



UNIVERSIDAD TECNOLÓGICA ISRAEL
ESCUELA DE POSGRADOS “ESPOG”

MAESTRÍA EN ELECTRÓNICA Y AUTOMATIZACIÓN

Resolución: RPC-SO-09-No.265-2021

PROYECTO DE TITULACIÓN EN OPCIÓN AL GRADO DE MAGÍSTER

Título del proyecto:
Sistema automático de ovoscopia para análisis de calidad de huevos de gallina mediante procesamiento digital de imágenes
Línea de Investigación:
Ciencias de la ingeniería aplicadas a la producción, sociedad y desarrollo sustentable
Campo amplio de conocimiento:
Ingeniería, industria y construcción
Autor/a:
Roberto Javier Miranda Coronel
Tutor/a:
Wilmer Fabián Albarracín Guarochico

Quito – Ecuador

2022

APROBACIÓN DEL TUTOR



Yo, Wilmer Fabián Albarracín Guarochico con C.I: 1713341152 en mi calidad de Tutor del proyecto de investigación titulado: Sistema automático de ovoscopia para análisis de calidad de huevos de gallina mediante procesamiento digital de imágenes.

Elaborado por: Roberto Javier Miranda Coronel, de C.I: 1804467338, estudiante de la Maestría en Electrónica y Automatización, de la **UNIVERSIDAD TECNOLÓGICA ISRAEL (UISRAEL)**, como parte de los requisitos sustanciales con fines de obtener el Título de Magister, me permito declarar que luego de haber orientado, analizado y revisado el trabajo de titulación, lo apruebo en todas sus partes.

Quito D.M., 9 de septiembre de 2022

Firma

DECLARACIÓN DE AUTORIZACIÓN POR PARTE DEL ESTUDIANTE



Yo, Roberto Javier Miranda Coronel con C.I: 1804467338, autor del proyecto de titulación denominado: Sistema automático de ovoscopía para análisis de calidad de huevos de gallina mediante procesamiento digital de imágenes. Previo a la obtención del título de Magister en Electrónica y Automatización.

1. Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar el respectivo trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Manifiesto mi voluntad de ceder a la Universidad Tecnológica Israel los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor@ del trabajo de titulación, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital como parte del acervo bibliográfico de la Universidad Tecnológica Israel.
3. Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de prosperidad intelectual vigentes.

Quito D.M., 9 de septiembre de 2022

Firma

Tabla de contenidos

APROBACIÓN DEL TUTOR	2
DECLARACIÓN DE AUTORIZACIÓN POR PARTE DEL ESTUDIANTE	3
INFORMACIÓN GENERAL	7
Contextualización del tema	7
Problema de investigación.....	7
Objetivo general	8
Objetivos específicos	8
Vinculación con la sociedad y beneficiarios directos:	8
CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO	10
1.1. Contextualización general del estado del arte	10
1.2. Proceso investigativo metodológico	16
CAPÍTULO II: PROPUESTA.....	19
2.1. Fundamentos teóricos aplicados	19
2.2. Descripción de la propuesta	22
2.3. Validación de la propuesta.....	32
2.4. Matriz de articulación de la propuesta	36
2.5. Análisis de Resultados. Presentación y discusión.	38
CONCLUSIONES.....	46
RECOMENDACIONES.....	47
BIBLIOGRAFÍA.....	48
ANEXOS	52

Índice de tablas

Tabla 1.....	24
Especificaciones técnicas de microcomputadoras para procesamiento de imágenes e interconexión de periféricos.....	24
Tabla 2.....	26
Especificaciones técnicas de cámaras digitales.	26
Tabla 3.....	32
Descripción de perfil de validadores.....	32
Tabla 4.....	33
Criterios de evaluación.	33
Tabla 5.....	33
Escala de evaluación otorgada por el M.Sc. Ángel Fernández E.....	33
Tabla 6.....	34
Escala de evaluación otorgada por el PhD. Martín González V.	34
Tabla 7.....	34
Escala de evaluación otorgada por el Mg. José Naranjo R.	34
Tabla 8.....	35
Escala de evaluación otorgada por el Mg. Franklin Salazar L.	35
Tabla 9.....	36
Matriz de articulación.	36
Tabla 10.....	39
Resultados de un set de imágenes analizadas por el sistema.	39
Tabla 11.....	41
Factores de estudio del análisis experimental.....	41
Tabla 12.....	41
Factores de estudio del análisis experimental.....	41
Tabla 13.....	42
Resultados del análisis experimental por repeticiones y características definidas.....	42

Índice de figuras

Figura 1.	17
Diagrama de flujo de las actividades desarrolladas en el proyecto.....	17
Figura 2.	23
Diagrama de bloques de interconexión de subsistemas del sistema automático de ovoscopia.	23
Figura 3.	23
Interconexión de dispositivos del sistema automático de ovoscopia.	23
Figura 4.	28
Banda transportadora desarrollada para el sistema automático de ovoscopia.....	28
Figura 5.	28
Eslabón que forma parte de las cadenas para el sistema automático de ovoscopia.	28
Figura 6.	29
Segmento de cadenas del sistema automático de ovoscopia.	29
Figura 7.	29
Engranaje para adaptación a los motores del sistema automático de ovoscopia.	29
Figura 8.	30
Engranaje para retroalimentación de cadenas del sistema automático de ovoscopia.....	30
Figura 9.	30
Cubeta del sistema automático de ovoscopia.	30
Figura 10.	42
Resultados finales del Tratamiento 1.	42
Figura 11.	43
Resultados finales del Tratamiento 2.	43
Figura 12.	43
Resultados finales del Tratamiento 3.	43
Figura 13.	44
Resultados finales del Tratamiento 4.	44
Figura 14.	44
Resultados finales del Tratamiento 5.	44
Figura 15.	45
Resultados finales del Tratamiento 6.	45

INFORMACIÓN GENERAL

Contextualización del tema

Los sistemas de procesamiento de imágenes son ampliamente utilizados para solventar requerimientos de pequeñas, medianas y grandes industrias que demandan mayor precisión sobre el tratamiento de sus productos. La necesidad de intensificar los controles de calidad crece constantemente ya que el consumidor se vuelve más exigente y las empresas buscan mantener la relación calidad-precio.

La investigación se centra en un prototipo de sistema automático de ovoscopía para análisis de calidad de huevos de gallina mediante procesamiento digital de imágenes, con el propósito de estudiar las características físicas de la cáscara, determinando posibles alteraciones o daños que puedan presentar internamente.

El prototipo cuenta con un sistema mecánico basado en una banda transportadora de cadena diseñada y adaptada a los requerimientos propuestos, cuyos elementos han sido impresos en 3D a base de PLA; un sistema informático encargado del proceso de visión artificial y almacenamiento de la información en una base de datos, adquiriendo imágenes de mayor calidad al emplear la cámara de un Smartphone vinculado a través de un servidor; y un sistema electrónico de control para regular la velocidad y accionamiento de los motores que permiten girar las cadenas de la banda, además de sincronizar la captura de imágenes. Cada uno de los componentes empleados en este proyecto tienen base en Hardware y Software libre, que constituye una reducción considerable en el aspecto económico frente a otros sistemas.

Problema de investigación

El presente proyecto plantea establecer un sistema de control de calidad de huevos de gallina de bajo costo, mediante procesamiento digital de imágenes, con enfoque a distintas aplicaciones dentro del sector avícola, a fin de automatizar este proceso encaminándolo a una reducción considerable de errores y agotamiento humano, implícitos en dicha actividad. Es un factor preponderante el hecho de que muchos de los rasgos físicos de los huevos de gallina, no se pueden apreciar ni mucho menos clasificar a simple vista, por lo que resulta necesario un diagnóstico interno pero que no implique la destrucción de los objetos de estudio.

Varios proyectos se centran en una sola característica como objeto de estudio, pero mientras más detallado sea el análisis, la repercusión en la industria avícola será mayor. Al contar con un sistema integral que permita verificar distintos factores físicos, sean estos externos o internos en los huevos de gallina, es posible parametrizar y clasificar adecuadamente los objetos de estudio, permitiendo

también hacer una retrospectiva en función de los análisis que se realicen para estimar otros problemas con base en los datos que se obtengan y así realizar investigaciones estadísticas futuras.

Además, existen pocos estudios que establecen un tiempo estimado de sus análisis para realizar proyecciones objetivas fuera de los ensayos de laboratorio, por lo que, en función a lo enunciado previamente se prevé precisar un tiempo de ejecución del sistema para cada análisis proyectando resultados a pequeña, mediana o gran escala.

Objetivo general

Desarrollar un prototipo de sistema automático de ovoscopia, mediante técnicas de procesamiento digital de imágenes, para analizar la calidad de huevos de gallina.

Objetivos específicos

Contextualizar los sistemas de ovoscopia mediante procesamiento digital de imágenes para establecer técnicas y métodos que conlleven a la evaluación de calidad de los huevos de gallina.

Establecer un prototipo de sistema automático de ovoscopia, mediante técnicas de procesamiento digital de imágenes, empleando hardware y software libre para optimizar los procesos de verificación de calidad de los huevos de gallina.

Evaluar la calidad de huevos de gallina de distintos proveedores de la provincia de Tungurahua, mediante el prototipo de sistema automático de ovoscopia, posterior a su desarrollo, para la comparación de características físicas de los objetos de estudio.

Vinculación con la sociedad y beneficiarios directos:

El presente proyecto de investigación es un aporte para la industria avícola y los consumidores de huevos de gallina o población en general, mediante el cual, cualquier industria, indistintamente de su tamaño, pueden ampliar su desarrollo en función de sus requerimientos invirtiendo o auspiciando la producción masiva del sistema. Es importante considerar que lo aquí planteado es un prototipo que puede ser mejorado u optimizado según el enfoque que se le dé, además, una de las características fundamentales es que se ha empleado recursos de hardware y software libre, es decir, de fácil acceso a nivel nacional para su réplica con un bajo costo.

Este estudio, se plantea como base investigativa o soporte para futuros profesionales que se especialicen en sistemas electrónicos, automatismos y sobre todo en temas de procesamiento de imágenes o visión artificial; así también, se considera como guía para el desarrollo de nuevos proyectos.

Las granjas avícolas tendrán un mejor control de sus productos (huevos de gallina), además, podrán analizar problemas en los procesos de producción o manipulación de los mismos. A su vez, los distribuidores y también los productores de pollos (incubadoras), podrán analizar la mejor opción en cuanto a sus proveedores, a fin de asegurar la adquisición de productos de calidad.

CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO

1.1. Contextualización general del estado del arte

Generalizando los conceptos teóricos en los que se fundamenta la presente investigación y en base a estudios previos realizados, se tiene lo siguiente:

1.1.1 Conceptos básicos

Ovoscopía

La ovoscopía es una técnica de análisis basada en la traslucidez de las cáscaras de los huevos, a través de los cuales se transmite un haz de luz, a fin de visualizar la estructura interna, sin recurrir a su apertura, mediante un dispositivo denominado ovoscopio. Esta técnica permite observar si el huevo se ha fecundado y si presenta alteraciones o defectos en su composición física. “Los huevos totalmente oscuros son huevos podridos” (Fernández y Jurado, 2018; Campos, 2010).

Usos de los huevos de gallina

Los principales usos, conocidos, de los huevos de gallina se presentan a continuación:

- Consumo masivo, para alimentación de la población humana por su alto índice de nutrientes.
- Industria alimenticia, para elaborar productos como salsas, harinas, postres, pan, etc.
- Incubación, para la producción de pollos a gran escala.

Problemas comunes presentes en los huevos de gallina

En la industria avícola, debido a la gran cantidad de producción y los distintos usos que se les da a los huevos de gallina, es imposible mantener un ciclo natural; normalmente se industrializan los procesos, por lo cual, es inherente la intervención o manipulación humana que puede repercutir en la integridad los productos en cuestión. En concordancia con los estudios realizados por Soler y Bueso (2017), Fernández y Jurado (2018) y Albán (2018), se presentan los siguientes problemas:

- Marcas de jaula: normalmente presentadas debido al contacto con estructuras metálicas en su almacenamiento o transporte.
- Grietas: que pueden extenderse longitudinalmente o marcarse con distintas formas como estrellas por golpes normalmente ocasionados durante el transporte.
- Rupturas o Agujeros: pueden ser marcas puntuales ocasionadas por el contacto con objetos agudos o golpes por una mala manipulación o transporte.

Además de problemas por manipulación o almacenamiento, pueden generarse problemas genéticos por mala alimentación de las ponedoras, su edad u otros factores. Haciendo una

retrospectiva a lo planteado previamente por Soler y Bueso (2017), Fernández y Jurado (2018) y Albán (2018), a continuación, se describen ciertos problemas fisiológicos de los huevos de gallina:

- Gránulos de material calcificado o cáscaras ásperas: se muestran como una anomalía sobre la superficie del huevo, denotando una capa de sales o exceso de calcio.
- Cáscaras finas, traslúcidas o moteadas: al emplear un ovoscopio, la cáscara se muestra más brillante que la de otros huevos en buen estado, esto puede presentarse en la totalidad de la cáscara o únicamente en ciertas partes como una especie de mancha.
- Cáscaras rugosas, o estriadas: se muestran como irregularidades, marcas o malformaciones en la cáscara, en pequeña escala.
- Cáscaras con exceso de poros: al visualizar los huevos en el ovoscopio estos poros se marcan como puntos traslúcidos a lo largo de toda la cáscara.
- Huevos con lombrices: son poco comunes, pero pueden presentarse si las lombrices pasan del intestino al oviducto de las ponedoras, insertándose en el huevo previo a su puesta.
- Huevo corregido en útero: presentan una especie de grietas longitudinales a través de la cáscara, como si se segmentara el huevo por una fuerza ejercida sobre el mismo.

Normativa para comercialización de huevos y ovoproductos en Ecuador

El Servicio Ecuatoriano de Normalización (INEN, 2013), determina, tanto para huevos frescos y ovoproductos de gallina y otras especies aviares, que:

- De acuerdo a su peso, los huevos de gallina frescos se clasifican en: Supergigantes (mayores o iguales a 76 g), Gigantes (mayores o iguales a 70 g y menores a 76 g), Extragrandes (mayores o iguales a 64 g y menores a 70 g), Grandes (mayores o iguales a 58 g y menores a 64 g), Medianos (mayores o iguales a 50 g y menores a 58 g), Pequeños (mayores o iguales a 46 g y menores a 50 g) e Inicial (menores a 46 g).
- Como disposiciones generales de comercialización: los huevos deben manipularse y transportarse asegurando las condiciones satisfactorias. Los huevos no deben contener restos químicos en cantidades fuera de lo tolerable según las normativas vigentes. Debe prevalecer la higiene en todas las instancias y procesos de comercialización. Además, se prohíbe comercializar huevos que presenten: putrefacción, manchas internas de sangre de más de 3 mm, embriones, moho, bacterias, parásitos, insectos, caducidad y organismos raros.

Procesamiento digital de imágenes

El procesamiento digital de imágenes, de acuerdo a los planteamientos de McAndrew (2004) y Pitas (2000), se refiere a un mecanismo de recopilación de la información de una imagen base,

descomponiéndola, mediante técnicas computarizadas, en contenido discreto, a fin de cambiar su esencia, obteniendo una imagen final con características diferentes previamente establecidas.

Uno de los objetivos de procesar o modificar una imagen es obtener la mejor información visual para la interpretación humana. Otro objetivo es adecuar la información para la percepción automática de una máquina. Bajo estas condiciones es importante precisar que, el uso de técnicas que permiten que una imagen se vea mejor según la percepción humana, es el peor procedimiento al emplear procesamientos automáticos y viceversa.

Visión artificial

El término visión artificial o visión por computadora se refiere al proceso automático de análisis de imágenes digitales, comparando su contenido original con los parámetros establecidos y modificando dicho contenido si es necesario. Este proceso se realiza mediante sistemas embebidos o computadoras y se basa en la teoría de procesamiento digital para obtener una imagen simbólica con información detallada. La visión artificial simula las funciones de la visión humana (Pitas, 2000).

Microcomputadora

Una microcomputadora es una computadora de tamaño reducido, equivalente al de una tarjeta de crédito, que permite la interconexión de periféricos para mantener la interacción humano-máquina, con ciertas limitaciones en su capacidad de procesamiento o almacenamiento de datos, pero que también permite controlar un sinnúmero de dispositivos a través de distintos métodos de comunicación avanzada o protocolos basados en estándares electrónicos.

