



**“UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA”
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN**

TRABAJO DE INTEGRACIÓN CURRICULAR

previo a la obtención del título de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

**“IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO, MEDIANTE EL USO
DE VISIÓN ARTIFICIAL PARA LA CLASIFICACIÓN DEL MARACUYÁ, SEGÚN
SU COLOR DE MADUREZ Y EL USO DE UN SISTEMA SCADA PARA EL
MONITOREO DE LA PRODUCTIVIDAD”**

AUTOR:

NICK BRYAN BRANQUE ROSALES

DOCENTE TUTOR:

ING. CARLOS SALDAÑA ENDERICA, MGT.

LA LIBERTAD – ECUADOR

2022

AGRADECIMIENTO

A **Dios** por darme la sabiduría y fortaleza, para poder culminar con éxito mis estudios y mi proyecto de grado.

A la universidad estatal península de Santa Elena, por permitirme adquirir todos los conocimientos necesarios para poder culminar con mis estudios; mis más profundos agradecimiento, siempre estaré en deuda con tal prestigiosa universidad.

A mi madre, **Flor Rosales** por inspirarme desde muy pequeño, por forjarme en la persona que hoy me he convertido, y demostrarme que todo con disciplina y coraje se puede lograr, por tu infinito amor, te agradezco madre mía.

A mi amada esposa, **Lorena Herrera** y a mi hijo **Nick Branque**, por ser mi principal fuente de motivación, y siempre estar en mis pensamientos, para nunca rendirme, por todo el apoyo que me han brindado, por todo el sacrificio que hacen para que yo pueda culminar con mis estudios y convertirme en una mejor persona, por todo eso y más, me faltara una vida para agradecerles.

A mi suegra, **Lorena Macias**, por brindarme todo su apoyo incondicional, por abrirme sus brazos cuando todo se puso difícil, por ser esa mujer luchadora y trabajadora que me hace tenerla de norte, gracias suegra querida por siempre creer en mí.

Al **Ing. Carlos Saldaña Enderica**, por ser mi tutor del proyecto, por brindarme todo su apoyo, y sus conocimientos, por su valioso tiempo y su infinita paciencia, gracias por ayudarme a culminar de manera exitosa mi proyecto de grado.

¡Mis más sinceros agradecimientos ¡

Nick Bryan Branque Rosales

DEDICATORIA

El presente trabajo de titulación se lo dedico a mi padre. Sé que hoy no estas, pero todas tus enseñanzas se quedaron en mi corazón y en mi memoria, forjando así a este ser humano, que hoy logra culminar sus estudios universitarios. Sé que allá en el cielo, estas muy feliz y orgulloso por tu hijo, esto es para ti papá

Hijo de mi vida, ahora eres muy pequeño para entender mis palabras, pero para cuando crezcas y seas capaz, quiero que entiendas lo que realmente significas para mí, eres la razón por la cual, pase muchos días sin dormir para poder cumplir con mis obligaciones de la universidad; quiero que sepas que eres mi principal fuente motivación. Hijo de mi corazón todo esto es por ti

Y finalmente a mis queridos hermanos; quiero decirles que los sueños se cumplen, cuando ponemos el corazón al frente y no cesamos ante nada ni nadie. Este trabajo es para ustedes hermanos de mi corazón

Nick Bryan Branque Rosales

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación denominado: **“Implementación de un sistema automatizado, mediante el uso de visión artificial para la clasificación del maracuyá, según su color de madurez y el uso de un sistema SCADA para el monitoreo de la productividad”**, elaborado por el estudiante Branque Rosales Nick Bryan, de la carrera de Electrónica y Automatización de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus parte y autorizo al estudiante que inicie los trámites legales correspondientes.

La Libertad, 30 de agosto del 2022.



Ing. Carlos Saldaña Enderica Mgt.

TRIBUNAL DE GRADO



Ing. Ronald Rovira Jurado, Ph.D
DIRECTOR DE LA CARRERA DE
ELECTRÓNICA Y AUTOMATIZACIÓN



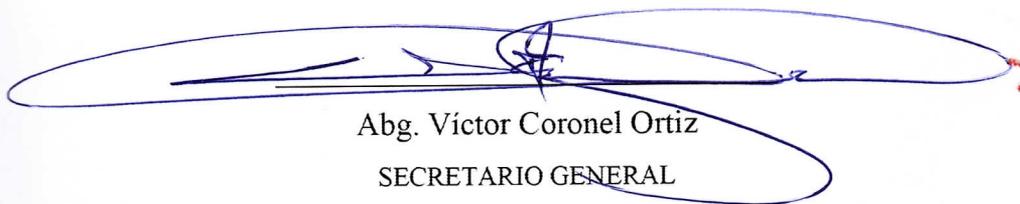
Ing. Carlos Saldaña Enderica Mgt.
DOCENTE TUTOR



Ing. Sendey Vera Gonzalez, Mgt.
DOCENTE ESPECIALISTA



Ing. Alfredo Tumbaco Reyes, Mgt.
DOCENTE GUÍA UIC



Abg. Víctor Coronel Ortiz
SECRETARIO GENERAL

RESUMEN

La automatización de procesos, avanza a pasos agigantados, gracias a los grandes beneficios que estos sistemas brindan, como el aumento de la productividad, la reducción de costos y confiabilidad. El país es uno de los principales exportadores del maracuyá, y las empresas que procesan estas frutas, no cuentan con estos sistemas automatizados de gran eficiencia, es por esto, la necesidad de implementar, sistemas automáticos para las diferentes líneas de producción de esta fruta exótica. El presente proyecto consiste en la implementación de un prototipo automatizado que sea capaz de clasificar el maracuyá en función de su color de maduración, mediante las nuevas técnicas de control, que se utilizan en la industria 4.0, como lo es la visión artificial.

Los instrumentos usados en esta técnica para este proyecto son: una Raspberry Pi 4 , ya que posee la capacidad de cálculo y memoria para el tratamiento de imágenes, también se usa el lenguaje de programación Python, y las librerías de OpenCV, donde se encuentra implementada la técnica HSV, que se la utiliza para la detección de los colores de la fruta, en tiempo real, por último se hace uso de la librería de código abierto Tensorflow, que sirve para el entrenamiento de una red neuronal que es capaz de detectar, si la fruta es un maracuyá o no.

Un controlador lógico programable S7-1200, está a cargo de la regulación, de la velocidad de la banda, por medio, de un variador de frecuencia, y a su vez está encargado de controlar los sensores del sistema de clasificación. El proyecto también consta, con el diseño de un sistema SCADA, para el monitoreo de la productividad, y por ultimo y no menos importante la implementación de un tablero de control, para mejorar la seguridad del sistema, en caso de que haya alguna falla técnica. La implementación del presente proyecto se realizó en el laboratorio de automatización de la “Universidad estatal península de Santa Elena”

PALABRAS CLAVES:

- VISIÓN ARTIFICIAL
- OPENCV
- HSV
- SCADA
- RED NEURONAL
- TENSORFLOW

ABSTRACT

Process automation is advancing by leaps and bounds, thanks to the great benefits that these systems provide, such as increased productivity, cost reduction and reliability. The country is one of the main exporters of passion fruit, and the companies that process these fruits do not have these highly efficient automated systems, which is why there is a need to implement automatic systems for the different production lines of this exotic fruit. This project consists of the implementation of an automated prototype capable of classifying passion fruit according to its ripening color, using new control techniques used in industry 4.0, such as artificial vision.

The instruments used in this technique for this project are: a Raspberry Pi 4, since it has the computing power and memory for image processing, also used the Python programming language, and OpenCV libraries, where the HSV technique is implemented, which is used for the detection of the colors of the fruit, in real time, finally using the open source library Tensorflow, which is used for training a neural network that is able to detect whether the fruit is a passion fruit or not.

A programmable logic controller S7-1200 is in charge of regulating the speed of the belt by means of a frequency variator, and in turn is in charge of controlling the sensors of the sorting system. The project also includes the design of a SCADA system to monitor productivity and, last but not least, the implementation of a control panel to improve the safety of the system in case of any technical failure. The implementation of this project was carried out in the automation laboratory of the "Santa Elena Peninsula State University".

KEYWORDS:

- ARTIFICIAL VISION
- OPENCV
- HSV
- SCADA
- NEURONAL NETWORK
- TENSORFLOW

INDICE GENERAL

AGRADECIMIENTO	I
DEDICATORIA	II
APROBACIÓN DEL TUTOR	III
TRIBUNAL DE GRADO	IV
RESUMEN	V
ABSTRACT	VI
INDICE GENERAL	VII
INTRODUCCIÓN	1
CAPÍTULO I	2
1.1. ANTECEDENTES	2
1.2. DESCRIPCIÓN DEL PROYECTO	2
1.3. OBJETIVOS	3
1.3.1. Objetivo general	3
1.3.2. Objetivos específicos	3
1.4. JUSTIFICACIÓN	4
1.5. METODOLOGÍA DE LA INVESTIGACIÓN	5
1.5.1. Investigación Exploratoria	5
1.5.2. Investigación Descriptiva	6
CAPÍTULO II	7
2.1. MARCO CONTEXTUAL	7
2.2. MARCO CONCEPTUAL	8
2.2.1. El maracuyá	8
2.2.2. Norma técnica ecuatoriana NTE INEN 1910:2012	8
2.2.2.1. Clasificación del maracuyá por estado de maduración	8
2.2.3. Sistemas automatizados	9
2.2.4. Controladores lógicos programables	10
2.2.5. Lenguaje de escalera	10
2.2.6. Sistema de transporte	10
2.2.6.1. Cinta transportadora	11
2.2.7. Sistemas electrónicos	12
2.2.8. Sensores de proximidad	12
2.2.9. Actuadores	12

2.2.10. Sistema scada	12
2.2.11. Sistema de visión artificial.....	13
2.2.11.1. Imagen.....	13
2.2.11.2. Etapas de un sistema de visión artificial	14
2.2.12. Red neuronal artificial	15
2.2.13. La iluminación	15
2.3. MARCO TEORICO	16
CAPITULO III	19
3.1. COMPONENTES DE LA PROPUESTA	19
3.1.1. Componentes físicos	19
3.1.1.1. Simatic PLC S7-1200 Siemens	19
3.1.1.2. Raspberry PI 4	20
3.1.1.3. Motor trifásico Siemens	21
3.1.1.4. Motor paso a paso Nema 23	22
3.1.1.5. Driver TB6560	23
3.1.1.6. Fuente conmutada WX-DC2412.....	24
3.1.1.7. Sinamics G120 Siemens.....	25
3.1.1.8. Cámara web.....	27
3.1.1.9. Sensor fotoeléctrico.....	28
3.1.1.10. Conductor eléctrico	29
3.1.1.12. Tablero de control eléctrico.....	29
3.1.1.14. Banda transportadora y sistema de clasificación.....	30
3.1.2. Componentes lógicos	30
3.1.2.1. Raspberry Pi Imager.....	31
3.1.2.2. Advanced IP Scanner	31
3.1.2.3. VNC Viewer.....	32
3.1.2.4. Python 3.9.2.....	32
3.1.2.5. Open CV.....	33
3.1.2.6. Tensorflow	33
3.1.2.7. TIA Portal V15.....	34
3.2. DISEÑO DE LA PROPUESTA	35
3.2.1. Diseño de la topología de comunicación	35
3.2.2. Diseño del sistema eléctrico.....	36
3.2.3. Instalación del sistema operativo a la Raspberry pi 4.....	39

3.2.4. Instalación de librerías y framework para visión e inteligencia artificial en la Raspberry	41
3.2.4.1. Instalación de OpenCV	41
3.2.4.2. Instalación de TensorFlow Lite.....	43
3.2.5. Desarrollo del sistema de inteligencia artificial.....	46
3.2.6. Desarrollo del sistema de visión artificial con Opencv	51
3.2.7. Configuración y programación del Sinamics G120 en el software TIA portal	55
3.2.8. Programación de los sensores y el sistema de conteo en TIA Portal.....	58
3.2.9. Programación del mecanismo de clasificación en la Raspberry	59
3.2.10. Comunicación entre la Raspberry Pi 4 y el PLC s7-1200	63
3.2.11. Diseño y configuración del sistema SCADA en la herramienta Ignition	69
3.2.11.1. Configuración de la herramienta SCADA Ignition.....	70
3.2.12. Costos de implementación de la propuesta.....	80
3.4. PRUEBAS Y RESULTADOS	81
3.4.1. Pruebas y resultados del sistema de detección de la visión artificial.....	81
3.4.2. Pruebas y resultados en la red de comunicación.....	87
3.4.3. Pruebas y resultados del sistema de clasificación del maracuyá.	89
3.4.4. Pruebas y resultados de la interfaz del sistema SCADA.	91
CONCLUSIONES.....	95
RECOMENDACIONES	97
BIBLIOGRAFÍA	98

INDICE DE FIGURAS

Figura 1. Producción de fruta Santa Elena [4].	4
Figura 2. El maracuyá.....	8
Figura 3. Partes de una cinta transportadora [15].	11
Figura 4. Imagen bidimensional procesada para obtener el brillo en función de la superficie [19].	13
Figura 5. Diagrama de bloques de las etapas típicas de un sistema de visión artificial.	15
Figura 5. Simatic S7-1200.....	20
Figura 6. Raspberry PI 4.....	21
Figura 7. Motor trifásico siemens.....	22
Figura 8. Motor paso a paso Nema 23.....	23
Figura 9. Motor paso a paso Nema 23.....	24
Figura 10. Motor paso a paso Nema 23.....	25
Figura 11. VDF Sinamics G120 Siemens	26
Figura 12. Módulo de potencia del Sinamics G120 Siemens.....	26
Figura 13. Unidad de control del Sinamics G120 Siemens.....	27
Figura 14. Cámara web.....	28
Figura 15. Sensor fotoeléctrico E18-D50NK	28
Figura 16. Rollo de conductor eléctrico	29
Figura 17. Tablero de control eléctrico	30
Figura 18. Banda transportadora y sistema de clasificación	30
Figura 19. Software Raspberry Pi Imager v1.7.2	31
Figura 20. Software Advanced IP Scanner.....	32
Figura 21. Software VNC Viewer.	32
Figura 22. Lenguaje de programación Python.	32
Figura 23. Librería de código abierto OpenCV.....	33
Figura 24. Librería de código abierto Tensorflow.....	34

Figura 25. Software TIA Portal.....	34
Figura 26. Topología de comunicaciones del sistema de clasificación.....	36
Figura 27. Circuito de fuerza del sistema eléctrico.....	37
Figura 28. Circuito de control del sistema eléctrico.....	38
Figura 29. Tablero de control con sus botoneras de mando.....	38
Figura 30. Conexiones del tablero de control eléctrico.....	39
Figura 31. Selección del sistema operativo desde la herramienta Raspberry Pi Imager.	39
Figura 32. Selección de la unidad SD, para la instalación del sistema Raspberry Pi OS 32 bit.....	40
Figura 33. Instalación del sistema operativo Raspberry Pi OS 32 bit en la tarjeta micro SD.....	40
Figura 34. Instalación del paquete completo de la biblioteca OpenCV.....	43
Figura 35. Verificación de la versión de openCV instalada en la tarjeta Raspberry Pi 4	43
Figura 36. Extracción y verificación de la carpeta que contiene el framework TensorFlow Lite en la tarjeta Raspberry Pi 4.....	45
Figura 37. Descarga de paquetes de instalación que contiene el archivo tensorflow- lite.sh.....	45
Figura 38. Instalación del framework tensorflow-lite mediante la ejecución del archivo setup.sh.....	46
Figura 39. Etiquetado de datos con el color y forma de la fruta para el entrenamiento del sistema de visión e inteligencia artificial.....	47
Figura 40. Archivo XML con las características relevantes a detectar.....	48
Figura 41. Instalación de paquetes necesarios para la construcción del modelo de inteligencia artificial.....	48
Figura 42. Librerías a utilizar importadas para la construcción del modelo de inteligencia artificial.....	48

Figura 43. Carga del directorio que contiene las muestras de imágenes previamente etiquetadas al sitio Google Drive.....	49
Figura 44. Entrenamiento y validación del modelo de inteligencia artificial.	49
Figura 45. Entrenamiento del modelo de detección de calidad de maracuyá.	51
Figura 46. Instrucción para exportar el respectivo modelo de detección de calidad del maracuyá.....	51
Figura 47. Codificación en Python para importar Librerías, framework y lectura de la entrada de video.....	52
Figura 48. Codificación en Python para inicializar el modelo de detección de objetos creado en TensorFlow Lite.....	53
Figura 49. Codificación para inicializar la captura de fotogramas provenientes de la cámara web USB.	54
Figura 50. Codificación en Python para realizar el procesado de imagen y detección de calidad del maracuyá.	54
Figura 51. Codificación para visualizar los FPS y la imagen resultante del sistema de detección de calidad.....	55
Figura 52. Asignación de dirección IP para el PLC S7-1200.....	56
Figura 53. Asignación de dirección IP variador de frecuencia Sinamics G120.....	56
Figura 54. Comunicación Profinet entre el PLC s7-1200 y el Sinamics G120.....	56
Figura 55. Parametrización del motor a controlar desde el TIA Portal.....	57
Figura 56. Selección del bloque SINA_SPEED.....	57
Figura 57. Programación y funcionamiento de bloque SINA_SPEED.....	58
Figura 58. Variables utilizadas para el sistema de clasificación y conteo.....	58
Figura 59. Programación escalera para el sistema de conteo de maracuyás pintonas...	59
Figura 60. Programación escalera para el sistema de conteo de maracuyás maduras...	59
Figura 61. Programación escalera para el sistema de conteo de maracuyás maduras...	59
Figura 62. Diagrama de conexiones Motores paso a paso y controlador TB6650.....	61
Figura 63. Diagrama de flujo del sistema de clasificación.....	62

Figura 64. Programación en Python de los pines GPIO de la tarjeta Raspberry Pi 4 a utilizar.....	62
Figura 65. Programación en Python del modo de trabajo e inicialización de los pines GPIO.....	62
Figura 66. Programación en Python para el accionamiento del motor paso a paso	63
Figura 67. Esquema de comunicación entre PLC S71200 y Raspberry Pi 4	64
Figura 68. Configuración de las señales de los sensores fotoeléctricos en un bloque de datos.....	65
Figura 69. Asignación de direcciones a las señales a enviar mediante el protocolo S7	65
Figura 70. Habilitación de la plataforma Node-Red en la tarjeta Raspberry Pi 4.....	66
Figura 72. Configuración de variables en Node-Red provenientes del PLC S7-1200..	67
Figura 73. Configuración del Pin GPIO para habilitar el controlador TB6650.	68
Figura 74. Programación de la conexión para las señales provenientes del PLC y emisión de dicho valor a un pin GPIO.	69
Figura 75. Arquitectura de comunicaciones SCADA-PLC Siemens S7-1200.....	70
Figura 76. Vista General de la herramienta Ignition.	70
Figura 77. Selección de la conexión con el controlador lógico programable S7-1200.	71
Figura 78. Configuración del canal de comunicación SCADA – PLC S7-1200.	71
Figura 79. Vista General Herramienta de diseño Ignition Designer Launcher.	72
Figura 80. Configuración de tags en la herramienta Ignition Designer Launcher.	73
Figura 81. Habilitación de lectura y escritura en el sistema SCADA.	75
Figura 82. Configuración de señal para guardar datos históricos	76
Figura 83. Base de datos creada en el web server de la herramienta Ignition.....	76
Figura 84. Interfaz de acceso al usuario al sistema SCADA.....	77
Figura 85. Interfaz del Sistema SCADA para el monitoreo de la productividad de maracuyá.....	78
Figura 86. Interfaz de visualización de datos históricos.....	79
Figura 87. Panel de Alarmas en el sistema SCADA	79

Figura 88. Alarmas históricas almacenadas en el sistema SCADA.....	80
Figura 89. Correcta detección de maracuyá madura para el primer modelo.....	82
Figura 90. Incorrecta detección de maracuyá madura para el primer modelo	82
Figura 91. Incorrecta detección de maracuyá madura para el primer modelo	83
Figura 92. Correcta detección de maracuyá madura para el segundo modelo	84
Figura 93. Correcta detección de maracuyá dañada para el segundo modelo.....	84
Figura 94. Incorrecta detección de maracuyá madura para el primer modelo	84
Figura 95. Correcta detección de maracuyá pintona para el tercer modelo	85
Figura 96. Correcta detección de maracuyá madura para el tercer modelo	86
Figura 97. Correcta detección de maracuyá dañada para el tercer modelo	86
Figura 98. Correcta detección de maracuyá dañada y madura a la misma vez para el tercer modelo	86
Figura 99. Eficiencia de los tres modelos de pruebas para la visión artificial	87
Figura 100. Ping 192.168.0.4 -t perteneciente al PLC S7-1200.....	88
Figura 101. Ping 192.168.0.10 -t perteneciente al Sinamics G120	88
Figura 102. Ping 192.168.0.10 -t perteneciente a la PC – SCADA	89
Figura 103. Ping 192.168.0.17 -t perteneciente a la Raspberry Pi 4	89
Figura 104. clasificación del maracuyá pintón.....	90
Figura 105. clasificación del maracuyá maduro.....	90
Figura 106. clasificación del maracuyá dañado	91
Figura 107. Pruebas del sistema SCADA en su estado inicial.....	91
Figura 108. Pruebas del sistema SCADA para el accionamiento del motor	92
Figura 109. Pruebas del sistema SCADA para los indicadores del sensor para el maracuyá pintona.....	93
Figura 110. Pruebas del sistema SCADA para los indicadores del sensor para el maracuyá madura.....	93
Figura 111. Pruebas del sistema SCADA para los indicadores del sensor para el maracuyá dañada.	94

ÍNDICE DE TABLAS

Tabla 1. Clasificación por estado de maduración.....	9
Tabla 2. Etapas de un sistema de visión artificial [19].....	14
Tabla 3. Tipos de sistemas de iluminación [22].....	16
Tabla 4. Características del PLC s7-1200 [26].....	20
Tabla 5. Características técnicas del Raspberry Pi 4 [27].	21
Tabla 6. Características técnicas del Motor trifásico siemens [29].....	22
Tabla 7. Características técnicas del Motor paso a paso Nema 23.....	23
Tabla 8. Características técnicas del Drivers BT6560 [31].....	24
Tabla 9. Características técnicas de la fuente conmutada WX-DC2412.....	25
Tabla 10. Características técnicas del módulo de potencia del Sinamics G120 Siemens.	26
Tabla 11. Características técnicas de la unidad de control del Sinamics G120 Siemens.	27
Tabla 12. Características técnicas de la cámara web.....	28
Tabla 13. Características técnicas del sensor fotoeléctrico.	28
Tabla 14. Direcciones IP utilizadas para los diferentes equipos de la propuesta.	35
Tabla 15. Secuencia de comando para la instalación de OpenCV	41
Tabla 16. Descripción de comandos para instalar las dependencias que necesita la biblioteca OpenCV	42
Tabla 17. Arquitectura de modelo de detección de objetos.....	50
Tabla 18. Objetos del framework TensorFlow Lite para inicializar el modelo de IA....	53
Tabla 19. Pines GPIO a utilizar para el control del sistema de clasificación.	60
Tabla 20. Equivalencia de direcciones de las señales en Node-Red y Step 7.	67
Tabla 21. Sintaxis de los respectivos tags en el sistema SCADA Ignition.	73
Tabla 22. Señales en el sistema SCADA.....	74
Tabla 23. Valores para la implementación de la propuesta.	80
Tabla 24. Pruebas para el modelo de 20 muestra de la visión artificial	81
Tabla 25. Pruebas para el modelo de 100 muestra de la visión artificial.	83
Tabla 26. Pruebas para el modelo de 200 muestra de la visión artificial.	84

INTRODUCCIÓN

La gran necesidad de establecer cambios que aporten mejoras en la economía y la productividad en las industrias, hace necesario la implementación de nuevas tecnologías en el área de la Electrónica y Automatización. En la actualidad, varios procesos de clasificación de frutas se realizan con máquinas diseñadas únicamente para estas tareas. Añadiendo el uso de la mano de obra para estas actividades, han llevado a generar oportunidades para integrar soluciones en sistemas basados en visión artificial. En la actualidad, esta tecnología se la utiliza en la industria alimentaria es en los procesos de control de calidad donde se establece una clasificación de frutas por color, características, peso, etc. En la industria alimentaria, específicamente en el área de exportación de frutas, existen requerimientos de productos con buena calidad, es por esto que este sector está pasando de ser dependiente de la mano de obra a un tipo de industria automatizada.

El proyecto consiste en la implementación de un sistema automatizado, capaz de detectar el estado de maduración del maracuyá, mediante la visión artificial, programada en una Raspberry Pi 4, se automatiza un sistema de transporte capaz de trasladar el maracuyá de un sistema de visión hacia el de clasificación, mediante un *Programmable Logic Controller* (PLC) y un *Variable Frequency Drive* (VFD), así mismo un sistema de clasificación del maracuyá con motores paso a paso, y un sistema *Supervisory Control And Data Acquisition* (SCADA) para el monitoreo de la productividad .

La propuesta tecnológica se encuentra conformada por cuatro capítulos; el primero capítulo comprende los antecedentes, descripción del proyecto y sus objetivos, justificación y la metodología de la investigación usada en el proyecto. En el segundo capítulo el lector podrá encontrar los el marco conceptual y contextual, así como los diferentes trabajos de titulación que sirvieron de referencia para poder culminar con el proyecto. El tercer capítulo contiene todos los componentes que se usaron en el proyecto tanto lógicos como físicos, y la explicación del desarrollo de cada una de las etapas de la propuesta, y a su vez las pruebas y resultados que se obtuvieron al culminar la implementación. Finalmente, en el capítulo cuatro se encuentran las conclusiones, recomendaciones, bibliografías y los anexos.

CAPÍTULO I

1.1. ANTECEDENTES

El campo de la agricultura a nivel mundial es muy extenso, el cual, producto de su naturaleza demanda mucha atención a los detalles con el fin de brindar a los diferentes usuarios productos de calidad aptos para el consumo humano.

Según la Organización de las Naciones Unidas para la Alimentación y la Agricultura, Ecuador es uno de los mayores productores del maracuyá, y el mayor exportador del concentrado de esta fruta exótica, seguido de Vietnam, Perú y Brasil [1]. Por esta razón, la implementación de un sistema automatizado de clasificación de frutas es de gran interés para las empresas exportadoras del país debido a sus beneficios tales como la reducción de costos, aumento de la productividad, disponibilidad, confiabilidad y rendimiento.

Las revistas digitales científicas promueven la elaboración de proyectos de agricultura de precisión basados en el uso de visión artificial. “La plataforma busca recolectar datos biofísicos del cultivo a lo largo de la evolución de sus estados fenológicos, sumado al monitoreo y evaluación de los mismos. Para obtener dichos datos la plataforma debe censar el cultivo mediante la captura de imágenes” [2].

La visión artificial permite obtener métodos de respuesta en tiempo real [3], lo que contribuye a que la información obtenida sea verídica en un porcentaje mayor, disminuyendo errores al momento de procesar productos agrícolas, como el maracuyá, que se esté clasificando, para ventas a nivel internacional.

1.2. DESCRIPCIÓN DEL PROYECTO

El presente proyecto tiene como tema, Implementación de un sistema automatizado, mediante el uso de visión artificial para la clasificación del maracuyá, según su color de maduración y el uso de un sistema scada para el monitoreo de la productividad, busca brindar a los agricultores, en especial aquellos que se dedican a la exportación de productos, un sistema automático que les permite agilizar los procesos de clasificación

del maracuyá cosechado. Este proyecto constará de tres etapas: se partirá con un sistema de visión artificial que, mediante el uso de una cámara de alta resolución, catará la imagen de la fruta, la cual será procesada en un computador junto con herramientas de software libre. Luego para la segunda etapa se constará con un sistema de clasificación de la fruta, según el color de maduración ubicándolas en distintos lotes, mediante sensores y actuadores, y finalmente el sistema SCADA contabilizará las frutas clasificadas en lotes, mediante la descripción de porcentajes, ayudando al empresario a mantener un control de la productividad.

1.3. OBJETIVOS

1.3.1. Objetivo general

Diseñar e implementar un prototipo, de un sistema automatizado capaz de detectar y clasificar el estado de maduración del maracuyá según su color, mediante el uso de visión artificial, y un sistema scada para el monitoreo de la productividad.

1.3.2. Objetivos específicos

Obtener los recursos informativos, para cumplir con las normativas de los diferentes tipos de color de maduración de la fruta, el diseño y desarrollo del prototipo mediante la investigación científica y la aplicación de metodología de la investigación.

Programar un sistema de visión artificial, que detecte el estado de madurez del maracuyá según su color, y la elaboración de un sistema que clasifique la fruta en lotes mediante sensores y actuadores.

Diseñar e implementar un sistema scada para el monitoreo de la productividad, y un tablero de control para mejorar la seguridad del sistema automatizado en casos de alguna falla técnica.

1.4. JUSTIFICACIÓN

El presente proyecto de investigación justifica la importancia de implementar la automatización de procesos en las diferentes pequeñas y medianas empresas del país, ya que, esto permite que el empresario ahorre costos y tiempos además que minimiza el número de errores cometidos en procesos manuales, aumentando así la efectividad en los distintos procesos.

Ecuador es uno de los mayores productores y exportadores de frutas exóticas del mundo, por ejemplo, el maracuyá es una de las frutas más producidas y demandadas a nivel mundial, teniendo a la provincia de Santa Elena con una producción de 760 TM anuales aproximadamente. [4]

Cultivos permanentes de mayor producción			
Cultivos permanentes	Superficie plantada (Ha)	Superficie cosechada (Ha)	Producción (TM.) anual
Banano	171	166	2447
Maracuyá	142	92	760
Otros	945	326	812

Figura 1. Producción de fruta Santa Elena [4].

A nivel mundial, Ecuador está en los tres primeros lugares de producción de maracuyá, misma que no solo es producida si no también procesada para la exportación a diferente mercado a nivel internacional. La empresa Ecuantropics S.A. ubicada en la provincia de Santa Elena, produce aproximadamente 50.000 kg de fruta a la semana; a la fecha uno de los procesos que recientemente han automatizado es un sistema de riego en el área de los cultivos, pero para otros tipos de proceso trabajan de manera manual, la falta de tecnificación de la fruta lleva a una reducción de la productividad [5].

