



**UNIVERSIDAD ESTATAL
PENÍNSULA DE SANTA ELENA**

**FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES**

**CARRERA DE ELECTRÓNICA Y
TELECOMUNICACIONES**

TRABAJO DE TITULACIÓN

Propuesta tecnológica, previo a la obtención del Título de:

**INGENIERA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

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

AUTOR

ALBA JAZMÍN ASECIO GONZABAY

PROFESOR TUTOR

ING. LUIS ENRIQUE CHUQUIMARCA JIMÉNEZ

LA LIBERTAD – ECUADOR

2019

AGRADECIMIENTO

La mejor manera de concluir con éxito mi proyecto de titulación es reconociendo totalmente mi gratitud a aquellos que de una u otra manera fueron parte de éste.

A **Dios**, padre celestial, por su inmenso amor eterno, sabiduría, fortaleza, perseverancia para enfrentar periodos difíciles y totalmente por las bendiciones que me ha otorgado, especialmente por permitir culminar mi carrera universitaria.

A mis amados padres **Isidro Asencio** y **Rosa Gonzabay**, quienes me inspiran a dar lo mejor de mí para lograr cada objetivo de mi vida, depositaron su confianza y me enseñaron a enfrentar cada reto, ser perseverante y no desmayar hasta lograr la meta, con mucho gusto reconozco mi endeudamiento con ustedes y profundamente agradezco su incesante apoyo en toda mi carrera de estudios.

A mi querido hermano **Wilmer Asencio**, siempre brindándome total apoyo, siendo perseverante y logrando sus objetivos propuestos ha llevado a proyectarme profesionalmente como él, siendo mi ejemplo a seguir y muestra de superación.

Al **Ing. Luis Chuquimarca Jiménez**, tutor del proyecto, expreso mi profundo agradecimiento por compartir sus conocimientos, experiencias, valioso tiempo y paciencia para guiarme durante todo el transcurso de elaboración del proyecto.

A mi compañero especial, quien estuvo a mi lado en los buenos y malos momentos, a mis amigas, las hermanas que me regaló la vida; gracias a Dios hemos culminado nuestra carrera universitaria con éxito.

¡Mi más sincero agradecimiento y aprecio!

Alba Jazmín Asencio Gonzabay.

DEDICATORIA

Isidro Asencio y Rosa Gonzabay, mis héroes, personas en mi vida a quienes les debo todo, sinceramente nunca encontraré manera de devolverles todo lo que han realizado por mí. Por todas las razones existentes, dedico mi proyecto de titulación, de todo corazón y con mucha humildad a mis amados padres, quienes son la fuente de inspiración de mi vida.

¡Los amo infinitamente!

Alba Jazmín Asencio Gonzabay.

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación denominado: **“Diseño e implementación de un sistema de gestión, monitoreo y digitalización para el proceso de selección de mangos en función de su color, enfocada a la industria 4.0 para optimizar el rendimiento de producción”**, elaborado por la egresada **Asencio Gonzabay Alba Jazmín**, de la carrera de Electrónica y Telecomunicaciones 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, 11 de Diciembre del 2019.



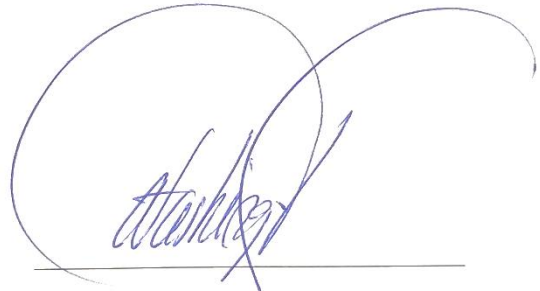
Ing. Luis Chuquimarca Jiménez, MSc.

TRIBUNAL DE GRADO



Ing. Freddy Villao Santos, MSc.

DECANO DE FACULTAD



Ing. Washington Torres Guin, MSc.

DIRECTOR DE CARRERA



Ing. Luis Chuquimarca Jiménez, MSc.

TUTOR DE TESIS



Ing. Samuel Bustos Gaibor, MSc.

TUTOR DE ÁREA



Abg. Víctor Coronel Ortiz.

SECRETARIO GENERAL

RESUMEN

Las empresas productoras de mangos realizan la clasificación de esta fruta de manera mecánica o artesanal basándose en el tamaño o color, esto enfoca a innovar el proceso de clasificación de mangos, aplicando nuevas tecnologías que se ajustan al nivel industrial actual, tales como, Visión Artificial Industrial, Dispositivos Automatas Programables, Internet Industrial de las Cosas, Sistema SCADA, Protocolo de Comunicación Profinet, que permiten la gestión y monitoreo del proceso en tiempo real con la finalidad de optimizar tiempo y eficiencia en la producción.

El proyecto de titulación consiste en aplicar visión artificial industrial, mediante un Raspberry Pi 3 con su respectiva cámara que captura la imagen real del mango y a la vez realiza la detección de los colores verde y amarillo que hacen referencia a los mangos verdes y maduros. Un controlador lógico programable realiza la automatización de un prototipo de banda transportadora, que traslada las frutas al sistema de clasificación de mangos. Mediante una pantalla de interfaz hombre-máquina y un sistema SCADA, se realiza el monitoreo y gestión de datos del sistema en tiempo real, además de una plataforma web que mediante el internet permite el monitoreo de datos del proceso desde cualquier sitio a través del dispositivo IoT2040. Una característica fundamental del proceso es el uso total de comunicación Profinet.

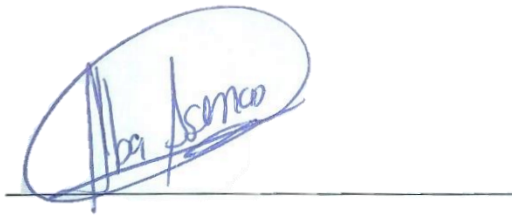
ABSTRACT

Mango producing companies classify this fruit in a mechanical or artisanal way based on size or color, this focuses on innovating the process of classification of mangoes, applying new technologies that adjust to the current industrial level, such as, Artificial Vision Industrial, Programmable Automated Devices, Industrial Internet of Things, SCADA System, Profinet Communication Protocol, which allow the management and monitoring of the process in real time in order to optimize production time and efficiency.

The titling project consists of applying industrial artificial vision, by means of a Raspberry Pi 3 with its respective camera that captures the real image of the handle and at the same time detects the green and yellow colors that refer to green and ripe mangoes. A programmable logic controller performs the automation of a prototype conveyor belt, which transfers the fruits to the mango classification system. Through a man-machine interface screen and a SCADA system, real-time monitoring and management of system data is carried out, in addition to a web platform that allows the monitoring of process data from any site through the device via the internet IoT2040. A fundamental characteristic of the process is the total use of Profinet communication.

DECLARACIÓN

El contenido del presente Trabajo de Graduación es de mi responsabilidad; el patrimonio intelectual del mismo pertenece a la Universidad Estatal Península de Santa Elena.



Asencio Gonzabay Alba Jazmín

ÍNDICE DE CONTENIDO

AGRADECIMIENTO	I
DEDICATORIA.....	II
APROBACIÓN DEL TUTOR.....	III
TRIBUNAL DE GRADO	IV
RESUMEN.....	V
ABSTRACT	VI
DECLARACIÓN.....	VII
ÍNDICE DE CONTENIDO.....	VIII
ÍNDICE DE FIGURAS.....	XI
ÍNDICE DE TABLAS	XVIII
LISTA DE ANEXOS.....	XIX
INTRODUCCIÓN	1
CAPÍTULO I.....	2
1.1 ANTECEDENTES	2
1.2 DESCRIPCIÓN DEL PROYECTO	3
1.3 OBJETIVOS DEL PROYECTO	4
1.3.1 Objetivo general.....	4
1.3.2 Objetivos específicos.	4
1.4 RESULTADOS ESPERADOS	5
1.5 JUSTIFICACIÓN	5
1.6 METODOLOGÍA	7
CAPÍTULO II.....	10
2.1 MARCO CONTEXTUAL	10
2.2 MARCO CONCEPTUAL.....	11
2.2.1 Raspberry Pi	11
2.2.2 Python	11
2.2.3 Camara Raspberry Pi	11
2.2.4 Controlador Lógico Programable	12
2.2.5 Variadores de frecuencia.....	12
2.2.6 Motores eléctricos.....	12
2.2.7 Interfaz Hombre – Máquina	13

2.2.8 Sistema SCADA	13
2.2.9 Protocolo de comunicación Profinet	13
2.2.10 Industria 4.0	14
2.2.11 Internet Industrial de las cosas	14
2.2.12 Visión Artificial Industrial	14
2.2.13 Clasificación de mangos en función de su color	15
2.3 MARCO TEÓRICO	16
CAPITULO III	17
3.1 COMPONENTES DE LA PROPUESTA	17
3.1.1 Componentes físicos	17
3.1.1.1 Simatic PLC s7-1200 Siemens	17
3.1.1.2 Sinamics G120 Siemens	18
3.1.1.3 Simatic Iot2040 Siemens	19
3.1.1.4 Raspberry Pi 3	20
3.1.1.5 Cámara Raspberry Pi	21
3.1.1.6 Prototipo de banda transportadora y sistema de clasificación	21
3.1.1.7 Motores DC	22
3.1.1.8 Motor trifásico Siemens	22
3.1.1.9 Pantalla HMI Delta	23
3.1.2 Componentes lógicos	23
3.1.2.1 TIA Portal V14	23
3.1.2.2 Raspbian Stretch	24
3.1.2.3 Python 3.0	24
3.1.2.4 OpenCV	25
3.1.2.5 Snap7 Python	25
3.1.2.6 PyCharm	25
3.1.2.7 WinSCP	26
3.1.2.8 Wonderware Intouch	26
3.1.2.9 Kepserverex 6	26
3.1.2.10 DOPSoft	27
3.1.2.11 Node-Red	27
3.1.2.12 Freeboard	27
3.1.2.13 Putty	28
3.2 DISEÑO DE LA PROPUESTA	29

3.2.1 Diagrama de red de comunicación del sistema	29
3.2.2 Diagrama de conexión de perifericos a la Raspberry Pi 3	29
3.2.3 Conexiones del dispositivo IoT2040	30
3.2.4 Diseño del sistema electromecánico de clasificación de mangos.....	31
3.2.5 Diagrama de conexiones del sistema de control de clasificación.....	31
3.2.6 Diseño del sistema de detección de mangos por visión artificial	32
3.2.7 Configuración de variador de frecuencia Sinamics G120.....	36
3.2.8 Configuración, diagrama lógico y programación de PLC s7-1200 en TIA Portal.....	39
3.2.9 Diseño de la interfaz HMI.....	43
3.2.10 Diseño de interfaz SCADA	47
3.2.10.1 Configuración de Kepsverex 6.0	47
3.2.10.2 Configuración y diseño en Wonderware Intouch	49
3.2.11 Configuración del dispositivo IoT2040	52
3.2.12 Programación de Plataforma Node Red.....	54
3.2.13 Plataforma web Freeboard.io.....	58
3.3 ESTUDIO DE FACTIBILIDAD	60
3.3.1 Factibilidad técnica	60
3.3.2 Análisis de costos del proyecto	61
3.4 PRUEBAS y RESULTADOS	62
CAPÍTULO IV	84
4.1 CONCLUSIONES	84
4.2 RECOMENDACIONES	85
4.3 BIBLIOGRAFÍAS	86
4.4 ANEXOS.....	89

ÍNDICE DE FIGURAS

Figura 1. PLC s7-1200.....	17
Figura 2. Sinamics G120 Siemens.....	18
Figura 3. Módulo de potencia de Sinamics G120.....	18
Figura 4. Unidad de control de Sinamics G120.	19
Figura 5. Panel operador de Sinamics G120.	19
Figura 6. Simatic Iot2040.....	20
Figura 7. Raspberry Pi 3.....	20
Figura 8. Módulo de cámara Raspberry Pi.....	21
Figura 9. Diseño de banda transportadora y clasificadora de mangos en SketchUp.	22
Figura 10. Motor DC.....	22
Figura 11. Motor trifásico Siemens.	22
Figura 12. Pantalla HMI Delta.	23
Figura 13. Software Siemens TIA Portal V14.....	24
Figura 14. Sistema operativo Raspbian Stretch.....	24
Figura 15. Lenguaje de programación Python.	24
Figura 16. Librería OpenCV.	25
Figura 17. Software Pycharm.	25
Figura 18. Software WinSCP.	26
Figura 19. Wonderware Intouch.	26
Figura 20. Software DOPSoft.....	27
Figura 21. Nodered.	27
Figura 22. Herramienta Freeboard.....	28
Figura 23. Software Putty.....	28
Figura 24. Diagrama de red de comunicación del sistema.....	29
Figura 25. Diagrama de conexión de periféricos a la Raspberry Pi 3.	30
Figura 26. Conexiones de Simatic IoT2040.....	31
Figura 27. Sistema de clasificación diseñado en SketchUp.....	31
Figura 28. Diagrama de conexiones de motores DC.	32
Figura 29. Detección de colores verde y amarillo.	32
Figura 30. Vista del video en tiempo real.	33

Figura 31. Rangos de tonos de colores HSV.....	33
Figura 32. Rectángulos producidos por la función cv2.rectangle	34
Figura 33. Círculos producidos por la función cv2.circle	34
Figura 34. Contorno producido por la función cv2.drawContours.....	35
Figura 35. Texto producido por la función putText.....	35
Figura 36. Sistema de detección diseñado en SketchUp.....	36
Figura 37. Asignación del nombre al variador de frecuencia Sinamics G120.	36
Figura 38. Asignación de dirección IP al variador de frecuencia Sinamics G120.	36
Figura 39. Selección del variador de frecuencia Sinamics G120.	37
Figura 40. Comunicación Profinet del Sinamics G120 con el controlador PLC s7-1200.....	37
Figura 41. Selección del Telegrama estándar en Sinamics G120.....	37
Figura 42. Asignación de dirección IP al PLC s7-1200.....	39
Figura 43. Habilitar opción PUT/GET en configuración de PLC s7-1200.....	39
Figura 44. Declaración de variables.	40
Figura 45. Diagrama lógico del proceso de clasificación de mangos.....	40
Figura 46. Conteo, producción estimada y producción alcanzada de mangos verdes.	41
Figura 47. Conteo, producción estimada y producción alcanzada de mangos maduros.	41
Figura 48. Diagrama lógico del encendido, apagado e inversión de giro del motor.	42
Figura 49. Encendido y apagado del motor de la banda transportadora.....	42
Figura 50. Diagrama lógico del ingreso de velocidad del motor.....	43
Figura 51. Ingreso de velocidad de motor en Rpm y conversión a Hz.	43
Figura 52. Configuración manual de IP y máscara de pantalla Delta.	44
Figura 53. Selección de modelo de pantalla Delta.	44
Figura 54. Configuración de comunicación.	44
Figura 55. Asignación de dirección IP y Máscara de la pantalla HMI.	45
Figura 56. Diagrama de bloques de acuerdo al diseño de la interfaz HMI.	45
Figura 57. Pantalla de ingreso al sistema.	46
Figura 58. Pantalla de control y monitoreo del sistema.	46

Figura 59. Gráfica de tendencia e historial del conteo de mangos verdes.	47
Figura 60. Grafica de tendencia e historial del conteo de mangos maduros.	47
Figura 61. Propiedades del canal de comunicación.	48
Figura 62. Propiedades del dispositivo.	48
Figura 63. Propiedades de variable creada.	49
Figura 64. Propiedades del Alias de comunicación.	49
Figura 65. Propiedades de Access Name.	50
Figura 66. Configuración de Tagname Dictionary.	50
Figura 67. Pantalla de Inicio de la interfaz del sistema SCADA.	51
Figura 68. Pantalla del control y monitoreo del proceso.	52
Figura 69. Instalación de imagen en tarjeta microSD.	52
Figura 70. Acceso al IoT2040 a través del software Putty.	53
Figura 71. Inicio de sesión del dispositivo IoT2040.	53
Figura 72. Configuración de dirección IP del IoT2040.	53
Figura 73. Configuración de dirección IP del computador.	54
Figura 74. Inicialización de plataforma Node Red.	54
Figura 75. Plataforma Node Red.	55
Figura 76. Instalación de nodos S7 y Dweetio.	55
Figura 77. Configuración de datos del PLC s7-1200 para el nodo S7.	56
Figura 78. Asignación de variables para el nodo S7.	56
Figura 79. Variables pertenecientes a la programación del PLC s7-1200 en TIA Portal.	56
Figura 80. Selección de variables para el nodo S7.	57
Figura 81. Configuración del nodo Dweetio.	57
Figura 82. Programación de nodos del sistema de clasificación de mangos.	58
Figura 83. Configuración de fuentes de datos en Freeboard.	58
Figura 84. Conexión de la fuente de datos.	59
Figura 85. Configuración de widgets en Freeboard.io.	59
Figura 86. Funcionamiento del widget Indicator Light en Freeboard.io.	59
Figura 87. Plataforma de Freeboard perteneciente al proceso de selección de mangos.	60
Figura 88. Ping 192.168.0.7 -t, correspondiente al PLC s7-1200.	63

Figura 89. Ping 192.168.0.7 -t, correspondiente al Raspberry Pi 3	63
Figura 90. PING 192.168.0.7 -t, correspondiente a PC – SCADA.....	63
Figura 91. PING 192.168.0.7 -t, correspondiente a Pantalla HMI.	64
Figura 92. Ping 192.168.0.7 -t, correspondiente al IoT2040.....	64
Figura 93. Detección del color verde de mangos.	65
Figura 94. Detección del color verde de mangos.	65
Figura 95. Detección del color amarillo (maduros) de mangos.	65
Figura 96. Detección de colores; verde y amarillo de mangos.	66
Figura 97. Diseño en SketchUp del sistema, con cámara a 0°, 15°, 30°, 45°, 60°, 85° para la detección.....	66
Figura 98. Detección de color verde y activación de salida Q0.0	67
Figura 99. Activación de salida Q0.0 correspondiente al primer actuador clasificador.	67
Figura 100. Detección de color amarillo y activación de salida Q0.4 para su conteo.....	67
Figura 101. Paso de mango maduro o amarillo sin activar ningún actuador.	68
Figura 102. Diseño en SketchUp del sistema, con cámara a 0° para la detección.	68
Figura 103. Diseño en SketchUp del sistema, con cámara a 15° para la detección.	68
Figura 104. Diseño en SketchUp del sistema, con cámara a 30° para la detección.	69
Figura 105. Diseño en SketchUp del sistema, con cámara a 45° para la detección.	69
Figura 106. Diseño en SketchUp del sistema, con cámara a 60° para la detección.	70
Figura 107. Diseño en SketchUp del sistema, con cámara a 85° para la detección.	70
Figura 108. Número vs grados de detección de mangos verdes.....	71
Figura 109. Número vs grados de detección de mangos maduros.	71
Figura 110. Sistema SCADA totalmente apagado.	72
Figura 111. Variables en KEPServer del sistema totalmente apagado.....	72
Figura 112. Prueba de ingreso de velocidad de 900 RPM mediante SCADA.	72

Figura 113. Prueba de la variable “INGRESO_VELOCIDAD_RPM” y “FRECUENCIA” en KEPServer.....	73
Figura 114. Prueba de ingreso de velocidad de 900 RPM mediante interfaz HMI.	73
Figura 115. Prueba de encendido del motor para la banda transportadora del sistema mediante SCADA.....	73
Figura 116. Prueba de la variable “ENCENDER_APAGAR_MOTOR_BANDA” en KEPServer.	73
Figura 117. Prueba de encendido del motor para la banda transportadora del sistema mediante interfaz HMI.....	74
Figura 118. Prueba del primer detector de mangos verdes, simulación del mango mediante SCADA y activación de su variable en KEPServer.	74
Figura 119. Prueba del primer detector de mangos verdes, simulación del mango mediante interfaz HMI.	74
Figura 120. Prueba del cuarto detector de mangos maduros, simulación del mango mediante SCADA y activación de su variable en KEPServer.	75
Figura 121. Prueba del cuarto detector de mangos maduros, simulación del mango mediante interfaz HMI.	75
Figura 122. Prueba del conteo de mangos verdes y maduros con su respectiva grafica de tendencia mediante SCADA.	76
Figura 123. Prueba de la variables “CONTADOR_VERDES” y “CONTADOR_MADUROS” en KEPServer.	76
Figura 124. Prueba del conteo de mangos verdes y maduros mediante interfaz HMI.....	76
Figura 125. Prueba de la gráfica de tendencia del respectivo conteo de mangos verdes mediante interfaz HMI.	76
Figura 126. Prueba de la gráfica de tendencia del respectivo conteo de mangos maduros mediante interfaz HMI.....	77
Figura 127. Prueba del ingreso producción estimada de mangos verdes y maduros mediante SCADA.	77
Figura 128. Prueba de las variables “PRODUCCION_ESTIMADA_VERDES” y “PRODUCCION_ESTIMADA_MADUROS” en KEPServer.	77

Figura 129. Prueba del ingreso producción estimada de mangos verdes y maduros mediante interfaz HMI.	78
Figura 130. Prueba de alarma activada al alcanzar la producción estimada de mangos maduros mediante SCADA.	78
Figura 131. Prueba de alarma activada al alcanzar la producción estimada de mangos maduros mediante interfaz HMI.	78
Figura 132. Prueba de alarma desactivada después de alcanzar la producción estimada de mangos maduros mediante SCADA.	79
Figura 133. Prueba de alarma desactivada después de alcanzar la producción estimada de mangos maduros mediante interfaz HMI.	79
Figura 134. Prueba del botón para reiniciar el conteo de mangos verdes y maduros mediante SCADA y activación de su variable en KEPServer.	79
Figura 135. Prueba del botón para reiniciar el conteo de mangos verdes y maduros mediante interfaz HMI.	80
Figura 136. Prueba del botón para reiniciar el conteo de mangos verdes y maduros mediante SCADA y activación de su variable en KEPServer.	80
Figura 137. Prueba del botón para reiniciar el conteo de mangos verdes y maduros mediante interfaz HMI.	80
Figura 138. Plataforma web Freeboard del sistema.	81
Figura 139. Programación de variables de velocidad y encendido/apagado de motor en NodeRed.	81
Figura 140. Indicadores de la interfaz de Freeboard perteneciente hora y fecha, velocidad (RPM-Hz) y encendido/apagado de motor.	81
Figura 141. Programación de variable del segundo detector de mangos verdes en NodeRed.	82
Figura 142. Indicadores de la interfaz de Freeboard perteneciente al segundo detector de mangos verdes.	82
Figura 143. Programación de variable del primer detector de mangos maduros en NodeRed.	82
Figura 144. Indicadores de la interfaz de Freeboard perteneciente al primer detector de mangos maduros.	82

Figura 145. Programación de variable del conteo de mangos verdes en NodeRed.	82
Figura 146. Indicadores de la interfaz de Freeboard perteneciente al conteo de mangos verdes y grafica de tendencias.	82
Figura 147. Programación de variable del conteo de mangos maduros en NodeRed.....	83
Figura 148. Indicadores de la interfaz de Freeboard perteneciente al conteo de mangos maduros y grafica de tendencias.	83
Figura 149. Programación de variable de la producción estimada de mangos verdes y maduros.	83
Figura 150. Indicadores de la interfaz de Freeboard perteneciente la producción estimada de mangos verdes y maduros.	83
Figura 151. Programación de variable de meta alcanzada de la producción estimada de mangos verdes y maduros.	83
Figura 152. Indicadores de la interfaz de Freeboard perteneciente alarma activada al alcanzar la producción estimada de mangos verdes y maduros.....	83

ÍNDICE DE TABLAS

Tabla 1. Características técnicas de PLC s7-1200 [23].	18
Tabla 2. Datos técnicos de Módulo de potencia.	18
Tabla 3. Datos técnicos de Unidad de control.	19
Tabla 4. Datos técnicos de IoT2040.	20
Tabla 5. Especificaciones técnicas de Raspberry Pi 3.	21
Tabla 6. Especificaciones técnicas de Motor Siemens.	22
Tabla 7. Datos técnicos de HMI Delta.	23
Tabla 8. Configuración - Telegrama Estándar 1. Manual Sinamics G120.	38
Tabla 9. Costos de equipos para implementación del proyecto.	61
Tabla 10. Costos de licencias de software utilizados en el proyecto.	61
Tabla 11. Costos de otros elementos del proyecto.	62
Tabla 12. Costos finales del proyecto.	62
Tabla 13. Direcciones IP de la red del sistema.	62
Tabla 14. Datos de la detección de mangos	66
Tabla 15. Datos de la clasificación de mangos.	68
Tabla 16. Datos de la clasificación de mangos.	69
Tabla 17. Datos de la clasificación de mangos.	69
Tabla 18. Datos de la clasificación de mangos.	69
Tabla 19. Datos de la clasificación de mangos.	70
Tabla 20. Datos de la clasificación de mangos.	70
Tabla 21. Datos generales de detección y clasificación de mangos.	71

LISTA DE ANEXOS

Anexo 1: Instalación de sistema operativo Raspbian Stretch.

Anexo 2: Instalación y funcionamiento de biblioteca Snap7.

Anexo 3: Instalación de librería OpenCV.

Anexo 4: Configuración PyCharm.

Anexo 5: Conexión WinSCP.

Anexo 6: Programación de transporte de mangos en software DOPSoft.

Anexo 7: programación de usuario y contraseña en software DOPSoft.

Anexo 8: programación de transporte de mangos en software Wonderware Intouch.

Anexo 9: prototipo de banda transportadora y clasificadora de mangos.

INTRODUCCIÓN

La clasificación de mangos en el Ecuador, en empresas como Refin S.A. y Agriproduct S.A. se realiza de manera manual o mecánica, hecho que conlleva a innovar dicho proceso, por lo que el proyecto consiste en el 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 optimizando la forma de producción.

Se aplica visión artificial que permite el procesamiento de imágenes de mangos en tiempo real, mediante un Raspberry Pi 3 con su respectiva cámara para detectar los colores de las frutas. Se automatiza el sistema de transporte y clasificación de mangos mediante un *Programmable Logic Controller* (PLC). El proceso de clasificación de mangos es monitoreado y gestionado en tiempo real mediante un sistema *Supervisory Control And Data Acquisition* (SCADA), una pantalla *Human Machine Interface* (HMI) y una plataforma web Freeboard conectada al dispositivo *Internet of Things* (IOT), que se encuentra en red con el PLC utilizando el protocolo de comunicación Profinet.

La propuesta tecnológica se encuentra conformada por cuatro capítulos; el primer capítulo se refiere a los antecedentes, descripción, objetivos del proyecto, además de los resultados esperados, justificación y metodología usada en el desarrollo. El segundo capítulo describe la teoría sobre los temas generales empleados en el proyecto. El tercer capítulo contiene los componentes usados; lógicos y físicos, además de la explicación de cada ejecución para el desarrollo de la propuesta tecnológica junto al respectivo estudio de factibilidad y también las pruebas y resultados obtenidos. Finalmente, el cuarto capítulo corresponde a las conclusiones, recomendaciones, bibliografías y anexos.

CAPÍTULO I

1.1 ANTECEDENTES

El mango, una fruta muy conocida mundialmente, se cultiva en lugares con climas intertropicales sobretodo en Sudamérica. Existen diversos tipos, varían su color, primero es verde de sabor ácido y luego de color amarillo con un sabor dulce cuando la fruta está madura, es muy antigua, cultivada desde más de 6000 años extendiendo su producción de cultivo desde India hasta América [1].

En la agroindustria, la productividad ha tomado manufactura artesanal desde el año 1960 en Latinoamérica, es decir, se realizaba mediante procesos de producción mecanizada sin un sistema de control automatizado.

En el intervalo de septiembre a enero es el tiempo de cosecha en Ecuador y se cuenta con plantas o fabricas para su tratamiento y empaquetado, este determinado tiempo de cosecha da prioridad de exportar antes que otras grandes empresas de otros países, tales como México y Brasil, superando niveles de hasta siete millones de cajas de mangos. El lugar principal del cultivo de esta fruta es la provincia del Guayas y cuenta con una área para la producción de 7700 hectáreas aproximadamente [2].

Una de las principales plantas productoras de Mango es el Grupo Agriproduct S.A., ubicada en la ciudad de Guayaquil, correspondiente a la zona 5, es considerada la más innovadora de Latinoamérica, gran infraestructura y aproximadamente un 40% de producción nacional de mangos. El proceso de selección de mangos se realiza de manera mecánica basándose en el tamaño y artesanal de acuerdo al color del mango.

La empresa Refin S.A., posee sus instalaciones en la ciudad de Guayaquil, realiza la adecuada clasificación de mangos. Los mangos son seleccionados manualmente según su tamaño y madurez al momento de pasar en una banda transportadora.

Sin embargo, las compañías mencionadas no cuentan con aspectos fundamentales, como la automatización del proceso de clasificación de mangos con nuevas tecnologías, visión artificial industrial, la digitalización y sistema SCADA que permite llevar un monitoreo y control de todo el proceso en tiempo real.

No se dispone de avances tecnológicos que permitan optimizar tiempos de producción y eficiencia en la planta, no se recibe información detallada en tiempo real, control, monitoreo de maquinarias que directamente ayudarían a mejorar la calidad del proceso.

1.2 DESCRIPCIÓN DEL PROYECTO

El proyecto tecnológico consiste en realizar el diseño e implementación de un prototipo del sistema automatizado de clasificación óptima de mangos según su color en tiempo real además de su digitalización, con el propósito de mejorar la productividad y eficiencia de la planta o empresa.

Primero se realiza el traslado de mangos mediante un prototipo de banda transportadora accionada por un motor siemens trifásico y un variador de frecuencia Sinamics G120 que utiliza comunicación profinet con el controlador PLC s7-1200.

El sistema de detección de colores de mangos; amarillos y verdes, se realiza al momento de la llegada de éste al final de la banda transportadora, en el cual, una cámara obtiene la señal de video, la cual es procesada mediante visión artificial industrial, que capta la imagen del mango en tiempo real y su vez se detecta el color y contorno utilizando un dispositivo con una plataforma llamada Raspberry Pi 3.

El proceso de Visión artificial tiene el objetivo de identificar el color de los mangos para realizar la clasificación, la Raspberry Pi 3 se comunica con el PLC s7-1200 mediante el protocolo de comunicación profinet, éste envía información a dicho controlador y cada vez que se detecte el color verde, este activan las salidas correspondientes, es decir, activan los actuadores electromecánicos ubicados al final de banda transportadora, logrando de esta manera realizar la clasificación de los mangos en función de su color.

