



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO AUTOMATIZADO
DE CLASIFICACIÓN MEDIANTE PROTOCOLO MODBUS TCP/IP**

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

AUTORES:

LUIS ADRIÁN ARCE SOLANO
ANTHONY JOSUÉ GUAMÁN GUALOTO

TUTOR:

Ing. RAFAEL FRANCO REINA, MSc

Guayaquil – Ecuador

2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Anthony Josué Guamán Gualoto con documento de identificación N° 0958394298 y Luis Adrián Arce Solano con documento de identificación N° 0955318480, manifestamos que:

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

Guayaquil, 28 de febrero del año 2025.

Atentamente,

Anthony Guamán

Anthony Josué Guamán Gualoto
095839429-8

Luis Arce

Luis Adrián Arce Solano
095531848-0

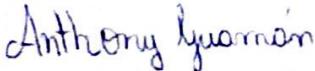
**CERTIFICADO DE SESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

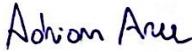
Nosotros, Anthony Josué Guamán Gualoto con documento de identificación N° 0958394298 y Luis Adrián Arce Solano con documento de identificación N° 0955318480, manifestamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: "DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO AUTOMATIZADO DE CLASIFICACIÓN MEDIANTE PROTOCOLO MODBUS TCP/IP", el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

Guayaquil, 28 de febrero del año 2025.

Atentamente,


Anthony Josué Guamán Gualoto
095839429-8


Luis Adrián Arce Solano
095531848-0

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Rafael Franco Reina con documento de identificación N°0923328629, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO AUTOMATIZADO DE CLASIFICACIÓN MEDIANTE PROTOCOLO MODBUS TCP/IP”, realizado por Guamán Gualoto Anthony Josué con documento de identificación N° 0958394298 y Arce Solano Luis Adrián con documento de identificación N° 0955318480, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 28 de febrero del año 2025.

Atentamente,



Ing. Rafael Franco Reina, MSc
C.I.: 0923328629

DEDICATORIA

A mi familia que ha sido parte fundamental de mi desarrollo académico y personal, por ser mi refugio constante, mi fuerza y mi mayor fuente de inspiración. A mis amigos, por acompañamiento y deseos que brindaron para sacar adelante este proyecto.

A mis profesores, por su orientación y enseñanzas que me han permitido crecer tanto académica y profesionalmente. Gracias por brindarme su conocimiento y por desafiarme a pensar de manera crítica, lo que ha sido clave en la realización de esta tesis.

Arce Solano Luis Adrián

DEDICATORIA

A mis padres y hermanas por ser el pilar fundamental de mi vida. Gracias por su amor, sacrificio y apoyo incondicional a lo largo de todos estos años. Su confianza en mí, incluso en los momentos de duda, me ha dado la fuerza para seguir adelante y alcanzar este logro. Sin ustedes, este sueño no habría sido posible. Cada sacrificio, cada palabra de aliento, cada gesto de cariño han sido la base que me ha permitido llegar hasta aquí.

A mis profesores y tutores, por su dedicación, paciencia y por guiarme en cada paso de este proceso académico. Gracias por compartir su conocimiento y por enseñarme a pensar de manera crítica, a cuestionar, a investigar y a no conformarme nunca con lo primero que se me presenta. Ustedes han sido esenciales en mi formación profesional y personal.

Este proyecto es el resultado de un esfuerzo colectivo y, por ello, lo dedico con todo mi corazón a quienes han sido mi fuente constante de inspiración. A ustedes, gracias.

Guamán Gualoto Anthony Josué

RESUMEN

Este proyecto de titulación se centra en el diseño e implementación de un prototipo automatizado para la clasificación de objetos basado en el protocolo de comunicación industrial Modbus TCP/IP. El sistema está compuesto por un controlador principal, sensores, y actuadores que permiten identificar y clasificar los objetos según criterios predefinidos como tamaño, peso y color. Para la comunicación entre los dispositivos involucrados, se utiliza Modbus TCP/IP, una tecnología ampliamente empleada en la automatización industrial, que asegura una transmisión de datos confiable.

La implementación del prototipo consta del desarrollo de una interfaz gráfica HMI en TIA Portal para el monitoreo y control del sistema, facilitando su operación y comprensión. Se utiliza un Arduino con un módulo Ethernet ENC28J60 para la comunicación con un PLC mediante el protocolo Modbus TCP/IP. El sistema integra sensores específicos para la clasificación de objetos: un sensor de color TCS3200 para identificar los colores rojo, azul y verde; un sensor ultrasónico HC-SR04 para determinar el tamaño de las cajas. Además, se ha incorporado un sensor de proximidad para detectar la presencia de los objetos. La clasificación se realiza mediante un servomotor S3003, activado según las condiciones predefinidas en las recetas configuradas en el HMI.

El proyecto ofrece una solución adaptable a líneas de producción, facilitando futuras mejoras en automatización. Además, fortalece el aprendizaje práctico en Ingeniería Electrónica mediante el uso de tecnologías y estándares industriales.

Palabras Clave: MODBUS TCP/IP, Automatización, Clasificación, Sensores, Banda transportadora.

ABSTRACT

This thesis project focuses on the design and implementation of an automated prototype for object classification based on the Modbus TCP/IP industrial communication protocol. The system consists of a main controller, sensors, and actuators that enable the identification and classification of objects according to predefined criteria such as size, weight, and color. Modbus TCP/IP, a widely used technology in industrial automation, is employed to ensure reliable data transmission between the devices involved.

The prototype implementation includes the development of an HMI graphical interface in TIA Portal for system monitoring and control, facilitating its operation and understanding. An Arduino with an ENC28J60 Ethernet module is used to communicate with a PLC via the Modbus TCP/IP protocol. The system integrates specific sensors for object classification: a TCS3200 color sensor to identify red, blue, and green colors; an HC-SR04 ultrasonic sensor to determine box sizes. Additionally, a proximity sensor has been incorporated to detect the presence of objects. The classification is performed using an S3003 servo motor, activated according to predefined conditions in the recipes configured in the HMI.

This project provides a solution adaptable to production lines, enabling future improvements in automation. Moreover, it strengthens practical learning in Electronic Engineering through the use of industrial technologies and standards.

Keywords: MODBUS TCP/IP, Automation, Classification, Sensors, Conveyor belt.

ÍNDICE

RESUMEN.....	1
ÍNDICE	3
ÍNDICE DE FIGURAS.....	6
ÍNDICE DE TABLAS.....	11
I INTRODUCCIÓN.....	1
II PROBLEMA.....	2
III OBJETIVOS	3
3.1 Objetivo general	3
3.2 Objetivos específicos	3
IV FUNDAMENTO TEÓRICO	4
4.1 Protocolo MODBUS.....	4
4.1.1 MODBUS TCP/IP	5
4.1.2 MODBUS RTU	5
4.1.3 MODBUS PLUS	6
4.1.4 MODBUS ASCII	6
4.2 Banda transportadora	7
4.2.1 Soporte de cinta	8
4.2.2 Poleas	8
4.2.3 Unidad de accionamiento.....	9
4.3 TIA Portal V18.....	10
4.3.1 Bloque de organización OB	11

4.3.2	Bloque de función FB	12
4.3.3	Función FC	12
4.3.4	Bloque de datos DB	13
4.4	Arduino IDE	13
4.5	Arduino UNO	15
4.6	Sensor de color TCS3200	15
4.7	Sensores Ultrasónico HC-SR04	17
4.8	Servomotor	18
4.9	Módulo Ethernet ENC28J60	20
4.10	Módulo LM2596	22
4.11	Sensor inductivo	22
V	MARCO METODOLÓGICO	24
5.1	Enfoque de la investigación	24
5.2	Diseño y planificación del prototipo	25
5.3	Implementación del prototipo	27
5.3.1	Armado de la banda transportadora	27
5.3.2	Implementación de sensores y actuadores	29
5.3.3	Conexión y configuración del motor trifásico	31
5.4	Programación del Arduino	34
5.4.1	Bibliotecas utilizadas	34
5.4.2	Declaración de variables y configuración de pines	35

5.4.3 Configuración Setup()	36
5.4.4 Ejecución en loop()	38
5.5 Programación del TIA Portal	40
5.6 Diseños de las interfaces del HMI	45
5.6.1 Tabla de variables del HMI	45
5.5.2 Pantalla de presentación	46
5.5.3 Ventana de selección de interfaces	47
5.5.3 Pantalla de usuarios	47
5.5.4 Pantalla de parámetros de productos	48
5.5.5 Pantalla de control del conveyor	49
5.5.6 Pantalla de Proceso	49
VI ANÁLISIS DE RESULTADOS	51
6.1 Evaluación del sistema	51
6.2 Prueba del sensor ultrasónico en el prototipo	57
6.2 Prueba del sensor de color en el prototipo	59
6.3 Prueba del servomotor en el prototipo	61
6.4 Prueba del sensor inductivo en el prototipo	63
6.5 Pruebas de recetas	64
6.5.1 Receta 1: Caja grande roja	64
6.5.2 Receta 2: Caja pequeña azul	65
6.5.3 Receta 3: Caja grande verde	66

6.5.4 Receta 4: Caja pequeña roja	67
6.5.5 Receta 5: Caja grande azul.....	68
6.5.6 Receta 6: Caja pequeña verde.....	69
6.6 Tabla de resultados	71
VII PRESUPUESTO	80
VIII CRONOGRAMA	81
IX CONCLUSIONES	82
X RECOMENDACIONES	83
XI BIBLIOGRAFÍA	84
XII ANEXOS	89

ÍNDICE DE FIGURAS

<i>Figura 1 Arquitectura protocolo MODBUS</i>	4
<i>Figura 2 Ejemplo de banda transportadora industrial</i>	7
<i>Figura 3 Ejemplo de soporte con su cinta</i>	8
<i>Figura 4 Estructura interna de una polea</i>	9
<i>Figura 5 Ejemplo de unidad de accionamiento</i>	10
<i>Figura 6 Entorno de desarrollo del TIA Portal</i>	11
<i>Figura 7 Bloque OB</i>	11
<i>Figura 8 Bloque FB</i>	12
<i>Figura 9 Función FC</i>	13
<i>Figura 10 Bloque de datos DB</i>	13
<i>Figura 11 Entorno de desarrollo de Arduino IDE.</i>	14

<i>Figura 12 Ejemplo de placa Arduino UNO.....</i>	<i>15</i>
<i>Figura 13 Ejemplo de sensor TCS3200.....</i>	<i>16</i>
<i>Figura 14 Ejemplo de tonalidades del sensor TCS3200</i>	<i>16</i>
<i>Figura 15 Diagrama de bloques del sensor de color</i>	<i>17</i>
<i>Figura 16 Estructura de un sensor ultrasónico.....</i>	<i>18</i>
<i>Figura 17 Servomotor S3003 360°.....</i>	<i>19</i>
<i>Figura 18 Estructura interna de un servomotor.....</i>	<i>20</i>
<i>Figura 19 Módulo ethernet mini ENC28J60</i>	<i>21</i>
<i>Figura 20 Conexión módulo Ethernet con Arduino UNO.....</i>	<i>21</i>
<i>Figura 21 Módulo Step-down LM2596.....</i>	<i>22</i>
<i>Figura 22 Sensor inductivo.....</i>	<i>23</i>
<i>Figura 23 Estructura del proyecto</i>	<i>24</i>
<i>Figura 24 Vista frontal del prototipo.....</i>	<i>26</i>
<i>Figura 25 Vista lateral del prototipo.....</i>	<i>26</i>
<i>Figura 26 Estructura principal de la banda transportadora</i>	<i>27</i>
<i>Figura 27 Rodillos de soporte</i>	<i>28</i>
<i>Figura 28 Prototipo de banda transportadora terminado</i>	<i>29</i>
<i>Figura 29 Conexión del sensor ultrasónico y módulo ethernet a la placa Arduino UNO.....</i>	<i>29</i>
<i>Figura 30 Conexión del sensor de color y módulo ethernet a la placa Arduino UNO.....</i>	<i>30</i>
<i>Figura 31 Conexión de servomotores a la placa Arduino UNO</i>	<i>31</i>
<i>Figura 32 Diagrama de conexión Doble Delta</i>	<i>32</i>
<i>Figura 33 Conexión Doble Delta en tiempo real</i>	<i>32</i>
<i>Figura 34 Diagrama de conexión para el variador de frecuencia.....</i>	<i>33</i>
<i>Figura 35 Alimentación para el PLC y HMI.....</i>	<i>34</i>
<i>Figura 36 Bibliotecas utilizadas.....</i>	<i>35</i>

<i>Figura 37 Pines del sensor TCS3200</i>	<i>35</i>
<i>Figura 38 Pines del sensor HC-SR04.....</i>	<i>35</i>
<i>Figura 39 Variables de los valores de color</i>	<i>36</i>
<i>Figura 40 Variables del sensor ultrasónico</i>	<i>36</i>
<i>Figura 41 Configuración de pines como entradas y salidas</i>	<i>37</i>
<i>Figura 42 Configuración de red local.....</i>	<i>37</i>
<i>Figura 43 Registros que almacenan los datos.....</i>	<i>38</i>
<i>Figura 44 Pines del servomotor</i>	<i>38</i>
<i>Figura 45 Función para leer datos de los sensores</i>	<i>38</i>
<i>Figura 46 Lectura de valores del servomotor</i>	<i>39</i>
<i>Figura 47 Envío de valores procesados</i>	<i>39</i>
<i>Figura 48 Imprime valores en el monitor serial.....</i>	<i>39</i>
<i>Figura 49 Bloque de datos utilizados en el TIA Portal V18.....</i>	<i>40</i>
<i>Figura 50 Bloque de organización “MAIN”</i>	<i>40</i>
<i>Figura 51 Bloque de función “FC_CONTROL”</i>	<i>41</i>
<i>Figura 52 Bloque de organización “OB_VDF”</i>	<i>42</i>
<i>Figura 53 Bloque MB_CLIENT.....</i>	<i>43</i>
<i>Figura 54 Programación SCL del OB_CHARACTERISTICAS.....</i>	<i>44</i>
<i>Figura 55 Programación SCL del FC_SECUENCIA.....</i>	<i>45</i>
<i>Figura 56 Tabla de variables del HMI.....</i>	<i>46</i>
<i>Figura 57 Pantalla de presentación</i>	<i>46</i>
<i>Figura 58 Ventana de selección de interfaces</i>	<i>47</i>
<i>Figura 59 Pantalla de administración de usuarios</i>	<i>47</i>
<i>Figura 60 Pantalla de parámetros - Tipos de productos</i>	<i>48</i>
<i>Figura 61 Pantalla de parámetros - Control del conveyor</i>	<i>49</i>

<i>Figura 62 Pantalla del Proceso.....</i>	<i>50</i>
<i>Figura 63 Diagrama de proceso.....</i>	<i>51</i>
<i>Figura 64 Marcha inicial.....</i>	<i>52</i>
<i>Figura 65 Lógica Set-Reset</i>	<i>52</i>
<i>Figura 66 Interfaz de marcha y paro del operador.....</i>	<i>53</i>
<i>Figura 67 Fases de detección.....</i>	<i>53</i>
<i>Figura 68 Parámetros asignados para análisis</i>	<i>54</i>
<i>Figura 69 Fase de procesamiento de datos</i>	<i>54</i>
<i>Figura 70 Clasificación según parámetros establecidos.....</i>	<i>55</i>
<i>Figura 71 Resultados de la clasificación.....</i>	<i>56</i>
<i>Figura 72 Rutas de clasificación.....</i>	<i>56</i>
<i>Figura 73 Intento inicial de la instalación del sensor ultrasónico.....</i>	<i>57</i>
<i>Figura 74 Prueba desde diferente ángulo con una caja de diferente color</i>	<i>58</i>
<i>Figura 75 Instalación correcta del sensor ultrasónico</i>	<i>59</i>
<i>Figura 76 Instalación del sensor TCS3200 en el prototipo.....</i>	<i>59</i>
<i>Figura 77 Prueba manual para verificar el funcionamiento.....</i>	<i>60</i>
<i>Figura 78 Prueba automática con la banda en movimiento</i>	<i>61</i>
<i>Figura 79 Prueba de movimiento de ángulos del servomotor en su posición inicial.....</i>	<i>62</i>
<i>Figura 80 Caja en movimiento a nueva posición</i>	<i>62</i>
<i>Figura 81 Prueba de sensor inductivo en el prototipo</i>	<i>63</i>
<i>Figura 82 Selección de caja roja grande en la receta.....</i>	<i>64</i>
<i>Figura 83 Selección de caja azul pequeña en la receta</i>	<i>65</i>
<i>Figura 84 Clasificación de caja azul pequeña</i>	<i>66</i>
<i>Figura 85 Selección de caja verde grande en la receta</i>	<i>66</i>
<i>Figura 86 Selección de caja roja pequeña en la receta</i>	<i>67</i>

<i>Figura 87 Clasificación de caja roja pequeña</i>	<i>68</i>
<i>Figura 88 Selección de caja azul grande en receta.....</i>	<i>69</i>
<i>Figura 89 Selección de caja verde pequeña en la receta</i>	<i>70</i>
<i>Figura 90 Clasificación de la caja verde pequeña.....</i>	<i>70</i>
<i>Figura 91 Cronograma de Gantt de actividades del desarrollo de la tesis</i>	<i>81</i>

ÍNDICE DE TABLAS

<i>Tabla 1</i>	<i>Tabla de resultados</i>	<i>76</i>
<i>Tabla 2</i>	<i>Tabla de análisis y rendimiento de sensores y actuadores;Error! Marcador no definido.</i>	
<i>Tabla 3</i>	<i>Tabla de estadísticas generales del sistema</i>	<i>79</i>
<i>Tabla 4</i>	<i>Tabla de Presupuesto Del Desarrollo Del Proyecto.....</i>	<i>80</i>

I INTRODUCCIÓN

En la era de la Industria 4.0, la automatización de procesos industriales se ha convertido en un factor clave para mejorar la eficiencia, la productividad y la calidad en la producción. Los sistemas de clasificación automatizados desempeñan un papel fundamental en este contexto, permitiendo la organización y distribución de productos en sectores como la logística y la manufactura. Estos sistemas utilizan tecnologías avanzadas para identificar y clasificar artículos según diversas características, optimizando el rendimiento y reduciendo errores humanos (LM Group, 2024).

El protocolo Modbus TCP/IP es ampliamente utilizado en sistemas de automatización industrial debido a su capacidad para facilitar la comunicación entre dispositivos a través de redes Ethernet. Esta versión del protocolo Modbus permite una integración sencilla en infraestructuras de red modernas, ofreciendo alta velocidad de datos y escalabilidad (AUMA Iberia, 2024). Su implementación no necesita componentes de hardware adicionales, lo que facilita su integración en diferentes entornos industriales.

Este proyecto desarrolla un prototipo automatizado de clasificación con comunicación Modbus TCP/IP, integrando sensores como el de detección de color TCS3200 y el sensor HC-SR04 en una red Ethernet para supervisión y control en tiempo real. El uso de este protocolo garantiza una comunicación estandarizada y correcta entre dispositivos de distintos fabricantes. (National Instruments, 2024).

Este prototipo demuestra cómo la automatización y los protocolos de comunicación optimizan procesos industriales, fomentando la adopción de soluciones avanzadas para una producción más inteligente.

II PROBLEMA

El presente estudio aborda la problemática de los errores humanos en los sistemas de trabajo, un factor que puede comprometer la eficiencia y seguridad de los procesos de manufactura. Estos errores, comúnmente, se asocian con el diseño inadecuado del sistema o con condiciones del entorno laboral, como cargas excesivas de trabajo, ritmos acelerados, tareas complejas e instrucciones ambiguas. Con la creciente necesidad de optimizar los procesos y minimizar la intervención humana, la industria ha comenzado a adoptar tecnologías de automatización avanzada. A pesar de los avances logrados, es crucial comprender cómo diversos factores asociados al entorno laboral afectan a los operarios, desencadenando errores que pueden resultar en accidentes, problemas de calidad y una producción deficiente (Torres-Medina, 2020) (Néstor Agudelo, 2020).

Por ejemplo, en tareas donde un trabajador debe realizar una inspección visual de detalles pequeños en un producto, la calidad de la iluminación es un aspecto del entorno de trabajo que influye directamente en su rendimiento. Otro aspecto a considerar es la postura, ya que mantener la vista fija en un plano horizontal puede requerir una flexión significativa del cuello, causando dolor y molestias. La fatiga visual también puede aparecer debido a la tensión en los músculos oculares cuando se mantiene la vista enfocada por largos periodos. Además, factores como la monotonía, la repetición de tareas y la presión de tiempo pueden interactuar, afectando el rendimiento del trabajador. Para reducir el error humano, es esencial identificar estos factores específicos y su influencia en el contexto de la tarea, implementando mejoras en el sistema de trabajo. La complejidad de la tarea es un factor relevante que puede aumentar la carga mental; por lo tanto, disminuir esta complejidad podría ser una medida efectiva para reducir los errores y mejorar el desempeño humano (Torres-Medina, 2020).

III OBJETIVOS

3.1 Objetivo general

Implementar un prototipo automatizado de clasificación mediante protocolo MODBUS TCP IP para la detección de color, tamaño y peso de productos.

3.2 Objetivos específicos

- Diseñar el sistema integrado de sensores para la detección de color, peso y tamaño, asegurando la correcta disposición física y conectividad para la clasificación automatizada en una banda transportadora.
- Implementar y programar los dispositivos de control y comunicación mediante el procesamiento de señales de los sensores en tiempo real, para clasificar los productos.
- Evaluar el prototipo utilizando herramientas de simulación adecuadas, como Arduino y TIA Portal, para validar su comportamiento y optimizar su diseño.