Microcontrolador

“El microcontrolador es un circuito integrado (chip) que contiene memoria (para almacenar programas), un procesador (para procesar y llevar a cabo los programas) y pines de entrada/salida (para conectar interruptores, sensores, y dispositivos de salida tales como motores). Los microcontroladores se compran en “blanco” (vacíos) luego se programan con un programa específico de control. Este programa es primero escrito en un ordenador y luego “descargado” en el chip del microcontrolador. Una vez programado, el microcontrolador se inserta dentro de un producto para hacer al producto más inteligente y fácil de utilizar” (Lascorz, 2009).

Python

“Python es un lenguaje de programación que permite trabajar con una velocidad mayor y sobre todo integrar distintos sistemas de manera eficiente. La utilización de este lenguaje de

programación puede aumentar la productividad en las empresas y reducir costos de mantenimiento. Al igual que otros lenguajes, Python puede ser empleado en distintos sistemas operativos como Windows, Linux o MAC, limitando su funcionalidad en cada uno de ellos en concordancia con los módulos que puedan emplearse y las versiones de Python” (Fernández y Jurado, 2018).

Banda transportadora

Una banda transportadora es una estructura simple y adaptable que puede ser construida con distintos tipos de materiales, a fin de crear una especie de anillo, generando un mecanismo de transporte continuo para mover materiales de un punto a otro, con el mínimo esfuerzo humano. Las bandas transportadoras son comúnmente usadas en industrias que requieren movilizar materiales de peso considerable a grandes distancias en el orden de Kilómetros, pero también suelen ser usadas en sistemas autónomos para transportar materiales sometidos a análisis o transformaciones, movilizándolos en cortas distancias en el orden de centímetros o unos pocos metros. Este tipo de mecanismo se complementa con el uso de motores de distinto tipo en concordancia con su aplicación, fuerza y velocidad requeridas (Alvarado, 2016; Huamán, 2014).

1.1.2 Estudios previos

De acuerdo con Guanjun et al. (2019), Aragua y Mabayo (2018), Nasir et al (2018), Alikhanov (2019) y Quillo (2018), se plantean distintas clasificaciones, según aspectos puntuales, de huevos de gallina, como huevos quebrados, estimaciones de forma y estimaciones de peso a través de técnicas de procesamiento de imágenes planteando distintos programas en función de los requerimientos establecidos. Algunos de los programas base mostrados en los artículos citados son: Matlab y LabVIEW, siendo estos, softwares de pago, cuyo costo de licencia es relativamente alto para mantenerlo en constante ejecución.

Guanjun et al. (2019) toma como referencia dentro del procesamiento de imágenes un operador o filtro LOG negativo para el alcance de sus objetivos. Aragua y Mabayo (2018), Nasir et al (2018) y Alikhanov (2019), analizan el tamaño de un huevo de gallina, determinando el volumen del objeto de estudio de acuerdo a las imágenes que se obtienen, factor de forma, índice de forma, la distancia del sistema de adquisición de imágenes y cálculos matemáticos con fórmulas obtenidas de estudios que anteceden. Quillo (2018), plantea un complemento del software EGGSOTic, previamente desarrollado, con actuadores para dinamizar la clasificación, obteniendo un tiempo de operación de 2,52 segundos por unidad.

Fernández-S et al. (2019), presentan un estudio de fertilidad de los huevos de gallina, determinando si un huevo es apto o no para incubación, desarrollando un sistema de desplazamiento basado en impresoras 3D para mover una cámara, sobre un tipo de cubeta perforada para el análisis colectivo de huevos de gallina dentro de un espacio cerrado que evita filtraciones de luz externa, iluminado por un panel de LEDs de luz blanca de alto brillo que actúa como ovoscopio. Este sistema emplea un mecanismo compuesto por elementos de hardware libre y un procesamiento de imágenes basado en software libre, haciendo énfasis en el uso de OpenCV con programación en Python y un tiempo de operación de 1,75 segundos por huevo de gallina.

Wang et al. (2020) analiza una característica en particular, que es la porosidad de la cáscara de los huevos de gallina, como parámetro de determinación de calidad de los mismos. Se adquiere la imagen de un huevo de gallina, se aplican filtros durante su procesamiento y, basándose en el algoritmo K-means programado en Matlab, se determina la característica indicada.

Varios proyectos se han planteado, haciendo uso de redes neuronales artificiales (ANN). Mota-Grajales et al. (2019) por ejemplo, analiza el factor de forma de los huevos de gallina, haciendo un análisis individual mediante un láser que se refleja sobre su superficie, adquiriendo las imágenes requeridas para el análisis y aplicando un método de ANN, que estima imperfecciones reflejadas sobre la curvatura del huevo de gallina y empleando el software Matlab. Fadchar y Dela Cruz (2020) procesan datos mediante una ANN, empleando también el software Matlab, para determinar la fertilidad de los huevos de gallina previo a su incubación, adquiriendo imágenes individuales de huevos de gallina, comparándolas con imágenes previamente procesadas. Okinda et al. (2020), estima el volumen de los huevos de gallina de forma colectiva adquiriendo imágenes a través de un Kinect con sensor depth. Estas imágenes se procesan en Matlab mediante ANN y se afirma que el sistema puede adaptarse a otros clasificadores. Nematina y Abdanan (2018), aplican ANN para evaluar las propiedades físico químicas de huevos de gallina, prediciendo su frescura mediante estimaciones de la unidad Haugh y pH, de forma individual. Este software ha sido desarrollado con programación en Matlab. Además, los procesos tuvieron fines destructivos de los productos para la verificación del funcionamiento del sistema.

Muchos de los estudios presentan la aplicación de redes neuronales convolucionales (CNN). Botta et al. (2022), realiza una detección de grietas en la cáscara de huevos de gallina mediante el uso de CNN, entrenando la red planteada con fragmentos de imágenes donde se presentaban distintos tipos de fisuras o grietas a fin de enfocarse únicamente en esa característica, realizando una comparación individual con los objetos de estudio al momento de analizarlos. En un entorno natural, se realiza un seguimiento o detección objetos como lo muestra Lubich et al. (2019),

basándose en la adquisición de imágenes generadas en lapsos de tiempo, mismas que son tratadas con un detector de disparo único (SSD) para la clasificación de múltiples objetos con una única toma, logrando identificar varias razas o tipos de huevos y pollos, reflejando estos datos en un entorno IoT y procesándolos a través de CNN, proyectando a que el proceso se realice en tiempo real. Nasiri et al. (2020) realiza una clasificación de huevos de gallina sin lavar para determinar tres tipos: intacto, ensangrentado y roto, mediante una CNN programada en Python complementándose con bibliotecas como Keras y TensorFlow. Valencia et al. (2021) estudia el nivel de suciedad y grietas en los huevos de gallina, analizando el exterior de los mismos en primera instancia para evaluar las características superficiales y posteriormente las características internas aplicando ovoscopía para adquirir imágenes que son procesadas mediante una CNN desarrollada bajo lenguaje de programación Python haciendo uso de la biblioteca OpenCV.

Ghaderi et al. (2018), plantearon un análisis para identificar embriones muertos y huevos infértiles durante la incubación, desarrollando una placa electrónica que consta de un microcontrolador, un componente de medición dieléctrica, un sensor capacitivo y su respectiva interfaz para comunicación con una computadora. Este análisis extrae información midiendo la capacitancia sobre el huevo de gallina, almacena datos en una hoja de cálculo de Excel y se procesan en el sistema WEKA, considerando dos métodos de clasificación, uno mediante ANN y otro conocido como máquina de vectores de soporte (SVM). Los datos fueron extraídos en el día 5 y 18 del proceso de incubación. Cruz-Tirado et al. (2021) realizaron un análisis empleando un espectrómetro NIR comercial, procesando los datos adquiridos mediante una máquina de vectores de soporte (SVM) a través de algoritmos desarrollados en Matlab para la predicción de la unidad Haugh y, en base a esto, se obtuvo una clasificación de huevos según su frescura.

Turkoglu (2020), que clasifica huevos de gallina por su nivel de suciedad, rotos y ensangrentados. Dicho estudio aplica un modelo BiLSTM y una CNN, programados en Matlab. Este sistema cuenta con una bandeja que integra un sistema de rotación constante a fin de procesar toda la superficie del huevo, con un clasificador mecánico para separar los huevos defectuosos, empleando una tarjeta Arduino. Dong et al. (2021) analiza la fertilidad de huevos de pato durante su proceso de incubación, entre el día 1 y 7, aplicando un sistema de ovoscopía para adquisición y procesamiento de imágenes a través de métodos de análisis discriminante lineal (LDA), clasificador Naive Bayes (NB) y SVM, considerando biblioteca OpenCV.

1.1.3 Estudios referentes presentados en la Universidad Tecnológica Israel

Dentro del repositorio de la institución no se cuenta con estudios sobre ovoscopía o procesamiento de imágenes para análisis de huevos de gallina, pero si existen varios sistemas de

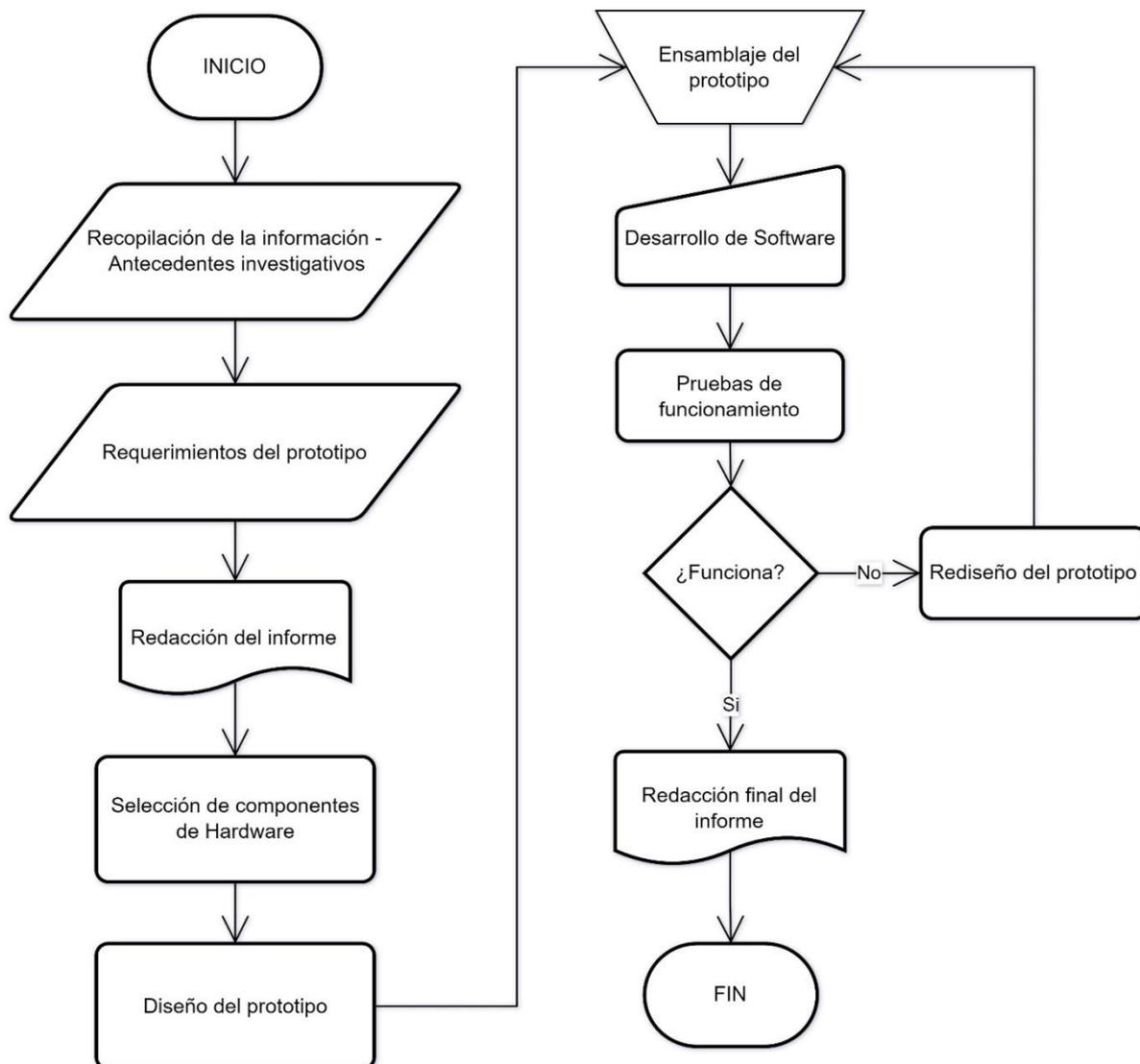
visión artificial con otros enfoques como es el caso de Chinchero (2019) y Baiza (2020), quienes emplean algoritmos desarrollados en Python, incluyendo la biblioteca OpenCV dentro de una Raspberry Pi ayudada por una cámara digital para adquirir imágenes; el primer autor en mención plantea extraer texto de imágenes y a su vez traducir este texto en audio para personas no videntes y el segundo detecta somnolencia en conductores a través de reconocimiento facial, analizando rasgos característicos en los ojos.

En la mayoría de sistemas planteados previamente, se analizan huevos de gallina de forma individual y estática, observando una sola característica, además del uso de software de pago, por lo que su aplicación a gran escala se ve limitada. Existen estudios donde se combina varios métodos o algoritmos a fin de aumentar la precisión y eficiencia de los análisis, considerando también que la velocidad y precisión de los sistemas de visión artificial son superiores a los métodos tradicionales que empleamos los seres humanos. Por lo antes expuesto se precisó un sistema dinámico que permite el análisis continuo de huevos de gallina desplazados por una banda transportadora, donde se hace énfasis en el uso de hardware y software libre para el procesamiento de imágenes que integra un clasificador de, no una sino, varias características o problemas que presente la cáscara de los huevos.

1.2. Proceso investigativo metodológico

En el presente proyecto de investigación se ha empleado un tipo de investigación bibliográfica y experimental, donde, a partir de trabajos previamente desarrollados, se ha planteado y desarrollado un prototipo obtenido mediante ensayos de prueba y error. A fin de cumplir el proceso sistemático para el desarrollo del presente proyecto, se ha seguido el diagrama de flujo presentado en la Figura 1.

Figura 1.
 Diagrama de flujo de las actividades desarrolladas en el proyecto.



Nota. Elaboración propia.

En términos generales el prototipo integra un sistema electrónico de control para sincronizar cada uno de los elementos o dispositivos, un sistema informático para procesamiento de imágenes y un sistema mecánico basado en una banda transportadora con diseño adaptado a los requerimientos del sistema.

Para la recolección de la información, el proyecto se ha basado netamente en una investigación bibliográfica, tomando como referencia artículos científicos y proyectos de titulación con un enfoque similar donde se emplean distintas técnicas de procesamiento de imágenes u otros mecanismos de análisis y a su vez distintas estructuras e interfaces.

Debido a la naturaleza del proyecto, puesto que no se está lanzando actualmente un producto al mercado o comparando características de distintos productos, por ser un prototipo, no se ha realizado

encuestas a una población determinada, es decir no se requiere población y muestra. En función de lo antes mencionado y basado en la investigación experimental se ha mantenido una metodología de investigación cualitativa para validación del sistema.

CAPÍTULO II: PROPUESTA

2.1. Fundamentos teóricos aplicados

El presente proyecto se centra en un prototipo desarrollado para el análisis de características superficiales de los huevos de gallina, orientado a la identificación de problemas en la producción de los mismos de acuerdo al enfoque del usuario. Este prototipo integra una banda transportadora adaptada que conduce un tipo de cubeta específica al punto de análisis, colocando los huevos de gallina sobre el sistema de iluminación u ovoscopio y bajo la cámara digital encargada de adquirir y transmitir la información de los huevos a trasluz, misma que es procesada por la Raspberry Pi en concordancia con los sensores integrados en el mecanismo y cuyas señales son receptadas por la misma placa y el microcontrolador Atmel dispuestos para el control y automatización del proceso.

Para el diseño y elaboración del prototipo se ha empleado distintos elementos electrónicos interconectados, con características de conocimiento general dentro del ámbito de automatización y programación, para la vinculación de los tres sistemas principales: mecánico, electrónico e informático.

