



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE MECATRÓNICA

**AUTOMATIZACIÓN DE UN SISTEMA IDENTIFICADOR Y
POSICIONADOR DE OBJETOS A TRAVÉS DE UN BRAZO
ROBÓTICO MEDIANTE VISIÓN ARTIFICIAL CON LENGUAJE
PYTHON**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Mecatrónica

AUTORES: Anthony Stalin Llerena Buenaño
Marc Jens Salazar Villamar
TUTOR: Ing. David Mateo Cortez Saravia

Guayaquil - Ecuador
2022

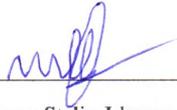
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Anthony Stalin Llerena Buenaño** con documento de identificación N° **1804885059** y **Marc Jens Salazar Villamar** con documento de identificación N° **0957893118**; manifestamos que:

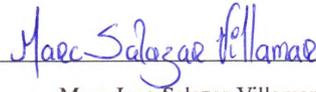
Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo.

Guayaquil, 03 de marzo del año 2022

Atentamente,



Anthony Stalin Llerena Buenaño
1804885059



Marc Jens Salazar Villamar
0957893118

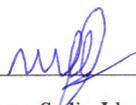
**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA
UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, **Anthony Stalin Llerena Buenaño** con documento de identificación N° **18044885059** y **Marc Jens Salazar Villamar** con documento de identificación N° **0957893118**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del **Dispositivo Tecnológico: AUTOMATIZACIÓN DE UN SISTEMA IDENTIFICADOR Y POSICIONADOR DE OBJETOS A TRAVÉS DE UN BRAZO ROBÓTICO MEDIANTE VISIÓN ARTIFICIAL CON LENGUAJE PYTHON**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

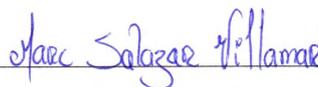
En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo a final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana

Guayaquil, 03 de marzo del año 2022

Atentamente,



Anthony Stalin Llerena Buenaño
18044885059



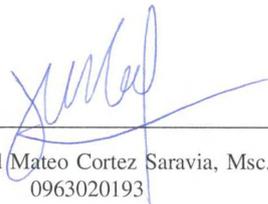
Marc Jens Salazar Villamar
0957893118

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **David Mateo Cortez Saravia**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **AUTOMATIZACIÓN DE UN SISTEMA IDENTIFICADOR Y POSICIONADOR DE OBJETOS A TRAVÉS DE UN BRAZO ROBÓTICO MEDIANTE VISIÓN ARTIFICIAL CON LENGUAJE PYTHON**, realizado por **Anthony Satalin Llerena Buenaño** con documento de identificación N° **1804885059** y por **Marc Jens Salazar Villamar** con documento de identificación N° **0957893118**, obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivo Tecnológico** que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 03 de marzo del año 2022

Atentamente,



Ing. David Mateo Cortez Saravia, Msc.
0963020193

ÍNDICE

I.	RESUMEN	1
II.	ABSTRACT	2
III.	INTRODUCCIÓN	3
IV.	PROBLEMA DE INVESTIGACIÓN	4
IV-A.	Importancia y alcance	4
V.	JUSTIFICACIÓN	5
VI.	OBJETIVOS	6
VI-A.	OBJETIVO GENERAL	6
VI-B.	OBJETIVOS ESPECIFICOS	6
VII.	MARCO TEÓRICO	7
VII-A.	FUNDAMENTO DEL BRAZO ROBÓTICO	7
VII-B.	TIPOS DE BRAZOS MANIPULADORES O BRAZOS ROBÓTICOS	7
VII-B1.	ROBOT CARTESIANO	7
VII-B2.	ROBOT CILÍNDRICO	7
VII-B3.	ROBOT POLAR	7
VII-B4.	ROBOT SCARA	7
VII-B5.	ROBOT ESFÉRICO O ARTICULADO	7
VII-B6.	ROBOT PARALELO	7
VII-C.	BRAZO ROBÓTICO XARM 1S	7
VII-C1.	ESPECIFICACIONES TÉCNICAS	7
VII-C2.	CARACTERÍSTICAS DEL BRAZO	7
VII-C3.	LISTA DE PARTES	8
VII-C4.	MORFOLOGÍA DEL ROBOT	8
VII-C5.	ESTRUCTURA MECÁNICA	8
VII-C6.	ELEMENTOS MECÁNICOS	8
VII-C7.	GRADOS DE LIBERTAD (GDL o DOF)	9
VII-C8.	SISTEMA DE TRANSMISIÓN	9
VII-C9.	ESLABONES Y ARTICULACIONES	9
VII-C10.	POSICIÓN Y ORIENTACIÓN	9
VII-C11.	SERVOS DE BUS INTELIGENTE	9
VII-D.	BANDA TRANSPORTADORA	9
VII-E.	CERRADURA ELÉCTRICA SOLENOIDE	10
VII-E1.	PUENTE H (DRIVER L298N)	10
VII-F.	VISIÓN ARTIFICIAL	10
VII-F1.	CONCEPTO DE VISIÓN ARTIFICIAL	10
VII-F2.	PRINCIPALES APLICACIONES DE VISIÓN ARTIFICIAL A NIVEL INDUSTRIAL	11
VII-F3.	MÉTODOS DE CAPTACIÓN DE IMÁGENES	11
VII-F4.	DIAGRAMA DE BLOQUES	12
VII-F5.	ELEMENTOS DE VISIÓN ARTIFICIAL	12
VII-G.	PROTOCOLOS DE COMUNICACIÓN	12
VII-G1.	PROTOCOLO TCP	12
VII-H.	SISTEMAS EMBEBIDOS Y CONTROL DEL ROBOT	13
VII-H1.	RASPBERRY-PI4	13
VII-I.	SISTEMA DE INTERACCIÓN CON EL USUARIO	13
VII-I1.	SOFTWARE: PYTHON	13
VII-I2.	MATLAB	14
VII-J.	SISTEMA DE CONTROL PARA BRAZO ROBOTICO	14
VII-J1.	TEORÍA DE LOS CONTROLADORES	14
VII-K.	CINEMÁTICA PARA UN BRAZO ROBÓTICO	14
VII-K1.	MODELO CINEMÁTICO	14
VII-K2.	CINEMÁTICA DE BRAZOS ROBÓTICOS	15

VIII. MARCO METODOLÓGICO	16
VIII-A. DIAGRAMA DE FLUJO DEL PROCESO	16
VIII-B. DIAGRAMA DE PARTES ELÉCTRONICAS CONECTADAS AL MICROCONTROLADOR	16
VIII-C. DISEÑO Y ELABORACIÓN DEL ESPACIO DE TRABAJO	17
VIII-C1. SISTEMA DE LLEGA DEL OBJETO	17
VIII-C2. SISTEMA DE DETECCIÓN	21
VIII-C3. SISTEMA DE RECOLECCIÓN Y POSICIONADOR	23
IX. RESULTADOS	36
IX-A. Microcontrolador	36
IX-B. Brazo robótico xARM1S	36
IX-C. Mesa de trabajo	37
IX-D. SOPORTE DE CÁMARA	38
IX-E. Resultados de práctica 1 (Calibración de colores)	39
IX-F. Resultados práctica 2 (Posicionamiento manual de servomotores por ángulos)	41
IX-G. Resultados práctica 3 (Identificación y posición de objetos según color)	44
X. CRONOGRAMA DE ACTIVIDADES	47
XI. PRESUPUESTO	48
XII. CONCLUSIONES	49
XIII. RECOMENDACIONES	49
XIV. ANEXOS	51
XIV-A. GUÍA DE PRÁCTICA 1	51
XIV-B. GUÍA DE PRÁCTICA 2	52
XIV-C. GUÍA DE PRÁCTICA 3	53
XIV-D. PRUEBAS DE VISIÓN ARTIFICIAL Y ENSAMBLE DEL BRAZO	54
XIV-E. DISEÑO 3D, SIMULACIÓN Y CONSTRUNCCIÓN DE ESTRUCTURAS	55

ÍNDICE DE FIGURAS

1.	Especificaciones técnicas	7
2.	Características técnicas del brazo	7
3.	Listado de partes del brazo, [12]	8
4.	Brazo robotico XARM 1S,[12]	8
5.	Eslabones y articulaciones del brazo robótico, [1]	8
6.	Clasificación de eslabones, Elaboración propia.	9
7.	Motores LX-225 y LX-15D, [12]	9
8.	Banda transportadora, [6]	10
9.	Driver controlador L298N, [16]	10
10.	Diagrama de Control de visión artificial	11
11.	Aplicación de visión artificial en la industria, [6]	11
12.	Diagrama de bloques, [6]	12
13.	Diagrama de bloques adaptado de, [6]	12
14.	Protocolo TCP, [9]	13
15.	Raspberry PI4	13
16.	Entorno del software Python	14
17.	Esquema de bloque de los controladores PID con dinámica invertida, H_{∞} , H_{∞} robustificado, [7]	14
18.	Esquema cinemática directa	15
19.	Esquema cinemática inversa	15
20.	Diagrama de Funcionamiento del proceso	16
21.	Elementos electrónicos	16
22.	Espacio de trabajo	17
23.	Fabricación de estructura	17
24.	Estructura para la llegada del objeto	18
25.	Propiedades de materiales para bandas planas y redonda	19
26.	Tambor motriz y tambor terminal	20
27.	Estructura 3D del soporte de para la cámara	21
28.	Importación de librerías en Python	21
29.	configuración de cámara	22
30.	Valores HSV de colores	22
31.	Contorno, Área, posición	22
32.	Montaje de estructura del brazo	23
33.	Subensambles de las partes del brazo	24
34.	Diseño 3D del brazo xARM1S	24
35.	Dimensiones del brazo robótico	25
36.	Sliders para selección o posición de motores	26
37.	Posicionamiento (Posición Home del brazo)	26
38.	Estado de pinza posición abierta/cerrada	26
39.	Secuencia de movimiento	27
40.	Recolección de objeto	27
41.	Traslado de objeto	28
42.	Clasificación de objeto	28
43.	Matrices rotacionales y traslacionales	29
44.	Matrices para la obtención del Jacobiano	29
45.	Declaración de función y dimensiones del robot	30
46.	Evaluación de puntos en los ejes x,y,z y Jacobiano	30
47.	Cálculo de psudoinversa, definición del controlador, cálculo de error y salida de la función	31
48.	Condiciones iniciales, ecuación diferencial	31
49.	Trayectori eslabón x_4, y_4, z_4	32
50.	Trayectoria eslabones (x_3, y_3, z_3) - (x_2, y_2, z_2) - (x_1, y_1, z_1)	32
51.	Caption	33
52.	Caption	33
53.	Trayectoria entre el origen y banda transportadora	34
54.	Trayectoria entre el origen y los puntos de clasificación	34
55.	Simulación de movimiento del brazo	35

56.	Microcontrolador seleccionado para el control del Sistema.	36
57.	Importación de librería xArm y servos de bus inteligente	36
58.	Simulación de posicionamiento	37
59.	Ensamble y diseño 3D	37
60.	Mesa de trabajo	38
61.	Ensamble del soporte para la cámara	38
62.	Visualización de valores HSV para calibración de color del objeto verde limón.	39
63.	Valores de sensibilidad y HSV de los objetos de color rojo, azul, verde y verde limón.	39
64.	Visualización de la Detección de color del objeto aplicando valores HSV	40
65.	Sliders para posicionamiento de cada servomotor del brazo	41
66.	Seteo de posición home del brazo	41
67.	Posición intermedia entre el origen y la cinta	41
68.	Configuración para posición deseada hasta la banda	42
69.	Posición deseada hasta la banda	42
70.	Posición de llegada del objeto color rojo	43
71.	Interfaz final para la tercera práctica	44
72.	Funcionamiento correcto de los electroimanes	44
73.	scale=1	45
74.	Recolección de objeto	45
75.	Clasificación de objeto por color	46
76.	Cronograma de actividades	47
77.	Presupuesto	48
78.	Guía de práctica 1	51
79.	Guía de practica 2	52
80.	Guía de práctica 3	53

I. RESUMEN

El presente proyecto técnico de titulación, “Automatización de un sistema identificador y posicionador de objetos a través de un brazo robotico mediante visión artificial con lenguaje Python” .Tiene como objetivo facilitar las prácticas de laboratorio de control de la UPS, realizando una maqueta didáctica, con la cual los estudiantes y maestros puedan enriquecer los procesos de aprendizaje y enseñanza.

Las prácticas didácticas tienen relación al uso de un brazo robótico, con el fin de ejecutar el traslado de un objeto seleccionado por color desde un punto A hacia un punto B que, al tener la capacidad de ser operado de forma remota, se acopla de manera óptima a las restricciones de aforo y distanciamiento social que conlleva la emergencia sanitaria.

Los elementos principales que componen el módulo son: Raspberry pi4 modeloB 2 GB, Brazo Robótico Lewansoul Hiwonder xArm1S, Cámara Logitech C270, Fuente 12v, servomotor Futaba s3003.

PALABRAS CLAVES: Phyton/ Raspberry/Servomotor/ Mecatronica/Lewansoul//

II. ABSTRACT

This project named, "Automation of an Object Identifier and Positioner System operated by a robotic arm through artificial vision with Python programming language". Its objective is to facilitate control in laboratory practices, making a didactic model where both, students and teachers can enrich their learning and teaching processes in relation to the use of a robotic arm, to transfer an object of a specific color from point A to point B which, having the ability to be operated remotely, is optimally coupled to the actual capacity restrictions, and social distancing due to the current sanitary emergency. The main elements that make up this project are the following: Raspberry pi4 model B 2 GB, Robotic arm Lewansoul Hiwonder xArm1S, Logitech camera C270, 12v power supply, Futaba servo s3003.

KEY WORDS: Python/ Raspberry/ Servo/ Mechatronics/Lewansoul/

III. INTRODUCCIÓN

La robótica ha evolucionado gracias a la tecnología moderna y el desarrollo de distintas herramientas informáticas, el entendimiento de la complejidad y aplicaciones de los robots requiere diversos conocimientos de distintas áreas de la ingeniería. En la actualidad un manipulador o brazo robótico, es indispensable dentro del sector industrial para solucionar posibles inconvenientes incrementando la eficiencia dentro de las líneas productivas.

Con la finalidad de garantizar la optimización de los procesos antes mencionados, se plantea la propuesta que consiste en la automatización de un sistema identificador y posicionador de objetos a través de un brazo robótico mediante visión artificial con lenguaje de programación Python. Este proyecto se llevará a cabo mediante la investigación del arte de máquinas, el diseño de la máquina se realizará con la ayuda del software de diseño SolidWorks, la implementación de un sistema ejecutado a través del desarrollo de una guía de prácticas.

IV. PROBLEMA DE INVESTIGACIÓN

El margen de error que existe al clasificar un gran volumen de objetos por color de forma manual es una de las más importantes limitaciones dentro de la industria de la producción en masa, y por ende, se considera como uno de los grandes desafíos que enfrenta la ingeniería hoy. Motivo por el cual, las empresas que desarrollan tecnología de punta han decidido incluir la robótica aplicada con el fin de automatizar la mayoría de sus procesos en la línea de producción, usualmente mediante el uso de un brazo Robótico, decisión que ha reducido drásticamente y definitivamente dicho margen.[2].

En la Universidad Politécnica Salesiana sede Guayaquil, específicamente en el laboratorio de Robótica podemos encontrar dos brazos de modelo KUKA KR AGILUS, mismos que operan por medio de un controlador remoto. Debido a la situación actual de aforo limitado a causa de la pandemia nos hemos visto obligados a recibir dichas prácticas de manera Online, encontrándonos con la problemática de no poder experimentar el control de dicho brazo de manera presencial, provocando que el proceso pedagógico presente ciertas dificultades.

IV-A. Importancia y alcance

Este proyecto brinda soporte para adquirir experiencia dentro de lo que implica realizar prácticas de manera remota, ya que, al contar con un proceso completamente automático, y lograr que, con la supervisión de su docente, los alumnos puedan conectarse a través de softwares con acceso al VNC experimentando bajo tutela el manejo de la Raspberry del modelo a escala. Otorgando así una experiencia introductoria a temática de la industria 4.0, más amigable y didáctica pese a las restricciones sanitarias.

V. JUSTIFICACIÓN

El presente proyecto tiene como objetivo eliminar la incertidumbre que existe al momento de clasificar un gran volumen de objetos de forma manual, transportados en una línea de producción. La propuesta para lograr dicho objetivo es el desarrollo de un sistema de clasificación de objetos mediante la implementación de un brazo robótico y reconocimiento de imágenes a través del uso de visión artificial en tiempo real. Facilitando el traslado y clasificación de productos por color, mismo que esta especialmente diseñado y programado para trabajos repetitivos y de gran volumen. Al momento de programar se requiere un lenguaje de fácil comprensión y con acceso libre a sus herramientas y librerías, por lo cual, utilizando el programa Python se realizará la codificación de órdenes en formato de algoritmos, con el fin de sistematizar el comportamiento físico y lógico. Se implementó la visión artificial para la selección o la identificación, caracterización, de objetos observados por las cámaras utilizando OpenCV.

Debido al gran requerimiento por parte del mercado en Latinoamérica de una industria más desarrollada y que responda a los avances tecnológicos de vanguardia, se propone el desarrollo del presente proyecto con el fin de aportar innovación y desarrollo y competitividad tecnológica aplicando la automatización dentro del mercado nacional e internacional dentro de la industria manufacturera [5].

Gracias a la implementación del brazo robótico para la clasificación de objetos por color tiene como resultado inmediato: La reducción de tiempos de manufactura, fallos, reducción significativa en los accidentes laborales, mejora de las condiciones de trabajo gracias al incremento de seguridad para el personal, cumplimiento de normativa y orden en las áreas de producción y almacén [13].

Las empresas que posean un inventario de accesorios que necesiten ser seleccionados por color para su almacenaje se vería altamente beneficiada debido a la eficiencia y agilidad con la cual los movimientos de los objetos serían clasificados y organizados de manera automática. [10].

La programación del software encargado del reconocimiento de imágenes de comparación con “objetos modelo” para su clasificación, puede ser desarrollada en diferentes plataformas que manejan sus propios softwares. Sin embargo, el proyecto propuesto tiene como prioridad la facilidad de comprensión y el acceso libre del software utilizado, cumpliendo con el objetivo de realización de prácticas de laboratorio de control en la Universidad Politécnica Salesiana. El principio fundamental de marcha es el transporte de objetos por una banda transportadora, localizar el accesorio u objeto con un sensor, mismo que lo indentifica por su color, capturar la imagen con una cámara compatible a la minicomputadora en tiempo real y procesarla por Python para clasificarlos en sus correspondientes espacios de almacenamiento. Con el proceso descrito se logró una clasificación de objetos eficiente. Finalmente, el proyecto busca generar un impacto en la educación universitaria del Ecuador, impulsando a que se invierta de manera eficiente en la educación tecnológica de la población y que los educadores trabajen en colaboración con el sector privado para desarrollar las capacidades necesarias para los empleos del futuro. Así también, esperando que los estudiantes y docentes que integran los centros de educación superior logren enriquecer su proceso de aprendizaje y enseñanza con esta maqueta.

VI. OBJETIVOS

VI-A. OBJETIVO GENERAL

Automatizar la clasificación de objetos en la línea de producción, incrementando la eficiencia y reduciendo los riesgos del proceso, mediante el desarrollo de un sistema identificador y posicionador de objetos con la ayuda de un brazo robótico a través de visión artificial con lenguaje Python.

VI-B. OBJETIVOS ESPECIFICOS

- Seleccionar los equipos y componentes a utilizar, de acuerdo con criterios técnicos y en cumplimiento de las normativas.
- Realizar el ensamble y diseño 3D del brazo robótico.
- Construir la estructura de la maqueta para el sistema propuesto.
- Programación del sistema propuesto mediante tarjeta de control usando python para catalogar los objetos.
- Elaborar propuesta de prácticas didácticas para enriquecer el conocimiento teórico adquirido en clases.