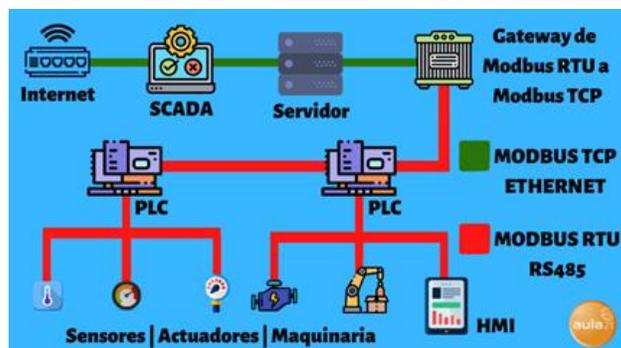
IV FUNDAMENTO TEÓRICO

4.1 Protocolo MODBUS

Modbus es un protocolo de comunicación ampliamente utilizado en redes industriales para intercambiar información entre dispositivos electrónicos. En este protocolo, el dispositivo maestro se encarga de solicitar datos, mientras que los dispositivos esclavos únicamente responden cuando se les consulta. El maestro puede tanto leer como escribir información en los registros de los esclavos, que poseen direcciones únicas dentro de un rango de 1 a 247 (Aula21, 2024).

Una red Modbus típica opera bajo un esquema maestro-esclavo, donde solo el maestro puede iniciar la comunicación. Este protocolo utiliza estándares de comunicación como RS232, RS485 o RS422, lo que lo hace compatible con diferentes tipos de hardware. Su estructura simple y su naturaleza abierta permiten a los fabricantes integrarlo en sus productos sin costos adicionales, favoreciendo su adopción masiva en la industria. Por esta razón, Modbus se ha convertido en una solución popular para conectar equipos electrónicos industriales de diversos fabricantes (Aula21, 2024). En la figura 1 se muestra la arquitectura básica del protocolo MODBUS.

Figura 1
Arquitectura protocolo MODBUS



Nota. Esta imagen muestra la estructura general del protocolo Modbus, incluyendo sus niveles de comunicación y formatos de mensaje (Aula21, 2024).

El protocolo MODBUS tiene 4 tipos de comunicación: TCP/IP, RTU, PLUS y ASCII.

4.1.1 MODBUS TCP/IP

MODBUS TCP/IP es una variante del protocolo MODBUS diseñada para la supervisión y control de equipos de automatización en redes ‘Intranet’ o ‘Internet’, aprovechando la infraestructura TCP/IP. Este protocolo es utilizado principalmente para conectar dispositivos como PLC, módulos de entrada/salida, y Gateway que enlazan con otros buses de campo o redes simples. Hoy en día, MODBUS TCP/IP es un estándar de facto en la automatización industrial. Dado que MODBUS es ampliamente conocido, este documento se centra en señalar las funciones que facilitan la interoperabilidad en equipos de automatización, destacando las "clases de conformidad" que diferencian entre mensajes comunes y opcionales, los cuales pueden variar según el tipo de dispositivo (Logicbus, 2024).

4.1.2 MODBUS RTU

El protocolo Modbus funciona como un medio de comunicación entre dispositivos electrónicos, y en muchas aplicaciones industriales, el modo predeterminado es Modbus RTU. Este modo está definido dentro las especificaciones del protocolo y se utiliza comúnmente por su eficiencia en la transmisión de datos en redes serie (IndHand, 2023).

La estructura de la trama en Modbus RTU sigue un formato estándar que incluye cuatro elementos principales: la dirección del dispositivo, el código de función que define la operación, los datos asociados a la solicitud o respuesta, y un campo de comprobación para garantizar la integridad de la comunicación. Este diseño simple y estructurado permite que Modbus RTU sea ampliamente adoptado en sistemas industriales que requieren comunicación fiable y directa entre múltiples equipos (IndHand, 2023).

4.1.3 MODBUS PLUS

Modbus Plus, conocido como MB+, es una red de alta velocidad diseñada específicamente para entornos de control industrial. Esta tecnología se clasifica como una red Fieldbus y utiliza un esquema típico de bus de tokens para gestionar la comunicación entre dispositivos. Además, opera como una red de área local (LAN), proporcionando una conexión correcta y confiable para aplicaciones industriales que requieren intercambio rápido de datos y control en tiempo real. Su diseño la hace ideal para integrar equipos en sistemas de automatización complejos (IndHand, 2023).

4.1.4 MODBUS ASCII

El protocolo Modbus ASCII es una variante que utiliza caracteres ASCII para codificar las instrucciones, a diferencia del formato binario compacto de Modbus RTU. Esta conversión implica que cada byte de datos debe ser representado por dos caracteres ASCII. Por ejemplo, un valor como "12", que en Modbus RTU solo necesita un byte para su transmisión, en Modbus ASCII requiere que el dígito "1" se convierta a su representación ASCII "31" y el "2" a "32". Esto aumenta significativamente el tamaño de los datos transmitidos, lo que hace que el protocolo sea más lento en comparación con Modbus RTU (IndHand, 2023).

Modbus ASCII ofrece una estructura más legible y puede ser útil en ciertos sistemas que requieren una comunicación más simple, su bajo rendimiento limita su uso en aplicaciones industriales que demandan alta velocidad y eficiencia. La baja tasa de utilización del ancho de banda convierte a esta variante en una opción menos adecuada para redes modernas, especialmente cuando se compara con otros modos de Modbus que están mejor optimizados para aplicaciones de tiempo crítico y con mayores volúmenes de datos (IndHand, 2023).

4.2 Banda transportadora

Las bandas transportadoras son dispositivos indispensables en la industria y el comercio, que agilizan el desplazamiento de materiales de manera continua y correcta. Al automatizar el transporte de productos o artículos a lo largo de una ruta determinada, estos sistemas contribuyen significativamente a optimizar los procesos logísticos y productivos. Su uso permite reducir tiempos de traslado y mejorar la organización en líneas de producción, lo que se traduce en una mayor productividad. Además, las cintas transportadoras ofrecen una solución segura y efectiva para manejar grandes volúmenes de materiales, adaptándose a diferentes tipos de entornos industriales (BeltingLab, 2023). En la figura 2 se observa una banda transportadora industrial.

Figura 2

Ejemplo de banda transportadora industrial



Nota. Imagen de una banda transportadora utilizada en procesos industriales para el traslado de cajas dentro de una línea de producción (Marcos, 2020).

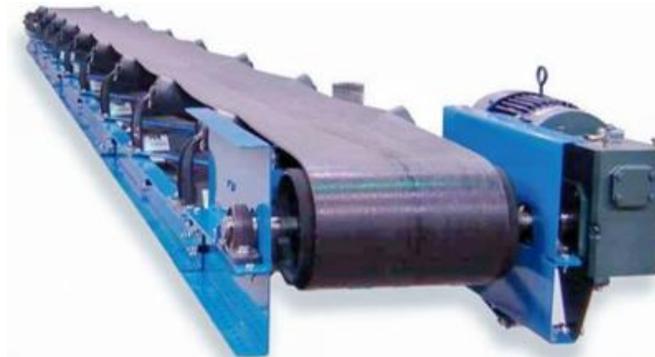
La estructura de una banda transportadora compone elementos como: Soporte de cinta, poleas y unidad de accionamiento.

4.2.1 Soporte de cinta

El soporte de la banda, figura 3, es un componente esencial en un sistema de cinta transportadora, ya que permite el movimiento fluido de la correa. Si este soporte no es lo suficientemente estable, pueden surgir problemas importantes. Por ejemplo, una banda que carece de soporte adecuado puede ceder bajo el peso de cargas pesadas, comprometiendo la eficiencia del traslado de materiales. Además, una cinta con flacidez puede presentar un movimiento irregular y lento, lo que impacta negativamente en la productividad del sistema (VENMIR, 2022).

Figura 3

Ejemplo de soporte con su cinta



Nota. Representación de un soporte estructural de una banda transportadora junto con su cinta de transporte. Este componente es esencial para el adecuado funcionamiento del sistema de transporte en procesos industriales (Smith, 2024).

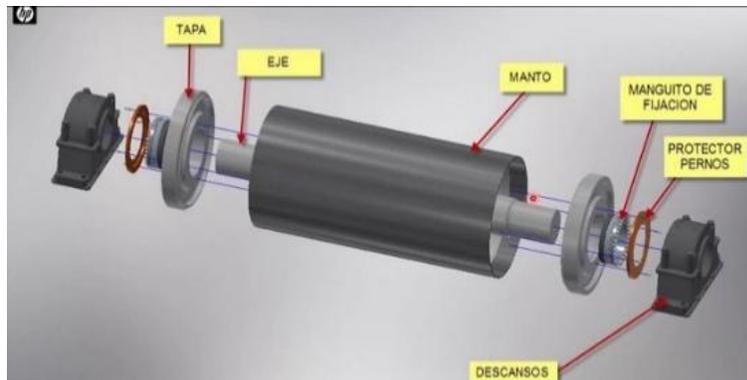
4.2.2 Poleas

Las poleas son elementos fundamentales para dirigir el movimiento de la correa en una cinta transportadora. Cada sistema cuenta con al menos dos poleas: una motriz, que recibe la fuerza de arrastre, y otra que permanece pasiva. En configuraciones más avanzadas, pueden añadirse poleas adicionales a lo largo del marco de la estructura. Las poleas están equipadas con cojinetes que optimizan el movimiento de las piezas y permiten que la correa se desplace en ambas

direcciones, especialmente en sistemas que requieren cambios constantes de dirección. Incluso en sistemas manuales, las poleas son esenciales para el buen funcionamiento de la banda transportadora (VENMIR, 2022). En la figura 4 se ve la estructura interna de una polea.

Figura 4

Estructura interna de una polea



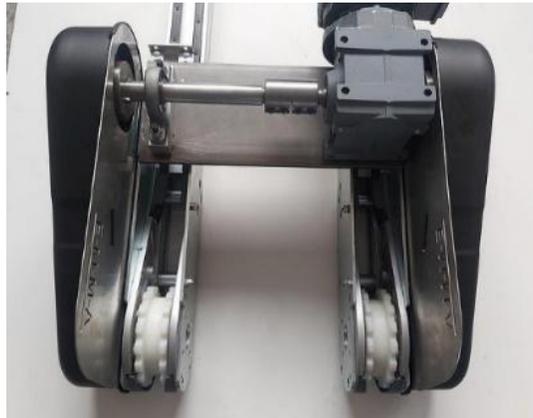
Nota. Vista detallada de la estructura interna de una polea, mostrando sus componentes principales y su función en la transmisión de movimiento en sistemas mecánicos (PRECISION, 2020).

4.2.3 Unidad de accionamiento

La unidad de accionamiento es el componente que proporciona el impulso necesario para el movimiento de la cinta transportadora en sistemas motorizados, figura 5, Su función es fundamental para asegurar un flujo estable y regulado de los materiales en transporte. Incluso en sistemas operados manualmente, esta unidad resulta importante para realizar ajustes y controlar la velocidad según se requiera (VENMIR, 2022).

Figura 5

Ejemplo de unidad de accionamiento

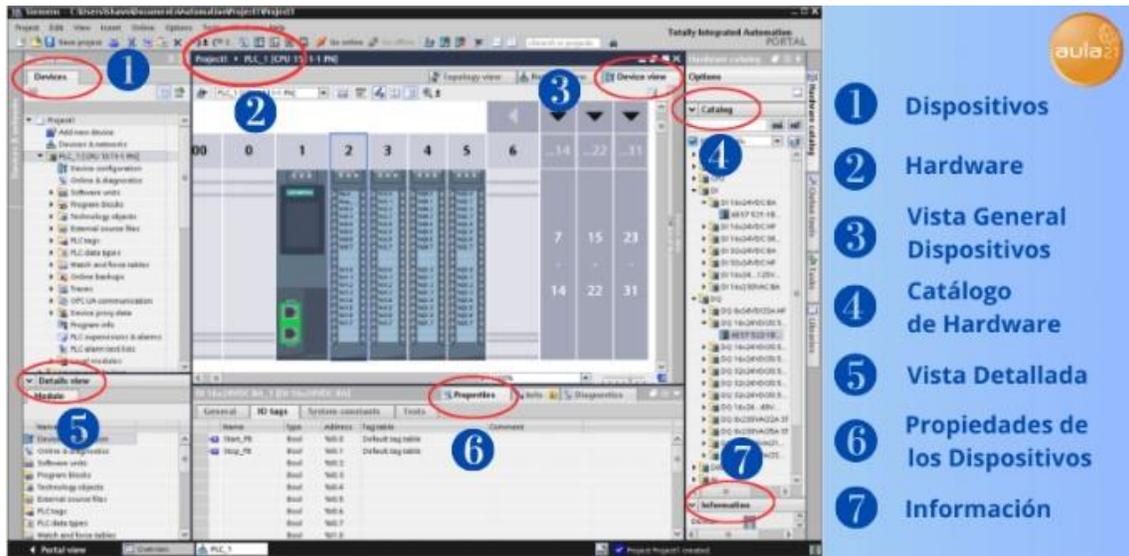


Nota. Representación de una unidad de accionamiento utilizada en sistemas mecánicos para el control del movimiento y la transmisión de potencia (IPV-Solution, 2020).

4.3 TIA Portal V18

TIA Portal es un software versátil, ampliamente utilizado en la industria, que se ha vuelto indispensable para los programadores industriales. Su amplia gama de funciones facilita el desarrollo, implementación y gestión de sistemas de automatización, mejorando la productividad en los procesos industriales. La figura 6 deja en evidencia el entorno de desarrollo del programa. Entre sus principales características, permite programar y depurar Controladores Lógicos Programables (PLCs) como S7-1200, S7-300, S7-1500 y S7-400 mediante lenguajes como Grafcet, Ladder y texto estructurado. Además, simplifica la configuración de interfaces HMI para el control y visualización en tiempo real de los procesos industriales, facilita la integración de dispositivos como el servomotor y variador de frecuencia en una sola plataforma, y optimiza el mantenimiento con funciones de diagnóstico, copias de seguridad y configuración de alarmas (WitAutomatización, 2023).

Figura 6
Entorno de desarrollo del TIA Portal



Nota. Captura de pantalla del entorno del TIA Portal, la imagen muestra diferentes secciones, como la vista de dispositivos, el catálogo de hardware y las propiedades del sistema (Aula21, 2024).

4.3.1 Bloque de organización OB

En TIA Portal, un bloque de organización (OB) gestiona las interrupciones del programa, figura 7, lo que permite detener temporalmente su ejecución cíclica. El OB principal, denominado OB1, es el punto de inicio para la programación y desde él se invocan otros bloques o bloques de organización adicionales si han sido configurados (TecnoPLC, 2023).

Figura 7
Bloque OB



Nota. La imagen muestra un Bloque de Organización (OB) en TIA Portal, utilizado para gestionar la ejecución del programa en un PLC Siemens. El OB1 es el principal, y otros OB manejan eventos o interrupciones específicas. (Siemens, 2025)

4.3.2 Bloque de función FB

Un bloque de función (FB) debe ser invocado desde otro bloque lógico, ya sea otro FB, un OB o una función FC. Cada vez que se llama a un FB, este cuenta con un grupo de datos asociado, lo que significa que las variables definidas dentro del bloque recibirán valores específicos en cada instancia, figura 8, a este conjunto de datos se le conoce como DB de instancia.

Figura 8
Bloque FB



Nota. La imagen muestra un Bloque de Función (FB) en TIA Portal, utilizado para ejecutar tareas específicas dentro de un PLC Siemens. Este bloque cuenta con un DB de instancia que almacena datos exclusivos en cada llamada (Siemens, 2025).

Por ello, es posible utilizar un mismo FB en diferentes secciones del programa, asignándole distintos bloques de datos (DB) que modificarán los valores almacenados en su interior (TecnoPLC, 2023).

4.3.3 Función FC

Una función es un bloque diseñado para ejecutar una operación específica dentro del programa general. Su uso permite organizar el proyecto de manera estructurada, asignando una función distinta a cada sección del programa, figura 9, por ejemplo, se puede desarrollar una función dedicada al control de alarmas generadas por los sensores o una función específica para la lectura de señales analógicas de cada sensor (TecnoPLC, 2023).

Figura 9
Función FC



Nota. La imagen representa un Bloque de Función (FC) en TIA Portal, utilizado para ejecutar operaciones específicas sin requerir un DB de instancia. Estos bloques optimizan la organización del programa en un PLC Siemens (Siemens, 2025).

4.3.4 Bloque de datos DB

En TIA Portal, los bloques de datos (DB) funcionan como áreas de memoria destinadas al almacenamiento de información generada durante la ejecución del programa, figura 10, además, pueden contener datos utilizados en diferentes secciones del código. Un DB global, en particular, permite el acceso a su información desde cualquier otro bloque del programa, lo que facilita el almacenamiento de los datos provenientes de los sensores a través de Modbus (TecnoPLC, 2023).

Figura 10
Bloque de datos DB



Nota. La imagen representa un Bloque de Datos (DB) en TIA Portal, utilizado para almacenar los datos recibidos de los sensores por el protocolo Modbus TCP/IP. Estos bloques optimizan la organización ya que pueden usarse en cualquier línea de código (Siemens, 2025).

4.4. Arduino IDE

El entorno de desarrollo integrado (IDE) de Arduino es una herramienta fundamental que simplifica la escritura, depuración y carga de código en los dispositivos Arduino. Gracias a su

diseño intuitivo, permite a los usuarios trabajar de manera agradable en proyectos electrónicos y de programación. Este software se destaca por ofrecer una interfaz amigable, accesible tanto para principiantes como para expertos. La facilidad con la que se pueden compilar y transferir programas al hardware ha contribuido significativamente al éxito de Arduino en la comunidad tecnológica. Además, su compatibilidad con múltiples sistemas operativos y su soporte para una amplia variedad de placas hacen que sea una opción versátil. Estas características han impulsado el uso de Arduino en proyectos educativos, de investigación y profesionales, consolidándolo como una plataforma accesible y poderosa en el ámbito de la electrónica y la automatización. (Arduino, 2020). Como se observa en la figura 11 el entorno de desarrollo del Arduino IDE.

Figura 11
Entorno de desarrollo de Arduino IDE.



```
sketch_dec10a.ino
1 #include <SPI.h>
2 #include <Ethernet.h>
3 #include <ModbusTCP.h>
4 #include <HX711.h>
5 #include <TCS3200.h>
6
7 // Configuración del Modbus TCP
8 ModbusTCPServer modbusTCPServer;
9 EthernetServer server(502); // Puerto por defecto de Modbus TCP
10
11 // Dirección MAC y IP estática (ajustar según tu red)
12 byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
13 IPAddress ip(192, 168, 1, 100);
14
15 // Configuración del sensor HX711 (celda de carga)
16 HX711 loadCell;
17 #define DOUT 3 // Pin de datos de HX711
18 #define CLK 2 // Pin de reloj de HX711
19
20 // Configuración del sensor TCS3200
21 #define S2 4
22 #define S3 5
23 #define OUT 6
24
25 TCS3200 tcs(S2, S3, OUT);
26
27 // Variables globales
28 long weight = 0;
29 uint16_t colorData = 0;
30
31 void setup() {
32 // Inicialización de comunicación serial
33 Serial.begin(9600);
34
```

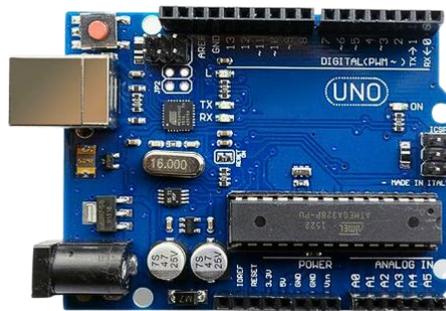
Nota. En la imagen se observa la pantalla principal de Arduino donde se escriben las líneas de programación correspondientes para el funcionamiento de los sensores (Arduino, 2020).

4.5 Arduino UNO

El Arduino UNO es una placa electrónica diseñada alrededor del microcontrolador ATmega328P, ideal para aplicaciones en automatización y programación. Posee 14 pines digitales configurables como entradas o salidas, y 6 de ellos son compatibles con señales PWM. Además, incluye 6 pines de entrada analógica, un cristal oscilador de 16 MHz que asegura su precisión, un puerto USB para comunicación y alimentación, un conector de corriente tipo jack, terminales para programación ICSP y un botón de reinicio. La placa incorpora todos los componentes necesarios para que el microcontrolador funcione correctamente, lo que permite usarla con facilidad. Puede alimentarse a través de un puerto USB o con un adaptador AC-DC, lo que la hace práctica y versátil para una amplia variedad de proyectos (Arduino, 2020). La figura 12 se observa el microcontrolador que se utilizara en el proyecto la cual es la placa de Arduino UNO.

Figura 12

Ejemplo de placa Arduino UNO



Nota. Imagen de la placa de desarrollo Arduino Uno, utilizada para programar microcontroladores. La placa incluye un microcontrolador ATmega328P, pines de entrada/salida digitales y analógicas y una interfaz de comunicación USB. (Arduino, 2020).

4.6 Sensor de color TCS3200

El sensor de color TCS230 el cual se muestra en la figura 13 es capaz de detectar la luz blanca, compuesta por los colores primarios rojo, verde y azul, cada uno asociado a una longitud de onda específica, Estos colores pueden combinarse para generar una amplia variedad de

tonalidades. Cuando la luz blanca incide sobre una superficie, ciertas longitudes de onda son absorbidas mientras que otras se reflejan, dependiendo de las características del material de la superficie. El color que se percibe corresponde a las longitudes de onda reflejadas que llegan a los ojos, lo que permite identificar la tonalidad de los objetos. Este principio hace posible que el sensor reconozca colores al analizar las propiedades de la luz reflejada (Paguayo, 2022).

Figura 13

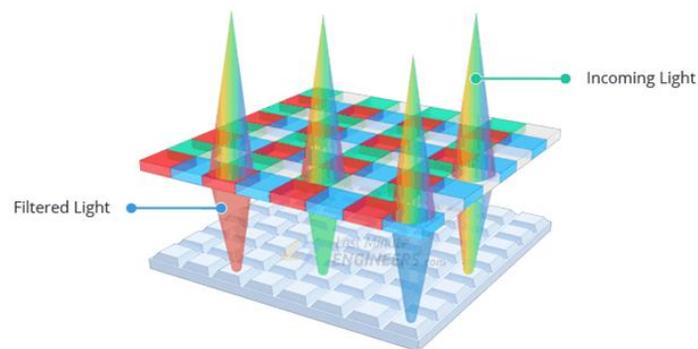
Ejemplo de sensor TCS3200



Nota. Imagen de un sensor de color **TCS3200**, utilizado para la detección y diferenciación de colores en aplicaciones de automatización y robótica. Funciona mediante una matriz de fotodiodos y un generador de frecuencia configurable (Novatronic, 2020).