La implementación de un prototipo de un sistema inteligente que seleccione la fruta por su estado de maduración, brindará la oportunidad a que el tiempo de estos procesos se reduzca y ayudará a que exista una mayor eficiencia y productividad en negocios comercializadores de frutas tanto a nivel nacional como a nivel internacional.

Elaborar este proyecto permitirá no solo ahorrar tiempo y costos sino también será un gran paso para la automatización y desarrollo del país en esta área del sector terciario que no se le ha brindado la relevancia necesaria hasta el momento.

Los estudiantes de la universidad estatal península de Santa Elena, en la carrera de electrónica y automatización podrán realizar diferentes tipos de prácticas de laboratorio, logrando que los estudiantes se familiaricen con procesos automatizados reales y no solo simulados que se realizan en las diferentes plataformas, ayudando así a que los alumnos salgan con conceptos más sólidos en el campo de la automatización.

1.5. METODOLOGÍA DE LA INVESTIGACIÓN

“El diseño de investigación constituye el plan general del investigador para obtener respuestas a sus interrogantes o comprobar la hipótesis de investigación. El diseño de investigación desglosa las estrategias básicas que el investigador adopta para generar información exacta e interpretable” [6].

El diseño de la investigación le da la apertura al investigador de conocer el terreno en el cual va a desarrollar su tema, todo con la finalidad de tener ideas claras sobre las estrategias que usará, así como los elementos y parámetros pertinentes para la correcta obtención de información.

1.5.1. Investigación Exploratoria

“La investigación exploratoria corresponde al primer acercamiento a un tema específico antes de abordarlo en un trabajo investigativo más profundo. Se trata de un proceso para tener información básica relacionada con el problema de investigación” [7].

Este tipo de investigación ayuda al investigador a obtener datos directamente de los personajes involucrados en ella, en este caso en particular podemos deducir que son los agricultores quienes trabajan con el maracuyá y conocen las características necesarias para considerarlas una fruta excelente, que han proveído de que parámetros deben ser tomados en cuenta para la implementación del proyecto.

1.5.2. Investigación Descriptiva

En contraste, este tipo de investigación ayuda a analizar profundamente o como lo dice su nombre, a describir. Permite registrar nuestras conclusiones en base a lo ya explorado y por ende comprendido además de tomar determinaciones con respecto al rumbo que tendrá el proyecto de investigación.

CAPÍTULO II

2.1. MARCO CONTEXTUAL

Ecuador es un país agrícola por naturaleza, el 8% del PIB (Producto Interno Bruto) lo contribuye exclusivamente este sector. La exportación de productos agropecuarios constituyó para el país 5 mil millones de dólares en el año 2020 [8]. En la provincia de Santa Elena, existen 20 haciendas que realizan cultivos de diferentes tipos, mismos que representaron USD 9,9 millones en utilidades para el año 2017. El tiempo de desarrollo de estas haciendas data de máximo 8 años atrás, por lo que, los procesos aún son muy básicos; el personal era nuevo en conocimientos agrícolas y les tocó empezar de cero. Todos los procesos de cultivo, cosecha, revisión y clasificación se realizan de manera artesanal [9].

La implementación de un sistema automatizado, que clasifica el maracuyá y monitorea la productividad, busca incrementar la producción de las empresas y disminuir los costos, ayudando así al crecimiento y desarrollo de la provincia, mediante la automatización de procesos que conlleven a la industrialización y a la obtención de utilidades que contribuyen al crecimiento económico no solo de Santa Elena y sus ciudadanos, sino también del país

Se realiza un sistema automático de transporte, capaz de trasladar la fruta hacia el sistema de visión artificial, que, mediante la adquisición y procesamiento de imagen, pueda detectar su color de maduración en tiempo real, luego de este procedimiento la banda transportará la fruta al sistema automático de clasificación, que mediante sensores y actuadores serán ubicados en los diferentes lotes. También se implementará un sistema de monitoreo en tiempo real, por medio del diseño de un sistema SCADA.

2.2. MARCO CONCEPTUAL

2.2.1. El maracuyá

El maracuyá (*Passiflora eduli*) o también conocida como la fruta de la pasión, es una fruta tropical con un sabor ácido y a la vez dulce, rica en hidratos de carbono y agua, y buena fuente de provitamina A, vitamina C, y minerales como magnesio, fósforo y potasio. Esta planta trepadora que crece en la zona centro sur de América como República Dominicana, Ecuador, Brasil entre otros [10].



Figura 2. El maracuyá

2.2.2. Norma técnica ecuatoriana NTE INEN 1910:2012

Según [11]: “Esta norma establece los requisitos generales que debe cumplir el maracuyá para ser consumido en estado fresco, incluido aquel destinados al procesamiento industrial.”

El maracuyá puede clasificarse por tamaño, por estado de maduración y color. Es por eso que para el proyecto solo nos enfocaremos en la clasificación de la fruta por el color del estado de maduración

2.2.2.1. Clasificación del maracuyá por estado de maduración

De acuerdo a esta misma norma, el maracuyá se clasifica de la siguiente manera:

Tabla 1. Clasificación por estado de maduración

CLASIFICACIÓN POR ESTADO DE MADURACIÓN		
Estado	Descripción	color
Maracuyá pintón	“Cuando el fruto no ha alcanzado la madurez de cosecha y el color característico de la variedad, no se extiende en toda la superficie del fruto; el mesocarpio es parcialmente verde, duro e intensamente brillante”.	
Madurez de cosecha	“Fruto que ha completado su desarrollo fisiológico dándole una consistencia firme y que conserva las características propias de la variedad, permitiendo su manipulación y transporte”.	

2.2.3. Sistemas automatizados

El proceso de automatización de las empresas es parte fundamental para el crecimiento y transformación de las mismas a nivel global. Con el fin de que puedan competir internacionalmente, las compañías deben buscar transformar sus entornos en modernos y dinámicos, de fácil y rápida adaptación a los requerimientos y cambios que el mercado les exija. También, ayudará a que los procesos disminuyan el margen de error en tareas cotidianas y se dedique mayor tiempo a tareas mucho más importantes y complejas [12].

La automatización de sistemas es el proceso que permite la transferencia de tareas productivas que por lo general solían ser realizadas por capital humano. El fin de los sistemas automatizados en las compañías es mejorar la producción y calidad de las empresas en acciones repetitivas que al realizarse automáticamente reducen costes de mano de obra.

2.2.4. Controladores lógicos programables

Estos equipos son comúnmente utilizados en industrias que buscan dar un salto significativo en la automatización de sus procesos. Estas computadoras industriales utilizadas por la ingeniería para la automatización de procesos, tiene como principal objetivo, dar instrucciones por medio de la programación para que las máquinas desarrollen efectivamente todas sus tareas. Estos se han convertido en una herramienta fundamental para el desarrollo tecnológico de las industrias y todo el entorno social [13].

Los controladores lógicos programables o más conocidos como PLC son dispositivos electrónicos y robustos fabricados fundamentales para la automatización de procesos industrial.

2.2.5. Lenguaje de escalera

“Este lenguaje permite representar gráficamente el circuito de control de un proceso, con ayuda de símbolos de contactos normalmente cerrados (N.C.) y normalmente abiertos (N.A.), relés, temporizadores, contadores, registros de desplazamiento, etc. Cada uno de estos símbolos representa una variable lógica cuyo estado puede ser verdadero o falso” [14].

El lenguaje de escalera o diagrama de escalera, es un tipo de esquema avanzado de programación que permite el registro de estructuras lógicas en un PLC. Como su nombre lo indica, este diagrama imita el diseño de una escalera ya que se compone de suministros de energía en forma de rieles verticales y de líneas horizontales que son simuladas por los circuitos de control.

2.2.6. Sistema de transporte

En la actualidad, los diferentes sistemas de transporte son de los elementos más importantes de la industria alimentaria; gracias al transporte, la movilización de la materia prima puede efectuarse en grandes cantidades y de forma más eficiente porque contribuye a la minimización de tiempo, costos, entre otros.

Existen múltiples tipos de sistemas de transporte como: de cinta, de cadena, de rodillos, entre otros. En el caso del presente proyecto, el sistema de transporte a utilizar será la “Cinta transportadora”

2.2.6.1. Cinta transportadora

La cinta de transporte está compuesta por dos tambores que giran en función del motor, sobre estos tambores descansa la banda continua. La superficie que sostiene todos estos implementos por lo general, se diseña de manera personalizada ya que, la resistencia de la misma varía de acuerdo al peso que se le vaya a colocar.

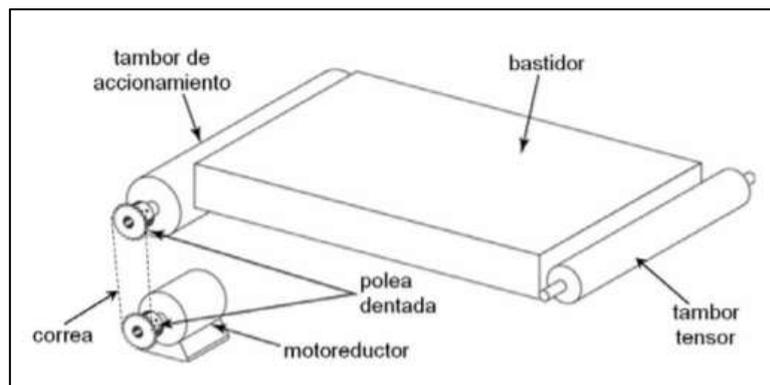


Figura 3. Partes de una cinta transportadora [15].

Los beneficios del uso de estas bandas transportadoras son múltiples, para resumir, podemos establecer las siguientes:

- Debido a la capacidad que tienen para transportar grandes cantidades de productos, ayudan a incrementar la eficiencia y productividad de las empresas.

- Su uso ayuda de manera directa al mejoramiento de las condiciones laborales ya que, el talento humano no tendrá la necesidad de realizar el trabajo de movilización de manera física reduciendo de igual manera los siniestros.

2.2.7. Sistemas electrónicos

Son todos aquellos elementos que ayudan a la generación y distribución de energía a las diferentes interfaces que requieran de su uso, el orden general de estos sistemas es: generación – transportación – transformación – consumo.

2.2.8. Sensores de proximidad

“El sensor de proximidad es un transductor que detecta la presencia de objetos u obstáculos sin la necesidad de contacto (...)” [16]. Estos sensores se usan en la industria para accionar algunos sistemas mecanizados, cuando estos son activados. A continuación, se detallan los más usados:

2.2.9. Actuadores

La función principal de un actuador es transformar, un determinado tipo de energía, en energía mecánica. Para realizar el movimiento o desplazamiento de algún objeto, en un sistema automatizado. Las principales fuentes de energía para lograr el accionamiento son, neumáticos, hidráulicos y eléctricos

“Un actuador lineal eléctrico es un dispositivo que convierte el movimiento rotatorio de un motor en movimiento lineal. Se puede integrar en cualquier equipo para empujar, tirar, levantar, bajar, posicionar o girar una carga” [17]

2.2.10. Sistema scada

Scada viene de acrónimo Supervisory Control And Data Acquisition (Supervisión Control y Adquisición de Datos). Un sistema scada es un conjunto de componentes de

software y hardware, que permite la supervisión y control de plantas industriales, tanto de forma local como remota. El scada también examina, recopila y procesa datos en tiempo real.

“Por medio de una sofisticada interfaz hombre-máquina ejecutada sobre un PC Industrial, es posible monitorizar de forma integral eventos, alarmas, análisis, visualización y control de la información adquirida desde los PLC’s que gobiernan cada celda tanto en corriente alterna como en continua” [18].

2.2.11. Sistema de visión artificial

La visión artificial tiene como único objetivo, la extracción de información del mundo físico mediante imágenes. Un sistema de visión artificial actúa sobre una representación de la realidad que le proporciona información sobre brillo, colores, formas. Estas captaciones suelen estar representadas de la siguiente forma: imágenes estáticas, escenas tridimensionales o imágenes en movimiento [19].

2.2.11.1. Imagen

Una imagen bidimensional se representa en un eje de coordenadas (x, y), que asocia un valor relativo de alguna propiedad del punto que representa su brillo o su matriz como se muestra en la Figura 4 “Una imagen acromática, sin información de color, en la que a cada punto se le asocia información relativa al brillo, se puede representar como una superficie, en la cual la altura de cada punto indica su nivel de brillo” [19].

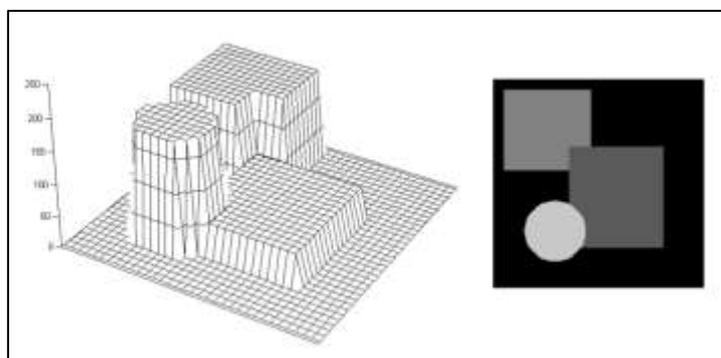


Figura 4. Imagen bidimensional procesada para obtener el brillo en función de la superficie [19].

2.2.11.2. Etapas de un sistema de visión artificial

Se conoce que el ser humano capta la luz a través de los ojos, y que esta información viaja a través de los nervios ópticos hasta llegar al cerebro donde es procesada. Por varias razones, se cree que el primer paso de este proceso, es encontrar elementos más simples que descomponer una imagen, tales como segmentos y arcos. Luego el cerebro interpreta y por último actúa. Con la visión artificial se busca interpretar este proceso, que tradicionalmente costa de cuatro etapas [19].

Tabla 2. Etapas de un sistema de visión artificial [19].

ETAPAS DE UN SISTEMA DE VISIÓN ARTIFICIAL	
Captura o adquisición de imagen	Esta etapa se dedica solo a la parte sensorial, consiste en captar imágenes digitales mediante algún tipo de sensor en su gran mayoría los CCD (Charge Coupled Device), dispositivo de carga acoplada.
Procesamiento	Luego de capturarla, la imagen pasa por un tratamiento digital, donde mediante filtros y transformaciones geométricas, donde se eliminan parte despreciables de la imagen o se realzan partes importantes de la misma.
Segmentación	Esta etapa consiste, en determinar con claridad el aislamiento de los elementos que importan de una escena para poder comprenderla
Reconocimiento o clasificación	Finalmente tenemos la etapa de reconocimiento. En esta se busca distinguir los objetos que han pasado por la etapa de segmentación, gracias al análisis que se establece previamente para identificarlos

Estas etapas, no siempre tienen que ejecutarse de manera secuencial, también se pueden realimentar, hacia una de las etapas anteriores, en caso de que una de ellas llegase a fallar

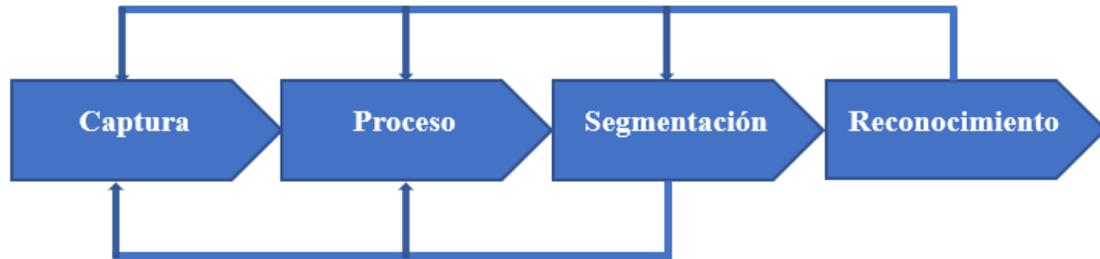


Figura 5. Diagrama de bloques de las etapas típicas de un sistema de visión artificial.

2.2.12. Red neuronal artificial

El objetivo principal de las redes neuronales artificiales es tratar de imitar alguna de las características del humano, como la capacidad de aprender, memorizar y de asociar hechos. “En definitiva, las redes neuronales no son más que un modelo artificial y simplificado del cerebro humano, que es el ejemplo más perfecto del que disponemos para un sistema que es capaz de adquirir conocimiento a través de la experiencia. [20]”

2.2.13. La iluminación

En los sistemas de visión artificial, la iluminación toma un papel muy importante. Esta se puede considerar la parte más crítica dentro de un sistema de visión artificial. Las cámaras son mucho menos sensibles, con respecto a la visión humana, y las condiciones de iluminación deben optimizarse al máximo, para que una cámara pueda captar una imagen, que tal vez el ojo humano podría distinguir con facilidad, sin necesidad de un sistema de iluminación tan especializado. [20]

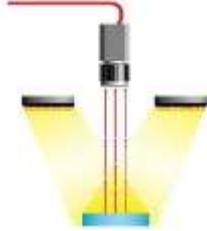
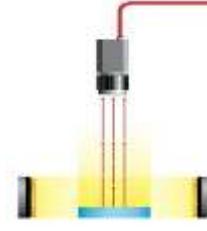
Según [20]: “Esto se hace mucho más evidente cuando el objeto a iluminar presenta formas complejas, o superficies muy reflectantes.”

Tipos y técnicas de sistemas de iluminación

“Las técnicas de iluminación es la manera en las fuentes de luz serán utilizadas, o instaladas. Con esto nos referimos a parámetros como: ángulos de incidencia de luz, distancia, zonas de reflexión y refracción” [21]. Cada sistema de visión artificial necesita

un método específico de iluminación. En la siguiente tabla se muestran las diferentes técnicas de iluminación.

Tabla 3. Tipos de sistemas de iluminación [22].

TIPOS DE SISTEMAS DE ILUMINACIÓN		
Nombre	Descripción	Imagen
Iluminación frontal	Este es uno de los sistemas más utilizados. en este caso la cámara se ubica mirando al objeto y en dirección de la luz. esta técnica es muy útil en superficies con pocos reflejos y se logra mediante anillos de luz.	
Iluminación lateral	esta es muy utilizada cuando se quiere apreciar con mucha nitidez los detalles de los objetos. Estos solo serán visibles en el caso de que la luz esté orientada de forma lateral a la posición de la cámara.	
Iluminación por campo oscuro	En esta técnica, el objetivo es resaltar los defectos superficiales (tales como grietas, surcos o manchas). Para poder detectarlos, se deberán detectar los códigos matrices u otros caracteres impresos en la superficie.	

2.3. MARCO TEORICO

En la siguiente sección se mostrarán las tesis de grados, que fueron analizadas por el autor, que sirvieron de guía y que están relacionadas con la presente propuesta de titulación.

El trabajo de titulación “Diseño e implementación de un sistema de gestión, monitoreo y digitalización para el proceso de selección de mangos en función de su color, enfocado a la industria 4.0 para optimizar el rendimiento de producción”, realizado en la

Universidad estatal península de Santa Elena, en el año 2019, por Alba Ascencio G., destacó la importancia de la innovación en los procesos de clasificación de frutas, mediante el uso de visión artificial, ya que esto, permite la gestión y monitoreo de los procesos en tiempo real, ayudando a la optimización de tiempos de producción. Destacando la importancia de los avances tecnológicos como lo es la realidad aumentada en la industria 4.0 [23].

El mismo año, en la Universidad de las fuerzas armadas, Luis Molina y Miguel Vargas realizaron su trabajo de titulación, “ESTUDIO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE CALIDAD PARA LA DETECCIÓN DE CONTAMINANTES SUPERFICIALES DE DIFERENTES TIPOS DE FRUTAS USANDO VISIÓN ARTIFICIAL”, tiene como objetivo la implementación de sistemas de control que ayudan a analizar los contaminantes en la corteza de las frutas y verduras con mayor exactitud, obteniendo productos de mayor calidad, junto a la optimización de recursos humanos y económicos. Dicha investigación, utiliza una interfaz gráfica que permite al operario tener más control en el proceso de clasificación [24].

En el año 2020, en la Escuela superior politécnica de Chimborazo, Johana Rodríguez y José Salazar mediante su tema de titulación, “DISEÑO, CONSTRUCCIÓN Y AUTOMATIZACIÓN DE UN PROTOTIPO DE MÁQUINA CLASIFICADORA DE PIÑAS DE ACUERDO AL COLOR Y FORMA POR MEDIO DE VISIÓN ARTIFICIAL”, la cual consistía, que por medio de la implementación de un sistema automatizado, y con el uso de visión artificial, para detectar el color de maduración de la piña, obtuvieron datos favorables en la clasificación por medio del prototipo diseñado con respecto a la clasificación manual, con una efectividad bastante favorable, y así mismo una reducción del 13% en los tiempos de clasificación. Finalmente se logró concluir que el prototipo realiza, la clasificación de manera efectiva, sin dañar la piña, también se recomienda utilizar procesadores más avanzados para agilizar los tiempos de ejecución [25].

El trabajo de titulación “Diseño e implementación de un sistema de reconocimiento y manipulación de frutas utilizando visión artificial y brazo robótico industrial”, realizado en la Escuela Superior Politécnica del Litoral, en el año 2016, por Ángel Valencia y Roger Idrovo, implementó un sistema automático de reconocimiento y manipulación de frutas, la propuesta consta de una solución basada en visión artificial para el reconocimiento del objeto, este sistema comprende una cámara industrial encargada de capturar información del entorno y un computador para el procesamiento de las imágenes basado en herramientas de software libre. Seguidamente el proceso de manipulación, esta tarea la realizó un brazo robótico industrial, cuya función es la de posicionar y orientar correctamente su herramienta para poder agarrar la fruta y colocarla de manera ordenada en una caja de empaque [27].

CAPITULO III

3.1. COMPONENTES DE LA PROPUESTA

En la siguiente sección se realizó la selección de cada uno de los componentes físicos y lógicos, que fueron utilizados para la implementación del presente proyecto, detallando las características técnicas más importantes, y una breve explicación de los mismos.

3.1.1. Componentes físicos

Para la implementación del sistema automatizado, que detecta el color de maduración del maracuyá mediante el uso de visión artificial, se utilizaron varios equipos y componentes eléctricos y electrónicos que son detallados a continuación.

3.1.1.1. Simatic PLC S7-1200 Siemens

Es un dispositivo electrónico robusto, con una amplia utilidad en la industria de la automatización. Es una buena opción cuando se necesita realizar tareas de forma flexible, por lo cual, consta de módulos expandibles de entradas y salidas tanto analógicas como digitales, también se le puede añadir módulos de comunicación, con el fin de ampliar las capacidades del PLC [26]. Este equipo es utilizado en el actual proyecto, para controlar el motor de la banda transportadora, el variador de frecuencias, los sensores y actuadores que conforman el sistema de clasificación del maracuyá, y su vez para comunicarse con la tarjeta Raspberry Pi 4 donde estará programado el sistema de visión artificial.



Figura 5. Simatic S7-1200

Tabla 4. Características del PLC s7-1200 [26].

CARACTERÍSTICAS TÉCNICAS	
Modelo	CPU 1212C AC/DC/RLY
Versión de firmware	V4.5
Tensión de alimentación	120 a 230V AC
Alimentación de sensores	24V DC
Entradas digitales	8 entradas; 24V DC
Salidas digitales	6 salidas; relé
Entradas analógicas	2; 0 a 10V
Tipo de interfaz	PROFINET
Dimensionamiento	90mm de ancho, 100mm de altura
Peso, aprox.	425g

3.1.1.2. Raspberry PI 4

Es una minicomputadora muy económica, del tamaño de una credencial, se puede conectar a una pantalla mediante el puerto HDMI, y ser utilizada por medio de un teclado y un mouse ya que posee varios puertos USB. Su sistema operativo Linux le permite ejecutar casi todas las tareas de un ordenador de escritorio, logrando ser usada en una amplia variedad de proyectos, desde páginas web, reproductores de música, robots cinemáticos, casas inteligentes, procesos de automatización a menor escala [27]. Este equipo será utilizado en el actual proyecto, para procesar los colores de los estados de maduración del maracuyá, y posteriormente enviar cuyos datos al PLC mediante comunicación Profinet.



Figura 6. Raspberry Pi 4

Tabla 5. Características técnicas del Raspberry Pi 4 [27].

CARACTERÍSTICAS TÉCNICAS	
Modelo	Raspberry Pi 4 B
CPU	4 núcleos a 1,5GHz con brazo Cortex-A72
GPU	VideoCore VI
Memoria	2GB LPDDR4 RAM
Conectividad	802.11ac Wi-Fi/Bluetooth 5.0, Gigabit Ethernet
Vídeo y sonido	2 puertos micro-HDMI; salida estéreo de 4 polos
Puertos	2 puertos USB 3.0; 2 puertos USB 2.0
Alimentación	5V/3A vía USB-C, 5V vía cabezal GPIO
Expansión	Cabezal GPIO de 40 pines

3.1.1.3. Motor trifásico Siemens

Es una máquina eléctrica que tiene la habilidad de convertir la energía eléctrica en energía mecánica, por medio de sus devanados internos que generan un campo electromagnético rotacional [28]. En la industria tiene una gran variedad de aplicaciones como, bandas transportadoras, sistemas de bombeo, ventiladores, mezcladoras, compresores, extractores y de más. Esta máquina eléctrica estará instalada en la banda transportadora, que junto con el variador de frecuencia le dará la velocidad adecuada a la cinta de transporte para que la fruta pueda ser trasladada tanto al sistema de visión artificial como al sistema de clasificación. En la tabla 7 se encuentran las características técnicas de este equipo.



Figura 7. Motor trifásico siemens

Tabla 6. Características técnicas del Motor trifásico siemens [29].

CARACTERÍSTICAS TÉCNICAS	
Voltaje de alimentación	220 V
Frecuencia	60 Hz
Potencia	0.75 HP
Corriente	2.60 A
Rpm	1735 rpm
Peso neto	14.5 Kg
Factor de potencia	0,77

3.1.1.4. Motor paso a paso Nema 23

Un motor paso a paso es un dispositivo de corriente continua sin escobillas, que su función principal es la de cualquier motor común y corriente, convertir la energía eléctrica en movimiento mecánico. Lo que lo hace diferente de los demás motores es que al recibir un pulso hace que gire un paso, y se necesitan de 200 pasos para que el motor gire 360°, esto quiere decir que cada paso hace que gire 1.8°, gracias a esta cualidad logran obtener un control muy preciso, es por eso, que son utilizados en equipos de gran precisión como las impresoras 3D, maquinas CNC, máquinas para el montaje y en la robótica, desarrollando un papel muy importante en la automatización de procesos y en la manufactura [30]. Este equipo estará implementado en el sistema de clasificación, desviando la fruta, con una paleta que tendrá adaptada en el eje del motor, para desviar la

fruta de la banda transportadora hacia los lotes según corresponda, a continuación, se detallaran las características técnicas



Figura 8. Motor paso a paso Nema 23

Tabla 7. Características técnicas del Motor paso a paso Nema 23

CARACTERÍSTICAS TÉCNICAS	
Modelo	KH56KM2U027G
Voltaje de referencia	1.76 V
Voltaje driver	12 V / 24 V DC
Corriente	2.1 A
Holdng torque	8.5 – 9.5 Kg/cm
pasos	200 pasos por vuelta
Diámetro del eje	6.35mm
Longitud del eje	23mm
Longitud del motor	53 mm

3.1.1.5. Driver TB6560

El driver TB6560 es un dispositivo de carácter electrónico, que fue diseñado para el control de los motores paso a paso. Según los pulsos que se le envíen por medio de la programación de un PLC o incluso un Arduino, este puede satisfacer las necesidades, según la aplicación que se les quiera dar a los motores, son fáciles de usar y usan acoplamientos de alta velocidad para que no existan perdidas en el control de los pasos que el motor suministra. Porta un disipador de calor, para que el módulo no se

sobrecaliente y no pueda afectar con el correcto funcionamiento, se los usa en máquinas de grabado, CNC, máquinas SMT, y en otros equipos [31]. Este módulo será utilizado en el proyecto, para controlar los motores paso a paso, que se encontraran en el sistema de clasificación, dándoles una mayor precisión al momento de clasificar el maracuyá en los diferentes lotes; a continuación, se presentan las características técnicas.

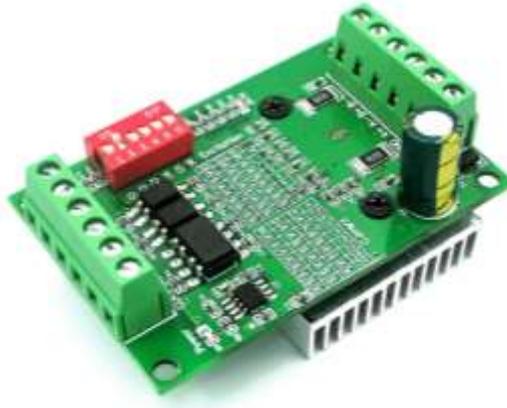


Figura 9. Drivers BT6560

Tabla 8. Características técnicas del Drivers BT6560 [31].

CARACTERÍSTICAS TÉCNICAS	
Tipo de chip	TB6560
Voltaje de alimentación	10 a 35 V DC (24 V Recomendado)
Corriente de funcionamiento	3A
Micro pasos	1, 1/2, 1/8, 1/16
Optoacoplador	6N137
Frecuencia de salida	16 KH z

3.1.1.6. Fuente conmutada WX-DC2412

Este módulo electrónico tiene incorporado un regulador de conmutación, que sirve para convertir la energía de corriente alterna, en cierto valor de energía de corriente continua, gracias a esta cualidad que posee este módulo, es muy utilizado en lugares donde se necesite alimentar varios circuitos al mismo tiempo, con una potencia de hasta 100 W. El módulo es utilizado en el presente proyecto para, la correcta alimentación de los dos drivers TB6560 que van a suministrar la corriente necesaria a los motores paso a paso.



Figura 10. Fuente conmutada WX-DC2412

Tabla 9. Características técnicas de la fuente conmutada WX-DC2412

CARACTERÍSTICAS TÉCNICAS	
Voltaje de alimentación	85 – 245V AC
Voltaje de salida	24V DC
Frecuencia	50Hz/60Hz
Corriente de salida	4 – 6 A
Potencia	100W

3.1.1.7. Sinamics G120 Siemens

Es un variador de frecuencia industrial con tecnología de punta, que está compuesto por una unidad de control y de potencia, tiene una amplia variedad de aplicaciones, las más comunes son los ventiladores industriales, accionamiento para equipos de bombeo, bandas y transportadores industriales, ascensores, tornos y fresadoras [31]. Este equipo inteligente se lo utilizara para controlar la velocidad del motor de la banda transportadora.



