

A dark blue vertical bar is on the left. A blue arrow points right from it, containing the date.

23-6-2017

# Sistema De Identificación y seguimiento de objetos de colores RGB para una plataforma rotatoria mediante visión artificial empleando el software de MATLAB

Several thin, curved lines in dark blue and light grey originate from the bottom left corner.

***Rafael Jesús Acosta Jiménez***  
***Carlos Alberto Rodríguez Moreno***

**UNIVERSIDAD DEL MAGDALENA**

**Sistema De Identificación y seguimiento de objetos de colores RGB para una  
plataforma rotatoria mediante visión artificial empleando el software de MATLAB**

**Autor**

**Rafael Jesús Acosta Jiménez  
Carlos Alberto Rodríguez Moreno**

**Director**

**Omar Francisco Rodríguez Álvarez,  
Ms.C. en educación**

**Universidad del Magdalena**

**Facultad de Ingeniería**

**Ingeniería Electrónica**

**Santa Marta**

**Magdalena**

**2017**

**DIRECTOR:**

Omar Francisco Rodríguez Álvarez, Ms.C. en educación

**ASESOR:**

José Ramón Iglesias Gamarra, Ph.D. en Automatización, Robótica e Ingeniería en  
materiales

Carlos Mario Ariza Gallejo, Ingeniero Electrónico

## CONTENIDO

<b>1. RESUMEN DEL PROYECTO .....</b>	<b>7</b>
<b>2. DESCRIPCIÓN DEL PROYECTO.....</b>	<b>10</b>
2.1. Descripción Del Problema .....	10
2.2. ¿Qué Problemas Se Identificaron? .....	10
2.2.1. Planteamiento De La Pregunta Problema .....	11
<b>3. JUSTIFICACIÓN .....</b>	<b>12</b>
<b>4. ALCANCE Y LIMITACIONES .....</b>	<b>13</b>
4.1. Alcance .....	13
4.2. Limitaciones .....	13
<b>5. MARCO TEÓRICO .....</b>	<b>15</b>
5.1. Visión Artificial .....	15
5.1.1. Aplicaciones de la Vision Artificial .....	16
5.1.2. Etapas De La Visión Artificial .....	16
5.1.2.1. Adquisición de Imágenes.....	17
5.1.2.1.1. Pre- Procesamiento de Imágenes.....	17
5.1.2.1.2. Conversión a Escala de Grises .....	17
5.1.2.2. Descripción.....	18
5.1.2.2.1. Segmentación.....	18
5.1.2.2.2. Selección Óptima del Umbral .....	19
5.1.2.2.3. Operaciones Morfológicas .....	20
5.1.2.2.3.1. Dilatación .....	20
5.1.2.2.3.2. Erosión .....	21
5.1.2.2.4. Filtro de Tamaño .....	22
5.1.2.2.5. Eliminación del Borde.....	23
5.1.2.2.6. Etiquetado.....	24
5.2. Visión Estereoscópica.....	25
5.3. MatLab .....	27
5.3.1. Interfaz Gráfica GUI.....	28
5.3.1.1. Creación de una GUI de MatLab de forma interactiva .....	28
5.3.1.2. Creación de una GUI de MatLab de forma programática .....	29

5.4.	Tarjeta de control “hardware” .....	29
5.4.1.	Arduino .....	29
5.4.1.1.	Tipos de Arduino .....	31
5.4.1.1.1.	Duemilanove .....	31
	Memoria.....	32
	Comunicación.....	32
	Programación .....	33
5.4.1.1.2.	Mega .....	33
	Alimentación .....	33
	Memoria.....	34
	Programación .....	34
	Características físicas .....	34
5.4.1.1.3.	Nano.....	35
	Alimentación .....	35
	Memoria.....	35
	Comunicación.....	36
	Programación .....	36
5.4.1.1.4.	Pro.....	36
	Alimentación .....	37
	Memoria.....	37
	Características físicas .....	37
<b>6.</b>	<b>ESTADO DEL ARTE.....</b>	<b>38</b>
<b>7.</b>	<b>OBJETIVOS.....</b>	<b>40</b>
7.1.	OBJETIVO GENERAL.....	40
7.2.	OBJETIVOS ESPECÍFICOS.....	40
<b>8.</b>	<b>DISEÑO METODOLÓGICO .....</b>	<b>42</b>
	Fase I: Obtención de información, para el desarrollo del proyecto: .....	42
	Fase II: Diseño del modelo de funcionamiento del software a desarrollar y el sistema de motores: .....	43
	Fase III: Análisis de las variables críticas del proyecto y escogencia de los elementos óptimos de construcción. ....	43
	Fase IV: Implementación y desarrollo de pruebas .....	44

<b>9. CRONOGRAMA .....</b>	<b>45</b>
<b>10. IMPACTOS ESPERADOS .....</b>	<b>46</b>
<b>11. DESARROLLO DEL PROYECTO .....</b>	<b>47</b>
11.1.    IMPLEMENTACIÓN DE SOFTWARE .....	48
11.1.1.    Software MatLab .....	48
11.1.1.1.    Obtención del video .....	50
11.1.1.2.    Pre-procesamiento .....	51
11.1.1.2.1.    Escala de grises.....	51
11.1.1.2.2.    Segmentación.....	51
11.1.1.2.3.    Binarización .....	52
11.1.1.2.4.    Filtrado de la imagen.....	53
11.1.2.    Software arduino.....	53
11.1.3.    Requisitos Para El Procesamiento .....	53
11.2.    Implementación De Hardware .....	55
11.2.1.    Dispositivo De Adquisición .....	55
11.2.2.    Sistema Rotatorio Con Dos Grados De Libertad .....	55
11.2.3.    Servomotores.....	56
11.2.4.    Tarjeta De Desarrollo Programable Arduino .....	57
<b>12. PRUEBAS Y RESULTADOS .....</b>	<b>59</b>
12.1.    EJECUCION INTERFAZ GRÁFICA DEL PROYECTO .....	59
12.2.    Escala de gris .....	61
12.3.    Segmentación.....	61
12.4.    Binarización.....	62
12.5.    Filtrado de la imagen.....	62
12.6.    Diagrama General De Comunicación .....	63
12.7.    Pre Procesamiento En MatLab .....	64
12.8.    Control Para El Movimiento .....	65
<b>13. Distancia Entre La Cámara Y Objeto A Seguir. ....</b>	<b>66</b>
13.1.    Tablas de Pruebas .....	66
<b>14. PRESUPUESTO .....</b>	<b>73</b>

14.1.	Recursos Humanos.....	73
14.2.	Recursos Institucionales.....	74
14.3.	Recursos Adicionales.....	74
14.3.1.	Presupuesto En Hardware.....	74
14.3.2.	Presupuesto En Software .....	75
14.4.	Presupuesto Varios .....	75
14.5.	Costo Total Del Proyecto.....	75
<b>15.</b>	<b>CONCLUSIONES .....</b>	<b>76</b>
<b>16.</b>	<b>REFERENCIAS BIBLIOGRAFICAS .....</b>	<b>78</b>
16.1.	Bibliografía .....	78
16.2.	Webgrafía.....	80
<b>17.</b>	<b>ANEXOS .....</b>	<b>81</b>
17.1.	Guía Para Realizar El Enlace Arduino MatLab.....	81
17.1.1.	Preguntas Frecuentes.....	82
17.1.2.	Procedimiento.....	82
17.2.	Manual de uso del software.....	84
17.2.1.	Funcionamiento del software .....	84
17.3.	Esquema de conexión de hardware. ....	90
17.4.	Imágenes Del Prototipo Terminado .....	91
17.5.	Colores usados en las pruebas .....	92
<b>18.</b>	<b>RECOMENDACIONES .....</b>	<b>98</b>
<b>19.</b>	<b>TRABAJO FUTURO .....</b>	<b>98</b>

## ÍNDICE DE IMÁGENES

Figura 1. Esquema de Bloques del sistema de Visión Artificial <sup>4</sup> .....	16
Figura 2. Ejemplo comparativo de una Imagen RGB y otra en escala de Grises <sup>6</sup> .....	18
Figura 3. Métodos de Segmentación <sup>7</sup> .....	19
Figura 4. a) Imagen binaria original; b) Imagen dilatada con un EE en forma de disco y de radio 2;.....	22
Figura 5. a) Imagen binaria dilatada; b) imagen filtrada por área $\leq 100$ pixeles <sup>12</sup> ....	23
Figura 6. a) Imagen resultante etapa anterior; b) Imagen sin borde <sup>13</sup> .....	24
Figura 7. Portada de Software MatLab.....	49
Figura 8. Interfaz Única de Video.....	50
Figura 9. Información del Equipo donde se ejecuta el Programa. ....	54
Figura 10. Cámara Star Tec.....	55
Figura 11. Base Servo motor .....	56
Figura 12. Servomotores.....	56
Figura 13. Placa Arduino UNO.....	58
Figura 14. Interfaz.....	59
Figura 15 Imagen Original sin Escala de Grises .....	61
Figura 16 Imagen en Escala de Grises .....	61
Figura 17 Imagen en Escala de grises .....	61
Figura 18 Imagen Segmentada (extracción del Color Rojo) .....	61
Figura 19 Imagen Binarizada .....	62
Figura 20 Imagen Segmentada.....	62
Figura 21 Imagen Filtrada .....	62
Figura 22 Esquema general de Funcionamiento.....	63
Figura 23. Interfaz de Usuario (UI).....	64
Figura 24. Interfaz Arduino.....	65
Figura 25 Serial Port No configurado y después de la configuración .....	86
Figura 26 Selección de colores .....	87
Figura 273 listo para presionar start.....	87
Figura 28 .....	88
Figura 29 .....	88
Figura 30 .....	89
Figura 31 Esquema de Hardware .....	90
Figura 32 paleta RGB de color rojo 1 .....	92
Figura 33 paleta RGB de color rojo 2 .....	93
Figura 34 paleta RGB de color Verde 1 .....	94
Figura 35 paleta RGB de color Verde 2 .....	95
Figura 36 paleta RGB de color Azul 1 .....	96
Figura 37 paleta RGB de color azul .....	97



## **1. RESUMEN DEL PROYECTO**

Es preciso reconocer que hoy por hoy la visión artificial, también conocida como visión por computador, es sin duda alguna, uno de los campos más interesantes y revolucionarios de la robótica y su estrecha relación con la inteligencia artificial lo hace verdaderamente interesante.

Actualmente, ya no hace falta dominar un lenguaje computacional para interactuar con los sistemas inteligentes, puesto que los sistemas de inteligencia artificial ya son capaces de entender y procesar nuestras órdenes empleando el lenguaje natural. Estamos acostumbrados a ver equipos inteligentes completamente autónomos, en entornos estrictamente industriales ya sea para la clasificación de formas o de objetos, control de nivel, seguimiento de objetos y personas entre otros, o por lo menos altamente controlados como en fábricas de automóviles. La aplicación de estos sistemas en entornos más dinámicos como nuestros hogares u oficinas, necesita de una enorme cantidad de sensores que permita al sistema inteligente percibir al medio con toda su complejidad para interactuar con él.

Igual que en los humanos, la entrada de datos más importante es la visión. En un sistema inteligente, normalmente se trata de una cámara compuesta por un lente o por una multitud de lentes, además de contar con sensores específicos para cada sistema y campo de aplicación. Una vez realizados todos los ajustes mecánicos, la visión artificial permite al sistema detectar la mayoría de los objetos para decidir cómo reaccionar y actuar frente a ellos. Este proceso está dotado de una inmensa complejidad, debido a la infinidad de situaciones u objetos, colores y formas a los que el sistema se puede enfrentar.

Una de las aplicaciones que tiene la visión artificial y la cual es el objetivo de este documento, es el reconocimiento de colores RGB lo cual traduce, color rojo, verde y azul.

Este documento de grado se ha dividido en cinco (5) ítems o capítulos, con el fin de organizar y entender mejor la relación que hay entre cada uno de ellos.

El primero hace referencia al título que debe tener el proyecto de grado, además de ser parte fundamental del objetivo general el cual enmarca la idea general de lo que se pretende desarrollar.

En el segundo ítem o capítulo, se podrá apreciar el resumen del proyecto el cual explica de manera resumida sobre qué consiste el trabajo.

En el tercer ítem el cual es uno de gran importancia, se dará a conocer la descripción del proyecto el cual abarca nueve (9) subcategorías las cuales brindaran una mayor información con el fin de entender y dejar más claro el contenido de este proyecto de grado.

En las subcategorías se expondrán el planteamiento del problema, en el cual se describe el problema detectado con la solución, al igual que la justificación, los alcances y limitaciones, el marco teórico, el estado del arte, los objetivos específicos y el general, además del diseño metodológico explicado en fases, complementándose con un cronograma de actividades detallado en semas y por último los impactos esperados.

En el cuarto inciso, se detalla el presupuesto de este anteproyecto, el cual tiene en cuenta los recursos humanos que describen a las personas que hacen parte de este proyecto, con su formación académica y su función dentro del mismo, los recursos institucionales con los que cuenta la universidad, los recursos adicionales, el presupuesto en hardware y en software, además el presupuesto varios el cual contiene gastos cotidianos como papelería, internet entre otros.

Finalmente el quinto apartado, tendrá las referencias bibliográficas y referencias de páginas web vistas y revisadas para poder dar construcción a este documento.

Siendo así, lo que este proyecto pretende es poder identificar objetos de cualquier forma por medio de una cámara, la cual cumplirá la función de sensor visual, es decir un ojo, pero con la particularidad y es que sean objetos o formas de colores específicos tales como el (Rojo), (verde) y (Azul), posteriormente después de esta identificación se realizará un seguimiento en tiempo real para evitar perderle de vista. Este seguimiento se hará posible ya que la cámara deberá ser instalada en una plataforma rotatoria la cual contará con dos servos motores que le permita realizar movimientos en dos direcciones (X, Y). a su vez estos elementos conocidos como servo motores dependerán de un driver y placa de desarrollo la cual recibirá las órdenes del algoritmo desarrollado en el software MatLab, el cual es parte indispensable, por lo que este algoritmo será el responsable de la identificación de los elementos de colores RGB.

## **2. DESCRIPCIÓN DEL PROYECTO**

### **2.1. Descripción Del Problema**

La tecnología de visión es una disciplina relativamente reciente que tuvo su introducción en la industria en la década de los 80. Aplicada en sistemas basados en la tecnología de los PC, se beneficia de los rápidos avances informáticos y de redes.

Se puede definir la “Visión Artificial” como un campo de la “Inteligencia Artificial” que, mediante la utilización de las técnicas adecuadas, permite la obtención, procesamiento y análisis de cualquier tipo de información especial obtenida a través de imágenes digitales. La visión industrial o Visión Artificial aplicada a la industria abarca la informática, la óptica, la ingeniería mecánica y la automatización industrial. A diferencia de la Visión Artificial académica, que se centra principalmente en máquinas basadas en el procesamiento de imágenes, las cuales si son enfocadas de manera correcta pueden dar origen a la elaboración de buenos proyectos investigación <sup>1</sup>.

### **2.2. ¿Qué Problemas Se Identificaron?**

A nivel académico observamos que la universidad del magdalena con su grupo de investigación MAGMA Ingeniería – Matemáticas aplicada a la ingeniería Universidad del Magdalena; y el Grupo de Investigación en Electrónica y Telecomunicaciones IET-UAC de la Universidad Autónoma del Caribe, pretenden llevar a cabo la realización de un proyecto en común que tiene como título: Identificación de objetivos y asignación de tareas a un agente robótico mediante visión estereoscópica. Este proyecto se verá enfocado en las Líneas de Investigación de: Automatización, Robótica y Procesamiento Digital de Imágenes de ambas instituciones.

<sup>1</sup> [http://dmi.uib.es/~ygonzalez/VI/Material\\_del\\_Curso/Teoria/Aplicaciones\\_VC.PDF](http://dmi.uib.es/~ygonzalez/VI/Material_del_Curso/Teoria/Aplicaciones_VC.PDF)

Como el proyecto de investigación de las dos universidades pretende basar toda su investigación con la estereovisión tradicional, y citando su planteamiento del problema el cual trata de que *“la estereovisión tradicional utiliza un par de imágenes estereoscópicas para realizar el análisis de profundidad de una escena basándose en características de forma o luminosidad. Y donde la información de movimiento adquiere un valor importante a la hora de eliminar falsas correspondencias. Si observamos que la estereovisión biológica depende directamente de la posición relativa a la convergencia de los ojos, el desplazamiento horizontal de las imágenes permite analizar la disparidad de cada uno de los píxeles.*

*Teniendo en cuenta la amplia información de entrada lo que pretendemos sería centrar la atención en aquellos objetos que sean de interés de acuerdo a unos determinados criterios, y conseguir mantener dicha atención a lo largo del tiempo, resolviendo la tarea de la visión selectiva. En esta primera aproximación los objetos que resultan de interés son aquellos en los que se detecta movimiento.*

*Por ello, la motivación principal de este proyecto de investigación se basa en visión estereoscópica y atención visual, con el fin de focalizar la atención en los objetos que nos resulten de interés, para discriminar situaciones y comportamientos predefinidos en escenarios complejos, utilizando pares de cámaras situados en unidades robóticas.”*

De lo descrito anteriormente, sobre lo que pretenden desarrollar las dos instituciones, nos inspiramos en la construcción de un proyecto que sirva como guía base para su futura implementación y dar parte a la solución generando así el planteamiento de la pregunta problema propuesta a continuación.

#### 2.2.1. Planteamiento De La Pregunta Problema

¿Qué aplicación software-hardware se puede desarrollar para poder centrar la atención y hacerle seguimiento en tiempo real a aquellos objetos que sean de interés de acuerdo a unos determinados criterios que para este caso son elementos de colores RGB, y lograr mantener dicha atención a lo largo del tiempo?

### 3. JUSTIFICACIÓN

Es importante recalcar que la visión artificial es la tecnología que permite detectar luz e interpretarla; logrando así, convertir cámaras pasivas en dispositivos inteligentes capaces de detectar eventos específico y posteriormente tomar decisiones que se encuentren programadas en su algoritmo base de procesamiento. Actualmente las ciudades se llenan de ojos artificiales que deben protegernos, pero esta protección no siempre es real. ¿De qué sirven tantas cámaras y tantas grabaciones de videos si nadie las mira?

Detección y clasificación de formas concretas, seguimientos de personas, seguimiento de objetos con características específicas, por citar un ejemplo “sean de colores como: El rojo, azul, amarillo, verde, etc.” en el campo visual; inteligencia artificial para el control y gestión de alarmas, son unas de la tantas tareas que pueden ser desarrolladas en este ámbito tecnológico, los cuales no se pueden notar en los sistemas actuales de la ciudad de Santa Marta.

Cabe resaltar, que la construcción de este trabajo de fin de carrera permitirá a la universidad del magdalena y específicamente al programa de ingeniería electrónica, contar con un documento de referencia que aborde esta línea de investigación sobre la temática de visión artificial, además que permita a futuros estudiantes fortalecer conocimientos en las áreas de programación y procesamiento de señales.

Por tal razón, lo que se pretende con este trabajo, es elaborar un aplicativo de la visión artificial que sirva como base y apoyo para la futura creación del proyecto de investigación que pretenden realizar la universidad del magdalena en conjunto con la universidad autónoma del caribe de la ciudad de barranquilla, el cual lleva como título Identificación de objetivos y asignación de tareas a un agente robótico mediante visión estereoscópica. Esto se llevaría a cabo con la solución a la pregunta problema que hace referencia sobre qué aplicación software-hardware se puede desarrollar para poder centrar la atención y hacerle seguimiento a aquellos objetos que sean de interés de

acuerdo a unos determinados criterios que en este caso son elementos de colores RGB, y mantener dicha atención a lo largo del tiempo que se esté ejecutando el aplicativo.

## **4. ALCANCE Y LIMITACIONES**

### **4.1. Alcance**

Este proyecto tiene como alcance desarrollar un aplicativo por software más un algoritmo de identificación y seguimiento de objetos de colores RGB, cuya naturaleza permitirá enviar órdenes a una tarjeta de desarrollo la cual se encontrara instalada en una plataforma mecánica rotatoria, que está conformada por dos servos motores los cuales hará que sea posible visualizar el movimiento mediante el análisis de imágenes que representan a objetos con características específicas, en este caso: colores en el rango de la paleta de colores RGB, trabajo que lleva como título Sistema De Identificación y seguimiento de objetos de colores RGB para una plataforma rotatoria mediante visión artificial empleando el software de MatLab.

De los resultados que arroje esta investigación, se podrá tener un documento que sirva como un fortalecimiento en el contenido de las materias de procesamiento de Señales y Robótica del programa de Ingeniería Electrónica de la Universidad del Magdalena.

### **4.2. Limitaciones**

Durante la realización de este proyecto, se darán a conocer algunas de las limitantes presentadas las cuales son:

- La complejidad de construcción del autómata, obligara a la mayor practicidad del mismo dejándolo más en una fase de prototipo que final.
- Como el sistema va a contar con una cámara de baja resolución, es posible que a grandes distancias el sistema no responda adecuadamente y no funcione de manera correcta. El rango de la distancia de operación a la que debe estar el prototipo del objeto que mide 10 x 10 cm es de 0.60 a 3.60 metros; superar esa distancia no garantiza que el dispositivo funcione de manera idónea.
- El proyecto no funcionará en ambientes de oscuridad o de poca iluminación, ya que la falta de ella afecta las propiedades del color, causando así que la identificación no sea posible; esto significa que para un funcionamiento correcto del dispositivo, se debe contar con un fondo blanco en el cual sea fácil resaltar las características del color del objeto y una iluminación con lámparas o focos de luces blancas
- Este dispositivo funcionará de manera limitada en algunos casos nula, para hacer seguimiento a objetos de colores en CMYK los cuales son otro tipo de formato de colores.
- Se debe colocar un solo objeto en el campo de visión, ya que al colocar dos o más del mismo color al que se le esté realizando el seguimiento, puede causar conflictos y más si los elementos se están moviendo en direcciones distintas.
- Toda la programación de procesamiento de video del proyecto, se encuentra desarrollada en el software de MatLab 2017, La razón por la cual solo funcionará en esta versión, es por una nueva instrucción desarrollada e implementada que lleva como nombre JSONLAB v1.5 la cual fue realizada el 01/02/2017 y consiste en un nuevo método para empaquetamiento de datos. Aquellos ordenadores que no cuenten con la nueva versión de MatLab 2017, no podrá ver el funcionamiento del prototipo.



