



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE MECATRÓNICA

**DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO DE
IMÁGENES PARA LA CLASIFICACIÓN DE LIMONES USANDO
RASPBERRY PI 4 CON PYTHON**

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

AUTORES: Tamara Sugey Jiménez Yagual
David Abraham Merchán Echeverría
TUTOR: David Mateo Cortez Saravia

Guayaquil - Ecuador
2024

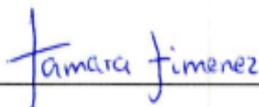
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Tamara Sugey Jiménez Yagual** con documento de identificación N° **0951009323** y **David Abraham Merchán Echeverría** con documento de identificación N° **0930545470**; manifestamos que:

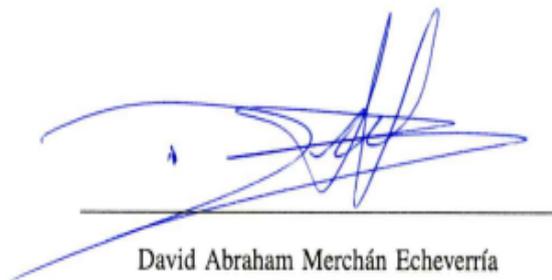
Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo.

Guayaquil, 30 de agosto del año 2024

Atentamente,



Tamara Sugey Jiménez Yagual
0951009323



David Abraham Merchán Echeverría
0930545470

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

Nosotros, **Tamara Sughey Jiménez Yagual** con documento de identificación N° **0951009323** y **David Abraham Merchán Echeverría** con documento de identificación N° **0930545470**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del **Dispositivo Tecnológico: DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO DE IMÁGENES PARA LA CLASIFICACIÓN DE LIMONES USANDO RASPBERRY PI 4 CON PYTHON**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

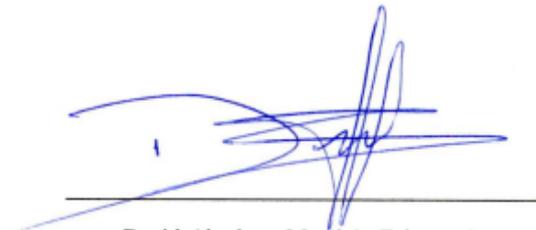
En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 30 de agosto del año 2024

Atentamente,



Tamara Sughey Jiménez Yagual
0951009323



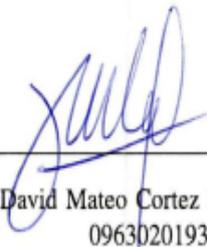
David Abraham Merchán Echeverría
0930545470

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **David Mateo Cortez Saravia**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO DE IMÁGENES PARA LA CLASIFICACIÓN DE LIMONES USANDO RASPBERRY PI 4 CON PYTHON**, realizado por **Tamara Sugey Jiménez Yagual** con documento de identificación N° **0951009323** y por **David Abraham Merchán Echeverría** con documento de identificación N° **0930545470**, obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivo Tecnológico** que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 30 de agosto del año 2024

Atentamente,



Ing. David Mateo Cortez Saravia, MSc.
0963020193

DEDICATORIA

Dedico esta etapa universitaria principalmente Dios, ya que gracias a él estoy cumpliendo una meta más en mi vida, también por la fortaleza y entendimiento que me ha dado durante este proceso.

Y a mi madre Martha Yagual por su apoyo y amor en absoluto a lo largo de mi vida; a mi tía Delia Yagual por guiarme en todo momento y estar siempre ahí cuidándome.

Tamara Sugey Jiménez Yagual

A mi padres, David y Angélica, quienes han sido un apoyo en mi vida y en carrera profesional, para que pueda alcanzar mis objetivos y metas propuestos. Cada paso que he dado en este largo camino ha sido posible gracias a su aliento constante y su fe en mí. Su confianza en mí fue lo que me impulso a no rendirme y continuar a pesar de los altos y bajos que tuve a lo largo de la carrera. A mi increíble familia, gracias por ser mi inspiración, cada logro académico lleva la huella de su amor y sacrificio. Gracias por creer en mí y por motivarme a perseguir mis sueños. Con mucho amor les dedico este inmenso logro.

David Abraham Merchán Echeverría

AGRADECIMIENTO

Agradezco a mis hermanos y familiares quienes han sido ese apoyo y motivación durante mi carrera universitaria.

A nuestra Universidad Politécnica Salesiana por darnos una excelente preparación como profesionales y nuestros amigos que formaron parte de este proceso.

Agradezco al Ing. Jonathan Paillacho por la paciencia y el asesoramiento en este trabajo de titulación al igual que el Ing. David Cortez.

Tamara Sugey Jiménez Yagual

Agradezco especialmente a Dios por darme salud y vida. También a mi querida madre, a mi padre, a mis hermanos, quienes fueron parte de mi largo trayecto en la universidad.

Agradezco a la Universidad Politécnica Salesiana, a los profesores y a mis amigos por la motivación dada en este transcurso del presente trabajo de investigación.

David Abraham Merchán Echeverría

RESUMEN

En el presente proyecto se tiene como objetivo la construcción de una banda transportadora automatizada mediante visión artificial logra clasificar limones en categorías de verdes, maduros y rechazos, dando como resultado una clasificación precisa y eficiente. La banda transportadora se encuentra equipada con sensores y cámaras que capturan imágenes de los limones en movimiento, estas son procesadas por un modelo automático de aprendizaje previamente entrenado con la herramienta digital Teachable Machine de Google. Este modelo automático clasifica limones por su color.

El proceso del proyecto empieza con la captura de imágenes en tiempo real a medida que los limones circulan por la banda transportadora. Las imágenes tomadas por la cámara son enviadas a un sistema de identificación que utiliza el modelo de inteligencia artificial previamente entrenada con imágenes de los estados de los limones. Se realiza un análisis de los componentes mecánicos y electrónicos para garantizar un proceso confiable y eficiente. Además, se realiza una investigación sobre los diversos métodos de clasificación de imágenes y técnicas para asegurar que el modelo sea capaz de llevar a cabo manejos en las variaciones de condiciones tales como, la luminosidad ya que puede afectar la identificación del modelo utilizado.

Como resultado, se tiene una banda transportadora automatizada que clasifica limones de manera efectiva mediante el uso de herramientas digitales y componentes electrónicos de fácil asequibilidad en el mercado. Este proyecto innova la manera de clasificar los limones en la industria agronómica que requieren de soluciones automatizadas para lograr un mejor desempeño en el manejo de sus operaciones.

Palabras claves: Visión Artificial, limones, reconocimiento de patrones, clasificación automatizada.

ABSTRACT

The objective of this project is to build an automated conveyor belt using artificial vision to classify lemons into green, ripe and rejected categories, resulting in a precise and efficient classification. The conveyor belt is equipped with sensors and cameras that capture images of the lemons in motion, which are processed by an automatic learning model previously trained with Google's Teachable Machine digital tool. This automatic model classifies lemons by their color.

The project process begins with the capture of real-time images as the lemons move along the conveyor belt. These images captured by the camera are sent to an identification process that uses the artificial intelligence model previously trained with images of lemons' conditions. An analysis of the mechanical and electronic components is carried out to ensure a reliable and efficient process. In addition, different image classification methods and techniques are investigated to ensure that the model is able to handle variations in conditions such as brightness, as this can affect the identification of the model used.

As a result, there is an automated conveyor belt that effectively sorts lemons by using digital tools and electronic components that are easily available on the market. This project innovates the way of sorting lemons in the agricultural industry that requires automated solutions to achieve better performance in the management of their operations.

Keywords: Machine Vision, lemons, pattern recognition, automated classification.

ÍNDICE

I.	Introducción	1
II.	Problema de estudio	2
III.	Justificación	3
IV.	Objetivos	4
IV-A.	Objetivo general	4
IV-B.	Objetivos específicos	4
IV-C.	Matriz de objetivo	4
V.	Marco Teórico	5
V-A.	La Inteligencia Artificial	5
V-B.	IA en la agricultura	5
V-C.	Limón	6
V-C1.	Cultivo	7
V-C2.	Cosecha	7
V-C3.	Post-cosecha	7
V-C4.	Identificación del estado del limón	8
V-D.	Visión artificial	9
V-E.	Herramientas de visión artificial	9
V-E1.	Transfer learning	9
V-F.	Redes neuronales	10
V-F1.	Redes neuronales convolucionales	11
V-F2.	Desarrollo de una red neuronal	11
V-G.	Lenguaje de programación	12
V-G1.	Python	12
V-G2.	TensorFlow	12
V-G3.	Modelo RGB	12
V-H.	Raspberry Pi 4	13
V-H1.	Redes Neuronales en Raspberry Pi 4	13
V-I.	Cámara web	14
V-J.	Servo motor	14
V-K.	Banda transportadora	15
V-L.	Acero inoxidable	15
V-M.	Impresión 3D	16
V-N.	PET-G	16
VI.	Metodología	17
VI-A.	Desarrollo y construcción de un mecanismo para el transporte y clasificación de limones	17
VI-A1.	Modelado en 3D de la banda transportadora	17
VI-A2.	Estructura base	18
VI-A3.	Estructuras adicionales	18
VI-A4.	Soporte de la cámara web	19
VI-A5.	Chumacera	19
VI-A6.	Ejes de acero	20
VI-A7.	Poleas	20
VI-A8.	Banda de caucho	22
VI-A9.	Clasificador	22

VI-B.	Sistema eléctrico	23
VI-B1.	Servo motor	23
VI-B2.	Cámara web	23
VI-C.	Selección y procesamiento de las imágenes	24
VI-D.	Desarrollo de la Red Neuronal	24
VI-E.	Implementación en Raspberry Pi	27
VI-E1.	Implementación de la pantalla	27
VI-E2.	Implementación de servomotores	27
VI-E3.	Implementación de la red	27
VI-F.	Desarrollo de una interfaz gráfica	27
VI-G.	Desarrollo e implementación de un contador de productos	28
VI-H.	Cálculos mecánicos de la banda transportadora	29
VI-H1.	Longitud de la banda	29
VI-H2.	Velocidad de la banda	29
VI-H3.	Tracción efectiva	30
VI-H4.	Tracción ajustada de la banda	30
VI-H5.	Tracción ajustada de la banda por milímetro de ancho de banda	30
VI-H6.	Inercia de carga	31
VI-I.	Simulación del diseño de la banda transportadora	31
VI-I1.	Simulación de torque	31
VI-I2.	Simulación de esfuerzo de eje conducido	32
VI-I3.	Simulación de movimiento de giro del soporte de banda	33
VI-I4.	Simulación de superficie de banda	33
VI-I5.	Simulación de giro de momento de uniones	34
VII.	Resultados	35
VII-A.	Sistema mecánico para el transporte de limones	35
VII-B.	Sistema de reconocimiento de imágenes para la clasificación de limones	35
VII-B1.	Programación implementada en el clasificador de limones	36
VII-C.	Validación del desempeño del clasificador de limones	37
VIII.	Cronograma	39
IX.	Presupuesto	40
X.	Conclusiones	41
XI.	Recomendaciones	42
	Referencias	43
	Anexo A: Planos	46
	Anexo B: Proceso de construcción del proyecto	59
	Anexo C: Código de programación para la Raspberry Pi 4	63

ÍNDICE DE FIGURAS

1.	Dron encargado del monitoreo y exploración de los cultivos [10]	5
2.	Sweeper [12]	6
3.	Limón sutil [14]	6
4.	Diagrama de flujo proceso del limón	7
5.	Limón verde	8
6.	Limón maduro	8
7.	Limón dañado	8
8.	Ejemplo visión artificial [20]	9
9.	Proceso de la cámara [21]	9
10.	Pasos para el entrenamiento de una red con Teachable Machine [22]	10
11.	Modelo de una neurona artificial [24]	10
12.	Red neuronal convolucional [25]	11
13.	Datos para hacer una red neuronal [27]	11
14.	Ejemplo programación en Python [30]	12
15.	Tensorflow [31]	12
16.	Raspberry Pi 4 [37]	13
17.	Estructura de procesamiento para el limón [40]	13
18.	Cámara web [41]	14
19.	Servo motor [42]	14
20.	Banda transportadora [44]	15
21.	PETG [48]	16
22.	Etapas del proyecto. Fuente: Los Autores	17
23.	Etapas 3 Propuesta del mecanismo. Fuente: Los Autores	17
24.	Banda transportadora. Fuente: Los Autores	18
25.	Estructura Base. Fuente: Los Autores	18
26.	Estructuras adicionales. Fuente: Los Autores	19
27.	Soporte de la cámara web. Fuente: Los Autores	19
28.	Chumacera. Fuente: Los Autores	20
29.	Ejes de acero. Fuente: Los Autores	20
30.	Poleas. Fuente: Los Autores	21
31.	Polea motriz. Fuente: Los Autores	21
32.	Polea conducida. Fuente: Los Autores	22
33.	Banda de caucho. Fuente: Los Autores	22
34.	Clasificador. Fuente: Los Autores	23
35.	Imágenes utilizadas para el entrenamiento de la red. Fuente: Los Autores	24
36.	Funcionamiento de la red neuronal. Fuente: Los Autores	24
37.	Valores de entrenamiento de la red neuronal. Fuente: Los Autores	25
38.	Entrenamiento de la red neuronal con teachable machine. Fuente: Los Autores	25
39.	Validación de la red neuronal. Fuente: Los Autores	26
40.	Exportar la red neuronal. Fuente: Los Autores	26
41.	Conexión Raspberry Pi 4 con la pantalla.	27
42.	Conexión Raspberry Pi 4 con los servomotores.	27
43.	Diseño de interfaz gráfica. Fuente: Los Autores	28
44.	Diagrama de flujo de la programación en la Raspberry Pi 4. Fuente: Los Autores	28
45.	Medidas del sistema mecánico.	29
46.	Simulación de torque F1. Fuente: Los Autores	31
47.	Simulación de torque F2. Fuente: Los Autores	32
48.	Simulación de esfuerzo de eje conducido. Fuente: Los Autores	32
49.	Simulación de movimiento de giro. Fuente: Los Autores	33

50.	Simulación de superficie de banda. Fuente: Los Autores	33
51.	Simulación de giro de momento de uniones. Fuente: Los Autores	34
52.	Prototipo implementado. Fuente: Los Autores	35
53.	Interfaz gráfica. Fuente: Los Autores	35
54.	Configuración limones verdes. Fuente: Los Autores	36
55.	Configuración limones dañados. Fuente: Los Autores	36
56.	Configuración limones maduros. Fuente: Los Autores	37
57.	Prueba de validación 1.	37
58.	Prueba de validación 2.	38
59.	Prueba de validación 3.	38
60.	Banda. Fuente: Los Autores	46
61.	Polea motriz. Fuente: Los Autores	47
62.	Polea conducida. Fuente: Los Autores	48
63.	Base. Fuente: Los Autores	49
64.	Rampa lateral A. Fuente: Los Autores	50
65.	Rampa lateral B. Fuente: Los Autores	51
66.	Base posterior. Fuente: Los Autores	52
67.	Brazo. Fuente: Los Autores	53
68.	Eje 1. Fuente: Los Autores	54
69.	Eje 2. Fuente: Los Autores	55
70.	Soporte servomotor. Fuente: Los Autores	56
71.	Tapa. Fuente: Los Autores	57
72.	Vista explosiva de los elementos utilizados del proyecto. Fuente: Los Autores	58
73.	Base del prototipo. Fuente: Los Autores	59
74.	Acople de los servomotores 180. Fuente: Los Autores	59
75.	Ensamblaje de la banda. Fuente: Los Autores	60
76.	Interfaz gráfica del prototipo. Fuente: Los Autores	60
77.	Pruebas de clasificación. Fuente: Los Autores	61
78.	Servomotor 360. Fuente: Los Autores	61
79.	Pruebas de prototipo con interfaz gráfica. Fuente: Los Autores	62
80.	Instalación de Raspberry Pi 4. Fuente: Los Autores	62

ÍNDICE DE TABLAS

I.	Tabla de matriz de objetivo. Fuente: Los Autores	4
II.	Propiedades del acero inoxidable 304 [45]	15
III.	Propiedades del PETG [47]	16
IV.	Características de los servomotores MG996R 180 grados y MG996R 360 grados	23
V.	Webcam-Logitech (C170) [45]	23
VI.	Media y mediana de medición del error porcentual	37
VII.	Resultado finales de validación	38
VIII.	Cronograma	39
IX.	Presupuesto	40

I. INTRODUCCIÓN

En un mundo de constante transformación en que las tecnologías se han convertido en grandes herramientas dentro del sector productivo por su capacidad para interpretar y clasificar imágenes de manera precisa lo cual ha abierto las puertas a variedades de aplicaciones en diversas industrias. Por tal razón expresada se desarrolló a escala una máquina clasificadora de limones con el objetivo de poder implementar un sistema que sea capaz de identificar automáticamente las diferentes variedades de limones como son limones verdes, limones maduros y los limones dañados, a partir de imágenes capturadas por una cámara conectada al Raspberry Pi 4, para facilitar la clasificación de estos productos y agilizar el proceso de su selección.

Se optó por elegir los limones como objeto de clasificación debido a su gran importancia económica en la industria alimentaria y agrícola ecuatoriana, ya que la mayoría de la industria culinaria en el país emplea el limón como un ingrediente, lo cual genera grandes ingresos económicos. En la clasificación de limones el proceso de reconocimiento de imágenes tiene el potencial de mejorar la precisión de la selección y clasificación del sector agrícola, lo que conllevará a beneficios económicos significativos al reducir el tiempo para el proceso de clasificación.

El uso de Raspberry Pi 4 como plataforma de hardware ofrece una solución y es económico ya que puede implementarse fácilmente dentro de los entornos agrícolas y más aún si se lo acompaña con un lenguaje de programación como es Python que proporciona un amplio catálogo de bibliotecas y herramientas para el uso de algoritmos durante procesamiento de imágenes.

II. PROBLEMA DE ESTUDIO

La clasificación precisa de la madurez es esencial en la industria agrícola, ya que asegura una buena calidad y una adecuada distribución en los procesos de las cosechas [1]. En el estado actual de las técnicas convencionales de clasificación de madurez de las frutas presentan desafíos significativos, la dependencia de procesos manuales y subjetivos ha sido identificada como una fuente clave de inconsistencias y vulnerabilidad a errores humanos; por este motivo la falta de precisión y objetividad en la evaluación de la madurez de las frutas pueden tener consecuencias importantes para las industrias agrícolas [2].

La carencia de un sistema automatizado y preciso se traduce a una serie de limitaciones para los productores. En primer lugar, la toma de decisiones informadas sobre el momento adecuado de la cosecha se ve comprometida, la falta de objetividad en la clasificación de la madurez puede llevar a cosechas prematuras o tardías, afectando directamente la condición y utilidad de dichos productos agrícolas [2].

Además, la ausencia de automatización impacta especialmente en la distribución agrícola. La falta de un sistema que proporcione mediciones precisas y consistentes de la madurez de limones puede generar retrasos, pérdida de calidad y mayores costos operativos en las distintas etapas de la producción; la insuficiencia de un sistema automatizado presenta una serie de desafíos y restricciones que no solo afecta la eficiencia de la cadena de suministro, sino también perjudica la competitividad del mercado agrícola en general [3].

III. JUSTIFICACIÓN

El presente estudio busca abordar este problema mediante el desarrollo de un prototipo mecatrónico en la cual integre tecnologías como Raspberry Pi e inteligencia artificial. La aplicación de visión artificial y su aprendizaje automatizado permite una clasificación objetiva de la maduración de los limones, mejorando la eficiencia del proceso y proporcionando a los productores un material confiable que ejecute de manera eficaz los cultivos y la cadena de suministro agrícola [4].

La clasificación de la madurez de los limones es un componente para la optimización eficiente de los procesos agrícolas. Desempeña un papel significativo al brindar información precisa y rápida, lo que, a su vez, contribuye a una toma de decisiones mas informada en todas las etapas desde la cosecha hasta la distribución de productos agrícolas [5].

La implementación de un prototipo para la clasificación de manera automatizada no solo acelera los procedimientos, sino que también mejora la precisión general. Al eliminar la influencia de los errores humanos asociados con las técnicas manuales de clasificación y se adopta la tecnología que desempeña un papel clave en el rendimiento y la minimización de los costos operativos, se garantiza una mayor consistencia y confiabilidad en los resultados obtenidos. Este enfoque basado en la automatización no solo ahorra tiempo, también minimiza los riesgos de inconsistencias en la calidad de los productos [6].

En el ámbito de este sistema mecatrónico de una maquina clasificadora de la limones mediante el empleo de Raspberry Pi e inteligencia artificial, se requiere la necesidad de mejorar el rendimiento y los tiempos de respuesta en la selección de los limones en las industrias agrícolas logrando así un resultado eficaz al momento de evaluar la calidad del producto.

IV. OBJETIVOS

IV-A. *Objetivo general*

Desarrollar un prototipo a escala de una máquina clasificadora de limones verdes, maduros y dañados basado en visión artificial mediante programación Python en Raspberry Pi 4.

IV-B. *Objetivos específicos*

- Construir un sistema mecánico para el transporte de limones que permita la implementación de una cámara.
- Desarrollar un sistema de reconocimiento de imágenes basado en visión artificial para la clasificación de limones con Raspberry Pi 4.
- Validar el desempeño del sistema mediante la exactitud de selección entre limones verdes, maduros y rechazados.

IV-C. *Matriz de objetivo*

Tabla I
TABLA DE MATRIZ DE OBJETIVO. FUENTE: LOS AUTORES

OBJETIVO	PLANTEAMIENTO	META	INDICADOR
Construir un sistema mecánico para el transporte de limones que permita la implementación de una cámara.	Crear un mecanismo que optimice el diseño detallado del sistema de transporte y del soporte de la cámara.	Lograr un rendimiento que permita el transporte de los limones con un margen de error inferior al 10 %.	Eficiencia del sistema mecánico de transporte de limones.
Desarrollar un sistema de reconocimiento de imágenes basado en visión artificial para la clasificación de limones con Raspberry Pi 4.	Implementar un sistema de identificación para diferenciar limones verdes, maduros y dañados en imágenes.	Alcanzar 85 % de eficiencia en la identificación de limones.	Presión del sistema de reconocimiento imágenes en la identificación de limones.
Validar el desempeño del sistema mediante la exactitud de selección entre limones verdes, maduros y dañados.	Realizar pruebas exhaustivas del sistema integrado de clasificación de limones.	Obtener una exactitud mayor del 80 % en la clasificación de limones.	Tasa de precisión del sistema integrado.

V. MARCO TEÓRICO

V-A. *La Inteligencia Artificial*

Son máquinas creadas por humanos pueden hacer algo más que un trabajo que requiere mucha mano de obra; lo que se busca es desarrollar Inteligencia similar a la humana. Conscientemente o no, la IA ha penetrado en nuestra vida diaria, desempeñando papeles novedosos en la industria, la salud, atención, transporte, educación y muchas más áreas cercanas al público en general. Es por eso que la IA es uno de los más importantes impulsos para cambiar la vida del cualquier persona pero además la IA contribuye al avance de la tecnología, comúnmente en forma de herramientas útiles para investigaciones innovadoras. Durante las últimas décadas, la IA ha atravesado etapas históricas de grandes avances e innovaciones [7].

La actual revolución tecnológica se enfoca en la IA, en dotar a las máquinas con capacidades cognitivas equiparables a los humanos. Se entiende que IA es la principal tecnología que realiza tareas que requerían intervención humana, como el aprendizaje automático y profundo [8].

Otra forma de verlo es que la IA nos centramos en las decisiones y en la resolución de los problemas mediante ciertos métodos cuya aplicación son de aprendizajes automático y logarítmico que posibilitan a tener un mejor rendimiento a medida que acumulan más datos y experiencia [9].

V-B. *IA en la agricultura*

La inteligencia artificial puede ser utilizada para incrementar la eficiencia en la agricultura, seguridad, potenciar la sostenibilidad y la garantía de los productos. No obstante, se debe estar pendiente de la legislación aplicable, pues se trabaja con datos masivos, aspectos que van de la mano con la inteligencia artificial. En esta nueva revolución tecnológica, lo que se busca es que el agricultor se valga de herramientas tecnológicas para mejorar su eficiencia y tiempo [10].

Algunos dispositivos de inteligencia artificial suelen predecir acontecimientos, como por ejemplo, futuras inclemencias meteorológicas, alteraciones del suelo por contaminación o plagas, Otros dispositivos utilizados pueden estar la programación el riego, determinar la medida adecuada para la fertilización de los cultivos, uso de drones que determinen el estado de productos como se puede ver en la figura 1 [10].