VII. MARCO TEÓRICO

VII-A. FUNDAMENTO DEL BRAZO ROBÓTICO

Un brazo robotico o brazo manipulador, es la agrupación de varios elementos electromagnéticos que se encargan del movimiento de un elemento terminal, consta una cadena de partes estructurales rígidos, enlaces o eslabones, que se acoplan mediante articulaciones que generan el movimiento de los eslabones. [6]

VII-B. TIPOS DE BRAZOS MANIPULADORES O BRAZOS ROBÓTICOS

VII-B1. ROBOT CARTESIANO: “Consta de 3 grados de libertad, uno rotacional en la base para la función de giro y dos de movimientos prismáticos, sus ejes coinciden con los ejes cartesianos”. [15]

VII-B2. ROBOT CILÍNDRICO: “Sus ejes forman un sistema de coordenadas cilíndricas. Son utilizados para soldadura por punto, fundición a presión y manipulación de máquinas y herramientas” [8].

VII-B3. ROBOT POLAR: Sus ejes forman un sistema polar de coordenadas, consta con dos articulaciones de rotación y lineales. Se utilizan para soldadura por punto, soldadura por gas y por arco, manipulación de máquinas y herramientas.

VII-B4. ROBOT SCARA: : “Se caracteriza por ser un robot de pequeñas dimensiones que principalmente ha sido diseñado para realizar trabajos repetitivos a gran velocidad y con mucha precisión. Realizan aplicaciones de coger y dejar piezas de un punto A a un punto B, que en inglés lo definimos como Pick and Place”.

VII-B5. ROBOT ESFÉRICO O ARTICULADO: Tiene como mínimo tres articulaciones rotativas. Cumple operaciones de ensamblaje, fundición a presión, soldadura a gas y por arco, pintado en aerosol, manipulación de máquinas herramientas [6].

VII-B6. ROBOT PARALELO: Tienen articulaciones prismáticas o rotatorias concurrentes. Son usados para la simulación de movimiento de vuelo, maquinado de piezas, transferencia de piezas a alta velocidad.

VII-C. BRAZO ROBÓTICO XARM 1S

Pertenece a la empresa LewanSoul, especializada en distribuir a nivel mundial componentes de robótica para uso educacional o industrial. Diseñado para ejecutar movimientos programados e implementarlo en áreas de trabajo. [12]

VII-C1. ESPECIFICACIONES TÉCNICAS: Algunas de las características adicionales incluyen un diseño ligero en la pinza y un indicador audible de engranaje de seguridad está incluido en las cinco cajas de engranajes evitando cualquier daño potencial del engranaje durante la operación.

ESPECIFICACIONES	DETALLES
Alimentación de entrada	100v-240v (50/60) Hz
Alimentación de salida	7.5v / 5 A
Material	Metal inoxidable

Figura 1. Especificaciones técnicas

VII-C2. CARACTERÍSTICAS DEL BRAZO: Se detallan características principales del robot.

CARACTERÍSTICAS	DETALLES
Peso	60 g
Dimensión	40*20*40,5mm
Velocidad	0,16 sec/g/60° 7,4v
Esfuerzo de torsión	15kg cm 6,6V; 17kg cm 7,4V
Voltaje de trabajo	6-8,4V
Corriente sin carga	100mA
Servo	Servos de bus inteligente de un solo eje/LX-15D/LX-225

Figura 2. Características técnicas del brazo

VII-C3. *LISTA DE PARTES*: Se detallan las piezas que necesitamos para el ensamble de nuestro brazo.

xArm 1S Robot Arm- Product list			
	Suction cup*4		Servo Controller*1pcs
	Bottom bracket*2pcs		Power adapter*1pcs
	Servo horn*1pcs		Mechanical gripper(with ID1 servo)*1pcs
	Big U-shaped bracket*1pcs		LX-15D servo(ID2 without servo horn)*1pcs
	Servo side cover*1pcs		LX-15D servo(ID3 with two servo horns)*1pcs
	Small plastic column*1pcs		LX-15D servo(ID4 with two servo horns)*1pcs
	Top-shaped bracket*1pcs		LX-22S servo(ID5 with two servo horns)*1pcs
	Bearing*1pcs		LX-15D servo(ID6 with one servo horn)*1pcs
	Rotational station*1set		Small plastic column*1pcs
	Wireless handle and receiver*1pcs		Big plastic column*1pcs
			Servo wires*5pcs
			USB cable*1pcs
			M4*8 machine screw*6
			M4*20 machine screw*6
			M2*6 self-tapping screw*10
			M4*32 copper column*4
			M2*5 self-tapping screw*6
			Rivets and rivets connector*4
			M4*16 machine screw*6
			M3*6 machine screw*26
			M2.6*5 self-tapping screw*3
			M2.5 machine screw*6
			M2.5 nut*12
			M2 nut*6
			Screwdriver (yellow)*1

Figura 3. Listado de partes del brazo, [12]

VII-C4. *MORFOLOGÍA DEL ROBOT*: Se conoce cómo morfología a la estructura y forma que tiene un robot.



Figura 4. Brazo robotico XARM 1S,[12]

VII-C5. *ESTRUCTURA MECÁNICA*: La representación de las partes y el armazón mecánico de un robot se denomina morfología. Un manipulador robotico está combinado por una serie de articulaciones y eslabones, cada articulación representa la interconexión a través de dos eslabones consecutivos.

VII-C6. *ELEMENTOS MECÁNICOS*:

- Desplazamiento (traslación)
- Giro (rotación)
- Combinación (roto-traslación)

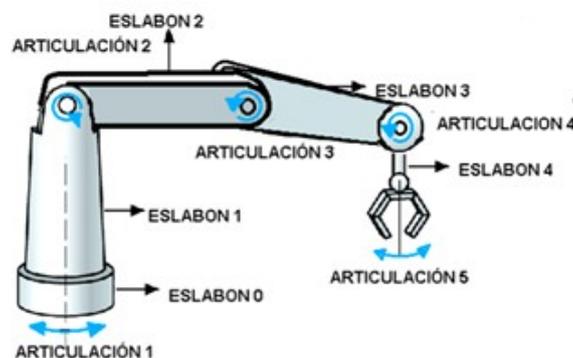


Figura 5. Eslabones y articulaciones del brazo robótico, [1]

VII-C7. GRADOS DE LIBERTAD (GDL o DOF): El desplazamiento de un sistema mecánico se puede catalogar de acuerdo con el número de GDL que dispone, su sistema es igual al número de parámetros autónomos que se necesitan para determinar de manera única su posición en el espacio en cualquier momento del tiempo, hay que observar que GDL se determina con respecto a valores de referencias seleccionados. Los GDL son la cantidad de parámetros independientes que definen la posición del elemento terminal del brazo robótico. [4].

VII-C8. SISTEMA DE TRANSMISIÓN: Son los componentes delegados de transmitir el movimiento desde los motores a las articulaciones, transforma el movimiento lineal-circular.

VII-C9. ESLABONES Y ARTICULACIONES: Las conexiones o enlaces son los bloques de construcción básicos de todos los mecanismos.

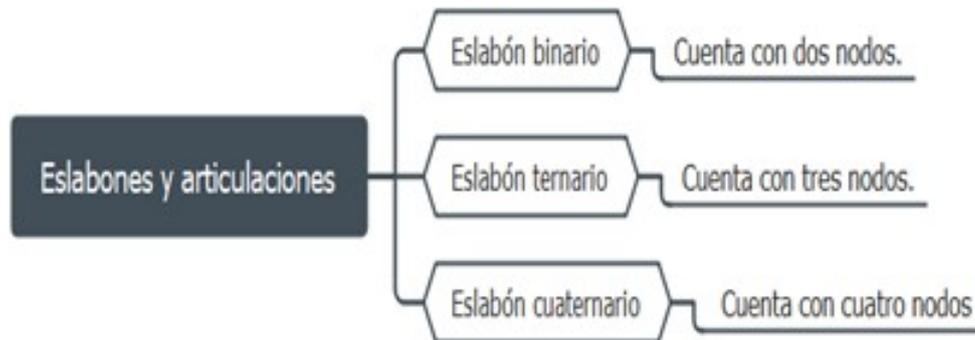


Figura 6. Clasificación de eslabones, Elaboración propia.

Un sistema mecánico está conformado por distintas articulaciones, produciendo movimientos lineales o rotacionales. Su sistema de medición para articulaciones lineales se da en metros (m), y las articulaciones rotacionales se dan en radianes o grados. [11]

VII-C10. POSICIÓN Y ORIENTACIÓN: La trayectoria de un robot involucra el movimiento espacial de cada uno de sus partes, con el objetivo de que un robot pueda retirar un objeto. Es necesario conocer los puntos de posición y orientación en relación con la base del robot.

VII-C11. SERVOS DE BUS INTELIGENTE: Comparados a otros servos de bus que existen en el mercado, estos tienen una estructura compacta, haciendo que el diseño del robot más biomiméticos y delicado. Utiliza engranajes metálicos extendiendo el tiempo de servicio o vida útil, diseñado para facilitar el cableado, contiene un indicador RGB para indicarnos temperatura, voltaje y ángulo de retroalimentación, protegiendo al servo de sobrecalentamientos.[12]



Figura 7. Motores LX-225 y LX-15D, [12]

VII-D. BANDA TRANSPORTADORA

Es un método de transporte el cual se basa en una banda que se mueve constantemente entre sus dos tambores, ésta banda es arrastrada por fricción por cualquier de los dos tambores, uno es accionado por un motor y el otro tambor hace de retorno a la banda girando libremente.

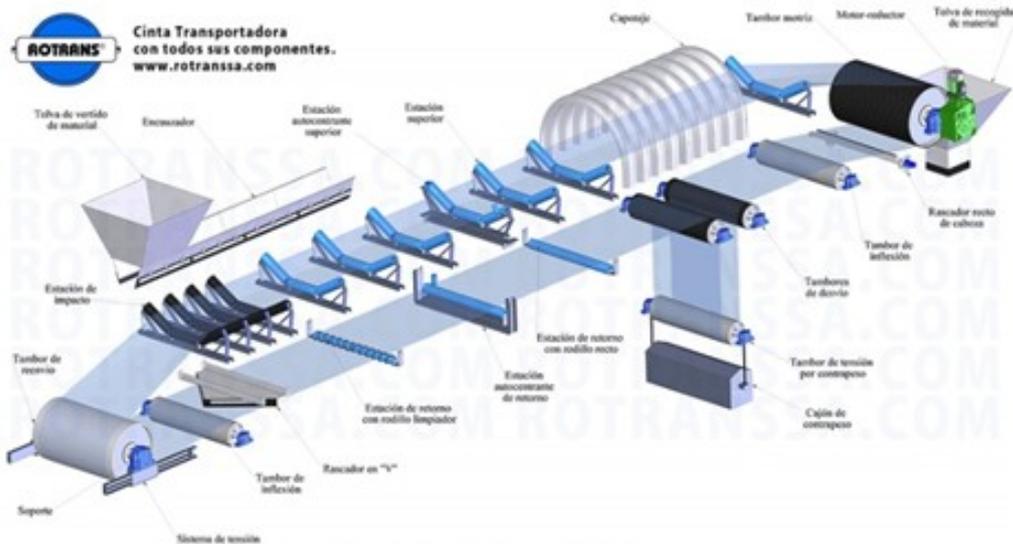


Figura 8. Banda transportadora, [6]

VII-E. CERRADURA ELÉCTRICA SOLENOIDE

Dispositivo encargado de convertir la energía eléctrica en energía mecánica (lineal), alimentado con una tensión adoptada de 12V, moviéndose a una distancia específica.

VII-E1. **PUENTE H (DRIVER L298N)**: Consta del circuito integrado L298N con un doble puente H permitiendo controlar el cambio de giro del motor, también activa y regula la velocidad del motor por medio de una señal de +5 V que puede ser del tipo PWM (Modulación por Ancho de Pulso). [16]

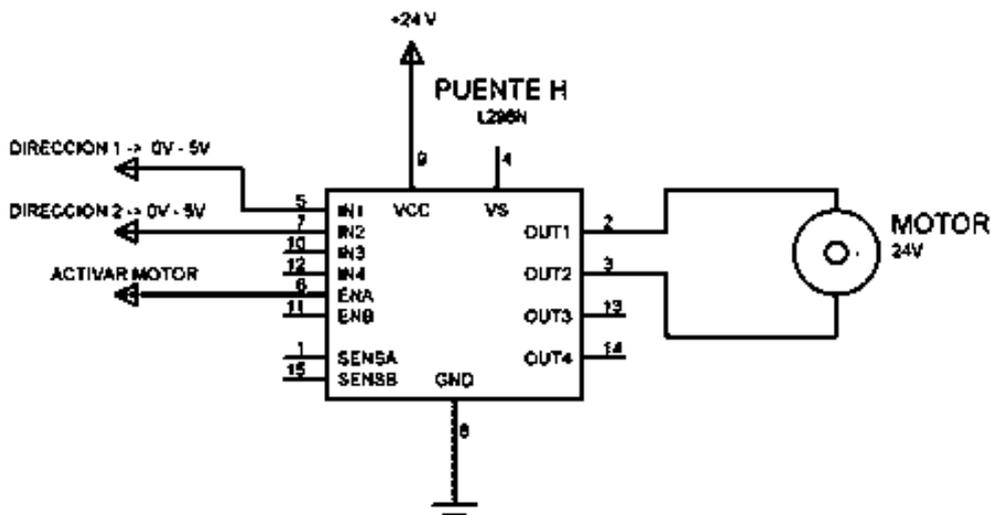


Figura 9. Driver controlador L298N, [16]

VII-F. VISIÓN ARTIFICIAL

VII-F1. **CONCEPTO DE VISIÓN ARTIFICIAL**: “La Inteligencia artificial es una rama de las ciencias informáticas y esta a su vez comprende la visión artificial, la cual ha sido desarrollada con el fin de obtener, procesar y analizar algún tipo de información conseguida a través de imágenes digitales por medio de un computador programado” [6]. Dicho conjunto de procesos destinado a la realización de análisis de imagen se basa en: captación de imágenes, memorización de la información, procesamiento e interpretación de los resultados.

Con la visión artificial se pueden realizar:

- Automatización en labores reiterativas de inspección realizadas por operadores.
- Desarrollar controles de calidad para los productos finales que no eran posibles verificar por métodos tradicionales.

- Realizar inspecciones de calidad a alta velocidad de los productos sin personal presente.
- Reducir el tiempo de los periodos en los procesos automatizados.
- Realizar inspecciones en procesos donde existe diversidad de piezas con cambios frecuentes de producción

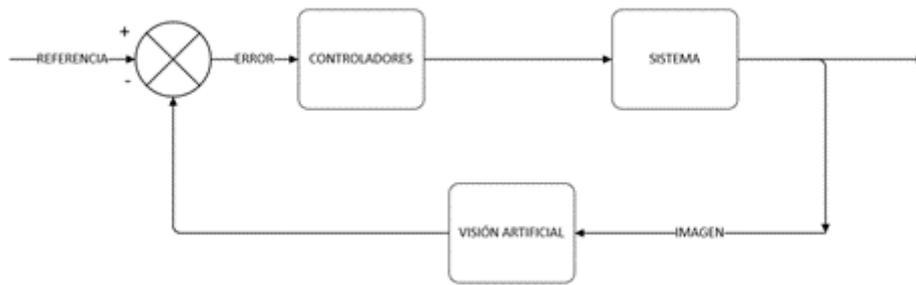


Figura 10. Diagrama de Control de visión artificial

VII-F2. PRINCIPALES APLICACIONES DE VISIÓN ARTIFICIAL A NIVEL INDUSTRIAL: Se realizan varios procesos de manera más rápida y eficaz que de la manera tradicional no se lograrían.

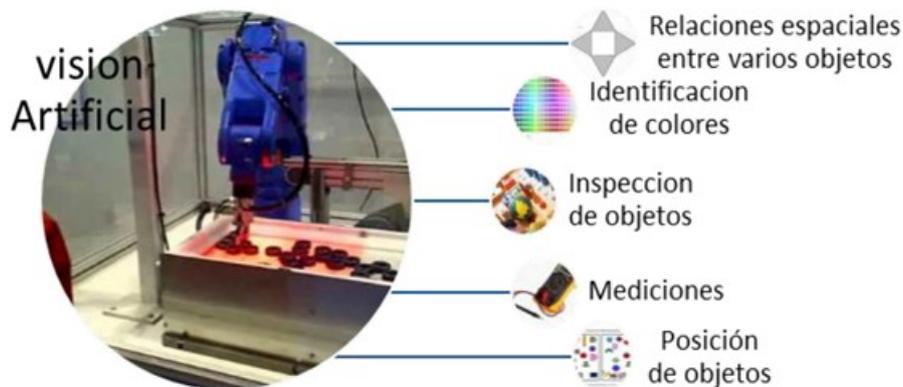


Figura 11. Aplicación de visión artificial en la industria, [6]

VII-F3. MÉTODOS DE CAPTACIÓN DE IMÁGENES:

- **DIGITAL** Los resultados obtenidos de la medida o muestreos realizados a intervalos de tiempo espaciados regularmente, siendo el valor de dicha función un número positivo y entero. Los valores que esta función recibe en cada punto dependen del brillo que presenta en esos puntos la imagen original.
- **PÍXEL** Una imagen digital se estima como una cuadrícula. Cada parte de esta cuadrícula se denomina Píxel (Picture element). La resolución o calidad estándar de una imagen digital se estima de 512x484 Píxel.
- **NIVEL DE GRISES** Cuando una imagen es digitalizada.
- **IMAGEN BINARIA** Sólo tiene dos niveles de grises; negro y blanco. Cada píxel se transforma en blanco o negro en función del nivel binario o UMBRAL.
- **ESCENA** Área de memoria donde se mantienen los parámetros de la inspección de un objeto en particular: imágenes de patrón, tolerancias, datos a visualizar, etc.
- **WINDOWS (VENTANAS DE MEDIDA)** Es el espacio específico de la imagen recogida que se quiere inspeccionar.
- **PROCESADO** Es el proceso de la imagen mediante las instrucciones aplicadas.
- **ACTUACIÓN** Sobre el entorno (aparato, objeto, elemento) en función del resultado obtenido.

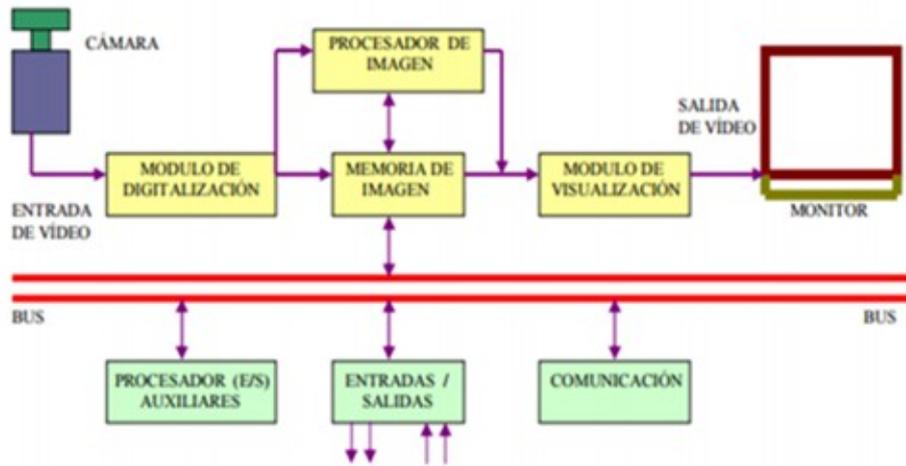


Figura 12. Diagrama de bloques, [6]

VII-F4. DIAGRAMA DE BLOQUES:

- **MEMORIA DE IMAGEN** Retiene la señal anteriormente del módulo de digitalización.
- **MÓDULO DE DIGITALIZACIÓN** Transforma la señal analógica captada por la cámara a una señal digital.
- **MÓDULO DE VISUALIZACIÓN** Cambia la señal digital de la memoria, en señal de video analógica para poder visualizarla en pantalla.
- **PROCESADOR DE IMAGEN** Soluciona e interpreta las imágenes captadas de la cámara.
- **MÓDULO DE ENTRADA Y SALIDAS** Administra la entrada de captación de imagen y las salidas de control que actúan dispositivos externos en función de resultado de la inspección.
- **COMUNICACIÓN** Vía I/O
- **CÁMARA** Para este rol es el dispositivo encargado de visualizar imagen o video a uno o varios objetos en particular, posteriormente procesándola y enviándola al software para que proceda a realizar el reconocimiento de la imagen.