- Los giros o movimientos realizados por el prototipo están limitados a las características de los servomotores que en este caso poseen un ángulo de rotación de 0° - 120° y al tipo de estructura utilizada, evitando así que se realicen movimientos de 360° en cualquiera de los dos ejes.

## 5. MARCO TEÓRICO

### 5.1. Visión Artificial

Uno de los propósitos de la visión artificial es lograr que un computador o robot "entienda" una escena o las características de una imagen y que este entendimiento le sea útil para llevar a cabo una misión. La visión estereoscópica es una de las técnicas de visión artificial donde existe más de una vista de una escena, tomada desde diferentes puntos. A través de estas imágenes se extraen las características tridimensionales de la escena y se hace el estudio pertinente a la aplicación en este caso gracias a la obtención de mapa de disparidades y el posterior tratamiento basándonos en colores<sup>2</sup>

Nuestros ojos proveen una gran cantidad de información del mundo externo, según Buhmann<sup>3</sup>, con nuestra visión logramos enterarnos de todos los objetos y seres vivientes que nos rodean y así representarlos en nuestro cerebro. Para realizar un reconocimiento de objetos, se requiere la asociación de una imagen con una memoria de ese objeto llamado modelo; estos no son innatos y tampoco son a priori, ya que nuestro cerebro no reconoce cosas que no hayamos visto antes, estos modelos se construyen a partir de una diaria experiencia visual.

Con todo lo anterior se evidencia que la visión trae beneficios para los seres vivos y esto ha motivado a investigar sobre los sistemas artificiales para el reconocimiento de imágenes. En una misma escena de visión artificial se presenta una gran cantidad de imágenes diferentes teniendo en cuenta la iluminación y el punto de vista.

<sup>2</sup> PEREZ, Lina M. Sistema de Visión Estereoscópica en El Proyecto Libélula. 29.11.2011

<sup>3</sup> BUHMANN, Joachim M.; MALIK, Jitendra; PERONA, Pietro. Image recognition: Visual grouping, recognition, and learning. PNAS \_ December 7, 1999 vol. 96 \_ no.25 \_ 14203 –14204. [www.pnas.org](http://www.pnas.org)

### 5.1.1. Aplicaciones de la Vision Artificial

Existen técnicas computacionales utilizadas en la visión artificial las cuales se combinan para generar la descripción correcta de una escena. El avance de esta potencia computacional a nivel hardware ha facilitado el abordaje de tareas complejas y la creación de muchos productos gracias a la visión computarizada, como por ejemplo:

- Buscadores de imágenes en base de datos de videos o en banco de imágenes modelos.
- Sistemas de vigilancia inteligente.
- Vehículos autónomos guiados por visión.
- Reconocimiento de rostros, de iris y de huellas dactilares.
- Reconocimiento de manuscritos (OCR).
- Identificación de ambientes para autómatas.
- Identificación de objetos en líneas de ensamble industrial.
- Control de calidad en proceso de productos.

### 5.1.2. Etapas De La Visión Artificial

El sistema de visión artificial está compuesto por cinco etapas fundamentales, las cuales son ejecutadas secuencialmente: Adquisición, Procesamiento, Descripción, identificación y Salida.(vease la Figura 1)



**Figura 1. Esquema de Bloques del sistema de Visión Artificial <sup>4</sup>**

<sup>4</sup> Acosta L(2009). *La visión Artificial*. Universidad Tecnologica de Bolivar, Cartagena, Colombia.

Los sistemas de visión artificial emplean métodos y técnicas distintas, los bloques de procesamiento, descripción e identificación pueden manipular conceptos variados en campos como matemática estocástica, probabilidad, inteligencia artificial, lógica, procesamiento de señales y biología.

#### 5.1.2.1. Adquisición de Imágenes

Esta etapa es la encargada de digitalizar las señales eléctricas análogas que vienen desde el dispositivo de captura, este proceso es muestrear, cuantificar, y adaptar la señal capturada para el pre-procesamiento, en la mayoría de los casos este proceso puede encontrarse inmerso en la cámara de captura y ésta entrega la señal ya tratada al ordenador.<sup>5</sup>

##### 5.1.2.1.1. Pre- Procesamiento de Imágenes

Es la transformación de la imagen original en otra imagen en la cual hayan sido eliminados los problemas de ruido granular de cuantización o de iluminación espacialmente variable. La utilización de estas técnicas permite el mejoramiento de las imágenes digitales adquiridas de acuerdo a los objetivos planteados en el sistema de visión artificial.<sup>6</sup>

##### 5.1.2.1.2. Conversión a Escala de Grises

Concepto implementado de forma exclusiva por las imágenes digitales bajo el cual una imagen independiente del formato y del tamaño en pixeles, es transformada mediante una relación preestablecida, de su apariencia original a un valor equivalente en una graduación de grises variando gradualmente del negro más profundo al blanco.<sup>6</sup>

<sup>5</sup> Rincones G. & Ruiz M. (2013). *Desarrollo de un prototipo electrónico basado en vision Artificial para la estimacion de la densidad de usuarios en Playas turísticas* (Tesis de Grado)(p. 55). Universidad del Magdalena, Santa Marta, Colombia.

<sup>6</sup> Rincones G. & Ruiz M. (2013). *Desarrollo de un prototipo electrónico basado en vision Artificial para la estimacion de la densidad de usuarios en Playas turísticas* (Tesis de Grado)(p. 56). Universidad del Magdalena, Santa Marta, Colombia.

Como sabemos el ojo percibe distintas intensidades de luz en función del color que se observe, esto es debido a la respuesta del ojo al espectro visible, por esa razón el cálculo de la escala de grises o luminancia de la imagen debe realizarse como una media ponderada de las distintas componentes de color de cada pixel.<sup>6</sup>

Como la variación de tonos grises se aplica a imágenes digitales, esta escala emplea un rango de variación de un octeto (8 Bits) permitiendo así entre el negro profundo y el blanco una gama de 256 posibles tonalidades de grises.<sup>6</sup>



**Figura 2. Ejemplo comparativo de una Imagen RGB y otra en escala de Grises<sup>6</sup>**

#### 5.1.2.2. Descripción

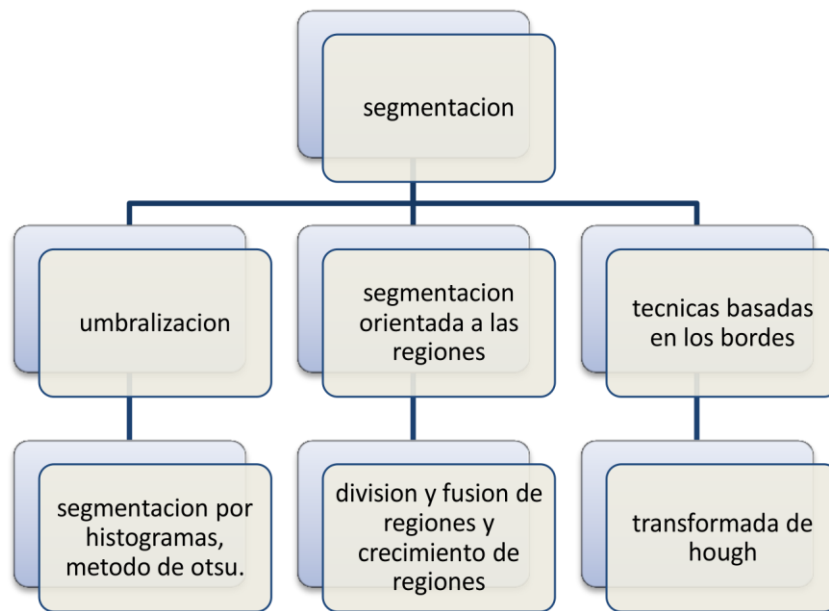
##### 5.1.2.2.1. Segmentación

La segmentación es el proceso mediante el cual una imagen se descompone en regiones o elementos que pueden corresponder a objetos o parte de objetos. El proceso de segmentación se encarga de evaluar si cada pixel de la imagen pertenece o no al objeto de interés.

---

<sup>6</sup> Rincones G. & Ruiz M. (2013). *Desarrollo de un prototipo electrónico basado en vision Artificial para la estimacion de la densidad de usuarios en Playas turísticas* (Tesis de Grado)(p. 56-57). Universidad del Magdalena, Santa Marta, Colombia.

Esta técnica de procesamiento de imágenes idealmente genera una imagen binaria, donde los pixeles que pertenecen al objeto se representan con un 1, mientras que los que no pertenecen al mismo se representan con un 0. Este tipo de particionamiento está basado en el análisis de alguna característica de la imagen, tal como los niveles de gris o la textura. En la actualidad existen diferente método de segmentación dependiendo del tipo de aplicación, tales como lo muestra la figura 3. <sup>7</sup>



**Figura 3. Métodos de Segmentación <sup>7</sup>**

#### 5.1.2.2.2. Selección Óptima del Umbral

Hay diferentes estrategias para la elección del umbral óptimo; entre ellas destacan aquellas cuyos fundamentos se basan en:

- Ajuste gaussiano.
- Minimización de la varianza interclase.

<sup>7</sup> Rincones G. & Ruiz M. (2013). *Desarrollo de un prototipo electrónico basado en vision Artificial para la estimacion de la densidad de usuarios en Playas turísticas* (Tesis de Grado)(p. 57). Universidad del Magdalena, Santa Marta, Colombia.

- Entropía del histograma.
- Análisis de la concavidad.
- Métodos basados en momentos.

Un resumen detallado de los distintos algoritmos se puede encontrar en revistas especializadas.<sup>8</sup>

#### 5.1.2.2.3. Operaciones Morfológicas

Las operaciones morfológicas son métodos para procesar imágenes binarias basadas en formas. Estas operaciones toman una imagen binaria como entrada y dan como resultado una imagen binaria como salida.<sup>9</sup>

El valor de cada pixel en la imagen de salida está basado sobre el correspondiente íxel de entrada y sus vecinos. Dentro de las operaciones morfológicas tenemos la dilatación y erosión, las cuales serán tratadas a continuación.<sup>9</sup>

##### 5.1.2.2.3.1. Dilatación

La operación de dilatar una imagen se puede describir como un crecimiento de píxeles, en otras palabras, se marca con 1 parte del fondo de la imagen que toque un pixel que forma parte de la región y así incrementarlo en dimensión y rellenar hoyos dentro de la región.<sup>10</sup>

$$\delta_B(X) = X \oplus B = \{x \mid X \cap B_x \neq \emptyset\}$$

<sup>8,9,10</sup>. Rincones G. & Ruiz M. (2013). *Desarrollo de un prototipo electrónico basado en vision Artificial para la estimación de la densidad de usuarios en Playas turísticas* (Tesis de Grado)(p. 58-59-60). Universidad del Magdalena, Santa Marta, Colombia.

Por tanto, el proceso de dilatación consiste en obtener la reflexión de B sobre su origen, después, cambiar esta reflexión por x. La dilatación de A por B es entonces, el conjunto de todos los desplazamientos x, tales que B y A se solapen en al menos un elemento distinto de cero.<sup>10</sup>

El conjunto B, se le conoce normalmente como el elemento de estructura de la dilatación. Como vemos, se toma el elemento de estructura B como una máscara de convolución. Aunque la dilatación se basa en operaciones de conjunto, mientras que la convolución se basa en operaciones aritméticas, el proceso básico de mover a B respecto a su origen, y desplazarlo después sucesivamente de tal forma que se deslice sobre el conjunto (imagen) A, es análogo al proceso de convolución expuesto anteriormente.<sup>10</sup>

Los componentes del conjunto B (elemento estructurador), pueden ser ceros o unos. La cantidad de ceros y/o unos determina, en conjunto con el tamaño del elemento estructurador, el efecto que produce su utilización en las operaciones morfológicas.<sup>10</sup>

#### 5.1.2.2.3.2. Erosión

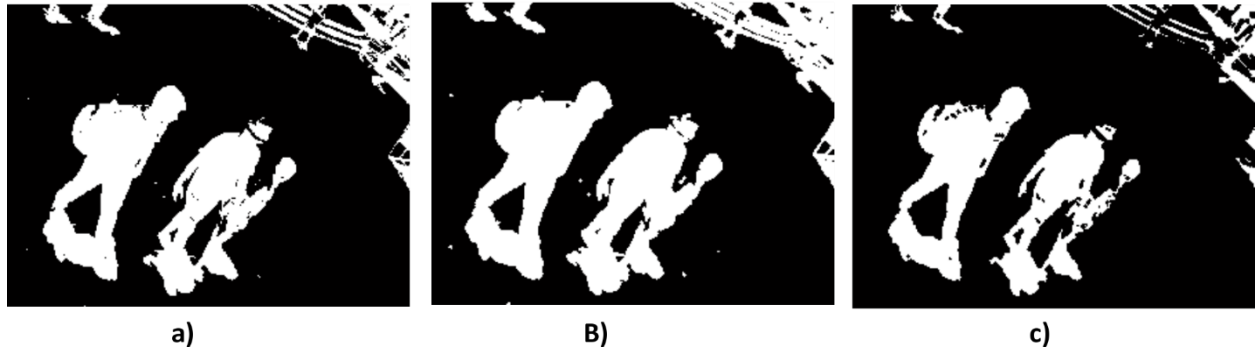
Se conoce como cerrado, en otras palabras, el proceso de cerrar morfológicamente una imagen consiste en marcar con un 1, los pixeles aislado que se encuentra entre ceros. Es decir, suaviza secciones del contorno, pero generalmente une separaciones estrechas y golfo delgado, elimina pequeños hoyos y rellena aberturas<sup>11</sup>

$$\varepsilon_B(X) = X \ominus B = \{x \mid B_x \subseteq X\}$$

La salida de la erosión es el conjunto de puntos barridos por el centro del elemento estructurante mientras se cumpla que todos los puntos de B están contenido en X. Es

<sup>10, 11</sup>. Rincones G. & Ruiz M. (2013). *Desarrollo de un prototipo electrónico basado en vision Artificial para la estimación de la densidad de usuarios en Playas turísticas* (Tesis de Grado)(p. 59-60). Universidad del Magdalena, Santa Marta, Colombia.

una operación anti extensiva, es decir la imagen contiene al resultado. Un ejemplo de dilatación y erosión se muestra en la figura 4.<sup>11</sup>



**Figura 4. a) Imagen binaria original; b) Imagen dilatada con un EE en forma de disco y de radio 2;**

**c) Imagen erosionada con un EE en forma de disco y de radio 2<sup>11</sup>**

#### 5.1.2.2.4. Filtro de Tamaño

El filtrado de tamaño en imágenes binarias está orientado, principalmente en dos objetivos:

- Eliminación de ruido: debido al proceso de umbralización pueden aparecer regiones en la imagen binaria que son consecuencia del ruido. Tales regiones son normalmente pequeñas.<sup>12</sup>

- Reducir el número de objetos de la imagen (hay objetos de un determinado tamaño que no interesan para la aplicación). En algunas ocasiones resulta interesante eliminar de una imagen binaria aquellos objetos que no superen un tamaño determinado, o por el contrario, eliminar los objetos que si superen este tamaño. Los objetos son filtrados<sup>12</sup>

teniendo en cuenta su tamaño o área o utilizando las operaciones de erosión y dilatación como se analizó anteriormente. Sin embargo, en muchas aplicaciones es conocido que

<sup>11, 12.</sup> Rincones G. & Ruiz M. (2013). *Desarrollo de un prototipo electrónico basado en vision Artificial para la estimacion de la densidad de usuarios en Playas turísticas* (Tesis de Grado)(p. 60-61). Universidad del Magdalena, Santa Marta, Colombia.



los objetos de interés son de un tamaño mayor que T píxeles (área). Todas las componentes de la imagen que tienen un tamaño igual o menor que T píxeles se eliminan; para ello se cambian los correspondientes píxeles al valor del fondo (0).

Tal y como se muestra en figura 5.<sup>12</sup>



**Figura 5. a) Imagen binaria dilatada; b) imagen filtrada por área  $\leq 100$  píxeles<sup>12</sup>**

#### 5.1.2.2.5. Eliminación del Borde

En esta parte se elimina el borde de la imagen, en otras palabras se borra cualquier objeto que se encuentre alrededor del borde de la imagen, ya que dichos objetos no tendrán la información completa, por lo que es mejor colocar en dicha región un (0) color del fondo, para que no se tenga en cuenta al momento de etiquetar la imagen. Tal como se muestra a continuación en la figura 6:<sup>13</sup>

<sup>12,13</sup>. Rincones G. & Ruiz M. (2013). *Desarrollo de un prototipo electrónico basado en visión Artificial para la estimación de la densidad de usuarios en Playas turísticas* (Tesis de Grado)(p. 61-62). Universidad del Magdalena, Santa Marta, Colombia.



**Figura 6. a) Imagen resultante etapa anterior; b) Imagen sin borde <sup>13</sup>**

#### 5.1.2.2.6. Etiquetado

En muchas aplicaciones se pretende diferenciar los objetos que aparecen en la imagen y/o contarlos, por lo que el proceso de segmentación se prolonga aplicando diversas técnicas sobre este tipo de imágenes. Existe una gran cantidad de técnicas de análisis para imágenes binarias, con propósitos tan variados, entre estas técnicas está el contar, etiquetar objetos y filtrado de objetos según su tamaño. <sup>14</sup>

Se ha de disponer de una definición consistente de conectividad para demarcar todos los objetos en imágenes binarias y ser capaces de idear algoritmos para etiquetarlos y contarlos.<sup>14</sup>

El etiquetado es una técnica que, partiendo de una imagen binaria, nos permite etiquetar cada uno de los objetos conectados presentes en la imagen. Esto va a posibilitar:<sup>14</sup>

- Distinguir los objetos respecto del fondo (propiedad intrínseca de la imagen binaria).<sup>14</sup>

<sup>13,14</sup>. Rincones G. & Ruiz M. (2013). *Desarrollo de un prototipo electrónico basado en vision Artificial para la estimacion de la densidad de usuarios en Playas turísticas* (Tesis de Grado)(p. 62-63). Universidad del Magdalena, Santa Marta, Colombia.

- Distinguir un objeto respecto de los demás objetos.
- Conocer el número de objetos en la imagen.

En general, los algoritmos para etiquetar dan buenos resultados con objetos convexos, pero presentan problemas cuando aparecen objetos que tienen concavidades (formas en U, donde diferentes partes de un mismo objeto pueden acabar con etiquetas distintas, incluso pueden aparecer colisiones de etiquetas.<sup>14</sup>

Sea utilizado rectángulos de color rojo para encerrar a los objetos que aparece en la imagen. Para distinguir si un objeto es una persona o no, se basa en el tamaño del área de cada objeto, y aquel objeto que supere un área determinada se contara como persona, el área del objeto de una imagen depende directamente de la posición de la cámara y el ángulo en que ésta esté, por lo tanto es válido decir que para cada posición de la cámara el algoritmo cambia en la parte de clasificación del objeto como persona.<sup>14</sup>

## **5.2. Visión Estereoscópica**

De manera natural nuestro mecanismo de visión es estéreo, es decir, somos capaces de apreciar, a través de la visión binocular, las diferentes distancias y volúmenes en el entorno que nos rodea. Nuestros ojos, debido a su separación, obtienen dos imágenes con pequeñas diferencias entre ellas, a lo que denominamos disparidad. Nuestro cerebro procesa las diferencias entre ambas imágenes y las interpreta de forma que percibimos la sensación de profundidad, lejanía o cercanía de los objetos que nos rodean. Este proceso se denomina estereopsis. En la estereopsis intervienen diversos mecanismos. Cuando observamos objetos muy lejanos, los ejes ópticos de nuestros ojos son paralelos. Cuando observamos un objeto cercano, nuestros ojos giran para que los ejes ópticos estén alineados sobre él, es decir, convergen. A su vez se produce la

<sup>14</sup>. Rincones G. & Ruiz M. (2013). *Desarrollo de un prototipo electrónico basado en vision Artificial para la estimacion de la densidad de usuarios en Playas turísticas* (Tesis de Grado)(p. 63). Universidad del Magdalena, Santa Marta, Colombia.

acomodación o enfoque para ver nítidamente el objeto. Este proceso conjunto se llama fusión<sup>15</sup>.

Un factor que interviene directamente en esta capacidad es la separación intraocular. A mayor separación entre los ojos, mayor es la distancia a la que apreciamos el efecto de relieve. Esto se aplica por ejemplo en los prismáticos, en los que, mediante prismas, se consigue una separación intraocular efectiva mayor que la normal, con lo que se consigue apreciar en relieve objetos distantes que en condiciones normales no seríamos capaces de separar del entorno.

Ésta técnica es aplicada en la fotografía aérea, en la que se obtienen pares estereoscópicos con separaciones de cientos de metros y en los que es posible apreciar claramente el relieve del terreno, lo que con la visión normal y desde gran altura sería imposible.

Una de las aplicaciones prácticas más antigua es la visualización y medición del relieve terrestre mediante fotografías aéreas. Si un avión toma dos fotografías de una zona de terreno con una cierta distancia calculada entre ellas, se obtiene un estéreo-par, que posteriormente puede verse en relieve con un estereoscopio especial. Si las tomas se realizan con la adecuada precisión, permiten calcular elevaciones en el terreno, para lo cual se emplean los estéreos-comparadores. A partir de datos del terreno pueden también generarse imágenes 3D simuladas mediante software específico<sup>16</sup>.

Considerando que la estereovisión biológica depende directamente de la posición relativa a la convergencia de los ojos, el desplazamiento horizontal de las imágenes permite analizar la disparidad de cada uno de los píxeles.

De la amplia información de entrada se pretende centrar la atención en aquellos objetos que sean de interés de acuerdo a unos determinados criterios, y conseguir mantener dicha atención a lo largo del tiempo, resolviendo la tarea de la visión

<sup>15</sup><http://sabia.tic.udc.es/gc/trabajos%202011-12/VisionEstereoscopica/principal.html>

<sup>16</sup><https://sites.google.com/site/grafcomputacional/estereoscopia/topografia>

selectiva. La visión artificial y la robótica, al unirse en distintos tipos de aplicaciones, han dado lugar a los sistemas de Robótica Guiada por Visión, que proporcionan un mayor grado de libertad a la robótica al poder trabajar en entornos de mayor versatilidad. Entre las principales ventajas inherentes a esta tecnología, destaca “la identificación de características estructuradas de alta complejidad. Con otras tecnologías, este tipo de piezas difícilmente pueden ser reconocidas, mientras que la utilización la visión artificial estéreo permite hacer esta identificación de forma más rápida y eficaz.