Figura 1. Dron encargado del monitoreo y exploración de los cultivos [10]

Actualmente, en la agricultura se están empleando los drones como herramientas para realizar recorridos más rápidos y revisar la situación de los cultivos desde la parte aérea proporcionando una visualización rápida y completa de los campos de cultivo no solo limitándose a ser simples cámaras de vídeo, sino que, al integrarles inteligencia artificial, pueden evaluar parámetros de gran importancia para los agricultores. Este aporte se da en dos aspectos el primero es la evaluación de la salud de las plantas según el color de sus hojas y el otro aporte es la generación de

mapas térmicos del terreno agrícola para evaluar la humedad y sequedad en diferentes áreas del suelo, permitiendo así un tratamiento individualizado para cada área [11].

Entre las grandes propuestas a futuro podremos encontrar proyectos financiados por la Unión Europea que propone la creación de robots que ayudados por inteligencia artificial puedan encargarse de la recolección de productos del campo, tal sea el caso de Sweeper un robot que recolecta pimientos dulces tal como se ve en la figura 2, Internet of Food & Farm, que busca hacer realidad la agricultura de precisión entre otros [12].



Figura 2. Sweeper [12]

V-C. *Limón*

Los limones es la tercer grupo de cítricos más importantes del mundo, solo debajo de las naranjas y las mandarinas, estos requieren de un clima subtropical pero a su vez tropical. Pero en los limoneros en los climas tropicales pueden crecer y dar frutos pero esto no garantiza que dicho producto tenga buena calidad, el cual se tiene una seleccionada area para este tipo de cultivo, ya que el cultivo de limonero tiene un clima más adecuado, y es por eso que las zonas de productores están ubicadas en áreas costeras [13].

La variedad de los limoneros es bastante reducida si los comparamos los naranjos con los mandarinos; las variantes de limones en distintos países son similares. Entre las principales variantes tenemos eureka, lisbon, femminello, verna, fino y génova. Sin embargo, en Ecuador la variedad de cultivo que conocemos es del limón sutil [14].



Figura 3. Limón sutil [14]

Existen 3 etapas en la elaboración de limones, cultivo, cosecha y post-cosecha. La etapa de cultivo es el desarrollo de la plantación y suele tener un tiempo de duración de 2 a 3 años, por otro lado, la cosecha y post-cosecha inician el 4 año. Desde este año, la vida económica de la planta es permanente [15].

V-C1. *Cultivo:*

En esta etapa se cultiva el limón por el método deseado [16]. En el caso del limón sutil, se utilizan las ramas de una planta adulta echan raíces, para lo cual se siguen los siguientes pasos:

1. Se obtiene la semilla de portainjertos seleccionada, se divide el limón, se saca la semilla y se deja secar.
2. Inicia la germinación, inicia a los 20 a 30 días y termina a los 40.
3. Las plantas se mantiene cultivada por un tiempo de entre 8 a 10 meses. Cuando alcanza los 20 a 30 cm, se realiza un trasplante a macetas de plástico negro.
4. Se realiza la injertación, una vez sé que el portainjerto alcanza el grosor deseado comúnmente a los 12 meses.
5. Se espera a que el limonero esté en condiciones de trasplante definitivo, que es de alrededor de 18 a 24 meses.
6. Se realizan procedimientos de cuidado como el riego y la restitución anual de nutrientes.

V-C2. *Cosecha:*

Este proceso empieza cuando el limón llega a su madurez, es decir, que sea color verde, forma redondeada, tenga piel lisa y la acidez esté entre un 4 y 7% y sin visos amarillos. Se debe tener cuidado, pues si el limón permanece en el árbol mucho tiempo se corre el riesgo de que tome un color amarillo al llegar a los mercados, es decir que maduro demasiado, además, se debe tener cuidado en no golpear ni maltratar la fruta y se recomienda realizar la cosecha después de las primeras horas de la mañana y cuando estén secas. Otra cosa a tener en cuenta es que se tiene que evitar dejar al limón al sol y mantenerlo en lugares sombreados. La etapa de cosecha es realizada manualmente para evitar dañar el limonero [17].

V-C3. *Post-cosecha:*

En esta etapa se realizan los cuidados después de haber realizado la cosecha, para lo cual enumeraremos los siguientes pasos [18].

En primer lugar las operaciones de cosecha, como transporte, cosecha y recolección, estas son importantes para garantizar la recolección del producto, teniendo en cuenta que la manipulación y almacenamiento cumplen operaciones como limpieza, secado, selección, almacenamiento y fumigación. Luego el procesamiento y preparación de productos alimenticios basados en limones. Finalmente, el almacenamiento, los limones son llevados al granel previa a su clasificación en verdes y amarillos. El tiempo de duración de la fruta almacenada suele ser de 6 a 8 meses.

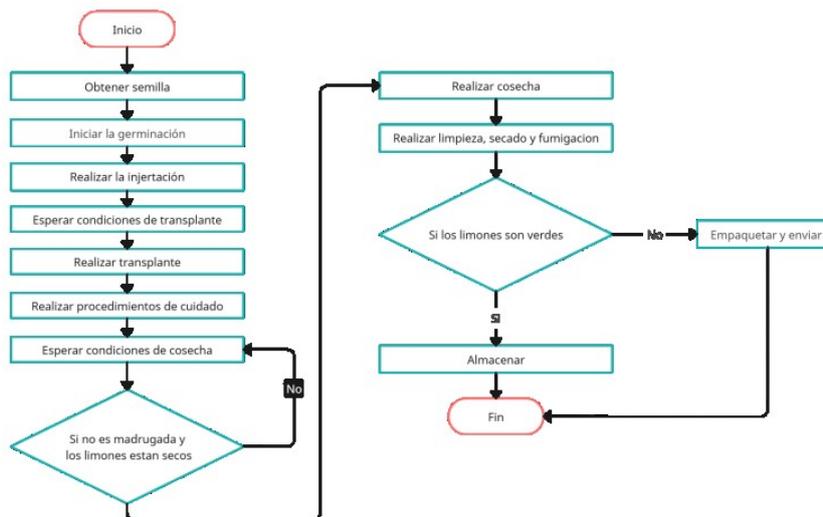


Figura 4. Diagrama de flujo proceso del limón

V-C4. Identificación del estado del limón:

Los limones en esta clasificación se pueden presentar en diferentes maneras dependiendo según su grado de condición que deseamos mostrar [19]. A continuación las diferencias entre un limón verde, maduro y dañado:

- Limón verde

Color: Verde Uniforme.

Textura: Firme y dura al tacto.

Cáscara: Lisa o ligeramente rugosa.



Figura 5. Limón verde

- Limón maduro

Color: Amarillo brillante.

Textura: Firme pero ligeramente más suave que el limón verde.

Cáscara: Suave o ligeramente rugosa, con un brillo natural.



Figura 6. Limón maduro

- Limón dañado

Color: Manchas marrones, negras o descoloridas en la cáscara.

Textura: Blando o esponjoso al tacto, con posibles zonas podridas.

Cáscara: Puede estar arrugada, seca, o con moho.



Figura 7. Limón dañado

V-D. Visión artificial

La rama de visión artificial en la inteligencia artificial cuyo enfoque se basa en entender y replicar los procesos de captación visual de los seres vivos mediante modelos matemáticos desarrollando herramientas que puedan simular estas capacidades visuales en el computador permitiendo así la interpretación del mundo tridimensional a partir de imágenes bidimensionales, ya sea a color o en su defecto en blanco y negro, sean estas captadas por una o varias cámaras a la vez, y estas pueden ser estáticas o dinámicas [20]. En términos de uso, la visión artificial se enfoca en determinar propiedades de los materiales, como su color y textura. En aplicaciones industriales, la visión artificial abarca disciplinas como principalmente la informática, automatización industrial, dispositivos médicos y la óptica. A continuación, se muestra en la figura 8 un ejemplo sobre como se puede utilizar la visión artificial para detectar automóviles y las señales de tránsito.

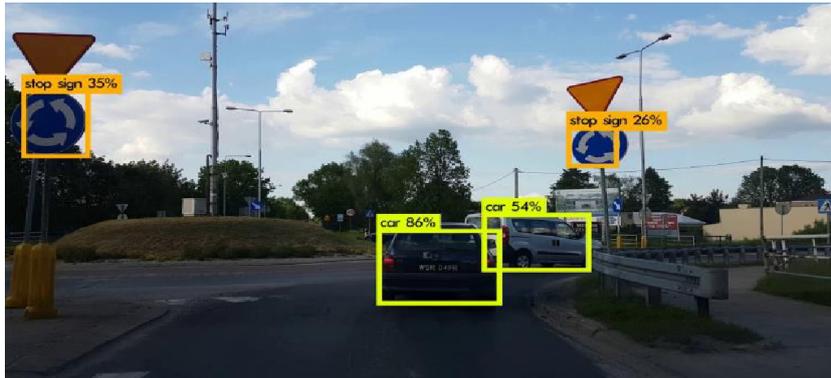


Figura 8. Ejemplo visión artificial [20]

V-E. Herramientas de visión artificial

La cámara es un periférico que establece conexión con el controlador, otorgándole una entrada a todo píxeles obtenidos por la visión de la cámara; se puede lograr diversos usos con estos elementos, brindando a los usuarios un servicio ya sea operativo o de visión artificial tenido en sí la capacidad de desarrollar soluciones personalizadas para sus proyectos específicos [21].

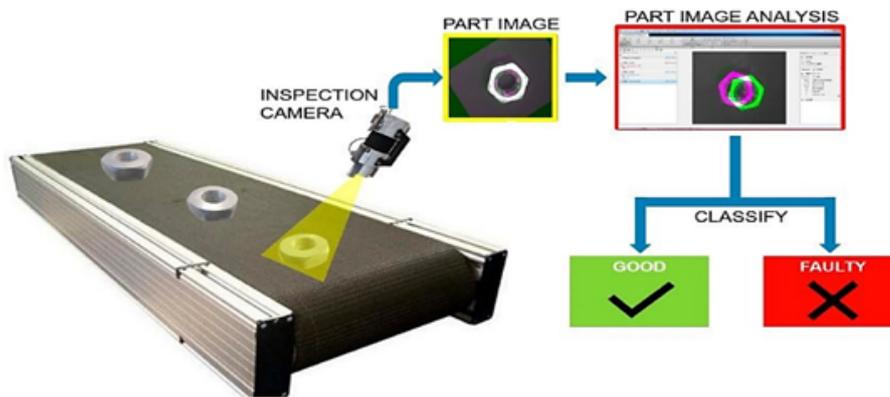


Figura 9. Proceso de la cámara [21]

V-E1. Transfer learning:

Esta técnica de aprendizaje automático utiliza modelos pre-entrenados, en este caso Google a desarrollado la herramienta Teachable machine, la cual puede ser utilizada en la web. Esta herramienta utiliza tecnologías tales

como TensorFlow.js y tiene como objetivo el poder crear modelos de aprendizaje automático que puedan ser realizados de manera, fácil, rápida y además ser accesible para todos, sin necesidad de tener de un conocimiento profundo sobre el tema o saber codificar [22]. A continuación. Se presentan las fases para utilizar teachable machine.

- **Gather:** Agrupar los ejemplos en clases o categorías en las que se desea que la computadora aprenda.
- **Train:** Entrenar el modelo y posteriormente probarlo con nuevos ejemplos para ver si es capaz de clasificarlos correctamente.
- **Export:** Exportar el modelo para el proyecto en el que se está trabajando: sitios web, aplicaciones y más. Se puede descargar el modelo y alojarlo en línea.

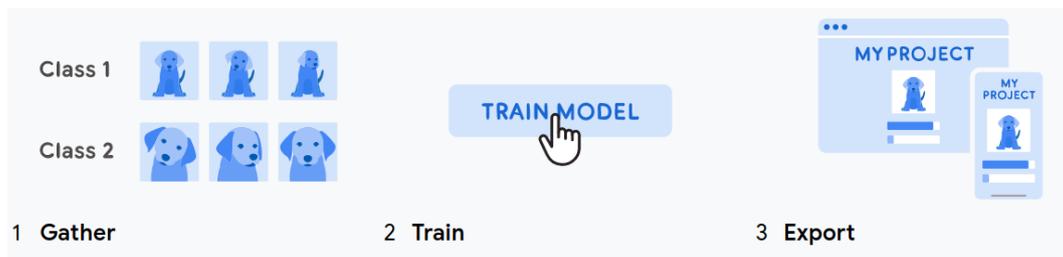


Figura 10. Pasos para el entrenamiento de una red con Teachable Machine [22]

Google Teachable Machine es una herramienta desarrollada por Google, que cumple la función de reconocer imágenes y sonidos. Y cuenta con la facilidad de usar y descargar archivos, permitiendo dar ese toque creativo a los usuarios de aprender de una manera rápida a crear todo mediante un proceso informático ya sea por aplicaciones y utilizando hardware [23].

V-F. Redes neuronales

En cierta forma, las redes neuronales pueden igualar a las estructuras del sistema nervioso, llevando a cabo sistemas de procesamiento idénticos, adaptativos y distribuidos. El elemento central de estas es la neurona artificial, la cual se encuentra agrupada en distintas capas que llegan a formar la red neuronal, y una neurona o varias conectadas pueden desarrollar un sistema solido de procesamiento [24]. A continuación, en la siguiente figura se puede observar el modelo de una neurona común.

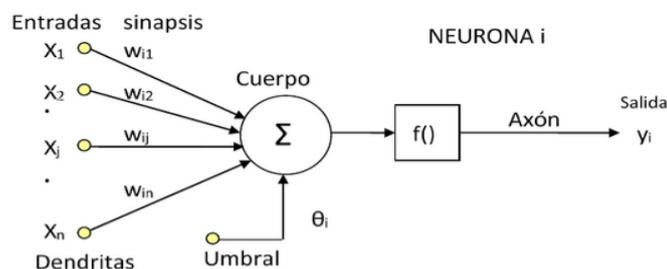


Figura 11. Modelo de una neurona artificial [24]

V-F1. *Redes neuronales convolucionales:*

Las redes neuronales convolucionales, es de las herramientas más representativas en el campo del aprendizaje para la inteligencia artificial. Por otro lado, la visión de imágenes por computador que esta basada en redes neuronales ha ayudado a las personas a desarrollar artefactos útiles para el ámbito de la tecnología y en la vida cotidiana. Estas herramientas son muy útiles para procesos tales como reconocimiento facial, vehículos autónomos y procedimientos médicos inteligentes [25].

Las redes neuronales de tipo convolucionales son redes que se distinguen de otras por su arquitectura y su comportamiento ante diferentes entradas como: audio, voz o imagen. Los tipos principales de capas, como lo son: las capas convolucionales, las capas de agrupamiento y capa total conectada. La capa de tipo convolucional es la primera etapa y es donde sucede la mayor parte de los cálculos de la red, esta capa necesita de algunos componentes, como lo son: los datos de entrada, un filtro de datos y un mapa de las características. La agrupación de estas capas, también conocida como reducción de muestreo, lo que hace es reducir la dimensionalidad lo cual implica reducir la cantidad de parámetros en la entrada de la red, aunque se suele perder mucha información en la capa de tipo agrupamiento, esto ayuda a reducir su complejidad, mejorando la eficacia y limita los riesgos del sobre ajuste. Finalmente, en la capa total conectada, se desarrolla el proceso de clasificación basada en los atributos extraídos a través de las capas anteriores y de sus diferentes filtros. Estas capas tienen sus propias funciones de activación. Por un lado, las capas de convolución y agrupadas usan las funciones ReLu y las capas totales conectadas aprovechan las funciones de activación del softmax para poder realizar la clasificación de las entradas satisfactoriamente, produciendo así una probabilidad de 0 a 1. [26].

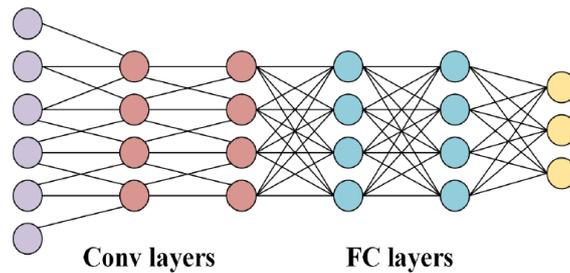


Figura 12. Red neuronal convolucional [25]

V-F2. *Desarrollo de una red neuronal:*

Existen algunos criterios que se necesitan tener en cuenta al momento de trabajar con redes neuronales como lo son los volúmenes de datos, estos están separados por datos de entrada/entrenamiento, validación y pruebas; estos datos de entrenamiento son con los cuales se entrenaran la neurona, los datos de validación sirven para hacer una evaluación imparcial del modelo, y también se tienen los datos de pruebas con los cuales se evaluará el funcionamiento de la red neuronal [27].



Figura 13. Datos para hacer una red neuronal [27]

V-G. Lenguaje de programación

Estos lenguajes tienen una serie de símbolos, normas sintácticas y semánticas que forman las disposiciones necesarias para que las máquinas puedan procesar los códigos para ejecutar una acción deseada; es decir que los lenguajes de programación son el puente para realizar algún cambio dentro de un lenguaje entendible para los humanos a instrucciones legibles por máquinas [28].

V-G1. Python:

Esa programación versátil puede funcionar prácticamente en diversas arquitecturas de sistemas o aplicarse en una variedad de áreas, abarcando desde el desarrollo web hasta el aprendizaje automático, entre otros campos [29]. Se optó por emplear este lenguaje de programación, ya que es compatible con Raspberry Pi 4 lo cual facilita la ejecución de los programas sin necesidad de realizar configuraciones adicionales al sistema, su facilidad de aprendizaje lo convierten en una elección preferida para el uso de distintas aplicaciones específicas como informáticas, domótica y visión artificial [30].

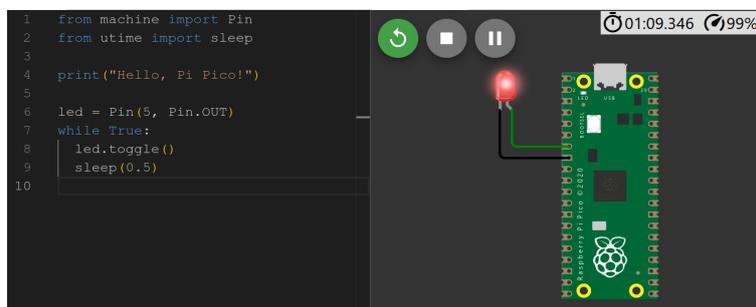


Figura 14. Ejemplo programación en Python [30]

V-G2. TensorFlow:

Es una biblioteca donde permite expresar, implementar o ejecutar algoritmos de aprendizaje automático [31]. Además, contiene una alta gama de aplicaciones, por medio de visión computacional [32]. Por otro lado, este se puede apoyar de OpenCV, que es un código abierto que está orientada al procesamiento de imágenes y la visión por computadora [33].



Figura 15. Tensorflow [31]

V-G3. Modelo RGB:

La visión artificial en conjunto con los modelos de colores que radica especialmente en la manera de capturar y procesar información visual, identificando los patrones sutiles de colores enfocándose en una clasificación más rápida y precisa [34].

En el modelo de colores RGB es importante para la construcción, presentación y manipulación de imágenes digitales, ya que accede a una representación contundente o variada de colores a través de la combinación de estos tres colores que son el rojo, verde y azul [35]. Cada color de la imagen está representado por una combinación,

lo que permite una representación útil en las aplicaciones de visión artificial debido a su capacidad para corregir cambios de color, facilitar la identificación de objetos y mejorar la claridad de las imágenes procesadas [36].

V-H. Raspberry Pi 4

Es una plataforma diseñada para el implementación de elementos y proyectos electrónicos, está equipado con un procesador de cuatro núcleos y un hardware moderno, este dispositivo ofrece tres configuraciones de memoria RAM: 1GB, 2GB o 4GB. Esto proporciona una capacidad gráfica mejorada y contiene una energía de consumo reducido. También, incluye Gigabit Ethernet, Wi-Fi 802.11ac y Bluetooth [37].

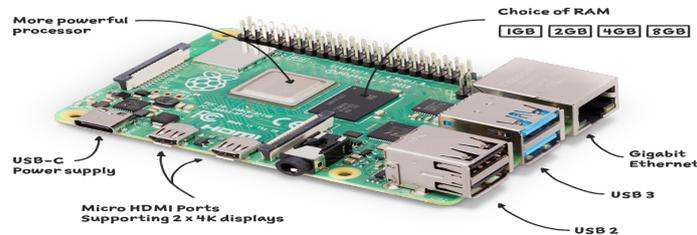


Figura 16. Raspberry Pi 4 [37]

V-H1. Redes Neuronales en Raspberry Pi 4:

Raspberry Pi 4 emplea redes neuronales convolucionales las cuales dentro del ámbito del deep learning son ampliamente empleadas y reconocidas. Estas redes están especialmente diseñadas para aplicaciones de visión artificial, utilizando para procesar información a través de imágenes, lo que les otorga su nombre [38].

Esta plataforma se emplea redes neuronales convolucionales las cuales dentro del ámbito del deep learning son ampliamente empleadas y reconocidas. Estas redes están especialmente diseñadas para aplicaciones de visión artificial, utilizando datos de imágenes que captar y procesar dicha información a lo que se les da un nombre [39].

Cuando se trata de la clasificación de limones a través del reconocimiento de imágenes, las redes neuronales juegan un papel fundamental ya que son aquellas que permiten analizar datos visuales para distinguir características clave como puede ser el color, textura y forma del objeto; al momento de implementar este proceso al Raspberry Pi 4, se obtiene una solución eficaz que tiene como resultado un mejor control de la calidad del producto agrícola [40].

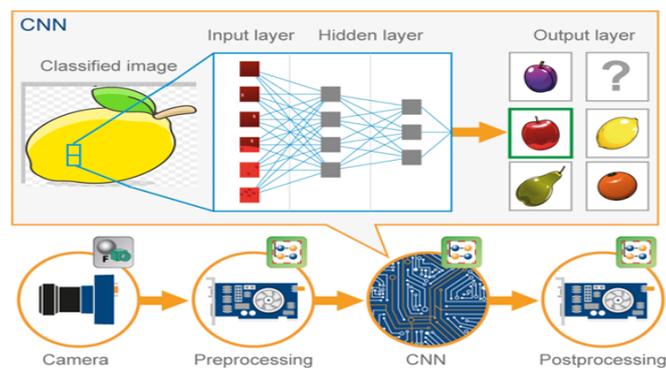


Figura 17. Estructura de procesamiento para el limón [40]

V-I. Cámara web

Al considerar las cámaras web, es esencial tener en cuenta su versatilidad y la diversidad de funciones que ofrecen para adaptarse a distintas necesidades. Elementos son fundamentales al evaluar los modelos a ser seleccionados para este proyecto. El uso de esta cámara web externa en lugar de la integrada en la computadora puede ofrecer una mayor flexibilidad en términos de uso. Las resoluciones de video más comunes suelen ser de 720p, 1080p e incluso 4K, dependiendo de las capacidades específicas de cada dispositivo [41].



Figura 18. Cámara web [41]

V-J. Servo motor

Es un tipo de motor incluye tanto un mecanismo como un algoritmo de control. Esto significa que dentro de este dispositivo mecánico hay un controlador que ajusta con precisión la posición del rotor del motor según la señal de control que recibe como entrada. Sin embargo, los servomotores pueden variar en cuanto a su capacidad de rotación, siendo comunes los modelos de 90, 180°y 360°. Visualmente este servomotor está constituido por tres cables: ground (0v), power (5v), control (PWM) los dos primeros elementos son empleados como fuente de energía, mientras que el cable de control se utiliza para transmitir la señal de ubicación del motor mediante una serie de pulsos PWM [42].



Figura 19. Servo motor [42]

V-K. Banda transportadora

La Banda transportadora es utilizada esencialmente para el transporte eficiente de materiales en diversas industrias. La composición de la banda puede variar, desde tela engomada, plásticos modulares hasta de materiales como caucho, adaptándose a diferentes necesidades industriales [43].

Las Bandas de caucho en particular poseen propiedades que las hacen resistente al calor, al desgaste, a quemaduras, aceites, álcalis, y cambios de temperatura. Principalmente empleada en sectores como minería, metalurgia, acero, energía, construcción, química, alimentaria, entre otros, para el traslado de materiales sólidos. Fabricada con múltiples capas de tela EP, garantiza durabilidad y eficiencia en su funcionamiento [44].



Figura 20. Banda transportadora [44]

V-L. Acero inoxidable

Se destaca por su capacidad para resistir la corrosión, gracias a la presencia de cromo en su composición. Esta aleación presenta una variedad de grados con diferentes mezclas de elementos como molibdeno, níquel y titanio. Entre los tipos más frecuentes se encuentran el 304 y el 316, siendo este último más resistente a la corrosión debido a la inclusión de molibdeno [45].