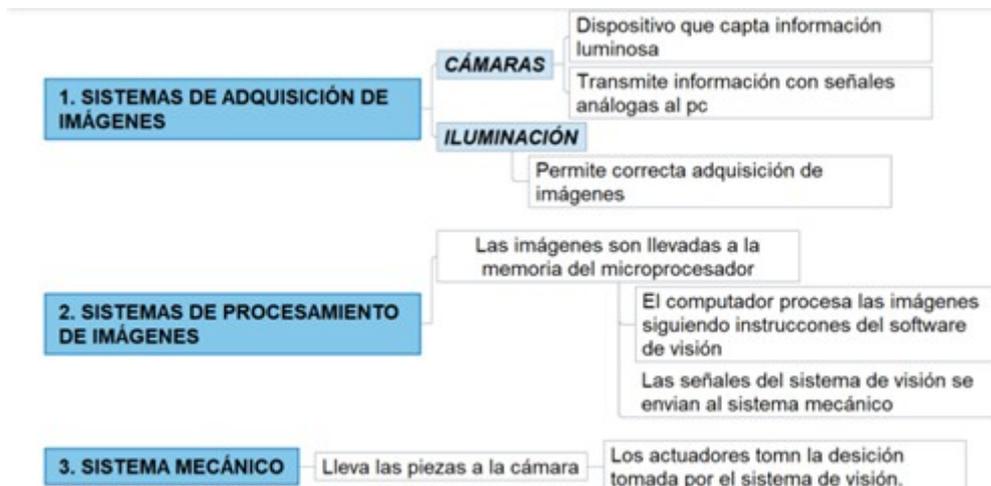


Figura 13. Diagrama de bloques adaptado de, [6]

VII-F5. ELEMENTOS DE VISIÓN ARTIFICIAL:

VII-G. PROTOCOLOS DE COMUNICACIÓN

Sirve con la finalidad de comunicar la visión artificial con los movimientos que empleara el robot, el autor [9], los presenta de la siguiente manera.

VII-G1. PROTOCOLO TCP: Es fundamental en Internet. Se trata de un protocolo seguro ya que dispone mecanismos que responden la integridad y la llegada en orden de los mensajes. Al tener estas garantías el protocolo utiliza comprobaciones adicionales y como consecuencia es menos eficiente que el UDP, aunque facilita la programación ya que no hay que tener en cuentas los problemas existentes en la red.

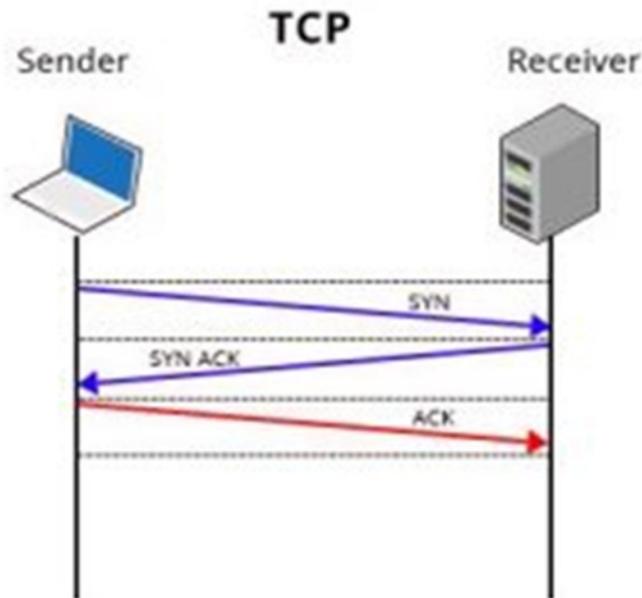


Figura 14. Protocolo TCP, [9]

VII-H. SISTEMAS EMBEBIDOS Y CONTROL DEL ROBOT

VII-H1. RASPBERRY-PI4: Se trata de un ordenador pequeño, que similar a una tarjeta de crédito y consta de muchos elementos. Consta con una placa base y sobre ésta es montado un procesador, una memoria RAM, y un chip gráfico.

Tiene un software de código abierto (Open Source), esto se puede modificar gracias a un programador, las raspberry pi tienen una memoria RAM, una GPU (unidad de procesamiento gráfico), procesador Broadcom, puertos USB, HDMI, Ethernet, 40 pines GP I/O (pines de entrada y salida), se pueden controlar por el usuario en tiempo de ejecución y un conector para cámara, carece del cable de alimentación y del disco duro para ello se reemplaza por una memoria USB.

[3].



Figura 15. Raspberry PI4

VII-I. SISTEMA DE INTERACCIÓN CON EL USUARIO

VII-II. SOFTWARE: PYTHON: Lenguaje de programación de alto nivel interpretado. La filosofía de diseño enfatiza la legibilidad del código con su notable uso. Su lenguaje, su enfoque orientado a objetos, tienen como objetivo apoyar a los programadores a escribir los códigos claros y lógicos para proyectos que sean a pequeña y gran escala.

Python se escribe dinámicamente. Admite diversos paradigmas de programación, incluyendo la estructurada (en particular, procedimental), orientada a objetos y funcional. Python se describe a menudo como un lenguaje debido a su amplia biblioteca estándar. [3].



Figura 16. Entorno del software Python

VII-12. *MATLAB*: Sistema interactivo cuyos componentes de datos es una matriz que no requiere dimensionamiento. Esto le permite resolver muchos problemas técnicos, problemas de computación, especialmente aquellos con formulaciones matriciales y vectoriales, disponiendo de librerías especializadas y de códigos básicos. [14]

VII-J. *SISTEMA DE CONTROL PARA BRAZO ROBOTICO*

Se han implementado y comparado cuatro tipos de controladores diferentes: un PID clásico, un PID con dinámica invertida, un controlador H_∞ y el mismo controlador H_∞ robustificado para considerar de forma expresa las incertidumbres de los datos dinámicos.

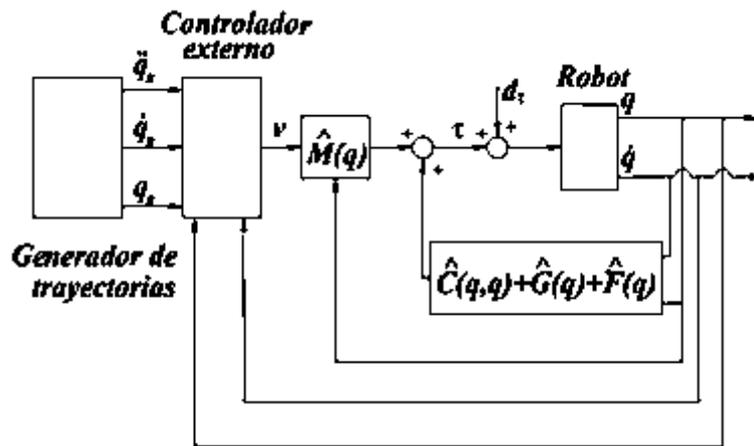


Figura 17. Esquema de bloque de los controladores PID con dinámica invertida, H_∞ , H_∞ robustificado, [7]

VII-11. *TEORÍA DE LOS CONTROLADORES*: Tanto el controlador PID como el controlador PID con dinámica invertida no son de nivel teórico, y su definición puede hallarse en cualquier libro básico de robótica. El controlador H_∞ desarrollado está basado en la teoría del control H_∞ no lineal.

VII-K. *CINEMÁTICA PARA UN BRAZO ROBÓTICO*

VII-K1. *MODELO CINEMÁTICO*: El Patrón cinemático del robot se basa en el análisis del movimiento con relación a un marco de referencia. Encontramos dos métodos fundamentales que son:

- Cinemática Directa.

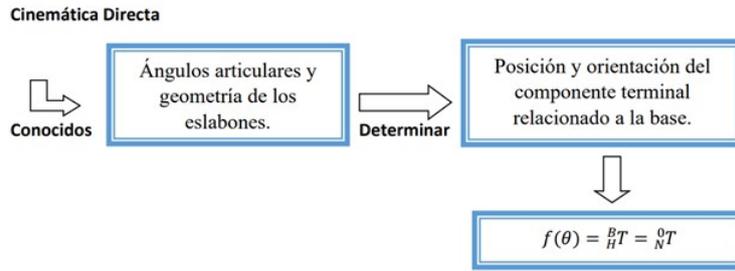


Figura 18. Esquema cinemática directa

■ Cinemática Inversa.

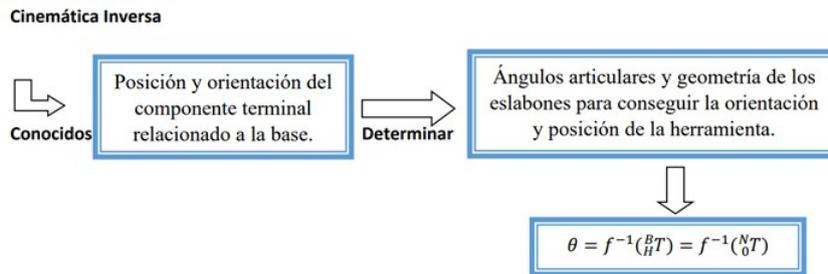


Figura 19. Esquema cinemática inversa

VII-K2. CINEMÁTICA DE BRAZOS ROBÓTICOS: La cinemática analiza la geometría del movimiento de los objetos físicos (la posición, velocidad, Aceleración) sin tomar en cuenta las fuerzas que lo provocan.

Se utiliza una matriz de transformación homogénea para representar la transformación de un vector de coordenadas homogéneas de un régimen de coordenadas a otro, con el método antes mencionado:

$$T = \begin{Bmatrix} R_{3 \times 3} & p_{3 \times 1} \\ f_{1 \times 3} & W_{3 \times 1} \end{Bmatrix} = \begin{Bmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escalado} \end{Bmatrix} \quad (1)$$

VIII. MARCO METODOLÓGICO

Considerando que el objetivo del estudio fue la automatización de un sistema identificador y posicionador de objetos por su color utilizando como recurso principal la visión artificial, recurrimos a un diseño experimental, mismo que nos ha permitido llevar a la práctica los conocimientos adquiridos dentro del periodo académico.

VIII-A. DIAGRAMA DE FLUJO DEL PROCESO

En la siguiente diagrama de proceso que observamos en la figura [20], observamos una línea de proceso que es repetitivo para la detección y clasificación de objetos.



Figura 20. Diagrama de Funcionamiento del proceso

VIII-B. DIAGRAMA DE PARTES ELÉCTRICAS CONECTADAS AL MICROCONTROLADOR

En este diagrama se muestra los componentes electrónicos que utilizaremos para nuestro proceso.

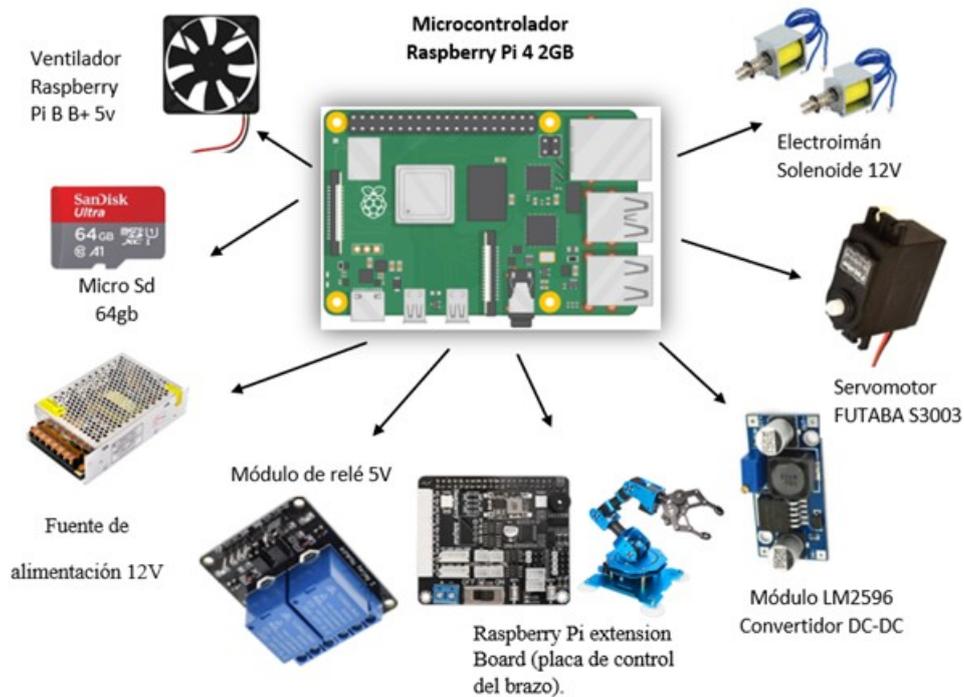


Figura 21. Elementos electrónicos

- **Electroimán:** Encargado de dar paso uno por uno a los objetos.
- **Servomotor Futaba S3003:** Cumple la función del movimiento de la banda mediante un tambor motriz.
- **MÓDULO LM2596:** Permite la modulación del voltaje de salida para nuestro Servomotor.
- **Placa de control del brazo:** Como su nombre lo indica, nos permite el control de los motores de nuestro brazo robótico.
- **Módulo relé de 5v:** Permite la activación de los electroimanes y del servomotor.
- **Fuente de alimentación 12v:** Sirve de alimentación para nuestros electroimanés, servomotor y ventilador.
- **Micro Sd de 64gb:** Sirve de almacenamiento de nuestra información que tengamos en la Raspberry.
- **Ventilador de Raspberry:** Evita el sobrecalentamiento de la Raspberry.

VIII-C. DISEÑO Y ELABORACIÓN DEL ESPACIO DE TRABAJO

El espacio de trabajo tiene una medida de (50 x 80 cm), dónde se colocará el sistema de llegada de los objetos, el brazo robótico, el sistema de detección de objetos y las deslizaderas para la clasificación por color de los objetos.

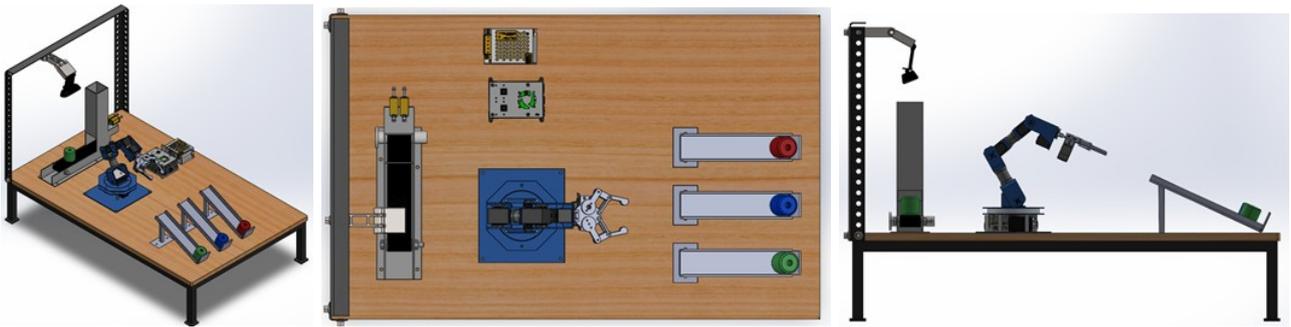


Figura 22. Espacio de trabajo

A continuación se detallará cada proceso, desde la recolección del objeto hasta su clasificación por color.

VIII-C1. SISTEMA DE LLEGA DEL OBJETO: Para poder ingresar el objeto creamos una estructura de acero A36, recubierta en pintura ploma, compuesta básicamente de un ducto vertical y un bastidor, al soltar de forma manual el objeto, se encuentra con un tope impulsado por dos cerraduras solenoides de 12v que controlan el paso del mismo.

■ ESTRUCTURA PRINCIPAL



Figura 23. Fabricación de estructura

■ CÁLCULO PARA SELECCIÓN DE BANDA TRANSPORTADORA

Para estos cálculos nos basamos con tablas impartidas en semestres previos, del tema transmisiones flexibles (Diseño Mecánico).

Propiedades de algunos materiales de las bandas planas y redondas. (Diámetro = d , espesor = t , ancho = w)

Material	Especificación	Tamaño, pulg	Diámetro mínimo de la polea, pulg	Tensión permisible por ancho unitario a 600 pies/min, lbf/pulg	Peso específico, lbf/pulg ³	Coefficiente de fricción	
Cuero	1 capa	$t = \frac{11}{64}$	3	30	0.035-0.045	0.4	
		$t = \frac{13}{64}$	$3\frac{1}{2}$	33	0.035-0.045	0.4	
	2 capas	$t = \frac{18}{64}$	$4\frac{1}{2}$	41	0.035-0.045	0.4	
		$t = \frac{20}{64}$	6 ^a	50	0.035-0.045	0.4	
		$t = \frac{23}{64}$	9 ^a	60	0.035-0.045	0.4	
Poliamida ^b	F-0 ^c	$t = 0.03$	0.60	10	0.035	0.5	
	F-1 ^c	$t = 0.05$	1.0	35	0.035	0.5	
	F-2 ^c	$t = 0.07$	2.4	60	0.051	0.5	
	A-2 ^c	$t = 0.11$	2.4	60	0.037	0.8	
	A-3 ^c	$t = 0.13$	4.3	100	0.042	0.8	
	A-4 ^c	$t = 0.20$	9.5	175	0.039	0.8	
	A-5 ^c	$t = 0.25$	13.5	275	0.039	0.8	
Uretano ^d	$w = 0.50$	$t = 0.062$	Vea la tabla	5.2 ^e	0.038-0.045	0.7	
	$w = 0.75$	$t = 0.078$	17-3	9.8 ^e	0.038-0.045	0.7	
	$w = 1.25$	$t = 0.090$		18.9 ^e	0.038-0.045	0.7	
	Redonda	$d = \frac{1}{4}$		Vea la tabla	8.3 ^e	0.038-0.045	0.7
		$d = \frac{3}{4}$		17-3	18.6 ^e	0.038-0.045	0.7
		$d = \frac{1}{2}$			33.0 ^e	0.038-0.045	0.7
		$d = \frac{3}{4}$			74.3 ^e	0.038-0.045	0.7

Figura 25. Propiedades de materiales para bandas planas y redonda

Se selecciono Poliamida (P) como material para la fabricación de la banda, dado que se tiene un eje con diametro $d = 2[\text{cm}] = 0.79 [\text{pulg}]$, basandonos en la tabla anterior (25), visualizamos las especificaciones de $(F - 0^c)$ $d_m = 0.6[\text{pulg}]$, y en $(F - 1^c)$ $d_m = 1[\text{pulg}]$, por lo tanto para obtener lo requerido para la banda tenemos que interpolar entre éstas dos especificaciones para llegar a los valores deseados.