### **5.3. MatLab**

MATLAB es la abreviatura de *MATrix LABoratory*, "laboratorio de matrices") es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X. Entre sus prestaciones básicas se hallan: <sup>17</sup>

- Manipulación de matrices
- Representación de datos y funciones
- Implementación de algoritmos
- Creación de interfaces de usuario (GUI)
- Comunicación con programas en otros lenguajes y con otros dispositivos

El hardware Matlab es un entorno de computación y desarrollo de aplicaciones totalmente integrado orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos.<sup>17</sup>

Matlab integra análisis numérico, cálculo matricial, proceso de señal y visualización gráfica en un entorno completo donde los problemas y sus soluciones son expresados

<sup>17</sup>. Rincones G. & Ruiz M. (2013). *Desarrollo de un prototipo electrónico basado en vision Artificial para la estimacion de la densidad de usuarios en Playas turísticas* (Tesis de Grado)(p. 39). Universidad del Magdalena, Santa Marta, Colombia.

del mismo modo en que se escribirían racionalmente, sin necesidad de hacer uso de la programación tradicional. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI).<sup>17</sup>

### 5.3.1. Interfaz Gráfica GUI

Las GUI (también conocidas como interfaces gráficas de usuario o interfaces de usuario) permiten un control sencillo (con uso de ratón) de las aplicaciones de software, lo cual elimina la necesidad de aprender un lenguaje y escribir comandos a fin de ejecutar una aplicación.<sup>18</sup>

Las apps de MatLab son programas autónomos de MatLab con un frontal gráfico de usuario GUI que automatizan una tarea o un cálculo. Por lo general, la GUI incluye controles tales como menús, barras de herramientas, botones y controles deslizantes. También es posible crear apps personalizadas propias, incluidas las interfaces de usuario correspondientes, para que otras personas las utilicen.<sup>18</sup>

#### 5.3.1.1. Creación de una GUI de MatLab de forma interactiva

GUIDE (entorno de desarrollo de GUI) proporciona herramientas para diseñar interfaces de usuario para Apps personalizadas. Mediante el editor de diseño de GUIDE, es posible diseñar gráficamente la interfaz de usuario. GUIDE genera entonces de manera automática el código de MatLab para construir la interfaz, el cual se puede modificar para programar el comportamiento de la app.<sup>19</sup>

<sup>17</sup>. Rincones G. & Ruiz M. (2013). *Desarrollo de un prototipo electrónico basado en vision Artificial para la estimacion de la densidad de usuarios en Playas turísticas* (Tesis de Grado)(p. 39). Universidad del Magdalena, Santa Marta, Colombia.

<sup>18,19</sup> <https://es.mathworks.com/discovery/matlab-gui.html>

#### 5.3.1.2. Creación de una GUI de MatLab de forma programática

A fin de ejercer un mayor control sobre el diseño y el desarrollo, también se puede crear código de MatLab que defina las propiedades y los comportamientos de todos los componentes. MatLab contiene funcionalidad integrada que le ayudará a crear la GUI para su App de forma programática. Cabe la posibilidad de agregar cuadros de diálogo, controles de interfaz de usuario (como botones y controles deslizantes) y contenedores (como paneles y grupos de botones).<sup>19</sup>

### 5.4. Tarjeta de control “hardware”

#### 5.4.1. Arduino

Arduino (Genuino a nivel internacional hasta octubre 2016), es una compañía de hardware libre y una comunidad tecnológica que diseña y manufactura placas computadora de desarrollo de hardware y software, compuesta respectivamente por circuitos impresos que integran un microcontrolador y un entorno de desarrollo (IDE), en donde se programa cada placa.<sup>20</sup>

Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Toda la plataforma, tanto para sus componentes de hardware como de software, son liberados con licencia de código abierto que permite libertad de acceso a ellos.<sup>20</sup>

El hardware consiste en una placa de circuito impreso con un microcontrolador, usualmente Atmel AVR, puertos digitales y analógicos de entrada/salida, los cuales

<sup>19</sup> <https://es.mathworks.com/discovery/matlab-gui.html>

<sup>20</sup> <https://es.wikipedia.org/wiki/Arduino>

pueden conectarse a placas de expansión (shields), que amplían las características de funcionamiento de la placa Arduino. Asimismo, posee un puerto de conexión USB desde donde se puede alimentar la placa y establecer comunicación con el computador.<sup>20</sup>

Por otro lado, el software consiste en un entorno de desarrollo (IDE) basado en el entorno de Processing y lenguaje de programación basado en Wiring, así como en el cargador de arranque (bootloader) que es ejecutado en la placa.<sup>4</sup> El microcontrolador de la placa se programa mediante un computador, usando una comunicación serial mediante un convertidor de niveles RS-232 a TTL serial.<sup>20</sup>

La primera placa Arduino fue introducida en 2005, ofreciendo un bajo costo y facilidad de uso para novatos y profesionales. Buscaba desarrollar proyectos interactivos con su entorno mediante el uso de actuadores y sensores. A partir de octubre de 2012, se incorporaron nuevos modelos de placas de desarrollo que usan microcontroladores Cortex M3, ARM de 32 bits, que coexisten con los originales modelos que integran microcontroladores AVR de 8 bits. ARM y AVR no son plataformas compatibles en cuanto a su arquitectura y por lo que tampoco lo es su set de instrucciones, pero se pueden programar y compilar bajo el IDE predeterminado de Arduino sin ningún cambio.<sup>20</sup>

Las placas Arduino están disponibles de dos formas: ensambladas o en forma de kits "Hazlo tú mismo" (por sus siglas en inglés "DIY"). Los esquemas de diseño del Hardware están disponibles bajo licencia Libre, con lo que se permite que cualquier persona pueda crear su propia placa Arduino sin necesidad de comprar una prefabricada. Adafruit Industries estimó a mediados del año 2011 que, alrededor de 300,000 placas Arduino habían sido producidas comercialmente y en el año 2013 estimó que alrededor de 700.000 placas oficiales de la empresa Arduino estaban en manos de los usuarios.<sup>20</sup>

<sup>20</sup> <https://es.wikipedia.org/wiki/Arduino>



Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash, Processing, Max/MSP, Pure Data, etc. Una tendencia tecnológica es utilizar Arduino como tarjeta de adquisición de datos desarrollando interfaces en software como JAVA, Visual Basic y LabVIEW 6. Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.<sup>20</sup>

#### 5.4.1.1. Tipos de Arduino

Arduino cuenta con varios prototipos y modelos de placas a elegir, dependiendo de qué tan grande es uso que le demos a esta. Cada una con características particulares y diferentes que las diferencian de las demás. Aquí se mencionan algunas de las más populares junto con sus características:<sup>21</sup>

##### 5.4.1.1.1. Duemilanove

El Arduino Duemilanove ("2009") es una placa con microcontrolador basada en el ATmega168 (datasheet) o el ATmega328 (datasheet). Tiene 14 pines con entradas/salidas digitales (6 de las cuales pueden ser usadas como salidas PWM), 6 entradas analógicas, un cristal oscilador a 16Mhz, conexión USB, entrada de alimentación, una cabecera ISCP, y un botón de reset. Contiene todo lo necesario para utilizar el microcontrolador; simplemente conéctalo a tu ordenador a través del cable USB o aliméntalo con un transformador o una batería para empezar a trabajar con él. <sup>21</sup>

El Arduino Duemilanove puede ser alimentado vía la conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente.

<sup>20</sup> <https://es.wikipedia.org/wiki/Arduino>

<sup>21</sup> <http://arduinodehtics.weebly.com/tipos-de-arduino.html>

Las fuentes de alimentación externas (no-USB) pueden ser tanto un transformador o una batería. La placa puede trabajar con una alimentación externa de entre 6 a 20 voltios. Si el voltaje suministrado es inferior a 7V el pin de 5V puede proporcionar menos de 5 Voltios y la placa puede volverse inestable, si se usan más de 12V los reguladores de voltaje se pueden sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.<sup>21</sup>

### *Memoria*

El ATmega328 tiene 32KB (el ATmega168 tiene 16 KB) de memoria flash para almacenar código (2KB son usados para el arranque del sistema (bootloader). El ATmega328 tiene 2 KB (Atmega168 1 KB) de memoria SRAM. El ATmega328 tiene 1KB (ATmega168 512 bytes) de EEPROM, que puede a la cual se puede acceder para leer o escribir con la [Reference/EEPROM |librería EEPROM]].<sup>21</sup>

### *Comunicación*

EL Arduino Duemilanove facilita en varios aspectos la comunicación con el ordenador, otro Arduino u otros microcontroladores. Tanto el ATmega328 como el Atmega168 proporciona comunicación vía serie UART TTL (5V), disponible a través de los pines digitales 0(RX) y 1(TX). Un chip FTDI FT232RL integrado en la placa canaliza esta comunicación serie a través del USB y los drivers FTDI (incluidos en el software de Arduino) proporcionan un puerto serie virtual en el ordenador. El software incluye un monitor de puerto serie que permite enviar y recibir información textual de la placa Arduino. Los LEDs RX y TX de la placa parpadearan cuando se detecte comunicación transmitida través del chip FTDI y la conexión USB (no parpadearan si se usa la comunicación serie a través de los pines 0 y 1).<sup>21</sup>

<sup>21</sup> <http://arduinothics.weebly.com/tipos-de-arduino.html>

## *Programación*

El ATmega328 y el ATmega168 en las placas Arduino Duemilanove viene precargado con un gestor de arranque (bootloader) que permite cargar nuevo código sin necesidad de un programador por hardware externo. Se comunica utilizando el protocolo STK500 original (archivo de cabecera C).<sup>21</sup>

### 5.4.1.1.2. Mega

El Arduino Mega es una placa microcontrolador basada ATmeg1280 (datasheet). Tiene 54 entradas/salidas digitales (de las cuales 14 proporcionan salida PWM), 16 entradas digitales, 4 UARTS (puertos serie por hardware), un cristal oscilador de 16MHz, conexión USB, entrada de corriente, conector ICSP y botón de reset. Contiene todo lo necesario para hacer funcionar el microcontrolador; simplemente conéctalo al ordenador con el cable USB o aliméntalo con un transformador o batería para empezar.<sup>21</sup>

## *Alimentación*

El Arduino Mega puede ser alimentado vía la conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente.<sup>21</sup>

La placa puede trabajar con una alimentación externa de entre 6 a 20 voltios. Si el voltaje suministrado es inferior a 7V el pin de 5V puede proporcionar menos de 5 Voltios y la placa puede volverse inestable, si se usan más de 12V los reguladores de voltaje se pueden sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.<sup>21</sup>

<sup>21</sup> <http://arduinothics.weebly.com/tipos-de-arduino.html>

## *Memoria*

El ATmega1280 tiene 128KB de memoria flash para almacenar código (4KB son usados para el arranque del sistema (bootloader)). El ATmega1280 tiene 8 KB de memoria SRAM. El ATmega1280 tiene 4KB de EEPROM, que puede a la cual se puede acceder para leer o escribir con la [Reference/EEPROM librería EEPROM].<sup>21</sup>

## *Programación*

El ATmega1280 en el Arduino Mega viene precargado con un gestor de arranque (bootloader) que permite cargar nuevo código sin necesidad de un programador por hardware externo. Se comunica utilizando el protocolo STK500 original (archivo de cabecera C).<sup>21</sup>

También te puedes saltar el gestor de arranque y programar directamente el microcontrolador a través del puerto ISCP (In Circuit Serial Programming).<sup>21</sup>

## *Características físicas*

La longitud y amplitud máxima de la placa Duemilanove es de 4 y 2.1 pulgadas respectivamente, con el conector USB y la conexión de alimentación sobresaliendo de estas dimensiones. Tres agujeros para fijación con tornillos permiten colocar la placa en superficies y cajas. Ten en cuenta que la distancia entre los pines digitales 7 y 8 es 160 mil (0,16"), no es múltiple de la separación de 100 mil entre los otros pines. <sup>21</sup>

<sup>21</sup> <http://arduinothics.weebly.com/tipos-de-arduino.html>

#### 5.4.1.1.3. Nano

El Arduino Nano es una pequeña y completa placa basada en el ATmega328 (Arduino Nano 3.0) o ATmega168 (Arduino Nano 2.x) que se usa conectándola a una protoboard.

Tiene más o menos la misma funcionalidad que el Arduino Duemilanove, pero con una presentación diferente. No posee conector para alimentación externa, y funciona con un cable USB Mini-B en vez del cable estándar. El nano fue diseñado y está siendo producido por Gravitech.<sup>21</sup>

##### *Alimentación*

El Arduino Nano puede ser alimentado usando el cable USB Mini-B, con una fuente externa no regulada de 6-20V (pin 30), o con una fuente externa regulada de 5V (pin 27). La fuente de alimentación es seleccionada automáticamente a aquella con mayor tensión.<sup>21</sup>

El chip FTDI FT232RL que posee el Nano solo es alimentado si la placa está siendo alimentada usando el cable USB. Como resultado, cuando se utiliza una fuente externa (no USB), la salida de 3.3V (la cual es proporcionada por el chip FTDI) no está disponible y los pines 1 y 0 parpadearán si los pines digitales 0 o 1 están a nivel alto.<sup>21</sup>

##### *Memoria*

El ATmega168 posee 16KB de memoria flash para almacenar el código (de los cuales 2KB son usados por el bootloader); el ATmega 328 posee 32KB, (también con 2 KB usados por el bootloader). El ATmega168 posee 1KB de SRAM y 512 bytes de EEPROM (la cual puede ser leída y escrita con la librería EEPROM); el ATmega328 posee 2 KB de SRAM y 1KB de EEPROM.<sup>21</sup>

<sup>21</sup> <http://arduinothics.weebly.com/tipos-de-arduino.html>

## *Comunicación*

El Arduino Nao tiene algunos métodos para la comunicación con un PC, otro Arduino, u otros microcontroladores. El ATmega168 y el ATmega328 poseen un módulo UART que funciona con TTL (5V) el cual permite una comunicación vía serie, la cual está disponible usando los pines 0 (RX) y 1 (TX). El chip FTDI FT232RL en la placa hace de puente a través de USB para la comunicación serial y los controladores FTDI (incluidos con el software de Arduino) provee al PC de un puerto COM virtual para el software en el PC.<sup>21</sup>

## *Programación*

El ATmega168 o ATmega328 del Arduino Nano vienen pre programados con un bootloader que te permite subir tu código al Arduino sin la necesidad de un programador externo. Se comunica usando el protocolo STK500 original (Archivos cabecera C).<sup>21</sup>

### 5.4.1.1.4. Pro

La Arduino pro es una placa con un microcontrolador ATmega168 (datasheet) o en el ATmega328 (datasheet). La Pro viene en versiones de 3.3v / 8 MHz y 5v / 16 MHz. Tiene 14 E/S digitales (6 de las cuales se puedes utilizar como salidas PWM), 6 entradas analógicas, un resonador interno, botón de reseteo y agujeros para el montaje de tiras de pines. Vienen equipada con 6 pines para la conexión a un cable FTDI o a una placa adaptadora de la casa Sparkfun para dotarla de comunicación USB y alimentación.<sup>21</sup>

La Arduino Mini Pro está destinada a instalaciones semi-permanentes en objetos o demostraciones. La placa viene sin conectores montados, permitiendo el uso de varios tipos de conectores o soldado directo de cables según las necesidades de cada proyecto en particular. La distribución de los pines es compatible con los shields de Arduino. Las versiones de 3.3v de la pro pueden ser alimentadas por baterías. <sup>21</sup>

<sup>21</sup> <http://arduinothics.weebly.com/tipos-de-arduino.html>

### *Alimentación*

La Arduino Pro puede ser alimentada por medio del cable USB, por baterías o mediante una fuente de alimentación. El conector de batería es del tipo JST, también se le puede soldar otro tipo de conector para alimentarla desde una fuente de alimentación externa.<sup>21</sup>

### *Memoria*

El ATmega168 tiene 16KB de memoria para el almacenamiento de sketches (de los cuales 2KB están reservados para el gestor de arranque). También tiene 1KB de SRAM y 512 bytes de EEPROM en los cuales se puede leer y escribir mediante la librería EEPROM. EL ATmega328 tiene 32 KB de flash, 2 KB de SRAM, y 1 KB de EEPROM. <sup>21</sup>

### *Características físicas*

La longitud y anchura máxima del PCB de la Pro son 5.2cm y 5.3cm respectivamente, con el conector de 6 pines y el selector de alimentación sobresaliendo ligeramente de los bordes. 4 perforaciones para tornillos permiten la fijación de la placa sobre una superficie o una caja. La distancia entre los pines 7 y 8 es de 4mm, no como los demás pines, que están separados por 2.5mm (separación normalizada entre pines).<sup>21</sup>

<sup>21</sup> <http://arduinothics.weebly.com/tipos-de-arduino.html>

## 6. ESTADO DEL ARTE

Recientemente ha venido aumentando el interés en el estudio de las ciencias de lo artificial, promovidas, entre otras cosas, por el asombroso avance de las tecnologías en la informática y la comunicación. En específico, a partir de la segunda mitad del siglo pasado una de las cuestiones que ha sido el principal foco de atención es la construcción de robots autónomos capaces de una interacción inteligente con los cambios en su entorno. En la última década, los sistemas de visión por computadora se han consolidado como una pieza fundamental en los sistemas de producción de las industrias. Para lograr entender mejor estos conceptos, es necesario resolver el problema de la percepción artificial y toma de decisiones frente a características determinadas. Este es precisamente el campo de investigación de este trabajo de grado. Históricamente, el problema se ha estudiado desde muchos puntos de vista.

Al colocar la vista internacionalmente y enfocada a nivel empresarial e industrial, se encuentra la empresa **Mathworks** la cual es pionera por su software de desarrollo conocido como MatLab, él cuenta con herramientas de gran utilidad las cuales ayudan a universidades y demás compañías a la investigación de temáticas referente a la visión artificial; además de otras áreas distintas de esta.

A nivel internacional y de carácter institucional se encuentra el “*Grupo de Neurocomputación Biológica (GNB) Dpto. de Ingeniería Informática Escuela Politécnica Superior Universidad Autónoma de Madrid SEPTIEMBRE 2011 con el proyecto de fin de carrera titulado Detección Y Seguimiento De Objetos Con Cámaras En Movimiento, desarrollado por AUTOR Héctor López Paredes y TUTOR: Manuel Sánchez Montañés*”.

En el ámbito nacional se encuentra el caso de la “*Universidad Tecnológica, Pereira, Colombia, del Departamento de Física, por medio de un artículo científico desarrollado por los señores Rodrigo Andrés Franco Luna y José Andrés Chaves Osorio en el mes de abril del año 2011, el cual lleva como título Desarrollo de un modelo para el seguimiento de patrones mediante reconocimiento óptico*”.



A nivel local, se destaca un trabajo de grado el cual utiliza visión artificial que tiene como título: Desarrollo de un aplicativo de software capaz de procesar imágenes para identificar patrones de irregularidad, y color en el melanoma cutáneo, desarrollado por el estudiante Miguel Zuleta en el año 2015 el cual consiste en procesar imágenes digitales del melanoma cutáneo, con el motivo de identificar sus características discriminantes y determinar malignidad o no como ayuda al diagnóstico parcial de esta lesión.

## **7. OBJETIVOS**

### **7.1. OBJETIVO GENERAL**

Desarrollar un Sistema De Identificación y seguimiento de objetos de colores RGB empleando visión artificial en MatLab, para una plataforma robótica rotatoria de tal manera que se pueda realizar un seguimiento en tiempo real para evitar perderle de vista el objeto.

### **7.2. OBJETIVOS ESPECÍFICOS**

- Escoger un método de diseño que permita al dispositivo el funcionamiento más cercano al que se espera llegar al final del proyecto.
- Emplear la metodología ya acordada para la conceptualización, verificación y documentación del dispositivo final a obtener en el proyecto
- Plantear un esquema de bloques que permita separar las distintas funcionalidades que maneja el dispositivo mecánico
- Construir en el software de desarrollo y programación cada bloque que hará parte de la máquina
- Integrar todos los bloques o etapas del software que hayan sido programados para la realización de pruebas de funcionamiento y corrección de errores
- Analizar cuál es la mejor tarjeta de desarrollo y materiales de movimiento mecánico del sistema que se adapten a los alcances del proyecto para su posterior compra

- Realizar la comunicación o cargar el programa previamente probado a la tarjeta de desarrollo seleccionada, para poder unificar con la estructura que ya se tiene clara va a utilizarse en el montaje del sistema mecánico.
- Probar el funcionamiento y realización de ajustes dejando en claro los posibles usos y mejoras que puede tener el proyecto
- Realizar pruebas que permitan medir la distancia mínima y máxima de detección y seguimiento de objetos de colores R, G, B.
- Clasificar las distintas gamas de colores rojo, verde y azul que puede detectar y seguir el prototipo.

## **8. DISEÑO METODOLÓGICO**

La metodología empleada se encuentra diseñada en el esquema por fases.

### **Fase I: Obtención de información, para el desarrollo del proyecto:**

En el esquema de trabajo a desarrollar se seguirán a manera de objetivos y como tareas principales, una serie de actividades que ayudaran de manera secuencial a llevar a cabo el proyecto y obteniendo de la forma más exacta llegar a los resultados esperados. Por eso es que en esta fase se comenzara por la recolección de toda la información posible, pertinente y necesaria para conseguir o anteriormente especificado. Para cual dividimos la fase en tres tareas:

1. Investigación de antecedente y proyectos similares desarrollados en el mundo, nuestro país, la región y por último la Universidad
2. Análisis de la información y depuración de todos los proyectos que se hayan encontrado y guarden similitudes al trabajo del cual este escrito tiene como objetivo, para así de estos mismos tomar las herramientas necesarias que permitan mejorar y optimizar y darle el enfoque necesario al trabajo respecto del objeto general.
3. Estudio de la viabilidad del proyecto, tarea que busca darle el enfoque correcto mediante el estudio de aplicabilidad de las técnicas y herramientas que no hayan sido desarrolladas por la literatura y llevarlas a desarrollo en la universidad.