La gestión y monitoreo del sistema automatizado se realiza mediante pantalla HMI Delta DOP-B03E211, y Software Wonderware Intouch en PC; ambas con comunicación profinet, estos admiten el control de encendido, apagado y variación de frecuencia de la banda transportadora, la visualización de una simulación de transporte de productos, indicadores de presencia de mangos, conteo de mangos

verdes (verde) y maduros (amarillo) con sus respectivas gráficas de tendencias e históricos de datos y alarmas.

En cuanto a la recopilación de datos, enviarlos a la nube y su digitalización se utiliza un dispositivo Siemens, Simatic IoT 2040, el cual consta de dos puertos Profinet configurados; el primero se conecta a la red local del sistema automatizado, para mediante nodos previamente programados realizar la lectura y escritura de datos en el controlador PLC s7-1200 y el segundo conectado directamente a internet, con la finalidad de enviar los datos a la nube para realizar la gestión y monitoreo del proceso desde cualquier lugar mediante una plataforma web también programada y diseñada, de esta forma analizar datos y lograr la digitalización del sistema productivo.

Una característica importante del sistema en general es el uso del protocolo de comunicación Profinet con funcionalidad TCP/IP; este es un protocolo fundamentado en Profinet aplicado a la automatización industrial para el acceso y control de dispositivos o equipos en tiempo real [3].

1.3 OBJETIVOS DEL PROYECTO

1.3.1 Objetivo general

Diseñar e implementar un sistema control, monitoreo y digitalización para el proceso de selección de mangos en función de su color, por medio de instrumentación industrial y tecnologías de la industria 4.0, para optimizar el rendimiento de producción.

1.3.2 Objetivos específicos.

- Diseñar, configurar e implementar la red de comunicación para la interconexión de los equipos de gestión, monitoreo y digitalización, mediante el protocolo Profinet.
- Desarrollar un sistema de control con visión artificial industrial que efectúe la detección de colores para la clasificación óptima de los mangos, utilizando las librerías de software OpenCV.

- Programar el PLC Siemens s7-1200 para controlar las entradas/salidas del sistema que corresponden al proceso de clasificación de mangos según su color.
- Diseñar e implementar una interfaz mediante SCADA y una pantalla HMI, que permita gestionar y monitorear los parámetros del proyecto.
- Enviar los parámetros digitalizados referentes al proyecto, mediante el equipo Siemens Simatic IOT 2040 enlazada a una base de datos en el Servidor.

1.4 RESULTADOS ESPERADOS

- Correcta implementación de la red de comunicaciones de todo el sistema con el respectivo protocolo Profinet para conseguir una transmisión de datos eficiente.
- Un sistema de control para el proceso mediante visión artificial con su respectiva programación del PLC s7-1200 y lograr cumplir las diferentes tareas de control del sistema.
- Interfaz hombre-máquina y SCADA para la gestión y monitoreo de los parámetros establecidos en el sistema, que permitan saber datos sobre el sistema y en otros casos detectar anomalías a través de una pantalla para solucionarlos mediante las acciones adecuadas en tiempo real.
- Recopilar información de datos del proceso de selección de mangos en función de su color, enviar a la nube y permitir visualizar estos datos desde cualquier sitio.

1.5 JUSTIFICACIÓN

La tecnología es un factor clave para incrementar la productividad, esto es evidente, ya que en muchos sectores o países, es notable su avance, obteniendo directamente una mejor calidad del producto y eficacia en la producción. Se deben aprovechar las tecnologías comenzando desde la educación, lograr una población especializada que conlleve a científicos o ingenieros a direccionarse hacia las necesidades empresariales en cuanto a innovación tecnológica y desarrollo de nuevos proyectos de procesos productivos industriales [4].

Refiriendo a la zona 5, exige incrementar la producción mediante el cambio de la matriz productiva reduciendo directamente costos operacionales y aumentando la producción de mangos mediante su clasificación. Implementar nuevas tecnologías físicas y digitales junto al desarrollo del proceso de clasificación, puesto que los sistemas que presentan actualmente son de manera artesanal, por lo tanto, no se obtiene una excelente eficiencia y eficacia en la producción. Claramente, para evitar el lento avance de innovaciones tecnológicas es totalmente necesario realizar fuertes inversiones en ello, es importante su desarrollo local basándose en políticas macroeconómicas [5].

Para complementar el proceso de planificación territorial, en la Zona 5, y el Plan Nacional del Buen Vivir; es necesario realizar fuertes inversiones científico tecnológicas que puedan ser realizadas localmente, esto se puede lograr mediante el apoyo de políticas macroeconómicas para que los capitales de inversión se enfoquen los escenarios tecnológicos necesarios para obtener los resultados deseados en cuanto a la transformación tecnológica de la matriz productiva del país.

Existen productores de mangos que realizan la selección de forma artesanal, por dicha razón, éste proyecto enfoca a otorgar a la industria una manera de innovar la gestión y monitoreo en tiempo real implementando un prototipo de sistema automatizado para la clasificación de mangos en función de su color mediante visión artificial y autómatas programables, además de nuevas tecnologías que se ajusten al nivel industrial, tales como variador de frecuencia, interfaz para SCADA, protocolos de comunicación Profinet, internet de las cosas, entre otras.

Realizar la digitalización de datos, es llevar información del proceso hacia la nube mediante el equipo Siemens Simatic IoT2040, éste permite el monitoreo del proceso desde cualquier lugar por medio de internet y una plataforma web, de esta forma permite analizar dichos datos mediante gráficas de tendencias y a la vez obtener históricos de la producción, con el fin de lograr un ahorro de tiempo y recursos, además de una mayor optimización del sistema productivo.

Este proyecto enfoca a la Industria 4.0, con el propósito de encaminar a las empresas artesanales de la Zona 5 hacia estas nuevas tecnologías industriales. Implementar esta nueva industria a la instrumentación de producción del Ecuador, es muy

importante, incentiva a efectuar los respectivos objetivos del Plan Nacional del Buen Vivir de la Secretaria Nacional de Planificación y Desarrollo [6].

1.6 METODOLOGÍA

Para realizar el proyecto se aplican las metodologías bibliográfica y diagnóstica.

- **INVESTIGACIÓN BIBLIOGRÁFICA**

Método en el cual se realiza consultas en fuentes confiables, tales como; libros, artículos científicos y páginas oficiales en internet, para extraer los conceptos, principios, leyes y métodos necesarios para el desarrollo del sistema de gestión del proceso de selección de mangos por medio de visión artificial, comunicación Raspberry Pi 3 y PLC s7-1200, actuadores, Variador de frecuencia, SCADA, pantalla HMI e IOT 2040 para la digitalización de proceso a través del Internet de las cosas.

- **INVESTIGACIÓN DIAGNÓSTICA**

Se emplean dispositivos y equipos electrónicos, con los cuales se realizan respectivas pruebas del sistema de selección de mangos en función de su color, diagnosticando errores, los cuales son corregidos hasta comprobar su correcto funcionamiento, logrando optimizar el sistema y aumentado directamente su confiabilidad y calidad.

A continuación, se explican las fases en las que se aplican las diferentes metodologías:

FASE 1

La construcción del prototipo del sistema automatizado se realiza mediante búsqueda y análisis de información técnica; la implementación de la banda transportadora es para el traslado de mangos, ésta es accionada por un motor siemens trifásico y un variador de frecuencia Sinamics G120 configurada en TIA Portal V14 que utiliza comunicación Profinet con el controlador PLC s7-1200.

FASE 2

Aplicando conocimientos adquiridos en la formación académica e investigando en bibliografías confiables toda información requerida para el uso los diferentes equipos del sistema. Instalación en Raspberry Pi 3 el Sistema Operativo Raspbian Stretch junto a Open Cv con enlaces de Python, luego realizar la programación adecuada para la detección de colores de la fruta con una cámara y finalmente lograr la comunicación Profinet entre la Raspberry Pi 3 y PLC s7-1200 por medio de la biblioteca Snap7.

FASE 3

Se diseña e implementa un actuador electromecánico con movimiento lineal que logre empujar la fruta y así lograr su clasificación, entonces, al detectar el rango de color verde de la fruta, mediante comunicación Profinet el Raspberry Pi 3 envía la señal al controlador s7-1200 y éste activa el actuador electromecánico correspondiente, a su vez en el controlador se realiza la programación en lenguaje Ladder para el conteo de las frutas verdes y maduras.

FASE 4

Para lograr la gestión y monitoreo del sistema automatizado mediante comunicación Profinet, se realiza técnicas de observación e investigación diagnóstica para el diseño de la interfaz gráfica y vinculación de sus variables; para lograrlo a través de pantalla HMI Delta se utiliza el software DopSoft y para el SCADA a través de PC se usa el software Wonderware con su respectivo servidor OPC Kepservex.

FASE 5

La recopilación de datos y envío a la nube por medio de internet se da a través del dispositivo Simatic IoT 2040, el cual se enlaza por comunicación Profinet con el PLC s7-1200 y extrae datos, éste requiere de un Sistema Operativo, programación en herramienta Node-Red y finalmente realizar la respectiva configuración en la plataforma Web para la gestión y monitoreo del sistema en tiempo real.

FASE 6

Analizar los resultados obtenidos durante el desarrollo del proyecto, comparar con los objetivos planteados, verificar si el sistema es óptimo y en caso de ser necesario realizar las respectivas correcciones para el proceso de selección de mangos en función de su color, con lo cual obtienen las respectivas conclusiones del proyecto.

CAPÍTULO II

2.1 MARCO CONTEXTUAL

En la zona 5 la producción de mangos junto a su exportación es muy importante; en la actualidad Agriproduct S.A. es una de las principales plantas productoras de mangos, situada en la ciudad de Guayaquil y cuenta con un proceso de selección de mangos de manera mecánica y artesanal de acuerdo al color de dicha fruta. De igual manera, la empresa Refin S.A., instalada en la ciudad de Guayaquil, realiza la clasificación de mangos manualmente basándose en su tamaño y madurez al pasar a través de una banda transportadora.

La implementación del sistema de gestión, monitoreo y digitalización del proceso de clasificación de mangos utilizando nuevas tecnologías busca incrementar la producción, disminuir costos operacionales e innovar el método de clasificación, ya que los sistemas que se usan actualmente en las empresas son mecánicos o artesanales.

Se realiza un sistema de control automático que permite la detección de colores verde y amarillo de mangos en tiempo real aplicando visión artificial. La detección se efectúa mediante una cámara que captura y procesa imágenes de los mangos al momento de su paso a través de un prototipo de banda transportadora y por medio de un sistema automatizado de actuadores electromecánicos se realiza la respectiva clasificación.

Además, se realiza el monitoreo y la gestión de datos del proceso en tiempo real mediante la implementación de un sistema SCADA, una Interfaz Hombre Maquina y la plataforma web Freeboard por medio de la tecnología IOT con enfoque a la Industria 4.0.

2.2 MARCO CONCEPTUAL

2.2.1 Raspberry Pi

Raspberry Pi comprende una familia de computadoras de bajo costo, con el objetivo de hacer la informática accesible para todos e incentivar a la nueva generación a tomar el control de la programación. Existen diversos modelos de Raspberry Pi; modelo 1 con versiones A, B y B+, modelo 2 con versión B y modelo 3 con versión B+. Además, consta con un conector *High-Definition Multimedia Interface* (HDMI) para conectarse a una pantalla, puertos *Universal Serial Bus* (USB) para mouse y teclado u otro dispositivo, conector Profinet para enlazar a la red, puerto *Camera Serial Interface* (CSI) para la conexión de la cámara, conectores *General Purpose Input/Output* (GPIO) para sensores o actuadores. La Raspberry Pi se ejecuta en un sistema operativo Linux, básicamente puede usarse como una PC, servidor web o controlador de dispositivos, además se pueden agregar compiladores y bibliotecas [7].

2.2.2 Python

Es un lenguaje de programación de alto nivel, orientada a objetos basado en comandos propios legibles, por lo tanto, da la ventaja de aprender a programar de una manera rápida y sencilla. Posee una amplia biblioteca estándar y están disponibles de forma gratuita en su página oficial. Python es una multiplataforma, puede ser usado en distintos entornos y sistemas operativos, tales como, Linux, Windows, Mac X, etc. [8].

2.2.3 Camara Raspberry Pi

Es un tipo de cámara especial para Raspberry Pi y trabaja con todos los modelos disponibles, se conecta directamente al módulo, ya que cuenta con una ranura especial de estándar CSI, ésta cámara es capaz de capturar imágenes o videos. Es comúnmente aplicado a la seguridad, detección de movimientos, visión artificial industrial, etc. [9].

2.2.4 Controlador Lógico Programable

Los PLC son equipos electrónicos comúnmente aplicados en la Industria para controlar procesos y resolverlos en tiempo real, mediante sus canales de entradas y salidas facilitan la conexión de sensores o actuadores u otros componentes, tal como el nombre lo indica, el dispositivo debe ser programado en su propio software, tomando en cuenta las variables de un proceso para adquirir, guardar y restablecer sus valores cíclicamente y finalmente realizar el control [10].

2.2.5 Variadores de frecuencia

Al variar la frecuencia de alimentación se logra variar la velocidad de un motor y se realiza mediante los conocidos variadores de frecuencia. Este es un equipo de electrónica de potencia, apto para modificar desde la mínima a la máxima frecuencia y se aplican en maquinarias comunes sin necesidad de devanado [11].

La utilización de motores ha cambiado notablemente gracias a los variadores de frecuencia. De una u otra manera estos han ayudado la operatividad de procesos orientados a la industria, tales como, bandas transportadoras de producción, actuadores mecánicos, mezclado de líquidos, desplazamiento de ascensores, etc.

2.2.6 Motores eléctricos

Los motores ejercen un movimiento rotatorio en su eje, en el cual se pueden ajustar sistemas mecánicos enfocados directamente al sector industrial y doméstico. Comúnmente son usados en ámbito de la automatización y de acuerdo al sistema de alimentación, estos motores pueden ser de corriente alterna o continua [12].

Estos equipos ejercen un importante rol, a través del suministro de potencia éste es apto para una variedad de aplicaciones; caseras, comerciales y sobretodo en el ámbito industrial. Su gran rango de variabilidad, bajo costo para operar, y otras características llevan a la conclusión de que este movimiento no puede ser asimilada por algún otro modo de potencia.

2.2.7 Interfaz Hombre – Máquina

Conocido por su acrónimo HMI, son equipos que admiten a las personas relacionarse o interactuar de manera sencilla con procesos automatizados. A través de éstos se pueden modificar datos o variables referentes a un proceso y ordenar la activación de sus actuadores, obtener información en tiempo real de un proceso que está siendo controlado por un PLC, además, permite la detección de errores en el sistema, generando alarmas, indicando al operador que debe actuar de manera rápida frente a estas fallas. Estos dispositivos suelen ser paneles de operación o pantallas táctiles [13].

2.2.8 Sistema SCADA

SCADA es un sistema de supervisión, control y adquisición de datos en tiempo real, siendo estas, variables de los dispositivos conectados directamente a los equipos de control o autómatas, admite una supervisión para tomar decisiones, adquisición, alarmas de avisos, visualización de estados de datos, guardarlos y realizar una evaluación posteriormente. SCADA cumple un papel muy importante en el sector de automatización industrial, se trata de llevar un sistema a un nivel superior, a través de software de monitoreo que permitan la interconexión entre equipos de control y equipos de gestión que a la vez implementan una determinada comunicación entre el operador y el proceso establecido [14].

2.2.9 Protocolo de comunicación Profinet

Profinet como estándar de comunicación para la automatización industrial, básicamente es una adaptación de Profinet, admite el uso de cableado y switches alcanzando una comunicación sencilla, veloz, manejable y abierta. Una de las características importantes de éste estándar es su aplicación enfocada a la automatización industrial, permitiendo comunicación entre sistemas electrónicos en tiempo real ya que posee una alta velocidad de transmisión y recepción de datos el cual beneficia directamente el óptimo funcionamiento de procesos [15].

Soporta ser aplicado en sistemas automatizados modernos, como los variadores de frecuencia que actualmente poseen tecnología Profinet, permitiendo su control en tiempo real mediante su programación y comunicación.

2.2.10 Industria 4.0

Conocida también como la cuarta revolución industrial, básicamente es llevar a las fábricas o plantas a ser más inteligentes, automatizadas y eficientes, aplicando nuevas tecnologías físicas, digitales junto a las etapas de desarrollo de un proceso para lograr eficiencia y aumento de productividad, permite el seguimiento y análisis de datos en tiempo real garantizando mayor asertividad en la toma de decisiones, saber todas las etapas del proceso y el momento en que se produzcan, monitoreo de forma remota del proceso para evitar posibles fallos. Se basa en varias tecnologías, tales como, Internet de las cosas, plataformas sociales, visión artificial, robótica, realidad aumentada, bigdata, entre otros [16].

2.2.11 Internet Industrial de las cosas

Un avance del internet de las cosas, es el *Industrial Internet of Things* (IIOT), es una de las tecnologías que conforman la Industria 4.0, esta tecnología permite la comunicación de una variedad de sensores o dispositivos que operan en el sector industrial, tales como, sensores de temperatura, presión, nivel, presencia, etc. Para llevar a cabo una gestión, monitoreo y control en tiempo real a través de una plataforma web. Existen diversas plataformas o software que permiten la conexión y comunicación de sensores u otros dispositivos físicos, captar ciertos valores reales de los dispositivos para realizar la gestión, monitoreo y control de éstos sin la necesidad de la presencia de un operador en el lugar del proceso físico [17].

2.2.12 Visión Artificial Industrial

La visión industrial forma parte de la Inteligencia artificial, incluye la instrumentación óptica, sistemas informáticos, automatización industrial, entre otros. Este tipo de visión artificial se basa en equipos o máquinas orientadas por procesamiento de imágenes y es aplicada a sistemas que se encargan de capturar imágenes que permitan accionar dispositivos de entrada o salida para lograr un

control de estos. Mediante estos sistema se pueden automatizar procesos que requieran acciones repetitivas en tiempo real, en casos no se pueden realizar verificaciones visuales de productos de manera tradicional y la visión artificial facilita este control sin total contacto físico y a gran velocidad. Estos sistemas analizan, miden colores o luz, presencia-ausencia, formas, bordes, dimensiones, áreas, posiciones, distancias, incluso permiten un análisis de objetos en movimiento o tiempo real. Se aplican en diversos sectores industriales, tales como, electrónica, alimentación, automoción, farmacéutico, etc. Para realizar control de procesos, control de calidad, visión estéreo, control de tráfico, clasificación de objetos, entre otros. Un sistema de visión artificial realiza la comunicación a través de estándares RS232, Profinet, I/O, Profinet. La cámara capta una señal analógica, la convierte y proporciona una digital para su almacenamiento y proceso, luego nuevamente convierte dicha señal en señal de video analógica permitiendo su visualización en una pantalla, además las señales pueden ser usadas para realizar el control de actuadores o dispositivos externos [18].

2.2.13 Clasificación de mangos en función de su color

La clasificación de mangos, al igual que otros procesos, es un requerimiento de producción de ésta fruta, consiste en separar los mangos de forma manual o mecanizada en función de sus características físicas, tales como el tamaño, color, peso y forma [19]. Tipos de clasificación:

- La clasificación manual consiste en separar el producto de acuerdo a su diámetro, color, peso, forma o madurez y se realiza usando una banda transportadora o rodillos giratorios permitiendo a los operarios escogerlos de acuerdo a parámetros asignados logrando la clasificación.
- La clasificación mecánica se realiza utilizando rodillos giratorios de igual separación, de esta forma los mangos del mismo tamaño pueden pasar por las separaciones y caer por debajo de los rodillos separándose de los de mayor tamaño.
- La clasificación basada en tecnología electrónica y visión artificial, obteniendo magnitudes físicas que permiten la clasificación en función de su

color, área, tamaño, peso, forma, etc. Reduciendo costos y aumentando rendimiento por su precisión y velocidad.

2.3 MARCO TEÓRICO

Seguidamente, se definen artículos y proyectos de titulación previamente diagnosticados y relacionados con la presente propuesta tecnológica, que sirven de orientación considerable para la realización de este proyecto.

“Sistema de clasificación por visión artificial de mangos tipo Tommy”, publicación de la revista UIS INGENIERÍAS, éste proyecto es realizado con el propósito de obtener una correcta clasificación en función del color y tamaño de tres mangos al mismo tiempo, utilizando visión artificial, Matlab y cámara para llevar a cabo tres fundamentales etapas de adquisición, procesamiento y reconocimiento de la imagen, el uso de una banda transportadora que lleve los productos al final de la banda para que mecanismos logren la clasificación, de igual manera se obtiene una elaboración de graficas referentes al proceso [20].

“Implementación de un módulo de clasificación por materiales y tamaños con PLC y pantalla táctil”, tesis desarrollada con el objetivo de implementar un módulo de clasificación automática de materiales según su tamaño, mediante un autómatas S7-1200 y HMI que permita el monitoreo y control de un proceso por medio de comunicación Profinet, comprobando su funcionalidad total al realizar el proceso en menor tiempo y de manera eficiente, además de permitir que los estudiantes tengan una mejor comprensión practica en el campo de automatización y control [21].

“Puesta en marcha de un sistema de automatización Siemens Simatic IOT2040”, en la Universidad del país de Vasco se implementó este proyecto, que permite la comunicación inalámbrica de sensores y actuadores con el Simatic Iot2040, se realiza la correcta configuración y programación con lenguaje de alto nivel del Iot2040 y plataformas en las que se accede a los datos del sistema automatizado desde cualquier lugar del planeta, llevando a la industria actual hacia la cuarta revolución industrial [22].

CAPITULO III

3.1 COMPONENTES DE LA PROPUESTA

En la siguiente sección se procedió a segmentar en dos componentes para el desarrollo de la propuesta de tesis, procediendo a describir los componentes físicos y detallando los componentes lógicos.

3.1.1 Componentes físicos

Para llevar a cabo el diseño e implementación del sistema para el proceso de selección de mangos en función de su color enfocada a la industria 4.0 se emplearon diversos componentes eléctricos y electrónicos, detallados a continuación:

3.1.1.1 Simatic PLC s7-1200 Siemens

Es un dispositivo controlador que tiene una gran variedad de usos, comúnmente es utilizado para procesos automatizados sencillos y exactos, controlando sensores y actuadores industriales, además permite la comunicación con dispositivos con equipos externos mediante el protocolo Profinet y ModBus. En el actual proyecto se emplea el PLC s7-1200 (ver Figura 1) para comunicarse con un dispositivo Raspberry Pi, además controla los actuadores y el variador de frecuencia [23]. En la tabla 1 se muestran detalles fundamentales del componente:



Figura 1. PLC s7-1200.

DATOS TÉCNICOS	
Modelo	CPU 1212C AC/DC/relé
Tensión de alimentación	120 - 240 V AC
Entradas digitales	8 entradas.
Salidas digitales	6 salidas; relé.
Entradas analógicas	2 entradas; 0-10V.
Dimensiones	90x100x75 mm
Interfaz	Profinet(Profinet Industrial)

Tabla 1. Características técnicas de PLC s7-1200 [23].

3.1.1.2 Sinamics G120 Siemens

El dispositivo Sinamics G120 es un variador de frecuencia industrial inteligente, adaptable a controladores Simatic, ideal para aplicaciones como maquinarias de embalaje, bombeo, bandas transportadoras, ventiladores, mezcladores entre otros (ver Figura 2). En el presente proyecto es usado para variar la velocidad y giro del motor que se encarga de la banda transportadora [24].



Figura 2. Sinamics G120 Siemens.

El variador de frecuencia Sinamics G120 posee módulos de adaptables para su funcionamiento, los cuales se mencionan a continuación:

Módulo de potencia: Es el modulo que proporciona potencia al motor (ver Figura 3). Los detalles que se consideraron se encuentran en la tabla 2.



Figura 3. Módulo de potencia de Sinamics G120.

DATOS TÉCNICOS	
Modelo	PM240-2
Entrada	1/3ØAC 200-240V ±10% 9.6/5.5A 47-63Hz
Salida	3ØAC 0-INPUT V In=4.2A
Rango de potencia	0.37-250 kW.
Dimensiones	196x73x165 mm

Tabla 2. Datos técnicos de Módulo de potencia.

Unidad de control: Controla y supervisa el módulo de potencia, además de la comunicación con otros dispositivos (ver Figura 4). Los detalles de la Unidad de Control se indican en la tabla 3.



Figura 4. Unidad de control de Sinamics G120.

DATOS TÉCNICOS	
Modelo	CU250S-2 PN
Comunicación	Profibus, Profinet, Can, RS485
Dimensiones	196x73x165 mm
E/S Digitales	6 ED (3ED F) 3 SD (1 SD F) 5 ED
E/S Analógicas	2EA 2SA

Tabla 3. Datos técnicos de Unidad de control.

Panel operador: Permite configurar y visualizar los parámetros del variador de frecuencia (ver Figura 5).



Figura 5. Panel operador de Sinamics G120.

3.1.1.3 Simatic Iot2040 Siemens

El dispositivo de última generación enfocado a la Industria 4.0, básicamente permite un almacenamiento, proceso y transferencia de información en campo industrial (ver Figura 6). Digitaliza los datos de sensores, actuadores u otros dispositivos, almacenando toda la información en la nube, permitiendo la

visualización en tiempo real mediante plataformas web [25]. Los parámetros técnicos se indican en la Tabla 4.



Figura 6. Simatic Iot2040.

DATOS TÉCNICOS	
Procesador	Intel Quark x1020
Memoria	1GB RAM
Comunicación	2xProfinet,2Xrs232/485
Lenguaje	Yocto Linux
Ampliación	Arduino y MiniPCIe

Tabla 4. Datos técnicos de IoT2040.

3.1.1.4 Raspberry Pi 3

Es un miniordenador de bajo costo, que funciona con un sistema operativo abierto (ver Figura 7). Se aplica en sistemas tales como, estaciones de temperatura, automatización de casa, incluso automatización de procesos industriales bajo protección [26]. En el presente proyecto el dispositivo Raspberry Pi 3 es utilizado como procesador para detectar los colores amarillo y verde de la fruta, a través de una cámara y por medio de comunicación Profinet envié los datos al PLC s7-1200. Las especificaciones se detallan en la tabla 5.



Figura 7. Raspberry Pi 3

ESPECIFICACIONES TÉCNICAS
Broadcom BCM2837B0, CorteC-A53 (ARMv8) SoC de 64 bits a 1,4 GHz
SDRAM LPDDR2 de 1 GB
LAN inalámbrica 2.4GHz y 5GHz, Bluetooth 4.2, BLE
Profinet Gigabit sobre USB 2.0 (rendimiento máximo de 300 Mbps)
Cabecera GPIO de 40 pines extendida
HDMI de tamaño completo
4 puertos USB 2.0
Puerto de cámara CSI para conectar una cámara Raspberry Pi
Puerto de pantalla DSI para conectar una pantalla táctil de Raspberry Pi
Salida estéreo de 4 polos y puerto de video compuesto
Puerto microSD para cargar su sistema operativo y almacenar datos
Entrada de corriente continua de 5V / 2.5A

Tabla 5. Especificaciones técnicas de Raspberry Pi 3.

3.1.1.5 Cámara Raspberry Pi

El dispositivo llamado Camara Raspberry Pi permite capturar las señales de imagen o video, y enviarlas al dispositivo electrónico Raspberry Pi 3, mediante la conexión al puerto CSI utilizando un bus de datos de 15 pines (ver Figura 8). Además, ofrece una resolución de 5 megapíxeles, mide 25x20x9mm y pesa 3g, trabaja ligado al procesador gráfico del módulo Raspberry Pi 3 y consume poca batería. Aplicado a proyectos desde vigilancia hasta visión artificial [27].



Figura 8. Módulo de cámara Raspberry Pi.

3.1.1.6 Prototipo de banda transportadora y sistema de clasificación

Un componente del proyecto es la banda transportadora, es controlada por el motor conectado al Sinamics G120 o variador de frecuencia. Sus medidas son 100cm de largo, 20.5cm de ancho y 90cm de alto. Este prototipo se encarga de transportar los mangos para que el sistema clasificador pueda ejercer su función (ver Figura 9).

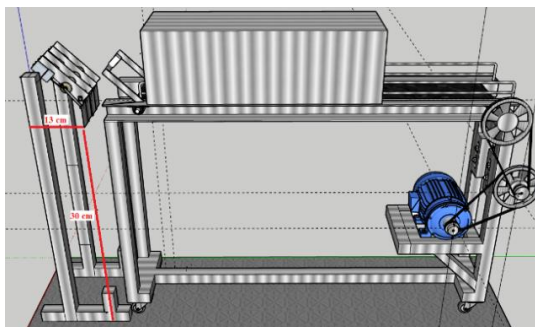


Figura 9. Diseño de banda transportadora y clasificadora de mangos en SketchUp.

3.1.1.7 Motores DC

En el presente proyecto se utilizan cuatro motores dc, para manipular los brazos que clasifican los mangos. Se alimentan a 12V y tiene una velocidad de 260 RPM (ver Figura 10).



Figura 10. Motor DC.

3.1.1.8 Motor trifásico Siemens

El Motor trifásico encargado de accionar la banda transportadora, conectado directamente al variador de frecuencia Sinamics G120 (ver Figura 11), el cual proporciona los parámetros necesarios para su funcionamiento. Los detalles técnicos del Motor se indican en la tabla 6.



Figura 11. Motor trifásico Siemens.

ESPECIFICACIONES TÉCNICAS	
Voltaje de alimentación	220 V
Frecuencia	60 Hz
Potencia	1 HP
Corriente	2.60/1.49 A
r/min	1735

Tabla 6. Especificaciones técnicas de Motor Siemens.

3.1.1.9 Pantalla HMI Delta

La pantalla HMI da la interfaz entre el sistema de clasificación de mangos y el operador, permitiendo el control y monitoreo de dicho proceso (ver Figura 12). Es programado mediante su propio software, y se comunica con el PLC s7-1200 para vincular las variables y permitir su funcionalidad utilizando el protocolo Profinet. En la tabla 7 se especifican sus principales características.



Figura 12. Pantalla HMI Delta.

HMI Delta	
Modelo	DOP-B03E211
Alimentación	24 V DC 2/300mA
Comunicación	Profinet y USB
Dimensiones	480x272 65536 colores
Software	DOPSoft

Tabla 7. Datos técnicos de HMI Delta.

3.1.2 Componentes lógicos

En la presente sección se detalla los diferentes softwares, que se utilizan para el diseño, configuración, control y comunicación de datos o parámetros, necesarios para el desarrollo del proyecto.

