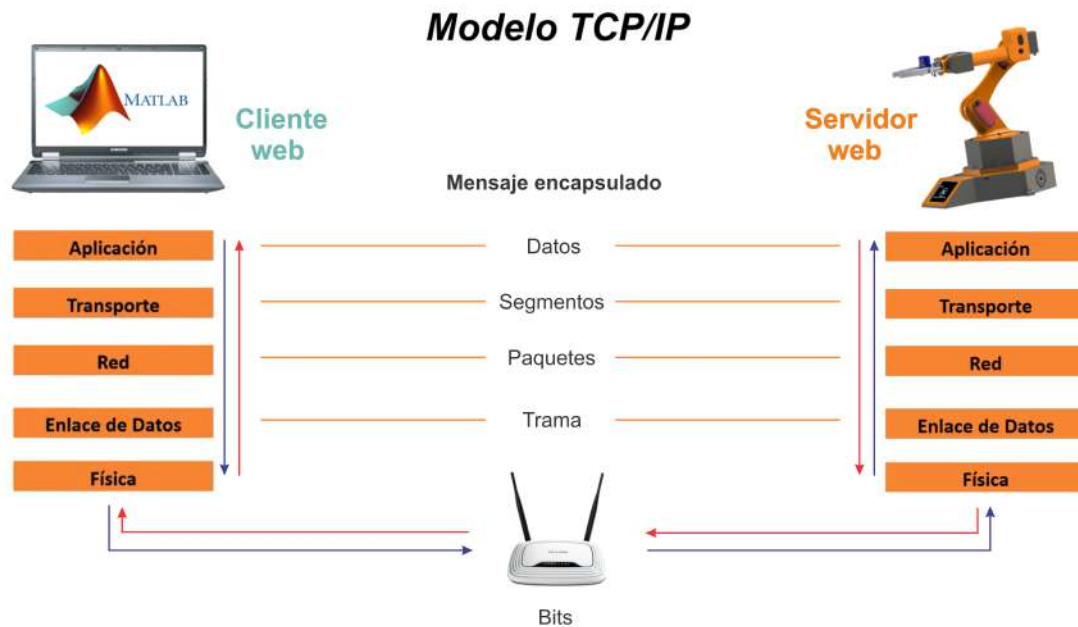


Figura 6.1: Protocolo TCP/IP entre prototipo físico y MATLAB. **Fuente:** Elaboración propia

El protocolo anteriormente mencionado se utiliza para enviar datos desde MATLAB tales como los puntos de referencia para el interpolador cúbico, que van a ser parte de la trayectoria, así mismo el ángulo de acercamiento y el número de puntos a interpolar entre cada tramo.

En el sistema embebido esp8266 se integró la cinemática inversa, el interpolador cúbico y el ajuste de ceros para que este cree la trayectoria final y se encargue de ejecutar el movimiento.

En un inicio se integró el ajuste de ceros en el software MATLAB y se envió desde este, la trayectoria hacia el prototipo físico, pero el modelo en el cual se genera la trayectoria en el sistema embebido tiene notables ventajas entre ellas:

- Mejora la velocidad en la comunicación, pues se disminuye notablemente los datos de envío.
- Mejora la precisión, ya que cuando se enviaba la trayectoria desde MATLAB se generaba pérdida de información.
- Se generan menos recursos computacionales y simplifica el código de programación en MATLAB.
- Obtiene una independencia el prototipo, del software, debido a que se puede usar otro método para enviar los puntos y que el sistema embebido genere las trayectorias.

Con el fin de ejemplificar las anteriores ventajas y agregar aún más funcionalidades al sistema que pueden ser de gran utilidad en cualquier campo de aplicación del prototipo, incluida la instrumentación quirúrgica, se implementa la conexión descrita en la figura 6.2 donde se muestra un flujo de información de MATLAB y Blynk hacia el prototipo.

Como se mencionó anteriormente desde MATLAB podemos enviar coordenadas tridimensionales para generar una trayectoria con el prototipo, y mediante Blynk podemos manipular cada grado de libertad uno a uno, guardar posiciones y hacer

que nuestro robot antropomórfico genere una trayectoria con ellas, de esta manera se visualiza de una forma muy amplia la última ventaja mencionada teniendo en cuenta que Blynk es un sistema el cual tiene su base de datos en la nube y nos permite manipular el robot sin tener en cuenta parámetros como la distancia existente entre ellos porque solo es necesario la conexión a internet, a gran diferencia del protocolo TCP/IP que conecta con MATLAB, pues requiere un enlace de los dos dispositivos a la misma red WLAN.

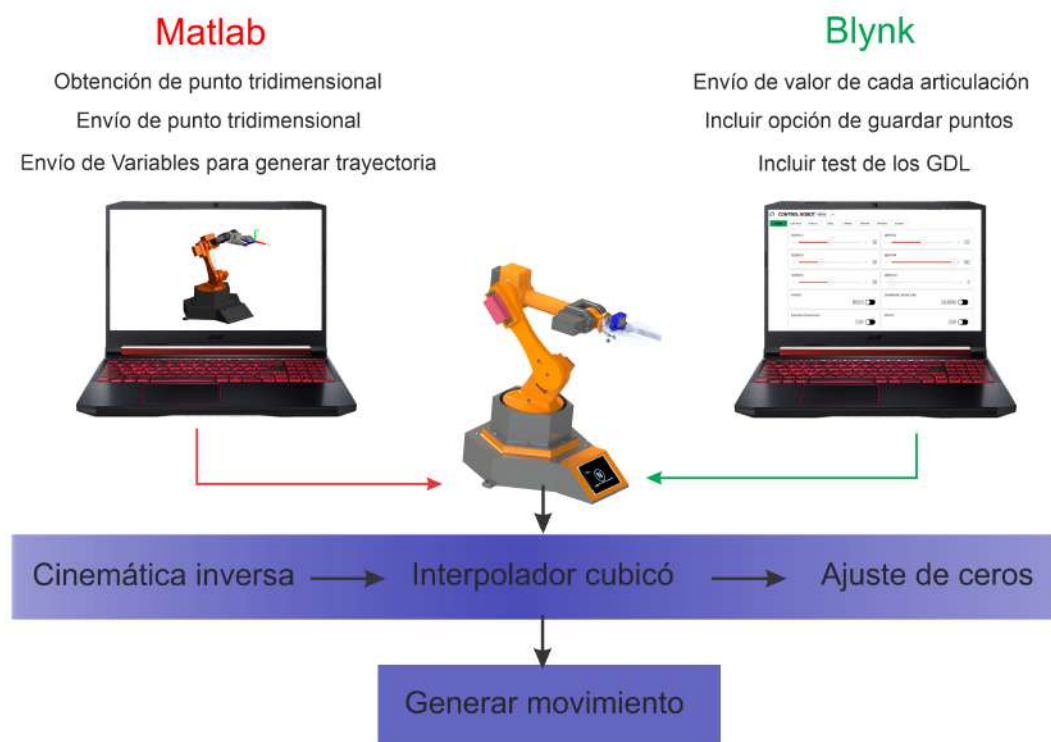


Figura 6.2: Flujo de información y modelo cinemático integrado en el sistema.
Fuente: Elaboración propia

Mediante la figura 6.3 se contempla el sistema robótico físico con el modelo cinemático ya implementado y con él, se ha obtenido la capacidad de manipulación eficaz de las herramientas quirúrgicas.

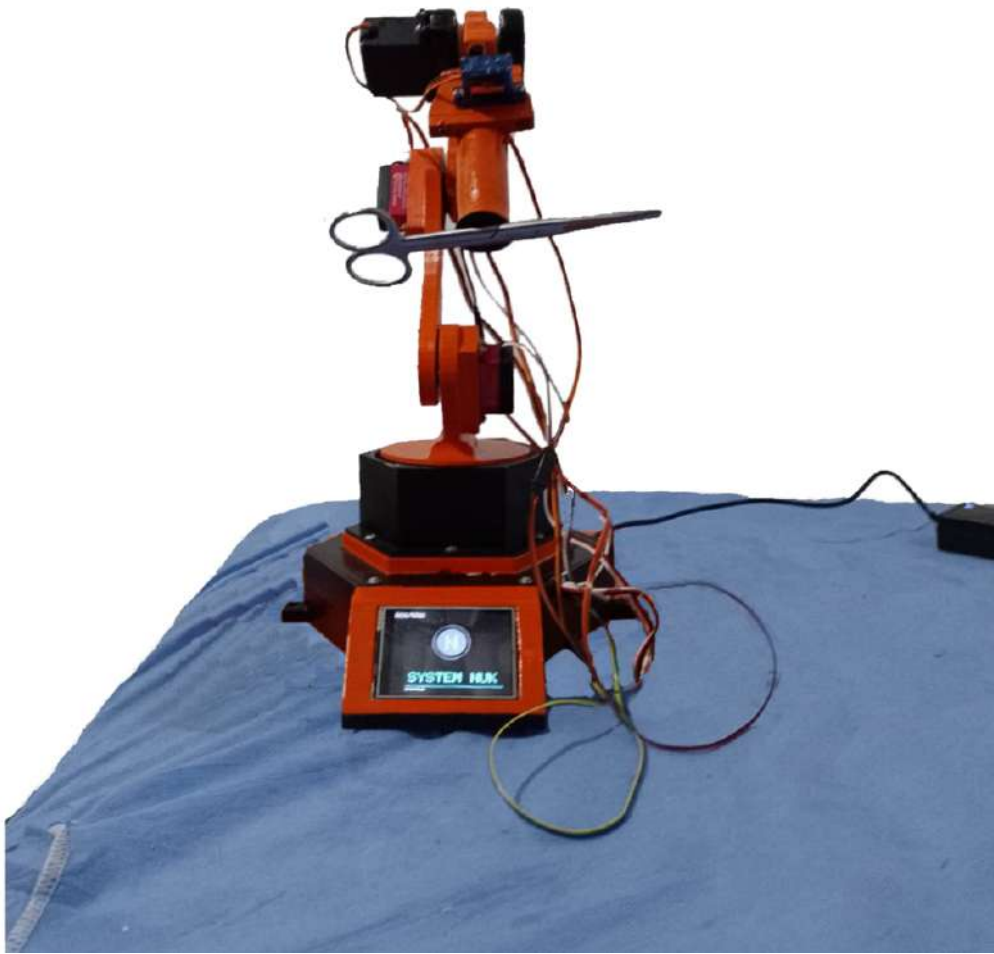


Figura 6.3: Sistema robótico físico con modelo cinemático implementado. **Fuente:** Elaboración propia

Con el fin de incorporar el campo de Deep Learning al sistema robótico, primero se debe definir de manera detallada el espacio de trabajo en el cual actuara, mediante SolidWorks y por medio de un editor de imágenes como lo es CorelDraw se obtuvo la imagen 6.4 donde se muestra el robot antropomórfico, dos soportes para las respectivas cámaras, los instrumentos quirúrgicos a manipular y una base en la cual se desarrolla todo el sistema.



Figura 6.4: Espacio de trabajo del sistema robótico. **Fuente:** Elaboración propia

La primera cámara ubicada en la parte inferior derecha del espacio de trabajo capturará los instrumentos disponibles, los ubicará dentro de la imagen y los clasificará según el nombre del instrumento, es desde este sitio de donde el robot facilitará el instrumental al personal médico cuando este lo solicite, en el segundo punto ubicado en la parte superior izquierda de la base del sistema se observa un

segundo soporte para cámara es aquí donde se entrega y se retira el instrumental, cuando el médico finalice o inicie el procedimiento con alguna herramienta, esta cámara tendrá la capacidad de detectar cuando se encuentre un instrumento en la imagen, clasificar el objeto y ubicarlo en la imagen para próximamente ser retirado y organizado.

6.2. Detectores de objetos y modelos CNN para la identificación de herramientas quirúrgicas

Existen varias estructuras de detectores de objetos, las cuales las podemos dividir en dos categorías, las de dos etapas y de una etapa.

Las de dos etapas son estructuras con una buena precisión, pero con una limitante referente a la velocidad de procesamiento necesaria para la predicción, lo que hace que no sean aptas para una detección en tiempo real, estas necesitan de dos algoritmos, el primero de proposición de cajas en las cuales es posible encontrar algún objeto y la segunda de clasificación por medio de redes neuronales convolucionales como forma de extracción de características.

Las estructuras de una etapa son estructuras con menos precisión que las de dos etapas, pero a diferencia de las primeras cuentan con una rapidez mucho mayor, generando la posibilidad de utilizarlas en tiempo real, en esta categoría se obtiene una proposición de cajas y la clasificación de ellas en un mismo algoritmo, al igual que las primeras necesita de redes neuronales convolucionales para la extracción de características.

Mediante la tabla 6.1 se categorizan algunas de las estructuras más utilizadas actualmente, para el prototipo en desarrollo se decide integrar una estructura de

una etapa, pues se requiere un tiempo de detección rápido con el que se realice el reconocimiento del objeto.

Tipos de detectores	
Una etapa	Dos etapas
YOLO	R-CNN
SSD	fast-RCNN
Retinanet	faster-RCNN
Efficientdet	mask-RCNN

Cuadro 6.1: Clases de detectores de objeto según su estructura. **Fuente:** Elaboración propia

La transferencia de aprendizaje suele ser en muchos casos una herramienta de gran utilidad con la cual se puede obtener un entrenamiento de mejor calidad y de una manera más sencilla, este procedimiento hace uso una red pre-entrenada para adaptarla a nuestro problema de reconocimiento, inicializando la red con los pesos ya entrenados y no aleatoriamente desde cero.

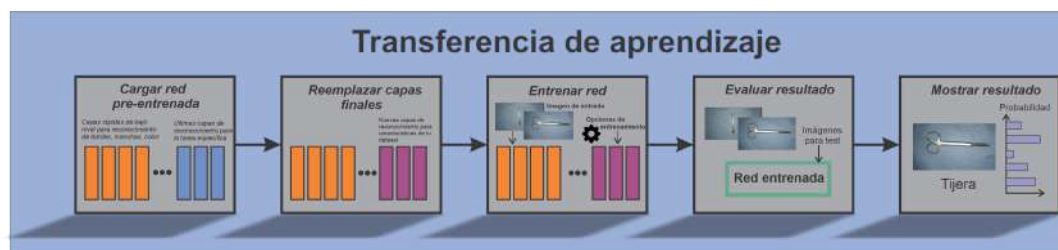


Figura 6.5: Algoritmo para la transferencia de aprendizaje. **Fuente:** Elaboración propia

La misma red convolucional ya entrenada sirve como mecanismo de extracción de las nuevas características, aunque hay que tener en cuenta que el tamaño de imagen de entrada a la red debe de adaptarse al del dataset con el que se encuentre entrenada, para que exista concordancia entre las capas de entrenamiento.

En el software MATLAB se encuentran varias redes neuronales pre-entrenadas disponibles, nosotros utilizaremos las redes neuronales Resnet50 junto con una estructura SSD, y Squeezenet en una estructura YOLO.

Para llevar a cabo el modelo de evaluación que define la calidad de la red entrenada se definen parámetros como la intersección sobre la unión (IOU) la cual refiere al porcentaje de superposición entre la caja limitadora real etiquetada por nosotros y la obtenida por la estructura de detección.