2.1.1. Sensores

Los sensores son los dispositivos electrónicos encargados de adquirir señales analógicas externas, a fin de continuar con el proceso de digitalización de las mismas dentro de la microcomputadora y el microcontrolador empleados. Dentro del proyecto se han empleado básicamente 2 tipos de sensores.

El primer tipo de sensor es de imagen, encargado de adquirir las señales lumínicas, mismo que se incluye en la cámara digital donde se realiza todo el proceso de conversión. El siguiente es un sensor infrarrojo de proximidad o interruptor fotoeléctrico, concretamente el dispositivo FS80NK, que, al ser digital, genera estados lógicos de 1 o 0, fácilmente leídos por los dispositivos de control; se emplea para determinar la presencia de objetos en la posición establecida donde el 1 lógico indica que no se ha detectado nada, mientras que el 0 lógico indica lo opuesto. Este sensor infrarrojo posee un potenciómetro que permite regular la distancia de detección desde 1 cm hasta 80 cm.

El sensor de imagen transmite la información digitalizada, en concordancia con el circuito integrado en la cámara, mediante comunicación serial a través de una conexión USB en la microcomputadora. Los sensores infrarrojos emiten los pulsos o información de las señales a los pines digitales que se configuran como entradas en el microcontrolador.

2.1.2. Actuadores

Los actuadores son los mecanismos de acción, generalmente mecánicos, que, referente a este sistema, se encargan de generar la fuerza para mover la banda transportadora, siendo el caso concreto de los motores paso a paso, controlados de forma secundaria por un driver o controlador físico que se encarga de recibir las señales lógicas enviadas a través de los pines de propósito general del microcontrolador, mismos que se establecen como salidas, y, además, se encarga de transformar estas señales lógicas en señales electrónicas de potencia para activar los motores, realizando el cambio de giro si fuera necesario y regulando su velocidad.

2.1.3. Pines de propósito general

Los pines de propósito general o GPIO tanto en la microcomputadora como en el microcontrolador son los periféricos que generan o reciben las señales electrónicas, ya sean analógicas o digitales. Estos pines pueden ser configurados como entradas o salidas. En el caso concreto de Raspberry Pi se cuenta únicamente con pines GPIO digitales que trabajan con un voltaje nominal de 3.3 V y en el caso de los microcontroladores Atmel, cuentan con pines GPIO digitales que pueden trabajar con un voltaje de 3.3 V o 5 V, adicionalmente cuentan con pines que solo pueden ser configurados como entrada analógica, para realizar una conversión analógica digital, más no viceversa.

2.1.4. Bibliotecas o librerías para programación en Python

Las bibliotecas o librerías para programación en Python son un conjunto de clases, métodos o fragmentos de código empaquetados en uno o varios scripts o archivos que deben ser descargados e instalados previamente, asociándolos con el lenguaje empleado y su versión. Existen varias bibliotecas que se instalan por defecto junto con el compilador con el que se trabaje para desarrollar el código, pero se pueden incluir bibliotecas adicionales externas a fin de optimizar los nuevos programas que se desarrollen, reduciendo notablemente las líneas de programación.

Una de las bibliotecas empleadas en el presente proyecto es OpenCV, misma que no se incluye en los compiladores, pero es ampliamente considerada en sistemas de visión artificial sobre todo basados en software libre. Esta biblioteca cuenta con varios métodos o funciones para almacenar imágenes, grabar videos, conversiones de color, añadir filtros, entre otros.

Qt5 es otra de las bibliotecas incluidas en el código, pero que tampoco se instala por defecto. Esta biblioteca contiene elementos predeterminados, como botones, etiquetas, cuadros de texto, ventanas, entre otros, fundamentales para el desarrollo de la interfaz gráfica.

Numpy es una biblioteca que suele instalarse por defecto, pero en ocasiones es necesario actualizarla para que sea compatible con otras bibliotecas; se emplea para el manejo de vectores incluidos en la segmentación de imágenes para su procesamiento y visualización en la interfaz de acuerdo a la ubicación de los objetos de estudio.

Para el manejo de bases de datos se emplea la biblioteca pymysql, misma que, a través de sus métodos puede crear o eliminar tablas, añadir datos a tablas existentes, editar datos, entre otras opciones basadas en la gestión de MySQL.

Sys es una de las bibliotecas que incluyen los compiladores y maneja funciones generales en los programas como las opciones de cerrar ventana, minimizar, maximizar, etc. Time y Datetime también se incluyen por defecto; estas bibliotecas controlan los tiempos de espera entre ejecuciones de código y pueden sincronizar la fecha y hora del sistema operativo respectivamente. Finalmente se emplea la biblioteca RPiGPIO, misma que sólo funciona en los compiladores de Raspberry Pi donde naturalmente se incluye para controlar los pines GPIO de la placa.

2.1.5. Filtros de imagen

Los filtros de imagen son métodos, funciones o códigos preestablecidos dentro de las bibliotecas que sirven generalmente para difuminar las imágenes, eliminar el ruido de las mismas o integrar efectos según los requerimientos del procesamiento. En caso de no usar bibliotecas, estos filtros pueden ser programados partiendo de funciones matemáticas, pero esto extendería el código. En el sistema se ha manejado el filtro gaussiano que, ante el filtro de media también probado, reduce mejor el ruido de las imágenes sin perder tanto detalle al aplicarlo; estos filtros solo se pueden usar posterior a una conversión de imagen a escala de grises. Adicional a esto, se ha aplicado otro filtro en función de umbrales para establecer características y métricas de comparación, mediante el algoritmo Canny.

2.1.6. Servidores

Los servidores son sistemas informáticos para interconexión o segmentación de redes, almacenamiento de datos y generalmente la atención de peticiones de clientes o dispositivos vinculados a un determinado sistema. En el proyecto se ha empleado un servidor LAMP basado en Linux, servidores web, MySQL y lenguaje PHP para el manejo de la base de datos localmente.

2.1.7. Interfaz gráfica

La interfaz gráfica es la parte visual que ejecuta el programa a fin de mostrar texto, imágenes, video, resultados y demás información fácilmente interpretable por el usuario. Una interfaz puede

diseñarse para, únicamente, monitorear o visualizar información o para interactuar con el usuario, permitiendo la inserción de información externa de forma manual en caso de ser requerida.

2.2. Descripción de la propuesta

El prototipo de sistema automático de ovoscopia, mediante técnicas de procesamiento digital de imágenes, plantea un mecanismo que emplea hardware y software libre a fin de brindar al usuario bajo costo, es decir, fácil acceso al mismo. Este mecanismo ha sido desarrollado a fin de tener versatilidad en los análisis de la industria avícola, concretamente en los análisis de huevos de gallina para sus distintos usos.

Mediante técnicas de procesamiento digital de imágenes, empleando un ovoscopio de mediana escala, el dispositivo es capaz de analizar las características físicas de la cáscara de los huevos de gallina, proporcionando información sobre sus características como, por ejemplo, cáscaras quebradas, cáscaras con exceso de poros, entre otras.

Los resultados obtenidos permiten evaluar no sólo la calidad de los huevos de gallina sino también el trasfondo de la misma, ya que puede verse afectada por una mala manipulación del personal encargado dentro o fuera de las granjas, problemas en la alimentación de las ponedoras, almacenamiento del producto, entre otros factores determinantes para las empresas de acuerdo a su enfoque o requerimientos en el mercado de interés.

a. Estructura general

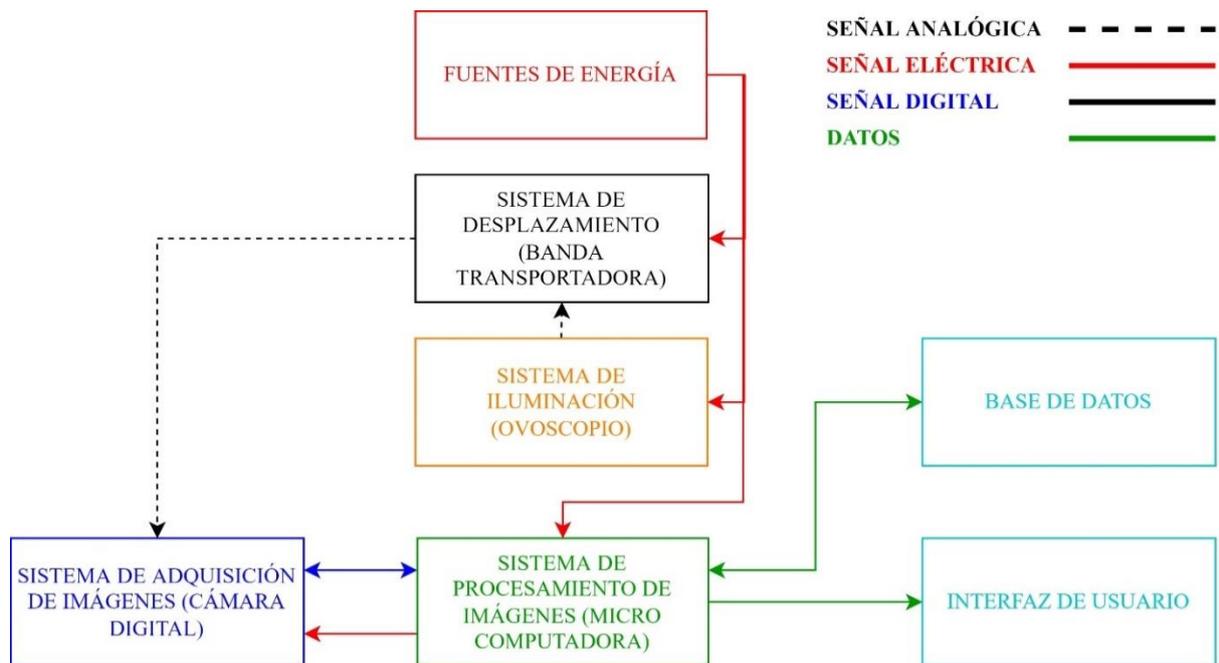
Para cumplir con los objetivos de análisis se ha establecido las siguientes especificaciones técnicas o requerimientos, separándolos en subsistemas interconectados:

- Un sistema de desplazamiento para dinamizar el proceso.
- Un sistema de iluminación para disponer los objetos de estudio a trasluz.
- Un sistema de adquisición de imágenes para recopilar la información.
- Un sistema de procesamiento de imágenes para la segmentación de la información.
- Una interfaz para la visualización de resultados.
- Una base de datos para almacenar los resultados y futuros análisis.

En la Figura 2 se presenta un diagrama de interconexión que permite entender de mejor manera la relación o comportamiento entre subsistemas.

Figura 2.

Diagrama de bloques de interconexión de subsistemas del sistema automático de ovoscopia.



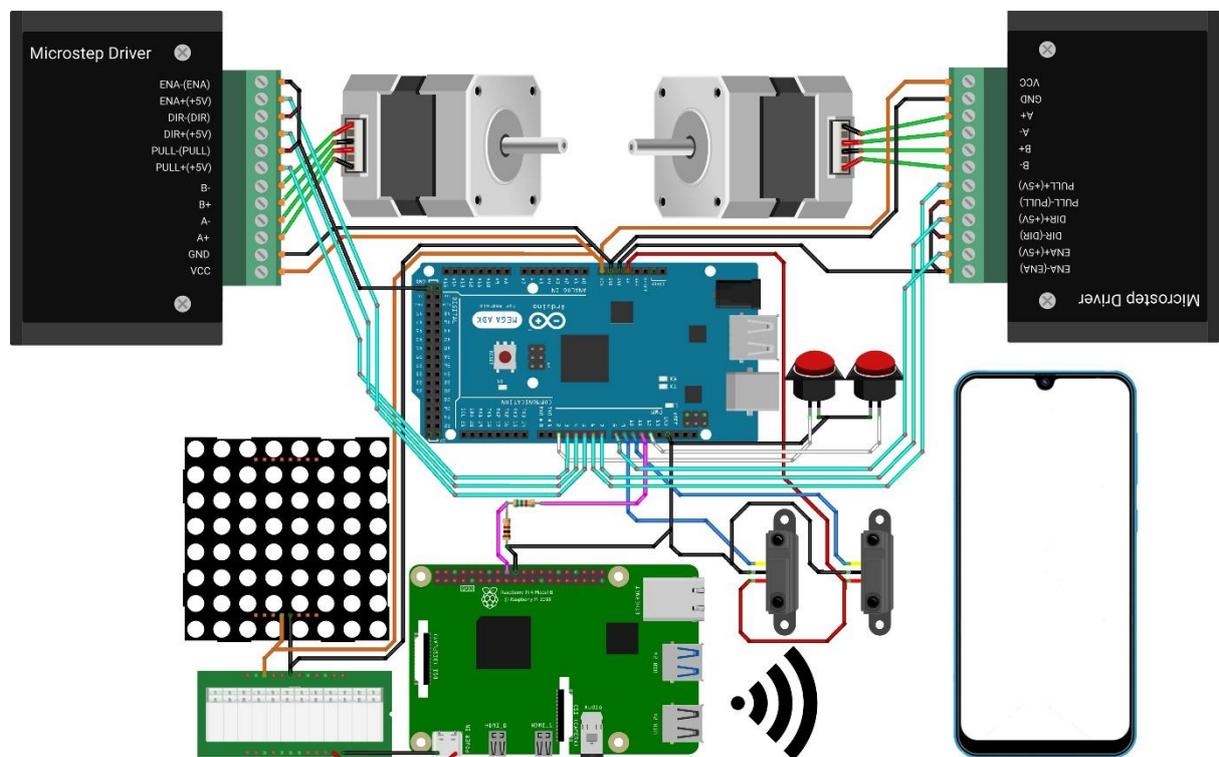
Nota. Elaboración propia.

b. Explicación del aporte

En la Figura 3 se presenta la interconexión de los dispositivos que constituyen el sistema.

Figura 3.

Interconexión de dispositivos del sistema automático de ovoscopia.



Nota. Elaboración propia.

En las tablas y análisis presentados a continuación se especifican las características de los componentes principales que constituyen el prototipo, comparándolos con dispositivos similares que han permitido determinar la factibilidad para el empleo de los mismos.

Microcomputadora

En la Tabla 1 se muestra un grupo de microcomputadoras con características similares, mismas que se han establecido en función de los requerimientos del sistema.

Tabla 1.

Especificaciones técnicas de microcomputadoras para procesamiento de imágenes e interconexión de periféricos.