Figura 14

Ejemplo de tonalidades del sensor TCS3200



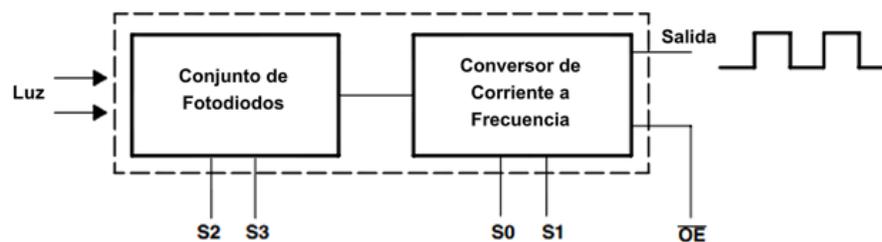
Nota. La imagen muestra un ejemplo de tonalidades del sensor TCS3200 representado como un diagrama de bloques (Paguayo, 2022).

El diagrama de bloques de la figura 14, muestra el funcionamiento interno de un sensor óptico de color, como el TCS3200. La luz incidente se detecta mediante un conjunto de fotodiodos, que convierten la intensidad luminosa en una señal eléctrica. Estos fotodiodos están organizados para captar diferentes colores (rojo, verde, azul o transparente), y su selección se realiza mediante

las entradas de control S2 y S3. La señal generada se envía a un convertor de corriente a frecuencia, que la transforma en una señal digital de pulsos cuya frecuencia es proporcional a la intensidad luminosa detectada. Este convertor puede configurarse en distintas escalas mediante los pines S0 y S1, permitiendo ajustar su sensibilidad. Finalmente, la salida del sensor, representada por la señal de pulsos, se activa o desactiva con el pin OE (Output Enable), que controla si la señal está disponible para ser procesada por un sistema externo. En la figura 15 se muestra un diagrama de bloques sobre el funcionamiento del sensor de color.

Figura 15

Diagrama de bloques del sensor de color



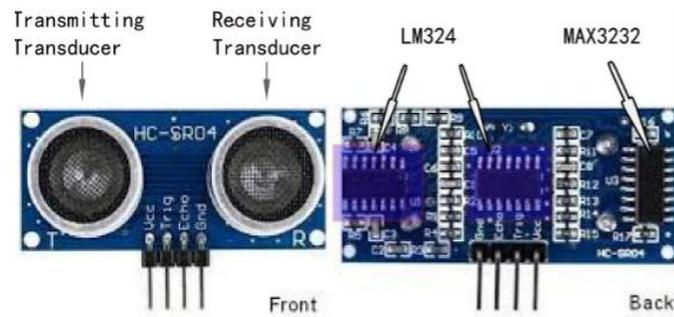
Nota. Imagen de un diagrama de bloques que explica el funcionamiento del sensor de color TCS3200, mostrando los diferentes componentes y cómo interactúan para detectar y diferenciar colores en aplicaciones de automatización (Robots.Argentina, 2018).

4.7 Sensores Ultrasónico HC-SR04

El sensor ultrasónico HC-SR04 mide la distancia a un objeto usando un sistema de sonar, similar al método que utilizan los murciélagos. Este sensor ofrece detección de rango sin contacto, con lecturas estables, y viene en un formato de fácil manejo. En su parte frontal, posee dos cilindros metálicos que funcionan como transductores. Estos dispositivos transforman fuerzas mecánicas en señales eléctricas. Uno de ellos actúa como transmisor y convierte la señal eléctrica en pulsos ultrasónicos de 40 KHz, mientras que el otro, como receptor, detecta esos pulsos. Al captar el eco de los pulsos, genera una señal de salida, cuyo ancho permite calcular la distancia recorrida por el sonido (Solectro, 2021). Se observa en la figura 16 la estructura de un sensor ultrasónico.

Figura 16

Estructura de un sensor ultrasónico



Nota. Imagen de un sensor ultrasónico HC-SR04, utilizado para medir la distancia de las cajas que pasan por la banda en el sistema de clasificación. El sensor funciona mediante la emisión y recepción de ondas ultrasónicas (Solectro, 2021).

4.8 Servomotor

El servomotor, comúnmente denominado servo, es un actuador diseñado para controlar de manera correcta tanto la posición como el movimiento de su eje, figura 17. A diferencia de los motores eléctricos convencionales, los servomotores destacan por su capacidad para ajustarse con exactitud en términos de ángulo, velocidad y posición en tiempo real. Esto los convierte en una herramienta ideal para aplicaciones que requieren un control detallado y exacto, permitiendo movimientos específicos según las necesidades del sistema. Su precisión y versatilidad hacen que sean ampliamente utilizados en automatización, robótica y sistemas de control industrial, donde el manejo preciso del eje es fundamental para el correcto funcionamiento del equipo o proceso (Egasen, 2021).

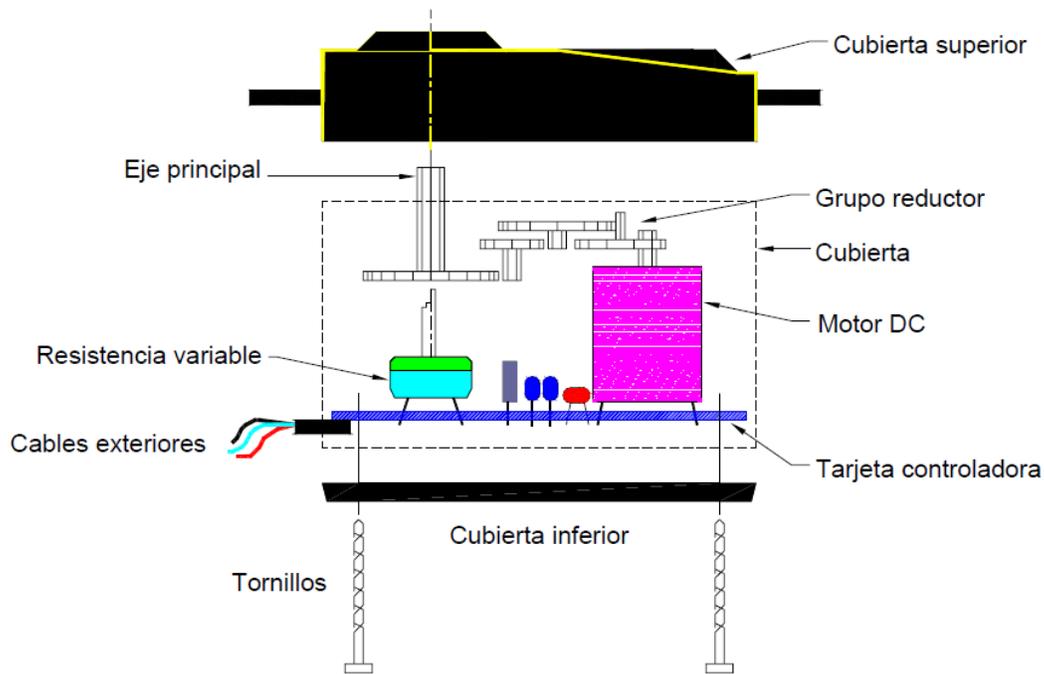
Figura 17
Servomotor S3003 360°



Nota. Imagen de un servomotor S3003, utilizado en el sistema de clasificación de cajas en la tesis. Este servomotor se emplea para el control de movimiento, permitiendo el posicionamiento preciso de las cajas según los criterios de color, tamaño y peso (Novatronic, 2020).

Un servomotor está compuesto por varios elementos esenciales que trabajan en conjunto para ofrecer un control preciso, figura 18, en su estructura interna se encuentra un motor DC, encargado de generar movimiento, y un grupo reductor de engranajes que disminuye la velocidad y aumenta el torque del eje principal. Este eje está conectado a una resistencia variable (potenciómetro), la cual envía información de posición al circuito de control. La tarjeta controladora, ubicada en la base del dispositivo, procesa las señales externas para ajustar el ángulo y la velocidad del motor según las órdenes recibidas. Todo el sistema está protegido por una cubierta superior e inferior, asegurada mediante tornillos, y cuenta con cables exteriores para la conexión eléctrica (Naylamp, 2023).

Figura 18
Estructura interna de un servomotor



Nota. Imagen de la estructura interna del servomotor S3003, mostrando los componentes clave como el motor, los cuales permiten el control preciso del movimiento y la posición en aplicaciones de automatización y clasificación de cajas (Naylamp, 2023).

4.9 Módulo Ethernet ENC28J60

El módulo Ethernet ENC28J60 es una solución que permite integrar microcontroladores, como Arduino o PIC, a redes Ethernet, facilitando la transmisión y recepción de datos a través de redes locales o Internet, figura 19, este módulo está basado en el controlador Ethernet ENC28J60 de Microchip, que cumple con los estándares IEEE 802.3 para garantizar compatibilidad con redes modernas. Entre sus características destacadas se incluye el Acceso Directo a Memoria (DMA), lo que optimiza la transferencia de datos, y un hardware dedicado para el cálculo automático de sumas de verificación (checksums) en protocolos IP. Además, incorpora funciones avanzadas como el filtrado de paquetes para mejorar la eficiencia y la seguridad en la comunicación, convirtiéndose en una herramienta clave para proyectos de conectividad en sistemas embebidos. Su diseño

compacto y funcionalidad lo hacen ideal para aplicaciones en automatización industrial (ElectrónicaTNC, 2022).

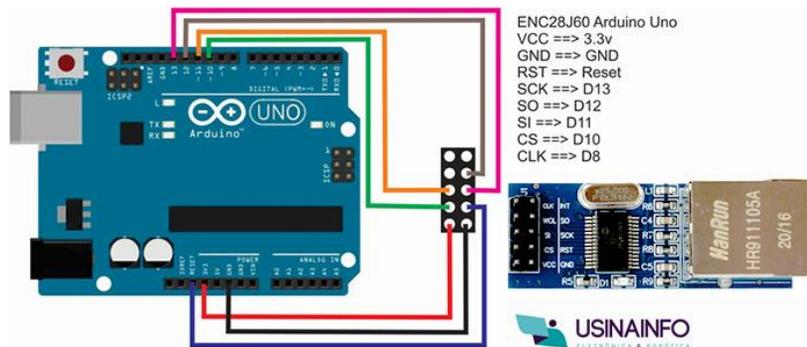
Figura 19
Módulo ethernet mini ENC28J60



Nota. Imagen de un módulo Ethernet ENC28J60, utilizado para la comunicación del Arduino al PLC por Modbus en el sistema de clasificación de cajas. Este módulo permite la conexión de Arduino a redes Ethernet mediante una interfaz SPI (Robotics, 2023).

El módulo Ethernet ENC28J60 se conecta al Arduino Uno usando la interfaz SPI, vinculando los pines MISO, MOSI, SCK y CS para establecer la comunicación en red. Esta conexión permite que el Arduino envíe y reciba datos a través de una red Ethernet, como se observa en la figura 20.

Figura 20
Conexión módulo Ethernet con Arduino UNO



Nota. Conexión del módulo Ethernet ENC28J60 al Arduino, esta conexión permite la comunicación entre el Arduino y redes Ethernet, facilitando la transmisión de datos entre el microcontrolador y otros dispositivos (Usinainfo, 2020).

4.10 Módulo LM2596

El regulador de voltaje LM2596, figura 21, es un módulo step-down diseñado para reducir la tensión de entrada de un circuito cuando se cuenta con una fuente de alimentación de mayor voltaje. Este dispositivo admite un rango de voltaje de entrada de 4V a 35V y permite obtener un voltaje de salida ajustable entre 2V y 28V. Además, puede manejar corrientes de hasta 3A. La regulación del voltaje de salida se realiza a través de un potenciómetro multivuelta, lo que facilita un ajuste preciso (UNIT Electronics, 2025).

Figura 21
Módulo Step-down LM2596



Nota. Módulo regulador step-down LM2596 con display de 7 segmentos para controlar el voltaje enviado al servomotor utilizado para la clasificación (ArduMótica, 2023).

4.11 Sensor inductivo

Un sensor inductivo, figura 22, es un dispositivo electrónico diseñado para identificar la presencia de objetos metálicos en su proximidad. Su funcionamiento se basa en la creación de un campo magnético que, al interactuar con un metal cercano, produce una alteración detectable. Estos sensores actúan como "detectores electrónicos", emitiendo un campo magnético de alta frecuencia que monitorea cambios en su entorno. Cuando se detecta una variación, el sensor genera una señal eléctrica que puede activar o desactivar otros dispositivos o sistemas. Gracias a esta

capacidad, son ampliamente utilizados en diversas aplicaciones industriales y tecnológicas (SDI Industrial, 2022).

Figura 22
Sensor inductivo



Nota. Sensor inductivo industrial con un rango de detección de 1 cm con el metal (*MicroChipOile, 2025*).

V MARCO METODOLÓGICO

5.1 Enfoque de la investigación

Este estudio sigue un enfoque aplicado y experimental, ya que se orienta al diseño e implementación de un sistema automatizado de clasificación de objetos mediante una banda transportadora de hierro controlada por un servomotor y gestionada a través de un PLC con comunicación Modbus TCP/IP.

El paradigma cuantitativo rige la investigación, dado que se fundamenta en la recopilación y análisis de datos obtenidos a partir de la interacción entre el Arduino, el PLC y la interfaz HMI en TIA Portal.

Figura 23
Estructura del proyecto



La figura 23 muestra un sistema de automatización basado en comunicación Modbus TCP/IP, donde varios sensores y actuadores están conectados a dos placas Arduino UNO. Cada Arduino está equipado con un módulo Ethernet para transmitir datos al PLC Siemens a través de un switch de red. Los sensores, incluyendo un sensor ultrasónico HC-SR04 y un sensor de color TCS3200, envían datos a un Arduino, mientras que otro Arduino controla el servomotor. El PLC recibe y procesa la información, donde permite enviar un dato del PLC al servo para su respectivo

movimiento, permitiendo la visualización y control del sistema mediante una HMI Siemens, facilitando la supervisión en tiempo real.

5.2 Diseño y planificación del prototipo

El proceso de creación del prototipo de la banda transportadora comenzó con una fase de diseño y planificación minuciosa, en la que se establecieron los requisitos mecánicos, electrónicos y funcionales del sistema. El propósito fundamental fue desarrollar una banda que pudiera clasificar objetos en función de características específicas como color, peso y tamaño, incorporando un sistema de control basado en el protocolo Modbus TCP/IP para asegurar su automatización y conectividad.

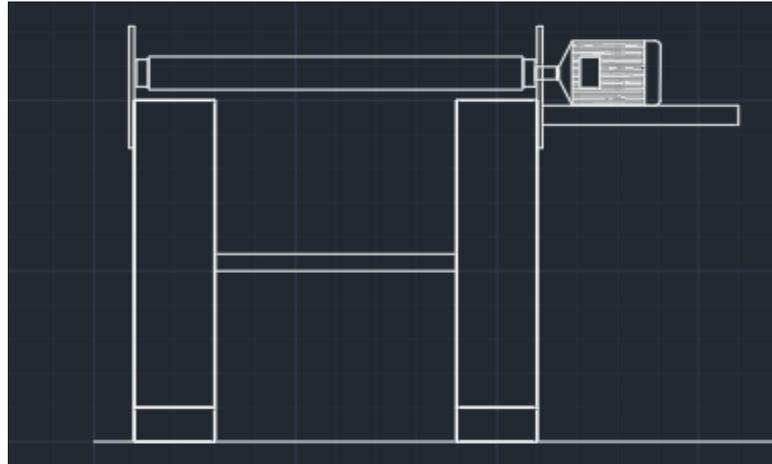
La estructura de la banda debía ser tanto resistente como ligera y modular. Por esta razón, se eligieron perfiles de metal, que ofrecen una alta resistencia mecánica y facilidad de montaje. Además, se seleccionaron los componentes principales, tales como motores de corriente continua con sus respectivos controladores, sensores TCS3200 para la identificación de color y un sensor ultrasónico para medir el tamaño de los objetos. La selección de estos componentes se realizó bajo criterios de precisión, compatibilidad y coste.

Durante esta fase, se elaboraron los planos y diagramas técnicos, que incluían la distribución de los componentes y los esquemas eléctricos iniciales. Este diseño permitió anticipar los puntos de montaje de cada elemento y optimizar la disposición, facilitando así un transporte adecuado de los objetos.

Además, se generaron representaciones visuales detalladas de la banda desde diferentes perspectivas, incluyendo vistas frontales, laterales y aéreas. Estas imágenes permiten comprender de manera clara la disposición de los componentes y la estructura general del sistema, facilitando

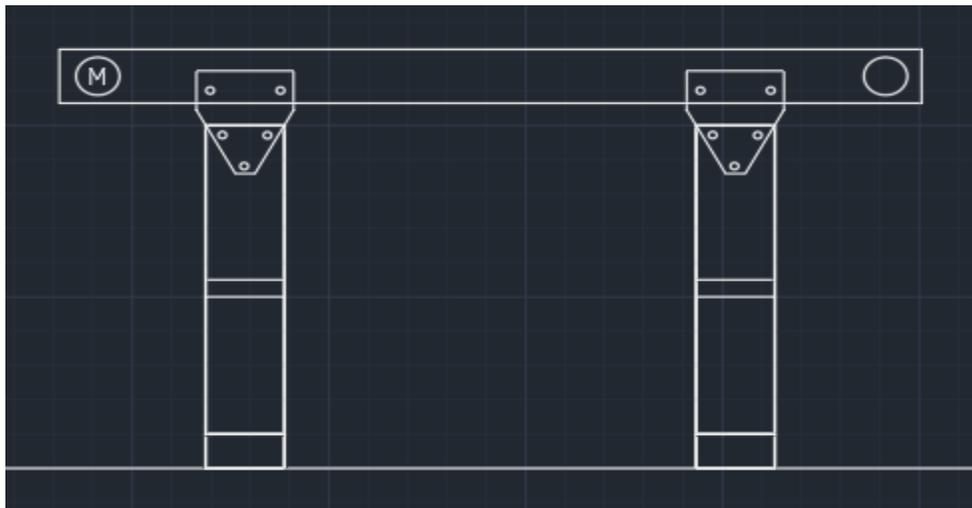
su interpretación y análisis. En la figura 24 demuestra una vista frontal del prototipo realizado en AutoCAD.

Figura 24
Vista frontal del prototipo



En la figura 25 se ve de manera lateral el prototipo de la banda con su mecanismo de movimiento y la ubicación del motor.

Figura 25
Vista lateral del prototipo



5.3 Implementación del prototipo

5.3.1 Armado de la banda transportadora

La construcción de la banda transportadora comenzó con el ensamblaje de la estructura principal, que fue diseñada para ofrecer estabilidad y resistencia. Para ello, se emplearon perfiles de metal de calidad, los cuales fueron conectados entre sí mediante conectores metálicos. Figura 26, Este diseño permitió obtener una base robusta, pero al mismo tiempo ligera y ajustable, que facilitaba la manipulación y el ajuste de la banda según las necesidades del proyecto.

Figura 26

Estructura principal de la banda transportadora



En los extremos de la estructura se instalaron rodillos de soporte, los cuales jugaron un papel crucial al garantizar el movimiento suave y continuo de la banda transportadora, figura 27. La banda misma fue fabricada con un material reforzado, conocido por su resistencia al desgaste y capacidad para soportar las exigencias del transporte de objetos de diferentes tamaños y pesos, sin comprometer su integridad a lo largo del tiempo.

Figura 27
Rodillos de soporte



Una vez que la estructura base estuvo ensamblada, se procedió a la instalación de los motores en posiciones estratégicas para asegurar un desplazamiento uniforme y controlado de la banda. Estos motores fueron acoplados a los rodillos mediante poleas y bandas sincronizadas, lo que garantizó una transmisión correcta de la energía mecánica, permitiendo el movimiento continuo de la banda sin interrupciones.

Además de estos elementos, se instalaron guías laterales ajustables a lo largo de la banda, con el fin de evitar que los objetos que se transportaban se deslicen fuera de la trayectoria. Estas guías fueron diseñadas para ser fácilmente ajustables, de modo que pudieran adaptarse a diferentes tamaños y formas de objetos, asegurando un flujo de trabajo sin obstáculos ni pérdidas de material. La combinación de estos componentes permitió la creación de una banda transportadora buena y duradera, como se observa en la figura 28, adecuada para su propósito de clasificación automatizada de objetos.

Figura 28

Prototipo de banda transportadora terminado

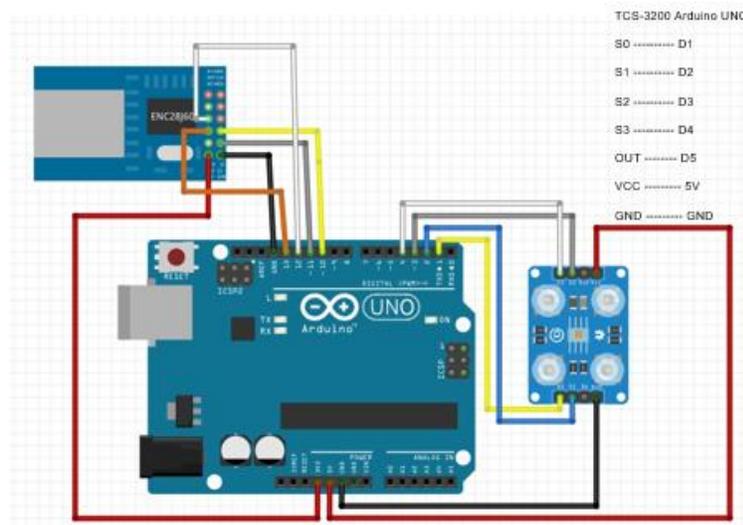


5.3.2 Implementación de sensores y actuadores

La implementación de este trabajo requiere de conexiones físicas las cuales ayudará con el intercambio de información y la lectura o escritura de los respectivos sensores y actuadores, Por lo que se realiza el siguiente esquema de conexiones. La figura 29 indica la manera en la que los sensores HC-SR04 estará conectados en el ARDUINO UNO, donde el VCC va conectado a los 5V de la placa, el GND al GND respectivamente, el pin TRIG al D7 y el ECHO al pin 8. Adicionalmente se conecta un módulo de comunicación ENC28J60 que es el que permite realizar la comunicación TCP/IP entre el ARDUINO y PLC.