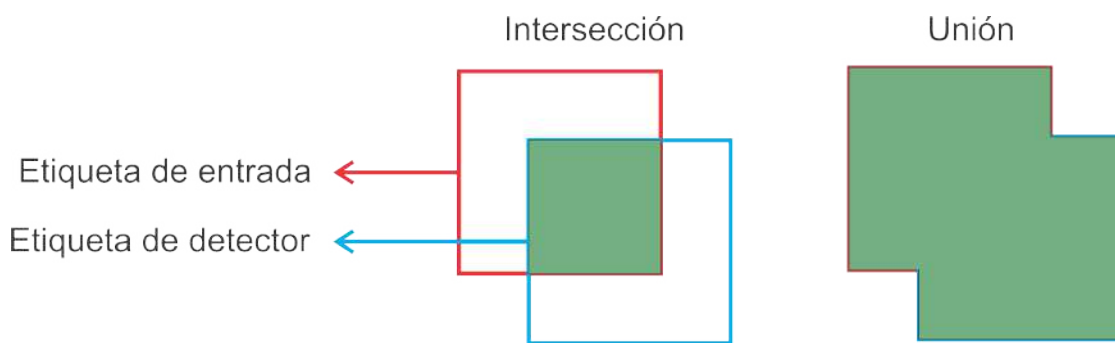


Figura 6.6: Intersección y unión en etiquetado de imágenes. **Fuente:** Elaboración propia

La expresión matemática de la IOU se indica de la siguiente forma:

$$IOU = \frac{I}{U} * 100$$

Mediante el cálculo de la intersección sobre la unión y un grado de confianza asignado por medio de código, es posible clasificar un acierto o desacierto en la predicción, por ejemplo, si establecemos un umbral del 50 %, todas las cajas delimitadoras que se tomaran como acertadas serán las que obtengan un valor IOU mayor a este, en caso contrario serán detecciones erróneas.

Otros parámetros a tener en cuenta son la precisión y el recuerdo, estas medidas nos dan una visión general de la funcionalidad del modelo entrenado. La

precisión es la fracción de aciertos con respecto a la totalidad de cajas delimitadoras generadas mediante la detección, y el recuerdo por su parte es la razón de los aciertos entre la cantidad de cajas delimitadoras etiquetadas por nosotros.

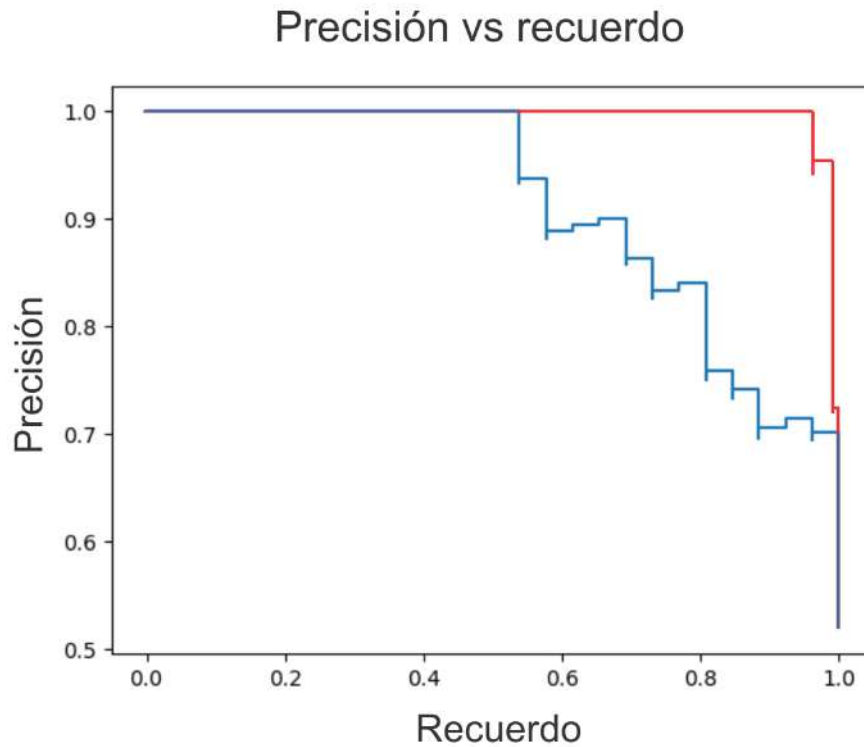


Figura 6.7: Gráfica de precisión con respecto al recuerdo **Fuente:** Elaboración propia

Los dos conceptos antes mencionados están estrechamente relacionados y tienden a actuar de forma inversa, es decir entre mayor sea el recuerdo menor precisión tiene el modelo, el objetivo de un buen entrenamiento es mantener una precisión alta durante el aumento del recuerdo, esto se logra observar mediante la figura 6.7, de la cual se puede concluir que los resultados del modelo expresados de color rojo son mejores que los de color azul.

6.2.1. Obtención de datasets

El dataset de entrada a una estructura de detección de objetos se realiza mediante la técnica de machine learning llamada aprendizaje supervisado, en la que cada dato debe contener una etiqueta.

Para obtener la cantidad de datos necesarios con los que se brinde la información que conlleve a una predicción fiable construimos una interfaz mediante código MATLAB, esta tiene como principal función facilitar la toma de imágenes y la organización de las mismas para próximamente ser etiquetadas.



Figura 6.8: Interfaz para la toma de imágenes. **Fuente:** Elaboración propia

Mediante la figura 6.8 se muestran las diferentes opciones integradas en el algoritmo de toma de datos, en la parte superior podemos elegir la cámara de trabajo, enseguida la ruta donde se alojarán las imágenes, podemos así mismo modificar parámetros como la velocidad entre cada foto tomada, el nombre con el que se guardarán, y el tamaño. Antes de iniciar con el proceso de capturas, podemos elegir entre una toma indefinida, es decir, se sigan obteniendo los datos

hasta pulsar el botón de pausa o se puede hacerlo por tramos, indicando el número de imagen inicial y final. En el transcurso se indicará tanto la cámara en vivo como la captura de la imagen inmediatamente anterior.



Figura 6.9: Muestra de dataset compuesto por 3000 imágenes. **Fuente:** Elaboración propia

Ejecutando el anterior sistema se creó dos datasets con el fin de definir la precisión obtenida con cada uno de ellos, en el primero se toma 3000 datos de 3 instrumentos en igual proporción y con un fondo estable, de los cuales se visualiza una muestra en la figura 6.9, este dataset aunque se tomó en un tamaño de 640x480 px se logra rescaldar mediante la función `imresize` de MATLAB, el segundo almacén de datos contiene 20000 imágenes, pero esta vez de 4 instrumentos en igual proporción y con un fondo variante como se observa en la figura 6.10.



Figura 6.10: Muestra de dataset compuesto por 20000 imágenes. **Fuente:** Elaboración propia

Ya que nuestro sistema también requiere de una cámara enfocando al lugar de donde el robot antropomórfico obtiene la herramienta quirúrgica y la escoge entre varias, se necesita un dataset en el que en una imagen se muestren diferentes objetos, esta toma de datos se realizó de forma similar a las anteriores, sin embargo, se ejecutó teniendo en cuenta diferentes combinaciones y apariciones de objetos en la imagen. La cantidad de fotos para este almacén de datos fue de 600 por cada una de 11 combinaciones.

Muestra de dataset 227x227 px con más de un objeto



Figura 6.11: Muestra de dataset con más de un instrumento enfocado. **Fuente:** Elaboración propia

Las combinaciones realizadas para la obtención de los datos de la figura 6.11 fueron:

- IB ■ IBT ■ IT ■ BT
- IBS ■ IS ■ BS ■ ST
- IBST ■ IST ■ BST

Siendo:

- I: Indicador quirúrgico
- B: Bisturí quirúrgico número 3

- S: Sonda acanalada
- T: Tijera recta

Con el fin de dar un etiquetado a las imágenes antes obtenidas hacemos uso de la herramienta imageLabeler de MATLAB con esto se puede clasificar cada uno de los instrumentos, para ejemplificar, se indica el caso del primer dataset en la figura 6.9 donde en la parte izquierda de la interfaz de la herramienta se tienen los tres labels correspondientes a la pinza de disección, la sonda acanalada y el mango del bisturí quirúrgico; con estas se señala el tipo y la ubicación del instrumento y es con estos datos que se entrenara las diferentes redes neuronales.

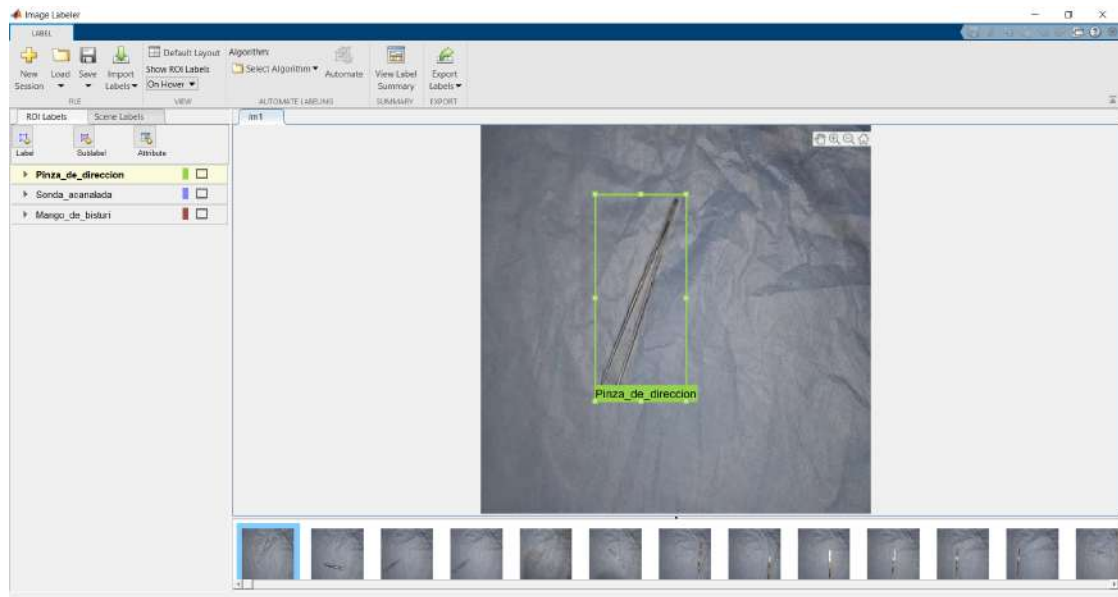


Figura 6.12: Etiquetado de imágenes mediante imageLabeler. **Fuente:** Elaboración propia

El modelo exportado de imageLabeler que se obtiene al finalizar el etiquetado se puede ver mediante la tabla 6.2, la primera columna corresponde a la ruta donde se encuentra las imágenes en cuestión, y en el caso del dataset de 3000 datos se

obtienen 3 columnas adicionales, una por cada herramienta quirúrgica, en ellas se indica la existencia de algún instrumento y su ubicación por medio de un vector de 4 posiciones, si hay ausencia del instrumento en la imagen se obtendrá un vector vacío.

Tabla de datos etiquetados			
Direccion	Pinza	Sonda	Bisturì
'D:\imagenes dataset\datadetecobject\pinzaim300x300\im1.png'	[80,51,94,161]	[]	[]
'D:\imagenes dataset\datadetecobject\pinzaim300x300\im2.png'	[69,199,140,55]	[]	[]
'D:\imagenes dataset\datadetecobject\pinzaim300x300\im3.png'	[14,173,112,99]	[]	[]
'D:\imagenes dataset\datadetecobject\pinzaim300x300\im4.png'	[10,173,113,97]	[]	[]
'D:\imagenes dataset\datadetecobject\pinzaim300x300\im5.png'	[55,102,115,136]	[]	[]
'D:\imagenes dataset\datadetecobject\pinzaim300x300\im6.png'	[103,111,114,125]	[]	[]
.	.	.	.
.	.	.	.
.	.	.	.
'D:\imagenes dataset\datadetecobject\pinzaim300x300\im1000.png'	[25,139,125,38]	[]	[]
'D:\imagenes dataset\datadetecobject\sondaim300x300\im1.png'	[]	[88,149,119,58]	[]
'D:\imagenes dataset\datadetecobject\sondaim300x300\im2.png'	[]	[73,5,76,184]	[]
'D:\imagenes dataset\datadetecobject\sondaim300x300\im3.png'	[]	[41,21,43,171]	[]
'D:\imagenes dataset\datadetecobject\sondaim300x300\im4.png'	[]	[91,111,80,148]	[]
'D:\imagenes dataset\datadetecobject\sondaim300x300\im5.png'	[]	[1,103,90,41]	[]
'D:\imagenes dataset\datadetecobject\sondaim300x300\im6.png'	[]	[64,173,62,127]	[]
.	.	.	.
.	.	.	.
.	.	.	.
'D:\imagenes dataset\datadetecobject\sondaim300x300\im1000.png'	[]	[22,59,96,138]	[]
'D:\imagenes dataset\datadetecobject\bisturiim300x300\im1.png'	[]	[]	[115,118,91,93]
'D:\imagenes dataset\datadetecobject\bisturiim300x300\im2.png'	[]	[]	[63,56,61,136]
'D:\imagenes dataset\datadetecobject\bisturiim300x300\im3.png'	[]	[]	[65,154,85,112]
'D:\imagenes dataset\datadetecobject\bisturiim300x300\im4.png'	[]	[]	[98,111,26,148]
'D:\imagenes dataset\datadetecobject\bisturiim300x300\im5.png'	[]	[]	[149,165,55,136]
'D:\imagenes dataset\datadetecobject\bisturiim300x300\im6.png'	[]	[]	[160,62,46,150]
.	.	.	.
.	.	.	.
.	.	.	.
'D:\imagenes dataset\datadetecobject\bisturiim300x300\im1000.png'	[]	[]	[23,184,110,48]

Cuadro 6.2: Modelo de datos exportado de imageLabeler. **Fuente:** Elaboración propia

6.2.2. Entrenamiento y validación con SSD

La estructura SSD o de disparo único indicada en la figura 6.13 se compone de cinco partes importantes desde la entrada de la imagen hasta la predicción, próximo al ingreso de la imagen a etiquetar se encuentran dos partes iniciales ne-

tamente convolucionales, la primera de ellas encargada de generar un mapa de características mediante una red neuronal convolucional en este caso se hará uso de una resnet50 pre-entrenada, la segunda obtiene las posibles cajas delimitadoras y etiquetas correspondientes agregándoles una probabilidad o confianza a cada una, es importante resaltar que en esta sección de convolución se agregan filtros y por ende la obtención de mapeos de características más pequeños. Todos estos son analizados dando la oportunidad de realizar una predicción a múltiples escalas. Habiendo obtenido ya las cajas delimitadoras con sus respectivas etiquetas y valores de confianza, se hace una exclusión de ellas, dejando solo a las de una probabilidad alta y de esta manera tendremos como salida una predicción más exacta.