Figura 11. VDF Sinamics G120 Siemens

El Sinamics G120 tiene módulos adaptables para su correcto funcionamiento a continuación se mostrarán las características técnicas de cada uno de los módulos.



Figura 12. Módulo de potencia del Sinamics G120 Siemens

Tabla 10. Características técnicas del módulo de potencia del Sinamics G120 Siemens.

CARACTERÍSTICAS TÉCNICAS	
Modelo	PM240-2
Entradas	1/3ØAC 200-240V ±10% 9.6/5.5A 47-63Hz
Salidas	3ØAC 0-INPUT V In=4.2A
Rango de potencia	0.37 – 250 kW
Dimensiones	196x73x1655mm



Figura 13. Unidad de control del Sinamics G120 Siemens

Tabla 11. Características técnicas de la unidad de control del Sinamics G120 Siemens.

CARACTERÍSTICAS TÉCNICAS	
Modelo	CU250S-2 PN
E/S Digitales	6 ED (3ED F) 3 SD (1 SD F) 5 ED
E/S Analógicas	2EA 2SA
Comunicación	RS485, Can, Profibus, Profinet
Dimensiones	196x73x1655mm

3.1.1.8. Cámara web

Este dispositivo permite capturar imágenes nítidas, ya que ofrece una resolución HD de 720 (1280 x 720) a una frecuencia de imagen de 30 fps, así mismo carece de compatibilidad plug-and-play a través de un cable USB 2.0 Tipo-A integrado. La cámara se utiliza para captar la imagen en tiempo real, para que la raspberry pueda realizar en procesamiento de la imagen mediante la programación en Python.



Figura 14. Cámara web

Tabla 12. Características técnicas de la cámara web.

CARACTERÍSTICAS TÉCNICAS	
Ajuste de enfoque	Manual
Tamaño de video predeterminado	1280 * 720P
Frame rate	30 FPS
Sensibilidad mínima	2.0V/Lux
Rango de enfoque:	2 cm a infinito
Interfaz USB	USB 2.0

3.1.1.9. Sensor fotoeléctrico

Es un dispositivo electrónico, que tiene la capacidad de censar cualquier objeto a cortas distancias, gracias a su sistema integrado, de transmisión y recepción de haz de luz; consta con un potenciómetro para poder regular la distancias a sensar. Este dispositivo es de gran utilidad para el sistema de clasificación así mismo como el conteo de la fruta.



Figura 15. Sensor fotoeléctrico E18-D50NK

Tabla 13. Características técnicas del sensor fotoeléctrico.

CARACTERÍSTICAS TÉCNICAS	
Modelo	E18-D50NK
Corriente de carga máxima	300mA (salida pulldown NPN)
Tiempo de respuesta	<2ms
Distancia de detección	3 – 80cm ajustable
Diámetro	196x73x1655mm

3.1.1.10. Conductor eléctrico

Es un elemento que está diseñado para conducir electricidad. Existen varios tipos de materiales con los que están fabricados, pero el más utilizado es el cobre, por su alto grado de conductividad; el conductor dependerá de su calibre para soportar los distintos niveles de corriente, el alambre está recubierto por un material aislante, para que la energía eléctrica no se escape, y pueda ser transmitida de principio a fin [33]. Este componente es muy importante en el proyecto, ya que servirá de transporte para suministrar energía a todos los dispositivos eléctricos.



Figura 16. Rollo de conductor eléctrico

3.1.1.12. Tablero de control eléctrico

Es un gabinete donde se encuentran instalados los equipos de protecciones, control y fuerza de un sistema eléctrico, es utilizado en los procesos más simple, como el paro y marcha de un motor, hasta procesos industriales muy complejos. El tablero del sistema eléctrico del presente proyecto, brinda una mayor seguridad al momento de utilizar los mandos de accionamiento del sistema, y a su vez para proteger al mismo de cualquier falla técnica que se pueda presentar ya que el tablero cuenta con un paro de emergencia (Ver Figura 17).



Figura 17. Tablero de control eléctrico

3.1.1.14. Banda transportadora y sistema de clasificación

Unos de los componentes que se utiliza en el proyecto, es la banda transportadora, ya que, en esta herramienta de transporte, se moviliza las frutas de los diferentes estados de maduración desde el sistema de visión artificial, hacia el sistema de clasificación; el variador de frecuencia es el encargado de controlar la velocidad de la banda mediante la programación del PLC S7-1200. Sus medidas son 91 cm de alto, 102 cm de largo y 21 cm de ancho (Ver Figura 18)



Figura 18. Banda transportadora y sistema de clasificación

3.1.2. Componentes lógicos

En la siguiente sección se mostrarán todos los softwares que se utilizaron en el presente proyecto para la implementación del mismo

3.1.2.1. Raspberry Pi Imager

El software Raspberry PI Imager instala de forma rápida y fácil el sistema operativo de Raspberry Pi y otros sistemas operativos en una tarjeta microSD [32]. Servirá para la descarga e instalación del sistema operativo para la Raspberry Pi 4.



Figura 19. Software Raspberry Pi Imager v1.7.2

3.1.2.2. Advanced IP Scanner

Este programa sirve para el escaneo de las direcciones IP de los dispositivos de red, le da acceso a las carpetas compartidas y a los servidores FTP [33]. Será utilizado para saber el direccionamiento IP de la raspberry Pi 4.



Figura 20. Software Advanced IP Scanner.

3.1.2.3. VNC Viewer

Permite controlar remotamente sus ordenadores desde otros dispositivos, en cualquier lugar del mundo [34]. Este programa será utilizado para acceder de forma remota a la tarjeta Raspberry Pi, mediante direccionamiento IP, y credenciales de acceso tales como usuario y la contraseña que posee la Raspberry.



Figura 21. Software VNC Viewer.

3.1.2.4. Python 3.9.2

Es un lenguaje de programación interpretado, cuyo mayor objetivo es la legibilidad. Este lenguaje es multiparadigma ya que soporta programación orientada a objetos, imperativa y funcional [35]. Al instalar el sistema Operativo Raspberry Pi OS se instala por default Python que en este caso es la versión 3.9.2.



Figura 22. Lenguaje de programación Python.

3.1.2.5. Open CV

Por sus siglas en inglés Open Computer Vision (Visión Artificial Abierta), es una librería de visión artificial de código abierto, enfocada en el tratamiento de imagen, escrita en C y C++, es ejecutada en los sistemas operativos como Linux, Windows, Mac OS X, y tiene interfaces en Matlab, Java y Python [38]. Esta librería es utilizada para programar el sistema de visión artificial, determinando el color del maracuyá para su respectiva clasificación.



Figura 23. Librería de código abierto OpenCV.

3.1.2.6. Tensorflow

Es una librería de código abierto, que sirve para la construcción y el entrenamiento de redes neuronales, está basado en el aprendizaje profundo (Deep learning), y puede ser ejecutado en los lenguajes de programación como Python y C++ [39]. Esta librería, es utilizada en el proyecto, para el entrenamiento de una red neuronal, que detecte si es un maracuyá.



Figura 24. Librería de código abierto Tensorflow.

3.1.2.7. TIA Portal V15

El portal de automatización totalmente integrado, es un software que brinda diversas funciones, que vinculan la automatización y la digitalización de una manera eficaz [39]. También, Permite la configuración y programación de equipos industriales altamente especializados del fabricante Siemens, consta de tres lenguajes de programación, diagrama de funciones, control estructurado y el lenguaje de escalera. En el presente proyecto se utiliza este software para la programación del PLC S7-1200, y el lenguaje a utilizar será el lenguaje de escalera.

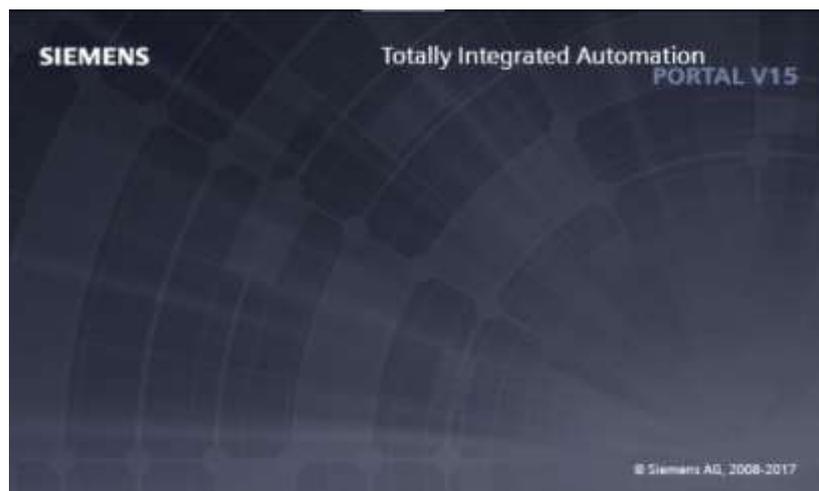


Figura 25. Software TIA Portal.

3.2. DISEÑO DE LA PROPUESTA

3.2.1. Diseño de la topología de comunicación

Para establecer comunicación con los elementos que poseen interfaz ethernet, tales como el PLC S7-1200, Raspberry Pi 4, variador de frecuencia y el computador donde se ejecuta el sistema SCADA, se empleó la red 192.168.0.0 en la máscara de red /24, dicho CIDR (Classless Inter-Domain Routing) /24 admite utilizar hasta 254 direcciones IP, esta máscara de red se utiliza en la propuesta ya que se utilizan 4 dispositivos con interfaz de red, los cuales se conectan a un switch de capa 2, quedando de esta manera disponible 250 direcciones IP utilizables para dispositivos adicionales que se requieran agregar en el segmento de red. En la tabla 14 se muestran las direcciones IP utilizadas para los dispositivos que se utilizan en la propuesta.

Tabla 14. Direcciones IP utilizadas para los diferentes equipos de la propuesta.

Direcciones IP de la propuesta	
Dispositivo	Dirección IP
Computador Sistema SCADA Ignition	192.168.0.124
PLC Siemens S7-1200	192.168.0.4
Variador de frecuencia Sinamics	192.168.0.10
Raspberry PI 4	192.168.0.17

La topología de red utilizada en la propuesta es estrella (Ver Figura 25), en la cual cada dispositivo utilizado se conecta a un switch de comunicaciones de 8 puertos con capacidad de conmutación de hasta 200Mbps, capacidad favorable para la conmutación rápida de paquetes ethernet provenientes del PLC S7-1200, variador de frecuencia y variables del sistema de visión artificial adquiridos desde la tarjeta Raspberry Pi 4 y de esta manera visualizar en tiempo real cada cambio de valores en el sistema SCADA. En esta topología de red se utilizará el protocolo Profinet entre el PLC S7-1200 y la Raspberry Pi 4 para el intercambio de datos de sensores y el sistema de visión artificial (datos provenientes de una cámara USB) para de esta manera efectuar la activación de los respectivos actuadores del sistema de clasificación.

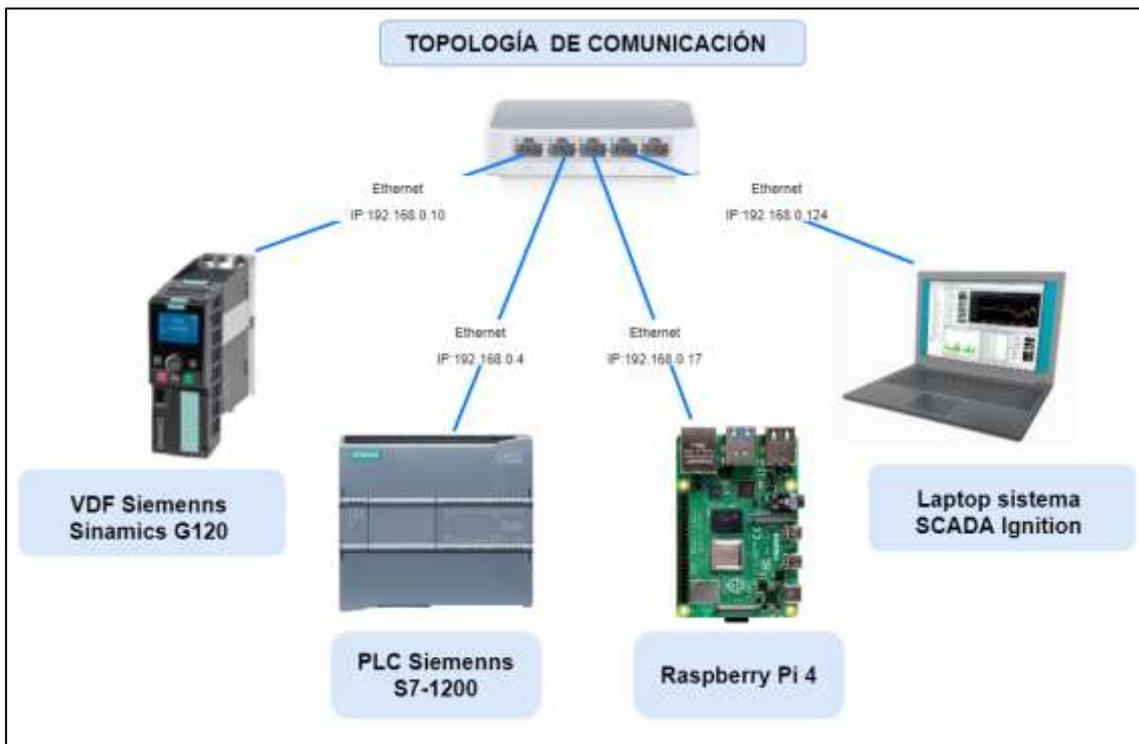


Figura 26. Topología de comunicaciones del sistema de clasificación

3.2.2. Diseño del sistema eléctrico

El diseño eléctrico se lo desarrollo en la herramienta CadeSimu, en la figura tal se muestra el circuito de fuerza, para el variador de frecuencias, este puede ser energizado de manera monofásica o trifásica (240V AC), en el diagrama está diseñado para operar de manera monofásica.

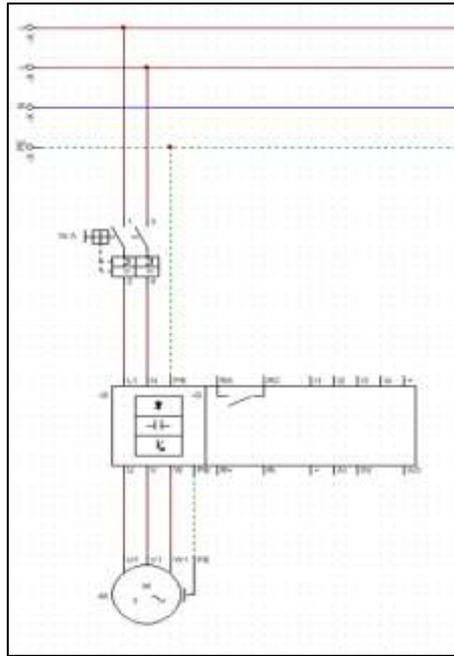


Figura 27. Circuito de fuerza del sistema eléctrico

Para el circuito de control se implementó un disyuntor termomagnético de 2 A para la protección del PLC en caso de alguna sobre carga, también cuenta con pulsadores de NO para la Marcha y dos NC, uno para el Paro y otro para el Paro de Emergencia, y por último una luz indicadora para determinar el estado de la banda.

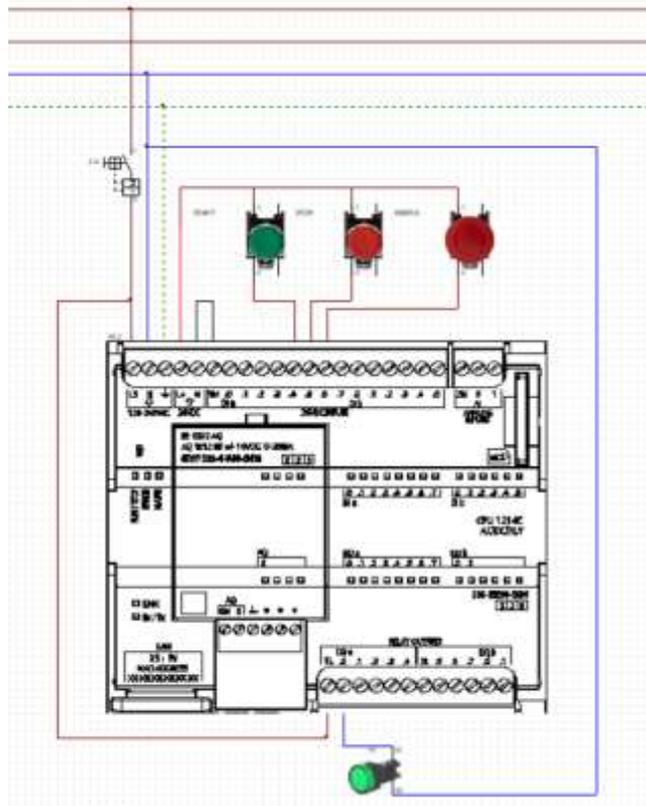


Figura 28. Circuito de control del sistema eléctrico

En el tablero de control consta de tres pulsadores y una luz indicadora en la parte de mandos, también se instaló las Raspberry Pi 4 con su propio disyuntor de 4 A ya que esta trabaja a 5V DC/ 3 A, así mismo se instaló el módulo de transformación de energía de 220V AC a 24V DC, para la correcta alimentación de los drivers que controlaran los motores paso a paso (ver figura)



Figura 29. Tablero de control con sus botoneras de mando

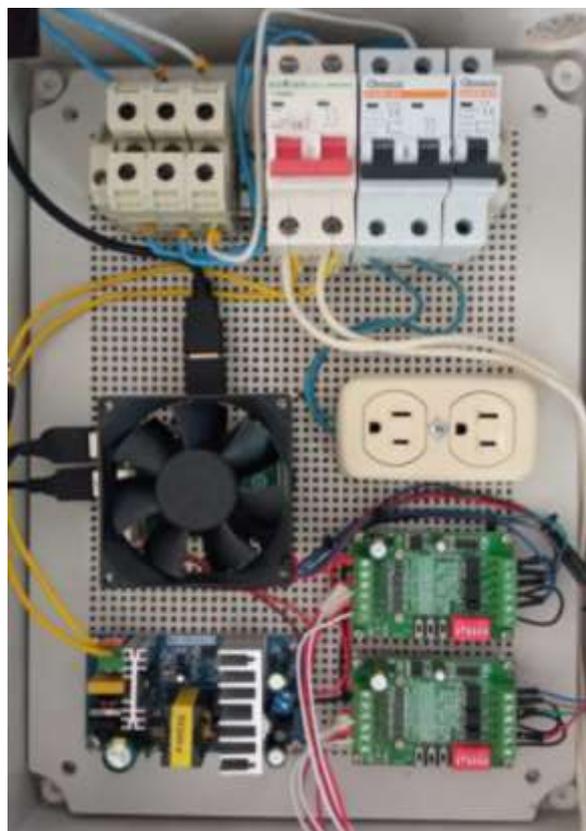


Figura 30. Conexiones del tablero de control eléctrico

3.2.3. Instalación del sistema operativo a la Raspberry pi 4

Para el funcionamiento y uso de la tarjeta Raspberry Pi 4 se realiza la instalación del sistema operativo compatible denominado Raspberry Pi OS, previamente se instala en un computador la herramienta Raspberry Pi Imager el cual ayuda a gestionar la instalación del sistema operativo. Al ejecutar la herramienta desde un computador se procede a seleccionar el sistema operativo, para lo cual el fabricante recomienda el sistema operativo Raspberry Pi OS de 32 bits (Ver Figura 31), esta selección dependerá del tipo de aplicaciones que se requieran instalar y de la capacidad de almacenamiento de la tarjeta microSD, en este caso se selecciona el sistema operativo recomendado el cual instala los puertos, aplicaciones básicas y el escritorio de la tarjeta Raspberry Pi, ya que posteriormente de manera manual procede a instalar las librerías y aplicaciones necesarias para el uso de visión artificial.

Las herramientas y sistemas operativos antes mencionados se encuentran disponibles en la página oficial de Raspberry Pi en el apartado de descargas.

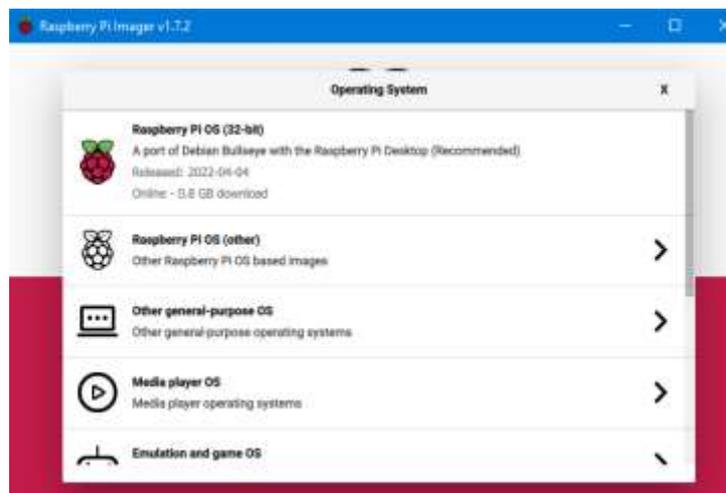


Figura 31. Selección del sistema operativo desde la herramienta Raspberry Pi Imager.

Al realizar la selección del sistema operativo requerido, se procede a colocar en la ranura SD del computador un adaptador SD con la respectiva tarjeta microSD inmediatamente la herramienta Raspberry Pi Imager reconocerá dicha tarjeta (Ver Figura

32), la capacidad de almacenamiento de la tarjeta SD se debe encontrar en el rango de 16GB a 64GB, dicha tarjeta deberá contener soporte para el sistema de archivos FAT16 o FAT32 ya que estas son las extensiones que soporta la Raspberry Pi 4 en los archivos del cargador de arranque, para este caso se hace la selección de la tarjeta micro SD SanDisk Ultra con capacidad de 16GB.

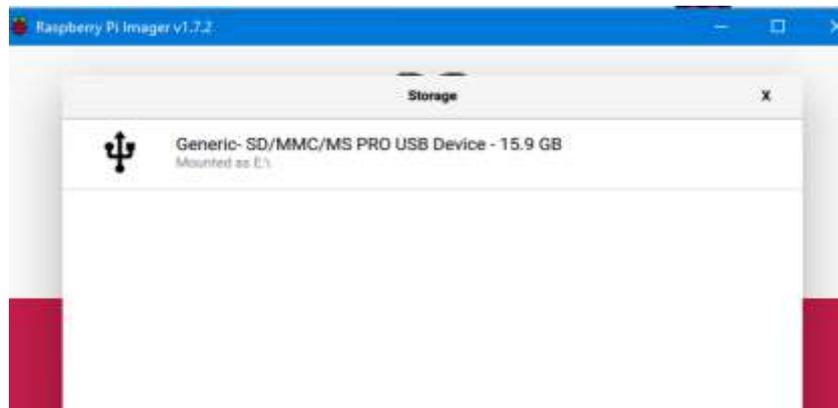


Figura 32. Selección de la unidad SD, para la instalación del sistema Raspberry Pi OS 32 bit.

Una vez seleccionada la tarjeta microSD se procede a escribir o cargar el sistema operativo, para lo cual se empezará a visualizar un Slider con el mensaje “Writing %”, en donde el número que acompaña al símbolo de porcentaje indica la escritura del sistema operativo en la tarjeta microSD, al momento que la escritura llega al 100% se procede a extraer la tarjeta microSD del computador y consecutivamente se conecta dicha tarjeta en la ranura microSD que posee la tarjeta Raspberry Pi 4.



Figura 33. Instalación del sistema operativo Raspberry Pi OS 32 bit en la tarjeta micro SD.

3.2.4. Instalación de librerías y framework para visión e inteligencia artificial en la Raspberry

Una vez instalado el sistema operativo Raspberry Pi OS, se procede a instalar mediante líneas de comandos la biblioteca OpenCV y el framework TensorFlow en su versión Lite, dichas herramientas ayudarán a visualizar y procesar imágenes en tiempo real, adicionalmente con la ayuda de Tensor Flow se procederá a realizar un sistema de aprendizaje automático que permita realizar la clasificación de maracuyá en función de su estado de madurez o si la fruta presentase daños en su superficie.

3.2.4.1. Instalación de OpenCV

La biblioteca de código abierto OpenCV es de vital importancia en el desarrollo de la propuesta ya que mediante esta herramienta se podrá realizar la adquisición de imágenes en tiempo real proveniente de una cámara web USB para su posterior tratamiento digital de dicha imagen, cambio de escala de colores, aplicación de filtros para eliminar ruido entre otras opciones que pueden ser aplicadas a un sistema de visión artificial. A continuación, en la tabla 15 se detallan los comandos a ejecutar en la terminal para la instalación de OpenCV en la tarjeta Raspberry Pi 4, en donde se resalta que cada comando deberá ser ejecutado secuencialmente.

Tabla 15. Secuencia de comando para la instalación de OpenCV

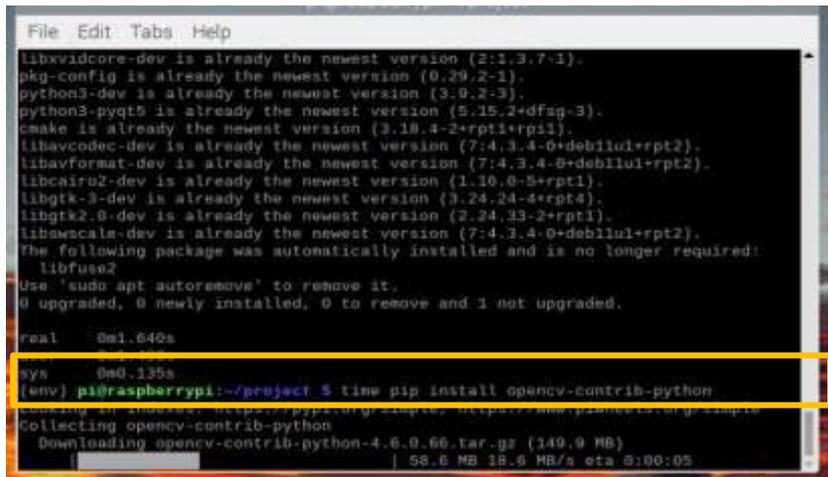
Número	Comandos a emitir en la terminal de la tarjeta Raspberry Pi 4
1	sudo apt-get update && sudo apt-get upgrade
2	sudo apt-get install build-essential cmake pkg-config
3	sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng-dev
4	sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
5	sudo apt-get install libxvidcore-dev libx264-dev
6	sudo apt-get install libfontconfig1-dev libcairo2-dev
7	sudo apt-get install libgdk-pixbuf2.0-dev libpango1.0-dev
8	sudo apt-get install libgtk2.0-dev libgtk-3-dev
9	sudo apt-get install libatlas-base-dev gfortran
10	sudo apt-get install libqtgui4 libqtwebkit4 libqt4-test python3-pyqt5
11	sudo apt-get install python3-dev

La tabla 16 se detalla la descripción de cada comando ejecutado de la tabla 15:

Tabla 16. Descripción de comandos para instalar las dependencias que necesita la biblioteca OpenCV

Número de comando	Descripción
1	Instalar actualizaciones y nuevas versiones de software
2	Instalar el entorno de desarrollo para compilar el código fuente de OpenCV
3	Instalar bibliotecas de Entrada/Salida de imágenes. Bibliotecas para leer y escribir varios tipos de imágenes
4,5	Instalar bibliotecas de Entrada/Salida de video
6	Instalar bibliotecas de gráficos vectoriales Cairo 2D
7	Instalar biblioteca para carga y almacenamiento de imágenes, escalado y carga de animaciones (GIF animados)
8	Instalar conjunto de herramientas multiplataforma para crear interfaces gráficas de usuario.
9	Paquetes necesarios para compilar y construir numpy
10	Instalar biblioteca para la gestión de gráficos
11	Instalar la herramienta de desarrollo para crear módulos de Python

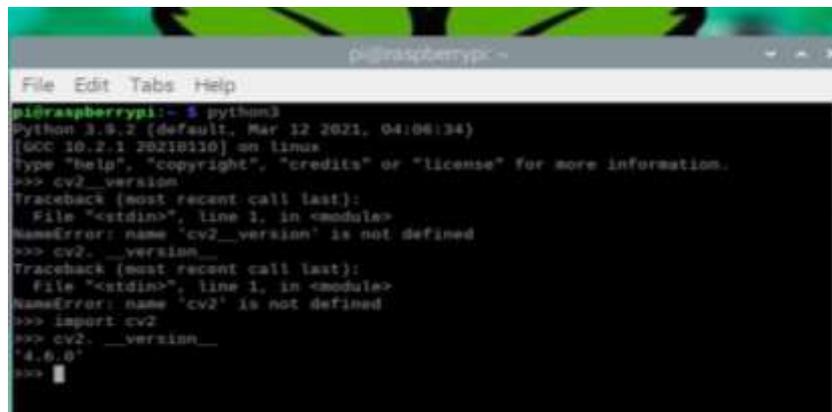
Cuando se finaliza la ejecución de los once comandos respectivamente se procede mediante el terminal de comandos y la herramienta que permite descargar e instalar paquetes de Python denominada pip, instalar el paquete completo de OpenCV el cual incluye módulos principales y módulos de contribución extra con la ejecución del comando *time pip install openCV-contrib-python* (Ver Figura 34).



```
libxvidcore-dev is already the newest version (2:1.3.7-1).
pkg-config is already the newest version (0.29.2-1).
python3-dev is already the newest version (3.9.2-3).
python3-pyqt5 is already the newest version (5.15.2+dfsg-3).
cmake is already the newest version (3.18.4-2+rpt1+rpt1).
libavcodec-dev is already the newest version (7:4.3.4-0+deb11u1+rpt2).
libavformat-dev is already the newest version (7:4.3.4-0+deb11u1+rpt2).
libcairo2-dev is already the newest version (1.16.0-5+rpt1).
libgtk-3-dev is already the newest version (3.24.24-4+rpt4).
libgtk2.0-dev is already the newest version (2.24.33-2+rpt1).
libhwloc-dev is already the newest version (7:4.3.4-0+deb11u1+rpt2).
The following package was automatically installed and is no longer required:
libfuse2
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
rsal 0s1.640s
sys 0s0.135s
[env] pi@raspberrypi:~/project $ time pip install opencv-contrib-python
Collecting opencv-contrib-python
  Downloading opencv-contrib-python-4.6.0.66.tar.gz (149.9 MB)
    | 58.6 MB 18.0 MB/s eta 0:00:05
```

Figura 34. Instalación del paquete completo de la biblioteca OpenCV.