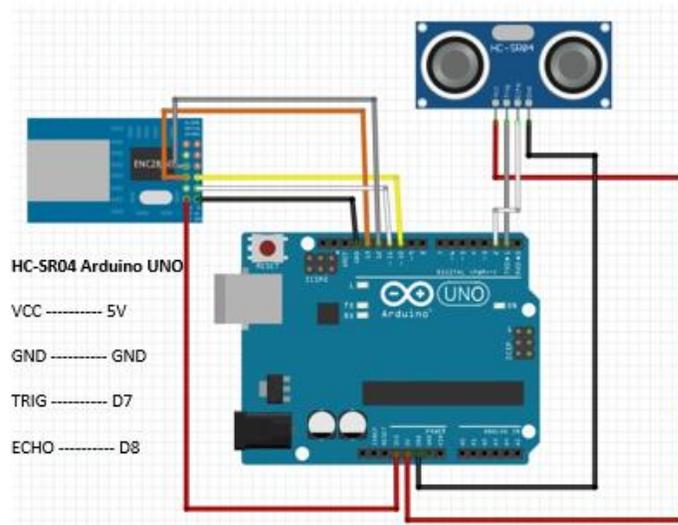
Figura 29

Conexión del sensor ultrasónico y módulo ethernet a la placa Arduino UNO



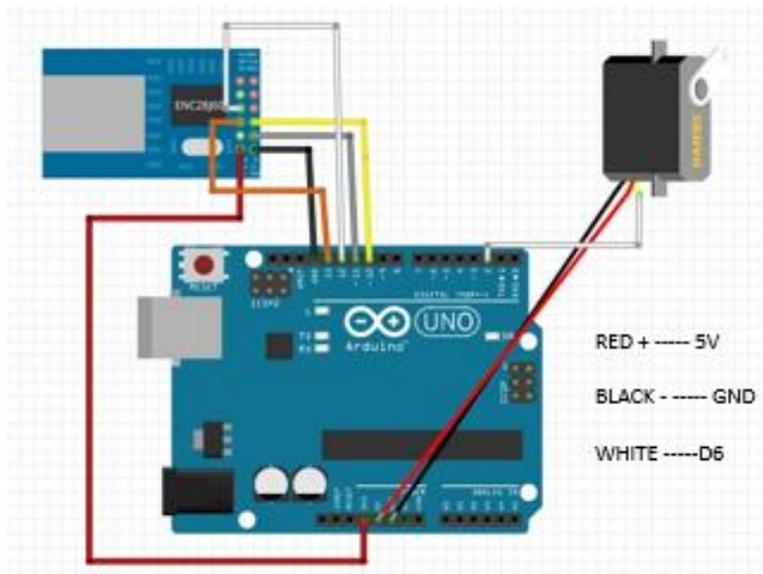
Además de los sensores HC-SR04 también se necesitará censar el color que tiene el producto y para esto se utiliza el sensor TCS3200, el cual está conectado como evidencia la figura 30, donde los pines del S0 al S3 val del D1 al D4 respectivamente, el ping OUT va conectado al D5, el VCC a los 5V y por último el GND al GND de la placa.

Figura 30
Conexión del sensor de color y módulo ethernet a la placa Arduino UNO



Una vez completado el proceso de detección se necesitará del uso de actuadores, Para este trabajo se utiliza el servomotor S3003, con la finalidad de que permitan clasificar el producto según las características anteriormente detectadas. El pin rojo va conectado a los 5V, el negro al GND de la placa y por último el pin blanco que es el de señal va en los pines D6 y D9 para cada servomotor se muestra en la figura 31.

Figura 31
Conexión de servomotores a la placa Arduino UNO



5.3.3 Conexión y configuración del motor trifásico

Luego de la conexión de los sensores en el Arduino para su comunicación, se procede a realizar la conexión del motor trifásico que ya se encuentra ensamblada anteriormente en el prototipo de la banda transportadora, en la figura 32 se ve el diagrama de la conexión a usar es Doble delta ya que con ella se obtiene mayor corriente de fase, línea y mayor voltaje de fase y de línea. En la figura 33 se puede ver la conexión en tiempo real.

Figura 32
Diagrama de conexión Doble Delta

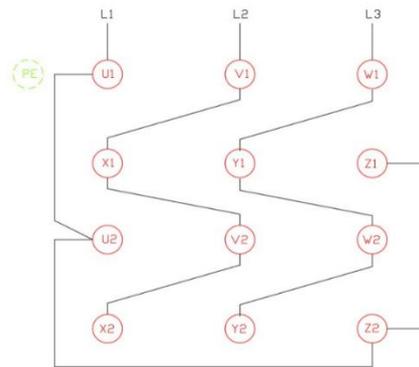


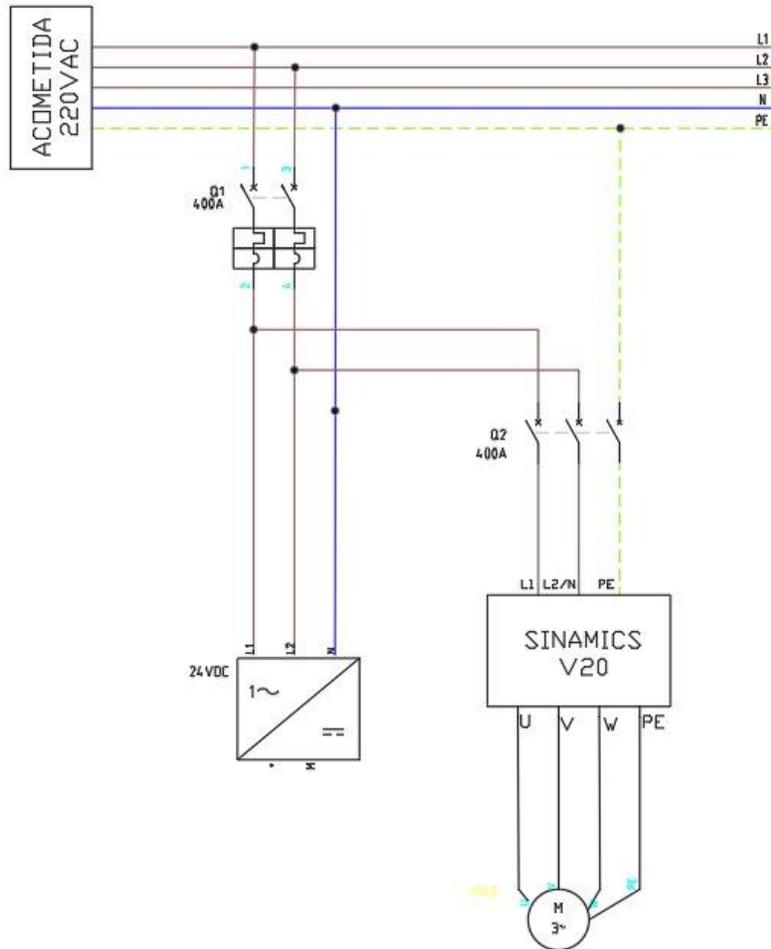
Figura 33
Conexión Doble Delta en tiempo real



El diagrama de la figura 34, representa la conexión eléctrica de un variador de frecuencia SINAMICS V20. La alimentación principal es de 220V AC y se distribuye a través de las líneas L1, L2, L3, N y PE. Para la protección del sistema, se incluyen dos interruptores automáticos de 400A (Q1 y Q2), encargados de prevenir sobrecargas y cortocircuitos. Además, se emplea una fuente de 24VDC para alimentar los circuitos de control.

Figura 34

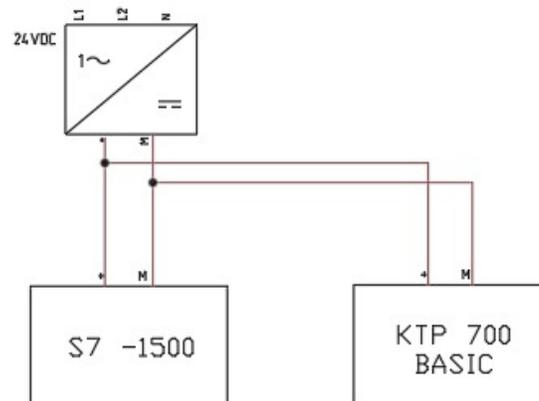
Diagrama de conexión para el variador de frecuencia



El variador SINAMICS V20 recibe la alimentación a través de L1, L2/N y PE, y suministra energía al motor mediante las salidas U, V y W. La conexión a tierra PE está presente en varios puntos para garantizar la seguridad del sistema.

El diagrama de la figura 35, muestra la conexión de una fuente de 24VDC que alimenta un PLC Siemens S7-1500 y una pantalla HMI KTP700 BASIC. La fuente convierte la corriente alterna en continua, suministrando energía a ambos dispositivos a través de las líneas de +24V y GND.

Figura 35
Alimentación para el PLC y HMI



Esta conexión es fundamental en el sistema de automatización, ya que el PLC S7-1500 se encarga del control del proceso, mientras que la HMI KTP700 permite la supervisión y operación mediante una interfaz gráfica.

5.4 Programación del Arduino

El desarrollo del sistema automatizado de clasificación se basa en la integración de sensores y una interfaz de comunicación industrial que permite la interacción con un PLC mediante el protocolo Modbus TCP/IP. Para lograr este objetivo, se utilizó un microcontrolador Arduino, el cual se encarga de recibir señales de los sensores, procesar la información y enviar comandos a los actuadores según los criterios establecidos para la clasificación de los objetos.

5.4.1 Bibliotecas utilizadas

Las bibliotecas utilizadas son esenciales para la gestión de los sensores, la librería ETHERCARD.H sirve para la comunicación ethernet con el módulo ENC28J60, la MODBUSETHERCARD.H que sirve para que el ENC28J60 pueda actuar como servidor o cliente modbus tcp, por otro lado, la librería MODBUS.H permite que el Arduino actúe como maestro o

esclavo modbus y el SERVO.H para controlar los movimientos del servomotor, como se observa en la figura 36.

Figura 36

Bibliotecas utilizadas

```
1  #include <EtherCard.h>
2  #include <ModbusEthercard.h>
3  #include <Modbus.h>
4  #include <Servo.h>
```

5.4.2 Declaración de variables y configuración de pines

Se definen las variables que almacenarán los valores capturados por los sensores, así como las constantes asociadas a la configuración de los pines de entrada y salida. En la figura 37 y 38 se configura los pines mediante una CONST INT para definir las constantes de tipo entero y que no cambien durante la ejecución del programa, se asignan a las conexiones físicas de los sensores (como el TCS3200 o el HC-SR04) y los actuadores (el servomotor). A través de esta asignación, se asegura que cada componente esté correctamente configurado para su funcionamiento.

Figura 37

Pines del sensor TCS3200

```
6  // Pines del sensor TCS3200
7  const int s0 = 1;
8  const int s1 = 2;
9  const int s2 = 3;
10 const int s3 = 4;
11 const int out = 5;
```

Figura 38

Pines del sensor HC-SR04

```
16 // Pines del sensor ultrasónico HC-SR04
17 const int trigPin = 7;
18 const int echoPin = 8;
```

Estas variables INT, guardan los valores de los colores (rojo, verde, azul) leídos por el sensor de color, y se coloca en 0 para que se inicien en ese valor, así como la duración de las

señales del sensor ultrasónico, y el valor del objeto detectado (si está cerca o no), como se observa en la figura 39 y 40.

Figura 39

Variables de los valores de color

```
13  int rojo = 0, verde = 0, azul = 0;
14  int colorValor = 0;
```

Figura 40

Variables del sensor ultrasónico

```
19  long duracion;
20  int objetoDetectado = 0;
```

5.4.3 Configuración Setup()

La función `setup()` se ejecuta una vez al inicio del programa para configurar todos los pines y los parámetros iniciales del sistema. Aquí es donde se establece la comunicación y se preparan los componentes para su uso. Se inicializa la comunicación con el Monitor Serie, lo cual es útil para la depuración y el monitoreo del sistema en tiempo real. Se configuran los pines de los sensores como salidas o entradas, según corresponda.

En la figura 41, se inicia con el `SERIAL.BEGIN` para la comunicación serial entre la placa Arduino y la computadora y se coloca 9600 para la velocidad de transmisión en bits por segundo. Para configurar los pines como entradas o salidas se usa `PINMODE`, para el sensor de color se configuran como salidas para controlar las señales de selección de color, mientras que el pin de salida del sensor se configura como entrada para leer los valores, de manera similar se configura los pines del sensor ultrasónico.

Figura 41

Configuración de pines como entradas y salidas

```
37 void setup() {
38   Serial.begin(9600);
39
40   // Configuración del sensor de color TCS3200
41   pinMode(s0, OUTPUT);
42   pinMode(s1, OUTPUT);
43   pinMode(s2, OUTPUT);
44   pinMode(s3, OUTPUT);
45   pinMode(out, INPUT);
46   digitalWrite(s0, HIGH);
47   digitalWrite(s1, HIGH);
48
49   // Configuración del sensor ultrasónico
50   pinMode(trigPin, OUTPUT);
51   pinMode(echoPin, INPUT);
```

Después de configurar los pines, en la figura 42, se establece la dirección IP de la red de la computadora y la MAC asignada a cada dispositivo de red para identificarlo en una red física para la comunicación de red del Arduino, lo que permite que el microcontrolador se comunique mediante Modbus TCP/IP. Luego en la figura 43 se ven los registros asignados para almacenar los datos, donde para el sensor de color y tamaño se usa MB. ADDIREG que sirve para definir el registro de entrada utilizado para leer los datos de dispositivos externos y para el servomotor el MB. HREG como registros de retención que son valores leídos o escritos por el cliente modbus.

Figura 42

Configuración de red local

```
32 // Configuración de Modbus TCP
33 ModbusEthercard mb;
34 byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
35 byte ip [] = {172, 18, 135, 77}; // Dirección IP del Arduino
```

Figura 43

Registros que almacenan los datos

```
53 | // Configuración de Modbus
54 | mb.config(mac, ip);
55 | mb.addIreg(SENSOR_COLORVAL);
56 | mb.addIreg(SENSOR_TAMANIO);
57 | mb.addHreg(SERVO1_POS, 90);
```

Finalmente se asignan los pines al servomotor, figura 44, donde el pin 6 va conectado al cable blanco para la señal, permitiendo que el Arduino controle sus posiciones en función a los datos que reciba.

Figura 44

Pines del servomotor

```
60 | // Configurar servomotores
61 | servo1.attach(6); // Servo 1 en pin 6
```

5.4.4 Ejecución en loop()

La función loop es donde el sistema realiza su operación continua, ejecutando los ciclos de lectura, procesamiento y control. Cada vez que el ciclo se repite, el sistema obtiene información de los sensores, ajusta la posición del servomotor y envía los valores a través de la red.

En la figura 45 se llaman a las funciones para leer los datos de los sensores. En este caso, la función color() captura los valores de color del sensor TCS3200, y la función medirTamaño() obtiene la distancia medida por el sensor ultrasónico.

Figura 45

Función para leer datos de los sensores

```
65 | void loop() {
66 |     // Leer color
67 |     color();
68 |
69 |     // Leer distancia y determinar si hay un objeto cerca
70 |     objetoDetectado = medirTamaño();
```

En la figura 46, después de leer los valores de los sensores, se obtiene la posición del servomotor a través de Modbus. Los valores de posición son limitados entre 0 y 180 grados utilizando la función constrain(). Luego, el servo se mueve a la posición correspondiente.

Figura 46

Lectura de valores del servomotor

```
int pos1 = mb.Hreg(SERVO1_POS); //Lectura de registro
servo1.write(pos1); //Escritura de registros
Serial.print("Servo1: "); Serial.print(pos1);
delay(500);
```

En la figura 47, se envían los valores procesados como el color detectado y la proximidad del objeto a través de la red utilizando los registros Modbus. La función task() gestiona el ciclo de ejecución de la comunicación Modbus

Figura 47

Envío de valores procesados

```
93 // Enviar valores a Modbus
94 mb.Ireg(SENSOR_COLORVAL, colorValor);
95 mb.Ireg(SENSOR_TAMANIO, objetoDetectado);
96
97 // Ejecutar tarea Modbus
98 mb.task();
```

En la figura 48, se imprime en el monitor serie los valores de los sensores y la posición del servomotor, lo que permite visualizar el comportamiento del sistema en tiempo real.

Figura 48

Imprime valores en el monitor serial

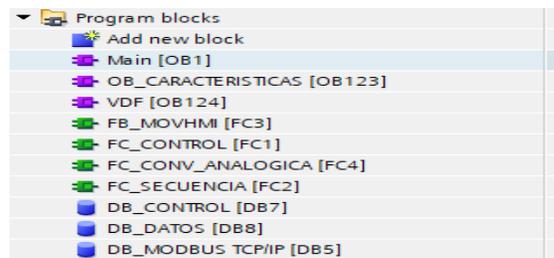
```
100 // Imprimir valores en el monitor serie
101 Serial.print("Color Valor: "); Serial.print(colorValor);
102 Serial.print(" Tamaño (1 = Objeto cerca): "); Serial.print(objetoDetectado);
103 Serial.print(" Servo1: "); Serial.print(posServo1);
```

5.5 Programación del TIA Portal

La figura 49, muestra los bloques de datos que se encargaran de interactuar entre sí para el funcionamiento del sistema además de almacenar los datos. Se realizo una programación de manera estructurada utilizando 3 bloques de organización (OB), 4 bloques de funciones (FC), y 3 bloques de datos (DB).

Figura 49

Bloque de datos utilizados en el TIA Portal V18.



En la figura 50, se puede observar el bloque de organización principal, el cual se encarga de llamar los bloques de funciones que darán comienzo a la comunicación entre bloque y así comenzando la interacción del sistema.

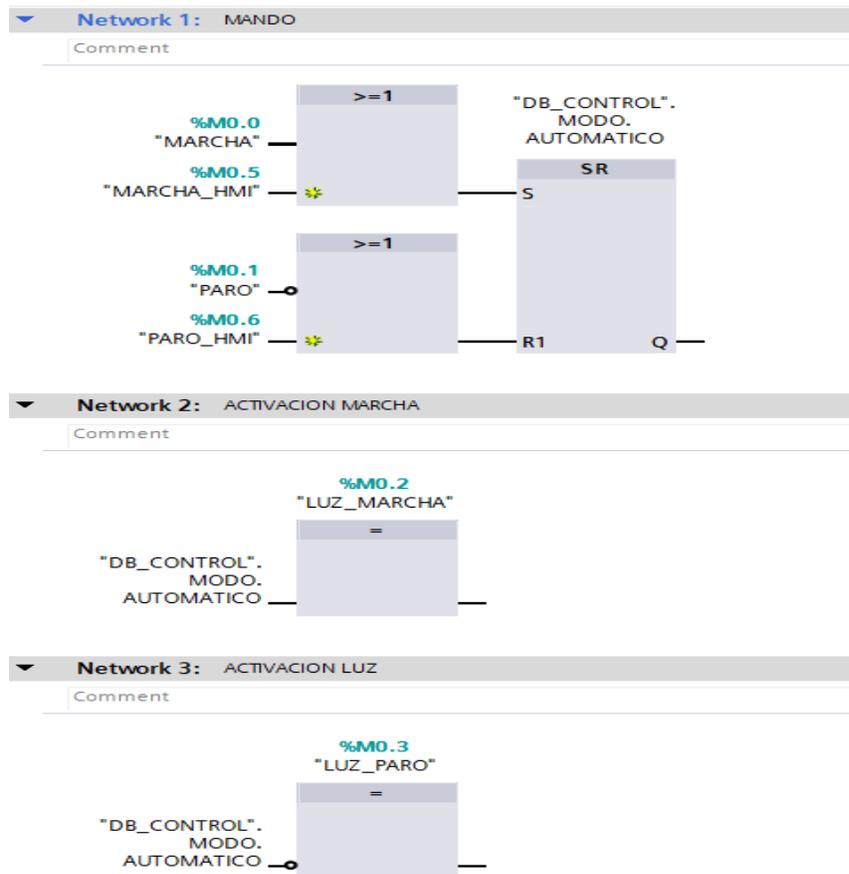
Figura 50

Bloque de organización "MAIN"



El bloque FC_CONTROL se encarga del mando de la operación, el cual cuenta con bloque SR (Set-Reset) para la activación y desactivación del modo automático. Además, este modo automático comanda el estado en el que las salidas que intervienen en el proceso actúan tal y como se evidencia en la figura 51.