DISPOSITIVO	RASPBERRY PI	ASUS TINKER BOARD	BANANA PI	PINE ROCK
DETALLE				
Modelo	4 B	2S	BPI-M5	Rock 64
Dimensiones	56 mm * 85 mm aprox.	56 mm * 85 mm aprox.	60 mm x 92 mm aprox.	56 mm * 85 mm aprox.
SoC	Broadcom BCM2711	Rockchip RK3399	Amlogic S905X3	Rockchip RK3328
Procesador	ARM Cortex-A72 de 4 núcleos de 64 bits	ARM Cortex-A72 de 2 núcleos; ARM Cortex-A53 de 4 núcleos	ARM Cortex-A55 de 4 núcleos	ARM Cortex-A53 de 4 núcleos
Velocidad de Reloj	4 núcleos a 1,5 GHz	2 núcleos a 2 GHz; 4 núcleos a 1,5 GHz	4 núcleos a 2 GHz	4 núcleos a 1,5 GHz
Memoria RAM	Hasta 8 GB LPDDR4	Doble canal 2 GB / 4 GB LPDDR4	4 GB LPDDR4	Hasta 4 GB LPDDR3
Memoria de Almacenamiento	No posee memoria integrada; puerto Micro SD	16 GB eMMC; puerto para Micro SD	16 GB eMMC; puerto Micro SD	Puerto eMMC; puerto Micro SD
Pines GPIO	Cabezal de 40 pines digitales	Cabezal de 40 pines digitales	Cabezal de 40 pines digitales	1 cabezal de 40 pines y 1 cabezal de 22 pines
Comunicación Red Alámbrica	Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet

Comunicación	Wi-Fi a 2,4 GHz y	Wi-Fi a 2,4 GHz	No integrada	No integrada
Red Inalámbrica	5GHz estándar IEEE 802.11 b/g/n/ac; Bluetooth 5.0 BLE	estándar IEEE 802.11 a/b/g/n/ac; Bluetooth 5.0		
GPU	Broadcom VideoCore VI	ARM Mali-T860 MP4	ARM Mali-G31	ARM Mali 450 MP2
Puertos	2 puertos USB 3.0 y 2 puertos USB 2.0	3 puertos USB 3.2 generación 1 tipo A y 1 puerto USB 3.2 generación 1 tipo C	4 puertos USB 3.0	2 puertos USB 2.0 y 1 puerto USB 3.0
Audio	Salida estéreo de 4 polos y puerto de video compuesto; HDMI; Jack 3.5 mm	HDMI; cabezal GPIO	HDMI; Jack 3.5 mm	HDMI; Jack 3.5 mm
Video	2 puertos micro HDMI (hasta 4Kp60); puerto para pantalla táctil MIPI DSI; puerto para cámara MIPI CSI	1 puerto HDMI-CEC; USB-C Alt 2.0; puerto para pantalla táctil MIPI DSI; puerto para cámara MIPI CSI-2	1 puerto HDMI 2.0 (hasta 4K@60Hz con HDR, CEC, EDID)	1 puerto HDMI (4K)
Sistemas Operativos Soportados	Distribuciones Linux (Raspberry Pi OS - oficial); Windows 10 IoT Core	Debian 9; Android 10	Distribuciones Linux; Android	Distribuciones Linux; Android; Yocto
Temperatura de funcionamiento	0 °C – 50 °C	0 °C – 60 °C	No especificado	No especificado

Fuente de energía requerida	5 V - 3 A, conector USB-C; cabezal GPIO; PoE	12 V a 19 V - 45 W, conector Jack (5.5/2.5 mm)	5 V - 3 A, conector USB-C	5 V - 3 A, Jack (3.5mm/1.35mm); PoE
Disponibilidad en el mercado local	Si	No	No	No
Costo promedio	\$150 (4GB)	\$120 (4GB)	\$90 (4 GB)	\$140 (4 GB)

Nota. Elaboración propia adaptado de Raspberry Pi (2021), ASUSTeK Computer Inc. (s.f.), Banana Pi team (2022) y Pine64 (2022).

En base a la información presentada, se seleccionó la Raspberry Pi modelo 4 B debido a su potente procesador, capacidad mayor de memoria RAM, versatilidad para la instalación de sistemas operativos, opciones extendidas de video y procesamiento de imágenes, facilidad para la interconexión de periféricos y sobre todo el fácil acceso a esta microcomputadora o disponibilidad en el mercado local, siendo un dispositivo conocido con gran cantidad de información de respaldo.

Cámara

En la Tabla 2 se analizan distintas cámaras, puesto que son una pieza fundamental para el sistema automatizado, encargadas de adquirir las imágenes a procesar por la microcomputadora seleccionada previamente.

Tabla 2.

Especificaciones técnicas de cámaras digitales.

DISPOSITIVO	CÁMARA	CÁMARA WEB	CÁMARA WEB	SMARTPHONE
DETALLE	RASPBERRY PI	GENIUS	FULL HD	
Modelo	V2.1	FaceCam 1000X	Genérica	Samsung Galaxy A50 – SM-A505G
Dimensiones	25 mm * 23 mm * 9 mm	22 mm * 60 mm *	60 mm x 92 mm	158,5 × 74,7 × 7,7 mm.
Peso	3 g	50 g	No especificado	No especificado
Longitud de cable	0,15 m	1,5 m	1,5 m	1,5 m
Sensor imagen	de CMOS Sony IMX219PQ - 8 Megapíxeles	CMOS 0,92 Megapíxeles	CMOS 2,07 Megapíxeles	Triple cámara 25.0 Megapíxeles + 5.0 Megapíxeles + 8.0 Megapíxeles

Resolución	1080p30, 720p60 y VGA90	720p30	1080p25, 720p25 y VGA25	1080p30
Ángulo de visión	62,2°	90°	No especificado	Apertura focal: F1.7 , F2.2 , F2.2
Micrófono integrado	No	Si	Si	Si
Tipo de conexión	MIPI CSI	USB	USB	Servidor
Disponibilidad en el mercado local	Si	Si	Si	Si
Costo promedio	\$45	\$20	\$20	\$180

Nota. Elaboración propia adaptado de Raspberry Pi (s.f.), Genius (s.f.) y Samsung (s.f.).

Por lo antes expuesto, la mejor opción habría sido la cámara propia de raspberry por sus especificaciones y mejor sensor de imagen, pero actualmente por las actualizaciones del sistema operativo Raspberry Pi OS, se han generado grandes problemas para emplear este dispositivo, generándose un sinnúmero de reportes referentes en la página oficial e incluso se encuentra en escasez dentro del mercado local. Por el fácil acceso, manipulación y resolución, puesto que las cámaras de los Smartphones tienen mejor calidad que las cámaras web u otros dispositivos, luego de un test previo, se empleó la cámara del Samsung Galaxy A50, que, entre otras especificaciones, se puede convertir en una cámara IP, creando un servidor con el uso de la aplicación IP Webcam, se facilita el acceso y la programación de la misma para adquirir las imágenes.

Microcontrolador

Por ser un dispositivo ampliamente utilizado a nivel global, de fácil acceso, y que, además, cumple con las condiciones de hardware y software libre con gran soporte e información, se ha empleado el ATmega2560 de Atmel, mismo que recibe las señales de los sensores infrarrojos de proximidad, así como de los botones para arranque y paro de emergencia; además, controla la velocidad y activación de los actuadores.

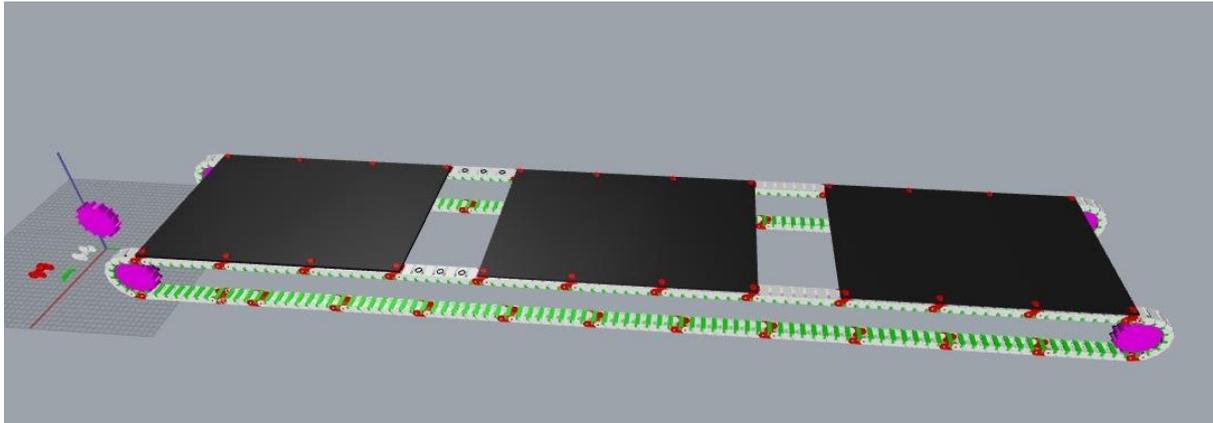
Banda transportadora

La banda transportadora ha sido desarrollada mediante impresión 3D empleando el material PLA para la constitución de su estructura. Esto, debido al particular diseño que se ha requerido, puesto que las bandas tradicionales son muy finas, lo que no permite albergar a los huevos de gallina de forma ordenada en una posición específica para su análisis y sobre todo que no se puede adaptar o cortar ya

que esto reduce su resistencia, por lo tanto, se ha requerido que la banda esté constituida por una especie de cadenas que hacen rotar longitudinalmente los ejes como se puede apreciar en la Figura 4, sobreponiendo una cubeta que cuenta con orificios donde se posicionan los huevos de gallina para que, en cuanto se llegue al punto de análisis, se pueda apreciar claramente la función del ovoscopio sobre todos los objetos de estudio segmentados en la estructura.

Figura 4.

Banda transportadora desarrollada para el sistema automático de ovoscopia.

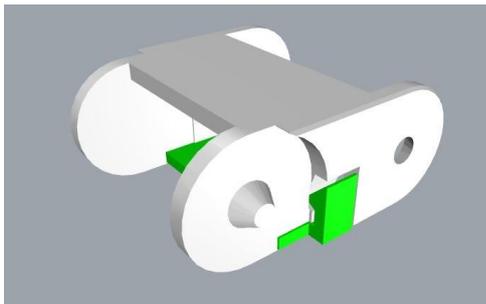


Nota. Elaboración propia.

En la Figura 5 se muestra un eslabón que constituye parte de la cadena, mismo que está compuesto por 2 piezas impresas para encajar en los ejes o engranajes que se sitúan en cada extremo de la estructura.

Figura 5.

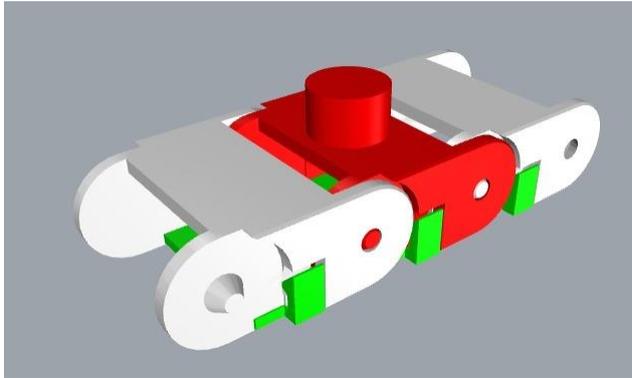
Eslabón que forma parte de las cadenas para el sistema automático de ovoscopia.



Nota. Elaboración propia.

En la Figura 6 se muestra un segmento de cadena enlazado, donde se puede apreciar también un eslabón especial que permite la sujeción de la cubeta.

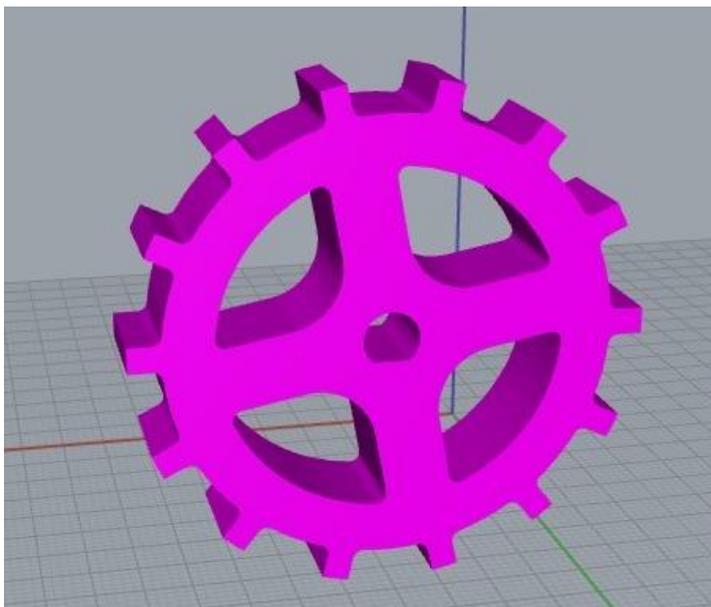
Figura 6.
Segmento de cadenas del sistema automático de ovoscopia.



Nota. Elaboración propia.

En las Figuras 7 y 8 se muestran los engranajes que se colocan en los motores y como soporte para retroalimentación de las cadenas respectivamente.

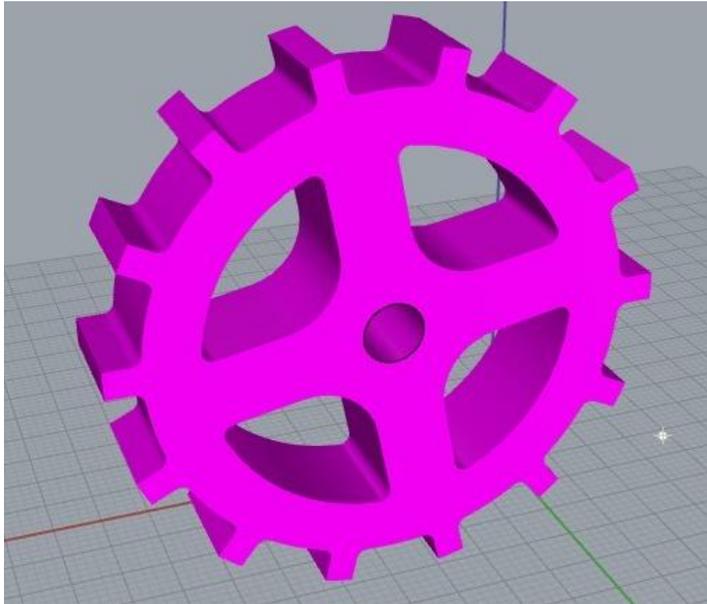
Figura 7.
Engranaje para adaptación a los motores del sistema automático de ovoscopia.



Nota. Elaboración propia.

Figura 8.

Engranaje para retroalimentación de cadenas del sistema automático de ovoscopia.

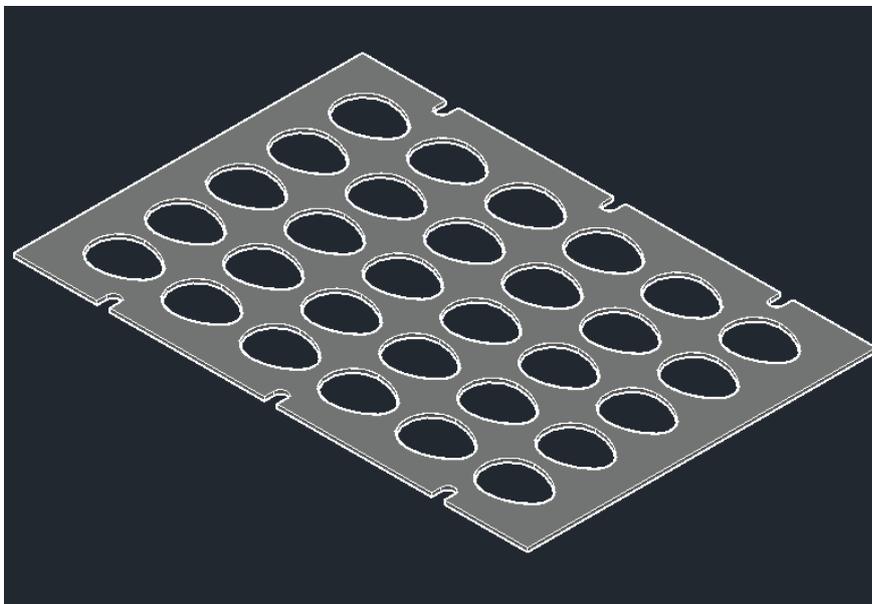


Nota. Elaboración propia.

Como complemento a la banda transportadora, se presenta la cubeta diseñada especialmente para este prototipo en la Figura 9.

Figura 9.

Cubeta del sistema automático de ovoscopia.



Nota. Elaboración propia.

Motores

Para desplazar la banda transportadora establecida como requerimiento del prototipo, se descarta previamente el uso de motores de corriente alterna o motores de carácter industrial, por su alto consumo de energía, costo elevado y por todos los implementos requeridos y manejo de pesos

relativamente bajos referente a los objetos de estudio (huevos de gallina), por lo cual, sólo se ha analizado motores de corriente continua, considerando contundentemente que los motores paso a paso tienen mayor precisión, alto torque y rotación continua, características fundamentales en el presente proyecto, son los mismos que se han seleccionado para el sistema.

Arbitrariamente se contempla el peso máximo de huevos grandes establecido por el INEN (2013) que es menor a 64 g por unidad. Para calcular un peso estimado se multiplica ese valor por 90 unidades, obteniendo un peso total de 5760 g, estableciendo a la par un máximo de 3 cubetas de 30 huevos de gallina cada una, colocados en la banda al mismo tiempo; una en la zona de carga, una en la zona de análisis y una en la zona de descarga. Por ende, se planteó el uso de 2 motores paso a paso Nema 23 con un torque de 12.2 Kg-cm que se ve reducido por las condiciones del mecanismo en la implementación real.

Sistema de iluminación

Para el sistema de iluminación, que cumple netamente con la función de ovoscopio a mediana escala por las características del prototipo, se emplea tecnología LED que actualmente es la más usada en relación a este apartado, “debido a su costo relativamente bajo con ventajas notables como mayor vida útil, bajo consumo de energía y baja emisión de calor” (Fernández y Jurado, 2018).