Procedimiento

Datos		
Especificación	Tamaño, pulg	Diámetro mínimo de polea
$(F - 0^c)$	$t = 0.03$	0.60
$(F - 1^c)$	$t = 0.05$	1.0

Interpolación

$$\begin{array}{l|l} x_1 = 0,60 & y_1 = 0,03 \\ x = 0,79 & y \\ x_2 = 1 & y_2 = 0,05 \end{array}$$

$$y = y_1 + \frac{y_2 - Y_1}{x_2 - x_1}(x - x_1) \tag{2}$$

$$y = 0,03 + \frac{0,05-0,03}{1-0,60}(0,79 - 0,60)$$

$$y = 0,040 = t$$

Para 0.79 [pulg], tenemos que el espesor de la banda a utilizar es $t=0.04[\text{pul}]$

Tensión permisible

$$\begin{array}{l|l} x_1 = 0,60 & y_1 = 10 \\ x = 0,79 & y \\ x_2 = 1 & y_2 = 35 \end{array}$$

$$y = 10 + \frac{35-10}{1-0,60}(0,79 - 0,60)$$
$$y = 21,875[\text{lb}/\text{pulg}]$$

Tensión permisible de 21.87 [lb/pulg] o 3.75 [kg/cm]. Peso específico 0,035[lb/pulg³] y coeficiente de fricción 0.5 no interpolamos debido a ser el mismo valor.

Una vez obtenido las propiedades de la banda en este caso plana horizontal se la puede fabricar con los datos obtenidos o con los datos de las especificaciones de ($F - 1^c$) ya que es preferible elegir el mayor valor.

Tipo de banda Banda plana de Poliamida ($F - 1^c$) con $d=0.79$ pulg, espesor $t=0.05$ pulg, peso específico 0,035[lb/pulg³] con coeficiente de fricción 0.5 y un ancho de $3,8\text{cm} = 1,5\text{pulg}$

■ EJES O TAMBORES

Son fabricados de material Duralon para cumplir la función de tambor motriz y tambor terminal.



Figura 26. Tambor motriz y tambor terminal

■ Cálculo de velocidad y Potencia del servomotor

Torque (6.0v) = 56.8 Oz/in (4.1kg.cm)

Velocidad rotacional para (6v) = 54 [rpm]

Velocidad

$$v = \frac{\pi * d * n}{12} \quad (3)$$

$$v = \frac{\pi(0,79)(54)}{12}$$

$$v = 11,16[\text{pie}/\text{min}] = 0,05[\text{m}/\text{s}]$$

Potencia

$$P = F * V \quad (4)$$

$$w = m * g \quad (5)$$

$$m = 18g$$

$$w = m * g = 8,16kg(9,8\text{m}/\text{s}^2) = 80N$$

$$P = 80N(0,05\text{m}/\text{s})$$

$$P = 4Nm/\text{s}$$

$$P = 4\text{segundosdewatt}$$

VIII-C2. *SISTEMA DE DETECCIÓN*: Para la detección con el metodo por colores, se seleccionó una cámara de 720mpx, y para poder posicionarla, se fabricó una base para poder mover la cámara y calibrarla con el fin de garantizar que el rango de la cámara cubra por completo la banda transportadora y tenga mejor detección.

La camara fue directamente conectada a la raspberry para poder observar, analizar y seleccionar el objeto de forma adecuada.

- ESTRUCTURA PARA LA CÁMARA

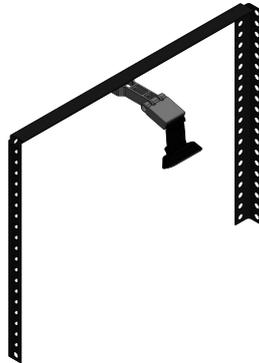


Figura 27. Estructura 3D del soporte de para la cámara

- CODIGO PARA DETECCIÓN

Para la detección de objetos se utilizaros librerias y variables específicas de Python para este método de detección por colores.

- Librerías

```

1 #####
2 ###Proyecto de titulación#####
3 #####Anthony Llerena.B#####
4 #####Marc Salazar.V#####
5 #####Tutor:Ing.David Cortez#####
6 #librerias
7 import cv2
8 import numpy as np
9 import time
10 import imutils
11 import tkinter as tk
12 import tkinter.messagebox
13 from PIL import Image, ImageTk
14 import serial
15 import serial.tools.list_ports
16 from math import *
17 #import xarm
18 import RPi.GPIO as GPIO
19 from time import sleep
20 from servo import *
21 path = '/home/pi/Desktop/LOGOUPS2.png'
22 path1 = '/home/pi/Desktop/MAQUETA.png'

```

Figura 28. Importación de librerias en Python

- Configuración de cámara

Se configura los parámetros de la cámara, el ancho y alto de la imagen .

```

50 camHeight = 480
51 camWidth = 640
52 nuevo_Contorno3=0
53 cam = cv2.VideoCapture(0)
54 cam.set(3, camWidth)
55 cam.set(4, camHeight)

```

Figura 29. configuración de cámara

- RANGO PARA VALORES SHV DE CADA COLOR

```

56 #Calibración de colores:
57 sensH=36
58 sensS=57
59 sensV=29
60 sensrojoH=15
61 sensrojoS=20
62 sensrojoV=30
63
64 val_azulH=107
65 val_azulS=169
66 val_azulV=219
67 val_verdeH=80
68 val_verdeS=159
69 val_verdeV=120
70 val_rojoH=178
71 val_rojoS=153
72 val_rojoV=165
73 azulBajo = np.array([val_azulH-sensH,val_azulS-sensS,val_azulV-sensV],np.uint8)
74 azulAlto = np.array([val_azulH+sensH,val_azulS+sensS,val_azulV+sensV],np.uint8)
75 verdeBajo = np.array([val_verdeH-sensH,val_verdeS-sensS,val_verdeV-sensV],np.uint8)
76 verdeAlto = np.array([val_verdeH+sensH,val_verdeS+sensS,val_verdeV+sensV],np.uint8)
77 rojoBajo = np.array([val_rojoH-sensrojoH,val_rojoS-sensrojoS,val_rojoV-sensrojoV],np.uint8)
78 rojoAlto = np.array([val_rojoH+sensrojoH,val_rojoS+sensrojoS,val_rojoV+sensrojoV],np.uint8)
79 redBajo1 = np.array([0,100,20],np.uint8)
80 redAlto1 = np.array([5,255,255],np.uint8)
81 redBajo2 = np.array([val_rojoH-sensrojoH,val_rojoS-sensrojoS,val_rojoV-sensrojoV],np.uint8)
82 redAlto2 = np.array([val_rojoH+sensrojoH,val_rojoS+sensrojoS,val_rojoV+sensrojoV],np.uint8)

```

Figura 30. Valores HSV de colores

- CONTORNO, UBICACIÓN EN EL PLANO x,y RESPECTO A LA CÁMARA

```

168 def dibujar_rojo(mask_rojo,color_rojo):
169     global area_rojo, nuevo_Contorno1,x1,y1,color_rojoa, c1, img
170
171     contornos1,hierachy = cv2.findContours(mask_rojo, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
172     for c1 in contornos1:
173         area_rojo = cv2.contourArea(c1)
174         if area_rojo > 1000:
175             a="rojo"
176             #print(a)
177             M = cv2.moments(c1)
178             if (M["m00"]==0): M["m00"]=1
179             x1 = int(M["m10"]/M["m00"])
180             y1 = int(M["m01"]/M["m00"])
181             rojo="rojo"
182             nuevoContorno1 = cv2.convexHull(c1)
183             cv2.circle(img, (x1,y1),30,(0,0,255),-1)#blue,green,red
184             cv2.putText(img, '{}.{}.{}'.format(x1,y1,rojo),(x1+10,y1), font, 0.75,(255,255,255),1,cv2.LINE_AA)
185             cv2.drawContours(img, [nuevoContorno1], 0, color_rojo, 3)
186             color_rojoa=color_rojo

```

Figura 31. Contorno, Área, posición

VIII-C3. SISTEMA DE RECOLECCIÓN Y POSICIONADOR:

■ BRAZO ROBÓTICO

Se procedió a realizar el montaje de la estructura del brazo xARM 1S

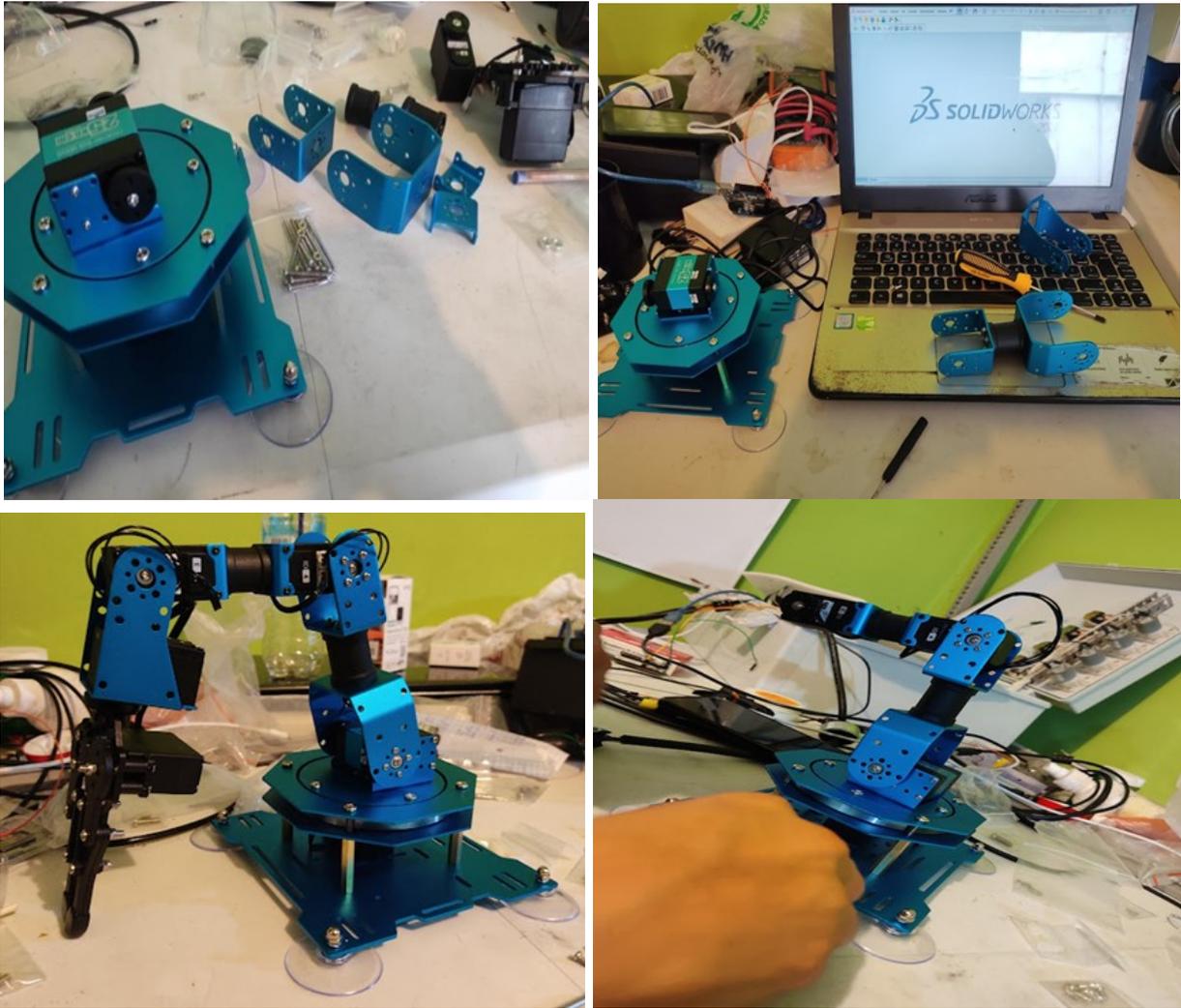


Figura 32. Montaje de estructura del brazo

■ DISEÑO 3D DEL BRAZO

Para nuestro diseño en 3D se tomaron las medidas correctas pieza por pieza del robot previamente desarmado.

Luego se realizó los subsensamblés, para así poder tener mejor maniobrabilidad al momento de hacer el ensamble total.

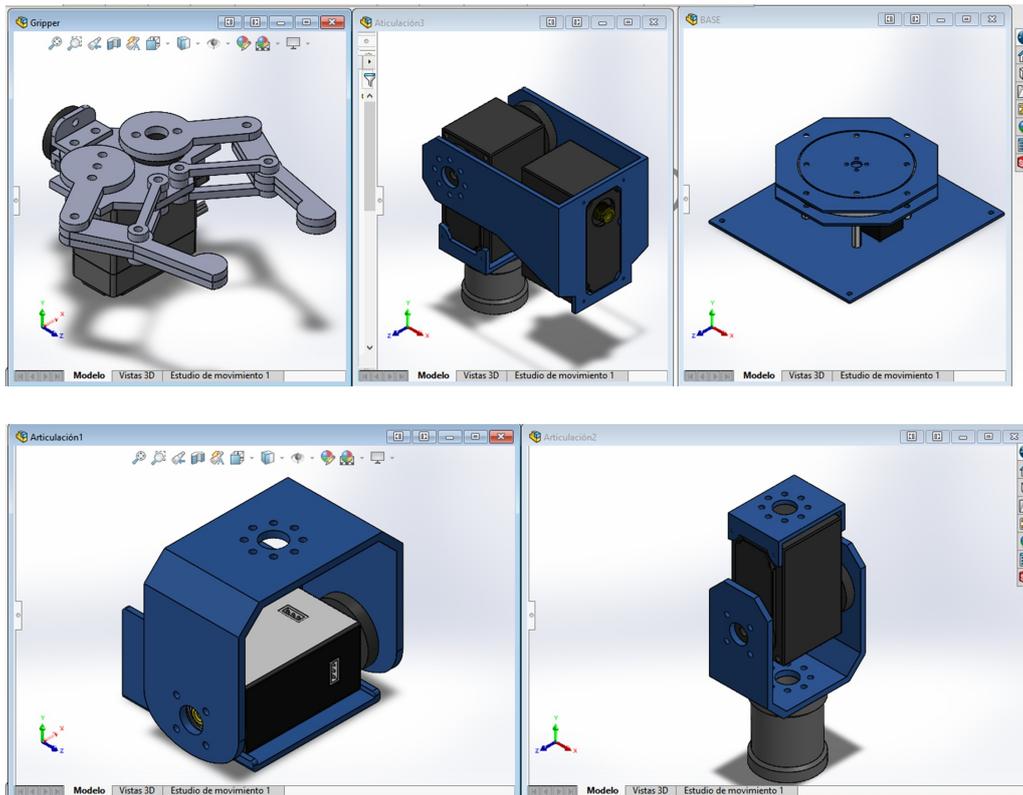


Figura 33. Subsensamblés de las partes del brazo

Se procedió a realizar el ensamble total de nuestro brazo.

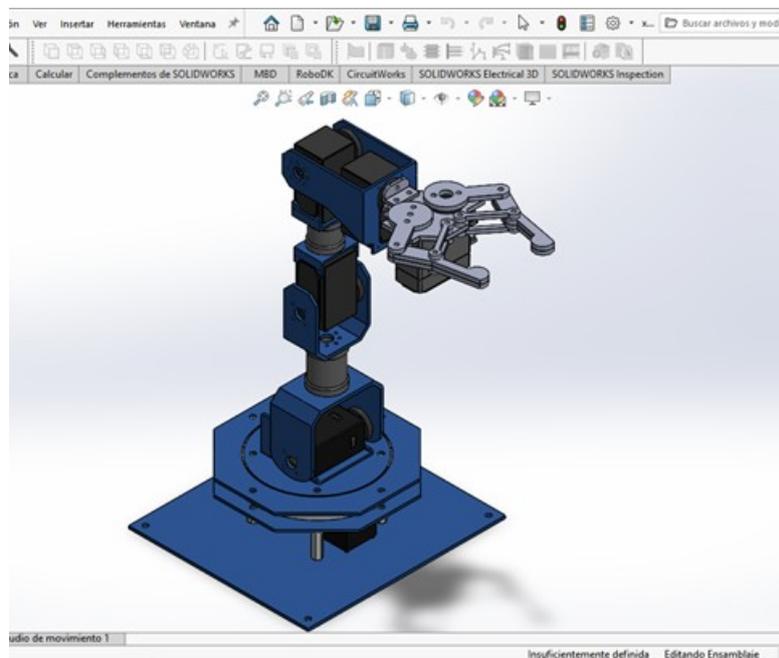


Figura 34. Diseño 3D del brazo xARM1S

■ DIMENSIONES DEL BRAZO xARM 1S

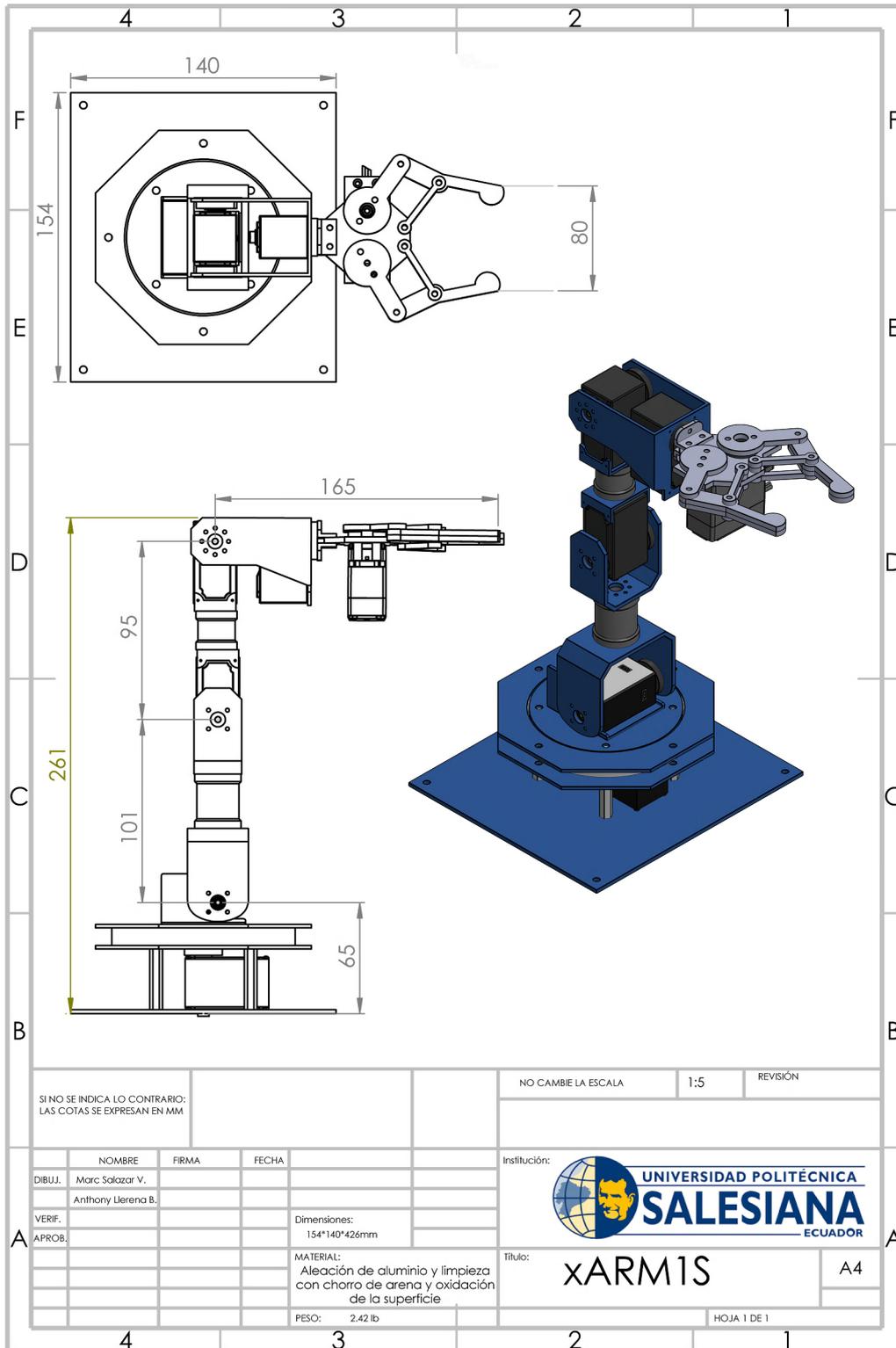


Figura 35. Dimensiones del brazo robótico

- CÓDIGOS PARA MOVIMIENTOS DEL BRAZO

- MOVIMIENTOS POR ÁNGULOS SETEADOS MANUALMENTE

Este método está enfocado a una de las prácticas que se van a realizar para movimientos de motores.