Tabla II
PROPIEDADES DEL ACERO INOXIDABLE 304 [45]

Propiedad	Valor
Densidad	7,93 g/cm ³
Punto de fusión	1398 – 1454 °C
Calor específico	500 J/(kg · K) a 20°C
Resistividad eléctrica	0,73 μΩ · m a 20°C
Permeabilidad magnética	1,02 (aproximado)
Módulo elástico	193 GPa (28 × 10 ⁶ psi)
Difusividad térmica	3,84 mm ² /s
Coefficiente de conductividad térmica	16,3 a 100°C 21,5 a 500°C
Coefficiente de dilatación lineal	17,2 (0 – 100°C) 17,8 (0 – 300°C) 18,4 (0 – 500°C)

V-M. Impresión 3D

Las impresoras 3D tiene un método de funcionamiento mediante fabricación por adición, que construye objetos añadiendo capas sucesivas de material como polímeros. Han encontrado aplicaciones en campos diversos, desde el arte hasta la medicina, donde han permitido la creación de prótesis económicas. Incluso se han utilizado en la Estación Espacial Internacional para fabricar herramientas [46].

V-N. PET-G

El PETG es preferido sobre el PET en la impresión 3D debido a que es más resistente al sobrecalentamiento, lo cual evita que se vuelva frágil como el PET. Esta resistencia se atribuye a su contenido de glicol. El PETG es más duradero, resistente al desgaste y los impactos. Otras ventajas incluyen su capacidad de esterilización y su capacidad para ser soldado por radiofrecuencia. Funcionalmente, el PETG es similar al ABS y fácil de imprimir como el PLA. Algunas de sus propiedades pueden verse en la tabla [47].

Tabla III
PROPIEDADES DEL PETG [47]

Propiedad	Valor
Densidad	1,27 g/cm ³
Punto de fusión	260 – 280 °C
Calor específico	1,3 J/(g · K)
Resistividad eléctrica	10 ¹⁵ – 10 ¹⁷ Ω · cm
Permeabilidad magnética	No magnético
Módulo elástico	2,1 – 3 GPa
Difusividad térmica	No disponible
Coefficiente de conductividad térmica	0,15 – 0,22 W/(m · K)
Coefficiente de dilatación lineal	80 – 180 × 10 ⁻⁶ /K

También muestra una buena adhesión entre capas, poca deformación, y resistencia a temperaturas bajas, resistencia química y resistencia al agua, además de no emitir olores durante la impresión. Se utiliza principalmente para piezas que requieren flexibilidad, resistencia a impactos y que pueden ser reciclables, como piezas sujetas a presión, protectoras o recipientes para alimentos [48].



Figura 21. PETG [48]

VI. METODOLOGÍA

La metodología empleada en este estudio se tiene propuesto implementar un sistema eficiente de la clasificación de limones en función de su estado mediante la identificación de imágenes. Esto se logra al aprovechar algoritmos y técnicas de visión por computadora, con esta tecnología se busca lograr una alta precisión en la clasificación de limones, lo que puede tener implicaciones significativas para industrias como las de la agricultura o el procesamiento de alimento. El proceso consta de algunas etapas como se muestra en la figura 22.

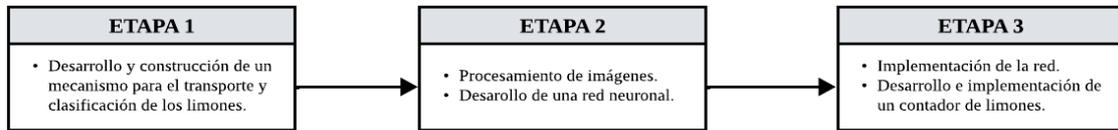


Figura 22. Etapas del proyecto. Fuente: Los Autores

VI-A. Desarrollo y construcción de un mecanismo para el transporte y clasificación de limones

El mecanismo que se decidió utilizar para la clasificación de limones se puede observar en la figura 23. Este mecanismo consta de una banda transportadora, la cuál será la encargada de transportar los limones. Posteriormente una cámara obtendrá las imágenes a la red neuronal la cual determinará por un lado si el limón está dañado o maduro y si ese es el caso entonces se activarán servomotores para redirigir los limones a cajas específicas. Por otro lado, si la red determina que el limón está en buen estado no se activará nada, pues el final de la banda transportadora se dirige a una caja dónde serán depositados los limones que se encuentran en buen estado.

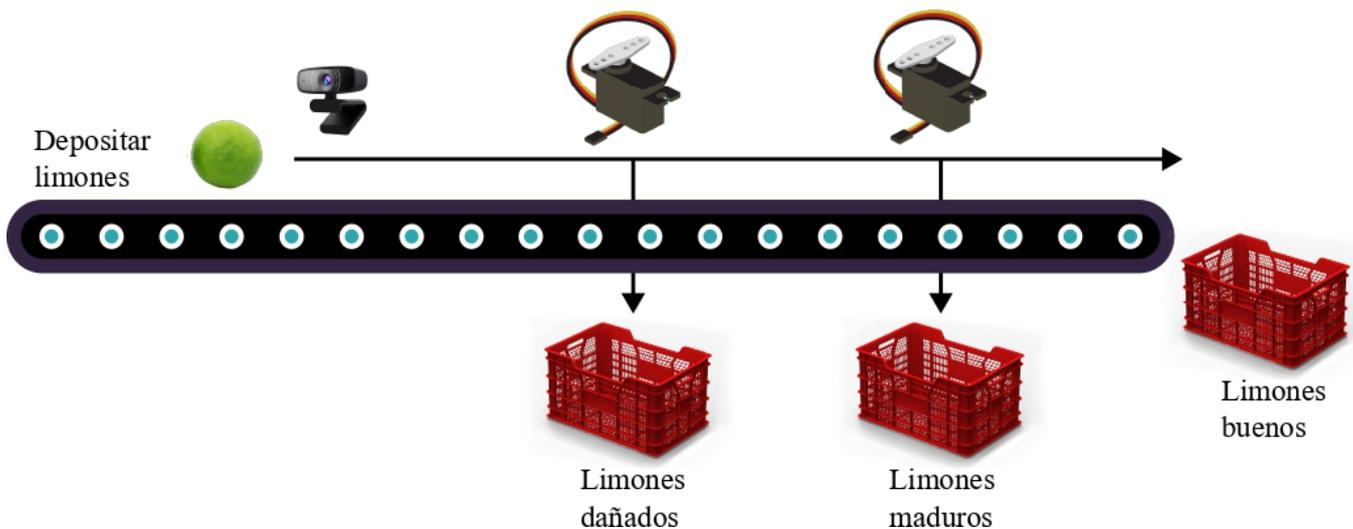


Figura 23. Etapa 3 Propuesta del mecanismo. Fuente: Los Autores

VI-A1. Modelado en 3D de la banda transportadora:

Este modelo fue diseñado en el software de modelamiento CAD, mismo en el que se han estructurado cada una de las piezas en base a la revisión de estudios bibliográficos que han permitido su correcto funcionamiento. Los componentes utilizados cuentan con su respectivo ensamblaje, permitiendo su simulación antes de ser testeado, utilizado y ejecutado de manera física, evitando así pérdidas económicas y reduciendo el tiempo de la prueba y error. La banda transportadora se integra de 22 elementos, clasificándose en 6 elementos que conforman el sistema eléctrico y 16 elementos que pertenecen a la estructura mecánica.

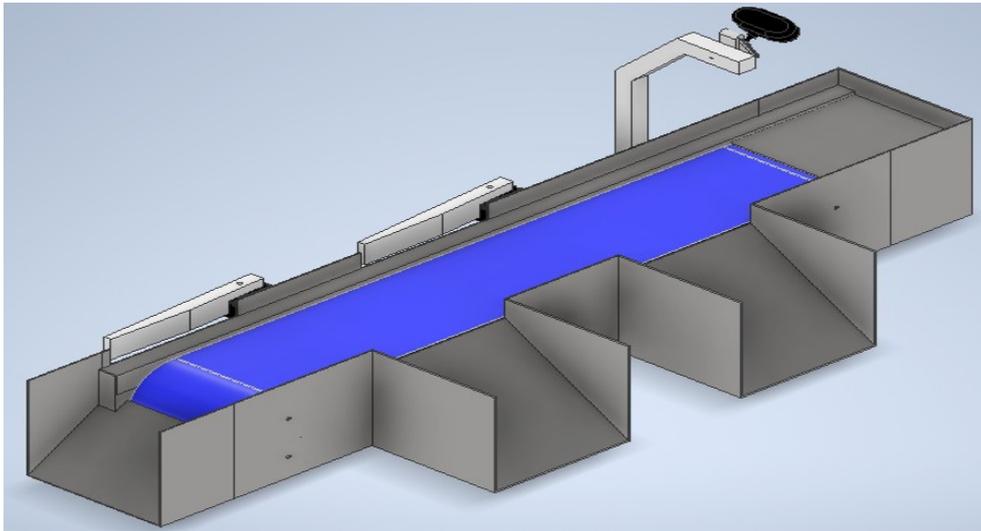


Figura 24. Banda transportadora. Fuente: Los Autores

VI-A2. Estructura base:

La estructura base, fabricada en acero inoxidable no solo soporta físicamente los componentes, sino que también facilita su integración y alineación precisa. Cada elemento, desde los motores hasta la banda y las poleas, se monta en un orden específico dentro de la estructura para asegurar una operación sincronizada. La estructura incluye soportes y montajes diseñados para cada componente, garantizando que estén fijos en su lugar y que puedan operar sin interferencias. Los soportes para los ejes y las poleas constan de puntos de ajuste para una calibración de la tensión, de la misma manera están alineados de manera precisa para asegurar que la banda transportadora se mantenga tensa y en movimiento constante, evitando deslizamientos y desalineaciones. La estructura está diseñada con consideraciones para el mantenimiento regular, permitiendo acceder rápidamente a los elementos esenciales como los motores, las poleas y los ejes. Esto permite realizar ajustes, reparaciones y reemplazos sin necesidad de desmantelar todas las partes del sistema.

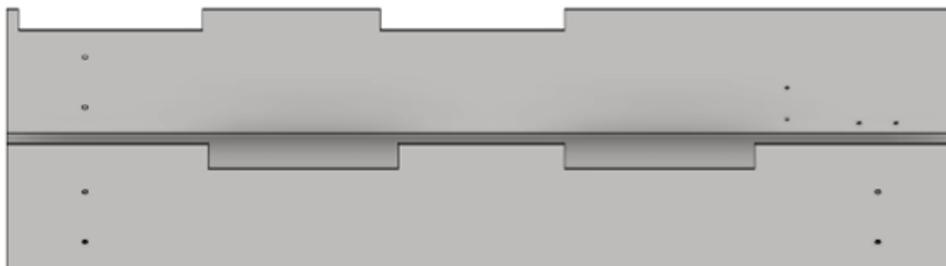


Figura 25. Estructura Base. Fuente: Los Autores

VI-A3. Estructuras adicionales:

Las estructuras adicionales de acero inoxidable están situadas desde el inicio hasta el final de la banda transportadora. A medida que se colocan los limones para su respectiva clasificación se hace uso de la estructura adicional o soporte inicial, seguido a esto se dirige automáticamente por la banda transportadora hasta la estructura tipo rampa correspondiente, en este caso los limones que están actos y cumplen con la calidad se dirigen por la banda de caucho sin ser interrumpidos por los brazos clasificadores, los limones maduros se dirigen a la primer rampa y los dañados usan la segunda rampa desde el punto inicial; por ello son una parte importante del proceso de clasificación garantizando que sólo se envíen limones de alta calidad a los clientes, ahorrando el tiempo para su

selección y almacenamiento. Es necesario que el diseño de estas rampas permita adaptarse al sistema y logre suplir todas las necesidades del proceso de clasificación.

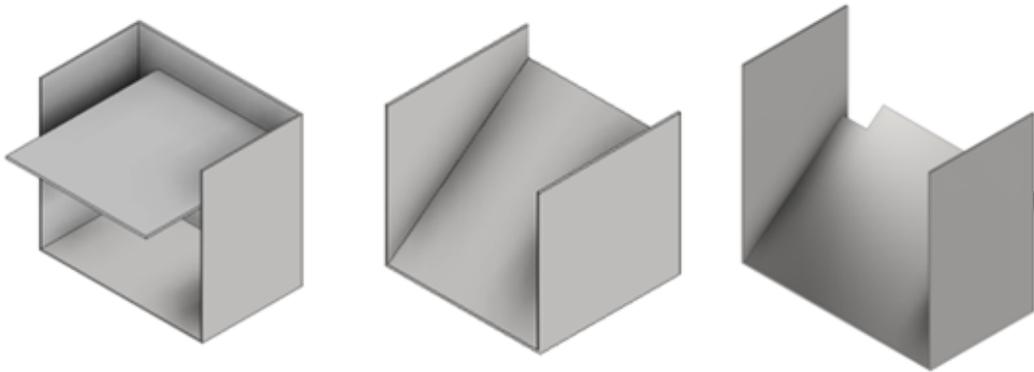


Figura 26. Estructuras adicionales. Fuente: Los Autores

VI-A4. Soporte de la cámara web:

El soporte de la cámara web está diseñado con un tubo de aluminio teniendo una altura x , necesaria para garantizar que el ajuste del posicionamiento sea logrando ya que su altura y ángulo no interfieran con la posición de los actuadores ni con el movimiento o vibración de otro elemento y así obtener una captura de imágenes consistente y precisa indistintamente del movimiento de la banda transportadora, la cual se mantendrá en funcionamiento constantemente.

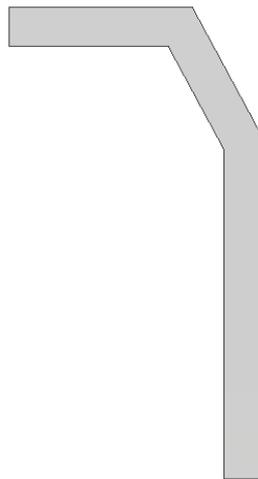


Figura 27. Soporte de la cámara web. Fuente: Los Autores

VI-A5. Chumacera:

Para seleccionar la chumacera adecuada en un mecanismo transportador de limones, se tomó en cuenta el diámetro de las poleas que tienen 10 mm. Siguiendo las especificaciones de montaje, se decidió por el modelo KFL000. Este tipo de chumacera de pared es perfecta para acoplamiento metálicos, está diseñada para ejes de 10 mm y cuenta con una carcasa de aleación de zinc que la hace resistente a la corrosión haciéndola extremadamente duradera. Este material tiene una alta resistencia al corrosión, lo que es particularmente importante en un entorno

donde la humedad y otros elementos pueden afectar los componentes metálicos. La elección de esta chumacera garantiza no solo un montaje adecuado, sino también una operación continua y sin problemas en la banda transportadora.

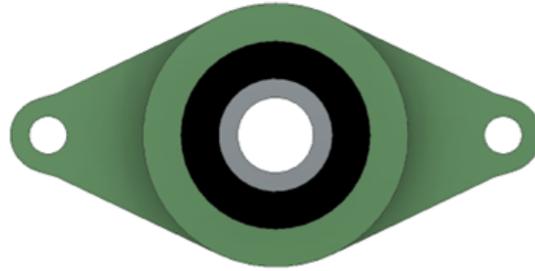


Figura 28. Chumacera. Fuente: Los Autores

VI-A6. Ejes de acero:

Los ejes de acero están diseñados para soportar las fuerzas ejercidas por las poleas y la tensión de la banda transportadora. El diámetro de los ejes, seleccionado en base a cálculos precisos, asegura que puedan manejar la carga sin deformarse ni sufrir desgaste prematuro. En este sistema, los ejes de 10 mm de diámetro se integran perfectamente con las chumaceras KFL000. La elección de acero como material garantiza su uso debido a su alta resistencia, en ambientes donde la humedad y otros factores pueden comprometer la integridad de los componentes metálicos. Además de su función primaria de soporte y transmisión de movimiento como los ejes que cumplen un uso primordial dentro de la integración de todo el sistema de clasificación. Los ejes permiten el montaje adecuado de las poleas y su interacción con la banda de caucho, garantizando que la tensión se mantenga constante y que la banda no se deslice.

Esta integración para mantener la precisión en la clasificación de los productos, asegurando que la cámara web pueda capturar imágenes claras y que los servomotores puedan operar los brazos clasificadores sin interferencias. En la figura 29 se tiene 2 ejes el mas grande mide 18 cm y hace soporte a la polea motriz , mientras el otro eje mide 16 cm y forma parte del soporte de la polea conducida.



Figura 29. Ejes de acero. Fuente: Los Autores

VI-A7. Poleas:

Las poleas de la banda transportadora han sido diseñadas garantizando su movimiento rotacional y a su vez evidenciando la funcionalidad antes de utilizar la impresora 3D utilizando el material Petg que proporciona alta resistencia a desgastes en corto tiempo.

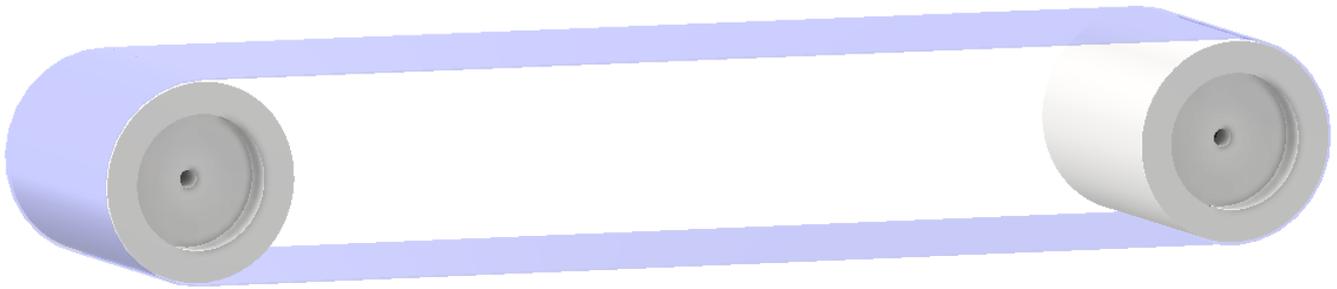


Figura 30. Poleas. Fuente: Los Autores

Polea motriz:

Considerando la optimización de espacio en la anchura de la banda transportadora se evidencia que el diseño de esta polea permite realizar la inserción de la chumacera en su circunferencia de 11cm de diámetro total y una anchura de 13.5cm.

Al tener una trayectoria de 72 cm, no se hace necesario hacer varias poleas motrices, en este sentido la primera polea se ha considerado como la polea motriz la cual va conectada al acople flexible que se ajusta por medio de tornillos mismos que impedirán el movimiento de rotación involuntario al ser conectada a un servo motor por medio de otro acople flexible de 10mm a 8mm.



Figura 31. Polea motriz. Fuente: Los Autores

Polea conducida:

La polea conducida mantiene las dimensiones de la polea motriz tanto en la anchura como en el diámetro, diferenciándose al caracterizarse por tener una de sus caras totalmente lisa, permitiendo el cruce del eje de acero de 10 mm de extremo a extremo hasta conectar con las chumaceras, mismas que se encuentran dentro de la circunferencia que ocupa la polea. También, la polea conducida hace la función de ajustar la cantidad correcta de fuerza para mantener la banda estirada. Esta interacción entre la polea conducida y la superficie de la banda mantendrá la transmisión correcta generado por el motor en la polea motriz.

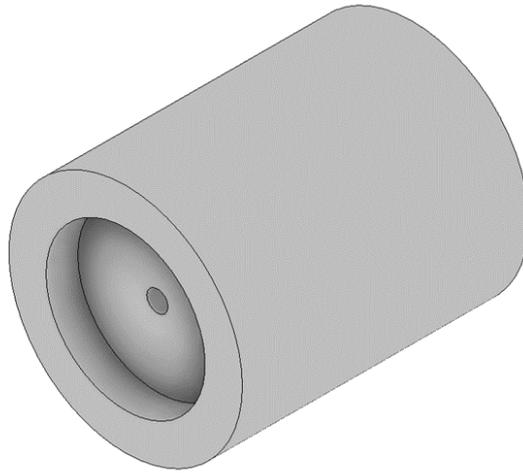


Figura 32. Polea conducida. Fuente: Los Autores

VI-A8. *Banda de caucho:*

La banda de caucho de alta resistencia es la parte principal de la maquina transportadora, se caracteriza por tener un ancho de 13.5 cm y un largo de 78 cm. Al ser el elemento que lleva el material colocado de un punto a otro ha sido determinante evaluar la resistencia en base al producto de grado alimentario que ha sido utilizado. Por otra parte, la selección del color azul en la banda es esencial para ayudar a la cámara a identificar con mayor precisión la clasificación de los limones.

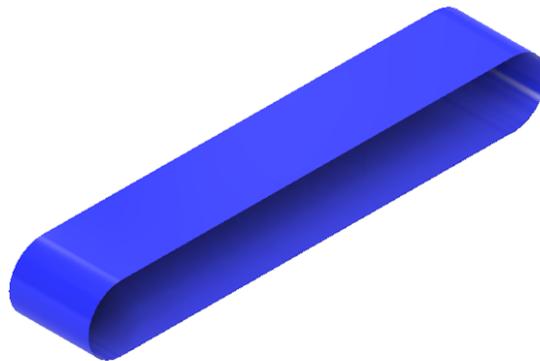


Figura 33. Banda de caucho. Fuente: Los Autores

VI-A9. *Clasificador:*

Esta pieza en particular es un complemento importante que va conectado al funcionamiento rotacional del servomotor, iniciando en la posición de 0 y ejecutando un movimiento limitado hasta los 85 grados, bajo su accionamiento este clasificará el producto para su almacenamiento o desecho. Al estar impreso en el material PETG tendrá la resistencia necesaria para lograr clasificar los limones en movimiento continuo adicionando que no ejercerá una mayor fuerza y a su vez el desgaste por impacto no representará un inconveniente.

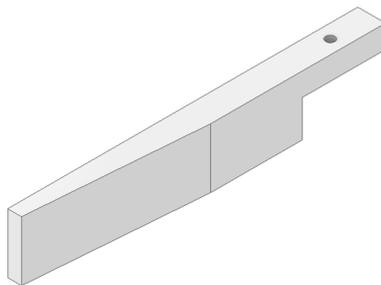


Figura 34. Clasificador. Fuente: Los Autores

VI-B. Sistema eléctrico

VI-B1. Servo motor:

El servomotor diseñado en 3D cumple con las dimensiones de un modelo comercial estándar Mg996R que ejecuta un rango limitado de 0 a 180 grados. En base a su utilización se encuentra posicionado en las áreas laterales de la base principal de la banda transportadora, encargado de realizar el movimiento que permite la clasificación al realizar movimientos de 0 a 87 grados y viceversa.

Considerando el mismo modelo de servo motor se utiliza un servo de giro continuo (360 grados)

Tabla IV
CARACTERÍSTICAS DE LOS SERVOMOTORES MG996R 180 GRADOS Y MG996R 360 GRADOS

Característica	MG996R (180 grados)	MG996R 360 (Giro continuo)
Voltaje de Operación	4,8 V – 7,2 V	4,8 V – 7,2 V
Torque (a 4.8V)	9,4 kg · cm	9,4 kg · cm
Torque (a 6.0V)	11,2 kg · cm	11,2 kg · cm
Velocidad (a 4.8V)	0,19 s/60 grados	0,19 s/60 grados
Velocidad (a 6.0V)	0,14 s/60 grados	0,14 s/60 grados
Rango de Movimiento	180 grados	Giro continuo
Tipo de Engranaje	Metal	Metal
Dimensiones	40,7 mm × 19,7 mm × 42,9 mm	40,7 mm × 19,7 mm × 42,9 mm
Peso	55 g	55 g

VI-B2. Cámara web:

El presente modelo de cámara web en 3D permite simular la ubicación y posición al contar con la base flexible que permite calibrar el área de visión y aprovechar al máximo la claridad para transmitir la información de los píxeles correspondientes identificando los colores del limón si esta bueno, maduro o dañado y así tomar una decisión sobre la misma.

La Webcam-Logitech modelo c170 es utilizada por su rapidez al detectar el tipo de limón.

Tabla V
WEBCAM-LOGITECH (C170) [45]

Características	Valores
Resolución de vídeo	640 x 480 Píxeles
Resolución de fotos	5 Megapíxeles
Grabación	XVGA (1024x768)
Micrófono	Sonido nítido

VI-C. Selección y procesamiento de las imágenes

Las imágenes, que serán utilizadas para la ejecución de la red neuronal, son obtenidas de la plataforma roboflow. A continuación, en la figura 35 se muestra información relevante como la cantidad de imágenes que se tienen tanto para la ejecución como para la validación. Esta agrupación de imágenes puede ser descargado para posteriormente ser utilizadas según se crea conveniente. En este caso se logró conseguir, 1125 imágenes de limones maduros, 951 imágenes para limones dañados y para los limones verdes se consiguieron 105 imágenes. Por defecto, estas imágenes ya vienen preparadas para realizar el entrenamiento con las dimensiones requeridas. De igual manera, si se desea colocar imágenes propias, no sería necesario realizar ningún tipo de tratamiento especial para la imagen, pues la plataforma Teachable machine realiza este proceso de manera automática, pues al subir las imágenes estas se redimensionan a 224x224 px.