Sistema informático

Por ser un lenguaje de programación incluido por defecto en la microcomputadora seleccionada y cumpliendo con lo establecido de emplear software libre, el sistema de procesamiento de imágenes junto con el servidor e interfaz, se han programado en Python. Como parte del sistema informático, el microcontrolador ha sido programado bajo el propio IDE de la empresa Arduino, considerado también como software libre basado en el lenguaje C++.

c. Estrategias y/o técnicas

Uno de los puntos críticos en el procesamiento de imágenes, al emplear una microcomputadora que cuenta con pines GPIO para la interconexión de elementos electrónicos externos, es el paralelismo referente a la ejecución de los métodos de análisis. Los retardos que deben generarse, para la activación de las bobinas de los motores paso a paso, interrumpe el accionar de la cámara por lo que se ha aislado el control de los motores, usando de forma auxiliar un microcontrolador mismo que se encarga de este proceso a fin de que la microcomputadora se dedique únicamente al procesamiento de imágenes en tiempo real, pero con la sincronía de activación del proceso en base a las señales de los sensores, creando un circuito cerrado entre los distintos procesos que conjuga el sistema.

El sistema integra hardware y software libre para el procesamiento de imágenes a fin de abaratar costos como se ha mencionado previamente y, además, se ha empleado tecnología de impresión 3D para la implementación de la banda transportadora especialmente diseñada para el prototipo desarrollado.

2.3. Validación de la propuesta

Para la elección de especialistas se ha considerado un perfil acorde a los siguientes criterios: formación académica relacionada con el tema investigativo, experiencia académica y/o laboral orientada a la gestión administrativa y motivación para participar. En la Tabla 3 se presenta la información detallada de los actores seleccionados para la validación del “Sistema automático de ovoscopia para análisis de calidad de huevos de gallina mediante procesamiento digital de imágenes”.

Tabla 3.

Descripción de perfil de validadores.

NOMBRES Y APELLIDOS		AÑOS DE EXPERIENCIA	TITULACIÓN ACADÉMICA	CARGO
Ángel Oliverio	23	-	Ingeniero en Alimentos	Docente
Fernández Escobar		-	Magister en Procesamiento de Alimentos	
Martín Armando	23	-	Ingeniero Zootecnista	Docente
González Vélez		-	Doctorado en Recursos Naturales y Gestión Sostenible	
José Ezequiel Naranjo Robalino	4	-	Ingeniero Industrial en Procesos de Automatización	Docente
		-	Magister en Producción Y Operaciones Industriales	
Franklin Wilfrido	6	-	Ingeniero Técnico en Telecomunicaciones, Especialidad en Sistemas de Telecomunicaciones	Docente
Salazar Logroño		-	Master Universitario en Automática y Robótica	

Nota. Elaboración propia.

Los objetivos perseguidos mediante la validación fueron los siguientes:

- Validar la metodología de trabajo aplicada en el desarrollo de la investigación.

- Aprobar los resultados, conclusiones y recomendaciones obtenidas.
- Redefinir (si es necesario) el enfoque de los elementos desarrollados en la propuesta, considerando la experiencia de los especialistas.
- Constatar las posibilidades potenciales de aplicación del modelo de gestión propuesto.

Tabla 4.

Criterios de evaluación.

CRITERIOS	DESCRIPCIÓN
Impacto	Representa el alcance que tendrá el modelo de gestión y su representatividad en la generación de valor público.
Aplicabilidad	La capacidad de implementación del modelo considerando que los contenidos de la propuesta sean aplicables
Conceptualización	Los componentes de la propuesta tienen como base conceptos y teorías propias de la gestión por resultados de manera sistémica y articulada.
Actualidad	Los contenidos de la propuesta consideran los procedimientos actuales y los cambios científicos y tecnológicos que se producen en la nueva gestión pública.
Calidad Técnica	Miden los atributos cualitativos del contenido de la propuesta.
Factibilidad	Nivel de utilización del modelo propuesto por parte de la Entidad.
Pertinencia	Los contenidos de la propuesta son conducentes, concernientes y convenientes para solucionar el problema planteado.

Nota. Elaborada por: Ing. Wilmer Fabián Albarracín Guarochico MBA.

En las Tablas 5, 6, 7 y 8, se presentan resultados, de los criterios de evaluación, consignados por cada uno de los evaluadores que figuran en la Tabla 3.

Tabla 5.

Escala de evaluación otorgada por el M.Sc. Ángel Fernández E.

CRITERIOS	EVALUACIÓN SEGUN IMPORTANCIA Y REPRESENTATIVIDAD				
	En Total Desacuerdo	En Desacuerdo	Ni de Acuerdo Ni en Desacuerdo	De Acuerdo	Totalmente Acuerdo
Impacto					X
Aplicabilidad					X
Conceptualización				X	

Actualidad		X
Calidad Técnica		X
Factibilidad	X	
Pertinencia		X

Nota. Elaboración propia.

Tabla 6.

Escala de evaluación otorgada por el PhD. Martín González V.

EVALUACIÓN SEGUN IMPORTANCIA Y REPRESENTATIVIDAD					
CRITERIOS	En Total Desacuerdo	En Desacuerdo	Ni de		Totalmente Acuerdo
			Acuerdo Ni en Desacuerdo	De Acuerdo	
Impacto					X
Aplicabilidad					X
Conceptualización				X	
Actualidad					X
Calidad Técnica					X
Factibilidad				X	
Pertinencia					X

Nota. Elaboración propia.

Tabla 7.

Escala de evaluación otorgada por el Mg. José Naranjo R.

EVALUACIÓN SEGUN IMPORTANCIA Y REPRESENTATIVIDAD					
CRITERIOS	En Total Desacuerdo	En Desacuerdo	Ni de		Totalmente Acuerdo
			Acuerdo Ni en Desacuerdo	De Acuerdo	
Impacto				X	
Aplicabilidad					X
Conceptualización				X	
Actualidad					X
Calidad Técnica				X	
Factibilidad				X	

Pertinencia	X
--------------------	---

Nota. Elaboración propia.

Tabla 8.

Escala de evaluación otorgada por el Mg. Franklin Salazar L.

EVALUACIÓN SEGUN IMPORTANCIA Y REPRESENTATIVIDAD					
CRITERIOS	En Total		Ni de		Totalmente
	Desacuerdo	En	Acuerdo Ni	De Acuerdo	
		Desacuerdo	en		Acuerdo
			Desacuerdo		
Impacto					X
Aplicabilidad					X
Conceptualización				X	
Actualidad					X
Calidad Técnica				X	
Factibilidad					X
Pertinencia					X

Nota. Elaboración propia.

2.4. Matriz de articulación de la propuesta

Posterior a la implementación del prototipo, se presenta un resumen de los elementos y métodos principales en la Tabla 9.

Tabla 9.

Matriz de articulación.

EJES O PARTES PRINCIPALES DEL PROYECTO	BREVE DESCRIPCIÓN DE LOS RESULTADOS DE CADA PARTE	SUSTENTO TEÓRICO QUE SE APLICÓ EN LA CONSTRUCCIÓN DEL PROYECTO	METODOLOGÍAS, HERRAMIENTAS TÉCNICAS Y TECNOLOGICAS QUE SE EMPLEARON
1 Determinación de elementos de hardware y software.	1.1 Identificación de problemas presentados en investigaciones previas en función del sustento bibliográfico. 1.2 Selección de componentes idóneos de acuerdo al análisis de las especificaciones técnicas de los componentes.	Hardware y Software Libre.	Factibilidad técnica mediante la identificación de elementos de hardware disponibles en el mercado local y acceso a los complementos de software requeridos.
2 Diseño e innovación del prototipo de acuerdo a los requerimientos.	2.1 Sistema de control electrónico basado en un microcontrolador Atmel que centraliza la adquisición de datos de los sensores y acciona los motores. 2.2 Sistema mecánico basado en una banda transportadora de cadena que consta de 2 motores paso a paso controlados por el	Microcontroladores. Mecanismos de impresión 3D. Desarrollo de algoritmos en lenguaje C++ y Python.	Modelado del prototipo en 2 y 3 dimensiones basado en sistemas CAD.

sistema previamente mencionado para su desplazamiento.

2.3 Sistema informático encargado de adquirir imágenes que son capturadas en concordancia con las señales del sistema de control, obteniendo un sistema de visión artificial que procesa automáticamente dichas imágenes y genera resultados visuales a través de una interfaz, almacenando la información relevante en una base de datos.

<p>3 Implementación y pruebas de funcionamiento del prototipo.</p>	<p>3.1 Funcionamiento independiente de cada elemento.</p> <p>3.2 Acoplamiento o ensamblaje de cada uno de los 3 sistemas previamente mencionados con funcionalidad independiente.</p> <p>3.3 Integración de los 3 sistemas en un solo sistema conjunto.</p>	<p>Importación de algoritmos externos.</p> <p>Aislamiento de etapas de potencia y control electrónico.</p> <p>Manejo de pines de propósito general GPIO.</p> <p>Configuración de motores paso a paso.</p>	<p>Complemento de los algoritmos a través de bibliotecas o librerías preexistentes.</p> <p>Interconexión de los elementos electrónicos y regulación de energía para alimentación de los mismos de acuerdo a las fichas técnicas u hojas de datos, manteniéndose dentro de los rangos permitidos.</p>
---	---	---	--

Nota. Elaboración propia.

2.5. Análisis de Resultados. Presentación y discusión.

De acuerdo a las pruebas de resistencia, evitando que las cadenas formadas por eslabones en cada extremo de la banda no se tensionen tanto al punto de romperse o simplemente separarse de su elemento sucesivo, por la presión generada, se ha dispuesto un conjunto ideal para el prototipo de 210 eslabones para formar cada una de las cadenas que constituyen la banda, teniendo un total de 420 eslabones. Cada eslabón consta de 2 piezas para encajar en los ejes o engranajes; 2 ejes van anclados, 1 en cada motor en el primer extremo de la estructura y 2 ejes más permiten la retroalimentación o retorno de las cadenas ubicados en el otro extremo de la estructura como se pudo apreciar en la Figura 4. Adicional a esto, se ha impreso 2 piezas como soporte para los motores. En resumen, se ha elaborado un total de 846 piezas en material PLA, empleando una impresora 3D.

La velocidad angular o RPM de los motores son controladas por la frecuencia de los pulsos que se generan en el microcontrolador, mientras que el torque depende de la corriente que suministran los drivers o controladores de potencia, que no debe sobrepasar la corriente nominal indicada por los fabricantes. Para la activación de los motores se ha configurado pasos completos a fin de tener el mayor torque posible como se especifica en las hojas de datos de los motores. Una revolución por motor, es decir 200 pasos, equivale a la distancia o recorrido de 14 eslabones de la banda, por lo cual, para dar una vuelta completa a la estructura, los motores deben dar 3000 pasos.

OpenCV establece un valor por defecto de 640x480 píxeles con relación de aspecto 1,33:1, para procesamiento de imágenes, pero estos valores pueden ser editados en concordancia con la cámara que se emplee. Después de varias pruebas realizadas con la cámara seleccionada se determinó que la mejor relación de aspecto para realizar las tomas, es 16:9 con una resolución de 1920x1080 píxeles a fin de que la pérdida de detalles sea mínima y no se sature el servidor. La cámara empleada en el prototipo tiene una resolución máxima para adquisición de imágenes, de 3840x2160 píxeles, pero mientras mayor tamaño presenten las imágenes, el procesamiento tarda aún más y los parámetros de codificación se tornan más complejos al momento de estandarizarlos. Para evitar la inserción de información innecesaria, como la estructura o reflejos en las imágenes por los bordes adicionales que sobrepasan la cubeta sobre la cual se coloca el sensor de imagen, se realiza un corte en cada captura obteniendo una imagen resultante de 990x1325 píxeles.

El proceso de clasificación de características en la cáscara los huevos analizados, inicia con la toma de una imagen generando a la par una matriz de 3x5 en cada captura, lo que representa 15 huevos de gallina por toma. Esta imagen se procesa convirtiéndola a escala de grises, posteriormente se aplica un filtro gaussiano, a continuación, se aplica otro filtro, detección de bordes y características relevantes; finalmente se segmenta la imagen procesada dividiéndola en 15 imágenes para así

comparar cada una de forma independiente con imágenes base de las características planteadas, esta comparación arroja un valor, que, mientras más se aproxime a 0 indicará una mayor coincidencia de características.

El tiempo cronometrado desde que se acciona el pulso de inicio hasta que la cubeta se detiene en la zona de descarga es de 19,36 segundos en promedio lo que representa un tiempo total del proceso de **0,645333 segundos por huevo de gallina**. Considerando únicamente el apartado de software, el procesamiento de cada captura de imagen, es decir 15 huevos a la vez, tarda 0,272831 segundos en promedio, medido con la función time() de la biblioteca time de Python lo que indica un tiempo total de procesamiento de **0,018189 segundos por huevo de gallina**.

En la Tabla 10 se puede apreciar las métricas obtenidas como resultado de la comparación de un set de imágenes, que han sido rotadas en 4 posiciones distintas a fin de determinar la eficiencia del sistema. Si la imagen gira, los valores cambian, pero se correlacionan en cada comparación arrojando el mismo resultado. Como parámetros relevantes dentro de los métodos empleados se ha establecido un kernel para el filtro gaussiano de 5x5 y como valores umbral de histéresis para aplicación del método Canny, se tiene, máximo 48 y mínimo 18.

Tabla 10.

Resultados de un set de imágenes analizadas por el sistema.

IMAGEN DE ENTRADA	PARÁMETROS DE COMPARACIÓN Y MÉTRICA DE RESULTADOS				
	SENTIDO DE GIRO	FISURA	MANCHAS	NORMAL	POROS
CON FISURA	Original	0.012189926 408618601	0.03486772572 7448895	0.26264968032 23428	0.15006669567 695563
	Anti horario -90°	0.014534808 98966808	0.03721260830 8498374	0.26016070042 837114	0.15241157825 80051
	Anti horario -180°	0.013083906 171765247	0.03576170549 059554	0.26211715981 326666	0.15096067544 010228
	Anti horario -270°	0.010531022 746520846	0.03320882206 535114	0.26487737956 88517	0.14840779201 485788
CON POROS	Original	0.103408692 92491184	0.08073089360 608154	0.32732713543 93281	0.03446807634 342519
	Anti horario -90°	0.104843273 20357106	0.08216547388 474077	0.32876171571 798735	0.03303349606 4765966

	Anti horario -180°	0.106555827 03971017	0.08387802772 087988	0.33047426955 412645	0.03132094222 862686
	Anti horario -270°	0.104849168 71370303	0.08217136939 487274	0.32876761122 81193	0.03302760055 4633996
	Original	0.020945380 53651418	0.00173241878 23161141	0.29494165767 50842	0.11693138873 182285
CON MANCHAS	Anti horario -90°	0.020995451 85041718	0.00168234746 84131146	0.29507270955 142095	0.11688131741 791985
	Anti horario -180°	0.021032450 48076463	0.00164534883 80656633	0.29517909845 789597	0.11684431878 75724
	Anti horario -270°	0.020982422 504250864	0.00169537681 45794301	0.29504796898 297037	0.11689434676 408617
	Original	0.151270074 0701799	0.17394787338 90102	0.10139919034 131537	0.28914684333 85169
NORMAL	Anti horario -90°	0.150799227 06107418	0.17347702637 990448	0.10181439522 93959	0.28867599632 94112
	Anti horario -180°	0.150372383 47252246	0.17305018279 135276	0.10218408810 975085	0.28824915274 08595
	Anti horario -270°	0.150843255 21233832	0.17352105453 116862	0.10176871498 437357	0.28872002448 067535
	Original	0.151270074 0701799	0.17394787338 90102	0.10139919034 131537	0.28914684333 85169

Nota. Elaboración propia.

Para evaluar la calidad de huevos de gallina de distintos proveedores de la provincia de Tungurahua, se consideró los factores de estudio presentados en la Tabla 11.

Tabla 11.*Factores de estudio del análisis experimental.*

FACTORES	CÓDIGO	NIVELES
Proveedores	A	a ₀ Proveedor 1
		a ₁ Proveedor 2
		a ₂ Proveedor 3
Color de los huevos de gallina	B	b ₀ Blancos
		b ₁ Cafés

Nota. Elaboración propia.