Para la codificación ingresamos los valores en ángulos de cada motor con ayuda de la interfaz que veremos en la siguiente imagen[36] ,y así poder codificar dichos valores para cada motor, como se muestra en la imagen [37].

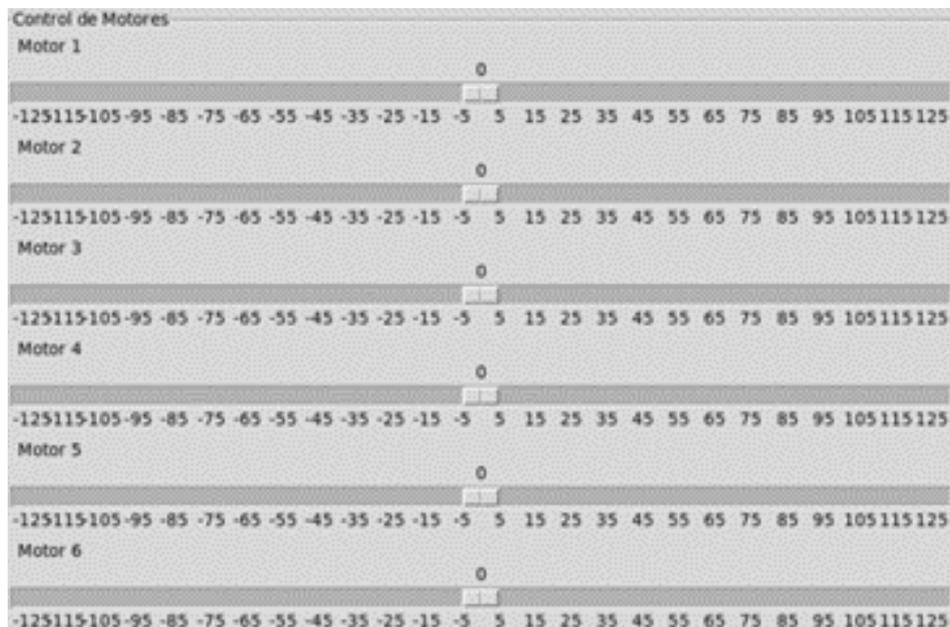


Figura 36. Sliders para selección o posición de motores

```
def ArmPosition_home():
    global arm

    arm.setPosition(2, 0.0, True)
    arm.setPosition(3, 0.0, True)
    arm.setPosition(4, 0.0, True)
    arm.setPosition(5, 0.0, True)
    arm.setPosition(6, 0.0, True)
```

Figura 37. Posicionamiento (Posición Home del brazo)

```
409 def ArmPinzaAbierta():
410     global arm
411     arm.setPosition(1, -64.0, True)
412
413 def ArmPinzaCerrada():
414     global arm
415     arm.setPosition(1, 64.0, True)
```

Figura 38. Estado de pinza posición abierta/cerrada

A continuación observaremos la secuencia a seguir para la recolección y clasificación del objeto color rojo, hacemos lo mismo para los otros colores.

```
if x1 >390:  
    print("Llegada Rojo")  
    GPIO21_State = True  
    GPIO.output(GPIO21, True)  
    ArmPosition_home()  
    gripper(True)  
    time.sleep(2)  
    ArmPosition_banda_intermedia()  
    time.sleep(2)  
    ArmPosition_banda()  
    time.sleep(2)  
    gripper(False)  
    time.sleep(2)  
    ArmPosition_banda_intermedia()  
    time.sleep(2)  
    ArmPosition_home()  
    time.sleep(2)
```

Figura 39. Secuencia de movimiento

Ilustración de la secuencia a seguir con el color rojo.

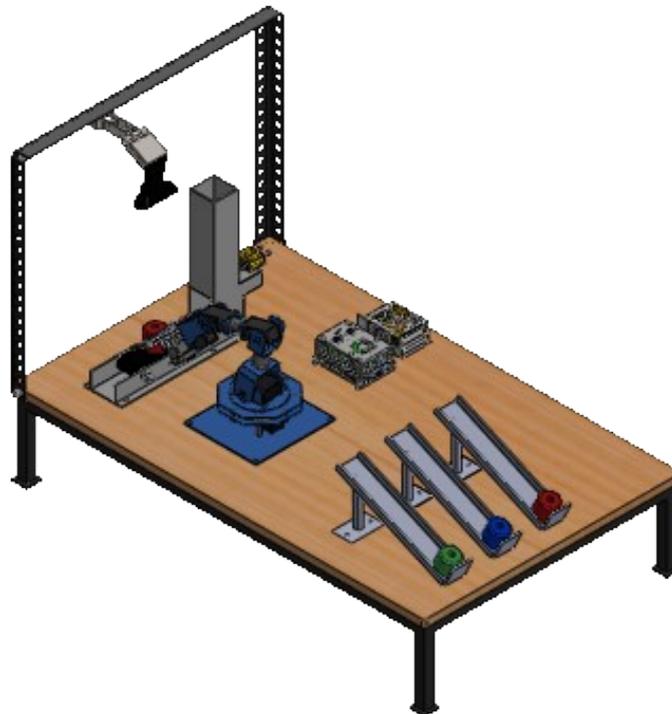


Figura 40. Recolección de objeto

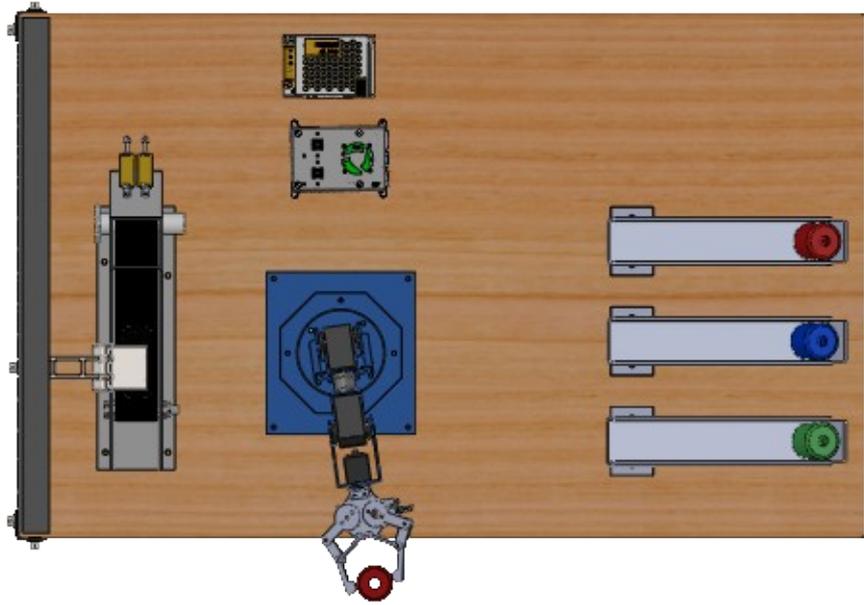


Figura 41. Traslado de objeto

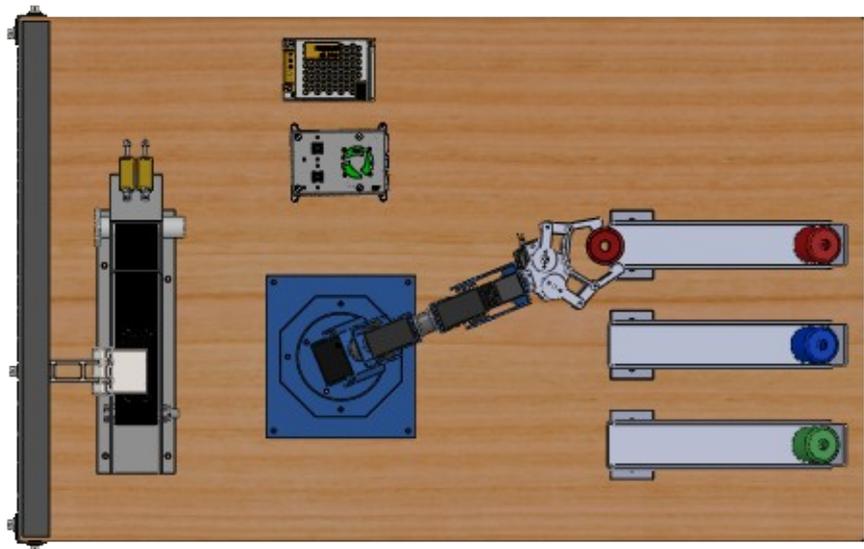


Figura 42. Clasificación de objeto

- CÓDIGOS PARA MOVIMIENTOS DEL BRAZO APLICANDO CINEMÁTICAS

Para este método utilizamos el software Matlab para así poder sacar las matrices de rotación y traslación con cinemática directa e inversa y luego obtener los movimientos en python de nuestro brazo robótico.

- CÁLCULOS DE MATRICES ROTACIONALES, DE TRASLACIÓN, HOMOGENEAS, JACOBIANAS

Procedemos a declarar las variables (q_i) que son las articulaciones de nuestro brazo, una vez obtenidas se procederá a realizar las matrices de rotación y traslación con fórmulas ya definidas.

```

1 - clear;
2 - clc;
3 - syms q1 q2 q3 q4 L1 L2 L3 L4
4   % L1=31;%cm
5   % L2=97.61;%cm
6   % L3=95.18;%cm
7   % L4=55.84+29.80+47.20;%cm
8
9   %Rotacion en Z , q1
10 - H1=[ cos(q1) -sin(q1) 0 0;
11        sin(q1) cos(q1) 0 0;
12        0 0 1 0;
13        0 0 0 1];
14   %traslación en Z , L1
15 - H2=[1 0 0 0;
16        0 1 0 0;
17        0 0 1 L1;
18        0 0 0 1];

```

Figura 43. Matrices rotacionales y traslacionales

Una vez todas las matrices adquiridas, procedemos a realizar el producto de todas ellas para obtener nuestra matriz homogénea.

Luego procedemos a derivar nuestra matriz respecto a los ejes x,y,z para obtener nuestra matriz Jacobiana.

```

54 %% Producto de matrices homogeneas
55 H=H1*H2*H3*H4*H5*H6*H7*H8
56
57 x=H(1,4)
58 y=H(2,4)
59 z=H(3,4)
60 %% Jacobiano X,Y,Z
61 J11 = diff (x,q1);
62 J12 = diff (x,q2);
63 J13 = diff (x,q3);
64 J14 = diff (x,q4);
65 %%
66 J21 = diff (y,q1);
67 J22 = diff (y,q2);
68 J23 = diff (y,q3);
69 J24 = diff (y,q4);
70 %%
71 J31 = diff (z,q1);
72 J32 = diff (z,q2);
73 J33 = diff (z,q3);
74 J34 = diff (z,q4);
75 %% MATRIZ JACOBIANA
76 J=[J11 J12 J13 J14;
77     J21 J22 J23 J24;
78     J31 J32 J33 J34]
79

```

Figura 44. Matrices para la obtención del Jacobiano

Después de obtener los puntos x,y,z y la matriz jacobiana, declaramos una función donde evaluamos la cinemática directa, obtenemos la PseudoInversa del jacobiano.

```

2  function Y =fcn control cinematico (t,u)
3  - dq = zeros(4,1); %3 Grados de libertad
4  %% Funcion de control Cinemático
5  %% dq=JT*(Xd_d+K*(Xd-X));
6  % dq =Velocidad Angular de cada articulación
7  % JT =Pseudo Inversa del jacobiano
8  % Xd_d =Velocidad cartesiana deseada(Trayectoria)
9  % K =Matriz de Ganancia del controlador
10 % Xd =Pocisión Cartesiana deseada (Trayectoria)
11 % X =Pocisión Cartesiana Real
12 %% Paso 1
13 %Declarar las dimensiones del Robot
14 - L1=31;%cm
15 - L2=97.61;%cm
16 - L3=95.18;%cm
17 - L4=55.84 + 29.80 + 47.2;%cm
18
19 %% Paso 2
20 % Entradas de la función
21 - q1=u(1);
22 - q2=u(2);
23 - q3=u(3);
24 - q4=u(4);
25 - ti=t; %tiempo

```

Figura 45. Declaracion de función y dimensiones del robot

```

26 %% Paso 3
27 % Evaluar cinemática directa
28
29 x =L2*cos(q1)*cos(q2 + pi/2) - L4*(cos(q4)*(cos(q1)*sin(q3)*sin(q2 + pi/2)
30 y =L2*cos(q2 + pi/2)*sin(q1) - L4*(cos(q4)*(sin(q1)*sin(q3)*sin(q2 + pi/2)
31 z =L1 + L3*(cos(q3)*sin(q2 + pi/2) + cos(q2 + pi/2)*sin(q3)) + L2*sin(q2 +
32 X=[x;y;z];
33 %% Paso 4
34 %Declarar el Jacobiano calculado previamente(revisar scrip de jacobiano)
35
36 J =[]
37 L3*(sin(q1)*sin(q3)*sin(q2 + pi/2) - cos(q3)*cos(q2 + pi/2)*sin(q1)) + L4
38 L2*cos(q1)*cos(q2 + pi/2) - L4*(cos(q4)*(cos(q1)*sin(q3)*sin(q2 + pi/2) -
39
40
41
42 %% Paso 5
43 x =L3*(cos(q1 + pi/2)*sin(q3)*sin(q2 + pi/2) + cos(q3)*cos(q1 + pi/2)*cos(
44
45
46 y =L3*(cos(q3)*cos(q2 + pi/2)*sin(q1 + pi/2) + sin(q3)*sin(q1 + pi/2)*sin(
47
48
49 z =L1 - L3*(cos(q3)*sin(q2 + pi/2) - cos(q2 + pi/2)*sin(q3)) - L2*sin(q2 +
50
51
52 J =[] - L3*(cos(q3)*cos(q2 + pi/2)*sin(q1 + pi/2) + sin(q3)*sin(q1 + pi/2)*

```

Figura 46. Evaluación de puntos en los ejes x,y,z y Jacobiano

Para la simulación se realiza la definición de trayectorias, velocidades, Definimos la matriz de ganancia del controlador P, cálculos de error y salida.

```

63      %% Calcular la pseudoInversa del jacobiano
64 -    JT=J'*inv(J*J')
65      %% Paso 6 SIMULACIÓN
66      %%Definición de la trayectoria en el espacio cartesiano
67      %%Para este caso una linea recta en el plano z=10;
68 -    w=2;%frecuencia
69 -    xd=100;
70 -    yd=100;
71 -    zd=100;
72 -    Xd=[xd; yd; zd]; %pocision deseada
73
74 -    xd_d=0;
75 -    yd_d=0;
76 -    zd_d=0;
77 -    Xd_d=[xd_d;yd_d;zd_d]; %velocidad deseada
78      %% Paso 7
79      %%Definición de la matriz de ganancia del controlador
80 -    K=[5  0  0;...
81        0  5  0;...
82        0  0  5];
83      %% Paso 8
84      %%Cálculo de la función dq y declaración de salida
85 -    error=Xd-X;
86 -    salida=[X ; Xd]
87 -    dq=JT*(Xd_d+K*(error))
88 -    Y= [dq];

```

Figura 47. Cálculo de psudoinversa, definicion del controlador, cálculo de error y salida de la función

Al obtener la cinemática inversa procedemos a realizar la solución de la trayectoria, resolviendo la ecuación diferencial integrando los valores, separamos los valores de (qi) y para finalizar mostramos la ventana de la trayectoria deseada.

```

1      %Solucion de la trayectoria
2 -    clc;clear;
3 -    t0=0; tf=1;
4 -    h=0.02;
5      %%condiciones iniciales
6 -    q0=[0,0,.1,.1];
7      %%resolvemos la ecuacion diferencial
8 -    [t,q]=ode45(@control_cinematico,[t0:h:tf],q0);
9 -    SALIDA=q*180/pi
10 -    TIEMPO=t
11     %%separamos los elementos de la matriz " q " obtenida
12 -    q1=q(:,1)*180/pi;
13 -    q2=q(:,2)*180/pi;
14 -    q3=q(:,3)*180/pi;
15 -    q4=q(:,4)*180/pi;
16 -    hold on
17 -    axis([0 360 0 360 0 360])
18 -    xlabel('x')
19 -    ylabel('Y')
20 -    zlabel('z')

```

Figura 48. Condiciones iniciales, ecuación diferencial

```

21 %Graficamos la trayectoria obtenida
22 %Es decir la trayectoria que realiza el efector final
23 %Medidas de brazo
24 - L1=31;%mm
25 - L2=97.61;%mm
26 - L3=95.18;%mm
27 - L4=55.84+29.80+47.20;%mm
28 %-----Cinemática Directa de todos los eslabones del robot
29 - x4=L2*cos(q(:,1)).*cos(q(:,2) + pi/2) - L4*(cos(q(:,4)).*(cos(q(:,1)).*sin(q(
30 sin(q(:,4)).*(cos(q(:,1)).*cos(q(:,3)).*sin(q(:,2) + pi/2) + cos(q(:,1)).
31 cos(q(:,1)).*cos(q(:,3)).*cos(q(:,2) + pi/2)));
32 - y4=L2*cos(q(:,2) + pi/2).*sin(q(:,1)) - L4*(cos(q(:,4)).*(sin(q(:,1)).*sin(q(
33 sin(q(:,4)).*(cos(q(:,3)).*sin(q(:,1)).*sin(q(:,2) + pi/2) + cos(q(:,2) +
34 cos(q(:,3)).*cos(q(:,2) + pi/2).*sin(q(:,1)));
35 - z4=L1 + L3*(cos(q(:,3)).*sin(q(:,2) + pi/2) + cos(q(:,2) + pi/2).*sin(q(:,3))
36 pi/2) + cos(q(:,2) + pi/2).*sin(q(:,3))) + sin(q(:,4)).*(cos(q(:,3)).*cos(
37 %plot3(x4,y4,z4)
38 %-----

```

Figura 49. Trayectori eslabón x4,y4,z4

```

38 %-----
39 - x3=L2*cos(q(:,1)).*cos(q(:,2) + pi/2) - L3*(cos(q(:,1)).*sin(q(:,3)).*sin(q(
40 - y3=L2*cos(q(:,2) + pi/2).*sin(q(:,1)) - L3*(sin(q(:,1)).*sin(q(:,3)).*sin(q(
41 - z3=L1 + L3*(cos(q(:,3)).*sin(q(:,2) + pi/2) + cos(q(:,2) + pi/2).*sin(q(:,3))
42 %plot3(x3,y3,z3)
43 %-----
44 - x2=L2*cos(q(:,1)).*cos(q(:,2) + pi/2);
45 - y2=L2*cos(q(:,2) + pi/2).*sin(q(:,1));
46 - z2=L1 + L2*sin(q(:,2) + pi/2);
47 %plot3(x2,y2,z2)
48 %-----
49 - x1=0;
50 - y1=0;
51 - z1=L1;
52 %-----
53 - x0=0;
54 - y0=0;
55 - z0=0;
56 %Grafico del Robot
57 - i=0;

```

Figura 50. Trayectoria eslabones (x3,y3,z3)- (x2,y2,z2)-(x1,y1,z1)

```
59
60 - for tiempo = t0:h:tf
61 -     i=i+1;
62 -     plot3([0;0],[0;0],[0;z1],'g')%Del origen al eslabon 1
63 -     plot3([x1;x2(i)],[y1;y2(i)],[z1;z2(i)],'r')%eslabon 2
64 -     plot3([x2(i);x3(i)],[y2(i);y3(i)],[z2(i);z3(i)],'b')%eslabon 3
65 -     plot3([x3(i);x4(i)],[y3(i);y4(i)],[z3(i);z4(i)],'m')%eslabon 3
66
67 -     pause(.4)
68
69
70 - end
71
72
73
74
75 - q=(q*180/pi);
76
```

Figura 51. Caption

o MOVIMIENTO DE MOTORES PYTHON

Declaramos posiciones del origen de nuestro brazo, tambien para las trayectorias X,Y,Z en cada punto (obtenidas con el analisis cinemático).

