

RAE

1. **TIPO DE DOCUMENTO:** Trabajo de grado para optar por el título de Ingeniero Mecatrónico.
2. **TÍTULO:** Control de posición de brazo robótico para clasificación de herramientas por medio de visión artificial.
3. **AUTORES:** Juan Sebastián Huertas Piedrahita, Jaime Arturo López Duque y Eduardo Luis Pérez Londoño.
4. **LUGAR:** Bogotá, D.C.
5. **FECHA:** Julio 06 de 2016.
6. **PALABRAS CLAVES:** Imagen, pixel, histograma, binarización, umbralización, escala de grises, RGB, caracterización, manipulador robótico, trayectoria, control de posición.
7. **DESCRIPCIÓN DEL TRABAJO:** Este trabajo de investigación plantea el desarrollo de un sistema de clasificación de figuras compuesto por un brazo robótico y una cámara, la cual realiza el proceso de caracterización de las figuras y las ubica en diferentes cubículos, con el fin de automatizar la recolección y organización.
8. **LINEAS DE INVESTIGACIÓN:** Robótica, procesamiento de imágenes y automatización de procesos.
9. **METODOLOGÍA:** Empírico Analítica.
10. **CONCLUSIONES:** Las capturas y procesado de imágenes depende mucho de la iluminación, ya que mejora o empeora los procesos que debe realizar el programa, debido a que las condiciones programadas se ven afectadas por dicha iluminación.

Las soluciones de clasificación en procesos industriales mejoran su rendimiento mediante el uso del procesamiento de imágenes, ya que un solo sensor (cámara) puede determinar las características de distintas variables y de esta manera realizar una clasificación robusta con un sistema más fácil de implementar, siempre y cuando se cuente con un ambiente controlado para evitar cambios en la iluminación.

**CONTROL DE POSICIÓN DE BRAZO ROBÓTICO PARA CLASIFICACIÓN DE
HERRAMIENTAS POR MEDIO DE VISIÓN ARTIFICIAL**

**JUAN SEBASTIÁN HUERTAS PIEDRAHITA
JAIME ARTURO LÓPEZ DUQUE
EDUARDO LUIS PÉREZ LONDOÑO**

**UNIVERSIDAD DE SAN BUENAVENTURA, BOGOTÁ
FACULTAD DE INGENIERÍA
INGENIERÍA MECATRÓNICA
BOGOTÁ 2016**

**CONTROL DE POSICIÓN DE BRAZO ROBÓTICO PARA CLASIFICACIÓN DE
HERRAMIENTAS POR MEDIO DE VISIÓN ARTIFICIAL**

**JUAN SEBASTIÁN HUERTAS PIEDRAHITA
JAIME ARTURO LÓPEZ DUQUE
EDUARDO LUIS PÉREZ LONDOÑO**

Tesis para optar al título de Ingeniero Mecatrónica

**Profesor Asesor:
Ing. Julián Severiano Rodríguez Acevedo M.Sc.**

**UNIVERSIDAD DE SAN BUENAVENTURA, BOGOTÁ
FACULTAD DE INGENIERÍA
INGENIERÍA MECATRÓNICA
BOGOTÁ 2016**

AGRADECIMIENTOS

Agradezco a cada una de las personas que fueron parte de este importante proceso en mi vida, principalmente a mi padre y mi madre que son y serán siempre la base fundamental para todos mis proyectos, también un apartado especial para mi hermano que siempre me ha apoyado en todas mis decisiones. Agradezco especialmente a mi novia por siempre estar cuando más lo he necesitado.

Quiero dar un agradecimiento especial a dos profesores que fueron parte fundamental de mi desarrollo académico, Julián Severiano Rodríguez Acevedo y Hernando Evelio Leyton Vásquez, quienes me guiaron en el arduo camino de la Ingeniería Mecatrónica y a los cuales aprecio enormemente.

JUAN SEBASTIÁN HUERTAS PIEDRAHITA

Agradezco en primer lugar a Dios, por darme sabiduría y paciencia durante el camino, por permitirme estudiar esta excelente carrera y cumplir este sueño de obtener el título de Ingeniero Mecatrónico.

Agradeciendo especialmente a mi familia, a mis padres, Luis y Ninfa, por su apoyo incondicional y confianza, por sus sacrificios y por darme fuerzas, y a mi hermano Luis, por siempre estar a mi lado ayudándome y apoyándome en todo momento, y a Nerys y a mi abuela Susana.

Mil gracias a mi novia Kiara y a sus padres Fidel y Divina, por confiar en mí, por sus palabras de aliento, por animarme a seguir adelante y por sus consejos.

También quiero agradecer al profesor Julián Severiano Rodríguez Acevedo, por su ayuda para poder culminar este trabajo de grado, y al profesor Hernando Evelio Leyton Vásquez, por hacer parte de mi proceso académico y compartir sus conocimientos.

¡INFINITAS GRACIAS!

EDUARDO LUIS PÉREZ LONDOÑO

En primer lugar, quiero agradecer a Dios por haberme guiado y por darme una familia tan maravillosa, que me ha ayudado a cumplir un logro importante en mi vida. A mis padres por la educación, que con ejemplo y amor me han inculcado valores, ayudándome a hacer una mejor persona. A mis hermanas por el apoyo constante en todo mi proceso de formación, por las conversas y enseñanzas compartidas.

Por ultimo quiero agradecer a la Universidad de San Buenaventura (sede Bogotá), al grupo de Ingenieros de la Facultad de Ciencias Básicas e Ingeniería, que a lo largo de la carrera aportaron su conocimiento y profesionalismo, brindándome muchas experiencias y momentos de aprendizaje. También quiero agradecer enormemente al Ingeniero Julián Severiano, por su dedicación, paciencia, y orientación en todas las etapas de desarrollo de este proyecto y de mi profesión como Ingeniero Mecatrónico.

A todos los que directa o indirectamente aportaron a la construcción de este trabajo de grado.

JAIME ARTURO LÓPEZ DUQUE

Nota de aceptación:

Firma jurado

Firma jurado

Bogotá D.C 06 de Julio del 2016

TABLA DE CONTENIDO

LISTA DE FIGURAS	1
LISTA DE TABLAS	3
LISTA DE ANEXOS	4
INTRODUCCIÓN.....	5
1. PLANTEAMIENTO DEL PROBLEMA	7
1.1 ANTECEDENTES DE LA INVESTIGACIÓN.....	7
2. JUSTIFICACIÓN	12
3. OBJETIVOS DE LA INVESTIGACIÓN	13
3.1 OBJETIVO GENERAL	13
3.2 OBJETIVOS ESPECÍFICOS.....	13
3.3 LIMITACIONES	13
4. MARCO REFERENCIAL.....	14
4.1 MARCO CONCEPTUAL	14
4.1.1 VISIÓN ARTIFICIAL.....	14
4.1.2 MANIPULADORES ROBOTICOS.....	14
4.2 MARCO TEÓRICO	14
4.2.1 Etapas para una aplicación en Visión Artificial	15
4.2.2 Características y clasificación de manipuladores robóticos	16
5. DISEÑO INGENIERIL	22
5.1 SOFTWARE Y LIBRERÍAS.....	22
5.2 VISIÓN ARTIFICIAL	22
5.2.1 ADQUISICIÓN DE IMAGEN.....	22
5.2.2 DETECCIÓN DE COLORES RGB	24
5.2.3 UMBRALIZACIÓN.....	25
5.2.4 CONTORNOS.....	29
5.2.5 EXTRACCIÓN DE COORDENADAS CARTESIANAS DE LAS FIGURAS A CLASIFICAR	31
5.3 DISEÑO MECÁNICO MANIPULADOR ROBÓTICO	36
5.3.1 Cinemática	37
5.3.2 Cinemática Inversa	38

5.3.2.1 Métodos Geométricos	39
5.3.3 Modelo Dinámico	43
5.3.4.1 Solución del modelo dinámico mediante formulación de Lagrange	44
5.4 CONTROL DE POSICIÓN	53
5.4.1 Generación de Trayectoria	53
5.4.2 Control de Posición del Segundo Eslabón	60
5.4.2.1 Diseño Control PI (Proporcional Integral)	70
6. RESULTADOS	80
7. CONCLUSIONES	82
BIBLIOGRAFIA	84
ANEXOS	87

LISTA DE FIGURAS

Figura 1. Etapas de una aplicación de visión artificial	16
Figura 2. Elementos de un manipulador robótico	17
Figura 3. Articulación lineal	17
Figura 4. Articulaciones rotacionales	17
Figura 5. Punto terminal de un manipulador con pinza	18
Figura 6. Analogía entre manipulador (brazo robótico) y brazo humano	18
Figura 7. Distintos Grados de Libertad.....	19
Figura 8. Logitech HD Webcam C270.....	23
Figura 9. Matrices RGB de una imagen Roja.....	24
Figura 10. Imagen original	27
Figura 11. Imagen en escala de grises	28
Figura 12. Histograma	28
Figura 13. Imagen umbralizada	28
Figura 14. Ejemplo de resultados del detector de bordes de Canny para dos imágenes distintas cuando los umbrales alto y bajo son 50 y 10.....	30
Figura 15. Ejemplo de resultados del detector de bordes de Canny para imagen capturada por la cámara.....	30
Figura 16. Coordenadas cartesianas del objeto reconocido en la imagen capturada.....	31
Figura 17. Diagrama de flujo RGB y Binarización	33
Figura 18. Diagrama de flujo de formas de los objetos	34
Figura 19. Diagrama de flujo de coordenada central objetos	35
Figura 20. Círculo reconocido mediante detección de número de bordes.....	36
Figura 21. Bosquejo de la vista lateral de la maqueta	37
Figura 22. Análisis Cinemático Inverso	39

Figura 23. Análisis Ley de Coseno.....	41
Figura 24. Representación ángulo θ_2	43
Figura 25. Análisis mediante formulación de Lagrange de un manipulador de dos grados de libertad	44
Figura 26. Primer Diseño Brazo Robótico	52
Figura 27. Diseño Final Brazo Robótico.....	53
Figura 28. Vista Superior Maqueta.....	54
Figura 29. Captura para determinar centímetros en el eje X	55
Figura 30. Captura para determinar centímetros en el eje Y	56
Figura 31. Comparación entre simulación en Matlab y captura original	57
Figura 32. Ejemplo trayectoria eje X positivo	58
Figura 33. Ejemplo trayectoria eje X negativo	59
Figura 34. Ubicación de cubículos	60
Figura 35. Análisis mediante formulación de Lagrange.....	61
Figura 36. Diagrama esquemático del servomotor.....	64
Figura 37. Diagrama de bloques del servomotor.....	64
Figura 38. Caja reductora del servomotor	69
Figura 39. Comportamiento función de transferencia a respuesta de escalón unitario en lazo abierto	72
Figura 40. Ubicación de a (cero).....	74
Figura 41. Función de transferencia y controlador PI en Simulink.....	76
Figura 42. Comportamiento del controlador PI.....	77
Figura 43. Diagrama de flujo parámetros de control en Arduino	78
Figura 44. Diagrama final control	79

LISTA DE TABLAS

Tabla 1. Configuraciones geométricas, estructura cinemática, espacio de accesibilidad y ejemplos de robots industriales.....	21
Tabla 2. Comparación entre cámaras HD	23
Tabla 3. Descripción de variables del manipulador robótico modelo cinemático inverso ..	40
Tabla 4. Descripción de variables del manipulador robótico modelo dinámico	45
Tabla 5. Descripción de variables para hallar la inercia del electroimán	50
Tabla 6. Descripción de variables del manipulador robótico modelo dinámico	62
Tabla 7. Descripción de los engranajes de la caja reductora del servomotor	69
Tabla 8. Matriz de confusión para colores.....	80
Tabla 9. Matriz de confusión para formas	81

LISTA DE ANEXOS

ANEXO A. Datasheet cámara.....	87
ANEXO B. Datasheet servo motor CYS S0150.....	89
ANEXO C. Datasheet rodamiento KML 6010.....	90
ANEXO D. Datasheet rodamiento 6008 Z.....	91
ANEXO E. Datasheet electroimán	93
ANEXO F. Datasheet Arduino UNO.....	95
ANEXOS G. PLANOS	97

INTRODUCCIÓN

El ser humano siempre ha buscado tener la capacidad de controlar las actividades realizadas con mayor eficacia y facilidad en todos los medios que lo rodean. A nivel industrial, la mayoría de empresas cuentan con un sistema automatizado para tener un mejor control de calidad, un proceso de elaboración más rápido y reducir costos a largo plazo.

Una de las formas como el hombre ha implementado el control industrial, ha sido asemejándolas a la manera como el ser humano recibe la información y hace las cosas; por eso uno de los grandes inventos en toda la humanidad ha sido la cámara, por medio de la fotografía analógica y posteriormente la fotografía digital [1].

El ser humano recibe el 50% de la información a través de los ojos [2], es decir, la mayoría de nuestras acciones están determinadas por estímulos que emite el cerebro, pero que este a su vez ha recibido unas señales de determinadas situaciones en el exterior enviadas a través del ojo. Por esta razón a nivel industrial se ha buscado controlar tareas por medio de cámaras para tener visión del proceso en todo momento.

El proceso de generar visión por computadora, se llama visión artificial. La visión artificial es una rama de la inteligencia artificial que permite la obtención, procesamiento y análisis de cualquier tipo de información especial obtenidas a través de imágenes digitales [1]. Actualmente la visión artificial se emplea en áreas manufactureras para el control de calidad y producción, ya que estos se pueden llevar acabo por medio de algoritmos de conteo, posición y reconocimiento de objetos que pueden ser medidos por una sola cámara, en esto se basa la visión artificial.

Por otra parte, las empresas utilizan en su línea de producción brazos robóticos o robots para el movimiento de piezas, o para el ensamble. Estos dispositivos deben estar previamente programados para su buen desempeño y deben tener una rutina lógica programada. Por medio de la visión artificial se desea controlar estos mecanismos para que el dispositivo tenga la capacidad de elegir cual es la siguiente acción a realizar, debido a la información que está recibiendo la cámara y debidamente procesada. Esto debe estar previamente programado, pero no debe tener una secuencia lógica, la idea es que la cámara decida dependiendo del objeto que se encuentra en escena.

1. PLANTEAMIENTO DEL PROBLEMA

A nivel industrial hoy en día existe la necesidad de tener operarios que puedan identificar los productos que se realizan en las empresas y poderlos distribuir todos en sus respectivos lugares.

Este problema se piensa solucionar con visión artificial, la cual identificará 2 piezas de una producción, estas llegarán de una etapa anterior una por una, para que la cámara pueda identificar la figura y se comunique con un brazo robótico el cual ordenará las figuras en cubículos especiales, dos para cada figura (cada figura tiene dos posibles colores), cada cubículo lo reconocerá la cámara por medio de coordenadas cartesianas.

Con la implementación de la visión artificial se evitarán inconvenientes en la organización de las piezas, ya que la cámara reconocerá la pieza y la llevará a su lugar respectivo, identificado previamente mediante coordenadas de la cámara.

1.1 ANTECEDENTES DE LA INVESTIGACIÓN

En Colombia y alrededor del mundo hay gran variedad de proyectos realizados con el tema de visión artificial, las investigaciones más relevantes en este tema y que son fundamentales para el proyecto planteado se inician presentando proyectos que ayudan al entendimiento de visión artificial y los pasos por los cuales se adquiere una imagen.

A continuación, se describen 13 proyectos realizados alrededor del mundo, de los cuales se extrajeron las características más importantes para la realización de este proyecto, se enunciarán títulos y breve descripción dada por el autor de los mismos.

Se presenta un proyecto en Querétaro, México, en el cual, mediante la adquisición de imágenes con una cámara digital, se requiere orientar a un robot manipulador KUKA de 6 grados de libertad para tareas de ensamblaje, este sistema es la base para una celda de manufactura en la misma ciudad [3].

Luego de adquirir las imágenes necesarias en cualquier proyecto de visión artificial se deben caracterizar dichas imágenes, en la Universidad Antonio Nariño, sede Ibagué se realizó un proyecto con título *“Estudio Comparativo de Técnicas Espaciales para la Identificación de Defectos en Textiles”*. Este artículo evalúa tres técnicas espaciales con descriptores estadísticos, para la extracción de características de diferentes imágenes de una fábrica de textiles para recolectar la información necesaria para detectar defectos de los mismos, pasando por una red neuronal para detectarlos [4].

Los proyectos de visión artificial pueden ser variados. Cualquier tipo de imagen se puede procesar e identificar ciertos patrones que caracterizarán esta imagen y dar una conclusión a un posible usuario como por ejemplo en el trabajo sobre el *“Sistema de Visión por Computadora para Control Industrial”* aplicado a la medición del color de forma automatizada realizado en la EUPM, en Barcelona, en este proyecto se realizó la medición pixel a pixel de una imagen, mediante un programa de computador diseñado con base en teoría de visión artificial, utilizando el sistema cartesiano (XYZ), ya que se intenta suplir la intervención de operarios en control industrial, y el sistema CIELAB [5].

Para el desarrollo de este proyecto es necesario tener en cuenta otro tema como lo es la robótica y poder enlazar los dos temas en uno solo, es decir encontrar la forma de comunicar la cámara con un robot o en este caso brazo robótico. Alrededor del mundo ya se han realizado trabajos de investigación enlazando estos dos temas.

En el artículo *“Aplicación de técnicas de correspondencia para determinar la posición del efector final de un manipulador robótico”* se evalúan tres tipos de técnicas de Matching, para hallar la posición final del último eslabón de un brazo robótico de cinco grados de libertad e incluir la posición de todos los objetos que rodean el brazo robótico, lo anterior realizando realimentación de imágenes para aplicar la técnica de control más efectiva [6].

Como se puede observar existen varias formas de adquirir imágenes y de comunicar las cámaras con un robot, en la Escuela Politécnica del Ejército en Latacunga, Ecuador, se realizó el proyecto *“Diseño e implementación de un proceso de envasado de agua empleando el brazo robótico Scorbote-er 4u Intelitek del laboratorio de robótica sincro y servos de la sede Latacunga”*. La cámara interactúa con el brazo robótico para realizar el análisis de una imagen comparándola con un modelo patrón, y esta se configuró para obtener una mejor resolución y calidad de la imagen. Para cumplir el objetivo del proyecto primero se posicionan las botellas previamente en un dispensador, tomando en cuenta y determinando el área de trabajo del brazo robótico con respecto a la ubicación de la cámara, porque es importante saber la limitación del espacio para el movimiento del robot, esto para la sujeción de las botellas desde el dispensador a la cámara donde se tomará una foto de la etiqueta de la botella para la discriminación de esta. La discriminación se refiere a la clasificación de las botellas según la etiqueta que estas contengan. Cuando el brazo robótico toma una botella del dispensador la cámara toma una foto y compara con la imagen patrón, si equivale a la misma imagen la envía por una banda transportadora para ser envasada, de lo contrario la envía a desechar si es una botella incorrecta [7].

Para poder adquirir las imágenes existen muchos softwares en los cuales se pueden trabajar. En la tesis *“Aplicación de Visión con LabVIEW para la Detección de Frascos Sin Tapa”*, la cual forma parte del tópico de graduación *“Plataforma para el desarrollo de proyectos con DSPICs y visión robótica con LabView”*, el

objetivo principal fue desarrollar un sistema de inspección automática utilizando técnicas de Visión Robótica que ofrece el paquete de visión NI IMAQ de LabView versión 8.5 compatible con Cámaras USB y el módulo Vision Development de National Instruments. Este programa brindó una aplicación práctica de la adquisición de datos mediante una cámara analizando imágenes y evaluando las características físicas del objetivo. Para este propósito es necesario disponer de imágenes o plantillas capturadas de un frasco modelo con y sin tapa, las cuales sirvieron como base en imágenes de mayor tamaño captadas de los frascos evaluados [8].

Para desarrollar este proyecto hay que tener en cuenta el control de un brazo robótico, por ende, el apoyo de un trabajo de maestría de la UNEXPO, *“Control de Brazo Robótico”*, en el que se desarrolla la comparación de controladores clásicos PID con el modelo ideal obtenido de la dinámica de LaGrange, para realizar el control óptimo de un brazo robótico de 4 grados de libertad, resulta esencial para la optimización de este proyecto [9].

Otra de las características cambiantes de los proyectos realizados en visión artificial y robótica es el método para controlar el robot. Un ejemplo de esto es el proyecto de investigación *“Seguimiento Autónomo de la Posición de un Objeto por Visión y Control Neurodifuso en MATLAB”* este artículo presenta el desarrollo de un sistema de visión artificial y control difuso, el cual tiene como objetivo principal realizar el seguimiento, con una cámara web, de un objeto de manera automática, todo esto montado en una estructura con un motor paso a paso controlado. La anterior implementación es la primera etapa para realizar la navegación autónoma de un robot que pueda cumplir varias tareas [10].

Otros trabajos de investigación abarcan el tema de visión artificial pero enlazado con robótica móvil como por ejemplo el proyecto de investigación *“Navegación de robot móvil usando Kinect, OpenCV y Arduino”*, este proyecto muestra los

resultados obtenidos de la investigación realizada de visión artificial, para apoyo a la navegación, teniendo en cuenta imágenes de profundidad y reconocimiento de objetos a través del sistema de sensor Kinect de Microsoft y el sistema Arduino [11].

El siguiente proyecto se relaciona con el antes mencionado ya que se desarrollan con algoritmos de programación para realizar el procesamiento de imágenes, el cuál es un proceso fundamental para el trabajo con cámaras.

En el artículo *“Control numérico de un brazo robótico, como respuesta a algoritmos de procesamiento de imágenes y sistemas expertos”*, se desarrolla un sistema de control para un brazo robótico de cinco grados de libertad, que tenga la capacidad de desempeñarse en el juego Tic-Tac-Toe (triqui), el brazo robótico está dotado de capacidad visual mediante una cámara para realizar los movimientos necesarios del juego y posee un algoritmo de inteligencia artificial diseñado para tener capacidad de planificar jugadas para ganar, todo lo anterior fue desarrollado en la plataforma de programación LabVIEW [12].

Otro trabajo que abarca el mismo tema de control de brazo robótico es el del *“Desarrollo de un instrumento virtual para el control remoto de un brazo manipulador de uso educacional”*, en este proyecto se enuncian los resultados obtenidos con respecto al diseño, desarrollo e implementación de un programa de computador, en el cual se obtenga el modelo cinemático inverso y directo de un brazo robótico de cinco grados de libertad, con el fin de que sea utilizado para determinar límites físicos y de operación para controlar de forma remota un brazo robótico [13].

En el proyecto *“Desarrollo de un sistema de control para un brazo robótico mediante adquisición y procesamiento de imágenes [14]”*, se presentan las fases de diseño y construcción, para un sistema de control en lazo abierto, con

procesamiento de imágenes, para un brazo robótico que posea la capacidad de transportar distintas piezas de un lugar determinado a otro igual y poder darles un orden predeterminado a dichos elementos.

2. JUSTIFICACIÓN

Hoy en día la producción es el punto de mira de todas las industrias, la calidad y la elaboración rápida de las etapas y/o procesos en la cadena de producción de cualquier producto o servicio que se ofrezca. Este proyecto propone una rápida implementación en cuanto a la selección y organización de las figuras, o productos que se deseen implementar. Es un proceso de automatización que presenta beneficios debido a que solo se necesitaría una supervisión por parte de un operario y no necesariamente para realizar una tarea, sino para estar pendiente de que el trabajo realizado por el brazo robótico sea el adecuado.

3. OBJETIVOS DE LA INVESTIGACIÓN

3.1 OBJETIVO GENERAL

Diseñar y construir un brazo robótico orientado por visión artificial para clasificar 2 piezas de una producción.

3.2 OBJETIVOS ESPECÍFICOS

- Desarrollar un algoritmo para el reconocimiento de las piezas a clasificar y los cubículos donde se depositarán.
- Diseñar y construir el brazo robótico que tenga la capacidad de recoger objetos para moverlos en un área determinada hasta encontrar el cubículo en el cual se depositará.
- Diseñar el control de posición del brazo que garantice la correcta ubicación en el cubículo correspondiente.

3.3 LIMITACIONES

- Las piezas a clasificar son dos figuras geométricas básicas (círculos y cuadrados) cada uno de 2 colores (rojo y azul).
- La cámara no debe cambiarse de la posición inicial debido a que cualquier cambio de posición afecta el proceso de reconocimiento.
- El funcionamiento del manipulador robótico es en un semicírculo debido al movimiento de los motores y al área que este abarca.
- La clasificación de las piezas no se debe realizar cercana a la base del manipulador.