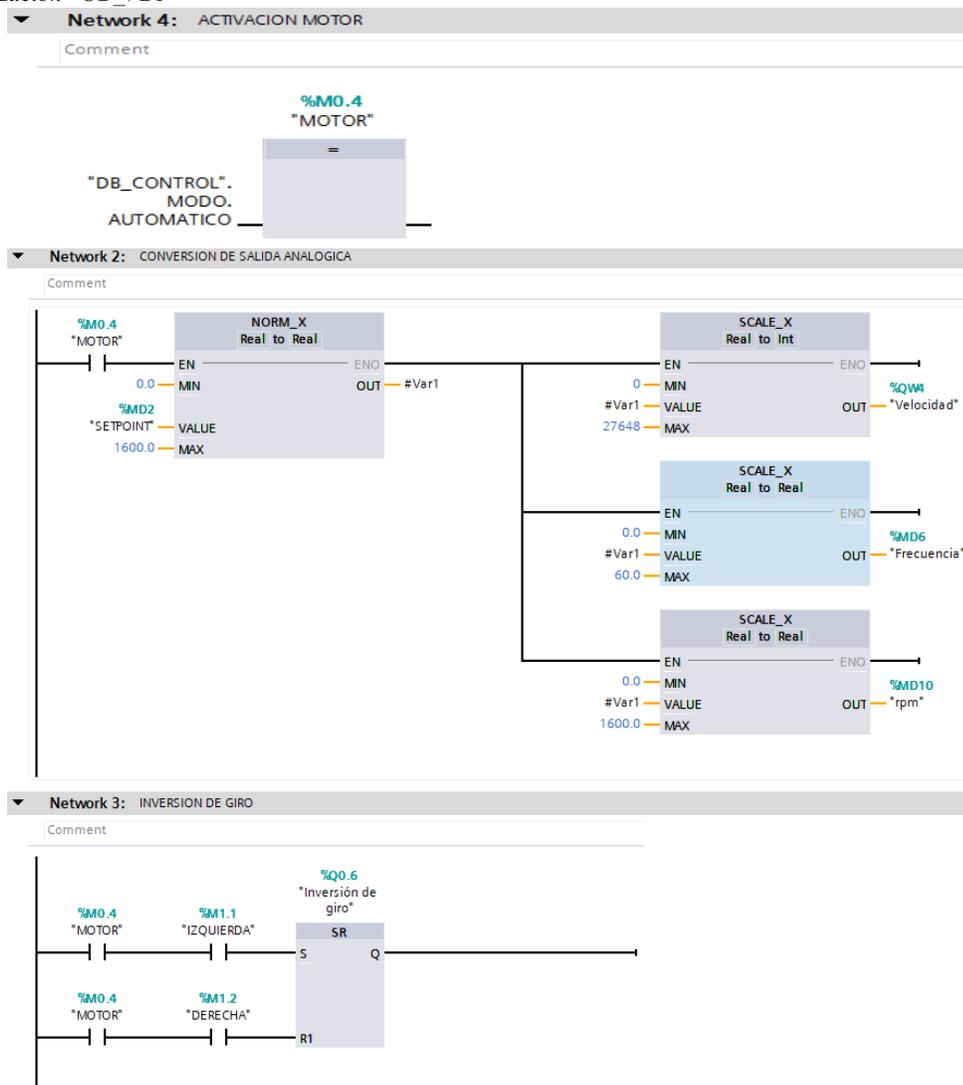
Figura 51
 Bloque de función "FC_CONTROL"



En la figura 52 se ve el control del motor el cual puede ser parametrizado a conveniencia además de que permite monitorear esos parámetros como la inversión de giro o la frecuencia de giro. Esto se puede realizar gracias al uso de los bloques de normalizado (NORM) en el que se coloca como valores máximo y mínimo las características del motor que se usará, en este caso será

de 0 – 1600 rpm mientras que el escalado (SCALE) realiza la conversión analógica que se recibe del normalizado y lo escala dentro del rango deseado en el mínimo y máximo. Por este motivo si yo quiero mostrar la frecuencia a la que está trabajando el motor debo poner la escala en la que trabaja el motor, en este caso de 0 a 60 Hz.

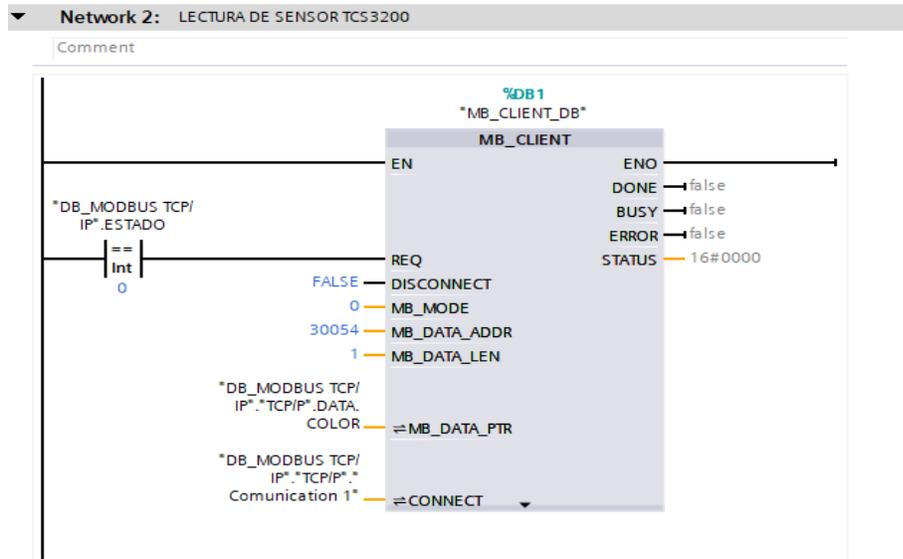
Figura 52
Bloque de organización "OB_VDF"



El bloque MB_CLIENT es el que permite establecer la comunicación MODBUS TCP/IP para el intercambio de datos entre el Arduino y el PLC figura 53. Estos parámetros del bloque se

los configura con la estructura de comunicación TCON_IP_v4 y se completa la información con respecto a la programación del Arduino. Es por eso por lo que en el parámetro MB_DATA_ADDR se coloca el 30054 por que el registro que se guarda en el microcontrolador es el 53.

Figura 53
Bloque MB_CLIENT



En la figura 54 se exhibe la programación que realiza la secuencia para que no haya conflicto entre los varios bloques de comunicación MB_CLIENT. Utilizando variables que traen por defecto los bloques como lo son las booleanas REQ, DONE y BUSY.

Figura 54
Programación SCL del OB_CHARACTERISTICAS

```
1 // SECUENCIA PARA EVITAR EL TRAFICO DE DATOS
2 CASE "DB_MODBUS TCP/IP".ESTADO OF
3
4 1:
5     IF NOT "MB_CLIENT_DB".BUSY AND NOT "MB_CLIENT_DB".DONE THEN
6         "MB_CLIENT_DB".REQ := TRUE;
7     ELSE
8         "MB_CLIENT_DB".REQ := FALSE;
9     END_IF;
10
11 2:
12     IF NOT "MB_CLIENT_DB_1".BUSY AND NOT "MB_CLIENT_DB_1".DONE THEN
13         "MB_CLIENT_DB_1".REQ := TRUE;
14     ELSE
15         "MB_CLIENT_DB_1".REQ := FALSE;
16     END_IF;
17
18 3:
19     IF NOT "MB_CLIENT_DB_2".BUSY AND NOT "MB_CLIENT_DB_2".DONE THEN
20         "MB_CLIENT_DB_2".REQ := TRUE;
21     ELSE
22         "MB_CLIENT_DB_2".REQ := FALSE;
23     END_IF;
24
25 4:
26     IF NOT "MB_CLIENT_DB_3".BUSY AND NOT "MB_CLIENT_DB_3".DONE THEN
27         "MB_CLIENT_DB_3".REQ := TRUE;
28     ELSE
29         "MB_CLIENT_DB_3".REQ := FALSE;
30     END_IF;
31
32 ELSE
33     "MB_CLIENT_DB".REQ := FALSE;
34     "MB_CLIENT_DB_1".REQ := FALSE;
35     "MB_CLIENT_DB_2".REQ := FALSE;
36     "MB_CLIENT_DB_3".REQ := FALSE;
37 END_CASE;
38
39 IF "MB_CLIENT_DB".DONE THEN
40     "DB_MODBUS TCP/IP".ESTADO := 1;
41 ELSIF "MB_CLIENT_DB_1".DONE THEN
42     "DB_MODBUS TCP/IP".ESTADO := 2;
43 ELSIF "MB_CLIENT_DB_2".DONE THEN
44     "DB_MODBUS TCP/IP".ESTADO := 3;
45 ELSIF "MB_CLIENT_DB_3".DONE THEN
46     "DB_MODBUS TCP/IP".ESTADO := 3;
47 ELSE
48     "DB_MODBUS TCP/IP".ESTADO := 0;
49 END_IF;
```

La programación que se ve en la figura 55, será la encargada de pedir y recibir los datos de los sensores para luego de eso proceder a la detección y clasificación con respecto a los valores enviados desde las recetas que se eligen. La secuencia se logra gracias al uso de las estructuras CASE, la cual sirve para evaluar variables y con respecto a eso realizar acciones, mientras que el IF indica las posibilidades de combinaciones a encontrar y el ELSE la acción que se realiza si la combinación no se cumple.

Figura 55

Programación SCL del FC_SECUENCIA

```
2 CASE "DB_MODBUS TCP/IP"."TCP/P".DATA.COLOR OF
3 // DETECCIÓN PARA PRODUCTOS DE COLOR ROJO
4 1:
5 IF ("DB_MODBUS TCP/IP"."TCP/P".DATA.TAMAÑO = "DB_DATOS".DATA_RECETA_1.TAMAÑO_RECETA_1) AND
6 ("DB_MODBUS TCP/IP"."TCP/P".DATA.PESO = "DB_DATOS".DATA_RECETA_1.PESO_RECETA_1) THEN
7 IF ("DB_DATOS".DATA_RECETA_1.COLOR_RECETA_1 = 1) THEN
8 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 180;
9 ELSE
10 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 0;
11 END_IF;
12 IF ("DB_DATOS".DATA_RECETA_2.COLOR_RECETA_2 = 1) THEN
13 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV02 := 180;
14 ELSE
15 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV02 := 0;
16 END_IF;
17 ELSE
18 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 0;
19 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV02 := 0;
20 END_IF;
21
22 // DETECCIÓN PARA PRODUCTOS DE COLOR AZUL
23 2:
24 IF ("DB_MODBUS TCP/IP"."TCP/P".DATA.TAMAÑO = "DB_DATOS".DATA_RECETA_1.TAMAÑO_RECETA_1) AND
25 ("DB_MODBUS TCP/IP"."TCP/P".DATA.PESO = "DB_DATOS".DATA_RECETA_1.PESO_RECETA_1) THEN
26 IF ("DB_DATOS".DATA_RECETA_1.COLOR_RECETA_1 = 2) THEN
27 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 90;
28 ELSE
29 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 0;
30 END_IF;
31 IF ("DB_DATOS".DATA_RECETA_2.COLOR_RECETA_2 = 2) THEN
32 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV02 := 90;
33 ELSE
34 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV02 := 0;
35 END_IF;
36 ELSE
37 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 0;
38 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV02 := 0;
39 END_IF;
40
41 // DETECCIÓN PARA PRODUCTOS DE COLOR VERDE
42 ..
```

5.6 Diseños de las interfaces del HMI

5.6.1 Tabla de variables del HMI

En la tabla de variables del HMI, figura 56, muestra las diferentes variables que están siendo monitoreadas o controladas a través de la interfaz, así como su estado o valor en un momento dado.

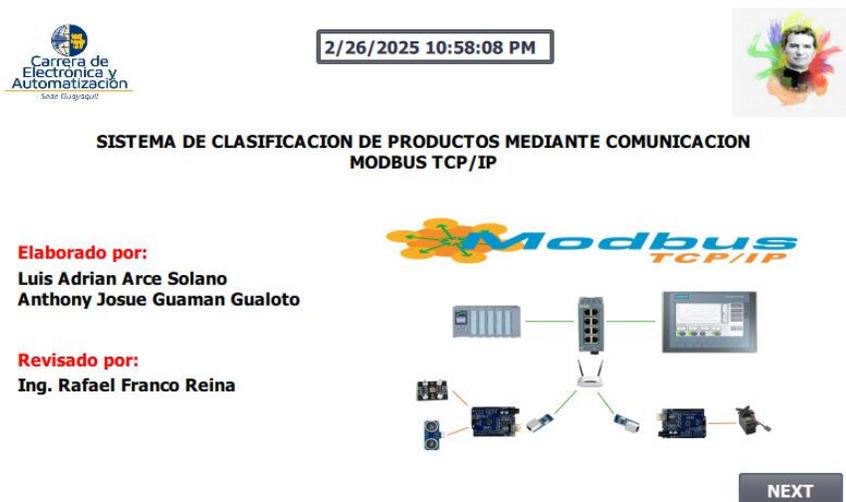
Figura 56
 Tabla de variables del HMI

Default tag table					
Name	Data type	Connection	PLC name	PLC tag	
Antihorario	Bool	HMI_Conne...	PLC_1	Antihorario	
CAJA	Int	HMI_Connectio...	PLC_1	CAJA	
DB_CONTROL_MODALO_AUTOM...	Bool	HMI_Connectio...	PLC_1	DB_CONTROL.MODO.AU...	
DB_DATOS_DATA_RECETA_1_C...	Int	HMI_Connectio...	PLC_1	DB_DATOS.DATA_RECET...	
DB_DATOS_DATA_RECETA_1_P...	Real	HMI_Connectio...	PLC_1	DB_DATOS.DATA_RECET...	
DB_DATOS_DATA_RECETA_1_T...	Bool	HMI_Connectio...	PLC_1	DB_DATOS.DATA_RECET...	
DB_DATOS_DATA_RECETA_1_T...	String	HMI_Connectio...	PLC_1	DB_DATOS.DATA_RECET...	
DB_DATOS_DATA_RECETA_2_P...	Real	HMI_Connectio...	PLC_1	DB_DATOS.DATA_RECET...	
DB_DATOS_DATA_RECETA_2_T...	Bool	HMI_Connectio...	PLC_1	DB_DATOS.DATA_RECET...	
DB_DATOS_DATA_RECETA_CO...	Int	HMI_Connectio...	PLC_1	DB_DATOS.DATA_RECET...	
DB_DATOS_DATA_RECETA_TIP...	String	HMI_Connectio...	PLC_1	DB_DATOS.DATA_RECET...	
Frecuencia	Real	HMI_Connectio...	PLC_1	Frecuencia	
Horario	Bool	HMI_Connectio...	PLC_1	Horario	
MARCHA_HMI	Bool	HMI_Connectio...	PLC_1	MARCHA_HMI	
MOTOR	Bool	HMI_Connectio...	PLC_1	MOTOR	
PARO_HMI	Bool	HMI_Connectio...	PLC_1	PARO_HMI	
rpm	Real	HMI_Connectio...	PLC_1	rpm	
SETPOINT	Real	HMI_Connectio...	PLC_1	SETPOINT	
Tag_ScreenNumber	UInt	<Internal tag>		<Undefined>	
<Add new>					

5.5.2 Pantalla de presentación

En la pantalla de presentación, figura 57, se muestra como una introducción visual y estructural a la aplicación, en ella detalla el título del proyecto junto con un gráfico representativo del sistema utilizado, los nombres de los tesistas y tutor, en la parte inferior derecha hay un botón que dirige hacia el inicio del proceso.

Figura 57
 Pantalla de presentación



5.5.3 Ventana de selección de interfaces

Esta pantalla es la que permite tener una vista previa además de dirigirse a cada pantalla seleccionando encima de cada recuadro, como se observa en la figura 58.

Figura 58

Ventana de selección de interfaces

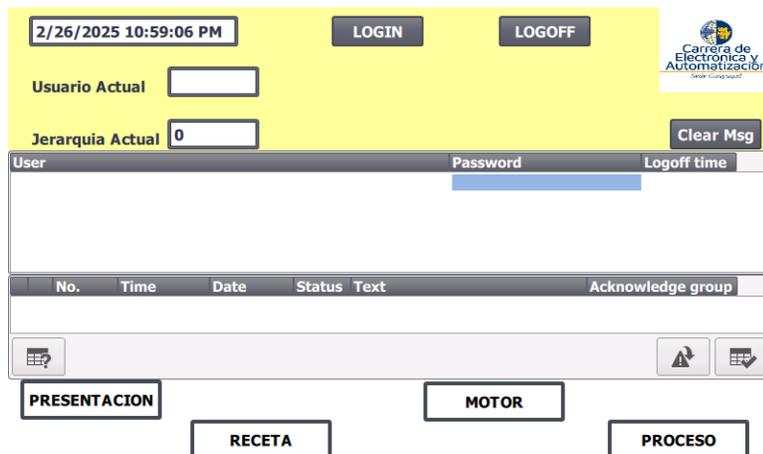


5.5.3 Pantalla de usuarios

La pantalla de usuarios muestra y permite un sistema de administración de usuarios, además, de llevar el registro de alarmas y movimientos que se realiza en el proceso y poder elegir la pantalla requerida a gusto del operador, figura 59.

Figura 59

Pantalla de administración de usuarios



5.5.4 Pantalla de parámetros de productos

La pantalla, figura 60, se divide en tres secciones, cada una destinada a configurar los parámetros de un servomotor específico. En la parte superior de cada sección, se encuentran dos listas desplegables: una para seleccionar la "RECETA SERVO" y otra para elegir el "TIPO DE PRODUCTO". Esto permite configurar tanto la receta de operación como el tipo de producto a clasificar.

Figura 60

Pantalla de parámetros - Tipos de productos

CARACTERÍSTICA	VALOR
COLOR	0
TAMAÑO	0
TIPO DE PRODUCTO	0
PESO	0

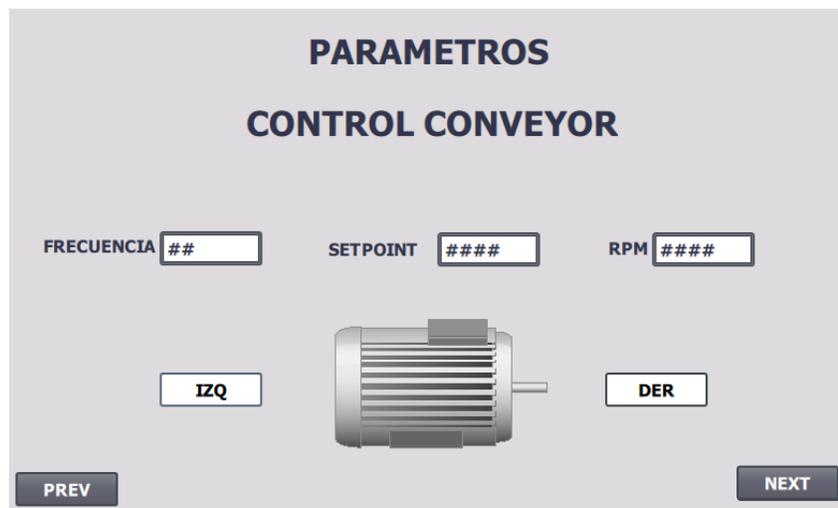
Debajo de las listas desplegables, se presenta una tabla con dos columnas: "CARACTERÍSTICA" y "VALOR". Esta tabla despliega información detallada sobre las propiedades del producto, como color, peso y altura, junto con sus respectivos valores. Además, hay un botón de descarga, lo que sugiere la opción de cargar o guardar configuraciones previamente definidas, facilitando la reconfiguración del sistema sin tener que ingresar manualmente todos los parámetros. En la parte inferior de la pantalla, los botones de navegación "PREV" y "NEXT" permiten moverse entre las distintas interfaces del HMI, siguiendo un flujo de trabajo lógico: primero se definen las recetas, luego se seleccionan los tipos de producto y finalmente se muestran sus características.

5.5.5 Pantalla de control del conveyor

En la figura 61, se observa la pantalla está diseñada para gestionar los parámetros operativos del motor de la banda transportadora. En la parte superior, se encuentran tres campos numéricos: FRECUENCIA, SETPOINT y RPM. El campo FRECUENCIA, con un valor predeterminado de 0, ajusta la frecuencia de operación del motor, lo que afecta directamente la velocidad de la banda. El campo SETPOINT, inicializado en 0, establece el objetivo de velocidad o posición del sistema, y RPM, también con valor predeterminado de 0, muestra la velocidad de rotación del motor en revoluciones por minuto, brindando una actualización constante del estado del motor.

Figura 61

Pantalla de parámetros - Control del conveyor



5.5.6 Pantalla de Proceso

La pantalla del proceso, figura 62, corresponde al monitoreo y control del sistema automatizado de clasificación. En la parte izquierda se encuentra el “Panel de monitoreo”, el cual permite visualizar el tipo de producto detectado y contadores asociados a cada categoría y al

reproceso. Además, con dos botones principales de “Marcha” para iniciar el sistema y” Paro” para detenerlo.

Figura 62
Pantalla del Proceso



A la derecha se observa una representación gráfica del sistema de clasificación, incluyendo una banda transportadora y un servomotor para desviar el producto según su categoría y en la esquina se ve el color, peso y tamaño requerido.

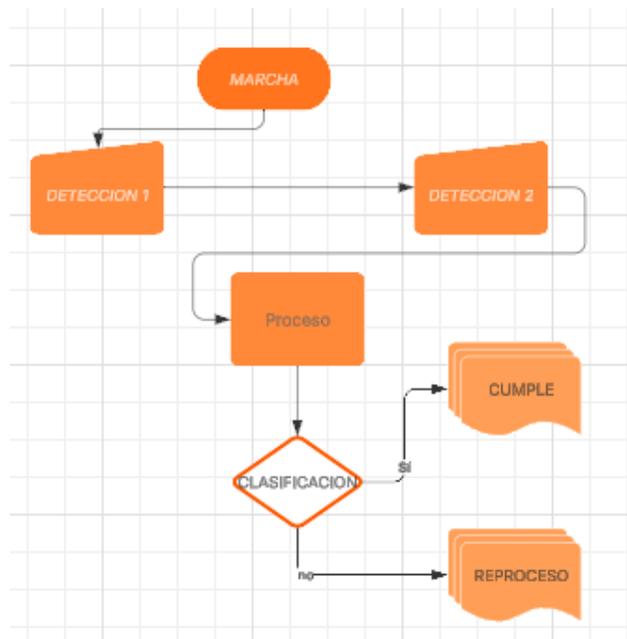
VI ANÁLISIS DE RESULTADOS

6.1 Evaluación del sistema

En este apartado se describe y demuestra las pruebas y modificaciones que se han realizado a lo largo de la puesta en marcha del prototipo de banda para clasificación de productos según sus características, en otras palabras, se comprobaba la fiabilidad y complementación de la parte mecánica y la parte del sistema de programación y la necesidad de ajustes. Es por ese motivo que se detalla desde la parte estructural del prototipo y el comportamiento de los algoritmos de programación, además de registrar los datos que arroja este tipo de pruebas para el análisis de rendimiento del sistema.