3.1.2.1 TIA Portal V14

Software que admite configurar y programar los distintos controladores de marca Siemens encargados de procesos industriales, además de realizar la programación de diversos parámetros a nivel industrial en el campo de la automatización (ver Figura 13) [28]. Permite tres tipos de lenguajes de programación:

- KOP (lenguaje de escalera) se basa en un lenguaje gráfico.
- FUP (diagrama de funciones) lenguaje con símbolos gráficos de algebra booleana.

- SCL (control estructurado) basado en texto con alto nivel. En la programación del PLC s7-1200 se utilizó el lenguaje de escalera.

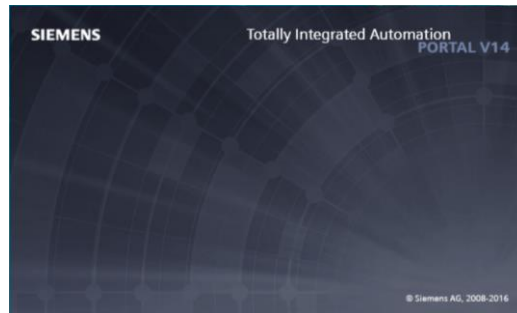


Figura 13. Software Siemens TIA Portal V14.

3.1.2.2 Raspbian Stretch

Sistema operativo gratuito y exclusivo para la plataforma de Raspberry Pi, es desarrollado en Linux, e incluye un escritorio que permite la interacción con el usuario, ofreciendo un óptimo funcionamiento [29] (ver Figura 14).

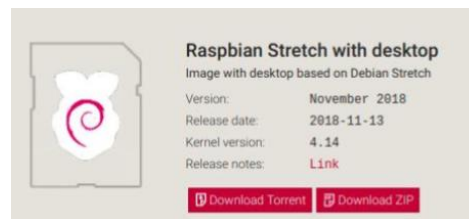


Figura 14. Sistema operativo Raspbian Stretch.

3.1.2.3 Python 3.0

El Lenguaje Python es de alto nivel, utiliza comandos que permiten al usuario desarrollar sus algoritmos de manera sencilla y rápida. Es un software gratuito, utilizable para sistemas operativos Linux, Windows y OS X [30] (ver Figura 15).



Figura 15. Lenguaje de programación Python.

3.1.2.4 OpenCV

La librería OpenCV es de código abierto, aplicado especialmente en visión artificial, dispone una extensa variedad de funciones que ayudan al procesamiento y reconocimiento de imágenes en tiempo real. Actualmente, esta librería es usada en lenguajes como Python, Matlab, Java entre otros (ver Figura 16) [31].



Figura 16. Librería OpenCV.

3.1.2.5 Snap7 Python

La biblioteca Snap7 permite la comunicación Profinet e interacción entre el Raspberry Pi 3 y el PLC s7-1200. Admite la transmisión/recepción de datos en tiempo real con alta velocidad en procesos industriales y es compatible con Python [32].

3.1.2.6 PyCharm

El software PyCharm definido como un *Integrated DeveLopment Environment* (IDE) muy completo para el lenguaje de programación Python; su interfaz es sencilla con un entorno editor inteligente y funciona con sistemas operativos tales como Linux, Windows y Mac [33] (ver Figura 17).



Figura 17. Software Pycharm.

3.1.2.7 WinSCP

Básicamente es un administrador de archivos con protocolo *Secure File Transfer Protocol* (SFTP), su objetivo principal es el intercambio de archivos entre una PC local y una remota. Permite editar, crear, mover, duplicar y eliminar elementos del Raspberry Pi 3 [34] (ver Figura 18).



Figura 18. Software WinSCP.

3.1.2.8 Wonderware Intouch

Software que permite el monitoreo y control de procesos de producción, cuyo objetivo es aumentar productividad y disminuir costos. Este software es comúnmente usado en el sector industrial, además admite el diseño de un sistema de supervisión, control y adquisición de datos facilitando la toma de decisiones en tiempo real [35] (ver Figura 19). La comunicación entre Intouch y el PLC s7-1200 es mediante el servidor OPC KEPServer.



Figura 19. Wonderware Intouch.

3.1.2.9 Kepserverex 6

La plataforma Kepserverex actúa como servidor a través del estándar OPC y empleando los protocolos de comunicación enfocados a TI, para facilitar a los usuarios datos industriales y lograr comunicar al Wonderware Intouch con el PLC

s7-1200, con el objetivo permitir el monitoreo y control de dispositivos conectados en procesos industriales y proporcionar al usuario los datos en tiempo real [36].

3.1.2.10 DOPSoft

Software perteneciente a la compañía Delta, permite programar las pantallas HMI de la serie Delta de una manera fácil, a través de un editor compatible con Windows con la finalidad de desarrollar una interfaz gráfica, que monitorea y controla los procesos industriales (ver Figura 20).

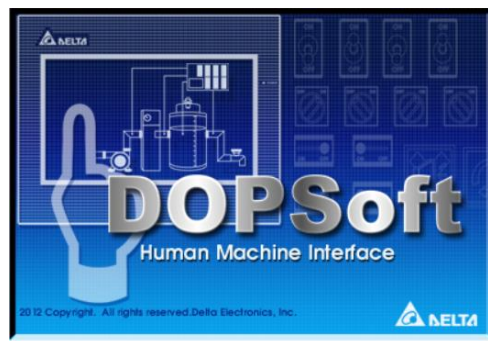


Figura 20. Software DOPSoft.

3.1.2.11 Node-Red

La plataforma node-red su interfaz se edita a través de un navegador web, utiliza un lenguaje de programación gráfica de nodos, que permiten la comunicación entre el PLC s7-1200, Iot2040 y plataforma web (ver Figura 21). Mediante diversos nodos, tales como, el S7 realiza la lectura/escritura de datos al PLC s7-1200 y el nodo Dweetio el envío de datos a la plataforma web determinada.

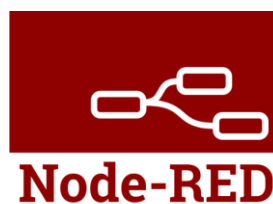


Figura 21. Nodered.

3.1.2.12 Freeboard

La plataforma web Freeboard permite visualizar los datos de los procesos, que son enviados a la red, además muestra la información en tiempo real de diferentes

equipos conectados. El flujo de datos recibidos desde Node-Red, a través del nodo Dweetio y son finalmente visualizados mediante widgets configurados (ver Figura 22).

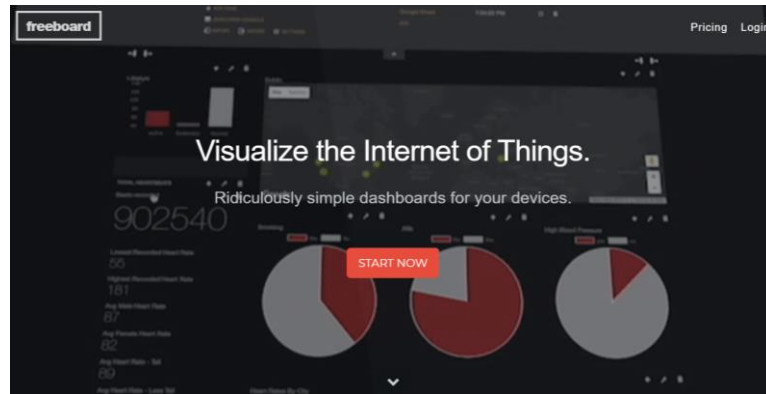


Figura 22. Herramienta Freeboard.

3.1.2.13 Putty

Programa de código abierto que establece la conexión mediante *Secure SHell* (SSH), permitiendo el acceso remoto al dispositivo Simatic Iot2040, para instalar los diferentes complementos o paquetes para el correcto funcionamiento del dispositivo IoT, estableciendo el monitoreo de la plataforma en tiempo real (ver Figura 23).

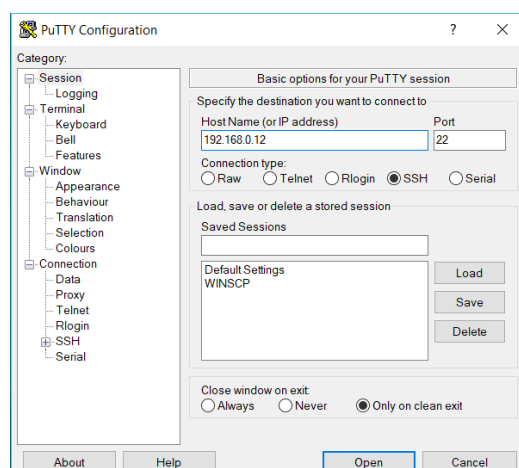


Figura 23. Software Putty.

3.2 DISEÑO DE LA PROPUESTA

3.2.1 Diagrama de red de comunicación del sistema

En todo el sistema se aplica el protocolo de comunicación Profinet, los equipos o dispositivos se encuentran comunicados a través de la red local 192.168.0.0/24, máscara de subred 255.255.255.0 y puerta de enlace 192.168.0.1, se asignaron las respectivas direcciones IP a los dispositivos (ver Figura 24).

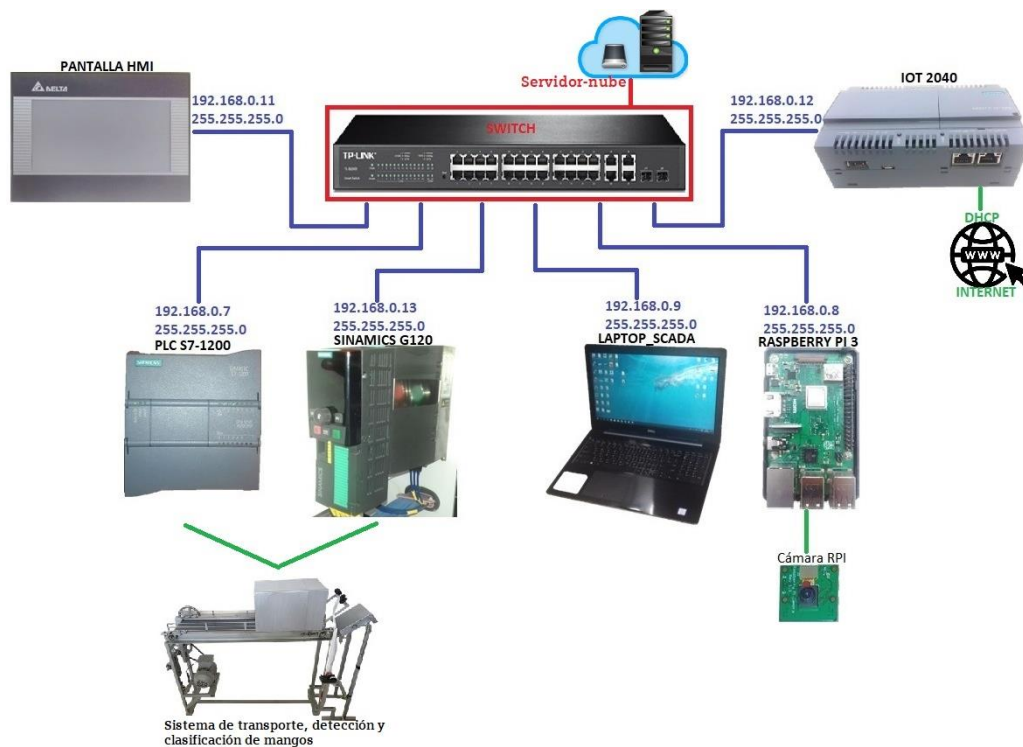


Figura 24. Diagrama de red de comunicación del sistema.

3.2.2 Diagrama de conexión de periféricos a la Raspberry Pi 3

Raspberry Pi 3 funciona realmente como una computadora, para ello, necesita la conexión de periféricos, tales como, monitor, mouse, teclado, cámara y alimentación, como se muestra en la Figura 25.

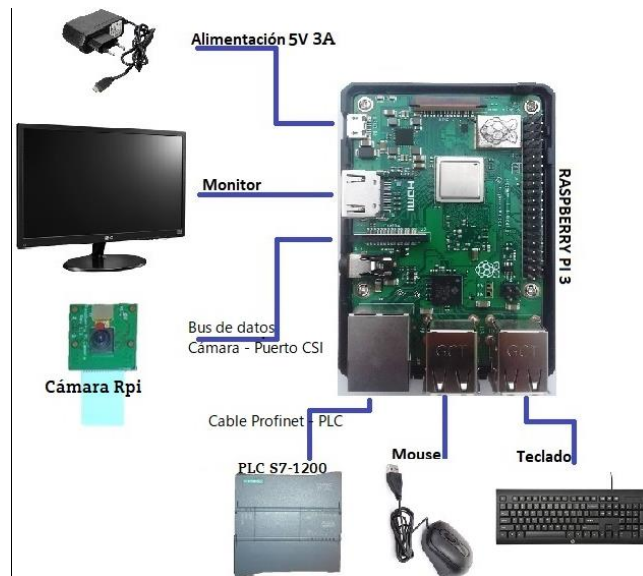


Figura 25. Diagrama de conexión de periféricos a la Raspberry Pi 3.

A través del monitor se indica la interfaz gráfica, donde se puede visualizar la ejecución de la programación, que permite la detección de colores de mangos. En el sistema operativo Raspbian Stretch se puede ingresar a la **Terminal de comandos**, se introduce el comando `cd proyecto` y luego `python DETECCION.py`, para ejecutar el archivo que permite la detección del sistema.

La conexión entre el PLC s7-1200 y la Raspberry Pi 3 se realiza mediante cable Profinet y la instalación de la respectiva biblioteca Snap7 que permite la comunicación entre dichos dispositivos.

La cámara Rpi se conecta mediante un cable bus de datos de 15 pines directamente al puerto CSI de la Raspberry Pi 3.

3.2.3 Conexiones del dispositivo IoT2040

El dispositivo Simatic IoT2040 se conecta a una alimentación de 12V, además en el puerto X1P1 se conecta el cable Profinet a la red local, en el segundo puerto X2P2 se conecta el cable Profinet con conectividad a internet y en la parte interior se coloca la tarjeta microSD en su respectiva ranura, (ver Figura 26).

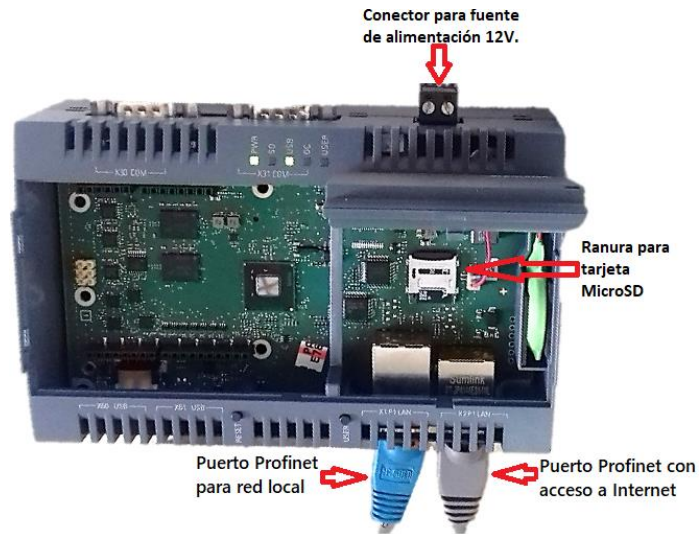


Figura 26. Conexiones de Simatic IoT2040.

3.2.4 Diseño del sistema electromecánico de clasificación de mangos

Sistema que permite realizar la clasificación de mangos, consta de cuatro motores dc controlados por el PLC s7-1200, dependiendo del estado de detección que proporciona la Raspberry Pi 3, al recibir las señales de detección de colores (ver Figura 27).

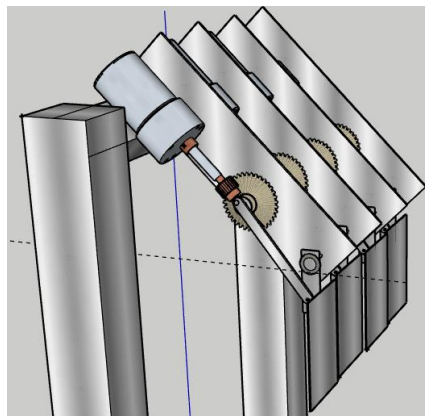


Figura 27. Sistema de clasificación diseñado en SketchUp.

3.2.5 Diagrama de conexiones del sistema de control de clasificación.

Los cuatro motores utilizados en el sistema de clasificación se encuentran a las salidas Q0.0, Q0.1, Q0.2, Q0.3 del PLC s7-1200, las cuales alimentan con 12V a cada uno de los motores, (ver Figura 28).

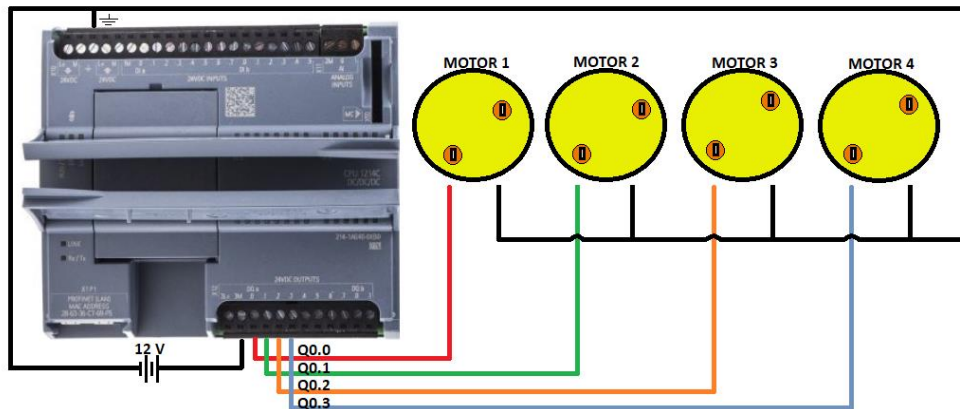


Figura 28. Diagrama de conexiones de motores DC.

3.2.6 Diseño del sistema de detección de mangos por visión artificial

El sistema de detección de mangos se realiza mediante visión artificial, consta de una cámara Rpi conectada directamente a la Raspberry Pi 3, permite la captura y procesamiento de video en tiempo real analizando los colores verde y amarillos de la fruta mediante la librería OpenCV y Python (ver Figura 29).

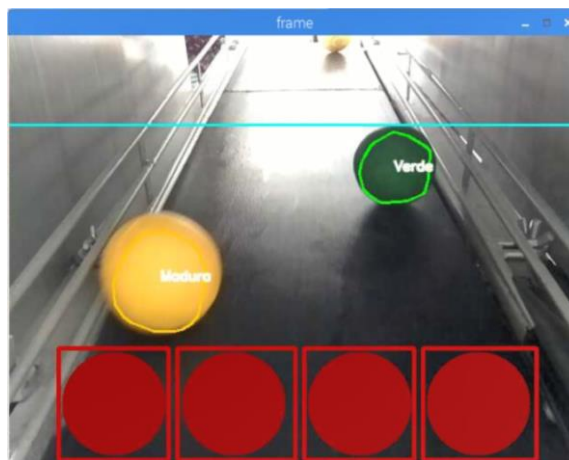


Figura 29. Detección de colores verde y amarillo.

A continuación se definen brevemente las principales funciones utilizadas en la codificación que permite la detección de colores:

- **PiCamera:** Paquete necesario para la cámara de Raspberry Pi, provee una interfaz de Python junto a los ajustes de configuración del módulo, tales como la resolución y velocidad de fotogramas o cuadros por segundos.
- **camera.capture_continuous:** Permite el acceso a la vista previa del video en tiempo real y de manera continua, usa un método que restituye un Frame

o cuadro por segundo (ver Figura 30) y contiene los siguientes tres parámetros:

- *RawCapture*, es el formato en el que se lee cada fotograma.
- *Format*, Se define que el formato de los canales de imagen sean BGR, tal como lo espera OpenCV.
- *Use_video_port*, parámetro que indica que la cámara está en modo video.

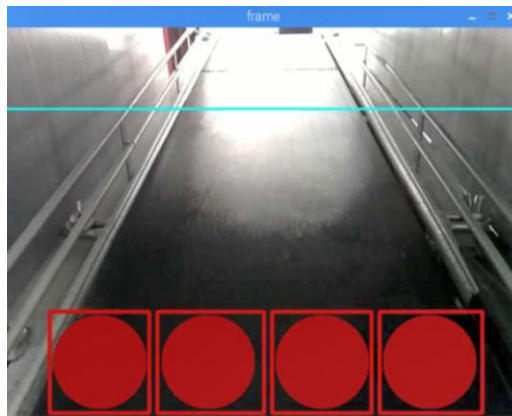


Figura 30. Vista del video en tiempo real.

- **cv2.cvtColor:** Función que establece el tipo de conversión de colores, contiene dos parámetros:
 - Definir el nombre de la imagen a convertir.
 - Parámetro que permite la conversión de colores de BGR→HSV (`cv2.COLOR_BGR2HSV`) (ver Figura 31).

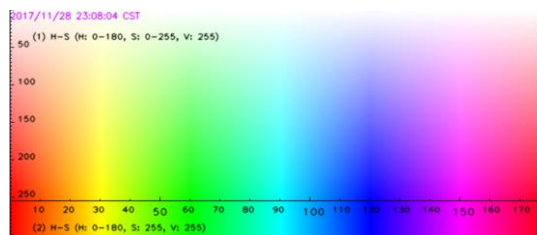


Figura 31. Rangos de tonos de colores HSV.

- **np.array:** Función que permite determinar el rango de colores a detectar.
- **cv2.rectangle:** Dibuja rectángulos (ver Figura 32) y sus parámetros son los siguientes:
 - Imagen en la cual se dibujan los rectángulos.
 - Ubicación de las líneas izquierda y arriba del rectángulo.

- Ubicación de las líneas de la derecha y abajo del rectángulo.
- Color del rectángulo dibujado.
- Grosor de las líneas del rectángulo.



Figura 32. Rectángulos producidos por la función `cv2.rectangle`

- **cv2.circle:** Dibuja círculos (ver Figura 33) y sus parámetros son los siguientes:
 - Imagen en la cual se dibujan los círculos.
 - Ubicación del centro del círculo a dibujar.
 - Radio del círculo a dibujar.
 - Grosor del contorno del círculo, si se requiere un círculo relleno se usa `-1`.



Figura 33. Círculos producidos por la función `cv2.circle`

- **cv2.moments:** Función que calcula todos los momentos de una gráfica detectada.
- **cv2.findContours:** Función que encuentra el contorno de una imagen, da puntos límites que permiten conectarse entre sí para formar un contorno más preciso. A continuación sus parámetros:
 - Nombre de la imagen a encontrar el contorno.
 - Modo de recuperación de contorno, se usa `CV_RETR_EXTERNAL` ya que solo se requiere obtener contornos de la parte exteriores de la imagen detectada.
 - Método de aproximación de contornos, `CV_CHAIN_APPROX_SIMPLE`, Comprime partes diagonales, verticales u horizontales y proporciona solo los puntos finales.

- **cv2.convexHull:** Después de encontrar los contornos, se utiliza esta función *convexHull* para encontrar el casco convexo de cada contorno, se especifica la variable que guarda los contornos contados.
- **cv2.drawContours:** Función que dibuja los contornos (ver Figura 34) y sus parámetros son los siguientes:
 - Imagen destino
 - Variable que almacena todos los contornos.
 - Se indica un contorno para dibujar.
 - Color de contornos.



Figura 34. Contorno producido por la función cv2.drawContours

- **cv2.putText:** Función utilizada para generar una cadena de texto sobre cualquier imagen (ver Figura 35) y sus parámetros son los siguientes:
 - Imagen sobre la cual se va a escribir el texto.
 - Texto a escribir en la imagen.
 - Coordenadas en la cual se va a escribir el texto.
 - Fuente para el texto, ejemplo: FONT_HERSHEY_SIMPLEX.
 - Escala de la fuente, esta se multiplica por el tamaño determinado de la fuente escogida.
 - Color del texto en espacio de color BGR (colores: *Blue*, *Green* y *Red*).
 - Grosor del texto.



Figura 35. Texto producido por la función putText.

Físicamente el sistema de detección de mangos está compuesta por una Cámara Rpi ubicada al final de la banda transportadora y permite variar el ángulo de posición para obtener una mejor detección (ver Figura 36).

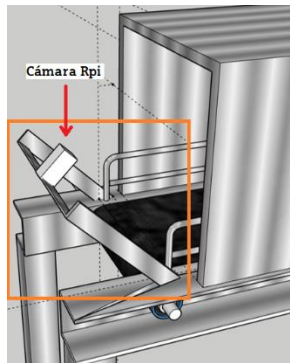


Figura 36. Sistema de detección diseñado en SketchUp.

3.2.7 Configuración de variador de frecuencia Sinamics G120

La configuración del Sinamics G120 se realiza en el software TIA Portal V14, mediante el acceso en línea de dispositivos accesibles en el rango de la red local se reconoce el dispositivo y se procede a realizar la debida configuración. Se asigna el nombre “SINAMICS_G120” y dirección IP “192.168.0.13” al dispositivo (ver Figura 37 y Figura 38).



Figura 37. Asignación del nombre al variador de frecuencia Sinamics G120.



Figura 38. Asignación de dirección IP al variador de frecuencia Sinamics G120.

Agregamos el variador al proyecto para lograr la comunicación con el controlador PLC s7-1200. Se busca el dispositivo en el **Catálogo Hardware** → **Otros dispositivo de campo** → **Profinet IO** → **Drives** → **SIEMENS AG** → **SINAMICS** → **SINAMICS G120 CU250S-2 PN VECTOR V4.7**. Finalmente se logra la comunicación Profinet del Sinamics G120 con el controlador PLC s7-1200 uniendo los puertos de cada dispositivo (ver Figura 39 y Figura 40).

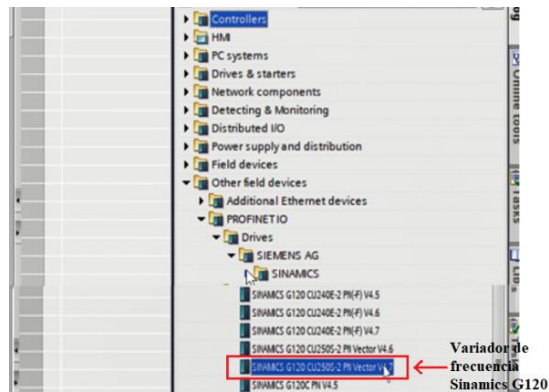


Figura 39. Selección del variador de frecuencia Sinamics G120.

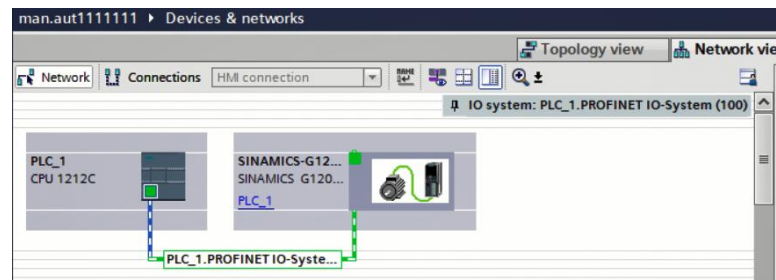


Figura 40. Comunicación Profinet del Sinamics G120 con el controlador PLC s7-1200.

En el TIA Portal se ingresa al dispositivo Sinamics G120 y se agrega el Telegrama Estándar en: **Catalogo Hardware** → **Otros dispositivo de campo** → **Profinet IO** → **Drives** → **SIEMENS AG** → **SINAMICS** → **Submodulos** → **Standard telegram 1, PZD-2/2** (ver Figura 41), el cual permite el acceso y uso de las palabras de mando del dispositivo Sinamics G120.

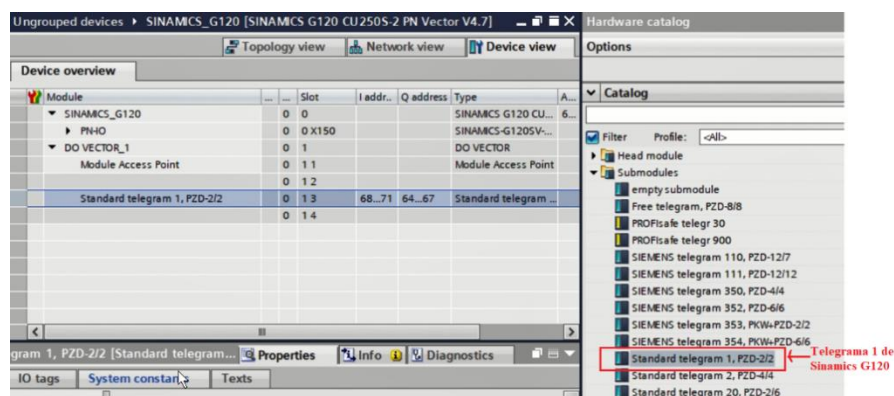


Figura 41. Selección del Telegrama estándar en Sinamics G120.

El telegrama Estándar 1, permite:

- Entradas de periferia (Enviar datos del PLC al variador):
 - Palabra de mando → QW64

- Consigna (Escritura) → QW66
- Salidas de periferia (Leer datos del variador y enviar al PLC):
 - Palabra de estado → IW68
 - Consigna lectura → IW70

El Telegrama Estándar 1 está constituido por 16 bits, cada bit tiene una función determinada y se indica en la siguiente tabla 8.

BIT	VALOR	DETALLES	VALOR DE COMANDO		
			Encender	Inversión de giro	Parada rápida
0	0	Motor se detiene con el tiempo de deceleración p1121 al llegar a la parada.	0	1	1
	1	El convertidor pasa al estado “Listo para servicio” con un flanco positivo, con el bit 3=1 adicional el convertidor conecta el motor.			
1	0	Desconectar inmediatamente el motor, se produce la parada natural.	1	1	1
	1	Conectar el motor.			
2	0	Parada rápida: El motor se detiene con el tiempo de deceleración OFF3 p1135 hasta la parada.	1	1	0
	1	Parada lenta.			
3	0	Desconectar inmediatamente el motor.	1	1	1
	1	Conectar el motor.			
4	0	La salida del generador de rampa se deshabilita (frenado rápido).	1	1	1
	1	Es posible la habilitación de generador de rampa.			
5	0	La salida del generador de rampa se detiene.	1	1	1
	1	Habilitar generador de rampa.			
6	0	El motor se detiene con el tiempo de deceleración p1121.	1	1	0
	1	El motor acelera con el tiempo de aceleración p1120 hasta alcanzar consigna.			
7	1	El fallo se confirma con un flanco positivo. Si todavía está presente la orden ON, el convertidor conmuta al estado “Bloqueo de conexión”.	0	0	0
8		No usado.	0	0	0
9		No usado.	0	0	0
10	0	Datos de proceso no válidos, se espera alguna señal.	1	1	1
	1	Mando vía bus campo, datos de proceso válidos.			
11	1	La consigna se invierte con el convertidor.	0	1	0
12		No usado.	0	0	0
13	1	Se aumenta la consigna almacenada en el potenciómetro motorizado.	0	0	0
14	1	Se reduce la consigna almacenada en el potenciómetro motorizado.	0	0	0
15	1	Conmutación entre ajustes para distintas interfaces de manejo.	0	0	0
			16#047E	16#0C7F	16#043B

Tabla 8. Configuración - Telegrama Estándar 1. Manual Sinamics G120.