4. MARCO REFERENCIAL

En este capítulo se enuncian y exponen los conocimientos conceptuales y teóricos, los cuales son la base principal para el desarrollo total del proyecto de investigación.

4.1 MARCO CONCEPTUAL

4.1.1 VISIÓN ARTIFICIAL:

Es un campo de la *“Inteligencia Artificial”* en el cual se extrae información y características esenciales, de una imagen digitalizada, utilizando técnicas de procesamiento y análisis, con el fin de imitar la percepción que poseen los humanos [1].

4.1.2 MANIPULADORES ROBÓTICOS:

La definición más aceptada con respecto a que es un manipulador robótico es la otorgada por la Asociación Francesa de Normalización (AFNOR), la cual define al manipulador como:

“Un mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico” [15].

4.2 MARCO TEÓRICO

En esta sección, se expone la teoría con respecto a las etapas que requiere una aplicación en Visión Artificial, las características principales de los manipuladores robóticos y su forma de clasificación, con el fin de enlazar los dos temas y luego desarrollar los tópicos principales para realizar la solución del problema planteado para este proyecto.

4.2.1 Etapas para una aplicación en Visión Artificial

Para comprender más como se realizan las diferentes aplicaciones de Visión Artificial, se explicarán unas etapas generales, que no siempre se abarcan con totalidad, pero algunas son indispensables para poder realizar una aplicación ideal.

La primera etapa es la **captura de la imagen**, no es el simple hecho de capturar una imagen de cualquier manera, se debe realizar la mejor captura posible, con los mejores elementos para poder realzar las características, que se necesitarán para la aplicación que se esté realizando. Para realzar estas características hay que tener en cuenta, que tipo de Cámara se está utilizando, el ambiente en el que se encuentra dicha cámara, la iluminación que posee el espacio en donde se realizará la captura. De igual manera, si la aplicación se realizará en tiempo real con video todas las anteriores características son primordiales para poder efectuar un buen proyecto [2].

La segunda etapa es el **pre-procesado de la imagen**, esta etapa tiene como objetivo mejorar la calidad de la imagen, para poder extraerle las características que sean consideradas relevantes, esta etapa incluye operaciones de mejora de la relación señal-ruido, filtrar la imagen para que el entorno no entorpezca la extracción de características y realzar los datos necesarios y relevantes de la imagen como bordes y contornos [2].

La **Figura 1** detalla un diagrama de las etapas y su orden para realizar en una aplicación básica de Visión Artificial.

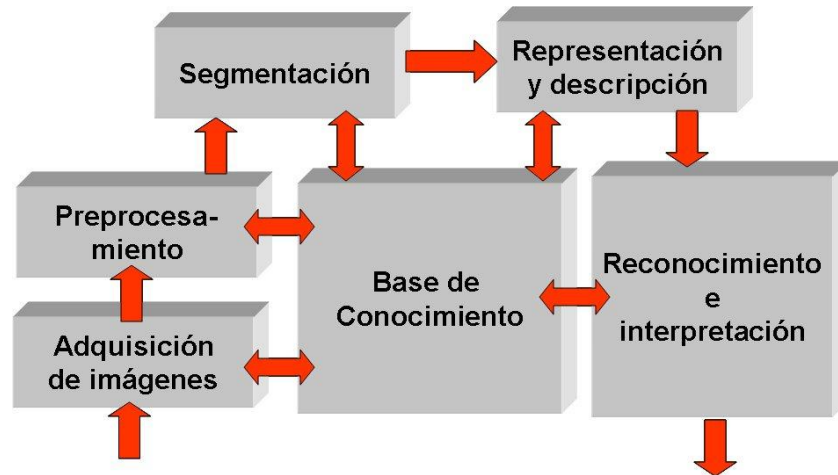


Figura 1. Etapas de una aplicación de visión artificial [2].

En el siguiente capítulo se definen las características y procesos que serán necesarios utilizar para este proyecto.

La tercera y última etapa, trata de **relacionar las características extraídas en la imagen con alguna acción o movimiento**, las cuales se pueden realizar con un sistema embebido conectado al computador para enviar una señal para que realice la acción.

4.2.2 Características y clasificación de manipuladores robóticos

Los manipuladores robóticos están constituidos por un grupo de elementos rígidos denominados eslabones, que se encuentran conectados entre sí mediante articulaciones, las cuales permiten el movimiento relativo de cada dos eslabones seguidos [15].

La **Figura 2** detalla los elementos principales de un manipulador robótico.

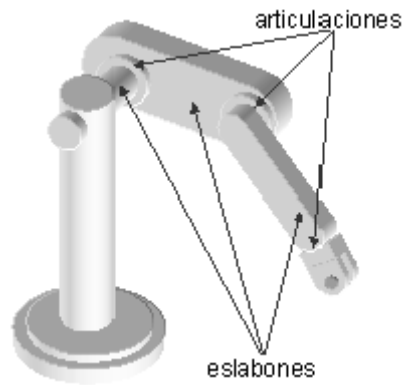


Figura 2. Elementos de un manipulador robótico [15].

Las articulaciones pueden ser lineales y rotacionales, según el movimiento que induzcan al eslabón [15].

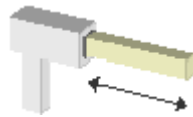


Figura 3. Articulación lineal [15].

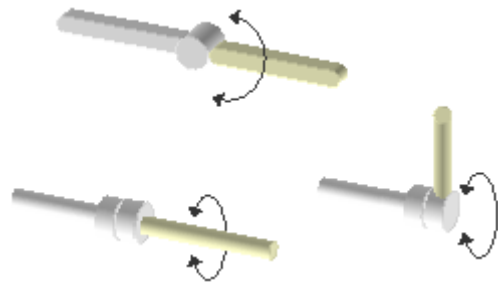


Figura 4. Articulaciones rotacionales [15].

Un conjunto de eslabones con sus respectivas articulaciones se denomina cadena cinemática. Una cadena cinemática es abierta, si la conexión de eslabones mediante articulaciones es lineal, es decir el primer eslabón está conectado al soporte y al segundo eslabón, el segundo eslabón está conectado al primero y al tercero y así sucesivamente sin intersección de eslabones, hasta el último eslabón cuyo extremo queda libre, al cual se le puede conectar algún elemento final dependiendo la aplicación [15].

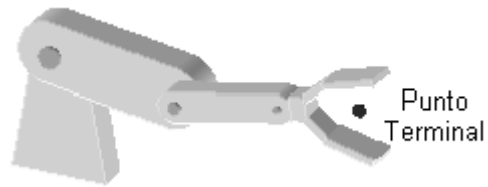


Figura 5. Punto terminal de un manipulador con pinza [15].

Los manipuladores robóticos más complejos son conocidos como brazos robóticos ya que poseen los movimientos característicos de los mismos.

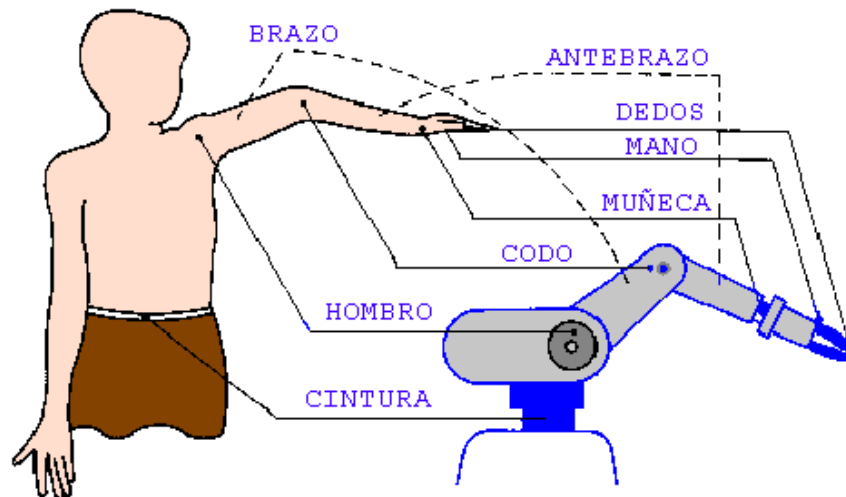


Figura 6. Analogía entre manipulador (brazo robótico) y brazo humano [15].

La relación de los movimientos que puede poseer el manipulador se denomina grados de libertad, lo que indica que dependiendo del número de grados de libertad que posea un manipulador robótico, es la cantidad de movimientos en diferentes planos de coordenadas independientes [15].



Figura 7. Distintos Grados de Libertad [15].

La mejor manera de describir el estado de un manipulador, es determinando la posición final del último eslabón, que es donde se encuentra el punto terminal, respecto a un sistema de coordenadas externos y fijo, denominado sistema mundo y determinando el movimiento que realiza cada eslabón dependiendo del actuador que aplica las fuerzas y momentos en cada uno de los eslabones [15].

Los tipos de análisis mecánicos que se le pueden realizar a un manipulador robótico son de tipo cinemático y dinámico, los cuales son con respecto a fuerza y momentos (cinemático) y con respecto a los movimientos realizados por el conjunto de eslabones totales (dinámico) [15].

Según la geometría de la estructura mecánica de los manipuladores robóticos se pueden clasificar de la siguiente manera:

Cartesiano: es el manipulador que se caracteriza por solo poseer movimientos lineales, por sus articulaciones del mismo tipo.

Cilíndricos: poseen mínimo tres eslabones, uno para movimiento rotacional que está anclado a una base y los otros eslabones poseen movimientos lineales.

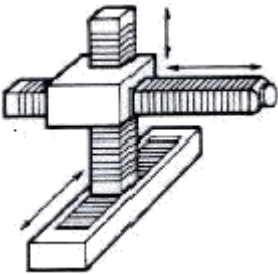
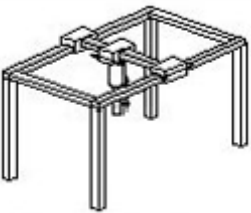
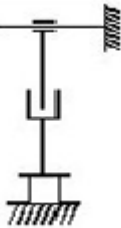
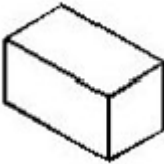
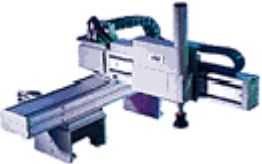
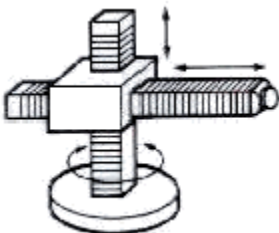
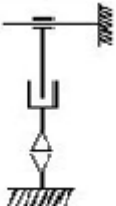


Polar: posee tres articulaciones, dos rotacionales y una lineal.

Esférico (brazo articulado): consta de tres articulaciones rotacionales.

Mixto: es una combinación de cualquiera de las anteriores configuraciones.

Otras formas de clasificar los manipuladores robóticos son según sus grados de libertad (movimientos en planos de coordenadas independientes), espacio de trabajo, capacidad de posicionamiento del punto terminal, capacidad de carga y velocidad.

En la **Tabla 1** se pueden observar los distintos tipos de caracterización que tienen los manipuladores robóticos.

Configuración geométrica	Estructura cinemática	Espacio de trabajo	Ejemplo
<p>cartesianos</p>  <p>tipo cantiléver</p>  <p>tipo pórtico</p>			
<p>cilíndrico</p> 			





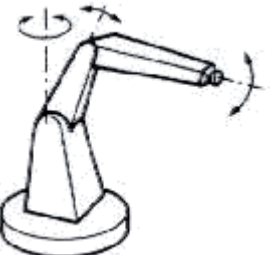



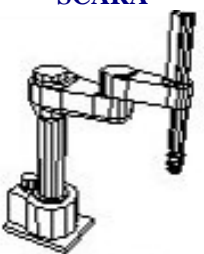
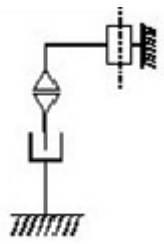
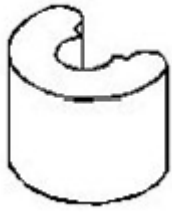


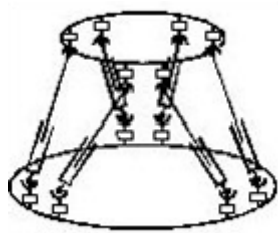


<p>polar</p> 			
<p>esférico</p> 			
<p>SCARA</p> 			
<p>paralelo</p> 			

Tabla 1. Configuraciones geométricas, estructura cinemática, espacio de accesibilidad y ejemplos de robots industriales [15].

En el siguiente capítulo se desarrollará la combinación de los dos tópicos anteriormente expuestos; Visión Artificial y Manipuladores Robóticos, que permiten la consecución de los objetivos planteados.

5. DISEÑO INGENIERIL

5.1 SOFTWARE Y LIBRERÍAS

El software utilizado para este proyecto es QT CREATOR, el cual es un ambiente de desarrollo integrado multiplataforma creado por Trolltech de uso libre para desarrollar aplicaciones con el uso de las bibliotecas Qt, la programación en QT CREATOR se realiza en el lenguaje de programación C++ [16].

Las librerías de Open CV son bibliotecas de uso libre para aplicaciones de Visión Artificial desarrolladas en su origen por Intel y contiene más de 500 funciones con respecto a temas de Visión Artificial. Estas abarcan muchas áreas en visión, incluyendo inspecciones de producción en empresas, aplicaciones médicas, seguridad, interfaz de usuario, calibración de cámara, visión estéreo y robótica [17].

La librería está escrita en C y C++ y funciona en los sistemas operativos de Linux, Windows y Mac OS X. Actualmente se desarrolla en interfaces para Python, Ruby, Matlab y otros lenguajes. Fue diseñada para eficiencia computacional y con un fuerte enfoque en aplicaciones en tiempo real [17].

Una de las metas de Open CV es proporcionar infraestructura de visión por computadora fácil de usar, que ayude a las personas a diseñar aplicaciones de visión rápidamente [17].

5.2 VISIÓN ARTIFICIAL

5.2.1 ADQUISICIÓN DE IMAGEN:

Para la adquisición de la imagen se empleó una cámara Logitech HD Webcam C270, con una resolución de 1280 x 720 pixeles en formato RGB (**Ver Figura 8**).



Figura 8. Logitech HD Webcam C270.

Fuente: <http://www.logitech.com/es-roam/product/hd-webcam-c270>

Esta cámara fue escogida teniendo en cuenta varias opciones y se decidió usar la cámara Logitech HD Webcam C270 por su bajo costo, a continuación, se ilustra una comparación entre la cámara escogida y otra cámara que se encuentra en el mercado.

Logitech HD Webcam C270	Cámara Hd Genius Facecam 2020
<ul style="list-style-type: none"> • Videoconferencias HD (1280 x 720 píxeles) con el sistema recomendado • Captura de vídeo: Hasta 1280 x 720 píxeles • Tecnología Logitech Fluid Crystal™ • Fotos: Hasta 3.0 megapíxeles (mejora por software) • Micrófono integrado con reducción de ruido • Certificación USB 2.0 de alta velocidad (se recomienda) • Clip universal para monitores LCD, CRT o portátiles • Velocidad máxima de cuadro 30 fotogramas por segundo 	<ul style="list-style-type: none"> • Interfaz: USB. • Resolución: 1280 x 720. • Sensor: CMOS. • Velocidad de captura de vídeo: 15 fotogramas por segundo. • Resolución: 2 megapíxeles.
\$70.000	\$86.600

Tabla 2. Comparación entre cámaras HD.

Como se puede observar la cámara Logitech tiene características comunes con otras cámaras con la misma resolución en el mercado, pero su precio es más bajo en comparación con la cámara Genius.

5.2.2 DETECCIÓN DE COLORES RGB

Cada imagen capturada por la cámara contiene tres matrices, una para cada color (R, G, B), las posiciones de cada matriz representan el nivel de color del pixel correspondiente en una escala desde 0 hasta 255 (**Figura 9. Matrices RGB de una imagen Roja**); esto quiere decir, por ejemplo, que si la posición (1,1) de la matriz R es 255, mientras que para la matriz B y para la G es 0, el píxel correspondiente a esta posición es únicamente rojo; de esta manera para valores intermedios se presenta la construcción de toda la gama de colores. Teniendo en cuenta lo anterior, la identificación del color se determinó calculando el promedio de los valores de cada matriz, si el color predominante en la imagen es el rojo, el promedio de los valores de la matriz R, será evidentemente mayor que el de la matriz G y la matriz B, de esta forma es posible determinar el color de la forma a reconocer dado que el fondo es negro. La razón para que, por ejemplo, en una imagen roja como la representada por las matrices de la **Figura 9** no todos los pixeles de R sean 255 ni todos los de las demás figuras sean 0, es que la imagen puede no ser roja completamente debido a variaciones en el color de la figura o por efectos del proceso de adquisición de la imagen.

R	200	240	240	225	...
	218	236	240	240	...
	215	237	245	245	...
	240	238	250	240	...

G	0	5	0	3	...
	0	5	2	2	...
	2	2	0	1	...
	4	0	1	4	...

B	1	3	0	2	...
	0	5	2	1	...
	2	1	0	1	...
	4	2	1	0	...

Figura 9. Matrices RGB de una imagen Roja.

Fuente: Autores

5.2.3 UMBRALIZACIÓN

Para llegar al objetivo de reconocer las formas finales, se requiere umbralizar la imagen, esto es, representarla en dos colores (blanco y negro), para esto el primer paso es convertir la imagen original en colores a escala de grises, lo que se consigue eliminando la información de tono y saturación, pero conservando la luminancia [18].

Uno de los métodos para utilizar un valor umbral de manera automática es conocido como el método de Otsu [19]. Este método utiliza técnicas estadísticas para conocer la varianza, midiendo la dispersión de los niveles de gris que se presentan en la imagen.

El método consiste en que haya la menor dispersión posible dentro de cada segmento, pero que a su vez haya la mayor dispersión posible entre segmentos diferentes, hallando automáticamente un valor umbral óptimo en el cual el cociente entre ambas varianzas sea el máximo y, por lo tanto, separando los objetos de nuestro interés del resto en una misma imagen.

Para el desarrollo del método de otsu es necesario partir de una imagen en escala de gris con N pixeles y L niveles diferentes.

La probabilidad de ocurrencia del nivel de gris i en la imagen está dada por la ecuación:

$$P_i = \frac{f_i}{N}$$

Donde f es la frecuencia de repetición del nivel de gris i-ésimo con $i = 1, 2, \dots, L$.

En el caso particular de realizar binarización, los píxeles se dividen en dos clases, C_1 y C_2 , con niveles de gris $[1, 2, \dots, t]$ y $[t + 1, t + 2, \dots, L]$ respectivamente donde las distribuciones de probabilidad de ambas clases son:

$$C_1: \frac{P_1}{\omega_1(t)}, \dots, \frac{P_t}{\omega_1(t)}$$

$$C_2: \frac{P_{t+1}}{\omega_2(t)}, \dots, \frac{P_{t+2}}{\omega_2(t)}, \dots, \frac{P_L}{\omega_2(t)}$$

Dónde:

$$\omega_1(t) = \sum_{i=1}^t P_i$$

$$\omega_2(t) = \sum_{i=t+1}^L P_i$$

Las medias para cada clase se definen como:

$$\mu_1 = \sum_{i=1}^t \frac{i \cdot P_i}{\omega_1(t)}$$

$$\mu_2 = \sum_{i=t+1}^L \frac{i \cdot P_i}{\omega_2(t)}$$

La intensidad media total de la imagen se define y se demuestra que:

$$\omega_1 \cdot \mu_1 + \omega_2 \cdot \mu_2 = \mu_T$$

$$\omega_1 + \omega_2 = 1$$

Otsu definió la varianza entre clases mediante el uso de un análisis discriminante y lo expresó en la siguiente ecuación:

$$\sigma_B^2 = \omega_1 \cdot (\mu_1 - \mu_T)^2 + \omega_2 \cdot (\mu_2 - \mu_T)^2$$

Luego de esto, otsu demostró que el umbral óptimo t es el siguiente:

$$t^* = \text{Max}\{\sigma_B^2(t)\}$$

Donde:

$$1 \leq t \leq L$$

Las imágenes 10 y 11 muestran la imagen original capturada por la cámara y la misma convertida a escala de grises respectivamente.

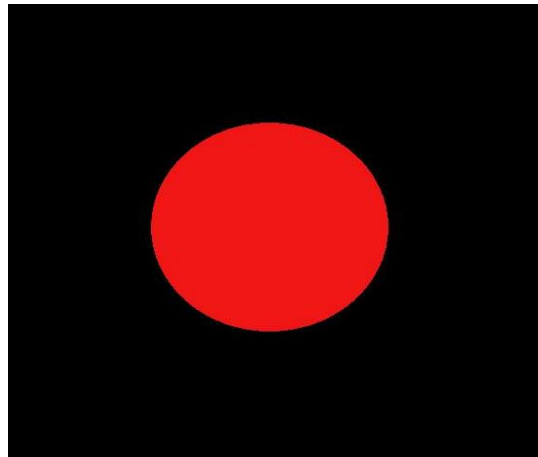


Figura 10. Imagen original.

Fuente: Autores

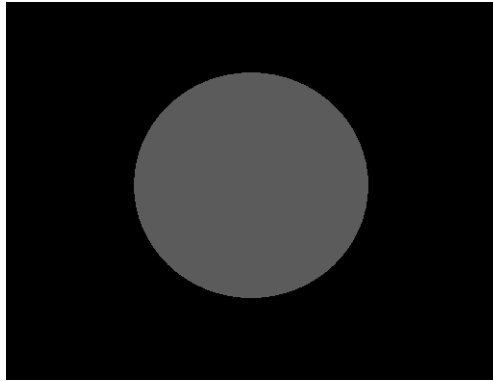


Figura 11. Imagen en escala de grises.

Fuente: Autores

Después de tener la imagen en escala de grises, se requiere realizar un proceso de umbralización de la imagen, para ello el método de otsu analiza el histograma de la imagen gris, el cual se ilustra en la **Figura 12** y determina automáticamente el umbral óptimo para obtener la imagen umbralizada (**Ver Figura 13**).

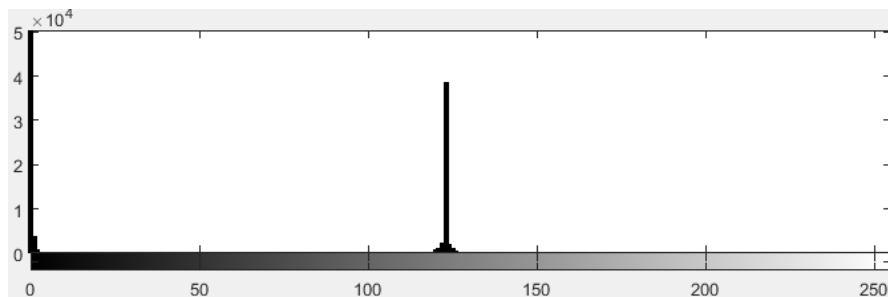


Figura 12. Histograma.

Fuente: Autores

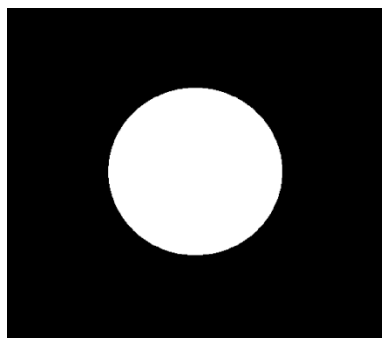


Figura 13. Imagen umbralizada.

Fuente: Autores

5.2.4 CONTORNOS

Luego de realizar un filtro al finalizar la binarización de la imagen para eliminar los defectos que esta pueda tener, se utiliza un algoritmo conocido como el detector de bordes de Canny, el cual sirve para detectar los mismos mediante un umbral, dejando únicamente estos de color blanco y la parte interna del objeto del mismo color del fondo (negro).

Al utilizar el algoritmo de Canny se obtiene una variable que almacena el número de bordes, mediante los cuales se pueden identificar las figuras a clasificar. Este método, a diferencia del algoritmo de Laplace, calcula en x y y las primeras derivadas, que luego son combinadas dentro de cuatro derivadas direccionales.