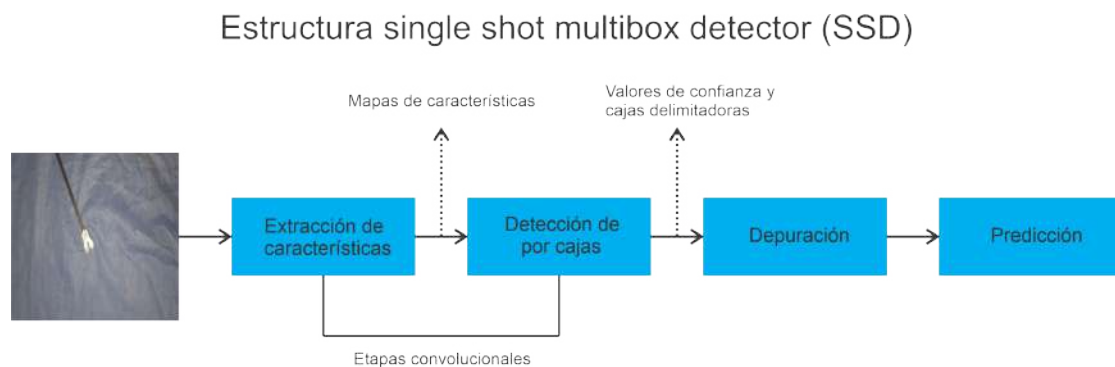


Figura 6.13: Estructura de detección single shot multibox detector. **Fuente:** Elaboración propia

En cuanto a la red pre-entrenada resnet50 utilizada como forma de extracción de características tiene una arquitectura que como su nombre lo indica es de 50 capas observadas en la tabla 6.3, para lograr una transferencia de aprendizaje con este tipo de redes se congelan las primeras capas las cuales ya tienen una aplicación de filtros efectivos en la detección de ciertas formas, colores, bordes . . . , y son remplazadas las últimas capas ajustando la salida a la cantidad de clases que se tiene en el problema en particular en este caso se tiene dos datasets uno de

3 clases y otro de 4.

Estructura de Resnet50				
Nombre de capa	Tamaño de salida	Capas		
Conv1	112X112	7X7, 64, <i>stride</i> 2		
Con2 x	56X56	3X3 <i>Maxpool</i> , <i>stride</i> 2		
	56X56		1X1, 64 3X3, 64 1X1, 256	<i>x</i> 3
Con3 x	28X28		1X1, 128 3X3, 128 1X1, 512	<i>x</i> 4
Con4 x	14X14		1X1, 256 3X3, 256 1X1, 1024	<i>x</i> 6
Con5 x	7X7		1X1, 512 3X3, 512 1X1, 2048	<i>x</i> 3
<i>Average pool</i> , 1000 <i>d fc</i> , <i>softmax</i>				

Cuadro 6.3: Arquitectura de Resnet50. **Fuente:** Elaboración propia

Durante el entrenamiento de las redes neuronales convolucionales se logran generar diferentes gráficas con el fin de detectar el correcto transcurso en el aprendizaje, entre estas está la pérdida o descenso del error y la precisión, la primera indica básicamente como va disminuyendo los desaciertos de la red entre cada iteración o época, esto lo logra calculando los gradientes utilizados para actualizar los pesos de la red mediante la función de descenso que cotidianamente ejecuta un proceso de regresión de datos usando el método estadístico del error cuadrático medio.

En cuanto la gráfica de precisión muestra el rendimiento obtenido de la red a través de las diferentes iteraciones, aunque se asocia muchas veces con el valor de exactitud o desempeño final, este cambia en el momento de hacer un test con datos diferentes a los de entrada.

Entrenamiento con un almacén de 3000 datos y 3 clases.

Mediante la figura 6.14 se muestra el descenso del error y la precisión para cada iteración obtenida durante el transcurso del aprendizaje, empleando una estructura SSD que tiene como entrada un dataset de 3000 imágenes de tres clases de instrumentos.

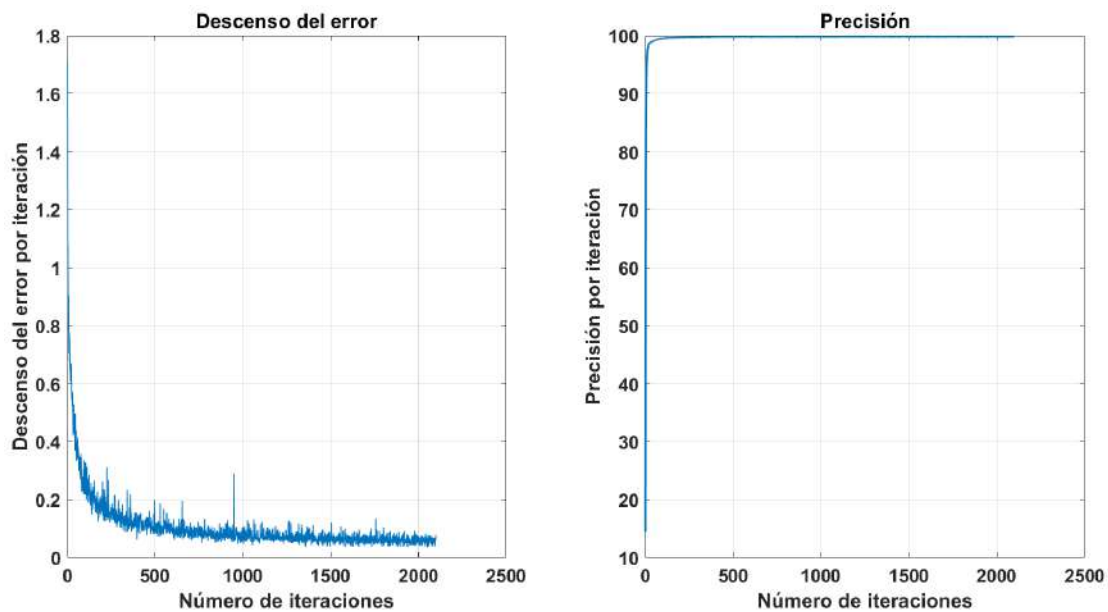


Figura 6.14: Descenso del error y precisión de SSD con 3000 imágenes. **Fuente:** Elaboración propia

Se logra apreciar que la señal de descenso del error no es estable, aunque presenta una pérdida significativa en las primeras 500 iteraciones, además no logra fijarse en un valor final muy cercano al objetivo, es decir, al cero; lo que nos indica una dudosa funcionalidad de este modelo para esta aplicación específica y con este tamaño de entrada de datos.

Con el fin de evaluar este detector se hace uso de los conceptos definidos al inicio del presente capítulo, referidos a la precisión y el recuerdo en un conjunto de

datos nuevo para el detector, este proceso se hizo con una cantidad de imágenes correspondientes al 30 % con respecto al dataset de entrenamiento. Los resultados obtenidos se muestran en la imagen 6.15.

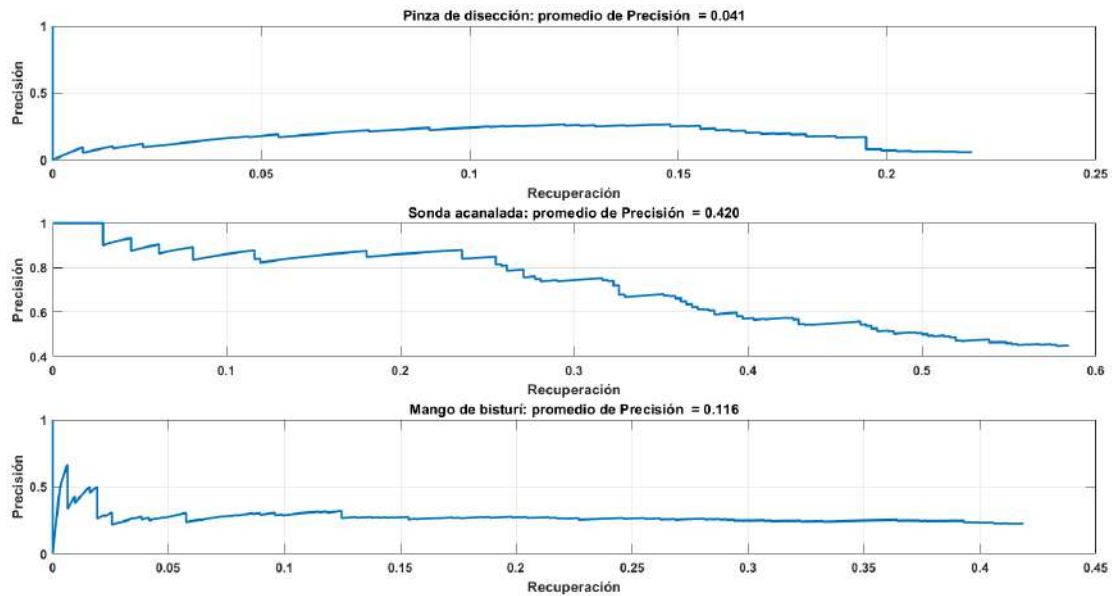


Figura 6.15: Precisión con respecto al recuerdo en SSD con 3000 imágenes. **Fuente:** Elaboración propia

De la figura 6.15 logramos concluir que la estructura SSD con 3000 datos de entrenamiento correspondientes a 3 clases y para esta aplicación en específico no se podría implementar en una puesta en marcha real, pues su desacierto es alto, ya que no logra mantener una buena precisión a medida que asciende el valor del recuerdo.

Se hizo pruebas visualizando el comportamiento de esta red tanto con los datos de entrenamiento como para los de validación como lo muestra la figura 6.16 y se logra observar que el detector indica el lugar correcto donde se encuentra un objeto, pero no siempre lo clasifica de manera adecuada.

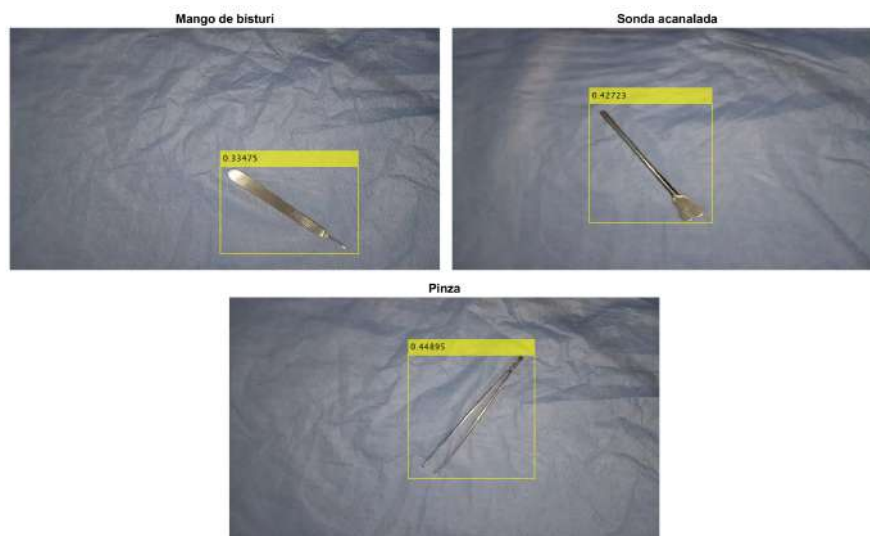


Figura 6.16: Funcionamiento de detector SSD con 3000 imágenes. **Fuente:** Elaboración propia

Entrenamiento con un almacén de 20000 datos y 4 clases.

Con el fin de realizar diferentes pruebas antes de descartar esta estructura para la aplicación en cuestión, se realizó un segundo entrenamiento con un dataset más amplio, agregando a la vez una cuarta clase a los datos.

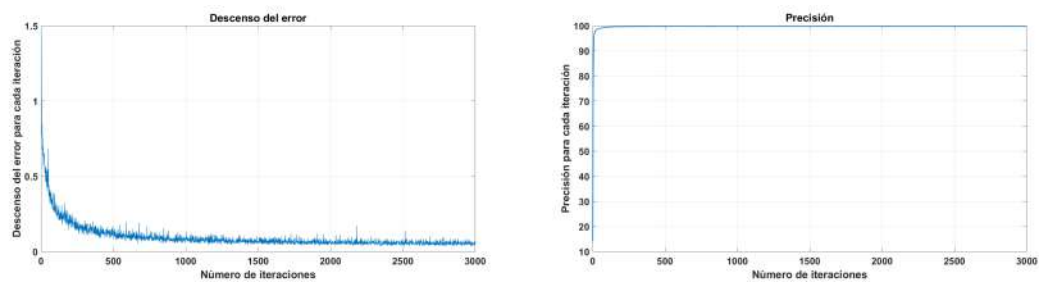


Figura 6.17: Descenso del error y precisión de SSD con 20000 imágenes. **Fuente:** Elaboración propia

Mediante la figura 6.17 se observa que el proceso finalizó con 3000 iteraciones,

al igual que en el entrenamiento anterior se encuentra una caída significativa del error en las primeras 500 iteraciones; al terminar el proceso el desacierto es más bajo que con el dataset de 3000 datos, pero su señal sigue siendo muy inestable.

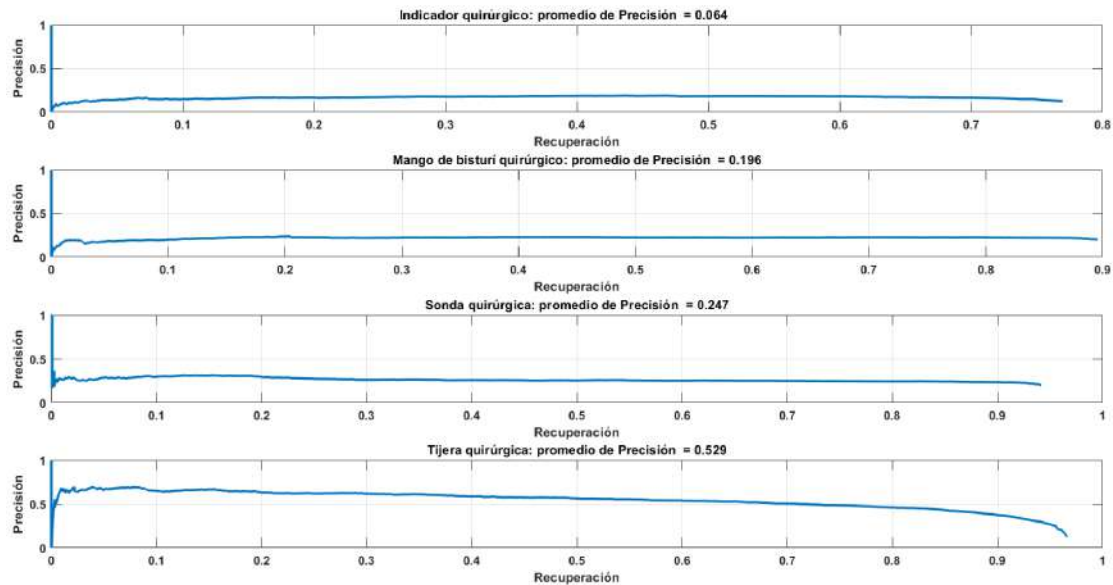


Figura 6.18: Precisión con respecto al recuerdo en SSD con 20000 imágenes. **Fuente:** Elaboración propia

El resultado del test realizado para la evaluación de este entrenamiento se muestra mediante la figura 6.18 en ella encuentran los promedios de precisión para cada clase con respecto a la recuperación. De estos valores se puede concluir que, aunque mejoran los resultados en comparación con el aprendizaje de la red anterior sigue siendo demasiado imprecisa para una tarea que requiere mucha más exactitud.

