



**UNIVERSIDAD ESTATAL DE MILAGRO
FACULTAD CIENCIAS E INGENIERÍA**

**INFORME DE PROYECTO INTEGRADOR
PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS COMPUTACIONALES**

TEMA: Diseño de un prototipo brazo robótico con sistema de visión por computador para la clasificación de empaques por colores

Autores:

Sr. Andrés Manuel Puca Flores

Sr. Víctor Emilio Rosado Rendón

Tutor:

Mgr. JHONNY DARWIN ORTIZ MATA

Milagro, Febrero 2020

ECUADOR

DERECHOS DE AUTOR

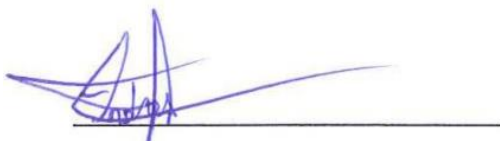
Ingeniero.
Fabricio Guevara Viejó, PhD.
RECTOR
Universidad Estatal de Milagro
Presente.

Yo, ANDRÉS MANUEL PUCA FLORES, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de integración curricular, modalidad presencial, mediante el presente documento, libre y voluntariamente procedo a hacer entrega de la Cesión de Derecho del Autor, como requisito previo para la obtención de mi Título de Grado, como aporte a la Línea de Investigación Tecnologías de la información y de la comunicación, de conformidad con el Art. 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, concedo a favor de la Universidad Estatal de Milagro una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservo a mi favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo a la Universidad Estatal de Milagro para que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

El autor declara que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

Milagro, 20 de Febrero del 2020



Puca Flores Andrés Manuel
Autor 1
CI: 0302438486

DERECHOS DE AUTOR

Ingeniero.

Fabricio Guevara Viejó, PhD.

RECTOR

Universidad Estatal de Milagro

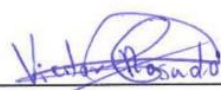
Presente.

Yo, VÍCTOR EMILIO ROSADO RENDÓN, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de integración curricular, modalidad presencial, mediante el presente documento, libre y voluntariamente procedo a hacer entrega de la Cesión de Derecho del Autor, como requisito previo para la obtención de mi Título de Grado, como aporte a la Línea de Investigación Tecnologías de la información y de la comunicación, de conformidad con el Art. 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, concedo a favor de la Universidad Estatal de Milagro una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservo a mi favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo a la Universidad Estatal de Milagro para que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

El autor declara que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

Milagro, 20 de Febrero del 2020



Rosado Rendón Víctor Emilio

Autor 2

CI: 0942054370

APROBACIÓN DEL TUTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Yo, JHONNY DARWIN ORTIZ MATA En mi calidad de tutor del trabajo de integración curricular, elaborado por los estudiantes ANDRÉS MANUEL PUCA FLORES y VÍCTOR EMILIO ROSADO RENDÓN, cuyo título es DISEÑO DE UN PROTOTIPO CON UN SISTEMA DE VISIÓN POR COMPUTADORA PARA LA CLASIFICACIÓN DE EMPAQUES POR COLORES, que aporta a la Línea de Investigación Tecnologías de la Información y de la Comunicación previo a la obtención del Título de Grado INGENIERO EN SISTEMAS COMPUTACIONALES; considero que el mismo reúne los requisitos y méritos necesarios en el campo metodológico y epistemológico, para ser sometido a la evaluación por parte del tribunal calificador que se designe, por lo que lo APRUEBO, a fin de que el trabajo sea habilitado para continuar con el proceso previa titulación de la alternativa de Trabajo de Integración Curricular de la Universidad Estatal de Milagro.

Milagro, 20 de Febrero del 2020


ORTIZ MATA, JHONNY DARWIN

Tutor

C.I: 0927159111

APROBACIÓN DEL TRIBUNAL CALIFICADOR

El tribunal calificador constituido por:

Mgtr. ORTIZ MATA JHONNY DARWIN

Mgtr. AREVALO GAMBOA LISSETT MARGARITA

Mgtr. MENDOZA CABRERA DENIS DARIO

Luego de realizar la revisión del Trabajo de Integración Curricular, previo a la obtención del título (o grado académico) de INGENIERO EN SISTEMAS COMPUTACIONALES Presentado por el estudiante PUCA FLORES ANDRÉS MANUEL

Con el tema de trabajo de Titulación: **DISEÑO DE UN PROTOTIPO BRAZO ROBÓTICO CON SISTEMA DE VISIÓN POR COMPUTADOR PARA LA CLASIFICACIÓN DE EMPAQUES POR COLORES.**

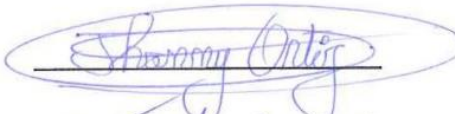
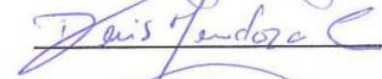

Otorga al presente Trabajo de Integración Curricular, las siguientes calificaciones:

Trabajo Curricular	Integración	[60]
Defensa oral		[40]
Total		[100]

Emite el siguiente veredicto: (aprobado/reprobado) APROBADO

Fecha: 20 de febrero de 2020

Para constancia de lo actuado firman:

	Nombres y Apellidos	Firma
Presidente	ORTIZ MATA JHONNY DARWIN	
Secretario /a	MENDOZA CABRERA DENIS DARIO	
Integrante	AREVALO GAMBOA LISSETT MARGARITA	

APROBACIÓN DEL TRIBUNAL CALIFICADOR

El tribunal calificador constituido por:

Mgtr. ORTIZ MATA JHONNY DARWIN

Mgtr. AREVALO GAMBOA LISSETT MARGARITA

Mgtr. MENDOZA CABRERA DENIS DARIO

Luego de realizar la revisión del Trabajo de Integración Curricular, previo a la obtención del título (o grado académico) de INGENIERO EN SISTEMAS COMPUTACIONALES presentado por el estudiante ROSASO RENDÓN VÍCTOR EMILIO

Con el tema de trabajo de Titulación: **DISEÑO DE UN PROTOTIPO BRAZO ROBÓTICO CON SISTEMA DE VISIÓN POR COMPUTADOR PARA LA CLASIFICACIÓN DE EMPAQUES POR COLORES..**

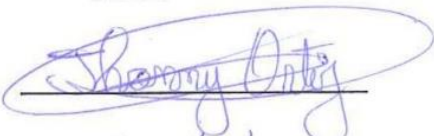
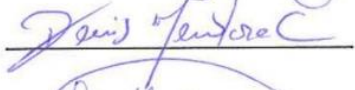

Otorga al presente Proyecto Integrador, las siguientes calificaciones:

Trabajo de Integración Curricular	[60]
Defensa oral	[40]
Total	[100]

Emite el siguiente veredicto: (aprobado/reprobado) APROBADO

Fecha: 20 de febrero de 2020

Para constancia de lo actuado firman:

	Nombres y Apellidos	Firma
Presidente	ORTIZ MATA JHONNY DARWIN	
Secretario /a	MENDOZA CABRERA DENIS DARIO	
Integrante	AREVALO GAMBOA LISSETT MARGARITA	

DEDICATORIA

El presente trabajo de titulación quiero dedicárselo como gesto sencillo de agradecimiento en primer lugar a Dios quien me ha permitido llegar a este punto de mi vida, a mis padres Carlos y Lucia quienes siempre me dieron su apoyo incondicional sin ellos no hubiera podido llegar hasta aquí.

A mis hermanos Anthony y Carlos quienes con sus consejos y apoyo en los momentos que los necesite siempre estuvieron presentes, a mi novia Valeria por su amor permanente su cariño y comprensión, a mis amigos quienes caminaron conmigo en las buenas y en las malas

Andrés Manuel Puca Flores

DEDICATORIA

Dedico este trabajo de titulación a Dios por darme la vida y la oportunidad de poder formarme como un profesional, por brindarme sabiduría y fortaleza, a mis padres Carlos Rosado y Narcisa Rendón los cuales me han ayudado con sus oraciones siempre, enseñándome a ser perseverante en la vida para lograr cada objetivo, a mi novia Ing. Katuska Rondan Pacheco quien ha sido un modelo a seguir en la formación profesional y laboral, por su apoyo incondicional en momentos difíciles, a mis hermanos Ángela Rosado, Carlos Rosado Rendón y Cuñada quienes han formado parte de mi guía en este camino, a mis sobrinos Alba, Sebastián, Santiago, Belén y Emanuel quienes me han brindado su amor y alegría, siendo parte fundamental en este proceso.

Víctor Emilio Rosado Rendón

AGRADECIMIENTO

Un agradecimiento especial al Msc. Jhonny Ortiz por su constante apoyo para que este trabajo fuera realizado impartiendo sus conocimientos en el tema de la electrónica punto clave en la elaboración de nuestro proyecto

A mi compañero Víctor Rosado por apoyarme en la elaboración de la tesis aportando en los momentos que no podía estar pendiente de la tesis.

A mis amigos más cercanos quienes en los momentos difíciles me brindaron un apoyo incondicional.

Andrés Manuel Puca Flores

AGRADECIMIENTO

A: Agradezco a Dios por darme la vida, permitir disfrutar con mi familia, brindarme sabiduría para poder cumplir cada una de las metas propuestas en mi vida.

A mis padres quienes han sido el pilar fundamental en mi vida, quienes con sus consejos y apoyo incondicional aportaron de manera fructífera en esta etapa para ser un profesional.

A mi novia Ing. Katuska Rondan Pacheco por brindarme su amor, su ayuda en este proceso aportando ideas positivas y ser guía en el ámbito educativo, motivándome así a seguir firme ante mi objetivo para poder ser un profesional.

A mis hermanos Ángela Rosado y Carlos Rosado por formar parte desde el inicio de la base primordial que logre obtener en mi formación como estudiante hasta el punto en el cual me encuentro ahora.

A mis sobrinos quienes me brindan alegría, me impulsan a seguir adelante para poder ser un modelo a seguir en su formación educativa, aquellos familiares y amigos los cuales con sus consejos me han ayudado a dar prioridad a los estudios.

A mí tutor Ing. Jhonny Ortiz Mata quien nos colaboró aportando con su conocimiento, dedicación y ayuda para desarrollar este proyecto.

A mi compañero de tesis Andrés Puca quien aportó con sus conocimientos, paciencia para poder realizar de forma exitosa este proceso.

A la Universidad Estatal De Milagro por brindarme la oportunidad de ejercerme como un profesional y a todo el personal que trabaja arduamente para sacar adelante a esta prestigiosa universidad

Víctor Emilio Rosado Rendón

ÍNDICE GENERAL

DERECHOS DE AUTOR.....	ii
APROBACIÓN DEL TUTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR.....	ii
DERECHOS DE AUTOR.....	iii
APROBACIÓN DEL TUTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR.....	iv
APROBACIÓN DEL TRIBUNAL CALIFICADOR	v
.....	v
APROBACIÓN DEL TRIBUNAL CALIFICADOR	vi
DEDICATORIA	vii
DEDICATORIA	viii
AGRADECIMIENTO.....	ix
AGRADECIMIENTO.....	x
ÍNDICE GENERAL.....	xi
ÍNDICE DE FIGURAS.....	xiv
ÍNDICE DE GRÁFICOS	xvii
RESUMEN.....	xviii
Job Title Curriculum Integration: DESIGN OF A ROBOTIC ARM PROTOTYPE WITH COMPUTER VISION SYSTEM FOR CLASSIFICATION OF COLOR PACKAGING.....	xx
ABSTRACT.....	xx
CAPÍTULO 1	1
1. INTRODUCCIÓN	1
1.1. Planteamiento del problema	1
1.2. Objetivos	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Especificos.....	2
1.3. Justificación.....	2
1.4. Marco Teórico.....	3
1.4.3 Comportamiento de la visión humana.....	7
1.4.3.1 Espectro visible de luz.....	8
1.4.3.2 Tipos de modelos de Color.....	9
1.4.3.3 Representación de los colores	11
1.4.3.4 Teoría de los espacios de colores	11
1.4.4 Visión por computadora versus visión humana.....	13
1.4.5 Robótica. - Definición.....	14
1.4.5.1 Brazo Robótico.....	17

1.4.5.2 KUKA YouBot.....	17
1.4.5.3 MeArm.....	18
1.4.5.4 ABB IRBI20	19
1.4.5.5 Kinova Jaco Re	19
CAPÍTULO 2.....	20
2. METODOLOGÍA.....	20
Fase 1: Planificación de Análisis inicial	20
Fase 2: Diseño	21
Fase 3: Programación	21
Fase 4: Pruebas.....	22
CAPÍTULO 3.....	24
3. RESULTADOS (ANÁLISIS O PROPUESTA)	24
3.1 Desarrollo.....	24
3.1.1 Fase de Análisis	24
Software a utilizar	24
LabVIEW.....	24
NI-IMAQ	26
Modulo NI-VISA.....	27
Tolkit SQL	27
SQL Server	28
Sentencias SQL	29
Arduino IDE.....	29
Hardware a Utilizar	30
Ordenador.....	30
Cámara USB.....	31
Placa Arduino Uno	32
Módulo Shield PCA9685 PWM 12-bit Driver I2C	34
Componentes Necesarios para el Prototipo de Visión por Computadora	34
Brazo Robótico.....	34
Banda Transportadora	36
3.1.2 Fase de Diseño	38
Diseño de Interfaz de Usuario en LabVIEW.....	38
Diseño del Menú Principal.....	39
Diseño de Interfaz del Sistema de Visión por Computadora.....	40
Diseño de Interfaz control Manual para el Brazo Robótico	40

Diseño de la Tabla UML de la Base de Datos	41
3.1.3 Programación	42
Programación en LabVIEW	42
Vis para del Reconocimiento de Color	42
Vis para la Conexión con la Cámara USB	42
Vis para el reconocimiento de Color en LabVIEW	44
Vis para el Enlace entre Labview y Arduino	45
Programación para el Enlace entre LabVIEW y la Base de Datos	46
Programación para registro en la base de datos de objetos detectados por color	46
Programación en SQL Server.....	47
Código DLL para la Creación de Base de Dato y Tabla a Utilizar.....	47
3.1.4 Pruebas	48
Pruebas individuales de sensibilidad de los colores bajo diferentes condiciones	50
Pruebas de Lectura de Colores en Tiempo Real	54
CONCLUSIONES	55
RECOMENDACIONES	56
REFERENCIAS BIBLIOGRÁFICAS	57
ANEXOS.....	59
ANEXOS 1: Tabla de Costo de Implementación del Sistema	59
ANEXOS 2: Figura de las Comunicaciones del Sistema con el Brazo y Banda.....	60
ANEXOS 3: Conexiones del Brazo Robótico a la Placa Arduino y Shiel.....	61
ANEXOS 4: Manual de Usuario para uso del Sistema	62
ANEXOS 5: Código Arduino para Comunicación con el Brazo Robótico y Banda Transportadora.....	65

ÍNDICE DE FIGURAS

Figura 1: Que Es la Visión Por Computadora	4
Figura 2: Áreas De La Visión por Computadora.....	5
Figura 3: Espectro Visual.....	7
Figura 4: Sensibilidad Espectral Del Color	8
Figura 5: Rango De Espectro De Luz	9
Figura 6: Circulo Cromático del Modelo RYB.....	10
Figura 7: Modelo de Color RGB.....	10
Figura 8: Circulo Del Modelo CMY.....	11
Figura 9: Espacio De Color RGB	12
Figura 10: Brazo Robótico KUKA Youbot.....	18
Figura 11: Brazo Robótico MeArm.....	19
Figura 12: Brazo Robótico ABB IRB1120.....	19
Figura 14: Icono de LabVIEW	25
Figura 15: Menú de complementos de Vision and Motion.....	26
Figura 16: Menú NI-IMAQ.....	27
Figura 17: Componentes del Módulo NI-VISA	27
Figura 18: Toolkit SQL.....	28
Figura 19: Logotipo de SQL Server.....	28
Figura 20: Logo de Arduino.....	30
Figura 21: Interfaz de la IDE de Arduino	30
Figura 22: Ordenador Toshiba	31
Figura 23: Cámara USB.....	32
Figura 24: Placa Arduino Uno	34
Figura 25: Modulo Shield PCA9685.....	34
Figura 26: Componentes del Brazo Robótico.....	36
Figura 27: Brazo Robótico LeArm.....	36
Figura 28: Componentes de Banda Transportadora	37
Figura 29: Diseño 3D de la Banda Transportadora	38
Figura 30: Diseño final de la Interfaz del Menú Principal.....	39
Figura 31: Diseño final de la Interfaz del Sistema de Visión por Computadora para la Detección de Colores.....	40
Figura 32: Diseño de Interfaz de Brazo Robótico.....	41
Figura 33: Diseño de Tabla UML Colores	42
Figura 34: Bloque para Conectar a la Cámara USB.....	44
Figura 35: Programación para Identificar Colores	45
Figura 36: Programa de enlace entre LabVIEW Y Arduino	45
Figura 37: Programación de Enlace Arduino LabVIEW.....	46
Figura 38: Contador de Total Objetos de Colores Detectados	46
Figura 39: Registro a la Base de Datos.....	47
Figura 40: Código Transact para la Creación de Base de Datos y Tabla	48
Figura 41: Diagrama de Flujo del Funcionamiento del Sistema	50
Figura 42: Comunicación del Sistema con el Brazo y la Banda	61
Figura 43: Conexión de los Servos del Brazo Robótico	61

Figura 44: Conexión Física del Brazo Robótica	62
Figura 45: Brazo Robótico Conectado a la Placa Shield y Arduino	62
Figura 46: Interfaz de Usuario.....	63
Figura 47: Configuración de puestos para la Conexión.....	64
Figura 48: Campos de Controles para Encendido y Apagado del Sistema	64
Figura 49: Determinando el Área que detectara los colores.....	65
Figura 50: Panel Indicador de Colores Detectados	65
Figura 51: Acompañamiento académico	72

ÍNDICE DE TABLAS

Tabla 1: Procesos de Visión por Computadora	5
Tabla 2: Tipos de espacios de color.....	12
Tabla 3: Tipos De Espacios y Características	12
Tabla 4: Ventajas y desventajas de visión por computadora y visión humana	14
Tabla 5: Campos de estudio de la Robótica.....	15
Tabla 6: Estructura y componentes de los robots	15
Tabla 7: Expansión Industrial de los Robots	17
Tabla 8: Características Brazo Robótico KUKA YouBot.....	17
Tabla 9: Características Brazo Robótico MeArm.....	18
Tabla 10: Características Brazo Robótico ABB IRBI20	19
Tabla 11: Características Brazo Robótico Kinova Jaco Re	20
Tabla 12: Flujograma de la metodología aplicada.....	23
Tabla 13: Módulos Requeridos para el sistema	25
Tabla 14: Sentencias Transact de SQL.....	29
Tabla 15: Características Principales del Ordenar Utilizado.....	31
Tabla 16: Características Principales de la Cámara USB	32
Tabla 17: Especificaciones Técnicas de la Placa Arduino.....	33
Tabla 18: Características Técnicas de la Shield.....	34
Tabla 19: Especificaciones Técnicas del Brazo Robótico	35
Tabla 20: Características del Brazo Robótico	35
Tabla 21: Tipos de Bandas Transportadoras	36
Tabla 22: Características Técnicas del Motor de Paso	37
Tabla 23: Registro de las Pruebas de Sensibilidad Color Rojo	50
Tabla 24: Registro de Prueba de Sensibilidad Color Verde	51
Tabla 25: Registro de Prueba de Sensibilidad Color Azul	52
Tabla 26: Registro de Prueba de Sensibilidad Color Blanco	53
Tabla 27: Pruebas de Efectividad de la Detección de Colores	54
Tabla 28: Costo de Implementación del Sistema en Industrias	59

ÍNDICE DE GRÁFICOS

Gráfico 1:Prueba de Sensibilidad del Color Rojo	50
Gráfico 2: Prueba de sensibilidad color verde	51
Gráfico 3:Prueba de Sensibilidad del Color Azul.....	52
Gráfico 4:Prueba de Sensibilidad Color Blanco.....	53
Gráfico 5:Resultado de la Prueba de Efectividad	54

Título de Trabajo Integración Curricular: DISEÑO DE UN PROTOTIPO BRAZO ROBÓTICO CON SISTEMA DE VISIÓN POR COMPUTADOR PARA LA CLASIFICACIÓN DE EMPAQUES POR COLORES

RESUMEN

La visión por computadora es un campo el cual trata de emular la visión que tiene un ser humano realizando de una forma casi igual el procesamiento de una imagen en tiempo real logrando así identificar los colores, formas y tamaños el presente trabajo se focalizo en crear un algoritmo capaz de identificar colores, al identificarse mediante la aplicación del prototipo de brazo robótico este realizara una determinada acción la cual estará programada en Arduino para el control del brazo y a su vez la programación está construido en LabVIEW un poderoso Software para programación gráfica que permite integrar hardware de una manera mucho más fácil, motivo por el cual es usado en muchas empresas de diversos campos, para el mejor control del algoritmo se realizó una interfaz sencilla para su uso donde por medio de esta se le dará instrucciones de qué color desea que el sistema detecte y la acción a realizar, tales como donde colocar cada objeto diferenciado por su color y colocarlo en un sitio específico, también se puede visualizar la cantidad de objetos por colores que el sistema ha detectado, todo esto en tiempo real para no perder los datos que se registraron se cuenta con una base de datos en donde se registra cada uno de los procesos que realizo con fecha y hora en la que esta fue realizada.