Estas direcciones son calculadas de la siguiente manera:

$$G = \sqrt{G_x^2 + G_y^2}$$

Donde:

G_x = derivada en la dirección vertical

G_y = derivada en la dirección horizontal

$$\Theta = \arctan = \frac{G_y}{G_x}$$

Los puntos en donde estas derivadas tienen el valor máximo son los pixeles que se unirán para formar los bordes. Los contornos son formados al aplicar un umbral de histéresis a los pixeles mencionados anteriormente. Esto significa que se tienen dos umbrales, uno alto y uno bajo. Si un pixel tiene un gradiente mayor que el umbral más alto, entonces es tomado como un pixel de borde; de lo contrario, si un pixel se encuentra por debajo que el umbral más bajo, entonces es rechazado. Si el gradiente de un pixel se encuentra entre los dos umbrales, entonces este

solo será aceptado si está conectado a un pixel que se encuentre por encima del umbral más alto [17]. (Ver Figura 14 y Figura 15).

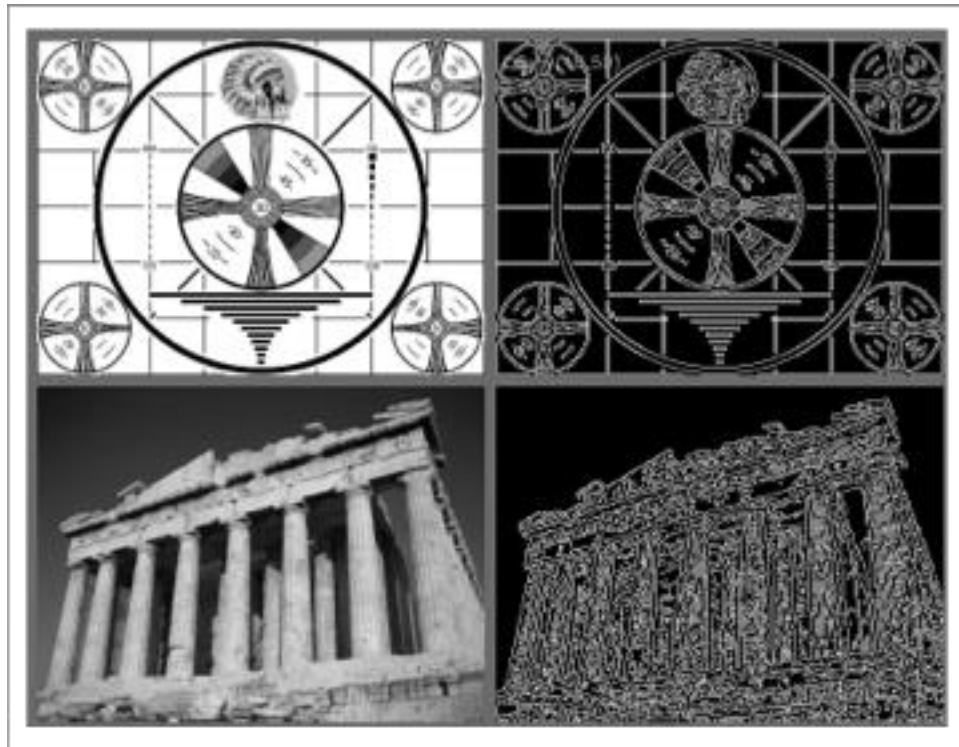


Figura 14. Ejemplo de resultados del detector de bordes de Canny para dos imágenes distintas cuando los umbrales alto y bajo son 50 y 10 [17].

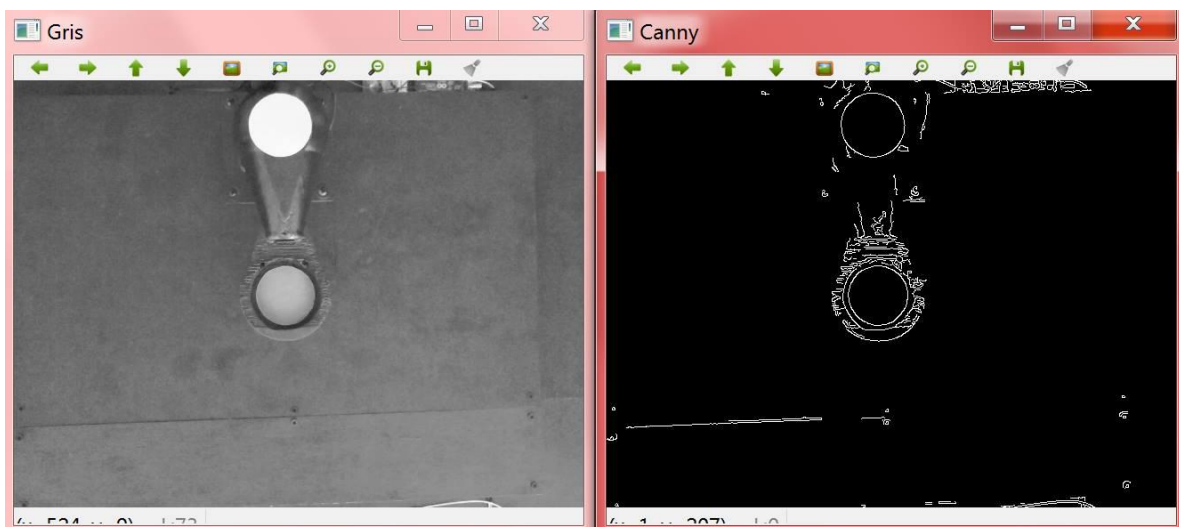


Figura 15. Ejemplo de resultados del detector de bordes de Canny para imagen capturada por la cámara.

Fuente: Autores

5.2.5 EXTRACCIÓN DE COORDENADAS CARTESIANAS DE LAS FIGURAS A CLASIFICAR

Para realizar el proceso de extracción de coordenadas cartesianas de la posición en la cual se encuentran las figuras a clasificar dentro del plano, se tuvo en cuenta que, al capturar una imagen con la cámara, se encontrarán varios objetos en la misma. Estos objetos deben ser reconocidos mediante técnicas de visión artificial, los objetos visibles en la imagen tomada corresponden a los cubículos donde se depositarán las piezas, las piezas a clasificar y el punto terminal del manipulador robótico. Al reconocer estos objetos, también es necesario determinar en qué posición del plano se encuentran. En la **Figura 16** se pueden observar tanto las coordenadas reales del centro del objeto, como las coordenadas calculadas del centro del objeto (texto al final de la imagen).

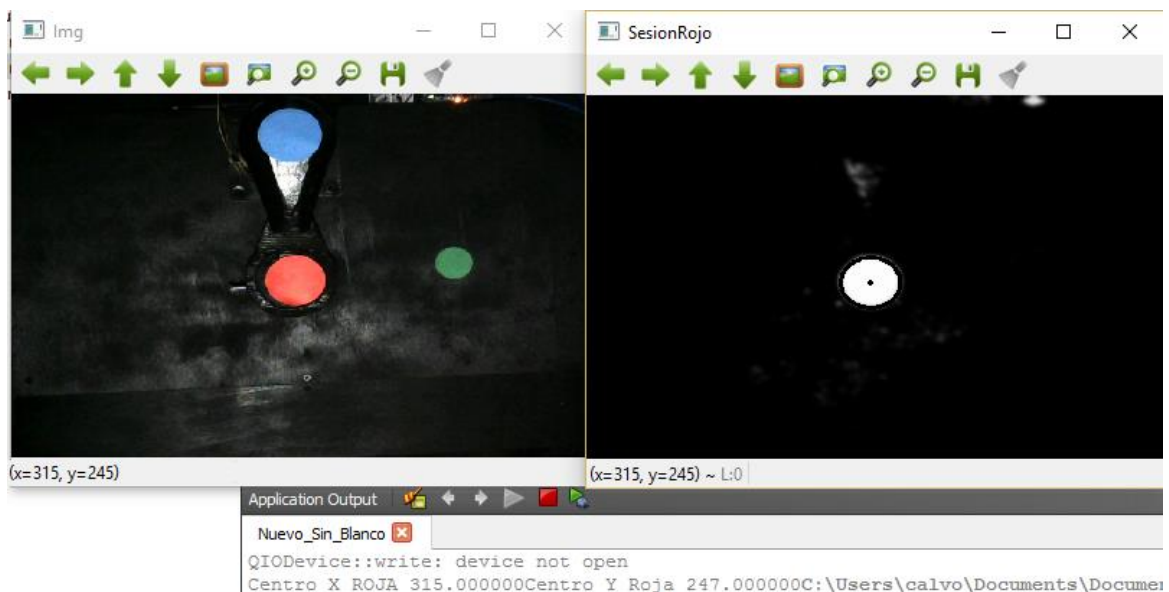


Figura 16. Coordenadas cartesianas del objeto reconocido en la imagen capturada.

Fuente: Autores

El primer paso para realizar una aplicación de visión artificial consiste en utilizar un umbral para determinar cuál de los colores RGB se había detectado, la figura 17 explica el proceso con el que se realizó el algoritmo. Posteriormente se binarizó la imagen con el fin de convertir los pixeles del color detectado en blanco y los

demás en negro. Luego de la binarización se recorren las matrices de izquierda a derecha empezando del punto (0,0) hasta finalizar en el punto (0,480), de allí se procede a recorrer la segunda fila comenzando en el punto (1,0) y se vuelve a recorrer de izquierda a derecha terminando en el punto (1,480), este proceso se repite hasta llegar al punto (640,480) que es el tamaño completo de la imagen y se cuenta el primer y último pixel de color blanco, estos valores se promedian para obtener las coordenadas del centro de la figura capturada. Finalmente, mediante el contorno de la imagen se detecta el número de bordes para determinar que figura es, teniendo en cuenta que si el número de bordes es igual a 4 es un cuadrado o si es mayor a 4 es un círculo (**Ver Figura 20**).

Las funciones de extracción de coordenadas cartesianas de las figuras a clasificar y detección de formas están detallados en los diagramas de flujo correspondientes a las Figuras 18 y 19 respectivamente.

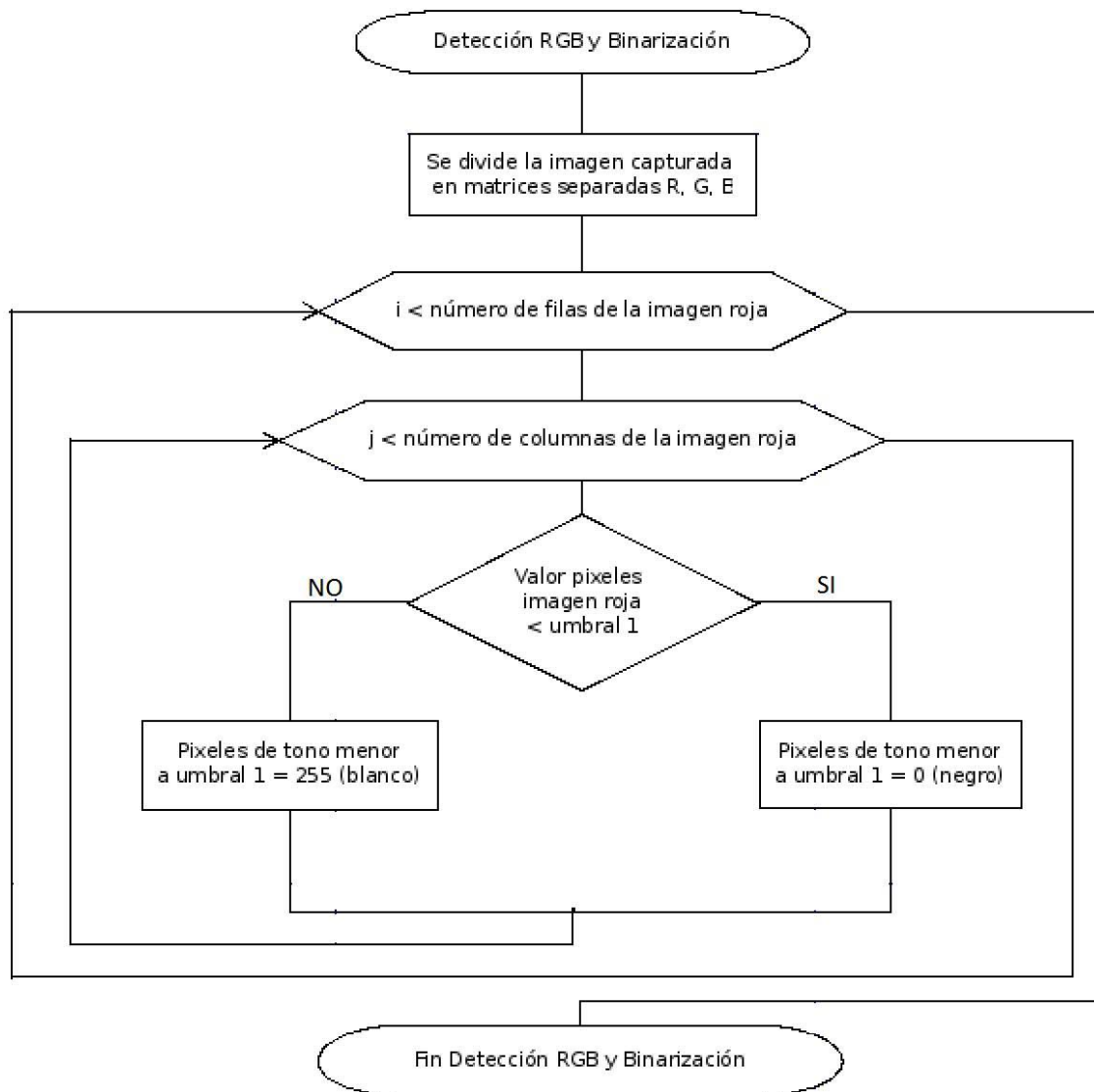


Figura 17. Diagrama de flujo RGB y Binarización.

Fuente: Autores

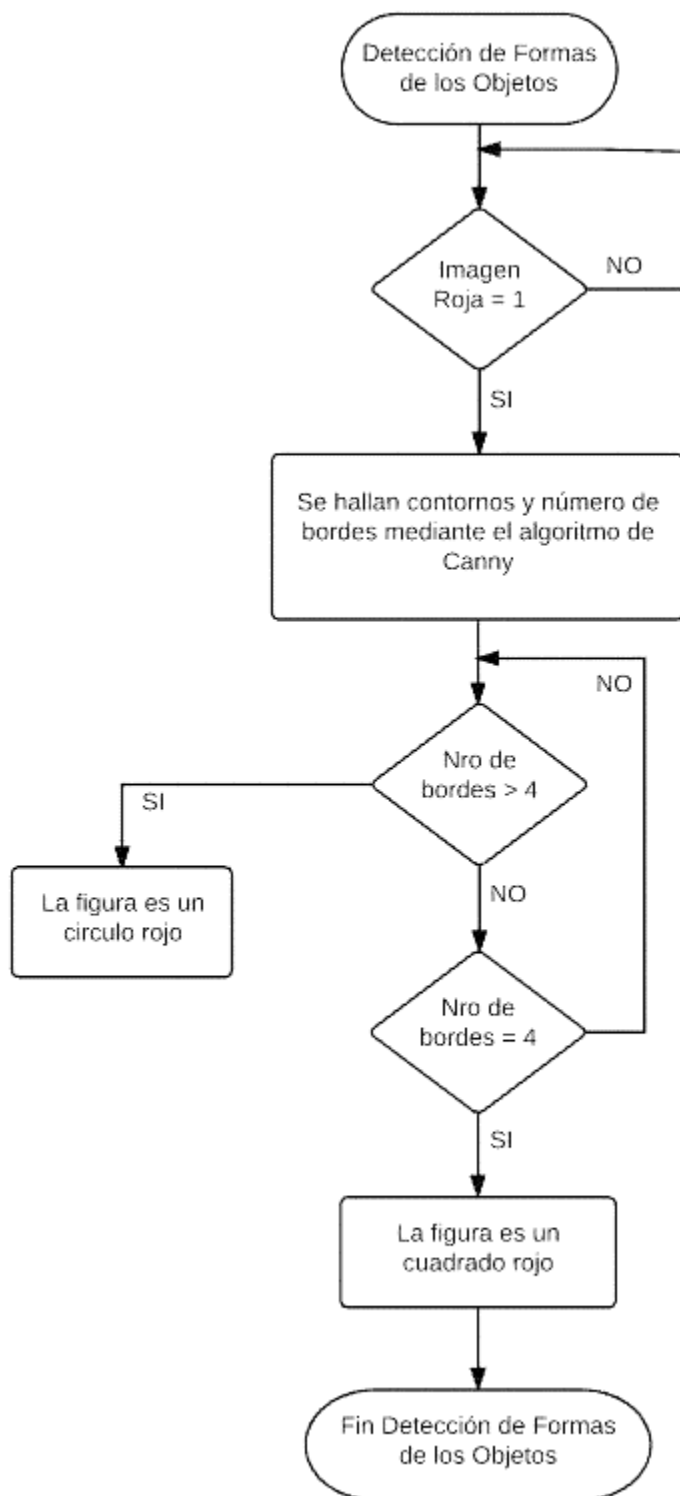


Figura 18. Diagrama de flujo de formas de los objetos.

Fuente: Autores

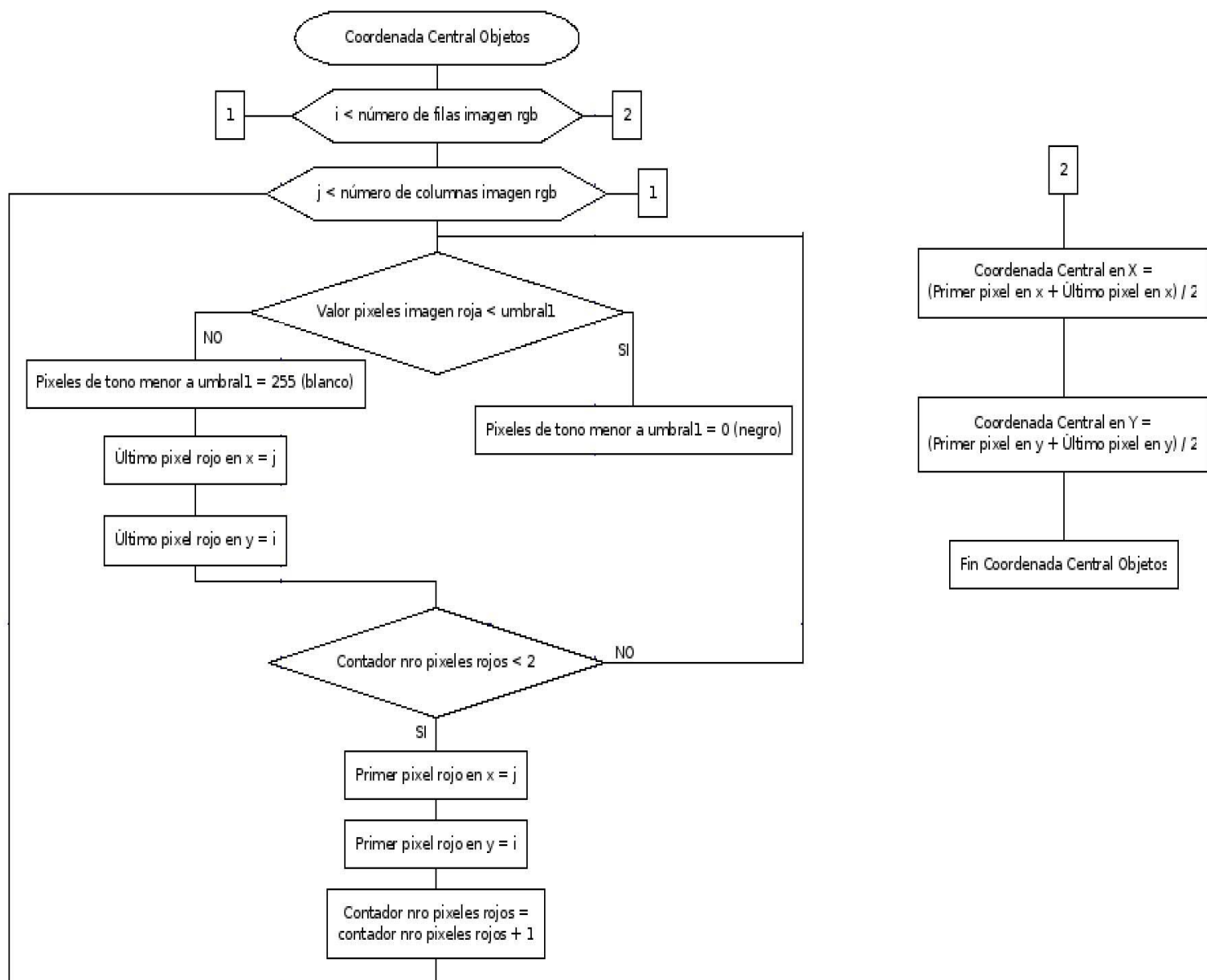


Figura 19. Diagrama de flujo de coordenada central objetos.

Fuente: Autores

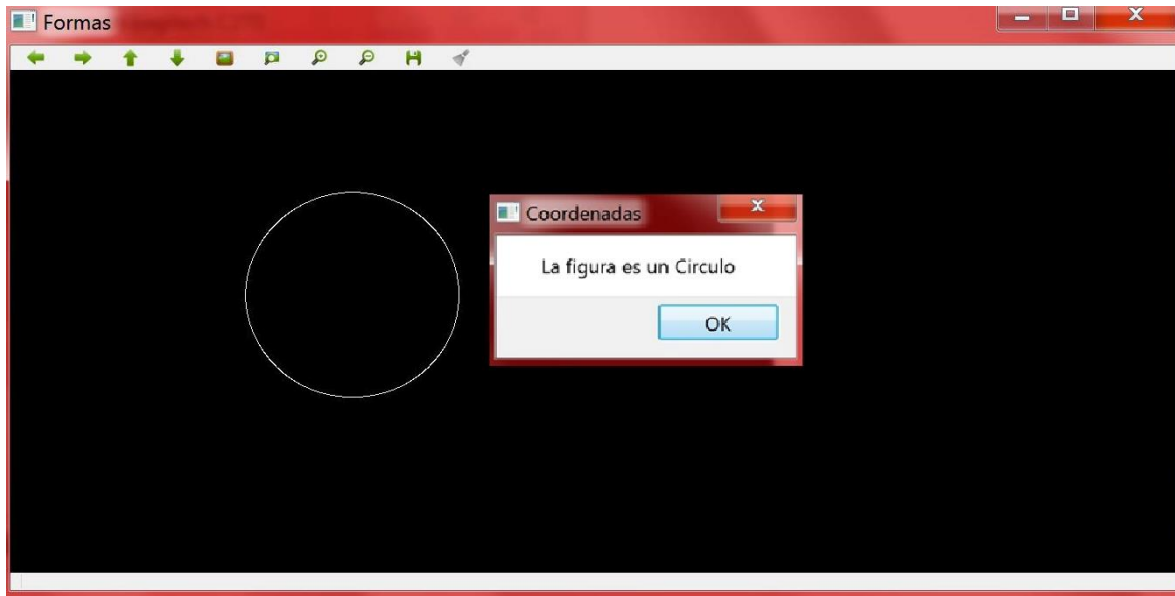


Figura 20. Círculo reconocido mediante detección de número de bordes.

Fuente: Autores

Las coordenadas de los centros de las figuras, son necesarias para realizar el control que será el encargado de organizar las figuras en sus cubículos correspondientes.

5.3 DISEÑO MECÁNICO MANIPULADOR ROBÓTICO

Teniendo en cuenta el marco teórico expuesto anteriormente se decidió diseñar un manipulador robótico de movimientos cartesianos ya que el manipulador es de dos grados de libertad y su área de trabajo está definida por la calidad de capturas que pueda tener la cámara. Para realizar este proceso se realizaron varias capturas y se determinó que la altura apropiada para que la cámara es de 75 centímetros en los cuales la cámara captura fotos de gran calidad y se tiene un área amplia de trabajo para el manipulador robótico.