De acuerdo a los factores descritos se presenta las combinaciones o tratamientos experimentales en la Tabla 12.

Tabla 12.*Factores de estudio del análisis experimental.*

TRATAMIENTOS	SIMBOLOGÍA	DESCRIPCIÓN
T ₁	a ₀ b ₀	Huevos blancos del Proveedor 1
T ₂	a ₀ b ₁	Huevos cafés del Proveedor 1
T ₃	a ₁ b ₀	Huevos blancos del Proveedor 2
T ₄	a ₁ b ₁	Huevos cafés del Proveedor 2
T ₅	a ₂ b ₀	Huevos blancos del Proveedor 3
T ₆	a ₂ b ₁	Huevos cafés del Proveedor 3

Nota. Elaboración propia.

Una vez definidos los tratamientos se realizaron 3 repeticiones de cada tratamiento, es decir 18 cubetas en total, validando el mejor Proveedor y su mejor producto en concordancia con los resultados estadísticos de los análisis realizados por el prototipo. Estos resultados se basan en las características físicas de los huevos definidas por el sistema y representadas en la Tabla 13 que se muestran a continuación.

Tabla 13.

Resultados del análisis experimental por repeticiones y características definidas.

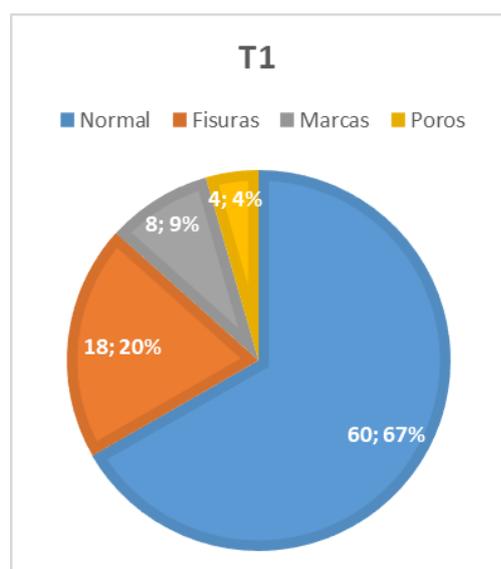
TRATAMIENTOS	REPETICIONES											
	Repetición 1				Repetición 2				Repetición 3			
	Normal	Fisuras	Marcas	Poros	Normal	Fisuras	Marcas	Poros	Normal	Fisuras	Marcas	Poros
T ₁	20	8	1	1	19	6	3	2	21	4	4	1
T ₂	12	1	10	7	13	1	9	7	15	3	4	8
T ₃	24	6	0	0	25	2	1	2	26	2	1	1
T ₄	17	2	6	5	15	4	5	6	17	1	4	8
T ₅	13	4	11	2	16	6	5	3	17	6	5	2
T ₆	6	3	8	13	10	2	6	12	12	3	5	10

Nota. Elaboración propia.

De los resultados presentados en la Tabla 13, se muestra un resumen o resultados globales por tratamiento en las Figuras 10, 11, 12, 13, 14 y 15 respectivamente.

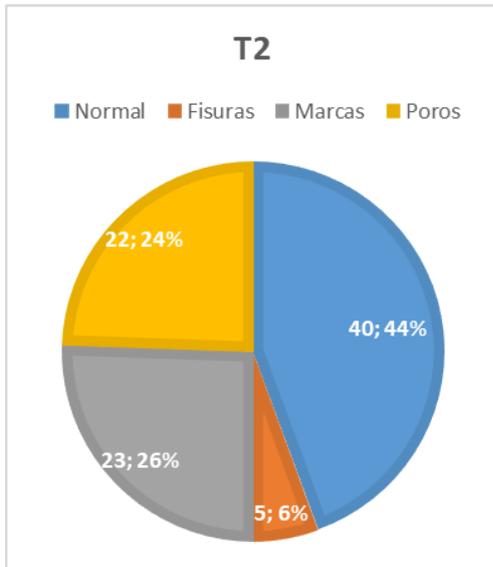
Figura 10.

Resultados finales del Tratamiento 1.



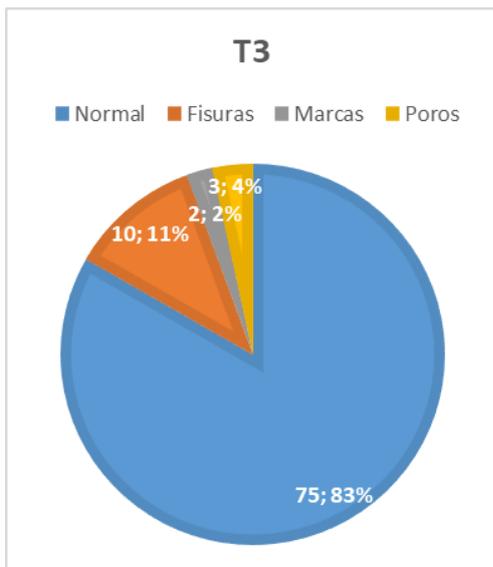
Nota. Elaboración propia.

Figura 11.
Resultados finales del Tratamiento 2.



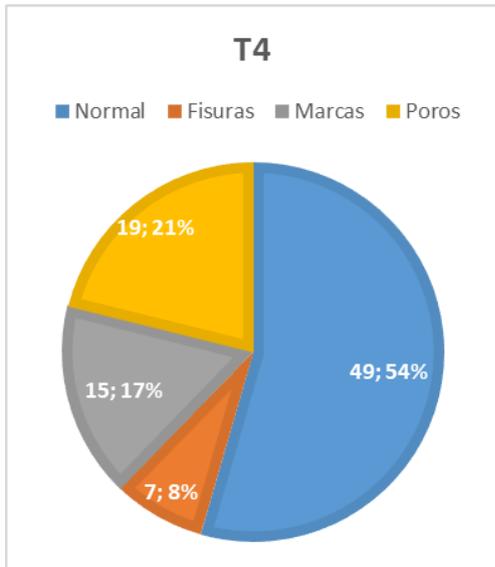
Nota. Elaboración propia.

Figura 12.
Resultados finales del Tratamiento 3.



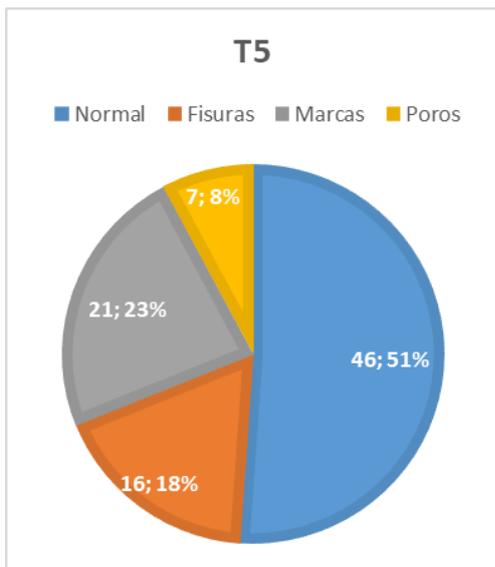
Nota. Elaboración propia.

Figura 13.
Resultados finales del Tratamiento 4.



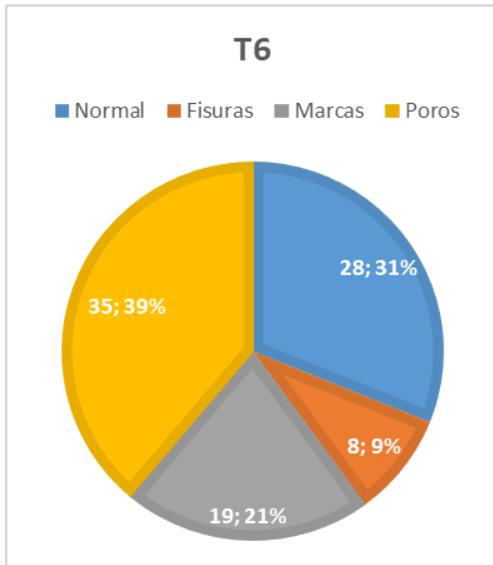
Nota. Elaboración propia.

Figura 14.
Resultados finales del Tratamiento 5.



Nota. Elaboración propia.

Figura 15.
Resultados finales del Tratamiento 6.



Nota. Elaboración propia.

Con estos resultados se evidencia que los huevos de gallina que presentaron mejor calidad son los correspondientes al Tratamiento 3 (T₃), que representa huevos de color blanco del Proveedor 2.

Finalmente se evaluó la eficiencia del sistema analizando cada cubeta por 3 ocasiones consecutivas a fin de determinar posibles alteraciones en los resultados. Reiterando que se analizó 18 cubetas de 30 huevos cada una y a su vez se procedió con 3 análisis de las mismas, se obtuvo un total de 1620 segmentos o imágenes procesadas por el sistema, de las cuales 67 presentaron errores en los resultados, por lo tanto, se tiene un error porcentual de 4,1358 %.

CONCLUSIONES

Los sistemas automáticos de ovoscopia mediante procesamiento digital de imágenes son complejos y conllevan una cantidad innumerable de información. Cada prototipo, desarrollado y en desarrollo, emplea métodos y técnicas diferentes para alcanzar los objetivos de su estudio por lo que es más complejo aún estandarizar estos procesos, aunque se empleen los mismos elementos de hardware o software puesto que estos siguen actualizándose constantemente. Los procesos idóneos para un análisis concreto basado en visión artificial no son idóneos para otro tipo de análisis, a menos que sea una réplica exacta del primero.

El prototipo desarrollado en la presente investigación, cumple con el objetivo de emplear completamente hardware y software libre lo que reduce considerablemente los egresos de los usuarios. A nivel de hardware la interconexión lógica y física, de cada uno de los elementos descritos en este documento, se realiza fácilmente con base en la hoja de datos de los mismos. A nivel de software, es más complejo comparar directamente las imágenes en su estado original a color con funciones como “subtract()” complementado con “split()” y “countNonZero(b)”, que determinan simplemente si una imagen es igual o no a otra, o a su vez el método “FlannBasedMatcher()” complementado con “xfeatures2d.SIFT_create()”, “detectAndCompute()” y “knnMatch()”, que analizan similitudes en las imágenes pero son imprecisos al momento de determinar características concretas. Estos métodos han sido testeados, determinando que el mejor proceso para analizar características físicas de las cáscaras de huevos de gallina es convertir la imagen a escala de grises con el método “cvtColor()”, a continuación aplicar un filtro gaussiano con “GaussianBlur()”, resaltar los bordes y descartar detalles innecesarios con “Canny()” para finalmente establecer una métrica más precisa de comparación con “matchShapes()”.

Los métodos empleados en el algoritmo de procesamiento de imágenes para análisis de calidad de huevos de gallina, pueden ser empleados en otro tipo de análisis que incluyan más características u otros objetos de estudio, pero no se debe aplicar los mismos parámetros ya que estos han sido calibrados específicamente para este prototipo.

Posterior a la aplicación del análisis experimental se establece que el mejor proveedor de huevos de gallina es el Proveedor 2 y su mejor producto son los huevos de color blanco, correspondientes al tratamiento a_1b_0 al obtener un mayor nivel de calidad en los mismos considerado el mayor número con características normales. Por proteger la integridad y no desprestigiar a los Proveedores, por la mala interpretación que pueda darse de los resultados presentados; no se ha utilizado los nombres reales de los mismos.

RECOMENDACIONES

A fin de implementar o innovar un sistema de procesamiento digital de imágenes, es recomendable delimitar concretamente su enfoque u objetivos, considerando los sistemas previos desarrollado, que no necesariamente tendrán el mismo punto de vista u objetos de estudio pero si cierta similitud, también es preciso analizar sus ventajas y desventajas, obteniendo así los recursos principales como punto de partida que permita testear los mismos para determinar las tecnologías y métodos idóneos, encaminando el nuevo estudio a un nivel más técnico y preciso.

Es necesario verificar siempre las hojas de datos de cada uno de los componentes electrónicos previo a su interconexión, determinando así la energía necesaria para su adecuado funcionamiento y los límites de la misma, a fin de evitar daños en los pines, interfaces de comunicación o etapas de potencia. Además, para el control de todos los mecanismos y procesamiento de imágenes, a nivel lógico o de software, se sugiere también verificar los algoritmos propios o externos basados en el lenguaje de programación que se esté empleando, para que estén acorde a los requerimientos del estudio en desarrollo, pero que a la par se cuente con las bases para implementarlos como en el caso de las redes neuronales, que exigen un número considerable, de por lo menos 1000 imágenes según registros previos, para establecer un bajo porcentaje de error por cada característica que se analice.

No es adecuado descartar un método o algoritmo en primera instancia si al testearlo no muestra los resultados esperados, por lo cual es preciso calibrar los parámetros requeridos en cada biblioteca que se ocupe hasta aproximarse al objetivo, comparando resultados con otros algoritmos hasta determinar cuál de ellos es el mejor para el estudio en desarrollo.

Procurar siempre que los estudios de comparación que se realicen en distintas empresas no afecten la integridad de las mismas, ya que los resultados obtenidos pueden ser contraproducentes a menos que se los realice con el aval o solicitud previa de las mismas.

BIBLIOGRAFÍA

- Albán Merino, T. (2018). *Determinación de la calidad física y organoléptica de los huevos comerciales de gallina doméstica que se expenden en los mercados del Distrito Metropolitano de Quito* [Tesis de Veterinaria, Universidad Central del Ecuador]. <http://www.dspace.uce.edu.ec/handle/25000/14288>
- Alikhanov, J., Penchev, S. M., Georgieva, T. D., Moldazhanov, A., Kulmakhambetova, A., Shynybay, Z., Stefanov, E. y Daskalov, P. I. (2019). Design and performance of an automatic egg sorting system based on computer vision. *TEM Journal: Technology, Education, Management, Informatics*, 8(4), 1319 -1325. <https://doi.org/10.18421/TEM84-31>
- Alvarado Maldonado, F. E. y Fuentes Orozco, M. A. (2016). *Propuesta de diseño y cálculo para una banda transportadora de rodillos para piedrín* [Tesis de Ingeniería, Universidad de San Carlos de Guatemala]. <https://core.ac.uk/download/pdf/35293832.pdf>
- Aragua, A. y Mabayo, V.I. (2018). A cost-effective approach for chicken egg weight estimation through computer vision. *International Journal of Agriculture Environment and Food Sciences*, 2(3), 82-87. <https://doi.org/10.31015/jaefs.18013>
- ASUSTeK Computer Inc. (s.f.). *Tinker Board 2S: Specifications*. Tinkerboard. <https://tinker-board.asus.com/product/tinker-board-2s.html>
- Baiza Lovato, C. P. y Jurado Pruna, F. (2020) *Sistema de detección y alerta del estado de somnolencia de conductores mediante visión artificial* [Tesis de Ingeniería, Universidad Tecnológica Israel]. <http://repositorio.uisrael.edu.ec/handle/47000/2441>
- Banana Pi team. (16 de junio de 2022). *Banana Pi BPI-M5*. Banana Pi Wiki. https://wiki.banana-pi.org/Banana_Pi_BPI-M5
- Botta, B., Gattam, S. S. R. y Datta, A. K. (2022). Eggshell crack detection using deep convolutional neural networks. *Journal of Food Engineering*, 315, 110798. <https://doi.org/10.1016/j.jfoodeng.2021.110798>
- Campos Lucas, M. I. (2010). *Un huevo en mi laboratorio*. Bubok. <https://books.google.com.ec/books?id=rlcB5NavrIMC>
- Chincherro Iza, J. L. y Parra Balza, F. D. (2019) *Desarrollo de un dispositivo que mediante visión artificial permita adquirir imágenes con palabras para la conversión a audio, orientado a la ayuda de personas invidentes* [Tesis de Ingeniería, Universidad Tecnológica Israel]. <http://repositorio.uisrael.edu.ec/handle/47000/2117>