Para la demostración del funcionamiento del sistema se requirió de una estructura de brazo robótico el cual está montado de una manera en la que pueda realizar los movimientos sin ningún problema, además se contó con una banda transportadora la cual tiene mucha importancia a la hora de realizar las pruebas de funcionamiento del sistema ya que esta

transporta los empaques de extremo a extremo al detectar un color la banda se detiene dando paso a la activación del brazo robótico.

Para la elaboración de cada uno de las partes del sistema se hizo uso de una metodología ágil de desarrollo de sistemas, determinando al modelo SCRUM como la base para el desarrollo total que se le dio al sistema, este modelo permitió separar en fases cada una de las tareas que se realizaron quedando de la siguiente manera:

- 1- Fase de Análisis
- 2- Fase de Diseño
- 3- Fase de Programación
- 4- Fase de Pruebas

Estas 4 fases mencionadas tienen el objetivo de mantener todo en orden para el desarrollo del sistema y la ejecución del producto final, en la Fase de análisis se determinaron los elementos a ser analizados como lo son la parte de hardware y software que se utilizó además de herramientas o componentes que sirvieron para el correcto funcionamiento del sistema, a su vez en la fase de Diseño se realizaron la parte visual del sistema tales como las interfaces de usuarios, diagrama de Clases para la tabla de base datos, en la fase de programación se realizó la codificación en el software de LabVIEW, Arduino y SQL Server, al finalizar se entregó pruebas del funcionamiento del software mediante tabulaciones con gráficas con los resultados de cada una de las pruebas demostrando la efectividad del sistema y como anexos se colocaron los componentes secundarios que sirvieron para el prototipo como lo es el brazo robótico junto a la banda transportadora.

PALABRAS CLAVE: PROCESAMIENTO, COLORES, LABVIEW, ARDUINO, ALGORITMO, VISIÓN

Job Title Curriculum Integration: DESIGN OF A ROBOTIC ARM PROTOTYPE WITH COMPUTER VISION SYSTEM FOR CLASSIFICATION OF COLOR PACKAGING

ABSTRACT

Computer vision is a field which tries to emulate the vision that a human being has, realizing in an almost equal way the processing of an image in real time, thus managing to identify the colors, shapes and sizes the present work focused on creating a algorithm capable of identifying colors, when identified through the application of the prototype of the robotic arm, this will perform a certain action which will be programmed in Arduino for the control of the arm and in turn the programming is built in LabVIEW a powerful Software for graphic programming that allows to integrate hardware in a much easier way, which is why it is used in many companies of various fields, for the best control of the algorithm a simple interface was made for its use where through this it will be given instructions of what color you want the system detect and the action to be performed, such as where to place each object differentiated by its color and c To place it in a specific site, you can also visualize the number of objects by colors that the system has detected, all this in real time so as not to lose the data that was recorded there is a database where each of the processes that I carry out with date and time in which it was carried out.

For the demonstration of the operation of the system, a robotic arm structure was required which is mounted in a way in which you can perform the movements without any problem, in addition there was a conveyor belt which is very important at the time of performing

The system's functional tests since it carries the gaskets from end to end when a color is detected, the band stops giving way to the activation of the robotic arm.

For the elaboration of each of the parts of the system, an agile methodology of system development was used, determining the SCRUM model as the basis for the total development that was given to the system, this model allowed to separate each one of the phases the tasks that were carried out being as follows:

- 1- Analysis Phase
- 2- Design Phase
- 3- Programming Phase
- 4- Testing Phase

These 4 phases mentioned have the objective of keeping everything in order for the development of the system and the execution of the final product, in the Analysis Phase the elements to be analyzed were determined as they are the part of hardware and software that was used in addition to tools or components that served for the correct functioning of the system, in turn in the Design phase the visual part of the system was made such as the user interfaces, Class diagram for the database table, in the programming phase performed the coding in the software of LabVIEW, Arduino and SQL Server, at the end of the day, tests of the software were delivered through tabulations with graphs with the results of each of the tests demonstrating the effectiveness of the system and as annexes the secondary components that were placed they served for the prototype as is the robotic arm next to the conveyor belt.

KEY WORDS: Processing, Colors, LabVIEW, Arduino, Algorithm, Vision

CAPÍTULO 1

1. INTRODUCCIÓN

En Ecuador existen varias empresas que se dedican a la fabricación de diversos artículos los cuales en la mayoría de los casos requieren de clasificación estas empresas al manejar una producción diversa en sus productos elaborados enfrentan varias situaciones las cuales suponen gastos para la empresa por la gran cantidad de personal requerido para las labores uno de estos casos es la clasificación, como se sabe este proceso es uno de los más importante ya que es aquí donde se separan los productos fabricados, separándolos por colores, formas y tamaños es por ello que la implementación de un sistema que realice este proceso de manera automatizada que no dependa de mucho personal para su control beneficiaria en muchos aspectos a la empresa comenzando por conseguir una mayor producción en menor tiempo, reducir gastos en sueldos y minimizar riesgos de errores nuestra propuesta se centraliza en diseñar un sistema que sea capaz de identificar en tiempo real colores permitiendo con esto su aplicación en sistemas mecánicos los cuales realicen tareas de clasificación como lo pueden ser brazos robóticos y bandas transportadoras estos implementos ayudan en mucho al crecimiento de las empresas que ya cuentan con estos mecanismos, es por ello que nuestro sistema cuenta con la capacidad de acoplarse a los mecanismos mencionados con la diferencia de que se podrá controlar de manera inteligente las operaciones de clasificación.

1.1. Planteamiento del problema

Actualmente la clasificación de empaques se la realiza de forma manual requiriendo de varias personas para dicho proceso, al mayor número de trabajadores mayor es el costo de producción que tiene la empresa ya que representa un valor significativo por coste de sueldos además de contar con una producción limitada por personas que laboran en dicha

área, los riesgos que pueden presentarse son altos debido que el personal puede sufrir algún percance causando que el proceso de clasificación no cumpla las metas establecidas.

1.2. Objetivos

1.2.1. Objetivo General

Diseñar un sistema de visión por computadora encargado de la detección de colores para la clasificación de empaques

1.2.2. Objetivos Específicos

- Diseñar un sistema de visión por computadora capaz de detectar en tiempo real colores.
- Realizar la programación del sistema en Labview y Arduino
- Crear un enlace con la base de datos SQL Server que permita generar reportes de total de producción.
- Enlazar los componentes de hardware y realizar pruebas de funcionamiento.

1.3. Justificación

Según (Pérez González, 2016) en su trabajo sobre visión por computadora muestra lo importante que es lograr la automatización en las empresas ya que estas al incluir tecnologías que agilicen los procesos que se manejan de forma manual permite que la empresa incremente su producción y a su vez reduzca los costos de operación, actualmente las tecnologías se encuentran en un gran crecimiento es por ello que hacer uso de las mismas se vuelve más accesible, la visión por computadora perteneciente a la rama de inteligencia artificial ha venido siendo estudiada para ser utilizada en muchos campos en nuestro caso el hacer uso de la visión por computadora nos permitirá crear mediante lenguajes de programación un sistema capaz de identificar de forma autónoma los colores y formas consiguiendo con esto darle instrucciones que saldrán de las necesidades de cada empresa, la finalidad de nuestro proyecto será el de aplicar un sistema para empresas en las

que realicen específicamente procesos de clasificación como es bien sabido estas empresas al contener mucho material elaborado que a su vez suelen ser de diferentes colores y tamaño se debe hacer uso de tecnologías como estas que agilicen la clasificación de cada empaque o producto elaborado, con la implementación de nuestro proyecto se conseguirá que se automatice en esta área dichos procesos resolviendo con esto la problemática de los tiempos que se demoran en realizar la clasificación de forma manual.

(Ingenier, Por, & Freddy, 2015) en su trabajo nos entrega una manera sencilla de trabajar con el reconocimiento de formas, enfocándose en las redes neuronales un campo muy estudiado por sus diversos beneficios que entrega, con dicho trabajo nos da un enfoque muy claro de cómo se puede realizar un trabajo dedicado en el área que vamos a aplicar nuestra investigación es por ello que los resultados para la detección de formas y colores no serán un obstáculo para el desarrollo de nuestro sistema.

1.4. Marco Teórico

1.4.1 Visión por computadora (concepto específico)

Según la publicación de (Infinita, Agropecuarias, & Ambientales, s. f.) denomina a la visión por computadora como un campo de investigación que trata de emular lo que el ojo humano hace es decir permite el procesamiento de una imagen captada en el caso del ser humano por medio de la retina ocular donde se logra la percepción de los colores y formas que tienen las figuras la visión por computadora sigue este principio del ojo humano al dividir los colores como si de un haz de luz pasara por un prisma para luego unirlos dando color a lo que vemos, en la visión por computadora este procesamiento comúnmente se lo logra usando equipos de hardware capaz de captar las luces que se encuentra para luego por medio de la asistencia de un computador el cual tenga consigo un software de procesamiento de imágenes, los continuos avances en la creación de algoritmos y aplicaciones en este campo hacen que esta disciplina sea una de las tecnologías con mayor

crecimiento como se lo ha visto en las últimas décadas, los grandes cambios que se ha visto en la visión por computadora, esto asido posible por el crecimiento de material visual que se maneja, tanto en imágenes como en videos, la capacidad que tienen los equipos electrónicos para procesar dicha información se lo puede encontrar en diversos dispositivos, además de contar con herramientas y lenguajes de programación que están dedicado netamente para este fin.

Existen varios sectores donde hoy en día la aplicación de la visión por computadora en vista en mayor porcentaje como principales podemos citar en la medicina, biología, seguridad, robótica, industria, agricultura, meteorología y la cartografía.

Pese a que el uso de la visión por computadora está bastante usado en las áreas mencionadas es muy complejo su aplicación requiriendo muchas veces de inversiones para investigaciones que toman un tiempo considerable para su puesta en práctica ya que estas tecnologías deben ser aplicadas con un nivel de robustez y gran velocidad de respuesta para hacer frente a la solución de los problemas que están siendo resueltos con dicha tecnolog

¿De qué se trata Visión por Computadora?



Figura 1: Que Es la Visión Por Computadora

En la actualidad contamos con propuestas interesantes que ofrecen buenos resultados pero a su vez presentan carencias en varios factores que a la hora de decidirse por un sistema de

estos llegue a ser determinante para no optar por uno una de las mayores razones es el factor económico que en muchos de los casos su adquisición es muy alto lo que desanima a las empresas en adquirir dichos sistemas, es por ello que este campo se encuentra en un punto muy amplio para ser explotado con la mira principal de conseguir resultados de calidad pero con costos reducidos en la aplicación de este tipo de sistemas.

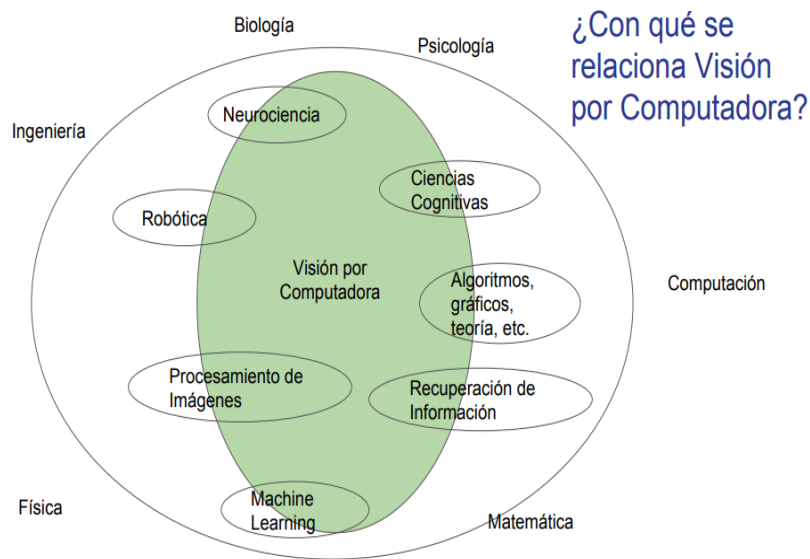


Figura 2: Áreas De La Visión por Computadora

1.4.2 Como nace la Visión por computadora

Según (Pajares & de la Cruz, 2007) la visión por computadora nació como intento de automatizar procesos con actividad visual haciendo uso de tratamiento de imágenes esto se lo lograría integrando a las maquinas existentes con un sistema de visión inteligente para ello se debería también hacer uso de varios factores tales como son las matemáticas, física, electrónica, robótica y computación todo esto en un solo dan como resultado un sistema óptimo de visión por computadora el autor además describió los pasos que se deben seguir para conseguir el correcto funcionamiento de la visión por computadora.

Tabla 1: Procesos de Visión por Computadora

Procesos	Nivel de visión	Entrada	Salida	Área
----------	-----------------	---------	--------	------

1. Captura	Bajo	Imagen	Imagen	Procesamiento de imágenes
2. Pre-procesamiento				
3. Segmentación	Medio	Imagen	Grupos de	Análisis de
4. Descripción		Objetos o regiones	pixeles en	imágenes
5. Reconocimiento (clasificación)		Información Cuantitativa	bruto(objetos o regiones) información cuantitativa de los objetos o regiones clasificados en categorías	
6. Interpretación	Alto	Objetos Clasificados en categoría	Comprensión de la escena	Visión por computadora

Fuente: González y Woods (2010)

En el proceso de Captura es donde se implica un elemento físico para obtener la imagen estos elementos pueden ser: cámara de video, cámara digital, escáner, telescopio, satélite, etc. Para el Pre-procesamiento se aplican técnicas de procesamiento de imágenes para la reducción de ruido, mejorar el contraste, realizar los detalles y hasta la restauración de la imagen. En la Segmentación ya se aplican procesos para dividir la imagen en objetos o regiones tomados de nuestro interés, esta es una de las etapas muy importantes que permiten lograr el éxito o fracaso de la aplicación.

En la descripción se obtendrán las características más importantes que permita diferenciar un objeto de otro, estas se las podrá sacar de niveles de intensidad del color tanto máximo como mínimo y patrones de texturas.

Para el Reconocimiento es el proceso en el que se clasificaran en categorías utilizando métodos estadísticos avanzados, estructurales, redes neuronales, algoritmos genéticos, etc.

La interpretación como punto final es la que trata de emular el ojo humano utilizando en ellas técnicas cognitivas para la toma de decisiones cabe recalcar que para llegar a este proceso se tiene que pasar por todos los mencionados anteriormente dependiendo de la

complejidad de lo analizado se podrá seguir una secuencia directa de lo presentado a su vez de no ser necesario se podrá omitir ciertos procesos.

1.4.3 Comportamiento de la visión humana

La visión humana presenta una complejidad muy grande por cómo es capaz de permitirnos ver el entorno por medio de nuestros ojos, para poder entendimiento de cómo se lleva aquello (Sucar,2018) determina en forma general que la visión humana es capaz de detectar ondas que se encuentran en rangos de percepción de colores como lo son los espectros electromagnéticos que tienen un rango partiendo desde los 400 nm en donde nos permite visualizar un color violeta y llegando a un tope de los 700 nm el cual se encuentra establecido en ese rango el color rojo, estos rangos se los conoce como continuo gradiente es decir una tabla de colores conocidos por cada uno de nosotros generando con esto un espectro visual que nos permitirá la detección de estos colores de cada cosa que observamos.

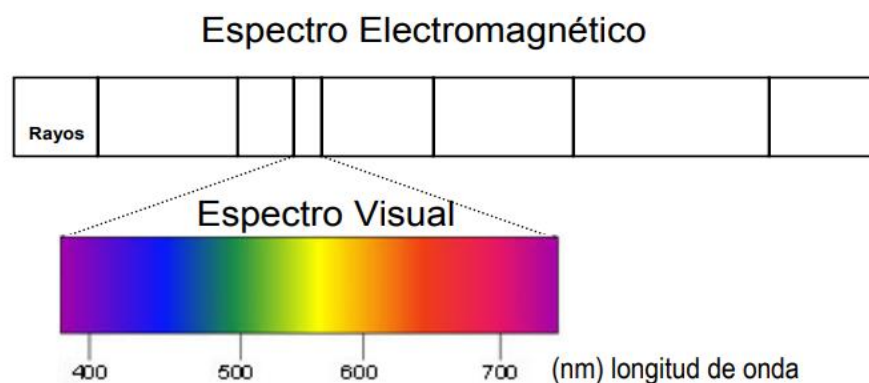


Figura 3: Espectro Visual

Al hablar de visión siempre se nos viene a la mente percepciones que son importantes tales como lo es la forma de algo, el color las distancias y muchos más aspectos los cuales solo

pueden ser vistos cuando se genera un proceso de ondas luminosas llegan a la retina, si estas ondas luminosas no se encuentran en una escala determina lo que provocaría sería es no producir impresión visual por algo, para conseguir visualizar los colores de manera clara se debe recibir correctamente una longitud de onda que va directamente lo que percibe la retina al momento de ver es una imagen del entorno pero invertida la nitidez de lo que vemos en nuestro entorno la retina debe encontrarse de forma sana capaz de captar la menor cantidad de espectro luminoso.

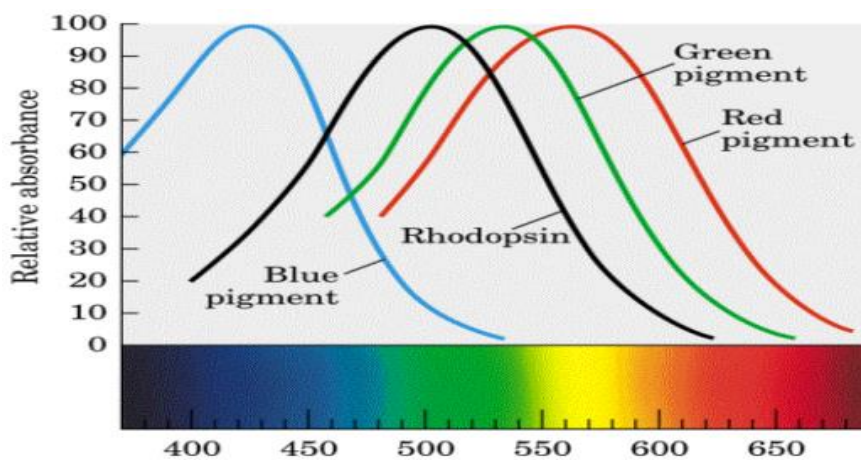


Figura 4: Sensibilidad Espectral Del Color

1.4.3.1 Espectro visible de luz

Para (Máster, 2014) en su trabajo describe un amplio concepto sobre el espectro visible de luz que percibe el ojo humano en el cual se encuentra determinado por varios factores uno de los más conocidos es que puede ser producido por una longitud de onda permitiendo con esto visualizar los colores de una manera clara, cada color está constituido por un porcentaje en una cadena de rangos los cuales al entrar en dichos rangos pueden ser diferenciados como un color en concreto como se observa en la figura 5 el rango es continuo por lo cual no existe cantidades entre cada color.