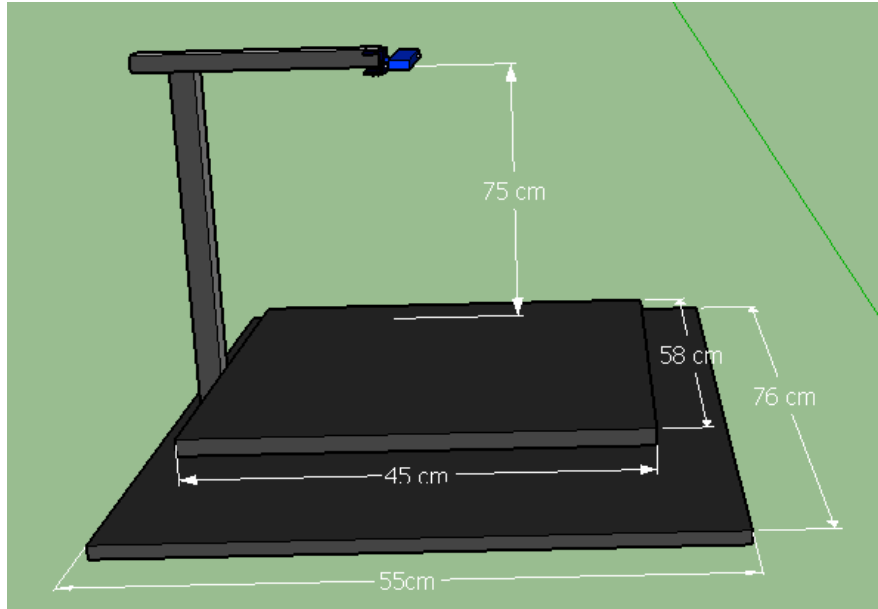


Figura 21. Bosquejo de la vista lateral de la maqueta.

Fuente: Autores

En la **Figura 21** se puede observar que la cámara está ubicada a 75 centímetros de una primera base en la cual se encuentra el brazo robótico y esta será el área de trabajo del mismo, también se observa una segunda base en la cual están ubicados los 4 cubículos y sirve para evitar que la cámara capture otros elementos que interfieran con la función.

5.3.1 Cinemática

La cinemática estudia el movimiento de sistemas sin tener en cuenta la fuerza ejercida para realizar dichos movimientos, como si lo hace la dinámica [21].

En aplicaciones con robots manipuladores se debe comprender que estos se mueven en el espacio, por ende, se le asignan sistemas de coordenadas, para representar posiciones y orientaciones, dependiendo de la posición inicial y final del efector, junto con la trayectoria que debe seguir el manipulador [22].

Con la cinemática directa se puede hallar la posición dependiendo de los ángulos de las articulaciones del manipulador robótico, mientras que, para hallar los

ángulos de las articulaciones con respecto a la posición, se aplica **cinemática inversa**. En el caso de este proyecto se quiere ubicar el brazo en una posición deseada dada por las coordenadas cartesianas de los objetos a clasificar por este motivo se requiere utilizar el método de **cinemática inversa**.

Se debe tener en cuenta que el modelo cinemático inverso es más complejo que el modelo cinemático directo, ya que para un manipulador robótico siempre es posible encontrar el modelo de cinemática directa, mientras que en la cinemática inversa pueden existir varias soluciones e inclusive no haber solución analítica alguna. En el caso de que no exista una solución analítica, se pueden aplicar los siguientes métodos: redes neuronales, métodos numéricos, iterativos o geométricos [21].

5.3.2 Cinemática Inversa

La cinemática inversa es un problema no lineal que relaciona los ángulos de las articulaciones en función de las coordenadas cartesianas y la orientación en la que se encuentra el extremo final del manipulador robótico [21].

Este modelo implica combinar las coordenadas extraídas con la cámara respecto al objeto que se va a clasificar, y los ángulos que requiere el manipulador para posicionar el efector final en el centro del objeto.

Para obtener el modelo cinemático inverso se pueden utilizar los siguientes métodos: métodos geométricos, matrices de transformación homogénea, desacoplamiento cinemático (robots de 6 GDL), entre otros.

Para poder realizar el modelo cinemático inverso se utilizó el método geométrico, ya que es el más adecuado para un manipulador robótico de dos grados de libertad, debido a que es un método fácil de implementar para manipuladores de 1 o 2 grados de libertad, con el cual conociendo las coordenadas en las cuales está

el objeto a clasificar, se determinarían los ángulos que debe poseer cada una de las articulaciones con respecto a sus ejes coordenados.

5.3.2.1 Métodos Geométricos

Este método fue utilizado para encontrar los ángulos necesarios en el manipulador robótico, con el fin de posicionarlo en el plano.

Para solucionar este método es necesario emplear relaciones trigonométricas y geométricas, como lo es la resolución de triángulos.

A continuación, se detalla la solución del modelo cinemático inverso realizado a partir de métodos geométricos, primero se ilustra en la **Figura 22** las variables que se tuvieron en cuenta en el manipulador.

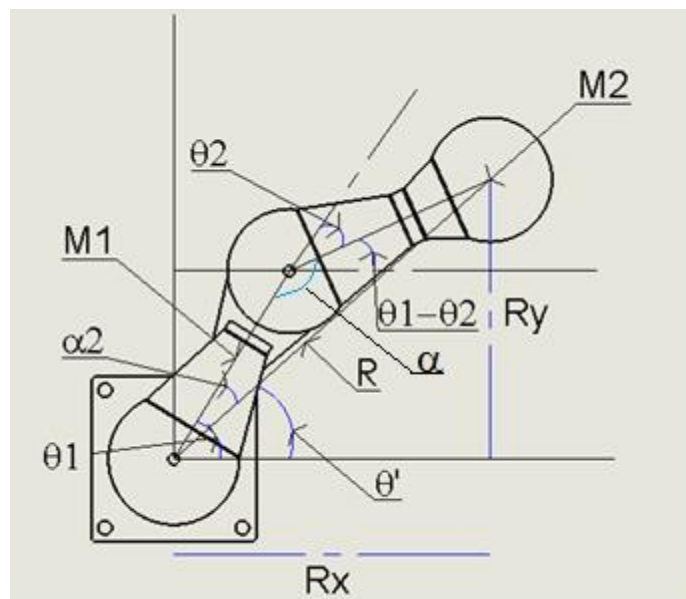


Figura 22. Análisis Cinemático Inverso.

Fuente: Autores

En la **Tabla 3** se observa la descripción de las variables con las cuales se realizaron los cálculos correspondientes a la cinemática inversa del manipulador robótico.

Variable	Descripción
M1	Longitud del Primer Eslabón
M2	Longitud del Segundo Eslabón
R	Magnitud del vector de la posición del objeto
R _x	Distancia en el eje x de la posición del objeto
R _y	Distancia en el eje y de la posición del objeto
θ_1	Ángulo de la primera articulación
θ_2	Ángulo de la segunda articulación
α	$180 - \theta_2$
α_2	Ángulo entre la Magnitud del vector de la posición del objeto y el Primer Eslabón
θ'	Ángulo entre la Magnitud del vector de la posición del objeto y el eje x

Tabla 3. Descripción de variables del manipulador robótico modelo cinemático inverso.

Para hallar los ángulos de las articulaciones se realizaron los siguientes cálculos, esto debido a que es necesario conocer el valor numérico de los ángulos θ_1 y θ_2 , para que el manipulador robótico logre llegar al objeto que se encuentra en el plano.

Estos ángulos hallados pertenecen a los grados que debe girar los servo motores, para poder general la trayectoria que debe seguir el manipulador robótico con el fin de desarrollar la aplicación propuesta.

Para comenzar a solucionar este método es necesario utilizar la ley de cosenos que se diagrama en la **Figura 23**, que representa el plano donde se ubican las piezas y en el cual los ejes representan la distancia en centímetros.

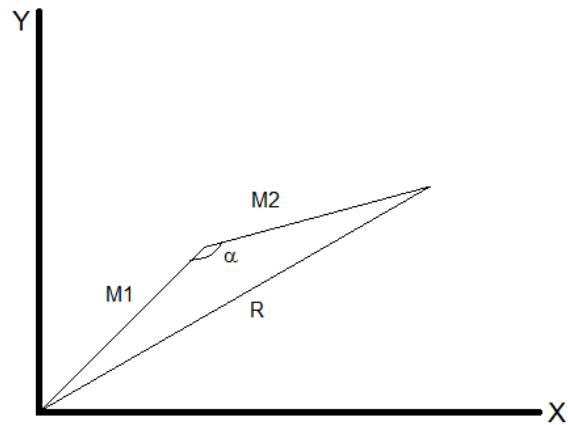


Figura 23. Análisis Ley de Coseno.

Fuente: Autores

$$a^2 = b^2 + c^2 - 2bc \cos A$$

$$R^2 = M_1^2 + M_2^2 - 2M_1M_2 \cos(\alpha)$$

Utilizando la identidad trigonométrica

$$\cos(a - b) = \cos(a) \cos(b) + \sin(a) \sin(b)$$

$$\cos(180 - \theta_2) = \cos(180) \cos(\theta_2) + \sin(180) \sin(\theta_2)$$

$$\cos(180 - \theta_2) = -\cos(\theta_2)$$

Reemplazamos en la primera ecuación

$$R^2 = R_x^2 + R_y^2 = M_1^2 + M_2^2 + 2M_1M_2 \cos(\theta_2)$$

Despejando el ángulo θ_2 de la anterior ecuación obtenemos:

$$\theta_2 = \cos^{-1} \left(\frac{Rx^2 + Ry^2 - M_1^2 - M_2^2}{2M_1M_2} \right)$$

Luego de hallar θ_2 se procede a despejar las otras variables en función de θ_2

Al conocer:

$$\alpha = 180 - \theta_2$$

Se aplica la Ley del Seno

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C}$$

$$\frac{R}{\sin(\alpha)} = \frac{M_2}{\sin(\alpha_2)}$$

$$\alpha_2 = \sin^{-1} \left(\frac{M_2 \sin(\alpha)}{R} \right)$$

$$\theta' = \tan^{-1} \left(\frac{Ry}{Rx} \right)$$

Se reemplaza en la ecuación de θ_1 y se halla en ángulo de la primera articulación.

$$\theta_1 = \alpha_2 + \theta'$$

La trayectoria que debe poseer el manipulador robótico queda definida por los ángulos θ_1 y θ_2 , siendo θ_1 el ángulo correspondiente a la primera articulación y θ_2 correspondiente a la segunda articulación, este θ_2 es la resta del θ_1 y θ_2 , para poder utilizar el eje coordenado ilustrado en la **Figura 24**.

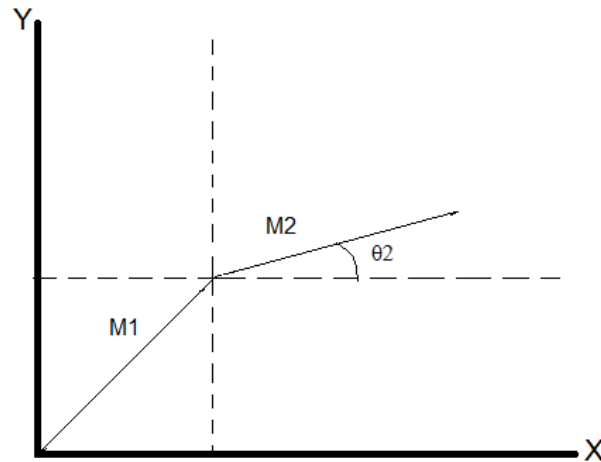


Figura 24. Representación ángulo θ_2 .

Fuente: Autores

Para poder comprobar que los ángulos que arrojan las ecuaciones son los necesarios para llegar a un punto deseado en el plano, se realizó un programa en Matlab para simular la obtención de los mismos y el posicionamiento del manipulador (**ver sección 5.4.1**).

5.3.3 Modelo Dinámico

Para poder realizar una buena simulación con respecto a los movimientos que posee el manipulador, es necesario realizar un modelo para verificar estos movimientos dependiendo de los pares de fuerzas que se emplearán. Por lo anterior es necesario calcular, por cada articulación, las ecuaciones de movimiento [21].

El modelo dinámico se soluciona mediante la formulación de Lagrange.

5.3.3.1 Solución del modelo dinámico mediante formulación de Lagrange

La formulación de Lagrange permite obtener el modelo matemático de un sistema, por medio del cálculo de energías de este.

A continuación, se detalla la solución del modelo dinámico, primero se ilustra en la **Figura 25** las variables que se tuvieron en cuenta en el manipulador de dos grados de libertad.

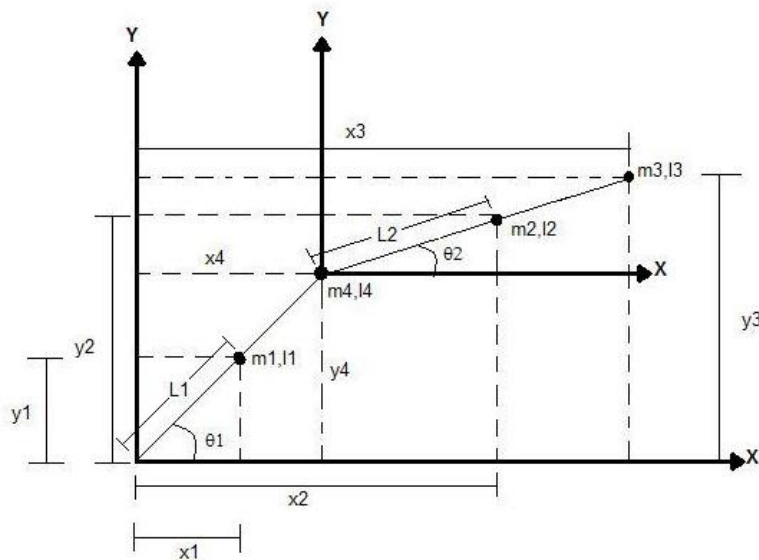


Figura 25. Análisis mediante formulación de Lagrange de un manipulador de dos grados de libertad.

Fuente: Autores

En la **Tabla 4** se observa la descripción de las variables con las cuales se realizaron los cálculos correspondientes a la dinámica del manipulador robótico de dos grados de libertad.

Variable	Descripción
L_1	Distancia desde la primera articulación hasta el centro masa del eslabón 1
L_2	Distancia desde la segunda articulación hasta el centro masa del eslabón 2

m_1	Masa del eslabón 1
m_2	Masa del eslabón 2
m_3	Masa electroimán
m_4	Masa rodamiento
I_1	Inercia del eslabón 1
I_2	Inercia del eslabón 2
I_3	Inercia electroimán
I_4	Inercia rodamiento
θ_1	Ángulo de la primera articulación
θ_2	Ángulo de la segunda articulación
x_1	La distancia en el eje x desde la primera articulación hasta el centro de masa del primer eslabón
x_2	La distancia en el eje x desde la primera articulación hasta el centro de masa del segundo eslabón
x_3	La distancia en el eje x desde la primera articulación hasta el electroimán
x_4	La distancia en el eje x desde la primera articulación hasta el rodamiento
y_1	La distancia en el eje y desde la primera articulación hasta el centro de masa del primer eslabón
y_2	La distancia en el eje y desde la primera articulación hasta el centro de masa del segundo eslabón
y_3	La distancia en el eje y desde la primera articulación hasta el electroimán
y_4	La distancia en el eje y desde la primera articulación hasta el rodamiento

Tabla 4. Descripción de variables del manipulador robótico modelo dinámico.

A continuación, se realiza el análisis dinámico por el método de Lagrange:

Análisis de posición del manipulador

$$x_1 = L_1 \cos \theta_1$$

$$y_1 = L_1 \sin \theta_1$$

$$x_2 = x_A + L_2 \cos \theta_2 = 2 L_1 \cos \theta_1 + L_2 \cos \theta_2$$

$$y_2 = y_A + L_2 \sin \theta_2 = 2 L_1 \sin \theta_1 + L_2 \sin \theta_2$$

$$x_3 = 2 L_1 \cos \theta_1 + 2 L_2 \cos \theta_2$$

$$y_3 = 2 L_1 \sin \theta_1 + 2 L_2 \sin \theta_2$$

$$x_4 = 2 L_1 \cos \theta_1$$

$$y_4 = 2 L_1 \sin \theta_1$$

Derivamos para encontrar la velocidad

$$\dot{x}_1 = -L_1 \dot{\theta}_1 \sin \theta_1$$

$$\dot{y}_2 = L_1 \dot{\theta}_1 \cos \theta_1$$

$$\dot{x}_2 = -2 L_1 \dot{\theta}_1 \sin \theta_1 - L_2 \dot{\theta}_2 \sin \theta_2$$

$$\dot{y}_2 = 2 L_1 \dot{\theta}_1 \cos \theta_1 + L_2 \dot{\theta}_2 \cos \theta_2$$

$$\dot{x}_3 = -2 L_1 \dot{\theta}_1 \sin \theta_1 - 2 L_2 \dot{\theta}_2 \sin \theta_2$$

$$\dot{y}_3 = 2 L_1 \dot{\theta}_1 \cos \theta_1 + 2 L_2 \dot{\theta}_2 \cos \theta_2$$

$$\dot{x}_4 = -2 L_1 \dot{\theta}_1 \sin \theta_1$$

$$\dot{y}_4 = 2 L_1 \dot{\theta}_1 \cos \theta_1$$

Se obtiene la velocidad de cada eslabón

$$\dot{\gamma}_1^2 = \dot{x}_1^2 + \dot{y}_1^2 = L_1^2 \dot{\theta}_1^2 \sin^2 \theta_1 + L_1^2 \dot{\theta}_1^2 \cos^2 \theta_1 = L_1^2 \dot{\theta}_1^2$$

$$\begin{aligned} \dot{\gamma}_2^2 = \dot{x}_2^2 + \dot{y}_2^2 &= 4 L_1^2 \dot{\theta}_1^2 \sin^2 \theta_1 + 4 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \sin \theta_1 \sin \theta_2 + L_2^2 \dot{\theta}_2^2 \sin^2 \theta_2 + 4 L_1^2 \dot{\theta}_1^2 \cos^2 \theta_1 \\ &\quad + 4 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \cos \theta_1 \cos \theta_2 + L_2^2 \dot{\theta}_2^2 \cos^2 \theta_2 \end{aligned}$$

$$\dot{\gamma}_2^2 = 4 L_1^2 \dot{\theta}_1^2 + 4 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 (\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2) + L_2^2 \dot{\theta}_2^2$$

$$\dot{\gamma}_2^2 = 4 L_1^2 \dot{\theta}_1^2 + 4 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2) + L_2^2 \dot{\theta}_2^2$$

$$\dot{\gamma}_3^2 = \dot{x}_3^2 + \dot{y}_3^2 = 4 L_1^2 \dot{\theta}_1^2 \sin^2 \theta_1 + 4 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 (\sin \theta_1 \sin \theta_2) + 4 L_2^2 \dot{\theta}_2^2 \sin^2 \theta_2 +$$

$$4 L_1^2 \dot{\theta}_1^2 \cos^2 \theta_1 + 4 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 (\cos \theta_1 \cos \theta_2) + 4 L_2^2 \dot{\theta}_2^2 \cos^2 \theta_2$$

$$\dot{\gamma}_3^2 = 4 L_1^2 \dot{\theta}_1^2 + 4 L_2^2 \dot{\theta}_2^2 + 4 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 (\sin \theta_1 \sin \theta_2 + \cos \theta_1 \cos \theta_2)$$

$$\dot{\gamma}_3^2 = 4 L_1^2 \dot{\theta}_1^2 + 4 L_2^2 \dot{\theta}_2^2 + 4 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2)$$

$$\dot{\gamma}_4^2 = \dot{x}_4^2 + \dot{y}_4^2 = 4 L_1^2 \dot{\theta}_1^2 \sin^2 \theta_1 + 4 L_1^2 \dot{\theta}_1^2 \cos^2 \theta_1$$

$$\dot{\gamma}_4^2 = 4 L_1^2 \dot{\theta}_1^2$$

Energía Cinética del manipulador (T)

$$T = \frac{1}{2} m_1 \dot{\gamma}_1^2 + \frac{1}{2} m_2 \dot{\gamma}_2^2 + \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_2^2 + \frac{1}{2} m_3 \dot{\gamma}_3^2 + \frac{1}{2} m_4 \dot{\gamma}_4^2 + \frac{1}{2} I_3 \dot{\theta}_2^2 + \frac{1}{2} I_4 \dot{\theta}_1^2$$

$$T = \frac{1}{2} m_1 L_1^2 \dot{\theta}_1^2 + 2 m_2 L_1^2 \dot{\theta}_1^2 + 2 m_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2) + \frac{1}{2} m_2 L_2^2 \dot{\theta}_2^2 + \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_2^2$$

$$+ 2 m_3 L_1^2 \dot{\theta}_1^2 + 2 m_3 L_2^2 \dot{\theta}_2^2 + 2 m_3 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2) + 2 m_4 L_1^2 \dot{\theta}_1^2 + \frac{1}{2} I_3 \dot{\theta}_2^2$$

$$+ \frac{1}{2} I_4 \dot{\theta}_1^2$$

Energía Potencial del manipulador (V)

$$V = m_1 g L_1 \sin \theta_1 + 2 m_2 g L_1 \sin \theta_1 + m_2 g L_2 \sin \theta_2 + 2 m_4 g L_1 \sin \theta_1 + 2 m_3 g L_1 \sin \theta_1$$

$$+ 2 m_3 g L_2 \sin \theta_2$$

Lagrangiano (L)

$$L = T - V$$

$$L = \frac{1}{2} (m_1 L_1^2 + 4 m_2 L_1^2 + I_1 + 4 m_3 L_1^2 + 4 m_4 L_1^2 + I_4) \dot{\theta}_1^2 + \frac{1}{2} (m_2 L_2^2 + I_2 + 4 m_3 L_2^2 + I_3) \dot{\theta}_2^2$$

$$+ 2 m_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2) + 2 m_3 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2)$$

$$- \sin \theta_1 (m_1 g L_1 + 2 m_2 g L_1 + 2 m_4 g L_1 + 2 m_3 g L_1) - \sin \theta_2 (m_2 g L_2$$

$$+ 2 m_3 g L_2)$$

Ecuación Diferencial de Lagrange para θ_1

$$\frac{\partial}{\partial t} \left\{ \frac{\partial L}{\partial \dot{\theta}_1} \right\} - \frac{\partial L}{\partial \theta_1} + \beta_1 \dot{\theta}_1 = \tau_1$$

$$\begin{aligned} \frac{\partial L}{\partial \theta_1} = & (m_1 L_1^2 + 4 m_2 L_1^2 + I_1 + 4 m_3 L_1^2 + 4 m_4 L_1^2 + I_4) \dot{\theta}_1 + 2 m_2 L_1 L_2 \dot{\theta}_2 \cos(\theta_1 - \theta_2) \\ & + 2 m_3 L_1 L_2 \dot{\theta}_2 \cos(\theta_1 - \theta_2) \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial t} \left\{ \frac{\partial L}{\partial \dot{\theta}_1} \right\} = & (m_1 L_1^2 + 4 m_2 L_1^2 + I_1 + 4 m_3 L_1^2 + 4 m_4 L_1^2 + I_4) \ddot{\theta}_1 + 2 m_2 L_1 L_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) - \\ & 2 m_2 L_1 L_2 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + 2 m_3 L_1 L_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) - 2 m_3 L_1 L_2 \dot{\theta}_2 \sin(\theta_1 - \theta_2) \end{aligned}$$

$$\begin{aligned} \frac{\partial L}{\partial \theta_1} = & -2 m_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) - 2 m_3 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) \\ & - \cos \theta_1 (m_1 g L_1 + 2 m_2 g L_1 + 2 m_4 g L_1 + 2 m_3 g L_1) \end{aligned}$$

$$\begin{aligned} \tau_1 = & (m_1 L_1^2 + 4 m_2 L_1^2 + I_1 + 4 m_3 L_1^2 + 4 m_4 L_1^2 + I_4) \ddot{\theta}_1 + 2 m_2 L_1 L_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) \\ & - 2 m_2 L_1 L_2 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + 2 m_3 L_1 L_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) \\ & - 2 m_3 L_1 L_2 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + 2 m_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) \\ & + 2 m_3 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) \\ & + \cos \theta_1 (m_1 g L_1 + 2 m_2 g L_1 + 2 m_4 g L_1 + 2 m_3 g L_1) + \beta_1 \dot{\theta}_1 \end{aligned}$$

τ_1 = torque del motor 1

Ecuación Diferencial de Lagrange para θ_2

$$\frac{\partial}{\partial t} \left\{ \frac{\partial L}{\partial \dot{\theta}_2} \right\} - \frac{\partial L}{\partial \theta_2} + \beta_2 \dot{\theta}_2 = \tau_2$$

$$\frac{\partial L}{\partial \theta_2} = (m_2 L_2^2 + I_2 + 4 m_3 L_2^2 + I_3) \dot{\theta}_2 + 2 m_2 L_1 L_2 \dot{\theta}_1 \cos(\theta_1 - \theta_2) + 2 m_3 L_1 L_2 \dot{\theta}_1 \cos(\theta_1 - \theta_2)$$

$$\frac{\partial}{\partial t} \left\{ \frac{\partial L}{\partial \dot{\theta}_2} \right\} = (m_2 L_2^2 + I_2 + 4 m_3 L_2^2 + I_3) \ddot{\theta}_2 + 2 m_2 L_1 L_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - 2 m_2 L_1 L_2 \dot{\theta}_1 \sin(\theta_1 - \theta_2) + 2 m_3 L_1 L_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - 2 m_3 L_1 L_2 \dot{\theta}_1 \sin(\theta_1 - \theta_2)$$

$$\frac{\partial L}{\partial \theta_2} = -2 m_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) - 2 m_3 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) - \cos \theta_2 (m_2 g L_2 + 2 m_3 g L_2)$$

$$\begin{aligned} \tau_2 = & (m_2 L_2^2 + I_2 + 4 m_3 L_2^2 + I_3) \ddot{\theta}_2 + 2 m_2 L_1 L_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - 2 m_2 L_1 L_2 \dot{\theta}_1 \sin(\theta_1 - \theta_2) \\ & + 2 m_3 L_1 L_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - 2 m_3 L_1 L_2 \dot{\theta}_1 \sin(\theta_1 - \theta_2) \\ & + 2 m_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + 2 m_3 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) \\ & + \cos \theta_2 (m_2 g L_2 + 2 m_3 g L_2) + \beta_2 \dot{\theta}_2 \end{aligned}$$

τ_2 = torque del motor 2

Los datos reales del manipulador robótico son los siguientes:

$$m_1 = 0.15 \text{ kg}$$

$$m_2 = 0.149 \text{ kg}$$

$$m_3 = 0.5 \text{ kg}$$

$$m_4 = 0.255 \text{ kg}$$

El valor de los ángulos es 90 debido a que es la posición inicial.

$$\theta_1 = \theta_2 = 90$$

Se utilizó una velocidad de

$$\dot{\theta}_1 = \dot{\theta}_2 = \frac{60^\circ}{0.14s} = 428.57 \frac{^\circ}{s} \cdot \frac{\pi}{180} \cdot 0.08m = 0.5984 \frac{m}{s}$$

Esta es la velocidad del servomotor a 6V, tomada del datasheet (**Ver Anexo C**).