De acuerdo a la configuración del Telegrama Estándar 1, los valores hexadecimales que se envían a la palabra de consigna son: 16#047E, 16#0C7F y 16#043B correspondientes a las funciones de “Encendido del motor”, “Inversión de giro” y “Parada rápida”.

3.2.8 Configuración, diagrama lógico y programación de PLC s7-1200 en TIA Portal

La configuración del PLC s7-1200 se realiza en el software TIA Portal V14, mediante el acceso en línea de dispositivos accesibles en el rango de la red local, se reconoce el dispositivo y se procede a configurar. Se asigna al PLC s7-1200 la dirección IP “192.168.0.7” (ver Figura 42).

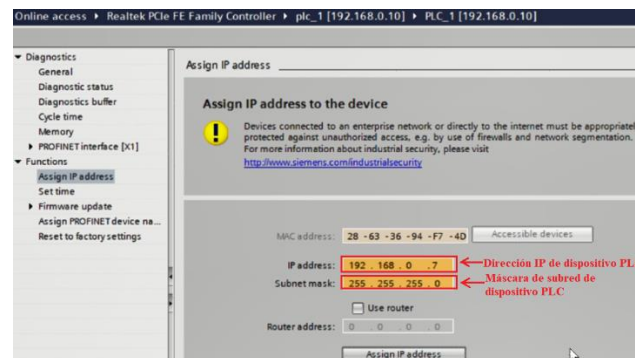


Figura 42. Asignación de dirección IP al PLC s7-1200.

Una propiedad importante es habilitar el PUT/GET en la opción de protección (ver Figura 43), la cual permite el acceso para la comunicación entre el PLC s7-1200, HMI y OPC.



Figura 43. Habilitar opción PUT/GET en configuración de PLC s7-1200.

Posterior, se realiza la declaración de variables a usar en la programación en lenguaje de escalera, para ello se le asigna un nombre, el tipo de dato y la dirección (ver Figura 44).

man.aut11111111 ▶ PLC_1 [CPU 1212C AC/DC/Rly] ▶ PLC tags ▶ Tabla de variables estándar [58]									
Tabla de variables estándar									
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment	
1	DETECTOR_VERDE_1	Bool	%Q0.0						
2	DETECTOR_VERDE_2	Bool	%Q0.1						
3	DETECTOR_VERDE_3	Bool	%Q0.2						
4	DETECTOR_VERDE_4	Bool	%Q0.3						
5	RESET1	Bool	%M7.6						
6	DETECTOR_MADURO_1	Bool	%Q0.4						
7	DETECTOR_MADURO_2	Bool	%Q0.5						
8	DETECTOR_MADURO_3	Bool	%Q0.6						
9	DETECTOR_MADURO_4	Bool	%Q0.7						
10	PRODUCCION_ALCANZADA_VE...	Bool	%M5.7						
11	PRODUCCION_ALCANZADA_M...	Bool	%M5.6						
12	ENCENDER_APAGAR_MOTOR_B...	Bool	%M7.5						
13	PALABRA_DE_MANDO	Word	%QW64						
14	tiempoverdes	Time	%MD105						
15	tiempomaduros	Time	%MD110						
16	CONSIGNA_ESCRITURA_VELOC...	Int	%QW66						
17	INGRESO_VELOCIDAD_RPM	Int	%QW68						
18	VELOCIDADX2	Real	%MD69						
19	INGRESO_VELOCIDAD_RPM_REAL	Real	%MD70						
20	FRECUENCIA	Int	%QW74						

Figura 44. Declaración de variables.

Se muestra un diagrama lógico de flujo del proceso de clasificación de mangos en función de su color (ver Figura 45).

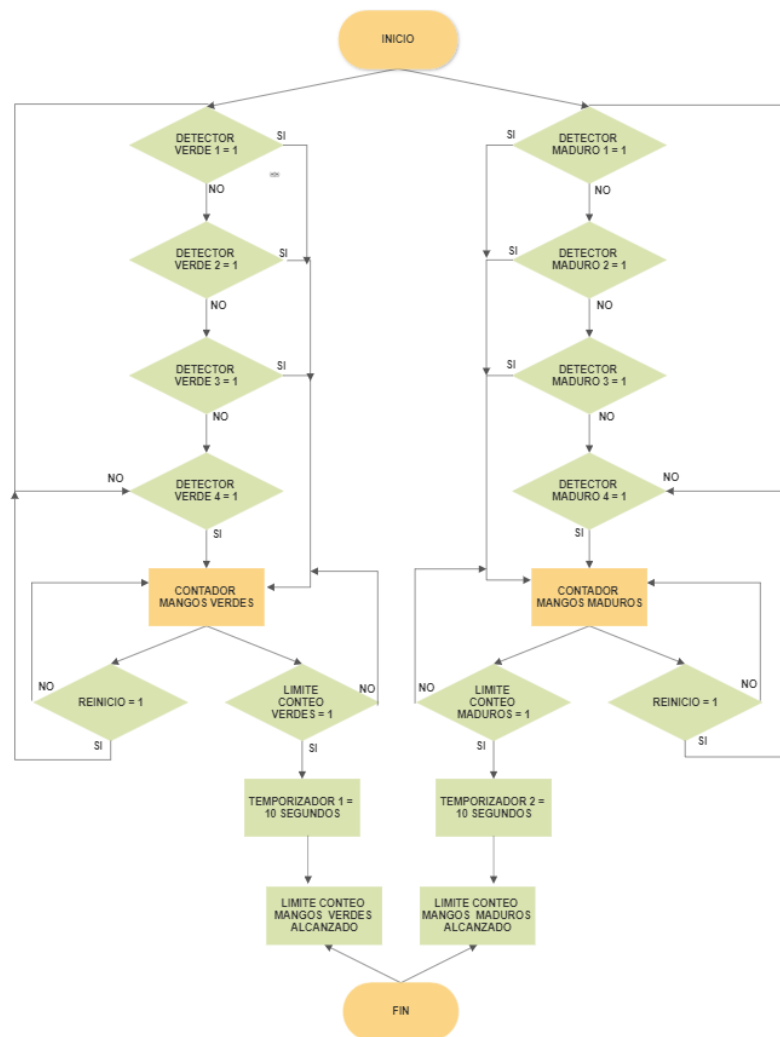


Figura 45. Diagrama lógico del proceso de clasificación de mangos.

A continuación, se indica la programación en el lenguaje de escalera, que permite el funcionamiento del sistema de clasificación de mangos (ver Figura 46 y Figura 47).

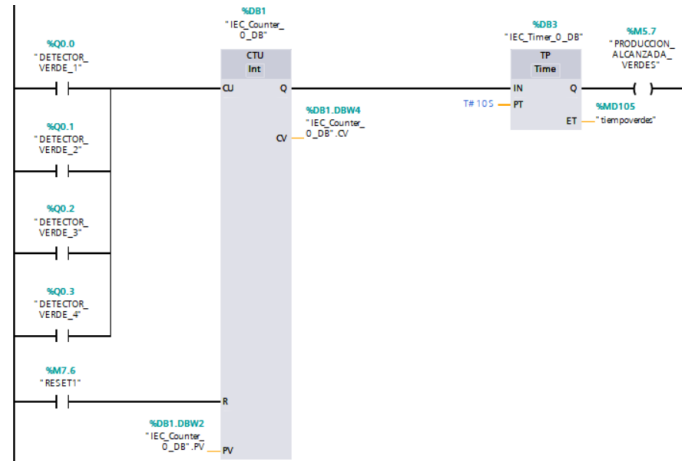


Figura 46. Conteo, producción estimada y producción alcanzada de mangos verdes.

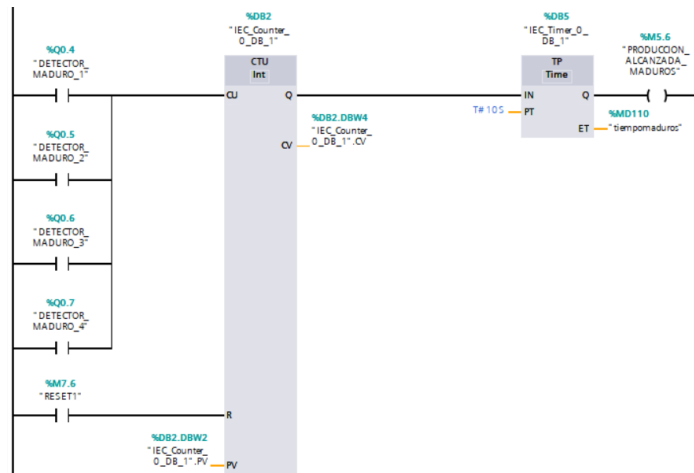


Figura 47. Conteo, producción estimada y producción alcanzada de mangos maduros.

Se muestra un diagrama lógico de flujo de acuerdo a la programación de encendido, apagado e inversión de giro del motor de la banda transportadora para el proceso de clasificación de mangos en función de su color (ver Figura 48).

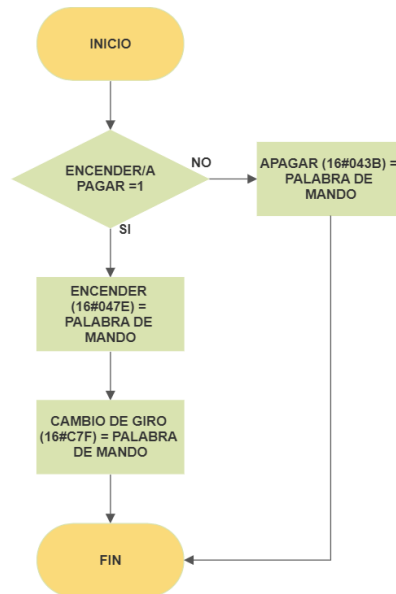


Figura 48. Diagrama lógico del encendido, apagado e inversion de giro del motor. Para el encendido y apagado del motor de la banda transportadora a través del Sinamics G120, basandose en el telegrama 1 de la configuracion del dispositivo, al activar (**M7.5**) se enciende y se invierte el giro mediante la palabra de mando ENCENDER e INVERSION DE GIRO que en hexadecimal son 16#047E y 16#C7F, respectivamente. Además, son asignadas las palabras de mando al registro **QW64:P**. Para parar el motor se realiza mediante la palabra de mando PARADA RAPIDA, que en hexadecimal es 16#043B asignada al registro **QW64:P** (ver Figura 49).

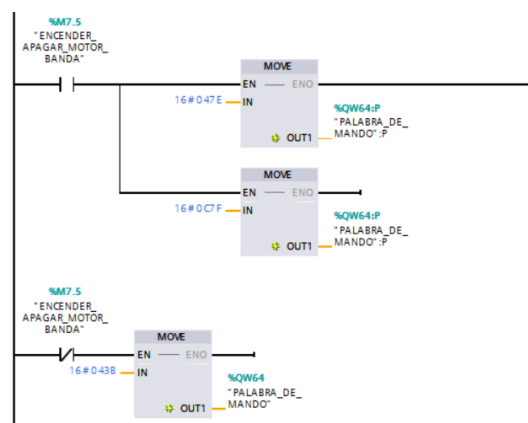


Figura 49. Encendido y apagado del motor de la banda transportadora.

A continuación, se muestra un diagrama lógico de flujo de acuerdo a la programación del ingreso de la velocidad del motor de la banda transportadora para el proceso de clasificación de mangos en función de su color (ver Figura 50).

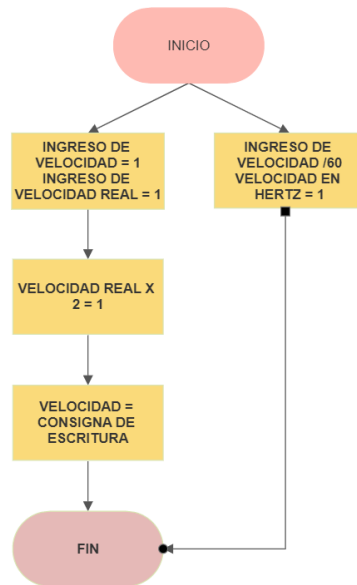


Figura 50. Diagrama lógico del ingreso de velocidad del motor.

Mediante el registro **QW68** se realiza el ingreso de la velocidad, la unidad de medida es de 0 a 2000rpm, el cual necesita ser convertido a una escala de 0 a 16384, para ello se realiza una conversión que es asignada en el registro **QW66**, de igual manera se realiza una operación que permite obtener el valor en Hz (ver Figura 51).

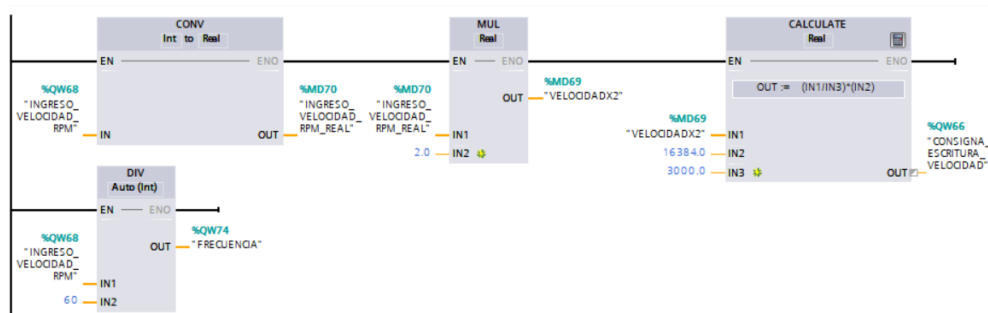


Figura 51. Ingreso de velocidad de motor en Rpm y conversión a Hz.

3.2.9 Diseño de la interfaz HMI.

En el desarrollo de la interfaz gráfica se utiliza el software DOPSoft de Delta, para realizar la comunicación basado en el protocolo Profinet entre el HMI y PLC s7-1200, se ingresa manualmente en la pantalla HMI la dirección IP: 192.168.0.11 y la máscara: 255.255.255.0 (ver Figura 52).

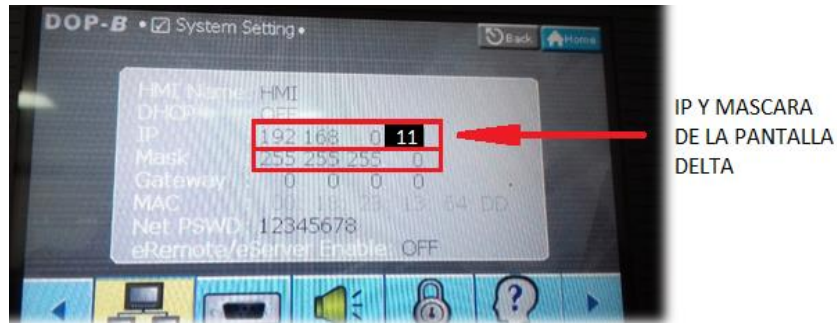


Figura 52. Configuración manual de IP y mascara de pantalla Delta.

Para iniciar la configuración en el software DOPSoft, se escoge el respectivo modelo de la pantalla Delta especificada en el datasheet del dispositivo, el cual es DOP-B03E211 (ver Figura 53).

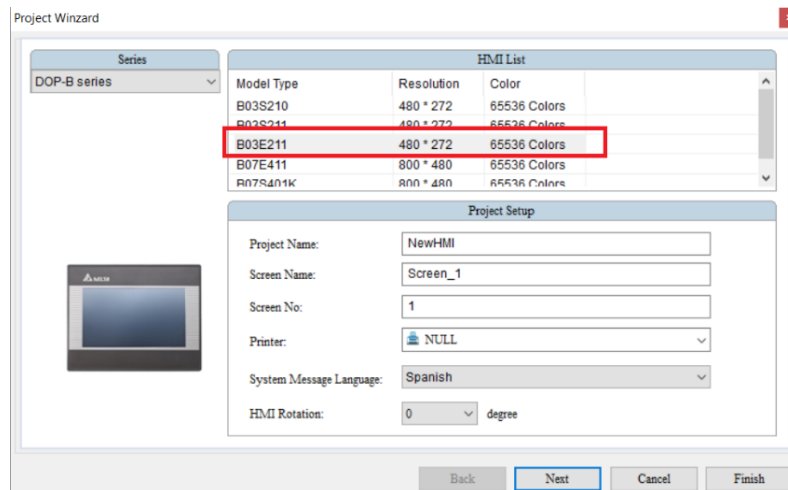


Figura 53. Selección de modelo de pantalla Delta.

Después de seleccionar el modelo de la pantalla Delta, es necesario definir la comunicación Profinet con las direcciones IP respectivas al controlador PLC s7-1200 y a la pantalla HMI (ver Figura 54 y Figura 55).

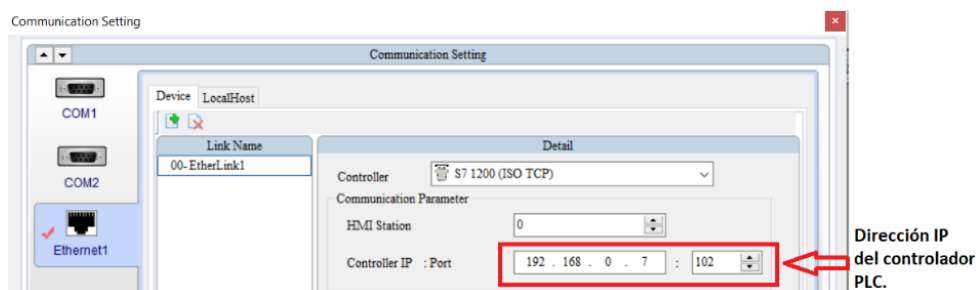


Figura 54. Configuración de comunicación.

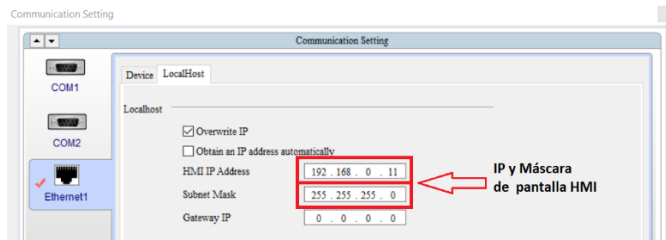


Figura 55. Asignación de dirección IP y Máscara de la pantalla HMI.

Se muestra un diagrama de bloques de acuerdo al diseño de la interfaz para el proceso de clasificación de mangos en función de su color (ver Figura 56).

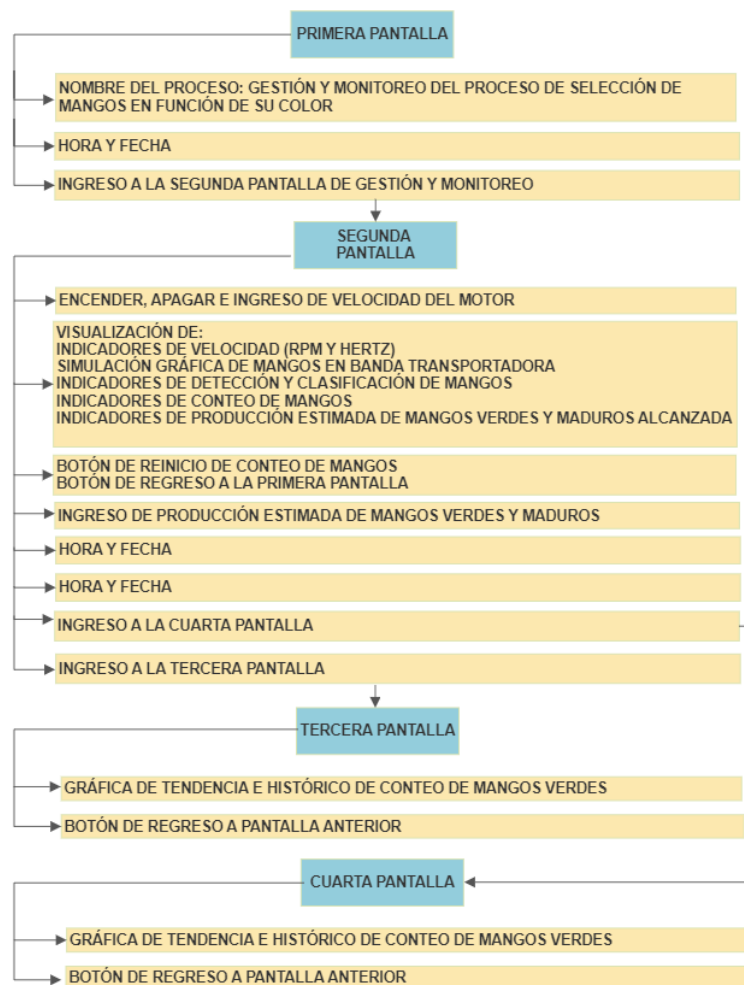


Figura 56. Diagrama de bloques de acuerdo al diseño de la interfaz HMI.

La interfaz gráfica del sistema de clasificación de mangos está compuesta por cuatro pantallas. La primera pantalla muestra el nombre del proceso, hora y fecha, también permite el ingreso al monitoreo y control del proceso (ver Figura 57).



Figura 57. Pantalla de ingreso al sistema.

En la segunda pantalla se realiza el control y monitoreo del proceso de clasificación de mangos (ver Figura 58). El usuario tiene el control para ingresar la velocidad del motor de la banda transportadora, encender o apagar mediante un interruptor gráfico tipo palanca, y visualizar los valores de velocidad en rpm y Hz del variador Sinamics G120. Además, se visualiza una simulación gráfica de mangos sobre la banda transportadora, la programación puede observarse en el Anexo 6. También se visualiza en dos indicadores la detección de mangos verdes y amarillos, de igual forma cuando los actuadores son accionados para realizar la clasificación los mangos. Existen dos indicadores en los cuales se encuentran asignadas las variables de conteo que permiten visualizar la cantidad de mangos verdes y maduros, incluye también un botón que permite manualmente reiniciar el conteo. Finalmente, se permite ingresar valores estimados de producción de mangos verdes y maduros a alcanzar, cuando la producción llega al valor predeterminado, éste envía una alarma mediante el encendido de un led y mostrando en pantalla un mensaje de “Producción alcanzada”.

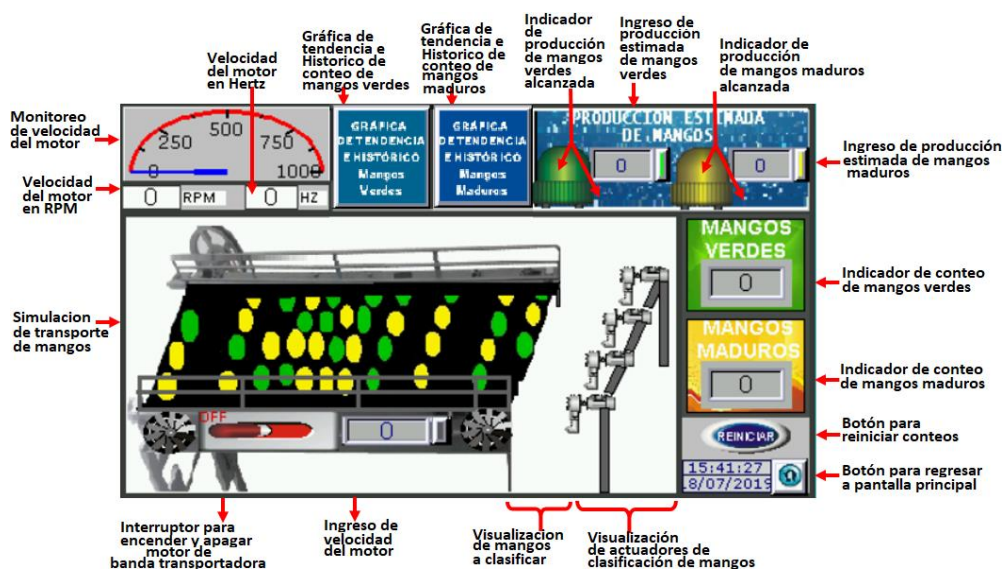


Figura 58. Pantalla de control y monitoreo del sistema.

La tercera y cuarta pantalla corresponden a las gráficas de tendencias e históricos de conteo de mangos verdes y maduros (ver Figura 59 y Figura 60).

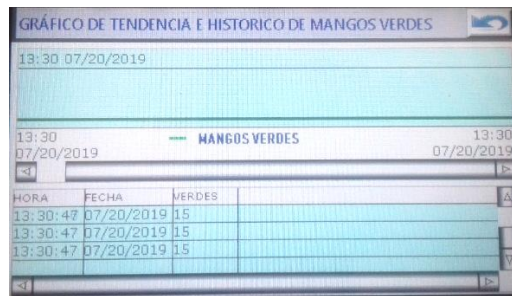


Figura 59. Gráfica de tendencia e historial del conteo de mangos verdes.

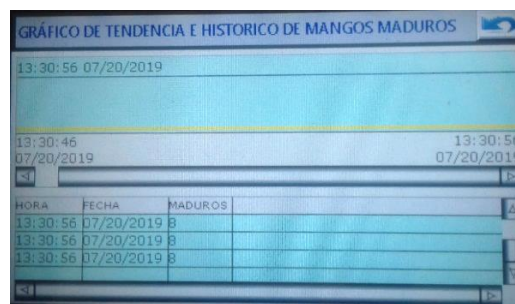


Figura 60. Grafica de tendencia e historial del conteo de mangos maduros.

Por último, cada pantalla contiene botones de navegación entre las diferentes pantallas, como también para regresar al menú principal.

3.2.10 Diseño de interfaz SCADA

La comunicación entre el software Wonderware Intouch y el PLC s7-1200 se realiza mediante el servidor Opc KepServerEx6.0, a continuación se procede a la configuración del mismo.

3.2.10.1 Configuración de Kepsverex 6.0

En la configuración, primero se agrega un nuevo canal de comunicación, se debe ubicar en *Connectivity*, y luego en *New Channel* se ingresa los siguientes parámetros (ver Figura 61):

- **Nombre del canal:** Canal del sistema
- **Tipo de comunicación:** Siemens TCP/IP Profinet
- **Adaptador de red:** Default.

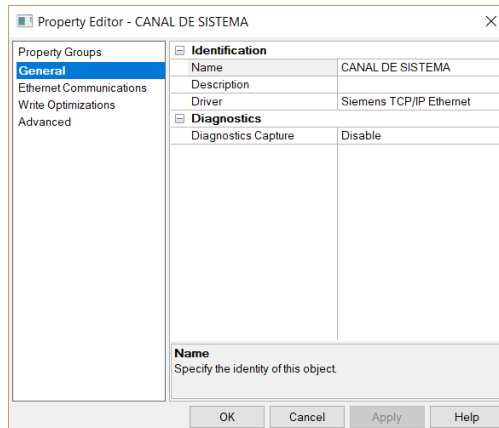


Figura 61. Propiedades del canal de comunicación.

Luego se agrega un nuevo dispositivo, se debe ubicar en el canal ya creado “CANAL DEL SISTEMA”, luego en *New Device* se ingresa los siguientes parámetros (Ver Figura 62):

- **Nombre del dispositivo:** SISTEMA
- **Nombre del canal asignado:** CANAL DEL SISTEMA
- **Tipo de comunicación:** Siemens TCP/IP Profinet
- **Modelo del controlador:** S7-1200
- **Dirección IP del controlador:** 192.168.0.7

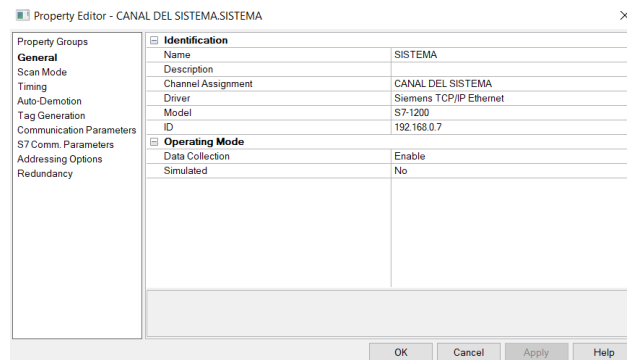


Figura 62. Propiedades del dispositivo.

Después se agregan las variables necesarias para la interfaz SCADA, se debe ubicar en el dispositivo ya creado “SISTEMA”, luego en *New Tag* se ingresa los siguientes parámetros (ver Figura 63):

- **Nombre de variable:** “Decisión del usuario”
- **Dirección y tipo de variable:** “Misma dirección de variables del controlador PLC”.

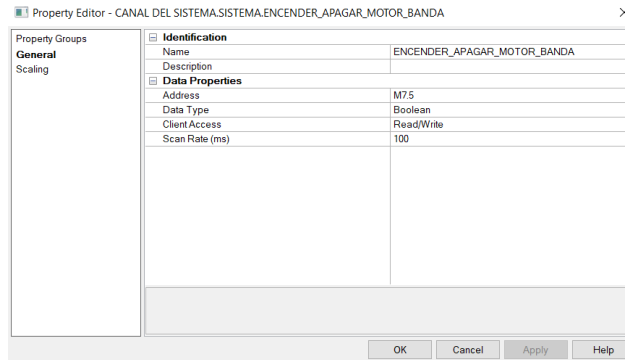


Figura 63. Propiedades de variable creada.

Y por último, se crea el Alias que permite la comunicación entre el Wonderware Intouch y el PLC s7-1200. Se debe ubicar en *Aliases*, luego en *New Alias* se ingresa los siguientes parámetros (ver Figura 64):

- **Nombre de Alias:** comunicacionprofinet
- **Localización del alias:** CANAL DEL SISTEMA.sistema.