Espectro visible de luz	
violeta	380–450 nm
azul	450–495 nm
verde	495–570 nm
amarillo	570–590 nm
rojo	590–650 nm

Figura 5: Rango De Espectro De Luz

1.4.3.2 Tipos de modelos de Color

Existen varias formas de tratar con el espectro visible de luz la cual permite visualizar los colores de forma en la que la apreciamos, algunas de las grandes teorías que se han propuesto se centran en dividir los colores que apreciamos en cantidades diferentes tomando como ejes los colores más relevantes que cada uno propuso, Goethe estableció un círculo con colores simétricos el cual está basado por el propuesto por Newton en donde indicaba que tal círculo se dividía en siete ángulos con colores desiguales, lo que diferencio de la teoría de Newton fue que Goethe si estableció una simetría dando a lugar al nacimiento de diferentes colores los cuales tomaban el lugar de otro color de dicha tabla circular.

Modelo RYB

Según (Rivera-mujica, Huamaní-navarrete, & Palma, 2018) Uno de los modelos también referencial en la actualidad es el de RYB, este modelo singular fija a 3 colores como los primarios de toda la gama de colores dejando como resultado al color rojo, amarillo y azul como colores primarios y a los demás con un término de colores puros los cuales se pueden generar al ir mezclando los colores primarios ya mencionados, este modelo es muy utilizado en las artes donde se utilizan pigmentos los cuales mediante la guía de este modelo pueden experimentar con los colores e ir creando nuevas gamas de colores.



Figura 6: Circulo Cromático del Modelo RYB

Modelo RGB

El modelo de Color RGB es uno de los más conocidos en la actualidad por sus múltiples aplicaciones que tiene en el área de la tecnología, la mezcla de 3 colores luz como el autor los define dan origen a los demás colores cuando se tiene ausencia de colores luz el color que se puede diferenciar es el color negro, mediante esta explicación es comprensible entender cómo es que este modelo es muy usado en los monitores, televisores, proyectores, etc.

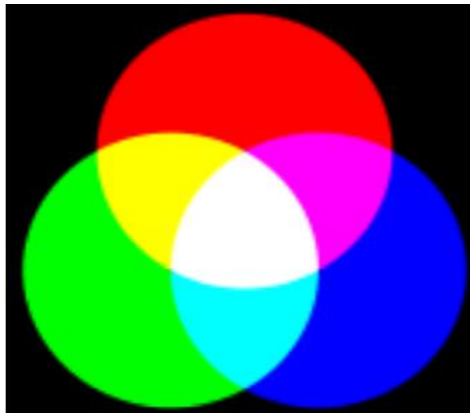


Figura 7: Modelo de Color RGB

Modelo CMY

El modelo CMY se caracteriza por crear el color negro de la mezcla de todos los colores y determina que el color blanco es la representación de la ausencia de cualquier otro color, también a dicho modelo se lo puede encontrar como modelo de color sustractivo, en este modelo se establece como colores primarios al cian, magenta y amarillo, al mezclar estos

colores logramos conseguir un color negro algo turbio, este tipo de modelo es muy utilizado para las mezclas de pigmentos, hay que tener en cuenta que los conocidos primarios ficticios como lo serían en este caso se los puede lograr conseguir de la mezcla de otros colores por lo tanto no son colores netamente puros.

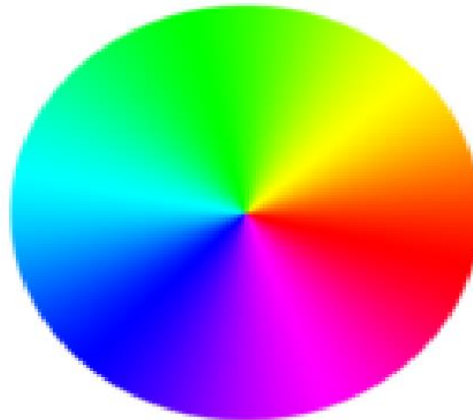


Figura 8: Circulo Del Modelo CMY

1.4.3.3 Representación de los colores

Para lograr un mayor entendimiento de los colores se representa en una rueda de 12 colores divididos en 3 secciones: 3 colores primarios, 3 colores secundarios y 3 colores terciarios la representación en un círculo con esta división de colores tiene el nombre de círculo cromático, el cual está basado principalmente del modelo RYB, estos modelos y rueda de representación es común verla en artistas de pintura, por otra parte en lo que a tecnología se refiere se usa mayormente la rueda RGB esta engloba al modelo CMY como se puede apreciar el conocer los aspectos fundamentales del reconocimiento de colores ayudan a poder crear aplicaciones con tecnologías actuales que dan como resultado poder captar los colores en tiempo real haciendo uso de tecnologías de reconocimiento.

1.4.3.4 Teoría de los espacios de colores

La teoría de los espacios de colores establece que cada color está compuesto por espacios el cual contiene una base de N vectores, si tomamos como ejemplo al espacio RGB el cual se divide en 3 colores podemos determinar que se está dividiendo en 3 vectores, la

combinación en forma lineal tiende a generar el espacio para los posibles colores visibles para el ojo humano, así también hay espacios de colores que suelen aislar los subconjuntos de ellos.

Tabla 2: Tipos de espacios de color

Una Dimensión	Dos dimensiones	Tres Dimensiones	Cuatro Dimensiones
Escala de grises	Sub-espacio rg	Espacio RGB	Espacio CMYK
Escala Jet	Sub-espacio xy	HSV	
		YCbCr	
		YUV	
		Y'I'Q'	

Fuente: (Adriana, Contreras, Garcia, & Calvo, 2014)

Los colores pueden ser identificados por coordenadas que indican una posición dentro de un espacio dando un color en específico, esto no quiere decir que dichas coordenadas nos den un color ya que estas solo nos están entregando el espacio donde se encuentra un color en particular, dichos espacios previamente fueron establecidos por cada color que se tratara de encontrar.

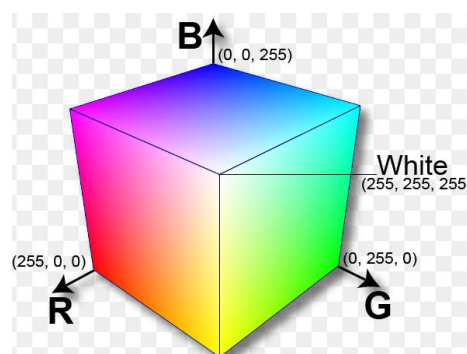


Figura 9: Espacio De Color RGB

Tabla 3: Tipos De Espacios y Características

Tipo de Espacio	Descripción	Características
RGB	Conocido como espacio aditivo por contar con colores primarios los cuales crean	<ul style="list-style-type: none"> Cuentan con 3 dimensiones

	mediante la combinación de luz a los otros colores.	<ul style="list-style-type: none"> • Los colores se determinan por coordenadas • Los colores extendidos son los más utilizados
CMY	Trabaja por medio de la absorción de la luz utilizando como su base los colores secundarios.	<ul style="list-style-type: none"> • Los colores visibles representan a la luz absorbida. • El color negro generado no es tan denso. • Al color negro se le añade un key.
YIQ	Nace de una recodificación para la televisión americana (NTSC), siendo compatible con la televisión en blanco y negro.	<ul style="list-style-type: none"> • El color negro solo requiere de iluminación • Generan cromaticidad de color • Genera una única señal luego de la suma de I más Q
HSV	Representa a un subconjunto del espacio de los valores de RGB.	<ul style="list-style-type: none"> • Posee un espacio cilíndrico • Se asocia a un cono en hexagonal

Fuente: (Barrio, 2011)

1.4.4 Visión por computadora versus visión humana

Luego de haber comprendido como se da la visión tanto por computadora como la visión humana se tiene claro que en marcos de definir quién es mejor a pura lógica podemos decir que la visión humana es la perfección por su forma tan compleja de realizar dicha tarea de entregarnos información del mundo real de una manera gráfica algo que hoy ni teniendo una tecnología muy avanzada se puede competir, la visión humana como se sabe realiza proceso de calibración denominada como visión espectral la cual entrega colores al entrar luz por su parte la visión por computadora depende de varios factores para poder realizar dicho proceso comenzando por la utilización de varios equipos que permitan la captación de información en tiempo real, se debe empezar por definir el tipo de cámaras que vamos a utilizar luego definir con qué tipo de imágenes se va a trabajar ya que estas pueden variar siendo imágenes de formato análogas, de colores o a escala de grises esto es sumamente

importante en el área de visión por computadora ya que dependiendo de a que se va enfocar se utilizara el formato adecuado para luego las imágenes captadas convertirlas a formato digital una vez que se ha conseguido digitalizar las imágenes realizando una serie de arreglos en matriz de pixeles las cuales almacenas diversa de información en forma de valores las cuales se ordenan mediante coordenadas definidas como X,Y. (Aminoff, Toneva, Shrivastava, Chen, & Misra, 2015)

Tabla 4: Ventajas y desventajas de visión por computadora y visión humana

Visión por computadora		Visión humana		
Ventajas	Desventajas	Aplicación	Ventajas	Desventajas
Captura de cuadros(frames) por segundo muy rápidas	Tiene un límite de 50 mega pixeles	Captura de imágenes	Reconoce infinidad de matrices	Captura de cuadros(frames) por segundo lentas
Reconoce imágenes complejas desde diferentes ángulos de percepción	No se adapta en situaciones imprevistas	Reconocimiento	Capacidad para adaptarse a situaciones imprevistas	Necesita tener un reconocimiento previo de objetos
Mejor exactitud de las magnitudes físicas, geométricas y espectro electromagnético	Costos muy altos en la aplicación de mediciones	Medición	x	No posee visión de espectro electromagnético
No presenta desgaste físico por tiempo de trabajo	No se toma en cuenta los tiempos que se le da uso, provocando que no se tenga en cuenta que puede dañarse.	Manejo de Rutinas	Tiene mayor tiempo de duración.	Presenta desgaste físico ocasionando errores
El tiempo de respuesta es superior	Limitaciones en la cantidad de almacenamiento	Procesamiento	Capacidad de almacenamiento ilimitada.	No pueden realizar cálculos complejos en corto tiempo
Realiza análisis minuciosos entregando resultados certeros	Requiere de calibración adecuada para obtener resultados óptimos	Efectos ópticos	Entregan resultados más rápidos por medio de la percepción	Tiende a variar los resultados finales si existen variación de ángulos de visión

Fuente: "Escuela superior politécnica del litoral," 2009

1.4.5 Robótica. - Definición

Según la definición de (José & Valero, 2006) el concepto de robótica nace tras la unión de varias disciplinas con el único fin de lograr crear algo semejante a lo que es el ser humano, al ser un ser humano muy complejo a la vista científica la tarea de un robot se centra en copiar una parte en específica de un ser humano donde tratara de realizar una tarea la cual asido previamente programada, pudiendo un robot lograr realizar una tarea manual de manera correcta y en muchas de las veces mejor a lo que lo hacía un ser humano muchas de estas tareas que realizan de manera mejor suelen ser las que requieren de fuerza física muy excesiva, los trabajos de cálculos en la actualidad se encuentran dominadas por equipos de robots que realizan dichas operaciones de una forma mucho más rápida y eficaz a la que un ser humano podría realizarla .

Por su parte (Miguel et al., 2018) indica que al hacer uso de la robótica en la educación en carreras técnicas ha permitido que la educación en estas áreas tengan un desempeño mucho mayor, los estudiantes tienden a dejar volar su imaginación logrando en muchos de los casos construir robots muy útiles para la sociedad, el autor también no indica que la aplicación de la robótica como en los campos de la educación técnica no es la única en la que esta disciplina asido implantada también se la pueda encontrar en áreas como, las matemáticas, las ciencias sociales, ciencias naturales, ciencias experimentales, ciencias de la información y la comunicación.

Tabla 5: Campos de estudio de la Robótica

Cinemática	Dinámica	Planificación de Sistemas
Control	Sensores	Lenguajes de Programación
Percepción	Arquitectura	Inteligencia Artificial

Fuente: Xavier, & Zatzabal, 2016

Tabla 6: Estructura y componentes de los robots

Componentes	Descripción
Manipulador	Este componente hace referencia a la estructura mecánica que

constituye a un robot		
Sensores	Internos	Permiten comunicar el estado del manipulador de los ángulos de articulaciones, fuerza o presión, etc.
	Externos	Entrega información del entorno y posición actual en la que se encuentra el robot.
Unidad de Control	Funciones Básicas	Inicia y finaliza los movimientos de los componentes.
		Almacenamiento en la memoria de datos la posición y secuencia de todos los movimientos
	Tipos	Facilita la interacción con el entorno usando los sensores
Unidad de Conversión de Potencia	Permite entregar la energía necesaria para poder activar los actuadores.	<u>Secuenciadores</u> <u>Secuenciadores Electrónicos</u> <u>Microordenadores</u> <u>Sistemas Lógicos Neumáticos</u>

Fuente: Xavier, & Zatzabal, 2016

(Odorico & Lage,2018)En su trabajo nos relatan como la robótica se ha ido expandiendo de una manera global siendo muy necesaria en muchos de los países desarrollados ya que han contribuido a su crecimiento, la aplicación de robótica en los campos de la industria son los que mayor se han beneficiado con esto, es por ello que estos países ven a la robótica como una disciplina que hay que explotar mucho más por lo cual realizan inversiones directas en este campo, desde los inicios la robótica a nacido para imitar el comportamiento humano con el fin de realizar tareas cotidianas especificas tanto asido el interés por lograr una imitación perfecta de un ser humano que los expertos en la materia han comenzado a relacionarse con la mente humana siendo este uno de los grandes retos el querer lograr que una máquina tenga autonomía y razonamiento como lo hace un ser humano, en la actualidad se ha conseguido que los robots puedan resolver problemas poco complejos, pero siempre con la esperanza que en un futuro los problemas que puedan resolver los robots sean de muchísima más complejidad.

Tabla 7: Expansión Industrial de los Robots

País	Densidad de robots	Índice de Paro	Nº de Robots
Japón	250	3%	413.578
Suecia	60	9%	5.911
Alemania	58	9%	56.175
Italia	55	12%	25.096
EEUU	35	7%	65.198
Francia	33	12%	14.376
España	22	22%	5.346

Fuente: This, Affairs, Question, & Register, 2018

1.4.5.1 Brazo Robótico

(Robótico et al., 2015) Uno de los campos que la robótica más es usada es en la industria, estas al manejar procesos sumamente precisos los cuales serían muy difíciles que los puede realizar una persona común es por ello que el uso de equipos especializados son de gran demanda en la industria, los más usados son los brazos robóticos por su capacidad para levantar objetos de diferentes tamaños y formas, por su gran capacidad de uso es fundamental en cada empresa se haga uno de estos sistemas, existen de muchos tipos los cuales pueden estar diferenciados por su construcción, tamaño y tipo de agarre a continuación se detallaran los más utilizados con su respectivas características más importantes.

1.4.5.2 KUKA YouBot

Este tipo de brazo robótico nace con el único objetivo de ser de ayuda para los centros de investigaciones de las universidades, al venir integrado con un software de código abierto permite que se puedan desarrollar aplicaciones con control móvil para su manipulación es considerado como el punto que conecta la robótica industrial actual con una mira a lo que será la robótica en el futuro.

Tabla 8: Características Brazo Robótico KUKA YouBot

Girados de libertad	5
Altura	655 mm
Volumen	0.513 m ³
Peso	6.3 kg
Carga	0.5 kg
Estructura	Magnesio Fundido
Comunicación	EtherCAT
Voltaje	24 V
Potencia limitada	80W

Fuente: Autores



Figura 10: Brazo Robótico KUKA Youbot

1.4.5.3 MeArm

Es lanzado como complemento de enseñanza en el campo de la electrónica principalmente es usado con componentes tales como Arduino, Rasperry y otras tarjetas electrónicas su fin principal consiste en que los alumnos aprendan de cómo se encuentra constituido un brazo robótico mediante el armado del mismo.

Tabla 9: Características Brazo Robótico MeArm

Grados de libertad	3
Controlador	Arduino, RaspBerry, etc.
Estructura	Acrílico
Cantidad servomotores	4 microservos

Fuente: Autores

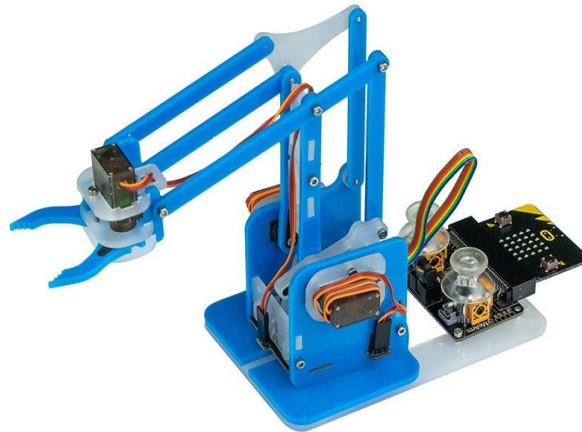


Figura 11: Brazo Robótico MeArm

1.4.5.4 ABB IRBI20

Bajo la empresa ABB especializada en la robótica industrial nace este proyecto como aporte a la educación de forma práctica en los campos de la automatización al ser un robot de sistema cerrado no permite su uso en el campo de la investigación.

Tabla 10: Características Brazo Robótico ABB IRBI20

Grados de libertad	6
Peso	25 kg
Carga máxima	4 kg
Alcance máximo	580 mm
Ruido	Max 70 dB

Fuente: Autores



Figura 12: Brazo Robótico ABB IRBI120

1.4.5.5 Kinova Jaco Re

Este brazo fue creado con la finalidad de ser parte para la investigación especialmente para ser usados en personas que padezcan alguna discapacidad limitada, también en la educación en las universidades tiene la característica de parecerse un brazo humano.

Tabla 11: Características Brazo Robótico Kinova Jaco Re

Grados de libertad	6
Estructura	Fibra de carbono y aluminio.
Peso	5.7 kg.
Altura máxima	90 cm.
Carga máxima	1.5 kg.
Programación	Windows y Ubuntu Linux.
Voltaje de entrada	18 V a 29 V
Velocidad máxima del brazo	15 cm/seg
Temperatura ambiente	0°C a 30°C

Fuente: Autores

CAPÍTULO 2

2. METODOLOGÍA

Para la creación de nuestro sistema el cual esta netamente enfocado a la industria, se realizó un previo análisis de las metodologías ágiles de desarrollos existentes encontrándonos con el adecuado el cual es el método SCRUM, esta metodología permitió enfocarnos de forma detallada en los procesos de cada etapa, permitiéndonos entregar un sistema completo que cumpla con las expectativas que se tenía al momento realizar nuestro proyecto.

Fase 1: Planificación de Análisis inicial

Paso 1: Analizar cómo usar el software LabVIEW y decidir que módulos usar en el desarrollo del sistema propuesto.

Como parte del proceso de análisis en esta parte nos centramos en conocer a fondo el software con el que desarrollaremos el sistema el cual es uno muy completo que está plenamente dirigido para el sector industrial, incluyendo en el mismo importantes diferencias empezando por integrar un lenguaje de programación propio, utilizando programación de bloques que nos permitirá crear desde la interfaz del sistema hasta la comunicación correcta con los diferentes elementos de hardware.

Paso 2: Analizar cómo se debe crear código en la IDE de Arduino de manera correcta

Conocer cómo usar la IDE de Arduino de manera correcta es uno de los principales pasos que tenemos que realizar, empezando por identificar cómo realizar una programación estructurada haciendo uso de librerías propias como también como crear nuestras propias funciones que ayuden a escribir código mucho más limpio y reutilizable.

Paso 3: Analizar los componentes de hardware que se utilizaran para el sistema.

Para que el sistema pueda funcionar de manera correcta se necesitó de varios componentes de hardware sumamente importantes que permitirán las conexiones físicas de un prototipo de brazo robótico, banda transportadora y cámara USB por motivo de reducir costos sin perder eficacia en el sistema se decidió utilizar la placa Arduino Uno, junto una shield para servos que permitirá alimentar de energía DC sin depender de la placa Arduino.

Fase 2: Diseño

Paso 4: Diseño de la interfaz de usuario en LabVIEW

En este paso se procedió a realizar el diseño de interfaz la cual será la ventana de control que tendrá a disposición el usuario, esta interfaz fue diseñada de manera que cualquier usuario lo pueda entender manejando activadores gráficos para el encendido y apagado de los servomotores además de un apartado para visualizar en tiempo real los objetos que se presenten en el momento.

Paso 5: Diseño de modelo de base de datos en SQL Server

En este paso se realizará un modelado de base de datos la cual servirá para almacenar los datos de la cantidad de botellas clasificadas con sus respectivas fechas y horas.