Para hallar la aceleración se divide la velocidad en 1.2 segundos el cual es el tiempo que transcurre desde la posición inicial del servo hasta llegar a velocidad constante.

$$\ddot{\theta}_1 = \ddot{\theta}_2 = \frac{0.5984 \frac{m}{s}}{1.2 s} = 0.4987 \frac{m}{s^2}$$

$$L_1 = L_2 = 0.08 m$$

Las inercias I_1, I_2, I_4 fueron tomadas del diseño realizado en SolidWorks teniendo en cuenta su respectivo material.

$$I_1 = 0.004 kg \cdot m^2$$

$$I_2 = 0.007 kg \cdot m^2$$

$$I_4 = 0.012 kg \cdot m^2$$

En la **Tabla 5** se observa la descripción de las variables con las cuales se realizaron los cálculos para hallar la inercia del electroimán (I_3).

Variable	Descripción
M_{ele}	Masa del electroimán
V_{ele}	Volumen del electroimán
ρ_{mat}	Densidad del material
d	Distancia del centro de la base al centro del electroimán
b	Base del electroimán
h	Altura del electroimán
z	Profundidad del electroimán

Tabla 5. Descripción de variables para hallar la inercia del electroimán.

Hallamos el Volumen del electroimán:

$$V_{ele} = b \cdot h \cdot z = 0.06 m \cdot 0.03 m \cdot 0.03 m = 0.000054 m^3$$

Hallamos la Masa del electroimán teniendo en cuenta la densidad del material ($7850 kg/m^3$):

$$M_{ele} = V_{ele} \cdot \rho_{mat} = 0.000054 m^3 \cdot 7850 \frac{kg}{m^3} = 0.4239 kg$$

Finalmente hallamos la inercia del electroimán:

$$\begin{aligned}
 I_3 &= \frac{1}{12} \cdot M_{ele} \cdot (b^2 + h^2) + M_{ele} \cdot d \\
 &= \frac{1}{12} \cdot 0.4239 \text{ kg} \cdot ((0.06 \text{ m})^2 + (0.03 \text{ m})^2) + 0.4239 \text{ kg} \cdot (0.32 \text{ m})^2 \\
 &= 0.0435 \text{ kg} \cdot \text{m}^2
 \end{aligned}$$

Los valores de β_1 y β_2 (coeficientes de fricción viscosa de los motores 1 y 2) son valores estándar altos para realizar la solución de este método con los valores más altos, con lo cual se obtiene como resultado el mayor torque posible.

$$\beta_1 = \beta_2 = 0.003 \frac{N \cdot m \cdot s}{rad}$$

Se procede a realizar los cálculos del torque de los servomotores teniendo en cuenta las ecuaciones obtenidas mediante la formulación de Lagrange.

$$\tau_1 = 2.34 \text{ Nm}$$

$$\tau_2 = 0.93 \text{ Nm}$$

El torque de los servomotores en kg por centímetro es 24.7, siendo

$$24.7 \text{ kgcm} \times \frac{1 \text{ m}}{100 \text{ cm}} \times 9.81 \frac{\text{m}}{\text{s}^2} = 2.42 \text{ Nm}$$

Por lo tanto, el torque de los servomotores escogidos cumple con el torque calculado necesario.

Luego de desarrollar el modelo cinemático inverso y dinámico se procedió a realizar el diseño del manipulador robótico

Para el diseño del manipulador robótico se tuvieron en cuenta las siguientes características:

- El diseño debe estar libre de concentradores de esfuerzos, para evitar fatiga del material. Por ello se le implementaron redondeces a un primer diseño en los puntos críticos (**Ver Figura 26 y Figura 27**).

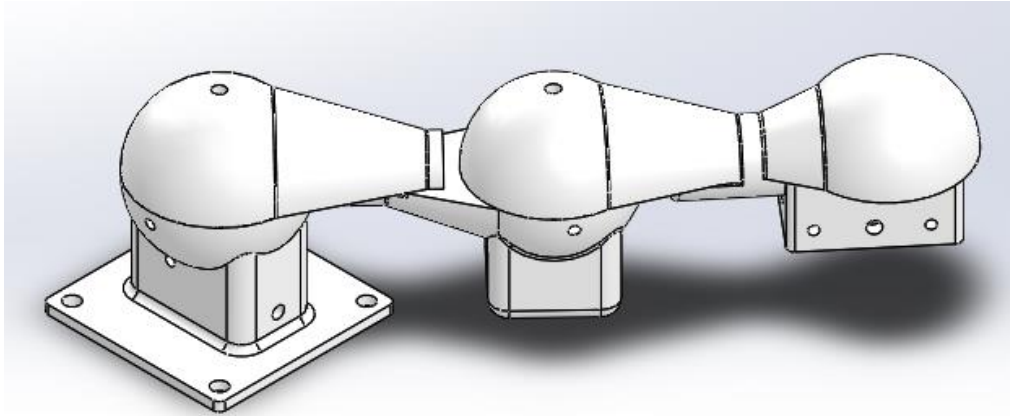


Figura 26. Primer Diseño Brazo Robótico.

Fuente: Autores

- Se tiene en cuenta que el material con el que se imprimió el prototipo no es isotrópico, esto da a entender que el material en el instante en que se imprime, no mantiene sus propiedades térmicas y mecánicas en todos sus ejes, las mantiene solo en el eje z [20].
- El prototipo se realizó en Impresora 3D, debido al costo que conlleva realizar este proyecto en materiales en los que fuera necesario mecanizar las piezas.

A continuación, se muestra el diseño final del manipulador robótico realizado en SolidWorks, teniendo en cuenta los anteriores conceptos.

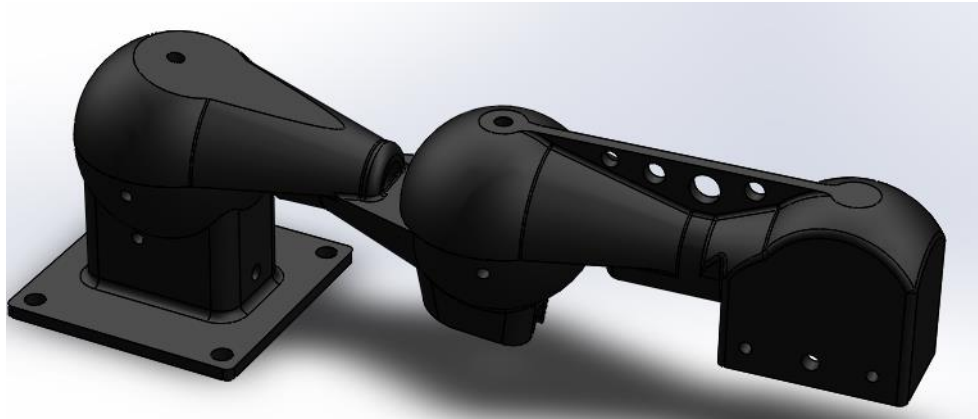


Figura 27. Diseño Final Brazo Robótico.

Fuente: Autores

5.4 CONTROL DE POSICIÓN

5.4.1 Generación de Trayectoria

Para realizar la generación de trayectoria para el manipulador robótico se utilizaron los métodos geométricos expuestos en la sección **5.3.2.1**, con los cuales se hallaron los ángulos necesarios para posicionar el manipulador en la coordenada donde se encontraba la pieza a clasificar.

El programa en Matlab se realizó con las ecuaciones expuestas en la sección 5.3.2.1 y con los datos reales en cuanto a las distancias entre los centros de los eslabones los cuales son 16 centímetros, esto con el fin de abarcar todo el ancho y alto de la maqueta final (Ver **Figura 28**).

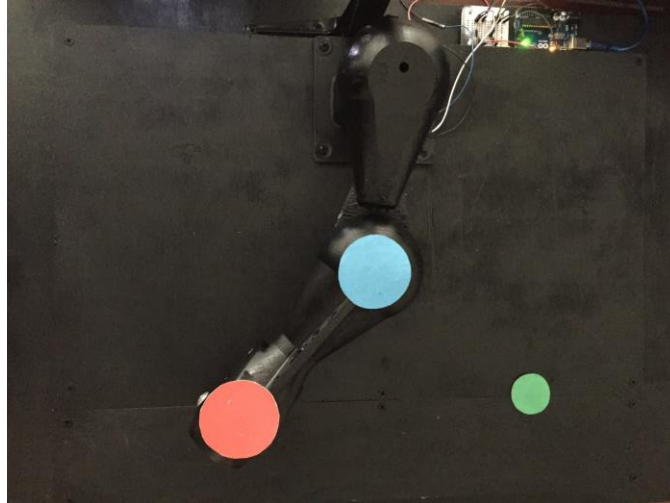


Figura 28. Vista Superior Maqueta.

Fuente: Autores

Para realizar el programa en Matlab se tuvo en cuenta que el tamaño en pixeles de la captura de la imagen es de 640 de ancho por 480 de alto, lo que indica que hay que convertir los valores en centímetros a pixeles para que las fórmulas funcionen correctamente y el brazo llegue a la posición deseada.

Para convertir los centímetros a pixeles se realizó una captura con la cámara al primer eslabón del brazo robótico con círculos de colores diferentes (rojo y azul) en cada articulación, en posiciones vertical y horizontal para medir cuantos pixeles hay en una distancia de 16 centímetros la cual es la medida que hay de centro a centro de cada círculo

En la Figura 29 y 30 se ilustran las dos capturas, en las cuales se tomaron los datos de ubicación de cada uno de los centros de los círculos.

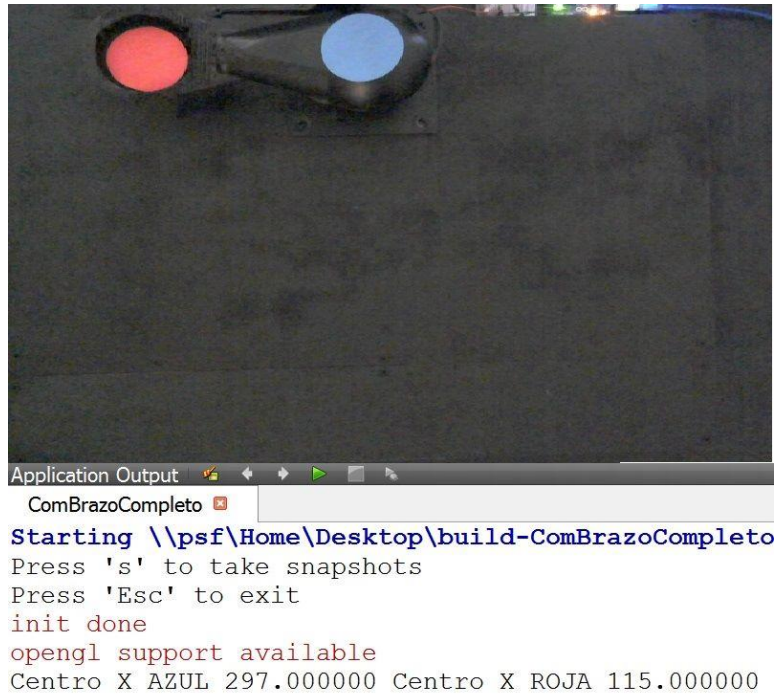


Figura 29. Captura para determinar centímetros en el eje X.

Fuente: Autores

Como se puede observar en la figura 29 la ubicación de centro del círculo azul es de 297 pixeles en X y la ubicación del centro del círculo rojo es de 115 pixeles en X, por consiguiente esta resta da el resultado de cuantos pixeles en X hay en 16 centímetros reales, que es la medida desde el centro del círculo azul hasta el centro del círculo rojo, el resultado de esta resta es de 182 pixeles en X para una distancia de 16 centímetros en X, esta relación entre pixeles y centímetros es válida siempre y cuando la cámara no cambie de altura de acuerdo a la **Figura 28**.



Figura 30. Captura para determinar centímetros en el eje Y.

Fuente: Autores

Como se puede observar en la figura 30 la ubicación de centro del círculo rojo es de 233 pixeles en Y y la ubicación del centro del círculo azul es de 47 pixeles en Y, por consiguiente esta resta da el resultado de cuantos pixeles en Y hay en 16 centímetros reales, que es la medida desde el centro del círculo azul hasta el centro del círculo rojo, el resultado de esta resta es de 186 pixeles en Y para una distancia de 16 centímetros en Y, esta relación entre pixeles y centímetros es válida siempre y cuando la cámara no cambie de altura de acuerdo a la **Figura 28**

Luego de tener claras las medidas en pixeles se procedió a diseñar el programa en Matlab, teniendo en cuenta el tamaño de la imagen (Captura 640x480), se introdujeron las fórmulas con las cuales el manipulador se debe posicionar en el lugar correspondiente, se tuvo en cuenta que el manipulador comenzaba en el punto (0,0) del plano cartesiano y de ahí realiza el recorrido hacia la pieza a clasificar y hacia los cubículos donde irán dichas piezas.

El punto (0,0) de la imagen es en la esquina superior izquierda de allí comienzan a contarse los pixeles hasta llegar a la esquina inferior derecha en la cual la coordenada es (640,480), en Matlab se realizó una simulación en un plano de ejes coordenados como ilustra la **Figura 31**.

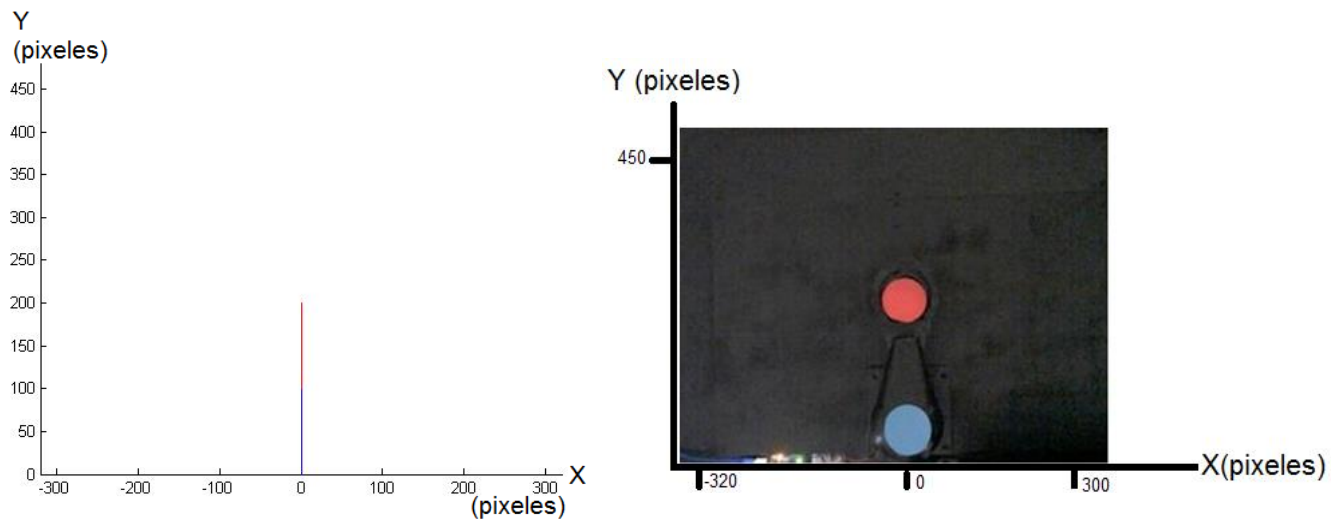


Figura 31. Comparación entre simulación en Matlab y captura original.

Fuente: Autores

Como se puede observar en la **Figura 31**, para realizar la simulación en Matlab se dividió el ancho de la imagen en 2, para que el punto de origen del manipulador robótico fuese el punto 0 en el plano cartesiano, por ende, se tienen valores positivos de 0 a 320 y valores negativos de 0 a -320.

Luego de configurar el tamaño del plano, se procede a configurar la entrega de los ángulos ya que para la parte negativa de X se requiere una condición que convierta los pixeles positivos a negativos, esto se logró restando 320 al número de pixeles que arrojaba la imagen siempre y cuando estos fueran mayores a 320.

A continuación, se muestran dos ejemplos, en los cuales se le ingresaron datos reales al programa realizado en Matlab y este arrojó los valores de los ángulos θ_1 y θ_2 necesarios para que el manipulador robótico llegara a la posición deseada.

Para el primer ejemplo se tomaron las siguientes coordenadas en donde se debe posicionar el manipulador robótico, el resultado se ilustra en la **Figura 32**:

x deseado = 30 cm

y deseado = 10 cm

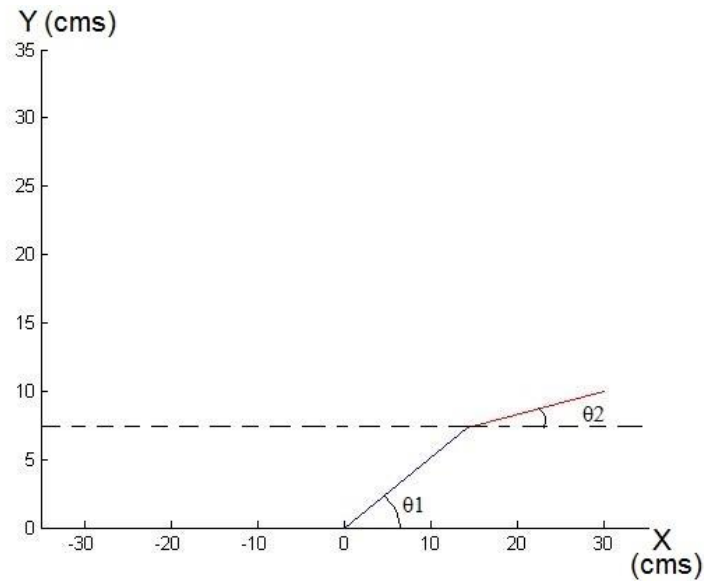


Figura 32. Ejemplo trayectoria eje X positivo.

Fuente: Autores

Donde:

$$\theta_1 = 27.2412$$

$$\theta_2 = 9.6287$$

Para el segundo ejemplo se tomaron las siguientes coordenadas en donde se debe posicionar el manipulador robótico, el resultado se ilustra en la **Figura 33**:

x deseado= -30 cm

y deseado =10 cm

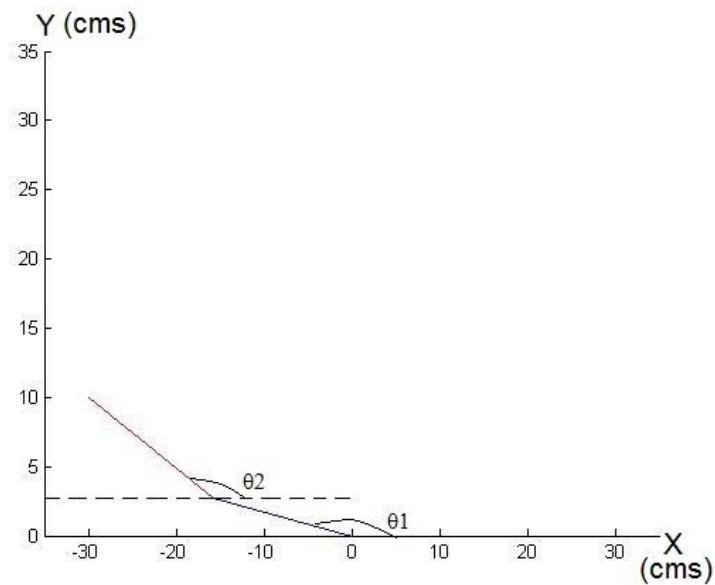


Figura 33. Ejemplo trayectoria eje X negativo.

Fuente: Autores

Donde:

$$\theta_1 = 170.3713$$

$$\theta_2 = 152.7588$$

Luego de tener el programa realizado y probado en Matlab se procedió a adquirir las coordenadas de los cubículos, los cuales siempre estarán en la misma posición por ende con una sola captura basta para procesar dicha posición y programar para que el brazo luego de ubicarse y recoger la pieza, proceda a clasificarla (ver **Figura 34**).

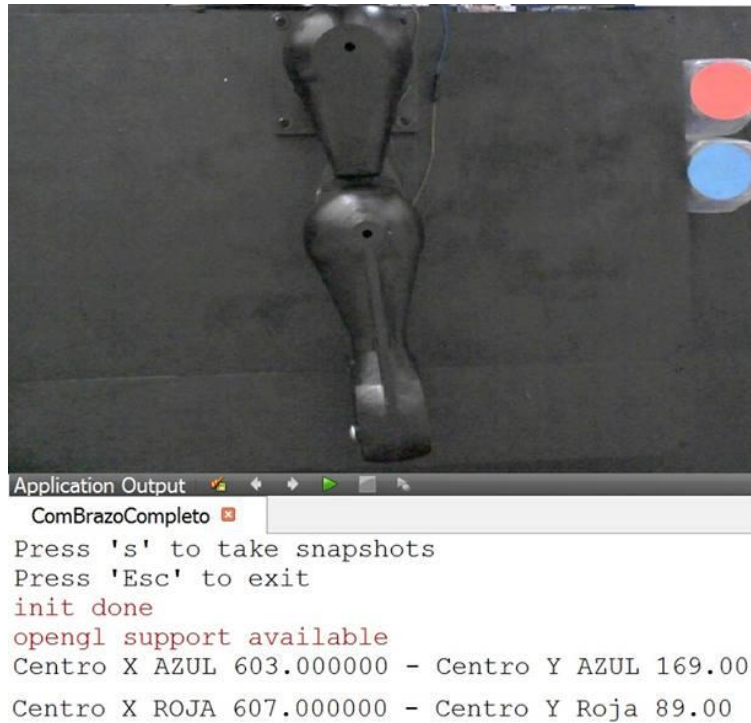


Figura 34. Ubicación de cubículos.

Fuente: Autores

Como se puede observar en la **Figura 34** se extraen de la captura las coordenadas donde se encuentran los cubículos. Los cubículos utilizados son 4 con el fin de clasificar círculos rojos y azules y cuadrados rojos y azules.

La generación de trayectoria presentó un problema con los servos, ya que el plano de coordenadas de trabajo del segundo servo dependía del movimiento del primer servo, por ende, es necesario realizar un control de posición para el segundo servo que será explicado a continuación.

5.4.2 Control de Posición del Segundo Eslabón

Para poder realizar el control, es necesario tener la función de transferencia de la planta, en este caso el segundo eslabón del brazo robótico, ya que es el que presenta el problema de posicionamiento, por ende, se realiza el análisis dinámico de un brazo robótico de un eslabón, solucionado a continuación:

A continuación, se detalla la solución del modelo Dinámico, primero se ilustra en la **Figura 35** las variables que se tuvieron en cuenta en el manipulador de un grado de libertad.