Realizando las pruebas de visualización nos damos cuenta que el haber agregado fondos variantes a las fotos tomadas genera un avance positivo en el reconocimiento acertado, al igual que el aumento importante en la cantidad de datos.

La detección generada por esta estructura se indica mediante la figura 6.19,

de ella podemos decir que al igual que en la anterior red detecta correctamente la posición de los objetos, pero no identifica de la mejor manera la clase de ellos, se puede inferir que esta estructura, en esta aplicación podría ayudar e implementarse si solo se requiriera la identificación de una clase.

Entre sus puntos negativos esta además el tiempo de entrenamiento, el cual con una base de datos de 3000 imágenes fue de una hora y quince minutos, con un dataset de 20000 imágenes fue de diez horas y treinta minutos. Se debe tener en cuenta que el hardware de entrenamiento cuenta con un procesador Intel core i5, memoria RAM de 8GB y una tarjeta gráfica NVIDIA GTX 1050

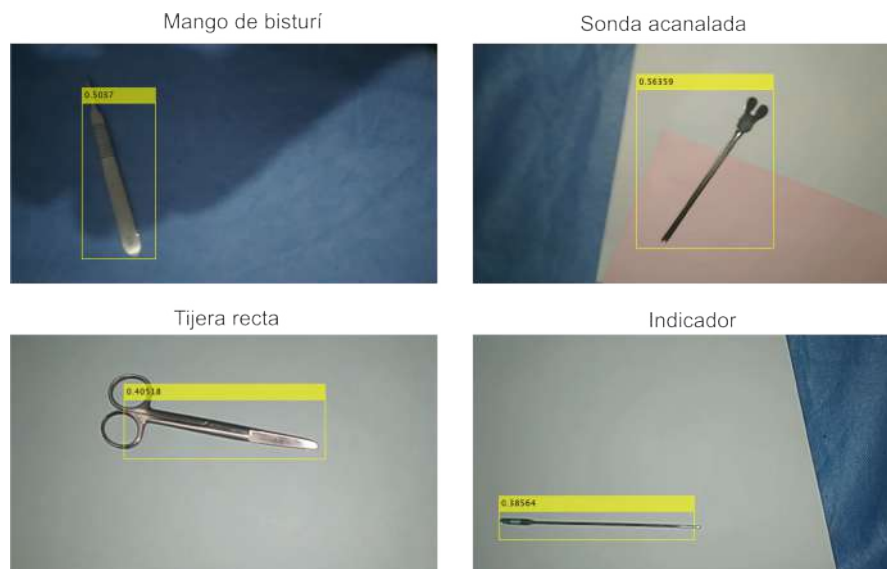


Figura 6.19: Funcionamiento de detector SSD con 20000 imágenes y 4 clases. **Fuente:** Elaboración propia

6.2.3. Entrenamiento y validación con YOLO v3

YOLO v3 es una estructura de detección de objetos que tiene como objetivo su implementación en tiempo real, con un grado de asertividad alto; esto lo logra

mediante el proceso mostrado en la figura 6.20 donde para el entrenamiento ingresa la imagen ya etiquetada y YOLO hace una partición de ella en forma de cuadrícula, y a cada uno de estos segmentos le asigna un vector de parámetros los cuales son:

- P_c = Grado de confianza
- C_n = Clases
- (x, y) = Coordenadas del centro del objeto respecto a la celda contenedora
- W = Alto de la caja delimitadora
- H = Ancho de la caja delimitadora

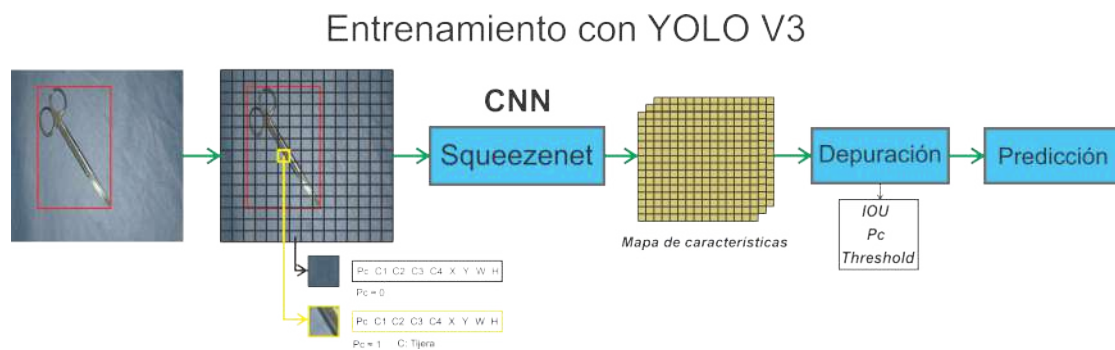


Figura 6.20: Estructura de detección you only look once. **Fuente:** Elaboración propia

El grado de confianza de cada partición es un porcentaje con el que se indica la probabilidad de existencia de un objeto en este segmento y con las variables de clase se muestra a cuál pertenece, la celda que es etiquetada por el algoritmo después de la partición es la que se encuentra en medio de la caja delimitadora, siendo está a la vez donde se encuentra el centro del elemento a detectar, es por ello que tendrá una confianza muy cercana al 100 %. Las coordenadas (x, y) por su

parte refieren a la posición exacta del centro del instrumento con respecto a la celda etiquetada, hay que tener en cuenta que (x, y) son medidas referenciadas acorde a la celda, mientras que W y H siendo el alto y ancho del objeto son referidas al tamaño de la imagen de entrada.

Una vez Incorporado el vector correspondiente a todas las celdas, los datos entran a la parte convolucional donde por medio de la red Squeezenet se obtienen los mapas de características asociados a la imagen, la idea principal de YOLO es que a cada partición de entrada le corresponde un segmento en el mapa de características, y de ellos se extrae los valores de salida.

Obtenido los valores de salida se depuran mediante diferentes métodos, ya que de no ser así obtendríamos muchas cajas delimitadoras propuestas, en diversos casos porque existen varios segmentos en los cuales hay presencia del objeto o por celdas muy cercanas al centro.

La primera variable que se utiliza para elegir las cajas delimitadoras con más alta probabilidad es el grado de confianza, con el que se establece un umbral en el que se admiten solo las que tengan un valor superior a este, además se utiliza el concepto de IOU explicado en el inicio del presente capítulo, con este se suprimen las cajas que se encuentren muy cercanas o sobrepuestas, ya que el valor de la intersección sobre la unión en estos casos es muy cercano a la unidad.

En seguida del proceso de depuración ya se hace posible realizar una predicción confiable de las cajas delimitadoras y etiquetas correspondientes a los objetos detectados.

Para la parte convolucional de YOLO v3 se hace uso del modelo de transferencia de aprendizaje usando Squeezenet, esta es una CNN pre-entrenada ofrecida en el paquete de MATLAB, su arquitectura está compuesta por una imagen de entrada

de tamaño 227x227x3 px y de esta se ejecutan una serie de convoluciones mostradas en la tabla 6.4 generando un mapa de características de 14x14x512 px.

Estructura de Squeezenet		
Nombre de capa	Tamaño de salida	Capas
Conv1	113X113X64	3X3, 64, <i>stride</i> 2
Max pool	56X56X64	3X3 <i>stride</i> 2
Fire2	56X56x128	<i>Squeeze, expand</i>
Fire3	56X56X128	<i>Squeeze, expand</i>
Max pool	28X28X128	3X3 <i>stride</i> 2
Fire4	28X28X256	<i>Squeeze, expand</i>
Fire5	28X28X256	<i>Squeeze, expand</i>
Max pool	14X14X256	3X3 <i>stride</i> 2
Fire6	14X14X384	<i>Squeeze, expand</i>
Fire7	14X14X384	<i>Squeeze, expand</i>
Fire8	14X14X512	<i>Squeeze, expand</i>
Fire9	14X14X512	<i>Squeeze, expand</i>
<i>Dropout, conv10 : 14X14X1000, Average pool, softmax</i>		

Cuadro 6.4: Arquitectura de Squeezenet. **Fuente:** Elaboración propia

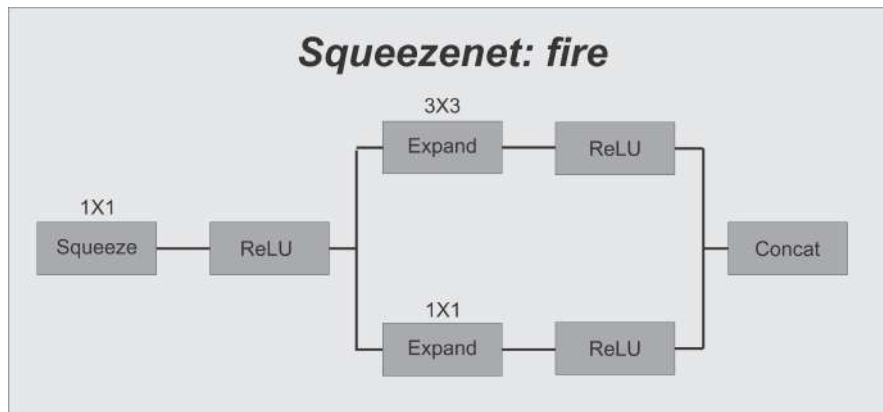


Figura 6.21: Estructura de un fire en Squeezenet. **Fuente:** Elaboración propia

La estructura Squeezenet incorpora los llamados bloques fire, compuestos por dos capas de convolución indicadas en la figura 6.21, la primera se denomina squee-

ze y en ella se aplican filtros 1x1 preparando la entrada a la capa expand, donde se emplean kernels tanto 3x3 y 1x1, disminuyendo los datos de procesamiento y haciéndola ideal para el uso en hardware no especializado.

Tomando como punto de partida lo observado mediante el entrenamiento de la red SSD, entrenamos la estructura YOLO con la base de datos de 20000 imágenes, pues queda evidenciado que entre más grande es el dataset aumentara la precisión de la red.

A continuación, comprobaremos la superioridad en la eficacia y asertividad de YOLO con respecto a la estructura SSD para esta aplicación, es por ello que se descarta la implementación de esta última.

Para una integración de YOLO a nuestro sistema vamos a dividir dos tipos de entrenamiento, uno en el que se enfoque en cámara un solo instrumento, esta red sería especializada para el reconocimiento en el espacio donde el robot antropomórfico deposita y recibe la herramienta que está en uso. El segundo tipo de entrenamiento se ejecutará con el dataset en el cual se enfocan más de un instrumento; este detector será utilizado en el lugar en el que el robot adquiere la herramienta quirúrgica según la orden dada por medio de comandos de voz.

Clasificación y detección de un solo instrumento enfocado en cámara

El entrenamiento de esta red se hizo mediante un dataset de 20000 imágenes en las cuales se utilizan diferentes fondos y posiciones de la herramienta quirúrgica, al terminar el entrenamiento se generó una gráfica de visualización en la cual podemos observar la tasa de aprendizaje con respecto a las iteraciones así mismo el descenso del error del cual podemos concluir que tuvo un decrecimiento significativo hasta un número de iteraciones aproximadas a las 200, a partir de este momento empieza

un descenso lento, pero cada vez más aproximándose al valor objetivo que sería el cero de la función, figura 6.22.

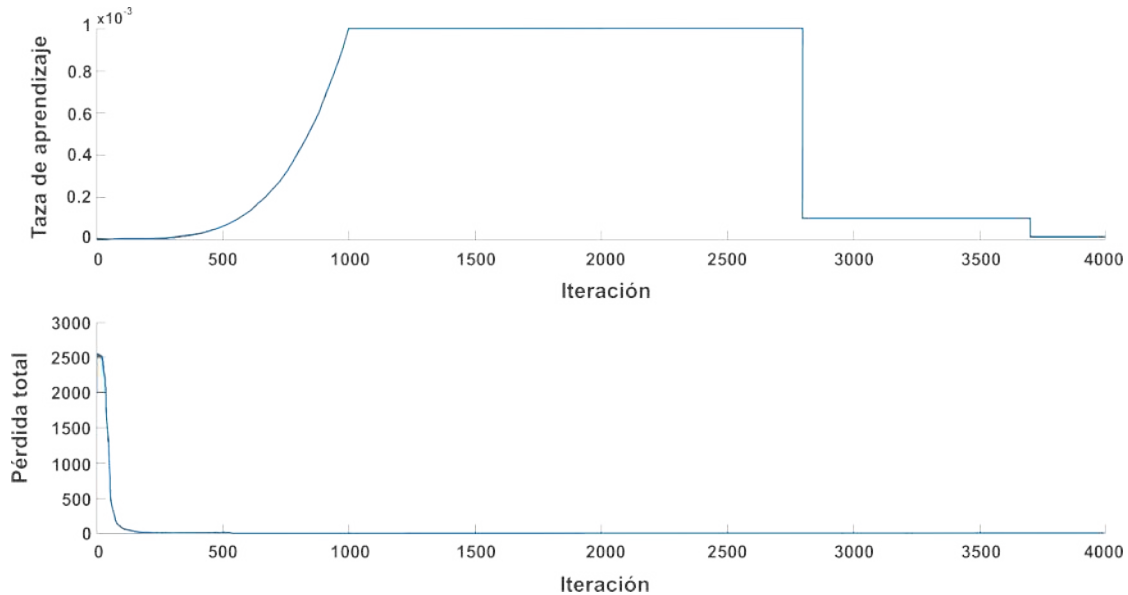


Figura 6.22: Descenso del error y tasa de aprendizaje de YOLO V3 enfocando un objeto en la imagen **Fuente:** Elaboración propia

La asertividad promedió obtenida en el test del detector por cada clase de instrumento la podemos obtener mediante la gráfica de la precisión con respecto al recuerdo mostrada en la figura 6.23.

De estos resultados podemos hacer conclusiones como que el valor promedio de precisión más bajo fue obtenido por el indicador, lo cual era algo esperado por su forma, al contrario de la tijera recta que obtuvo la precisión más elevada; aunque ninguna de estas estimaciones está por debajo del 96

También se establece que la precisión se mantiene a medida del aumento del recuerdo, lo cual nos da constancia de una buena funcionalidad en su implementación.

Si se compara los resultados de la red entrenada con la estructura SSD y el mismo almacén de datos, nos damos cuenta de que YOLO V3 es mucho más asertivo en las 4 clases de herramientas quirúrgicas.