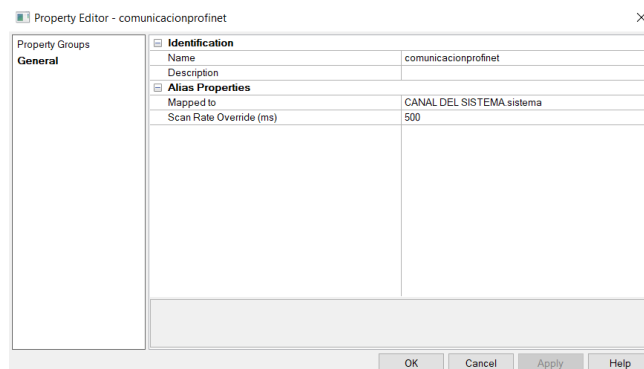


Figura 64. Propiedades del Alias de comunicación.

3.2.10.2 Configuración y diseño en Wonderware Intouch

Se realiza la configuración que permite la comunicación entre el Wonderware Intouch y el PLC s7-1200, se trata del **Access Name**, al crear el proyecto se accede en *Special*, luego en *Access Name* y finalmente ingresamos los datos (ver Figura 65):

- **Access:** comunicación
- **Node Name:** localhost
- **Application Name:** server_runtime
- **Topic Name:** comunicacionprofinet.

Se debe tener en consideración el Topic Name debe ser igual al Nombre de Alias configurado en KepServerEx.

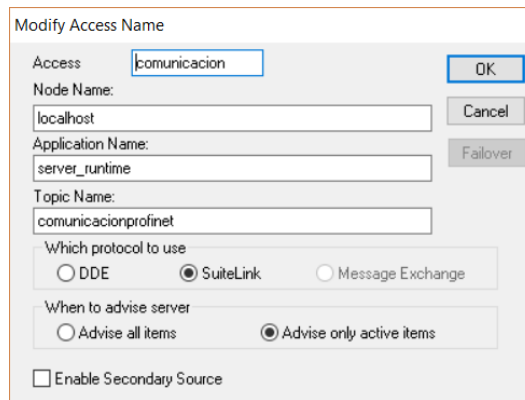


Figura 65. Propiedades de Access Name.

Para agregar variables o tags se realiza la configuración en el Tag Dictionary, se accede en *Special* y luego en *Tag Dictionary*, en la nueva ventana ingresar el Nombre de la variable o **Tagname**, **Tipo de variable**, **Access Name** (Se escoge el creado anteriormente) y la dirección de la variable o **Item** (Misma dirección de variables del controlador PLC s7-1200 (ver Figura 66).

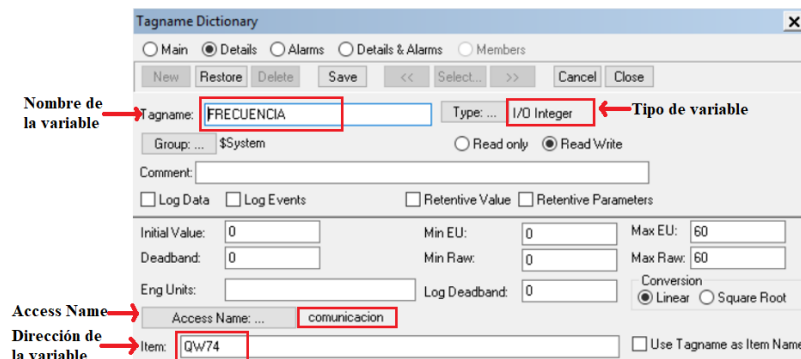


Figura 66. Configuración de Tagname Dictionary.

La interfaz gráfica del sistema SCADA permite el monitoreo y control del proceso desde una computadora, tiene una pantalla principal que muestra el nombre del proceso, hora y fecha (ver Figura 67). Además, permite el acceso al control y monitoreo mediante un usuario y contraseña ya definido. La programación puede observarse en el Anexo 7.



Figura 67. Pantalla de Inicio de la interfaz del sistema SCADA.

En la figura 68 se muestra la pantalla que permite el control y monitoreo el proceso del Sistema SCADA, en el cual el operador tiene el control de encender, apagar el motor de la banda transportadora e ingresar o variar la velocidad, además de visualizar los valores de dicha velocidad en rpm y Hz.

En la misma pantalla se observa la simulación gráfica de mangos sobre la banda transportadora, la programación puede observarse en el Anexo 8, también se visualiza el mango verde o maduro cada vez que pase por el sistema de clasificación, así mismo se visualiza cuando el actuador acciona para realizar la clasificación los mangos, todo en tiempo real.

Mediante un indicador se muestra el conteo de mangos verdes y maduros, consta de un botón que permite el reinicio de dicho conteo y una gráfica de tendencia en tiempo real, también un botón que permite regresar a la pantalla principal, además de mostrar la fecha y hora.

En la interfaz se ingresan valores estimados de producción de mangos verdes y maduros, cuando la producción haya llegado al valor antes definido, éste envía una alarma, mostrándola en el histórico de alarmas.

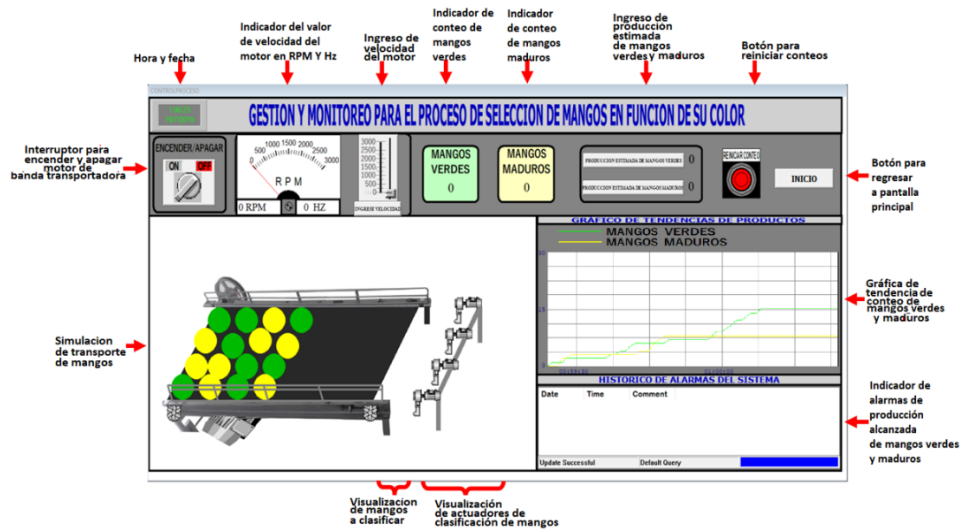


Figura 68. Pantalla del control y monitoreo del proceso.

3.2.11 Configuración del dispositivo IoT2040

Mediante este dispositivo nos enfocamos a la Industria 4.0, para realizar el almacenamiento y transferencia de datos del proceso en tiempo real se realiza la siguiente configuración y programación.

Primero se descarga la imagen a instalar proporcionada en la página oficial de Siemens, necesariamente debe ser en una tarjeta microSD mayor a 8Gb. La instalación se realiza mediante el software Win32 Disk Imager (ver Figura 69), luego de la debida instalación se coloca la tarjeta microSD en la ranura correspondiente del Iot2040 para su correcto funcionamiento.

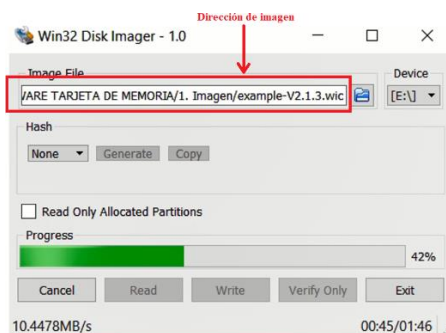


Figura 69. Instalación de imagen en tarjeta microSD.

Se enciende el dispositivo IoT2040 y nos guiamos de su funcionamiento por medio de sus indicadores leds; el primer led enciende al instante, éste indica la correcta alimentación del dispositivo, el segundo led enciende luego de unos segundos e indica el arranque del IoT, el tercer led enciende de manera intermitente mientras

se procesa el arranque del SO instalado en la tarjeta microSD y cuando este indicador se apague el dispositivo está listo para ser usado.

El dispositivo IoT2040 tiene por defecto la dirección IP 192.168.200.1, para su acceso por medio de la PC, este debe tener configurada una dirección IP en el rango 192.168.200.2 – 192.168.200.254. La dirección IP configurada del dispositivo IoT2040 es la 192.168.200.9, después se accede a través del software Putty, se configura una conexión SSH, ingresa la dirección IP 192.168.200.1 (ver Figura 70).

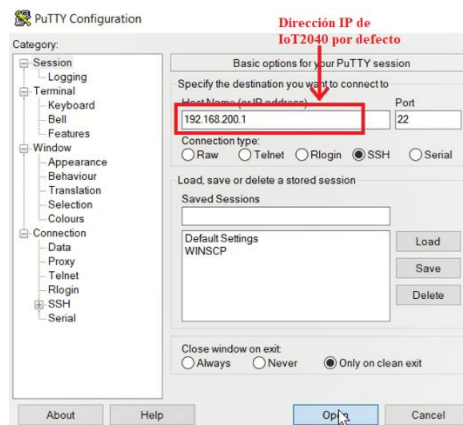


Figura 70. Acceso al IoT2040 a través del software Putty.

Al iniciar la sesión del dispositivo, se ingresa el **Login as** que por defecto es **root**, y se procede a cambiar la dirección IP del dispositivo, una dirección que se encuentre en la misma red de trabajo del sistema general, para esta configuración se ingresa el comando **iot2000setup** (ver Figura 71) y se configura la dirección IP 192.168.0.12 (ver Figura 72).

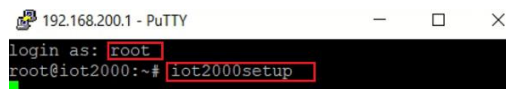


Figura 71. Inicio de sesión del dispositivo IoT2040.

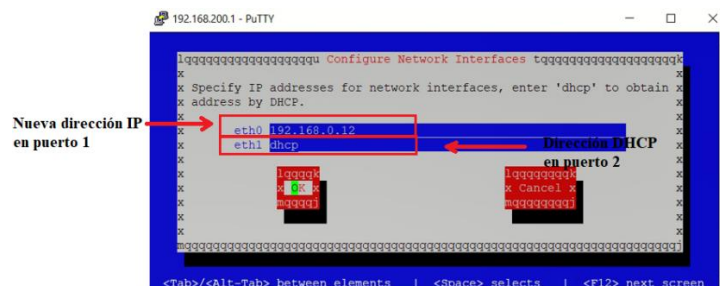


Figura 72. Configuración de dirección IP del IoT2040.

Finalmente, se configura en la PC una dirección IP que se encuentre en red con el dispositivo IoT2040 para una correcta comunicación, por ejemplo la dirección IP 192.168.0.9 (ver Figura 73).

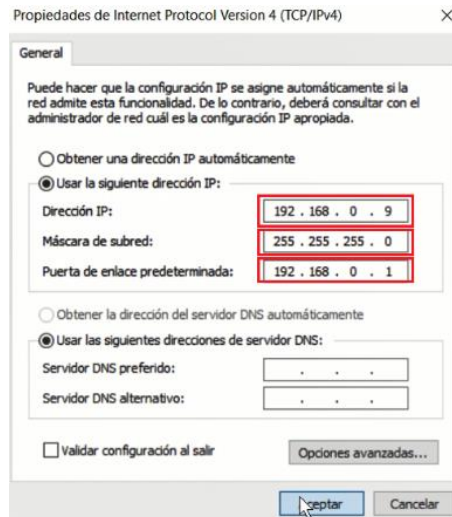


Figura 73. Configuración de dirección IP del computador.

3.2.12 Programación de Plataforma Node Red

Con el software Putty ingresamos al IoT2040 de manera normal y al iniciar sesión ingresamos el Login as: root. Iniciamos la plataforma Node Red ingresando el comando **node /usr/lib/node_modules/node-red/red &** (ver Figura 74).

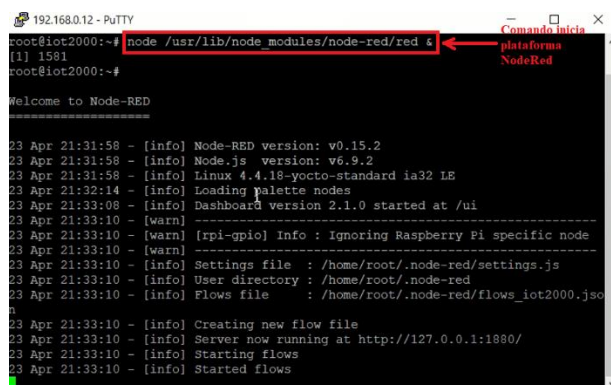


Figura 74. Inicialización de plataforma Node Red.

Se accede al editor de Node Red a través de un navegador web ingresando la dirección 192.168.0.12:1880 (ver Figura 75).

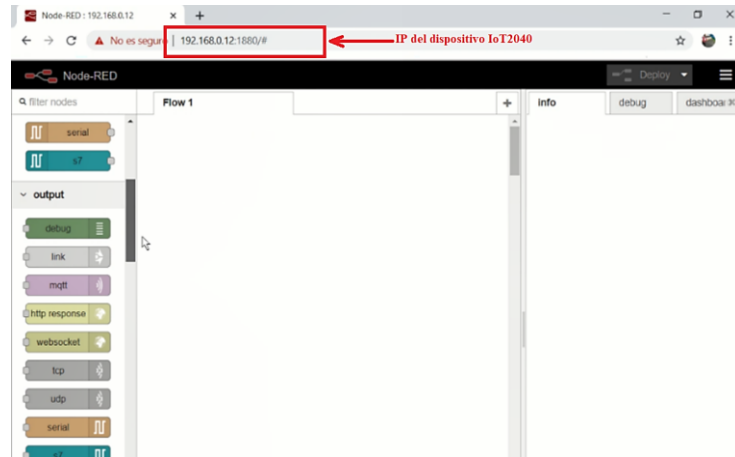


Figura 75. Plataforma Node Red.

La herramienta cuenta con nodos de programación básicos, en este caso se requiere del **Nodo S7** que permite la lectura o escritura de datos del PLC s7-1200 y el **Nodo Dweetio**, que permite enviar y recibir datos a la plataforma Freeboard. Los nodos antes mencionados no se encuentran instalados, para ello en el mismo sitio se tiene la opción *Manage palette* para buscar los nodos e instalarlos (ver Figura 76).

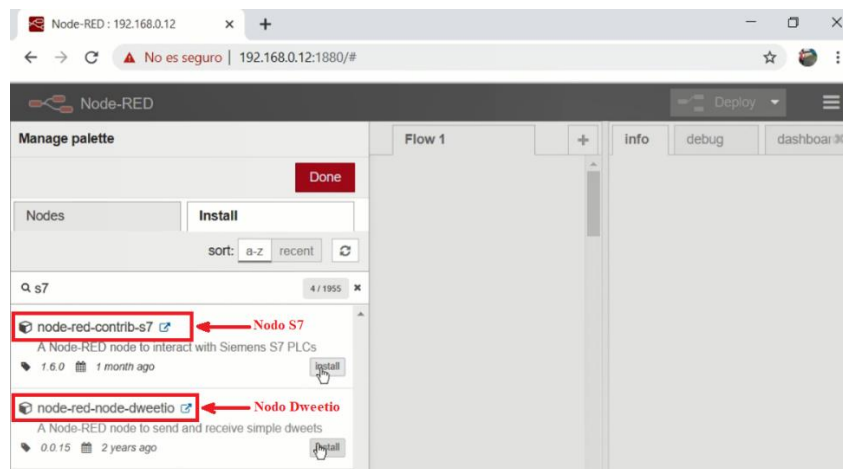


Figura 76. Instalación de nodos S7 y Dweetio.

La conexión se configura a través del nodo S7, es el encargado de leer los datos del PLC s7-1200 (ver Figura 77). Para la conexión se ingresan los siguientes datos:

- **Dirección IP:** 192.168.0.7
- **Modo:** Rack/Slot
- **Número de Rack:** 0
- **Número de Slot:** 1.

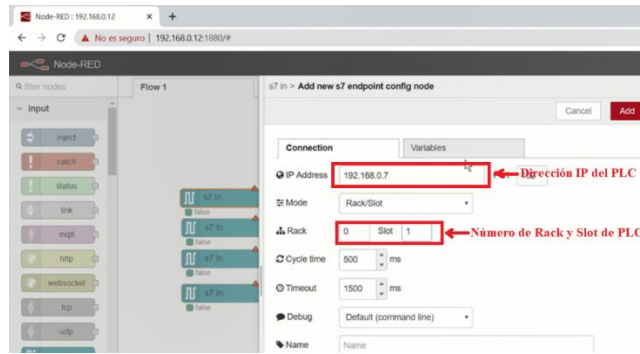


Figura 77. Configuración de datos del PLC s7-1200 para el nodo S7.

Además, se debe ingresar las variables que va a leer (ver Figura 78), las direcciones de las variables deben ser las mismas declaradas antes en la programación del PLC s7-1200 en TIA Portal (ver Figura 79).

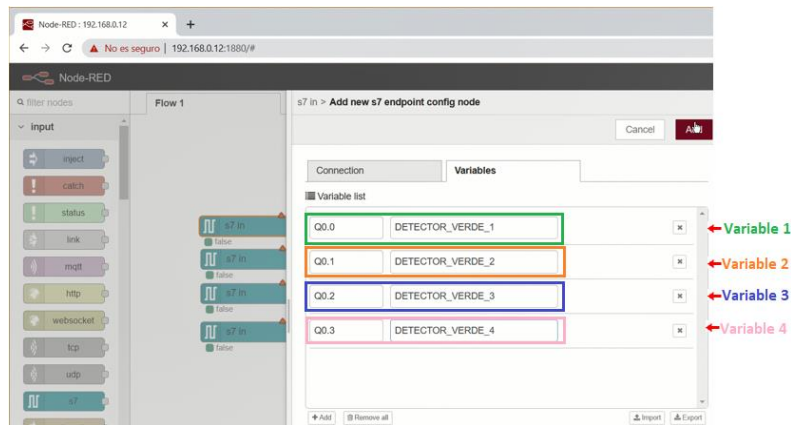


Figura 78. Asignación de variables para el nodo S7.

man.aut1111111 > PLC_1 [CPU 1212C A/DC/Rly] > PLC tags > Tabla de variables estándar [66]

Name	Data type	Address	Retain	Access	Write	Visibl.	Comment
DETECTOR_VERDE_1	Bool	%Q0.0					
m1	Bool	%M0.0					
DETECTOR_VERDE_2	Bool	%Q0.1					
m2	Bool	%M0.1					
DETECTOR_VERDE_3	Bool	%Q0.2					
m3	Bool	%M0.2					
DETECTOR_VERDE_4	Bool	%Q0.3					
m4	Bool	%M0.3					
RESET1	Bool	%M7.6					
DETECTOR_MADURO_1	Bool	%Q0.4					
m5	Bool	%M0.4					
DETECTOR_MADURO_2	Bool	%Q0.5					
m6	Bool	%M0.5					

Figura 79. Variables pertenecientes a la programación del PLC s7-1200 en TIA Portal.

Por último, se debe editar la configuración del nodo S7 con los parámetros de conexión del PLC s7-1200, con cada una de las variables predeterminadas (ver Figura 80).

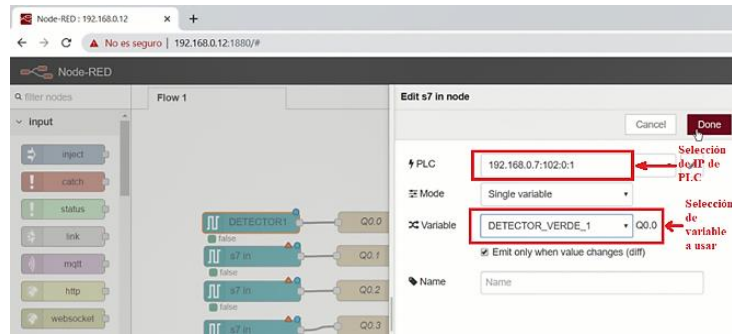


Figura 80. Selección de variables para el nodo S7.

El nodo Dweetio cumple la función de enviar datos leídos por el nodo S7 a la plataforma web Freeboard para su monitoreo (ver Figura 81), al nodo Dweetio se le asigna dos datos:

- **Thing:** DETECTOR_VERDE_1 (esta variable debe ser igual en la configuración de Freeboard).
- **Name:** Q0.0 (el nombre puede variar).

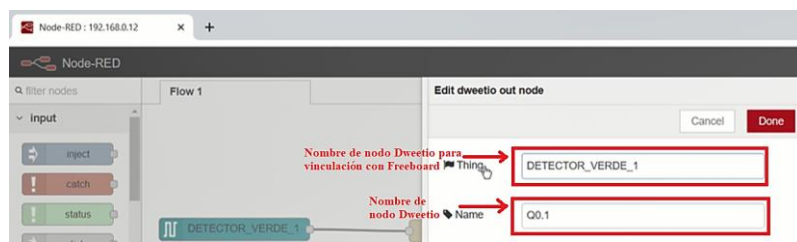


Figura 81. Configuración del nodo Dweetio.

En la Figura 82 se muestra la programación de los nodos del sistema de clasificación de mangos. Los nodos s7 adquieren los siguientes datos referentes al sistema:

- Detector de mangos.
- Conteo de mangos.
- Valor de producción estimada de mangos.
- Producción estimada de mangos alcanzada.
- Encendido y apagado del motor de la banda transportadora.
- Velocidad ingresada al motor de la banda transportadora.
- Valores de velocidad del motor de la banda transportadora en Hz y Rpm.

Después de adquirir los datos, estos son enviados a través de los nodos Dweetio a una plataforma web para su visualización.

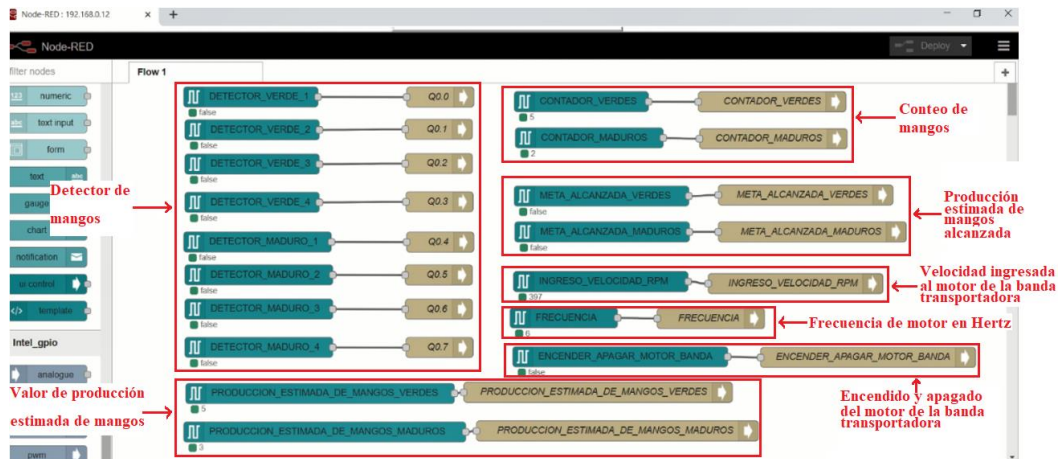


Figura 82. Programación de nodos del sistema de clasificación de mangos.

3.2.13 Plataforma web Freeboard.io

La visualización de datos referentes al sistema de clasificación de mangos se realiza a través de la plataforma web Freeboard.io, adquiere los datos del Node Red a través del nodo Dweetio, recibe los datos y los muestra en la plataforma web mediante widgets; para acceder a esta plataforma el usuario debe tener registrada una cuenta.

Al realizar un nuevo proyecto se deben añadir las fuentes de datos que son adquiridas de las variables configuradas en Nodered. En la sección de *DATA SOURCES* en la pestaña *ADD*, luego escogemos la opción Dweet.io que permite la comunicación con el nodo de Nodered y agregamos el *Nombre* de la variable con el *THING NAME*, este debe ser igual a la variable designada en el nodo Dweetio del Nodered (ver Figura 83). Ya agregado podemos comprobar su conexión, observando la fuente de datos junto a la hora (ver Figura 84).

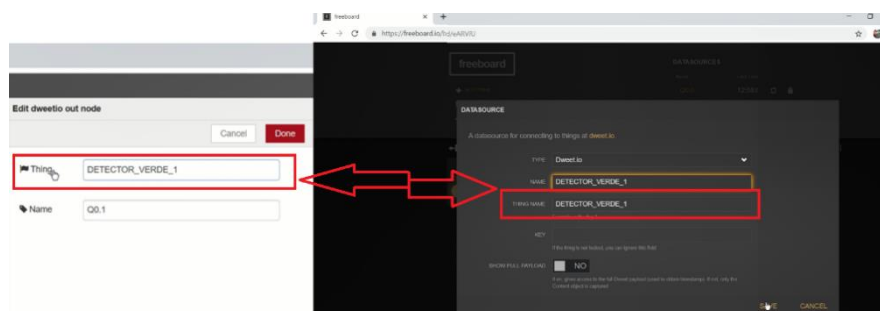


Figura 83. Configuración de fuentes de datos en Freeboard.

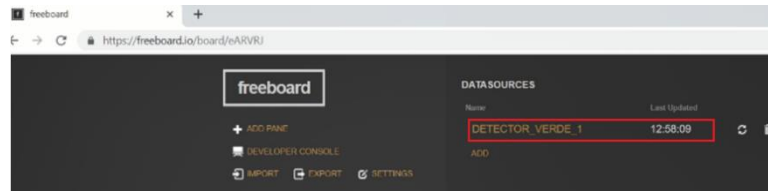


Figura 84. Conexión de la fuente de datos.

Lo siguiente es agregar los widgets que permiten la visualización en la plataforma, en *ADD PANE*, se muestran los widgets disponibles y se selecciona el necesario, en este caso el *Indicador Light*.

Al seleccionar el widget *Indicator Light*, se ingresa el *Nombre* y el *Datasource* respectivo a la fuente de datos creada anteriormente junto al *Payload* (ver Figura 85). Ya creado el widget podemos observar su funcionamiento (ver Figura 86).

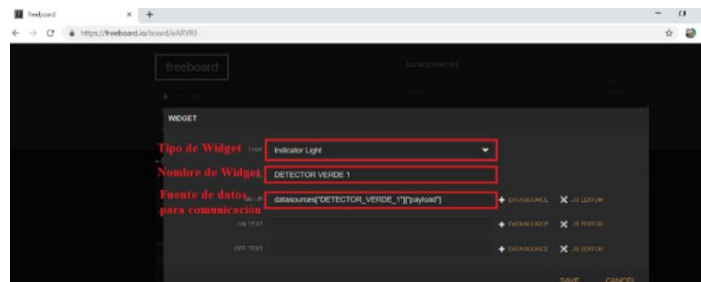


Figura 85. Configuración de widgets en Freeboard.io.



Figura 86. Funcionamiento del widget Indicator Light en Freeboard.io.

La plataforma de Freeboard perteneciente al proceso de selección de mangos está compuesta por una pantalla que permite visualizar (ver Figura 87):

- Indicadores de cada detector de mangos.
- Indicadores de encendido y apagado del motor de la banda transportadora.
- Indicador de producción estimada de mangos alcanzada.
- Valores del conteo de mangos.
- Valores de producción estimada de mangos.
- Valores de velocidad ingresada al motor de la banda transportadora.
- Valores de velocidad del motor de la banda transportadora en Hz y Rpm.

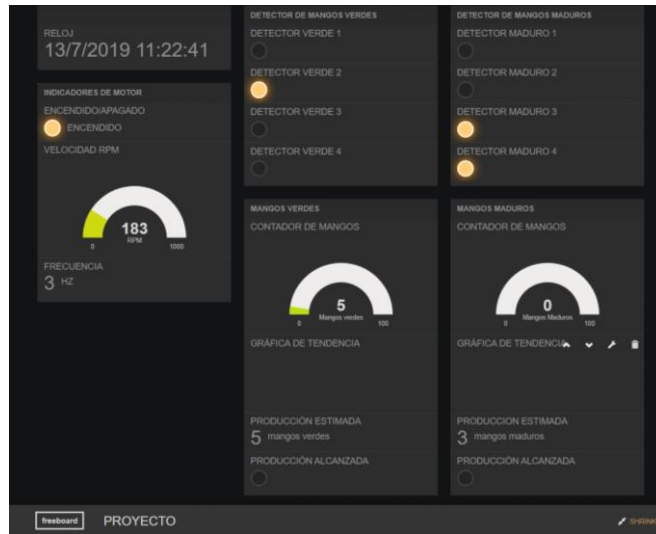


Figura 87. Plataforma de Freeboard perteneciente al proceso de selección de mangos.

3.3 ESTUDIO DE FACTIBILIDAD

3.3.1 Factibilidad técnica

Se utilizan equipos, tales como PLC s7-1200, pantalla HMI Delta, Variador de frecuencia Siemens Sinamics G120, Raspberry Pi3, Cámara Raspberry Pi, IoT2040, utilizando comunicación Profinet e I2C, además del prototipo de banda transportadora con el respectivo motor trifásico y el sistema de clasificación de mangos.

Se utiliza un Raspberry Pi 3 con su respectiva cámara, como solución de bajo costo que permite su aplicación a la automatización industrial, la programación adecuada que permite la detección de mangos de acuerdo a su color se realiza en el software OpenCV y Python.

La configuración y programación del PLC s7-1200 que permite el control del sistema de selección de mangos se realiza en lenguaje de escalera en el software TIA PORTAL v14.

Wonderware Intouch es el software en el cual se realiza el sistema SCADA, permite optimizar la interacción entre el sistema de automatización y el usuario en tiempo real, es sencillo de utilizar, además su comunicación es a través de un OPC llamado KEPServer que permite el envío y recepción de datos al sistema SCADA. También

se realiza un monitoreo y control del sistema a través de una Pantalla HMI delta B03S211 con comunicación Profinet.

En cuanto al enfoque del sistema a la industria 4.0, se utiliza la plataforma NodeRed y Freeboard.io que admite el almacenamiento de datos en su respectivo servidor por un bajo precio mensual.

Los equipos, dispositivos o software empleados en la implementación del proyecto se adquieren en el mercado nacional, mediante las pruebas que se le realizan al sistema se demuestra su sencilla operatividad y su correcto funcionamiento.

3.3.2 Análisis de costos del proyecto

Para determinar el costo de la implementación del sistema de gestión, monitoreo y digitalización para el proceso de selección de mangos en función de su color, con enfoque a la industria 4.0 se detallan los costos de equipos, licencias de software, mano de obra u otros elementos utilizados.