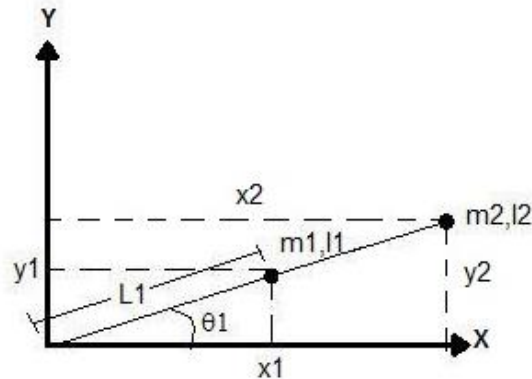


Figura 35. Análisis mediante formulación de Lagrange.

Fuente: Autores

En la **Tabla 6** se observa la descripción de las variables con las cuales se realizaron los cálculos correspondientes a la dinámica del manipulador robótico.

Variable	Descripción
L_1	Distancia desde la articulación hasta el centro masa del eslabón
m_1	Masa del eslabón
m_2	Masa electroimán
I_1	Inercia del eslabón
I_2	Inercia electroimán
θ_1	Ángulo de la articulación
x_1	La distancia en el eje x desde la articulación hasta el centro de masa del eslabón
x_2	La distancia en el eje x desde la articulación hasta el electroimán
y_1	La distancia en el eje y desde la articulación hasta el centro de masa del eslabón
y_2	La distancia en el eje y desde la articulación hasta el electroimán
A	Ganancia de corriente del amplificador (Puente H)
k_s	Sensibilidad del comparador
k_p	Ganancia conversor entre Voltaje y PWM
k_a, k_b	Coeficientes del Motor D.C
R_a	Resistencia del Motor D.C
p	Ganancia sensor de posición (potenciómetro)

V_i	Voltaje que envía el Arduino al Motor D.C (entrada al comparador)
n	Relación de engranajes

Tabla 6. Descripción de variables del manipulador robótico modelo dinámico.

A continuación, se realiza el análisis dinámico por el método de Lagrange:

Posición

$$x_1 = L_1 \cos \theta_1$$

$$y_1 = L_1 \sin \theta_1$$

$$x_2 = 2 L_1 \cos \theta_1$$

$$y_2 = 2 L_1 \sin \theta_1$$

Derivamos para encontrar la velocidad

$$\dot{x}_1 = -L_1 \dot{\theta}_1 \sin \theta_1$$

$$\dot{y}_1 = L_1 \dot{\theta}_1 \cos \theta_1$$

$$\dot{x}_2 = -2 L_1 \dot{\theta}_1 \sin \theta_1$$

$$\dot{y}_2 = 2 L_1 \dot{\theta}_1 \cos \theta_1$$

$$\dot{\gamma}_1^2 = \dot{x}_1^2 + \dot{y}_1^2 = L_1^2 \dot{\theta}_1^2 \sin^2 \theta_1 + L_1^2 \dot{\theta}_1^2 \cos^2 \theta_1 = L_1^2 \dot{\theta}_1^2$$

$$\dot{\gamma}_2^2 = \dot{x}_2^2 + \dot{y}_2^2 = 4 L_1^2 \dot{\theta}_1^2 \sin^2 \theta_1 + 4 L_1^2 \dot{\theta}_1^2 \cos^2 \theta_1 = 4 L_1^2 \dot{\theta}_1^2$$

Energía Cinética

$$T = \frac{1}{2} m_1 \dot{\gamma}_1^2 + \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} m_2 \dot{\gamma}_2^2 + \frac{1}{2} I_2 \dot{\theta}_1^2$$

$$T = \frac{1}{2} m_1 L_1^2 \dot{\theta}_1^2 + \frac{1}{2} I_1 \dot{\theta}_1^2 + 2 m_2 L_1^2 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_1^2$$

Energía Potencial

$$V = m_1 g L_1 \sin \theta_1 + 2 m_2 g L_1 \sin \theta_1$$

Lagrangiano

$$L = T - V$$

$$L = \frac{1}{2} (m_1 L_1^2 + I_1 + 4 m_2 L_1^2 + I_2) \dot{\theta}_1^2 - \sin \theta_1 (m_1 g L_1 + 2 m_2 g L_1)$$

Ecuación Diferencial de Lagrange para θ_1

$$\frac{\partial}{\partial t} \left\{ \frac{\partial L}{\partial \dot{\theta}_1} \right\} - \frac{\partial L}{\partial \theta_1} + \frac{\partial R_1}{\partial \dot{\theta}_1} = \tau_1$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = (m_1 L_1^2 + I_1 + 4 m_2 L_1^2 + I_2) \dot{\theta}_1$$

$$\frac{\partial}{\partial t} \left\{ \frac{\partial L}{\partial \dot{\theta}_1} \right\} = (m_1 L_1^2 + I_1 + 4 m_2 L_1^2 + I_2) \ddot{\theta}_1$$

$$\frac{\partial L}{\partial \theta_1} = -\cos \theta_1 (m_1 g L_1 + 2 m_2 g L_1)$$

$$\frac{\partial R_1}{\partial \dot{\theta}_1} = \beta_1 \dot{\theta}_1$$

$$\tau_1 = (m_1 L_1^2 + I_1 + 4 m_2 L_1^2 + I_2) \ddot{\theta}_1 + \cos \theta_1 (m_1 g L_1 + 2 m_2 g L_1) + \beta_1 \dot{\theta}_1$$

Para poder obtener la función de transferencia para el control de posición es necesario analizar el modelo del servomotor (Ver Figura 36 y 37), señalado a continuación:

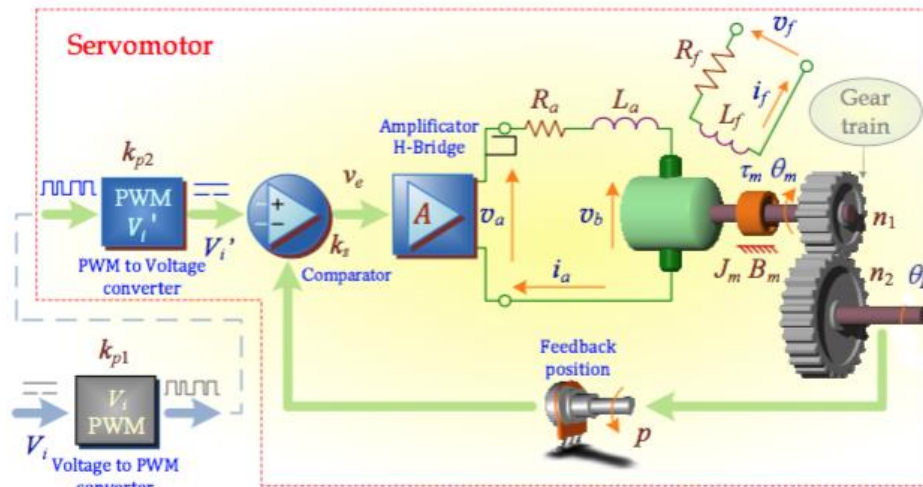


Figura 36. Diagrama esquemático del servomotor.

Fuente: [23]

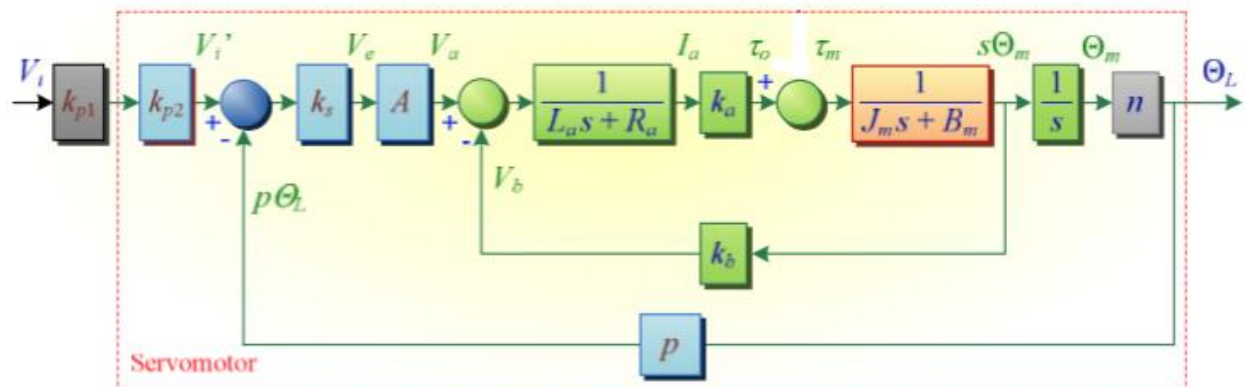


Figura 37. Diagrama de bloques del servomotor.

Fuente: [23]

Ecuación Diferencial de un Motor Eléctrico D.C

$$V_a = L_a \frac{dia}{dt} + R_a ia + ea = L_a \frac{dia}{dt} + R_a ia + k_b \dot{\theta}_m$$

$$\tau_m = K\phi ia = k_1 k_2 ia = k_a ia$$

Analizando el diagrama de bloques del servomotor

$$\tau_o = \tau_m$$

Ecuación mecánica del servomotor

$$\tau_m = J_m \ddot{\theta}_m + B_m \dot{\theta}_m$$

Reemplazando se obtiene:

$$\tau_o = J_m \ddot{\theta}_m + B_m \dot{\theta}_m$$

$$\tau_o = k_a ia$$

Despejando ia

$$ia = \frac{\tau_o}{k_a}$$

Reemplazando en la ecuación diferencial del Motor Eléctrico D.C

$$V_a = \frac{L_a}{k_a} \dot{\tau}_o + \frac{R_a}{k_a} \tau_o + k_b \dot{\theta}_m$$

Reemplazando la ecuación mecánica se obtiene:

$$V_a = \frac{L_a}{k_a} (J_m \ddot{\theta}_m + B_m \dot{\theta}_m) + \frac{R_a}{k_a} \tau_o + k_b \dot{\theta}_m$$

Despejamos τ_o

$$\tau_o = - \frac{(B_m L_a \ddot{\theta}_m + J_m L_a \ddot{\theta}_m + k_a (k_b \dot{\theta}_m - V_a))}{R_a}$$

Debido a que la inductancia $L_a = 0$, y reemplazando $\theta_m = \frac{1}{n} \theta$ (teniendo en cuenta la relación de engranajes), la ecuación de torque del servomotor queda de la siguiente manera:

$$\tau_o = \frac{(V_a - k_b \frac{1}{n} \dot{\theta}) k_a}{R_a}$$

Se reemplaza el voltaje V_a

$$V_a = A k_s k_{p1} k_{p2} V_i - A k_s p \theta$$

Se igualan los torques hallados mediante la formulación de Lagrange y diagramas del servomotor:

$$\tau_1 = \tau_o$$

$$\begin{aligned} & \frac{A k_a k_s k_{p1} k_{p2} V_i - (A k_a k_s p) \theta - \left(k_a k_b \frac{1}{n} \right) \dot{\theta}}{R_a} \\ & = (m_1 L_1^2 + I_1 + 4 m_2 L_1^2 + I_2) \ddot{\theta} + \cos \theta (m_1 g L_1 + 2 m_2 g L_1) + \beta_1 \dot{\theta} \end{aligned}$$

Se despeja la posición con respecto a V_i para obtener la función de transferencia:

$$\begin{aligned}
 A k_a k_s k_{p1} k_{p2} V_i &= (R_a m_1 L_1^2 + R_a I_1 + 4 R_a m_2 L_1^2 + R_a I_2) \ddot{\theta} \\
 &+ \cos \theta (R_a m_1 g L_1 + 2 R_a m_2 g L_1) + R_a \beta_1 \dot{\theta} + (A k_a k_s p) \theta + \left(k_a k_b \frac{1}{n} \right) \dot{\theta}
 \end{aligned}$$

Se aplica la transformada de Laplace a la ecuación:

$$\begin{aligned}
 A k_a k_s k_{p1} k_{p2} V_i(s) &= (R_a m_1 L_1^2 + R_a I_1 + 4 R_a m_2 L_1^2 + R_a I_2) s^2 \theta(s) \\
 &+ (R_a m_1 g L_1 + 2 R_a m_2 g L_1) \theta(s) + R_a \beta_1 s \theta(s) + (A k_a k_s p) \theta(s) \\
 &+ \left(k_a k_b \frac{1}{n} \right) s \theta(s)
 \end{aligned}$$

$$\begin{aligned}
 \left[(R_a m_1 L_1^2 + R_a I_1 + 4 R_a m_2 L_1^2 + R_a I_2) s^2 + \left(k_a k_b \frac{1}{n} + R_a \beta_1 \right) s \right. \\
 \left. + (R_a m_1 g L_1 + 2 R_a m_2 g L_1 + A k_a k_s p) \right] \theta(s) = (A k_a k_s k_{p1} k_{p2}) V_i(s)
 \end{aligned}$$

Considerando que $k_{p1} k_{p2} = k_p$, se obtiene la función de transferencia:

$$\begin{aligned}
 G(s) &= \frac{\theta(s)}{V_i(s)} \\
 &= \frac{A k_a k_s k_p}{(R_a m_1 L_1^2 + R_a I_1 + 4 R_a m_2 L_1^2 + R_a I_2) s^2 + \left(k_a k_b \frac{1}{n} + R_a \beta_1 \right) s + (R_a m_1 g L_1 + 2 R_a m_2 g L_1 + A k_a k_s p)}
 \end{aligned}$$

Los datos reales del manipulador robótico son los siguientes:

$$m_1 = 0.15 \text{ kg}$$

$$m_2 = 0.149 \text{ kg}$$

$$m_3 = 0.5 \text{ kg}$$

$$m_4 = 0.255 \text{ kg}$$

$$L_1 = 0.08 \text{ m}$$

La inercia del eslabón fue tomada del diseño realizado en SolidWorks teniendo en cuenta el material (ABS).

$$I_1 = 0.007 \text{ kg} \cdot \text{m}^2$$

La inercia del electroimán fue tomada del cálculo hallado en la sección **5.3.3.1**.

$$I_2 = 0.0435 \text{ kg} \cdot \text{m}^2$$

Para los demás parámetros de los servomotores se consideraron los siguientes valores [24]:

$$k_a = 0.35 \frac{\text{N} \cdot \text{m}}{\text{A}}$$

$$k_b = 0.04 \frac{\text{V} \cdot \text{s}}{\text{rad}}$$

$$R_a = 1.6 \Omega$$

$$A = 10$$

$$k_s = 6$$

$$k_p = 1$$

$$p = 1$$

Para hallar la relación de engranajes del servomotor se obtuvo el número de dientes de cada engranaje observando la caja reductora de este (**Ver Figura 38**).

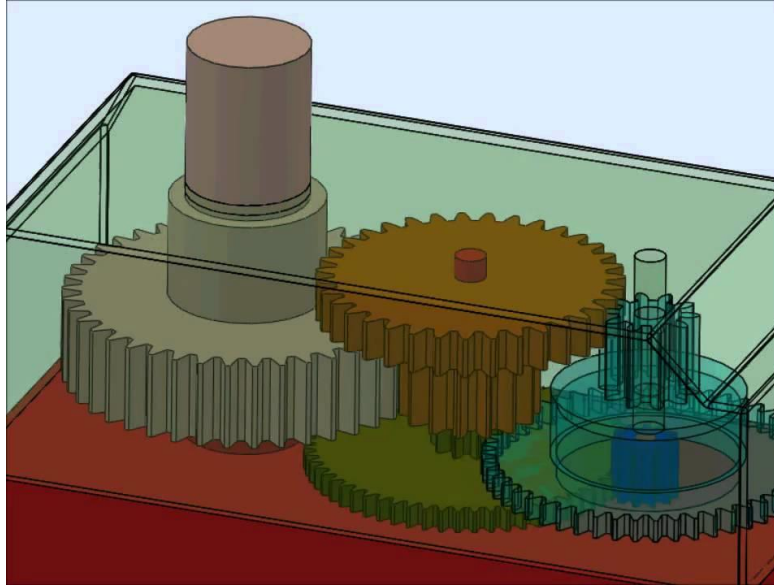


Figura 38. Caja reductora del servomotor.

Fuente: <https://www.youtube.com/watch?v=cCBcUG-cY7A>

En la **Tabla 7** se observa la descripción de los engranajes de la caja reductora del servomotor, clasificados por número, color y su respectivo número de dientes.

Número Engranaje	Color y Tamaño del Engranaje	Número de dientes del engranaje
1	Aguamarina Grande	48
2	Aguamarina Pequeño	10
3	Azul Oscuro	10
4	Verde Grande	62
5	Verde Pequeño	12
6	Naranja Grande	48
7	Naranja Pequeño	16
8	Gris	42

Tabla 7. Descripción de los engranajes de la caja reductora del servomotor.

Con la información de la Tabla 7 se procede a calcular la relación, sabiendo que en el numerador van los engranajes conducidos y en el denominador los conductores.

$$n = \frac{No^{\circ} 4}{No^{\circ} 3} \times \frac{No^{\circ} 5}{No^{\circ} 1} \times \frac{No^{\circ} 6}{No^{\circ} 2} \times \frac{No^{\circ} 8}{No^{\circ} 7} = \frac{62}{10} \times \frac{12}{48} \times \frac{48}{10} \times \frac{42}{16} = 20:1$$

Debido a que el valor del coeficiente de fricción viscosa es cercano a cero ($\beta_1 = 0.003 \frac{N \cdot m \cdot s}{rad}$), este se desprecia y la función de transferencia queda de la siguiente manera:

$$G(s) = \frac{\theta(s)}{V_i(s)} = \frac{Ak_a k_s k_p}{(R_a m_1 L_1^2 + R_a I_1 + 4 R_a m_2 L_1^2 + R_a I_2) s^2 + \left(k_a k_b \frac{1}{n}\right) s + (R_a m_1 g L_1 + 2 R_a m_2 g L_1 + Ak_a k_s p)}$$

A continuación, se reemplazan los valores reales del manipulador robótico y el servo motor:

$$G(s) = \frac{\theta(s)}{V_i(s)} = \frac{21}{0.088439s^2 + 0.287s + 21.562}$$

5.4.2.1 Diseño Control PI (Proporcional Integral)

El primer paso para realizar el controlador es definir las condiciones de diseño escogidas, a continuación, enunciadas:

$$ts = 1s \text{ (Tiempo de establecimiento)}$$

$$Mp = 10\% \text{ (Máximo sobreimpulso)}$$

$$ts = \frac{4}{\xi \omega_n} ; \xi \omega_n = 4$$

$$Mp = 100 e^{\frac{-\xi \pi}{\sqrt{1-\xi^2}}} ; \xi = 0.59$$

$$\omega_n = \frac{4}{\xi} = 6.76$$

$$\omega_d = \omega_n \sqrt{1 - \xi^2} = 5.45$$

Los polos de diseño son:

$$z_{1,2} = e^{(-\xi\omega_n \pm i\omega_d)Ts} = 0.573874 \pm 0.347909i$$

Debido a que el controlador se empleará en digital es necesario discretizar la función de transferencia hallada anteriormente, para realizar este proceso es necesario utilizar la transformada z la cual convierte una señal del dominio del tiempo al dominio de la frecuencia.

El primer paso para realizar la discretización es elegir el retenedor a utilizar, en este caso se utilizó el retenedor de orden cero debido a que es el método empleado por los dispositivos de adquisición

El tiempo de muestreo empleado se calculó teniendo en cuenta las siguientes ecuaciones:

$$\omega_s = 10 * \omega_n \text{ Valor que corresponde a 5 veces la frecuencia de Nyquist}$$

$$\omega_s = \frac{2\pi}{T}$$

Donde T es igual al tiempo de muestreo, por ende:

$$10 * \omega_n = \frac{2\pi}{T}$$

$$\text{Donde } \omega_n = 6.76$$

Por consiguiente T es igual a:

$$T = \frac{2\pi}{10 * \omega_n} = 0.092536$$

Se procedió a tomar $T = 0.1$ debido a su cercanía con el valor calculado y teniendo en cuenta que cada instrucción de Arduino se ejecuta a 62.5 nanosegundos que corresponden a una frecuencia de reloj de 16 MHz, el tiempo de muestreo es el indicado, ya que es mayor al mínimo permitido en Arduino.

El proceso de discretización se realizó en Matlab con la función `c2d`, esta función discretiza con retenedor orden cero y con un tiempo de muestreo de 0.1, comprobado anteriormente y como resultado se encuentra la siguiente función de transferencia:

$$G(z) = \frac{1.602 z + 1.403}{z^2 + 1.309 z + 0.7286}$$

Para revisar el comportamiento de la planta, esta función de transferencia se introdujo en Matlab y se comprobó su comportamiento en lazo abierto (ver **Figura 38**).

La **Figura 39** demuestra el comportamiento de la función de transferencia teniendo una entrada de escalón unitario en lazo abierto.

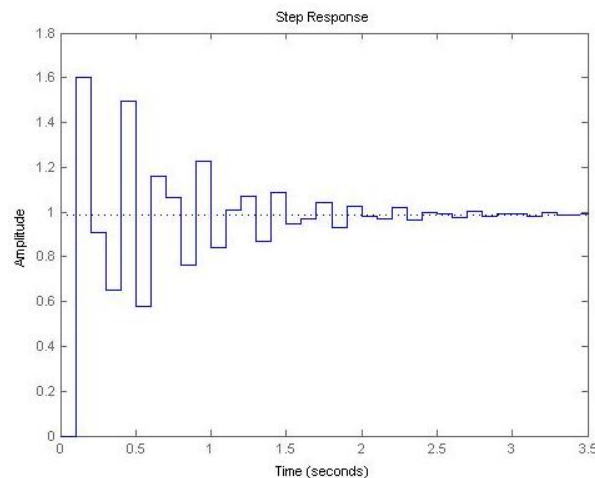


Figura 39. Comportamiento función de transferencia a respuesta de escalón unitario en lazo abierto.

Fuente: Autores

Como se observar en la **Figura 39** la función se estabiliza a los 2 segundos, pero tiene un sobre impulso de más del 50% por ello se requiere realizar un control PI que disminuya el sobre impulso a como máximo un 10% y estabilice más rápido.

Teniendo en cuenta que el tiempo de muestreo para la discretización de la función de transferencia fue de $T=0.1s$, es necesario diseñar el control PI con el mismo tiempo de muestreo.

$$Ts = T = 0.1 s \text{ (Tiempo de muestreo)}$$

Para obtener la solución del controlador con posibilidad de establecer condiciones iniciales se realizó el control con el método del lugar geométrico de las raíces, en este método la expresión del controlador PI discreto es la siguiente:

$$G_{PI}(z) = \left(Kp + Ki \frac{T}{2} \right) * \frac{z + \frac{KiT - 2Kp}{KiT + 2Kp}}{z - 1} = K_{PI} \frac{z + a}{z - 1}$$

Para obtener el cero del controlador, se debe aplicar la condición de ángulo, la cual se define de la siguiente manera:

$$\text{Condición de fase: } \angle G_{PI}(z) + \angle G(z) = 180^\circ$$

Reemplazando los valores de la función de transferencia del manipulador ($G(z)$) $G_{PI}(z)$ la condición de fase queda de la siguiente manera:

$$\frac{\angle z + a}{\angle z - 1} + \frac{\angle 1.602 z + 1.403}{\angle z^2 + 1.309 z + 0.7286} = 180^\circ$$

De la anterior ecuación se despeja $\angle z + a$ el cual es el valor del polo y cero requerido para solucionar el control.

$$(\angle(z + a) - \angle(z - 1) + (\angle(1.602z + 1.403) - \angle(z^2 + 1.309z + 0.7286))) = 180$$

$$\angle(z + a) = 180 + \angle(z - 1) - \angle(1.602z + 1.403) + \angle(z^2 + 1.309z + 0.7286)$$

Luego de despejar $\angle z + a$ se reemplaza por el polo deseado y se hayan los ángulos correspondientes

$$\angle(z + a) = 180 + 144.7702 - 13.4954 + 26.8543$$

$$\angle(z + a) = 334.1291$$

Como el valor de $\angle(z + a)$ es 334.1291 la ubicación de a (cero) se utiliza el ángulo correspondiente -26 como ilustra la figura 40.