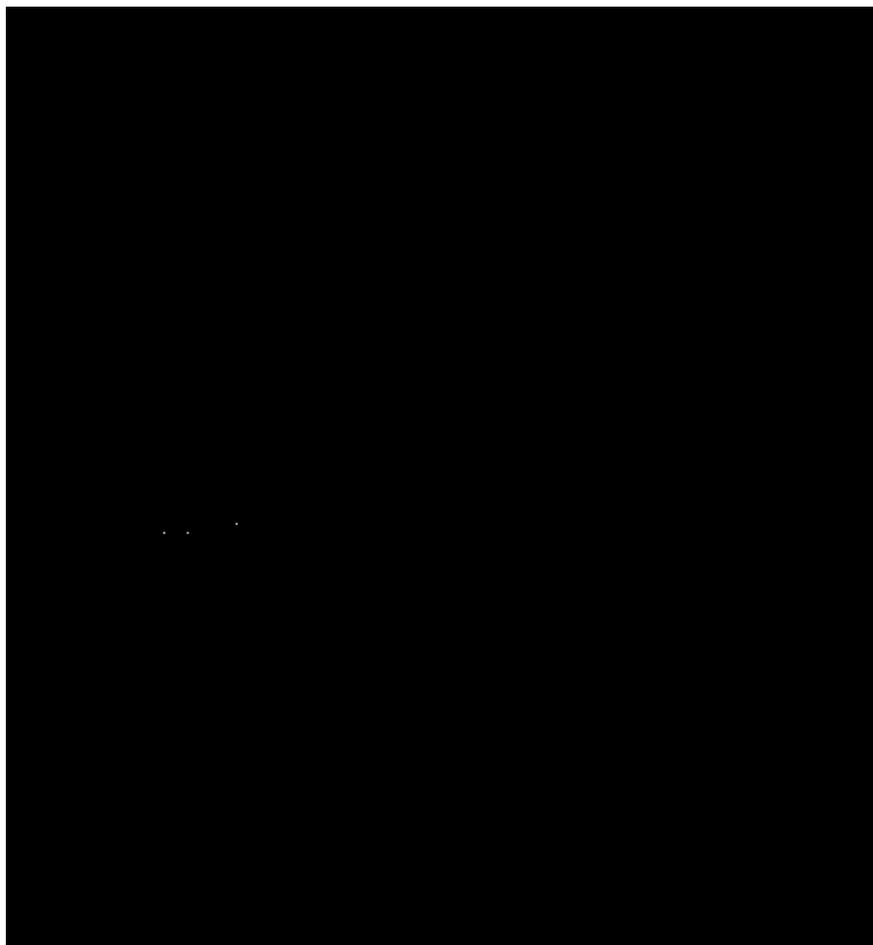


Figura 52. Caption

Luego seteamos los puntos y determinamos dos tipos de trayectoria, uno entre el origen y la cinta y el otro tipo de trayectoria entre el origen y los puntos de clasificación.



Figura 53. Trayectoria entre el origen y banda transportadora



Figura 54. Trayectoria entre el origen y los puntos de clasificación

○ SIMULACION DE MOVIMIENTO DE LAS ARTICULACIONES DEL BRAZO

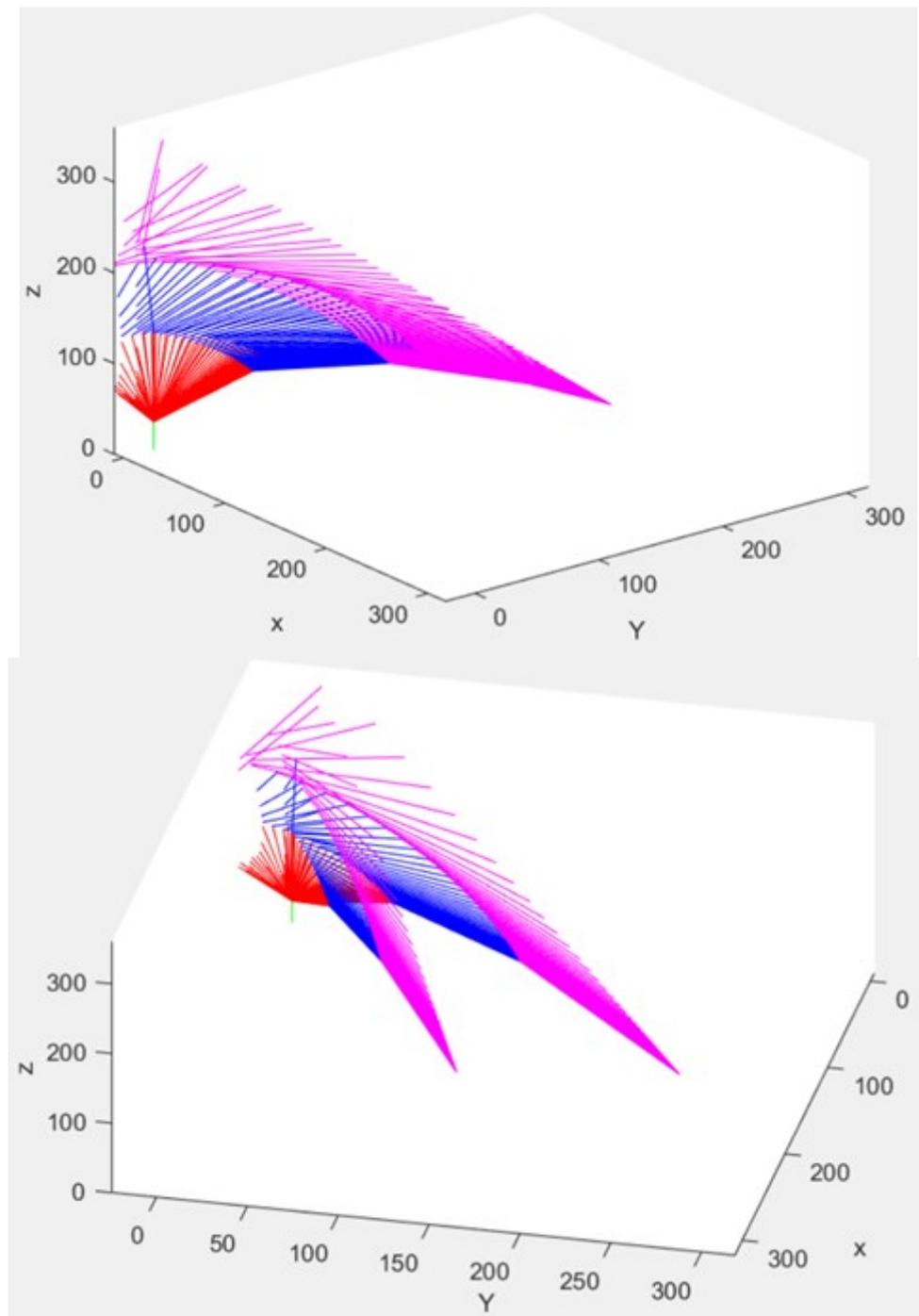


Figura 55. Simulación de movimiento del brazo

IX. RESULTADOS

De acuerdo con las investigaciones, se dió a conocer un manejo eficiente referente al estándar de visualización, mediante la ejecución del sistema propuesto, lo cual se analizó los siguientes resultados obtenidos al momento de realizar el prototipo del sistema.

IX-A. Microcontrolador

Se alcanzó un resultado positivo, al seleccionar el microcontrolador raspberry pi 4 model B con sistema operativo Raspbian, empleando técnicas de procesamiento con la librería de programación OpenCv, se desarrolló la aplicación de visión artificial con lenguaje de programación Python.

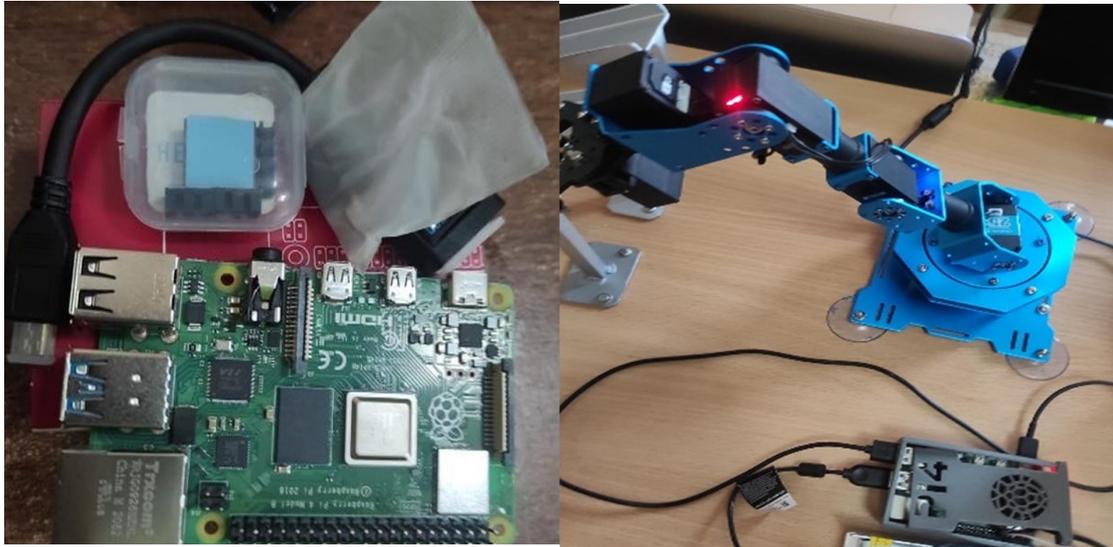


Figura 56. Microcontrolador seleccionado para el control del Sistema.

IX-B. Brazo robótico xARMIS

■ IMPORTACIÓN DE LIBRERÍA

Cuenta con una librería (**import xarm**) ejecutable en python, facilitando el control del movimiento de los servos de bus inteligentes.

```
Detección de objetos.py ✕
1 #####
2 ###Proyecto de titulación#####
3 #####Anthony Llerena.B#####
4 #####Marc Salazar.V#####
5 #####Tutor:Ing.David Cortez#####
6 #librerias
7 import cv2
8 import numpy as np
9 import time
10 import imutils
11 import tkinter as tk
12 import tkinter.messagebox
13 from PIL import Image, ImageTk
14 import serial
15 import serial.tools.list_ports
16 from math import *
17 #import xarm
```



Figura 57. Importación de librería xArm y servos de bus inteligente

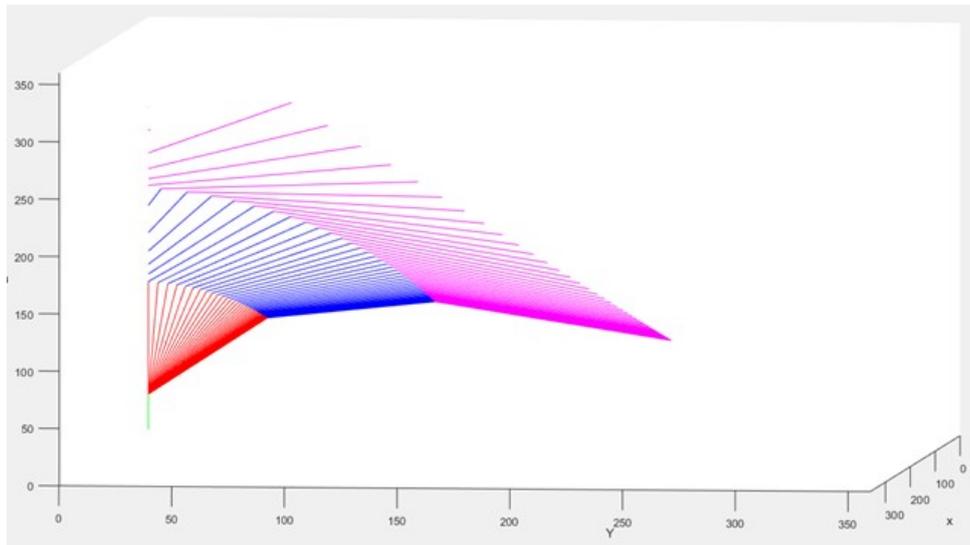


Figura 58. Simulación de posicionamiento

■ **SIMULACIÓN DE ARTICULACIONES EN MATLAB**

La simulación del posicionamiento del brazo robótico se realizó en MATLAB, mediante cálculos cinemáticos.

- **DISEÑO Y ENSAMBLE EN SOLIDWORKS** Se obtuvo como resultado el diseño de cada parte de nuestro brazo y por último el ensamble completo.

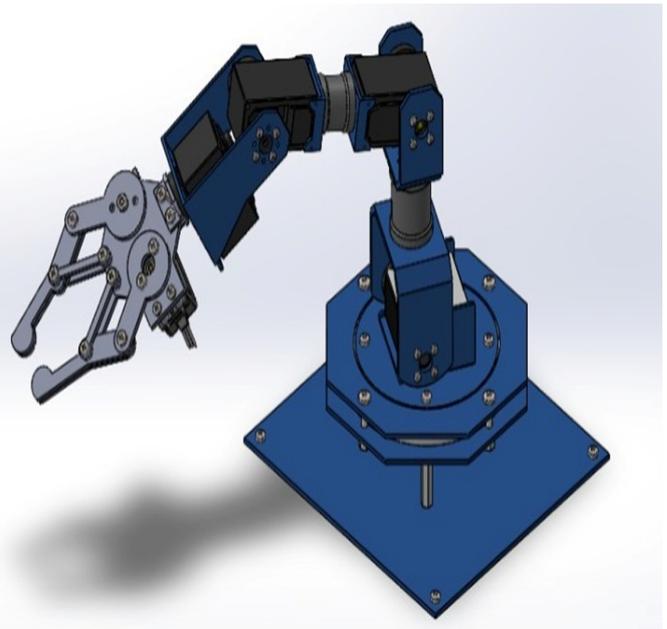


Figura 59. Ensamble y diseño 3D

IX-C. Mesa de trabajo

Se realizó una mesa de trabajo con medidas de 50x80 cm y con una base metálica con una altura de 15 cm.



Figura 60. Mesa de trabajo

IX-D. SOPORTE DE CÁMARA

La estructura que se diseñó es eficiente, podemos ajustar moviendo de manera vertical y horizontal nuestra cámara.



Figura 61. Ensamble del soporte para la cámara

IX-E. Resultados de práctica 1 (Calibración de colores)

En esta práctica se obtienen los resultados de los valores HSV del color verde limón.

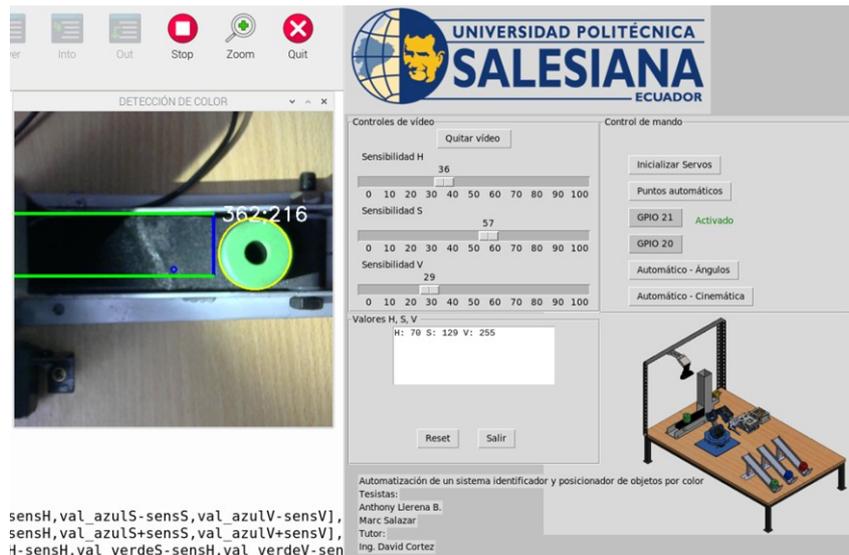


Figura 62. Visualización de valores HSV para calibración de color del objeto verde limón.

Una vez inicializada la programación y teniendo visible la imagen de nuestra cámara, dando clic con el scroll de nuestro mouse para seleccionar nuestro color. En este caso los valores hsv del color verde limón seleccionado son: H=70 ; S=129 ; V=255 (ventana inferior izquierda), y para ajustar nuestra sensibilidad obtenemos otros valores SHV que son: H= 36 ; S=57 ; V=29 (ventana superior izquierda).

```

56 #Calibración de colores:
57 sensH=36
58 sensS=57
59 sensV=29
60 sens rojoH=15
61 sens rojoS=20
62 sens rojoV=30
63
64 val_azulH=107
65 val_azulS=169
66 val_azulV=219
67 val_verdeH=80
68 val_verdeS=159
69 val_verdeV=120
70 val_rojoH=178
71 val_rojoS=153
72 val_rojoV=165
73
74 #NUEVO COLOR
75 val_vlimonH=70
76 val_vlimonS=129
77 val_vlimonV=255
    
```

Figura 63. Valores de sensibilidad y HSV de los objetos de color rojo, azul, verde y verde limón.

Finalmente se observa el propósito de la práctica, donde se visualiza la detección del objeto verde limón en la cámara, De esta manera practicando con varios colores se logra una buena calibración recordando que la segmentación es lo más importante a la hora de detectar a un objeto.

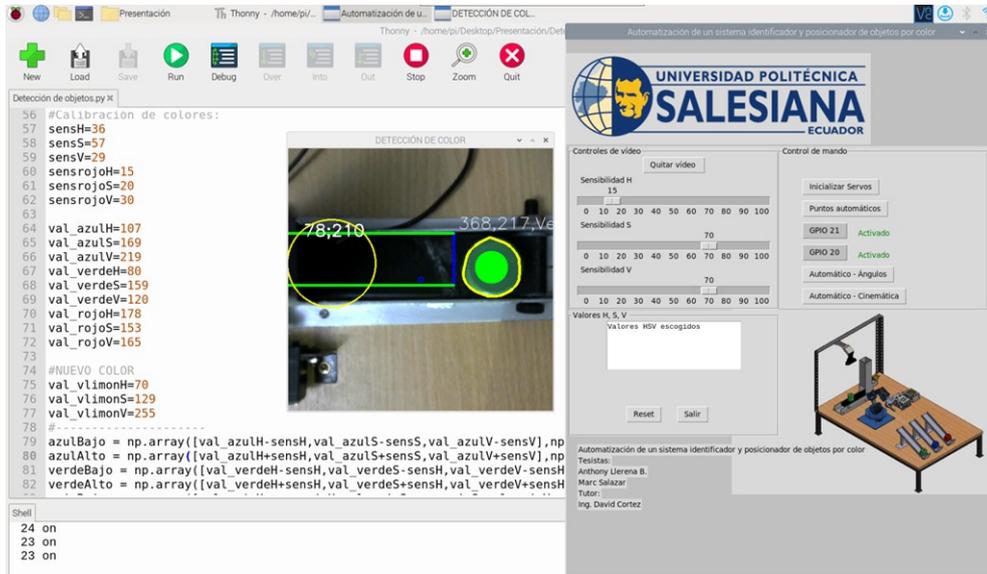


Figura 64. Visualización de la Detección de color del objeto aplicando valores HSV

Conclusión:

Conocer la importancia de los valores HSV, logrando de esta forma que la cámara pueda realizar la detección del objeto. Recordando que se utiliza un soporte para la cámara ya que debe permanecer estática y en el punto adecuado; También se guardan los valores obtenidos en la programación de la práctica final ya que están enlazadas en el proceso de funcionalidad.