Fase 3: Programación

Paso 6: Realizar el código pertinente para la detección de color

Al tener estudiado el uso del software de LabVIEW podemos determinar que la codificación se la realiza por medio de bloques las cuales están integradas entre sí para que

puedan trabajar con el hardware necesario en este caso se trabajara con el código que permita determinar el color de los objetos, este código permitirá que el sistema pueda clasificar los colores y entregarnos un resultado final lo más eficaz posible.

Paso 7: Realizar el código en la IDE de Arduino para la comunicación entre LabVIEW, cámara USB, Brazo Robótico y Banda Transportadora.

Uno de los principales retos es realizar la codificación en la IDE Arduino que permita la comunicación integrada de todos los elementos de hardware que se utilizaran, Arduino nos permite hacer uso de librerías propias como lo es para el control de servomotores y la comunicación Serial las cuales permitirán que el sistema tenga el funcionamiento correcto.

Fase 4: Pruebas

Paso 8: Realizar pruebas de conexiones de los componentes de hardware.

Es preciso realizar pruebas en las conexiones realizadas de todos los componentes para evitar errores con el funcionamiento del sistema, es por ello que se revisaran cada uno de las conexiones de cables realizando los ajustes respectivos si hubiera el caso de hacerlo.

Paso 9: Realizar las pruebas de calibración de los ángulos de los servos del brazo robótico

El sistema cuenta con un prototipo de brazo robótico el cual realiza diferentes acciones al detectar la cámara un color pre establecido es por ello que se realizó pruebas constantes hasta conseguir la postura correcta de los ejes para que este pueda realizar las acciones en tiempo real sin errores.

Paso 10: Realizar pruebas del desplazamiento de los objetos por la banda transportadora

Al integrar una banda transportadora para que esta desplace los objetos hacia un punto específico se verifico que los objetos lleguen de manera correcta sin sufrir algún

contratiempo estos se pueden dar por falta de grasa en los rulimanes o en peor de los casos encontrar fallas en el motor de paso.

Paso 11: Ejecución de pruebas de funcionamiento del sistema

Para poder entregar un sistema funcional se realizaron varias pruebas empezando por analizar los tiempos de respuestas el cual debe ser optimo consiguiendo con esto que los procesos se realicen de manera autónoma y seguro, además se probara la base de datos realizando ingresos de datos como pruebas para seguido de la visualización de los mismos, como sensor principal del sistema se realizaron varias pruebas con cámaras distintas pudiendo utilizar cámaras USB de diferentes resoluciones de video.

A continuación, se representará el flujograma de la metodología implementada

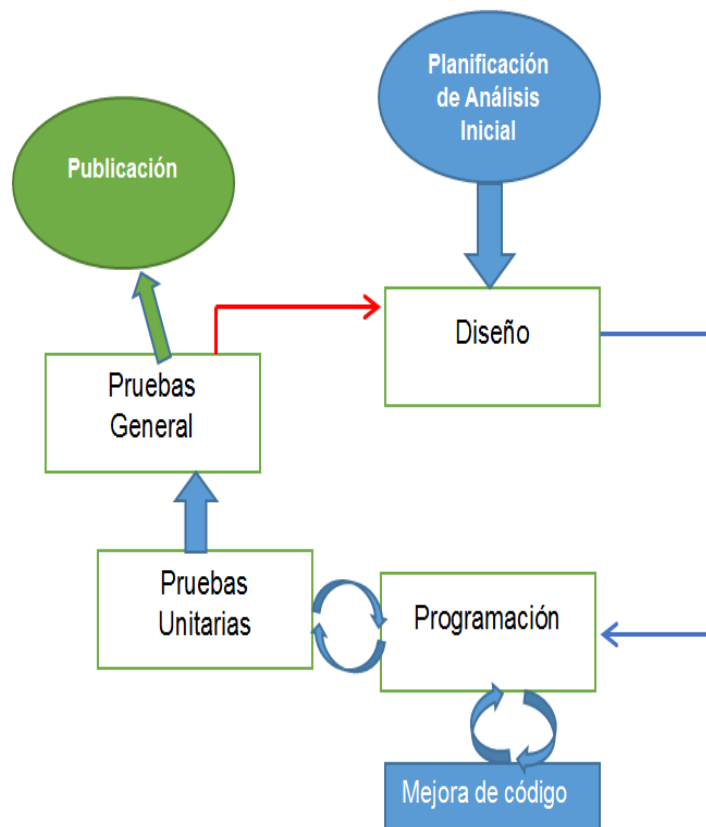


Tabla 12: Flujograma de la metodología aplicada

CAPÍTULO 3

3. RESULTADOS (ANÁLISIS O PROPUESTA)

3.1 Desarrollo

3.1.1 Fase de Análisis

Software a utilizar

LabVIEW

Permite realizar una serie de aplicaciones con diferentes funciones, en este caso se destaca en el procesamiento de imagen y visión por computador, debido al enfoque que tiene en este proyecto, brinda una codificación interactiva debido a que su lenguaje es gráfico (programación por bloque), ahorrando tiempo en escribir largas líneas de código y a destacar, no se tiene inconvenientes al usar con otros sistemas operativos (Linux, Mac, Windows), conjuntamente podemos acoplarlo con otros lenguajes de programación (Matlab, Visual Basic, C++).(Quiñones & Bernal, 2011)

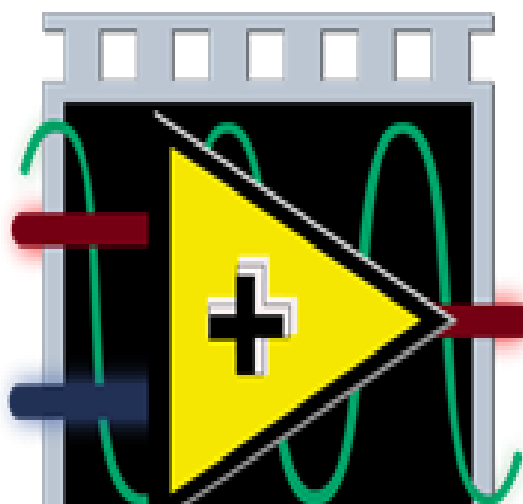


Figura 13: Icono de LabVIEW

Para este proyecto se utilizaron varios módulos que a continuación se describirán:

Tabla 13: Módulos Requeridos para el sistema

Módulos	Descripción	Detalles
Visión	Permite trabajar con el procesamiento de imágenes y visión artificial.	En Windows trabaja en cualquier arquitectura y está disponible hasta W7, lo tenemos en idioma inglés.
NI-Imaq	Esta plataforma permite crear programas usando notación grafica que simulan elementos físicos unidos de manera virtual.	Trabaja en sistema operativo de Windows, viene en el idioma inglés.
NI-VISA	Permite la comunicación con implementos de E/S, siendo un estándar para configurar, programar y depurar sistemas de instrumentación.	La última versión es la 19.0, viene en varios idiomas, es compatible con los sistemas Windows y Macs(McLoughlin, 2009).

Fuente: Autores

Módulo Vision and Motion

En el menú de Vision and Motion, se encuentran ubicado todas las funciones como NI-IMAQ, Image Processing, donde estas sirven para la visión artificial.

Al acceder a NI-IMAQ se presentan las funciones para las cámaras, adquirir imágenes, mientras tanto en el bloque de image processing permite, filtrar, analizar las imágenes originales además de obtener los histogramas, cambiar el contraste entre otras funciones.

En la figura 11 se puede observar el menú de Vision and Motion

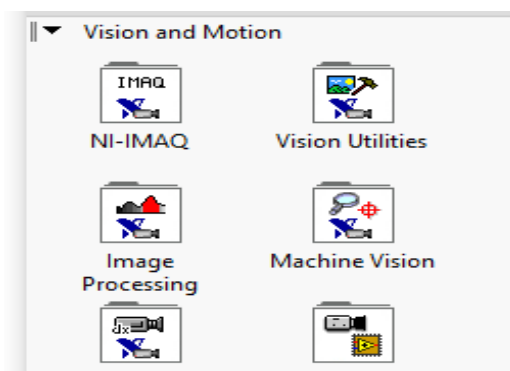


Figura 14: Menú de complementos de Vision and Motion

NI-IMAQ

Para poder realizar el procesamiento de imágenes en tiempo real necesitamos usar los complementos de NI-IMAQ en el cual encontramos todo lo necesario empezando por realizar la comunicación del computador con una cámara USB, seguido se debe crear una unidad virtual de almacenamiento en donde todo lo que valla grabando se almacene en tiempo real para poder visualizar el video en caso de errores tenemos el complemento que nos cierra el programa evitando que nuestro sistema sufra un error irreparable.

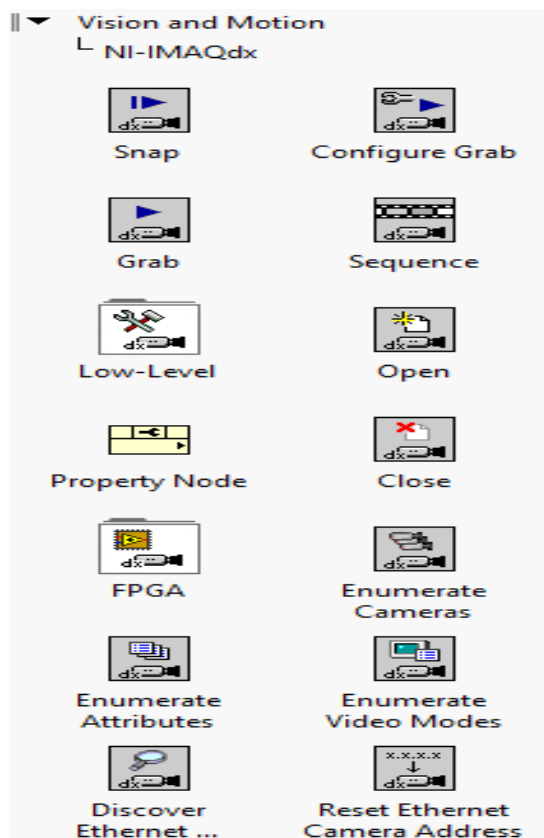


Figura 15: Menú NI-IMAQ

Modulo NI-VISA

NI-VISA (Virtual Instrument Software Architecture) es un controlador el cual permite tener las conexiones sea serial, PXI, USB, Ethernet, entre otros dispositivos, las aplicaciones básicas de NI-VISA es abrir sesión, cerrar, manejos de errores que ocurran, escritura y lectura a los dispositivos.

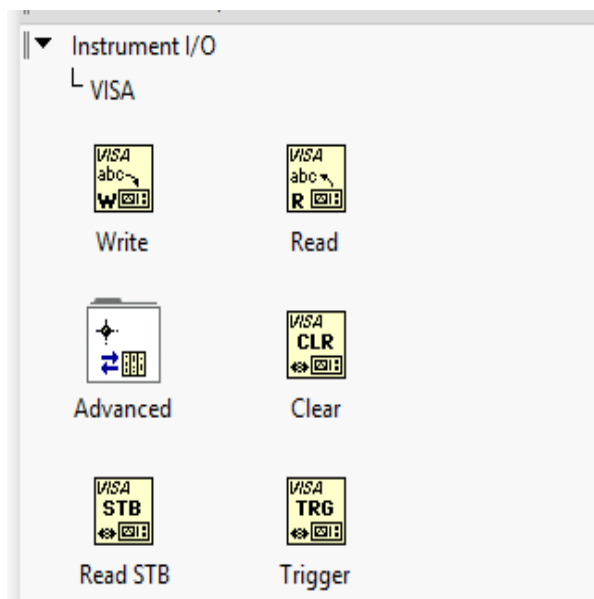


Figura 16: Componentes del Módulo NI-VISA

Tolkit SQL

El manejo de bases de datos ha permitido que las empresas presenten reportes mensuales trimestrales o anuales con el fin de mantener informado de la situación en la que la empresa se encuentra es por ello que para poder utilizar una de las bases de datos más conocidas como lo es SQL Server es necesario contar con un toolkit para el manejo de base de datos, este Toolkit permite tener los complementos para conectar una base de datos y poder enviar las sentencias que permitan presentar, eliminar, editar y actualizar.

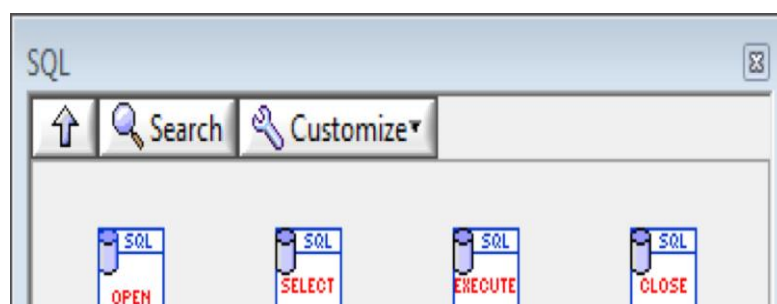


Figura 17: Toolkit SQL

SQL Server



Figura 18: Logotipo de SQL Server

Brinda facilidad en la conexión con otros tipos de lenguaje de programación (C++, Visual Basic, Matlab, Lavbiew, entre otros.), se puede crear, manipular bases de datos en las cuales podemos almacenar una gran cantidad de información, Transact-SQL es su lenguaje de programación, está hecho en el modelo relacional. La comunicación con el software de LabVIEW es óptima donde se almacenará la información registradas de los objetos, además de darnos una serie de beneficios tales como:

- **Portabilidad:** Permite trabajar dese cualquier computadora porque podemos generar script o copiar el código en un texto de nota, para usarlo en otras computadoras.
- **Compatibilidad:** No se necesita tener la misma arquitectura del computador (32 y 64 bits) o version del software SQL.

- **Conectividad:** Facilita trabajar con otro sistema de BD, donde podemos visualizar, almacenar información, comunicándose con otro software.
- **Seguridad:** Da integridad sobre los datos en los cuales podemos crear niveles de usuarios, evitando que haya una mala manipulación de datos.
- **Administración:** Optimiza los recursos del computador, tiene un agente XPs en el cual podemos programar tareas que se ejecuten automáticamente sea por días, semanas o meses.
- **Rendimiento:** Procesa rapido las sentencias ejecutadas sin destabilizar el computador, debido a que no consume muchos recursos. (Clark, Cobb, Kapfhammer, Jones, & Harrold, 2011)

Sentencias SQL

SQL maneja tres sentencias específicas, cada una tienen sus descripciones, funcionalidades y sus instrucciones las cuales cumplen cada una.

Tabla 14: Sentencias Transact de SQL

Lenguaje	Descripción	Funcionalidad	Instrucciones
DLL	Lenguaje de definición de datos	Son sentencias las cuales permiten trabajar en las bases de datos, dichas sentencias nos permiten la creación de vistas, procedimientos almacenados, tablas, trigger, índices	<ul style="list-style-type: none"> • Create • Alter • Drop
DML	Lenguaje de manipulación de datos	Teniendo nuestra base de datos sea llena o vacía se usa sentencia distinta las cuales nos permiten el ingreso, actualizar o borrar datos.	<ul style="list-style-type: none"> • Insert • Update • Delete • Select
DCL	Lenguaje de control de datos	Permite crear restricciones o niveles de usuarios	<ul style="list-style-type: none"> • Grant • Revoke

Fuente: Los Autores

Arduino IDE



Figura 19: Logo de Arduino

Es un software liviano, de código libre, basado en la estructura C++, disponible para todo tipo de sistema operativo, tiene disponibilidad de agregar algunas librerías para comunicar la placa con otros circuitos externos, para tener comunicación entre el servomotor y la placa Arduino se necesita cargar en el IDE la librería Servo.h. Las ventajas para usar este IDE es que no es tedioso al traer 5 botones básicos y nos da la facilidad de verificar si el código está mal escrito o se necesita una librería antes de ser almacenado en el microcontrolador, su monitor serial nos permite también saber si la tarjeta Arduino está mandando la ejecución correcta al servomotor.

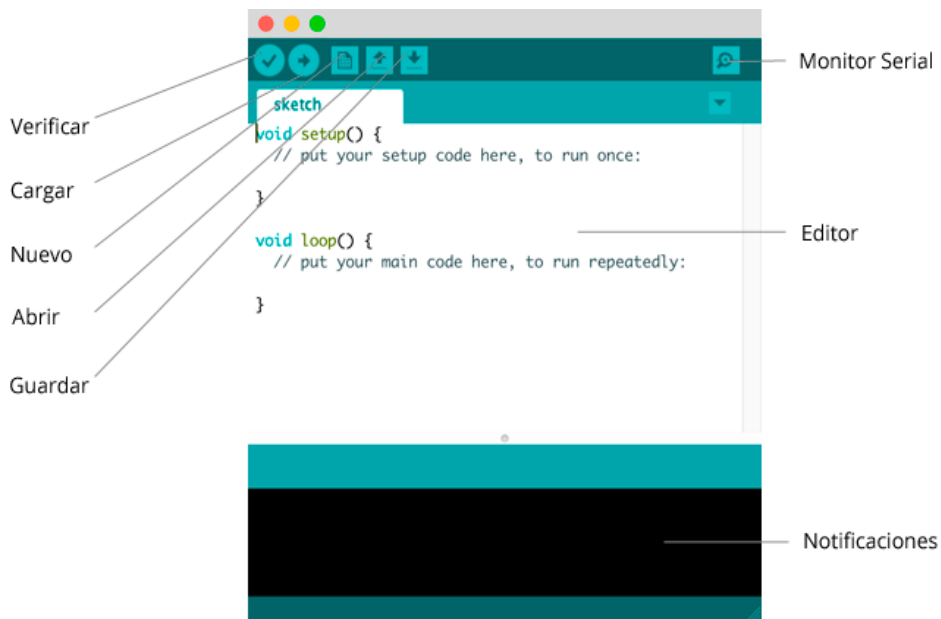


Figura 20: Interfaz de la IDE de Arduino

Hardware a Utilizar

Ordenador

Se trabajó con un computador portátil de marca Toshiba Satélite S55-B5268, sus especificaciones son las siguientes:

Tabla 15: Características Principales del Ordenar Utilizado

Características	Detalles
Procesador	Intel Core I7 4510U 2.0GHz
Pantalla	15.6 HD , táctil
RAM	12GB
Disco Duro	1 TB
Tarjeta de video	2GB
Peso	2.2 kg

Fuente: Autores

Las especificaciones descritas muestran que trabajar con esta laptop no presentaremos inconvenientes al tener 12GB de memoria RAM, que da una respuesta rápida de la ejecución del sistema y a destacar que viene con una tarjeta de video dedicado optimizando el uso de la visión artificial.



Figura 21: Ordenador Toshiba

Cámara USB

La cámara elegida es de marca Omega con conexión USB para la adquisición de imágenes, a continuación, se describen sus detalles técnicos:

Tabla 16: Características Principales de la Cámara USB

Características	Detalles
Resolución	640 * 480 hasta 1600*1200
Enfoque manual	360 grados
Efecto de video	30 cuadros por segundo
Sensor	CMOS
Megapíxeles	8
Conexión	USB 2.0

Fuente: Autores

Por su conexión USB nos da la facilidad de trabajar y de ubicarla a un lado de la banda transportador para detectar los objetos que se vayan a identificar aplicando visión por computador en tiempo real.



Figura 22: Cámara USB

Placa Arduino Uno

Integrado por un micro controlador y una placa. Es muy utilizado actualmente por brindar un sin números de funciones permitiendo leer datos sea de sensores, interruptores, motores o luces. Se lo implementa por su bajo costo, facilidad en la configuración, posee características muy importantes que se detallan a continuación:

Tabla 17: Especificaciones Técnicas de la Placa Arduino

ESPECIFICACIONES	
Microcontroller	ATmega328
Operating Voltaje	5V
Input Voltage (Recomendado)	7-12V
Input Voltage (Limite)	6-20V
Digital I/O Pins	14(of which 6 provide PWM output)
Analog Input Pins	6
DC Current for I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Fuente: Autores

La comunicación que se puede realizar utilizando la placa Arduino con el ordenador es mediante conexión serial la cual se establece conectado un cable con conexión USB, una vez conectado a uno de los puertos del computador se deberá ver mediante la IDE de Arduino a que puerto COM se encuentra conectado, hacer uso de una placa Arduino tiene muchos beneficios uno de los principales es su forma sencilla de cargar código que contienen comandos que permiten que la placa permita interactuar con diversos dispositivos electrónicos pudiendo realizar proyectos interesantes.