Se comienza desde la visión general del proceso, figura 63, esto refiere a la demostración y comprobación de la parte inicial del mismo. El algoritmo que obedece la programación, el contenido de la programación y hasta la forma que permite interactuar con el operador.

Figura 63
Diagrama de proceso



En la figura 64, se demuestra la forma general de la cual se riga este algoritmo, empezando desde la puesta en marcha para luego pasar a la fase de detección de características, en esta parte recibiendo la retroalimentación de los sensores el cual permite realizar la comparación de valores entre la receta del producto que se desea y los valores que estoy recibiendo para así determinar que clasificación deberá actuar o si el producto necesita entrar a una fase de reproceso.

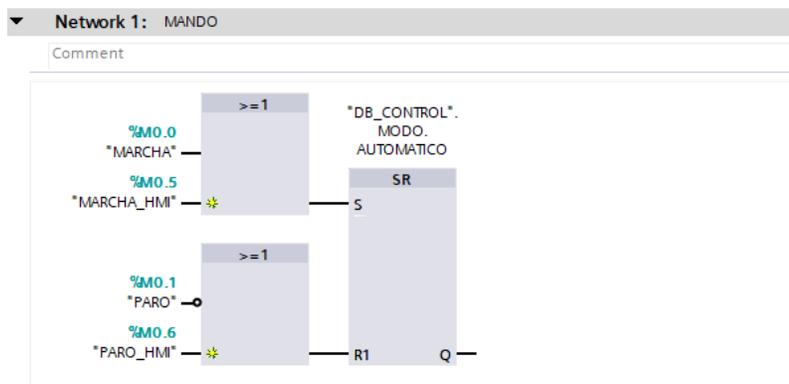
Figura 64
Marcha inicial



La principal habilitación del proceso, la marcha o mando se encarga de dar inicio al algoritmo dando resultado al cambio de estado de algunas salidas que intervienen tales como el motor o la comunicación. Desde el enfoque de la programación del controlador en el TIA PORTAL se puede encontrar este inicio en el FC_CONTROL, el cual se encuentra en programación FBD.

La figura 65, demuestra la lógica SR (Set-Reset) que se utiliza.

Figura 65
Lógica Set-Reset



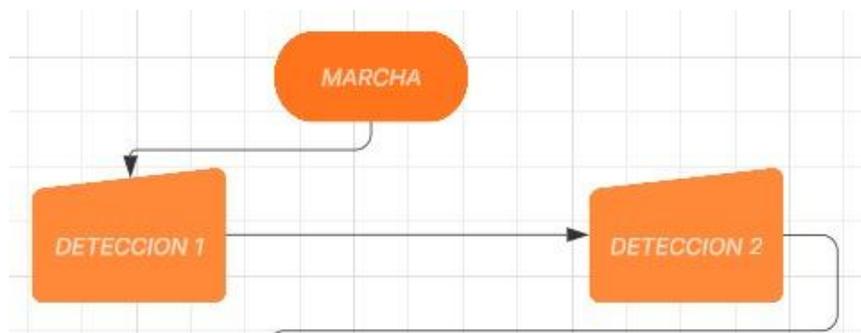
Mientras que, en el HMI, permite interactuar esta sección, dando inicio y paro al proceso y visualizar el conteo correcto de los productos, la cual es esencial, ya que permite al operador mantener bajo su control el estado de la operación, figura 66.

Figura 66
Interfaz de marcha y paro del operador



Como consecuencia de la marcha aparecen las detecciones, figura 67, que es la del sensor de color TCS3200 y ultrasónico HC-SR04, donde cada una está enfocada en la medición de parámetros del objeto a clasificar.

Figura 67
Fases de detección



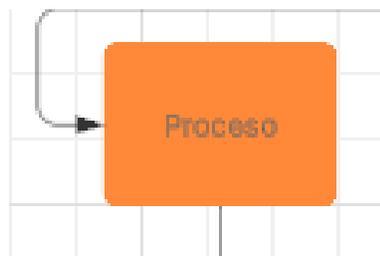
En esta etapa de detección 1, los sensores realizan mediciones de los objetos en tiempo real y los datos utilizados se envían al PLC mediante Modbus TCP/IP para su análisis, mientras que en la detección 2 el PLC recibe un valor de peso simulado establecido. Este análisis se realiza mediante los parámetros asignados en el apartado de recetas, como se ve en la figura 68.

Figura 68
Parámetros asignados para análisis

CARACTERISTICA	VALOR
COLOR	3
TAMAÑO	1
TIPO DE PRODUCTO	CAJA VERDE GR...
PESO	7

Luego de asignar los parámetros, viene el apartado de proceso, figura 69, donde una vez obtenidas todas las mediciones, el PLC procesa los datos recibidos para determinar si el objeto cumple con los parámetros establecidos.

Figura 69
Fase de procesamiento de datos



En esta fase, figura 70, se ejecutan cálculos y comparaciones utilizando las siguientes referencias:

- Color esperado según el producto a clasificar.
- Altura mínima y máxima permitida.
- Peso dentro del rango de tolerancia.

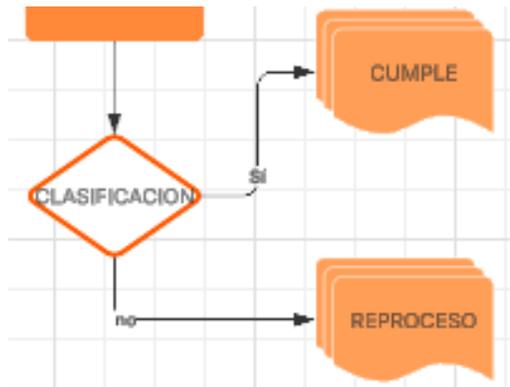
Figura 70
Clasificación según parámetros establecidos



El sistema realiza una evaluación lógica y determina si el objeto debe ser aceptado o rechazado. Esta decisión es tomada con base en reglas programadas en el software del PLC, el cual genera una señal de control para el servomotor encargado de la clasificación.

Por último, la evaluación de cumplimientos (clasificación), donde una vez analizados los datos, el sistema entra en fase de decisión, donde se establecen dos posibles resultados, como se ve en la figura 71.

Figura 71
Resultados de la clasificación

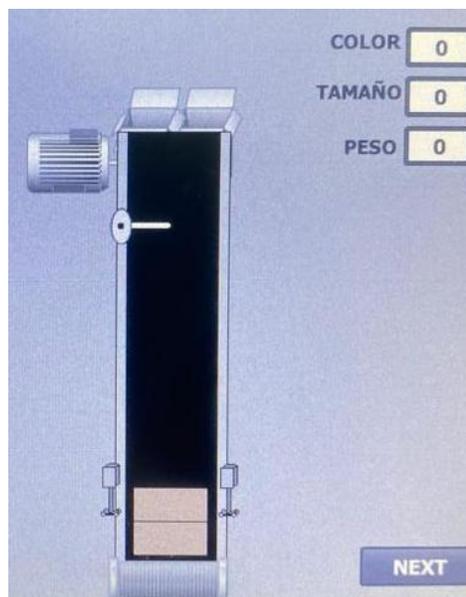


-Cumple con los requisitos: Si el objeto pasa todas las validaciones, el sistema lo dirige a la categoría "CUMPLE", asegurando que se encuentra dentro de los estándares de calidad requeridos.

-No cumple con los requisitos: Si el objeto presenta desviaciones en alguna de las mediciones, es enviado a "REPROCESO", donde puede ser analizado nuevamente o descartado.

Este proceso de clasificación, figura 72, se ejecuta mediante un servomotor que, según la decisión tomada por el PLC, redirigen el objeto hacia la ruta correspondiente.

Figura 72
Rutas de clasificación



6.2 Prueba del sensor ultrasónico en el prototipo

Para evaluar el funcionamiento del sistema automatizado, se realizaron 8 pruebas con objeto de instalación y calibración del sensor ultrasónico como se ve en la figura 73, durante este proceso, se presentaron diversos inconvenientes relacionados con la correcta fijación del sensor y su alineación con la cinta transportadora, como la altura debida del sensor sobre la banda, donde inicialmente se coloca 5 cm pero no reconocía el producto a esa distancia, al final se opta por una distancia de 10 cm donde detecta de una manera correcta.

Figura 73

Intento inicial de la instalación del sensor ultrasónico



En las primeras pruebas, se intentó colocar el sensor en una posición estratégica de 90° y pegado a la canaleta para que permitiera la detección de las cajas. Sin embargo, se observaron dificultades en la estabilidad del montaje, lo que generaba mediciones inconsistentes porque el pegamento no era muy bueno para sostener el sensor. Se realizaron ajustes en la estructura de soporte donde en vez de pegar el sensor, se coloca dentro de la canaleta y su detección sale por medio de una abertura.

Posteriormente, en la figura 74, se llevó a cabo una nueva prueba, sacando el sensor de la canaleta para verificar que no haya dañado su funcionamiento y se lo coloca desde un ángulo de 45° y con una caja de distinto tamaño para verificar la respuesta del sensor. Durante esta prueba, se corroboró su funcionamiento a pesar de los cambios por la inclinación o la distancia del sensor respecto a la superficie de las cajas.

Figura 74

Prueba desde diferente ángulo con una caja de diferente color



Finalmente, en la figura 75, tras intentos y modificaciones se volvió a colocar el sensor dentro de la canaleta a la altura de 35cm por sobre la banda, se logró una correcta fijación del sensor ultrasónico. Este avance permitió que el sistema detectara de manera correcta los objetos en la cinta transportadora,

Figura 75

Instalación correcta del sensor ultrasónico



6.2 Prueba del sensor de color en el prototipo

Para estas pruebas, las cuales fueron 10 se instaló el sensor de color TCS3200 dentro del prototipo, figura 76. El objetivo es verificar su capacidad para identificar correctamente los colores de los objetos y analizar su respuesta en condiciones de operación.

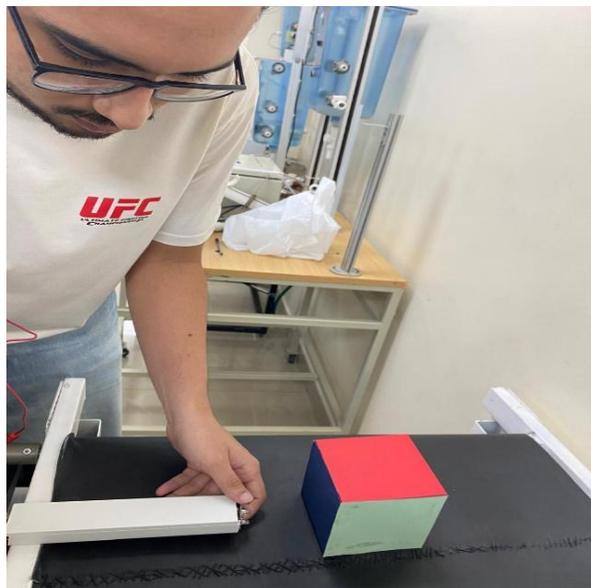
Figura 76

Instalación del sensor TCS3200 en el prototipo



Inicialmente, se realizó una prueba manual del sensor para comprobar su funcionamiento básico, figura 77. Se sostuvo el sensor sobre la banda transportadora para verificar su capacidad de lectura de colores en diferentes superficies. En esta fase, se utilizó un cubo con distintas caras de colores para analizar la precisión del sensor. Se acercó cada cara al sensor y se registraron los valores de salida, de rojo=1, azul=2 y verde=3, asegurando su detección es correcta siempre que esté dentro de una distancia de no más de 5cm con el objeto. Esta prueba permitió verificar que el sensor podía diferenciar con claridad los colores presentes en los objetos que serán clasificados en el sistema.

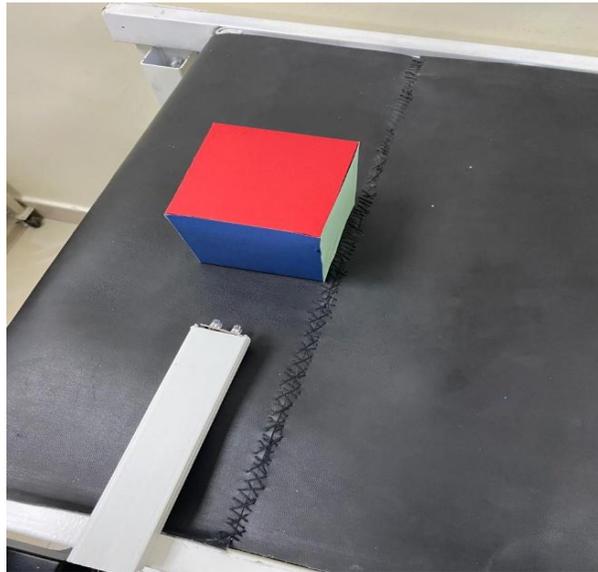
Figura 77
Prueba manual para verificar el funcionamiento



Finalmente, en la figura 78, se realizó una prueba en condiciones reales, con el sensor montado ya fijado en la estructura del sistema dentro de una canaleta de 20 cm y la banda transportadora en movimiento. Se observaron aspectos como la estabilidad de las mediciones y la necesidad de calibración para optimizar la detección.

Figura 78

Prueba automática con la banda en movimiento



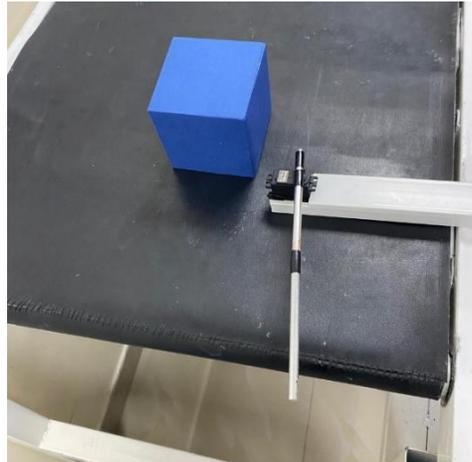
6.3 Prueba del servomotor en el prototipo

Para las pruebas del servomotor, se realizaron 18 muestras. Se conectó el servomotor a la fuente de alimentación y al microcontrolador, permitiendo así la comunicación y el control del movimiento mediante señales PWM por momentos ya que por la conexión de varios jumpers el servo comenzó a vibrar por la insuficiente llegada de voltaje.

En la prueba inicial, figura 79, el servomotor se conectó al Arduino con un módulo regulador de voltaje LM2596 y con cables n°22, se lo programó para moverse en ángulos de 0 a 90°, verificando su precisión y tiempo de respuesta de unos 3 segundos. Se utilizó un brazo acoplado al eje del servomotor para realizar la acción de empuje sobre un objeto de prueba.

Figura 79

Prueba de movimiento de ángulos del servomotor en su posición inicial



Las imágenes muestran el proceso de prueba, donde el servomotor desplaza un cubo de diferentes colores. En la primera imagen, el brazo del servomotor se encuentra en posición inicial de 0° antes de realizar el empuje. En la segunda imagen, figura 80, el servomotor está realizando un movimiento a una posición de 90° para aplicar la fuerza necesaria, moviendo el objeto hacia una nueva posición.

Figura 80

Caja en movimiento a nueva posición



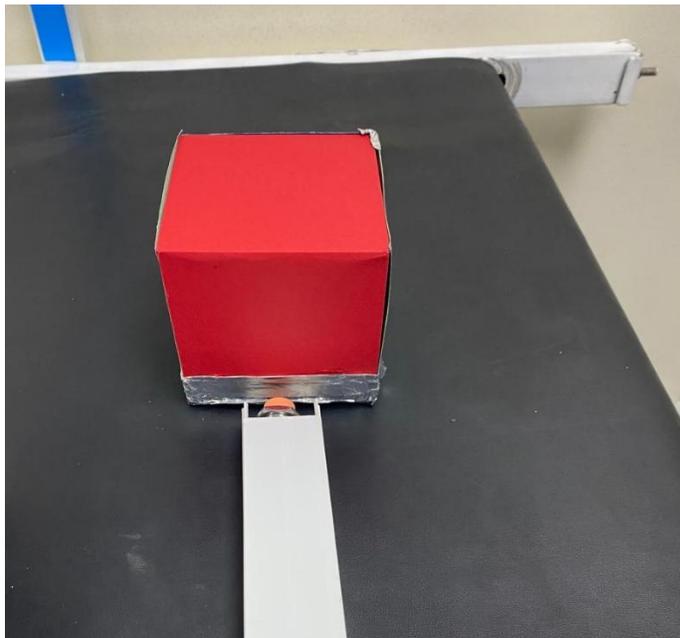
Esta prueba permitió comprobar el correcto funcionamiento del sistema de clasificación, asegurando que el servomotor ejecuta la tarea de desplazamiento de manera correcta y con la fuerza adecuada para modificar la ubicación del objeto sin generar inestabilidad en la estructura.

6.4 Prueba del sensor inductivo en el prototipo

Para las pruebas del sensor inductivo, 8 pruebas permitieron ajustar la calibración del sensor en la estructura de soporte colocada al inicio del proceso de la banda, asegurando su fijación para evitar desplazamientos no deseados, así, se conectó el sensor a la fuente de alimentación, al neutro y a la señal, permitiendo la comunicación del sensor con el TIA Portal para la detección del metal y detener el avance por unos 3

En la prueba inicial, figura 81, el sensor se conectó al PLC y se lo programó para que al detectar detenga el proceso durante 3 segundos.

Figura 81
Prueba de sensor inductivo en el prototipo



6.5 Pruebas de recetas

6.5.1 Receta 1: Caja grande roja

Antes de iniciar las pruebas, se configuró la receta en la interfaz HMI, figura 82, estableciendo que la caja a clasificar debía ser de color rojo y de tamaño grande. La imagen a continuación muestra la configuración de la receta en el HMI.

Figura 82

Selección de caja roja grande en la receta

CARACTERISTICA	VALOR
COLOR	1
TAMAÑO	1
TIPO DE PRODUCTO	CAJA ROJA GRA...
PESO	7

Se colocó una caja grande y roja en la banda transportadora, la cual avanzó hasta ser detectada por el sensor inductivo. Este detuvo la banda para que los sensores de color y ultrasónico realizaran sus mediciones. El sensor ultrasónico confirmó que la caja era grande, enviando una señal de "objeto detectado: sí", mientras que el sensor de color identificó el tono rojo, asignándole el valor 1. Estos datos fueron enviados al Arduino y al PLC para su procesamiento.

Como la caja cumplía con los criterios de la receta, el sistema activó el servomotor, desviando la caja hacia una esquina de la banda para su clasificación. Las pruebas demostraron que el sistema es capaz de identificar y clasificar correctamente las cajas grandes y rojas. El número de muestras fue de cinco para esta receta.

6.5.2 Receta 2: Caja pequeña azul

Para estas pruebas, se configuró la receta en el HMI, figura 83, indicando que la caja a clasificar debía ser de color azul y de tamaño grande. La imagen a continuación muestra la configuración de la receta en el HMI. Fueron 4 las muestras que se realizaron en esta receta.

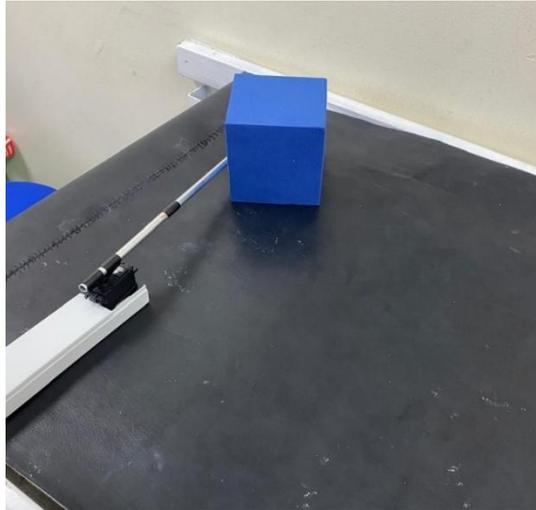
Figura 83

Selección de caja azul pequeña en la receta

CARACTERISTICA	VALOR
COLOR	2
TAMAÑO	0
TIPO DE PRODUCTO	CAJA AZUL PEQ...
PESO	2

Se colocó una caja pequeña y azul en la banda transportadora. Al ser detectada por el sensor inductivo, la banda se detuvo brevemente para que los sensores realizaran sus mediciones. El sensor ultrasónico determinó que la caja era pequeña, enviando una señal de "objeto detectado: no", mientras que el sensor de color identificó el tono azul, asignándole el valor 2. Estos datos fueron enviados al Arduino y al PLC para su análisis.

Figura 84
Clasificación de caja azul pequeña



En la figura 84, dado que la caja cumplía con el criterio de tamaño pequeño, el sistema activó el servomotor, permitiendo que la caja continuara su camino y ser clasificada. La prueba confirmó que el sistema ignora adecuadamente las cajas que no cumplen con los requisitos establecidos.

6.5.3 Receta 3: Caja grande verde

En estas pruebas, se configuró la receta en el HMI, figura 85, para clasificar cajas de color verde y tamaño grande. La imagen a continuación muestra la configuración de la receta en el HMI.

Figura 85
Selección de caja verde grande en la receta

CARACTERISTICA	VALOR
COLOR	3
TAMAÑO	1
TIPO DE PRODUCTO	CAJA VERDE GR...
PESO	7

Se colocó una caja grande y verde en la banda transportadora. Al avanzar, fue detectada por el sensor inductivo, el cual detuvo la banda para que los sensores realizaran sus mediciones. El sensor ultrasónico confirmó que la caja era grande, enviando una señal de "objeto detectado: sí", mientras que el sensor de color identificó el tono verde, asignándole el valor 3. Estos datos fueron enviados al Arduino y al PLC para su procesamiento.