## **Fase II: Diseño del modelo de funcionamiento del software a desarrollar y el sistema de motores:**

Conseguir que las entidades que se estén desarrollando cumplan con los alcances del proyecto requiere de estudios minuciosos, basados en investigaciones y cálculos que permitirán la escogencia de las herramientas correctas que ayudaran el proceso de creación del software y el dispositivo manejado mediante este:

1. Desarrollo de la parte del software en MatLab para detección de objetos de color ROJO
2. Desarrollo de la parte del software en MatLab para detección de objetos de color VERDE
3. Desarrollo de la parte del software en MatLab para detección de objetos de color AZUL

## **Fase III: Análisis de las variables críticas del proyecto y escogencia de los elementos óptimos de construcción.**

En esta etapa se busca un consenso después de obtenidas determinadas respuestas en pruebas de diseño, dejando ya en fase transitoria al montaje del proyecto:

1. Integración y prueba de las 3 partes del software
2. Escogencia y compra de la tarjeta de desarrollo que más se ajuste a las necesidades que se observen.
3. Compra y ensamble de estructura rotatoria con 2 grados de libertad

#### **Fase IV: Implementación y desarrollo de pruebas**

1. Programación en tarjeta de desarrollo para recepción de datos de MatLab y proceder a realizar seguimiento del objeto
2. Integración y prueba del software de MatLab y la tarjeta de desarrollo
3. Mejoras al proyecto

## 9. CRONOGRAMA

Actividad	Semanas															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Fase 1: Investigación de antecedente y proyectos similares																
Fase 1: Análisis de la información																
Fase 1: Estudio de la viabilidad del proyecto																
Fase2: Desarrollo de la parte del software en MatLab para detección de objetos de color ROJO																
Fase2: Desarrollo de la parte del software en MatLab para detección de objetos de color VERDE																
Fase 2: Desarrollo de la parte del software en MatLab para detección de objetos de color AZUL																
Fase 3: Integración y prueba de las 3 partes del software																
Fase 3: Compra y ensamble de estructura rotatoria con 2 grados de libertad																
Fase 4: Programación en tarjeta de desarrollo para recepción de datos de MatLab y proceder a realizar seguimiento del objeto																
Fase 4: Integración y prueba del software de MatLab y la tarjeta de desarrollo																
Fase4: Mejoras al proyecto																
Realización del documento final del proyecto																

## **10. IMPACTOS ESPERADOS**

Este proyecto va enfocado a los campos de investigación en un área necesaria a la hora de trabajar aplicaciones con procesamiento de señales visuales todo esto mediante el manejo de las estrategias tecnológicas desarrolladas en procesos de aprendizajes con agentes robóticos, buscando así como se ha dicho en anteriores ocasiones un desarrollo de prototipo robótico aplicado. Esta Generación de tecnologías propias asociadas con agentes robóticos ayudará a las distintas instituciones de investigación desarrollando nuevas tecnologías. Es bueno aclarar que para esto se tendrá muy en cuenta la implementación de las estrategias tecnológicas desarrolladas en procesos industriales con agentes robóticos dado que a partir de estos podríamos facilitar la comprensión de porque se hacen necesarios dispositivos de este tipo. Por último se desea dejar una evidencia escrita detallada de las estrategias tecnológicas desarrolladas en procesos de aprendizajes con agentes robóticos para que en su momento puedan ser estudiadas y sean tenidas en cuenta al momento de desarrollar las temáticas de enseñanza en las asignaturas pertinentes



## 11. DESARROLLO DEL PROYECTO

Para la elaboración del proyecto que tiene como título **Sistema De Identificación y seguimiento de objetos de colores RGB para una plataforma rotatoria mediante visión artificial empleando el software de MatLab**, se inicia dándole cumplimiento al cronograma de actividades descrito en el contenido de este documento, el cual inicia en la **Fase 1** que consiste en realizar una investigación de antecedentes y proyectos similares relacionados con la temática de interés que es la visión artificial. Seguido del análisis de la misma y dando como resultado la viabilidad del proyecto en cuestión.

Luego de realizar las revisiones detalladas de las investigaciones consultadas se encontraron tres proyectos cuyo nombre son:

- **DESARROLLO DE UN PROTOTIPO ELECTRÓNICO BASADO EN VISION ARTIFICIAL PARA LA ESTIMACION DE LA DENSIDAD DE USUARIOS EN PLAYAS TURÍSTICAS**, elaborada por los estudiantes GUSTAVO ADOLFO RINCONES PAVI y MARIO ELÍAS RUIZ MÁRQUEZ bajo la dirección del PHD. JOHN ALEXANDER TABORDA GIRALDO en el año del 2013.
- **DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE VISIÓN POR COMPUTADOR Y PROCESAMIENTO DE VIDEO PARA LA RECONSTRUCCIÓN DE TRAYECTORIAS CAÓTICAS DE PECES**, desarrollado por los estudiantes MAILE NIETO MUÑOZ y VÍCTOR ALTAFULLA PÉREZ bajo la dirección del PHD. JOHN ALEXANDER TABORDA GIRALDO en el año 2013.
- **PROTOTIPO PARA LA AUTOMATIZACIÓN DEL PROCESO DE CLASIFICACIÓN DEL MANGO TIPO EXPORTACIÓN (Mangifera Indica L) BASADO EN VISIÓN ARTIFICIAL**, desarrollado por los estudiantes ADOLFO MORENO BERMUDEZ y DEIMER BALLESTEROS PADILLA. Bajo la dirección del señor GERMAN SANCHEZ (director) y CLAUDIO DELRIEUX (codirector) en el año 2010.

Dichos proyectos fueron de gran importancia para la realización de esta investigación puesto que se tomaron como referencias de fortalecimiento para este trabajo, dado que la temática que manejan en común estas investigaciones es relacionada en el campo de la visión artificial o también llamada visión por computadora.

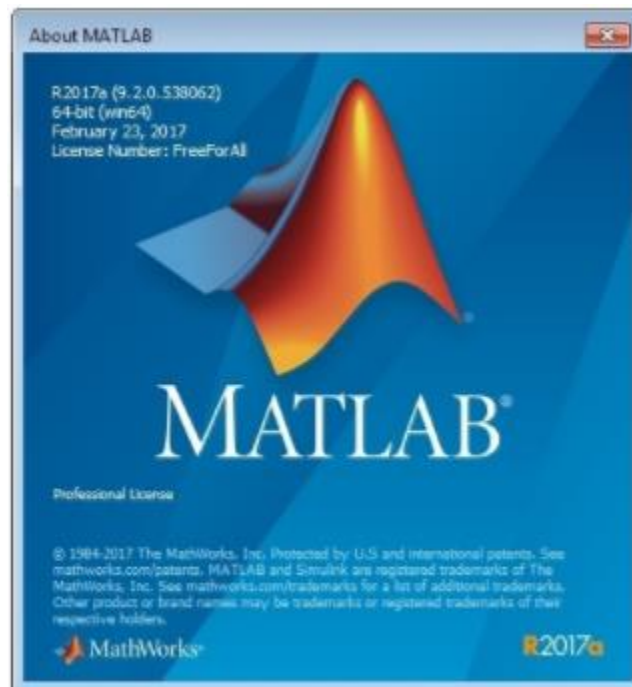
## **11.1. IMPLEMENTACIÓN DE SOFTWARE**

### **11.1.1. Software MatLab**

La implementación de algoritmos en visión por computador resulta de mucha inversión de tiempo ya que se requiere de la manipulación y conocimiento de softwares que permitan de una manera “relativamente fácil”, llevar a cabo el desarrollo de dicha investigación, cabe anotar que se debe tener en cuenta otros recursos como, el tipo de procesador que debe tener la computadora que permita soportar tanto los programas por defecto que trae consigo como el software con el cual se llevara a cabo la construcción de la programación, la cantidad de memoria, etc.

Hacerlo en cualquier otro tipo de lenguaje supondría la inversión de tiempo y sin la seguridad de que la implementación funcionará. Además de tener en cuenta el periodo de prueba, esto exige un tiempo normal de corrección de errores debidos al proceso de implementación del algoritmo. Todos estos problemas pueden ser resueltos si la implementación de prueba es realizada en MatLab, utilizando su toolbox de procesamiento de imágenes con ello el tiempo de implementación se convierte en el mínimo con la confianza de utilizar algoritmos científicamente probados y robustos. El toolbox de procesamiento de imágenes contiene un conjunto de funciones de los algoritmos más conocidos para trabajar con imágenes binarias, transformaciones geométricas, morfología y manipulación de color que junto con las funciones ya integradas en MatLab permite realizar análisis y transformaciones de imágenes a través de simples instrucciones y comandos.

Por su gran versatilidad se seleccionó MatLab por ser el software idóneo para aplicaciones de visión artificial y además por la facilidad al permitir crear fácilmente una interfaz de usuario (GUI) que para este proyecto es pieza fundamental, sumándole a esto que ya se pueden conectar hardware y software libres como es el caso de Arduino. La versión utilizada fue **MatLab 9.2.0.5380624** (R2017a), en la **Figura 7** observamos su logotipo.

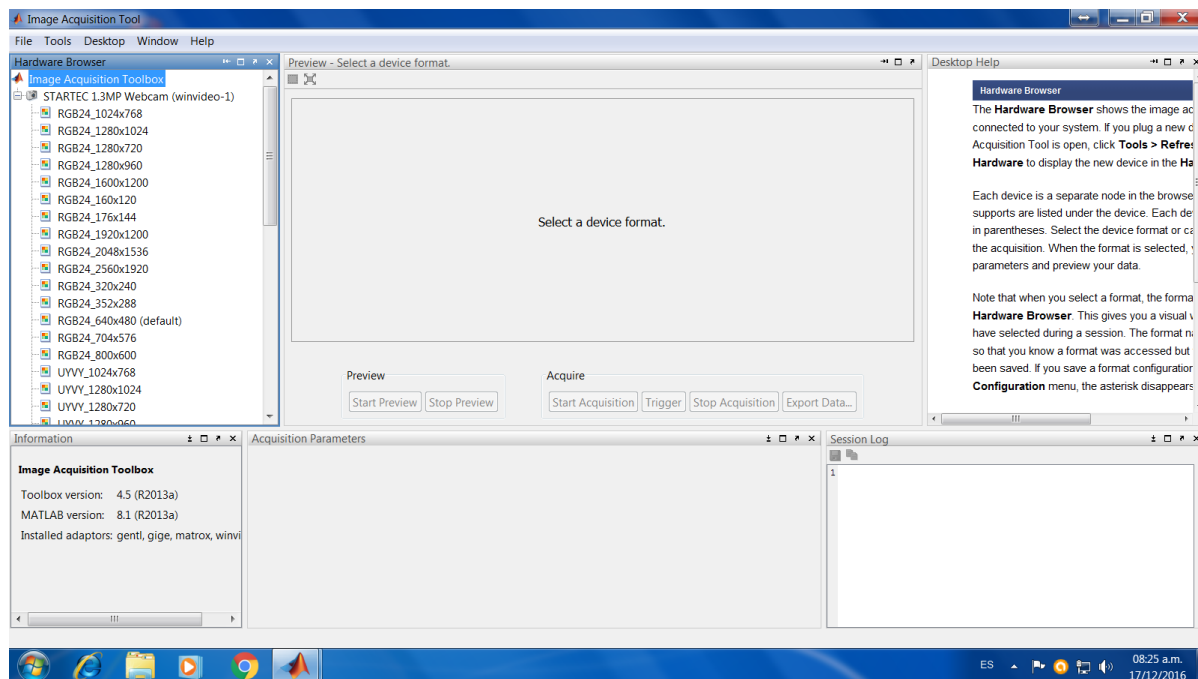


**Figura 7. Portada de Software MatLab**

De lo escrito anteriormente y de tener claro los antecedentes de los proyectos relacionados con la temática, se procede a darle forma y construir la **Fase 2**. El cual consiste en desarrollar por separado las partes del software que se encarguen en detectar objetos de color **Rojo** además de mostrar en pantalla las coordenadas del objeto y su respectivo seguimiento en tiempo real. Al cumplir con este objetivo, solo es cuestión de modificar los parámetros que permitan hacer lo mismo con el color **Verde** y posteriormente con el color **Azul**.

#### 11.1.1.1. Obtención del video

Para llevar a cabo esta etapa, se procede primero a instalarse el driver controlador de la cámara web, para que MatLab pueda reconocer que tipo de cámara está conectada además de mostrar información de la misma. Para verificar que todo fue instalado correctamente, se emplea el comando ***Imaqtool*** el cual desplegara una ventana donde se puede apreciar ciertos tipos de parámetros. Ver **Figura 8**



**Figura 8. Interfaz Única de Video**

El video viene configurada por defecto con una resolución de 640 x 480, sin embargo ésta resolución se puede variar como lo muestra en la figura 3, dependiendo de las propiedades técnicas de la cámara. Pero se deja por defecto para evitar mayor carga en el procesamiento, además de ser un tamaño adecuado para visualizar el funcionamiento.

#### 11.1.1.2. Pre-procesamiento

El objetivo de esta etapa es de mejorar las condiciones de las imágenes obtenidas como muestras, dichas muestras se extraen de la cadena de **frames que el sistema va almacenando**, todo esto se lleva a cabo con el fin de prepararla para su análisis en las etapas posteriores, las cuales inician con la captura de pantalla con el comando **snapshot** a uno de los tantos frames que está recibiendo MatLab por medio de la cámara, luego seguiría la conversión de la muestras originales a escala de grises, seguido de la binarización para poder calcular el área del objeto en cuestión; además de filtros que mejoren la calidad de los resultados.

##### 11.1.1.2.1. *Escala de grises*

Para llevar a cabo este proceso, Se toman una muestra o captura de pantalla de los frames por medio del comando snapshot, los cuales están siendo recibidos por MatLab de manera constante. Posteriormente, a esta captura de pantalla que en principio sería una imagen original ya que cuenta con todas sus características, se le aplica la siguiente instrucción **rgb2gray** para convertir la imagen original a escala de gris. Esto se hace porque una imagen en escalas de grises tiene una menor cantidad de datos lo cual favorece a la velocidad de procesamiento y facilita los algoritmos para la detección de detalles y propiedades de los objetos individuales en el video.

##### 11.1.1.2.2. *Segmentación*

Lo que se lleva a cabo en este proceso es la extracción del parámetro de color que en este caso sería el color seleccionado por el usuario (rojo, verde o azul), este paso se realiza al tener la imagen en escala de grises. El comando utilizado para la extracción del color de la imagen en escala de grises almacenada en la variable `data` es:

```
diff_im = imsubtract(data(:,:,1));
```

Donde:

**diff\_im:** Es la variable que almacenara el resultado al implementar el comando

**Imsubtract:** Comando utilizado para la extracción del parámetro

**Data:** Es donde se almacena los datos de la imagen en escala de gris

Y el número (:,:,1); corresponde a la constante del parámetro de un color que para este caso es el color rojo; para el color verde el valor cambiaria a 2 y para el color azul el valor seria 3

Luego se procede a la binarización

#### *11.1.1.2.3. Binarización*

Consiste en una fase o herramienta que debe cumplirse en aquellos proyectos que realicen procesamiento de video o imágenes y que deba arrojar ciertos resultados, como el tamaño del área del objeto que se esté analizando, esto quiere decir, que para calcular el tamaño del área del objeto es necesario banalizarla y filtrarle todo los ruidos. Todo es con la finalidad arrojar un mejor resultado. La instrucción que se utiliza para este caso es:

**diff\_im = im2bw(diff\_im,0.18);**

Donde

**Im2bw:** Es el comando que lleva a cabo el proceso de binarización a la variable **diff\_im** que contiene la imagen en escala de gris

#### 11.1.1.2.4. Filtrado de la imagen

Es el paso siguiente que complementa el proceso de binarización, se utiliza para cada color que se desea procesar, con el objetivo de tener una imagen limpia de ruidos o impurezas que capte la cámara. La instrucción utilizada es:

```
diff_im = medfilt2(diff_im, [3 3]);
```

Donde

**medfilt2:** Es el comando de filtrado seguido de la variable a la cual se le está aplicando el filtrado

#### 11.1.2. Software arduino

Se elabora un código en Arduino el cual permitirá la interacción entre el MatLab y la placa de desarrollo Arduino con el fin de recibir las coordenadas de posición del objeto captado por la cámara web y posteriormente transmitido a la placa, para que se pueda realizar los movimiento en los dos servos motores

#### 11.1.3. Requisitos Para El Procesamiento

Se procedieron a realizar las pruebas del sistema de visión artificial, en una computadora con las siguientes especificaciones del sistema. Ver **Figura 9**.

---

## Ver información básica acerca del equipo

### Edición de Windows

---

Windows 7 Professional


Copyright © 2009 Microsoft Corporation. Reservados todos los derechos.

Service Pack 1

[Obtener más características con una nueva edición de Windows 7](#)

### Sistema

---

Fabricante:	Dell
Evaluación:	 <a href="#">Evaluación de la experiencia en Windows</a>
Procesador:	Intel(R) Core(TM) i5-3330 CPU @ 3.00GHz 3.00 GHz
Memoria instalada (RAM):	8,00 GB (7,89 GB utilizable)
Tipo de sistema:	Sistema operativo de 64 bits

***Figura 9. Información del Equipo donde se ejecuta el Programa.***

En caso tal de utilizar un equipo con recursos más limitados, causaría que todo el proyecto funcionara con mayor lentitud, y en el peor de los casos bloqueo de todo el sistema operativo del computador, causando el error que muchos conocen de pantalla azul, el cual indica que todos los recursos de memoria y procesamiento son consumidos por algún proceso.



## 11.2. Implementación De Hardware

Se darán a conocer los elementos utilizados para la construcción del dispositivo

### 11.2.1. Dispositivo De Adquisición

Para la adquisición, se utiliza una cámara web de marca **STAR TEC** la cual cuenta con las siguientes especificaciones técnicas:



*Figura 10. Cámara Star Tec*

Modelo: ST-HC-326

Balance de blancos

Interface: USB

Enfoque manual

Botón de obturador

Resolución: 1.3 Mpx

### 11.2.2. Sistema Rotatorio Con Dos Grados De Libertad

Este sistema rotatorio se adquiere teniendo en cuenta que son partes utilizadas para la construcción de brazos robóticos y que se encuentran elaborados en aluminio, material resistente y liviano. <sup>22</sup>

<sup>22</sup>[http://www.didacticasselectronicas.com/index.php/search?keyword=pan+and+tilt&limitstart=0&option=com\\_virtuemart&view=category&virtuemart\\_category\\_id=0](http://www.didacticasselectronicas.com/index.php/search?keyword=pan+and+tilt&limitstart=0&option=com_virtuemart&view=category&virtuemart_category_id=0)



**Figura 11. Base Servo motor**

A continuación, se dan a conocer las especificaciones y los elementos que trae la estructura:

- Soporte multifuncional de Aluminio (1 unidades)
- Soporte en forma de U de Aluminio (1 unidad)
- Soporte tipo copa (1 unidad)
- Juegos de tuercas (1 juego)

### 11.2.3. Servomotores



**Figura 12. Servomotores**

Básicamente un servomotor es un motor de corriente continua con un potenciómetro que le permite saber la posición en la que se encuentra y así poder controlarla.

Para controlar el servomotor se le envía pulsos cada 20 ms es decir 50Hz. La anchura del pulso es lo que codifica el ángulo de giro, es decir lo que se conoce como PWM,

codificación por ancho de pulso. Esta anchura varía según el servomotor pero normalmente va entre 0.5 y 2.5 ms aunque pueden variar.

Dependiendo del tamaño del servo y su consumo es posible que no puedas alimentarlo desde tu placa Arduino, para este caso, se utilizan servos estándar de referencia DF05BB por lo que consume poca corriente y se puede alimentar directamente por el Arduino.

Se emplean los servos motores de referencia DF05BB, los cuales cuentan con las especificaciones idóneas para lo que se pretende desarrollar.<sup>23</sup>

Especificaciones de los servos DF05BB:

- Voltaje de operación: 4.8 ~ 6V
- Corriente de trabajo: 0,1 ~ 0,8A
- Torque: 4.8Kg.cm a 4.V y 5.5Kg.cm a 6V
- Velocidad media: 60 rpm
- Gama de rotación: 0 - 120°
- Temperatura de funcionamiento: 0°C ~ 55°C
- Tamaño: 41x20x38mm
- Peso: ~ 48g

#### 11.2.4. Tarjeta De Desarrollo Programable Arduino

Para darle cumplimiento de uno de los objetivos específicos, el cual consiste en Realizar la comunicación o cargar el programa previamente probado a la tarjeta de desarrollo seleccionada, para poder unificar con la estructura que ya se tiene identificada a utilizarse en el montaje del sistema mecánico, se utiliza la plataforma de desarrollo Arduino, el cual esta se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas

<sup>23</sup> <http://www.electroensaimada.com/servomotor.html>

embebidos en proyectos multidisciplinarios. Toda la plataforma, tanto para sus componentes de hardware como de software, son liberados con licencia de código abierto que permite libertad de acceso a ellos.

El hardware consiste en una placa de circuito impreso con un microcontrolador, usualmente Atmel AVR, puertos digitales y analógicos de entrada/salida, los cuales pueden conectarse a placas de expansión (shields), que amplían las características de funcionamiento de la placa Arduino. Asimismo, posee un puerto de conexión USB desde donde se puede alimentar la placa y establecer comunicación con el computador.