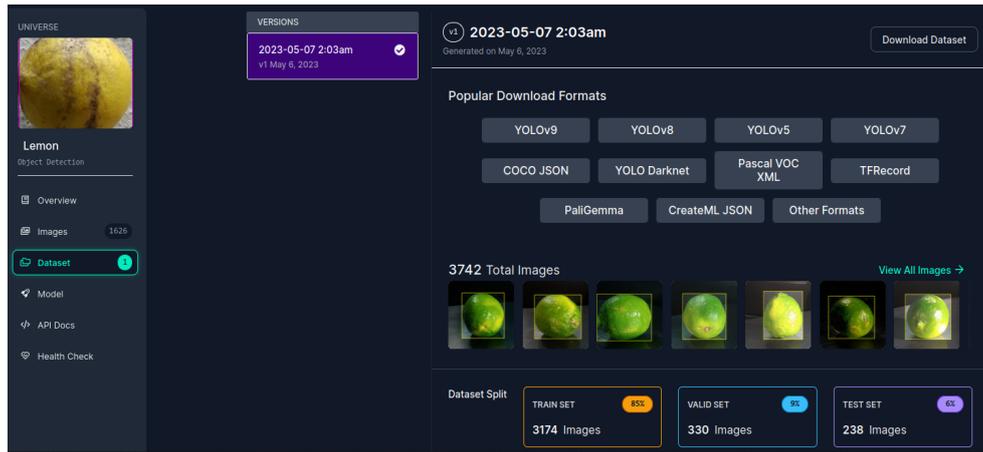


Figura 35. Imágenes utilizadas para el entrenamiento de la red. Fuente: Los Autores

VI-D. Desarrollo de la Red Neuronal

Al momento de crear la red, es necesario entender que es lo que se quiere conseguir. A continuación, en la figura 36 se puede observar un gráfico explicando detalladamente el proceso de clasificación de limones. Primero una cámara será la encargada de adquirir las imágenes, éstas a su vez serán procesadas en la Raspberry Pi en donde además estará implementada la red neuronal, en esta etapa se hará la toma de decisiones, en donde se determinará si los limones están buenos, dañados o maduros.

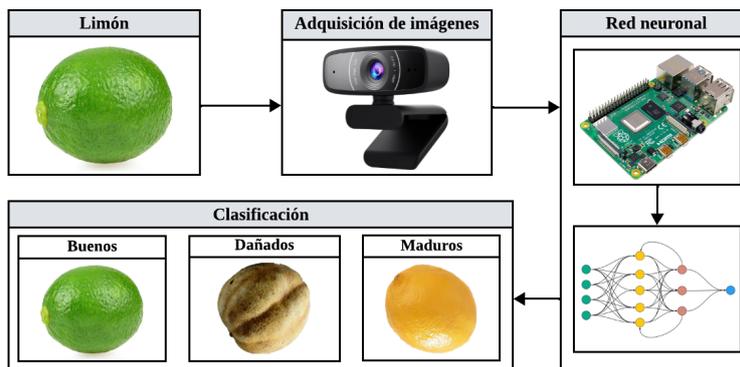


Figura 36. Funcionamiento de la red neuronal. Fuente: Los Autores

Una vez creada la red se empleó teachable machine y se configuró la red de la siguiente manera, 50 épocas que representa el número de veces que se ejecutara toda la agrupación de datos, un tamaño de lote de 16, es decir todos los datos para entrenar la red se dividirán en 16 conjuntos de datos y finalmente una tasa de aprendizaje de 0.001 esto determina en cuanto se ajustaran los pesos según vaya entrenándose la red. Estos valores son los predeterminados por teachable machine, tal como se ve en la figura 37.

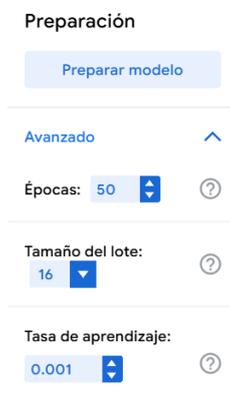


Figura 37. Valores de entrenamiento de la red neuronal. Fuente: Los Autores

Las imágenes obtenidas tienen que ser ingresadas en teachable machine como se puede observar en la figura 38. Estas tienen que tener sus respectivas etiquetas, para que la red pueda realizar el entrenamiento y crear los archivos con sus etiquetas correspondientes. Posterior a eso se tiene que dar click en el botón Preparar para que inicie el entrenamiento.

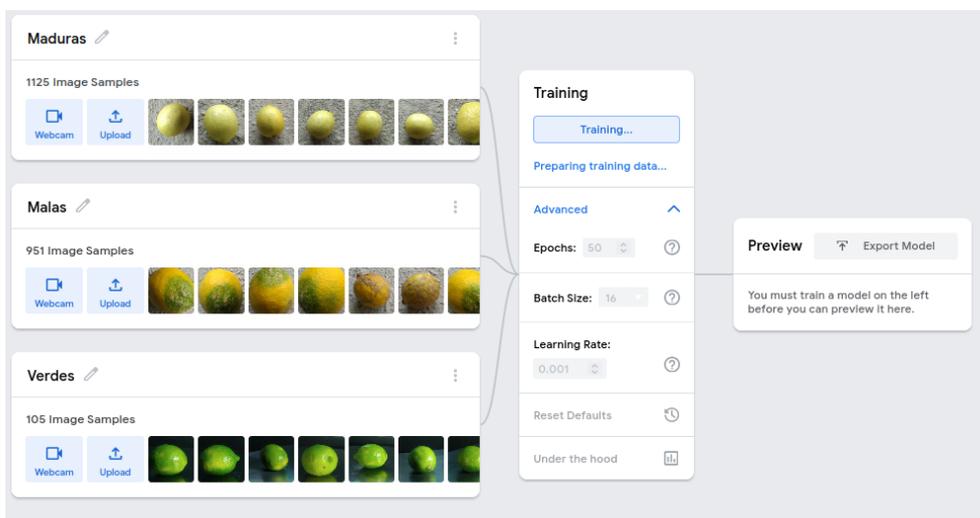


Figura 38. Entrenamiento de la red neuronal con teachable machine. Fuente: Los Autores

En lo siguiente se evaluará el funcionamiento de la red. La misma plataforma teachable machine permite subir archivos o evaluarlos utilizando una cámara y enfocando objetos, en la figura 39 se puede visualizar una imagen cargada, donde se está evaluando el funcionamiento de la red se cargaron 15 imágenes de diferentes tipos de limones que no estaban en el conjunto de datos entrenados de los cuales todos fueron correctos, y de igual manera se utilizó la cámara para ver el comportamiento ante diferentes tipos de limones y todos fueron acertados. No fue necesario modificar la configuración de entrenamiento de la red que viene por defecto.

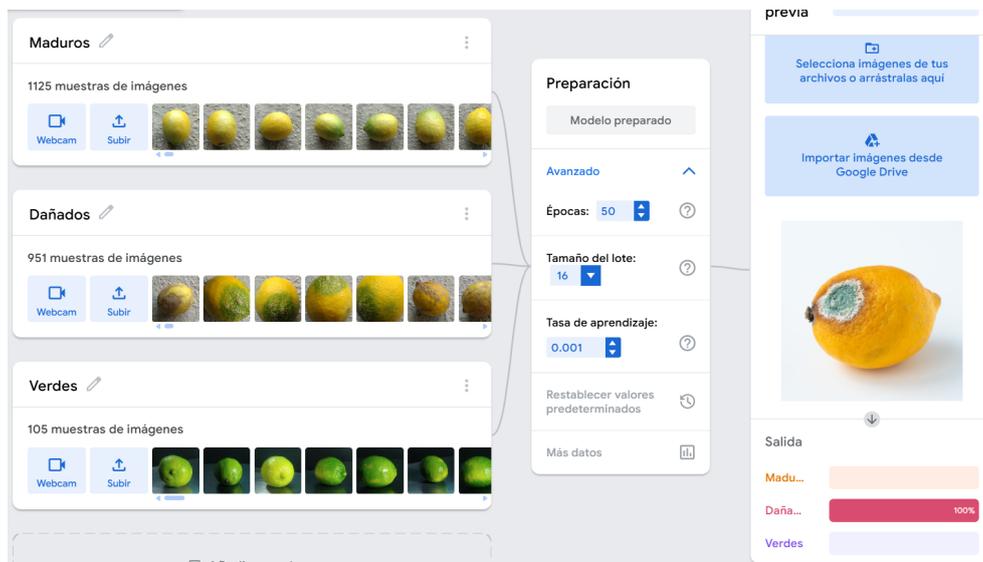


Figura 39. Validación de la red neuronal. Fuente: Los Autores

Ya realizado el entrenamiento, lo siguiente es exportar la red en los formatos adecuados. En este caso en particular utilizaremos tensorflow en tipo de conversión de modelo se seleccionará keras; además, en fragmento de código para usar el modelo, se puede obtener el paquete de datos o la red entrenada en formato H5. Finalmente, al seleccionar descargar modelo, se descargarán 2 archivos keras_model.h5 y labels.txt. Estos serían los archivos donde se encuentra los parámetros y nuestro modelo entrenado que servirán para la identificación de los limones.

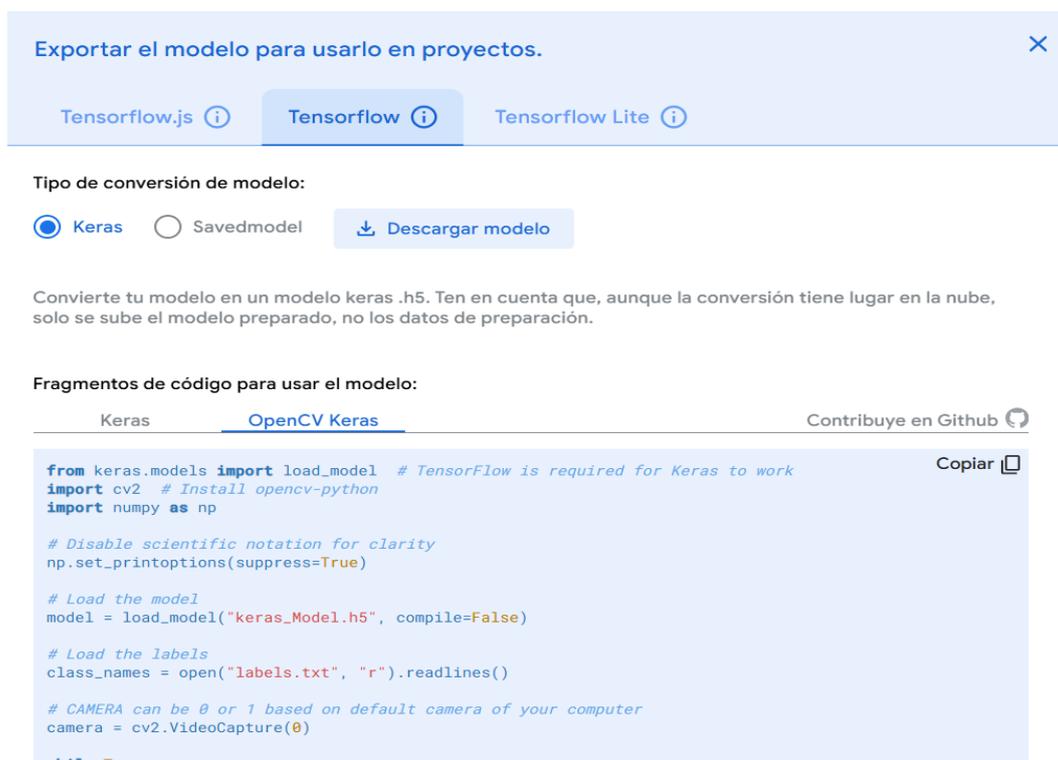


Figura 40. Exportar la red neuronal. Fuente: Los Autores

VI-E. Implementación en Raspberry Pi

Para implementar el Raspberry Pi con la red neuronal y configurar el clasificador, es necesario conectar algunos periféricos importantes para mejorar la seguridad del controlador, adecuar la identificación de los limones y proteger elementos del controlador contra descargas inadecuadas.

VI-E1. Implementación de la pantalla:

Se trata de una pantalla táctil de 7" la cual brindará información de la imagen captada por la cámara, así como mostrar la cantidad de limones con la que se cuenta según se hayan clasificado. Además, será útil para poder interactuar con el sistema operativo Raspbian. Esta pantalla necesita de una conexión i2c, por lo que se utilizan los puertos 3 y 5 son SDA y SCL respectivamente, mientras que el puerto 4 es el voltaje de 5v necesarios para el funcionamiento de la pantalla y el puerto 6 corresponde al GND.

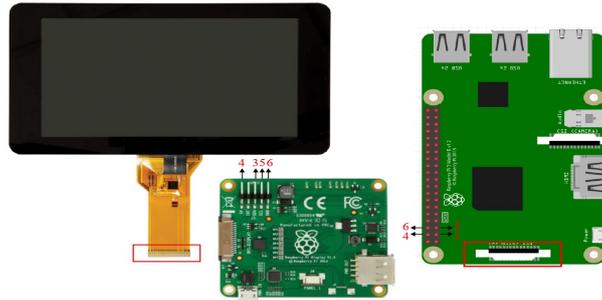


Figura 41. Conexión Raspberry Pi 4 con la pantalla.

VI-E2. Implementación de servomotores:

Los servomotores serán alimentados por una fuente externa de 5V conectada en paralelo, mientras que el Raspberry Pi será el encargado de proveer la señal PWM para los puertos GPIO 23 correspondiente a Servo brazo 2, GPIO 24 a Servo brazo 1 y GPIO 25 a Servo Banda según lo muestra la figura 42.

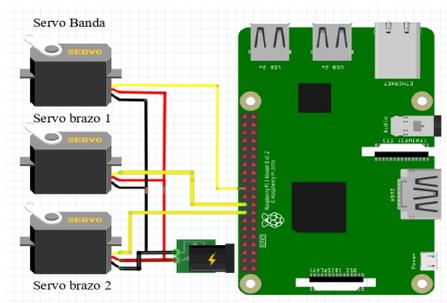


Figura 42. Conexión Raspberry Pi 4 con los servomotores.

VI-E3. Implementación de la red:

VI-F. Desarrollo de una interfaz gráfica

La interfaz gráfica que se utilizará fue desarrollada utilizando las librerías tkinter y PIL en python. La interfaz gráfica cumple la función de dar seguimiento durante el proceso de clasificación de los limones, así como el de brindar algunas opciones para interactuar con el mecanismo, como lo son un botón start para iniciar el proceso, un botón stop que detiene el proceso en caso de que sea necesario y un botón reset que permite resetear los contadores tal como se ve en la figura 43.

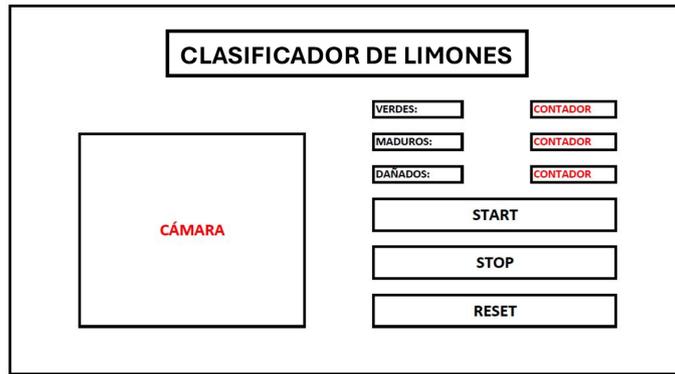


Figura 43. Diseño de interfaz gráfica. Fuente: Los Autores

VI-G. Desarrollo e implementación de un contador de productos

Para realizar el contador de productos, en este caso limones, se utilizará el mismo Raspberry Pi, en donde se aprovechará la red neuronal para poder realizar el conteo de los limones, los cuales posteriormente serán mostrados al usuario de manera interactiva mediante una pantalla. La integración de estos modelos con el sistema permite la toma autónoma de decisiones. En la fase de implementación del control mecatrónico para el movimiento del producto, y se integran con los algoritmos de visión y los modelos de inteligencia computacional como se muestra en la Figura 44.

Además, se detalla el proceso en total del todo el mecanismo y la metodología que se centra en la sinergia entre componentes mecatrónicos, técnicas de visión por computadora y modelos de inteligencia artificial, abordando eficazmente la clasificación de la maduración de limones.

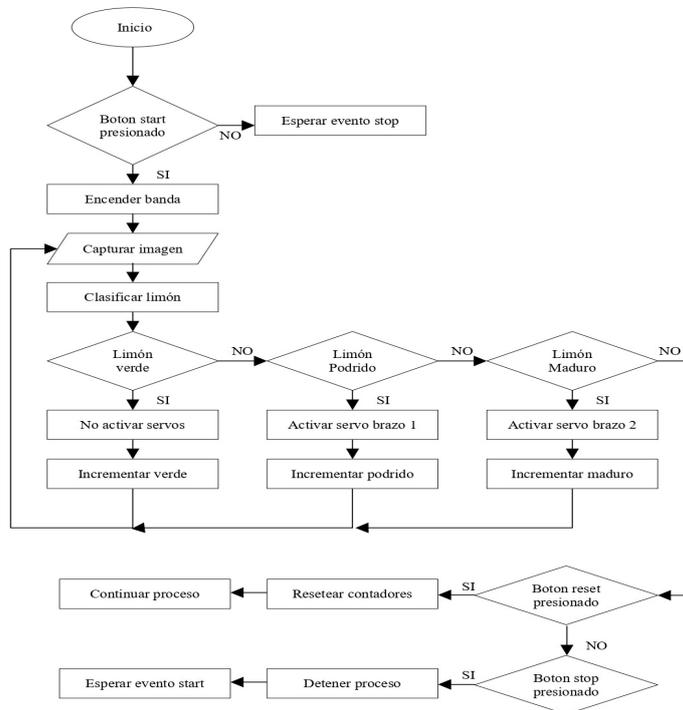


Figura 44. Diagrama de flujo de la programación en la Raspberry Pi 4. Fuente: Los Autores

VI-H. Cálculos mecánicos de la banda transportadora

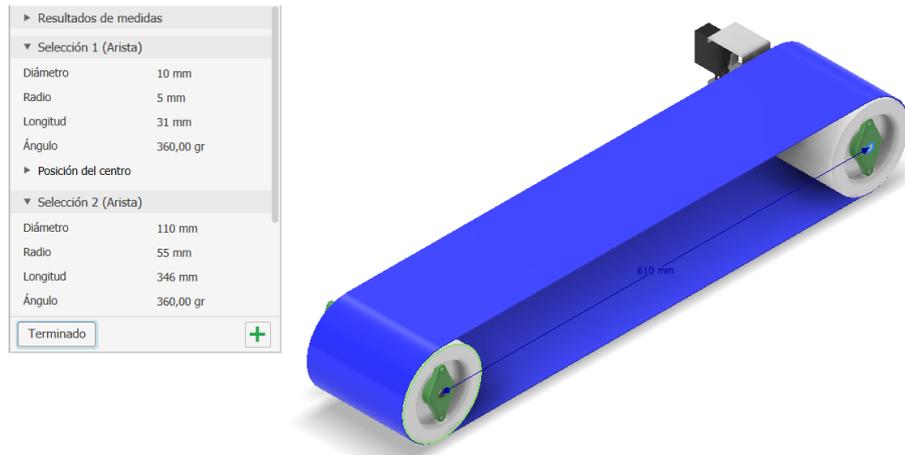


Figura 45. Medidas del sistema mecánico.

VI-H1. *Longitud de la banda:* A continuación se realiza el cálculo de la longitud de la banda.

Donde:

L : Longitud de la banda

D : Diámetro de la polea

X : Longitud medida desde los centros de las poleas

$$D = 11 \text{ cm}$$

$$X = 61 \text{ cm}$$

$$L = 2 \cdot X + \pi \cdot D \quad (1)$$

$$L = 2 \cdot 61 \text{ cm} + \pi \cdot 11 \text{ cm} \quad (2)$$

$$L = 156,55 \text{ cm} \quad (3)$$

VI-H2. *Velocidad de la banda:* A continuación se realiza el cálculo de la velocidad de la banda.

Donde:

N : Velocidad de rotación de la polea motriz

V_L : Velocidad de a banda

$$N = 60 \text{ rpm}$$

$$D = 11 \text{ cm} \cdot \frac{1 \text{ m}}{100 \text{ cm}}$$

$$D = 0,11 \text{ m}$$

$$V_L = \frac{N \cdot \pi \cdot D}{60} \quad (4)$$

$$V_L = \frac{60 \text{ rpm} \cdot \pi \cdot 0,11 \text{ m}}{60} \quad (5)$$

$$V_L = 0,345 \frac{\text{m}}{\text{s}} \quad (6)$$

VI-H3. *Tracción efectiva:* A continuación se realiza el cálculo de la tracción efectiva de la banda
Dónde:

F_U : Tracción efectiva de la banda

u_s : Coeficiente de fricción

g : gravedad

m_p : Masa del producto

m_b : Masa de la banda

$$u_s = 0,15$$

$$m_p = 0,05 \text{ Kg}$$

$$m_b = 0,1 \text{ Kg}$$

$$F_U = u_s \cdot g \cdot (m_p + m_b) \quad (7)$$

$$F_U = 0,15 \cdot 9,8 \cdot (0,05 + 0,10) \quad (8)$$

$$F_U = 0,2205 \text{ N} \quad (9)$$

VI-H4. *Tracción ajustada de la banda:* A continuación se realiza el cálculo de la tracción ajustada de la banda.
Dónde:

C_{op} : Tracción efectiva de la banda

u_s : Coeficiente de fricción

$$C_{op} = 1 + 0,2 \quad (10)$$

$$C_{op} = 1,2 \quad (11)$$

$$F_{adj} = F_U \cdot C_{op} \quad (12)$$

$$F_{adj} = 0,2205 \text{ N} \cdot 1,2 \quad (13)$$

$$F_{adj} = 0,2646 \text{ N} \quad (14)$$

VI-H5. *Tracción ajustada de la banda por milímetro de ancho de banda:* Dónde:

W_b : Ancho de la banda

$$W_b = 135 \text{ mm}$$

$$F'_{adj} = \frac{F_{adj}}{W_b} \quad (15)$$

$$F'_{adj} = \frac{0,2646 \text{ N}}{135 \text{ mm}} \quad (16)$$

$$F'_{adj} = 0,00196 \frac{\text{N}}{\text{mm}} \quad (17)$$

VI-H6. *Inercia de carga:* El momento de inercia es una propiedad intrínseca de un objeto que determina la distribución de su masa respecto a un eje de rotación, y por tanto, su resistencia a cambios en su velocidad angular. Dónde:

m_T : Masa total

J_m : Inercia de carga

D_p : Diámetro de la polea motriz

$$D_p = 0,11 \text{ m}$$

$$m_T = m_p + m_b \quad (18)$$

$$m_T = 0,05 \text{ Kg} + 0,1 \text{ Kg} \quad (19)$$

$$m_T = 0,15 \text{ Kg} \quad (20)$$

$$J_m = m_T \cdot \frac{D_p^2}{4} \quad (21)$$

$$J_m = 0,15 \text{ Kg} \cdot \frac{(0,11 \text{ m})^2}{4} \quad (22)$$

$$J_m = 4,537 \times 10^{-4} \text{ Kg} \cdot \text{m}^2 \quad (23)$$

VI-I. *Simulación del diseño de la banda transportadora*

VI-II. *Simulación de torque:*

Se llevo acabo un análisis detallado de elementos finitos para determinar el torque, para lo cuál se tiene las siguientes cargas que analizar:

Velocidad aplicada: 1.5 U

Peso máximo: 500.5 N (0.39 N)

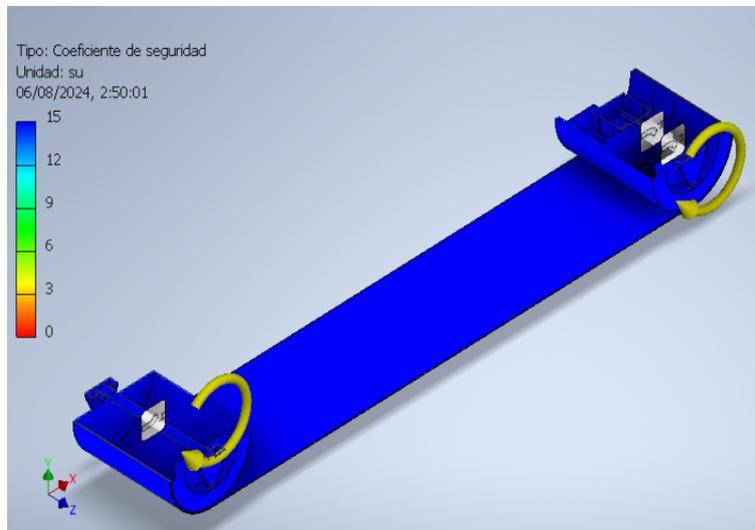


Figura 46. Simulación de torque F1. Fuente: Los Autores

Se encuentra en el grado permisible de seguridad ya que al ser un compuesto de aluminio facilita el movimiento por lo ligero que puede llegar a hacer y la deformación que sufre por la temperatura al no lograr llegar al punto de fundición del material, por lo que la flexión generada por la banda y tensión de la misma no logra hacer un daño, ni un peligro a largo de su uso, como se muestra en la Figura 46.