Figura 23: Placa Arduino Uno

Módulo Shield PCA9685 PWM 12-bit Driver I2C

Permite ampliar las conexiones de las tarjetas de circuito, brindando una mayor funcionalidad, su costo es accesible. Consta de 16 pines, se consigue 992 PWM de salidas, se describe las características más importantes de la tarjeta Shield.

Tabla 18: Características Técnicas de la Shield

Características	Descripción
Tipo de tarjeta	Shield PCA9685
Voltaje de operación	DC 3-5 V
Voltaje máximo	6 V
Frecuencia	40-1000HZ
Resolución	12 bits
Peso	40 g
Dimensiones	6.2x2.5x1.2 cm

Fuente: Los autores

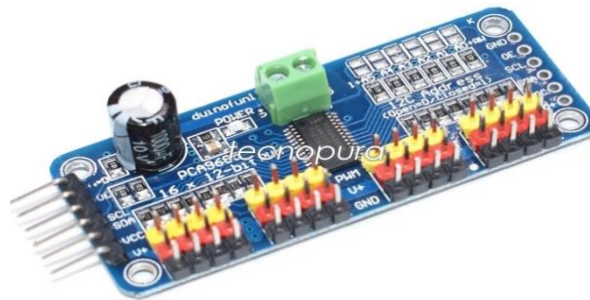


Figura 24: Modulo Shield PCA9685

Componentes Necesarios para el Prototipo de Visión por Computadora

Brazo Robótico

Diseñado por un conjunto de circuitos, tales como sensores, transmisores, entre otros, permitiendo ejecutar una serie de movimientos programados para implementarlos en

algunas áreas de trabajo, destacándolo en el área industrial para la clasificación de empaquetados, una de sus ventajas que permite ser controlado desde cualquier modelo de celular (IOS o Android) y ordenador para la implementación de nuestro sistemas fue necesario hacer uso de un brazo robótico por el cual se decidió adquirir uno que nos facilite el montaje de nuestro sistema de visión por computadora, por lo cual el brazo robótico elegido para nuestro prototipo es el conocido LeArm Robot distribuido por la empresa LewanSoul , especializada en distribuir a nivel mundial componentes de robótica para uso educacional o industrial, todos los dispositivos que la empresa distribuye viene desmontado por lo cual la persona que los compre deberá armarlo a continuación se presentara las características y especificaciones con las que cuenta nuestro prototipo de brazo robótico.

Tabla 19: Especificaciones Técnicas del Brazo Robótico

Especificaciones	Detalles
Alimentación de entrada	100 v- 240 V (50/60 HZ)
Alimentación de salida	7.5 V /5 A
Material	Metal inoxidable

Fuente: Lewansoul

Tabla 20: Características del Brazo Robótico

Características	Detalle
Peso	60 g
Dimensión	40*20*40.5mm
Velocidad	0.16sec/60°7.4V
Esfuerzo de torsión	15kg.cm 6.6V; 17kg.cm 7.4V
Voltaje de trabajo	6-8.4V
Corriente sin carga	100mA
Tamaño del cable	30cm

Fuente: Lewansoul

 Long U-shaped bracket	3pcs	 Side cover bracket	1pcs	 Small U-shaped bracket	1pcs	 Multifunctional Bracket	2pcs	
 Rotational station	A Set	 Bearing	1pcs	 Power Adapter	1pcs	 Large bottom plate	1pcs	
 Clamper	1pcs	 LDX-218 digital servo	2pcs	 LFD-06 digital servo	2pcs	 LD-1501 MG digital servo	1pcs	
 Wireless handle and handle receiver	A Set	 Controller	1pcs	 Winding tube	Several	 Screwdriver	1pcs	
 M4*8 Copper Column 4pcs	 M3*8 Nylon Column 10pcs	 Small Bearing 1pcs	 M4 Screw Nut 20pcs	 M3 Screw Nut 10pcs	 M2 Screw Nut 10pcs	 M4*8 Screw	 M3*10 Screw	 M3*8 Screw

Figura 25: Componentes del Brazo Robótico



Figura 26: Brazo Robótico LeArm

Banda Transportadora

La banda transportadora permitirá que nuestro sistema pueda realizar operaciones de manera automatizada ya que estará comunicándose con el sistema el cual le indicará cuando activarse o cuando detenerse, existen gran variedad de bandas transportadoras en el mercado las cuales tienen un uso diferente dependiendo para que material estén destinadas a transportar como se observa en la siguiente tabla para una mayor comprensión.

Tabla 21: Tipos de Bandas Transportadoras

Tipos de Productos	
Bulks	Son utilizadas para productos de gran volumen que no se encuentran empacados. Esta banda es sugerida para transportar carbón, cemento, granos u otros materiales.
Unidad de manipulación	Están diseñados para productos empacados en cajas o envases de bajo peso o volumen.

No acumulador	Solo sirven para procesos de carga y descarga deteniéndose cada vez que se vaya a realizar una de las tareas descritas.
---------------	---

Tipos de Conveyor

Cinta con rodillos	Se caracterizan por estar conformadas por conjuntos de rodillos. Se utilizan para el traslado de productos durante la producción y la distribución.(M & E, 2011)
Banda de cinta	Se caracteriza por ser bandas de transporte continuo donde no se presentan interrupciones a la hora de su funcionamiento.

Fuente: Autores

Para nuestro sistema de Visión por computadora para la clasificación por color se decidió utilizar una banda transportadora de cinta con rodillos y a su vez destinada a productos empacados en cajas pequeñas.

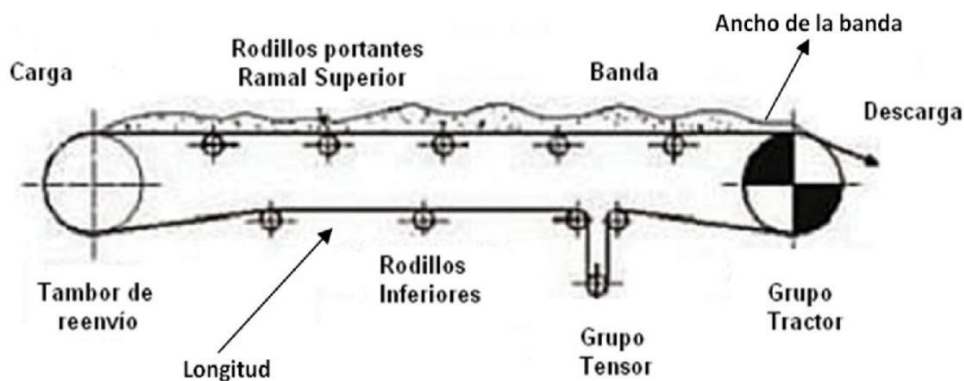


Figura 27: Componentes de Banda Transportadora

Para que la banda transportadora pueda moverse se utilizó un motor de paso con las siguientes características:

Tabla 22: Características Técnicas del Motor de Paso

Especificaciones	Detalles
Tamaño	42.3x48mm, sin tomar medir el eje
Peso	350 gramos (13 Oz)
Diámetro del eje	25 mm
Pasos por vuelta	200 (1.8º/paso)
Corriente	1.2 Amperios por bobinado
Tensión	4 V
Resistencia	3.3 Ohm por bobina
Torque	3.2 kg/cm (44 oz-in)
Inductancia	2.8 mH por bobina

Fuente: Autores

El diseño de la banda Transportadora fue creado a propuesta del tutor académico escogiendo como material principal para la estructura la madera y para los rodillos tubos de fibra de nylon, en la figura 25 se muestra cómo quedaría la estructura de la banda transportadora.

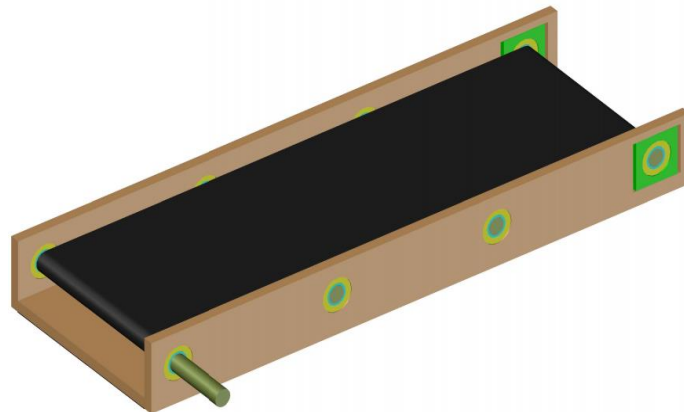


Figura 28: Diseño 3D de la Banda Transportadora

3.1.2 Fase de Diseño

Diseño de Interfaz de Usuario en LabVIEW

La interfaz de usuario es una de las partes primordiales del sistema en ella se podrán realizar las distintas operaciones con los que nuestro sistema contara como lo es el detectar en tiempo real los colores mediante el uso de una cámara USB, al detectar el color indicado el sistema alertara de manera visible por medio de la pantalla del ordenar el color que se ha detectado logrando encender unos leds digitales con lo que se reflejara el color tomado por la cámara.

El diseño de la interfaz contara de tres fases las cuales son:

1. Diseño del Menú Principal

2. Diseño de Interfaz del Sistema de Visión por Computadora para la Detección de Colores
3. Diseño de Interfaz del modo control Manual para el Brazo Robótico

Luego del diseño de la interface se procederá a diseñar la tabla de la base de datos la cual será representada en diagrama de clases.

Diseño del Menú Principal

El menú principal corresponde a la pantalla inicial que vera el usuario al iniciar el sistema contara de botones que nos llevara a la interfaz seleccionada. Para esta parte el diseño contiene los nombres de los integrantes, nombre de la institución y nombre dado al Sistema.



Figura 29: Diseño final de la Interfaz del Menú Principal

Gracias al entorno de desarrollo que tiene LabVIEW se pueden realizar interfaces de una manera mucho más rápido que en otras lenguas, al ser un entorno de desarrollo grafico bastara con arrastrar los elementos que necesitemos le coloquemos el nombre, color, tamaño que se apege a nuestros gustos como se puede apreciar en la figura 30 solo se ha

colocado botones imágenes y texto todo esto dentro de un reducido marco permitiendo así tener un menú principal sencillo para su manipulación.

Diseño de Interfaz del Sistema de Visión por Computadora

Como punto principal del sistema tenemos a la interfaz de usuario creada en LabVIEW, siendo esta de fácil manejo por la simplicidad en sus controles, al necesitar de solo visualizar como se está llevando a cabo la detección de imagen el sistema no necesita contener en su interfaz principal muchas opciones que al final complicaran al usuario, solo se colocara los campos visuales y los componentes que permitan la conexión con la cámara y comunicación con Arduino.



Figura 30: Diseño final de la Interfaz del Sistema de Visión por Computadora para la Detección de Colores

Como se puede observar en la figura 31 la interfaz de usuario es muy sencilla a primera vista por no contener botones que puedan causar el mal funcionamiento del mismo, bastara con seleccionar en los campos de configuración la cámara a la que se va a conectar como el puerto COM para la comunicación con Arduino.

Diseño de Interfaz control Manual para el Brazo Robótico

A pesar del sistema estar orientado para que se maneje de manera autónoma mediante la detección de color también se decidió colocarle una Interfaz para el modo manual donde se podrá controlar los servos con los que esta ensamblado el Brazo Robótico permitiéndole al usuario poder mover el brazo mediante el ordenador con tan solo mover los ejes virtuales que se han colocado en la interfaz

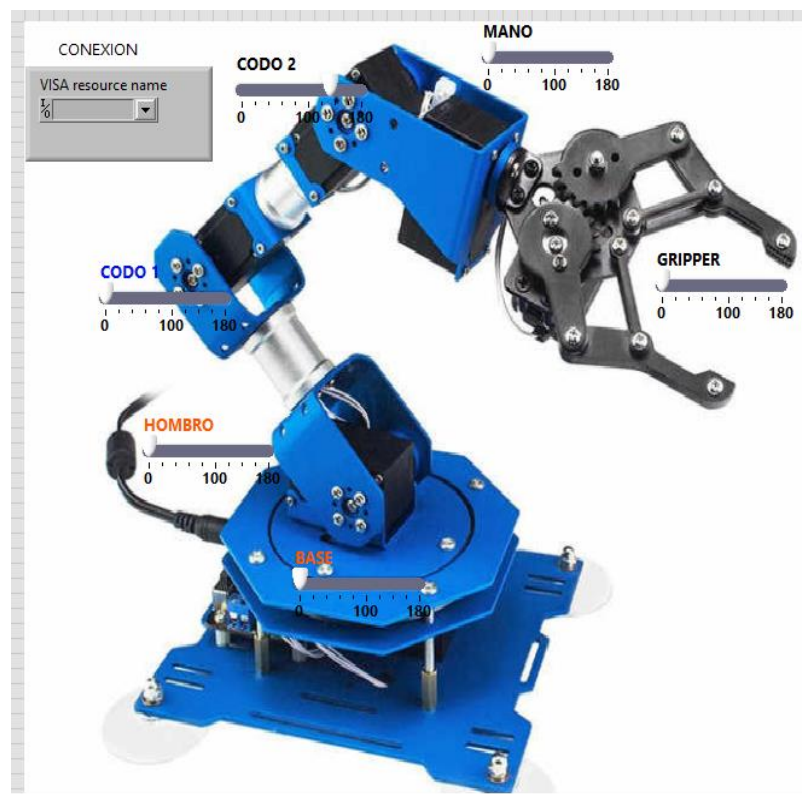


Figura 31: Diseño de Interfaz de Brazo Robótico

Diseño de la Tabla UML de la Base de Datos

Al ser un sistema integrado con varias funcionalidades se tuvo que incluir una conexión a la base de datos SQL Server para que se almacene el total de objetos detectados captó por día el sistema con este añadido permite al usuario comparar si el sistema le ha aumentado

la calidad de clasificación ya que si este fuera el caso los números diarios tendrán que aumentar a comparación a como lo hacía una empresa de manera manual.



Figura 32: Diseño de Tabla UML Colores

3.1.3 Programación

Programación en LabVIEW

Vis para del Reconocimiento de Color

Lo primero que tenemos que comprender que para poder reconocer los colores se debe tener una conexión con la cámara en tiempo real la cual permitirá que esta vaya almacenando de manera local lo que va captando para luego aplicando elementos propios de LabVIEW pueda reconocer los colores para dicho trabajo se está utilizando la técnica de espectro de colores, es decir dividir los colores por su luz y a su vez separarlos por matriz enviándole rangos al cual previa identificación de los colores que se requerirán se colocaron dichos rangos, a continuación se detallara la programación para obtener imagen en tiempo real.

Vis para la Conexión con la Cámara USB

Una vez abierto un nuevo proyecto procedemos a codificar mediante bloques lo primero que se va a colocar será los bloques que se encuentran en la parte de Vision and Motion aquí encontraremos bloques que están dirigidos para el tratamiento de imágenes una vez ubicado seleccionamos la que dice NI-IMAQdx seguido buscamos el bloque de Open la cual servirá para inicializar la cámara luego colocaremos la de Configure Grab y finalmente la de Grab

con estos 3 bloques tendremos nuestra estructura inicial, Una vez inicializado los bloques de apertura configuración y grabación procedemos a colocar un ciclo while el que estará en el apartado llamado Programming seguido de Structures y procedemos a seleccionar el ciclo llamado While loop. Nos dirigimos al entorno que se encuentra diferenciado por cuadrículas y luego vamos al apartado de bloques y seleccionamos donde dice Visión aquí procedemos a seleccionar el que dice ImageDisplay(Classic) este servirá para verlo que la cámara está captando en ese momento.

Procedemos a colocar un bloque que permita se almacene la imagen de manera virtual con esto podemos garantizar que al detener el programa esta se detenga y cargue la imagen final en la que se detuvo para ello vamos a Vision Utilities seguido a Image Management y finalmente seleccionamos la que tiene el nombre de IMAQ Create y la colocamos fuera del ciclo While una vez enlazado con el bloque Grab que está dentro del ciclo While procedemos a fijar el puntero en el punto de Image Name damos clic derecho y seleccionamos Create y colocamos constante y le damos un nombre que queramos. Como punto final colocaremos el bloque de finalización y a su vez un botón de Stop , para detener el proceso de encendido de la cámara debemos colocar el bloque de Close que se encuentra en NI-IMAQdx esta se colocara fuera del ciclo While que hemos colocado, una vez realizado esto procederemos a ubicar el punto rojo que se encuentra dentro del ciclo while le damos clic derecho en el punto que aparece y procedemos a seleccionar Create Control, también en el bloque de Open debemos colocar el indicador de la cámara a la cual queremos acceder por lo cual daremos clic derecho en la primera entrada seguido de Create y luego Constant una vez hecho esto en la constante creada le daremos clic en la flecha desplegable y seleccionamos cam0.

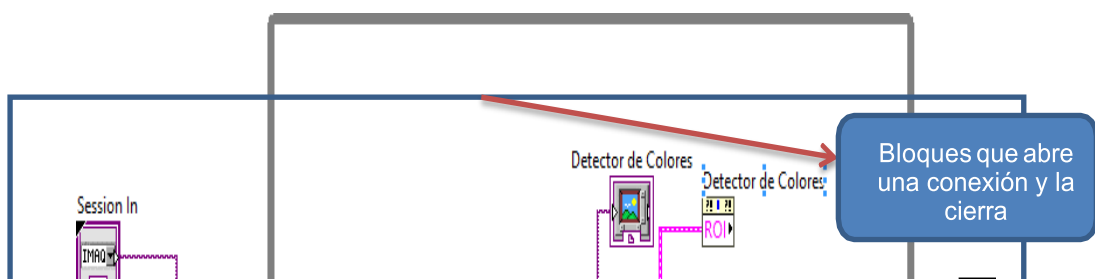
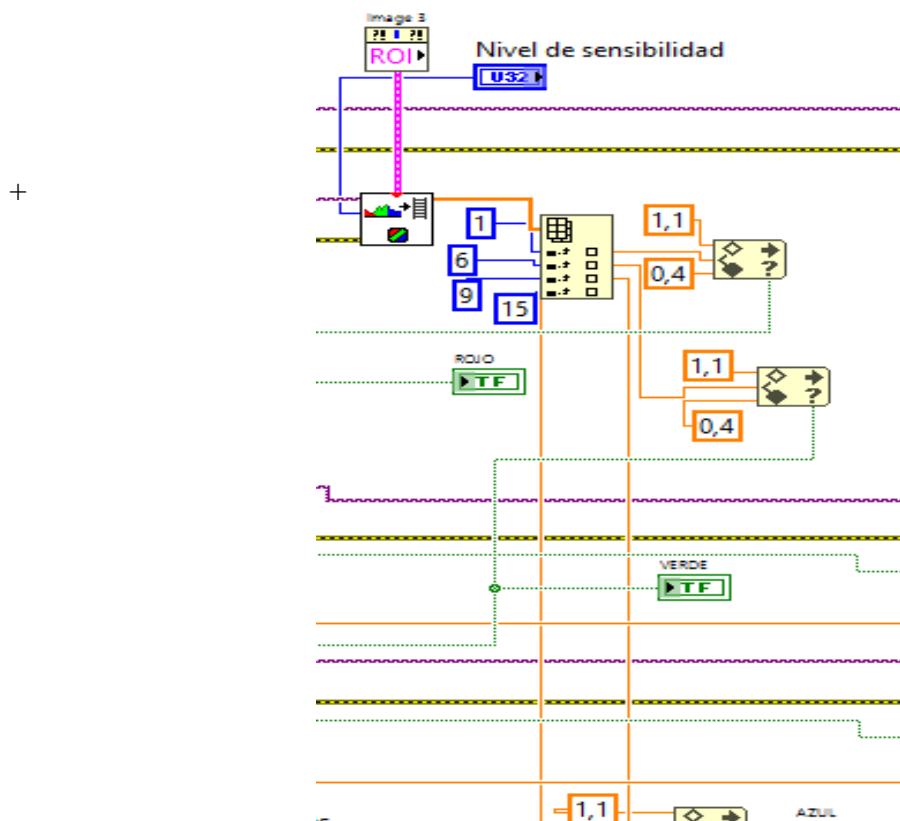


Figura 33: Bloque para Conectar a la Cámara USB

Vis para el reconocimiento de Color en LabVIEW

Al tratar con procesamiento de imágenes podemos acogernos a varias funciones que trae la propia aplicación para la detección de colores una de ellas que nos pareció interesante es con el manejo de espectros de colores la cual posee las imágenes con RGB es por ello que el código implementado se basa en aquello en usar el espectro en nivel bajo el cual subdivide los colores en 7 secciones de 2 partes es decir lo divide en 14 partes en donde se comienza desde abajo donde captara colores con intensidad un poco más claras, esta técnica permite definir qué color buscar indicando valores de comparación de rangos entre 1,1 y 0,4



Programación para el Enlace entre LabVIEW y la Base de Datos

Para realizar una conexión entre Labview y SQL Server se haría uso de un Toolkit el cual nos entrega de manera gráfica los diagramas de conexión en forma de bloques la cual resulta muy sencilla de enlazar tanto para realizar sentencias de consultas y también para realizar transacciones el Toolkit nos entrega 4 bloques (Open, Select, Execute, Close)

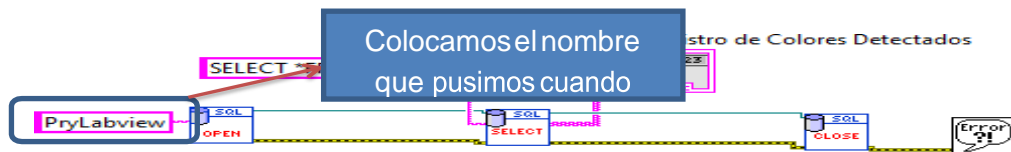


Figura 36: Programación de Enlace Arduino LabVIEW

Programación para registro en la base de datos de objetos detectados por color

La programación para registro de colores detectados es muy corta por lo fácil que nos entrega el toolkit de SQL el cual con 4 bloques nos permite realizar dicha operación, los registros se lo realizara cuando le demos en el botón detener, esto para que solo se guarde el último registro total de lo que se contó en el día.