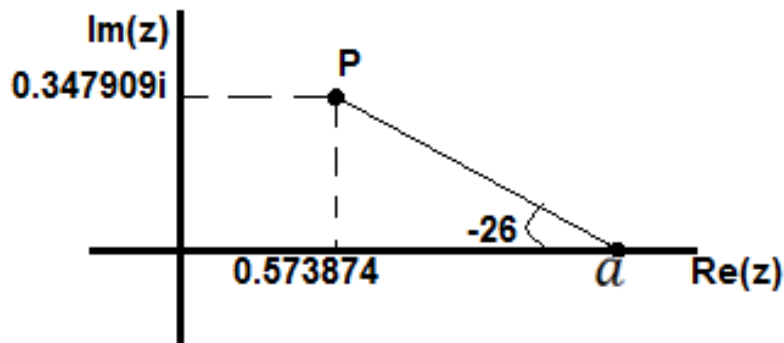


Figura 40. Ubicación de a (cero)

Fuente: Autores

A continuación, se haya el valor de a :

$$\tan(26^\circ) = \frac{0.347909}{a - 0.573874}$$

$$0.487733(a - 0.573874) = 0.347909$$

$$0.487733a - 0.279897 = 0.347909$$

$$a = \frac{0.347909 + 0.279897}{0.347909} \Rightarrow a = 1.2874$$

Para obtener K_{PI} se aplica la condición de modulo o magnitud.

$$\frac{1}{|K_{PI}|} = \frac{|z + a|}{|z - 1|} \cdot \frac{|1.602z + 1.403|}{|z^2 + 1.309z + 0.7286|}$$

Las magnitudes se hallan reemplazando el polo deseado y hallando las magnitudes de la siguiente manera:

$$\frac{1}{|K_{PI}|} = \frac{1.89353}{0.550087} \cdot \frac{2.38826}{1.89214}$$

$$\frac{1}{K_{PI}} = 4.34479$$

$$K_{PI} = 0.2303$$

Por lo tanto, el controlador PI discreto es:

$$G_{PI}(z) = K_{PI} \frac{z + a}{z - 1}$$

$$G_{PI}(z) = 0.2303 \cdot \frac{z + 1.2874}{z - 1}$$

$$K_{PI} = \left(Kp + Ki \frac{T}{2} \right)$$

$$0.2303 = (Kp + Ki0.05)$$

$$0.2303 - Ki0.05 = Kp$$

$$a = \frac{KiT - 2Kp}{KiT + 2Kp}$$

$$1.2874 = \frac{Ki0.1 - 2Kp}{Ki0.1 + 2Kp}$$

$$1.2874(Ki0.1 + 2Kp) = Ki0.1 - 2Kp$$

$$0.12874Ki + 2.5748Kp = Ki0.1 - 2Kp$$

$$2.5748Kp + 2Kp = Ki0.1 - 0.12874Ki$$

$$4.5748Kp = -0.02847Ki$$

$$Ki = -160.688Kp$$

$$Ki = -160.688(0.2303 - Ki0.05)$$

$$Ki = -37.004 + 8.0344Ki$$

$$37.004 = 8.0344Ki - Ki$$

$$Ki = 5.26043$$

$$0.2303 - Ki0.05 = Kp$$

$$Kp = -0.032722$$

A continuación, se describe la forma de probar el controlador PI discreto en Matlab (Figura 41) y su respectiva respuesta (Figura 42).

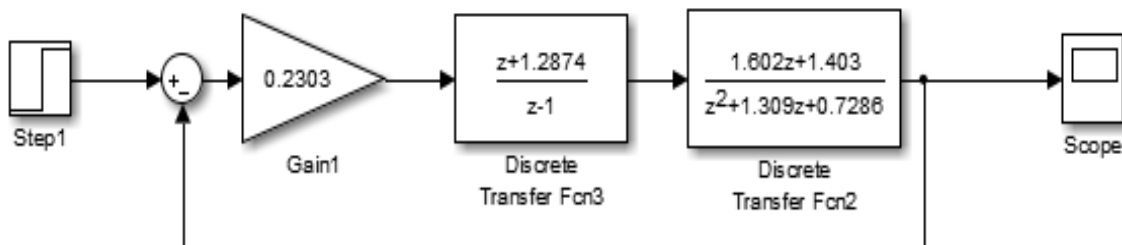


Figura 41. Función de transferencia y controlador PI en Simulink.

Fuente: Autores

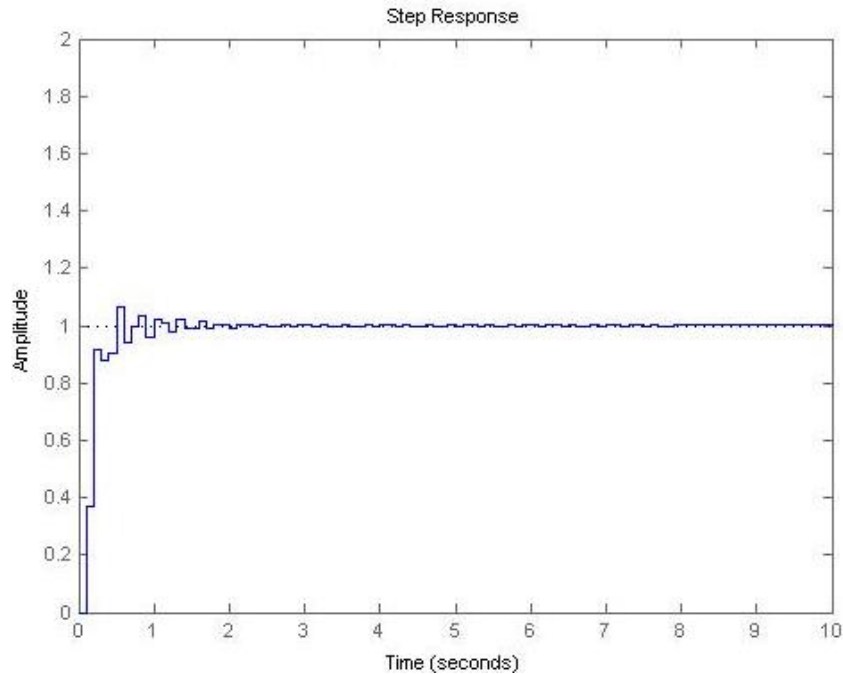


Figura 42. Comportamiento del controlador PI.

Fuente: Autores

Como se observa en la figura 42 el controlador PI cumple con las condiciones de diseño estipuladas, las cuales son el tiempo de establecimiento = 1 segundo y el máximo sobre impulso como se puede observar es menor del 10%.

La función de transferencia que corresponde al control y que será utilizada en el algoritmo de control realizado en Arduino es la siguiente:

$$G_{PI}(z) = 0.2303 \cdot \frac{z + 1.2874}{z - 1}$$

Con sus respectivos valores de K_p y K_i

$$K_i = 5.26043$$

$$K_p = -0.032722$$

Para poder programar esta ecuación en el Arduino es necesarios seguir los pasos explicados en la **Figura 43**.

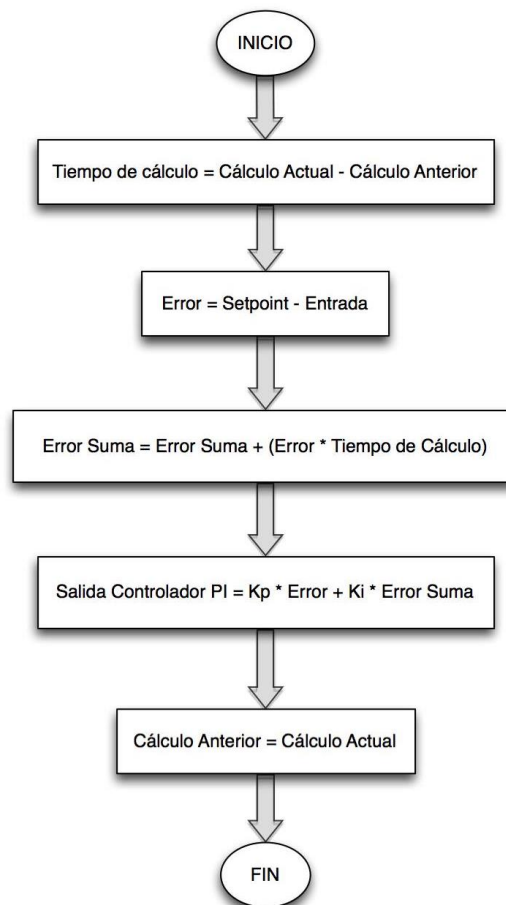


Figura 43. Diagrama de flujo parámetros de control en Arduino.

Fuente: Autores

La **Figura 44** contiene el diagrama final para realizar el control e integración con Arduino.

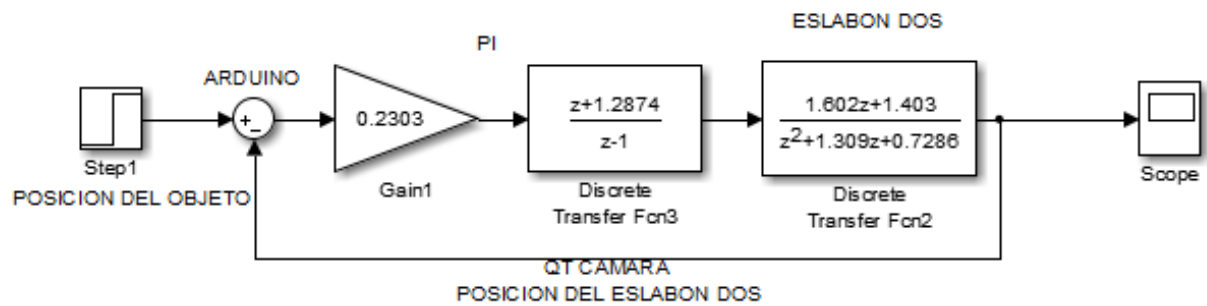


Figura 44. Diagrama final control.

Fuente: Autores

Como se puede observar en la **Figura 44** los datos que arrojará el PI dependen exclusivamente de la retroalimentación, es decir de la cámara, la cual extraerá la posición en X y Y del objeto a clasificar y estas mismas posiciones del efector final del eslabón 2. Con estas distancias se procede a hallar el error en el que está posicionado el eslabón 2 con respecto al objeto a clasificar y pasa por el control que arroja el valor del PWM para que el eslabón 2 se posicione en el lugar correspondiente.

6. RESULTADOS

Se realizaron pruebas de reconocimiento de 30 cuadrados y 30 círculos, de los cuales existían 15 de cada color para cada una de las formas. Respecto al reconocimiento de colores, como se tienen también 30 figuras de cada color, se obtuvo la matriz de confusión de la **Tabla 8**.

	Rojo	Verde	Azul
Rojo	28	2	0
Verde	4	25	1
Azul	2	4	24

Tabla 8. Matriz de confusión para colores.

En la **Tabla 8** se observa que 28 de las 30 figuras rojas se reconocieron como tales, mientras que 4 de las verdes se reconocieron como rojas y 1 como azul, y de los 30 azules 4 fueron identificadas como verdes y 2 como rojas, esto es debido a las condiciones de iluminación y el brillo generado al adquirir la imagen.

De la misma forma, la **Tabla 9** muestra la matriz de confusión para las formas reconocidas. Se aprecia que los peores resultados corresponden a la figura cuadrado, los 15 aciertos corresponden a más del 65%. Lo que indica que el reconocimiento de los cuadrados varía mucho debido a la iluminación y brillo. Los falsos positivos encontrados, obedecen a inconvenientes en el proceso de filtrado y dilatación de las imágenes que generaron más brillo en la adquisición.

	Cuadrado	Circulo
Cuadrado	20	10
Circulo	5	25

Tabla 9. Matriz de confusión para formas.

Las pruebas realizadas con respecto a las organización de las piezas se realizaron con 30 cuadrados y 30 círculos, de los cuales 15 eran de color rojo y 15 de color azul, en estas pruebas se tuvo en cuenta que reconociera bien el color, la forma, que llegara a la pieza correspondiente y que la clasificara en su debido cubículo, a continuación se describen los resultados.

Para los colores y las formas, los porcentajes de reconocimiento no variaron en gran cantidad, siempre y cuando la iluminación permanezca en perfecto estado la cámara reconocerá bien tanto formas como colores.

Con respecto a la ubicación del manipulador robótico en las piezas a clasificar, se requirió una segunda captura debido a que el movimiento del primer servo cambia el origen del segundo servo, por ende fue necesario esta segunda captura para conocer la posición en la que se encontraba el segundo eslabon con respecto al la pieza a clasificar y se comprobó que el control solucionado teóricamente funcionó muy bien en la práctica, ya que el manipulador clasificó las piezas rápido y con precisión, esto, siempre y cuando la cámara determinara los colores y formas correctamente. Se presentó una limitación con respecto a los ángulos, ya que los servos de 180° no permitían ángulos negativos los cuales eran necesarios en algunos casos.

Para la clasificación de la piezas con respecto a los cubículos, como son coordenadas fijas no se presentó ningún problema en el momento de clasificar las piezas, siempre y cuando los procesos anteriores resultaran exitosos.

7. CONCLUSIONES

Las capturas y procesamiento de imágenes depende mucho de la iluminación, ya que mejora o empeora los procesos que debe realizar el programa, debido a que las condiciones programadas se ven afectadas por dicha iluminación.

El software de procesamiento de imágenes empleado fue desarrollado en un pc, sin embargo el hecho de utilizar Open CV en un entorno de desarrollo libre permite que se pueda implementar en un dispositivo embebido como por ejemplo Raspberry.

El controlador PI implementado se desarrolló por el método del lugar geométrico de las raíces ya que este método es útil para analizar sistemas tipo SISO (Single Input Single Output).

Se demuestra en su análisis de deformación y tensión que el brazo robótico está diseñado para levantar cargas levemente pesadas, si este proyecto es considerado para la industria es necesario cambiar los materiales y el análisis de deformación y tensión dependiendo del objetivo a realizar.

Se decidió implementar los métodos geométricos para la generación de trayectoria debido a que el brazo robótico es de dos grados de libertad y la solución se presenta de una manera mas eficaz con respecto a matrices de transformación homogénea, que son comunmente mas utilizadas en manipuladores de mayor número de grados de libertad.

Se encontraron diversas limitaciones con respecto a la clasificación de las piezas debido a que los servos son de 180° y solo abarcan una región semi-circular de la maqueta (mayoría), exceptuando la región mas cercana al origen del manipulador.

Las soluciones de clasificación en procesos industriales mejoran su rendimiento mediante el uso del procesamiento de imágenes, ya que un solo sensor (cámara) puede determinar las características de distintas variables y de esta manera realizar una clasificación robusta con un sistema más fácil de implementar, siempre y cuando se cuente con un ambiente controlado para evitar cambios en la iluminación.

BIBLIOGRAFÍA

- [1] Conceptos básicos de Visión Artificial. Consultado en Octubre del 2015. Visión Artificial, Aplicación práctica de la Visión Artificial en el control de procesos industriales, Febrero 2012, Gobierno de España Ministerio de Educación Disponible en: http://visionartificial.fpcat.cat/wp-content/uploads/UD_1_didac_Conceptos_previos.pdf
- [2] Platero Dueñas Carlos. (2011). Universidad Politécnica de Madrid, Dpto. Electrónica, Automática e Informática Industrial, Robótica y Visión Artificial, Capitulo 1 Introducción a la Visión Artificial páginas 11 - 32. Disponible en: <http://www.elai.upm.es/moodle/mod/resource/view.php?id=131>
- [3] Peña, M., Juárez, I. L., Corona, J., & Ordaz, K. (2003, December). Visión para robots en tareas de ensamble. In Proceedings, of the Second National Congress on Mecathronics.
- [4] Fernández G, José Armando, José David Alvarado Moreno, Diego Mauricio Florián P, and César Augusto Sánchez S. working paper., Estudio Comparativo de Técnicas Espaciales para la Identificación de Defectos en Textiles, 2010. .
- [5] Corbalán, M. (n.d.), Dpt.E.E. EUPM. 08240 Manresa (Barcelona). Sistema de visión por computador para control industrial.
- [6] Hoyos, José G, Jaiber E Cardona, Luís M Capacho, and Mónica L Hernández. "Aplicación de técnicas de correspondencia para determinar la posición del efector final de un manipulador robótico." xv simposio de tratamiento de señales, imágenes y visión artificial, XV simposio de tratamiento de señales, imágenes y visión artificial – STSIVA 2010: 88-94.
- [7] Casnanzuela Guamushig, W. J; Vega Salazar, J. N. (n.d.). Diseño e implementación de un proceso de envasado de agua empleando el brazo robótico scorbot-er 4u intelitek del laboratorio de robótica sincro y servos de la espe sede Latacunga, Escuela politécnica del ejército sede Latacunga (2008).

- [8] Macías Loza, Leandro Danilo, Freddy Israel Mata Zavala, and Carlos Valdivieso. "Aplicación de Visión con LabVIEW para la Detección de Frascos Sin Tapa." Revista Tecnológica ESPOL.
- [9] Piñero, J; Velásquez, S. (n.d.). Control Brazo Robótico, Memorias de las IX Jornadas de Investigación 2011 – UNEXPO Puerto Ordaz - 6 al 9 de Julio de 2011.
- [10] García Enríquez, Irma Josefina, Ramírez Cortés Juan Manuel, Ibarra Bonilla Mariana Natalia, and Gómez Gil Ma. Del Pilar. "Seguimiento Autónomo de la Posición de un Objeto por Visión y Control Neurodifuso en MATLAB." 6th International Conference on Electrical and Electronics Engineering Research. : 335-340, Noviembre 2008.
- [11] Díaz Celis, César Augusto, and César Augusto Romero Molano. "Navegación de robot móvil usando Kinect, OpenCV y Arduino." Prospect. no. 1 (2012): pags 71 -78, Prospect. Vol. 10, No. 1, Enero - Junio de 2012, págs. 71-78.
- [12] Calderón, C. Ramirez, Tayron. Mora, Cristian. (n.d.). Control numérico de un brazo robótico, como respuesta a algoritmos de procesamiento de imágenes y sistemas expertos. Universidad Técnica Particular de Loja.
- [13] Aguilar Nieto, F. J. Chica Barrera, J. J. Mola Mateos, M.F. Andújar Márquez, J.M. (n.d.). Desarrollo de un instrumento virtual para el control remoto de un brazo manipulador de uso educacional. Universidad de Huelva, Dpto. de Ingeniería Electrónica, Sistemas Informáticos y Automática (DIESIA), Universidad de Huelva.
- [14] Basantes Ortiz, J. C. Torres Tufiño, F. R. (2009). Desarrollo de un sistema de control para un brazo robótico mediante adquisición y procesamiento de imágenes. Escuela Politécnica Nacional.
- [15] Tecnologías de la Información y de la Comunicación. Capítulo 6, Programación y control de procesos. Juan A. Alonso, Santiago Blanco A., Santiago Blanco S., Roberto escribano, Víctor R. González, Santiago Pascual, Amor Rodríguez. Editorial Ra-Ma 2004.
- [16] Conceptos básicos de QT Creator. Disponible en: <http://www.qt.io/services/>

- [17] Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. "O'Reilly Media, Inc."
- [18] J. A. Cortes, «Técnicas Alternativas para la conversión de imágenes a color a escala de grises en el tratamiento digital de imágenes,» Scientia et technica., vol. 1, nº 47, pp. 201-212, 2011.
- [19] Otsu, N. (1975). A threshold selection method from gray-level histograms. Automatica, 11(285-296), 23-27
- [20] Beer, F., Dorador G, J. and Murrieta, J. (2010). Mecánica de materiales. México: McGraw-Hill...
- [21] Reyes, F. (2012). MATLAB Aplicado a Robótica y Mecatrónica. México: Alfaomega Grupo Editor.
- [22] Mendoza Poveda, C. I. (2014). Modelado, simulación y control de un manipulador robótico de 2 grados de libertad empleando Adams/Matlab.
- [23] Urrea, E. C. y Kern, J., A New Model for Analog Servomotors. Practical Implementations, Canadian Journal on Automation, Control and Intelligent Systems 2, 29-38 (2011).
- [24] Urrea, E. C. y Kern, J., Modelamiento y simulación de un robot redundante de tipo manipulador SCARA, 831 (2011).

ANEXO A. Datasheet cámara



Logitech HD Webcam C270

Requisitos del sistema

Lo que necesita:

Windows Vista®, Windows® 7 (32 bits o 64 bits) o Windows® 8

Requisitos básicos:

- 1 GHz
- 512 MB de RAM o más
- Disco duro con 200 MB de espacio libre
- Conexión a Internet
- Puerto USB 1.1 disponible (se recomienda 2.0)

Para videoconferencias HD 720p y grabaciones de vídeo HD:

- Intel® Core™2 Duo a 2,4 GHz
- 2 GB de RAM
- Disco duro con 200 MB de espacio libre
- Puerto USB 2.0
- Velocidad de carga de 1 Mbps o superior
- Resolución de pantalla de 1280 x 720

Información de garantía

Dos años de garantía de hardware limitada

Esta caja contiene

- Cámara Web con cable de 1,5 m
- Documentación del usuario

Especificaciones

- Especificaciones:
 - Videoconferencias HD (1280 x 720 píxeles) con el sistema recomendado
 - Captura de vídeo: Hasta 1280 x 720 píxeles
 - Tecnología Logitech Fluid Crystal™
 - Fotos: Hasta 3.0 megapíxeles (mejora por software)
 - Micrófono integrado con reducción de ruido
 - Certificación USB 2.0 de alta velocidad (se recomienda)
 - Clip universal para monitores LCD, CRT o portátiles
- Software de cámara Web Logitech:
 - Controles de panorámico, inclinación y zoom
 - Captura de vídeo y fotos
 - Seguimiento facial
 - Detección de movimiento

Fuente: <http://www.logitech.com/es-roam/product/hd-webcam-c270>

ANEXO B. Datasheet Servo HS-805BB Hitec

Basic Information

Modulation: Analog

Torque: 4.8V:
275.0 oz-in (19.80 kg-cm)

6.0V:
343.0 oz-in (24.70 kg-cm)

Speed: 4.8V:
0.19 sec/60°

6.0V:
0.14 sec/60°

dWeight: 5.36 oz (152.0 g)

Length:
2.60 in (66.0 mm)

Dimensions: Width:
1.18 in (30.0 mm)

Height:
2.27 in (57.7 mm)

Motor Type: 3-pole

Gear Type: Plastic

Rotation/Support: Dual Bearings



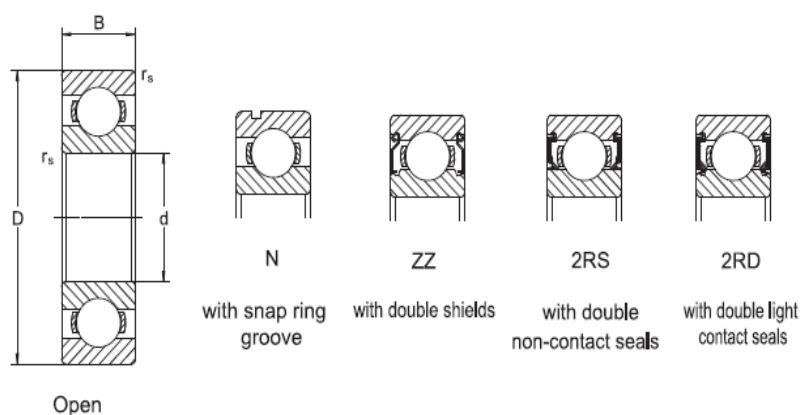
Fuente: <http://www.servodatabase.com/servo/hitec/hs-805bb>

ANEXO C. Datasheet rodamiento KML 6010



Metric Size Bearings

6700,6800,6900,6000,16000,6200,6300,6400 Series



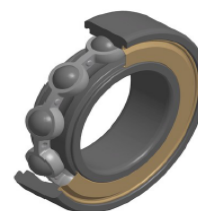
KML No.	Boundary dimensions mm				Basic load ratings N		Limiting speeds RPM		Mass Kg
							Lubrication		
	d	D	B	rs min	Dynamic Cr	Static Cor	Grease ZZ 2RS 2RD	Oil open	open type
6810	50	65	7	0.3	6600	6100	9600	11000	0.052
6910		72	12	0.6	15600	12200	8900	11000	0.132
16010		80	10	0.6	13200	11300	8400	9800	0.180
6010		80	16	1	21800	16600	8400	9800	0.261
6210		90	20	1.1	35000	23200	7100	8300	0.454
6310		110	27	2	62000	38500	6400	7500	1.070

Fuente: <http://www.kml-bearing.com/wp-content/uploads/2015/09/KML-Ball-Bearings-C2013.pdf>