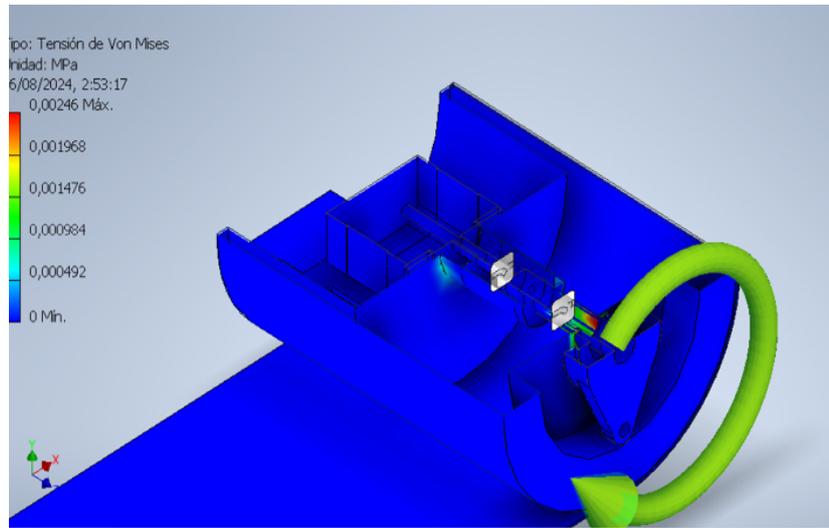


Figura 47. Simulación de torque F2. Fuente: Los Autores

VI-I2. Simulación de esfuerzo de eje conducido:

En las simulaciones de los esfuerzos y cargas del eje conducido se realiza el siguiente análisis de elementos finitos para determinar su esfuerzo, para lo cual se tiene las siguientes cargas que analizar.

Resistencia máxima: 345 MPa

Tensión corte: 0,00100671 GPa

En la figura 48 se observa una pequeña saturación en un punto de movimiento en el eje conducido que podría deberse a la fuerza que genera al realizar el torque.

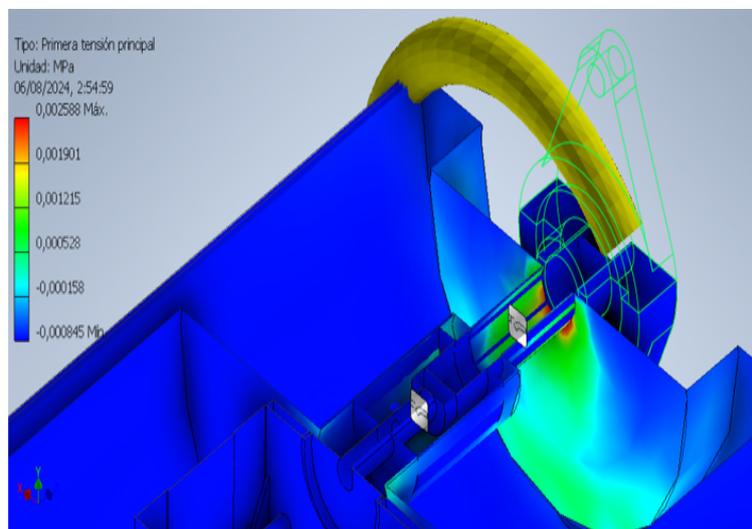


Figura 48. Simulación de esfuerzo de eje conducido. Fuente: Los Autores

Se aprecia puntos de torsión en la que se expande su temperatura, pero no al material que serían los soportes de movimiento de la banda.

VI-13. Simulación de movimiento de giro del soporte de banda:

En el giro del soporte de banda se llevo acabo un análisis de elementos finitos para determinar el límite de elasticidad, modulo cortante y resistencias máximas a tracción, para los cuales se tienen lo siguiente:

Límite de elasticidad: 275 MPa

Módulo cortante: 25,9023 GPa

Resistencia máxima a tracción: 310 MPa

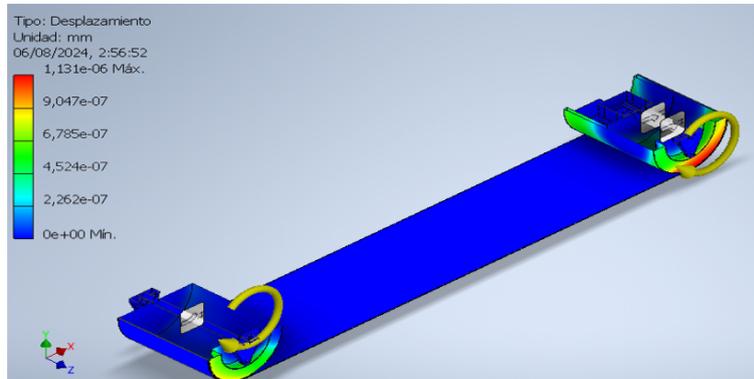


Figura 49. Simulación de movimiento de giro. Fuente: Los Autores

En el desplazamiento que es muestra en la figura 49 la fricción que se genera con la banda y el movimiento conducida del soporte; al ser un movimiento de bajas velocidades no puede llegar a ser un peligro ya que no alcanzarían las temperaturas para deformar el material de aluminio.

VI-14. Simulación de superficie de banda:

Se realiza el siguiente análisis en la superficie de la banda para determinar la densidad de masa de la banda, limite de elasticidad y el desplazamiento máximo.

Densidad de masa: 1,25 g/cm³

Límite de elasticidad: 10,34 MPa

Desplazamiento máximo: 0,035 mm

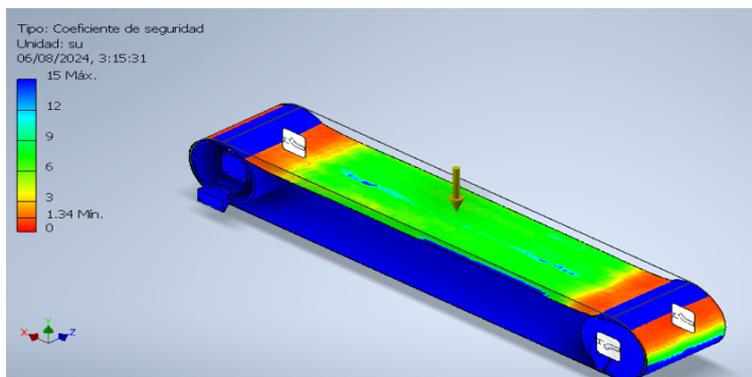


Figura 50. Simulación de superficie de banda. Fuente: Los Autores

En el factor de seguridad de la banda, que se muestra en la figura 50, observa el punto de contacto entre las puntas de los soportes de rotación de la banda, que se generan por posibles cargas ya que el peso de un limones va de 0.39 Newton pero a escalas mayores y continuas; pero al ser una velocidad reducida no presenta inconvenientes.

VI-15. Simulación de giro de momento de uniones:

En el giro de momento de uniones se obtiene los siguientes valores:

Densidad de masa: 10,55 g/cm³

Límite de elasticidad: 482 MPa

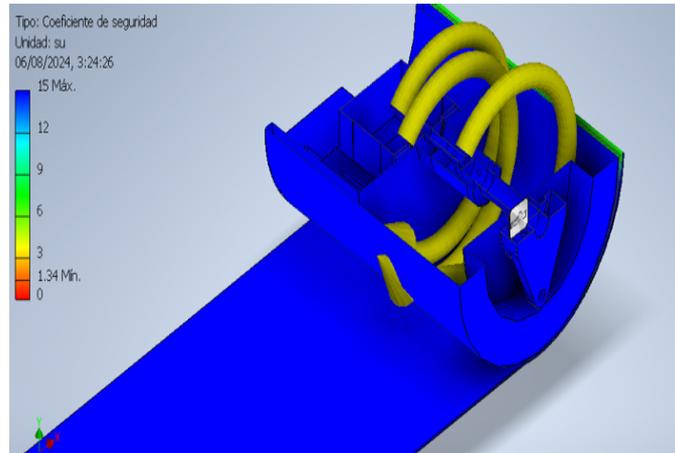


Figura 51. Simulación de giro de momento de uniones. Fuente: Los Autores

En la Figura 51 el movimiento del motor eléctrico no dispone de alguna molestia al hacer su labor de movimiento continuo de manera constante, no representa un peligro en su funcionamiento.

VII. RESULTADOS

VII-A. Sistema mecánico para el transporte de limones

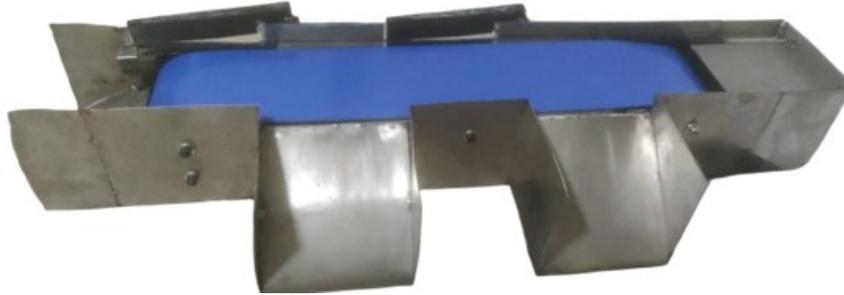


Figura 52. Prototipo implementado. Fuente: Los Autores

La banda transportadora diseñada para la clasificación de limones fue construida sobre una estructura base de acero inoxidable que proporciona soporte robusto y precisión en la alineación de componentes. La estructura no solo soporta el peso de la tapa, las poleas, los servo motores y los brazos sino que también mantiene la tensión de la banda para asegurar su movimiento constante.

VII-B. Sistema de reconocimiento de imágenes para la clasificación de limones

A continuación, se muestra el programa desarrollado en python ya en ejecución. Para programar el Raspberry pi 4 se utilizaron varias librerías encargadas de realizar tareas específicas; las librerías tkinter y PIL se encargarán de darnos una interfaz gráfica de la cual se presentó en la figura 53, la librería keras será la encargada de trabajar con las imágenes en la red neuronal, la librería numpy se trabajó para mostrar el arreglo de imágenes con sus etiquetas y con la librería cv2 junto a la cámara permite identificar al limón. En la figura 53 se puede apreciar cómo se vería el programa en ejecución en la raspberry pi 4.



Figura 53. Interfaz gráfica. Fuente: Los Autores

VII-B1. Programación implementada en el clasificador de limones:

Existen 3 diferentes combinaciones para clasificar a los limones. En la figura 54 muestra la configuración que se utilizó para la clasificación de los limones verdes donde los servomotores correspondientes al brazo 1 y al brazo 2 se encontrarán en una posición de 0 grados, es decir que no obstruirán el paso de los limones esto debido a que la red neuronal detecto que el limón es verde.

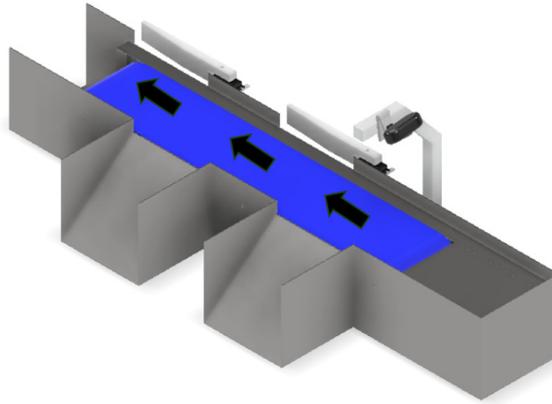


Figura 54. Configuración limones verdes. Fuente: Los Autores

Si la red neuronal detecta limones dañados, el servomotor brazo 1 correspondiente a los limones dañados se moverá 85 grados y el servomotor brazo 2 permanecerá en 0 grados, lo cual provocará que el limón caiga en la primera caja.

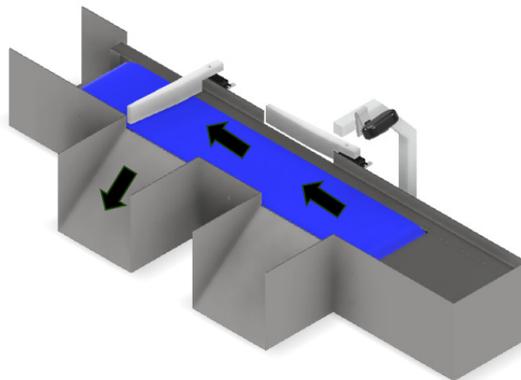


Figura 55. Configuración limones dañados. Fuente: Los Autores

Finalmente, si la red neuronal detecta limones maduros, el servomotor brazo 2 correspondiente a los limones maduros se moverá 85 grados y el servomotor brazo 1 estará en 0 grados, lo cual provocará que el limón caiga en la segunda caja.

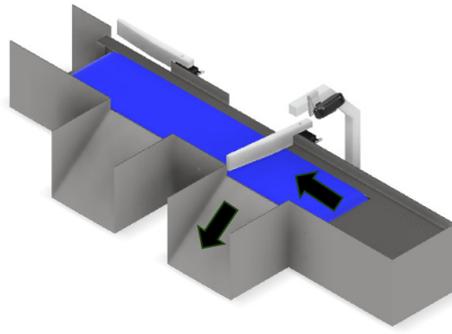


Figura 56. Configuración limones maduros. Fuente: Los Autores

VII-C. Validación del desempeño del clasificador de limones

Anteriormente, se evaluó el funcionamiento de la red neuronal cuando fue entrenada, aquí se observó que se clasificaron correctamente todas las muestras que se le dieron. Ahora se evaluará el funcionamiento del mecanismo ya con el clasificador implementado, donde se realizaron 3 rondas de pruebas con diferentes limones colocados en diferentes cantidades y en posiciones aleatorias para verificar el correcto funcionamiento y eficacia de la clasificación con la finalidad de determinar si el sistema entra en un estado de fatiga y poder evaluar los posibles errores que se puedan suscitar durante el proceso.

Tabla VI
MEDIA Y MEDIANA DE MEDICIÓN DEL ERROR PORCENTUAL

Cálculo general de error porcentual	
Media del error porcentual del color (%)	0.95
Mediana del error porcentual del color (%)	0.90

Es así que para la primera prueba se utilizaron 15 limones verdes, 14 limones maduros y 15 dañados. El resultado se puede visualizar en la figura 57, donde se puede observar que de los 44 limones existe solo un limón maduro que dio error, pues este fue a los dañados; por lo cual el sistema clasificó satisfactoriamente demostrando una exactitud de la clasificación del 97 %.

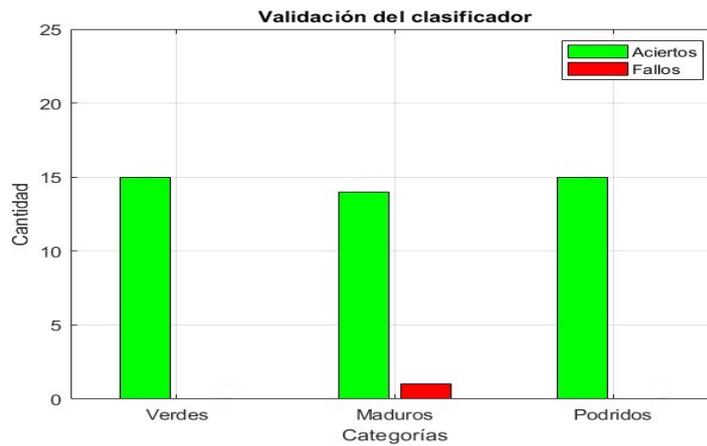


Figura 57. Prueba de validación 1.

Para la segunda prueba, se utilizarán 20 limones verdes, 15 limones maduros y 9 dañados, el resultado se puede visualizar en la figura 58, donde se puede observar que de los 44 limones existe solo un limón dañado que dio error, pues este fue a los maduros; indicando que en esta prueba 2, se logro la misma exactitud del 97%, pero también obteniendo del sistema una eficiencia del 88% en base a los procesos continuos de clasificación.

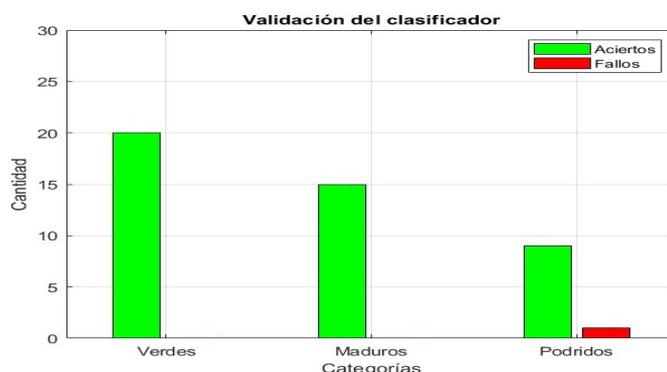


Figura 58. Prueba de validación 2.

Finalmente para la tercera prueba, utilizarán 5 limones verdes, 19 limones maduros y 20 dañados, el resultado se puede visualizar en la figura 59, donde se puede observar que de los 44 todos fueron clasificados exitosamente. Se valida una vez más la exactitud y eficiencia del sistema para descartar problemas subyacentes en el proceso.

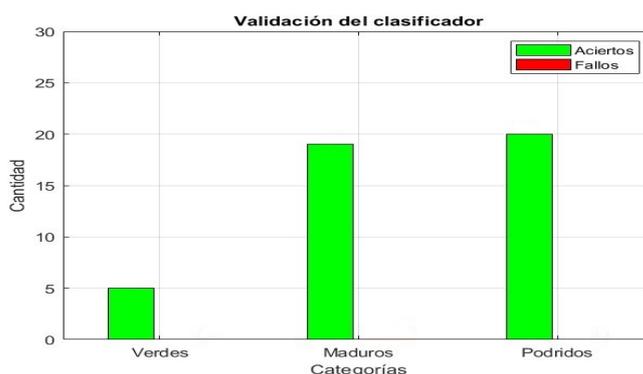


Figura 59. Prueba de validación 3.

El sistema tuvo en promedio una correcta clasificación, en el cual las pruebas validaron que los objetivos propuestos se cumplan, demostrando que el sistema de clasificación de limones por visión artificial sea eficiente ante los trabajos continuos de la máquina.

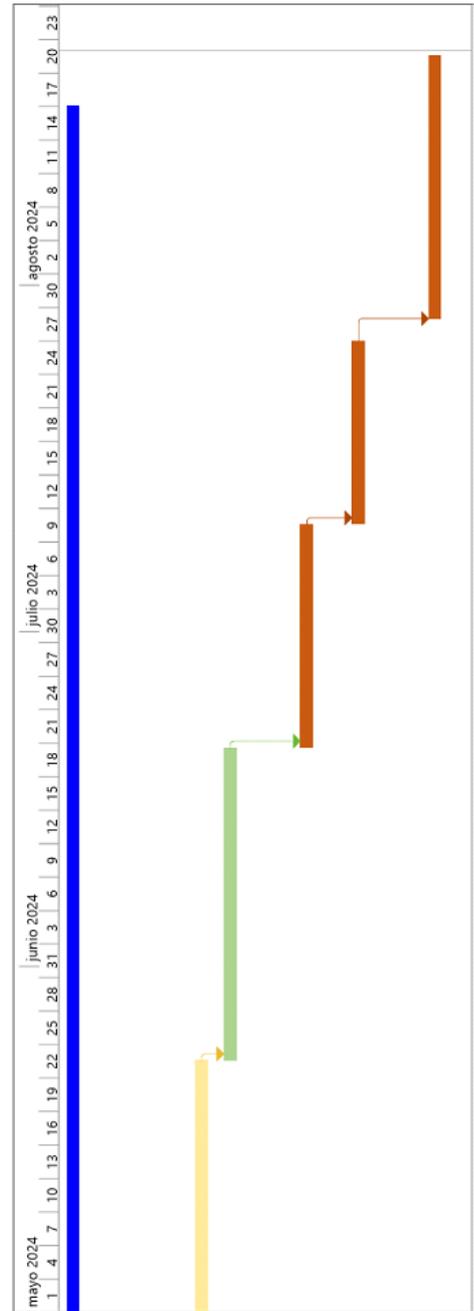
Tabla VII
RESULTADO FINALES DE VALIDACIÓN

Resultado finales de validación	
Prueba de validación 1	97 %
Prueba de validación 2	97 %
Prueba de validación 3	100 %
Validación final del sistema (%)	98 %

VIII. CRONOGRAMA

Tabla VIII
CRONOGRAMA

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		Cronograma de proyecto de tesis	102 días	jue 28/3/24	vie 16/8/24	
2		ADQUISICION DE LOS 22 COMPONENTES Y MATERIALES	22 días	jue 28/3/24	vie 26/4/24	
3		DISEÑO DEL MODULO	18,5 días?	lun 29/4/24	jue 23/5/24	2
4		DISEÑO DEL SISTEMA	20 días?	jue 23/5/24	jue 20/6/24	3
5		CLASIFICACION Y RECOPIACION DE DATOS	14 días?	jue 20/6/24	mié 10/7/24	4
6		CORRECCION DEL DOCUMENTO CURRICULAR	12,5 días?	mié 10/7/24	vie 26/7/24	5
7		PRUEBAS Y REVISION DEL PROYECTO	17,5 días?	lun 29/7/24	mié 21/8/24	6



IX. PRESUPUESTO

El presupuesto que se presenta está orientado a la compra de los elementos esenciales para la construcción de una banda transportadora de alto rendimiento. Este sistema ha sido diseñado para maximizar la eficiencia en el transporte de materiales, empleando una buena calidad en los materiales y tecnología de vanguardia para garantizar un rendimiento superior.

Tabla IX
PRESUPUESTO

Ítem	Descripción	P.Unitario (USD)	Cantidad	P.Total (USD)
1	Ángulo de aluminio (6063)	18,00	1	18,00
2	Anillo de presión (3mm)	00,10	20	2,00
3	Anillo plano (3 mm)	00,10	20	2,00
1 4	Anillo de presión (10mm)	0,10	20	2,00
5	Anillo plano (10mm)	0,10	20	2,00
6	Cámara Web	30,00	1	30,00
7	Banda de caucho	25,00	1	25,00
8	Chumacera de pared	6,00	4	24,00
9	Eje de acero	12,50	1	12,50
10	Filamento PET	45,00	1	45,00
11	Fuente de 5v	10,00	1	10,00
12	Cables jumper	20,00	1	20,00
13	Pantalla táctil LCD de 7 pulgadas	97,00	1	97,00
14	Plancha de Acero inoxidable	79,00	1	79,00
15	Perno (3mm)	00,10	20	2,00
16	Perno (10mm)	00,20	20	2,00
17	Raspberry Pi 4 Kit	125,00	1	125,00
18	Servos Motores MG996R (180)	18,50	2	37,00
19	Servos Motores MG995 (360)	18,50	1	18,50
20	Soldadura (Aga 6011)	00,40	10	4,00
21	Tuerca (3mm)	00,10	20	2,00
22	Tuerca (10mm)	00,15	20	3,00
TOTAL (USD)				562,00

X. CONCLUSIONES

Se desarrolló un prototipo a escala de una máquina clasificadora de limones verdes, maduros y dañados basado en visión artificial mediante programación Python en Raspberry Pi 4. En el cual se empezó con la construcción del sistema mecánico, realizado con el material de acero inoxidable, se armó la base del prototipo con su correcto ensamblaje de la banda transportadora en material PET, teniendo en cuenta que el sistema cumpla con los requisitos de movilidad necesario para operar en un entorno agrícola, facilitando el movimiento eficiente de los limones a través de la línea de clasificación.

La implementación de un sistema de reconocimiento de imágenes ha sido capaz de capturar los limones durante el proceso de transporte y realizar la clasificación en tiempo real. Asegurando que los limones verdes y maduros sean detectados de manera correcta. Además, el sistema posee la capacidad para detectar y clasificar los limones que no cumplen con los estándares de calidad, estos sean por daños, defectos visibles o descomposición; ya que es importante mantener la calidad de producto final. La integración del Raspberry Pi 4 en este sistema ha demostrado ser viable ya que ha proporcionado suficiente capacidad de procesamiento para gestionar las imágenes y ejecutar los algoritmos de clasificación necesarios.