IX-F. Resultados práctica 2 (Posicionamiento manual de servomotores por ángulos)

Fundamentalmente los estudiantes pueden obtener los ángulos de posicionamiento de los servomotores con la ayuda de la programación realizada que se encuentra guardada en el microcontrolador y utilizada para el movimiento de los motores en el espacio de trabajo.

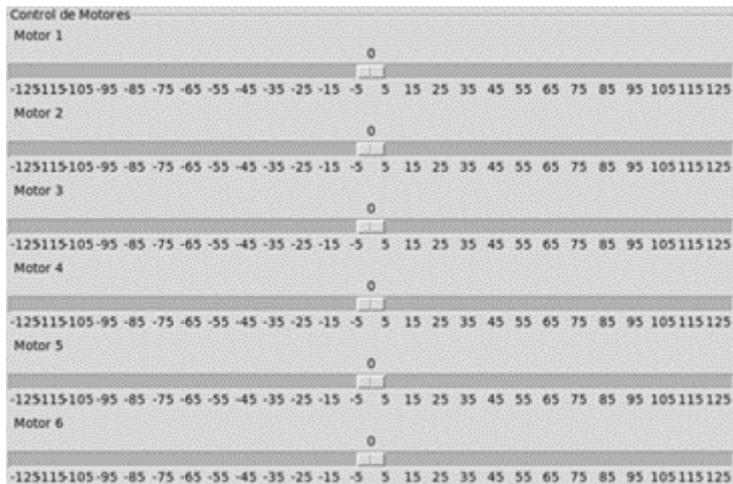


Figura 65. Sliders para posicionamiento de cada servomotor del brazo

```
326 def ArmPosition_home():
327     global arm
328
329     arm.setPosition(2, 0.0, True)
330     arm.setPosition(3, 0.0, True)
331     arm.setPosition(4, 0.0, True)
332     arm.setPosition(5, 0.0, True)
333     arm.setPosition(6, 0.0, True)
```



Figura 66. Seteo de posición home del brazo

Los ángulos intermedios que seteamos, nos servirán para evitar llegas bruscas a los puntos seteados.

```
343
344 def ArmPosition_banda_intermedia():
345     global arm
346
347     arm.setPosition(2, -84.0, True)
348     arm.setPosition(3, 0.0, True)
349     arm.setPosition(4, 0.0, True)
350     arm.setPosition(5, -5.0, True)
351     arm.setPosition(6, -77.0, True)
```

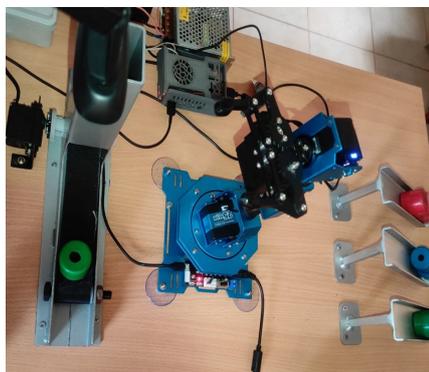


Figura 67. Posición intermedia entre el origen y la cinta

Del mismo modo realizamos el proceso, obteniendo la posición para llegar a la banda.

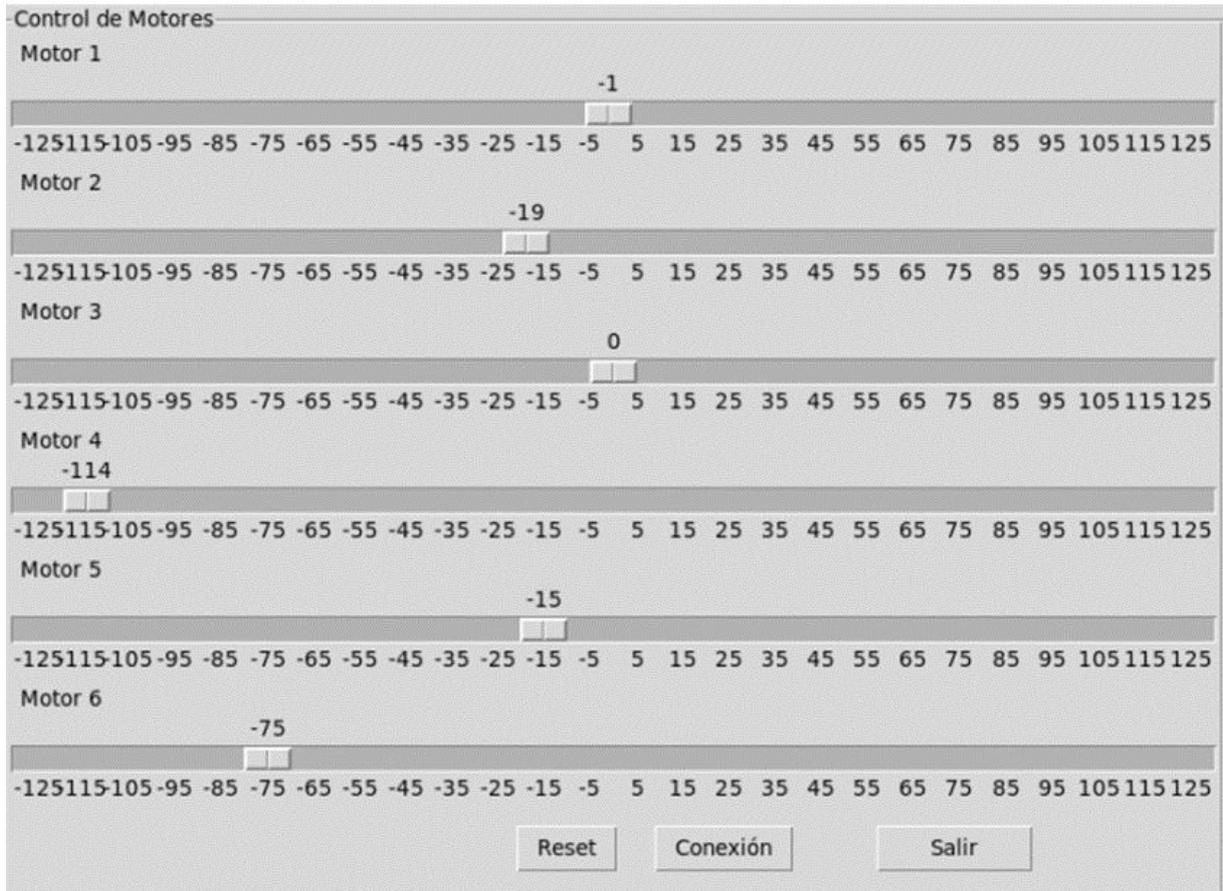
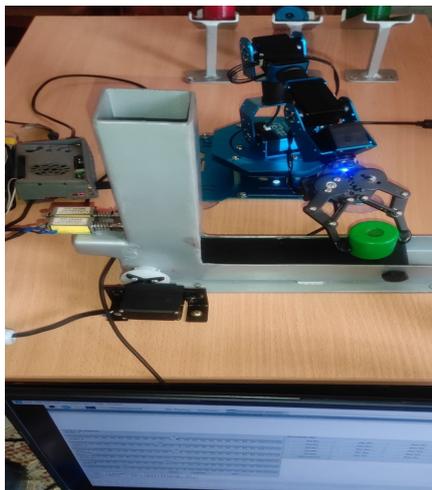


Figura 68. Configuración para posición deseada hasta la banda



```
335 def ArmPosition_banda():
336     global arm
337
338     arm.setPosition(2, -19.0, True)
339     arm.setPosition(3, 0.0, True)
340     arm.setPosition(4, -114.0, True)
341     arm.setPosition(5, -15.0, True)
342     arm.setPosition(6, -75.0, True)
```

Figura 69. Posición deseada hasta la banda

Se clasifica de manera óptima los colores previamente configurado a cada punto de llegada.

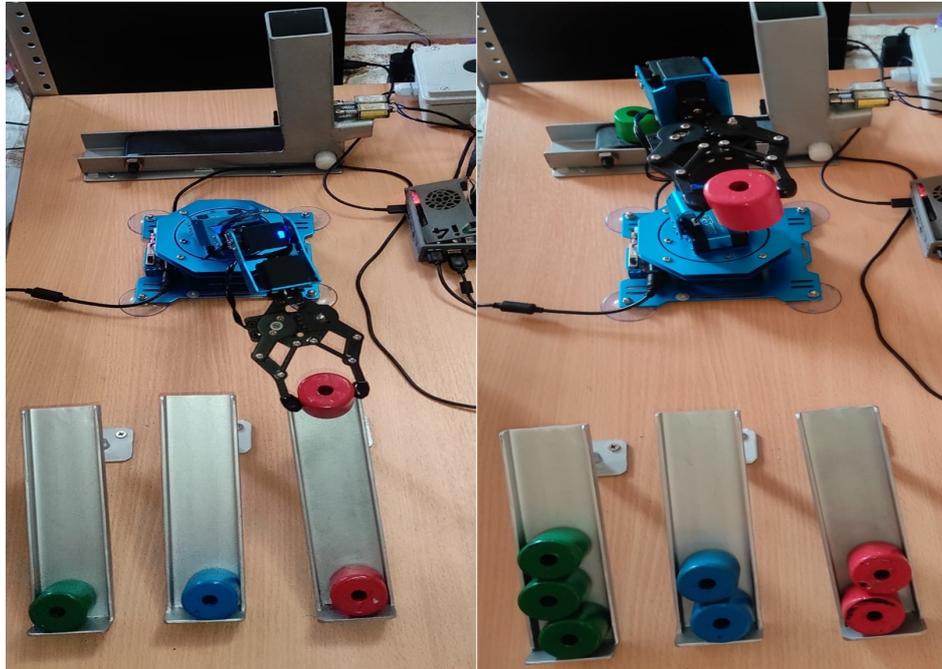


Figura 70. Posicion de llegada del objeto color rojo

Se tiene que tener en cuenta que los posicionamientos varían dependiendo de los puntos de llegadas como se observa en la figura [70].

Conclusión:

Los estudiantes fueron introducidos al manejo manual de posicionamiento mediante ángulos y conocimientos de cada uno de los comandos que genera la programación para efectuar el requerimiento de la práctica.

IX-G. Resultados práctica 3 (Identificación y posición de objetos según color)

Para esta práctica se unen las otras dos prácticas anteriores. Realizando la detección del color, la recolección y la clasificación.



Figura 71. Interfaz final para la tercera práctica

- **Proceso 1** Para el inicio del proceso se comprueba que los electroimanes funcionan correctamente, evitando el paso de los objetos.

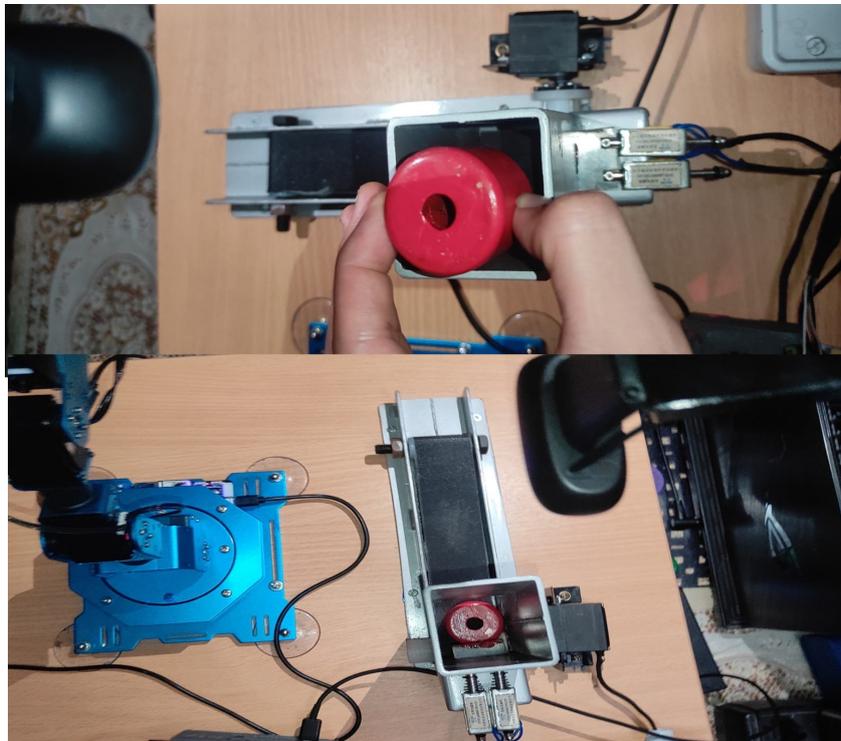


Figura 72. Funcionamiento correcto de los electroimanes

Una vez se accione el electroimán, éste se retrae y vuelve a su posición inicial rápidamente, logrando el paso de un solo objeto a la vez.

- **PROCESO 2** En el proceso mostrado en la figura [73], observamos que nuestro objeto desde su punto inicial, llega hasta un punto específico (línea azul), estas coordenadas (x,y) son respecto a la visión de nuestra cámara, por ende pueden variar al momento de moverla.

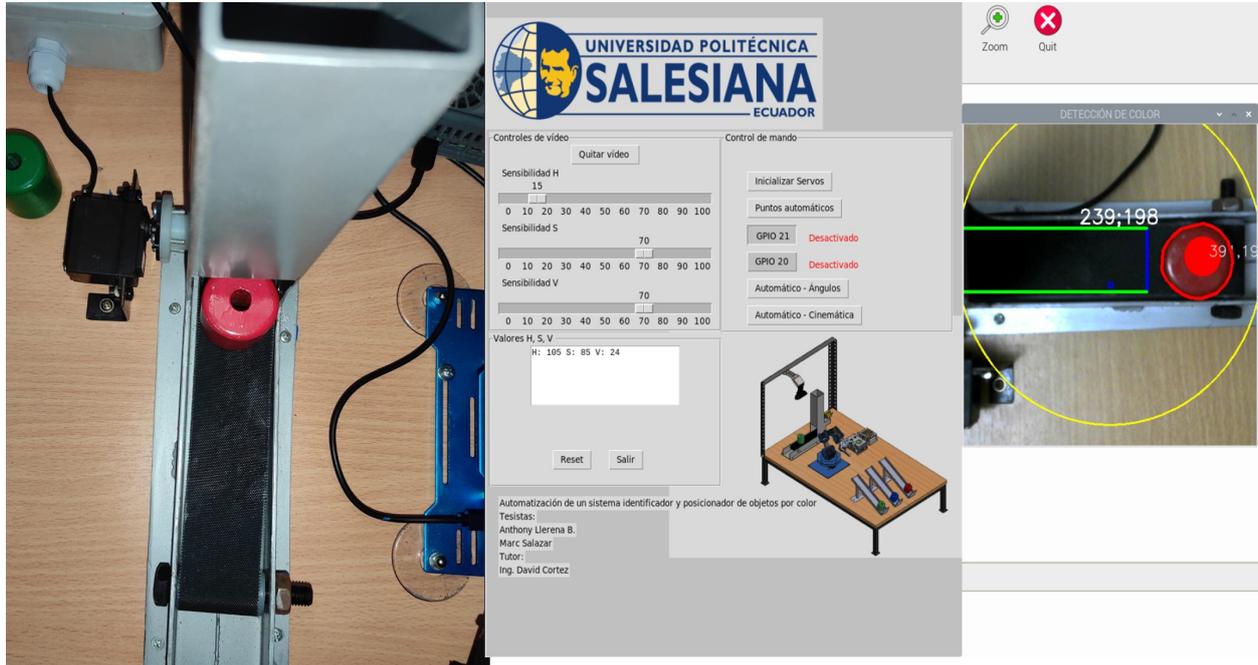


Figura 73. Posición inicial y llegada del objeto

- **PROCESO 3** En este proceso se observa la recolección y clasificación de objeto (rojo). En la figura [74], nuestro brazo se dirige al punto de llegada del objeto, para luego trasladarlo a los puntos específicos para cada color, como se muestra en la figura [75].



Figura 74. Recolección de objeto

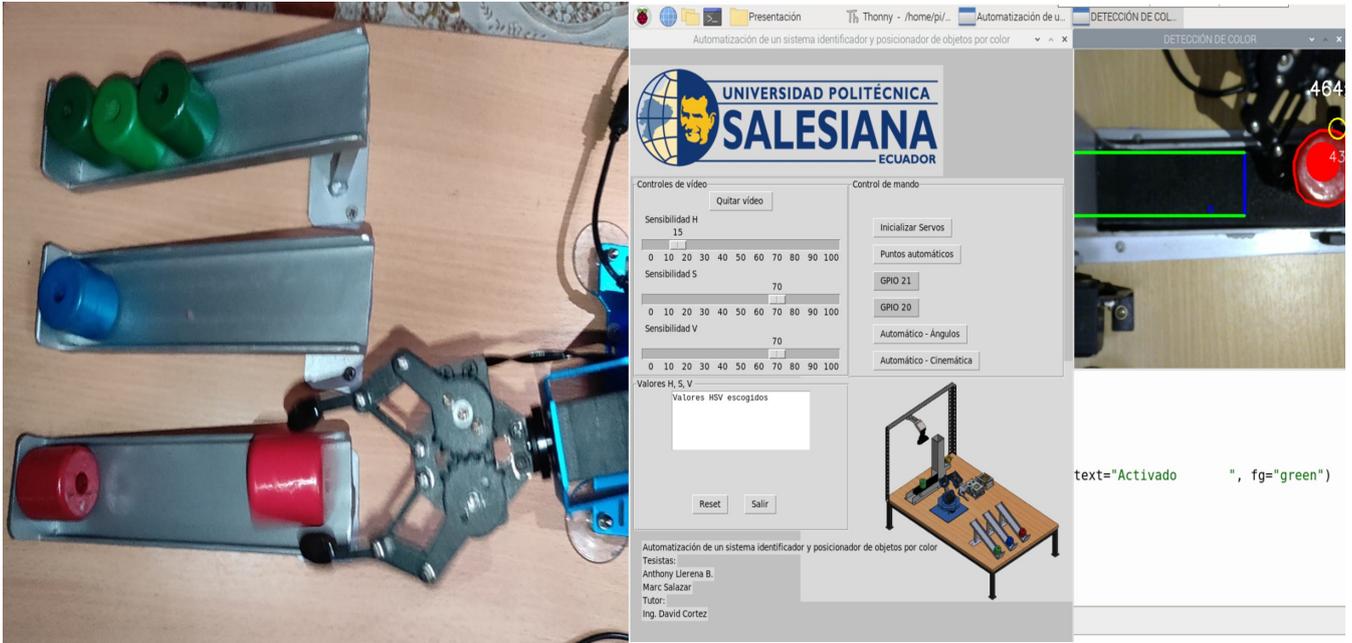


Figura 75. Clasificación de objeto por color

XI. PRESUPUESTO

Desarrollo de un sistema indentificador y posicionador de objetos mediante un brazo robótico a través de visión artificial con lenguaje Python.					
Tabla de Presupuestos					
Elemento	Tipo de recurso	Cantidad	Medida	Costo unitario (\$)	Costo total
Borneras	Conexión electrica	10	Unidad	\$ 2,00	\$ 20,00
Boton	Paro de emergencia	1	Unidad	\$ 10,00	\$ 10,00
Cables	Conexión	10	Metros	\$ 1,60	\$ 16,00
Camara HD	Camara HD	2	Unidad	\$ 80,00	\$ 160,00
Capacitación	Curso Control Industrial	2	Unidad	\$ 20,00	\$ 40,00
Capacitación	Curso Python	2	Unidad	\$ 30,00	\$ 60,00
Cinta o Banda	Banda Transportadora	1	Unidad	\$ 200,00	\$ 200,00
Computadora	Diseño y Programación	1	Unidad	\$ 800,00	\$ 800,00
Disyuntores	Protección	3	Unidad	\$ 15,00	\$ 45,00
Estructura	Banda Transportadora	1	Unidad	\$ 150,00	\$ 150,00
Fuentes	Energización	2	Unidad	\$ 20,00	\$ 40,00
Fusibles	Protección	15	Unidad	\$ 2,00	\$ 30,00
JETSON NANO 2G	Microcontrolador	1	Unidad	\$ 250,00	\$ 250,00
Material	Sistemas de circuitos electricos	40	Unidad	\$ 3,00	\$ 120,00
Material	Resistencia,-Breadboard	1	100 Unidades	\$ 2,00	\$ 2,00
OVI 535	Brazo robótico	1	Unidad	\$ 250,00	\$ 250,00
Personal	Horas de trabajo	960	Horas	\$ 3,33	\$ 3.196,80
Personal	Viajes	5	Unidad	\$ 50,00	\$ 250,00
Personal	Visitas a empresas	3	Unidad	\$ 20,00	\$ 60,00
Personal especializado	Ayuda Profesional	1	Unidad	\$ 500,00	\$ 500,00
Piezas	Diseño	3	Unidad	\$ 40,00	\$ 120,00
Planos	Diagramas de Fuerza y Control	16	Unidad	\$ 10,00	\$ 160,00
Raspberry PI 4 2G	Microcontrolador	1	Unidad	\$ 200,00	\$ 200,00
Sensores	Protección	10	Unidad	\$ 15,00	\$ 150,00
Servomotores	Servomotores	3	Unidad	\$ 30,00	\$ 90,00
Transmisión	Tambores y Motores	1	Unidad	\$ 350,00	\$ 350,00
Varios	Laboral	1	Unidad	\$ 300,00	\$ 300,00
Viaticos	Transporte	150	Unidad	\$ 2,00	\$ 300,00
Viaticos	alimentación	240	Unidad	\$ 3,00	\$ 720,00
Viaticos	Hospedaje	10	Unidad	\$ 40,00	\$ 400,00
TOTAL					\$ 8.989,80