Consecutivamente cuando se culmina la instalación de los módulos principales y módulos de contribución de la biblioteca OpenCV, se procede a verificar que se haya realizado de manera satisfactoria dicha instalación, lo cual es posible verificar mediante los comandos `python3`, `import cv2`, `cv2.__version__`, este último comando ayudara a visualiza la versión respectiva correspondiente a la biblioteca OpenCV (Ver Figura 35).



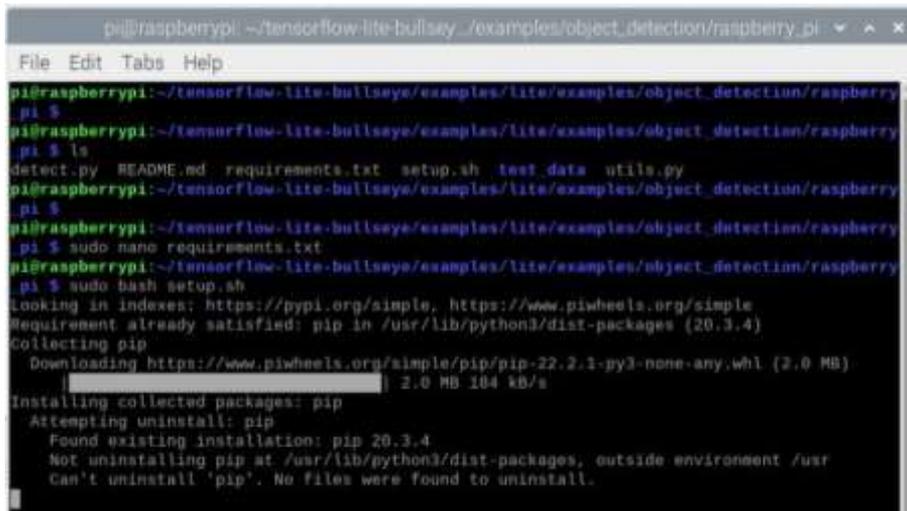
```
pi@raspberrypi:~$ python3
Python 3.9.2 (default, Mar 12 2021, 04:06:34)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> cv2.__version__
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'cv2' is not defined
>>> cv2.__version__
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'cv2' is not defined
>>> import cv2
>>> cv2.__version__
'4.6.0'
>>>
```

Figura 35. Verificación de la versión de openCV instalada en la tarjeta Raspberry Pi 4

3.2.4.2. Instalación de TensorFlow Lite

En este apartado se detalla la instalación del framework TensorFlow Lite, el cual servirá para el desarrollo del modelo de aprendizaje automático, este framework se encuentra disponible en el sitio web GitHub, el cual es una plataforma web que almacena herramientas, aplicaciones de desarrolladores. A continuación, se detallan los pasos a seguir para una correcta instalación:

1. Ingresar a el terminal de comandos de la tarjeta Raspberry Pi 4 y verificar que se encuentre conectado con el usuario “pi”.
2. Una vez ingresado con el usuario “pi” ejecutar en la terminal el comando `sudo git clone https://github.com/freedomwebtech/tensorflow.git`, dicho comando permite extraer el archivo existente del repositorio GitHub y almacenarlo en el directorio home de la respectiva tarjeta Raspberry Pi 4.
3. Luego mediante el comando `ls`, se verifica que la respectiva carpeta se encuentre disponible en el directorio home, para de esta manera asegurar que el comando ejecutado en el paso número 2 se realizó satisfactoriamente (Ver Figura 36).
4. En este paso se ingresa a la carpeta “tensorflow-lite-bullseye” mediante el comando `cd tensorflow-lite-bullseye/`, una vez dentro de dicho directorio se emite el comando `ls` para verificar que exista el archivo `tensorflow-lite.sh`, dicha extensión `.sh` hace referencia a un archivo de Shell Unix, internamente es un script que posee instrucciones en lenguaje bash (lenguaje de comandos).
5. Una vez que se verifica la existencia del archivo `tensorflow-lite.sh` en el directorio `tensorflow-lite` se procede a emitir el comando `sudo chmod 775 tensorflow-lite.sh`, para establecer permisos de ejecución y lectura para el respectivo usuario.
6. Luego se procede a descargar cada uno los paquetes de instalación que contiene el archivo `tensorflow-lite.sh` ejecutando el comando `sudo ./tensorflow-lite.sh`, con lo cual la respectiva descarga se almacenará la carpeta `tensorflow-lite-bullseye` (Ver Figura 37). Una vez que se realizó la descarga se visualizarán dos carpetas una con el nombre “examples” y “tensorflow-lite.sh”.
7. En este paso se accede a una serie de rutas disponibles dentro de la carpeta “examples” con la finalidad de localizar el archivo `setup.sh` el cual permite la instalación del framework `tensorflow-lite` en la tarjeta Raspberry Pi 4, por lo que se emite los siguientes comandos `cd examples/`, `cd lite/`, `cd examples/`, `cd object_detection/`, `cd raspberry_pi/`, teniendo en cuenta que dichos comandos deben emitirse en el respectivo orden.
8. Al momento de ingresar a la carpeta `raspberry_pi` se procede a emitir el comando `ls` para localizar los archivos `requirements.txt` y `setup.sh`.
9. El archivo de extensión `.txt` se procede a editar ejecutando el comando `sudo nano requirements.txt`, en el cual se deberá ingresar la respectiva versión de `openCV`, en este caso se ingresa la versión 4.6.0 ya que es la versión instalada en el apartado 3.2.4.1.



```
pi@raspberrypi: ~/tensorflow-lite-bullseye/examples/object_detection/raspberry_pi
pi $
pi $ ls
detect.py README.md requirements.txt setup.sh test data utils.py
pi $ sudo nano requirements.txt
pi $ sudo bash setup.sh
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: pip in /usr/lib/python3/dist-packages (20.3.4)
Collecting pip
  Downloading https://www.piwheels.org/simple/pip/pip-22.2.1-py3-none-any.whl (2.0 MB)
    | 2.0 MB 184 kb/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.3.4
    Not uninstalling pip at /usr/lib/python3/dist-packages, outside environment /usr
    Can't uninstall 'pip'. No files were found to uninstall.
```

Figura 38. Instalación del framework tensorflow-lite mediante la ejecución del archivo setup.sh.

3.2.5. Desarrollo del sistema de inteligencia artificial

La primera etapa para realizar el sistema de detección de la calidad de la fruta mediante el uso de visión e inteligencia artificial consiste en recolectar el mayor número de imágenes en las cuales se visualicen maracuyá madura, pintona y con daños en la superficie de dicha fruta, con la finalidad de etiquetar cada uno de estos estados de calidad para el entrenamiento de un modelo de inteligencia artificial que permita detectar la calidad de un maracuyá.

La herramienta que permite etiquetar cada una de las imágenes recolectadas con cada uno de los estados de calidad, es el software labelImg el cual es instalado en la tarjeta Raspberry Pi 4, en dicha herramienta es posible marcar cada una de las áreas de interés de las imágenes recolectadas (Ver Figura 39) y etiquetar cada una de estas áreas respectivas con los estados siguientes etiquetas:

- Maracuyá Pintona
- Maracuyá Madura
- Maracuyá con daño en su superficie

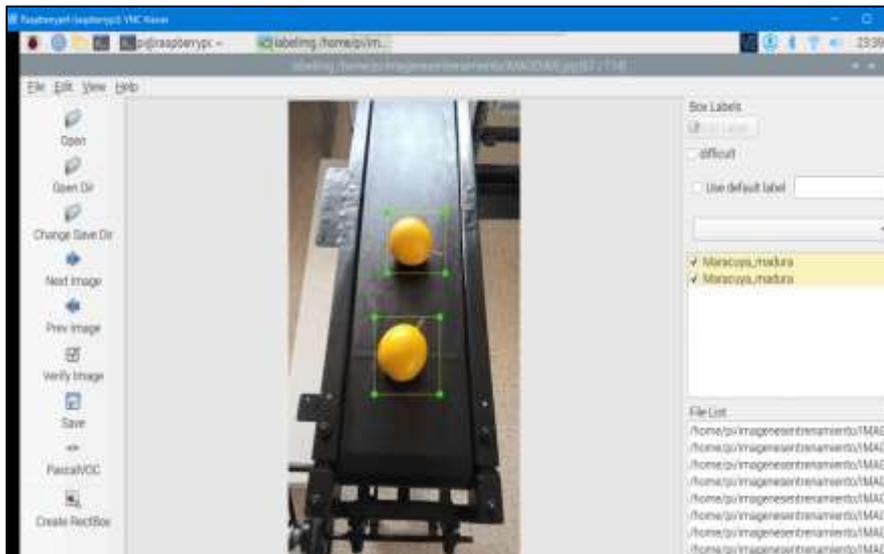


Figura 39. Etiquetado de datos con el color y forma de la fruta para el entrenamiento del sistema de visión e inteligencia artificial.

Cada una de estas etiquetas realizadas en el software labelImg son guardadas en archivos de extensión XML (Extensible Markup Language), el cual contiene el nombre de la etiqueta y el tamaño del área y pixeles que indican la información que se requiere detectar, cada uno de estos archivos XML se guardaron en dos diferentes directorios dentro de la Raspberry Pi en donde se colocó las muestras de las imágenes de datos para el entrenamiento del modelo e imágenes para la validación del respectivo modelo de visión e inteligencia artificial.

```

File Edit Search View Document Help
<annotation>
  <folder>imagenesentrenamiento</folder>
  <filename>IMAGEN1.JPG</filename>
  <path>/home/pi/imagenesentrenamiento/IMAGEN1.JPG</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>814</width>
    <height>410</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>Maracuya Madura</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>173</xmin>
      <ymin>106</ymin>
      <xmax>398</xmax>
      <ymax>319</ymax>
    </bndbox>
  </object>
</annotation>

```

Figura 40. Archivo XML con las características relevantes a detectar.

A continuación, se usa la herramienta Google Colab, para realizar el entrenamiento de un modelo de detección de objetos mediante el framework TensorFlow Lite Model Maker para posteriormente utilizar dicho modelo de detección de la calidad de maracuyá en la tarjeta Raspberry Pi.

En primer lugar, se realiza la instalación de los paquetes necesarios a utilizar, esto incluye los paquetes “model Maker” y la biblioteca “pycocotools” la cual será utilizada para evaluar el modelo de inteligencia artificial, mediante el comando pip install (Ver Figura 41), además se importa cada una de las librerías requeridas tales como numpy, os, de igual forma se indica mediante la instrucción from los objetos en específicos que se requieren importar (Ver Figura 42).

```
!pip install -q tflite-model-maker
!pip install -q tflite-support
```

Figura 41. Instalación de paquetes necesarios para la construcción del modelo de inteligencia artificial.

```
import numpy as np
import os

from tflite_model_maker.config import ExportFormat, QuantizationConfig
from tflite_model_maker import model_spec
from tflite_model_maker import object_detector

from tflite_support import metadata

import tensorflow as tf
assert tf.__version__.startswith('2')

tf.get_logger().setLevel('ERROR')
from absl import logging
logging.set_verbosity(logging.ERROR)
```

Figura 42. Librerías a utilizar importadas para la construcción del modelo de inteligencia artificial

Luego de realizar la respectiva instalación de paquetes e importar cada una de las librerías requeridas, se procede con la carga de cada una de las muestras de imágenes previamente etiquetadas con cada uno de los estados de calidad del maracuyá a la plataforma Google drive, resaltando que previamente el directorio el cual contiene las muestras para el entrenamiento y la validación del modelo de inteligencia artificial debe de ser un archivo de extensión ZIP. Luego de haber cargado el archivo ZIP proveniente

de Google Drive se procede a descomprimir el directorio para de esta manera cargar cada una de las muestras obtenidas al modelo de inteligencia artificial en desarrollo, mediante el comando `unzip/mydrive/InteligenciaArt_tesis.zip` es posible descomprimir el respectivo directorio.

```
from google.colab import drive
drive.mount('/content/gdrive')
!ln -s /content/gdrive/My\ Drive/ /mydrive
!ls /mydrive
```

Figura 43. Carga del directorio que contiene las muestras de imágenes previamente etiquetadas al sitio Google Drive.

Al momento de descomprimir cada uno de los datos y ser cargados al modelo de inteligencia artificial en desarrollo, se procede cargar dichos datos en variables para el posterior entrenamiento y validación del conjunto de muestras recolectadas, las cuales previamente fueron etiquetados en el software labelImg, estas muestras se encuentran en archivos XML en formato de datos Pascal VOC (Pascal Visual Object Classes), con lo cual para poder cargar y acceder a estos datos se utiliza la instrucción `object_detector_DataLoader.from_pascal_voc`, en donde esta instrucción tiene como argumentos la ruta del directorio en el cual se encuentran las imágenes sin procesar, la ruta del directorio en donde se ubican las imágenes con las respectivas etiquetas y como tercer parámetro las respectivas cadenas de texto utilizadas para la detección de la calidad de la maracuyá (Ver Figura 44).

```
train_data = object_detector.DataLoader.from_pascal_voc(
    'InteligenciaArt_tesis/train',
    'InteligenciaArt_tesis/train',
    ['Maracuya_madura', 'Maracuya_semimadura', 'Maracuya_con_daño ']
)

val_data = object_detector.DataLoader.from_pascal_voc(
    'InteligenciaArt_tesis/validate',
    'InteligenciaArt_tesis/validate',
    ['Maracuya_madura', 'Maracuya_semimadura', 'Maracuya_con_daño ']
)
```

Figura 44. Entrenamiento y validación del modelo de inteligencia artificial.

Una vez que se guardan los datos, se selecciona la arquitectura que tendrá el modelo de inteligencia artificial para la detección de calidad del maracuyá, el framework TensorFlow Lite Model Maker permite 5 diferentes tipos de modelos con los cuales se pueden realizar detección de objetos, la principal diferencia en estos 5 modelos radica en el tiempo de respuesta y el porcentaje de precisión promedio del modelo para detectar correctamente el o los objetos de interés (Ver Tabla 17).

Tabla 17. Arquitectura de modelo de detección de objetos

Arquitectura de modelo de detección de objetos			
Arquitectura del Modelo	Tamaño (MB)	Latencia (ms)	Precisión del modelo
EfficientDet-Lite0	4,4	37	25,69%
EfficientDet-Lite1	5,8	49	30,55%
EfficientDet-Lite2	7,2	69	33,97%
EfficientDet-Lite3	11,4	116	37,70%
EfficientDet-Lite4	19,9	260	41,96%

Como se visualiza en la tabla A, a medida que el modelo incrementa de nivel, el tamaño y el tiempo de respuesta aumentan proporcionalmente lo cual hace referencia a un modelo un poco lento en la detección de los respectivos objetos, para el caso de la precisión de la arquitectura de cada modelo, esta incrementa a medida de que el nivel de arquitectura aumenta, para el caso de la presente propuesta se selecciona un modelo de arquitectura 0 el cual tiene un tamaño de 4.4MB, un rápido tiempo de respuesta de 37ms y una precisión promedio alrededor del 26%. Para definir la arquitectura a utilizar en el presente modelo se utilizó la instrucción:

- `spec = model_spec.get('efficientdet_lite0')`

Luego de seleccionar la arquitectura requerida para el sistema de detección de calidad del maracuyá, se procede a crear y entrenar el respectivo modelo mediante la instrucción `object_detector.create`, con el cual se ejecutara una serie de iteraciones para el entrenamiento con los respectivos datos, estas iteraciones se ejecutarán hasta que el sistema detecte que los valores de pérdida de validación (`val_loss`) dejen de disminuir, adicionalmente mediante el parámetro `train_whole` establecido en el valor lógico True, se logra afinar todo el modelo en lugar de solo entrenar la capa principal, de esta forma se

mejora el modelo de detección, una desventaja de aquello es que el entrenamiento del modelo puede durar más tiempo.

```
model = object_detector.create(train_data, model_spec=spec, batch_size=4, train_whole_model=True, epochs=20, validation_data=val_data)

Epoch 1/20
19/19 [=====] - 121s 4s/step - det_loss: 1.7226 - cls_loss: 1.1382 - box_loss: 0.0118 - reg_l2_loss: 0.0630 - loss
Epoch 2/20
19/19 [=====] - 73s 4s/step - det_loss: 1.6114 - cls_loss: 1.0738 - box_loss: 0.0100 - reg_l2_loss: 0.0630 - loss
Epoch 3/20
19/19 [=====] - 74s 4s/step - det_loss: 1.4470 - cls_loss: 0.9488 - box_loss: 0.0100 - reg_l2_loss: 0.0630 - loss
Epoch 4/20
19/19 [=====] - 74s 4s/step - det_loss: 1.2717 - cls_loss: 0.8889 - box_loss: 0.0093 - reg_l2_loss: 0.0630 - loss
Epoch 5/20
```

Figura 45. Entrenamiento del modelo de detección de calidad de maracuyá.

Después de realizar el proceso de entrenamiento y validación del modelo de detección de la calidad del maracuyá, se procede a exportar dicho modelo mediante la instrucción mostrada en la siguiente figura, la cual permite especificar el formato del modelo de TensorFlow, este es de tipo tflite, además se especifica el directorio al cual se exporta el respectivo modelo, un dato relevante el cual es la técnica de cuantificación utilizada posteriormente al entrenamiento es full integer quantization, dicha técnica permite que el modelo entrenado en el framework TensorFlow Lite sea más pequeño y se ejecute más rápido en la CPU de la tarjeta Raspberry Pi 4.

```
model.export(export_dir='.', tflite_filename='ModeloIA_detectorcalidadMaracuya.tflite')
```

Figura 46. Instrucción para exportar el respectivo modelo de detección de calidad del maracuyá.

El modelo resultante será utilizado en conjunto con la biblioteca OpenCV para de esta forma obtener datos de video en tiempo real proveniente de la cámara web USB, procesar las respectivas datos e imágenes obtenidas del video, extraer la información de las etiquetas detectadas por el sistema de inteligencia artificial y de esta manera realizar la activación de los actuadores correspondientes al sistema de clasificación.

3.2.6. Desarrollo del sistema de visión artificial con Opencv

Para el desarrollo del sistema de visión artificial y la utilización del sistema de detección de maduración de la fruta desarrollado en TensorFlow Lite, es necesario utilizar la biblioteca de OpenCV, para de esta manera capturar los fotogramas provenientes de la cámara web USB, enmarcar cuadros sobre los objetos detectados, mostrar textos, entre otras opciones. La siguiente figura muestra la codificación inicial en Python y OpenCV en la cual se importan las respectivas librerías y framework a utilizar, primordialmente se utiliza la instrucción `cv2.Capture`, con esta instrucción se establece conexión con la cámara USB conectada en la tarjeta Raspberry Pi, de esta manera se adquiere las respectivas imágenes o fotogramas para el sistema de detección de calidad del maracuyá.

```
#Importar biblioteca de opencv y módulos necesarios|
import cv2
import argparse
import sys
import time
import utils
# Importar framework de TensorFlow Lite
from tflite_support.task import core
from tflite_support.task import processor
from tflite_support.task import vision
# Empezar a capturar la entrada del video de la cámara web
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, width) #ancho del marco de la cámara a visualizar
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, height)#alto del marco de la cámara a visualizar
time.sleep(2)
```

Figura 47. Codificación en Python para importar Librerías, framework y lectura de la entrada de video.

Luego de inicializar la entrada de video correspondiente a la cámara web USB, se procede a iniciar el modelo de detección de calidad de maracuyá, utilizando instrucciones de los objetos `core`, `processor` y `vision` importados del framework TensorFlow Lite con las cuales es posible realizar el sistema de inteligencia artificial, a continuación, en la tabla 18 se detallan los atributos relevantes de los objetos previamente mencionados.

```
# Inicializar el modelo de detección de la calidad de maracuyá
base_options = core.BaseOptions(
    file_name=model, use_coral=enable_edgetpu, num_threads=num_threads)
detection_options = processor.DetectionOptions(
    max_results=3, score_threshold=0.3)
options = vision.ObjectDetectorOptions(
    base_options=base_options, detection_options=detection_options)
detector = vision.ObjectDetector.create_from_options(options)
```

Figura 48. Codificación en Python para inicializar el modelo de detección de objetos creado en TensorFlow Lite.

Tabla 18. Objetos del framework TensorFlow Lite para inicializar el modelo de IA.

Objetos del framework TensorFlow Lite		
Objeto	Atributo	Descripción
processor.DetectionOptions	max_results	Número máximo de resultados de clasificación mejor puntuados devuelto por el atributo.
	score_threshold	Los resultados por debajo de este valor se rechazan
vision.ObjectDetectorOptions	base_options	Campo de clase de datos
	detection_options	Campo de clase de datos
core.BaseOptions	file_name	Ruta al modelo de inteligencia artificial
	use_coral	Si es verdadero, la inferencia se delegará a un dispositivo coral Edge TPU conectado

Una vez inicializado la entrada de video de la cámara web se procede a validar mediante un ciclo while el inicio de captura de los respectivos fotogramas mediante la instrucción isOpened(), con esto se asegura que el proceso de adquisición de fotogramas se inicie exitosamente, luego mediante la instrucción read() se realiza la lectura de cada uno de los fotogramas que forman el video proveniente de la webcam para de esta manera guardar en una variable la respectiva imagen, el contenido de dicha variable será una

matriz de datos que contiene cada uno de los pixeles que forman dicha imagen, adicionalmente mediante la instrucción `cv2.flip` se realiza un giro sobre el eje Y la imagen adquirida es decir se voltea horizontalmente la imagen para obtener una correcta orientación y visualización del sistema de detección de calidad de maracuyá.

```
# Ciclo while, capturar continuamente imágenes desde la cámara web USB
# al momento de que se ha comenzado a capturar los fotogramas:
while cap.isOpened():
    # success = True ? se obtuvo el marco de la cámara
    #imagen_webcam: guardar la imagen en dicha variable
    success, imagen_webcam = cap.read() # Leer el video proveniente de la webcam
    if not success:
        sys.exit(
            'No se puede leer el video desde la cámara web USB'
        )
    #variable contador frames
    contador += 1
    # Girar las imágenes alrededor del eje Y=1, =0 giro alrededor del eje X
    imagen = cv2.flip(imagen_webcam, 1)
```

Figura 49. Codificación para inicializar la captura de fotogramas provenientes de la cámara web USB.

En la siguiente etapa de codificación mostrada en la siguiente figura, se realiza el procesado de la imagen adquirida de la cámara web, en el cual el primer paso es convertir la imagen full color a una escala RGB mediante la instrucción `cv2.cvtColor`, luego se procede realizar la estimación de la primitiva de los patrones geométricos denominados tensores, estos tensores dependen de la forma del objeto a detectar, en este caso se utiliza la instrucción `vision.TensorImage.create_from_array`, posteriormente cada uno de estos patrones geométricos encontrados en los fotogramas son enmarcados mediante cuadros los cuales permitirán visualizar la respectiva detección de calidad de maracuyá.

```
# Convertir la imagen de BGR a RGB para utilizar el modelo de TensorFlow Lite
rgb_image = cv2.cvtColor(imagen, cv2.COLOR_BGR2RGB)
# Crear un objeto de imagen de tensor a partir de la imagen RGB
input_tensor = vision.TensorImage.create_from_array(rgb_image)
# Ejecutar la estimacion de la detección de objetos usando el modelo
# de inteligencia artificial creado
detection_result = detector.detect(input_tensor)
# Dibujar puntos claves y bordes en la imagen
imagen = utils.visualize(imagen, detection_result)
```

Figura 50. Codificación en Python para realizar el procesado de imagen y detección de calidad del maracuyá.

En esta sección final de la codificación en Python del sistema de visión artificial, se calculan los fotogramas por segundo (FPS) para de esta manera visualizar la fluidez con la cual se visualizan los fotogramas en el monitor respectivo conectado en el puerto micro HDMI de la tarjeta Raspberry Pi 4, cuanto menor sea el valor de los fotogramas por segundo la visualización del video proveniente de la cámara web USB se observara de manera lenta, mientras que si se obtiene un valor de FPS mayor el sistema de detección se visualizará de manera fluida, este valor de FPS depende mucho del CPU ARM Cortex-A72 incorporado en la Raspberry Pi 4, en este caso dicho CPU permite la decodificación de video 4K a 60 fotogramas por segundo. Para mostrar el resultado de la imagen previamente procesada y enmarcada mediante rectángulos la respectiva detección, se utiliza la instrucción imshow la cual tiene como atributos el titulo con el cual se desea visualizar el marco de la cámara y la respectiva imagen resultante.

```

# Calcular los FPS
if contador % fps_avg_frame_count == 0:
    end_time = time.time()
    fps = fps_avg_frame_count / (end_time - start_time)
    start_time = time.time()

# Mostrar los FPS
fps_text = 'FPS = {:.1f}'.format(fps)
text_location = (left_margin, row_size)
cv2.putText(imagen, fps_text, text_location, cv2.FONT_HERSHEY_PLAIN,
            font_size, text_color, font_thickness)

# Fin del programa si se presiona la Tecla ESC o "s"
if cv2.waitKey(1) & 0xFF == ord('s'):
    break
cv2.imshow('Detector calidad de maracuyá', imagen)

cap.release()
cv2.destroyAllWindows()

```

Figura 51. Codificación para visualizar los FPS y la imagen resultante del sistema de detección de calidad.

3.2.7. Configuración y programación del Sinamics G120 en el software TIA portal

Para el correcto funcionamiento de variador de frecuencia y su correcta comunicación con el PLC vía Profinet, se requiere hacer previas configuraciones desde el Tia Portal, estas configuraciones comprenden, el direccionamiento correcto de la IP de este equipo, ya que, los equipos a comunicar se tienen que encontrar en la misma red para establecer correcta comunicación. (Ver Figura 52 Y 53)

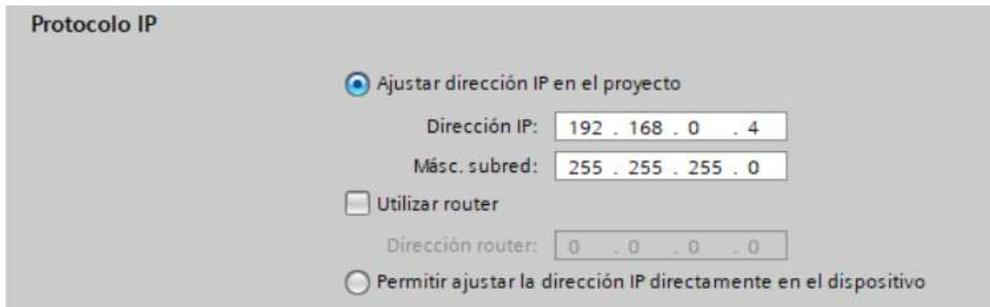


Figura 52. Asignación de dirección IP para el PLC S7-1200

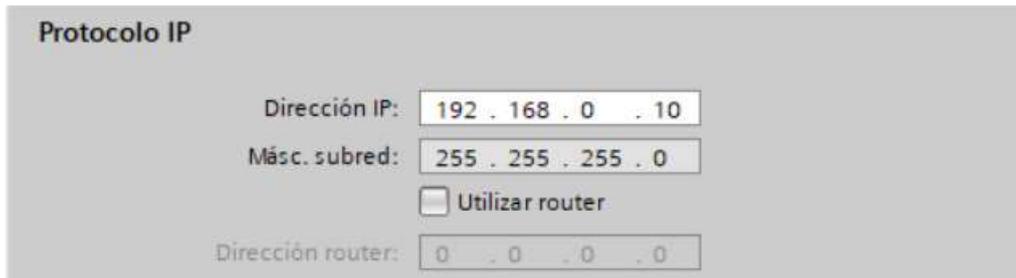


Figura 53. Asignación de dirección IP variador de frecuencia Sinamics G120

Para establecer la comunicación entre el plc y el variador de frecuencia, se da click en dispositivos y redes, luego en vista de redes para enlazar los dispositivos que se van a comunicar (Ver Figura 54).

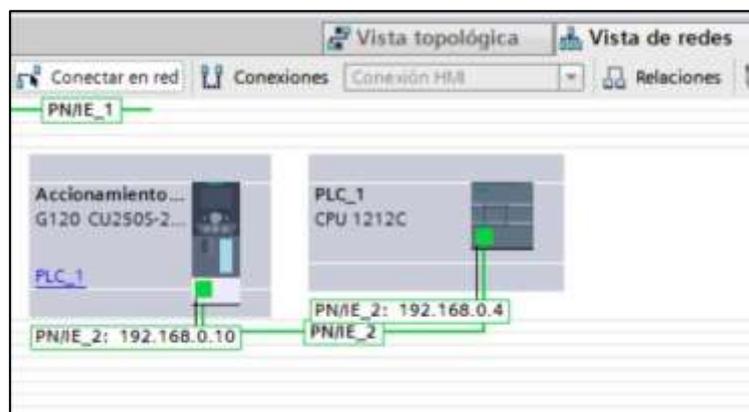


Figura 54. Comunicación Profinet entre el PLC s7-1200 y el Sinamics G120

El siguiente paso es dar click en puesta en servicio, seguido a esto en el asistente de puesta en marcha para Parametrizar el motor a controlar por el variador de frecuencia G120 desde el software TIA Portal (Ver Figura 55).



Figura 55. Parametrización del motor a controlar desde el TIA Portal

El bloque que se necesita para programar el variador de frecuencia desde el TIA Portal, es el SINA_SPEED, donde se podrá programar la velocidad del motor, como se muestra en la figura 56 y 57.