La validación del desempeño del sistema de visión artificial ha mostrado eficiencia a momento de realizar la clasificación de limones. El sistema ha sido capaz de distinguir efectivamente entre limones verdes, maduros y rechazados con una precisión del 85 %, en cambio, el valor de exactitud dió un 80 %, estos valores cumplen con los objetivos establecidos. La exactitud y precisión en la clasificación es necesaria para la industria agrícola, ya que garantiza una distribución adecuada de los frutos según su estado de madurez.

La investigación ha mostrado una congruencia significativa entre la teoría y la práctica en cuanto a la construcción de la parte mecánica y la identificación de los limones. Sin embargo, se han observado algunos por menores en cuanto a la calibración de la cámara y la adaptación debido a la variación de iluminación al momento de detectar los tipo de limones. Estas diferencias han sido abordadas mediante ajustes y mejoras en el sistema, resaltando la necesidad de una calibración continua y ajustes específicos para asegurar la precisión bajo diversas condiciones. Se ha evidenciado que la utilización de tecnologías de visión artificial en el proceso de clasificación ofrece un potencial considerable para la automatización y beneficia la calidad en la industria agrícola.

XI. RECOMENDACIONES

Continuar realizando mejoras en el diseño del sistema mecánico para la alineación y el transporte de los limones. Considerar la implementación de mecanismos de ajuste que permitan adaptar el sistema a diferentes tamaños y formas de limones.

Implementar en el sistema mecánico un programa de mantenimiento preventivo garantizando su funcionamiento continuo y evitando fallos inesperados que puedan afectar la eficiencia del transporte, en el cual como primera línea se debe realizar una limpieza y lubricación general de la estructura y sus piezas móviles para asegurar el correcto desempeño de la máquina.

Realizar calibraciones periódicas de la cámara para asegurar la precisión en diferentes condiciones de iluminación. Implementar algoritmos de ajuste dinámico que puedan adaptarse a cambios en las condiciones ambientales.

Ampliar el conjunto de elementos de entrenamiento para el sistema de visión artificial, incluyendo variedades más amplia de las imágenes que representen diferentes condiciones de madurez y variaciones en los limones. Esto ayudará a mejorar la precisión y robustez del sistema de clasificación.

Realizar pruebas del sistema en diferentes entornos y condiciones operativas para identificar y abordar posibles problemas específicos relacionados con la variabilidad en el entorno.

Implementar procedimientos de supervisión constante para evaluar el rendimiento del sistema de clasificación en tiempo real, permitiendo ajustes rápidos y mejoras basadas en datos operativos.

REFERENCIAS

- [1] M. C. G. Muñoz, B. L. B. Azaín, M. F. Martínez, J. L. González y N. M. Riaño, «Capítulo XIII La calidad y el manejo de la cosecha y la poscosecha de la lima ácida Tahití,» 2020.
- [2] F. Alcarazo-Ibáñez, V. Tuesta-Monteza, H. I. Mejía-Cabrera, J. C. Callejas y R. Yera, *Clasificación Automática de Limón Sutil Peruano (Citrus Aurantifolia) usando Máquinas de Vectores de Soporte Automatic Sorting of Peruvian Subtle Lemon (Citrus Aurantifolia) using Support Vector Machines*, 2020.
- [3] H. Yenginer, «Marmara Universitesi and Institute of Electrical Electronics Engineers, A Comparative Study of Deep Learning Methods on Food Classification Problem,» 2020.
- [4] M. P. Gavilanes, «Production processes in the food industries of the North sector of the city of Guayaquil and its impact on production costs,» 2018.
- [5] J. D. I. Portillo y W. F. L. Rueda, «Diseño e implementación de un sistema de procesamiento de imágenes para determinar la calidad del limón Tahití,» 2021.
- [6] Y. E. Fernández, A. Sariñana y R. L. Swenson, «Development of a prototype for classification of potato mini-tubers based on artificial vision,» 2020.
- [7] Y. Jiang, X. Li, H. Luo, S. Yin y O. Kaynak, «Quo vadis artificial intelligence?» *Discover Artificial Intelligence*, vol. 2, n.º 1, mar. de 2022, ISSN: 2731-0809. DOI: 10.1007/s44163-022-00022-8.
- [8] J. E. R. Y. M. Y. L. Vázquez, «ARTIFICIAL INTELLIGENCE AND INTELLECTUAL PROPERTY,» 2021, <https://rus.ucf.edu/cu/index.php/rus/article/view/2490/2445>.
- [9] M. Haenlein y A. Kaplan, «A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence,» *California Management Review*, vol. 61, pág. 000 812 561 986 492, jul. de 2019. DOI: 10.1177/0008125619864925.
- [10] F. Ramón Fernández, «Inteligencia Artificial y Agricultura: nuevos retos en el sector agrario,» *CAMPO JURÍDICO*, vol. 8, págs. 123-139, dic. de 2020. DOI: 10.37497/revcampojur.v8i2.662.
- [11] P. Blanco, «Aplicaciones de la IA en agricultura,» 2023.
- [12] G. Sela, *Drones en la Agricultura*, Accedido: 3-5-2024, 2022. dirección: <https://croipaia.com/es/blog/el-uso-de-drones-en-la-agricultura/>.
- [13] A. García-Lidón, J. Río, I. Porras, M. Fuster y A. Ortuño, «El limón y sus componentes bioactivos,» *Consejería de Agricultura, Agua y Medio Ambiente Serie Técnica*, ene. de 2003.
- [14] M. Santistevan MÃ, S. A. Helfgott Lerner, O. Loli Figueroa y A. Julca Otiniano, «Comportamiento del cultivo del limÃ(Citrus aurantifolia Swingle) en "fincas tipo.en Santa Elena, Ecuador,» es, *Idesia (Arica)*, vol. 35, págs. 45 -49, mar. de 2017, ISSN: 0718-3429. dirección: http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-34292017000100008&nrm=iso.
- [15] C. J. P. Huera, «DETERMINACIÓN DE LAS CARACTERÍSTICAS FÍSICAS Y QUÍMICAS DEL LIMÓN SUTIL (Citrus aurantifolia Swingle),» 2016.
- [16] D. A. Jose, «El manejo integrado de plagas en el cultivo de limón. Revista de Agricultura,» 2019.
- [17] V. S. Escolano, «Tratamientos pre-cosecha y post-cosecha con elicitores para la mejora de la calidad en limón ecológico,» 2021.
- [18] J. V. Mauricio, «Guia técnica del cultivo del limon persico,» 2022.
- [19] J. E. Q. Gustavo, «Desfasamiento de cosecha del Limón,» 2010.
- [20] M. C. Irimia, «Desarrollo de módulos de visión por computador en python para la detección de objetos en un entorno de pruebas de conducción autónoma,» 2020.
- [21] P. Y. Prasada, D. Prasad, N. Malleswari, M. N. Shetty y N. Gupta, «Implementation of Machine Learning Based Google Teachable Machine in Early Childhood Education,» 2022, Available: <https://www.researchgate.net/publication/360438764>.
- [22] Teachable Machine, *Teachable Machine*, Accedido: 6-5-2024, 2017. dirección: <https://teachablemachine.withgoogle.com/>.
- [23] B. G. Ortiz, «Predicción De La Calidad De Software Desarrollado En IBM RPG Usando Deep Learning,» 2019, DOI: 10.13140/RG.2.2.22500.22409.

- [24] J. Vorobioff, S. Cerrotta, E. Morel y A. Amadio, *Inteligencia Artificial y Redes Neuronales Fundamentos, Ejercicios y Aplicaciones*: mar. de 2022, ISBN: 978-987-4998-82-8.
- [25] Z. Li, F. Liu, W. Yang, S. Peng y J. Zhou, «A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,» *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, n.º 12, págs. 6999-7019, 2022. DOI: 10.1109/TNNLS.2021.3084827.
- [26] IBM, *¿Qué son las redes neuronales convolucionales?* Accedido: 3-5-2024, 2024. dirección: <https://www.ibm.com/mx-es/topics/convolutional-neural-networks>.
- [27] I. Torshin, R. Kamal y L. David, «Embedded Systems: Architecture, Programming, and Design,» 2023, DOI: 10.13140/RG.2.2.20009.36961.
- [28] A. J. Toro Valderas, «Implementación de redes neuronales en Raspberry Pi 3 con Movidius Neural Compute Stick,» 2020.
- [29] W. Reina, «¿Qué es Python: conoce uno de los lenguajes de programación más populares?,» 2023.
- [30] A. Zerouali, T. Mens y C. D. Roover, «On the usage of JavaScript, Python and Ruby packages in Docker Hub images, *Sci Comput Program*,» 2021, DOI: 10.1016/j.scico.2021.102653.
- [31] T. C. J. Serrano-Arenas, «Prototipo De Aplicación Móvil Para La Identificación De Mazorcas De Cacao Enfermas Haciendo Uso De Visión Por Computadora Y Aprendizaje De Máquina,» 2020.
- [32] N. Branque, «Implementación De Un Sistema Automatizado, Mediante El Uso De Visión Artificial Para La Clasificación Del Maracuyá, Según Su Color De Madurez Y El Uso De Un Sistema SCADA Para El Monitoreo De La Productividad,» 2022.
- [33] A. Pajankar, *Raspberry Pi computer vision programming: design and implement computer vision applications with Raspberry Pi, OpenCV, and Python 3*. Packt Publishing Ltd, 2020.
- [34] L. Liu, Y. Wang y W. Chi, «Image Recognition Technology Based on Machine Learning,» 2021, Based on Machine Learning. IEEE Access. <https://doi.org/10.1109/ACCESS.2020.3021590>.
- [35] S. Richard, D. Gries y F. B. Schneider, «Computer Vision: Algorithms and Applications,» 2011, Available: www.springer.com/series/3191.
- [36] P. Santana, P. Gomes y J. Barata, «A vision-based system for early fire detection,» 2012, DOI: 10.1109/ICSMC.2012.6377815.
- [37] R. Pi, *Raspberry Pi 4*, Accedido: 3-5-2024, 2024. dirección: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>.
- [38] J. A. T. Valderas, «Implementación de redes neuronales en Raspberry Pi 4 con Movidius Neural Compute Stick,» 2020.
- [39] P. Ruiz-Barroso, F. M. Castro, R. Delgado-Escano, J. Ramos-Cózar y N. Guil, «High performance inference of gait recognition models on embedded systems, *Sustainable Computing: Informatics and System*,» 2022, DOI: 10.1016/j.suscom.2022.100814.
- [40] C.-W. Hsu, Y.-H. Huang y N.-F. Huang, «Clasificación de madurez de Dragonfruit en tiempo real Sistema con Edge Computing basado en Red neuronal convolucional,» 2022, DOI:10.1109/ICOIN53446.2022.9687292.
- [41] Hiraoka, *Cámara web: ¿Cómo elegir la webcam ideal?* Accedido: 2-5-2024, 2024. dirección: <https://hiraoka.com.pe/blog/post/camara-web-como-elegir-la-webcam-ideal#:~:text=Una%20c%C3%A1mara%20web%20funciona%20al,para%20capturar%20im%C3%A1genes%20y%20videos..>
- [42] M. Tronics, «Servomotor MG90S Tower Pro,» 2024.
- [43] ROLTIA, *¿Qué es una cinta transportadora?* Accedido: 3-5-2024, 2024. dirección: <https://www.acelerapyme.gob.es/novedades/pildora/usuarios-y-ventajas-de-la-impresion-3d#:~:text=Las%20impresoras%203D%20ofrecen%20la,y%20el%20coste%20en%20electricidad..>
- [44] Utuo, *CINTA*, Accedido: 4-4-2024, 2024. dirección: <https://www.utuobelting.com/es/rubber-conveyor-belt.html>.
- [45] Reliance Foundry, *Acero Inoxidable 304 versus Acero Inoxidable 316*, Accedido: 3-5-2024, 2024. dirección: <https://www.reliance-foundry.com/blog/acero-inoxidable-304-vs-316-es#:~:text=El%20acero%20inoxidable%20304%20es,cantidades%20de%20carb%C3%B3n%20y%20manganeso..>

- [46] GCFGlobal, *¿Qué es una Impresora 3D?* Accedido: 3-5-2024, 2024. dirección: <https://edu.gcfglobal.org/es/cultura-tecnologica/que-es-una-impresora-3d/1/>.
- [47] Esun, *PETG*, Accedido: 3-5-2024, 2024. dirección: <https://www.esun3d.com/petg-product/>.
- [48] FILAMENT2PRINT, *¿Qué es el PETG?* Accedido: 3-5-2024, 2024. dirección: https://filament2print.com/es/blog/49_petg.html.

ANEXO A
PLANOS

Planos de las piezas acotadas en inventor:

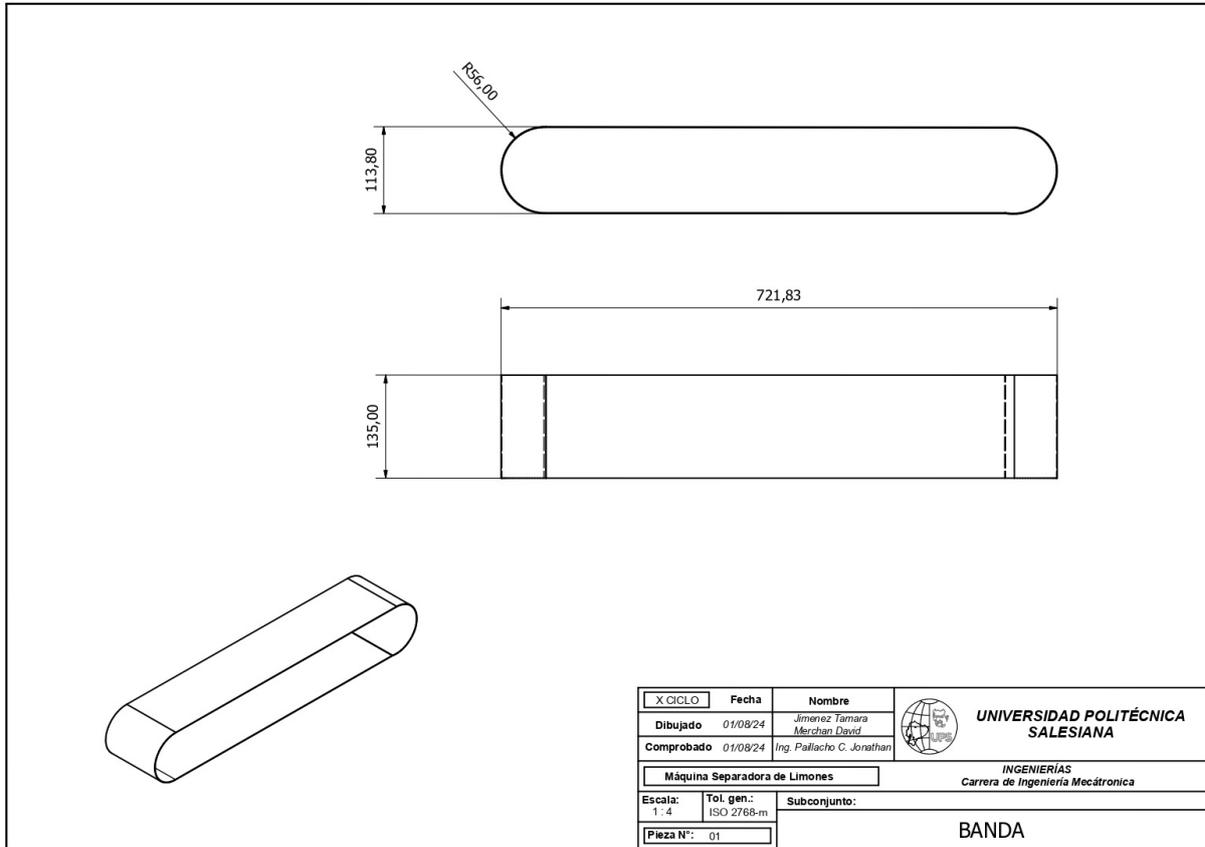


Figura 60. Banda. Fuente: Los Autores

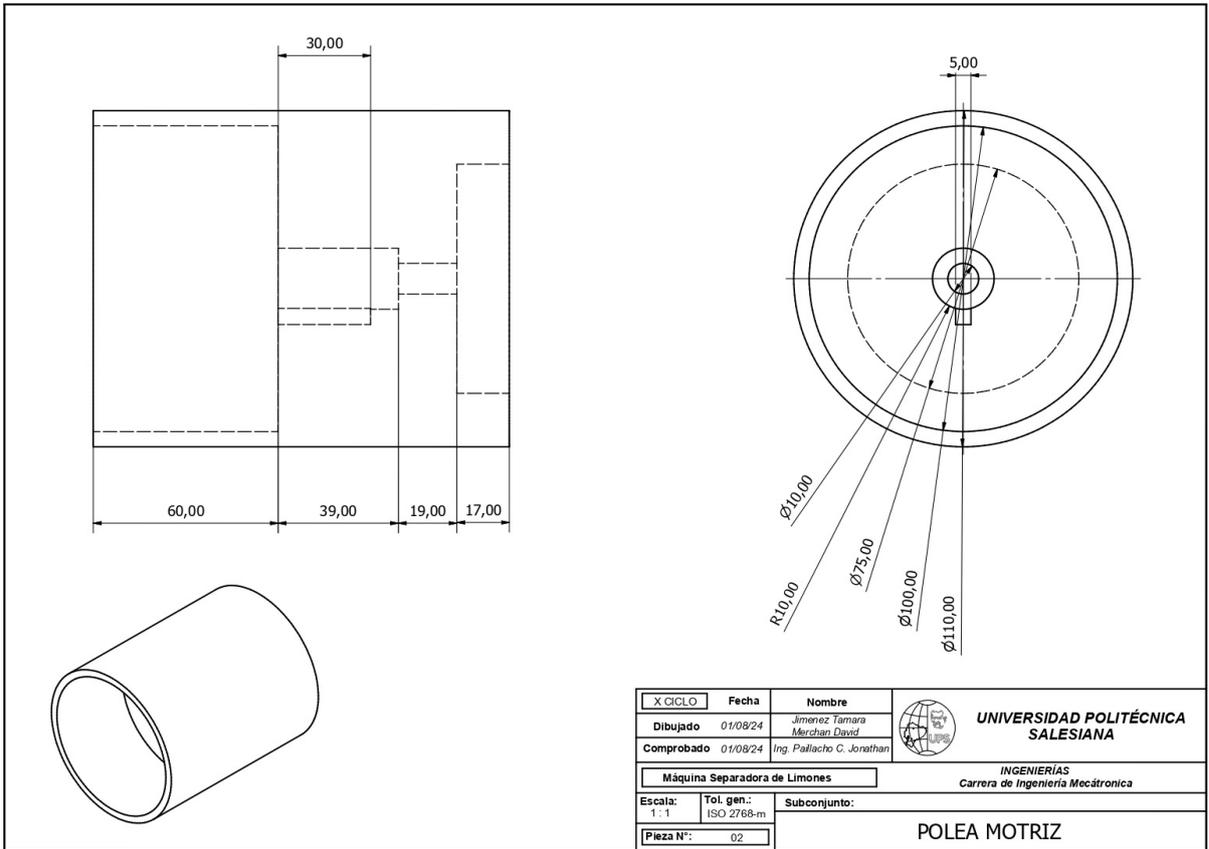


Figura 61. Polea motriz. Fuente: Los Autores

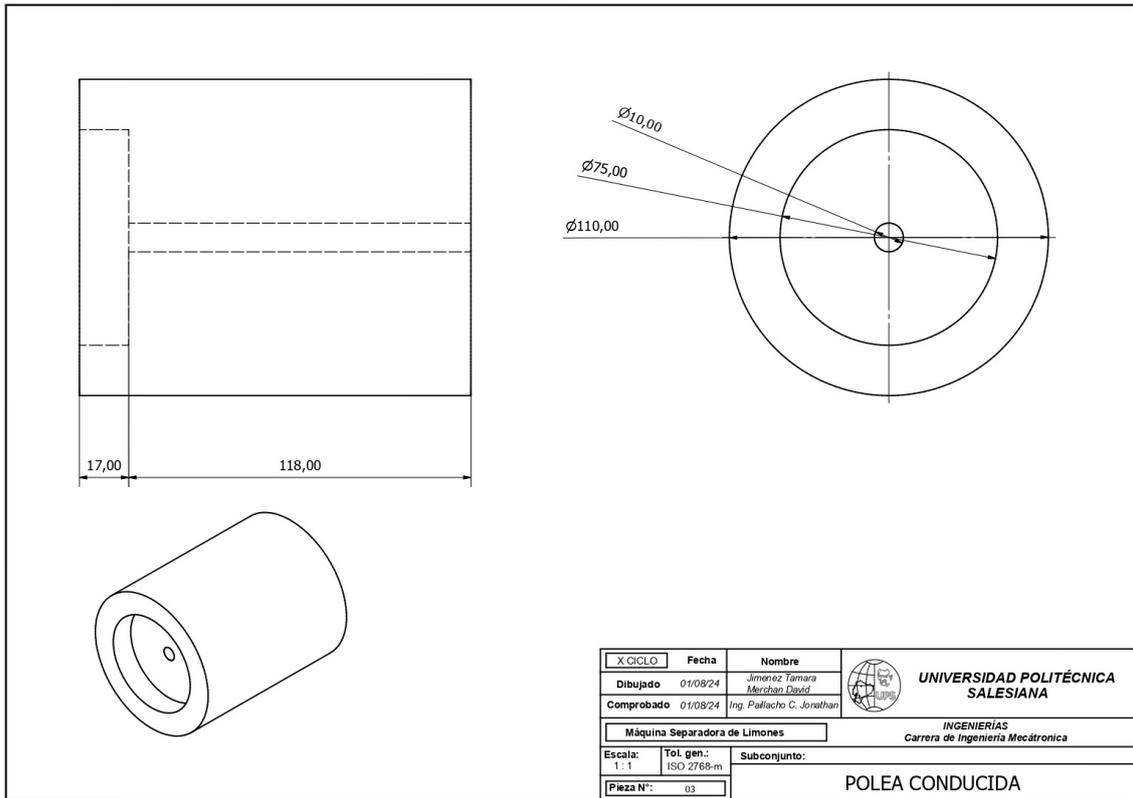


Figura 62. Polea conducida. Fuente: Los Autores

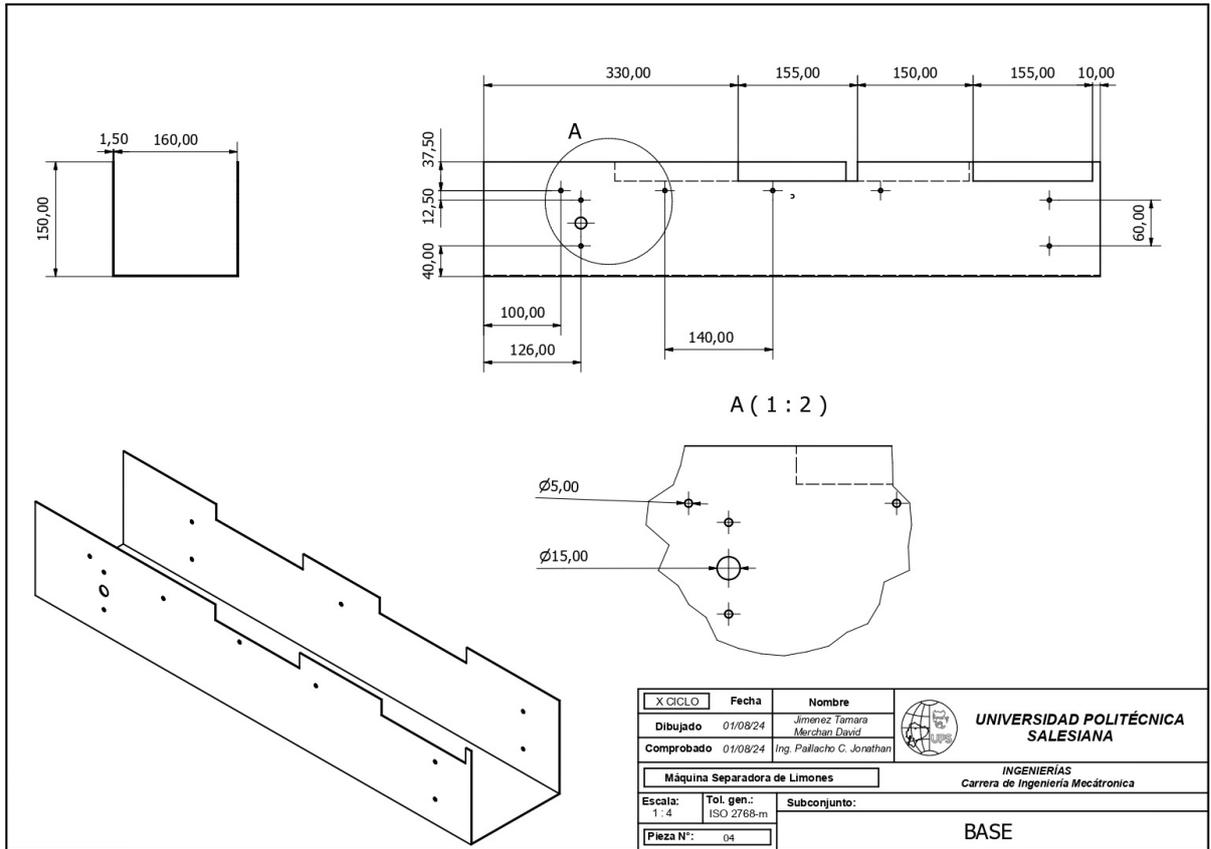


Figura 63. Base. Fuente: Los Autores

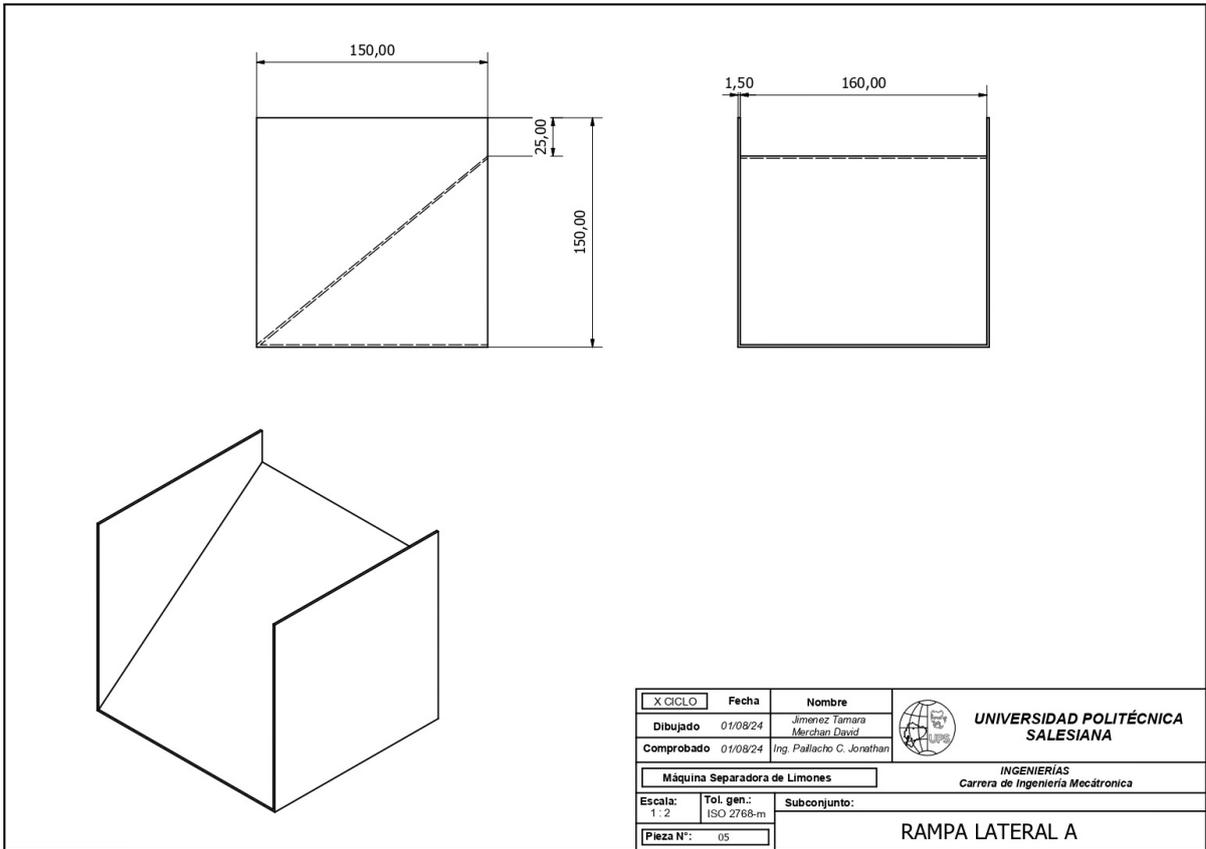


Figura 64. Rampa lateral A. Fuente: Los Autores

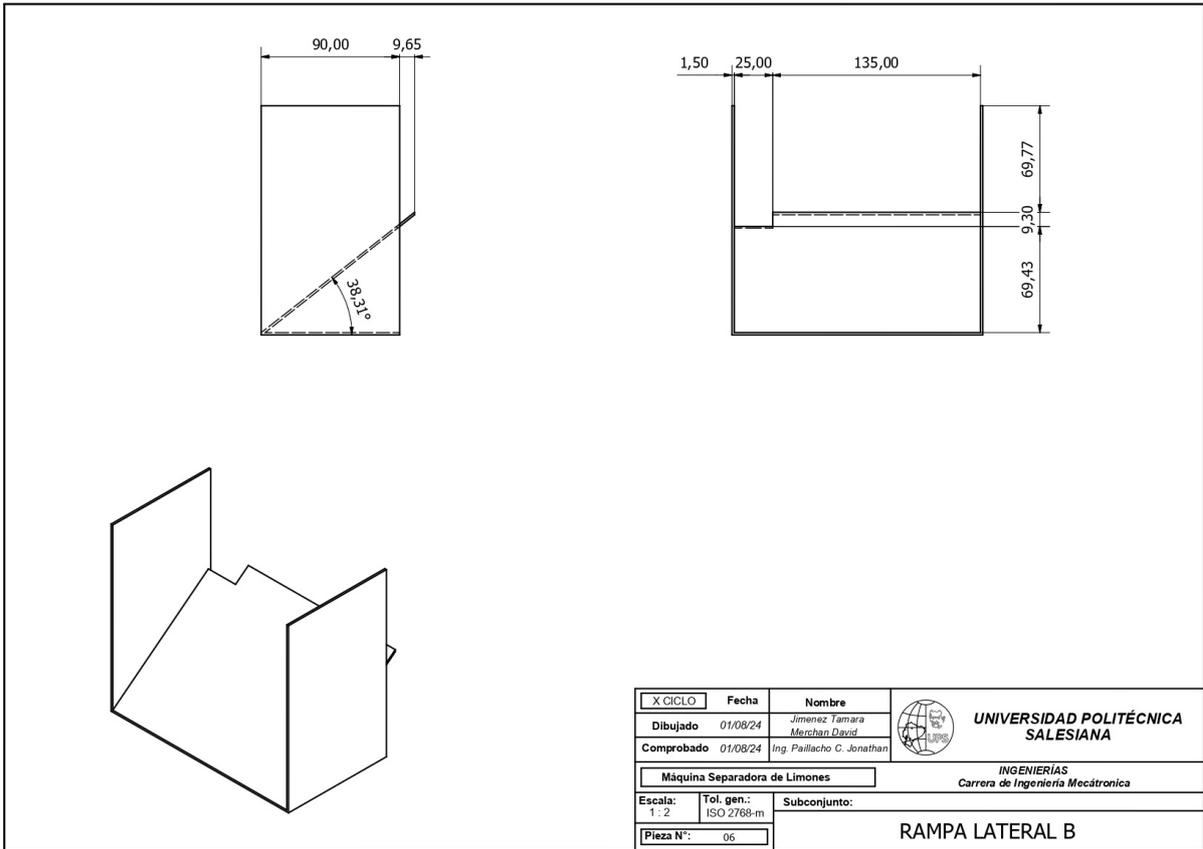


Figura 65. Rampa lateral B. Fuente: Los Autores

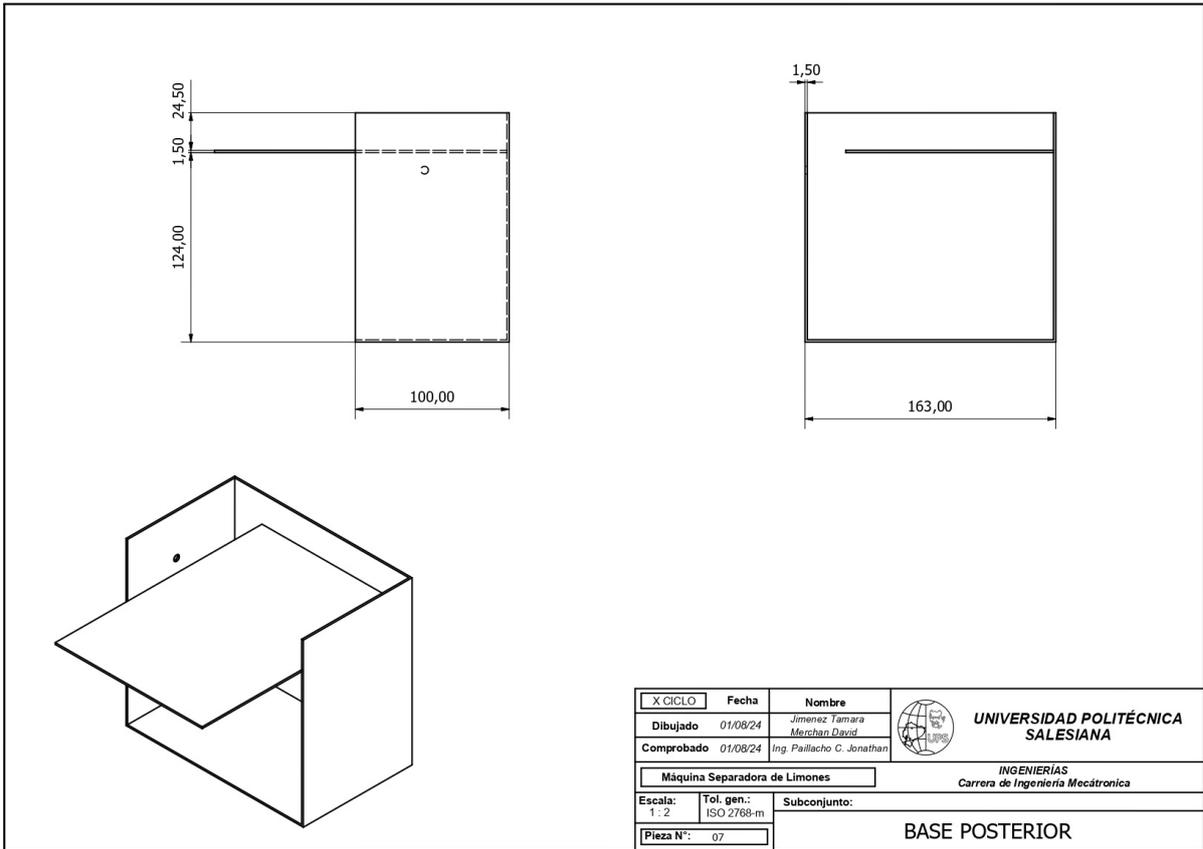


Figura 66. Base posterior. Fuente: Los Autores

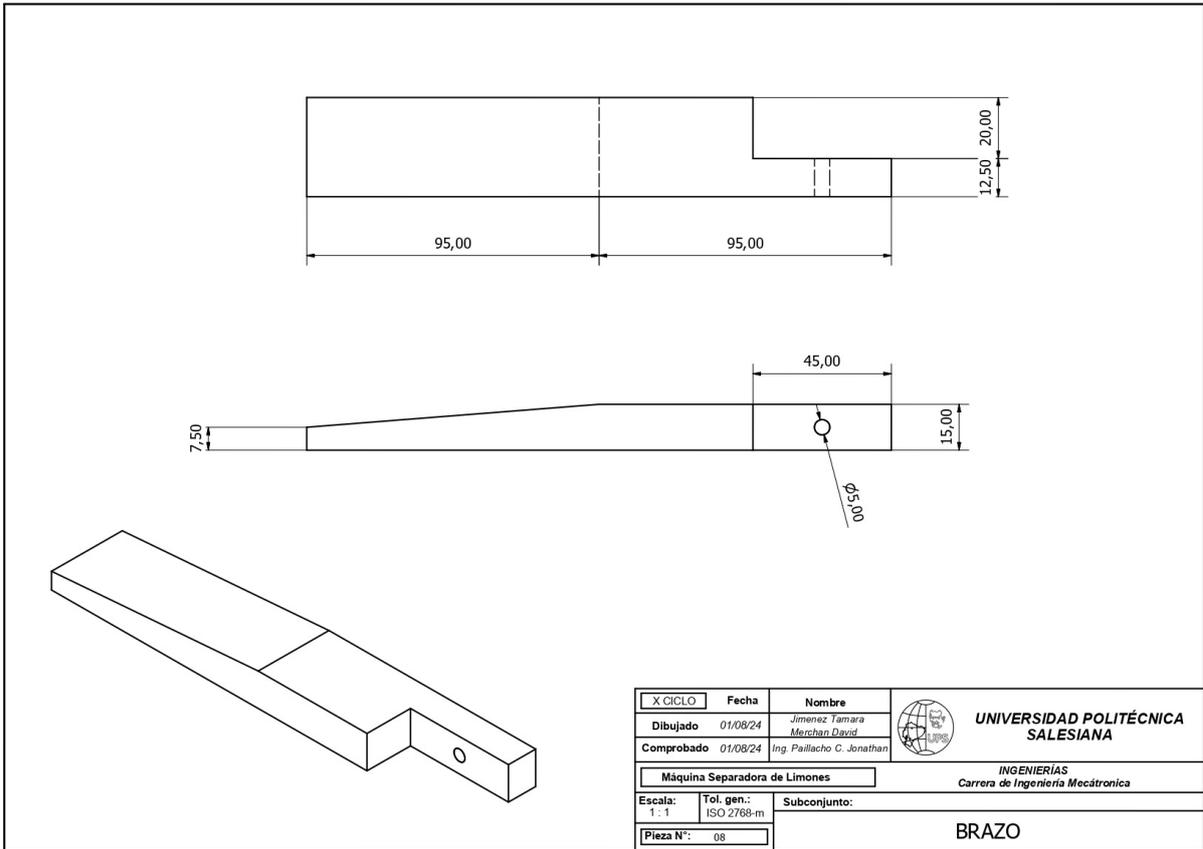


Figura 67. Brazo. Fuente: Los Autores

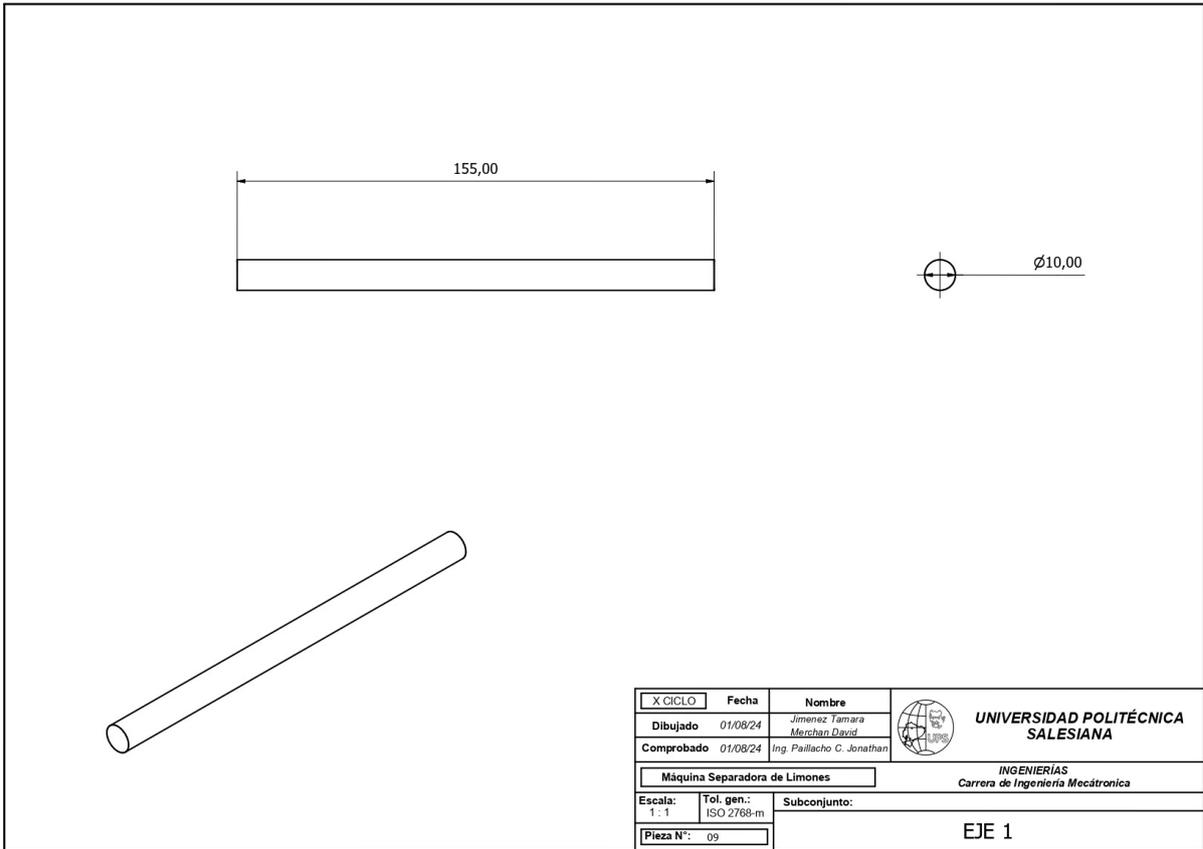


Figura 68. Eje 1. Fuente: Los Autores

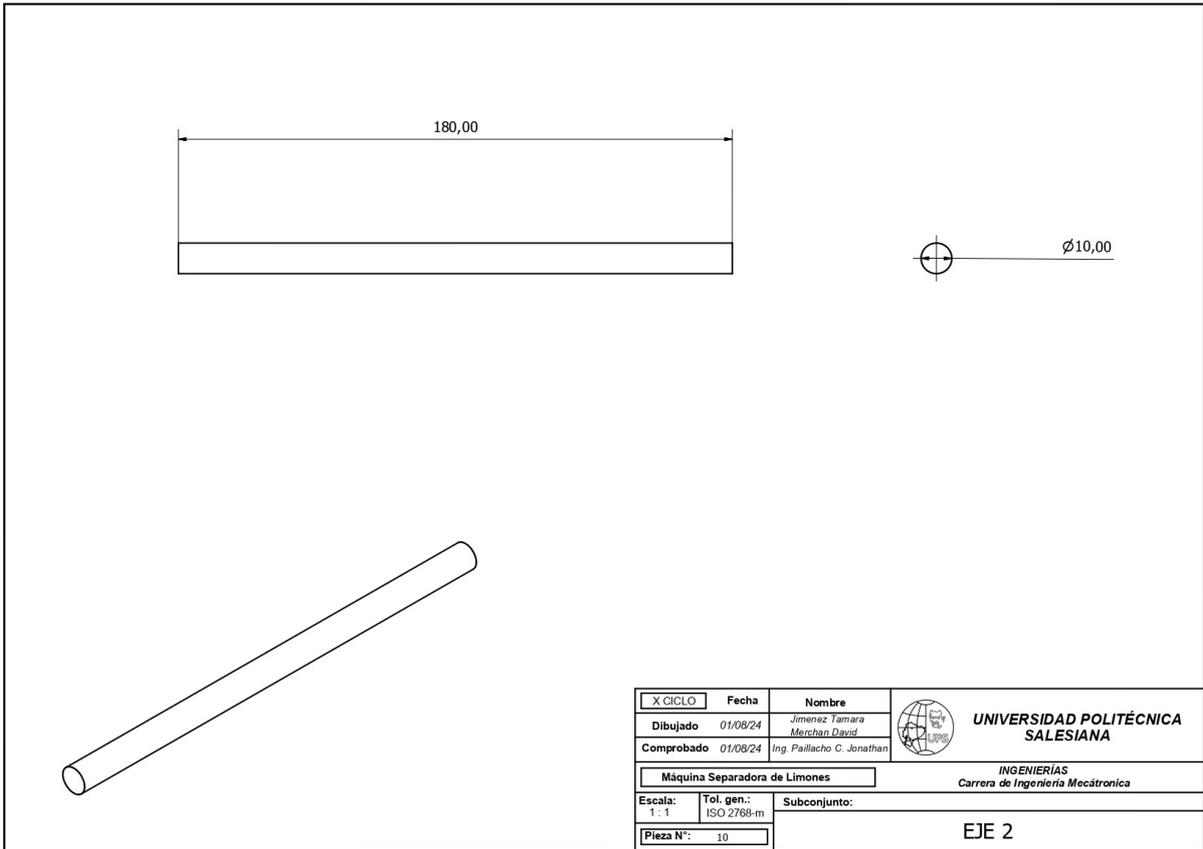


Figura 69. Eje 2. Fuente: Los Autores

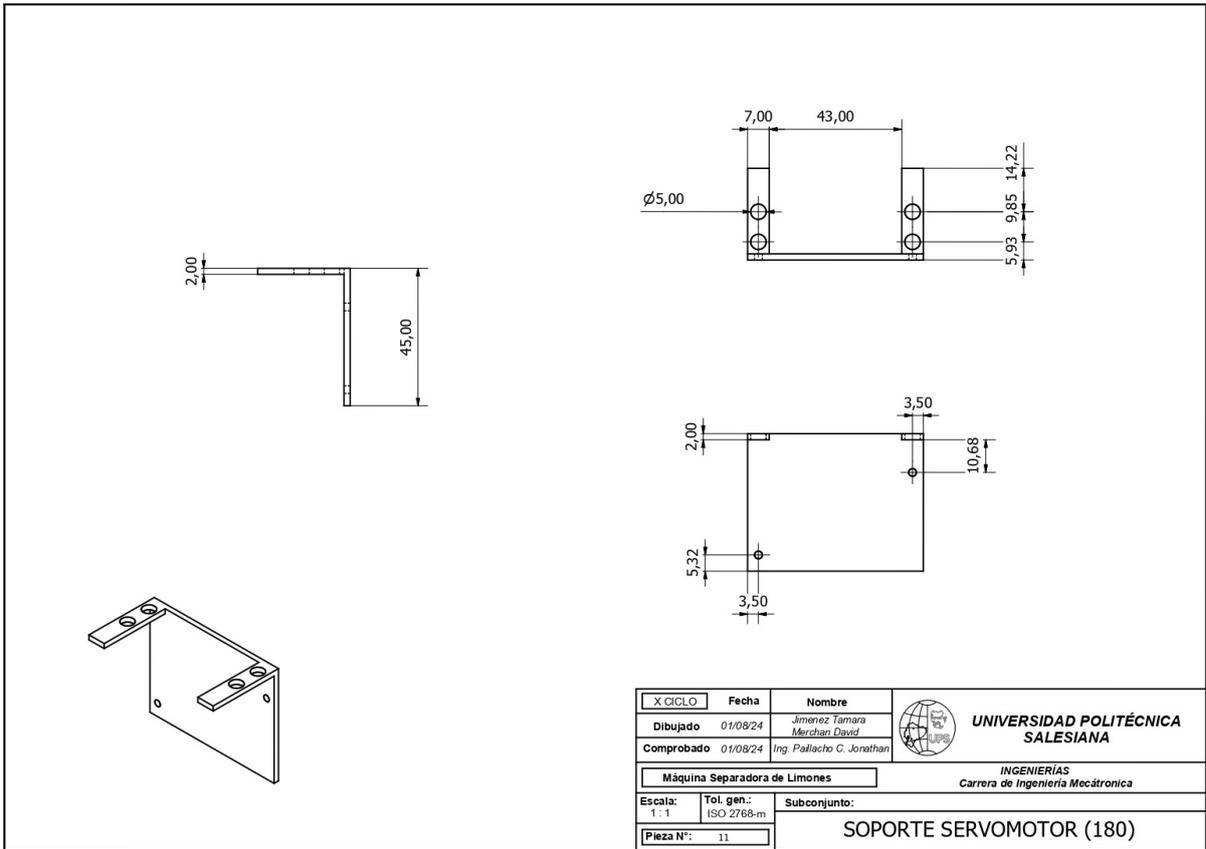


Figura 70. Soporte servomotor. Fuente: Los Autores

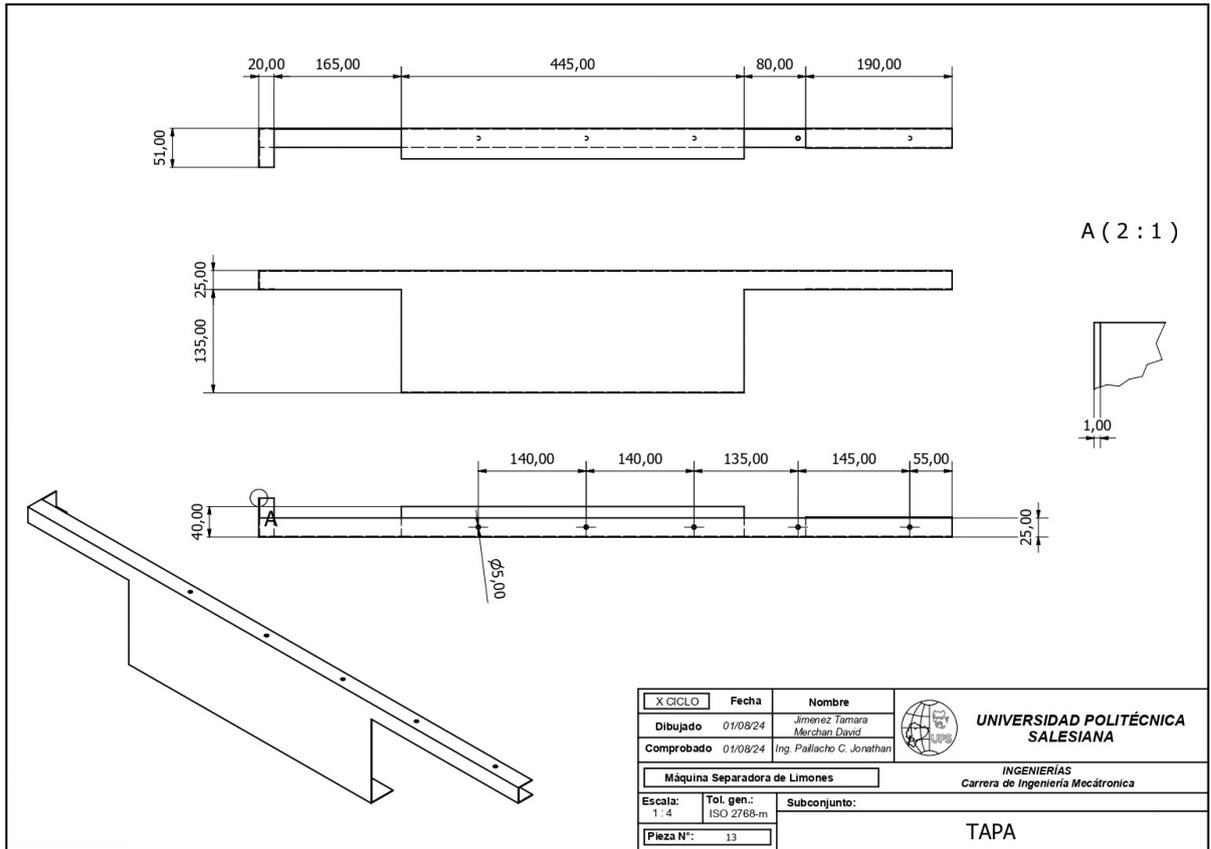


Figura 71. Tapa. Fuente: Los Autores

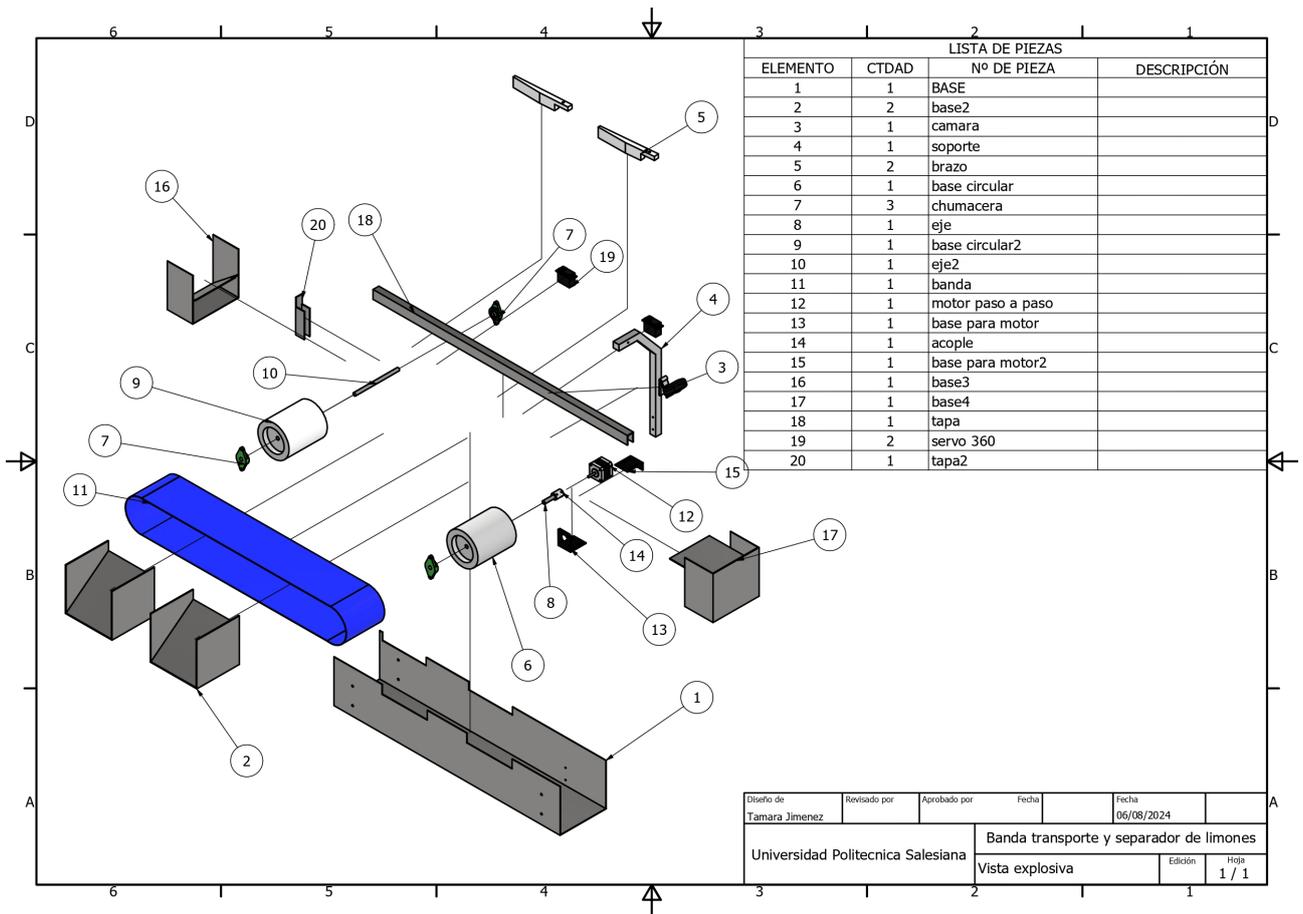


Figura 72. Vista explosiva de los elementos utilizados del proyecto. Fuente: Los Autores

ANEXO B
PROCESO DE CONSTRUCCIÓN DEL PROYECTO



Figura 73. Base del prototipo. Fuente: Los Autores



Figura 74. Acople de los servomotores 180. Fuente: Los Autores



Figura 75. Ensamblaje de la banda. Fuente: Los Autores



Figura 76. Interfaz gráfica del prototipo. Fuente: Los Autores

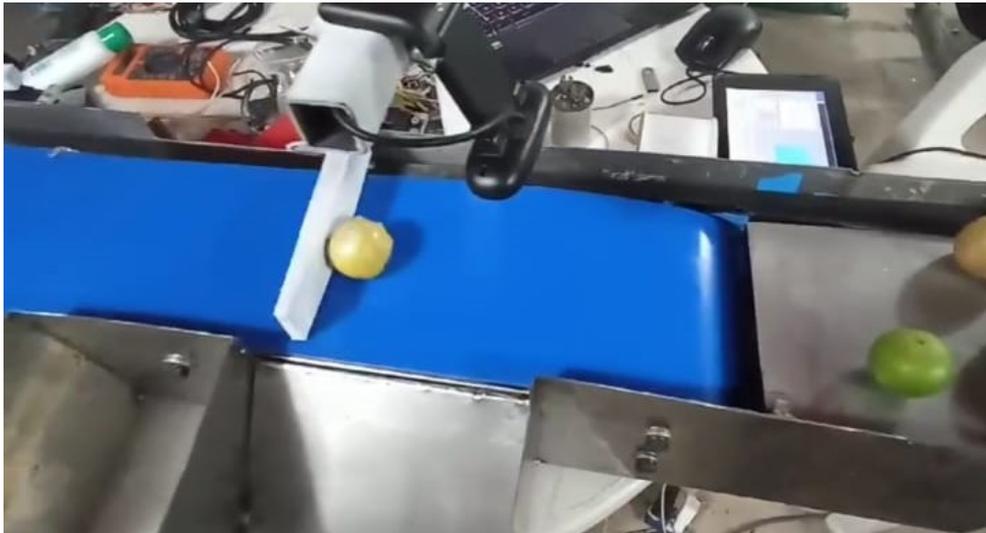


Figura 77. Pruebas de clasificación. Fuente: Los Autores



Figura 78. Servomotor 360. Fuente: Los Autores

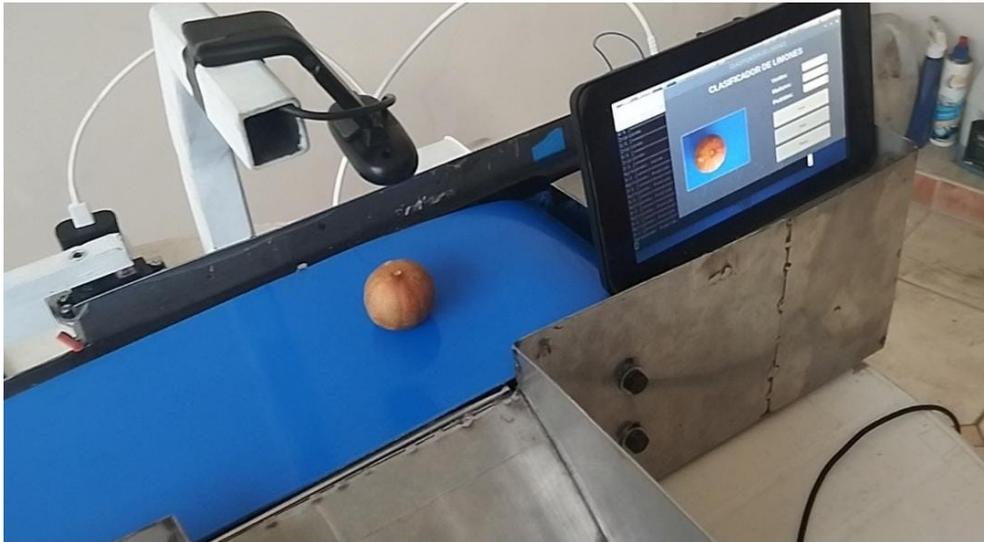


Figura 79. Pruebas de prototipo con interfaz gráfica. Fuente: Los Autores

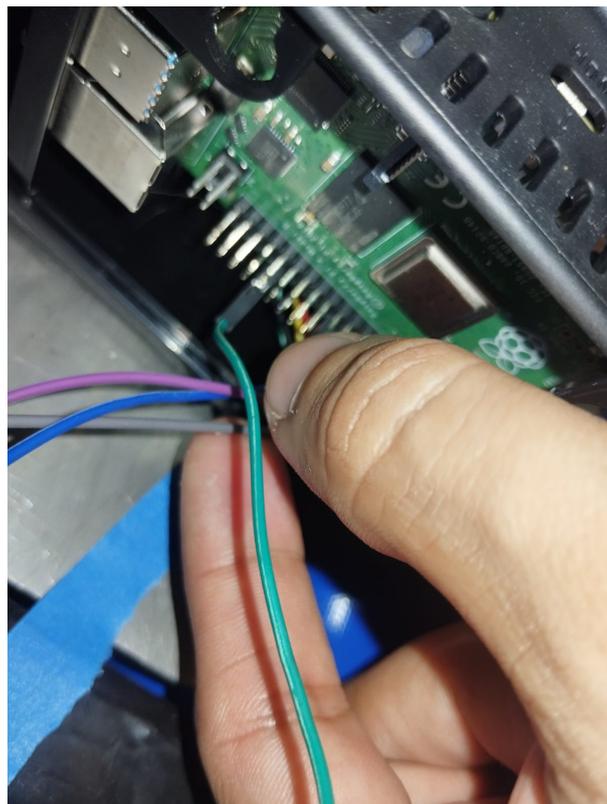


Figura 80. Instalación de Raspberry Pi 4. Fuente: Los Autores

ANEXO C

CÓDIGO DE PROGRAMACIÓN PARA LA RASPBERRY PI 4

A continuación se adjunta la programación:

```
import tkinter as tk # Librería para la interfaz grafica
from keras.models import load_model # TensorFlow is required for Keras to work
import cv2 # Librería para utilizar la cámara
import numpy as np # Librería para hacer operaciones con las imágenes
import time # Librería para los tiempos
import threading # Librería para manejar hilos
from PIL import Image, ImageTk # Librería para la interfaz grafica
import RPi.GPIO as GPIO # Librería para los pines
# Variable global
running = False
# Se crea la clase LemonClassifierApp
class LemonClassifierApp:
def __init__(self, root):
### ATRIBUTOS
# Atributos para la interfaz
self.root = root
self.root.title("CLASIFICADOR DE LIMONES")
# Configuración de los pines GPIO y PWM
self.SERVO_MADURO_PIN = 23
self.SERVO_DANADO_PIN = 12
self.SERVO_BANDA_PIN = 25
self.SERVO_MADURO_PWM_FREQ = 50
self.SERVO_DANADO_PWM_FREQ = 50
self.SERVO_BANDA_PWM_FREQ = 50
# Inicialización de GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(self.SERVO_MADURO_PIN, GPIO.OUT)
GPIO.setup(self.SERVO_DANADO_PIN, GPIO.OUT)
GPIO.setup(self.SERVO_BANDA_PIN, GPIO.OUT)
# Configuración de PWM para cada servo
self.servo_maduro_pwm = GPIO.PWM(self.SERVO_MADURO_PIN,
self.SERVO_MADURO_PWM_FREQ)
self.servo_danado_pwm = GPIO.PWM(self.SERVO_DANADO_PIN,
self.SERVO_DANADO_PWM_FREQ)
self.servo_banda_pwm = GPIO.PWM(self.SERVO_BANDA_PIN,
self.SERVO_BANDA_PWM_FREQ)
self.servo_maduro_pwm.start(0)
self.servo_danado_pwm.start(0)
self.servo_banda_pwm.start(0)
# Frame principal

```

```
46
self.main_frame = tk.Frame(self.root, bg="gray")
self.main_frame.pack(fill=tk.BOTH, expand=True)
# Título de la interfaz
self.title_label = tk.Label(
self.main_frame,
text="CLASIFICADOR DE LIMONES",
font=("Arial", 18, "bold"),
bg="gray",
fg="white",
)
self.title_label.pack(pady=10, anchor=tk.N)
# Frame para la cámara (lado izquierdo)
self.camera_frame = tk.Frame(self.main_frame, width=224,
height=224, bg="black")
self.camera_frame.pack(side=tk.LEFT, padx=40, pady=10)
# Frame para los contadores (lado derecho)
self.counter_frame = tk.Frame(self.main_frame, width=341,
height=600, bg="gray")
self.counter_frame.pack(side=tk.RIGHT, padx=40, pady=10, fill=tk.Y)
# Definir las variables para los contadores
self.verdes_count = 0
self.maduros_count = 0
self.podridos_count = 0
# Etiqueta para el contador Verdes
self.verdes_label = tk.Label(
self.counter_frame,
text="Verdes:",
bg="gray",
fg="white",
font=("Arial", 12, "bold"),
anchor=tk.W,
)
self.verdes_label.grid(row=0, column=0, sticky=tk.W, padx=20, pady=5)
# Etiqueta para el contador Maduros
self.maduros_label = tk.Label(
self.counter_frame,
text="Maduros:",
bg="gray",
fg="white",

```

```

font=("Arial", 12, "bold"),
anchor=tk.W,
)
self.maduros_label.grid(row=1, column=0, sticky=tk.W, padx=20, pady=5)
47
# Etiqueta para el contador Dañados
self.podridos_label = tk.Label(
self.counter_frame,
text="Dañados:",
bg="gray",
fg="white",
font=("Arial", 12, "bold"),
anchor=tk.W,
)
self.podridos_label.grid(row=2, column=0, sticky=tk.W, padx=20, pady=5)
# Cajas para los números de contadores
self.verdes_count_var = tk.StringVar()
self.verdes_count_var.set("0")
self.verdes_count_label = tk.Label(
self.counter_frame,
textvariable=self.verdes_count_var,
bg="white",
fg="black",
font=("Arial", 12),
width=10,
anchor=tk.E,
relief=tk.RIDGE,
)
self.verdes_count_label.grid(row=0, column=1, sticky=tk.E,
padx=20, pady=5)
self.maduros_count_var = tk.StringVar()
self.maduros_count_var.set("0")
self.maduros_count_label = tk.Label(
self.counter_frame,
textvariable=self.maduros_count_var,
bg="white",
fg="black",
font=("Arial", 12),
width=10,

```

```

relief=tk.RIDGE,
)
self.maduros_count_label.grid(row=1, column=1, sticky=tk.E,
padx=20, pady=5)
self.podridos_count_var = tk.StringVar()
self.podridos_count_var.set("0")
self.podridos_count_label = tk.Label(
self.counter_frame,
textvariable=self.podridos_count_var,
bg="white",
fg="black",
font=("Arial", 12),
48
width=10,
anchor=tk.E,
relief=tk.RIDGE,
)
self.podridos_count_label.grid(row=2, column=1, sticky=tk.E,
padx=20, pady=5)
# Crear un frame para los botones
self.buttons_frame = tk.Frame(self.counter_frame, bg="gray")
self.buttons_frame.grid(row=3, column=0, columnspan=2, pady=10,
sticky=tk.NSEW)
# Boton Start
self.start_button = tk.Button(
self.buttons_frame,
text="Start",
command=self.start_all, # Ejecutar el metodo start_all
font=("Arial", 10),
width=26,
height=2,
)
self.start_button.pack(padx=5, pady=5)
# Boton Stop
self.stop_button = tk.Button(
self.buttons_frame,
text="Stop",
command=self.stop_all, # Ejecutar el metodo stop_all
font=("Arial", 10),
width=26,

```

```

height=2,
)
self.stop_button.pack(padx=5, pady=5)
# Boton Reset
self.reset_button = tk.Button(
self.buttons_frame,
text="Reset",