- Cruz-Tirado, J. P., Lucimar da Silva Medeiros, M. y Barbin, D. F. (2021). On-line monitoring of egg freshness using a portable NIR spectrometer in tandem with machine learning. *Journal of Food Engineering*, 306, 110643. <https://doi.org/10.1016/j.foodeng.2021.110643>
- Dong, J., Lu, B., He, K., Li, B., Zhao, B. y Tang, X. (2021). Assessment of hatching properties for identifying multiple duck eggs on the hatching tray using machine vision technique. *Computers and Electronics in Agriculture*, 184, 106076. <https://doi.org/10.1016/j.compag.2021.106076>
- Fadchar, N. A. y Dela Cruz, J. C. (2020). Prediction Model for Chicken Egg Fertility Using Artificial Neural Network. In *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, 916–920. <https://doi.org/10.1109/ICIEA49774.2020.9101966>
- Fernández-S. Á., Salazar-L. F., Jurado M., Castellanos E.X., Moreno-P. R. y Buele J. (2019) Electronic System for the Detection of Chicken Eggs Suitable for Incubation Through Image Processing. In Rocha, Á., Adeli, H., Reis, L., Costanzo, S. (eds) *New Knowledge in Information Systems and Technologies. WorldCIST'19 2019. Advances in Intelligent Systems and Computing. Springer, Cham*, 931, 208-218. https://doi.org/10.1007/978-3-030-16184-2_21
- Fernández Soria, Á. P. y Jurado Lozada, M. A. (2018) *Sistema electrónico para el control de calidad de huevos de gallina mediante procesamiento de imágenes* [Tesis de Ingeniería, Universidad Técnica de Ambato]. <https://repositorio.uta.edu.ec/jspui/handle/123456789/28950>
- Genius. (s.f.). *FACECAM 1000X 720P HD WEBCAM* Recuperado el 25 de junio de 2022 de <https://pe.geniusnet.com/product/facecam-1000x/>
- Ghaderi, M., Banakar, A. y Masoudi, A. A. (2018). Using dielectric properties and intelligent methods in separating of hatching eggs during incubation. *Measurement*, 114, 191-194. <https://doi.org/10.1016/j.measurement.2017.09.038>
- Guanjun, B., Mimi, J., Yi, X., Shibo, C. y Qinghua, Y. (2019). Cracked egg recognition based on machine vision. *Computers and Electronics in Agriculture*, 158, 159–166. <https://doi.org/10.1016/j.compag.2019.01.005>
- Huaman Valencia, E. R. (2014). *Tecnología de bandas transportadoras* [Tesis de Ingeniería, Universidad Nacional de San Agustín de Arequipa]. <http://repositorio.unsa.edu.pe/handle/UNSA/2912>
- Lasorz Salazar, P. (2009). *Robótica y automática en el aula: Trabajando con microcontroladores Picaxe*. <https://books.google.com.ec/books?id=6vDxAgAAQBAJ>

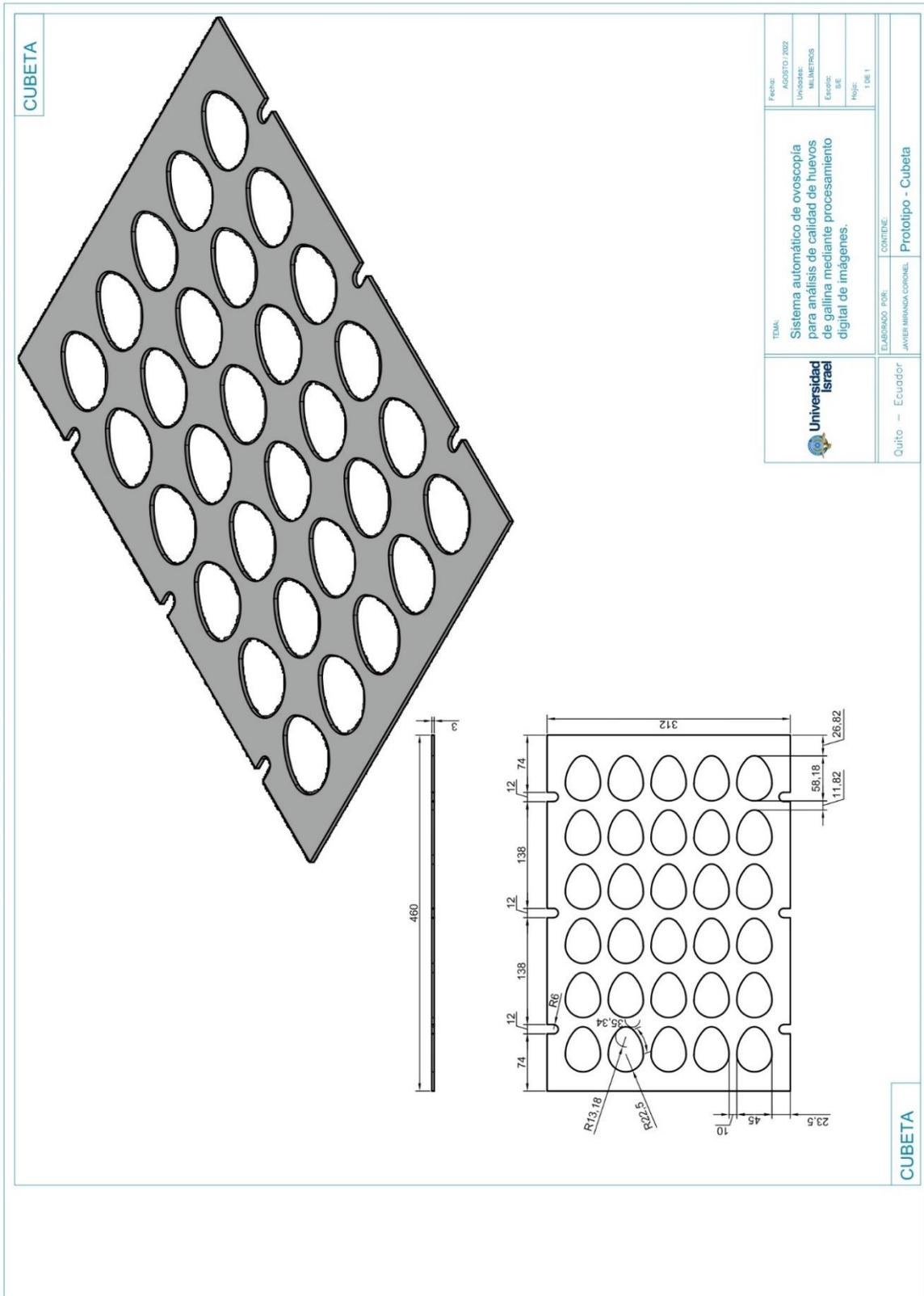
- Lubich, J., Thomas, K., & Engels, D. W. (2019). Identification and classification of poultry eggs: a case study utilizing computer vision and machine learning. *SMU Data Science Review*, 2(1), 20. <https://scholar.smu.edu/datasciencereview/vol2/iss1/20>
- McAndrew, A. (2004). *An introduction to digital image processing with matlab notes for scm2511 image processing*. School of Computer Science and Mathematics, Victoria University of Technology, 264(1), 1-264. <http://202.91.76.90:81/fdScript/RootOfEBooks/ECE/IntroductionToDigitalImageProcessingWithMatlab.pdf>
- Mota-Grajales, R., Torres-Peña, J. C., Camas-Anzueto, J. L., Pérez-Patricio, M., Grajales Coutiño, R., López-Estrada, F. R., Escobar-Gómez, E. N. y Guerra-Crespo, H. (2019). Defect detection in eggshell using a vision system to ensure the incubation in poultry production. *Measurement*, 135, 39–46. <https://doi.org/10.1016/j.measurement.2018.09.059>
- Nasir, A. F. A., Sabarudin, S. S., Majeed, A. P. P. A. y Ghani, A. S. A. (2018). Automated egg grading system using computer vision: Investigation on weight measure versus shape parameters. In *IOP Conference Series: Materials Science and Engineering*, 342(1). <https://doi.org/10.1088/1757-899x/342/1/012003>
- Nasiri, A., Omid, M., & Taheri-Garavand, A. (2020). An automatic sorting system for unwashed eggs using deep learning. *Journal of Food Engineering*, 283, 110036. <https://doi.org/10.1016/j.jfoodeng.2020.110036>
- Nematinia, E., Abdanan Mehdizadeh, S. (2018). Assessment of egg freshness by prediction of Haugh unit and albumen pH using an artificial neural network. *Food Measure* 12, 1449–1459. <https://doi.org/10.1007/s11694-018-9760-1>
- Okinda, C., Sun, Y., Nyalala, I., Korohou, T., Opiyo, S., Wang, J., & Shen, M. (2020). Egg volume estimation based on image processing and computer vision. *Journal of Food Engineering*, 283, 110041. <https://doi.org/10.1016/j.jfoodeng.2020.110041>
- Pitas, I. (2000). *Digital image processing algorithms and applications*. John Wiley & Sons. https://books.google.com.ec/books?id=VQs_Ly4DYDMC
- Pine64. (28 de abril de 2022). *ROCK64*. Pine64 Wiki. <https://wiki.pine64.org/wiki/ROCK64>
- Quillooy, E. P., Suministrado, D. C., & Bato, P. M. (2018). Single-line automated sorter using mechatronics and machine vision system for Philippine table eggs. *African Journal of Agricultural Research*, 13(17), 918-926. <https://doi.org/10.5897/AJAR2018.13113>

- Raspberry Pi. (Junio de 2021). *Raspberry Pi 4 Computer: Model B* [Archivo PDF]. <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf>
- Raspberry Pi. (2022). *Raspberry Pi 4 Computer: Model B* [Archivo PDF]. <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf>
- Raspberry Pi. (s.f.). *Raspberry Pi Documentation: Camera*. Recuperado el 25 de junio de 2022 de <https://www.raspberrypi.com/documentation/accessories/camera.html>
- Samsung. (s.f.). *Galaxy A50* Recuperado el 29 de agosto de 2022 de <https://www.samsung.com/latin/smartphones/galaxy-a/galaxy-a50-blue-64gb-sm-a505gzbtqa/#specs>
- Servicio Ecuatoriano de Normalización. (2013). *Huevos comerciales y ovoproductos: Requisitos. Primera edición* (NTE INEN 1973:2013). <https://www.normalizacion.gob.ec/buzon/normas/1973-2.pdf>
- Soler Castillo, R. y Bueso Ródenas, J. (2018). Análisis de las alteraciones de la cáscara del huevo de gallina. *Nereis*, (10), 137-147. <http://hdl.handle.net/20.500.12466/869>
- Turkoglu, M. (2021). Defective egg detection based on deep features and Bidirectional Long-Short-Term-Memory. *Computers and Electronics in Agriculture*, 185, 106152. <https://doi.org/10.1016/j.compag.2021.106152>
- Valencia, Y. M., Majin, J. J., Taveira, V. B., Salazar, J. D., Stivanello, M. E., Ferreira, L. C. y Stemmer, M. R. (2021). A novel method for inspection defects in commercial eggs using computer vision. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43, 809-816. <https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-809-2021>
- Wang, C., Zhou, J., Wu, H., Li, J., Chunjiang, Z., & Liu, R. (2020). Research on the Evaluation Method of Eggshell Dark Spots Based on Machine Vision. *IEEE Access*, 8, 160116–160125. <https://doi.org/10.1109/ACCESS.2020.3020260>

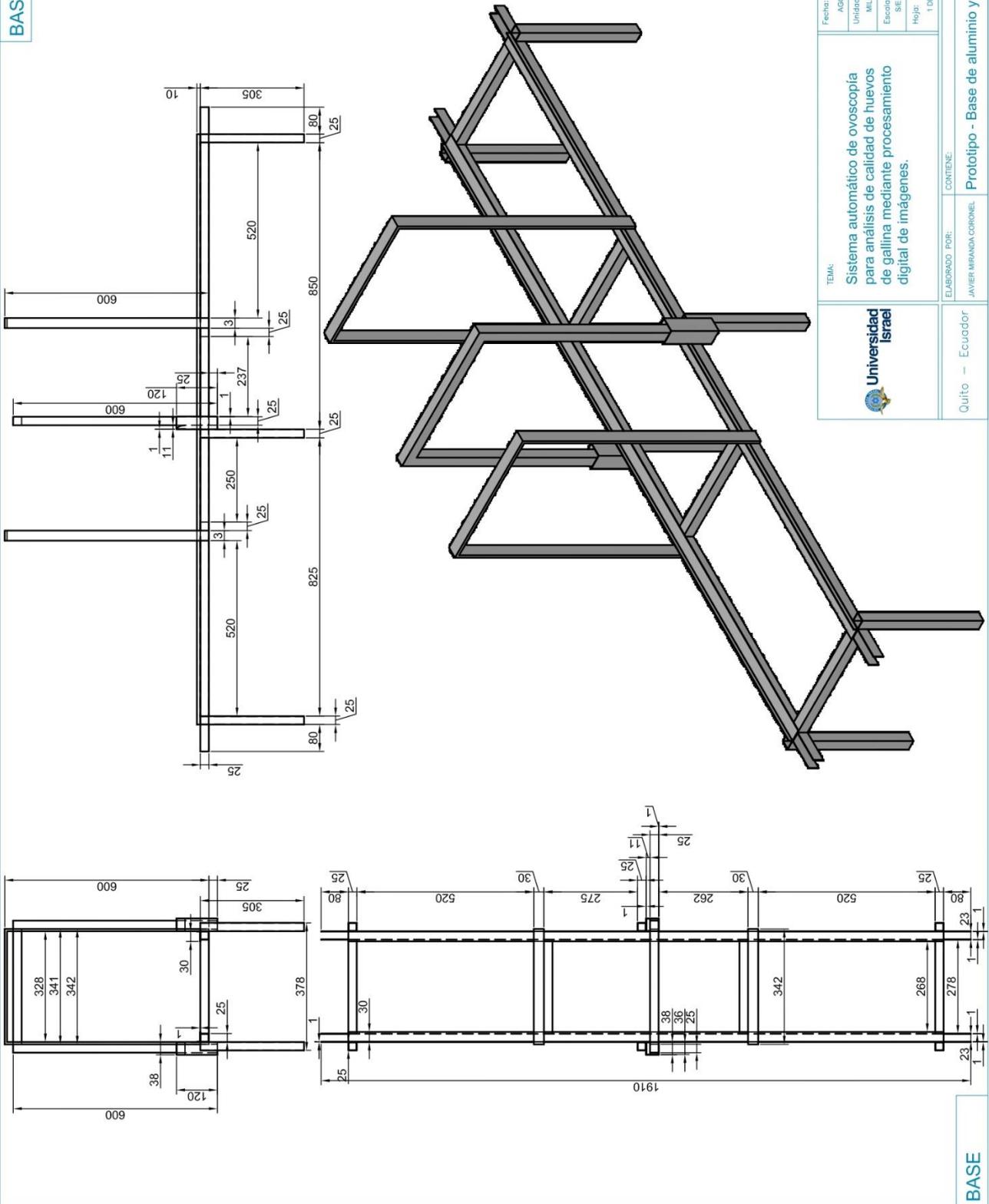
ANEXOS

ANEXO 1

PLANOS DE LAS ESTRUCTURAS DESARROLLADAS PARA EL PROTOTIPO

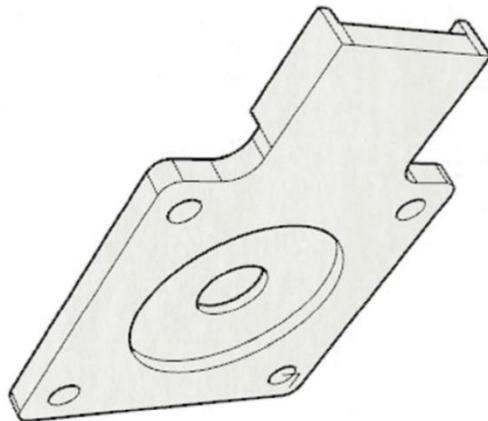
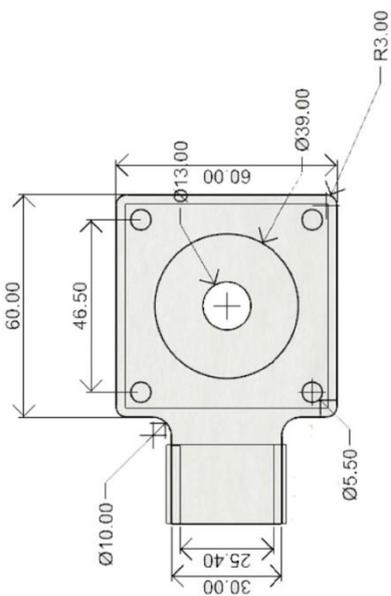
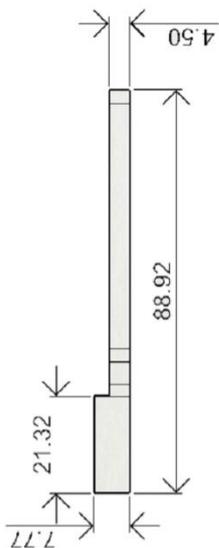