COSTOS DE EQUIPOS

CANT.	DESCRIPCIÓN	COSTO
1	MODULO DE SINAMICS G120 Y MOTOR TRIFÁSICO SIEMENS.	\$ 1482,00
1	SIMATIC IOT2040	\$ 415,80
1	PLC s7-1200 CPU 1212C AC/DC/relé	\$ 473,00
1	PANTALLA HMI DELTA DOP-B03E211	\$ 419,00
1	RASPBERRY PI 3	\$ 72,00
1	CAMARA RASPBERRY PI	\$ 25,00
1	PROTOTIPO DE BANDA TRANSPORTADORA	\$ 400,00
1	PROTOTIPO DE SISTEMA DE CLASIFICACIÓN	\$ 80,00
4	MOTORES DC	\$ 96,00
TOTAL		\$ 3462,80

Tabla 9. Costos de equipos para implementación del proyecto.

COSTOS DE LICENCIAS DE SOFTWARE

CANT.	DESCRIPCIÓN	COSTO
1	LICENCIA WONDERWARE INTOUCH	\$ 2500,00
1	LICENCIA KEPSERVEREX V6	\$ 438,00
1	ALMACENAMIENTO ANUAL EN PLATAFORMA FREEBOARD	\$ 264,00
TOTAL		\$ 3202,00

Tabla 10. Costos de licencias de software utilizados en el proyecto.

COSTOS DE OTROS ELEMENTOS

CANT.	DESCRIPCIÓN	COSTO
2	TARJETAS DE MEMORIA DE 16GB	\$ 24,00
	CABLE ELÉCTRICO #12	\$ 25,00
	CANALETAS	\$ 6,00
	CABLE FLEX PARA CÁMARA	\$ 10,00
	METROS DE CABLE UTP CAT 6	\$ 15,00
	VENTILADOR PARA RASPBERRY PI	\$ 9,00
	FUENTE DE ALIMENTACIÓN PARA RASPBERRY PI 3	\$ 9,00
TOTAL		\$ 98,00

Tabla 11. Costos de otros elementos del proyecto.

COSTO FINAL

DESCRIPCIÓN	COSTO
COSTOS DE EQUIPOS	\$ 3462,80
COSTOS DE LICENCIAS DE SOFTWARE	\$ 3202,00
COSTOS DE OTROS ELEMENTOS	\$ 98,00
COSTO DE MANO DE OBRA	\$ 1700,00
TOTAL	\$ 8462,80

Tabla 12. Costos finales del proyecto.

3.4 PRUEBAS Y RESULTADOS

Pruebas y resultados de la red de comunicación del sistema

Se realiza la asignación de direcciones IP para la correcta comunicación de los equipos o dispositivos usados en el sistema, como se indica en la tabla 13.

EQUIPO O DISPOSITIVO	DIRECCIONES IP
PLC S7-1200	192.168.0.7
RASPBERRY PI 3	192.168.0.8
PC-SCADA	192.168.0.9
PANTALLA HMI	192.168.0.11
IOT 2040	192.168.0.12
SINAMICS G120	192.168.0.13

Tabla 13. Direcciones IP de la red del sistema.

Para comprobar la comunicación entre los diferentes equipos o dispositivos se utiliza el comando PING en la consola de CMD desde la PC (ver Figura 88 a Figura 92). Se logra verificar el envío y respuesta de paquete de datos con su respectivo tiempo.


```
Símbolo del sistema - PING 192.168.0.11 -t
C:\Users\DELL>PING 192.168.0.11 -t

Haciendo ping a 192.168.0.11 con 32 bytes de datos:
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo=2ms TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo=1ms TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo=1ms TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo<1m TTL=100
Respuesta desde 192.168.0.11: bytes=32 tiempo=1ms TTL=100
Estadísticas de ping para 192.168.0.11:
    Paquetes: enviados = 36, recibidos = 36, perdidos = 0
              (0% perdidos)
    Tiempos aproximados de ida y vuelta en milisegundos:
              Mínimo = 0ms, Máximo = 3ms, Media = 0ms
```

Figura 91. PING 192.168.0.7 -t, correspondiente a Pantalla HMI.

```
Símbolo del sistema - PING 192.168.0.12 -t
C:\Users\DELL>PING 192.168.0.12 -t

Haciendo ping a 192.168.0.12 con 32 bytes de datos:
Respuesta desde 192.168.0.12: bytes=32 tiempo=2ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=6ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=3ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.12: bytes=32 tiempo=6ms TTL=64
Estadísticas de ping para 192.168.0.12:
    Paquetes: enviados = 35, recibidos = 35, perdidos = 0
              (0% perdidos)
    Tiempos aproximados de ida y vuelta en milisegundos:
              Mínimo = 1ms, Máximo = 6ms, Media = 1ms
```

Figura 92. Ping 192.168.0.7 -t, correspondiente al IoT2040.

Pruebas y resultados del sistema de detección de colores de mangos.

Al realizar las pruebas del sistema de detección de colores de mangos aplicando visión artificial mediante librerías de OpenCV y Python, se comprueba la correcta detección de colores: verde y amarillo (ver Figura 93 a Figura 96).

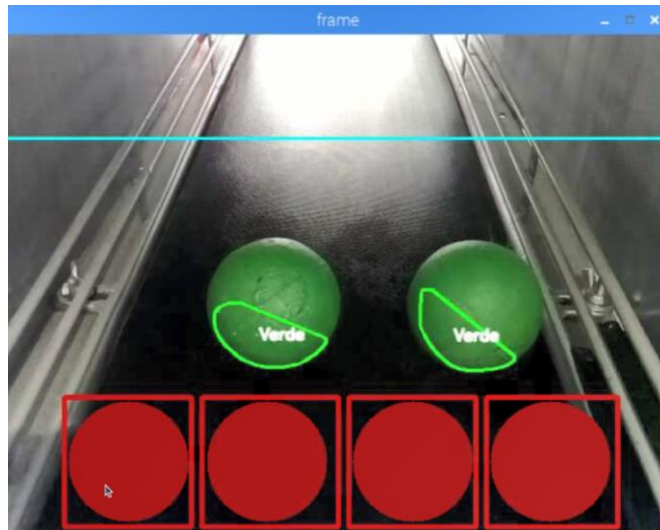


Figura 93. Detección del color verde de mangos.

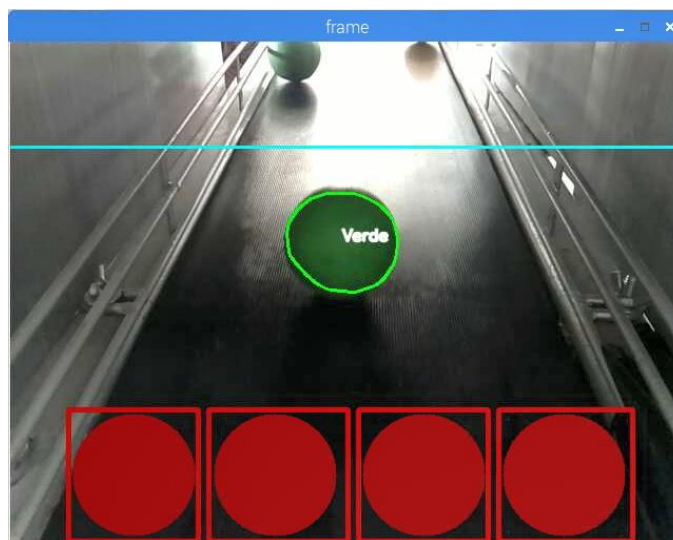


Figura 94. Detección del color verde de mangos.

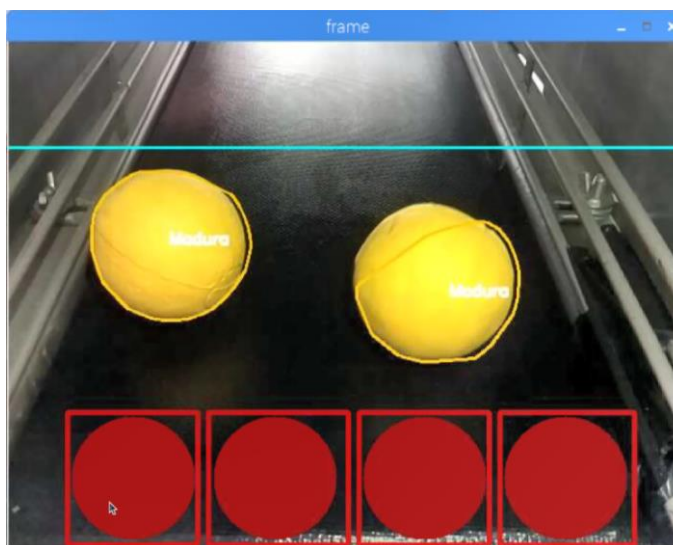


Figura 95. Detección del color amarillo (maduros) de mangos.

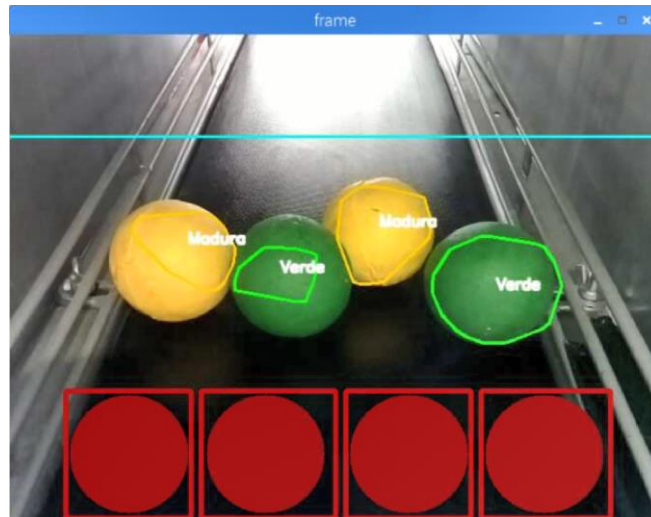


Figura 96. Detección de colores; verde y amarillo de mangos.

Se realizan pruebas para la detección de colores; verde y amarillo, mediante la cámara Raspberry Pi a 0, 15, 30, 45, 60 y 85 grados (ver Figura 97), obteniendo los resultados de la tabla 14.

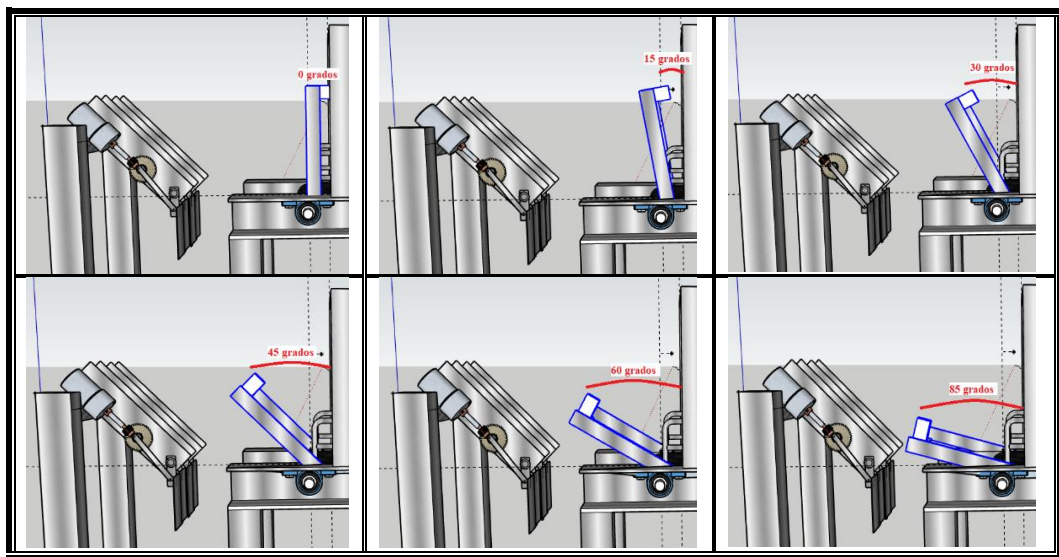


Figura 97. Diseño en SketchUp del sistema, con cámara a 0°, 15°, 30°, 45°, 60°, 85° para la detección.

DETECCIÓN DE MANGOS				
GRADOS DE CAPTURA DE FRUTA	N° DE MANGOS VERDES		N° DE MANGOS MADUROS	
	CON ÉXITO	SIN ÉXITO	CON ÉXITO	SIN ÉXITO
0, 15, 30, 45, 60 y 85 grados	20	0	20	0

Tabla 14. Datos de la detección de mangos

Pruebas y resultados del sistema de clasificación de mangos en función de su color.

Se realiza la programación en lenguaje de escalera en el software Tia Porta, cada vez que se detecta el color verde o amarillo, se recibe la señal de la Raspberry Pi 3 y este activa la determinada salida del sistema que corresponden al proceso de clasificación de mangos según su color (ver Figura 98 a Figura 101).

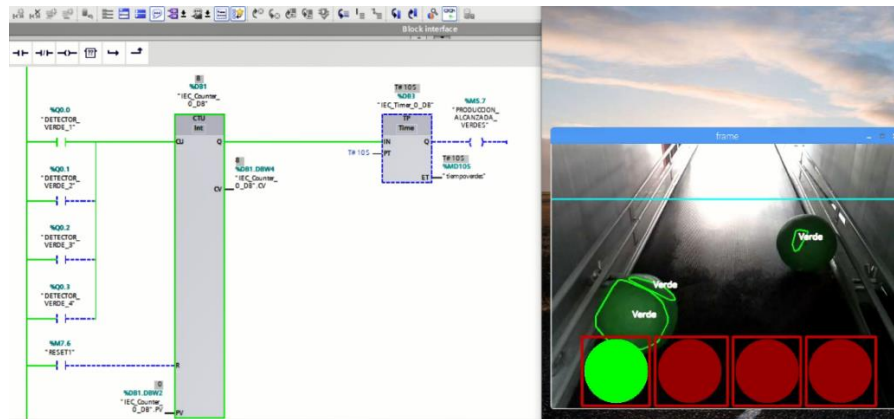


Figura 98. Detección de color verde y activación de salida Q0.0

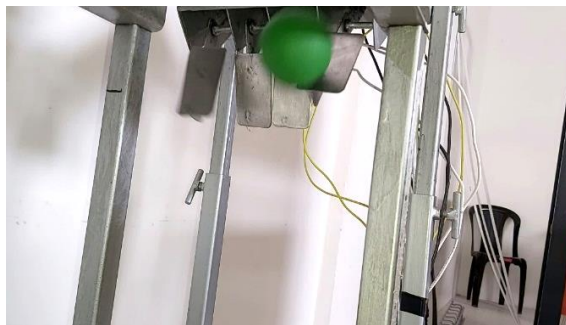


Figura 99. Activación de salida Q0.0 correspondiente al primer actuador clasificador.

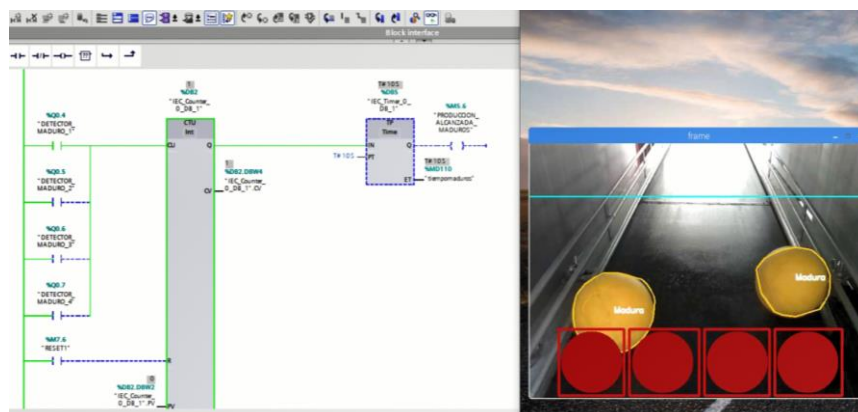


Figura 100. Detección de color amarillo y activación de salida Q0.4 para su conteo.

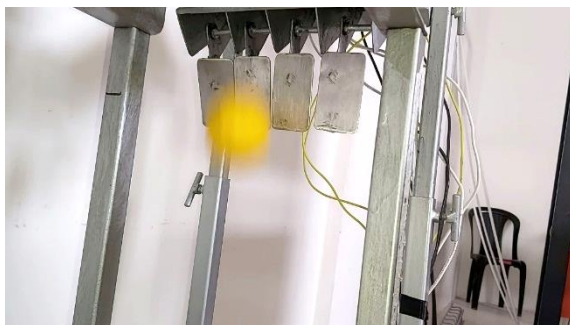


Figura 101. Paso de mango maduro o amarillo sin activar ningún actuador.

Se realizan pruebas para la detección de colores; verde y amarillo a 0, 15, 30, 45, 60 y 85 grados para su correcta clasificación (ver Figura 102 a Figura 107), obteniendo los siguientes resultados (ver Tabla 15 a Tabla 20).

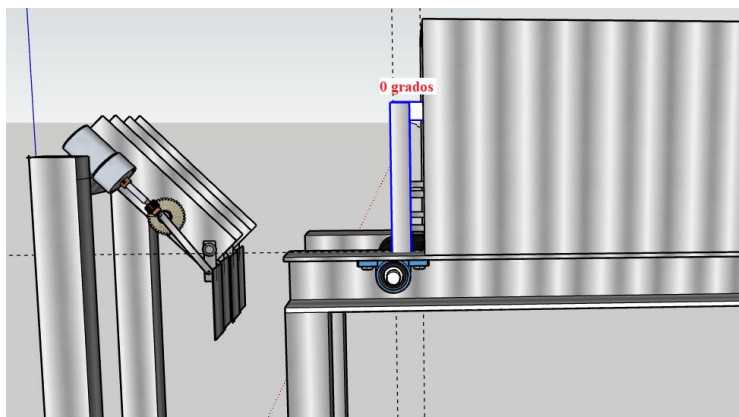


Figura 102. Diseño en SketchUp del sistema, con cámara a 0° para la detección.

CLASIFICACIÓN DE MANGOS				
GRADOS DE CAPTURA DE FRUTA	N° DE MANGOS VERDES		N° DE MANGOS MADUROS	
	20		20	
0 grados	CON ÉXITO	SIN ÉXITO	CON ÉXITO	SIN ÉXITO
		5	15	20

Tabla 15. Datos de la clasificación de mangos.

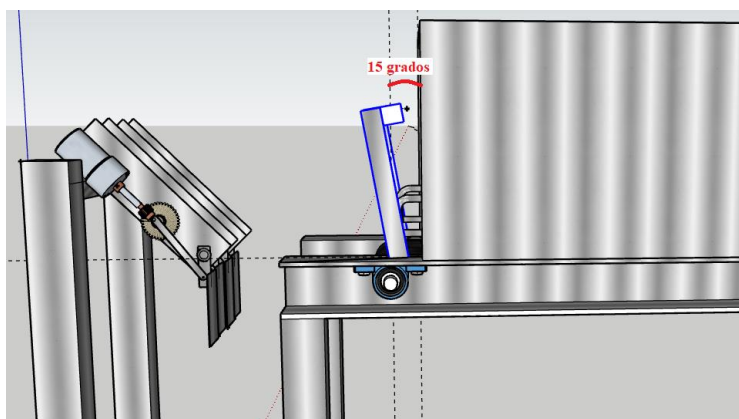


Figura 103. Diseño en SketchUp del sistema, con cámara a 15° para la detección.

CLASIFICACIÓN DE MANGOS				
GRADOS DE CAPTURA DE FRUTA	N° DE MANGOS VERDES		N° DE MANGOS MADUROS	
	20		20	
15 grados	CON ÉXITO	SIN ÉXITO	CON ÉXITO	SIN ÉXITO
	8	12	20	0

Tabla 16. Datos de la clasificación de mangos.

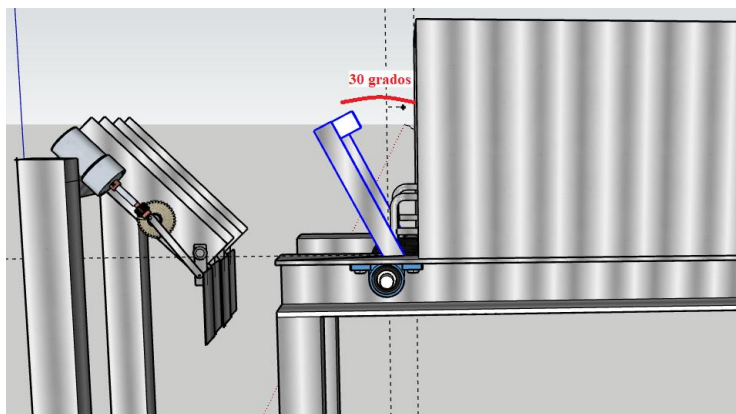


Figura 104. Diseño en SketchUp del sistema, con cámara a 30° para la detección.

CLASIFICACIÓN DE MANGOS				
GRADOS DE CAPTURA DE FRUTA	N° DE MANGOS VERDES		N° DE MANGOS MADUROS	
	20		20	
30 grados	CON ÉXITO	SIN ÉXITO	CON ÉXITO	SIN ÉXITO
	13	7	20	0

Tabla 17. Datos de la clasificación de mangos.

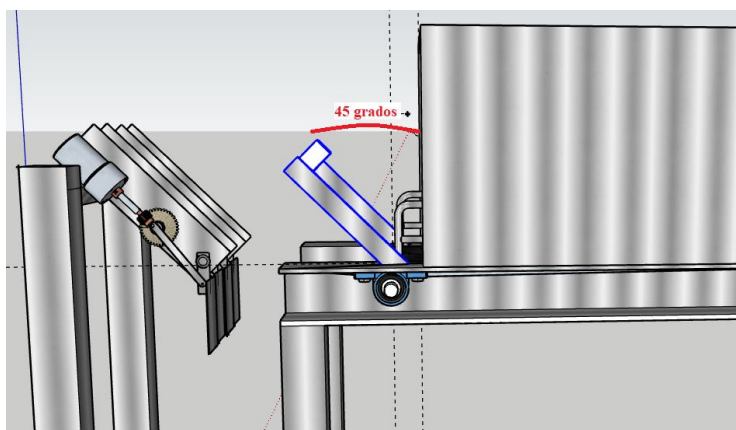


Figura 105. Diseño en SketchUp del sistema, con cámara a 45° para la detección.

CLASIFICACIÓN DE MANGOS				
GRADOS DE CAPTURA DE FRUTA	N° DE MANGOS VERDES		N° DE MANGOS MADUROS	
	20		20	
45 grados	CON ÉXITO	SIN ÉXITO	CON ÉXITO	SIN ÉXITO
	18	2	20	0

Tabla 18. Datos de la clasificación de mangos.

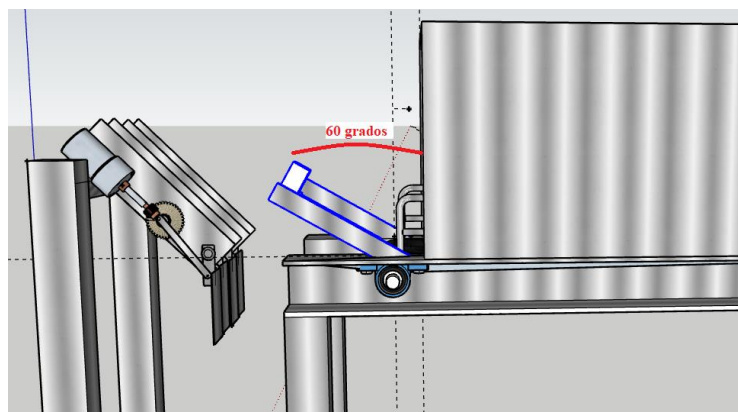


Figura 106. Diseño en SketchUp del sistema, con cámara a 60° para la detección.

CLASIFICACIÓN DE MANGOS				
GRADOS DE CAPTURA DE FRUTA	N° DE MANGOS VERDES		N° DE MANGOS MADUROS	
	CON ÉXITO	SIN ÉXITO	CON ÉXITO	SIN ÉXITO
60 grados	15	5	20	0

Tabla 19. Datos de la clasificación de mangos.

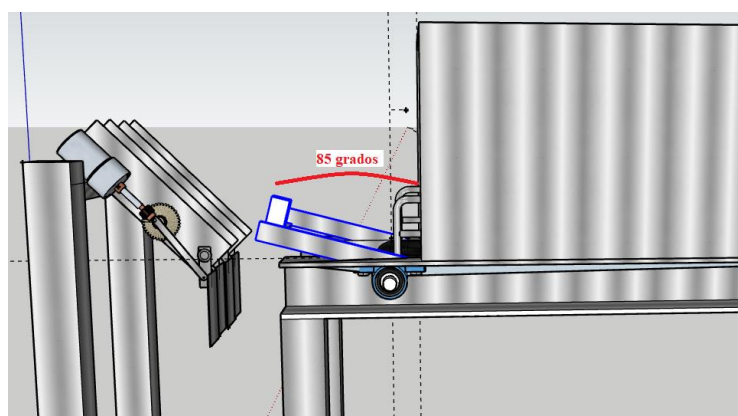


Figura 107. Diseño en SketchUp del sistema, con cámara a 85° para la detección.

CLASIFICACIÓN DE MANGOS				
GRADOS DE CAPTURA DE FRUTA	N° DE MANGOS VERDES		N° DE MANGOS MADUROS	
	CON ÉXITO	SIN ÉXITO	CON ÉXITO	SIN ÉXITO
85 grados	5	15	20	0

Tabla 20. Datos de la clasificación de mangos.

De acuerdo a los grados de detección para la clasificación de mangos en función de su color se obtuvo el siguiente resultado (ver Tabla 21) (ver Figura 108 y Figura 109):

DETECCIÓN Y CLASIFICACIÓN DE MANGOS				
Grados de detección	MANGOS VERDES		MANGOS MADUROS	
	Con éxito	Sin éxito	Con éxito	Sin éxito
0°	5	2	20	0
15°	8	12	20	0
30°	13	7	20	0
45°	18	2	20	0
60°	15	5	20	0
85°	5	15	20	0

Tabla 21. Datos generales de detección y clasificación de mangos.

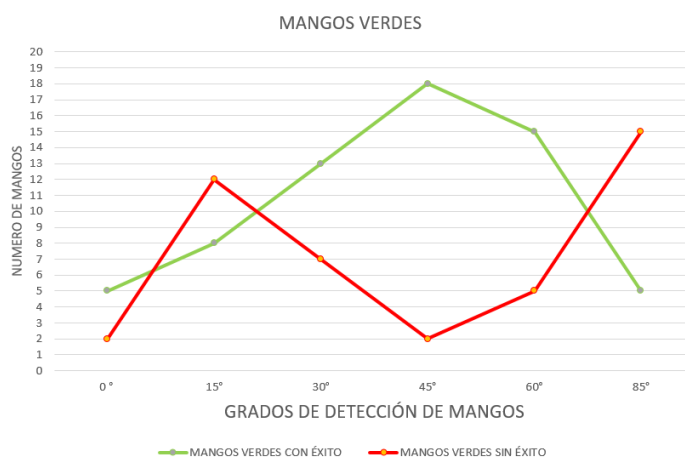


Figura 108. Número vs grados de detección de mangos verdes.

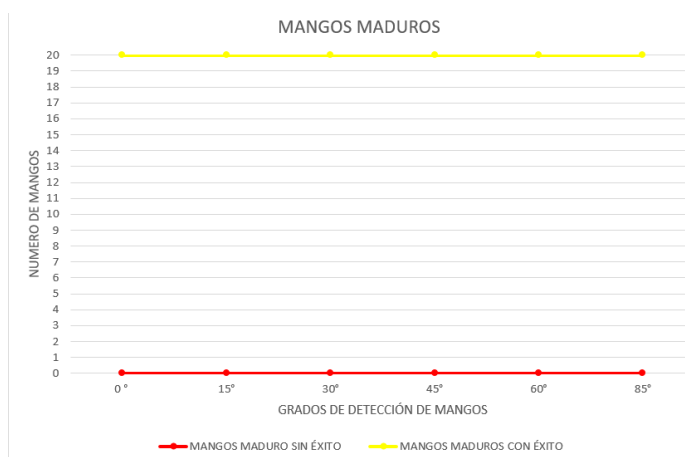


Figura 109. Número vs grados de detección de mangos maduros.

Pruebas de las interfaces gráficas del sistema SCADA Y HMI

El PLC s7-1200 interactúa con el sistema SCADA a través del Opc KEPServer y la interfaz gráfica de la pantalla HMI Delta. En la Figura 110, se muestra el sistema SCADA y en la Figura 111 se muestran las variables del sistema totalmente apagado.



Figura 110. Sistema SCADA totalmente apagado.

Item ID	Data Type	Value	Timestamp	Quality
● CANAL DEL SISTEMA SISTEMA RESET1	Boolean	0	12:52:10.306	Good
● CANAL DEL SISTEMA SISTEMA PRODUCCION_ESTIMADA_VERDES	Word	0	12:53:08.839	Good
● CANAL DEL SISTEMA SISTEMA PRODUCCION_ESTIMADA_MADUROS	Word	0	12:53:11.844	Good
● CANAL DEL SISTEMA SISTEMA PRODUCCION_ALCANZADA_VERDES	Boolean	0	12:53:18.970	Good
● CANAL DEL SISTEMA SISTEMA PRODUCCION_ALCANZADA_MADUROS	Boolean	0	12:53:21.465	Good
● CANAL DEL SISTEMA SISTEMA INGRESO_VELOCIDAD_RPM	Word	0	12:53:16.450	Good
● CANAL DEL SISTEMA SISTEMA FRECUENCIA	Word	0	12:53:16.450	Good
● CANAL DEL SISTEMA SISTEMA ENCENDER_APAGAR_MOTOR_BANDA	Boolean	0	12:53:20.955	Good
● CANAL DEL SISTEMA SISTEMA DETECTOR_VERDE_4	Boolean	0	12:52:10.306	Good
● CANAL DEL SISTEMA SISTEMA DETECTOR_VERDE_3	Boolean	0	12:52:10.306	Good
● CANAL DEL SISTEMA SISTEMA DETECTOR_VERDE_2	Boolean	0	12:52:10.306	Good
● CANAL DEL SISTEMA SISTEMA DETECTOR_VERDE_1	Boolean	0	12:52:10.306	Good
● CANAL DEL SISTEMA SISTEMA DETECTOR_MADURO_4	Boolean	0	12:52:10.306	Good
● CANAL DEL SISTEMA SISTEMA DETECTOR_MADURO_3	Boolean	0	12:52:10.306	Good
● CANAL DEL SISTEMA SISTEMA DETECTOR_MADURO_2	Boolean	0	12:52:10.275	Good
● CANAL DEL SISTEMA SISTEMA DETECTOR_MADURO_1	Boolean	0	12:52:10.275	Good
● CANAL DEL SISTEMA SISTEMA CONTADOR_VERDES	Word	0	12:52:48.793	Good
● CANAL DEL SISTEMA SISTEMA CONTADOR_MADUROS	Word	0	12:52:48.793	Good
● CANAL DEL SISTEMA SISTEMA_Slot	Byte	1	12:52:10.275	Good
● CANAL DEL SISTEMA SISTEMA_Rack	Byte	0	12:52:10.275	Good

Figura 111. Variables en KEPServer del sistema totalmente apagado.