Figura 77. Presupuesto

XII. CONCLUSIONES

- La verificación de las conexiones de los componentes con equipos técnicos seleccionados como un multímetro evitan que al instante o al paso de tiempo se quemen o se dañen por no utilizar los métodos de evitar riesgos aprendidos en ingeniería.
- Se logró el ensamblaje y diseño 3D que se realizó mediante el software SolidWorks, este diseño fue realizado mediante la medición del brazo adquirido, aplicando conocimientos aprendidos en la UPS.
- Se logró obtener las medidas del espacio de trabajo para poder mandar hacer la mesa donde están incorporados los equipos y componentes del sistema.
- Se realizaron con éxito las pruebas de funcionamiento del brazo robótico, Las cuales fueron realizadas mediante lenguaje de programación Python y el software MATLAB.
- Se logró crear una programación eficiente para catalogar objetos, con cuales se realizaron las pruebas convenientes con el uso de la cámara comprobando la detección objeto por color y llevando de un punto A, a un punto B mediante el uso de las articulaciones del brazo robótico.
- Según los resultados obtenidos en las practicas desarrolladas, los estudiantes tendrán la oportunidad de documentar, analizar, e interactuar, con el robot, enriqueciendo así su proceso de aprendizaje.
- Las guías de prácticas se elaboraron de forma que se desarrolla la programación, con explicación adecuada para que al momento de realizar el docente o alumno pueda seguir las instrucciones y pueda efectuar de manera correcta.

XIII. RECOMENDACIONES

- Se recomienda utilizar el componente LM2596 que es un convertidor de voltaje regulable utilizado para conectar desde la fuente de 12V a servomotor S3003 de 6V; Logrando enviar la tensión requerida y no causar daño interno al equipo electrónico.
- Se recomienda para proyectos similares utilizar tarjeta controladora NVIDIA Jetson GPU TX2, la diferencia de la raspberry pi 4 model B es que esta cuenta con tarjeta de video.
- Es recomendado no utilizar dos softwares de diseño al momento del ensamble se podrá tener inconvenientes.
- Podemos ver que para la cámara se necesita un soporte por el motivo de que debe estar situada en la parte superior del funcionamiento del sistema.
- Se sugiere comprar cursos en línea de Python para utilizar de guía en el proyecto que se esté efectuando.
- Poseer conocimiento previo relacionado a diseño, programación, dimensionamiento de motores y todo tema afín al desarrollo de este tipo de tecnología para lograr el mayor nivel de comprensión al momento del análisis del equipo.
- Establecer las especificaciones y límites del proyecto antes de su desarrollo, con el fin de reducir correcciones.
- Es importante que al momento de realizar las prácticas estén conectados correctamente todos los equipos a utilizar en especial el brazo conectado a su fuente ya que si no se prende una alarma.
- Reutilizar acero empleando técnicas de desoxidación.

REFERENCIAS

- [1] K. A. Calderón Martínez, *DESARROLLO DE UN SISTEMA DE CONTROL PARA UN BRAZO ROBÓTICO UTILIZANDO VISION ARTIFICIAL PARA LOS LABORATORIOS DE MECATRÓNICA DE LA UTN*. Universidad Técnica del norte, Ibarra, Ecuador, 2017.
- [2] M. Costa, R. Gouveia y F. Silva, «How to solve quality problems by advanced fully-automated manufacturing systems,» *International Journal of Advanced Manufacturing Technology*, vol. 94, págs. 3041-3063, 2018.
- [3] D. S. Escalante Cordova y D. M. Vargas Ospina, «Raspberry PI: la tecnología reducida en placa,» 2019.
- [4] S. S. Escandón Vizueta y M. C. Trujillo Torres, *Análisis cinemático de un brazo robótico de cinco grados de libertad e implementación de un control en Matlab*. Guayaquil, Ecuador, 2017.
- [5] ESPOL, A. U. B. Esperanza y L. B. Y. Yohany, «Automatización e implementación de un sistema de Administración de Procesos de Negocio (BPM) para el proceso de administración de proyectos basado en las metodologías Capability Maturity Model Integration (CMMI - DEV) nivel 3 para Servicios Profesionales Cima-E S.A.,» *Revistaespacios*, vol. 39, pág. 19, 2017.
- [6] E. D. Garcés Hernández, *Implementación de un prototipo de brazo robótico que realiza control de calidad con visión artificial, para el laboratorio de automatización de la Escuela de Ingeniería Industrial de la ESPOCH*. Escuela Superior Politécnica de Chimborazo, Ecuador, 2017.
- [7] J. Gómez Ortega, F. Dreoni, M. Ortega y F. R. Rubio, «Identificación t control de un robot manipulador,»
- [8] R. Jara Ruiz, M. E. Rodríguez Franco, Y. F. López Álvarez y R. D. Robles Valenciano, «Cylindrical robot control,» *Revista de Ingeniería Biomédica y Biotecnología*, vol. 2, n.º 5, págs. 22-28, 2018.
- [9] E. Laona Jiva, *DESARROLLO DE LA TELEOPERACIÓN DE ROBOTS INDUSTRIALES Y COLABORATIVOS MEDIANTE TÉCNICAS AVANZADAS DE VISIÓN ARTIFICIAL*. Universitat Politècnica de València, España, 2019.
- [10] N. López y C. Ruiz, *Diagnóstico de la implementación del sistema de inventario KANBAN en el área de corte AALFS uno, Sébaco Matagalpa, segundo semestre 2012*. Universidad Nacional Autónoma de Nicaragua, Managua, 2017.
- [11] G. N. Montenegro Alcívar y M. L. Yaguachi Ríos, *DISEÑO, CONSTRUCCIÓN Y CONTROL DE UN PROTOTIPO DE BRAZO ROBÓTICO ANTROPOMÓRFICO PARA LA SOLDADURA POR PUNTO DE DIFERENTES FIGURAS, MEDIANTE EL USO DE CINEMÁTICA INVERSA Y VISIÓN ARTIFICIAL*. Quito, UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO, 2020.
- [12] A. M. Pucas Flores y V. E. Rosado Rendón, *Diseño de un prototipo brazo robótico con sistema de visión por computador para la clasificación de empaques por colores*. Milagro, UNIVERSIDAD ESTATAL DE MILAGRO FACULTAD CIENCIAS E INGENIERÍA, 2020.
- [13] F. J. Rechi, *Análisis de riesgos laborales en el sector de la electrónica y la automática*. Valencia, España, 2019.
- [14] J. Rodríguez y J. García, *MATLAB: guía de aprendizaje*. Córdoba, Argentina: Jorge Sarmiento Editor - Universitas, 2020.
- [15] C. L. Rodríguez Rodríguez, *DISEÑO Y CONSTRUCCIÓN DE UN ROBOT CARTESIANO CON UN CONTROL DE POSICIÓN PUNTO A PUNTO*. UNIVERSIDAD LIBRE FACULTAD DE INGENIERÍA INGENIERÍA MECÁNICA, BOGOTÁ D.C., 2017.
- [16] E. S. Valencia Olvera, *Desarrollo e implementación de un sistema de control para un ROV Lynx*. Santiago de Querétaro, Qro., México, 2019.

XIV. ANEXOS

XIV-A. GUÍA DE PRÁCTICA 1

		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / ROBÓTICA – PARA DOCENTES	
CARRERA: Ingeniería Mecatrónica			
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Calibración de colores.	
OBJETIVO GENERAL. Reconocimiento de objeto por su color mediante cámara USB.			
OBJETIVOS ESPECÍFICOS: - Encendido del sistema. -Manejar la pantalla de control. -Analizar y visualizar la segmentación de colores. -Obtener los valores HSV. -Introducir valores encontrados en la programación para comprobación del funcionamiento.			
INSTRUCCIONES (Detallar las instrucciones que se dará al estudiante):		<ol style="list-style-type: none"> 1. Conectar los equipos adecuadamente; Principalmente la raspberry y el brazo. 2. Abrir carpeta presentación y abrir la programación que está en lenguaje Python denominada detección de objetos.py. 3. Insertar en ventana de control mostrar video y con el mouse dando clic en el medio va a seleccionar el objeto donde en una pantalla pequeña blanca le saldrán los valores HSV.} 4. Dirigirse al código y buscar las líneas con el nombre #Calibración de colores y remplazar con los valores obtenidos. 	
ACTIVIDADES POR DESARROLLAR			
<ol style="list-style-type: none"> 1. Encendido del brazo, microcontrolador y fuente para activación del sistema. 2. Conectarse mediante VNC a la computadora del laboratorio y a la vez activar anyDesk para manera remota. 3. Reconocer las librerías openCv (cv2) que es la librería utilizada para la visión artificial. 4. Conectar correctamente los cables de comunicación desde los brazos robóticos al módulo de control y la conexión del módulo de control con el computador. 5. Poder manejar el panel de control para poder obtener los valores HSV y poder colocar en la programación que está modificada para la facilidad de los estudiantes. 6. Colocar los valore en el código en la carpeta mencionada y darle Run a la programación. 			
RESULTADO(S) OBTENIDO(S): El estudiante puede conectarse de manera remota y obtener los resultados HSV que es el valor de la matriz de saturación, con el valor podremos detectar el objeto mediante la cámara en tiempo real.			
CONCLUSIONES: El estudiante adquirirá el conocimiento de como ingresar los valores de HSV en la programación y ver como resultado la detección con una circunferencia en la parte superior del objeto de su mismo color.			
RECOMENDACIONES: todos los equipos que compone el sistema para que la programación de calibración sea ejecutada correctamente; Verificar la claridad del lugar donde se va a realizar la calibración del color debido, revisar que la cámara este posicionada en un lugar adecuado a donde está situado el objeto.			

Docente: Ing. DAVID MATEO CORTEZ SARAVIA, MSc.

Firma: _____

Figura 78. Guia de práctica 1

		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / ROBÓTICA – PARA DOCENTES	
CARRERA: Ingeniería Mecatrónica			
NRO. PRÁCTICA: 2		TÍTULO PRÁCTICA: Calibración de posicionamiento de servomotores para movimientos de punto A, a punto B.	
OBJETIVO GENERAL. Obtener las posiciones mediante ángulos utilizando una programación en Python.			
OBJETIVOS ESPECÍFICOS: - Calibrar los servomotores al punto home. - Conocer el límite de libertad de cada eslabón - Manejar el panel de control de la programación. - Colocar en un punto requerido el brazo robótico - Guardar valores encontrados en cada servomotor. - Implementar valores de las qi (ángulos) del servomotor en la programación final.			
INSTRUCCIONES (Detallar las instrucciones que se dará al estudiante):		5. Abrir la carpeta programación y seleccionar el archivo mov_motor_xarm.py. 6. Colocar Run a la programación, revisar si no existe error de sintaxis. 7. Colocar las barras de cada servomotor en 0. 8. Dirigirse de punto A, a punto B manualmente y guardar el valor de cada motor que sale encima de la barra que es de -100-0-100.	
ACTIVIDADES POR DESARROLLAR			
7. Encendido del brazo, microcontrolador en este caso no es necesario la fuente que este activada. 8. Ubicar un punto A mediante las barras recordando que en el caso de la pinza abre en (-) y cierra en (+), un poco distinto para cada articulación una sube y la otra baja, recordando anotar los ángulos de cada motor al mover. 9. Ubicar el punto B y anotar para las practica final. 10. Abrir programación detección de objetos.py y colocar los valores en la parte de automático por ángulos y luego darle Run para ver el funcionamiento requerido.			
RESULTADO(S) OBTENIDO(S): El estudiante podrá controlar desde su computadora el brazo robótico, a la vez podrá guardar los datos y colocando en la programación visualizar de modo automático por ángulos que se encontraron al principio.			
CONCLUSIONES: El estudiante adquiere interés por los procesos autónomos y también experimentan los pasos a seguir dando una experiencia de un proceso de manufactura.			
RECOMENDACIONES: Colocar los valores obtenidos de los ángulos de cada servomotor en automático ángulos no en automático cinemáticas; Recordar anotar los valores es importante.			

Docente: Ing. DAVID MATEO CORTEZ SARAVIA, MSc.

Firma: _____

Figura 79. Guía de practica 2

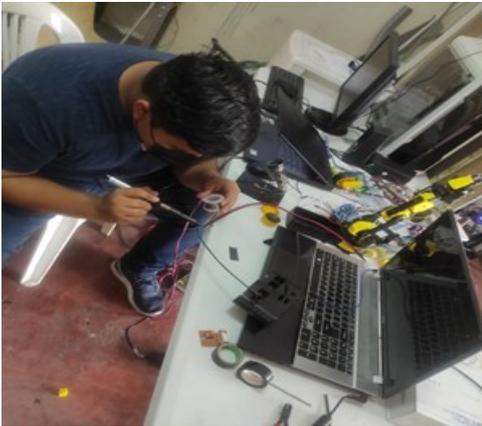
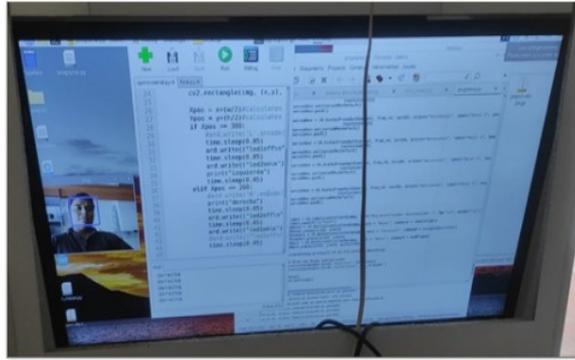
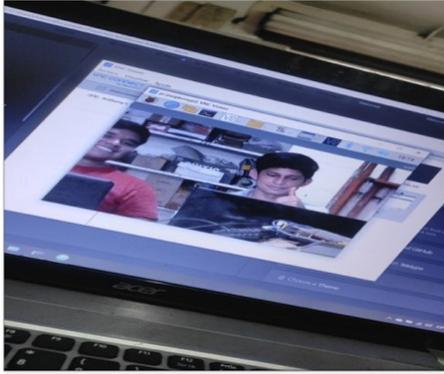
		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / ROBÓTICA – PARA DOCENTES	
CARRERA: Ingeniería Mecatrónica			
NRO. PRÁCTICA:	3	TÍTULO PRÁCTICA: Identificación y posicionamiento de objetos según su color.	
OBJETIVO GENERAL.			
OBJETIVOS ESPECÍFICOS: -Clasificar los objetos por su color. -Detección de color verde, rojo y azul. -Funcionamiento mediante ángulos el proceso del sistema. -Funcionamiento mediante cálculos cinemáticos el proceso del sistema. -Verificar funcionamiento de componentes y equipos utilizados. -Ubicación del objeto al punto de llegada según su color.			
INSTRUCCIONES (Detallar las instrucciones que se dará al estudiante):		9. Revisar conexiones de raspberry, brazo robótico y fuente. 10. Abrir la carpeta programación y seleccionar el archivo detección de objetos.py. 11. Dar Run a la programación y seleccionar en panel de control mostrar video. 12. Comprobar con los botones GPIO 21 y GPIO 20 el funcionamiento de las solenoides y banda transportadora. 13. En el panel de control seleccionar el botón automático-Ángulo. 14. Para poder activar los botones puntos automáticos y Automático-cinemática se tiene que oprimir el botón inicializar servos y los dos botones antes mencionados funcionaran.	
ACTIVIDADES POR DESARROLLAR			
11. El primer punto es colocar objetos en el orificio de la estructura de la banda transportadora. 12. En cuanto a poner en marcha se debe dar Run a la programación antes mencionada en instrucciones y seleccionar mostrar cámara y verificar que tenga un enfoque a la banda transportadora, realizado lo anterior damos clic en el botón en el panel de control Automático-Ángulos y observamos el funcionamiento. 13. En relación con el movimiento anterior seleccionando primero al inicializar servos se podrá dar clic en puntos automáticos y observar los movimientos que realiza al reconocer un objeto y posicionar en su lugar de llegada y por último inicializar Automático-cinemática que hace el mismo funcionamiento con las diferencias que se obtiene cálculos cinemáticos y con una ganancia (velocidad)colocada en programación servos.py. 14. Colocar más objetos de color rojo, azul y verde una vez inicializado el proceso el brazo comenzara a clasificar por su color y colocar en su diferente punto de llegada.			
RESULTADO(S) OBTENIDO(S): Mediante los dos botones seleccionados uno Automático-Ángulo y el otro Automático-cinemática se observa la detección y posicionamiento del objeto mediante el color captado por la cámara, además comprobamos que los valores obtenidos en las anteriores practicas son parte de la final para lograr el movimiento y la detección.			
CONCLUSIONES: Por todo lo mencionado podemos concluir que la librería openCv (cv2), los valores HSV, ángulos de movimiento de servomotores y los cálculos cinemáticos mediante matriz homogénea y jacobiano, entre otros...; Se logra el objetivo principal de seleccionar y posicionar los objetos mediante color.			
RECOMENDACIONES: Colocar los objetos de diferente color con el objetivo de comprobar la clasificación según su color; La posición de la cámara tienen que estar apuntando a la cinta transportadora.			

Docente: Ing. DAVID MATEO CORTEZ SARAVIA,MSc.

Firma: _____

Figura 80. Guía de práctica 3

XIV-D. PRUEBAS DE VISIÓN ARTIFICIAL Y ENSAMBLE DEL BRAZO



XIV-E. DISEÑO 3D, SIMULACIÓN Y CONSTRUCCIÓN DE ESTRUCTURAS