Como la caja cumplía con los criterios de la receta, el sistema activó el servomotor, desviando la caja hacia una esquina de la banda para su clasificación. La prueba fue exitosa, validando que el sistema puede identificar y clasificar correctamente las cajas grandes y verdes, es así que, luego de 4 pruebas se concluyeron con los ajustes antes mencionados

6.5.4 Receta 4: Caja pequeña roja

Para estas pruebas, las cuales fueron 4 se configuró la receta en el HMI, figura 86, para clasificar cajas de color rojo y tamaño grande. La imagen a continuación muestra la configuración de la receta en el HMI.

Figura 86

Selección de caja roja pequeña en la receta

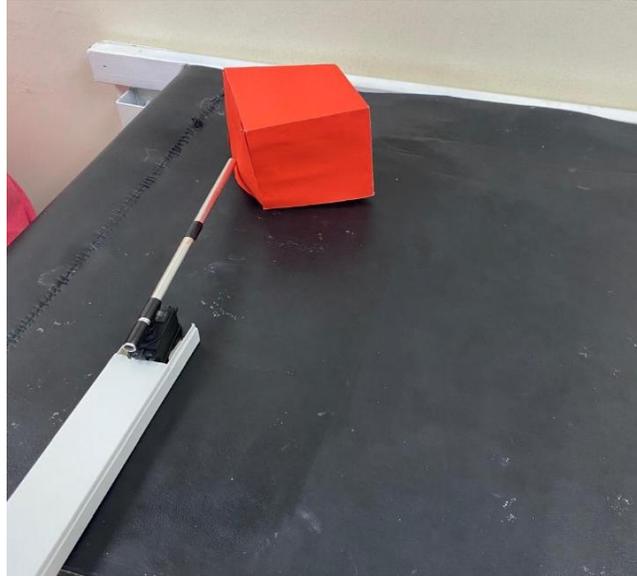
CARACTERISTICA	VALOR
COLOR	1
TAMAÑO	0
TIPO DE PRODUCTO	CAJA ROJA PEQ...
PESO	2

Se colocó una caja pequeña y roja en la banda transportadora, figura 87, al ser detectada por el sensor inductivo, la banda se detuvo para permitir que los sensores realizaran sus

mediciones. El sensor ultrasónico determinó que la caja era pequeña, enviando una señal de "objeto detectado: no", mientras que el sensor de color identificó el tono rojo, asignándole el valor 1. Estos datos fueron enviados al Arduino y al PLC para su análisis.

Figura 87

Clasificación de caja roja pequeña



Dado que la caja cumplía con el criterio de tamaño pequeño, el sistema activó el servomotor, permitiendo que la caja siguiera su camino hasta ser clasificada. La prueba demostró que el sistema es capaz de ignorar correctamente las cajas que no cumplen con los requisitos establecidos y clasificar la caja con las características pedidas.

6.5.5 Receta 5: Caja grande azul

En esta prueba, se configuró la receta en el HMI, figura 88, para clasificar cajas de color azul y tamaño grande. La imagen a continuación muestra la configuración de la receta en el HMI.

Figura 88

Selección de caja azul grande en receta

CARACTERISTICA	VALOR
COLOR	2
TAMAÑO	1
TIPO DE PRODUCTO	CAJA AZUL GRA...
PESO	7

Se colocó una caja grande y azul en la banda transportadora, la cual avanzó hasta ser detectada por el sensor inductivo. Este detuvo la banda para que los sensores realizaran sus mediciones. El sensor ultrasónico confirmó que la caja era grande, enviando una señal de "objeto detectado: sí", mientras que el sensor de color identificó el tono azul, asignándole el valor 2. Estos datos fueron enviados al Arduino y al PLC para su procesamiento.

Como la caja cumplía con los criterios de la receta, el sistema activó el servomotor, desviando la caja hacia una esquina de la banda para su clasificación. La prueba fue exitosa, confirmando que el sistema puede identificar y clasificar correctamente las cajas grandes y azules. Se realizaron 4 muestras para esta receta.

6.5.6 Receta 6: Caja pequeña verde

Para estas pruebas, se realizaron 4 muestras que dio como resultado la configuración en la receta en el HMI, figura 89, para clasificar cajas de color verde y tamaño grande. La imagen a continuación muestra la configuración de la receta en el HMI.

Figura 89
Selección de caja verde pequeña en la receta

**PARAMETROS
TIPOS DE PRODUCTO**

RECETA SERVO #1
RECETA_SERVO#1

TIPOS DE PRODUCTO #1
VERDE P

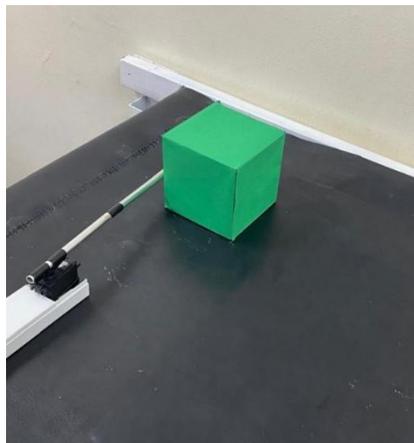
CARACTERISTICA	VALOR
COLOR	3
TAMAÑO	0
TIPO DE PRODUCTO	CAJA VERDE PE...
PESO	2

PREV NEXT

Se colocó una caja pequeña y verde en la banda transportadora. Al ser detectada por el sensor inductivo, la banda se detuvo para permitir que los sensores realizaran sus mediciones. El sensor ultrasónico determinó que la caja era pequeña, enviando una señal de "objeto detectado: no", mientras que el sensor de color identificó el tono verde, asignándole el valor 3. Estos datos fueron enviados al Arduino y al PLC para su análisis.

Dado que la caja incumplía el criterio de tamaño grande, el sistema activó el servomotor, permitiendo que la caja continuara su camino y ser clasificada como se ve en la figura 90.

Figura 90
Clasificación de la caja verde pequeña



6.6 Tabla de resultados

A continuación, en la tabla 1, se presenta el análisis de errores y rendimiento del sistema, gracias a las pruebas realizadas a los distintos componentes del prototipo, incluyendo el sensor TCS3200 para clasificación de colores, el sensor inductivo, el sensor ultrasónico y el servomotor. Se muestran los valores reales versus los valores detectados por cada componente, junto con el porcentaje de error asociado.

Tabla 1

Tabla de análisis y rendimiento de los sensores y actuadores

Componente	Muestra	Valor real	Valor detectado	Error %	Conclusión
Sensor TCS3200 (5 cm)	1	Rojo	Rojo	0 %	Detección correcta de colores a corto alcance.
	2	Verde	Verde	0 %	Responde de manera consistente bajo condiciones normales de luz.
	3	Azul	Azul	0 %	Baja tasa de error en la clasificación de colores.
	4	Rojo	Rojo	0 %	Mantiene precisión en diferentes pruebas.
	5	Verde	Verde	0 %	Precisión alta en múltiples pruebas.
	6	Azul	Azul	0 %	Correcta identificación de colores bajo iluminación artificial.

	7	Rojo	Verde	100 %	Error debido a lo alejado del objeto.
	8	Azul	Azul	0 %	Consistencia en la detección del color azul.
	9	Verde	Azul	100 %	Confusión ocasional en colores oscuros.
	10	Verde	Verde	0 %	Detección correcta en condiciones controladas.
Sensor inductivo	1	5 cm	No detecta	0 %	No detecta objeto debido al alcance del sensor
	2	4 cm	No detecta	0 %	No detecta objeto debido al alcance del sensor
	3	3 cm	No detecta	0%	No detecta objeto debido al alcance del sensor
	4	2 cm	No detecta	0%	No detecta objeto debido al alcance del sensor
	5	1 cm	Detecta	100%	Detecta el objeto
	6	1 cm	No detecta	0%	No detecta el objeto por desviación
	7	1.5 cm	Detecta	100%	Detecta el objeto
	8	1.5 cm	Detecta	100%	Detecta el objeto
Sensor HC-SR04 (35 cm de altura)	1	5 cm	5 cm	0 %	Objeto no detectado
	2	6 cm	6 cm	0%	Objeto no detectado
	3	7 cm	7 cm	0%	Objeto no detectado

	4	8 cm	8 cm	0%	Objeto no detectado
	5	9 cm	9 cm	0%	Objeto no detectado
	6	10 cm	10 cm	100%	Objeto detectado
	7	9.5 cm	9.5 cm	100%	Objeto detectado
	8	11 cm	11 cm	100%	Objeto detectado
Servomotor S-3003	1	90°	0°	100%	No hay comunicación
	2	90°	0°	100%	No hay comunicación
	3	90°	0°	100%	No hay movimiento, si comunicación
	4	90°	0°	100%	No hay movimiento, si comunicación
	5	90°	90°	0%	Hay movimiento y comunicación luego de alimentación independiente de servomotor
	6	90°	90°	0%	Posicionamiento correcto
	7	90°	90°	0%	Posicionamiento correcto
	8	90°	90°	0%	Posicionamiento correcto, pero existe perturbación en el movimiento
	9	90°	90°	0%	Posicionamiento correcto, pero existe perturbación en el movimiento
	10	90°	0°	100%	Perdida de voltaje debido a empate en cables

11	90°	90°	0%	Alimentación con calibre de cable mayor. Movimientos correctos
12	90°	90°	0%	Posicionamiento correcto, pero existe perturbación en el movimiento
13	90°	90°	0%	Cambio de acople más ligero acabo con la perturbación
14	90°	90°	0%	Movimiento y comunicación correcta
15	90°	90°	0%	Movimiento y comunicación correcta
16	90°	90°	0%	Movimiento y comunicación correcta
17	90°	90°	0%	Movimiento y comunicación correcta
18	90°	90°	0%	Movimiento y comunicación correcta

En las pruebas realizadas el sensor TCS3200 mostró una alta precisión en la clasificación de colores rojo, verde y azul, con un 100 % de aciertos en 8 de las 10 pruebas realizadas. En condiciones normales de iluminación, la detección de colores fue consistente y sin errores. Sin embargo, en dos casos (muestras 7 y 9), el sensor presentó confusión en la detección de colores debido a la distancia del objeto y a la tonalidad oscura de la superficie. En general, la tasa de precisión del sensor fue del 80 %, con un 20 % de error en condiciones desfavorables como distancia variable o iluminación insuficiente.

El sensor inductivo presentó limitaciones en su rango de detección, mostrando que solo es capaz de identificar objetos cuando están a 1 cm o menos de distancia. En las pruebas realizadas con distancias mayores a 1 cm, el sensor no detectó el objeto en ninguno de los casos, lo que indica una dependencia estricta de la distancia. En pruebas con objetos ubicados exactamente a 1 cm, el sensor tuvo un 75 % de efectividad, aunque en una de las pruebas no logró detectar el objeto debido a una posible desviación en la colocación. La precisión del sensor en su rango de detección óptimo fue del 100 %, pero fuera de este rango su desempeño es nulo.

El servomotor presentó fallos en las primeras pruebas debido a problemas en la alimentación y el cableado. En 4 de las pruebas iniciales, el servomotor no se movió debido a la falta de comunicación, mientras que en otras 2 pruebas la comunicación se restableció, pero no hubo movimiento. A partir de la alimentación independiente y el uso de un cable de mayor calibre, el servomotor logró ejecutar movimientos correctos en 14 pruebas consecutivas (93.3 % de efectividad). No obstante, se registraron perturbaciones en el movimiento en 3 de las pruebas, lo que sugiere la necesidad de un ajuste en la estabilidad del sistema mecánico. La solución fue reemplazar el acople del servomotor por una versión más ligera, lo que eliminó la perturbación y permitió un control preciso del posicionamiento en todas las pruebas siguientes.

El sensor ultrasónico HC-SR04 demostró una alta precisión en la detección, lo que llevó a un proceso de ajuste progresivo en los parámetros de detección. A medida que se calibraron los valores y se establecieron umbrales más adecuados, el sensor logró alcanzar una precisión del 100 % en la detección. Esto indica que el sensor es altamente confiable, permitiendo obtener mediciones estables en todo su rango operativo.

El análisis de pruebas demuestra que el sensor de color TCS3200 es altamente confiable en condiciones controladas, con un rendimiento del 80 %, pero requiere ajustes en condiciones de baja iluminación o superficies oscuras. El sensor inductivo solo es efectivo en un rango de 1 cm, con un 100 % de precisión en distancias ideales, pero sin detección a mayores distancias. Finalmente, el servomotor logró un 93.3 % de efectividad, con fallos iniciales que se resolvieron mediante ajustes en la alimentación y el acople.

En la tabla 2, de esta sección, se muestra la tabla de los resultados obtenidos dentro de las pruebas realizadas de la comunicación entre las recetas enviadas por el PLC S7-1500 y Arduino, además de la detección de los objetos.

Tabla 2
Tabla de resultados prototipo

PRUEBA	RECETA	TAMAÑO	COLOR	PESO	SERVO	RESULTADO
1	Objeto grande rojo	Grande	Rojo (1)	Objeto pesado: si	No activado	Fallido, el sensor inductivo no detectó el metal de la caja
2	Objeto grande rojo	Grande	Rojo (1)	Objeto pesado: sí	No activado	Fallido, la frecuencia del motor no fue suficiente para detener la caja en los sensores
3	Objeto pequeño azul	Pequeño	Azul (2)	Objeto pesado: no	No activado	Fallido, los sensores no tuvieron el suficiente tiempo para detectar.
4	Objeto pequeño azul	Pequeño	Azul (2)	Objeto pesado: no	No activado	Fallido, los sensores no tuvieron el suficiente tiempo para detectar.

5	Objeto grande verde	Grande	Verde (3)	Objeto pesado: sí	No activado	Fallido, los sensores no tuvieron el suficiente tiempo para detectar.
6	Objeto grande verde	Grande	Verde (3)	Objeto pesado: sí	Activado	Correcto
7	Objeto pequeño rojo	Pequeño	Rojo (1)	Objeto pesado: no	No activado	Fallido, el acople del motor se salió de su posición
8	Objeto pequeño rojo	Pequeño	Rojo (1)	Objeto pesado: no	Activado	Correcto
9	Objeto grande azul	Grande	Azul (2)	Objeto pesado: sí	Activado	Fallido, el servomotor se salió de su posición por el tamaño de la caja.
10	Objeto grande azul	Grande	Azul (2)	Objeto pesado: sí	Activado	Correcto
11	Objeto pequeño verde	Pequeño	Verde (3)	Objeto pesado: no	No activado	Fallido, el servomotor no recibía comunicación por la pérdida de voltaje
12	Objeto pequeño verde	Pequeño	Verde (3)	Objeto pesado: no	Activado	Correcto
13	Objeto grande rojo	Grande	Rojo (1)	Objeto pesado: sí	Activado	Correcto
14	Objeto grande rojo	Grande	Rojo (1)	Objeto pesado: sí	Activado	Correcto.
15	Objeto grande rojo	Grande	Rojo (1)	Objeto pesado: sí	Activado	Retraso en la comunicación, no hubo clasificación.

16	Objeto pequeño azul	Pequeño	Azul (2)	Objeto pesado: no	Activado	Correcto.
17	Objeto pequeño azul	Pequeño	Azul (2)	Objeto pesado: no	No activado	Fallido, el sensor de color confundió el color
18	Objeto grande verde	Grande	Verde (3)	Objeto pesado: si	Activado	Correcto.
19	Objeto grande verde	Grande	Verde (3)	Objeto pesado: sí	Activado	Correcto.
20	Objeto pequeño rojo	Pequeño	Rojo (1)	Objeto pesado: no	Activado	Correcto.
21	Objeto pequeño rojo	Pequeño	Rojo (1)	Objeto pesado: no	Activado	Correcto.
22	Objeto grande azul	Grande	Azul (2)	Objeto pesado: si	Activado	Correcto.
23	Objeto grande azul	Grande	Azul (2)	Objeto pesado: sí	Activado	Correcto.
24	Objeto pequeño verde	Pequeño	Verde (3)	Objeto pesado: no	Activado	Correcto.
25	Objeto pequeño verde	Pequeño	Verde (3)	Objeto pesado: no	Activado	Correcto.

El análisis de 25 pruebas mostró un 68 % de éxito en la clasificación de objetos, mientras que en 32 % de los casos se identificaron fallos por detección incorrecta, comunicación deficiente o problemas mecánicos. Se observó que en 4 pruebas los sensores no tuvieron suficiente tiempo para detectar, lo que sugiere ajustes en la velocidad de la banda. El servomotor presentó fallas en 4 pruebas (16 % del total) debido a caídas de voltaje, retrasos en la comunicación o desajustes mecánicos del prototipo. Los objetos grandes rojos y pequeños azules tuvieron la menor tasa de

éxito (60 %), mientras que los grandes azules y pequeños verdes lograron una clasificación perfecta (100 %).

En la tabla 3 se muestra las estadísticas generales del sistema con todas las muestras realizadas a lo largo del proceso de pruebas.

Tabla 1

Tabla de estadísticas generales del sistema

COMPONENTE	PRUEBAS REALIZADAS	ÉXITO (%)	FALLOS (%)	Conclusiones
Sensor TCS3200	10	80%	20%	Confusión en colores oscuros y objetos alejados.
Sensor Inductivo	8	50%	50%	No detecta objetos a más de 1.5 cm.
Sensor HC-SR04	8	50%	50%	Se debe ajustar en la programación la distancia a detectar
Servomotor S3003	18	78%	22%	Problemas de comunicación y alimentación.
Clasificación de objetos	25	68%	32%	Fallas en detección, tiempo de respuesta y comunicación.

VII PRESUPUESTO

En la tabla 4, detalla el costo de los materiales a utilizar para el desarrollo del diseño del proyecto, con los cuales ciertos materiales y elementos que se utilizaron en este proyecto, se dispone en el Laboratorio de Automatización II de la Universidad Politécnica Salesiana, los cuales serían Motor trifásico, PLC S7-1500, Actuador Eléctrico, VDF.

Tabla 2
Tabla de Presupuesto Del Desarrollo Del Proyecto

CANTIDAD	MATERIALES	PRECIO UNITARIO	TOTAL
2	Arduino UNO	\$15.00	\$30.00
1	Sensor TCS3200	\$12.00	\$12.00
2	Módulo LM2596	\$5.00	\$10.00
2	Módulo ENC28J60	\$10.00	\$20.00
1	Sensor HC-SR04	\$3.00	\$3.00
40	Jumpers	\$0.10	\$4.00
2	Servomotor	\$8.00	\$16.00
60	Horas de ingeniería	\$2.87	\$172.2
1	Prototipo banda transportadora	\$ 150.00	\$150.00
TOTAL			\$ 417.2

VIII CRONOGRAMA

En la figura 91 se presenta el cronograma de actividades a realizar durante el desarrollo de la tesis.

Figura 91

Cronograma de Gantt de actividades del desarrollo de la tesis

Actividades	SEPTIEMBRE					OCTUBRE				NOVIEMBRE					DICIEMBRE				
	S1	S2	S3	S4	S5	S1	S2	S3	S4	S1	S2	S3	S4	S5	S1	S2	S3	S4	
Diseño del anteproyecto	■	■	■																
Revisión bibliográfica sobre sistemas Modbus			■	■															
Diseño del sistema de clasificación				■	■														
Selección de componentes y hardware						■													
Configuración del protocolo Modbus							■	■											
Programación del PLC								■	■	■									
Integración de sensores y actuadores											■	■	■						
Pruebas de comunicación Modbus												■	■						
Pruebas funcionales del sistema												■	■						
Ajustes y optimización													■	■	■				
Redacción y corrección del informe final																■	■		
Entrega final de la tesis escrita																			■

IX CONCLUSIONES

El desarrollo del prototipo permitió validar el diseño mecánico y eléctrico de la banda transportadora, asegurando su correcto funcionamiento en la clasificación de objetos según color, peso y tamaño. Se comprobó la fiabilidad de los sensores utilizados: el sensor TCS3200 para la detección de color, el sensor HC-SR04 ultrasónico para la determinación del tamaño.

La programación del Arduino y del PLC Siemens S7-1500 fue fundamental para el correcto funcionamiento del sistema de clasificación. En el Arduino, se estructuró el código para adquirir y procesar datos de los sensores de manera correcta, asegurando una comunicación estable con el PLC a través de Modbus TCP/IP. En TIA Portal, la lógica de control implementada con el bloque MB_CLIENT permitió una sincronización eficiente entre la detección de datos y la activación de los actuadores.

El protocolo Modbus TCP/IP demostró ser eficaz para la transmisión de datos en sistemas de automatización industrial, permitiendo interoperabilidad entre dispositivos de diferentes fabricantes. El HMI se confirmó como una herramienta esencial para la configuración y supervisión del sistema, aunque se identificaron oportunidades de mejora, como la implementación de funciones avanzadas, el prototipo sirve como base para futuras mejoras en velocidad, escalabilidad y aplicación en otros sistemas de automatización.

Las pruebas experimentales demostraron que el sistema clasifica las cajas de manera correcta y confiable, cumpliendo con los criterios predefinidos de color, peso y tamaño de la receta. Se verificó la estabilidad del proceso, con un margen del 68% en la detección y ejecución de las acciones de clasificación en las primeras nueve muestras, es así, que se realizó ajustes en el prototipo, con esto se evidencia un incremento en la efectividad del sistema con las últimas 16 pruebas, observando un margen de estabilidad del 81%.

X RECOMENDACIONES

Para mejora de la estructura se debe observar la estabilidad y durabilidad del sistema, se recomienda optimizar el diseño mecánico del prototipo. Esto incluye el uso de diferentes mecanismos que ayuden a aumentar el torque y control en la velocidad. Estas mejoras son clave para garantizar un desempeño confiable en operaciones prolongadas y entornos industriales.

Es importante modularizar el código en Arduino, implementar técnicas como interrupciones y reducir tiempos de muestreo innecesarios ayudará a mejorar la eficiencia y minimizar retardos en la transmisión de datos al PLC.

Para evaluar el desempeño de este sistema en condiciones reales, se recomienda realizar pruebas en entornos donde factores como iluminación variable, vibraciones puedan influir en el funcionamiento de los sensores. Estas pruebas permitirán identificar posibles limitaciones y realizar ajustes antes de su implementación definitiva.