Figura 56. Selección del bloque SINA_SPEED

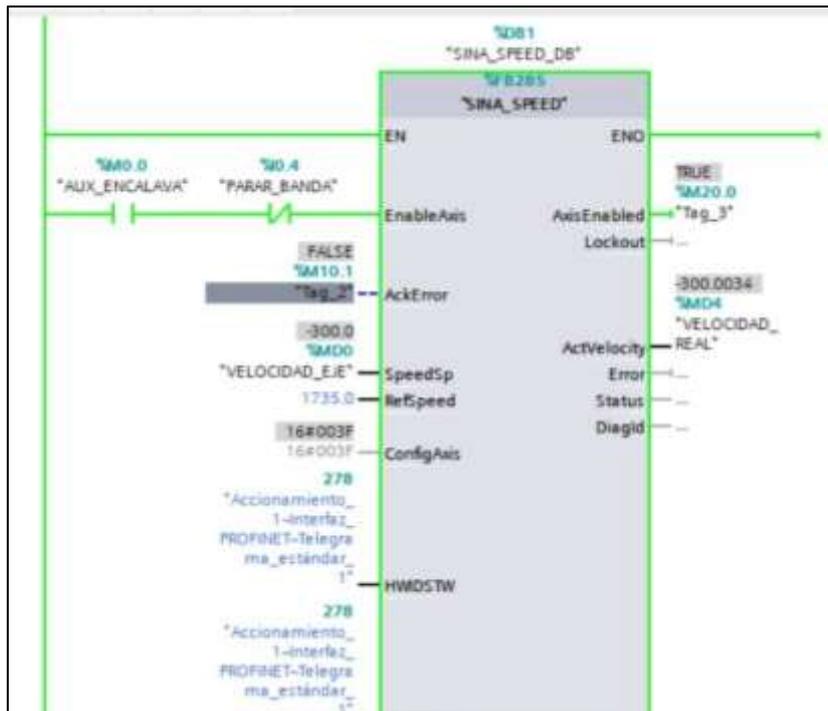


Figura 57. Programación y funcionamiento de bloque SINA_SPEED

3.2.8. Programación de los sensores y el sistema de conteo en TIA Portal

Para la programación de los sensores y el sistema que va a contar cuanto maracuyá han sido clasificada por su estado de maduración, necesitamos crear las variables, donde se encuentran tres tipos de datos, booleanos, reales y enteros. Los sensores fueron asignados con el nombre SEN_MS, SEN_MM y SEN_MD (Ver Figura 58)

Variables PLC				
	Nombre	Tabla de variables	Tipo de datos	Dirección
1	Tag_2	Tabla de variables e.	Bool	%M10.1
2	Tag_3	Tabla de variables e.	Bool	%M20.0
3	VELOCIDAD_EJE	Tabla de variables e.	Real	%MD0
4	VELOCIDAD_REAL	Tabla de variables e.	Real	%MD4
5	ACCIONAR_BANDA	Tabla de variables e.	Bool	%I0.3
6	PARAR_BANDA	Tabla de variables e.	Bool	%I0.4
7	AUX_ENCALAVA	Tabla de variables e.	Bool	%M0.0
8	INDICADOR_BANDA	Tabla de variables e.	Bool	%Q0.0
9	SEN_MS	Tabla de variables e.	Bool	%I0.0
10	SEN_MM	Tabla de variables e.	Bool	%I0.2
11	SEN_MD	Tabla de variables e.	Bool	%I0.7
12	RESET	Tabla de variables e.	Bool	%M0.1
13	CONTADOR_MS	Tabla de variables e.	Int	%MW106
14	CONTADOR_MM	Tabla de variables e.	Int	%MW102
15	CONTADOR_MD	Tabla de variables e.	Int	%MW104

Figura 58. Variables utilizadas para el sistema de clasificación y conteo

Los sensores van a enviar datos booleanos por lo que, cada vez que detecte un maracuyá, independientemente por su estado de maduración, dicha señal entrara al bloque

CUT, para contar los maracuyás ascendentemente, también se creó una variable con el nombre RESET, del tipo memoria para que dese el SCADA se pueda resetear el conteo (Ver Figura 59 a 61).

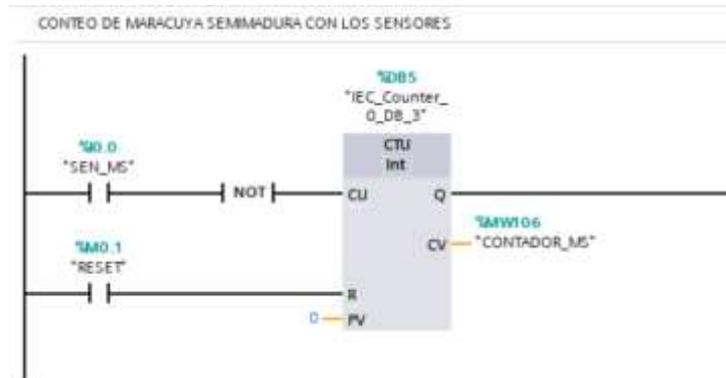


Figura 59. Programación escalera para el sistema de conteo de maracuyás pintonas

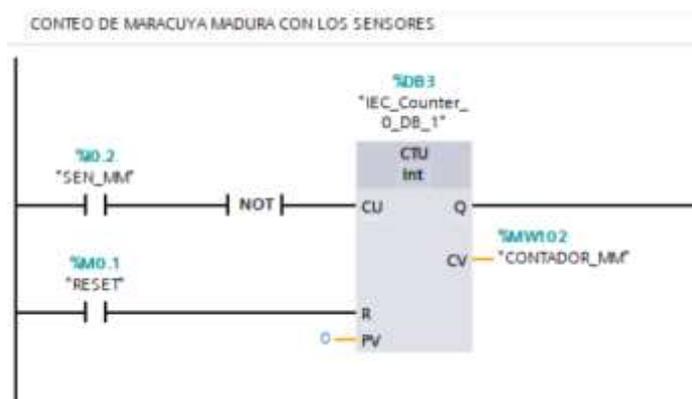


Figura 60. Programación escalera para el sistema de conteo de maracuyás maduras

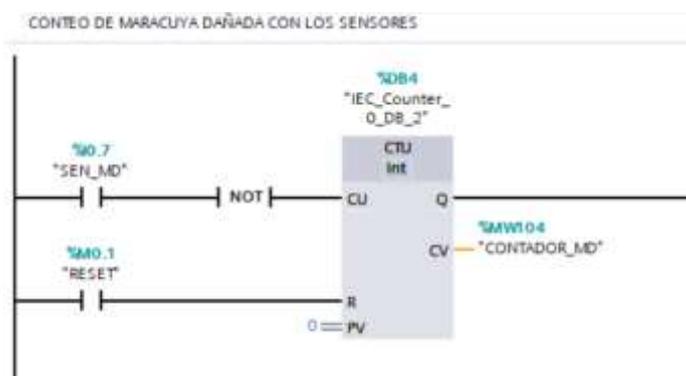


Figura 61. Programación escalera para el sistema de conteo de maracuyás maduras

3.2.9. Programación del mecanismo de clasificación en la Raspberry

En este apartado se realiza la programación en lenguaje Python, instalado en la tarjeta Raspberry Pi 4 para el mecanismo de clasificación de maracuyá de acuerdo a la calidad esto, incluye estado de madurez y presencia de daño en la superficie de la fruta, dicha clasificación se la realiza en base al estado detectado por el sistema de visión artificial, los cuales pueden ser fruta pintona, madura y con presencia de daño en la superficie, según el estado, se activa el sistema de actuadores conformado por motores paso a paso, los cuales se encargan de distribuir la maracuyá en un respectivo recipiente.

Para activar cada uno de los motores paso a paso, se hace uso de controladores TB6650 los cuales permiten suministrar el amperaje necesario para el funcionamiento de los motores paso a paso, además mediante este controlador es posible suministrar funciones de sentido de giro, pulso de trabajo, habilitación o desactivación del respectivo controlador, dichas funciones se utilizarán mediante el uso de pines de entrada o salida de propósito general (GPIO) de la tarjeta Raspberry Pi 4 conectados a los pines de funciones del controlador TB6650, en la siguiente tabla se detalla los pines GPIO que se utilizan en el sistema de clasificado, adicionalmente en la figura 62 se muestra el diagrama de conexiones de los motores paso a paso con los respectivos controladores.

Tabla 19. Pines GPIO a utilizar para el control del sistema de clasificación.

Motor	Pines GPIO Raspberry Pi 4	Pin controlador TB6650	Descripción
Motor Paso a Paso 1	GPIO 2	CLK+	Pin para el pulso de trabajo Motor 1
	GPIO 3	CW+	Pin para el sentido de giro Motor 1
	GPIO 4	EN+	Pin para habilitar el controlador Motor 1
Motor Paso a Paso 2	GPIO 17	CLK+	Pin para el pulso de trabajo Motor 2
	GPIO 22	CW+	Pin para el sentido de giro Motor 2
	GPIO 27	EN+	Pin para habilitar el controlador Motor 2

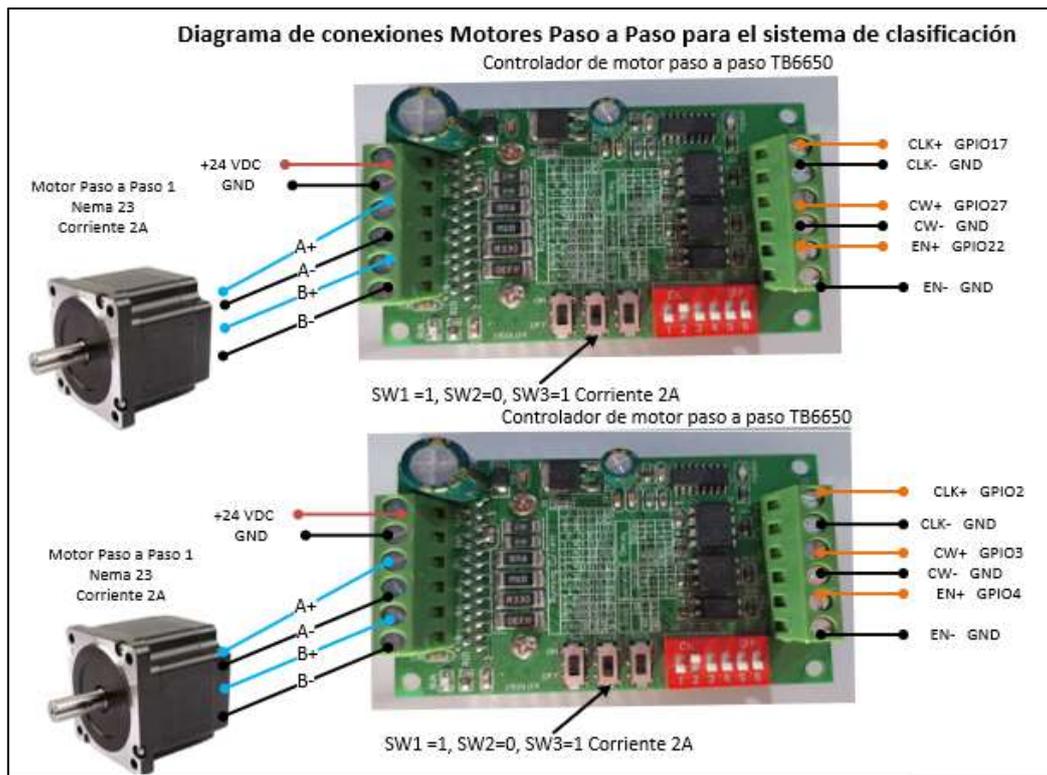


Figura 62. Diagrama de conexiones Motores paso a paso y controlador TB6650.

Previamente a la realización de la respectiva programación del sistema de clasificación se diseña un diagrama de flujo mostrado en la figura 63.

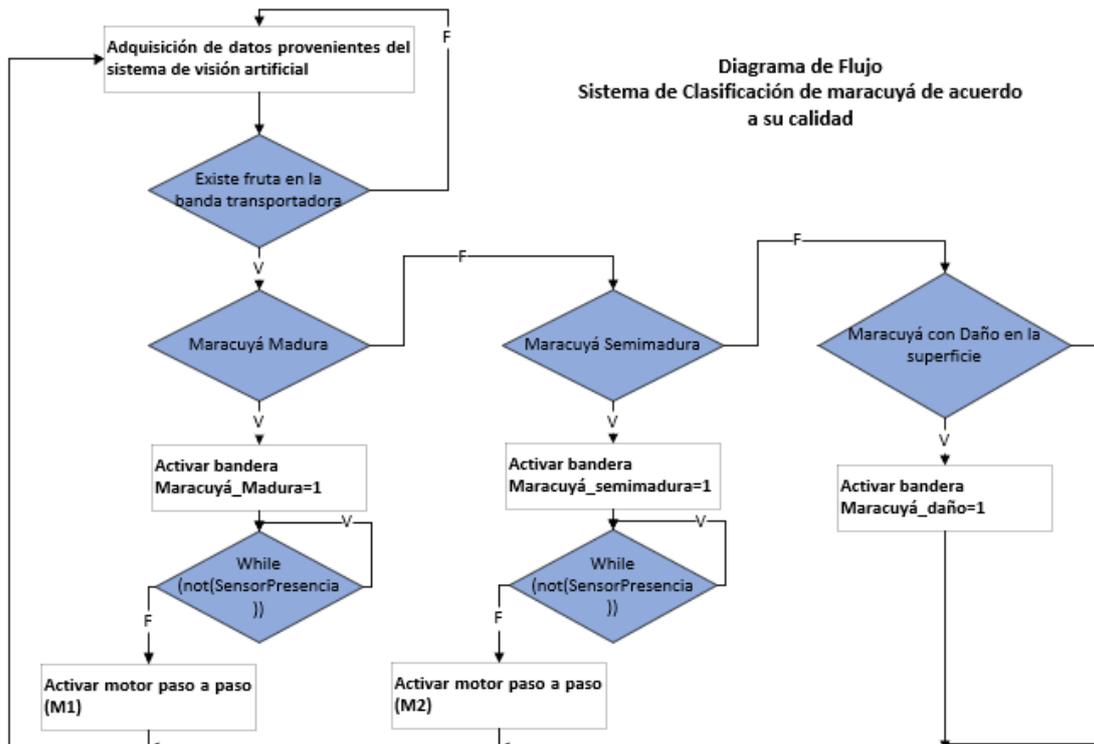


Figura 63. Diagrama de flujo del sistema de clasificación

Mediante Thonny el cual es un IDE de Python instalado en la tarjeta Raspberry Pi 4 se realiza la respectiva programación del sistema de clasificación, en el cual, en primer lugar, se declara los seis pines GPIO a utilizar para el control de los motores paso a paso (Ver Tabla 18), los cuales permiten habilitar el controlador, el sentido de giro y el pulso de trabajo.

```
# Pines GPIO Motor Paso a Paso 1
PULM1 = 17 # Step Bit, Pulsos M1 (PIN CLK+ TB6650)
DIRM1 = 27 # Direction Bit Controlar sentido de giro M1 (PIN CW+ TB6650)
ENAM1 = 22 # Enable Bit (High Habilita / LOW Deshabilita) M1 (Pin EN+ TB6650)
# Pines GPIO Motor Paso a Paso 2
PULM2=2    # Step Bit, Pulsos M2 (PIN CLK+ TB6650)
DIRM2=3    # Direction Bit Controlar sentido de giro M2 (PIN CW+ TB6650)
ENAM2=4    # Enable Bit (High Habilita / LOW Deshabilita) M2 (Pin EN+ TB6650)
```

Figura 64. Programación en Python de los pines GPIO de la tarjeta Raspberry Pi 4 a utilizar.

Cuando se indica los pines GPIO a utilizar de la tarjeta Raspberry Pi 4, consecutivamente se denota el modo de trabajo de dichos pines los cuales pueden ser configurados como entrada (INPUT) o salida (OUTPUT), en este caso para el control de los motores paso a paso los seis pines en uso se establecen en modo salida, ya que , adicionalmente para evitar errores o accionamientos inesperados al encender la tarjeta Raspberry Pi se coloca en estado bajo (LOW) todos los pines en uso, esto ocasionará que el controlador TB6650 se encuentre desactivado y por consiguiente ambos motores a paso inicializaran apagados.

```
GPIO.setmode(GPIO.BCM)
#Declaración de pines como salida
GPIO.setup(PULM1, GPIO.OUT)
GPIO.setup(DIRM1, GPIO.OUT)
GPIO.setup(ENAM1, GPIO.OUT)
GPIO.setup(PULM2, GPIO.OUT)
GPIO.setup(DIRM2, GPIO.OUT)
GPIO.setup(ENAM2, GPIO.OUT)

#Colocar en Stop M1 y M2 al momento de iniciar el proceso
GPIO.output(DIRM1, GPIO.LOW)
GPIO.output(PULM1, GPIO.LOW)
GPIO.output(ENAM1, GPIO.LOW)
GPIO.output(DIRM2, GPIO.LOW)
GPIO.output(PULM2, GPIO.LOW)
GPIO.output(ENAM2, GPIO.LOW)
```

Figura 65. Programación en Python del modo de trabajo e inicialización de los pines GPIO.

Al momento de que el sistema realiza la validación de que se encuentra presente en la banda transportadora una maracuyá, el sistema de visión e inteligencia artificial determina la calidad de la misma, dicha calidad de la fruta es almacenada mediante variables de tipo String, con la cual es posible activar el motor paso a paso de acuerdo a la calidad de la maracuyá detectada, en donde se resalta que previamente para activar el motor paso a paso respectivo, se codificó un ciclo while con el dato del sensor fotoeléctrico proveniente del PLC S7-1200, para de esta manera validar que el sistema de clasificación se acciona al momento de detectar el paso de la maracuyá.

```
image = utils.visualize(image, detection_result) # Dibujar bordes y etiquetas en la imagen adquirida
# desde la cámara USB
for detection in detection_result.detections:
    categoria = detection.classes[0]
    clase=categoria.class_name # Tipo de fruta detectada
    if clase == 'MaracuyaMadura':
        flag_MaracuyaMadura=1
        while not(SensorM1==0): #Condicion de monitoreo de sensor fotoeléctrico
            GPIO.output(ENAM1,GPIO.HIGH) # Habilitar el controlador TB6658
            sleep(0.5) # Pausa de 0.5 segundos
            GPIO.output(DIRM1, GPIO.LOW) #Sentido de giro en contra de las manecillas de reloj
            #Ciclo for para realizar el accionamiento del motor a paso
            for x in range(500):
                # Pulso de trabajo a frecuencia de 1kHz
                GPIO.output(PULM1,GPIO.HIGH)
                sleep(0.0005)
                GPIO.output(PULM1,GPIO.LOW)
                sleep(0.0005)
```

Figura 66. Programación en Python para el accionamiento del motor paso a paso

3.2.10. Comunicación entre la Raspberry Pi 4 y el PLC s7-1200

En este apartado se describe la comunicación entre el controlador lógico programable S7-1200 y la tarjeta Raspberry Pi 4 para el envío de datos, esta comunicación es bidireccional, es decir el controlador lógico realiza envíos de datos hacia la Raspberry Pi 4, y de igual forma la tarjeta Raspberry puede enviar datos hacia el controlador lógico. Esta comunicación se la realiza mediante el uso del software multiplataforma Snap 7 instalada en la tarjeta Raspberry, la cual permite comunicar con el controlador lógico Siemens S7 1200 mediante tecnología ethernet y protocolo S7 protocolos nativos de controladores lógicos de Siemens, en la siguiente figura se muestra el esquema de comunicación entre estos dos dispositivos.

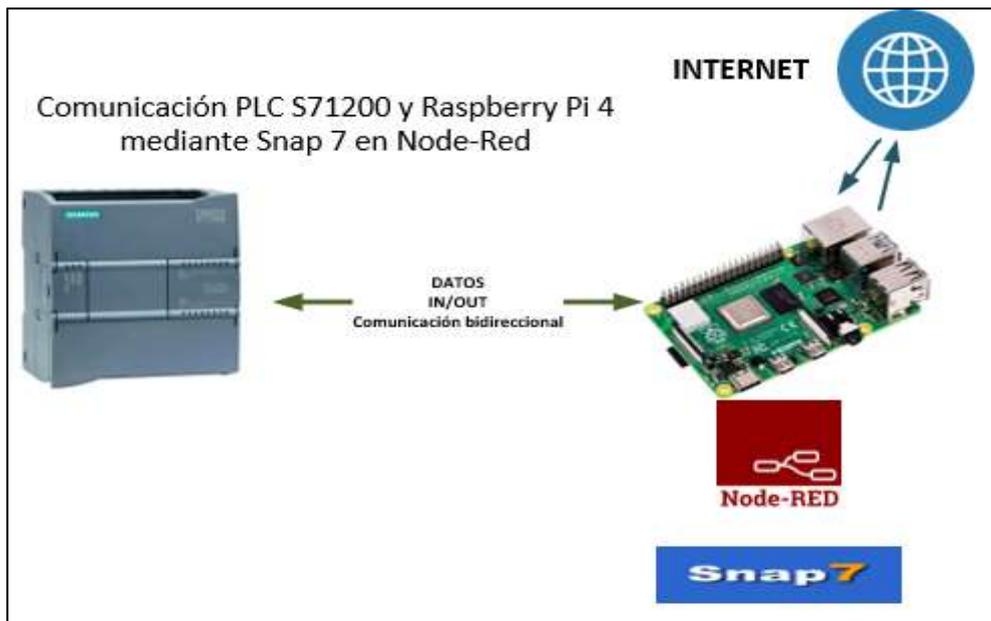


Figura 67. Esquema de comunicación entre PLC S71200 y Raspberry Pi 4

Las señales de los respectivos sensores fotoeléctricos leídas mediante entradas digitales físicas del PLC S7-1200 son guardadas en un bloque de datos en la configuración que se realiza en el software TIA PORTAL con la cual se asigna una dirección a estos valores de dichas señales para que de esta forma estos valores puedan ser leídos en Node-Red y consecutivamente al adquirir estas señales ejecutar activaciones en la tarjeta Raspberry Pi, tales como la activación de los controladores de los motores paso a paso que permiten clasificar la maracuyá de acuerdo a su estado de madurez, en las siguientes figuras se muestran las señales de los sensores fotoeléctricos creadas en un bloque de datos, adicionalmente en la figura se visualizan las respectivas direcciones asignadas a estas variables por el software TIA PORTAL.

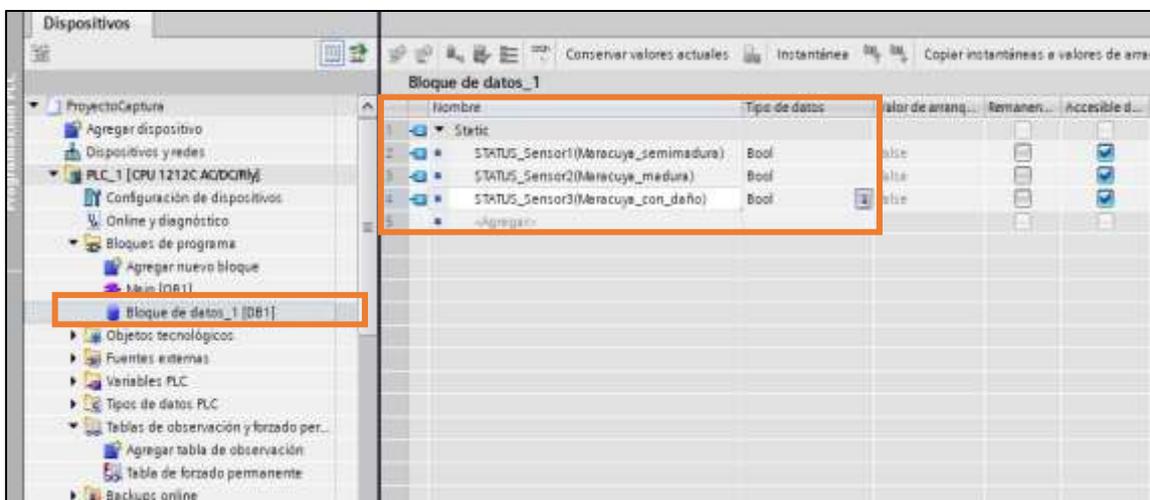


Figura 68. Configuración de las señales de los sensores fotoeléctricos en un bloque de datos.

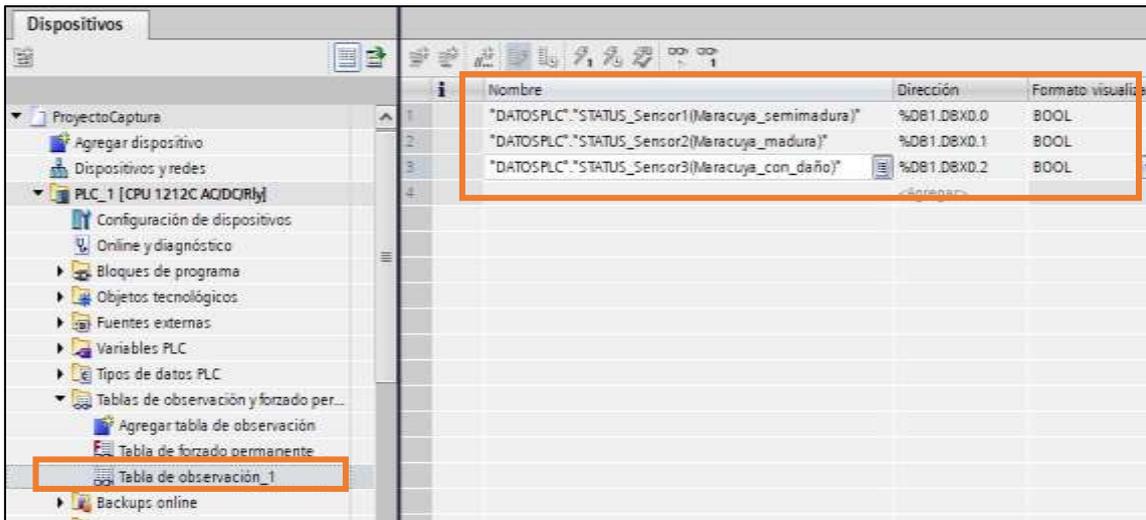


Figura 69. Asignación de direcciones a las señales a enviar mediante el protocolo S7

Una vez configuradas las variables de cada uno de los sensores fotoeléctricos en el software TIA PORTAL, se procede a inicializar la herramienta Node-Red, para lo cual se inicia el terminal de comandos de la tarjeta Raspberry Pi digitando “red-node-star”, el resultado de dicho comando dará como resultado la habilitación de la herramienta, además mostrara el socket TCP con el cual es posible ingresar a Node-Red mediante un navegador web (Ver Figura 70), para de esta manera realizar la respectiva programación que permita comunicar las variables emitidas desde entradas físicas del controlador lógico con la tarjeta Raspberry.

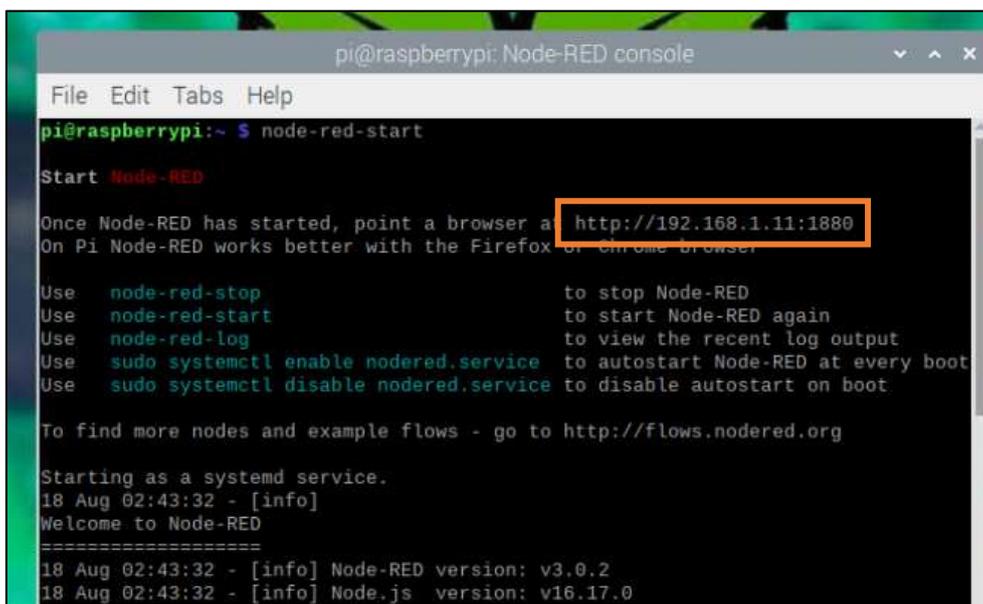


Figura 70. Habilitación de la plataforma Node-Red en la tarjeta Raspberry Pi 4

Luego de habilitar Node-Red en la tarjeta Raspberry Pi 4, se ingresa con el respectivo socket TCP mediante un navegador web a dicha herramienta, en el momento de la apertura de Node-Red se visualizarán varios bloques para el desarrollo de la programación de manera visual. En primer lugar, se selecciona el bloque “S7 in” en el que se configuran los parámetros del tipo de conexión, la respectiva dirección IP del PLC S7-1200, número de slot en el cual se encuentra ubicada la CPU del PLC, el tiempo de peticiones de datos entre Node-Red y el PLC, un tiempo de espera de reconexión en caso de una pérdida de comunicación entre el PLC y la tarjeta Raspberry Pi 4, estos parámetros fueron configurados en los siguientes valores:

- Tipo de conexión: Ethernet TCP
- Dirección IP: 192.168.1.3
- Rack: 0, Slot: 1
- Cycle Time: 2000 ms o 2 segundos
- Timeout: 3000ms o 3 segundos (Ver Figura 71).