Virtualmente mediante la interfaz de Intouch se realiza el ingreso de la velocidad en RPM, en este caso 900 RPM y automáticamente se muestra la velocidad también en Hz (ver Figura 112). Inmediatamente se ve modificada la variable “INGRESO_VELOCIDAD_RPM” en KEPServer (ver Figura 113). De igual manera, se puede verificar en la pantalla HMI (ver Figura 114).

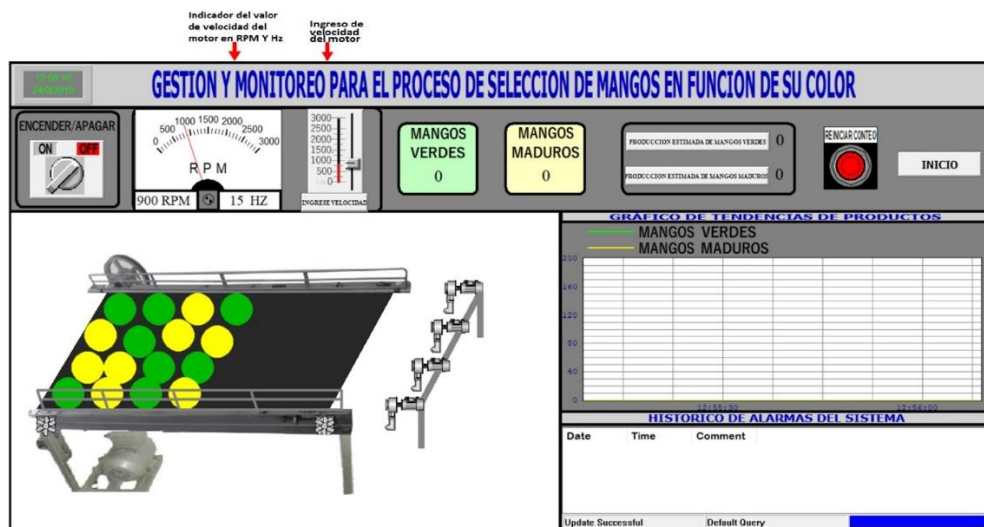


Figura 112. Prueba de ingreso de velocidad de 900 RPM mediante SCADA.

Item ID	Data Type	Value	Timestamp	Quality
CANAL DEL SISTEMA.SISTEMA.RESET1	Boolean	0	12:52:10.306	Good
CANAL DEL SISTEMA.SISTEMA.PRODUCCION_ESTIMADA_VERDES	Word	0	12:53:08.839	Good
CANAL DEL SISTEMA.SISTEMA.PRODUCCION_ESTIMADA_MADUROS	Word	0	12:53:11.844	Good
CANAL DEL SISTEMA.SISTEMA.PRODUCCION_ALCANZADA_VERDES	Boolean	0	12:53:18.970	Good
CANAL DEL SISTEMA.SISTEMA.PRODUCCION_ALCANZADA_MADUROS	Boolean	0	12:53:21.465	Good
CANAL DEL SISTEMA.SISTEMA.INGRESO_VELOCIDAD_RPM	Word	900	12:56:03.603	Good
CANAL DEL SISTEMA.SISTEMA.FRECUENCIA	Word	15	12:56:03.603	Good
CANAL DEL SISTEMA.SISTEMA.ENCENDER_APAGAR_MOTOR_BANDA	Boolean	0	12:53:20.955	Good

Figura 113. Prueba de la variable “INGRESO_VELOCIDAD_RPM” y “FRECUENCIA” en KEPServer.



Figura 114. Prueba de ingreso de velocidad de 900 RPM mediante interfaz HMI.

En la interfaz de Intouch se realiza el encendido del motor para la banda transportadora del sistema (ver Figura 115). Inmediatamente se ve modificada la variable “ENCENDER_APAGAR_MOTOR_BANDA” en KEPServer (ver Figura 116). De igual manera, se puede verificar en la pantalla HMI (ver Figura 117).

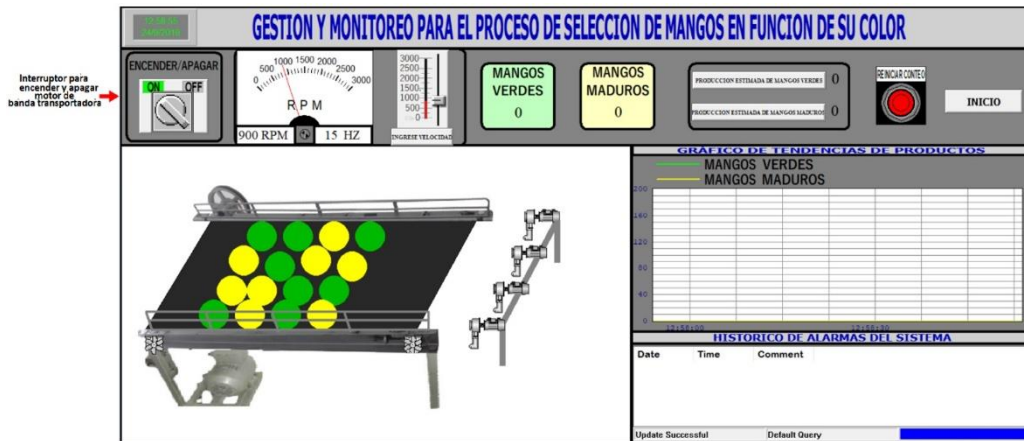


Figura 115. Prueba de encendido del motor para la banda transportadora del sistema mediante SCADA.

Item ID	Data Type	Value	Timestamp	Quality
CANAL DEL SISTEMA.SISTEMA.RESET1	Boolean	0	12:52:10.306	Good
CANAL DEL SISTEMA.SISTEMA.PRODUCCION_ESTIMADA_VERDES	Word	0	12:53:08.839	Good
CANAL DEL SISTEMA.SISTEMA.PRODUCCION_ESTIMADA_MADUROS	Word	0	12:53:11.844	Good
CANAL DEL SISTEMA.SISTEMA.PRODUCCION_ALCANZADA_VERDES	Boolean	0	12:53:18.970	Good
CANAL DEL SISTEMA.SISTEMA.PRODUCCION_ALCANZADA_MADUROS	Boolean	0	12:53:21.465	Good
CANAL DEL SISTEMA.SISTEMA.INGRESO_VELOCIDAD_RPM	Word	900	12:56:03.603	Good
CANAL DEL SISTEMA.SISTEMA.FRECUENCIA	Word	15	12:56:03.603	Good
CANAL DEL SISTEMA.SISTEMA.ENCENDER_APAGAR_MOTOR_BANDA	Boolean	1	12:58:53.392	Good
CANAL DEL SISTEMA.SISTEMA.DETECTOR_VERDE_4	Boolean	0	12:52:10.306	Good

Figura 116. Prueba de la variable “ENCENDER_APAGAR_MOTOR_BANDA” en KEPServer.



Figura 117. Prueba de encendido del motor para la banda transportadora del sistema mediante interfaz HMI.

Prueba de la identificación en SCADA del accionamiento de los actuadores, que clasifican los mangos verdes y maduros (ver Figura 118 y Figura 119). De igual manera, se puede verificar en la pantalla HMI (ver Figura 120 y ver Figura 121).

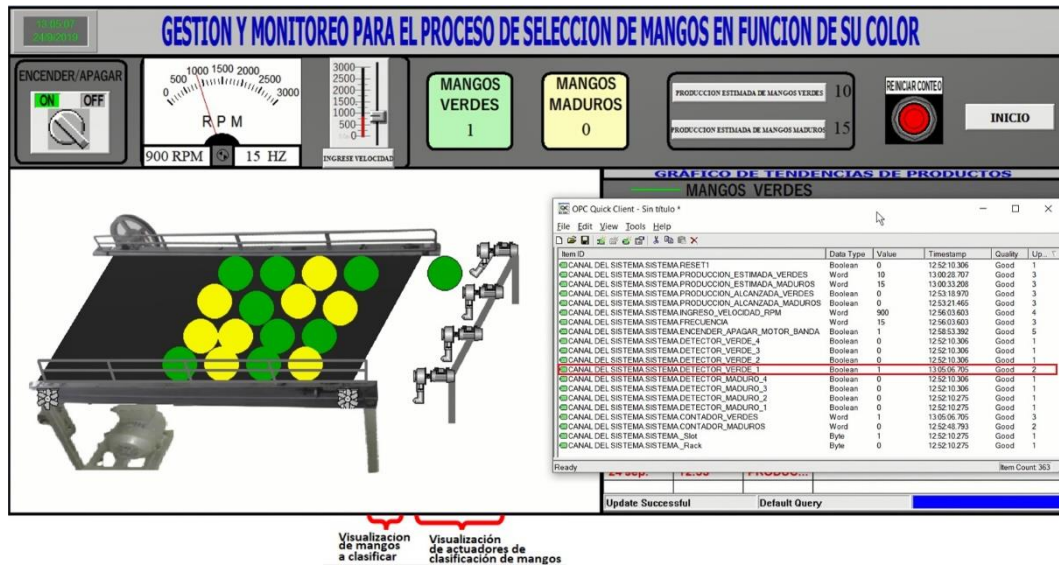


Figura 118. Prueba del primer detector de mangos verdes, simulación del mango mediante SCADA y activación de su variable en KEPServer.



Figura 119. Prueba del primer detector de mangos verdes, simulación del mango mediante interfaz HMI.

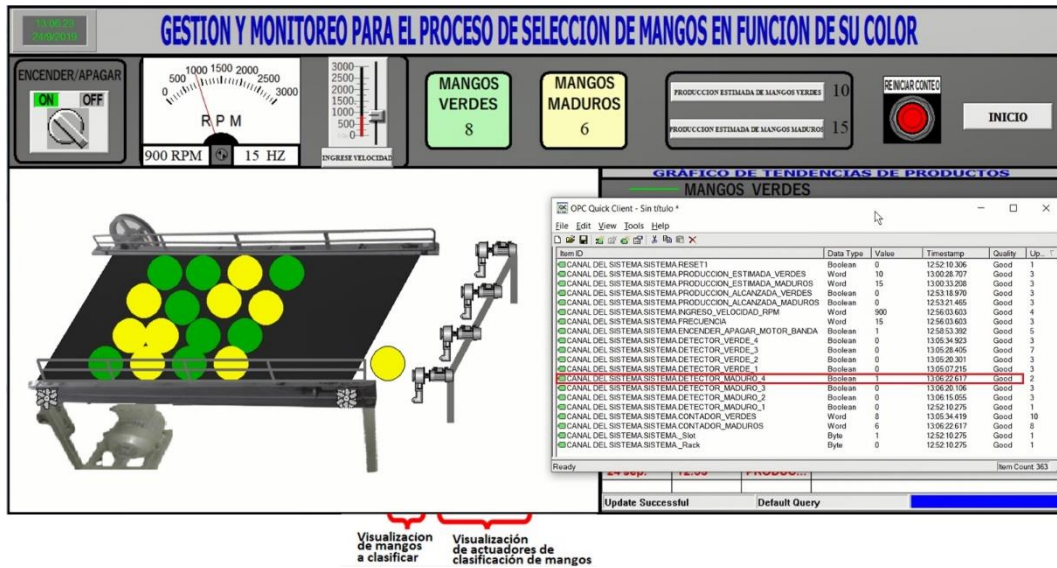


Figura 120. Prueba del cuarto detector de mangos maduros, simulación del mango mediante SCADA y activación de su variable en KEPServer.



Figura 121. Prueba del cuarto detector de mangos maduros, simulación del mango mediante interfaz HMI.

En la interfaz SCADA también se realiza el conteo de mangos verdes y maduros con su respectiva gráfica de tendencia de productos (ver Figura 122). También pruebas de las identificaciones respectivas a las variables del “CONTADOR_VERDES” y “CONTADOR_MADUROS” en KEPServer (ver Figura 123). De igual manera, se puede verificar en la pantalla HMI (ver Figura 124) y en sus respectivos históricos (ver Figura 125 y ver Figura 126).

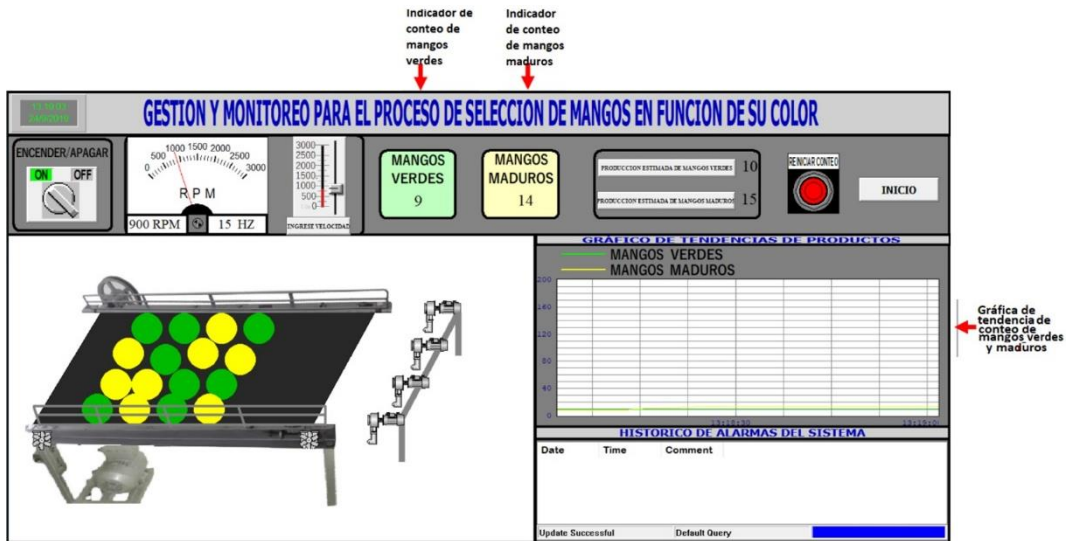


Figura 122. Prueba del conteo de mangos verdes y maduros con su respectiva grafica de tendencia mediante SCADA.

CANAL DEL SISTEMA SISTEMA DETECTOR_MADURO_2	boolean	0	13:16:24.621	Good
CANAL DEL SISTEMA SISTEMA DETECTOR_MADURO_1	boolean	0	12:52:10.275	Good
CANAL DEL SISTEMA SISTEMA CONTADOR_VERDES	Word	9	13:17:51.492	Good
CANAL DEL SISTEMA SISTEMA CONTADOR_MADUROS	Word	14	13:18:24.114	Good
CANAL DEL SISTEMA SISTEMA_Slot	Byte	1	12:52:10.275	Good
CANAL DEL SISTEMA SISTEMA_Rack	Byte	0	12:52:10.275	Good

Figura 123. Prueba de la variables “CONTADOR_VERDES” y “CONTADOR_MADUROS” en KEPServer.



Figura 124. Prueba del conteo de mangos verdes y maduros mediante interfaz HMI.

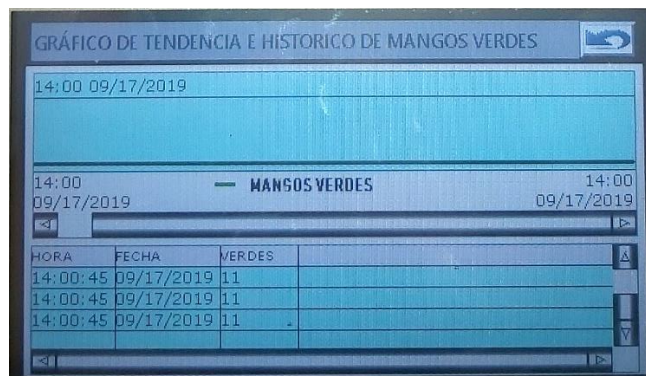


Figura 125. Prueba de la gráfica de tendencia del respectivo conteo de mangos verdes mediante interfaz HMI.

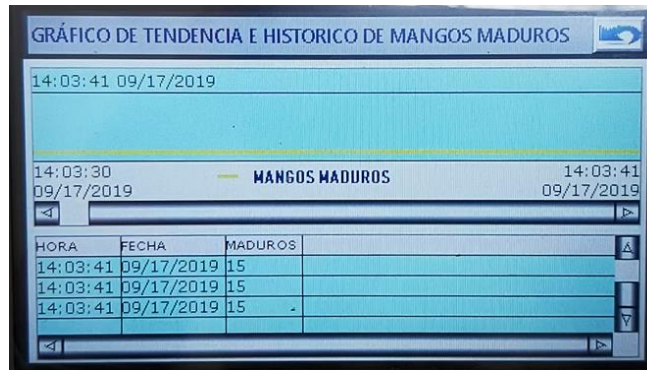


Figura 126. Prueba de la gráfica de tendencia del respectivo conteo de mangos maduros mediante interfaz HMI.

Prueba en la interfaz SCADA del ingreso de la producción estimada de mangos verdes y maduros, que sirven para obtener una alarma al alcanzar el valor asignado (ver Figura 127) y las variable “PRODUCCION_ESTIMADA_VERDES” y “PRODUCCION_ESTIMADA_MADUROS” predeterminadas en KEPServer (ver Figura 128). De igual manera, se puede verificar en la pantalla HMI (ver Figura 129).



Figura 127. Prueba del ingreso producción estimada de mangos verdes y maduros mediante SCADA.

Item ID	Data Type	Value	Timestamp	Quality
● CANAL DEL SISTEMA.SISTEMA RESET1	Boolean	0	12:52:10.306	Good
● CANAL DEL SISTEMA.SISTEMA PRODUCCION_ESTIMADA_VERDES	Word	10	13:00:28.707	Good
● CANAL DEL SISTEMA.SISTEMA PRODUCCION_ESTIMADA_MADUROS	Word	15	13:00:33.208	Good
● CANAL DEL SISTEMA.SISTEMA PRODUCCION_ALCANZADA_VERDES	Boolean	0	12:53:18.970	Good

Figura 128. Prueba de las variables “PRODUCCION_ESTIMADA_VERDES” y “PRODUCCION_ESTIMADA_MADUROS” en KEPServer.

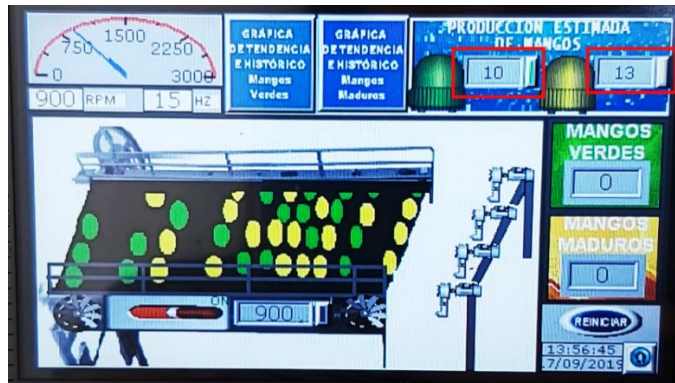


Figura 129. Prueba del ingreso producción estimada de mangos verdes y maduros mediante interfaz HMI.

Prueba de activación y desactivación de alarmas, estas son generadas al alcanzar la producción estimada de mangos verdes y maduros (ver Figura 130 y Figura 131). De igual manera, se puede verificar en la pantalla HMI (ver Figura 132 y 133).



Figura 130. Prueba de alarma activada al alcanzar la producción estimada de mangos maduros mediante SCADA.



Figura 131. Prueba de alarma activada al alcanzar la producción estimada de mangos maduros mediante interfaz HMI.



Figura 132. Prueba de alarma desactivada después de alcanzar la producción estimada de mangos maduros mediante SCADA.



Figura 133. Prueba de alarma desactivada después de alcanzar la producción estimada de mangos maduros mediante interfaz HMI.

Se realiza la prueba en la interfaz SCADA sobre el reinicio del conteo de mangos verdes y maduros mediante un botón (ver Figura 134 y Figura 135). De igual manera, se puede verificar en la pantalla HMI (ver Figura 136 y Figura 137).



Figura 134. Prueba del botón para reiniciar el conteo de mangos verdes y maduros mediante SCADA y activación de su variable en KEPServer.

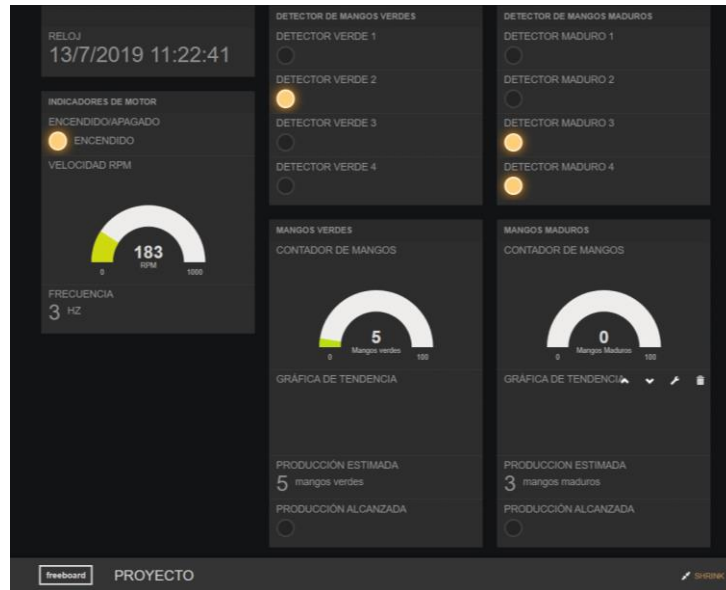


Figura 138. Plataforma web Freeboard del sistema.

Prueba de indicadores de: ingreso de velocidad (RPM - Hz) y encendido/apagado del motor de la banda transportadora del sistema (ver Figura 139 y Figura 140).

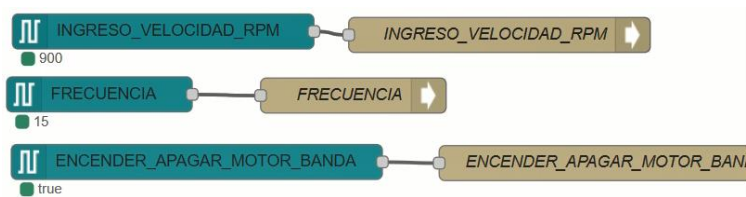


Figura 139. Programación de variables de velocidad y encendido/apagado de motor en NodeRed.



Figura 140. Indicadores de la interfaz de Freeboard perteneciente hora y fecha, velocidad (RPM-Hz) y encendido/apagado de motor.

Prueba de indicadores del accionamiento de los detectores de mangos verdes (ver Figura 141 y Figura 142) y maduros (ver Figura 143 y Figura 144).

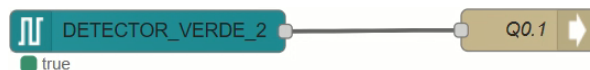


Figura 141. Programación de variable del segundo detector de mangos verdes en NodeRed.

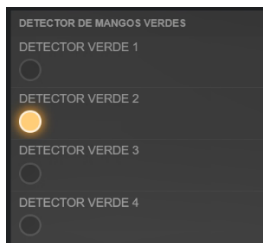


Figura 142. Indicadores de la interfaz de Freeboard perteneciente al segundo detector de mangos verdes.

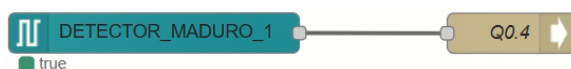


Figura 143. Programación de variable del primer detector de mangos maduros en NodeRed.



Figura 144. Indicadores de la interfaz de Freeboard perteneciente al primer detector de mangos maduros.

Prueba de indicadores del conteo de mangos verdes (ver Figura 145 y Figura 146) y maduros (ver Figura 147 y Figura 148) con su respectiva grafica de tendencia de productos.



Figura 145. Programación de variable del conteo de mangos verdes en NodeRed.



Figura 146. Indicadores de la interfaz de Freeboard perteneciente al conteo de mangos verdes y grafica de tendencias.



Figura 147. Programación de variable del conteo de mangos maduros en NodeRed.



Figura 148. Indicadores de la interfaz de Freeboard perteneciente al conteo de mangos maduros y grafica de tendencias.

Prueba de indicadores de la producción estimada de mangos verdes y maduros (ver Figura 149 y Figura 150), con su respectiva alarma (ver Figura 151 y Figura 152).

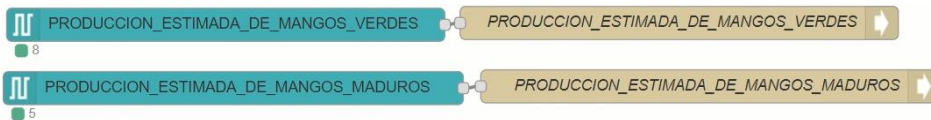


Figura 149. Programación de variable de la producción estimada de mangos verdes y maduros.

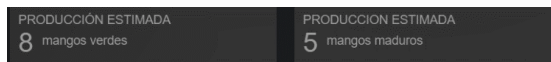


Figura 150. Indicadores de la interfaz de Freeboard perteneciente la producción estimada de mangos verdes y maduros.

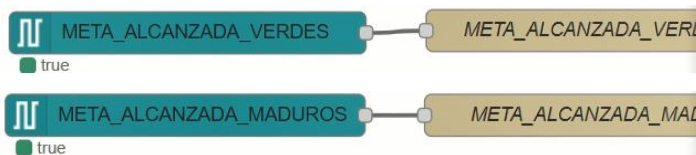


Figura 151. Programación de variable de meta alcanzada de la producción estimada de mangos verdes y maduros.

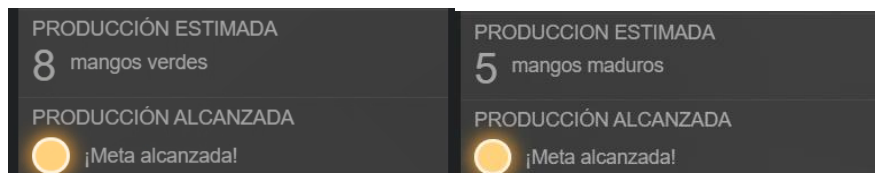


Figura 152. Indicadores de la interfaz de Freeboard perteneciente alarma activada al alcanzar la producción estimada de mangos verdes y maduros.

CAPÍTULO IV

4.1 CONCLUSIONES

Se realizó con éxito la transmisión y recepción de información entre los dispositivos del sistema SCADA, HMI, variador de frecuencia, PLC s7-1200, Raspberry Pi 3 y Simatic IoT2040 con un tiempo de respuesta mínimo de 0,8ms y máximo de 5,6ms con cero paquetes perdidos, incorporando la tecnología Profinet.

La implementación del sistema de visión artificial utilizando Python y OpenCV logró detectar los colores verde y amarillo con una eficiencia de 100% a distintos ángulos de incidencia.

El sistema electromecánico lineal permitió la clasificación y actuación del sistema para la clasificación de los mangos verdes.

La implementación de las interfaces HMI y SCADA permitieron visualizar en tiempo real el comportamiento del proceso de clasificación de mangos en función de su color.

La configuración del dispositivo Siemens Simatic IoT 2040 admitió la comunicación PLC s7-1200 con la plataforma NodeRed, logrando la visualización de los datos del proceso en la plataforma Freeboard.

4.2 RECOMENDACIONES

En cuanto al sistema de clasificación, se recomienda utilizar motores paso a paso y aplicar otros mecanismos que permitan movimientos de manera rápida y eficiente.

Se recomienda obtener las licencias de los softwares utilizados; Intouch y KEPServer, con la finalidad de no presentar interrupciones o pérdidas de comunicación, puesto que el desarrollo del sistema SCADA se realizó en versiones DEMO que presentan interrupciones cada 2 horas.

Para futuros trabajos, el sistema de detección puede ser modificado en la programación de Python y OpenCV permitiendo la detección de otros tipos de frutas; en cuanto a tamaño, se debe modificar la variable que define el espacio de detección, a colores, se modifica el rango de colores a detectar y a la forma, se realiza la detección capturando el contorno de la forma la fruta.

Al trabajar con el dispositivo Simatic y Node Red es necesario mantener activa su sesión en el software Putty ya que este es su servidor y permite el envío y recepción desde el PLC s7-1200 a la plataforma web utilizada.

Se desarrolló el monitoreo del proceso en la plataforma web Freeboard que permite un número limitado de variables y widgets solo para monitoreo, si se requiere trabajar en la misma plataforma web y se necesita control o más opciones sobre el proceso, se recomienda obtener una cuenta de pago que permita una cantidad ilimitada de variables, widgets y dashboards.