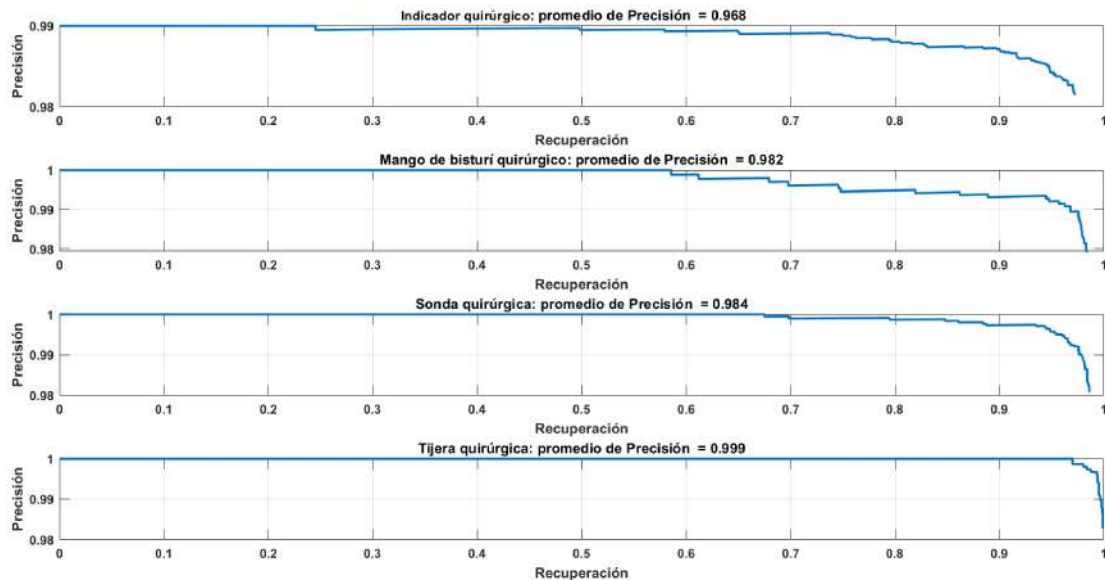


Figura 6.23: Precisión con respecto al recuerdo en YOLO V3 enfocando un solo objeto. **Fuente:** Elaboración propia

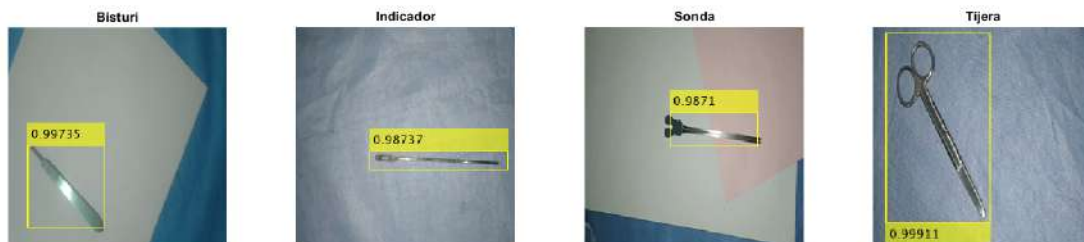


Figura 6.24: Funcionamiento de detector YOLO V3 con 20000 imágenes y 4 clases. **Fuente:** Elaboración propia

La figura 6.24 muestra de forma clara la detección generada por esta red entrenada, además se puede apreciar los altos porcentajes de certeza para cada instrumento y se da muestra de la eficacia para una integración en tiempo real.

Clasificación y detección de más de un instrumento enfocado en cámara

Para entrenar este detector se le agregó al dataset de 20000 imágenes 6600 datos más, lo que equivale a 600 fotos por cada combinación nombrada en el presente capítulo en el apartado de obtención de datasets.

Al igual que en el entrenamiento anterior se estableció un límite de iteraciones igual a 4000, se observa que en la gráfica de descenso del error de la figura 6.25, se tiene un comportamiento similar al entrenamiento de un solo instrumento enfocado en cámara, pues tiene un descenso significativo hasta un valor aproximado de 200 iteraciones, después de ello se realiza un descenso lento acercándose al valor objetivo es decir a cero.

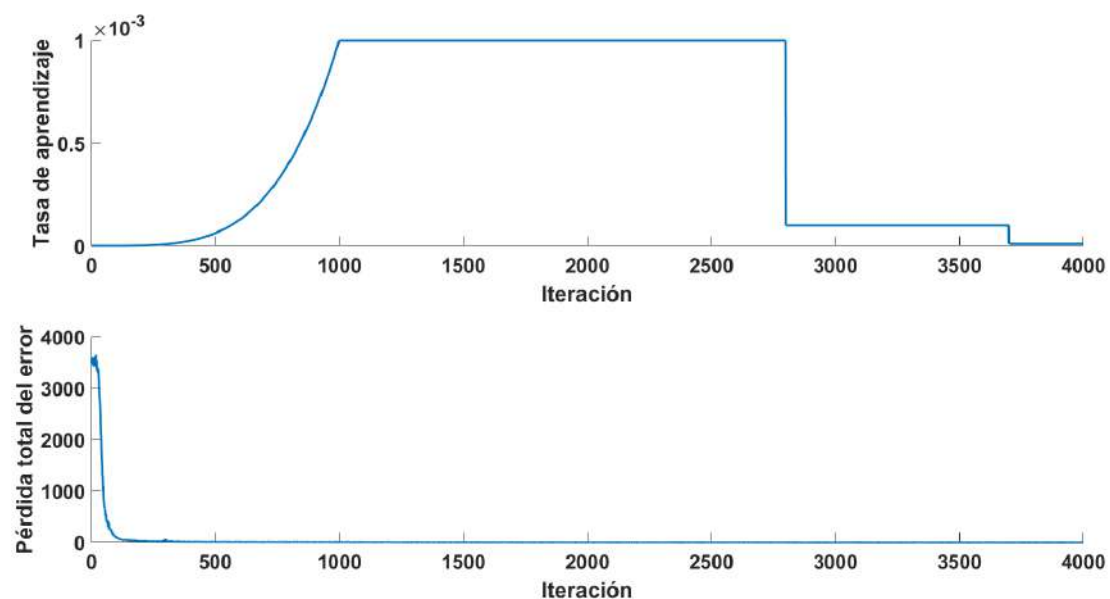


Figura 6.25: Descenso del error y tasa de aprendizaje de YOLO V3 enfocado más de un objeto en la imagen **Fuente:** Elaboración propia

De la gráfica de la precisión con respecto al recuerdo se puede decir que se obtiene una leve mejora en el comportamiento comparado con la red de una sola

herramienta quirúrgica enfocada en cámara. Se mantiene la precisión a medida que sube el valor de recuerdo y la confianza menor obtenida es de 96 % atribuida nuevamente por el indicador quirúrgico.

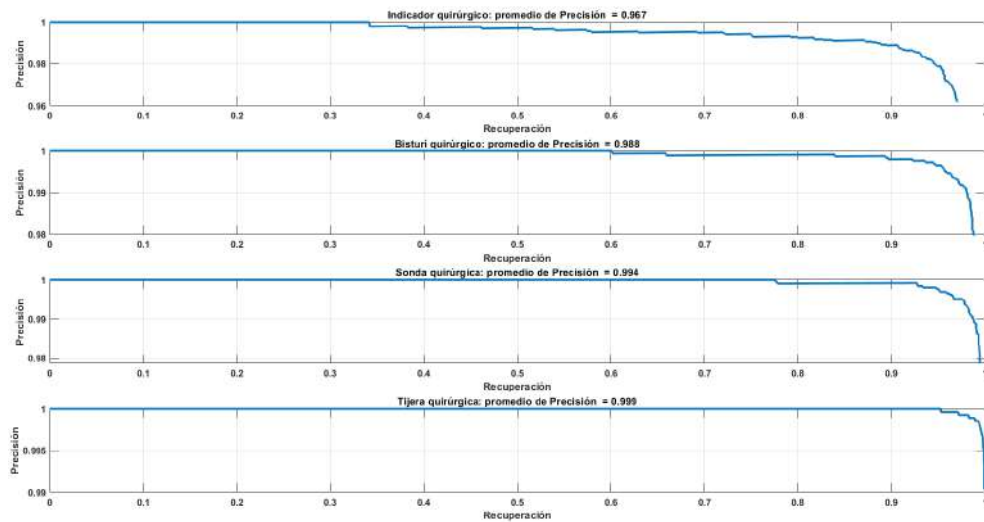


Figura 6.26: Precisión con respecto al recuerdo en YOLO V3 enfocando más de un objeto en la imagen. **Fuente:** Elaboración propia

En la figura 6.27 se indica el funcionamiento de este detector donde se evidencia el correcto comportamiento y la identificación de diferentes instrumentos en una misma imagen, etiquetándolos de una forma acertada.



Figura 6.27: Funcionamiento de detector YOLO V3 enfocando más de un objeto en una imagen. **Fuente:** Elaboración propia

6.2.4. Punto medio de la herramienta y ángulo de orientación

La posición y orientación de la herramienta quirúrgica en la imagen es importante para la correcta manipulación por parte del robot antropomórfico mediante la figura 6.28 en el apartado a se observa la disposición ya planteada del área de trabajo del sistema, la ubicación de las dos cámaras que enfocarán el lugar donde estarán ubicados inicialmente todos los instrumentos, y donde se depositará el requerido mediante comandos de voz.

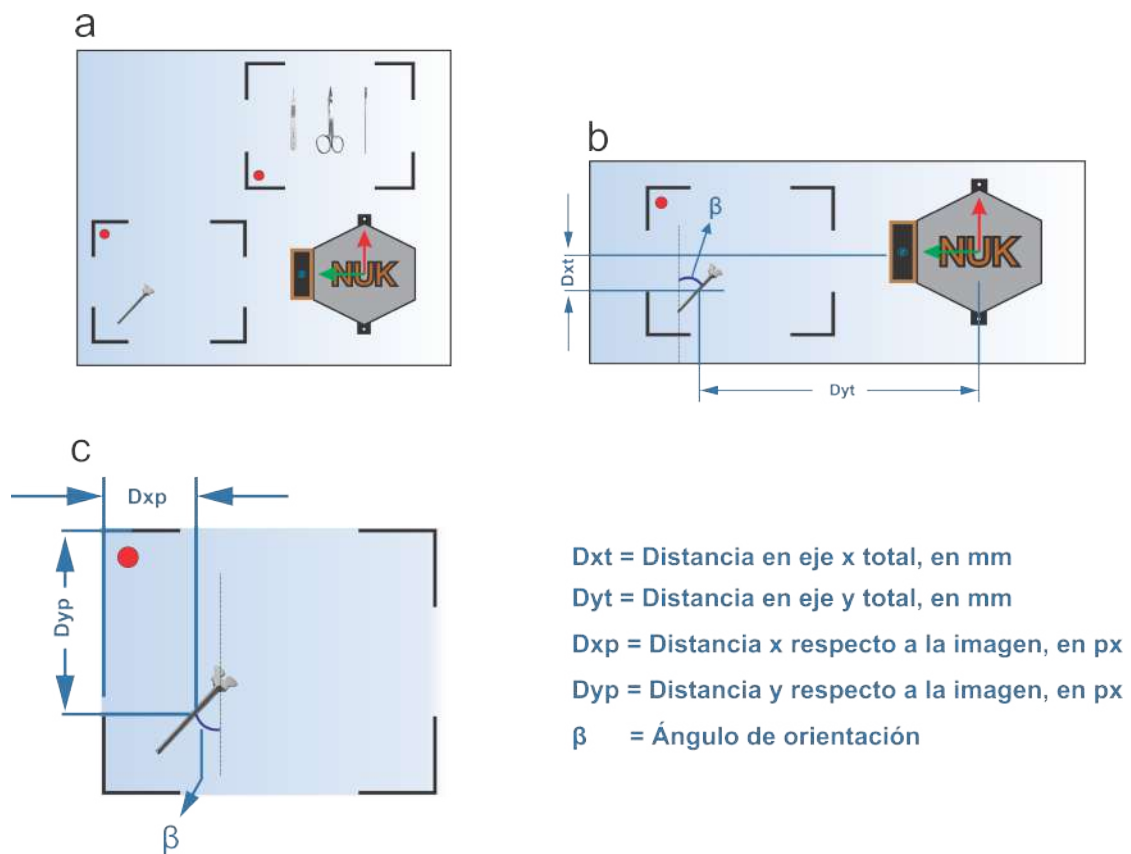


Figura 6.28: Posición del instrumento quirúrgico respecto al robot en el área de trabajo. **Fuente:** Elaboración propia

En el apartado b se analiza el caso de la cámara en la cual enfoca un solo instrumento, adicionalmente se acotan los valores Dyt, Dxt Y β siendo en su orden la distancia que se encuentra la herramienta en el eje y, en el eje x y el ángulo existente con este último, respecto al sistema coordenado cero del robot; estos son los valores que necesita la cinemática para ubicar el objeto, sujetarlo y orientarlo.

En el apartado c se especifica como se mide las distancias en la imagen, estando estas en unidades de píxeles, es por esto que se debe encontrar una relación con las medidas en mm, el ángulo β por su parte si es posible calcularlo partiendo de estas unidades.

Funcionamiento de redes neuronales para obtener orientación de los objetos

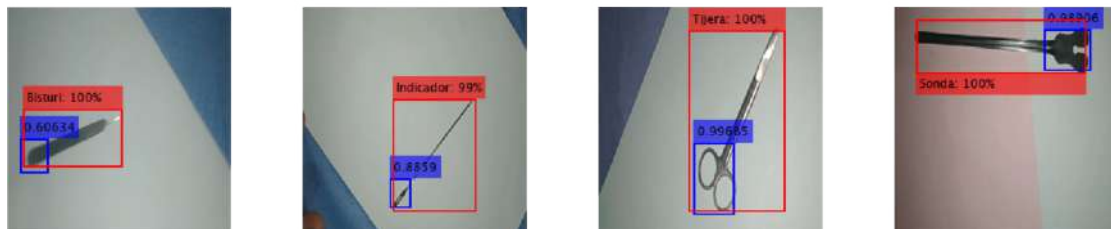


Figura 6.29: Funcionamiento en la detección del objeto y su extremo. **Fuente:** Elaboración propia

Para lograr medir el ángulo β correctamente se entrena una red neuronal que además de identificar la herramienta quirúrgica también sea capaz de ubicar su extremo, este funcionamiento se observa en la figura 6.30 donde se obtienen las cajas delimitadoras necesarias para el cálculo.

El resultado arrojado por las redes neuronales en cada detección se almacena en tres variables, la primera corresponde a la etiqueta, es decir, el tipo de instrumento al que pertenece, como puede ser tijera, sonda, bisturí e indicador o sus extremos; La segunda almacena el vector con los valores de la caja delimitadora y la tercera la precisión con la cual hace la predicción.

Variables para la obtención de la orientación y el punto medio del objeto

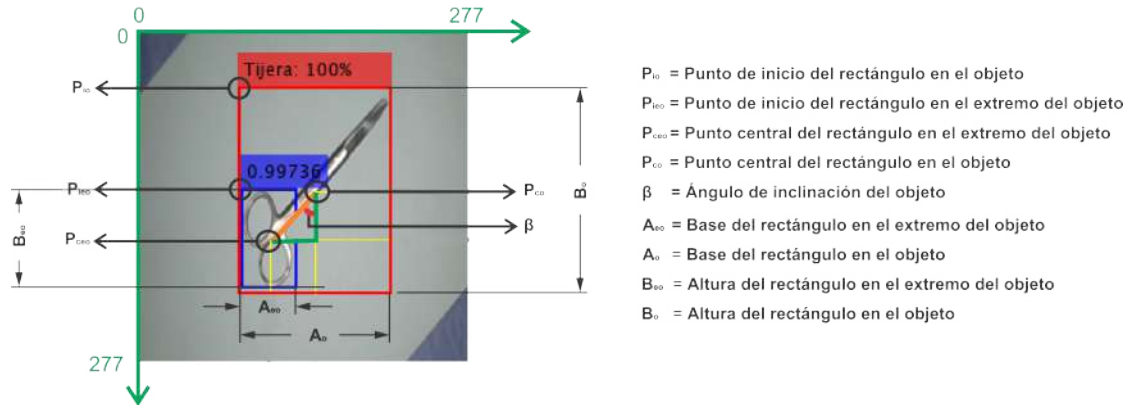


Figura 6.30: Variables para la obtención de la posición y orientación del instrumento en la imagen. **Fuente:** Elaboración propia

En la figura 6.30 se muestra las variables con las que se obtiene los puntos medios de las cajas delimitadoras, así mismo indica el esquema realizado para encontrar la orientación del instrumento. El procedimiento analítico se describe en seguida.