Se emplea el ARDUINO UNO R3 el cual cumple con las características idóneas para la implementación del prototipo <sup>24</sup>



**Figura 13. Placa Arduino UNO**

**Características:**

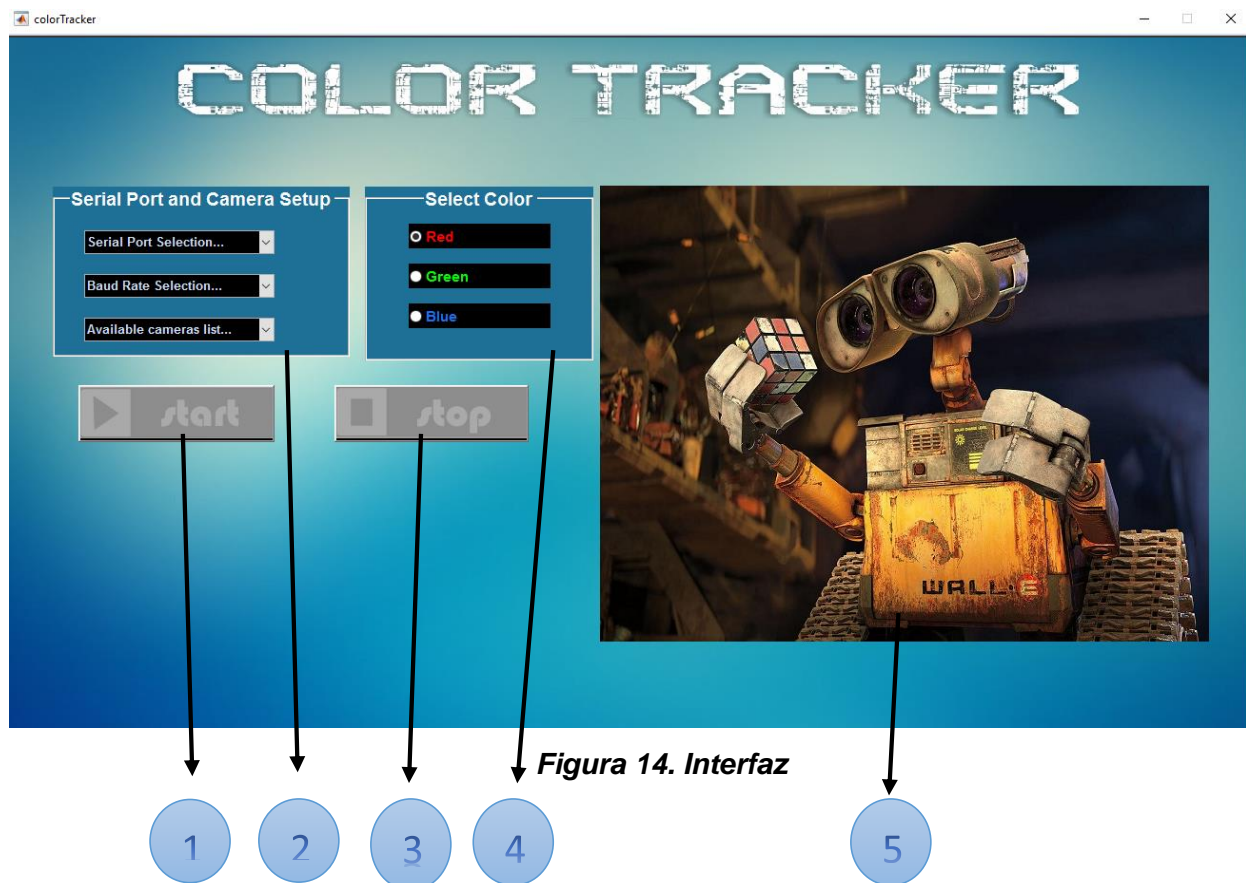
- Micro controlador ATmega328.
- Voltaje de entrada 7-12V.
- 14 pines digitales de I/O (6 salidas PWM).
- 6 entradas análogas.
- 32k de memoria Flash.      -Reloj de 16MHz de velocidad.

<sup>24</sup> <http://arduino.cl/arduino-uno/>

## 12. PRUEBAS Y RESULTADOS

### 12.1. EJECUCION INTERFAZ GRÁFICA DEL PROYECTO




Al ejecutar el código “**trackercolor.m**”, Se mostrará un GUIDE, ver figura. Y corresponde la interfaz inicial del programa desarrollado en MatLab



A continuación se hará una descripción de las partes que conforman la interfaz inicial del programa.

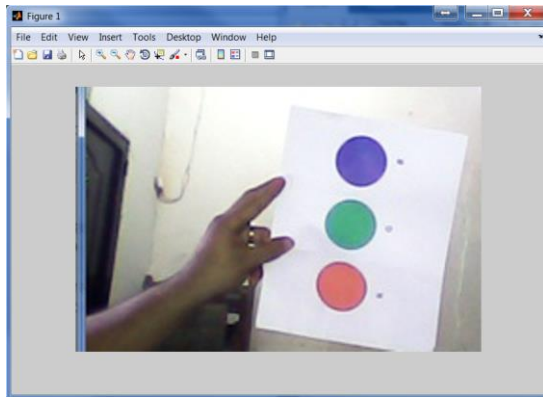
- 1 **Botón START**, Es el encargado de dar inicio o de colocar en funcionamiento el procesamiento del código.
- 2 **Serial Port and Camera Setup**, contiene tres listas, en la primera se encontraran los puertos utilizados para la comunicación del arduino, es decir el COM que

utiliza la placa de desarrollo. La segunda lista corresponde a la velocidad de comunicación que se desea utilizar, este valor varía dependiendo de la placa utilizada. Como en este proyecto se está utilizando la placa arduino Mega, se utiliza un baut rate de 115200. Si se escoge un valor diferente, este debería ser remplazado en la programación del arduino y reprogramar la placa. Y la tercera lista se encontraran las cámaras que tenga conectado el ordenador; este sería un caso específico para laptops los cuales traen incorporada una webcam, y como el prototipo cuenta con una, se hace necesario identificarlas para que el usuario escoja la del proyecto y no la del computador.

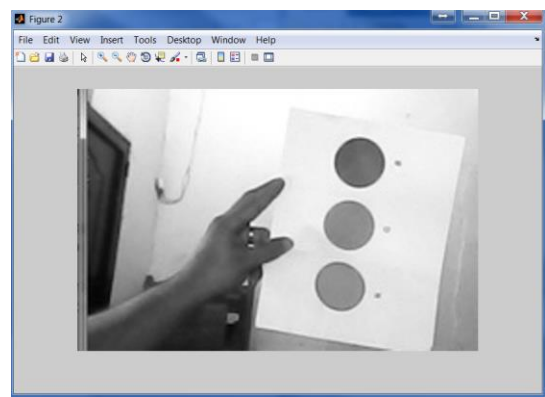
-  **Botón STOP**, Es el encargado de detener en cualquier momento la ejecución del código, esto será posible siempre y cuando el usuario desee detener la ejecución del programa, además al ser pulsado finalizara toda la comunicación serial, además de realizar una limpieza de los registros de memoria, esto se hace con el fin de evitar que el programa no consuma todo los recursos del computador
-  **Select Color**, Corresponde al recuadro en el cual se podrá seleccionar el color deseado al que se desea realizar el seguimiento. Por defecto, cada vez que se proceda a ejecutar el archivo trackercolor.m, este mostrará seleccionado el color rojo.
-  Es el recuadro que permitiría observar el video en tiempo real que es capturado por la cámara web; adicionalmente se podrá apreciar un recuadro que encierra el objeto en cuestión con su respectivas coordenadas en eje X y en el eje Y.

## 12.2. Escala de gris

El resultado obtenido al aplicar el comando descrito anteriormente en el desarrollo es:



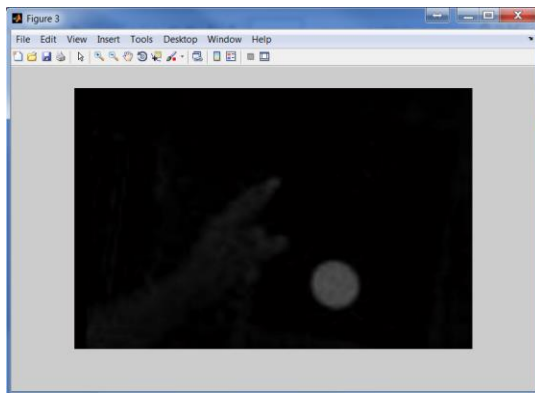
**Figura 15 Imagen Original sin Escala de Grises**



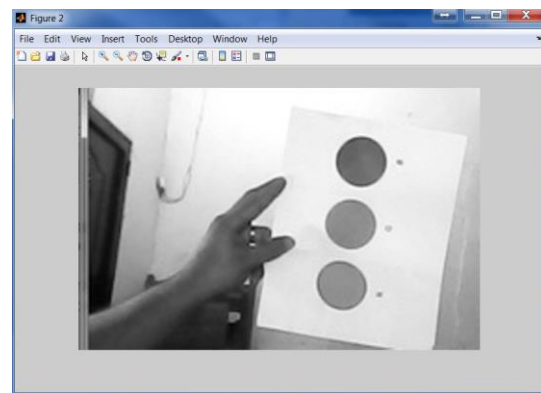
**Figura 16 Imagen en Escala de Grises**

## 12.3. Segmentación

El resultado obtenido al aplicar el comando descrito anteriormente en el desarrollo es:



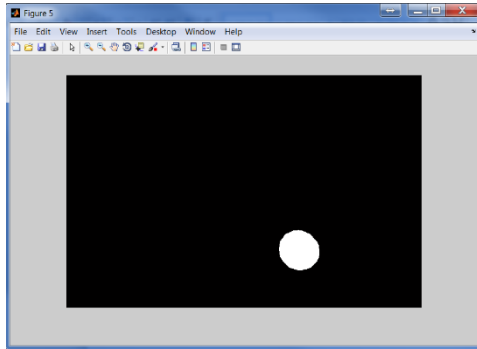
**Figura 18 Imagen Segmentada (extracción del Color Rojo)**



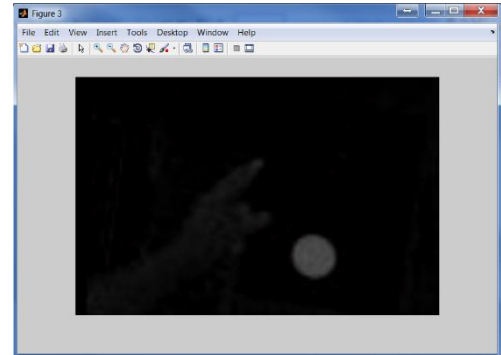
**Figura 17 Imagen en Escala de grises**

## 12.4. Binarización

El resultado obtenido al aplicar el comando descrito anteriormente en el desarrollo es:



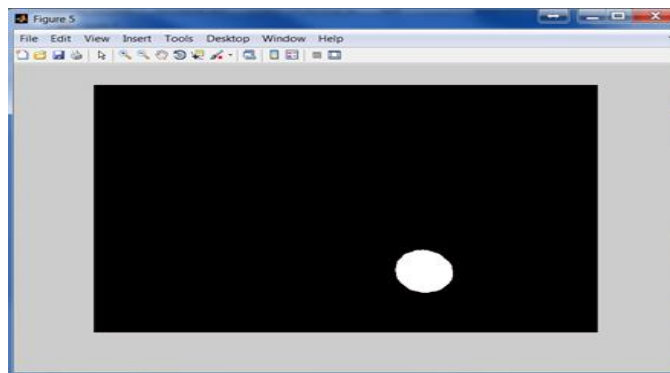
*Figura 20 Imagen Segmentada*



*Figura 19 Imagen Binarizada*

## 12.5. Filtrado de la imagen

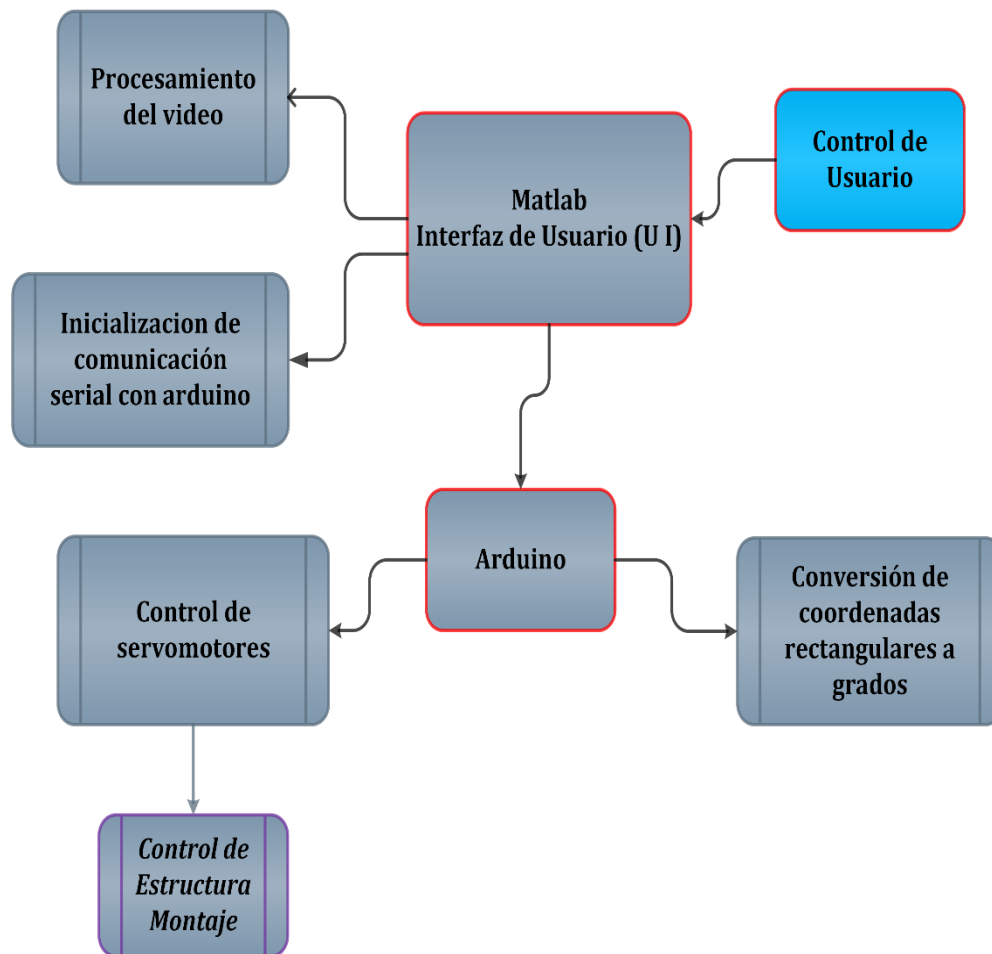
El resultado obtenido al aplicar el comando descrito anteriormente en el desarrollo es:



*Figura 21 Imagen Filtrada*

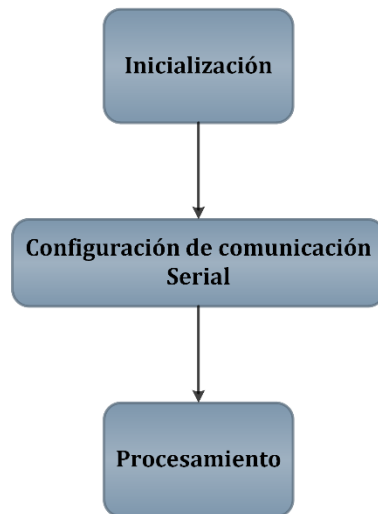


## 12.6. Diagrama General De Comunicación

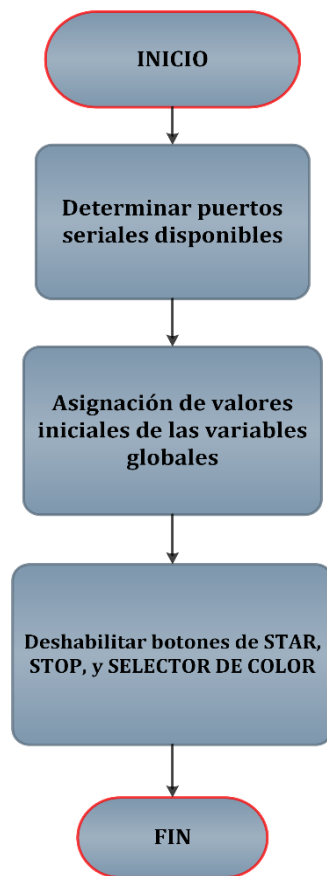


**Figura 22 Esquema general de Funcionamiento**

## 12.7. Pre Procesamiento En MatLab

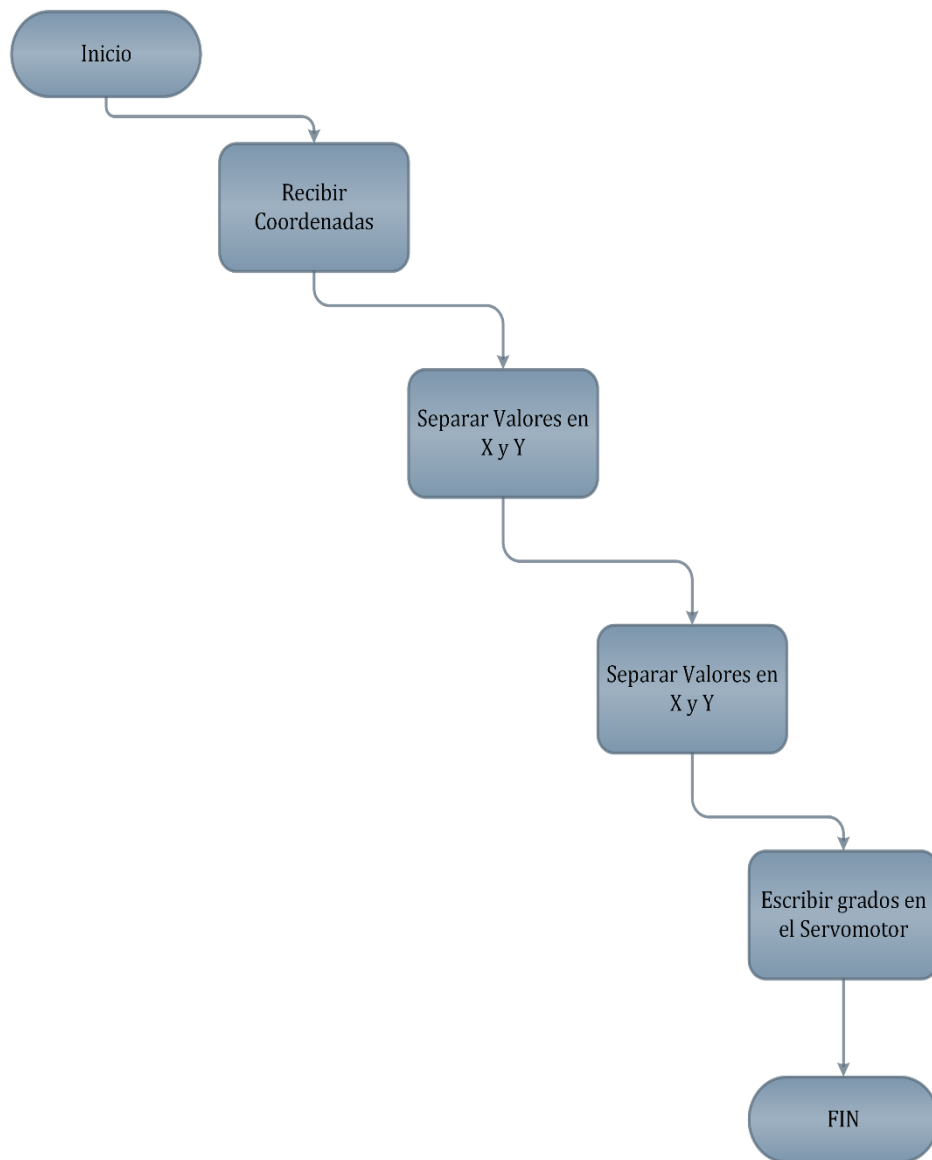


**Figura 23. Interfaz de Usuario (UI)**



**Figura 23.1. Inicialización Procesamiento**

## 12.8. Control Para El Movimiento



**Figura 24. Interfaz Arduino**

### 13. Distancia Entre La Cámara Y Objeto A Seguir.

Al realizarse las respectivas pruebas, se logra determinar que la distancia, mínima y máxima recomendada para el buen funcionamiento en la detección y seguimiento de objetos e independientemente del color y tipo de muestra utilizada es en promedio de 0.60 m (distancia mínima) a 3.60 metros (distancia máxima). Si el objeto a seguir ya sea Rojo, Verde o Azul, sobrepasa esta distancia, pueda que el funcionamiento del prototipo se vea afectado y no se presente ya que se saldría del rango de visión. Para tener mayor detalle de lo descrito en el presente párrafo, se debe revisar los encisos referentes a las pruebas y resultados del proyecto.

#### 13.1. Tablas de Pruebas

A continuación se dará a conocer las pruebas realizadas al dispositivo con el fin de medir los límites de reconocimiento de objetos mediante el dispositivo en metros

**Caso 1:** Prueba de distancia, con un tamaño del objeto de 10x10 cm, y distintas muestras del Rojo puro hasta el Rojo más oscuro, ver Tabla 1 y muestras del Rojo claro hasta el Rojo más puro, ver Tabla 2. Complementado el caso 1 con figura 25 en el cual se evidencia el resultado.

Prueba	Tamaño del objeto a Seguir	Distancia entre el prototipo y el objeto de color a seguir (Metros)	Muestras de Color Rojo		
			R	G	B
1	10 x 10 Cm	3.40 a 3.60	255	0	0
2	10 x 10 Cm	3.40 a 3.60	229	0	0
3	10 x 10 Cm	3.40 a 3.60	204	0	0
4	10 x 10 Cm	3.25 a 3.40	178	0	0

5	10 x 10 Cm	3.20 a 3.30	153	0	0
6	10 x 10 Cm	3.10 a 3.20	127	0	0
7	10 x 10 Cm	2.40 a 2.50	102	0	0
8	10 x 10 Cm	60 cm a 70 cm	76	0	0
9	10 x 10 Cm	<b>No Detectado</b>	51	0	0
10	10 x 10 Cm	<b>No Detectado</b>	25	0	0

**Tabla 1:** Color Rojo puro hasta el más oscuro

Prueba	Tamaño del objeto a Seguir	Distancia entre el prototipo y el objeto de color a seguir (Metros)	Muestras de Color Rojo		
			R	G	B
1	10 x 10 Cm	<b>No Detectado</b>	255	229	229
2	10 x 10 Cm	<b>No Detectado</b>	255	204	204
3	10 x 10 Cm	<b>No Detectado</b>	255	178	178
4	10 x 10 Cm	1.90 a 2.00	255	153	153
5	10 x 10 Cm	3.10 a 3.20	255	127	127
6	10 x 10 Cm	3.15 a 3.25	255	102	102
7	10 x 10 Cm	3.40 a 3.60	255	76	76
8	10 x 10 Cm	3.40 a 3.60	255	51	51
9	10 x 10 Cm	3.40 a 3.60	255	25	25

**Tabla 2:** Color Rojo Claro hasta el más puro



**Figura 25.** Prueba del funcionamiento Color rojo detectado; Muestra: 102.0.0

**Caso 2:** Prueba de distancia, con un tamaño del objeto de 10x10 cm, y distintas muestras del Verde puro hasta el Verde más oscuro, ver Tabla 3 y muestras del Verde claro hasta el Verde más puro, ver Tabla 4. Complementando el caso 2 con figura 26 en el cual s evidencia el resultado.