4.3 BIBLIOGRAFÍAS

- [1] L. Segundo Santiago, «ANÁLISIS DE LA PRODUCCIÓN Y COMERCIALIZACIÓN DEL MANGO,» BUENAVISTA, SALTILLO, COAHUILA, MÉXICO, 2005.
- [2] «Fundación Mango Ecuador,» Fundación Mango Ecuador, [En línea]. Available: <https://www.mangoecuador.org/index.php>.
- [3] The Siemon Company, Siemon Business Park y Siemon Company Drive, «SIEMON,» The Siemon Company, 2019. [En línea]. Available: https://www.siemon.com/la/white_papers/03-10-13-ethernet-ip.asp.
- [4] «El papel de la Ventaja competitiva en el desarrollo economico de los paises,» p. 73.
- [5] «Secretaría Nacional de Planificación y Desarrollo - Agenda-zona-5,» 2015. [En línea]. Available: <http://www.planificacion.gob.ec/wp-content/uploads/downloads/2015/11/Agenda-zona-5.pdf>.
- [6] A. R. G. VILLACÍS, «MODELO DE NEGOCIO PARA LA CREACIÓN DE SMART,» Quito, 2017.
- [7] G. Halfacree y E. Upton, Meet the Raspberry Pi, John Wiley & Sons, 2012.
- [8] G. Van Rossum y D. Fred L., An Introduction to Python, 2003.
- [9] A. Pajankar y A. Kakkar, Raspberry Pi By Exaple, 2016.
- [10] F. Torres y C. Jara, Automatización - Autómatas programables I., España, 2011.
- [11] J. C. Martín Castillo y M. P. García García, Automatismos industriales, Madrid: Editex S.A..
- [12] J. C. Martín Castillo y M. P. García García, Automatismos Industriales, Editex, 2016.
- [13] J.-Y. Fiset, Human-Machine Interface Design for process control applications, 2009.
- [14] A. Daneels y W. Salter, «What is SCADA,» 2005.
- [15] V. Guerrero, L. Martínez y R. Yuste, Comunicaciones Industriales, Marcombo, 2009.
- [16] L. Joyanes Aguilar, Ciberseguridad: la colaboración público-privada en la era de la cuarta revolución industrial., 2017.
- [17] H. Breivold y K. Sandstrom, Internet of Things for Industrial Automation - Challenges and Technical Solutions., Sydney, 2015.
- [18] IES-SEP La Garrotxa, Centre integral de formació professional de La Gar, Miguel Altuna Institutua, Interplast, Creacions Tecnologiques, Omron, Lazpiur y Ikuspe, Vision artificial - Aplicación práctica de la visión artificial, España, 2012.
- [19] R. Báez Sañudo, Manejo Postcosecha del Mango, 1997.

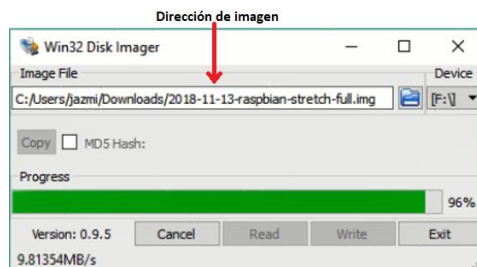
- [20] A. Romero Acero, A. Marín Cano y J. Jiménez Builes, «Sistema de clasificación por visión artificial de mangos tipo Tommy,» REVISTE UIS INGENIERÍAS, Universidad Industrial de Santander, 2015.
- [21] M. Carrillo López y F. Morales Mayorga, Implementación de un modulo de clasificacion por materiales y tamaños con Plc y pantalla tactil para el laboratorio de control y automatización de procesos industriales de la escuela Ingenieria Mecánica de la ESPOCH, Riobamba, 2015.
- [22] A. Torre Escudero, Puesta en marcha de un sistema de automatización Siemens Simatic IOT2040, Bilbao, 2018.
- [23] SIEMENS, «SIEMENS,» [En línea]. Available: <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/s7-1200.html>.
- [24] S. AG, «SINAMICS G: Speed Control of a G110M/G120,» Julio 2018. [En línea]. Available: [https://support.industry.siemens.com/cs/document/70155469/sinamics-g%3A-speed-control-of-a-g120-\(startdrive\)-with-s7-1200-via-profinet-profibus-dp-with-safety-integrated-\(via-terminal\)-and-hmi?dti=0&lc=en-WW](https://support.industry.siemens.com/cs/document/70155469/sinamics-g%3A-speed-control-of-a-g120-(startdrive)-with-s7-1200-via-profinet-profibus-dp-with-safety-integrated-(via-terminal)-and-hmi?dti=0&lc=en-WW). [Último acceso: 21 Septiembre 2019].
- [25] SIEMENS, «SIMATIC,» Octubre 2016. [En línea]. Available: https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/pc_industriales/pages/simatic-iot2040.aspx.
- [26] E. Upton y G. Halfacree, Raspberry Pi user guide, Wiley, 2012.
- [27] G. Senthilkumar, K. Gopalakrishnan y V. Sathish Kumar, «EMBEDDED IMAGE CAPTURING SYSTEM,» International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol. 3, pp. 213-215, 2014.
- [28] S. AG, «Totally Integrated Automation Portal,» Siemens, 2018.
- [29] RASPBERRY PI FOUNDATION, «RASPBERRY PI,» 2019. [En línea]. Available: <https://www.raspberrypi.org/downloads/raspbian/>.
- [30] Software Foundation Python, «Python,» [En línea]. Available: <https://docs.python.org/3/>.
- [31] G. Bradski y A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library, 2008.
- [32] M. Gijs y P. Stephan, «python snap7,» [En línea]. Available: <https://python-snap7.readthedocs.io/en/latest/introduction.html>.
- [33] JetBrains s.r.o., «JetBrains,» 2019. [En línea]. Available: <https://www.jetbrains.com/help/pycharm/installation-guide.html#toolbox>.
- [34] Winscp, «WinSCP,» 2019. [En línea]. Available: <https://winscp.net/eng/docs/lang:es>.

- [35] H. Rachmat y T. Mulyana, «Website design of EMS-SCADA for AC usage on a building,» International Conference on Information and Communication Technology, pp. 17-22, 2015.
- [36] D. J. Barrera, J. A. Mendoza, L. E. Neira y A. P. García, «Enlace entre PLC Siemens LOGO! 12/24 RCE y Wonderware InTouch para supervisión y control de nivel Link between PLC Siemens LOGO! 12/24 RCE and Wonderware InTouch for monitoring and level,» 2019.
- [37] «infoPLC,» [En línea]. Available: <http://www.infoplcn.net/actualidad-industrial/item/102902-isa101-hmi>.
- [38] «El productor,» [En línea]. Available: <http://elproductor.com/quienes-somos/>.
- [39] A. F. Montoro, Python 3 Al Descubierto, RC Libros, 2013.
- [40] C. F. Tedesco, Ascensores electrónicos y variadores de velocidad, Argentina: Librería y editoril Alsina, 2010.
- [41] I. G. Enríquez Harper, EL ABC DE LAS MAQUINAS ELÉCTRICAS, México: Limusa Noriega Editores, 2004.
- [42] R. Villena Albert, Proyecto de integración de autómatas programables Simatic S7-1200 en red IOT, Valencia, 2017.
- [43] A. Moreno Bermúdez, D. Ballesteros Padilla y G. Sánchez Torres, «UN PROTOTIPO MECÁNICO PARA LA AUTOMATIZACIÓN,» Ingenierías Universidad de Medellín, 2012.

4.4 ANEXOS

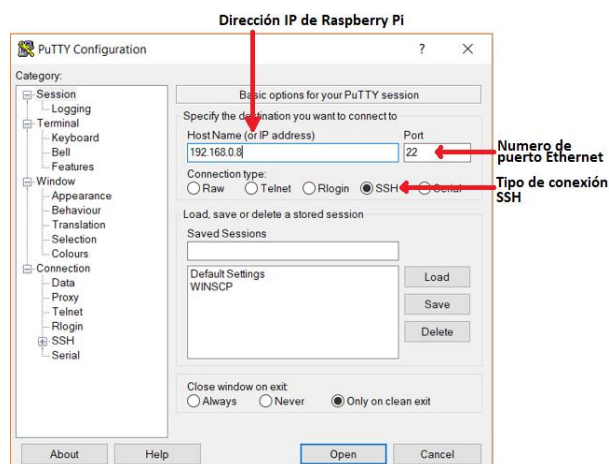
Anexo 1: Instalación de sistema operativo Raspbian Stretch.

Se realiza la descarga de la imagen del sistema operativo Raspbian Stretch con escritorio desde el sitio web oficial de Raspberry Pi, para un correcto funcionamiento se recomienda trabajar en una tarjeta de memoria mayor a 16Gbyte. Se crea la imagen del *Operating System* OS a través del software Win32 Disk Imager, terminada la creación de la imagen se coloca la tarjeta microSD en el dispositivo Raspberry Pi 3, posteriormente se enciende, lo cual se verifica en la interfaz gráfica donde se selecciona las configuraciones básicas del OS y finalmente está listo para usar.



Anexo 2: Instalación y funcionamiento de biblioteca Snap7

Se realiza la instalación de la biblioteca Snap7 que permite la comunicación Profinet entre Raspberry Pi 3 y PLC s7-1200. Para iniciar se accede al Raspberry Pi 3 desde Putty con conexión SSH e ingresando la dirección IP de la misma.



Ingresamos el usuario de la Raspberry Pi 3 junto a la contraseña e iniciamos sesión, ejecutamos el comando **wget** junto al link de Snap7 como se muestra en la figura

31 e iniciamos la descarga, se debe tomar en cuenta el nombre de la descarga. Luego se descomprime el archivo ejecutando el comando **tar -zxvf** junto al nombre de la descarga.

```

pi@raspberrypi:~$ wget https://sourceforge.net/projects/snap7/files/1.4.0/snap7-full-1.4.0.tar.gz/download
--2019-01-29 00:34:37-- https://sourceforge.net/projects/snap7/files/1.4.0/snap7-full-1.4.0.tar.gz/download
Resolviendo sourceforge.net (sourceforge.net)... 216.105.38.13
pi@raspberrypi:~$ tar -zxvf download
snap7-full-1.4.0/

```

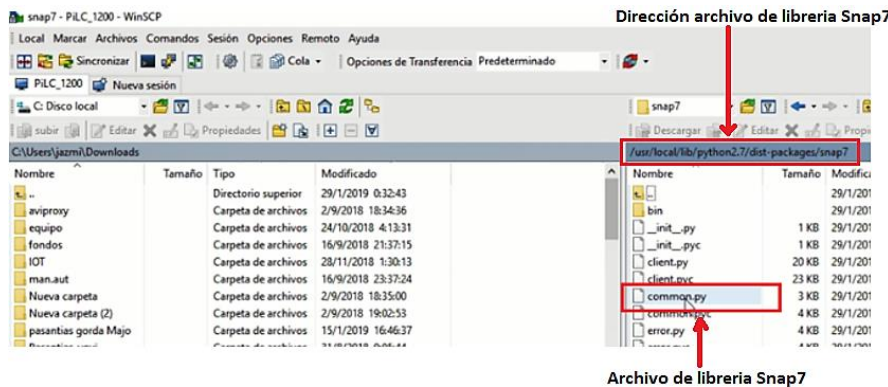
Lo siguiente es ejecutar los comandos para buscar en la carpeta de compilación **snap7-full-1.4.0** hasta llegar a Unix y se ejecuta **sudo make -f arm_v6_linux.mk** para la construcción de la biblioteca y haciendo referencia a la CPU ARMV6 que posee el Raspberry Pi 3.

```

pi@raspberrypi:~/snap7-full-1.4.0/build/bin$
snap7-full-1.4.0/win-clean.bat
pi@raspberrypi:~/snap7-full-1.4.0$ ls
Desktop Documents download Downloads MapPi Music Pictures Public snap7-full-1.4.0 Templates Videos
pi@raspberrypi:~/snap7-full-1.4.0$ cd snap7-full-1.4.0/
pi@raspberrypi:~/snap7-full-1.4.0$ ls
Makefile README.txt arm_v6_linux.mk common.mk examples gpi.txt i386_linux.mk i386_solaris_cc.mk i386_solaris_gcc.mk linux.mk
mips_linux.mk mips_solaris_cc.mk mips_solaris_gcc.mk README.txt release rich-demos src utility win-clean.bat
x86_64_linux.mk x86_64_solaris_cc.mk x86_64_solaris_gcc.mk
pi@raspberrypi:~/snap7-full-1.4.0/build$ cd build
pi@raspberrypi:~/snap7-full-1.4.0/build$ cd unix
pi@raspberrypi:~/snap7-full-1.4.0/build/unix$ ls
arm_v6_linux.mk common.mk i386_linux.mk i386_solaris_gcc.mk README.txt x86_64_linux.mk x86_64_solaris_gcc.mk
arm_v7_linux.mk i386_bad.mk i386_solaris_cc.mk mips_linux.mk mips_solaris_cc.mk x86_64_bad.mk x86_64_solaris_cc.mk
pi@raspberrypi:~/snap7-full-1.4.0/build/unix$ sudo make -f arm_v6_linux.mk
g++ -c ../../src/sys/snap_magsock.cpp -O3 -g -fPIC -mword-relocations -pedantic -o ../temp/arm_v6/sys_snap_magsock

```

Se necesita editar la ubicación de la librería en el archivo **common.py** que se encuentra en la dirección **/usr/local/lib/python2.7/dist-packages/snap7/**, se agrega la línea **lib_location = '/usr/local/lib/libsnap7.so'**, para crearlo en esta ubicación y se guarda los cambios.



```
class Snap7Library(object):
    """
    Snap7 loader and encapsulator. We make this a singleton to make
    sure the library is loaded only once.
    """
    _instance = None
    def __new__(cls, *args, **kwargs):
        if not cls._instance:
            cls._instance = object.__new__(cls)
            cls._instance.lib_location = None
            cls._instance.cdll = None
            return cls._instance

    def __init__(self, lib_location=None):
        lib_location = '/usr/local/lib/libsnap7.so'
        if self.cdll:
            return
        self.lib_location = lib_location or self.lib_location or find_library('snap7')
        if not self.lib_location:
            msg = "can't find snap7 library. If installed, try running ldconfig"
            raise Snap7Exception(msg)
        self.cdll = cdll.LoadLibrary(self.lib_location)
```

Linea agregada al archivo de libreria Snap7

Se ejecuta el comando `ls -la` para verificar si la librería Snap7 se ha ubicado correctamente en la dirección `usr/local/lib`.

```
pi@raspberrypi:~/usr/lib $ ls -la
-rw-r--r-- 1 root root 1116892 sep  5 2016 libqwt-qt5.so.6.1.2
drwxr-xr-x 5 root root 4096 nov 13 08:58 libreoffice
lrwxrwxrwx 1 root root 25 sep 19 2016 libresid-builder.so.0 -> libresid-b
-rw-r--r-- 1 root root 1045928 feb  6 2016 libamumps_seq-4.10.0.so
-rwxr-xr-x 1 root root 1084144 ene 29 11:03 libsnap7.so
drwxr-xr-x 3 root root 4096 nov 13 08:27 xorg
```

Libreria Snap7, ubicada en usr/local/lib.

Para poder instalar otros módulos es necesario instalar la herramienta pip de Python, para ello ejecutamos el comando `sudo apt-get install Python-pip`, el cual es un requerimiento para la instalación de Python-Snap7.

```
pi@raspberrypi: ~
pi@raspberrypi:~ $ sudo apt-get install python-pip
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
```

Luego, se instala Python-Snap7 a través del comando `sudo pip install python-snap7`, este es un contenedor y desarrollador para la biblioteca Snap7 en Python.

```
pi@raspberrypi: ~
pi@raspberrypi:~ $ sudo pip install python-snap7
Collecting python-snap7
Downloading https://files.pythonhosted.org/packages/
```

Para comprobar la correcta instalación se necesita instalar ipython, es un “Python interactivo”, ejecutamos el comando `sudo apt-get install ipython`.

```
pi@raspberrypi: ~
Proceed (y/n)? y
Successfully uninstalled ipython-2.3.0
pi@raspberrypi:~$ sudo apt-get install ipython
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
```

Una vez instalado ipython, se ejecutan los siguientes comandos:

- `Import snap7`
- `Plc = snap7.client.Client()`
- `Plc.connect('192.168.0.7',0,1)` ← Dirección IP del Plc s7-1200.

Luego de la ejecución de los comandos indicados, en la terminal no debe existir ningún mensaje de error, por lo cual se concluye que se ha realizado de manera completa la comunicación entre el Raspberry Pi 3 y PLC s7-1200. Sin embargo, si existe algún error indicado en la terminal se debe proceder a la reinstalación del sistema.

```
IPython: home/pi
login as: pi
pi@raspberrypi:~$ ipython
Python 2.7.13 (default, Sep 26 2018, 18:42:22)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
?quickref        -> Quick reference.
?help            -> Python's own help system.
?object?         -> Details about 'object', use 'object??' for extra details.

In [1]: import snap7 ← Importar biblioteca Snap7

In [2]: plc = snap7.client.Client() ← Asignación de Snap 7 como Cliente al Plc

In [3]: plc.connect('192.168.3.147',0,1) ← Ingreso de dirección IP del Plc para la
                                             conexión

In [4]:
```

Anexo 3: Instalación de librería OpenCV

Se instala la librería OpenCV en el dispositivo Raspberry Pi 3 con un OS Raspbian Stretch. Se ingresa mediante el software Putty, con la dirección IP de la Raspberry Pi 3 y con conexión SSH.

Primero se deben actualizar los paquetes existentes ejecutando el comando **sudo apt-get update && sudo apt-get upgrade**, luego se instalan las herramientas del desarrollador que sirven de apoyo en la construcción de OpenCV con el comando **sudo apt-get install build-essential cmake pkg-config**.

Para poder cargar archivos de imágenes del disco en formatos Jpeg, png o tiff se deben instalar bibliotecas de entrada/salida de imágenes, se realiza con el comando

sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev. Tambien, las bibliotecas de entrada/salida de video **sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev** y **sudo apt-get install libxvidcore-dev libx264-dev.**

Además, se instala la biblioteca GTK para ejecutar el modulo HighGUI que muestra las imágenes o videos en pantalla, luego se ejecuta el comando **sudo apt-get install libgtk2.0-dev libgtk-3-dev.**

La Raspberry Pi 3 es un dispositivo limitado en cuanto a recursos, para esto existen bibliotecas que permiten la optimización, con el comando **sudo apt-get install libatlas-base-dev gfortran.** Al instalar el sistema operativo Raspbian Stretch por defecto lleva instalado las versiones 2.7 y 3 de Python, sin embargo se instala nuevamente ejecutando el comando **sudo apt-get install python2.7-dev python3-dev.**

Para descargar el archivo de OpenCV desde el sitio web oficial de OpenCV, se utiliza el comando **Wget -O opencv.zip**, que se encuentra ubicado en la pagina web **<https://github.com/Itseez/opencv/archive/3.3.0.zip>**, luego se procede a descomprimir con el comando **unzip opencv.zp.**

Se descarga OpenCV Contrib, esta biblioteca da la ventaja de acceder a la mayoría de funciones y tener un OpenCV más completo, ingresando el comando **wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.3.0.zip** que incluye la direccion del sitio oficial y se descomprime con **unzip opencv_contrib.zip.** Se debe tomar en cuenta que OpenCV y opencv_contrib sean de versiones iguales.

Al trabajar con un procesamiento numerico, para ello se necesita de una herramienta que se encuentra en el paquete Numpy de Python, se procede a instalar con el comando **pip install numpy.**

Para compilar e instalar el OpenCV se utiliza el comando **CMake**, verificando que el interprete corresponda a Python 2.7, posterior se ejecutan los siguientes comandos:

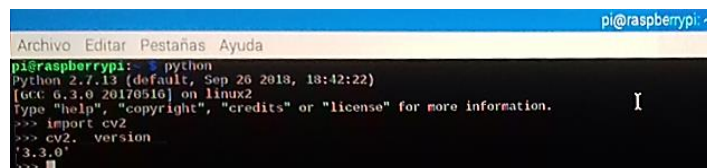
- `Cd ~/opencv-3.3.0/`

- Mkdir build
- Cd build
- Cmake -D CMAKE_BUILD_TYPE=RELEASE \
- -D CMAKE_INSTALL_PREFIX=/usr/local \
- -D INSTALL_PYTHON_EXAMPLES=ON \
- -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.3.0/modules \
- -D BUILD_EXAMPLES=ON ..

Para no tener problemas de espacio de almacenamiento en memoria, se procede a incrementar el espacio ayudando a que la compilación se realice en los cuatro núcleos de la Raspberry Pi 3. Ingresamos a `/etc/dphys-swapfile` seguido del comando `CONF_SWAPSIZE=1024`, este permite compilar en todos los núcleos de la Raspberry Pi 3. Para habilitar el espacio incrementado anteriormente con los comandos `sudo /etc/init.d/dphys-swapfile stop` y `sudo /etc/init.d/dphys-swapfile start`. Compilamos OpenCV ejecutando el comando `make -j4` y finalmente instalamos OpenCV ejecutando `sudo make install` y `sudo ldconfig`.

Para instalar OpenCV se debe considerar la siguiente dirección `/usr/local/lib/python2.7/dist-packages`, para ello ingresamos el comando `ls -l /usr/local/lib/python2.7/dist-packages/`, luego se introduce el comando `total 1852` y después el comando `-rw-r--r 1 root staff 1895772 Mar 20 20:00 cv.so`.

Para verificar la correcta instalación, ingresamos a Python e importamos OpenCV.



```

pi@raspberrypi:~$ python
Python 2.7.13 (default, Sep 26 2018, 18:42:22)
[GCC 6.3.0 20170516] on linux2
Type "help", "copyright", "credits" or "license()" for more information.
>>> import cv2
>>> cv2.__version__
'3.3.0'
>>>

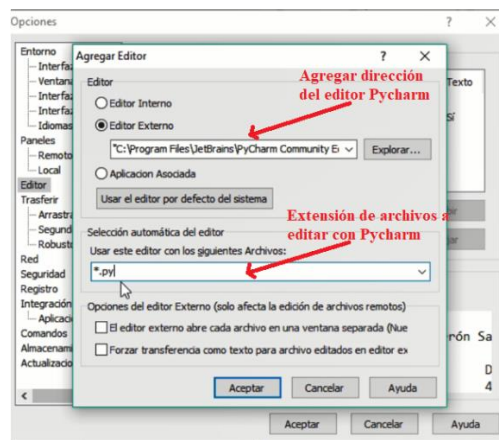
```

Anexo 4: Configuración PyCharm

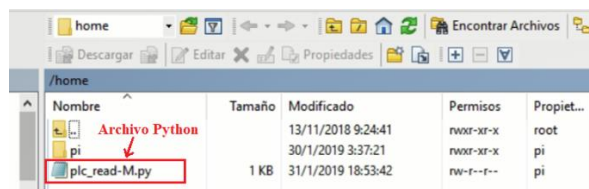
PyCharm es un software usado para la creación y edición del archivo programado con lenguaje Python, que permite la detección de colores del mango y la comunicación Raspberry Pi 3 – PLC s7-1200.

Se realiza una configuración para permitir crear archivos .py o Python, primero se debe agregar el editor, siendo este Pycharm, en **Opciones** → **Editor** → **Agregar**. Seguidamente, se selecciona **Editor Externo**, además, se busca el editor en el disco

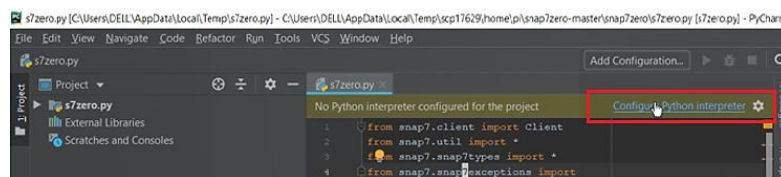
local C para agregarlo y se escribe la extensión de los archivos a editar, siendo estos .py.



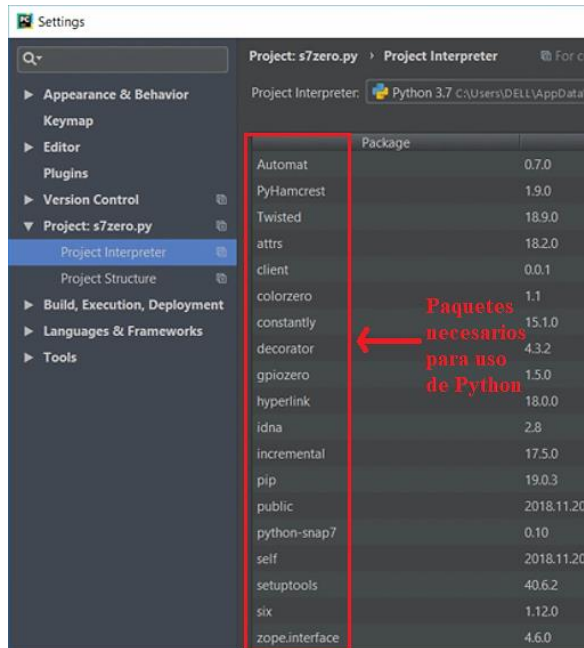
Se crea el archivo en Python y se le asigna un nombre con la extensión .py.



Al abrir este archivo se procede a instalar Pycharm y se procede a abrir el archivo que contiene el algoritmo para la detección con el editor Pycharm. El editor no trae configurado el interpretador que permite la edición, por lo cual se procede a configurarlo.

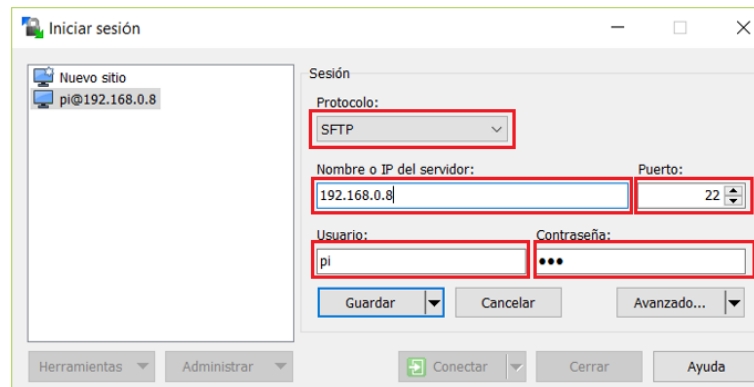


Para configurar el interpretador se accede a **Configure Python interpreter** → **Show all** → **Python 3.7**, logrando que se agreguen todos los paquetes necesarios. Y finalmente se puede editar de manera normal el archivo.



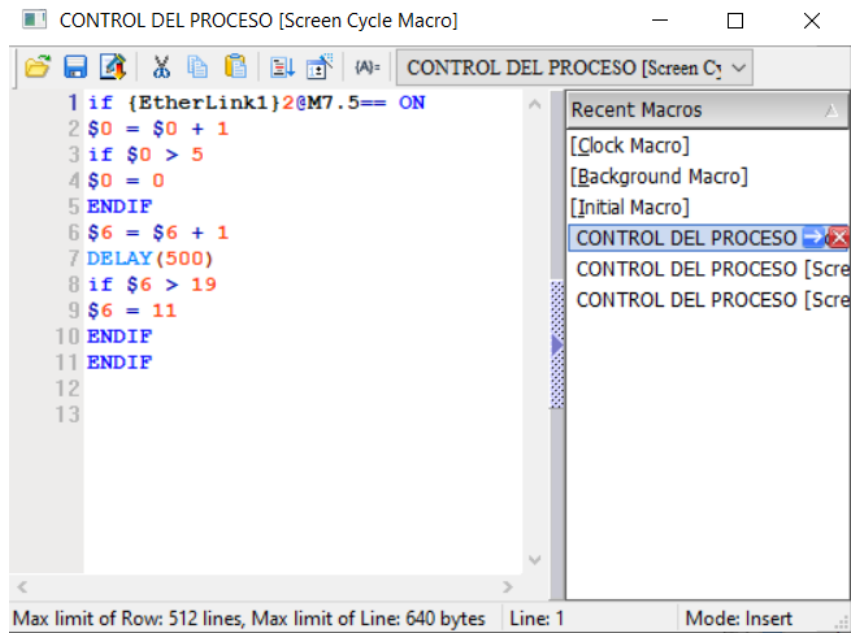
Anexo 5: Conexión WinSCP

La conexión de forma remota a la Raspberry Pi 3 desde una computadora se realiza a través de WinSCP, para ello, se selecciona el protocolo SFTP, la dirección IP del Raspberry Pi 3: 192.168.0.8, el número del puerto es 22, usuario y contraseña.

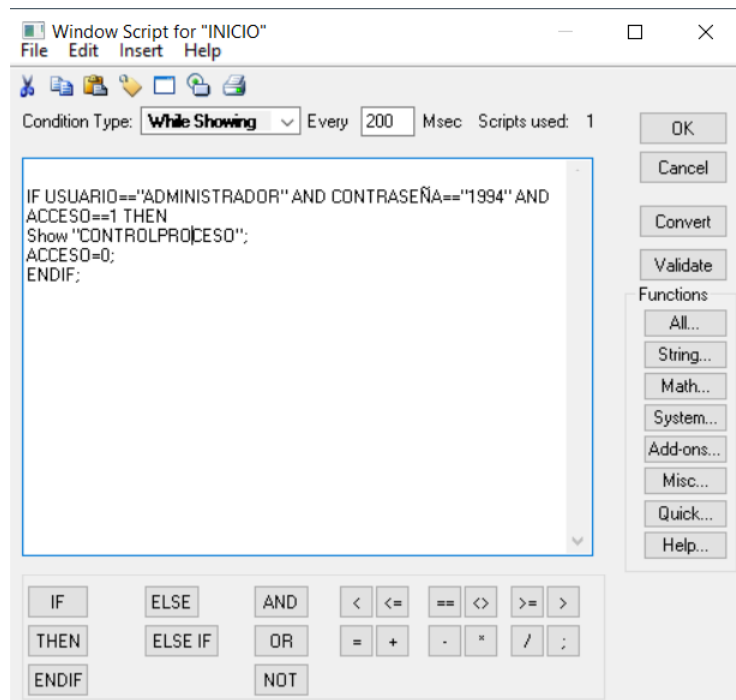


Iniciada la sesión remota es posible editar, crear, mover, duplicar y eliminar elementos del Raspberry Pi 3. En el proyecto, WinSCP es usado para la creación y edición del archivo de Python.

Anexo 6: Programación de transporte de mangos en software DOPSoft.

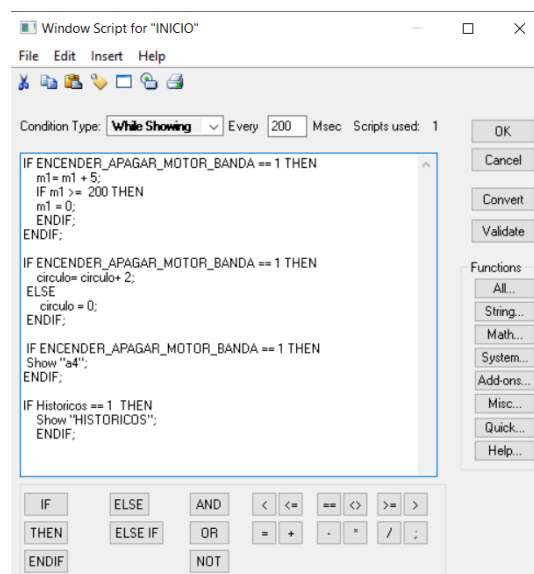


Anexo 7: Programación de usuario y contraseña en software DOPSoft.





Anexo 8: Programación de transporte de mangos en software Wonderware Intouch.



Anexo 9: Prototipo de banda transportadora y clasificadora de mangos.