Los siguientes vectores son obtenidos mediante redes neuronales y corresponden a las cajas delimitadoras de los objetos y sus extremos:

$$\text{Caja delimitadora del objeto} = [P_{ioy}, P_{ioy}, A_o, B_o]$$

$$\text{Caja delimitadora del extremo del objeto} = [P_{ieoy}, P_{ieoy}, A_{eo}, B_{eo}]$$

Calculó de los centros en las cajas delimitadoras:

$$P_{coy} = P_{ioy} + A_o/2$$

$$P_{ceoy} = P_{ieoy} + A_{eo}/2$$

$$P_{coy} = P_{ioy} + B_o/2$$

$$P_{ceoy} = P_{ieoy} + B_{eo}/2$$

Cálculo para orientación del objeto:

$$\beta = \arctan \frac{P_{ceox} - P_{cox}}{P_{ceoy} - P_{coy}}$$

Con el fin de corroborar lo anterior se incorpora el procedimiento en software MATLAB y se obtienen los resultados de la figura 6.31 donde se visualiza la detección del instrumento con una anotación de la clase correspondiente, la precisión y el grado de orientación, además de una línea que tiene como origen el centro de la herramienta quirúrgica y finaliza en el centro de su extremo.

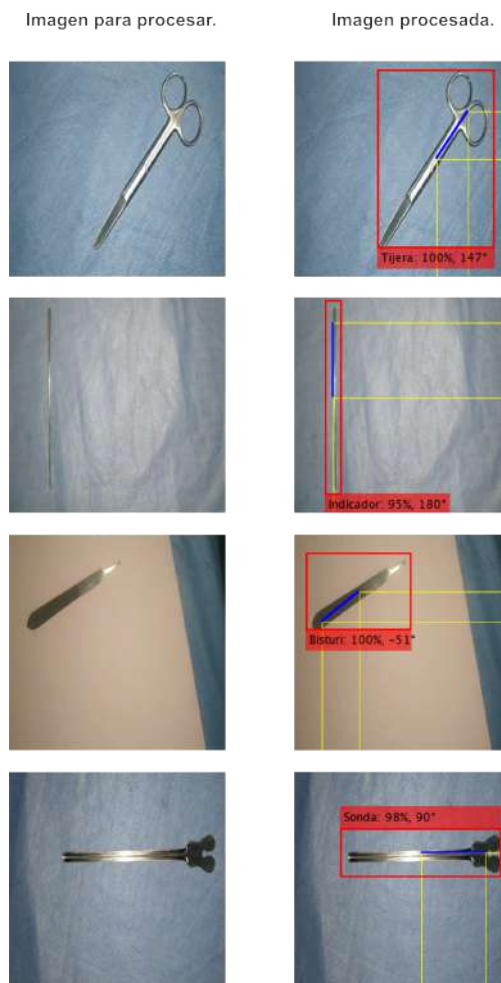


Figura 6.31: Posición y orientación del instrumento en la imagen. **Fuente:** Elaboración propia

En cuanto a la asertividad del detector entrenado para el reconocimiento y ubicación del extremo en el material quirúrgico se refleja en las gráficas de precisión con respecto al recuerdo de la figura 6.32, de manera general se observa un decremento en los valores promedio de acierto, esto debido a que el tamaño de las cajas delimitadoras a detectar es más pequeño, aunque sigue teniendo un margen de error aceptable, tanto el porcentaje del extremo de la tijera recta como de la sonda acanalada sigue siendo superior al 96%, mientras que en el indicador y el bisturí quirúrgico se redujo alrededor del 73 %.

Un indicio de la correcta funcionalidad de la red está en que la gráfica de precisión con respecto al recuerdo se mantuvo a pesar de su aumento, el menor dato obtenido corresponde al bisturí con un valor aproximado del 85 %.

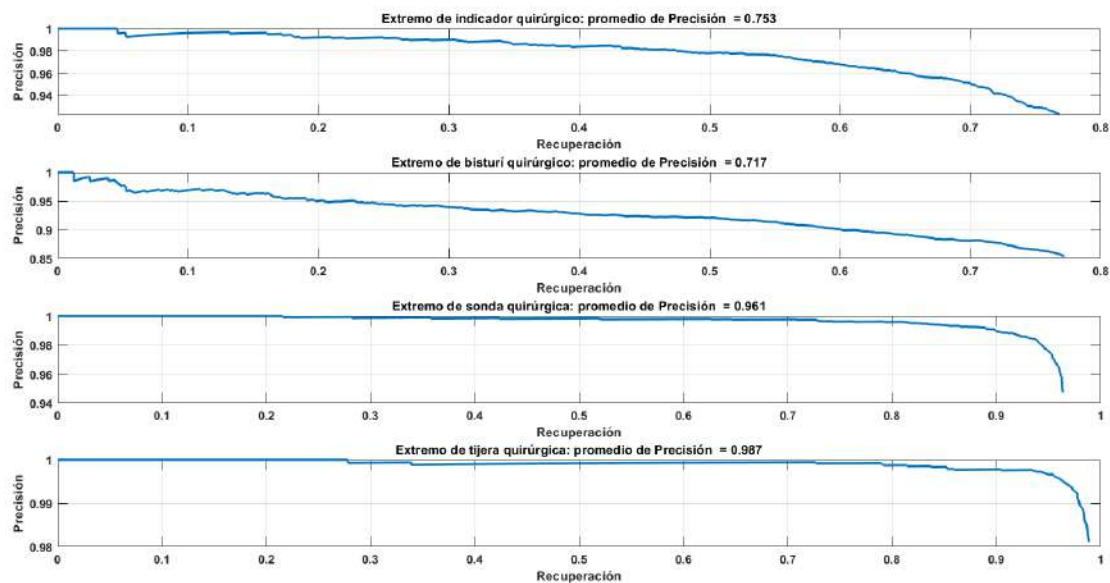


Figura 6.32: Precisión con respecto al recuerdo en la detección del extremo de los instrumentos. **Fuente:** Elaboración propia

Previo al escado de px a mm y al envío de esta información al prototipo robótico se debe tener en cuenta el ajuste de las cámaras, estas pueden ser integradas al

sistema de dos formas ya sean fijas o movibles. Ya que este proyecto debe ser lo más robusto posible en cuanto a su tolerancia a perturbaciones, se decidió integrar cámaras movibles y realizar un ajuste automático en su calibración.

La implantación de cámaras fijas, aunque requieren de menos esfuerzos, están sujetas a que, si en algún momento, ya sea por desgaste, mal uso, o accidentalidad cambian su posición de forma drástica o leve, el sistema perderá eficacia y en muchos casos precisión.

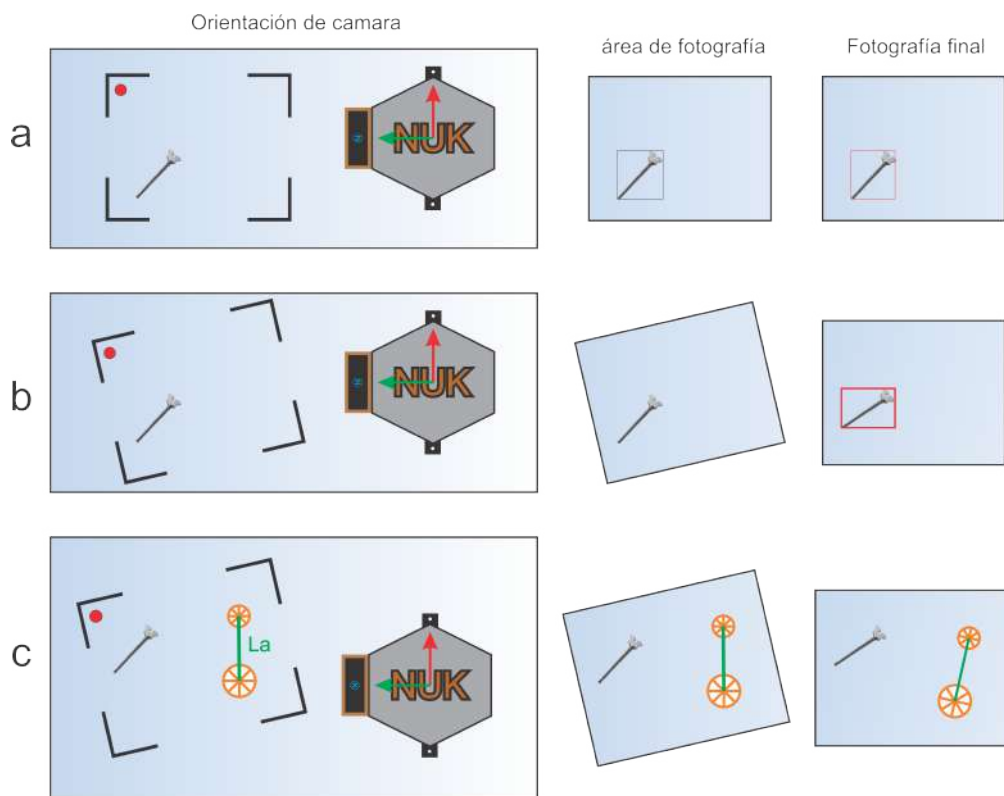


Figura 6.33: Posición del objeto en la imagen según la orientación de la cámara.
Fuente: Elaboración propia

El análisis que conlleva el tipo de cámaras seleccionadas se presenta mediante la figura 6.33, en el apartado a nos encontramos un enfoque de la imagen totalmente alineado con los ejes coordenados del sistema robótico y como consecuencia no se

genera ningún cambio entre el área de la imagen y la fotografía final, si solo fuera este el caso, para encontrar el centro del instrumento respecto al robot se podría tomar fijas las medidas desde el eje x y y de éste a los laterales del fotograma, hacer el escalado de px a mm teniendo en cuenta el tamaño verdadero de la herramienta quirúrgica, y sumar o restar los valores obtenidos de la posición del centro del material quirúrgico respecto a la imagen con la del robot.

En el apartado b y c se muestra una cámara con un enfoque no alineado con el sistema coordinado al sistema robótico, como se puede observar el área de la fotografía ya no corresponde a la fotografía final, presenta una rotación que cambia la posición y orientación de la herramienta en la imagen, la solución planteada consiste en añadir dos figuras circulares en las que sus centros formen una línea paralela al eje x del robot y de esta manera realizar los cálculos de posición tomando como eje de referencia la línea entre las circunferencias de la fotografía final.

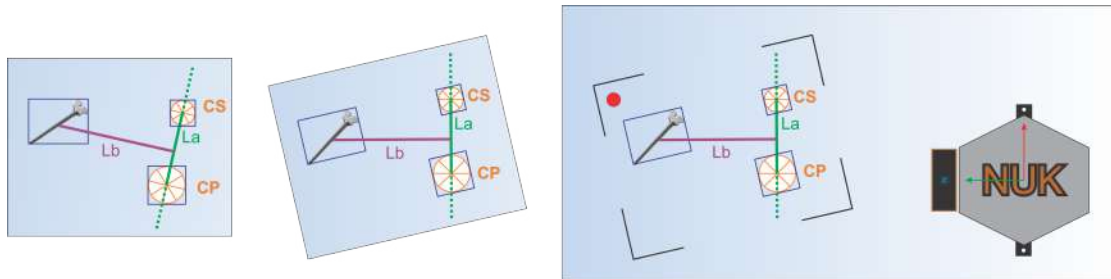


Figura 6.34: Obtención de posición del instrumento en el área de trabajo con cámara movable. **Fuente:** Elaboración propia

Para encontrar la posición del objeto respecto al robot se parte de la fotografía final y se pretende de ella encontrar las coordenadas reales de su ubicación, este procedimiento se logra observar en la figura 6.34 donde Cp refiere a la circunferencia principal, Cs a la circunferencia secundaria, La a la recta coincidente con los centros de los círculos y Lb a la distancia que hay desde el centro del instrumento a la línea La , formando una perpendicular a esta. La línea Lb es el objetivo a

encontrar en este procedimiento, ya que es la distancia real en píxeles que existe desde la vertical de los círculos hasta el objeto en el eje y del sistema robótico.

El desarrollo analítico inicia por hallar las coordenadas de los centros de Cp, Cs y el objeto a partir del vector que contiene la información de las cajas delimitadoras como resultado de las redes neuronales entrenadas.

$$\text{Caja delimitadora del objeto} = [P_{iox}, P_{ioy}, A_o, B_o]$$

$$\text{Caja delimitadora del círculo principal} = [P_{icpx}, P_{icpy}, A_{cp}, B_{cp}]$$

$$\text{Caja delimitadora del círculo secundario} = [P_{icsx}, P_{icsy}, A_{cs}, B_{cs}]$$

Cálculo de los centros P_{co} , P_{cpx} y P_{cs} correspondientes al objeto, al círculo principal y círculo secundario respectivamente.

$$P_{cox} = P_{iox} + A_o/2 \quad P_{cpx} = P_{icpx} + A_{cp}/2 \quad P_{csx} = P_{icsx} + A_{cs}/2$$

$$P_{coy} = P_{ioy} + B_o/2 \quad P_{cpy} = P_{icpy} + B_{cp}/2 \quad P_{csy} = P_{icsy} + B_{cs}/2$$

Encontrar ecuación de la recta que pasa por los centros de las circunferencias.

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{P_{csy} - P_{cpx}}{P_{csx} - P_{cpx}}$$

$$y - y_1 = m(x - x_1)$$

$$y - P_{cpx} = m(x - P_{cpx})$$

$$y - P_{cpx} = mx - mP_{cpx}$$

$$y - P_{cpx} - mx + mP_{cpx} = 0$$

$$-mx + y + mP_{cpx} - P_{cpx} = 0$$

$$a = -m; b = 1; c = mP_{ccpx} - P_{ccpy}$$

Encontrar la distancia del punto en el centro del objeto a la recta.

$$d = \frac{ax_p + by_p + c}{\sqrt{a^2 + b^2}}$$

$$Lbp = \frac{-mP_{cox} + P_{coy} + mP_{ccpx} - P_{ccpy}}{\sqrt{(-m)^2 + 1}}$$

$$Lbp = \frac{m(P_{ccpx} - P_{cox}) + P_{coy} - P_{ccpy}}{\sqrt{m^2 + 1}}$$

Una vez encontrada la distancia Lbp correspondiente a la medida de Lb en píxeles, se hace necesario calcular Dxtp, formando el triángulo rectángulo de la figura 6.35.