Considerar en el diseño de la estructura la forma de que una celda de carga sea instalada en el prototipo, es por eso que, en este proyecto se procedió a realizar la simulación del peso por la poca factibilidad de ensamblaje en el diseño.

Se sugiere probar el sistema en un entorno real, donde factores como iluminación irregular, vibraciones o polvo puedan afectar su funcionamiento. Además, es crucial calibrar los sensores regularmente y monitorear el desgaste del servomotor para evitar fallos en operaciones continuas. Estas pruebas prácticas ayudarán a identificar y corregir limitaciones antes de su implementación definitiva.

XI BIBLIOGRAFÍA

Arduino. (2020). *Software de Arduino*. Obtenido de Arduino: <https://arduino.cl/programacion/>

Ardumótica. (2023). *MODULO STEP DOWN LM2596 AJUSTABLE CON DISPLAY*. Obtenido de Ardumótica: <https://ardumotica.cl/producto/modulo-step-down-lm2596-ajustable-con-display/>

Aula21. (2024). *Modbus: Qué es y cómo funciona*. Obtenido de Aula21: <https://www.cursosaula21.com/modbus-que-es-y-como-funciona/>

Aula21. (2024). *TIA Portal: ¿Qué es y para qué sirve?* Obtenido de Aula21: <https://www.cursosaula21.com/tia-portal/>

AUMA Iberia. (12 de Julio de 2024). *Implementación de Modbus en sistemas de automatización industrial*. Obtenido de AUMA Iberia: https://aumaiberia.com/modbus-en-sistemas-de-automatizacion-industrial/?utm_source=chatgpt.com

BeltingLab. (29 de Mayo de 2023). *Qué es una cinta transportadora*. Obtenido de BeltingLab: <https://beltinglab.com/noticias/que-es-una-cinta-transportadora-funcionamiento-y-aplicaciones>

Egasen. (13 de Octubre de 2021). *Qué es un servomotor y para qué se utiliza*. Obtenido de Egasen: <https://www.egasen.com/es/blog/noticias/que-es-servomotor-para-que-se-utiliza>

ElectrónicaTNC. (2022). *Modulo Ethernet ENC28J60*. Obtenido de ElectrónicaTNC: <https://electronicatnc.com.co/producto/modulo-ethernet-enc28j60/>

ElectronPerdido. (2024). *Módulo de pesaje electrónico HX711 (ref: 0015)*. Obtenido de ElectronPerdido: <https://electronperdido.com/shop/sensores/fuerza/modulo-de-pesaje-electronico-hx711/>

Flintec. (2024). *¿Cómo funciona una celda de carga?* Obtenido de Flintec: <https://www.flintec.com/co/aprender/sensor-de-peso/celulas-de-carga>

IndHand. (2023). *Qué es Modbus y cómo funciona en la industria.* Obtenido de InHand: <https://www.inhand.com/es/support/blogs/what-is-modbus/>

IPV-Solution. (2020). *Unidad de accionamiento final para cinta transportadora articulada, Versaflex.* Obtenido de IPV-Solution: <https://www.ipv-solution.com/es/anuncios/1985074-unidad-de-accionamiento-final-para-cinta-transportadora-articulada-versaflex-a06de2-mk-como-nueva-50>

LM Group. (7 de Mayo de 2024). *La importancia de los sistemas de clasificación automatizados.* Obtenido de LM Group: https://lmgroup-es.com/la-importancia-de-los-sistemas-de-clasificacion-automatizados/?utm_source=chatgpt.com

Logicbus. (30 de Noviembre de 2024). *Protocolos de comunicación: MODBUS TCP/IP.* Obtenido de Logicbus: <https://www.logicbus.com.mx/blog/modbus-tcp-ip/>

Marcos. (29 de Noviembre de 2020). *Banda transportadora estrecha para aplicaciones de clasificación y desvío de pedidos.* Obtenido de ARIA CORP: <https://nordica.co/es/banda-transportadora-estrecha/>

MicroChipOtle. (2025). *Sensor inductivo: Funcionamiento y Conexión.* Obtenido de MicroChipOtle: <https://microchipotle.com/sensor-inductivo-funcionamiento-y-conexion/>

National Instruments. (8 de Julio de 2024). *Desarrollo de aplicaciones con Modbus.* Obtenido de National Instruments: https://www.ni.com/es/shop/seamlessly-connect-to-third-party-devices-and-supervisory-system/application-development-with-modbus.html?srsId=AfmBOopleOkNHCLcr-J3_cbmMhB5XdgCIKiY7L5FPFEbkdWFfaI6FP--&utm_source=chatgpt.com

Naylamp. (2023). *Tutorial uso de servomotores con arduino*. Obtenido de Naylamp:
https://naylampmechatronics.com/blog/33_tutorial-uso-de-servomotores-con-arduino.html

Néstor Agudelo, G. T. (2020). *Ingeniero Vizcaino*. Obtenido de
<https://ingenierovizcaino.com/ecci/aut1/corte1/articulos/Historia%20de%20la%20Automatizacion.pdf>

Novatronic. (2020). *Sensor de Color TCS3200*. Obtenido de Novatronic:
<https://novatronicec.com/index.php/product/sensor-de-color-tcs3200/>

Novatronic. (2020). *Servomotor S3003 360 grados*. Obtenido de Novatronic :
<https://novatronicec.com/index.php/product/servomotor-s3003-12kg-360-grados/>

Paguayo. (26 de Diciembre de 2022). *Usando el sensor de color TCS230/TCS3200 con Arduino*. Obtenido de MCI educación: <https://cursos.mcielectronics.cl/2022/12/26/interfaz-del-sensor-de-color-tcs230-tcs3200-con-arduino/>

PRECISION. (14 de Septiembre de 2020). *Webinar - Poleas Revesol para fajas transportadoras*. Obtenido de Youtube: https://www.youtube.com/watch?app=desktop&v=afsYwiPEQ_k

Robotics. (2023). *ETHERNET MINI ENC28J60*. Obtenido de Robotics:
<https://roboticsec.com/producto/ethernet-mini-enc28j60/>

RobotsArgentina. (21 de Diciembre de 2018). *Arduino: reconocer colores con el módulo TCS230 – TCS3200*. Obtenido de Robotsargentina: <https://robots-argentina.com.ar/didactica/arduino-reconocer-colores-con-el-modulo-tcs230/>

SDI Industrial. (2022). *Sensor inductivo*. Obtenido de SDI Industrial:
<https://sdindustrial.com.mx/blog/sensores-inductivos/>

- Siemens. (2025). *TIA Portal (Totally Integrated Automation Portal)*. Obtenido de Siemens:
<https://www.siemens.com/ar/es/productos/automatizacion/software-industrial/tia-portal.html>
- Smith, J. (14 de Junio de 2024). *Cómo instalar una cinta transportadora: pautas, mantenimiento y solución de problemas*. Obtenido de GramConveyor:
<https://www.gramconveyor.com/es/how-to-install-conveyor-belt/>
- Solectro. (12 de Octubre de 2021). *¿Cómo funciona el sensor de ultrasonidos medidor de distancia?* . Obtenido de Solectro: https://solectroshop.com/es/blog/como-funciona-el-sensor-de-ultrasonidos-medidor-de-distancia--n99?srsId=AfmBOorPOll2KYnp9Y4pFdTruReZul_LdB95-WPsB70oIezO8Focmrl
- TecnoPLC. (2023). *Tipos de bloques en TIA Portal descripción y funcionamiento*. Obtenido de TecnoPLC: https://www.tecnoplac.com/tipos-de-bloques-en-tia-portal-descripcion-y-funcionamiento/#Bloque_de_organizaci%C3%B3n_OB
- Torres-Medina, Y. (16 de julio de 2020). *El análisis del error humano en la manufactura*. Obtenido de Revista UIS ingeniería: <file:///C:/Users/Antho/Downloads/10827-Textodelarticulo-76699-1-10-20200729.pdf>
- UNIT Electronics. (2025). *LM2596 Regulador Step Down 25W 3A*. Obtenido de UNIT Electronics: <https://uelectronics.com/producto/modulo-regulador-ajustable-lm2596-dc-dc-step-down-3a-1-25-30v/#:~:text=INFORMACI%C3%93N,alimentaci%C3%B3n%20con%20un%20voltaje%20mayor.>

UnitElectronics. (2023). *Celdas de Carga 1/5/10 kg con Modulo HX711*. Obtenido de UnitElectronics: <https://uelectronics.com/producto/celdas-de-carga-1-5-10-kg-con-modulo-hx711/>

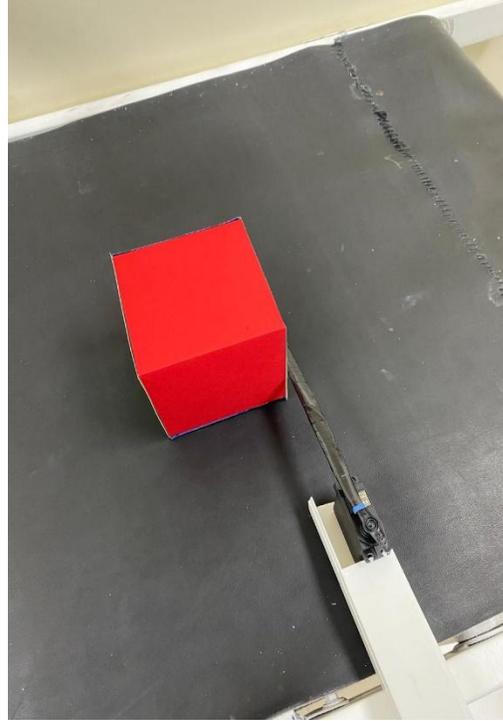
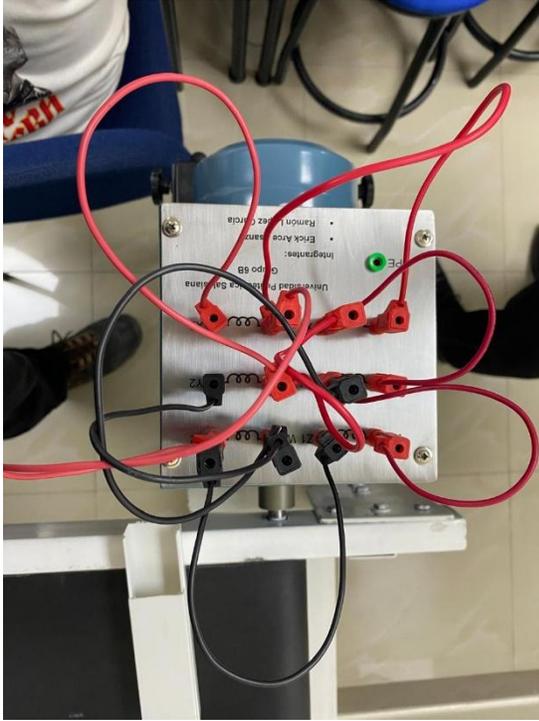
Usinainfo. (2020). *Módulo Ethernet Mini ENC28J60*. Obtenido de Usinainfo: <https://www.usinainfo.com.br/ethernet-e-wifi-arduino/modulo-ethernet-enc28j60-2732.html>

VENMIR. (21 de Febrero de 2022). *Partes esenciales de una banda transportadora: ¿Cuáles son?* Obtenido de VENMIR: <https://venmir.com/partes-esenciales-de-una-de-bandas-de-transportadora/>

WitAutomatización. (2023). *¿Qué es TIA Portal y para qué sirve?* Obtenido de WitAutomatización: <https://witautomatizacion.es/que-es-tia-portal-y-para-que-sirve/>

XII ANEXOS

Anexo 1. Pruebas generales del prototipo



Anexo 2. Programación general Arduino Uno

```
// Código que permite escritura de accionamiento para clasificación
#include <EtherCard.h> //librería de comunicación modulo ENC28J60
#include <ModbusEthercard.h> //librería de Modbus TCP en el modulo ENC28J60
#include <Modbus.h> //librería Modbus
#include <Servo.h> //librería para controlar servos

// Dirección de registro para comunicación
Servo servo1;
const int SERVO1_POS = 55

// Configuración de Modbus TCP
ModbusEthercard mb;
byte mac2[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xEE}; //dirección mac del modulo ENC28J60
byte ip2[] = {172, 18, 135, 78}; //dirección ip de arduino UNO

void setup() {
  Serial.begin(9600); //Inicialización de comunicación a 9600 baudios
  servo1.attach(6); //Pines de conexión
  mb.config(mac2, ip2); // Configuración de la MAC e IP para el modulo
  mb.addHreg(SERVO1_POS, 90); //Registro para almacenar angulo
}

void loop() {
  mb.task(); //Tarea para actualizar registros
  int pos1 = mb.Hreg(SERVO1_POS); //Lectura de registro
  servo1.write(pos1); //Escritura de registros
  Serial.print("Servo1: "); Serial.print(pos1);
  delay(500);
}

// Dirección MAC del módulo Ethernet ENC28J60
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};

// Dirección IP del Arduino en la red
byte ip [] = {172, 18, 135, 77};

void setup() {
  Serial.begin(9600); // Inicializa la comunicación serie a 9600 baudios

  // === Configuración del sensor de color TCS3200 ===
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(out, INPUT); // La salida del sensor es de tipo frecuencia, por lo que se

  // Configuración de escala máxima de frecuencia en el sensor
  digitalWrite(s0, HIGH);
  digitalWrite(s1, HIGH);

  // === Configuración del sensor ultrasónico HC-SR04 ===
  pinMode(trigPin, OUTPUT); // El pin TRIG se configura como salida
  pinMode(echoPin, INPUT); // El pin ECHO se configura como entrada

  #include <EtherCard.h> //librería de comunicación modulo ENC28J60
  #include <ModbusEthercard.h> //librería de Modbus TCP en el modulo ENC28J60
  #include <Modbus.h> //librería Modbus

  // === Pines del sensor de color TCS3200 ===
  const int s0 = 1; // Control de escala de frecuencia S0
  const int s1 = 2; // Control de escala de frecuencia S1
  const int s2 = 3; // Selección de filtro de color S2
  const int s3 = 4; // Selección de filtro de color S3
  const int out = 5; // Salida de frecuencia del sensor de color

  // Variables para almacenar los valores RGB
  int rojo = 0, verde = 0, azul = 0;
  int colorValor = 0; // Variable que almacena el color detectado

  // === Pines del sensor ultrasónico HC-SR04 ===
  const int trigPin = 7; // Pin para activar el pulso ultrasónico
  const int echoPin = 8; // Pin que recibe el eco del sensor ultrasónico
  long duracion; // Variable para almacenar el tiempo de retorno del pulso
  int objetoDetectado = 0; // Variable que indica si hay un objeto (1 = sí, 0 = no)
  // === Direcciones de registros Modbus para enviar datos al PLC ===
  const int SENSOR_COLORVAL = 53; // Registro Modbus para el valor del color detectado
  const int SENSOR_TAMANIO = 54; // Registro Modbus para el tamaño (detección de objeto)

  void loop() {
    // === Leer color del sensor TCS3200 ===
    color();

    // === Leer tamaño del objeto con el sensor ultrasónico ===
    objetoDetectado = medirTamaño();

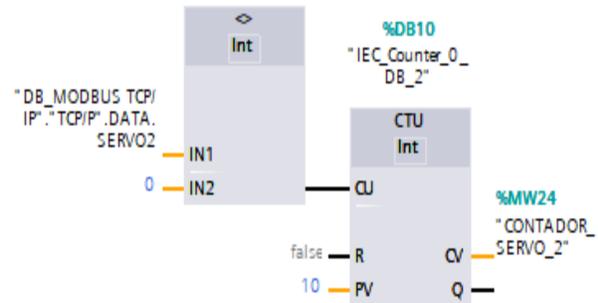
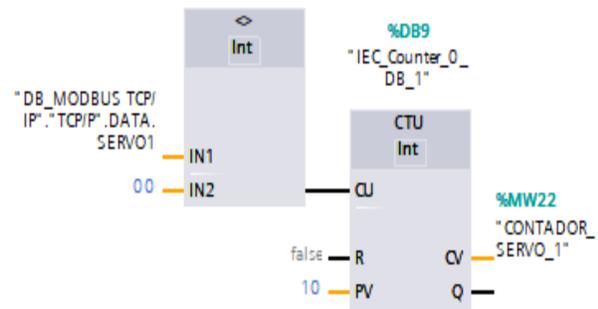
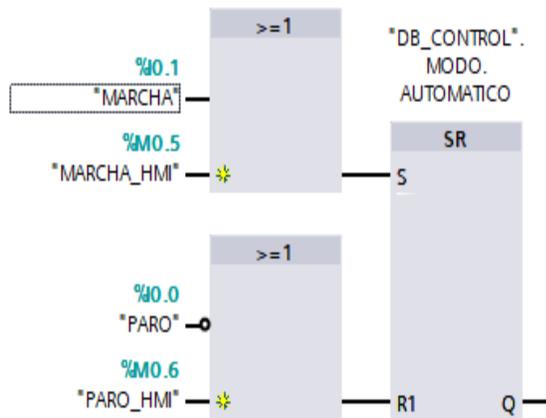
    // === Determinar el color basado en los valores RGB obtenidos ===
    if (rojo < azul && verde > azul && rojo < 35) {
      colorValor = 1; // Se detecta color rojo
    } else if (azul < rojo && azul < verde && verde < rojo) {
      colorValor = 2; // Se detecta color azul
    } else if (rojo > verde && azul > verde) {
      colorValor = 3; // Se detecta color verde
    } else {
      colorValor = 0; // Sin color definido
    }

    // === Enviar valores al PLC mediante Modbus ===
    mb.Ireg(SENSOR_COLORVAL, colorValor); // Envía el valor del color detectado
    mb.Ireg(SENSOR_TAMANIO, objetoDetectado); // Envía el valor de detección de objeto
    mb.task(); // Ejecuta la tarea Modbus para actualizar los registros

    // === Imprimir en el monitor serie para depuración ===
    Serial.print("Color Valor: "); Serial.print(colorValor);
    Serial.print(" Tamaño (1 = Objeto cerca: "); Serial.println(objetoDetectado);
  }
}
```

Anexo 2. Programación general TIA Portal

```
1 CASE "DB_MODBUS TCP/IP"."TCP/P".DATA.COLOR OF
2 // DETECCIÓN PARA PRODUCTOS DE COLOR ROJO
3
4 1:
5 IF ("DB_MODBUS TCP/IP"."TCP/P".DATA.TAMAÑO = "DB_DATOS".DATA_RECETA_1.TAMAÑO_RECETA_1) AND
6 ("DB_MODBUS TCP/IP"."TCP/P".DATA.PESO = "DB_DATOS".DATA_RECETA_1.PESO_RECETA_1) THEN
7 IF ("DB_DATOS".DATA_RECETA_1.COLOR_RECETA_1 = 1) THEN
8 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 180;
9 ELSE
10 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 0;
11 END_IF;
12 IF ("DB_DATOS".DATA_RECETA_2.COLOR_RECETA_2 = 1) THEN
13 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV02 := 180;
14 ELSE
15 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV02 := 0;
16 END_IF;
17 ELSE
18 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 0;
19 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV02 := 0;
20 END_IF;
21 // DETECCIÓN PARA PRODUCTOS DE COLOR AZUL
22 2:
23 IF ("DB_MODBUS TCP/IP"."TCP/P".DATA.TAMAÑO = "DB_DATOS".DATA_RECETA_1.TAMAÑO_RECETA_1) AND
24 ("DB_MODBUS TCP/IP"."TCP/P".DATA.PESO = "DB_DATOS".DATA_RECETA_1.PESO_RECETA_1) THEN
25 IF ("DB_DATOS".DATA_RECETA_1.COLOR_RECETA_1 = 2) THEN
26 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 90;
27
28
29
30
31
32
33
34
35
36
37
38
39
40 // DETECCIÓN PARA PRODUCTOS DE COLOR VERDE
41 3:
42 IF ("DB_MODBUS TCP/IP"."TCP/P".DATA.TAMAÑO = "DB_DATOS".DATA_RECETA_1.TAMAÑO_RECETA_1) AND
43 ("DB_MODBUS TCP/IP"."TCP/P".DATA.PESO = "DB_DATOS".DATA_RECETA_1.PESO_RECETA_1) THEN
44 IF ("DB_DATOS".DATA_RECETA_1.COLOR_RECETA_1 = 3) THEN
45 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 45;
46 ELSE
47 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 0;
48 END_IF;
49 IF ("DB_DATOS".DATA_RECETA_2.COLOR_RECETA_2 = 3) THEN
50 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV02 := 45;
51 ELSE
52 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV02 := 0;
53 END_IF;
54 ELSE
55 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 0;
56 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV02 := 0;
57 END_IF;
58 ELSE
59 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV01 := 0;
60 "DB_MODBUS TCP/IP"."TCP/P".DATA.SERV02 := 0;
61 END_CASE;
62
63
```



DB_DATOS

Name	Data type	Start value
Static		
DATA_RECETA_1	Struct	
PESO_RECETA_1	Real	0.0
COLOR_RECETA_1	Int	0
TAMAÑO_RECETA_1	Int	0
TIPO DE PRODUCT...	String	"
DATA_RECETA_2	Struct	
PESO_RECETA_2	Real	0.0
COLOR_RECETA_2	Int	0
TAMAÑO_RECETA_2	Int	0
TIPO DE PRODUCT...	String	"

DB_MODBUS TCP/IP

Name	Data type	Start value
Static		
ESTADO	Int	0
TCP/IP	Struct	
DATA	Struct	
PESO	Real	0.0
COLOR	Int	0
TAMAÑO	Int	0
PRESENCIA	Int	0
SERVO1	Int	0
SERVO2	Int	0
Comunication 1	TCON_IP_v4	
InterfaceId	HW_ANY	64
ID	CONN_OUC	16#1
ConnectionType	Byte	16#0B
ActiveEstablish...	Bool	TRUE
RemoteAddress	IP_V4	
RemotePort	UInt	502
LocalPort	UInt	0
Comunication 2	TCON_IP_v4	
Comunication 3	TCON_IP_v4	
Comunication 4	TCON_IP_v4	
Comunication 5	TCON_IP_v4	