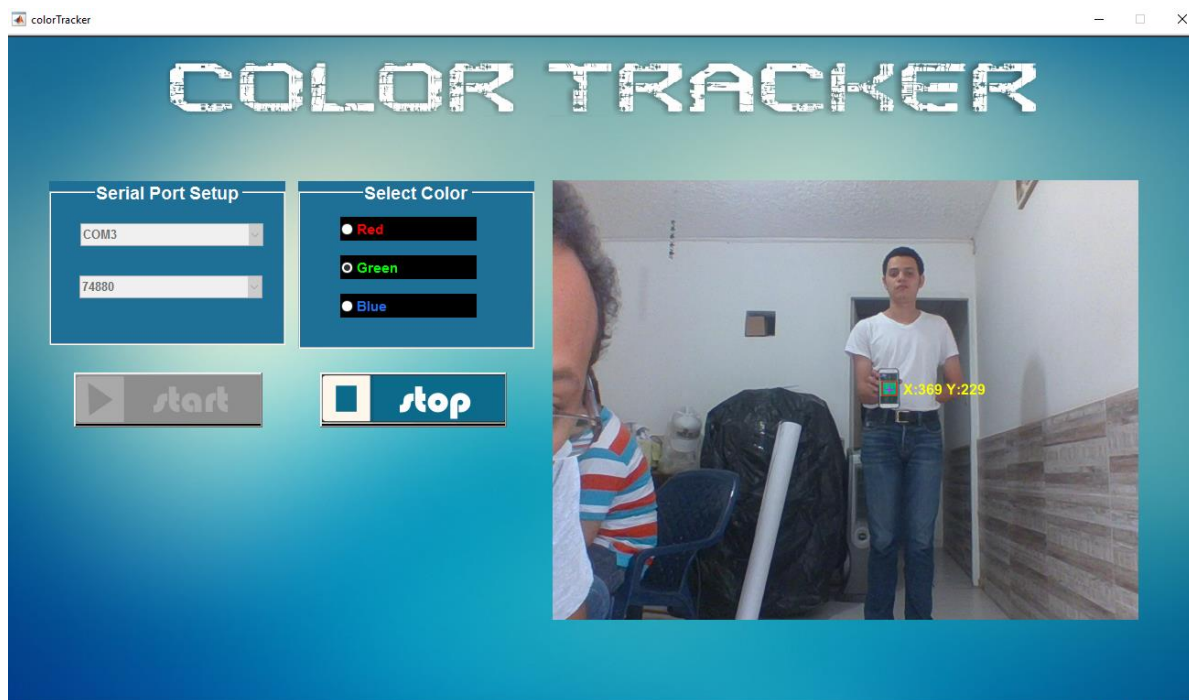
Prueba	Tamaño del objeto a Seguir	Distancia entre el prototipo y el objeto de color a seguir (Metros)	Muestras de Color Rojo		
			R	G	B
1	10 x 10 Cm	3.40 a 3.60	0	255	0
2	10 x 10 Cm	3.40 a 3.60	0	229	0
3	10 x 10 Cm	3.40 a 3.60	0	204	0
4	10 x 10 Cm	3.40 a 3.60	0	178	0
5	10 x 10 Cm	3.40 a 3.60	0	153	0
6	10 x 10 Cm	3.20 a 3.30	0	127	0

7	10 x 10 Cm	1.80 a 1.90	0	102	0
8	10 x 10 Cm	No reconocido	0	76	0
9	10 x 10 Cm	No reconocido	0	51	0
10	10 x 10 Cm	No reconocido	0	25	0

**Tabla 3:** Color Verde puro hasta el más oscuro

Prueba	Tamaño del objeto a Seguir	Distancia entre el prototipo y el objeto de color a seguir (Metros)	Muestras de Color Rojo		
			R	G	B
1	10 x 10 Cm	No reconocido	229	255	229
2	10 x 10 Cm	No reconocido	204	255	204
3	10 x 10 Cm	No reconocido	178	255	178
4	10 x 10 Cm	No reconocido	153	255	153
5	10 x 10 Cm	20 cm a 40 cm	127	255	127
6	10 x 10 Cm	2.30 a 2.40	102	255	102
7	10 x 10 Cm	2.70 a 2.80	76	255	76
8	10 x 10 Cm	2.80 a 3.00	51	255	51
9	10 x 10 Cm	No reconocido	25	255	25

**Tabla 4:** Color Verde claro hasta el más puro



**Figura 26.** Prueba del funcionamiento Color verde detectado, Muestra: 0.0.102

**Caso 3:** Prueba de distancia, con un tamaño del objeto de 10x10 cm, y distintas muestras del Azul puro hasta el Azul más oscuro, ver tabla 5 y muestras del Azul claro hasta el Azul más puro, ver Tabla 6. Complementando el caso 3 con figura 27 en el cual se evidencia el resultado.

Prueba	Tamaño del objeto a Seguir	Distancia entre el prototipo y el objeto de color a seguir (Metros)	Muestras de Color Rojo		
			R	G	B
1	10 x 10 Cm	3.10 a 3.20	0	0	255
2	10 x 10 Cm	2.90 a 3.00	0	0	229
3	10 x 10 Cm	2.80 a 2.90	0	0	204
4	10 x 10 Cm	2.80 a 2.90	0	0	178
5	10 x 10 Cm	2.70 a 2.80	0	0	153

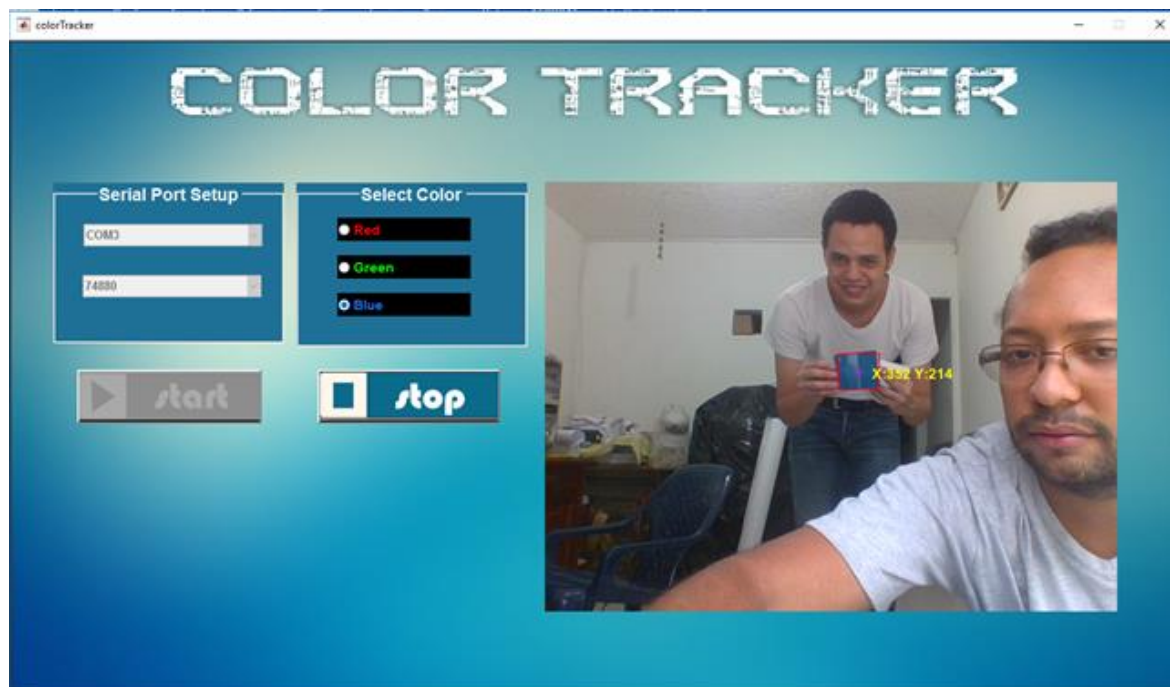


<b>6</b>	10 x 10 Cm	2.60 a 2.65	0	0	127
<b>7</b>	10 x 10 Cm	2.55 a 2.60	0	0	102
<b>8</b>	10 x 10 Cm	1.20 a 1.35	0	0	76
<b>9</b>	10 x 10 Cm	No reconocido	0	0	51
<b>10</b>	10 x 10 Cm	No reconocido	0	0	25

**Tabla 5:** Color Azul puro hasta el más oscuro

Prueba	Tamaño del objeto a Seguir	Distancia entre el prototipo y el objeto de color a seguir (Metros)	Muestras de Color Rojo		
			R	G	B
<b>1</b>	10 x 10 Cm	No reconocido	229	229	255
<b>2</b>	10 x 10 Cm	2.60 a 2.70	204	204	255
<b>3</b>	10 x 10 Cm	3.00 a 3.10	178	178	255
<b>4</b>	10 x 10 Cm	3.20 a 3.30	153	153	255
<b>5</b>	10 x 10 Cm	3.30 a 3.40	127	127	255
<b>6</b>	10 x 10 Cm	3.30 a 3.40	102	102	255
<b>7</b>	10 x 10 Cm	3.30 a 3.40	76	76	255
<b>8</b>	10 x 10 Cm	3.30 a 3.40	51	51	255
<b>9</b>	10 x 10 Cm	2.90 a 3.00	25	25	255

**Tabla 6:** Color Azul claro hasta el más puro



**Figura 27.** Prueba del funcionamiento Color Azul detectado, Muestra: 0.0.76

## 14. PRESUPUESTO

### 14.1. Recursos Humanos

A continuación se describen a las personas que hacen parte de este proyecto, con su formación académica y su función dentro del mismo.

PERSONAL	FORMACION ACADEMICA	FUNCION DENTRO DEL PROYECTO	COSTO
Omar Francisco Rodríguez Álvarez	Ms.C. en educación	Director	\$ 0
José Ramón Iglesias Gamarra	Ph.D. en Automatización, Robótica e Ingeniería en materiales	Asesor	\$ 0
Carlos Alberto	Estudiante de ingeniería electrónica	Desarrollador del proyecto	\$ 0
Rafael Jesús Acosta Jiménez	Estudiante de ingeniería electrónica	Desarrollador del proyecto	\$ 0
<b><i>Costo Total de Presupuesto Humano</i></b>			<b>\$ 0</b>

## 14.2. Recursos Institucionales

Son recursos necesarios para el desarrollo del proyecto. Estos recursos hacen parte de nuestra institución y abarcan los recursos físicos, tecnológicos y de software

RECURSO	CANTIDAD	VALOR DEL RECURSO	VALOR TOTAL
Computador de escritorio DELL OptiPlex 9020	1	\$ 2.700.000	\$ 2.700.000
Laboratorio de procesamiento de Señales	1	\$ 0	\$ 0
Licencia del Software de Matlab para uso académico con los toolbox requeridos	1	\$ 1.800.000	\$ 1.800.000
<b>Costo Total de Presupuesto Institucional</b>			<b>\$ 4.500.000</b>

## 14.3. Recursos Adicionales

A continuación se describirán los materiales y elementos que se requieren comprar tanto en el contexto de hardware, software, papelería, transporte, uso de internet, entre otros.

### 14.3.1. Presupuesto En Hardware

MATERIALES	CANTIDAD	VALOR / UNIDAD	TOTAL
Tarjeta de desarrollo <b>Arduino UNO R3</b>	1	\$ 50.000	\$ 50.000
Servo motores Estándar <b>DF05BB</b>	2	\$ 25.000	\$ 50.000
Estructura rotatoria con dos grados de libertar	1	\$ 50.000	\$ 50.000
Cámara web	1	\$ 150.000	\$ 150.000
Conectores para la tarjeta de desarrollo	10	\$ 2.000	\$ 20.000
<b>Costo Total de Presupuesto en Hardware</b>			<b>\$ 320.000</b>

#### 14.3.2. Presupuesto En Software

HERRAMIENTA DE SOFTWARE	CANTIDAD	VALOR / UNIDAD	TOTAL
Software para la tarjeta de desarrollo Arduino UNO	1	\$ 0	\$ 0
<b>Costo Total de Presupuesto en Software</b>			<b>\$ 0</b>

#### 14.4. Presupuesto Varios

Descripción	TOTAL
Papelería	\$ 300.000
Internet	\$ 220.000
Transporte	\$ 150.000
Otros	\$ 150.000
<b>Costo Total de Presupuesto Varios</b>	<b>\$ 820.000</b>

#### 14.5. Costo Total Del Proyecto

El presupuesto total para el desarrollo del proyecto está determinado por el costo total del presupuesto humano, institucional y adicional.

<i>Costo Total de Presupuesto Humano</i>	\$ 0
<i>Costo Total de Presupuesto Institucional</i>	\$ 4.500.000
<i>Costo Total de Presupuesto Adicional</i>	\$ 1.140.000
<b>Costo Total de Presupuesto del proyecto</b>	<b>\$ 5.640.000</b>

## 15. CONCLUSIONES

A manera de conclusión, se puede dar a conocer que para la construcción del proyecto de grado que lleva como título Sistema De Identificación y seguimiento de objetos de colores RGB para una plataforma rotatoria mediante visión artificial empleando el software de MatLab, inicia como un prototipo base para apoyar a futuro, otros trabajos de investigación; Como es el caso entre la Universidad Del Magdalena y La Universidad Autónoma Del Caribe de la ciudad de Barranquilla la cual pretenden realizar un proyecto de investigación de mayor complejidad que tiene como título Identificación de objetivos y asignación de tareas a un agente robótico mediante visión estereoscópica.

Al analizar la parte estructural que sería la encargada de llevar instalados dos servo motores los cuales permitirían el movimiento, y al detallar el material que se pretendía utilizar que en principio era la madera por sus propiedades de fácil tallado, se observó que la simetría que se debía tener entre las partes que conformarían la estructura, tenían que ser lo más precisa posible y construidas en una máquina de gran precisión tipo cnc, ya que si no se lograba, esto causaría esfuerzos y movimientos mecánicos no adecuado del agente robótico. Por lo anterior mencionado y con la finalidad de evitar estos inconvenientes y muchos más, lo utilizado fueron las partes de la base de un brazo robótico, las cuales están construidas en aluminio, material resistente y liviano. Ver figura 11 del ítem implementación de hardware.

Uno de los inconvenientes encontrados consistía en que para poder realizar el movimiento de los dos grados de libertad, fueron necesarias tomar las coordenadas que nos arrojaba el programa de MatLab ya que dichos valores eran en números enteros y las coordenadas que los servomotores pueden leer para sus movimientos son en grados. Este inconveniente se solucionó haciendo una conversión de los datos.

La tarjeta de desarrollo seleccionada es la de Arduino, específicamente la Arduino uno ya que brinda una facilidad de compra al igual que su programación la cual es viable para lo que se pretendía desarrollar, además de ser una plataforma de código abierto, razón

por la cual se emplea para abaratar costos tanto en la compra de la tarjeta como la licencia del software que para la plataforma Arduino es Open Source, la cual es una Iniciativa para utilización de Código Abierto tanto para software como hardware.

Al realizarse las respectivas pruebas que permitieron determinar la distancia, mínima y máxima recomendada para el buen funcionamiento en la detección y seguimiento de objetos, independientemente del color y tipo de muestra utilizada que se encuentre en el rango idóneo para ser detectado por el dispositivo. Se determina que el rango de la distancia detectable es de 0.60 m (distancia mínima) a 3.60 metros (distancia máxima). Si el objeto a seguir ya sea Rojo, Verde o Azul, sobrepasa esta distancia, el funcionamiento del prototipo se verá afectado y no se presenta de la mejor manera ya que se saldría del rango de focalización de la cámara. Y si las pruebas son realizadas en lugares donde la luminancia sea deficiente, tampoco se observaría el buen funcionamiento

La velocidad de respuesta al momento de aparecer en distintos puntos la muestra de color es la adecuada para lo que se requiere mostrar en el proyecto. Este punto será observable para los usuarios que realicen la prueba, ya que es imposible demostrar por medio de imágenes los resultados de dicha prueba.

Para los usuarios que no tengan en cuenta las pautas de como colocar en funcionamiento el dispositivo, se les podría dificultar ya que en estos pasos se explican los elementos que conforman a la interfaz además de su configuración y función.

Ya para finalizar este documento, es importante hacer ver que a la hora de construir el prototipo y el software que hacen posible que este proyecto funcione a cabalidad, nos damos cuenta que todas y cada una de las asignaturas cursadas a lo largo de la carrera fueron indispensables para llegar al resultado final que se esperaba obtener y que se está dando a conocer. Permitiéndonos así cumplir con el objetivo general del trabajo el cual consistía en desarrollar un sistema de identificación y seguimiento de objetos de colores RGB empleando visión artificial en MatLab, para una plataforma robótica rotatoria de tal manera que se pueda realizar un seguimiento en tiempo real para evitar perderle de vista el objeto.

## 16. REFERENCIAS BIBLIOGRAFICAS

### 16.1. Bibliografía

[1] PEREZ, Lina M. Sistema de Visión Estereoscopica en El Proyecto Libélula.  
29.11.2011

[2] BUHMANN, Joachim M.; MALIK, Jitendra; PERONA, Pietro. Image recognition: Visual grouping, recognition, and learning. PNAS \_ December 7, 1999 vol. 96 \_ no.25 \_ 14203 –14204.

[3] Acosta L(2009). La visión Artificial,Universidad Tecnologica de Bolivar, Cartagena, Colombia.

[4] Rincones G. & Ruiz M. (2013). *Desarrollo de un prototipo electrónico basado en vision Artificial para la estimacion de la densidad de usuarios en Playas turísticas* (Tesis de Grado). Universidad del Magdalena, Santa Marta, Colombia.

[5] Muñoz M., Altafulla V. *Diseño E Implementación De Un Sistema De Visión Por Computador Y Procesamiento De Video Para La Reconstrucción De Trayectorias Caóticas De Peces* (Tesis de Grado)( 2013). Universidad del Magdalena, Santa Marta, Colombia.

[6] Moreno A., Ballesteros D. *Prototipo Para La Automatización Del Proceso De Clasificación Del Mango Tipo Exportación (Mangifera Indica L) Basado En Visión Artificial.* (tesis de Grado)(2010). Universidad del Magdalena, Santa Marta, Colombia.

[7] Proakis, Jhon G., Manolakis, Dimitris G., Tratamiento Digital de Señales, Prentice Hall, 4ta edición, 2007.



- [8] Soliman, Samir S., Srinath, Mandyam D., Señales y Sistemas Continuos y Discretos, 2da Edición, Prentice Hall, 1999
- [9] Lathi, B. P, Signal Processing and Linear Systems, Berkeley - Cambridge Press, 1998.
- [10] Mallat, Stéphane. A Wavellet Tour in Signal Processing 2do Edition, Academic Press, 1999.
- [11] Wavelets and Signal Processing - An Application-Based Introduction, Springer, 2005
- [12] Bovid, Alan C., Handbook of Image and Video processing 1st edition, Academic Press, 2000.
- [13] Gonzalez, Rafael C., Woods, Richard E., Digital Image Processing, 2nd Edition, Prentice Hall, 2002

## 16.2. Webgrafía

[1] [http://dmi.uib.es/~ygonzalez/VI/Material\\_del\\_Curso/Teoria/Aplicaciones\\_VC.PDF](http://dmi.uib.es/~ygonzalez/VI/Material_del_Curso/Teoria/Aplicaciones_VC.PDF)

[2] [www.pnas.org](http://www.pnas.org)

[3] <http://sabia.tic.udc.es/gc/trabajos%202011-12/VisionEstereoscopica/principal.html>

[4] <https://sites.google.com/site/grafcomputacional/estereoscopia/topografía>

[5] <https://es.mathworks.com/discovery/matlab-gui.html>

[6] <https://es.wikipedia.org/wiki/Arduino>

[7] <http://arduinodhtics.weebly.com/tipos-de-arduino.html>

[8] [http://www.didacticaselctronicas.com/index.php/search?keyword=pan+and+tilt&limit\\_start=0&option=com\\_virtuemart&view=category&virtuemart\\_category\\_id=0](http://www.didacticaselctronicas.com/index.php/search?keyword=pan+and+tilt&limit_start=0&option=com_virtuemart&view=category&virtuemart_category_id=0)

[9] <http://www.electroensaimada.com/servomotor.html>

[10] <http://www.tech-faq.com/rgb.html>

[11] <http://arduino.cl/arduino-uno/>

[12] <https://es.slideshare.net/lonely113/procesamiento-digital-de-imgenes-con-matlab>

## **17. ANEXOS**

### **17.1. Guía Para Realizar El Enlace Arduino MatLab**

Lo que se desea enseñar a través de esta guía es como realizar un enlace o conexión entre la placa ARDUINO y MatLab. Entre las prestaciones básicas del MATLAB están: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware.

El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI).

Además, se pueden ampliar las capacidades de MatLab con las cajas de herramientas (toolbox); y las de Simulink con los paquetes de bloques (blocksets). Este párrafo anterior nos indica la gran capacidad de funcionalidad del MatLab, este es el motivo por el cual se realiza esta guía, porque de esta forma podemos tener un gran software para analizar, evaluar y controlar nuestra placa ARDUINO.

<sup>25</sup> <https://es.slideshare.net/lonely113/procesamiento-digital-de-imgenes-con-matlab>

### 17.1.1. Preguntas Frecuentes

- **¿Qué software necesito?**

Se debe tener descargada la última versión del ARDUINO, alguna versión superior a la 2010a del MATLAB y un paquete desarrollado para MATLAB el cual nos permitirá hacer el enlace.

ARDUINO: <https://www.arduino.cc/en/Main/Donate>

Paquete de soporte Matlab – Arduino:

<http://www.mathworks.com/matlabcentral/fileexchange/32374-matlab-support-package-for-arduino-aka-arduinoio-package?download=true>

- **¿Con cuales tipos de placa ARDUINO es funcional?**

Esta guía estará dedicada a la versión de la placa: “ARDUINO UNO”.