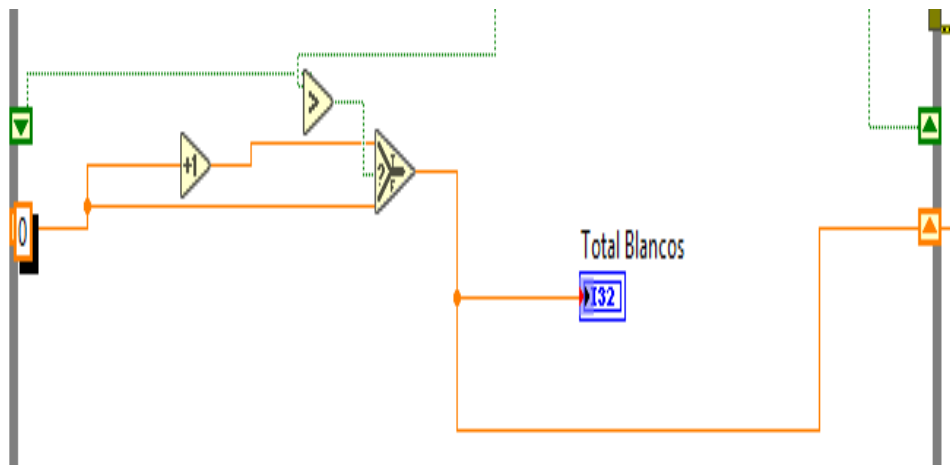


Figura 37: Contador de Total Objetos de Colores Detectados

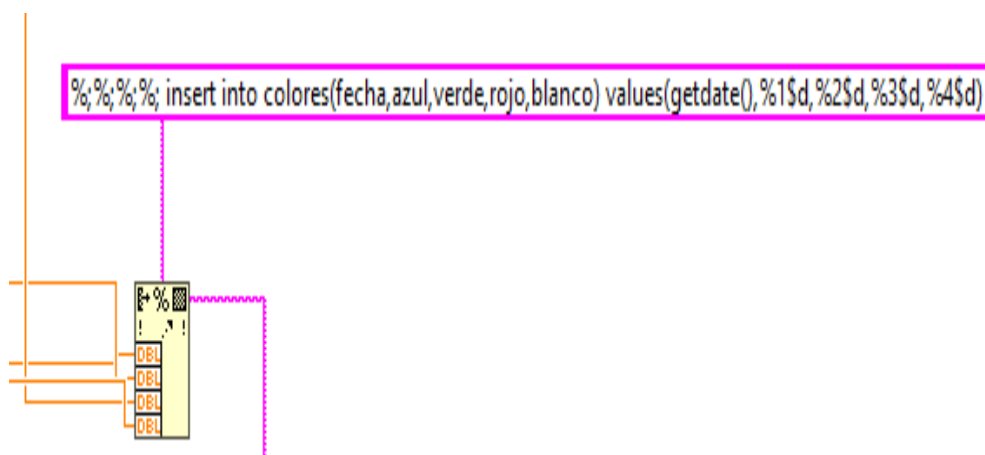


Figura 38: Registro a la Base de Datos

Programación en SQL Server

Código DLL para la Creación de Base de Dato y Tabla a Utilizar.

La creación de bases de datos resulta muy simple si se conoce las líneas de código correctas consiguiendo reducir el tiempo de escritura ya que integra comandos que hacen que escribir código en esta IDE sea más sencillo por su gran variedad de librerías que engranan.

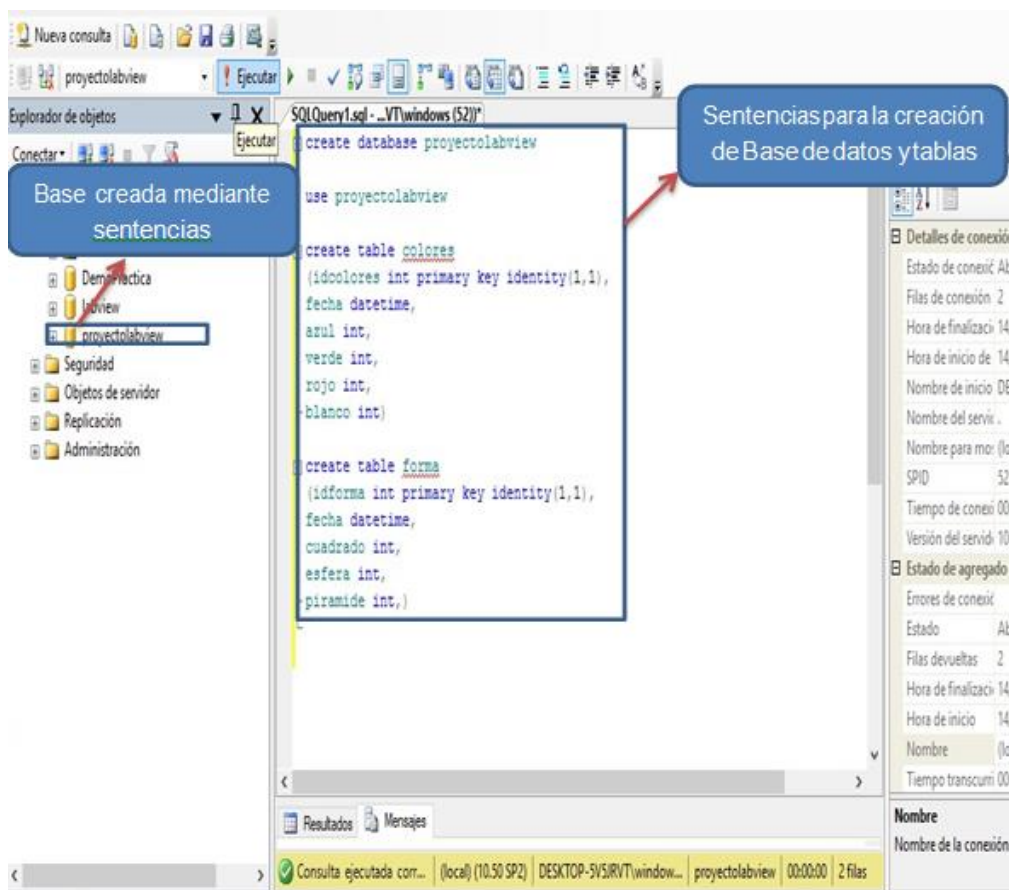


Figura 39: Código Transact para la Creación de Base de Datos y Tabla

3.1.4 Pruebas

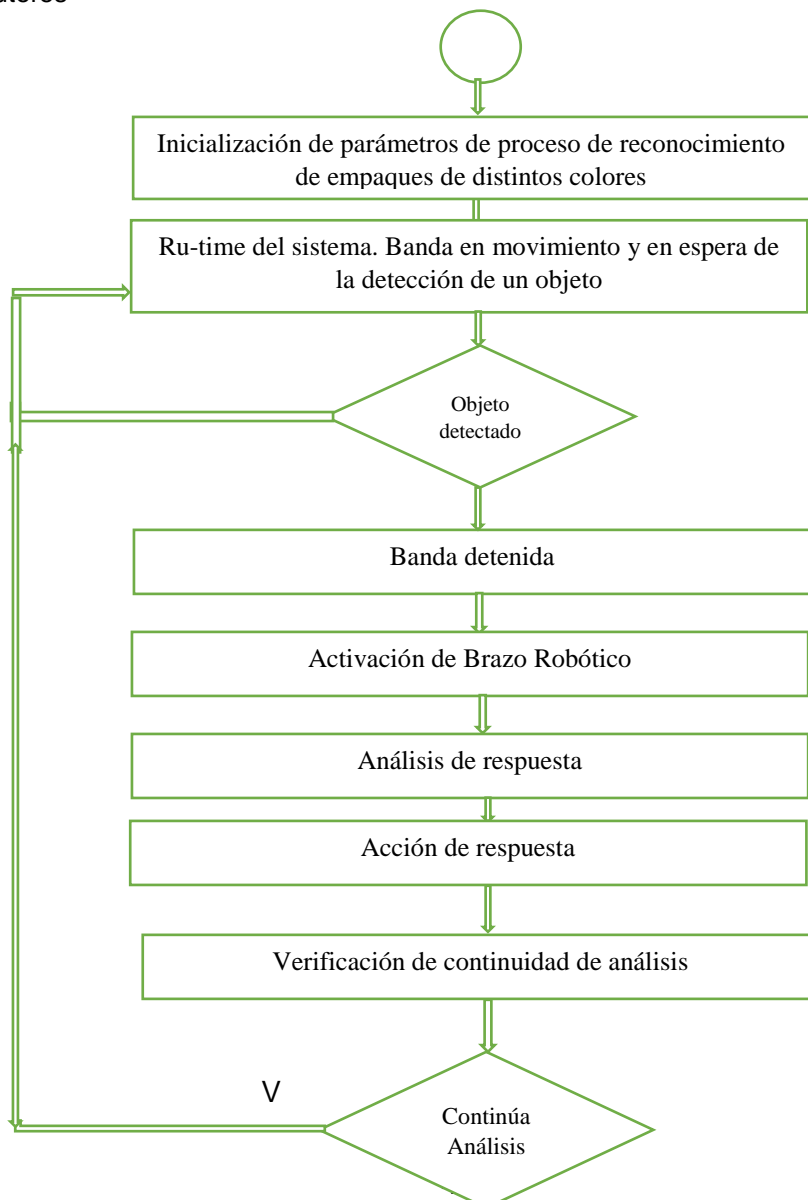
Una vez que se logró el desarrollo del sistema de visión por computadora es necesario realizar una serie de pruebas para determinar el funcionamiento del sistema por lo cual se han establecido parámetros de pruebas los cuales van desde realizar mediciones de los tiempos de respuestas del sistema junto al prototipo de brazo robótico y banda transportadora, para esto entregara gráficos con las pruebas ejercidas entregando datos reales del funcionamiento del sistema.

Al ser nuestro sistema enfocado a las técnicas de espectro de color se tiene que tener en cuenta varios factores que puedan perjudicar a la efectividad de nuestro sistema como lo indica en su trabajo (Manera, Rodriguez, Delrieux, & Coppo, 2009) pueden ser los factores climáticos, al trabajar con descomposición de matriz de colores RGB corremos el riesgo que los colores se vean afectados por un mínimo cambio de luz es por ello que la estructura en donde se montara la cámara paso por muchas pruebas de luminosidad y de distancia del objetivo a analizar.

Para nuestra primera prueba sometimos nuestro sistema comprobando la eficacia de la detección de los colores haciendo 10 pruebas en diferentes condiciones de luz estableciendo como valor singular a todos los colores que evaluamos los cuales son el rojo, verde, azul y blanco tratando que la evaluación se de en condiciones iguales se estableció un rango de sensibilidad que va desde un 0 hasta un 1.2, valores los cuales podemos ir verificando con el sistema al utilizar la herramienta de Color Spectrum, esta herramienta propia de LabVIEW nos entrega una especie de matriz donde cada vector representa a un color en donde más sube el porcentaje podemos darnos cuenta que ha ese color hace referencia, si lo ocupamos para nuestra prueba nos arrojaría en tiempo real como varia la

sensibilidad de cada color en cada condición de luz como se sabe la luz influye mucho en los colores dando a personas diferentes tonalidades pero viéndolo desde una fuente científica es normal que este fenómeno ocurra como lo indica (La, Del, & Del, 2017) en su trabajo especifica que el color que captamos cada uno los seres humanos no es el mismo visto desde los otros ojos esto se debe a la iluminación que refleje un color en específico

Fuente: Autores



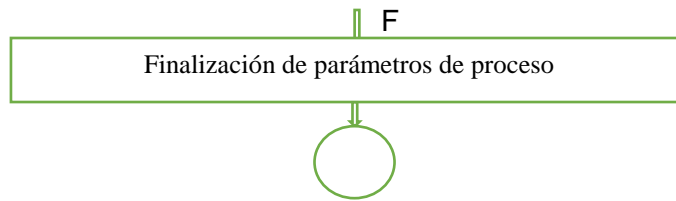


Figura 40: Diagrama de Flujo del Funcionamiento del Sistema

Pruebas individuales de sensibilidad de los colores bajo diferentes condiciones

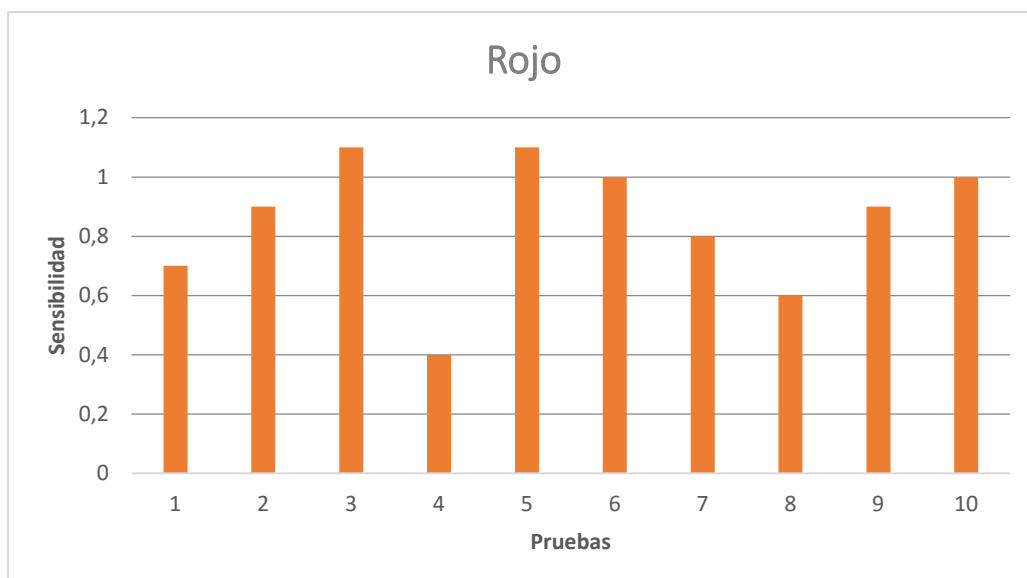
Empezamos con el color rojo el cual fue sometido a diez pruebas de sensibilidad obteniendo como resultado que de las 10 pruebas solo en la cuarta sufrió una bajada considerable de sensibilidad siendo el culpable la luz solar que en ese rato estaba demasiado fuerte, la variación de patrones de sensibilidad entrega como efecto que el color rojo tienda a no detectar bien el color cuando los parámetros son menores a un 0,4

Tabla 23: Registro de las Pruebas de Sensibilidad Color Rojo

Número de Pruebas	Sensibilidad
1	0,7
2	0,9
3	1,1
4	0,4
5	1,1
6	1
7	0,8
8	0,6
9	0,9
10	1

Fuente: Autor

Gráfico 1: Prueba de Sensibilidad del Color Rojo



Gráfica 1: Prueba de Sensibilidad del Color Rojo

Gráfica 2:

Fuente: Autores

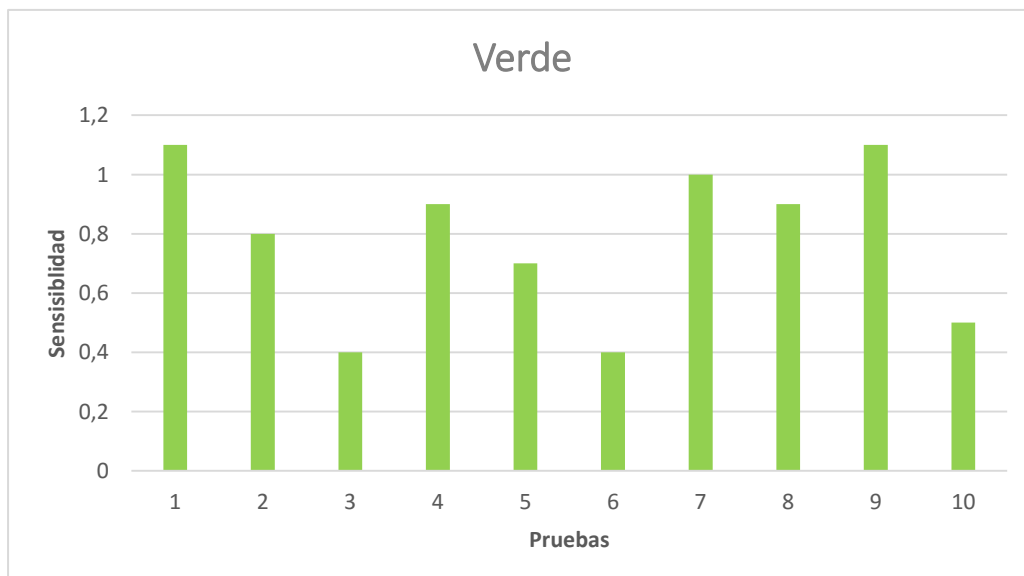
Luego de la prueba al color rojo se procedió con el color verde arrojándonos valores muy diferentes, el color rojo solo tuvo una bajada en la cuarta prueba, pero por su parte el color verde se mantuvo inestable desde el inicio marcando valores bajos en tres ocasiones dejándonos claro que el color verde en condiciones de poca luz no se puede identificar el color, por lo que hay que tener mucho cuidado con querer trabajar con este color.

Tabla 24: Registro de Prueba de Sensibilidad Color Verde

Número de Pruebas	Sensibilidad (Tiempo)
1	1,1
2	0,8
3	0,4
4	0,9
5	0,7
6	0,4
7	1
8	0,9
9	1,1
10	0,5

Fuente: Autores

Gráfico 2: Prueba de sensibilidad color verde



Fuente: Autores

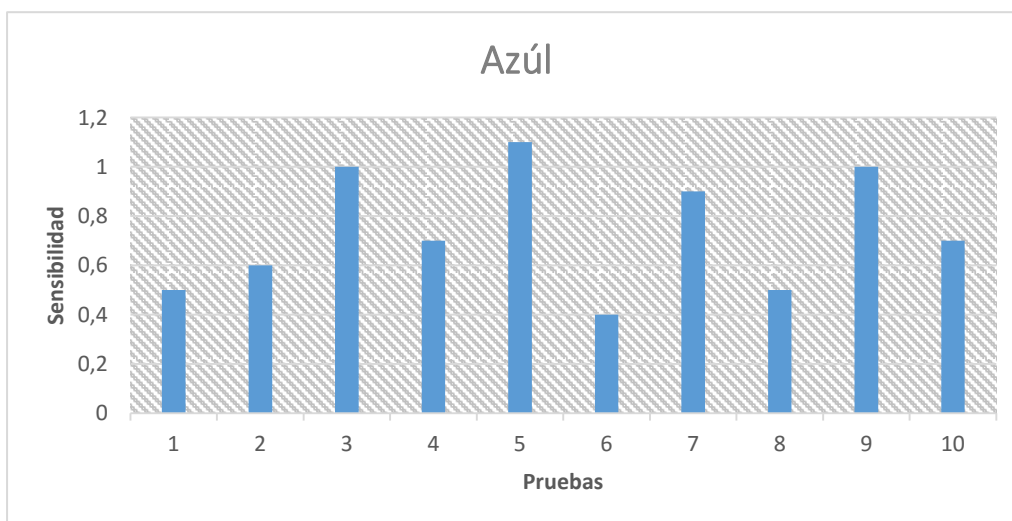
Seguido con la evaluación del color azul podemos indicar que su evaluación resulto ser muy parecida al verde, con pequeñas diferencias a las pruebas a la que se sometió, el resultado en este caso presenta un poco más de estabilidad y tiende a variar cuando recibe poca iluminación.