ANEXO D. Datasheet rodamiento 6008 Z



Ficha técnica PDF 6008.Z

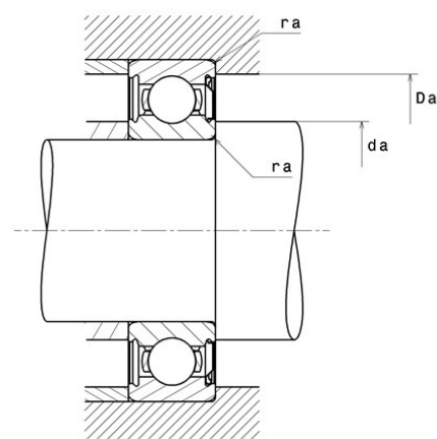
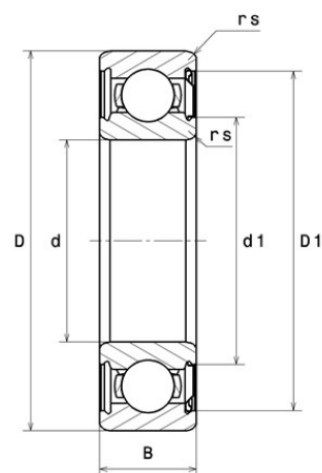


Rodamientos de bolas de 1 hilera

Rodamiento rígido de bolas, contacto radial, jaula de chapa, deflector en un lado

Características técnicas	
d	40 mm
D	68 mm
B	15 mm
d1	46,80 mm
D1	61,40 mm
rs min	1 mm
Clase de Juego Radial	CN
Peso	0,19 kg
Marca	SNR

Prestaciones	
Capacidad de carga dinámica C	16 400 daN
Capacidad de carga estática, C0	11 500 daN
Carga límite de fatiga, Cu	520 daN
f0	15.2
Nref	11 000 r.p.m.
Nlim	17 000 r.p.m.
Temperatura mínima, Tmin	-40 °C
Temperatura máxima, Tmax	120 °C
Frecuencia propia jaula, FTF	0,43 Hz
Frecuencia propia cuerpos rodantes, BSF	6,66 Hz
Frecuencia propia anillo exterior, BPFO	5,12 Hz
Frecuencia propia anillo interior, BPFI	6,88 Hz



Definición de ejes y apoyos

da min	45 mm
da max	46,80 mm
Da max	63 mm
ra max	1 mm

Coeficientes de cálculo

Carga radial dinámica equivalente

$$P = X \cdot Fr + Y \cdot Fa$$

$\frac{f_0 F_a}{C_0}$	e	Fa / Fr ≤ e		Fa / Fr > e	
		X	Y	X	Y
0.172	0.19	1	0	0.56	2.3
0.345	0.22				1.99
0.689	0.26				1.71
1.03	0.28				1.55
1.38	0.3				1.45
2.07	0.34				1.31
3.45	0.38				1.15
5.17	0.42				1.04
8.89	0.44				1

Equivalente carga radial estática

$$P_0 = X_0 \cdot Fr + Y_0 \cdot Fa$$

X_0	Y_0
0.6	0.5

En el caso de rodamiento individual o asociación DT :

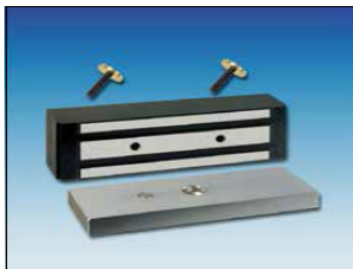
Si $P_0 < F_r$, entonces considerar $P_0 = F_r$

Fuente: <https://eshop.ntn-snr.com/es/6008-Z-646713.html>

ANEXO E. Datasheet electroimán

DICTATOR

Electroimanes rectangulares



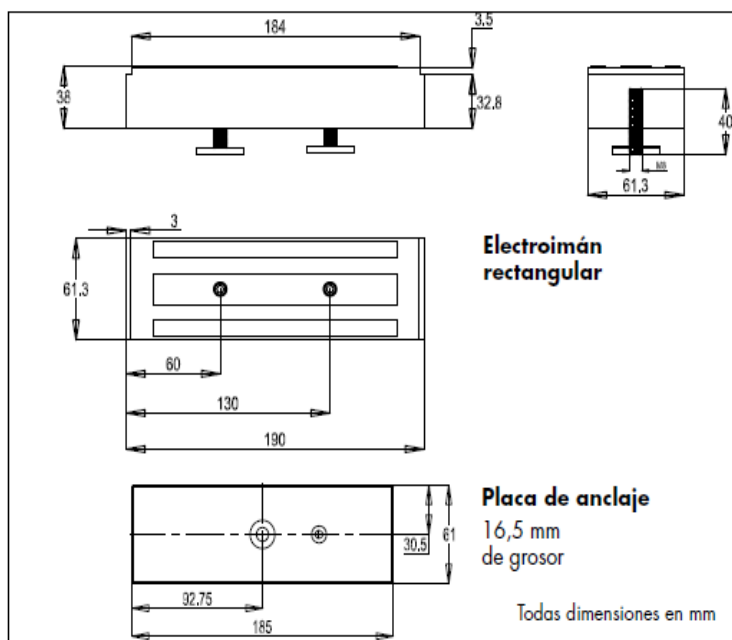
Electroimán rectangular FH550U fuerza 3600 N, montaje encastrado

El electroimán rectangular FH550U está disponible con y sin contacto de efecto Hall integrado. La ejecución con contacto de efecto Hall es intentado para sistemas de bloqueo o de esclusa, como ellos necesitan una información sobre el estado de la puerta.

Para la alimentación puede optar por 12 VDC o 24 VDC. El ajuste de la tensión necesaria se hace en la placa de conexión en posicionando puentes enchufables de manera correspondiente.

El montaje se hace por el lado posterior con dos tornillos de tope.

Dimensiones FH550U



Datos técnicos

Tensión de la conexión	12 VDC ± 10 %	24 VDC ± 10 %
Consumo	480 mA	240 mA
Fuerza de retención	3600 N	
Duración de la conexión	100 %	
Temperatura de trabajo	-10 °C hasta +55 °C	
Remanencia	0 N (por perno de empuje)	
Acabado	zincado	
Protección	IP 42	
Contacto relé	con o sin contacto de efecto Hall	
Indicador LED	ninguno	
Conexión eléctrica	placa de conexión	

Datos de pedido

** ensayado según EltVTR y aprobado con el terminal de salida de emergencia del sistema esclusa (certificado P-3250/08).*

Electroimán rectangular FH550U	Art. N° DD 040678SET
para montaje encastrado y con placa de anclaje	
Electroimán rectangular FH550UK *	Art. N° DD 040679SET
para montaje encastrado y con contacto de efecto Hall, placa de anclaje	

© DICTATOR Española, S.A.U. · C/Mogoda, 20 - 24 · Pol. Ind. Can Salvatella
08210 Barberá del Vallés/Barcelona · España
Tfn. 93/7191314 · Fax 93/7182509 · E-mail dictator@dictator.es · 1113

Página 08.045.00

Fuente: http://dictator.es/wp-content/uploads/2014/01/807-Es-131122_Electroimanes_rectangulares1.pdf

ANEXO F. Datasheet Arduino UNO



Technical specs

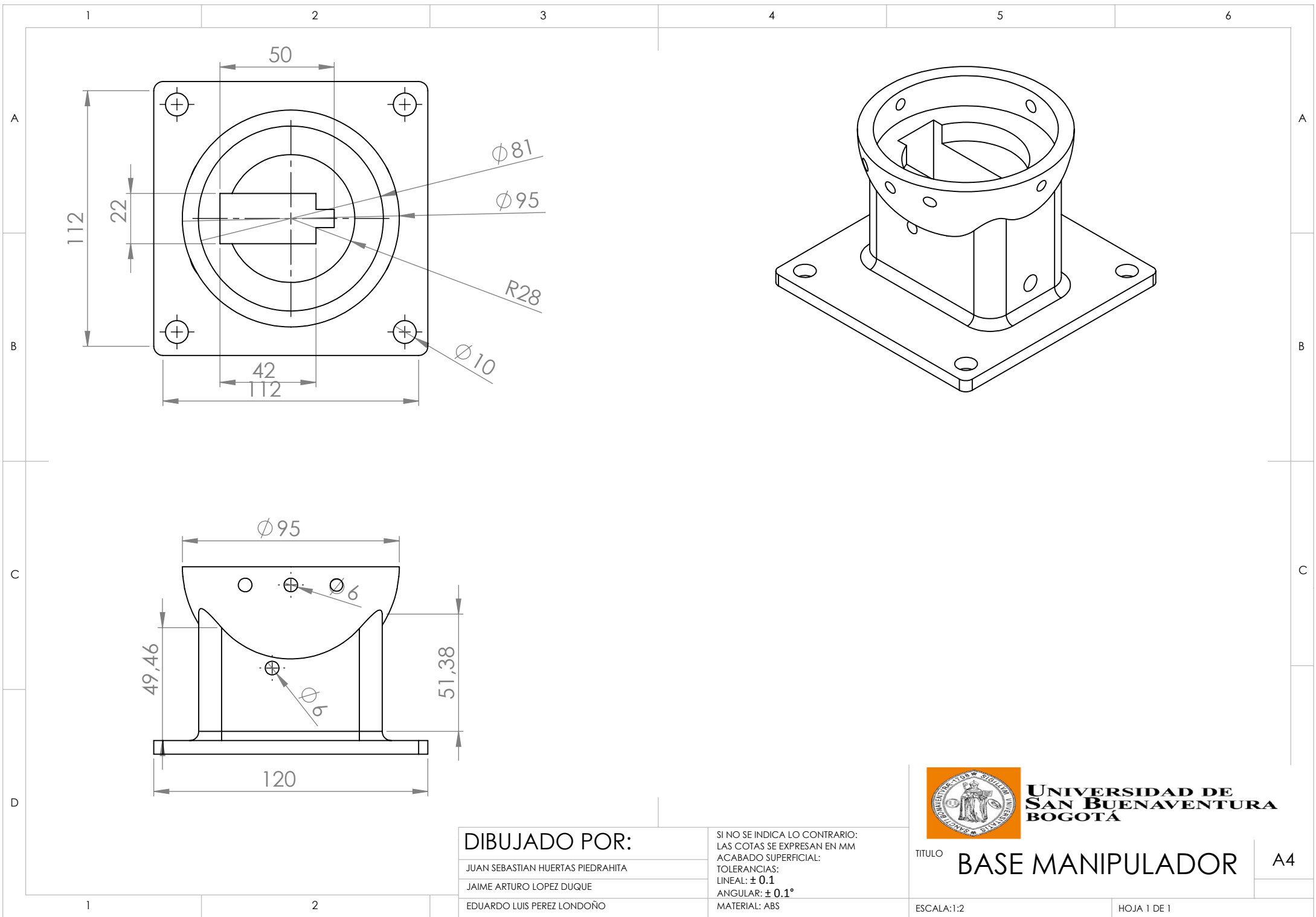
Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Atmega168 Pin Mapping

Arduino function				Arduino function
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13) analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12) analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11) analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10) analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9) analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8) analog input 0
VCC	VCC	7	22	GND GND
GND	GND	8	21	AREF analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5) digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4) digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3) digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1) digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Fuente: <https://www.arduino.cc/en/Main/ArduinoBoardMega>



DIBUJADO POR:

JUAN SEBASTIAN HUERTAS PIEDRAHITA

JAIME ARTURO LOPEZ DUQUE

EDUARDO LUIS PEREZ LONDOÑO



UNIVERSIDAD DE
SAN BUENAVENTURA
BOGOTÁ

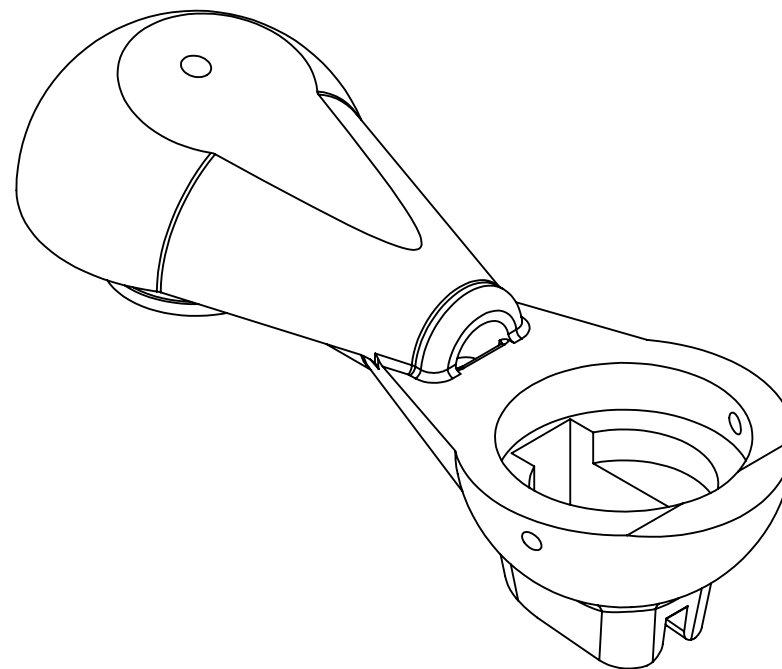
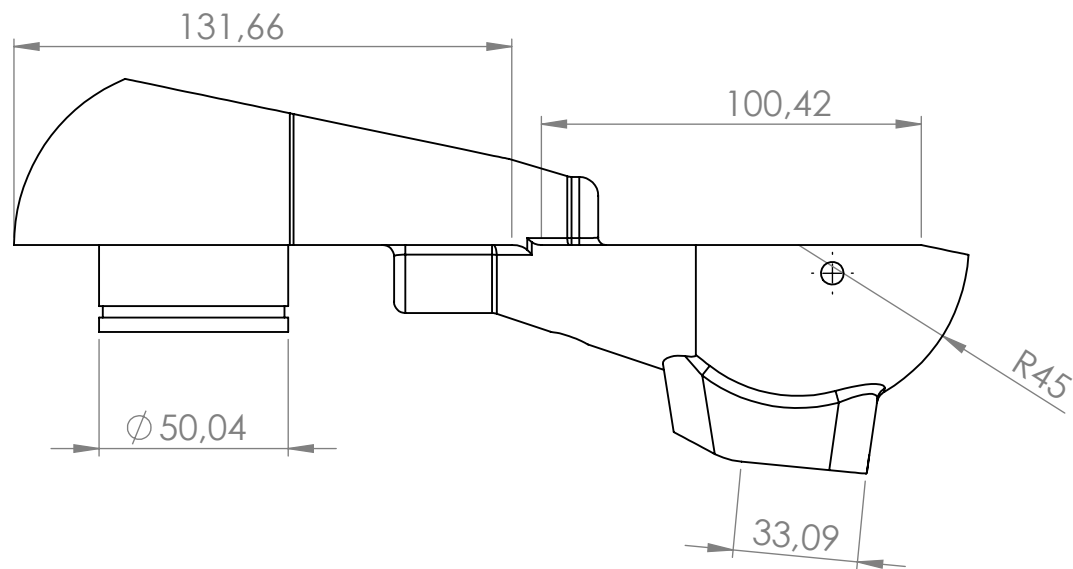
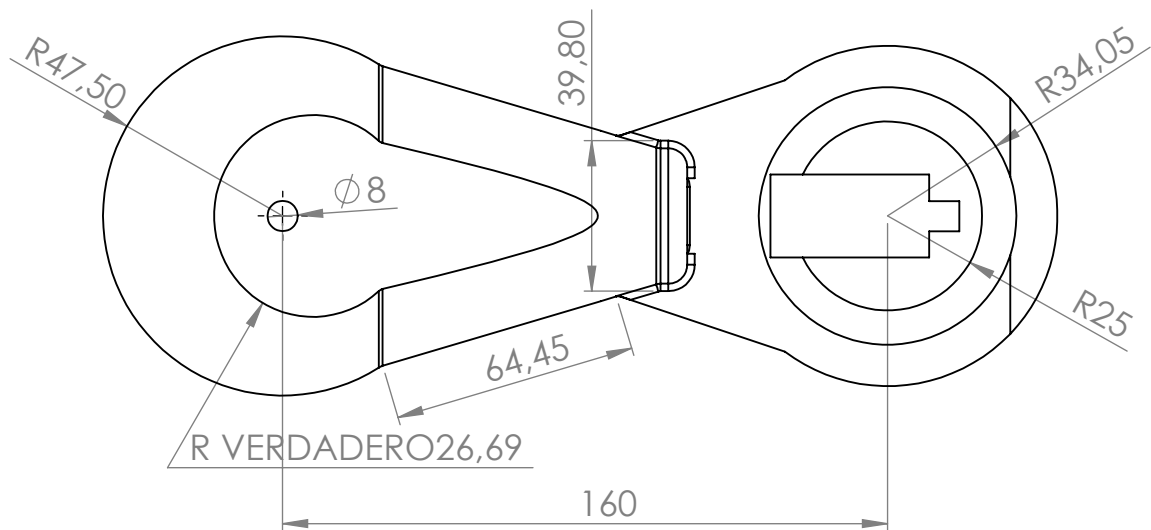
TITULO

BASE MANIPULADOR

A4

ESCALA: 1:2

HOJA 1 DE 1



DIBUJADO POR:

JUAN SEBASTIAN HUERTAS PIEDRAHITA

JAIME ARTURO LOPEZ DUQUE

EDUARDO LUIS PEREZ LONDOÑO

SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN MM
ACABADO SUPERFICIAL:
TOLERANCIAS:
LINEAL: ± 0.1
ANGULAR: $\pm 0.1^\circ$
MATERIAL: ABS



UNIVERSIDAD DE
SAN BUENAVENTURA
BOGOTÁ

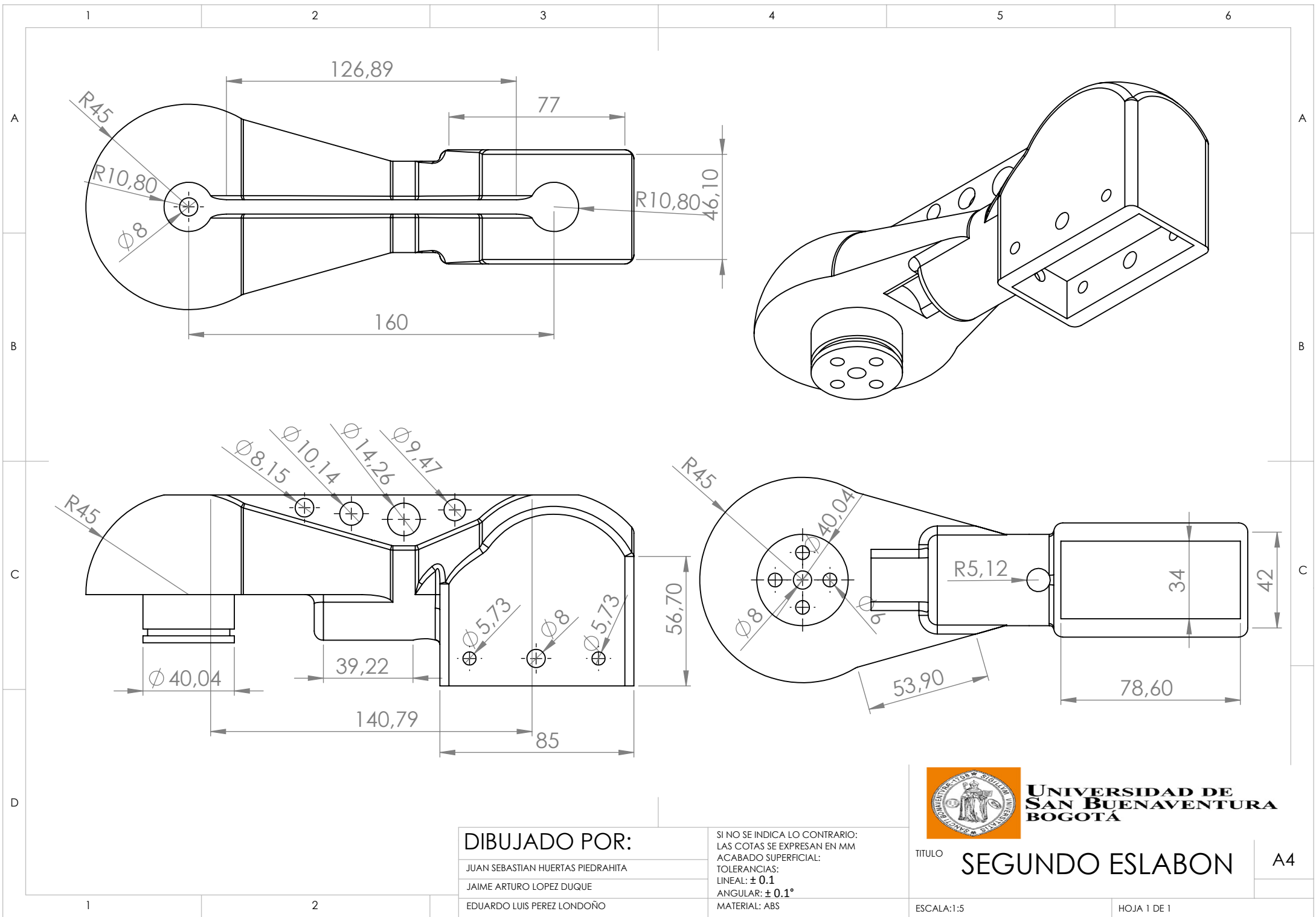
TITULO

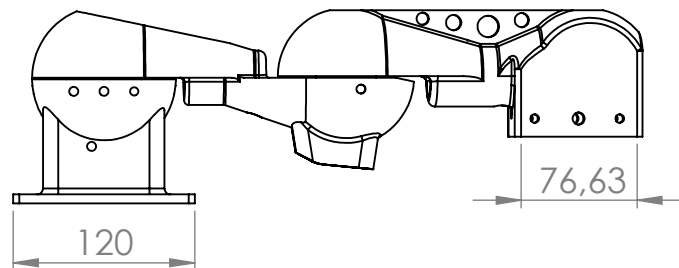
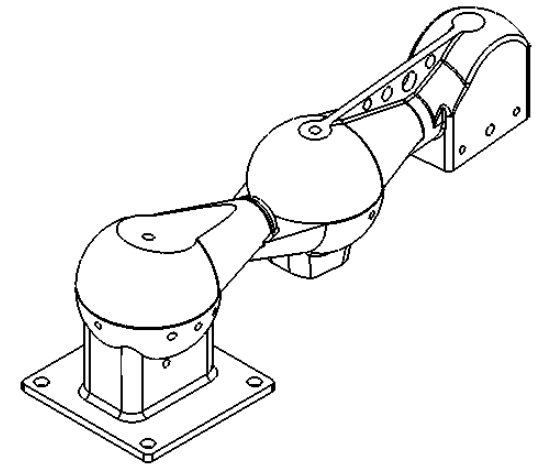
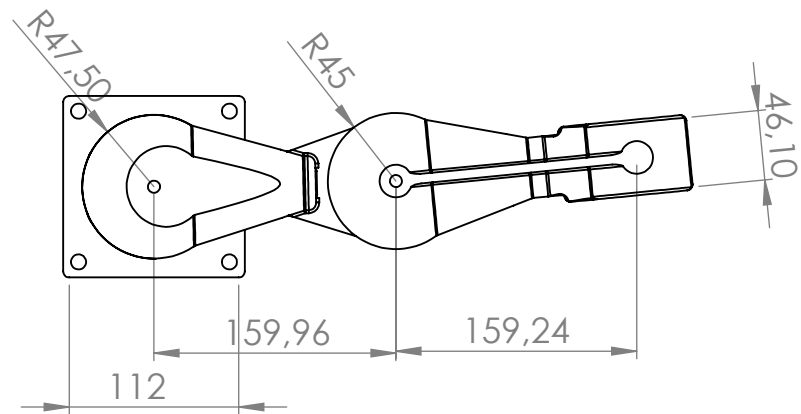
PRIMER ESLABON

A4

ESCALA: 1:5

HOJA 1 DE 1





DIBUJADO POR:

JUAN SEBASTIAN HUERTAS PIEDRAHITA

JAIME ARTURO LOPEZ DUQUE

EDUARDO LUIS PEREZ LONDOÑO

SI NO SE INDICA LO CONTRARIO:
LAS COTAS SE EXPRESAN EN MM
ACABADO SUPERFICIAL:
TOLERANCIAS:
LINEAL: ± 0.1
ANGULAR: $\pm 0.1^\circ$
MATERIAL: ABS



UNIVERSIDAD DE
SAN BUENAVENTURA
BOGOTÁ

TITULO

MANIPULADOR

A4

ESCALA:1:5

HOJA 1 DE 1