- **¿Qué podemos hacer con este enlace ARDUINO - MATLAB?**

Esto es tan funcional, que podríamos crear un programa dentro del MATLAB haciendo uso del GUI, el cual al hundir algún botón podamos observar una acción en la vida real por parte de la placa. Ejemplo: Apagar o encender un LED o bombillo, controlar un brazo con servomotores, entre otros proyectos que pueden llegar a ser muy ambiciosos.

### 17.1.2. Procedimiento

1. Tener instalado el MATLAB, y descargar los archivos mencionados en las preguntas frecuentes.
2. Abrir la carpeta “arduino-1.0.4-windows” y luego el “ARDUINO”.

3. Conectar la placa ARDUINO a través del cable USB.
4. Verificar por cual “COM” quedo conectado. Luego dirigirse a Inicio > Equipo “Click derecho” > Administrar > Administrador de dispositivos > Puertos > Arduino Mega 2560, al lado derecho de “Arduino Mega 2560” sale el COMX, siendo X un nuevo entre 0 y 6 usualmente.
5. Dentro del ARDUINO ir a Herramientas > Tarjeta y elegir la “placa ARDUINO MEGA 2560”
6. De nuevo ir a Herramientas>Puerto Serial y seleccionar nuestro puerto COMX, que ubicamos en el paso 4.
7. Vamos a introducir un sketch dentro de nuestra placa ARDUINO. Este sketch nos permitirá hacer la conexión entre el ARDUINO Y MATLAB, se le instala este sketch a la placa para que pueda entender las órdenes enviadas desde el MATLAB.
8. Dentro del Arduino: Archivo > Abrir Buscamos dentro de los archivos descargados en la siguiente dirección: ArduinoIO>pde>adiosrv>adiosrv. Pde
9. Teniendo el adiosrv abierto le damos a “cargar” y esperamos hasta que diga “Carga terminada.”
10. Cerramos el “ARDUINO” para dejar de tener el puerto ocupado, así poder establecer ahora la conexión a través del MATLAB.
11. Se abre como administrador el MATLAB, MATLAB > „Click derecho”>Ejecutar como administrador.
12. El directorio “Current Folder” lo ubicamos dentro de la carpeta ArduinoIO.

13.Procedemos a realizar la instalación adecuada insertando los comandos:

14.Digitamos “install\_arduino” (sin comillas). Si sale error es que hicieron mal el paso 11, o escribieron mal “install\_arduino”.

15.Ahora digitamos “a=arduino ('COMX')” (sin comillas). Donde X es el número hallado en el paso 4. Justo ahora debería de aparecer muchas instrucciones. Dichas instrucciones entendiéndolas orientaran al usuario en la programación.

16.Tras haber ejecutado todo el procedimiento, justo ahora tenemos ya la conexión hecha entre MATLAB y ARDUINO. Luego solo se debe cargar el programa adjunto en el archivo.

## **17.2. Manual de uso del software**

### **17.2.1. Funcionamiento del software**

Para colocar en funcionamiento el software, se debe verificar que se cumplan los 3 primeros pasos que son fundamentales

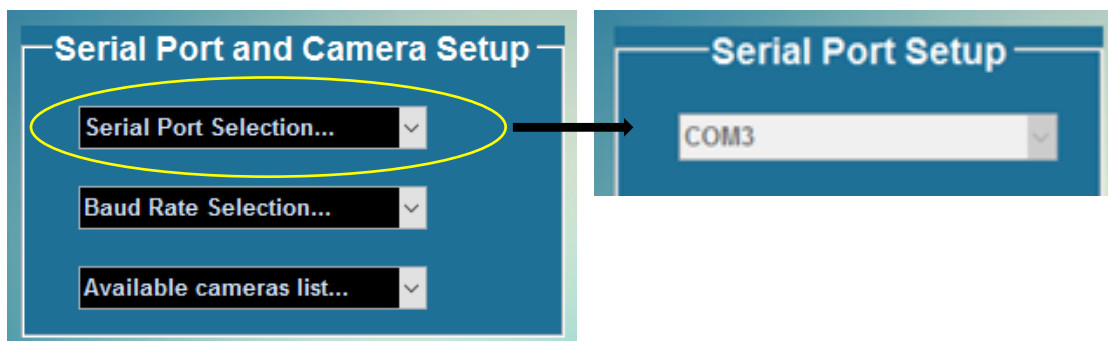
1. Tener instalado el software de MatLab, versión 2017
2. Instalar los driver controladores de la webcam Star Tec Modelo: ST-HC-326
3. Realizar la conexión arduino MatLab descrita en el ítem 16.1 correspondiente a los Anexos
4. Al realizar los 3 pasos anteriores, se procede a ejecutar el archivo “trackercolor.m” o “trackercolor.fig” que dará como resultado a la siguiente interfaz

- Para colocar en funcionamiento la interfaz, se procede a realizar los siguientes pasos



**Figura 28** interfaz inicializada

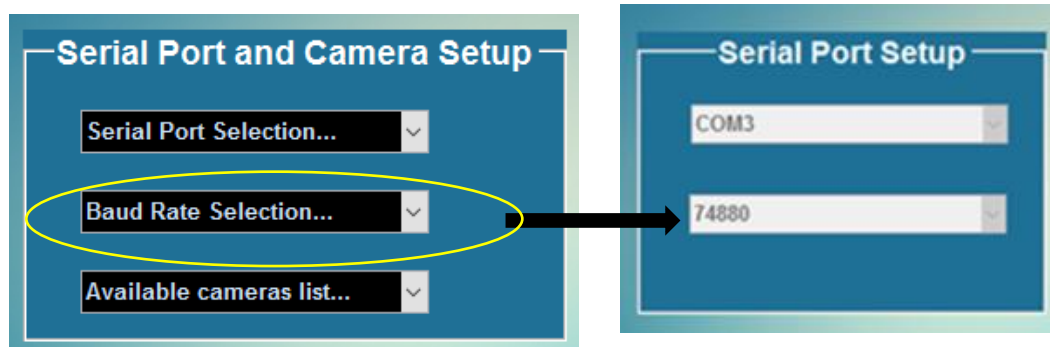
- Se debe seleccionar el puerto serial para que se presente la comunicación Arduino MatLab



**Figura 29** Selección puerto serial en interfaz

- seleccionar el valor de baud rate que corresponde a la velocidad de transferencia y comunicación para el envío del empaquetamiento de coordenadas hacia el Arduino. Se observara un listado con los valores permitidos a utilizar, pero se debe seleccionar el valor de 115200, ya que en el código del Arduino se está utilizando

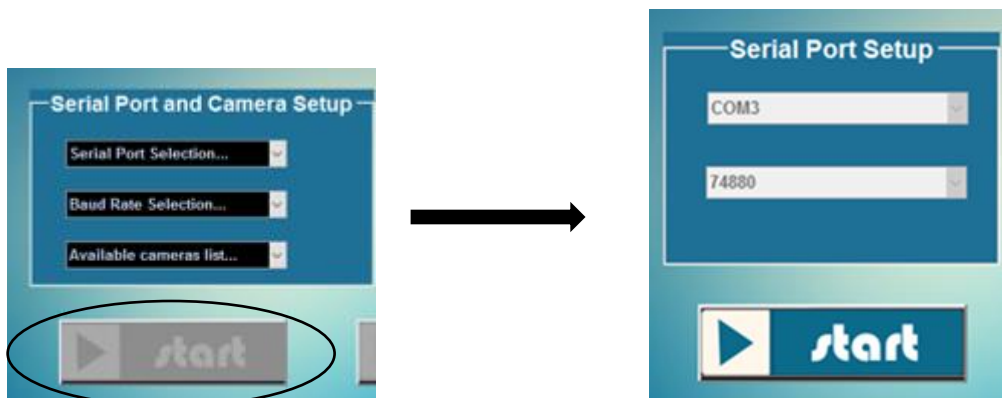
dicho valor. Sí se selecciona un valor distinto al descrito anteriormente y no sea cambiado en el código del Arduino, el aplicativo no funcionará.



**Figura 30** Selección de Rata de Baudios

8. Seleccionar la cámara instalada en el dispositivo de nombre Star tec

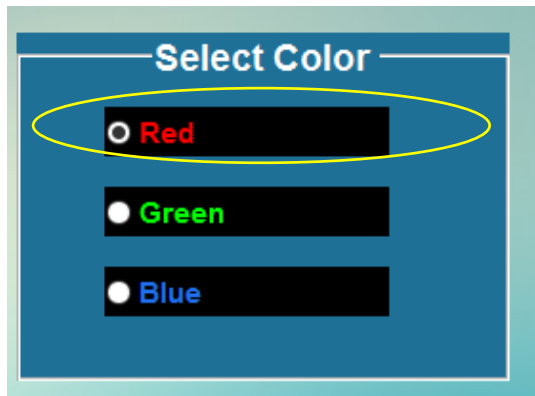
Nota: Si el recuadro de serial port and camera setup no se configura, no se habilitara el botón Start



**Figura 25** Serial Port No configurado y después de la configuración



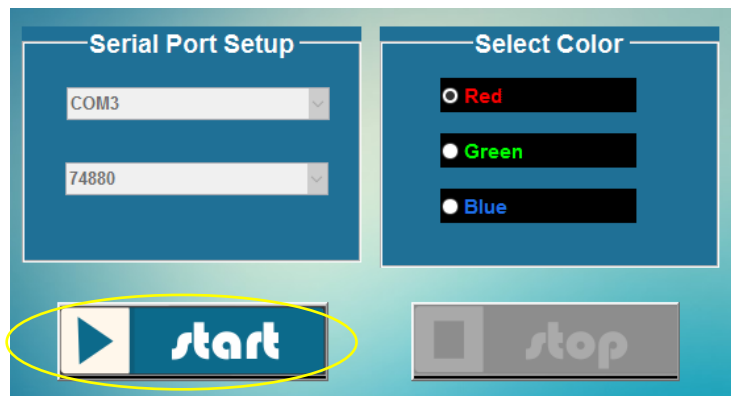
9. se debe escoger uno de las 3 opciones de colores,



**Figura 26** Selección de colores

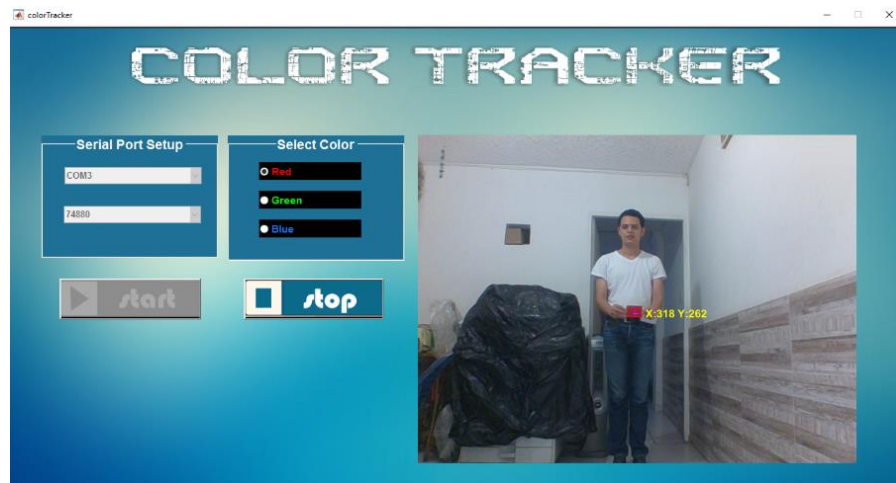
Nota: Por defecto siempre quedara seleccionado el color rojo, pero se puede escoger otro color antes de presionar el botón Start

10. el aplicativo se encuentra listo para que se presione el botón start



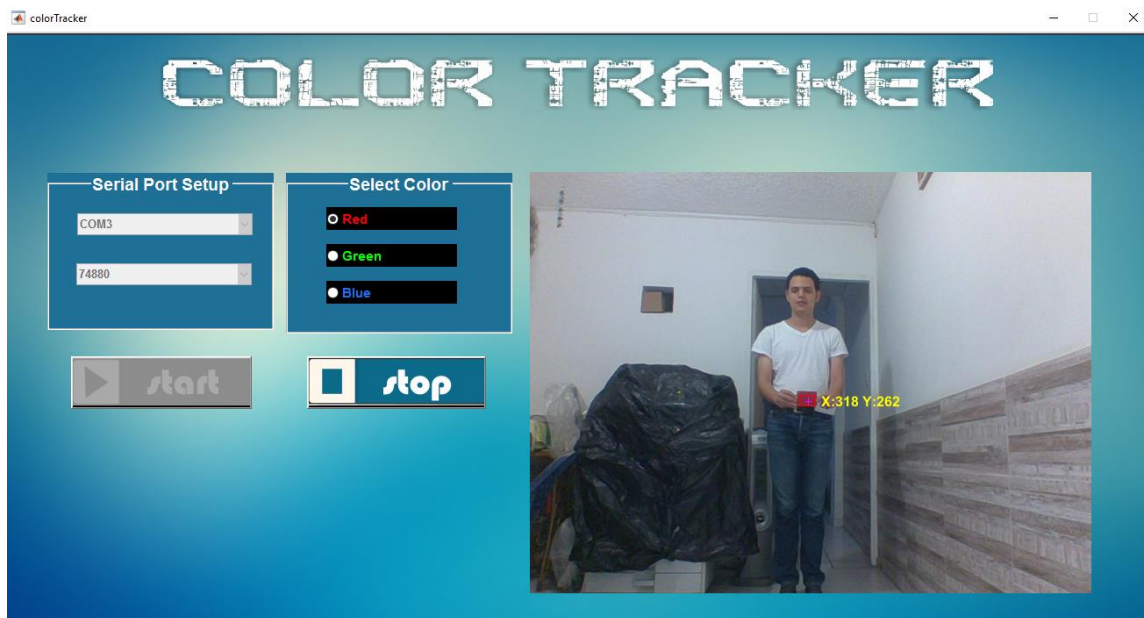
**Figura 273** listo para presionar start

11. al presionar el botón Start, se deshabilita el recuadro serial port setup y el botón start, esto se realiza para evitar que el usuario presione accidentalmente cualquier opción y altere la configuración y el funcionamiento del proyecto.



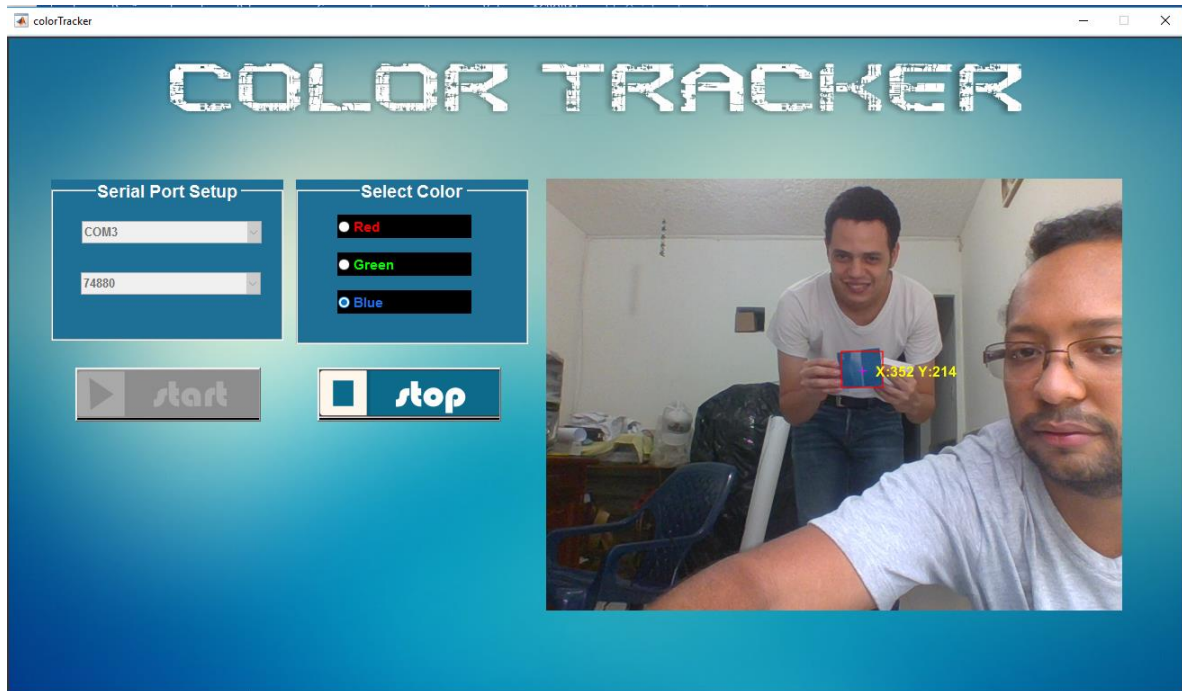
**Figura 28**

12. se observara la detección y el seguimiento del objeto con el color seleccionado en la parte derecha de la ventana.



**Figura 29**

14. proyecto, se puede cambiar el color a seguir sin necesidad de colocar en Stop el aplicativo.



**Figura 30**

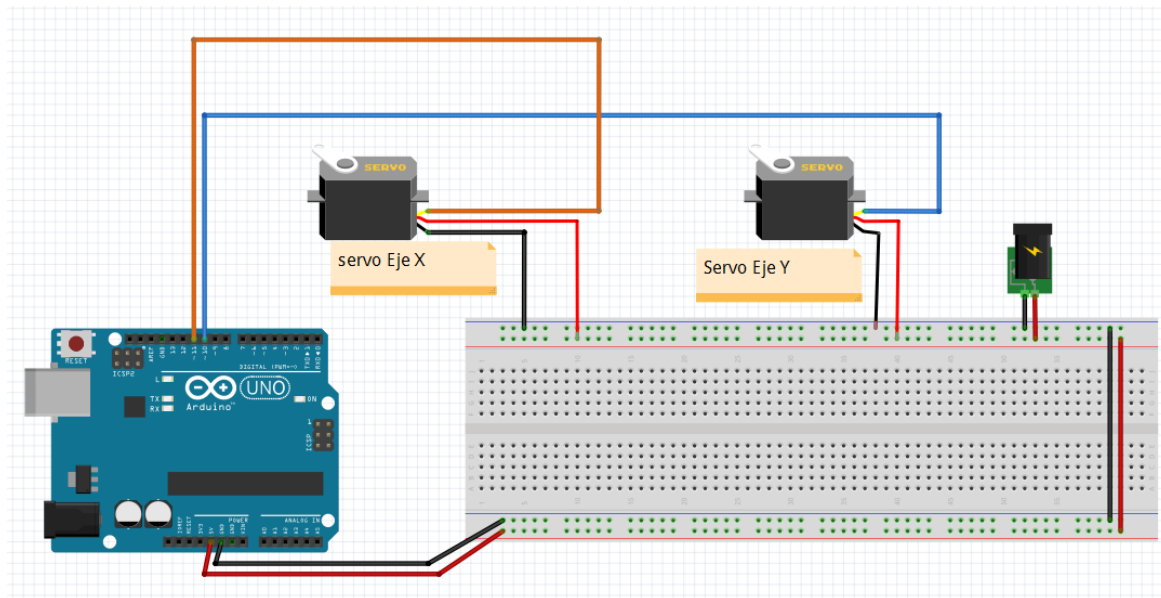
15. para detener toda la simulación, se debe presionar el botón stop

Nota: Si se cierra la ventana del aplicativo sin detener la simulación con el botón Stop, y al momento de querer colocarla nuevamente en funcionamiento el prototipo, es posible que se genere error, puesto que la comunicación serial port no se ha cancelado de un todo. La solución es reiniciar todo el software de MatLab y configurar nuevamente los pasos 6 y 7.

### 17.3. Esquema de conexión de hardware.

Conexión que se debe tener en cuenta para el funcionamiento del prototipo

Como el proyecto de investigación utiliza dos servos motores para realizar el movimiento tanto en el **Eje X** como en el **Eje Y**, y estos a su vez manejan 3 cables. El tipo de conexión que se debe realizar se mostrara a continuación.