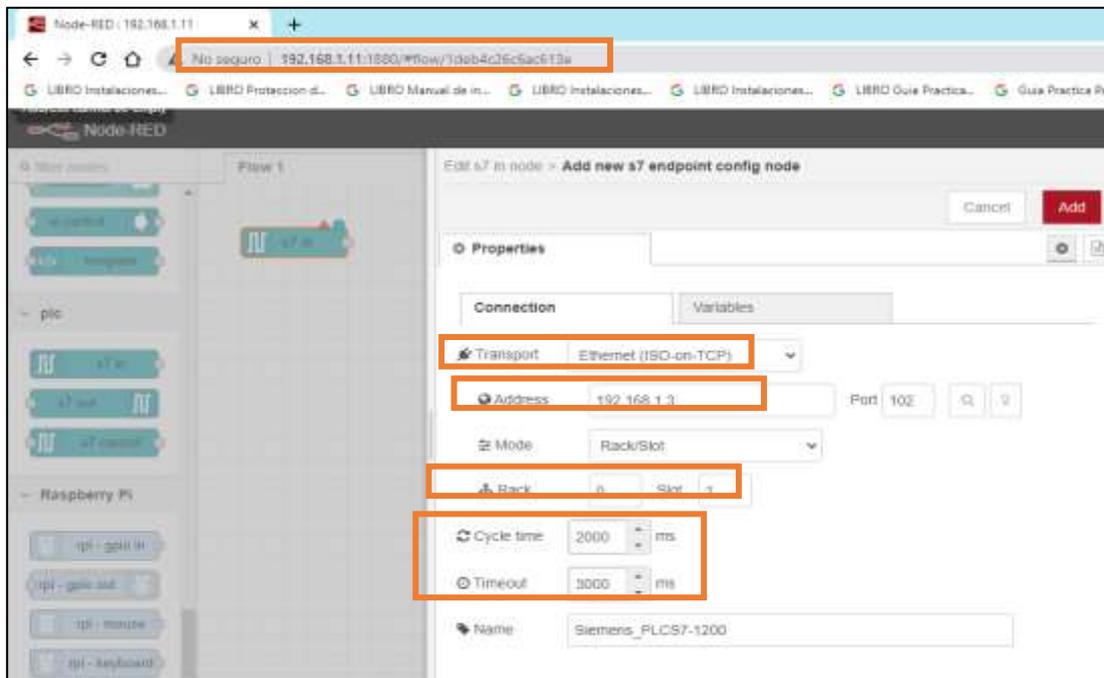


Figura 71. Configuración de la conexión del PLC S7-1200 en Node-Red

Una vez configurada la conexión del PLC en Node-Red, se procede a configurar cada una de las variables provenientes del PLC para que de esta forma se puedan leer y

visualizar en la herramienta Node-Red y por consiguiente activar los pines GPIO de la tarjeta Raspberry Pi que permiten habilitar los controladores de los motores paso a paso del sistema de clasificación, las direcciones a configurar siguen un formato diferente al mostrado en el software TIA PORTAL, a continuación se describe de manera general el formato a configurar en la herramienta:

- DB1,X0.0: Bit 0 del byte 0 correspondiente al bloque de datos DB1.

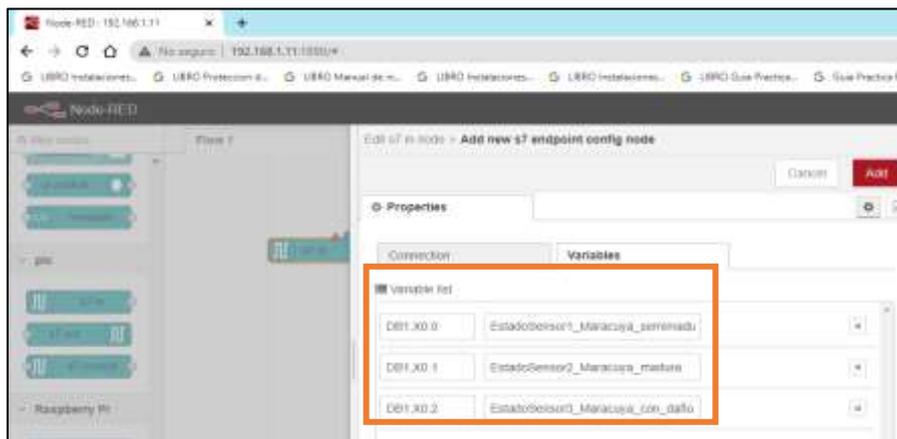


Figura 72. Configuración de variables en Node-Red provenientes del PLC S7-1200.

Tabla 20. Equivalencia de direcciones de las señales en Node-Red y Step 7.

Direcciones de señales en Node Red y TIA PORTAL		
Señal del PLC S7-1200	Dirección Node Red	Step 7 equivalente (TIA PORTAL)
STATUS_Sensor 1 (Maracuyá semi madura)	DB1,X0.0	DB1.DBX0.0
STATUS_Sensor 2 (Maracuyá madura)	DB1,X0.1	DB1.DBX0.1
STATUSSensor 3 (Maracuyá dañada)	DB1,X0.2	DB1.DBX0.2

La señal booleana adquirida mediante Node-Red proveniente de los sensores fotoeléctricos conectados en las respectivas entradas físicas del PLC, se utilizan para

activar el respectivo controlador TB6650 de los motores paso a paso, de acuerdo al valor booleano adquirido en Node-Red es posible enviar el valor de 0 lógico(0V) o 1 lógico(5V) hacia un pin GPIO de la tarjeta Raspberry (Ver Figura 73), adicionalmente en la figura A se muestra la respectiva programación para concatenar dichas señales adquiridas desde el PLC para de esta forma activar los motores a paso que permiten clasificar las maracuyás en un recipiente de acuerdo al estado de madurez que se detecte en el sistema de visión artificial.

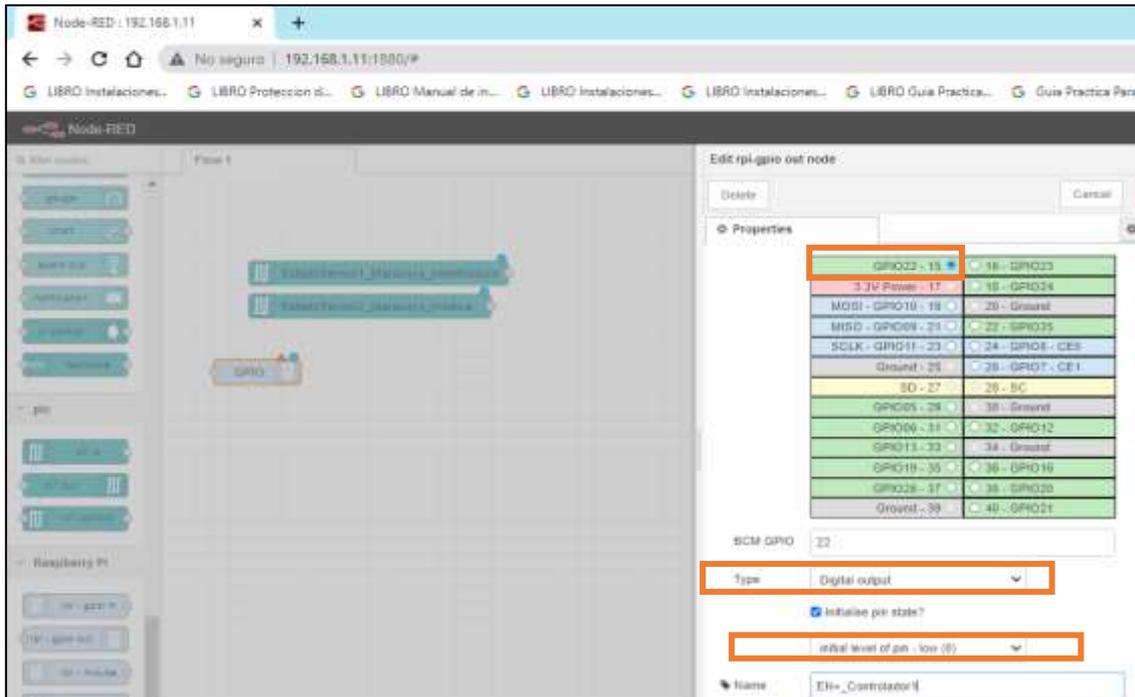


Figura 73. Configuración del Pin GPIO para habilitar el controlador TB6650.

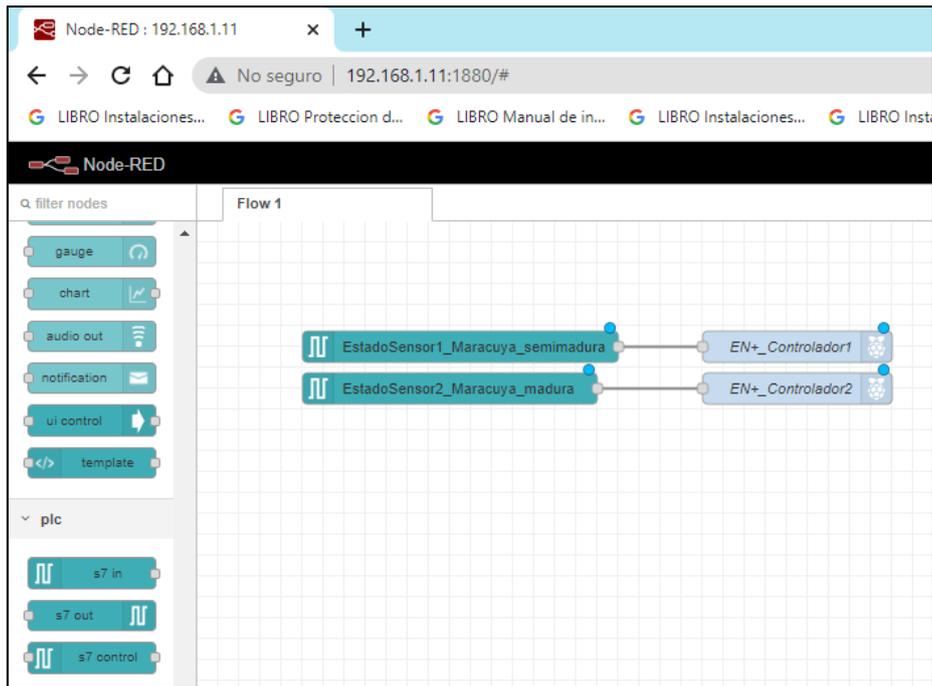


Figura 74. Programación de la conexión para las señales provenientes del PLC y emisión de dicho valor a un pin GPIO.

3.2.11. Diseño y configuración del sistema SCADA en la herramienta Ignition

Para el diseño del sistema SCADA el cual permite monitorear la productividad y la calidad de maracuyá de acuerdo con su estado de madurez, se tiene dos herramientas, la primera es el web server en donde se configura la conexión OPC, y la conexión con el PLC S7-1200, para el diseño de cada una de las interfaces del sistema SCADA se utiliza la herramienta Ignition Designer Launcher utilizando la siguiente arquitectura de comunicaciones (Ver Figura 75).

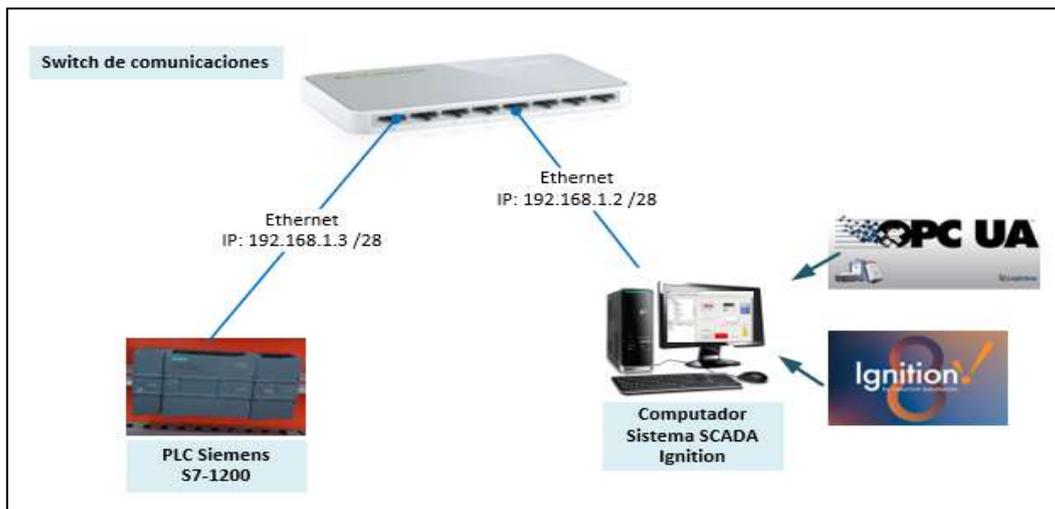


Figura 75. Arquitectura de comunicaciones SCADA-PLC Siemens S7-1200.

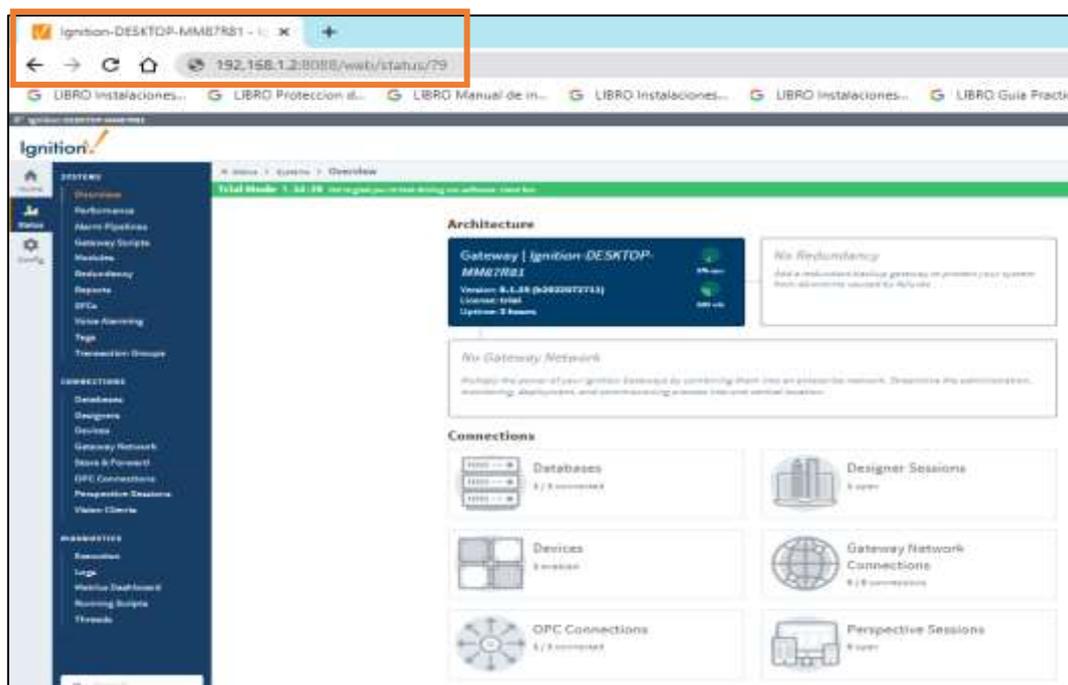


Figura 76. Vista General de la herramienta Ignition.

3.2.11.1. Configuración de la herramienta SCADA Ignition

La herramienta Ignition permite la comunicación con varios fabricantes de PLC tales como Allen-Bradley CompactLogix o MicroLogix, Siemens PLC S7, además posee varios protocolos utilizados en el ámbito de la automatización industrial como, ModBus RTU o TCP el cual puede ser serial o ethernet, BacNET/TCP y DNP3, para el desarrollo de la propuesta se selecciona la conexión con el PLC de Siemens S7-1200 que brinda la herramienta Ignition (Ver Figura 77).

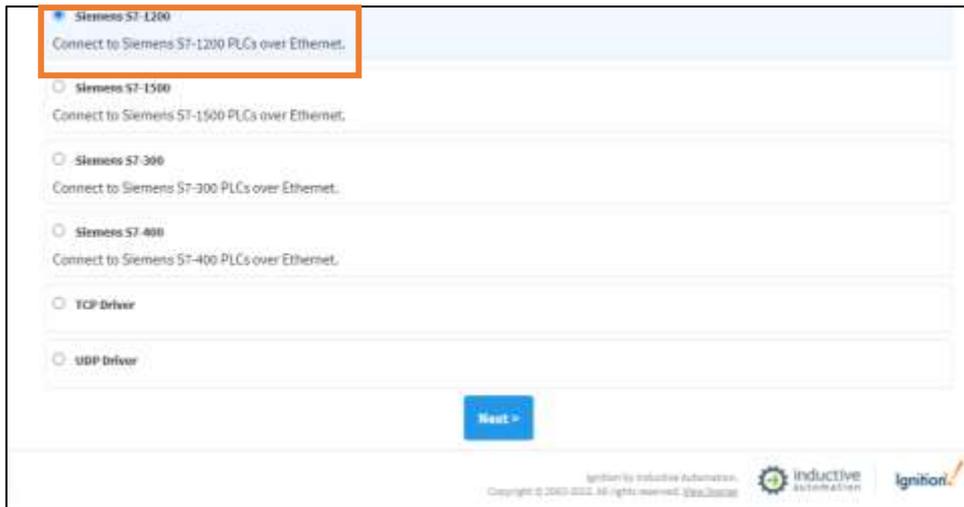


Figura 77. Selección de la conexión con el controlador lógico programable S7-1200.

Una vez seleccionado el dispositivo o controlador lógico programable a utilizar con la herramienta Ignition, se configuran los parámetros del canal de comunicación entre estos dos elementos, como el nombre de la conexión en el que se resalta que dicho nombre se toma en cuenta para direccionar las señales desde el controlador lógico S7-1200 hacia el Ignition, un texto descriptivo, la dirección IP del controlador lógico y el tiempo de espera en milisegundos que deberá esperar el Ignition en caso de no establecer comunicación con el controlador Lógico (Ver Figura 78), adicionalmente se habilita la casilla Enabled para de esta manera habilitar el respectivo canal de comunicación.

General	
Name	PLCS71200
Description	Controlador Lógico Programable S7-1:
Enabled	<input checked="" type="checkbox"/> (default: true)
Connectivity	
Hostname	192.168.1.3
Local Address	Address of network adapter to connect from. (default:)
Timeout	3000 (default: 2,000)
<input type="checkbox"/> Show advanced properties	

Figura 78. Configuración del canal de comunicación SCADA – PLC S7-1200.

Luego de crear el canal de comunicación se realiza el diseño de cada una de las interfaces del sistema SCADA en la herramienta Ignition Designer Launcher, esta herramienta posee componentes tales como cuadros de textos, botones, gráficos los cuales se insertarán en el diseño de cada una de las respectivas ventanas, además es posible configurar cada uno de los tags del servidor OPC en el cual se almacenan cada uno de los datos de las señales del PLC, para que de esta forma el sistema SCADA pueda leer los respectivos y por ende se puedan visualizar los respectivos datos mediante indicadores numéricos o booleanos.

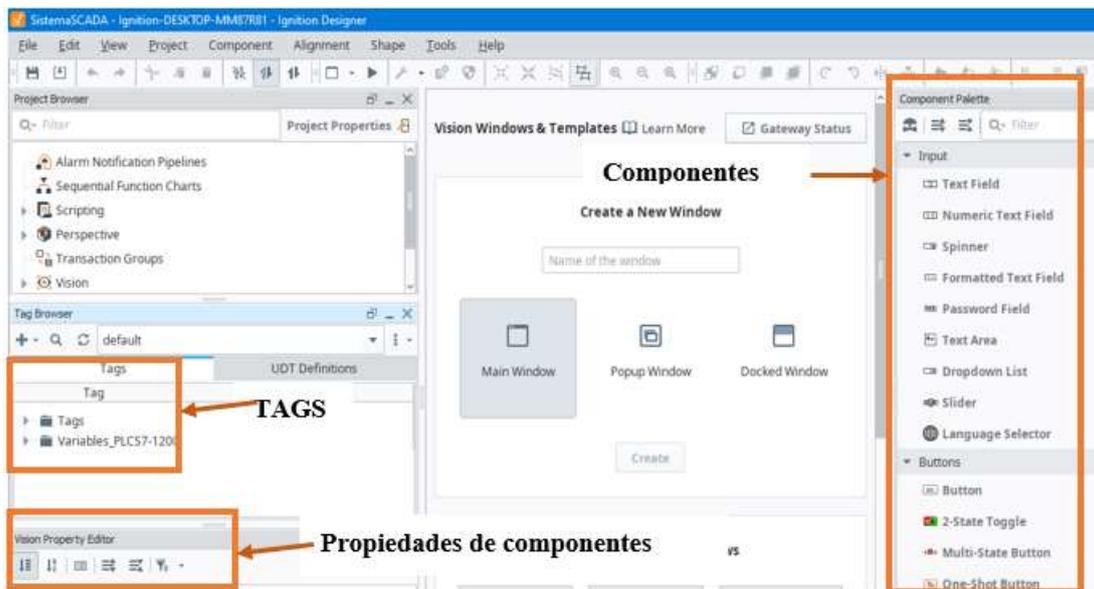


Figura 79. Vista General Herramienta de diseño Ignition Designer Launcher.

Para poder acceder en el sistema SCADA a los respectivos valores de las señales del PLC, se crean todos los tags necesarios para estas señales como se muestra en la tabla AA, en el apartado “Tag Browser” se adiciona cada uno de los respectivos tags siguiendo la respectiva sintaxis:

- Área + Tipo de Dato + offset

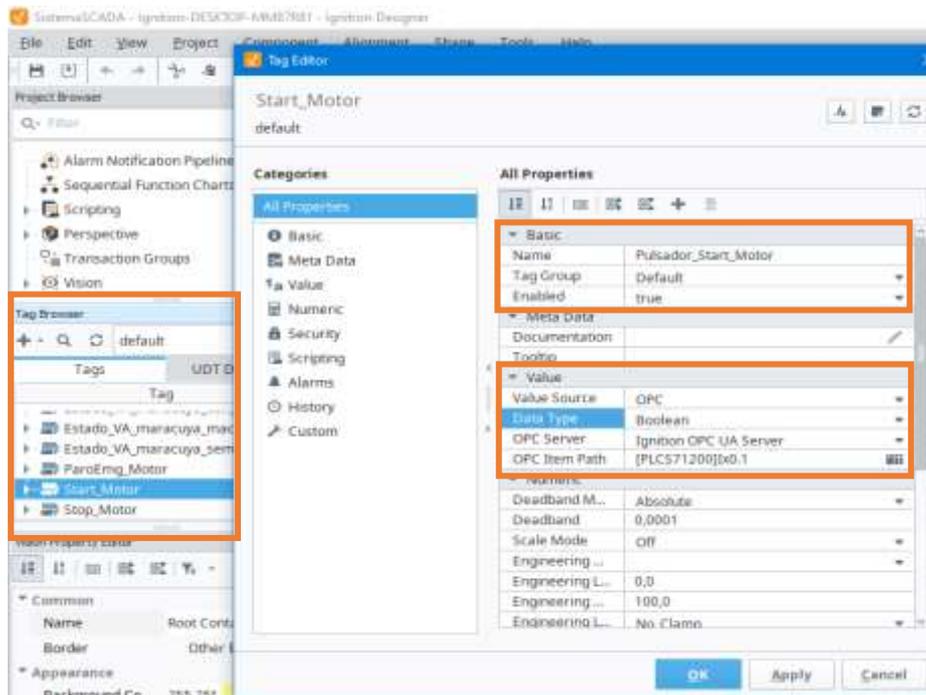


Figura 80. Configuración de tags en la herramienta Ignition Designer Launcher.

En las siguientes tablas se muestra la descripción de la sintaxis para las respectivas direcciones.

Tabla 21. Sintaxis de los respectivos tags en el sistema SCADA Ignition.

Sintaxis área + Tipo de dato			
Área	Sintaxis	Tipo de Dato	Sintaxis
Data Bloks	DBn	Bit	X
Entradas	I	Byte	B
Salidas	Q	Char	C
Memoria	M	Word	W
Temporizadores	T	Int	I
Contadores	C	Dword	DW
		Dint	DI
		Real	REAL
		String	STRING

Posteriormente se muestra la tabla 21, donde se detallan el tipo de dato y la dirección de cada una de las variables del PLC, y como son recibidos estos datos en el sistema SCADA para luego ser visualizadas cada una de las activaciones de las señales

Tabla 22. Señales en el sistema SCADA.

Señales en el sistema SCADA		
Descripción Señal	Dirección SCADA Ignition	Dirección TIA PORTAL
Pulsador Start Motor Banda Transp.	[PLCS71200]Ix0.1	%I0.1
Pulsador Stop Motor Banda Transp.	[PLCS71200]Ix0.2	%I0.2
Pulsador PE Motor Banda Transp.	[PLCS71200]Ix0.3	%I0.3
Sensor 1 (Maracuyá semimadura)	[PLCS71200]Ix0.4	%I0.4
Sensor 2 (Maracuyá madura)	[PLCS71200]Ix0.5	%I0.5
Sensor 3 (Maracuyá con daño)	[PLCS71200]Ix0.6	%I0.6
Indicador Motor Banda Transp. ON	[PLCS71200]Qx0.1	%Q0.1
Indicador Motor Banda Transp. OFF	[PLCS71200]Qx0.2	%Q0.2
Indicador Motor Banda Transp. PE	[PLCS71200]Qx0.3	%Q0.3
Contador Maracuyá semimaduras	[PLCS71200]MW1	%MW1
Contador Maracuyá maduras	[PLCS71200]MW3	%MW3
Contador Maracuyá con daño	[PLCS71200]MW5	%MW5
Indicador Maracuyá semimaduras	[PLCS71200]DB1	%DB1.
Indicador Maracuyá maduras	[PLCS71200]DB1	%DB1.
Indicador Maracuyá con daño	[PLCS71200]DB1	%DB1.

Posteriormente de realizar la configuración de cada uno de los respectivos tags se habilita la opción “Comm Read/Write” la cual permite leer las respectivas variables del PLC hacia el sistema SCADA y además permite el envío de valores digitales hacia el PLC, para este caso esta opción se utilizará para que desde el sistema SCADA se pueda enviar mediante botones comandos de arranque, stop o paro de emergencia de la banda transportadora.

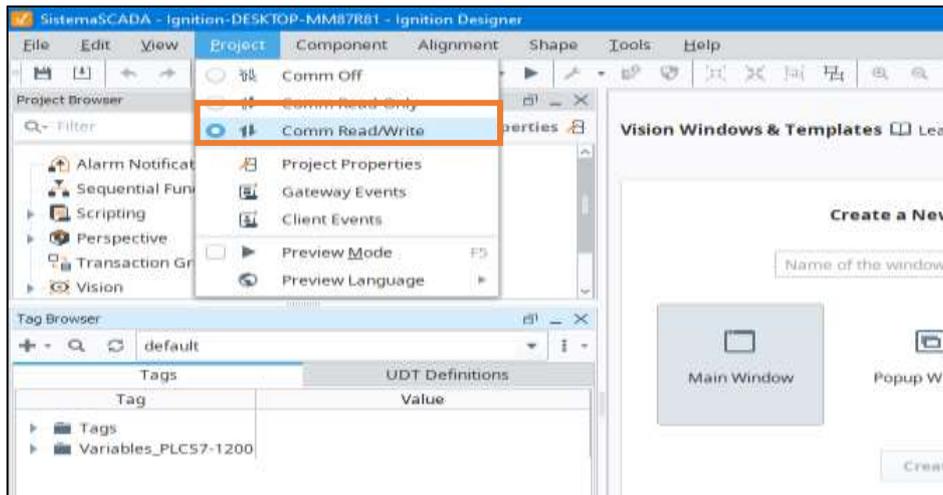


Figura 81. Habilitación de lectura y escritura en el sistema SCADA.

En el sistema SCADA es posible configurar cada uno de los valores de las señales provenientes del PLC como datos históricos, cada valor en cada instante de tiempo es almacenado en una base de datos la cual es creada en el web server de la herramienta Ignition (Ver Figura 83), para las respectivas señales que se requieran almacenar como valores históricos se habilita el campo “History Enabled” y en el campo “Storage Provider” se configura el nombre de la respectiva base de datos en donde se almacenaran los respectivos valores (Ver Figura 82).

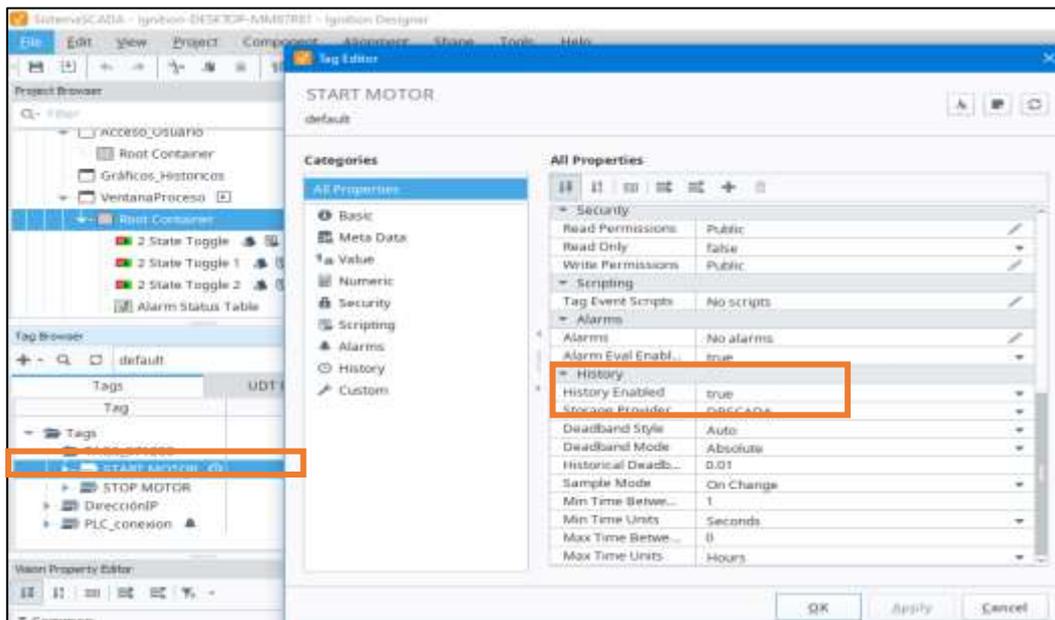


Figura 82. Configuración de señal para guardar datos históricos

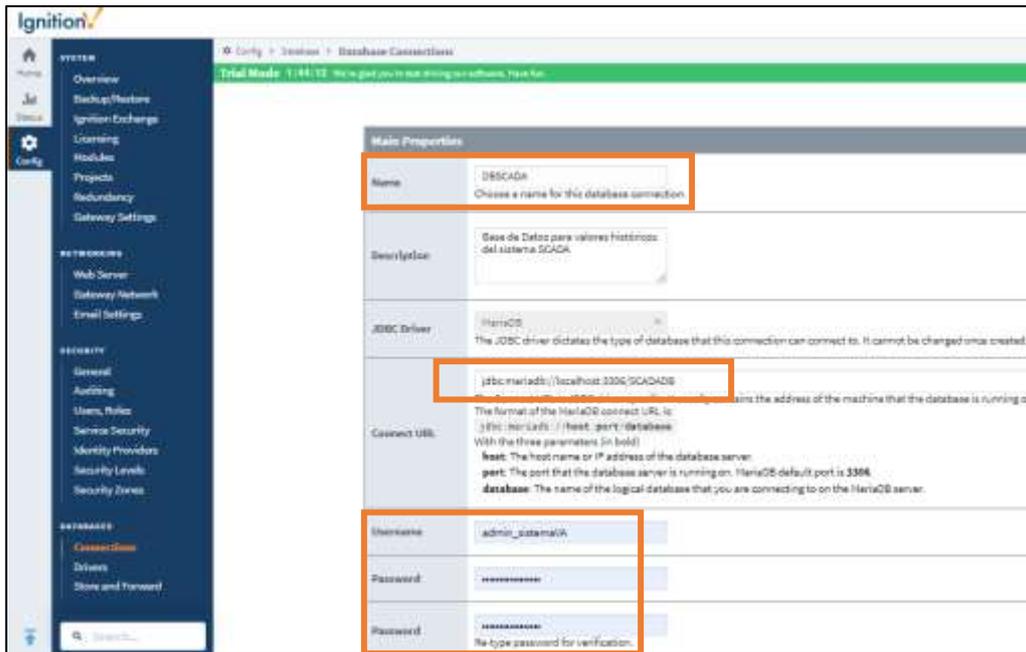


Figura 83. Base de datos creada en el web server de la herramienta Ignition

3.2.11.2. Diseño Interfaz SCADA en la herramienta Ignition

En este apartado se diseñan cada una de las interfaces del sistema SCADA en las cuales se diseñará una interfaz de monitoreo, interfaz de alarmas e interfaz de visualización de datos históricos es decir el valor que toman las variables en cada instante de tiempo, cada una de estas interfaces se crean en la herramienta Ignition Designer Launcher. Principalmente para poder acceder al sistema SCADA el usuario deberá tener un usuario y contraseña (Ver Figura 84), para de esta forma tener un acceso restringido al SCADA, ya que únicamente podrán acceder los usuarios previamente registrados.