# Ejecutar el metodo reset_counts
command=self.reset_counts,
font=("Arial", 10),
width=26,
height=2,
)
self.reset_button.pack(padx=5, pady=5)
# Variable para la imagen de la cámara
self.camera_image = None
# Variables para control de la cámara
49
self.camera = None
self.is_camera_running = False
# Modelo de clasificación y etiquetas
self.model = None
self.class_names = None
self.vector = []
self.vector_len = 8
# METODOS
# Metodo para ejecutar la cámara
def start_camera(self):
if not self.is_camera_running:
# Usar la cámara predeterminada
self.camera = cv2.VideoCapture(0)
self.is_camera_running = True
# Llamar al metodo update_camera
self.update_camera()
# Metodo para refrescar la cámara
def update_camera(self):
if self.is_camera_running:
ret, frame = self.camera.read()
if ret:

```

```

# Ajustar tamaño sin deformar la imagen
height, width, _ = frame.shape
aspect_ratio = width / height
# Ancho deseado del frame de la cámara
target_width = 224
target_height = int(target_width / aspect_ratio)
frame = cv2.resize(frame, (target_width, target_height))
frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
img = Image.fromarray(frame)
imgtk = ImageTk.PhotoImage(image=img)
# Etiqueta cámara
if not hasattr(self, "camera_label"):
self.camera_label = tk.Label(self.camera_frame)
self.camera_label.pack()
self.camera_label.configure(image=imgtk)
self.camera_label.image = imgtk
# Actualizar cada 10ms
self.root.after(10, self.update_camera)
# Ejecutar run.clasification como threading
def start_classification(self):
threading.Thread(target=self.run_classification).start()
50
# Ejecutar la clasificación utilizando la red
def run_classification(self):
# Cargar el modelo y las etiquetas
np.set_printoptions(suppress=True)
self.model = load_model("keras_model.h5", compile=False)
self.class_names = open("labels.txt", "r").readlines()
# Clasificar los limones
while self.is_camera_running:
ret, image = self.camera.read()
if ret:
image = cv2.resize(image, (224, 224),
interpolation=cv2.INTER_AREA)
image = np.asarray(image, dtype=np.float32)
image = np.reshape(1, 224, 224, 3)
image = (image / 127.5) - 1
prediction = self.model.predict(image)
index = np.argmax(prediction)
class_name = self.class_names[index]
confidence_score = prediction[0][index]

```

```

# Variable que contiene el tipo de limón
estado = class_name[2:].strip()
# Condiciones para ejecutar los servos
if estado == "Nada":
    print("Iniciando")
else:
    self.vector.append(estado)
    print("Agregando: {}".format(estado))
    if (
        self.vector.count("Verde")>=self.vector.count("Podrido")
        and self.vector.count("Verde")>=self.vector.count("Maduro")
        and len(self.vector) == self.vector_len
    ):
        self.verdes()
        self.verdes_count += 1
        self.update_counters()
        self.vector = []
    elif (
        self.vector.count("Podrido")>=self.vector.count("Verde")
        and self.vector.count("Podrido")>=self.vector.count("Maduro")
        and len(self.vector) == self.vector_len
    ):
        self.motor_podrito()
        self.podridos_count += 1
        self.update_counters()
        self.vector = []
    elif (
        self.vector.count("Maduro")>=self.vector.count("Verde")
        and self.vector.count("Maduro")>=self.vector.count("Podrido")
        and len(self.vector) == self.vector_len
    ):
        self.motor_maduros()
        self.maduros_count += 1
        self.update_counters()
        self.vector = []
# Metodo para detener la camera
def stop_camera(self):
    if self.is_camera_running:
        self.camera.release()

```

```

self.is_camera_running = False
# Metodo para resetear los contadores
def reset_counts(self):
    self.verdes_count = 0
    self.maduros_count = 0
    self.podridos_count = 0
    self.update_counters()
# Metodo para ir acualizando los contadores
def update_counters(self):
    self.verdes_count_var.set(f"{self.verdes_count:3d}")
    self.maduros_count_var.set(f"{self.maduros_count:3d}")
    self.podridos_count_var.set(f"{self.podridos_count:3d}")
# Encender el motor para limones podridos
def motor_podrito(self):
    print("Iniciando motor podrido")
    global running
    running = True
    self.thread = threading.Thread(target=self.run_servo,args=(7,))
    self.thread.start()
    self.set_servo_angle(self.servo_danado_pwm, 0)
    time.sleep(0.5)
    espera = 6
    while running == True and espera > 0:
        print(running)
        self.set_servo_angle(self.servo_danado_pwm, 80)
        time.sleep(1)
    #print(espera)
    if espera == 1 and running == True:
        self.set_servo_angle(self.servo_danado_pwm, 0)
        time.sleep(0.5)
        espera = espera - 1
    print("Terminado motor podrido")
    51
# Encender el motor para limones maduros
def motor_maduros(self):
    print("Iniciando motor maduros")
    global running
    running = True
    self.thread = threading.Thread(target=self.run_servo,args=(4,))
    self.thread.start()
    self.set_servo_angle(self.servo_maduro_pwm, 0)

```

```

time.sleep(0.5)
espera = espera - 1
print("Terminado motor danado")
# Decisiones al tener limones verdes
def verdes(self):
    global running
    running = True
    print("Encender motor verde")
    servo_thread = threading.Thread(target=self.run_servo, args=(9,))
    servo_thread.start()
    time.sleep(9)
    print("Terminado motor verde")
    # Girar el motor un ángulo deseado
    def set_servo_angle(self, pwm, angle):
        duty_cycle = (angle / 18) + 2
        pwm.ChangeDutyCycle(duty_cycle)
    time.sleep(1)
    pwm.ChangeDutyCycle(0)
    # Start - ejecuta al precionar el botón
    def start_all(self):
        self.set_servo_angle(self.servo_maduro_pwm, 0)
        self.set_servo_angle(self.servo_danado_pwm, 0)
        time.sleep(0.5)
        self.vector = []
        self.start_camera()
        self.start_classification()
    # Stop - ejecuta al precionar el botón
    def stop_all(self):
        global running
        53
        running = False
        print("Stop Ejecutado")
        self.stop()
        self.stop_camera()
    # Ejecutar el servo de la banda
    def run_servo(self, tiempo):
        global running
        time.sleep(1)
        while tiempo > 0 and running == True:
            #print("Ejecutando servo")
            self.servo_banda_pwm.ChangeDutyCycle(10)

```

```

time.sleep(1)
tiempo = tiempo - 1
self.servo_banda_pwm.ChangeDutyCycle(7)
time.sleep(0.5)
# Método stop para detener la banda
def stop(self):
    self.servo_banda_pwm.ChangeDutyCycle(7)
    time.sleep(1)
    print("Detener todo")
    # Ejecutar el código
    if __name__ == "__main__":
        # Ejecutar la interfaz
        root = tk.Tk()
        app = LemonClassifierApp(root)
        root.mainloop()

```