**Figura 31** Esquema de Hardware

#### 17.4. Imágenes Del Prototipo Terminado



17.5. Colores usados en las pruebas

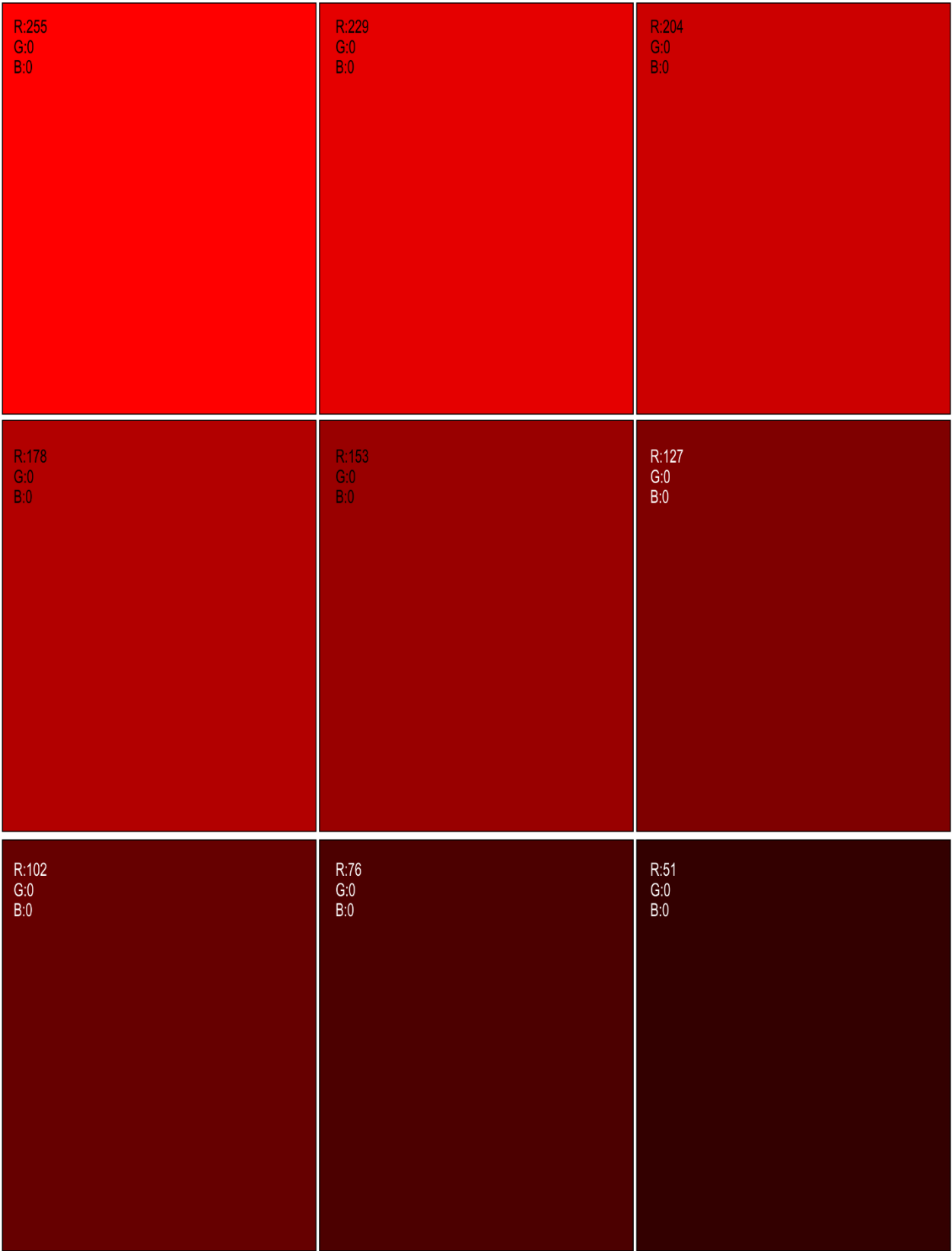
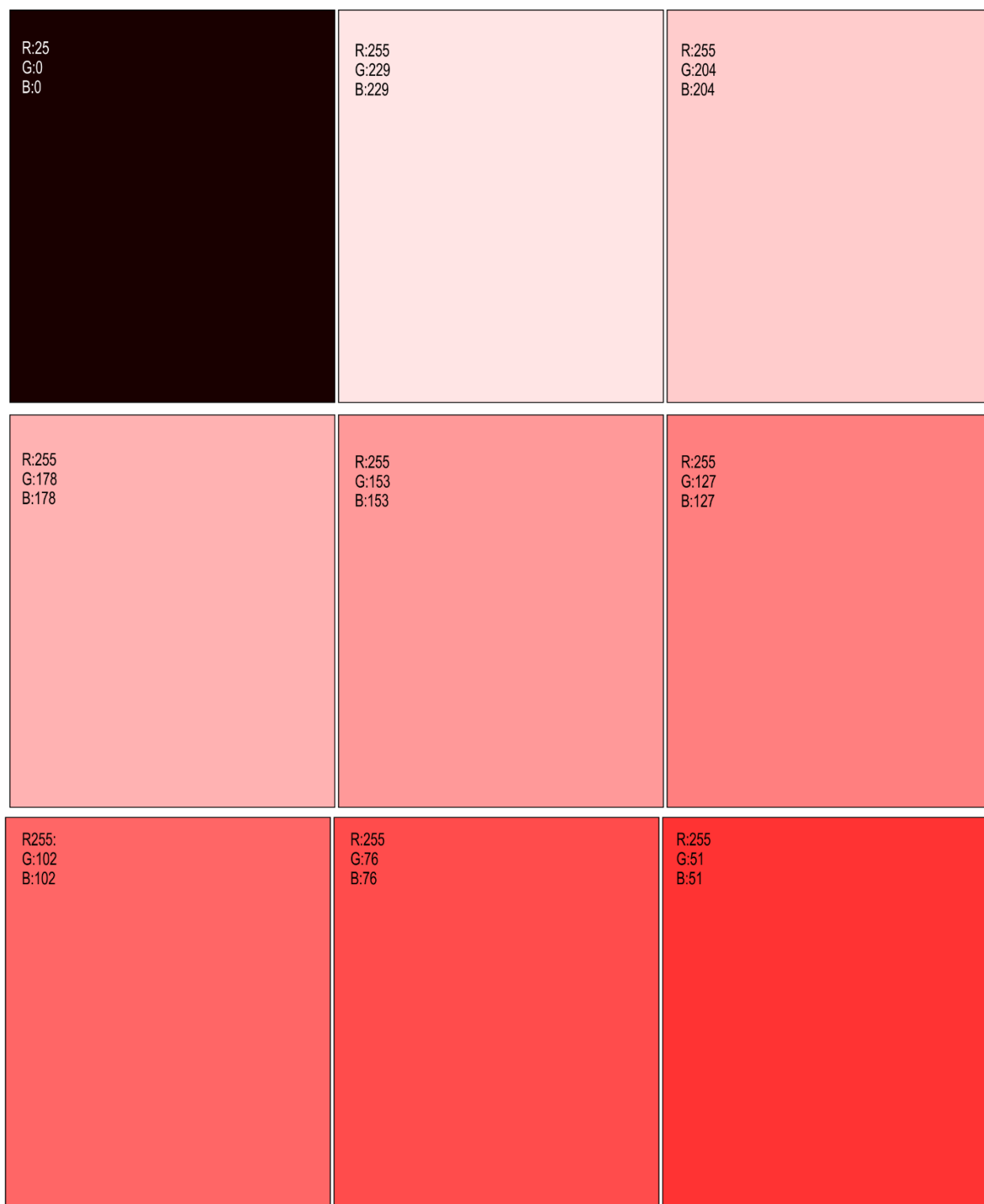
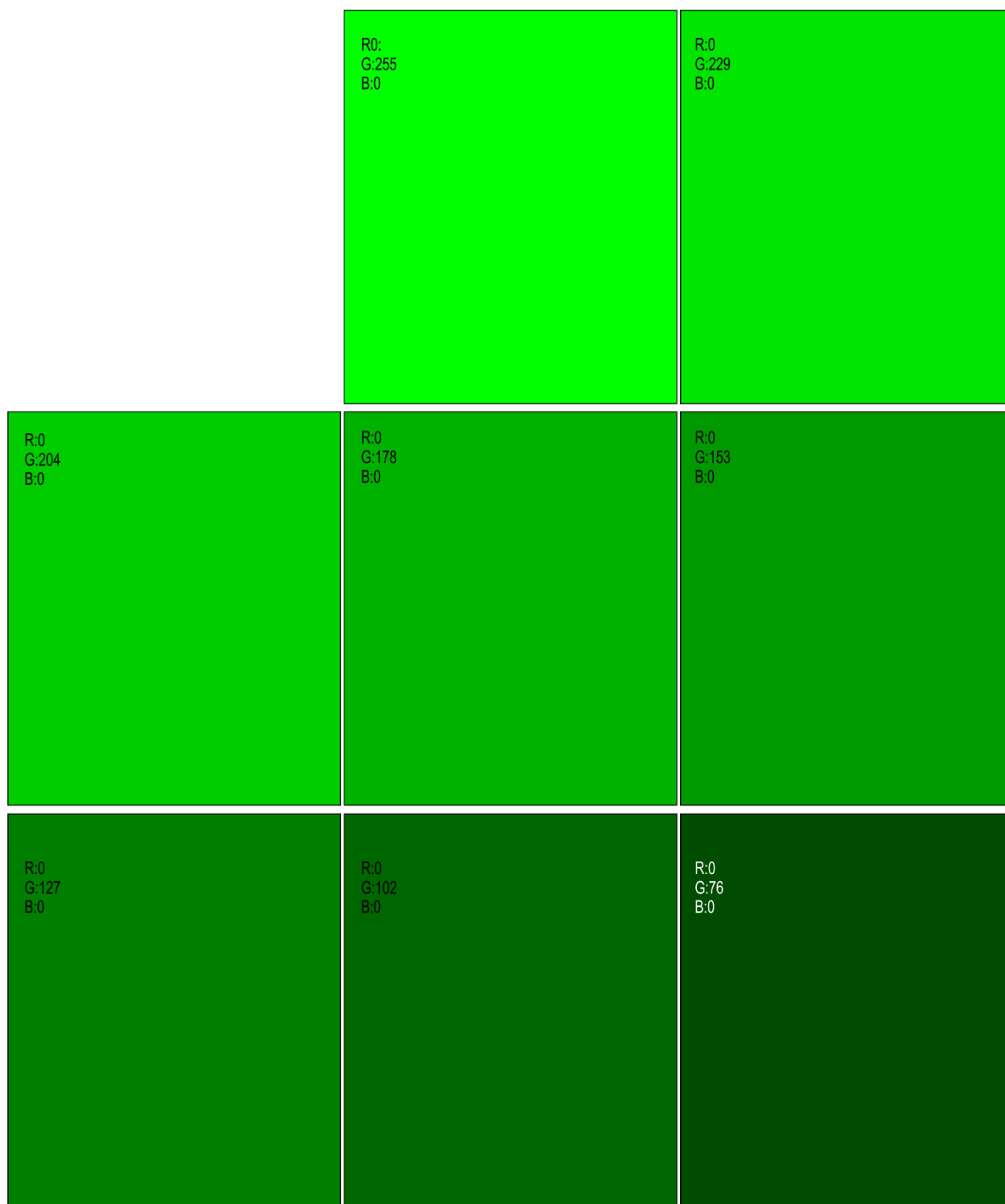


Figura 32 paleta RGB de color rojo 1

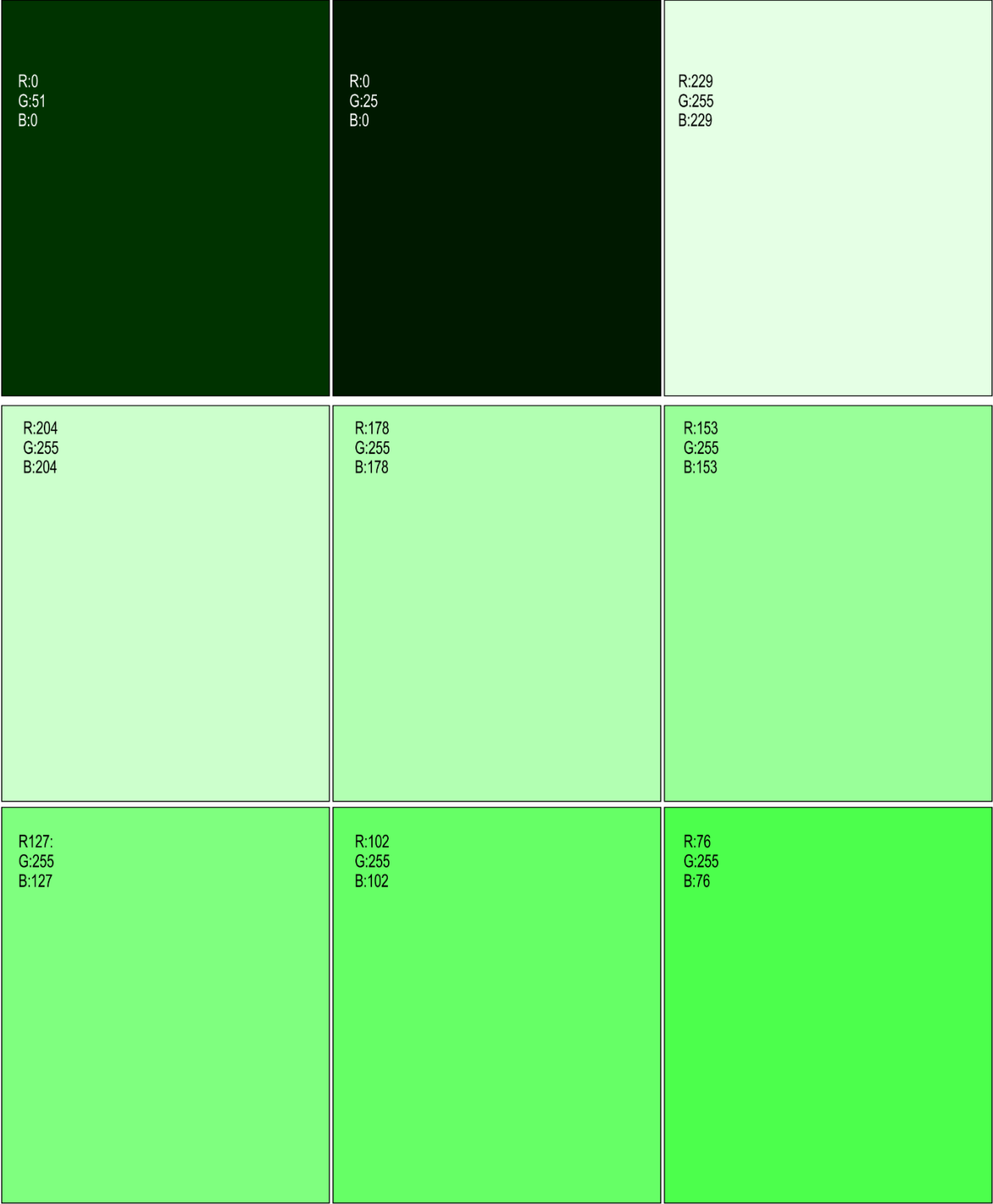


**Figura 33 paleta RGB de color rojo 2**

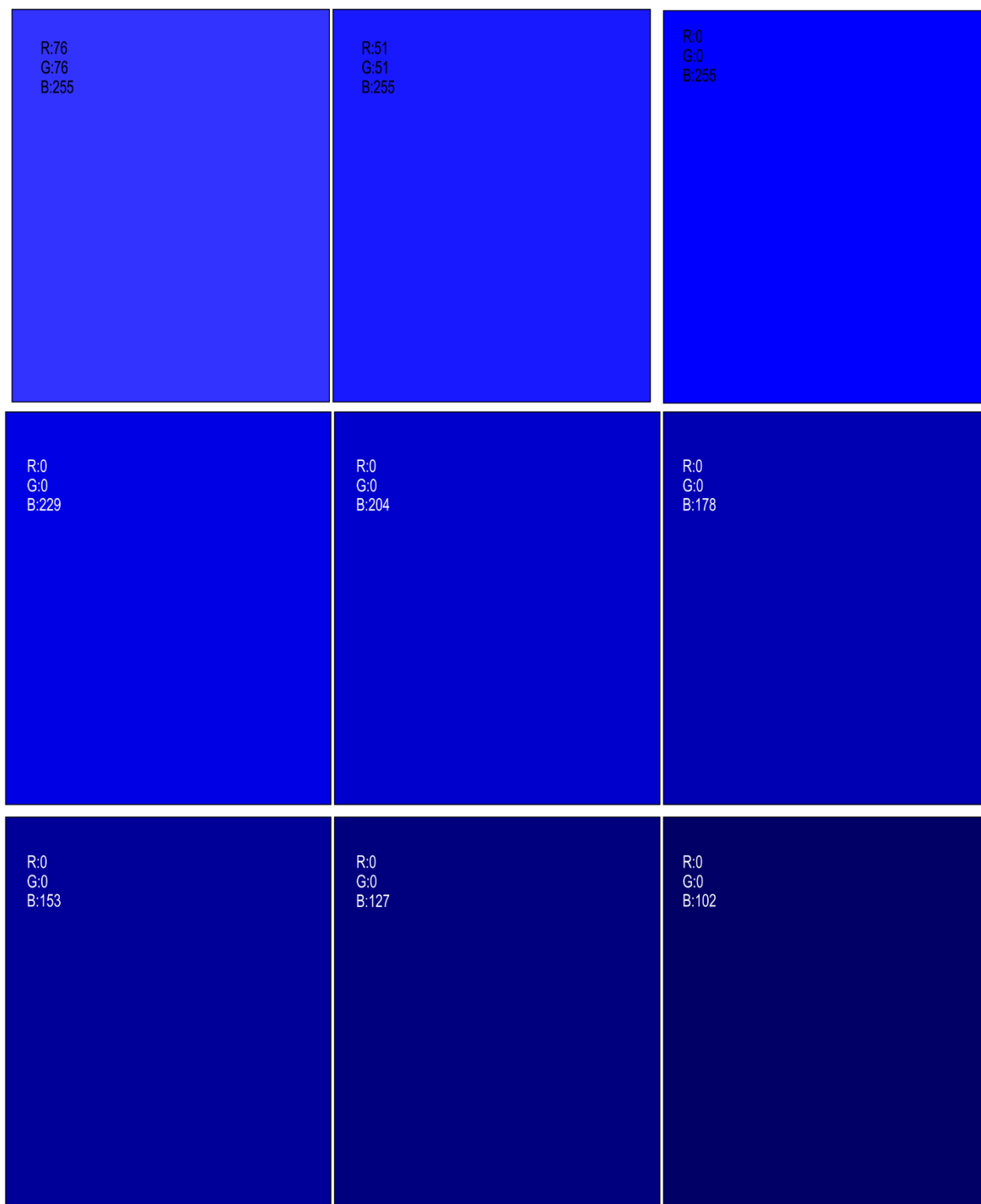


***Figura 34 paleta RGB de color Verde 1***

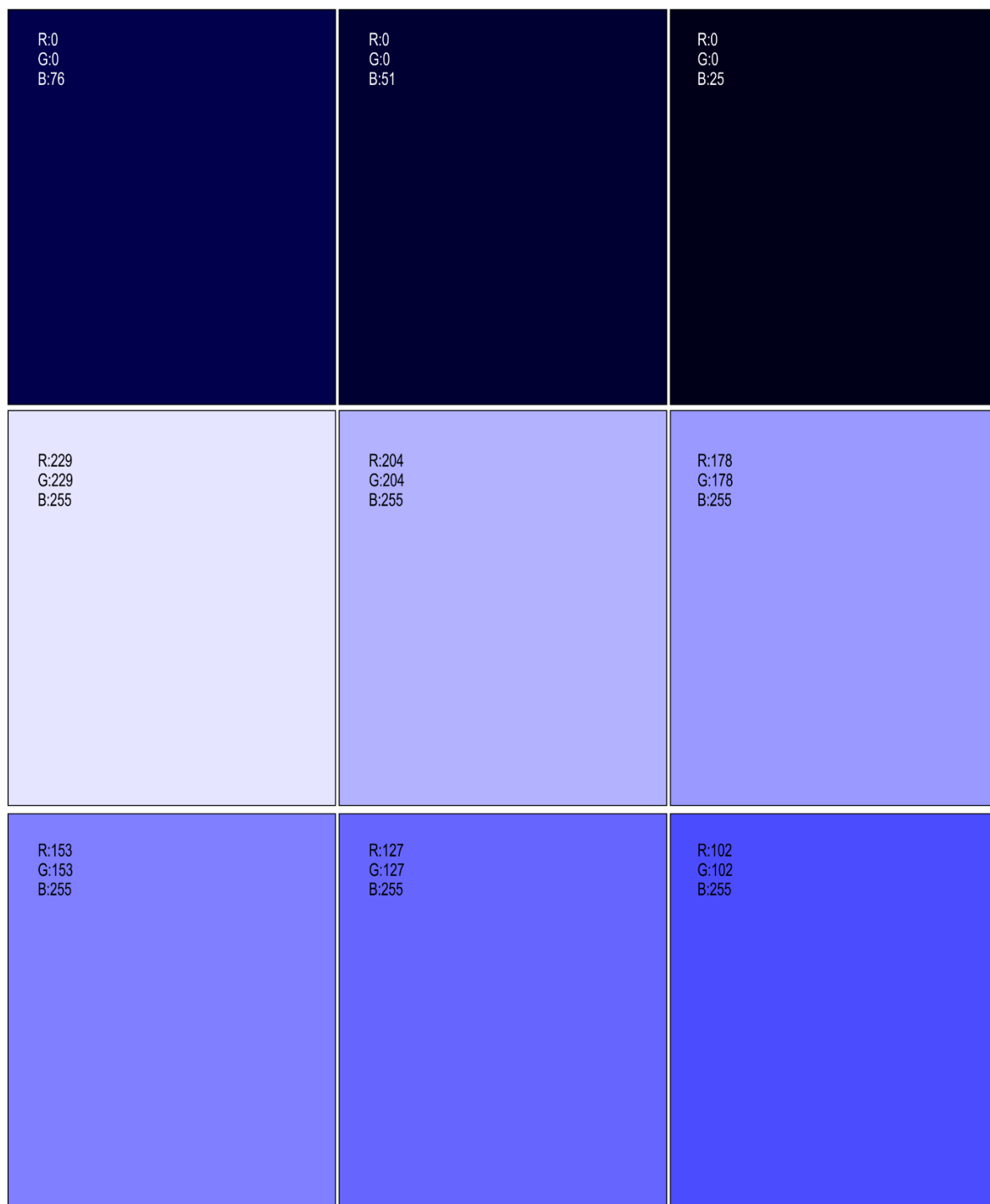




**Figura 35 paleta RGB de color Verde 2**



***Figura 36 paleta RGB de color Azul 1***



***Figura 37 paleta RGB de color azul***

## **18. RECOMENDACIONES**

El mecanismo de rotación implementado para este proyecto, es utilizado para la construcción de brazos robóticos y dichos elementos son controlados por diversos motores, para nuestro caso se utilizaron servos motores con un margen de movimiento de 0° a 120, no se implementaron otros tipos de motores “giros continuos” ya que la estructura limitaría los movimientos.

Para un buen funcionamiento del prototipo se recomienda que el entorno al cual sea colocado no tenga elementos de colores que afecten el funcionamiento, es preferible que tenga un fondo blanco y buena iluminación, ya que en entornos oscuros no podría funcionar perfectamente. Cabe resaltar la revisión del ítem sobre el manual de uso de la interfaz del software.

## **19. TRABAJO FUTURO**

Para mejorar el proyecto de investigación implementado se puede pensar en una estructura que permita realizar movimientos de 360° tanto en el eje **X** como en el eje **Y**, ya que a una estructura de este tipo y complementándola con los servomotores correctos que permitan realizar dichos movimientos, se lograría que el aplicativo no presentara limitación alguna al momento de realizar el seguimiento.

Para la parte del procesamiento se implementaría inicialmente una cámara de mayor resolución permitiendo así obtener más detalles de resolución del color a seguir, además de ampliar la gama de formatos de colores a seguir.

Otro apunte para un trabajo a futuro seria agregarle otros dispositivos de hardware que permitan al proyecto ser controlados remotamente.