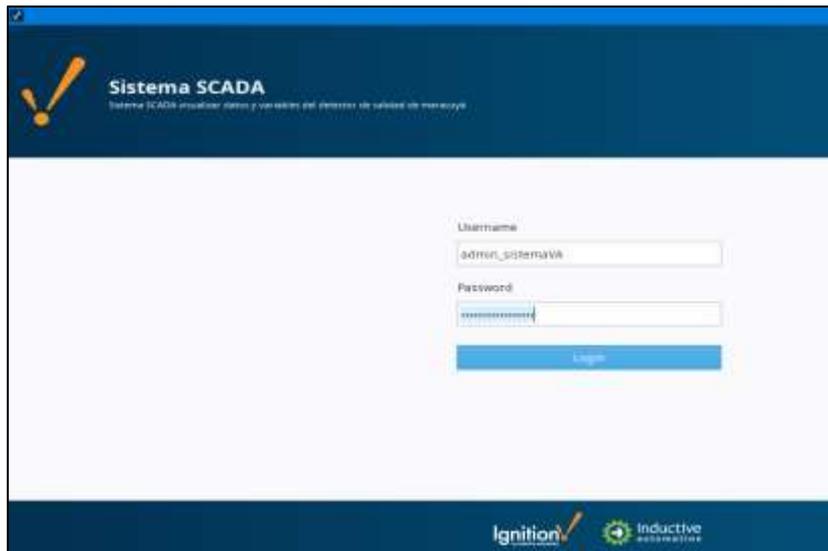


Figura 84. Interfaz de acceso al usuario al sistema SCADA.

Una vez que el usuario accede al sistema SCADA visualizara la interfaz principal del proceso, el cual contiene indicadores booleanos y numéricos, además de botones de control con los cuales el usuario puede ejecutar operaciones de mando desde el sistema SCADA tales como activar, colocar en stop o realizar un paro de emergencia al motor de la banda transportadora. Mediante indicadores booleanos el usuario podrá visualizar el estado de madurez de la respectiva maracuyá, este dato es proveniente desde el sistema de visión artificial, de igual manera se podrá visualizar el estado del motor de la banda transportadora en el cual si el motor se encuentra encendido se visualizara un indicador de color verde, un indicador de color rojo significara que el motor de la banda transportadora se encuentra en stop, mientras que un indicador de color amarillo indicara ocurrió un paro de emergencia. Para monitorear constantemente el estado de la conexión entre el sistema SCADA y el PLC S7-1200 se colocó un indicador de estado, el cual mostrara un color verde con el mensaje “Conectado” si se estableció de manera correcta la comunicación entre estos dos dispositivos, mientras que, si existe un error en el canal o la red de comunicación se visualizará el mismo indicador de color rojo con el mensaje “Desconectado”, en esta ventana principal del monitoreo del proceso se adiciono indicadores numéricos para de esta manera visualizar la producción de maracuyá madura, semi madura y con daños en la superficie de la fruta.

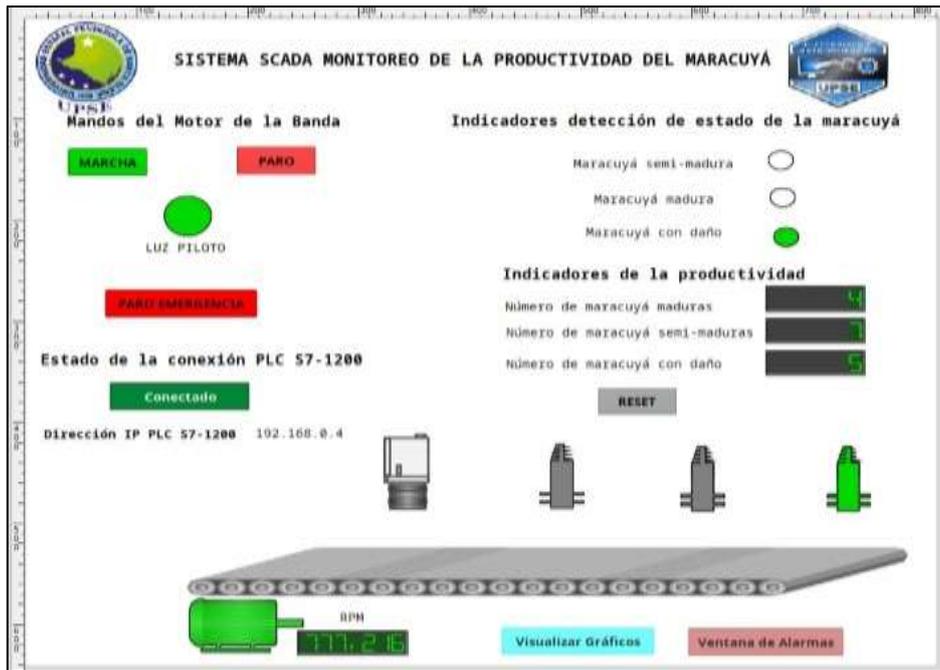


Figura 85. Interfaz del Sistema SCADA para el monitoreo de la productividad de maracuyá.

Con la finalidad de que el usuario pueda tener un control de datos históricos en las variables que indican la producción de la calidad de maracuyá tales como las señales de los contadores e indicadores de la calidad de maracuyá detectada, se crea una interfaz en donde se pueda visualizar de manera grafica la variación de cada uno de los valores de producción en cada instante de tiempo, esto se realiza mediante el componente Easy Chart que se encuentra disponible en la paleta de componentes de la herramienta Ignition Designer Launcher.

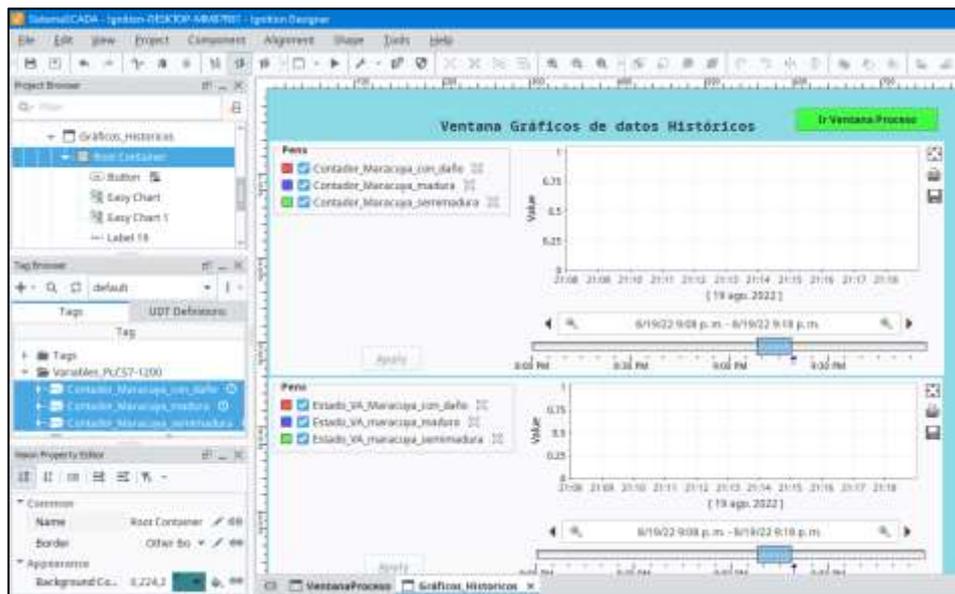
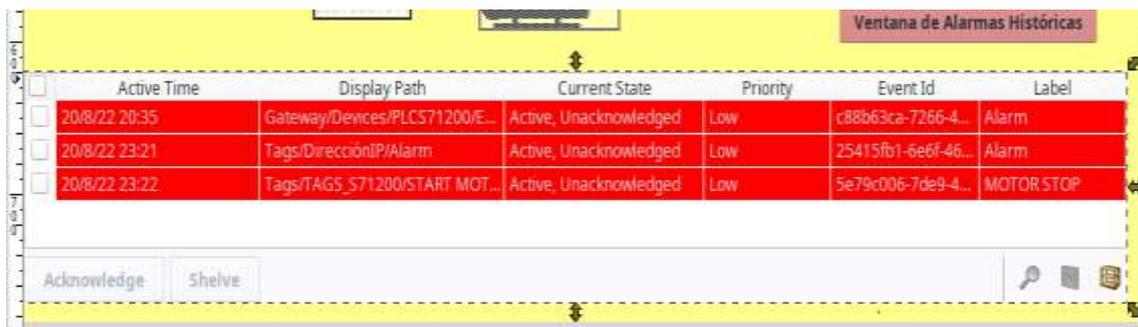


Figura 86. Interfaz de visualización de datos históricos

El sistema SCADA desarrollado para el monitoreo de la productividad de maracuyá posee un panel de alarmas con el objetivo de visualizar una señal de aviso intermitente al instante en que las señales monitoreadas cambien de estado (Ver Figura 87), por ejemplo, que una señal cambie digital cambie de estado, es decir del estado High o Alto cambio al estado Low o Bajo, si se detecta este cambio se emitirá una alarma en el panel, otro caso puede ser la activación del pulsador que coloca en Stop el motor de la banda transportadora, en este caso se visualizará una alarma indicando este evento . Cada alarma o cambio de los valores de las respectivas señales se almacenará en un panel histórico de alarmas el cual se puede acceder desde la interfaz principal del proceso, mediante este panel se podrá supervisar cada uno de los eventos que se susciten en el proceso de clasificación de maracuyá.



Active Time	Display Path	Current State	Priority	Event Id	Label
20/8/22 20:35	Gateway/Devices/PLC571200/E...	Active, Unacknowledged	Low	c88b63ca-7266-4...	Alarm
20/8/22 23:21	Tags/DirecciónIP/Alarm	Active, Unacknowledged	Low	25415fb1-6e6f-46...	Alarm
20/8/22 23:22	Tags/TAGS_S71200/START MOT...	Active, Unacknowledged	Low	5e79c006-7de9-4...	MOTOR STOP

Figura 87. Panel de Alarmas en el sistema SCADA

Event Time	Event Id	Display Path	Event State	Priority	System Ev...	Ack'ed By	Event Value	Current St...	Label
20/8/22 23:22	Se79c00...	Tags/TAGS_57...	Active	Low	False			Active, U...	MOTOR S...
20/8/22 23:21	14e11d6...	Tags/TAGS_57...	Active	Low	False			Active, U...	Alarm
20/8/22 23:21	25415fb1...	Tags/Dirección...	Active	Low	False		192.168.1.4	Active, U...	Alarm
20/8/22 20:35	c88b63ca...	Gateway/Devic...	Active	Low	False		1	Active, U...	Alarm
20/8/22 20:34	3da1723...	evtSystem Star...	Active	Low	True			Active, U...	System S...

Figura 88. Alarmas históricas almacenadas en el sistema SCADA.

3.2.12. Costos de implementación de la propuesta

Para determinar el costo de la propuesta de la implementación de un sistema automatizado, mediante el uso de visión artificial para la clasificación del maracuyá, según su color de madurez y el uso de un sistema scada para el monitoreo de la productividad, se detallan los costos en la tabla 23.

Tabla 23. Valores para la implementación de la propuesta.

CANT.	DESCRIPCIÓN	COSTO
1	MODULO DE SINAMICS G120 Y MOTOR TRIFÁSICO SIEMENS.	\$ 1490,00
1	PLC s7-1200 CPU 1212C AC/DC/RELÉ	\$ 475,00
2	MOTOR PASO A PASO NEMA 23	\$ 80,00
2	DRIVERS BT6560	\$ 52,00
3	SENSORES FOTOELÉCTRICOS	\$ 24,00
1	PROTOTIPO DE LA BANDA TRANSPORTADORA	\$ 400,00
1	PROTOTIPO DEL SISTEMA DE CLASIFICACIÓN	\$ 20,00
1	RASPBERRY PI 4	\$ 200,00
1	CAMARA WEB	\$ 25,00
1	FUENTE CONMUTADA WX-DC2412	\$ 28,00

1	TABLERO DE CONTROL	\$ 80,00
1	CONDUCTOR ELECTRICO	\$ 30,00
1	OTROS GASTOS	\$ 60,00
TOTAL		\$ 2964,00

3.4. PRUEBAS Y RESULTADOS

3.4.1. Pruebas y resultados del sistema de detección de la visión artificial

En esta sección se muestran las diferentes pruebas que se realizaron, en total se realizaron tres pruebas para el sistema de visión artificial, con tres diferentes modelos de entrenamiento de IA. En la siguiente tabla se observa el modelo de entrenamiento con veinte muestras (Ver Tabla 24).

Tabla 24. Pruebas para el modelo de 20 muestra de la visión artificial

MODELO DE IA CON 20 MUESTRAS				
N° de Pruebas	Detección de la IA (%)	Detección Humana	Aciertos	
			Correctas	Incorrectas
1	Maracuyá Madura (0,51)	Pintona		✓
2	Maracuyá Madura (0,64)	Pintona		✓
3	Maracuyá Madura (0,47)	Madura	✓	
4	Maracuyá Madura (0,65)	Madura	✓	
5	Maracuyá Madura (0,68)	Dañada		✓
6	Maracuyá Madura (0,62)	Dañada		✓
7	Maracuyá Madura (0,65)	Dañada		✓
8	Maracuyá Madura (0,63)	Pintona		✓
9	Maracuyá Madura (0,67)	Madura	✓	
10	Maracuyá Madura (0,75)	Pintona		✓

Como se muestra en la tabla 22, se puede observar que todas las detecciones del sistema de visión artificial fueron de maracuyá maduras, no detectaba las semi maduras

ni las dañadas, como se puede observar en la figura 90 se muestra la prueba número 10, que para visión artificial la detecta como madura, mientras que para visión humana es detectada como semi madura.

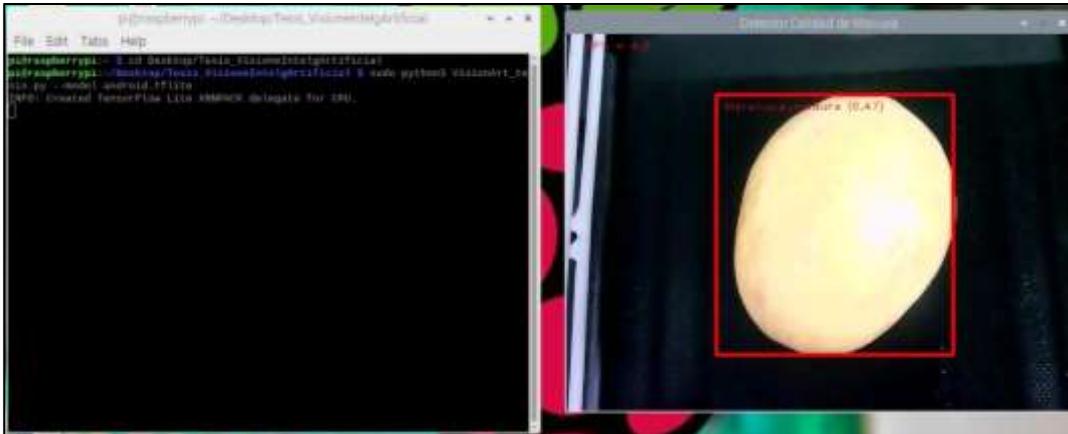


Figura 89. Correcta detección de maracuyá madura para el primer modelo

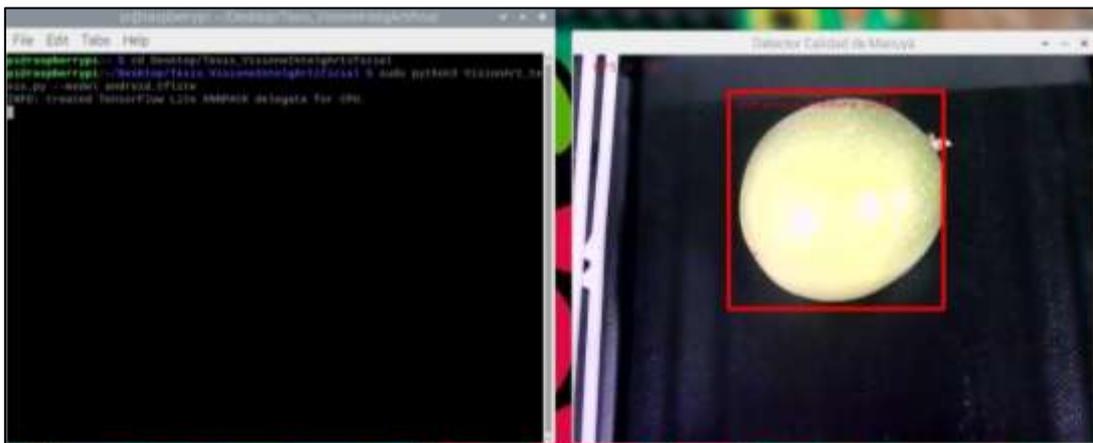


Figura 90. Incorrecta detección de maracuyá madura para el primer modelo

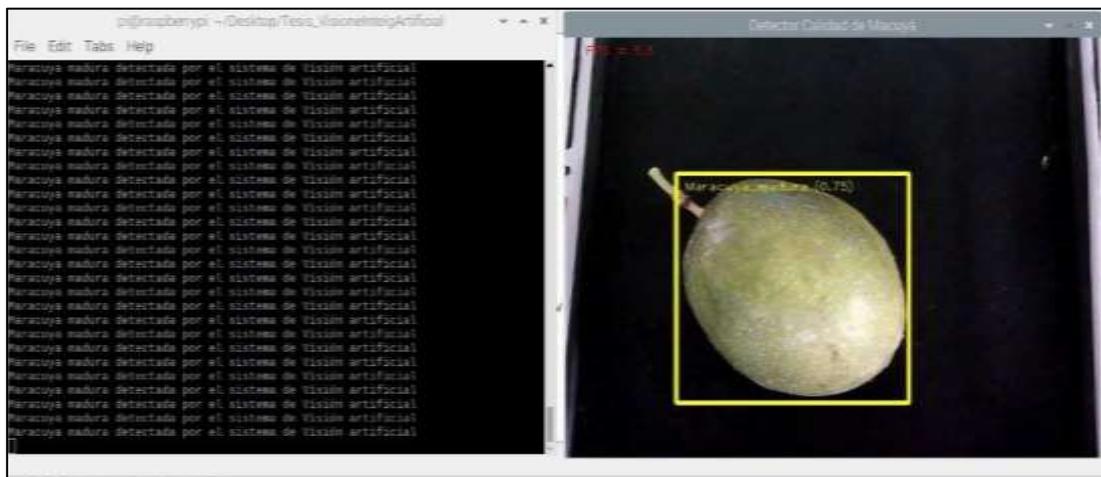


Figura 91. Incorrecta detección de maracuyá madura para el primer modelo

Para las siguientes pruebas de la visión artificial se realiza un modelo IA con un total de cien muestras (Ver Tabla 25).

Tabla 25. Pruebas para el modelo de 100 muestra de la visión artificial.

MODELO DE IA CON 100 MUESTRAS				
N° de Pruebas	Detección de la IA (%)	Detección Humana	Aciertos	
			Correctas	Incorrectas
1	Maracuyá Madura (0,7)	Madura	✓	
2	Maracuyá Madura (0,72)	Dañada		✓
3	Maracuyá Madura (0,71)	Pintona		✓
4	Maracuyá Dañada (0,63)	Dañada	✓	
5	Maracuyá Dañada (0,64)	Dañada	✓	
6	Maracuyá Madura (0,72)	Madura	✓	
7	Maracuyá Madura (0,74)	Pintona		✓
8	Maracuyá Madura (0,77)	Dañada		✓
9	Maracuyá Madura (0,64)	Madura	✓	
10	Maracuyá Madura (0,68)	Madura	✓	

En estas pruebas realizadas para el modelo de cien muestras para la IA, se puede analizar que, los maracuyás maduros son detectas, para las dañadas también se detectan con margen de error, pero las semi maduras siguen sin ser detectadas

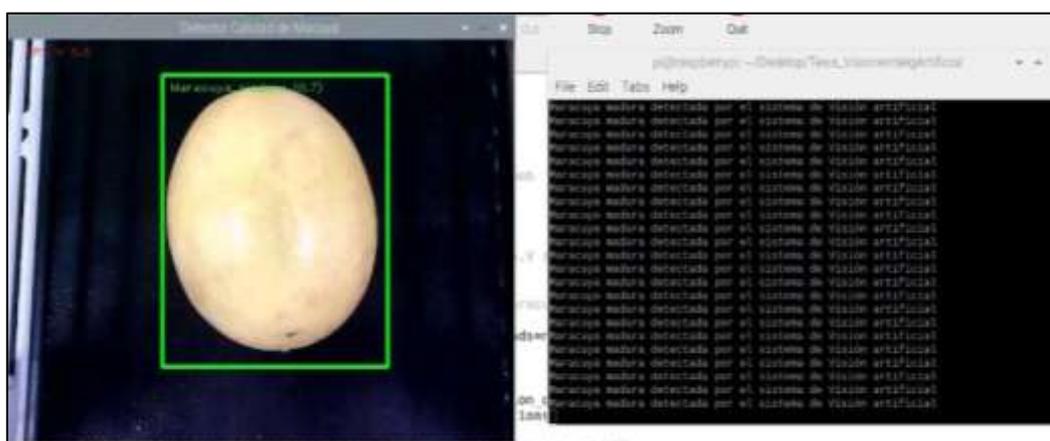


Figura 92. Correcta detección de maracuyá madura para el segundo modelo

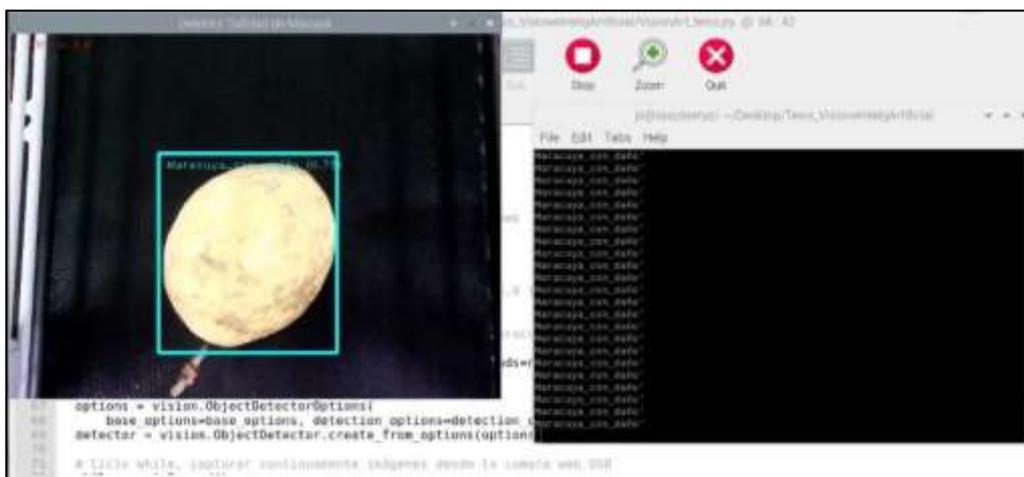


Figura 93. Correcta detección de maracuyá dañada para el segundo modelo



Figura 94. Incorrecta detección de maracuyá madura para el primer modelo

Para las últimas pruebas de la visión artificial se realiza un modelo de IA, con un total de doscientas muestras, logrando que todas las pruebas sean detectadas correctamente (Ver Tabla 26).

Tabla 26. Pruebas para el modelo de 200 muestra de la visión artificial.

MODELO DE IA CON 200 MUESTRAS				
Nº de Pruebas	Detección de la IA (%)	Detección Humana	Aciertos	
			Correctas	Incorrectas

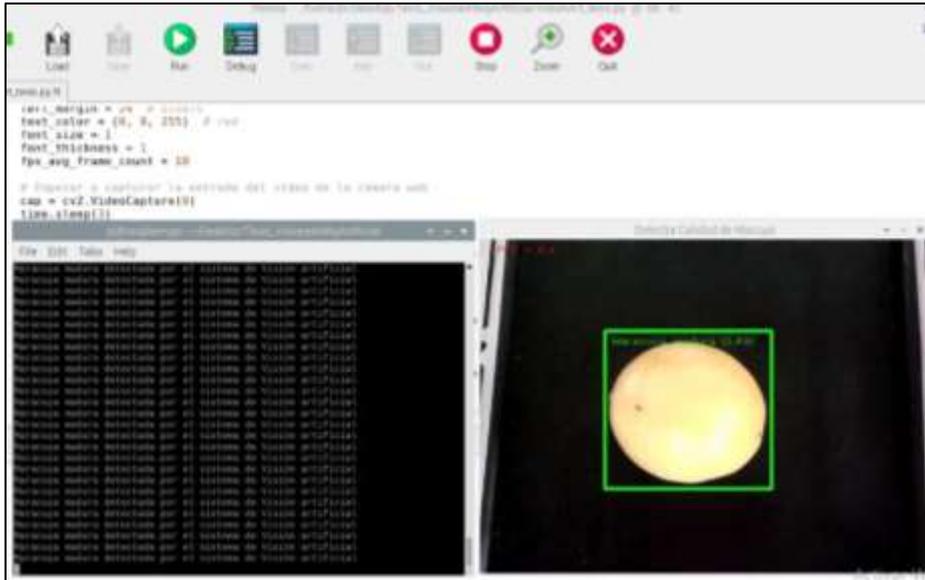


Figura 96. Correcta detección de maracuyá madura para el tercer modelo

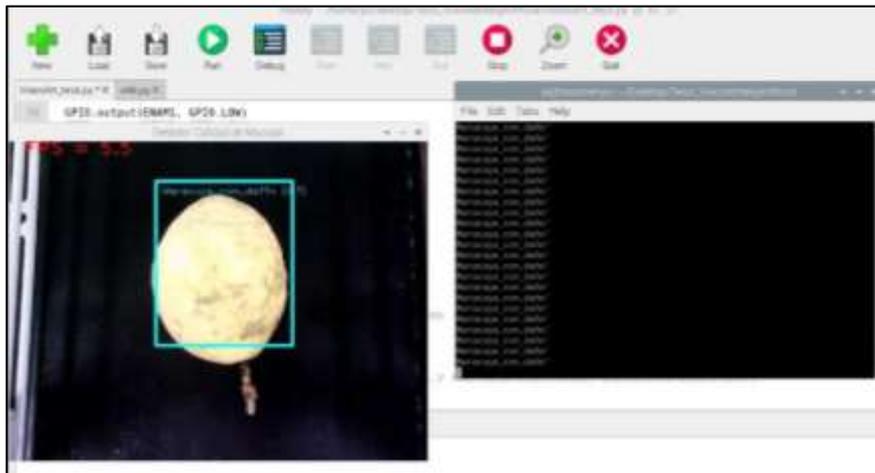


Figura 97. Correcta detección de maracuyá dañada para el tercer modelo

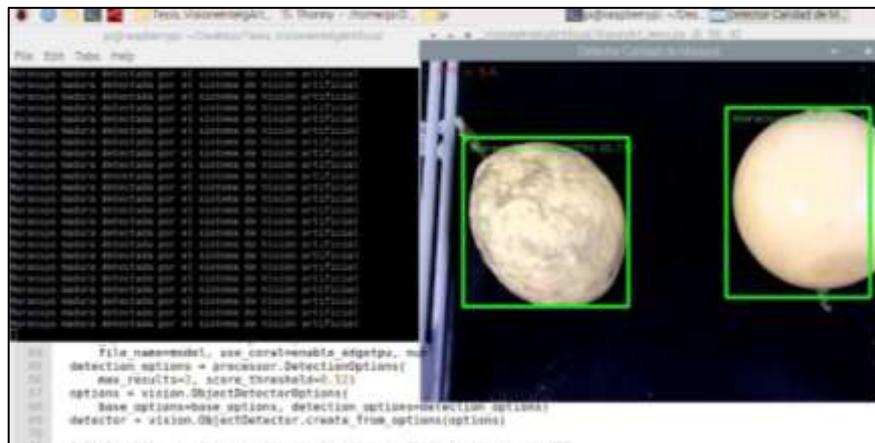


Figura 98. Correcta detección de maracuyá dañada y madura a la misma vez para el tercer modelo

De acuerdo a las pruebas que se realizaron para los distintos modelos de entrenamiento de la visión artificial, se obtuvieron los siguientes resultados que son evidenciados en la figura tal



Figura 99. Eficiencia de los tres modelos de pruebas para la visión artificial

3.4.2. Pruebas y resultados en la red de comunicación

En la tabla 13 mostrada anteriormente se pueden visualizar las diferentes direcciones IP asignadas a cada uno de los equipos o dispositivos que se encuentran conectados. Para comprobar el correcto funcionamiento de la comunicación, se procede a abrir la consola CMD desde el ordenador, que debe estar conectado a la red mediante cable RJ45, con el comando PING y con la dirección IP de los dispositivos, seguido de esto se comprueba los datos enviados y recibidos (Ver Figura 100 a 103).