Tabla 25: Registro de Prueba de Sensibilidad Color Azul

Número de Pruebas	Sensibilidad (Tiempo)
1	0,5
2	0,6
3	1
4	0,7
5	1,1
6	0,4
7	0,9
8	0,5
9	1
10	0,7

Fuente: Autores

Gráfico 3: Prueba de Sensibilidad del Color Azul



Fuente: Autores

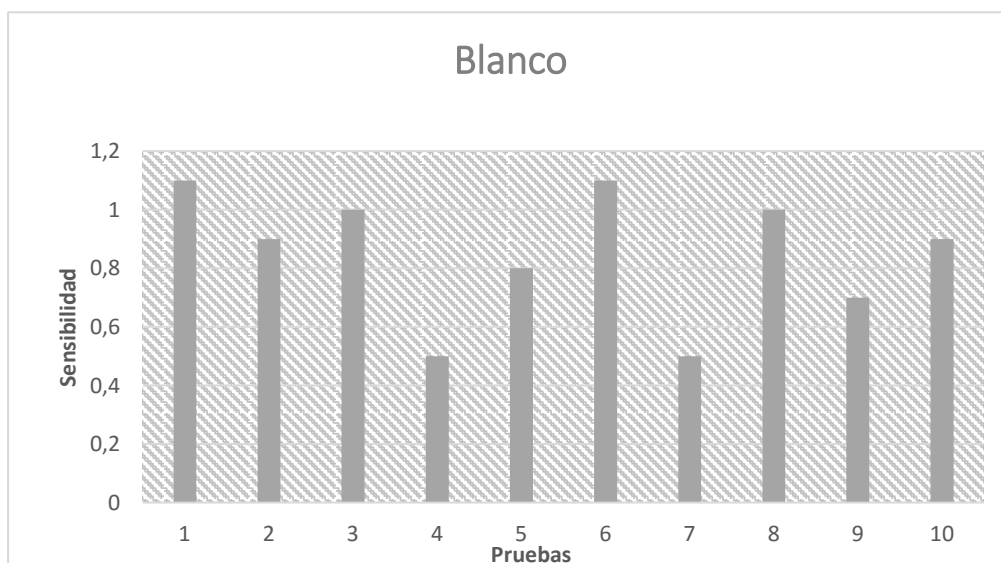
Como último color analizado tenemos al que mejor se comportó en las situaciones el cual es el color blanco comportándose en condiciones de luz bajo muy aceptable y en condiciones de luz alta mejorando por lo tanto es recomendable que si la cámara a utilizar no es de una resolución muy grande se debe colocar un tipo de luz alado de la caja para que esta pueda mantener estable el rendimiento de reconocimiento de color.

Tabla 26: Registro de Prueba de Sensibilidad Color Blanco

Número de Pruebas	Sensibilidad
1	1,1
2	0,9
3	1
4	0,5
5	0,8
6	1,1
7	0,5
8	1
9	0,7
10	0,9

Fuente: Autores

Gráfico 4: Prueba de Sensibilidad Color Blanco



Fuente: Autores

Otra de las pruebas que se realizó es la de calcular de las veces que fallan si son mayores o menores a las que acierta la identificación de color el resultado nos confirma lo que ya se había planteado desde el inicio donde se indicaba que el sistema estaba funcionando de manera óptima, las veces que hubo fallos se debió a problemas de reflejos de luz por lo que se decidió colocar una fuente de luz cercana a la cámara que ayudada a evitar los errores que se dieron en las pruebas.

Pruebas de Lectura de Colores en Tiempo Real

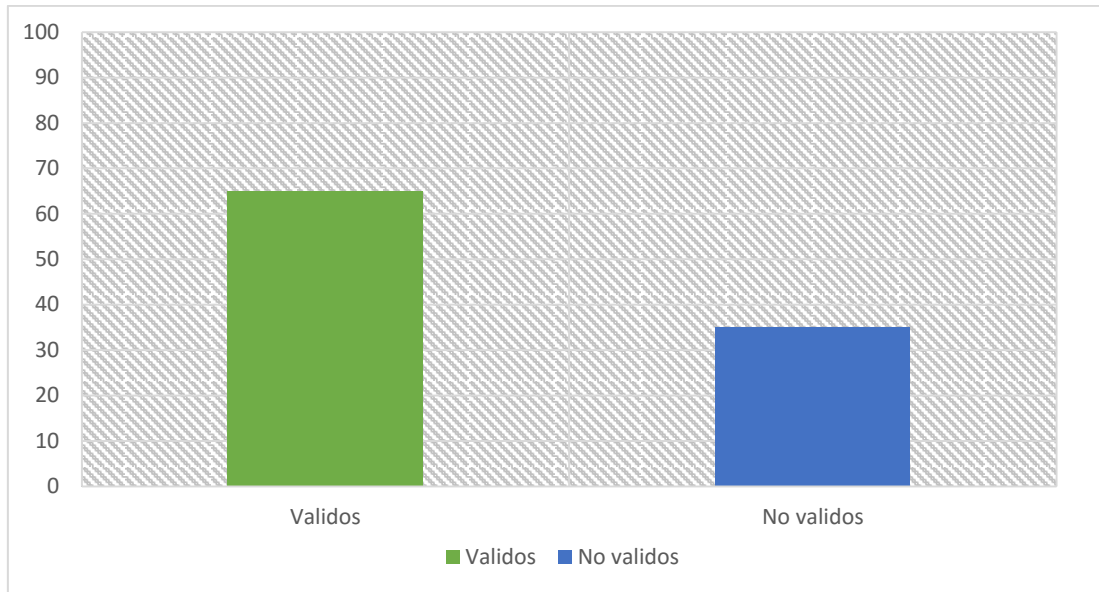
La prueba realizada consistió en medir la eficacia de detección de cada uno de los colores en tiempo real por lo que se han realizado 100 pruebas, comprendidas en 25 pruebas por cada color como se observa en la tabla 27 se han definido 2 variables (validos, no validos) realizando a los 25 muestras de cada color el análisis usando el sistema desarrollado observando que cada color tiende a tener una diferente sensibilidad lo cual hace que el sistema en condiciones ajenas no detecte el color esta prueba es de vital importancia ya que nos permitió ajustar los colores a valores que se encuentren más sensibles a cualquiera de las condiciones en la que se esté.

Tabla 27: Pruebas de Efectividad de la Detección de Colores

Total de colores	Colores Elegidos	Colores Validos Detectados con la Aplicación LabVIEW	Colores No Validos Detectados con la Aplicación LabVIEW
25	Rojo	17	8
25	Azul	16	9
25	Blanco	18	7
25	Verde	14	11

Fuente: Autores

Gráfico 5: Resultado de la Prueba de Efectividad



Fuente: Autores

CONCLUSIONES

El sistema cumplió con las expectativas y objetivos planteados dejando buenos resultados en todas las pruebas que fue sometido, cabe destacar que pese a no contar con un ordenar en óptimas condiciones el sistema se mantuvo estable en la mayoría de tiempos, la falta de luz ambiente puede ser uno de los peores inconvenientes a la hora de identificar los colores por el tipo de configuración a la que está enfocada, la toma de referencia del espectro de los colores enfocándonos en cómo obtener los colores en tiempo real con dicha técnica como la que se da a los colores RGB, ubicándolos en una cadena de vectores a cada uno de los colores, al querer distinguir los colores observamos que según su clasificación están en la categoría de colores que emiten luz es por ello que hubieron pequeños contratiempos a la hora de probar el prototipo en exteriores donde el sol pegaba directamente.

Las contribuciones que nos deja la implementación de este sistema son varias siendo necesarias nombrarlas entre las principales tenemos:

- Lograr Integrar un sistema propio con varias tecnologías que están siendo muy usadas como lo es la inteligencia Artificial
- Conseguir mantener funcional un prototipo a escala sin que presente fallas.
- Incluir bases de datos en proyectos de Automatización
- Poder implementar el prototipo en empresas locales con el fin de lograr un crecimiento económico de dichas empresas.

RECOMENDACIONES

- Realizar más pruebas en torno a los colores más usados.
- Realizar pruebas con diferentes cámaras con mayor resolución
- Crear parámetros de validaciones al detectar los colores, para que no se filtren colores falsos.
- Realizar pruebas en diferentes ambientes donde la luminosidad pueda afectar al sistema para hacer posibles correcciones.
- Probar diferentes conexiones a bases de datos utilizando conexión ODBC
- Realizar pruebas con varias figuras de muestras tales como piezas mecánicas o piezas electrónicas para comprobar si el programa las puede identificar

REFERENCIAS BIBLIOGRÁFICAS

- Adriana, L., Contreras, L., Garcia, D. F., & Calvo, A. (2014). software de diagnóstico en la percepción del dolor Diagnostic Software in color perception, *11*(2), 80-91.
- Aminoff, E. M., Toneva, M., Shrivastava, A., Chen, X., & Misra, I. (2015). Applying artificial vision models to human scene understanding, *9*(February), 1-14.
<https://doi.org/10.3389/fncom.2015.00008>
- Barrio, J. (2011). La complejidad de las mezclas sustractivas y su predicción a partir del análisis espectral, *8*, 427-436.
- Clark, S. R., Cobb, J., Kapfhammer, G. M., Jones, J. A., & Harrold, M. J. (2011). Localizing SQL faults in database applications. En *2011 26th IEEE/ACM International Conference on Automated Software Engineering, ASE 2011, Proceedings* (pp. 213-222). <https://doi.org/10.1109/ASE.2011.6100056>

- Escuela superior politécnica del litoral. (2009).
- Infinita, T., Agropecuarias, I., & Ambientales, C. (s. f.). La visión artificial y los campos de aplicación, *1*, 94-103.
- Ingenier, E. P. D. E., Por, P., & Freddy, N. P. (2015). Bach. CASTILLO ORTIZ JONATHAN.
- José, P., & Valero, S. (2006). INTRODUCCIÓN A LA ROBÓTICA.
- La, P., Del, C., & Del, C. (2017). DISEÑO DE UN SISTEMA DE VISIÓN POR ORDENADOR PARA LA CARACTERIZACIÓN DEL COLOR DEL TOMATE (*Solanum lycopersicum* L.) TRABAJO FIN DE MÁSTER.
- M, R. A. G., & E, A. A. C. (2011). ANALYSIS OF IMPLEMENTATION CONVEYOR SYSTEMS IN STORAGE YARDS IN COAL MINING COMPANIES WITH DISCRETE SIMULATION AND.
- Manera, J., Rodriguez, L., Delrieux, C., & Coppo, R. (2009). Adquisición y Procesamiento de Imágenes Aéreas para Sensado Remoto. *Ieee - Conicet*, 1-9.
- Máster, T. F. De. (2014). ESPECTRO ELECTROMAGNÉTICO Y EFICACIA DE LOS FILTROS ÓPTICOS EN PATOLOGÍAS OCULARES . Revisión bibliográfica.
- McLoughlin, M. (2009). Synoptic pericope order. *Ephemerides Theologicae Lovanienses*, 85(1), 71-97. <https://doi.org/10.15446/acag.v64n3.42657>
- Miguel, J., Somoza, G., Antonio, J., Josemiguelmerinouclmes, J. M. M., Rafaelvillenaaluclmes, R. V., & Somoza, J. A. G. (2018). Análisis del efecto de la robótica en la motivación de estudiantes de tercero de Educación Primaria durante la resolución de tareas de interpretación de planos tasks of interpretation of plans, 2.
- Odorico, A. H., & Lage, F. (s. f.). Didáctica en Robótica : Un software de simulación para aprendizaje significativo de la localización espacial Resumen.
- Pajares, G., & de la Cruz, J. (2007). Visión por computador: Imágenes digitales y aplicaciones. *Editorial Ra-ma*, 4, 102-105.
- Pérez González, C. (2016). Detección y seguimiento de objetos por colores en una plataforma Raspberry Pi, 80.
- Quiñones, C., & Bernal, M. (2011). LabVIEW y la instrumentación virtual aplicados a la docencia y la investigación en ciencias básicas.
- Rivera-mujica, E., Huamaní-navarrete, P., & Palma, U. R. (2018). Aplicación de la Lógica Difusa en el Control de un Proceso de Prototipo de Pintado, (July), 19-21.
- Robótico, B., Sistema, D. E. C. O. N., Por, M., Usuario, E. L., Fines, P., En, I., ... Bravo, S. (2015). Pontificia universidad católica del Perú.
- Sucar, L. E. (s. f.). Visi ´ on Computacional Giovanni G ´ Helmholtz Zentrum Munchen.
- This, D., Affairs, L., Question, E. S., & Register, T. (s. f.). general public Table of contents.
- Titulación, P. D. E., Xavier, H., & Zatzabal, M. (2016). UNIVERSIDAD DE GUAYAQUIL.

ANEXOS

ANEXOS 1: Tabla de Costo de Implementación del Sistema

Tabla 28: Costo de Implementación del Sistema en Industrias

	DESCRIPCIÓN	CANTIDA D	UNIDAD	VALOR UNITARIO	VALOR TOTAL
1	Ingeniero	1	Desarrollador	\$ 1200.00	\$ 1200.00
2	Transporte	25	Viajes	\$ 5	\$ 125.00
3	Papelería	550	Hojas	\$ 0.10	\$ 55.00
4	Impresiones	200	Hojas	\$ 0.30	\$ 60.00
5	Piezas para construcción				
5.1	Motor Eléctrico de 1 HP	1	Pieza	\$ 260.00	\$ 260.00

5.2	Extractor industrial 254 mm	1	Pieza	\$ 165.00	\$ 165.00
5.3	Polea en V diámetro 131,4 mm	1	Pieza	\$ 45.00	\$ 45.00
5.4	Polea en V diámetro 76,2 mm	1	Pieza	\$ 28.00	\$ 28.00
5.5	Chumaceras y rodamientos	2	Pieza	\$ 48.00	\$ 96.00
5.6	Malla N 16	2	Pieza	\$ 35.00	\$ 70.00
5.7	Resortes	4	Pieza	\$ 6.50	\$ 26.00
5.8	Eje con excéntrica	1	Pieza	\$ 64.00	\$ 64.00
5.9	Ángulos metálicos y laminas	40	Kg	\$ 3.60	\$ 144.00
5.10	Consumibles	varios	Pieza	\$ 87.00	\$ 87.00
6	Hardware				
6.1	Computador de 7ma generación	1	Computo	\$ 600.00	\$ 600.00
6.2	Monitor de 17 pulgadas	1	Computo	\$ 150.00	\$ 150.00
6.3	Cámara lineal	1	Computo	\$ 3745.00	\$ 3745.00
6.4	Tarjeta de adquisición de imágenes	1	Computo	\$ 255.00	\$ 255.00
6.5	Controlador Industrial	1	Computo	\$ 2715.00	\$ 2715.00
7	Software				
7.1	Labview Profesional	1	Licencia	\$ 5500.00	\$ 5500.00
7.2	Módulo Visión Development	1	Licencia	\$ 4210.00	\$ 4210.00
7.3	Modulo Real-Time	1	Licencia	\$ 2960.00	\$ 2960.00
8	Complemento				
8.1	Brazo robótico	1	Mecánico	\$8000.00	\$8000.00
9	Mano de Obra	1	Mecánico	\$ 900.00	\$ 900.00
TOTAL					\$ 32080.00

Fuente: Autores

ANEXOS 2: Figura de las Comunicaciones del Sistema con el Brazo y Banda

Fuente: Autores

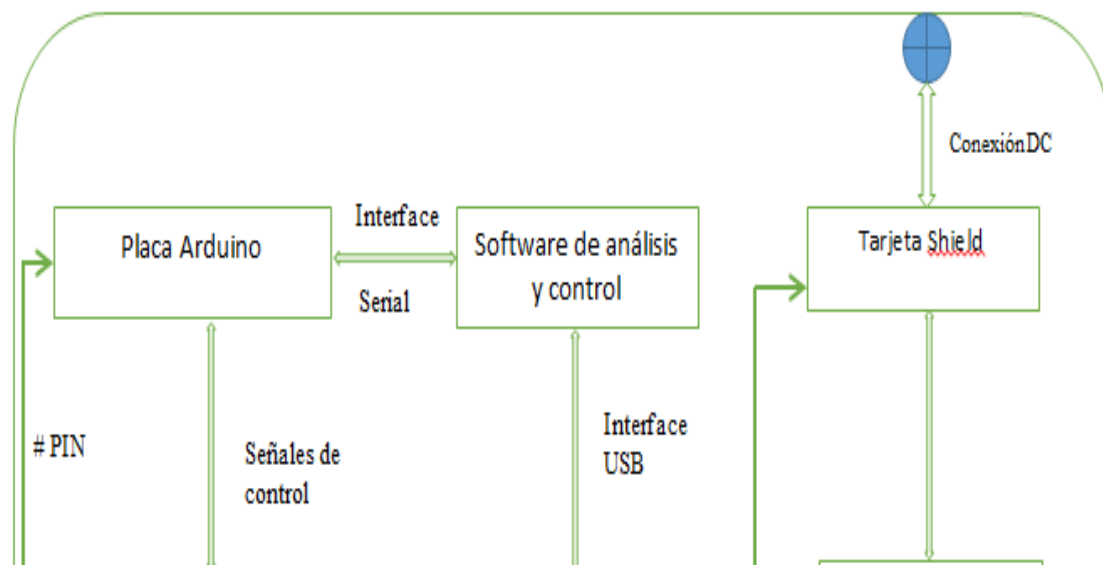


Figura 41: Comunicación del Sistema con el Brazo y la Banda

ANEXOS 3: Conexiones del Brazo Robótico a la Placa Arduino y Shiel

Al contar el brazo Robótico con 6 servo motores se requirió de 6 pines libres dedicados para la conexión con Arduino por lo cual se procedió a utilizar la shield como conector de energía para los servos dejando libre solo a la conexión que es para pin a continuación se detalla cómo está conectado la estructura de brazo robótico con la shield.