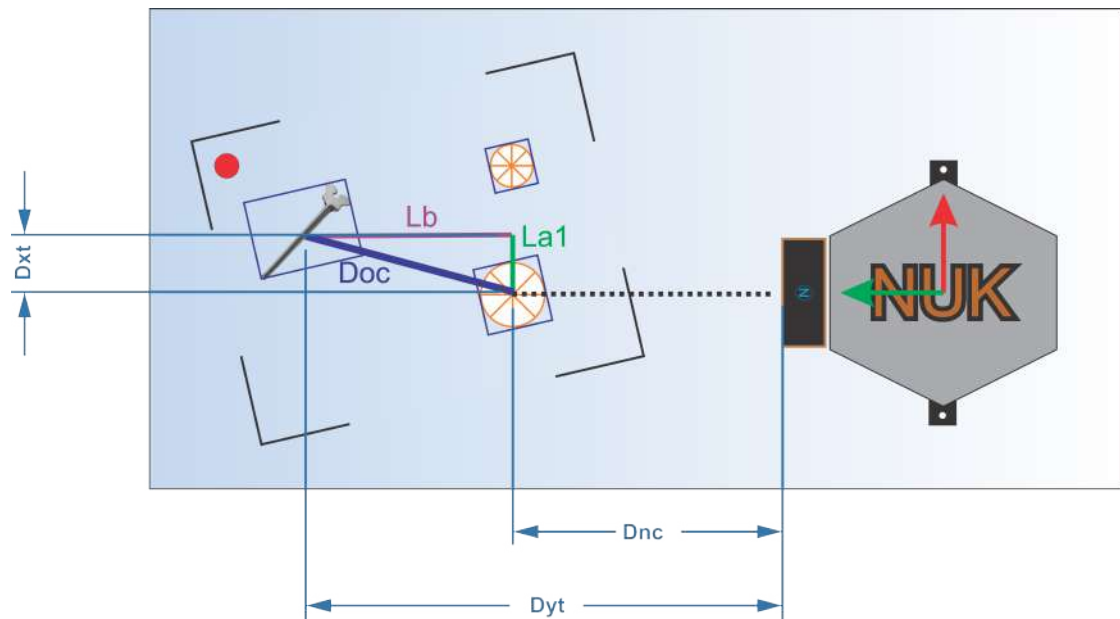


Figura 6.35: Coordenadas del centro del instrumento respecto al sistema robótico.
Fuente: Elaboración propia

El desarrollo analítico que permite la obtención de Dxtp se realiza de la siguiente forma:

$$Docp = \sqrt{(P_{cox} - P_{ccpx})^2 + (P_{coy} - P_{ccpy})^2}$$

$$Docp^2 = (P_{cox} - P_{ccpx})^2 + (P_{coy} - P_{ccpy})^2$$

$$Docp^2 = Lbp^2 + Dxtp^2$$

$$Dxtp = \sqrt{(P_{cox} - P_{ccpx})^2 + (P_{coy} - P_{ccpy})^2 - Lbp^2}$$

Encontrados estos dos valores, es posible realizar un escalado de px a mm, en este caso se hizo uso del diámetro del círculo, utilizar la medida de esta figura soluciona el error de distorsión de los ejes x y y al momento de cambiar de tamaño la imagen para ingresarla a la red neuronal, pues nos es posible generar una escala para cada eje.

$$\text{Caja delimitadora del círculo principal} = [P_{icpx}, P_{icpy}, A_{cp}, B_{cp}]$$

Escala en eje x:

$$E_x = \frac{D_{cp}}{A_{cp}}$$

Escala en eje Y:

$$E_y = \frac{D_{cp}}{B_{cp}}$$

Donde:

D_{cp} = Diámetro del círculo principal en mm

Conversiones de Lbp y Dxtp a mm:

$$Lb = Lbp * E_x$$

$$Dxt = Dxtp * E_y$$

Con el fin de ubicar la herramienta respecto al robot, teniendo la posición de esta referida al centro de la circunferencia principal, se dejó fija la posición del

círculo, haciéndolo coincidente su centro al eje y del prototipo robótico y situándolo a una distancia Dnc .

De lo anterior se puede definir las coordenadas del objeto respecto al robot como $(Dxt, Lb + Dnc)$

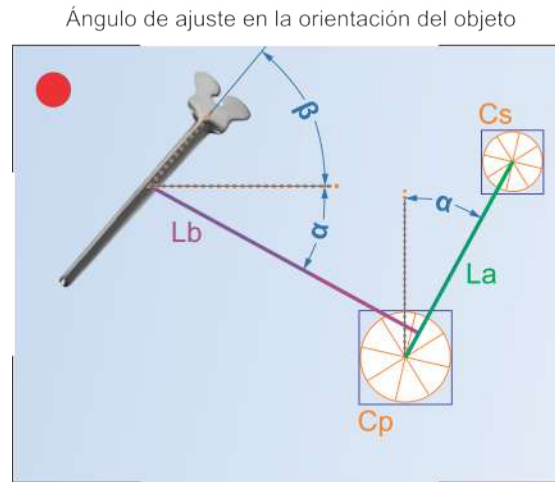


Figura 6.36: Ángulo de ajuste en la orientación del objeto. **Fuente:** Elaboración propia

Del análisis de la figura 6.30 se llegó anteriormente a estimar la orientación según la ecuación:

$$\beta = \arctan \frac{P_{ceox} - P_{cox}}{P_{ceoy} - P_{coy}}$$

En el caso de tener una cámara movable que capture la imagen con una cierta rotación, esta ecuación necesita un ángulo de ajuste denominado en la figura 6.36 como α . La orientación final del instrumento respecto a la línea La se expresa mediante la adición de β y α , para calcular esta última lo podemos hacer mediante las coordenadas del centro en el círculo tanto principal como secundario.

$$\alpha = \arctan \frac{P_{ccsx} - P_{ccpx}}{P_{ccsy} - P_{ccpy}}$$

Ingresado los anteriores cálculos en software MATLAB e incluyendo los detectores necesarios para este procedimiento, se muestra la figura 6.37 con la cual se puede evidenciar el correcto funcionamiento del sistema de detección, posición y orientación del objeto en la imagen.

Funcionamiento de cálculo para la posición y orientación del objeto

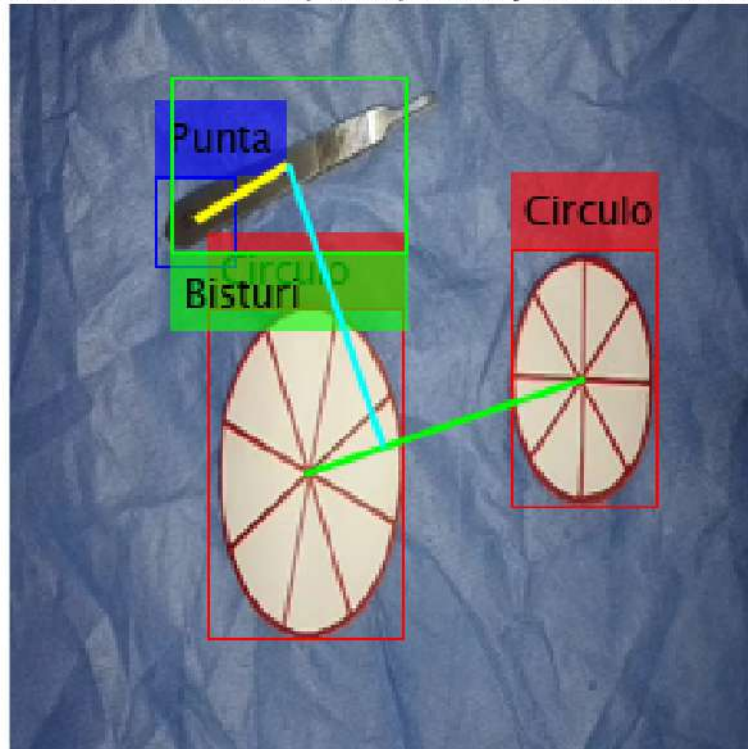


Figura 6.37: Funcionamiento del sistema de detección, posición y orientación del objeto en la imagen. **Fuente:** Elaboración propia

6.3. Implementación de comandos de voz

La integración de un asistente de voz se realiza como se indica en la figura 6.38, para ello usa el servidor IFTTT con el fin de crear acciones según los comandos especificados, estas acciones permiten enviar información a diferentes receptores como puede ser adafruit si se desea conectar el asistente directamente con sistemas embebidos, o como es nuestro caso se ejecuta el método webhooks para colocar datos en canales del software ThingSpeak.

La razón de emplear ThingSpeak se fundamenta en que sirve como facilitador del acceso de datos por parte de MATLAB para luego ser analizados.



Figura 6.38: Implementación de Google assistant al sistema. **Fuente:** Elaboración propia

Los comandos creados mediante IFTTT para la ejecución de una tarea se des-

criben mediante la tabla 6.5 junto con la función que conllevara cada uno dentro del sistema.

Comandos de voz	
Comando	Función
Ok Google activar encender cámaras	Enciende las cámaras en la interfaz de control e inicia los detectores entrenados
Ok Google activar apagar cámaras	Apaga las cámaras y los detectores en la interfaz, así mismo cierra el programa de control y finaliza el procedimiento quirúrgico
Ok Google activar Indicador	Asigna el instrumento indicador como el requerido por el personal médico
Ok Google activar Bisturí	Asigna el instrumento bisturí como el requerido por el personal médico
Ok Google activar Sonda	Asigna el instrumento sonda como el requerido por el personal médico
Ok Google activar Tijera	Asigna el instrumento tijera como el requerido por el personal médico

Cuadro 6.5: Comandos de voz definidos mediante IFTTT para la ejecución de una tarea. **Fuente:** Elaboración propia

6.4. Diseño de la interfaz de unión entre los sistemas

En el diseño de una interfaz que permita la integración del sistema robótico, el sistema de visión artificial, y el de reconocimiento de voz, es importante tener en cuenta que debe de ser intuitiva, capaz de ser manipulada fácilmente por personal no experto en las áreas que integran la creación del presente proyecto. Además, debe de guardar un historial con los datos relevantes del procedimiento quirúrgico, como se estipuló mediante los resultados del desenvolvimiento en el capítulo dos.

Nuestro planteamiento incorpora tres tiempos principales el primero refiere a la pantalla con la cual se encuentra el usuario al iniciar la interfaz, mostrada en la

figura 6.39, para su implementación se consideró diseñar los campos de recepción de datos del procedimiento quirúrgico, la información que se desea ser ingresada corresponde al formulario de la lista de verificación de la seguridad en la cirugía propuesto por la OMS, figura 2.2. En la parte izquierda de esta pantalla se puede observar el método de conexión de la interfaz con el sistema robótico.

Figura 6.39: Interfaz de unión en el tiempo uno. **Fuente:** Elaboración propia

El segundo tiempo importante es cuando el usuario ha terminado de ingresar la información y ha realizado la verificación de conexión con el prototipo robótico, en este momento se activa el botón de registrar con el que se da inicio al procedimiento y guarda los datos ingresados en un historial escribiéndolos en un archivo tipo Excel y siendo guardado en el ordenador asignándole la fecha y hora de la intervención como nombre, 6.40.

Al seleccionar registrar se cerrará automáticamente esta pantalla y se abrirá la indicada en la figura 6.41 siendo el tercer y último tiempo, en ella en un inicio estarán las cámaras desactivadas las cuales serán iniciadas mediante comando de voz, mostrara también los datos más relevantes en cuanto a la ocupación del

quirófano con los cuales se podría comprobar su disponibilidad, una vez activadas las cámaras y los detectores el sistema indicara mediante vídeo en tiempo real el reconocimiento realizado por las redes neuronales entrenadas. Al finalizar se desactivan las cámaras mediante el asistente de Google y se cerrara de forma automática la interfaz dando fin a la intervención quirúrgica.

Figura 6.40: Interfaz de unión en el tiempo dos. **Fuente:** Elaboración propia

Figura 6.41: Interfaz de unión en el tiempo tres. **Fuente:** Elaboración propia

Capítulo 7

Puesta en marcha

En este capítulo se describirá el funcionamiento del sistema en conjunto y su desempeño respecto a la asistencia en la instrumentación quirúrgica.

Inicialmente, al encender el prototipo robótico, este realiza su conexión a red wifi y su estado será visualizado en la pantalla integrada en el robot. Seguido a eso se activa como servidor web y el router o modem le asigna una ip disponible, es esta ip la que se debe definir en el software MATLAB para ejecutar el envío de datos por medio del protocolo tcp/ip; puesto que esta variable cambia según el modem al cual realiza su conexión y a los dispositivos que se encuentren conectados a él, se envía esta dirección haciendo uso de la base de datos firebase, es así como de forma automática el software obtiene la variable ip y hace la conexión sin importar su cambio de valor, figura 7.1.

Firebase es una base de datos en la nube y es por ello que para hacer el intercambio de información con MATLAB se realiza mediante la API de REST permitiéndonos leer, crear, actualizar o suprimir información estructurada en este almacén.

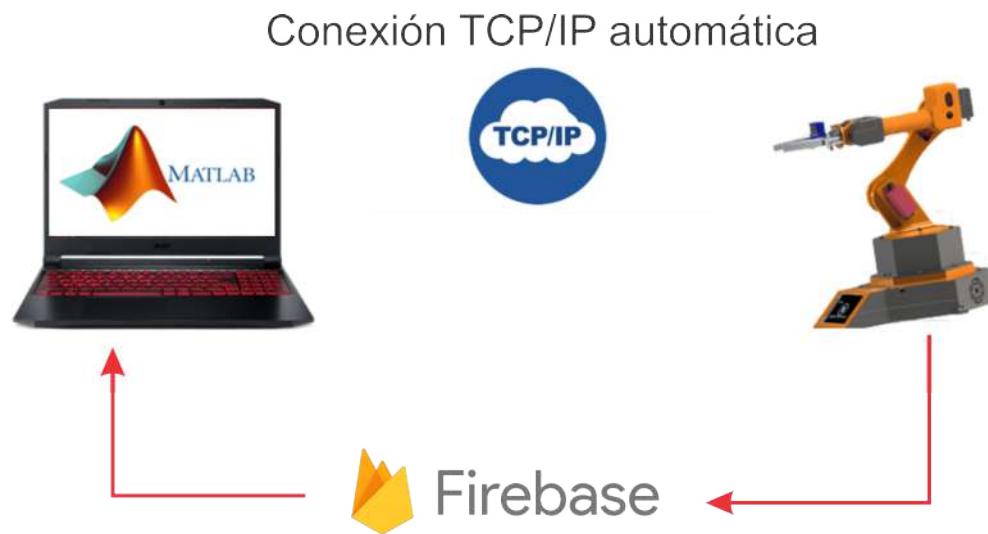


Figura 7.1: Conexión para TCP/IP automática mediante firebase. **Fuente:** Elaboración propia

Además de la conexión tcp/ip se programó el robot antropomórfico para que sea capaz de recibir información simultáneamente desde la plataforma Blynk, es por ello que después de estar el sistema conectado a la red se es posible manipularlo por medio de estos dos métodos, cada uno de ellos para esta aplicación poseen ventajas y desventajas, tcp/ip por su parte necesita que el prototipo esté conectado a la misma red WLAN es decir hay una restricción en la distancia del robot al ordenador, pero es posible sincronizar muchas de las funcionalidades de MATLAB con su funcionamiento por ejemplo las redes neuronales convolucionales. Blynk siendo una plataforma en línea no requiere estar cerca del dispositivo, es posible controlar su posición desde cualquier lugar, solo como requisito se tendría que los dos dispositivos estén conectados a internet, pero su uso se limita a la creación de una interfaz utilizando elementos predeterminados como son pulsadores y sliders, figura 7.2.