Figura 104. clasificación del maracuyá pintón



Figura 105. clasificación del maracuyá maduro

Cuando el sistema de visión artificial detecte un maracuyá dañado, el sistema de clasificación la dejará pasar de largo cayendo en un lote donde se almacenarán las frutas con cualquier tipo de deformación en la superficie o tenga un color diferente al programado en el sistema de visión artificial (Ver Figura 106)



Figura 106. clasificación del maracuyá dañado

3.4.4. Pruebas y resultados de la interfaz del sistema SCADA.

En la figura 107 se visualiza el estado inicial del sistema en el cual el motor de la banda se encuentra en estado stop, tal como se muestra en el SCADA el indicador luz piloto de color rojo, además se muestra la programación en TIA portal que los contactos NO se encuentran en su estado inicial, manteniendo des energizada el enclavamiento que activa la señal para encender el motor de la banda

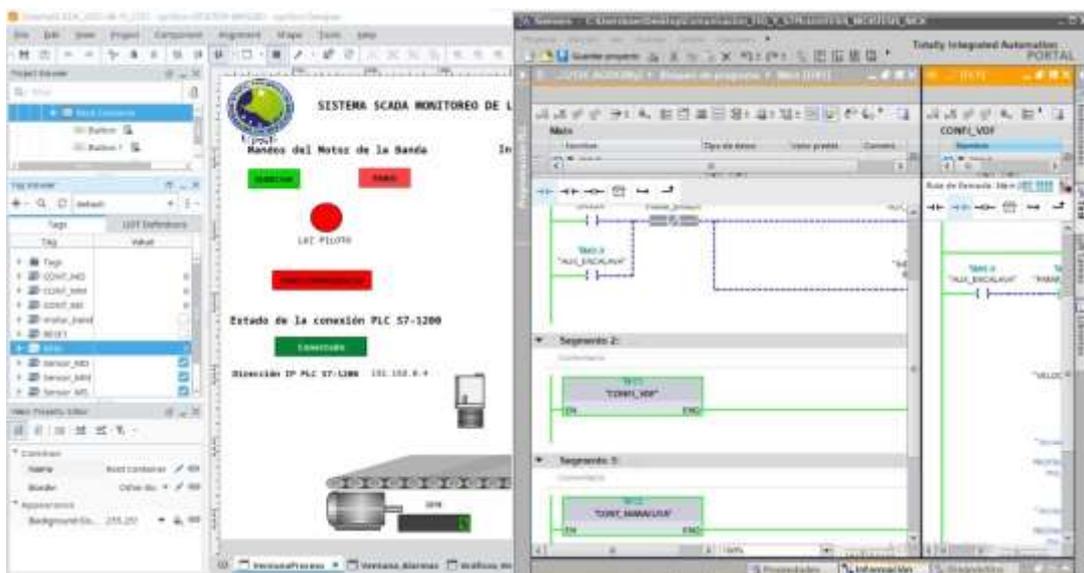


Figura 107. Pruebas del sistema SCADA en su estado inicial

Para comprobar el correcto funcionamiento del sistema SCADA, se da click en el botón de marcha que se encuentra en la interfaz, para arrancar el motor de la banda, luego en el TIA Portal se observa que el contacto NO, perteneciente a la variable de marcha para el motor de la banda, se acciona logrando encender el motor, y la luz indicadora piloto se pone de color verde a igual que el motor (Ver Figura 108).

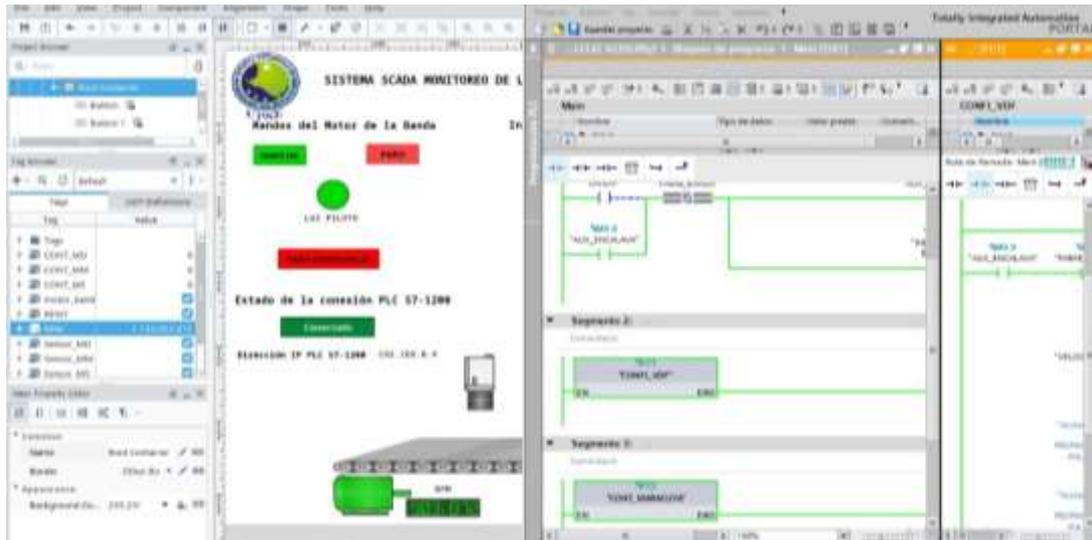


Figura 108. Pruebas del sistema SCADA para el accionamiento del motor

En la interfaz del SCADA también se realizaron pruebas del conteo de las frutas clasificadas según el estado de maduración, como se puede observar en las figuras 109 a la 111, los sensores que se encuentran arriba de la banda, cuando se encuentran inactivos es porque no han detectado las maracuyás, y se visualizan de color gris, al momento que los sensores fotoeléctricos detectan el maracuyá, los sensores cambian a color verde en la interfaz, a la misma vez mediante la programación en TIA Portal, se realiza operaciones de conteo ascendentemente, que también pueden ser visualizadas en el SCADA, adicional a esto el botón de reseteo cuando es pulsado, vuelve a cero el conteo de las pantallas que muestra, el conteo de las maracuyás clasificadas

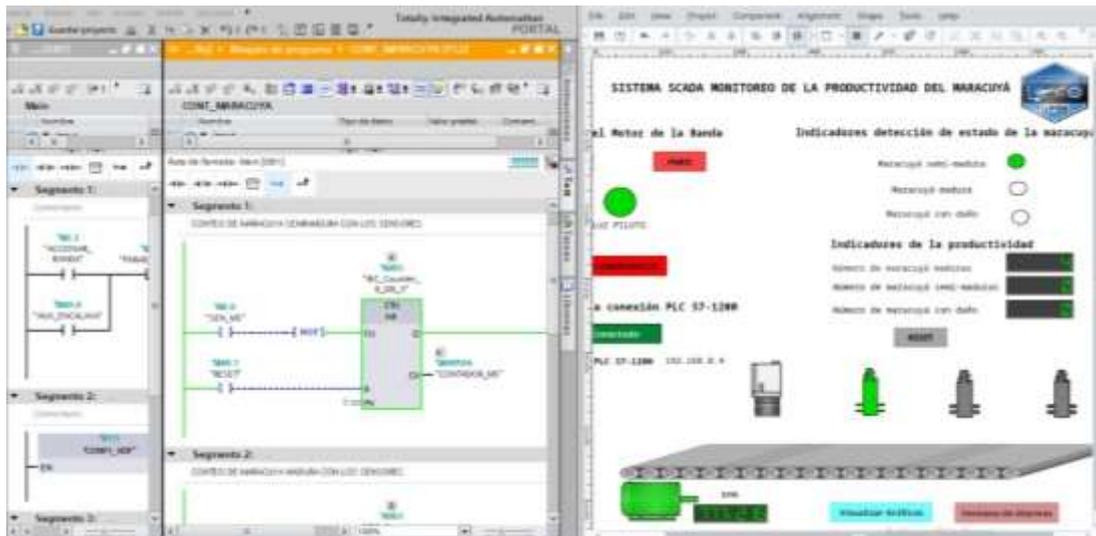


Figura 109. Pruebas del sistema SCADA para los indicadores del sensor para el maracuyá pintona.

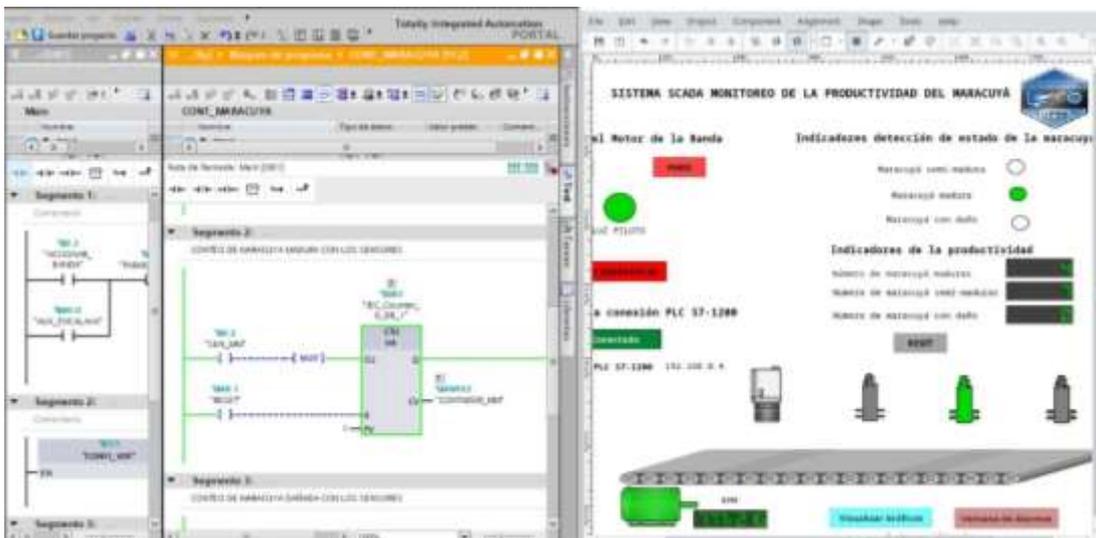


Figura 110. Pruebas del sistema SCADA para los indicadores del sensor para el maracuyá madura.

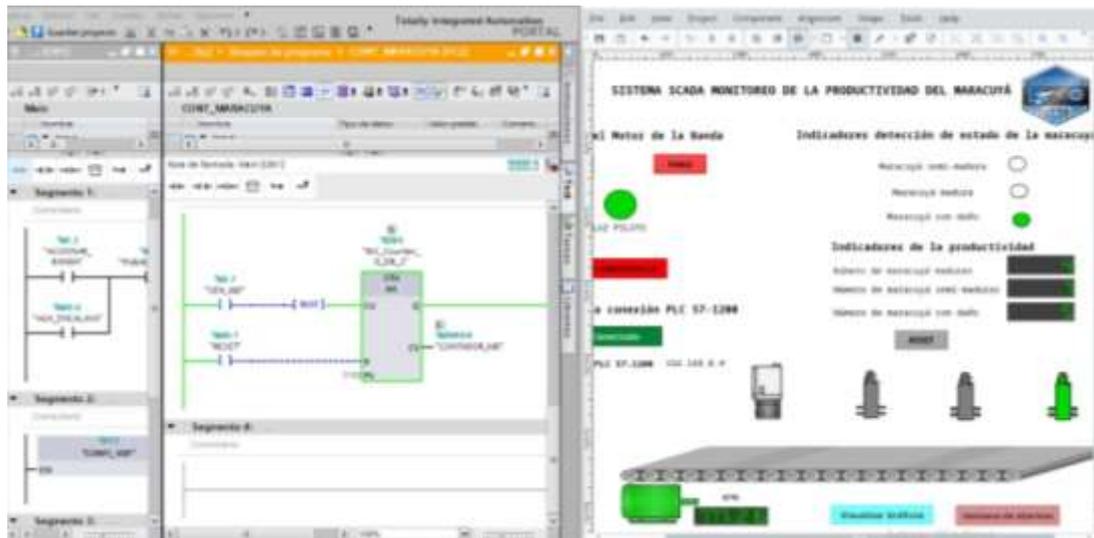


Figura 111. Pruebas del sistema SCADA para los indicadores del sensor para el maracuyá dañada.

CAPÍTULO IV

CONCLUSIONES

En la etapa de visión artificial implementada en la tarjeta Raspberry Pi 4, se desarrolló un algoritmo en Python 3, empleando la biblioteca de OpenCV y el Framework TensorFlow Lite para el reconocimiento en tiempo real según el estado de maduración del maracuyá, estas bibliotecas permitieron adquirir fotogramas en tiempo real y realizar el procesamiento adecuado de la imagen, logrando así implementar una solución de software veloz y escalable, ya que en las bibliotecas utilizadas en la presente propuesta se puede adicionar instrucciones para robustecer el sistema de detección.

Se comprobó que el sistema automatizado, el cuál consta de dos etapas visión artificial y clasificación, puede realizar adecuadamente el trabajo de diferenciar y clasificar el tipo de maracuyá, verdes, maduras y dañadas, obteniendo un 100% de detección para el caso del modelo entrenado con 200 imágenes, y para los modelos de visión artificial entrenados con 20 y 100 imágenes se obtuvo un 30% y 60% de detección, quedando descartados estos modelos por su baja precisión al momento de realizar la detección del maracuyá.

Se implemento un tablero de control eléctrico en el cual se realizó el respectivo diseño y dimensionamiento de los conductores y protecciones eléctricas, tomando en consideración las respectivas corrientes nominales de los diferentes sensores, actuadores, Raspberry Pi 4 y controlador lógico programable, para de esta manera proteger cada uno de estos equipos en caso de que se presente alguna falla eléctrica en el sistema.

En la etapa de clasificación se verificó que el sistema mecánico junto con los motores paso a paso, permitieron clasificar el maracuyá en su respectivo lote de acuerdo a su estado de maduración. Con los motores paso a paso se obtuvo precisión velocidad y torque adecuado para no provocar maltratos o golpes en las frutas durante su funcionamiento, con el fin de mantener la calidad y evitar pérdidas de producción.

Se logro diseñar e implementar un sistema SCADA en la herramienta Ignition, capaz de adquirir y enviar comandos de accionamientos en tiempo real a cada uno de los

actuadores, leer estados digitales y valores analógicos provenientes de los sensores fotoeléctrico, variador de frecuencia, pulsadores, sistema de visión artificial, para el caso de las señales digitales, estas señales reflejaron su valor al instante que la señal cambiaba de alto a bajo o viceversa, encendiendo así indicadores de color verde en la interfaz de monitoreo de procesos del sistema SCADA.

RECOMENDACIONES

Para obtener un porcentaje de confiabilidad mayor al 83% en el sistema de detección de visión artificial, se debe seleccionar una cámara de características mejoradas con respecto a las que se trabajó en esta propuesta, las cuales fueron de calidad VGA, con unas dimensiones de 640X480px, una calidad HD o FHD, ayudaría a aumentar la confiabilidad en el sistema de detección, ya que se tendría mayor tonalidad en los colores, enfoque en los objetos y un aumento en la ventana de detección (1280x720px para HD y 1920x1080px para FHD).

Si se requiere aumentar la robustez del sistema de detección el cual incluye inteligencia artificial, se puede instalar otros framework en la Raspberry Pi, tales como Keras o TensorFlow en su versión normal, adicionalmente a los modelos de entrenamiento se requiere el ingreso de un mayor número de imágenes con los objetos de interés que se requiera detectar

El sistema SCADA diseñado en la herramienta Ignition cuenta con un periodo de licencia gratuita de un tiempo limitado de 2 horas, al cumplirse este tiempo, el sistema se bloquea, impidiendo el monitoreo, esto se resuelve ingresando las credenciales de acceso nuevamente, esto en un ámbito industrial real, genera inconformidad con lo cual se sugiere que si se desea operara y monitorear fluidamente un proceso, se debe consultar y adquirir una licencia del software, con la empresa desarrolladora Inductive Automation.

Al momento de utilizar la tarjeta Raspberry Pi 4 en conjunto con el sistema visión artificial, dicha tarjeta consume la mayor parte de los recursos de memoria RAM lo que provoca que la CPU se sobre caliente, es por esto que se recomienda utilizar un sistema de refrigeración el cual puede ser un ventilador con motores brushless, los cuales tienen un bajo consumo de amperaje alrededor de los 120 mA y una alta ventilación, otra alternativa es hacer uso de disipadores de calor conectados en la CPU .

Si se desea replicar el sistema de clasificación desarrollado en esta propuesta, el material con el cual se le adapte a los motores paso a paso deberá ser ligero, y que no contenga partes que pueda afectar la superficie de la fruta como el metal, o de ser así preferible que estén forradas por algún tipo de material que amortigüe el golpe a la fruta, como el caucho.

BIBLIOGRAFÍA

- [FAO, «FAO,» 28 04 2015. [En línea]. Available: [https://www.fao.org/in-](https://www.fao.org/in-1 action/agronoticias/detail/es/c/515873/)
1 [action/agronoticias/detail/es/c/515873/](https://www.fao.org/in-action/agronoticias/detail/es/c/515873/).
]
- [H. R. S. Claudia Russo, «Universidad Nacional del Noroeste de la Provincia de
2 Buenos Aires (UNNOBA),» 2018. [En línea]. Available:
] [https://digital.cic.gba.gob.ar/bitstream/handle/11746/8636/11746_8636.pdf-](https://digital.cic.gba.gob.ar/bitstream/handle/11746/8636/11746_8636.pdf-PDFA.pdf?sequence=1&isAllowed=y)
PDFA.pdf?sequence=1&isAllowed=y.
- [S. J., G. A. y C. A, «Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y
3 Electrónica,» 01 01 2014. [En línea]. Available:
] <file:///C:/Users/nickb/Downloads/1659.pdf>.
- [INEC, «INEC,» 08 2011. [En línea]. Available:
4 [https://www.ecuadorencifras.gob.ec/wp-](https://www.ecuadorencifras.gob.ec/wp-content/descargas/Presentaciones/espac_2010.pdf)
] [content/descargas/Presentaciones/espac_2010.pdf](https://www.ecuadorencifras.gob.ec/wp-content/descargas/Presentaciones/espac_2010.pdf).
- [REVISTA LÍDERES, «REVISTA LÍDERES,» 21 06 2015. [En línea]. Available:
5 <https://www.revistalideres.ec/lideres/cultivo-maracuya-produccion-ecuador.html>.
]
- [ROLDÁN, COMO HACER UN PROYECTO DE INVESTIGACIÓN, Universidad
6 de Alicante.
]
- [J. ORTIZ, «lifeder.com,» 2017. [En línea]. Available:
7 <https://www.lifeder.com/investigacion-exploratoria/>.
]
- [MINISTERIO DE AGRICULTURA, 12 2020. [En línea]. Available:
8 <https://www.agricultura.gob.ec/exportaciones-impulsan-a-ecuador-como-potencia->
] [agropecuaria/](https://www.agricultura.gob.ec/exportaciones-impulsan-a-ecuador-como-potencia-agropecuaria/).
- [DIARIO EL COMERCIO, «EL COMERCIO,» 09 2018. [En línea]. Available:
9 <https://www.elcomercio.com/actualidad/ecuador/ecuador-santaelena-granero->
] [agricola-produccion.html](https://www.elcomercio.com/actualidad/ecuador/ecuador-santaelena-granero-agricola-produccion.html).
- [L. VANGUARDIA, «LA VANGUARDIA,» 08 04 2021. [En línea]. Available:
1 [https://www.lavanguardia.com/comer/materia-](https://www.lavanguardia.com/comer/materia-0-prima/20180711/45803718261/maracuya-fruta-de-la-pasion-propiedades-valor-)
0 [prima/20180711/45803718261/maracuya-fruta-de-la-pasion-propiedades-valor-](https://www.lavanguardia.com/comer/materia-0-prima/20180711/45803718261/maracuya-fruta-de-la-pasion-propiedades-valor-nutricional-beneficios.html)
] [nutricional-beneficios.html](https://www.lavanguardia.com/comer/materia-0-prima/20180711/45803718261/maracuya-fruta-de-la-pasion-propiedades-valor-nutricional-beneficios.html).
- [INEN, «DOCPLAYER,» 2012. [En línea]. Available:
1 <https://www.normalizacion.gob.ec/buzon/normas/1971.pdf>.

1

]

[MECALUX, 24 12 2021. [En línea]. Available:
1 <https://www.mecalux.es/blog/sistemas-automatizados-ejemplos>.

2

]

[I. GSL, «Industrias GSL,» 2022. [En línea]. Available:
1 <https://industriasgsl.com/blogs/automatizacion/que-es-un-plc-y-como-funciona>.

3

]

[J. M. J. C. S.-T. E. F. A. S. José Carlos Quezada, «Diagrama escalera red de Petri
1 (LDPN),» vol. 4, nº 7, 2016.

4

]

[A. G. B. L. M. L. C. Á. Vicente Díaz. [En línea]. Available:
1 [http://ocw.uc3m.es/cursos-archivados/ingenieria-de-](http://ocw.uc3m.es/cursos-archivados/ingenieria-de-transportes/pract_1_cinta_transportadora.pdf)
5 [transportes/pract_1_cinta_transportadora.pdf](http://ocw.uc3m.es/cursos-archivados/ingenieria-de-transportes/pract_1_cinta_transportadora.pdf).

]

[MECATRONICALATAM, «MECATRONICALATAM,» 05 05 2021. [En línea].
1 Available: [https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-](https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-proximidad/)
6 [proximidad/](https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-proximidad/).

]

[TiMOTION, «TiMOTION,» 10 Agosto 2021. [En línea]. Available:
1 [https://www.timotion.com/es/news-and-articles/part-1-what-is-an-electric-linear-](https://www.timotion.com/es/news-and-articles/part-1-what-is-an-electric-linear-actuator-and-how-to-choose-it#:~:text=Un%20actuador%20lineal%20el%C3%A9ctrico%20es,un%20motor%20en%20movimiento%20lineal.&text=Por%20ejemplo%2C%20es%20posible%20obtener,la%20i)
7 [actuator-and-how-to-choose-](https://www.timotion.com/es/news-and-articles/part-1-what-is-an-electric-linear-actuator-and-how-to-choose-it#:~:text=Un%20actuador%20lineal%20el%C3%A9ctrico%20es,un%20motor%20en%20movimiento%20lineal.&text=Por%20ejemplo%2C%20es%20posible%20obtener,la%20i)
] [it#:~:text=Un%20actuador%20lineal%20el%C3%A9ctrico%20es,un%20motor%20en%20movimiento%20lineal.&text=Por%20ejemplo%2C%20es%20posible%20obtener,la%20i](https://www.timotion.com/es/news-and-articles/part-1-what-is-an-electric-linear-actuator-and-how-to-choose-it#:~:text=Un%20actuador%20lineal%20el%C3%A9ctrico%20es,un%20motor%20en%20movimiento%20lineal.&text=Por%20ejemplo%2C%20es%20posible%20obtener,la%20i).

[S. S.A., «SICA S.A.,» 2017. [En línea]. Available:
1 [https://sica.es/es/productos/sistemas-de-control-local/subestaciones-gis-distribucion-](https://sica.es/es/productos/sistemas-de-control-local/subestaciones-gis-distribucion-traccion/scada#:~:text=SCADA%20permite%20la%20supervisi%C3%B3n%20y,realizadas%20dentro%20de%20las%20subestaciones.&text=La%20aplicaci%C3%B3n%20permite%20la%20generaci%C3)
8 [traccion/scada#:~:text=SCADA%20permite%20la%20supervisi%C3%B3n%20y,rea-](https://sica.es/es/productos/sistemas-de-control-local/subestaciones-gis-distribucion-traccion/scada#:~:text=SCADA%20permite%20la%20supervisi%C3%B3n%20y,realizadas%20dentro%20de%20las%20subestaciones.&text=La%20aplicaci%C3%B3n%20permite%20la%20generaci%C3)
] [lizadas%20dentro%20de%20las%20subestaciones.&text=La%20aplicaci%C3%B3n%20permite%20la%20generaci%C3](https://sica.es/es/productos/sistemas-de-control-local/subestaciones-gis-distribucion-traccion/scada#:~:text=SCADA%20permite%20la%20supervisi%C3%B3n%20y,realizadas%20dentro%20de%20las%20subestaciones.&text=La%20aplicaci%C3%B3n%20permite%20la%20generaci%C3).

[A. B. M. D. Á. S. C. J. L. E. S.-M. José Francisco Vélez Serrano, VISIÓN POR
1 COMPUTADORA, 2016.

9

]

[INFAIMON, «INFAIMON,» 26 febrero 2018. [En línea]. Available:
2 <https://blog.infaimon.com/sistemas-de-iluminacion/>.

0

]

[DACHS-SOLUTIONS, «DACHS-SOLUTIONS,» [En línea]. Available:
2 <https://dachssolutions.com/vision-artificial/tipos-de-iluminacion/>.

1

]

[INFAIMON, 2 junio 2020. [En línea]. Available: [https://blog.infaimon.com/tecnicas-](https://blog.infaimon.com/tecnicas-de-iluminacion-sector-industrial/)
2 [de-iluminacion-sector-industrial/](https://blog.infaimon.com/tecnicas-de-iluminacion-sector-industrial/).

2

]

[A. J. A. GONZABAY, "Diseño e implementación de un sistema de gestión, monitoreo
2 y digitalización para el proceso de selección de mangos en función de su color,
3 enfocado a la industria 4.0 para optimizar el rendimiento de producción", La libertad:
] Univesidad estatal península de santa elena, 2019.

[V. C. C. E. Molina Llunitasig Luis Miguel, "Estudio e implementación de un sistema
2 de control de calidad para la detección de contaminantes superficiales de diferentes
4 tipos de frutas usando visión artificial", Latacaunga: Universidad de las fuerzas
] armadas, 2019.

[J. I. S. P. Johana Beatriz Rodríguez Bosquez, "Diseño, construcción y automatización
2 de un prototipo de máquina clasificadora de piñas de acuerdo al color y forma por
5 medio de visión artificial", Riobamba: Escuela superior politécnica de Chimborazo,
] 2020.

[SIEMENS, «SIEMENS,» [En línea]. Available:
2 [https://new.siemens.com/mx/es/productos/automatizacion/systems/industrial/plc/s7-](https://new.siemens.com/mx/es/productos/automatizacion/systems/industrial/plc/s7-61200.html)
6 [1200.html](https://new.siemens.com/mx/es/productos/automatizacion/systems/industrial/plc/s7-61200.html). [Último acceso: 22 06 2022].

]

[<https://www.raspberrypi.org/>, «Raspberry Pi Foundation,» 06 07 2022. [En línea].
2 Available: <https://raspberrypi.cl/que-es-raspberry/>.

7

]

[S. J. Chapman, MÁQUINAS ELÉCTRICAS, México: The McGraw-Hill, 2012, p. 1.

2

8

]

[SIEMENS, «SIEMENS,» [En línea]. Available:
2 [https://assets.new.siemens.com/siemens/assets/api/uuid:782b04d0-82f4-488a-936b-](https://assets.new.siemens.com/siemens/assets/api/uuid:782b04d0-82f4-488a-936b-9514cd8bb5d5f/version:1580149154/arteimpresioncatalogosimotics.pdf)
9 [514cd8bb5d5f/version:1580149154/arteimpresioncatalogosimotics.pdf](https://assets.new.siemens.com/siemens/assets/api/uuid:782b04d0-82f4-488a-936b-9514cd8bb5d5f/version:1580149154/arteimpresioncatalogosimotics.pdf). [Último
] acceso: 13 07 2022].

[E. Components, «Electronic Components,» 08 09 2020. [En línea]. Available:
3 [https://www.tme.eu/es/news/library-articles/page/41861/Motor-paso-a-paso-tipos-y-](https://www.tme.eu/es/news/library-articles/page/41861/Motor-paso-a-paso-tipos-y-0)
0 [ejemplos-del-uso-de-motores-paso-a-paso/](https://www.tme.eu/es/news/library-articles/page/41861/Motor-paso-a-paso-tipos-y-0). [Último acceso: 20 07 2022].
]

[SIEMENS, «SIEMENS,» [En línea]. Available:
3 [https://siemensmexico.com.mx/productos/variadores-de-velocidad/sinamics-](https://siemensmexico.com.mx/productos/variadores-de-velocidad/sinamics-1)
1 [g120.html#:~:text=SINAMICS%20G120%20es%20el%20variador,general%20com](https://siemensmexico.com.mx/productos/variadores-de-velocidad/sinamics-1)
] [o%2C%20por%20ejemplo%2C%20sistemas](https://siemensmexico.com.mx/productos/variadores-de-velocidad/sinamics-1). [Último acceso: 14 07 2022].

[R. Pi, «Raspberry Pi,» 12 07 2022. [En línea]. Available:
3 <https://www.raspberrypi.com/software/>.
2
]

[A. I. Scanner, «Advanced IP Scanner,» 12 07 2022. [En línea]. Available:
3 <https://www.advanced-ip-scanner.com/es/>.
3
]

[filehorse, «filehorse,» 01 07 2022. [En línea]. Available:
3 <https://www.filehorse.com/es/descargar-vnc-viewer/>. [Último acceso: 12 07 2022].
4
]

[J. S. N. Valenzuela, Python: Aplicaciones prácticas, Madrid: RA-MA Editorial, 2018,
3 p. 14.
5
]

[INIAP, «INIAP,» 2014. [En línea]. Available:
3 <http://tecnologia.iniap.gob.ec/index.php/explore-2/mfruti/rmaracuya>.
6
]

[K. CORPORATION, «KEYENCE CORPORATION,» 2022. [En línea]. Available:
3 <https://www.keyence.com.mx/ss/products/sensor/sensorbasics/ultrasonic/info/>.
7
]

[Fidemar, «Fidemar,» 2019. [En línea]. Available:
3 <https://www.fidemar.com.uy/producto/sensores-proximidad/>.
8
]

[CREATIVISION, «CREATIVISION,» [En línea]. Available:
3 <https://sensordeproximidad.com/>.

9

]

[SEPIA, «SEPIA,» 2018. [En línea]. Available: <https://www.sepia.mx/sabe-como-4-funcionan-los-sensores-capacitivos/>.

0

]

[NEOTEO, «NEOTEO,» [En línea]. Available: <https://www.neoteo.com/servomotores-el-primer-paso-hacia-tu-robot>.

1

]

[LINAK, «LINAK,» 2022. [En línea]. Available: <https://www.linak-latinamerica.com/productos/actuadores-lineales/>.

2

]