Fuente: Autores

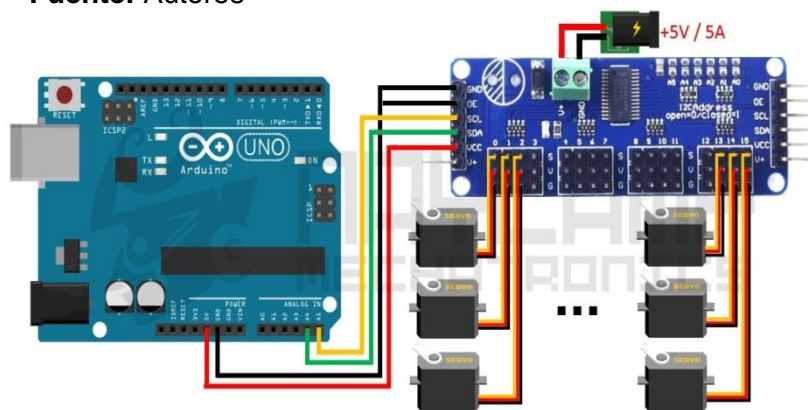


Figura 42: Conexión de los Servos del Brazo Robótico

Como se puede apreciar en las conexiones todos los componentes funcionales van conectados a la placa Shield quien alimentara con el voltaje correcto a cada uno de los servomotores

Fuente: Autores

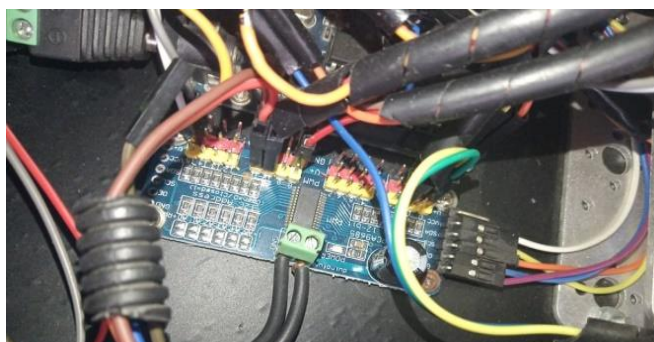


Figura 43: Conexión Física del Brazo Robótica

Fuente: Autores

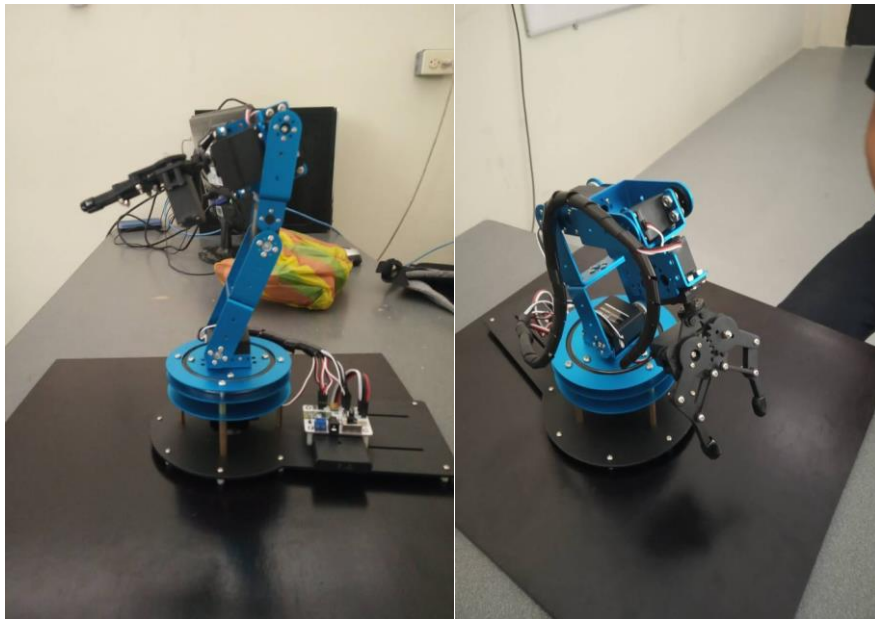


Figura 44: Brazo Robótico Conectado a la Placa Shield y Arduino

ANEXOS 4: Manual de Usuario para uso del Sistema

Lo primero que se deber hacer al querer trabajar con el sistema será abrir el menú principal donde contara con 3 opciones las cuales son:

1. Detección de Colores
2. Informe de Registros
3. Control Manual Brazo

Cada uno de estos botones cumple una acción en específico, abrirla interfaz correspondiente que se requiera, al darle clic sobre el botón de la primera opción nos aparecerá la ventana del Sistema de detección de Colores.

Fuente: Autores

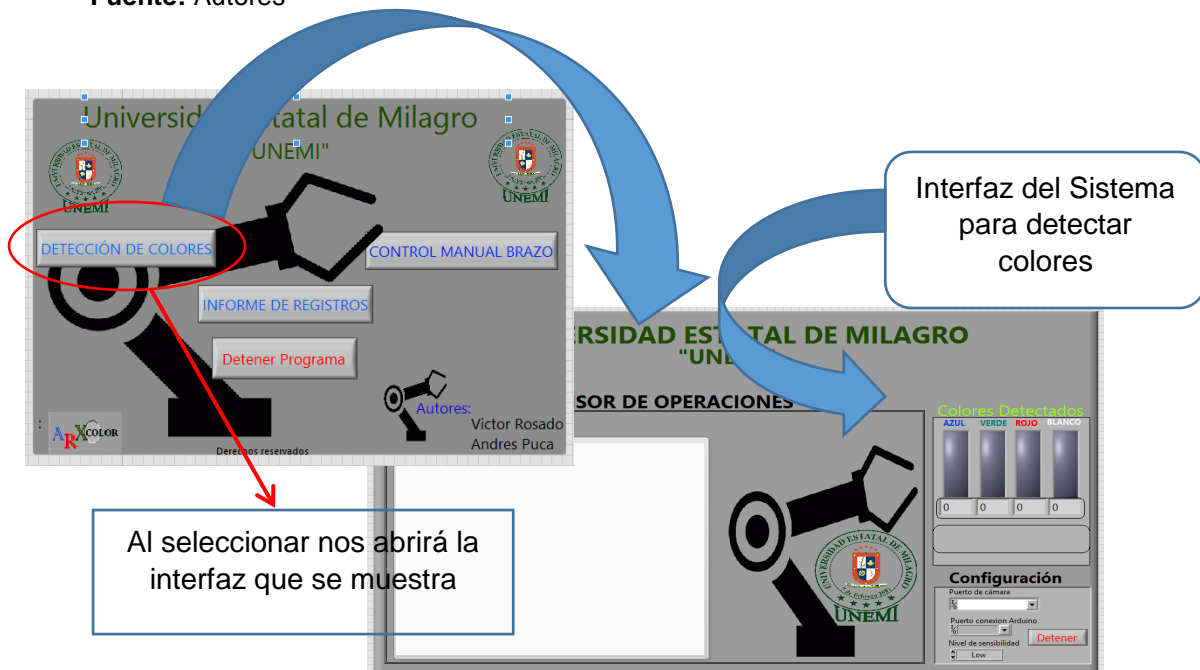
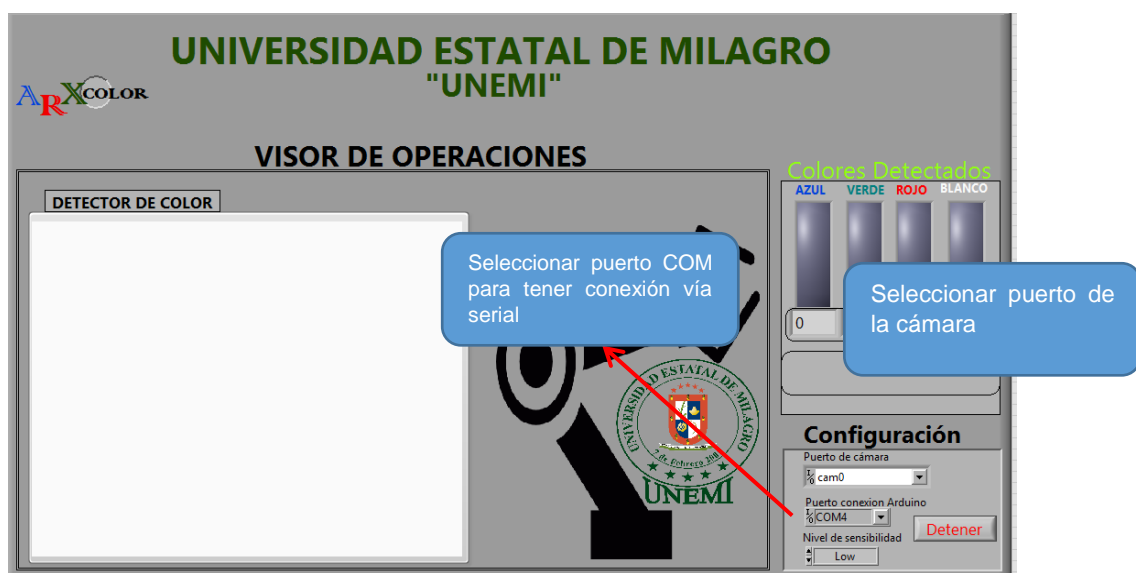


Figura 45: Interfaz de Usuario

Luego que estemos en la Interfaz del sistema debemos realizar las configuraciones de conexión con la cámara y también a la placa Arduino, dichas opciones se encuentran ubicadas en un parte muy sencillas de realizar las configuraciones iniciales.

Fuente: Autores



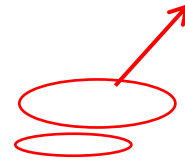


Figura 46: Configuración de puestos para la Conexión

Para iniciar a la ejecución del programa bastara con darle clic en la parte de superior del programa donde tiene un pericido a una flecha

Fuente: Autores



Figura 47: Campos de Controles para Encendido y Apagado del Sistema

Al iniciar el sistema la parte donde se encuentra un display visualizará lo que se esté grabando en ese momento para que el sistema realice la verificación de un color se debe seleccionar un área en la pantalla esta servirá para fijar un punto específico en donde los colores al entrar en esa área el sistema tendrá la cualidad de verificar que color es.

Fuente: Autores



ROI.-Área dedicada para la detección de colores

Figura 48: Determinando el Área que detectara los colores

Como último punto el usuario debe conocer el área donde podrá visualizar los colores que ha detectado en la parte derecha se ha colocado uno visores que al detectar dichos colores se encenderán y a su vez en su parte superior marcara las veces que ha detectado dichos colores.

Fuente: Autores



Figura 49: Panel Indicador de Colores Detectados

ANEXOS 5: Código Arduino para Comunicación con el Brazo Robótico y Banda Transportadora

La codificación realizada en Arduino tiene como fin conectarse con labVIEW haciendo uso de sus librerías que permitan una rápida interacción mediante conexión serial donde llegaran en forma de código ASCII y en compilador lo sabrá entender y haciendo que se active lo establecido mediante código.

Código

```
#define SERVO_BASE 7 //BASE
#define SERVO_HOMB 9 //HOMBRO
#define SERVO_COD1 10 // CODO1
#define SERVO_COD2 11 // CODO2
#define SERVO_MAN 12 // MANO
#define SERVO_GRIP 13 //GRIPPER
#define rotor 6 //GRIPPER
#include <Servo.h>

Servo servobase; // crea objeto con las propiedades de la libreria Servo
Servo servohombro; // crea objeto con las propiedades de la libreria Servo
Servo servocodo1; // crea objeto con las propiedades de la libreria Servo
Servo servocodo2; // crea objeto con las propiedades de la libreria Servo
Servo servomano; // crea objeto con las propiedades de la libreria Servo
Servo servogripper; // crea objeto con las propiedades de la libreria Servo
Servo servorotor;

void setup() {
  //-----Definimos Los Pines a Usar-----
  servobase.attach(SERVO_BASE); // pin 7 base
  servohombro.attach(SERVO_HOMB); // pin 9 hombro
  servocodo1.attach(SERVO_COD1); // pin 10 codo 1
  servocodo2.attach(SERVO_COD2); // pin 11 codo 2
  servomano.attach(SERVO_MAN); // pin 12 mano
  servogripper.attach(SERVO_GRIP); // pin 13 gripper
  servorotor.attach(rotor);

  //-----Creamos la Conexion Serial-----
  Serial.begin(9600);
}

void loop() {

  //----- Establecemos valores de arranque-----
  servobase.write(90);
  servohombro.write(90);
  servocodo1.write(95);
  servocodo2.write(180);
  servomano.write(180);
  servogripper.write(40);

  inicio();

}

//-----Creamos funciones para los Angulos-----
void control_base_derecha(int angulomin, int angulomax)
{
  for (int angulo = angulomin; angulo < angulomax; angulo++) // 5 150
  { // un ciclo para mover el servo entre los 0 y los 180 grados
```



```

servobase.write(angulo);          // manda al servo la posicion
delay(15);                        // espera unos milisegundos para que el servo llegue a su posicion
}
delay(500);
}
//_____

void control_base_izquierda(int angulomin, int angulomax)
{
for (int angulo = angulomax; angulo > angulomin; angulo --) //150 5
{ // un ciclo para mover el servo entre los 0 y los 180 grados
servobase.write(angulo);          // manda al servo la posicion
delay(15);                        // espera unos milisegundos para que el servo llegue a su posicion
}
delay(500);
}
//_____

void control_hombro_arriba(int angulomin, int angulomax)
{
for (int angulo = angulomin; angulo < angulomax; angulo ++) // 5 65
{ // un ciclo para mover el servo entre los 0 y los 180 grados
servohombro.write(angulo);        // manda al servo la posicion
delay(20);                        // espera unos milisegundos para que el servo llegue a su posicion
}
delay(500);
}
//_____

void control_hombro_abajo(int angulomin, int angulomax)
{
for (int angulo = angulomin; angulo > angulomax; angulo --) // 5 65
{ // un ciclo para mover el servo entre los 0 y los 180 grados
servohombro.write(angulo);        // manda al servo la posicion
delay(20);                        // espera unos milisegundos para que el servo llegue a su posicion
}
delay(500);
}
//_____

void control_codo1_abajo(int angulomin, int angulomax)
{
for (int angulo = angulomin; angulo < angulomax; angulo ++) // 5 65
{ // un ciclo para mover el servo entre los 0 y los 180 grados
servocodo1.write(angulo);         // manda al servo la posicion
delay(15);                        // espera unos milisegundos para que el servo llegue a su posicion
}
delay(500);
}
//_____

void control_codo1_arriba(int angulomin, int angulomax)
{
for (int angulo = angulomax; angulo > angulomin; angulo --) // 65 5
{ // un ciclo para mover el servo entre los 0 y los 180 grados
servocodo1.write(angulo);         // manda al servo la posicion
delay(15);                        // espera unos milisegundos para que el servo llegue a su posicion
}
delay(500);
}
}

```

```

// _____

void control_codo2_abajo(int angulomin, int angulomax)
{
  for (int angulo = angulomin; angulo < angulomax; angulo ++) // 5 65
  { // un ciclo para mover el servo entre los 0 y los 180 grados
    servocodo2.write(angulo); // manda al servo la posicion
    delay(15); // espera unos milisegundos para que el servo llegue a su posicion
  }
  delay(500);
}
// _____

void control_codo2_arriba(int angulomin, int angulomax)
{
  for (int angulo = angulomax; angulo > angulomin; angulo --) // 65 5
  { // un ciclo para mover el servo entre los 0 y los 180 grados
    servocodo2.write(angulo); // manda al servo la posicion
    delay(15); // espera unos milisegundos para que el servo llegue a su posicion
  }
  delay(500);
}
// _____

void control_mano_derecha(int angulomin, int angulomax)
{
  for (int angulo = angulomin; angulo < angulomax; angulo ++)
  { // un ciclo para mover el servo entre angulo min 20 y nagulo max 70
    servomano.write(angulo); // manda al servo la posicion
    delay(15); // espera unos milisegundos para que el servo llegue a su posicion
  }
  delay(500);
}
// _____

void control_mano_izquierda(int angulomin, int angulomax)
{
  for (int angulo = angulomax; angulo > angulomin; angulo --)
  { // un ciclo para mover el servo entre 20 y 70
    servomano.write(angulo); // manda al servo la posicion
    delay(15); // espera unos milisegundos para que el servo llegue a su posicion
  }
  delay(500);
}
// _____

void control_gripper_cerrar(int angulomin, int angulomax)
{
  for (int angulo = angulomin; angulo < angulomax; angulo ++)
  { // un ciclo para mover el servo entre angulo min 20 y nagulo max 70
    servogripper.write(angulo); // manda al servo la posicion
    delay(20); // espera unos milisegundos para que el servo llegue a su posicion
  }
  delay(500);
}
// _____

void control_gripper_abrir(int angulomin, int angulomax)
{

```

```

for (int angulo = angulomin; angulo > angulomax; angulo --)
{ // un ciclo para mover el servo entre angulo min 20 y angulo max 70
  servogripper.write(angulo); // manda al servo la posicion
  delay(20); // espera unos milisegundos para que el servo llegue a su posicion
}
delay(500);
}

```

```

void inicio()
{
  if (Serial.available()>0)//Constatamos que tenemos conexion entre Arduino y Labview
  {
    char inChar = Serial.read();//Enviamos la variable que esta leyendo el puerto serie

    if (inChar == 'r')// Preguntamos si es la letra definida para un color y de ser este caso entraria las
    funciones para el color rojo
    {
      colorrojo();
      control_codo1_abajo(70, 100); //angulo mini y angulo maximo
      control_gripper_abrir(180, 10); //angulo mini y angulo maximo
      control_codo2_abajo(100, 130); //angulo mini y angulo maximo
      control_mano_derecha(30, 90); //angulo mini y angulo maximo
      control_hombro_abajo(140, 60); //angulo mini y angulo maximo
      control_gripper_cerrar(10, 180); //angulo mini y angulo maximo
      control_hombro_arriba(60, 150); //angulo mini y angulo maximo
      control_base_izquierda(30, 140); //angulo mini y angulo maximo
      control_hombro_abajo(140, 60);
      control_hombro_abajo(140, 60);
      control_gripper_abrir(180, 10);
      control_hombro_arriba(60, 150);
    }else if (inChar=='s'){
      servobase.write(90);
      servohombro.write(90);
      servocodo1.write(95);
      servocodo2.write(180);
      servomano.write(180);
      servogripper.write(40);
    }

    if (inChar == 'v')// Preguntamos por el color Verde
    {
      colorverde();
      control_codo1_abajo(90, 125); //angulo mini y angulo maximo
      control_gripper_abrir(180, 10); //angulo mini y angulo maximo
      control_codo2_abajo(120, 130); //angulo mini y angulo maximo
      control_mano_derecha(30, 90); //angulo mini y angulo maximo
      control_hombro_abajo(140, 60); //angulo mini y angulo maximo
      control_gripper_cerrar(10, 180); //angulo mini y angulo maximo
      control_hombro_arriba(60, 150); //angulo mini y angulo maximo
      control_base_derecha(50, 150); //angulo mini y angulo maximo
      control_hombro_abajo(140, 60);
      control_hombro_abajo(140, 60);
      control_gripper_abrir(180, 10);
      control_hombro_arriba(60, 150);
    }else if (inChar=='t'){
      servobase.write(90);
      servohombro.write(90);
    }
  }
}

```

```

servocodo1.write(95);
servocodo2.write(180);
servomano.write(180);
servogripper.write(40);
}

if (inChar == 'z')// Preguntamos por el color Azul
{
//control_base_izquierda(30, 160);
colorazul();
control_codo1_abajo(90, 125); //angulo mini y angulo maximo
control_gripper_abrir(180, 10); //angulo mini y angulo maximo
control_codo2_abajo(120, 130); //angulo mini y angulo maximo
control_mano_derecha(30, 90); //angulo mini y angulo maximo
control_hombro_abajo(140, 60); //angulo mini y angulo maximo
control_gripper_cerrar(10, 180); //angulo mini y angulo maximo
control_hombro_arriba(60, 150); //angulo mini y angulo maximo
control_base_izquierda(50, 145); //angulo mini y angulo maximo
control_hombro_abajo(140, 60);
control_hombro_abajo(140, 60);
control_gripper_abrir(180, 10);
control_hombro_arriba(60, 150);
}else if (inChar=='o'){
servobase.write(90);
servohombro.write(90);
servocodo1.write(95);
servocodo2.write(180);
servomano.write(180);
servogripper.write(40);
}

if (inChar == 'b')//Preguntamos por el color Blanco
{
colorblanco();
control_codo1_abajo(90, 125); //angulo mini y angulo maximo
control_gripper_abrir(180, 10); //angulo mini y angulo maximo
control_codo2_abajo(120, 130); //angulo mini y angulo maximo
control_mano_derecha(30, 90); //angulo mini y angulo maximo
control_hombro_abajo(140, 60); //angulo mini y angulo maximo
control_gripper_cerrar(10, 180); //angulo mini y angulo maximo
control_hombro_arriba(60, 150); //angulo mini y angulo maximo
control_base_derecha(10, 160); //angulo mini y angulo maximo
control_hombro_abajo(140, 60);
control_hombro_abajo(140, 60);
control_gripper_abrir(180, 10);
control_hombro_arriba(60, 150);
}else if (inChar=='p'){
servobase.write(90);
servohombro.write(90);
servocodo1.write(95);
servocodo2.write(180);
servomano.write(180);
servogripper.write(40);
}

}
}
void colorrojo(){
servorotor.write(25);

```

```
}  
  
void colorverde(){  
    servidorotor.write(45);  
}  
void colorazul(){  
    servidorotor.write(65);  
}  
void colorblanco(){  
    servidorotor.write(95);  
}
```



Figura 50: Acompañamiento académico