Como se mencionó en el apartado de integración de los comandos de voz,

tanto MATLAB como el robot independientemente pueden enlazarse con Google assistant, pero para esta aplicación se facilita hacer la conexión directamente en el software que contendrá la detección de objetos.

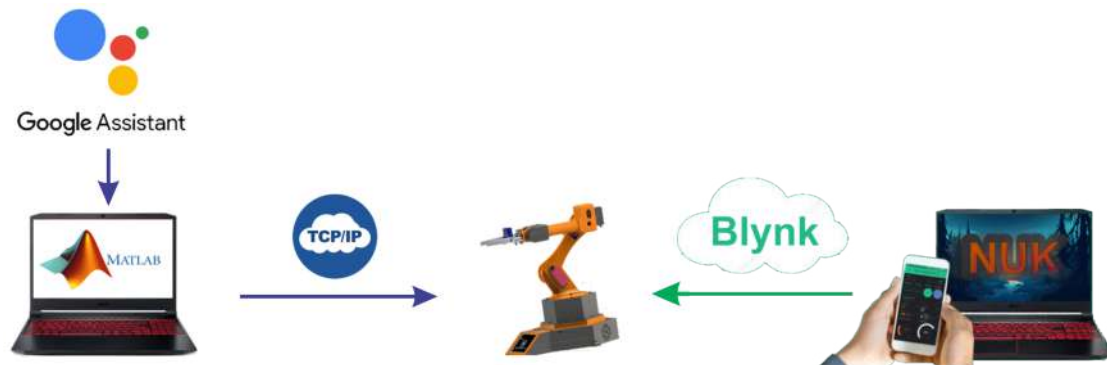


Figura 7.2: Software para manipular el sistema robótico. **Fuente:** Elaboración propia

Mediante la figura 7.3 se muestran las dos interfaces creadas para cada forma de control, en el apartado de diseño de la interfaz de unión del sistema de visión artificial y robótico se describe detalladamente el modelo generado mediante protocolo tcp/ip, en el que se inicia con la recepción de datos a guardar como historial de cada procedimiento quirúrgico, siguiente a esto se verifica la conexión y se activan los detectores entrenados para realizar la manipulación del instrumental médico, que consiste en entregarle a este la herramienta quirúrgica que solicite mediante comandos de voz y retirarla en el momento que finalice su uso, con el fin validar este funcionamiento en el prototipo se realizó el reconocimiento de cuatro clases de instrumentos, en los cuales están una sonda acanalada, un mango de bisturí quirúrgico número tres, una tijera de mayo recta y un indicador.

Respecto a la interfaz realizada en Blynk logra ser de gran utilidad en el momento de verificar el funcionamiento electromecánico del robot, así como la correcta

construcción de trayectorias, pues mediante esta es posible mover cada grado de libertad a un ángulo establecido, también integra la función demo la cual se encarga de realizar un movimiento predeterminado de cada articulación y finalizar su movimiento en la posición inicial para la ejecución de trayectorias. Una característica implementada en este modelo es la capacidad de guardar puntos tridimensionales establecidos mediante el posicionamiento cada gdl, tiene como limite la selección de treinta puntos y puede generar una trayectoria entre ellos.



Figura 7.3: Interfaces de manipulación del sistema robótico. **Fuente:** Elaboración propia

Finalizamos el presente capítulo con la figura 7.4 donde se muestra el funcionamiento del sistema en su área de trabajo.

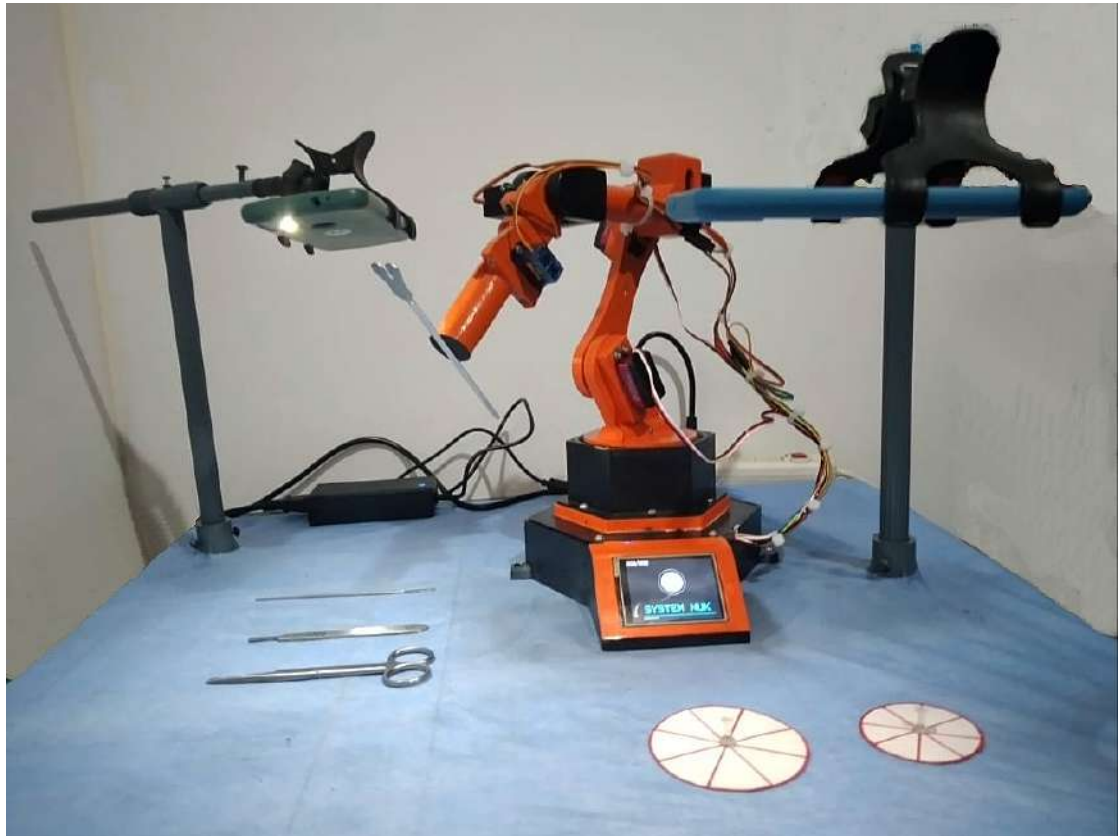


Figura 7.4: Funcionamiento del sistema robótico y de visión artificial para la asistencia en instrumentación quirúrgica. **Fuente:** Elaboración propia

Capítulo 8

Análisis de resultados

El prototipado del sistema basado en robótica y visión artificial para la asistencia en instrumentación quirúrgica, se creó siguiendo los lineamientos obtenidos a partir de un estudio de desenvolvimiento del producto, obteniendo de ello las características principales con las cuales contaría el proyecto.

Para el sistema robótico se realizó un planteamiento electrónico compuesto por sistemas embebidos, generando como principales características de ello, una conexión a la red WLAN, el correcto funcionamiento de los actuadores y de los dispositivos de visualización.

El diseño mecánico para el robot antropomórfico se ejecutó mediante software CAD en el que se obtuvo como resultado un modelo virtual del ensamblado del prototipo, con este modelo se hizo el estudio cinemático y se lo simuló en MATLAB.

Como método de manufactura en las piezas diseñadas se utilizó la impresión 3D, con la que se alcanzó una buena precisión en las dimensiones de cada parte, una vez obtenidos todos los componentes para el ensamble de la estructura física, se hizo un proceso de acabado para generar un dispositivo visualmente agradable.

Ensamblado y calibrado el sistema robótico físico se le realizó pruebas del correcto posicionamiento tridimensional. De este procedimiento se puede concluir que el sistema robótico cumplió con las expectativas esperadas, existiendo una relación precisa entre los resultados del modelo cinemático simulado y el real.

La obtención de trayectorias se implementó siguiendo el método del interpolador cúbico, y este proceso fue integrado en el sistema embebido controlador, ya que ejecutar este algoritmo en el ordenador y enviar los datos por comunicación tcp/ip genera un retraso considerable además de pérdida de información debido a su cantidad.

Finalizado el sistema del robot antropomórfico, se procedió a la realización e implementación de un software que sea capaz de manipularlo con el fin de cumplir la funcionalidad en la asistencia quirúrgica.

La principal característica del software está en la integración de algoritmos de deep learning para obtener el reconocimiento, la posición y la orientación del instrumental quirúrgico. Para este proceso se ejecutaron pruebas mediante dos estructuras de detección, como lo son SSD y YOLO V3, en cada modelo de detección se realizó entrenamientos con almacenes de datos de diferentes tamaños; llegando a la conclusión de que, en ambas estructuras entre más grande sea el dataset de aprendizaje más precisa será la red.

El detector que obtuvo mejores resultados mediante los test en los que se analizó la precisión con respecto al recuerdo fue YOLO V3, así mismo se observó una asertividad más alta de esta estructura en la implementación en tiempo real.

Para la integración del sistema robótico con el de visión artificial se creó una interfaz la cual, además de esta unión, realiza un proceso de recepción de información del proceso quirúrgico y lo almacena en forma de historial.

Para la validación de este prototipo se hizo un reconocimiento de 4 clases de instrumentos quirúrgicos con un tamaño de dataset igual a 20000 imágenes etiquetadas, para trabajos futuros a este o la implementación de este modelo en un ámbito quirúrgico se ve necesario aumentar las clases del instrumental a detectar y como consecuencia los datos de entrada para el entrenamiento de la red. Así mismo se verían mejoras teniendo en cuenta la normativa de asepsia y el método de limpieza del equipo.

Bibliografía

- [1] M. Tolino, “Oblitos. causas, consecuencias clínicas y legales. prevención.” *Rev. Argent. Cir*, vol. 13, p. 89, 2009.
- [2] M. Velasco, J. López, V. Pradel, C. Vázquez, y R. García, “Oblitoma en la cirugía ginecoobstétrica: un problema creciente. reporte de un caso,” 2017. [En línea]. Disponible: <https://pesquisa.bvsalud.org/portal/resource/pt/biblio-892524>
- [3] S. Rodríguez, “Olvidados e infrecuentes: oblitos. recopilacion de casos - revista electrónica de portalesmedicos.com,” *Ginecologia y Obstetricia*, 2011. [En línea]. Disponible: <https://www.portalesmedicos.com/publicaciones/articles/2886/2/Olvidados-e-infrecuentes%3Aoblitos.-Recopilacion-de-casos>
- [4] MathWorks, “Redes neuronales convolucionales,” 2022. [En línea]. Disponible: <https://es.mathworks.com/discovery/convolutional-neural-networkmatlab.html#c%C3%B3mo-funcionan-las-cnn>
- [5] C. SolidWorks, “Introducción a solidworks,” 2022. [En línea]. Disponible: https://my.solidworks.com/solidworks/guide/SOLIDWORKS_Introduction_ES.pdf

- [6] T. Carrillo, D. Robles, y J. Rueda, “Diseño, construcción y control de un brazo robótico tipo antropomórfico de 6 grados de libertad y de un brazo robótico tipo scara de 4 grados de libertad,” Universidad San Francisco de Quito, Tech. Rep., 2022. [En línea]. Disponible: <http://repositorio.uac.edu.co/handle/11619/1658>
- [7] A. Barrientos, A. Cruz, L. Peñín, y C. Balaguer, *Fundamentos de robótica*. McGraw-Hill, 2007. [En línea]. Disponible: <https://books.google.com.co/books?id=ArEMPAAACAAJ>
- [8] Autodesk, “History of 3d printing: it’s older than you think,” 2022. [En línea]. Disponible: <https://redshift.autodesk.com/history-of-3d-printing>
- [9] F. Pugin, P. Bucher, y P. Morel, “History of robotic surgery : From aesop and zeus to da vinci,” *Journal of Visceral Surgery*, vol. 148, no. 5, Supplement, pp. e3–e8, 2011, robotic surgery. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/S1878788611000324>
- [10] J. Pransky, “Robodoc - surgical robot success story,” *Industrial Robot: An International Journal*, vol. 24, no. 3, pp. 231–233, Jan 1997.
- [11] R. Valero, Y. Ko, S. Chauhan, O. Schatloff, A. Sivaraman, R. Coelho, F. Ortega, K. Palmer, R. Sanchez, H. Davila, X. Cathelineau, y V. Patel, “Cirugía robótica: Historia e impacto en la enseñanza,” *Actas Urológicas Españolas*, vol. 35, no. 9, pp. 540–545, 2011. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/S0210480611001902>
- [12] B. Davies, “A review of the state-of-the-art of ”smart”systems in surgery,” in *2008 15th International Conference on Mechatronics and Machine Vision in Practice*, 2008, pp. 151–155. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/4749523>

- [13] C. Michael y C. Eric, “Twenty-first century surgery using twenty-first century technology: Surgical robotics,” *Current Surgery*, vol. 61, no. 5, pp. 466–473, 2004. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/S014979440400042X>
- [14] F. Santos, “Seguimiento de gases quirúrgicos en imágenes de laparoscopia empleando redes neuronales convolucionales,” Universidad de Valladolid, Tech. Rep., 2020. [En línea]. Disponible: <https://uvadoc.uva.es/handle/10324/45061>
- [15] D. Sanz, “Detección de instrumentos quirúrgicos en imágenes para cirugía laparoscópica robotizada,” Universidad de Valladolid, Tech. Rep., 2021. [En línea]. Disponible: <https://uvadoc.uva.es/handle/10324/47195>
- [16] Z. Zhao, T. Cai, F. Chang, y X. Cheng, “Real-time surgical instrument detection in robot-assisted surgery using a convolutional neural network cascade,” *Healthc Technol Lett*, vol. 6, no. 6, pp. 275–279, Nov. 2019.
- [17] J. Rojas, S. Escrucería, M. Suárez, y C. Peña, “Diseño e implementación de un brazo robótico de bajo costo para la automatización en el proceso de análisis bacteriológico,” *INGE CUC*, vol. 8, no. 1, 2012. [En línea]. Disponible: <https://revistascientificas.cuc.edu.co/ingecuc/article/view/258>
- [18] A. Naar y F. Barreto, “Modelo de red neuronal convolucional para el diagnóstico de neumonía en imágenes radiológicas,” Universidad del Sinú, Tech. Rep., 5 2019. [En línea]. Disponible: <http://repositorio.unisinucartagena.edu.co:8080/xmlui/handle/123456789/50>
- [19] O.M.S, *Lista de verificación de la seguridad de la cirugía*. Organización Mundial de la Salud, 8 2022.

[20] R.A.E, “Oblito,” *Diccionario de la real academia Española*, 5 2022.