BASE



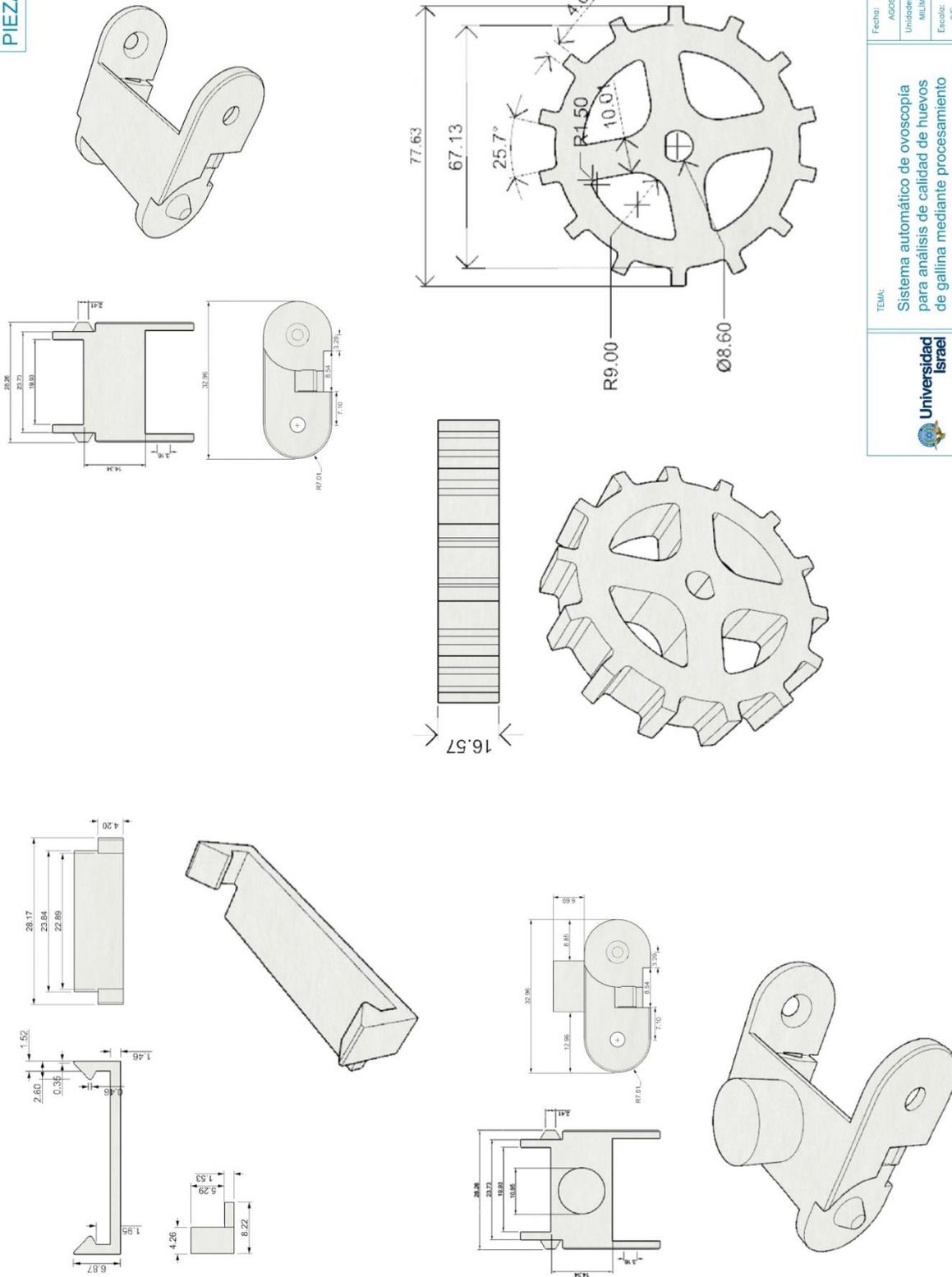
<p>FECHA: AGOSTO / 2022</p> <p>UNIDADES: MILIMETROS</p> <p>ESCALA: 3/4</p> <p>HOJA: 1 DE 1</p>	<p>TEMA: Sistema automático de ovoscopia para análisis de calidad de huevos de gallina mediante procesamiento digital de imágenes.</p>	<p>ELABORADO POR: JAVIER MIRANDA CORONEL</p>	<p>CONTIENE: Prototipo - Base de aluminio y plástico</p>
	<p>Quito - Ecuador</p>		

BASE



	TEMA:	Sistema automático de ovoscopia para análisis de calidad de huevos de gallina mediante procesamiento digital de imágenes.
	ELABORADO POR:	JAVIER MIRANDA CORONEL
	CONTIENE:	Prototipo - Soporte para motores Nema 23
	FECHA:	AGOSTO / 2022
	UNIDAD:	MILIMETROS
	ESCALA:	5:1
	HOJA:	1 DE 1

PIEZAS



<p>FECHA: AGOSTO / 2022</p> <p>UNIDADES: MILIMETROS</p> <p>ESCALA: 5/8</p> <p>HOJA: 1 DE 1</p>	<p>TEMA: Sistema automático de ovoscopia para análisis de calidad de huevos de gallina mediante procesamiento digital de imágenes.</p>	<p>ELABORADO POR: JAVIER MIRANDA CORONEL</p>	<p>CONTIENE: Prototipo - Piezas PLA para cadena</p>
	<p>Quito - Ecuador</p>		

PIEZAS

ANEXO 2

CÓDIGO DESARROLLADO EN ARDUINO PARA EL CONTROL DEL SISTEMA

```
#include <SoftReset.h>

//pines de control drivers tb6600

#define paso1 3

#define dir1 4

#define ena1 5

#define paso2 6

#define dir2 7

#define ena2 8

//retardo entre pulsos

#define pausa 3500

//pines de la botonera y sensores

#define scubeta 10

#define sfinal 9

#define paro 2

#define inicio 12

//pin para activar cámara

#define camara 50

int contador=0;

void setup() {

    //pines de los drivers como salidas

    pinMode(paso1, OUTPUT);

    pinMode(dir1, OUTPUT);

    pinMode(ena1, OUTPUT);
```

```

pinMode(paso2, OUTPUT);

pinMode(dir2, OUTPUT);

pinMode(ena2, OUTPUT);

digitalWrite(dir1, HIGH); //HIGH DERECHO - LOW IZQUIERDO

digitalWrite(dir2, LOW);

digitalWrite(ena1, LOW);

digitalWrite(ena2, LOW);

//pines de los sensores y botonera como entradas

pinMode(paro, INPUT_PULLUP);

pinMode(inicio, INPUT_PULLUP);

pinMode(scubeta, INPUT);

pinMode(sfinal, INPUT);

//pin raspberry

pinMode(camara, OUTPUT);

//INTERRUPCION PARO DE EMERGENCIA

attachInterrupt(digitalPinToInterrupt(2), interrupcion, RISING);

}

void interrupcion(){

int resto = 1000 - contador;

digitalWrite(ena1, HIGH);

digitalWrite(ena2, HIGH);

if(resto==1000){

soft_restart();

}

while(digitalRead(inicio)==HIGH){

```

```

}

digitalWrite(ena1, LOW);

digitalWrite(ena2, LOW);

for (int i = 0; i < resto; i++) {

    digitalWrite(paso1, HIGH);

    digitalWrite(paso2, HIGH);

    delayMicroseconds(4000);

    digitalWrite(paso1, LOW);

    digitalWrite(paso2, LOW);

    delayMicroseconds(4000);

    //Serial.println("Reiniciando...");

}

soft_restart();

}

void detectado(){

    delay(2000);

    //MOVIMIENTO 2

    digitalWrite(ena1, LOW);

    digitalWrite(ena2, LOW);

    delay(100);

    for (int i = 0; i < 161; i++) {

        digitalWrite(paso1, HIGH);

        digitalWrite(paso2, HIGH);

        delayMicroseconds(pausa);

        digitalWrite(paso1, LOW);

```

```

digitalWrite(paso2, LOW);

delayMicroseconds(pausa);

contador++;

}

delay(100);

digitalWrite(ena1, HIGH);

digitalWrite(ena2, HIGH);

delay(6000);

//MOVIMIENTO 3

digitalWrite(ena1, LOW);

digitalWrite(ena2, LOW);

delay(100);

for (int i = 0; i < 310; i++) {

digitalWrite(paso1, HIGH);

digitalWrite(paso2, HIGH);

delayMicroseconds(pausa);

digitalWrite(paso1, LOW);

digitalWrite(paso2, LOW);

delayMicroseconds(pausa);

contador++;

}

delay(100);

// digitalWrite(ena1, HIGH);

// digitalWrite(ena2, HIGH);

while(true){

```

```

    if(digitalRead(sfina1)==HIGH){

        delay(5000);

        break;

    }

}

//MOVIMIENTO 4

// digitalWrite(ena1, LOW);

// digitalWrite(ena2, LOW);

delay(100);

for (int i = 0; i < 222; i++) {

    // These four lines result in 1 step:

    digitalWrite(paso1, HIGH);

    digitalWrite(paso2, HIGH);

    delayMicroseconds(pausa);

    digitalWrite(paso1, LOW);

    digitalWrite(paso2, LOW);

    delayMicroseconds(pausa);

    contador++;

}

delay(100);

contador=0;

digitalWrite(ena1, HIGH);

digitalWrite(ena2, HIGH);

delay(2000);

}

```

```

void loop() {
//MOVIMIENTO 1
while(true){
    digitalWrite(ena1, LOW);
    digitalWrite(ena2, LOW);
    if(digitalRead(inicio)==LOW){
        delay(100);
        for (int i = 0; i < 307; i++) {
            // These four lines result in 1 step:
            digitalWrite(paso1, HIGH);
            digitalWrite(paso2, HIGH);
            delayMicroseconds(pausa);
            digitalWrite(paso1, LOW);
            digitalWrite(paso2, LOW);
            delayMicroseconds(pausa);
            contador++;
        }
        delay(100);
        digitalWrite(ena1, HIGH);
        digitalWrite(ena2, HIGH);
        delay(4500);
        break;
    }
}
}
if(digitalRead(scubeta)==LOW){

```

```
digitalWrite(camara, HIGH);

delay(750);

digitalWrite(camara, LOW);

detectado();

}else{

digitalWrite(ena1, LOW);

digitalWrite(ena2, LOW);

delay(100);

for (int i = 0; i < 693; i++) {

// These four lines result in 1 step:

digitalWrite(paso1, HIGH);

digitalWrite(paso2, HIGH);

delayMicroseconds(pausa);

digitalWrite(paso1, LOW);

digitalWrite(paso2, LOW);

delayMicroseconds(pausa);

contador++;

}

delay(100);

contador=0;

digitalWrite(ena1, HIGH);

digitalWrite(ena2, HIGH);

}

}
```

ANEXO 3

CÓDIGO DESARROLLADO EN PYTHON ORIENTADO AL SISTEMA OPERATIVO RASPBERRY PI OS PARA EL PROCESAMIENTO DE IMÁGENES

INTERFAZ

```
#Importación de módulos o librerías
from PyQt5 import QtGui
from PyQt5.QtWidgets import *
from PyQt5.QtGui import QPixmap, QFont
from PyQt5.QtCore import pyqtSignal, pyqtSlot, Qt, QThread
import sys
import cv2
import time
import threading
import numpy as np
from datetime import datetime
import RPi.GPIO as GPIO
import pymysql

GPIO.setwarnings(False) #desactivar las alertas de los pines GPIO
#Creación de métodos
class video(QThread): #QThread se inicializa para controlar subprocessos
    #se inicializa una nueva señal con un arreglo matricial para adecuar las imágenes o frames
    varpixmap = pyqtSignal(np.ndarray)

    def __init__(self): #se crea una función principal
        super().__init__() #se crea un objeto temporal con acceso a los métodos de la clase principal
        self.varbool = True #se inicializa una variable auxiliar para ingresar a un ciclo

    def run(self): #se crea un método para inicialización de captura de video de la cámara seleccionada
        captura = cv2.VideoCapture("http://192.168.1.37:8080/video") #camara de servidor #2160x3840
        #captura = cv2.VideoCapture(0) #camara usb
        #    captura.set(cv2.CAP_PROP_FRAME_WIDTH, 1920)
        #    captura.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080)
        while self.varbool: #se inicia un ciclo continuo mientras la variable auxiliar sea verdadera (true)
            #VideoCapture().read() devuelve 2 valores:
```

```

#en la variable valid se almacena un valor booleano que determina si la imagen es adquirida o
no
#en la variable cv_imag se almacena un vector tridimensional que
#corresponde a las imágenes o frames que se adquieren por la cámara
valid, cv_imag = captura.read()
#print('Resolution: ' + str(cv_imag.shape[0]) + ' x ' + str(cv_imag.shape[1]))
if valid: #se verifica el valor de la variable valid
    self.varpixmap.emit(cv_imag) #mientras la variable valid sea True, se transmite la imagen
#si se cambia el valor de la variable auxiliar, se suspende la adquisición de imágenes
captura.release()

def stop(self): #método para cerrar correctamente el programa
    self.varbool = False #se cambia el valor de la variable auxiliar a False para cerrar
    self.wait() #se espera a que el ciclo se deje de ejecutar para cerrar

class aplicacion(QWidget): #QWidget se inicializa para adicionar objetos a la interfaz
def __init__(self): #se crea una función principal
    super().__init__() #se crea un objeto temporal con acceso a los métodos de la clase principal
    self.setWindowTitle("Análisis") #se le da un nombre a la ventana
    self.definir = False #inicialización de variables auxiliares
    self.definir2 = False
    self.activaraux = False
    #se crean etiquetas para mostrar imágenes dentro de la ventana
    vbox = QVBoxLayout() #se crea un contenedor para disposición vertical de objetos y se almacena
en una variable
    hboxenc = QHBoxLayout()
    vbox1 = QVBoxLayout()
    hbox = QHBoxLayout() #se crea un contenedor para disposición horizontal de objetos y se
almacena en una variable
    hbox2 = QHBoxLayout()
    self.basefisuras=0
    self.basemanchas=0
    self.basenormal=0
    self.baseporos=0

```

```

self.etiqimagen = QLabel(self) #creación de la etiqueta dentro de la ventana principal
#el método setSizePolicy establece las dimensiones del objeto
#QSizePolicy.Expanding estable las dimensiones máximas de acuerdo con la ventana
#setStyleSheet establece estilos en los objetos
#border: establece un borde con un grosor expresado en pixeles (px)
#un tipo de línea que se define a continuación y un color de la misma
self.etiqimagen.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.Expanding)
#self.etiqimagen.setStyleSheet("border: 3px solid blue;")
self.etiqimagen2 = QLabel(self)
self.etiqimagen2.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.Expanding)
#self.etiqimagen2.setStyleSheet("border: 3px solid black;")
self.etiqimagen3 = QLabel(self)
self.etiqimagen3.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.Expanding)
#self.etiqimagen3.setStyleSheet("border: 3px solid yellow;")
self.etiqlogo = QLabel(self)
self.etiqlogo.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.Expanding)
logo=QPixmap('LOGO_UISRAEL.png').scaled((self.size()/1.7), Qt.KeepAspectRatio) #se carga y
escala el logo deseado
self.etiqlogo.setPixmap(logo) #se visualiza el logo dentro de la etiqueta
self.etiqtema = QLabel("SISTEMA AUTOMÁTICO\n" #se crea una etiqueta con el texto deseado
"DE OVOSCOPIA.\n"
"Ing. Javier Miranda",self)
self.etiqtema.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.Expanding)
self.etiqtema.setFont(QFont('Bauhaus 93',16, QFont.Bold)) # se añade la fuente y tamaño al texto
self.etiqtema.setAlignment(Qt.AlignLeft) #se alinea el texto
# self.botoncap = QPushButton("Capturar") #creación del botón que emula la señal del sensor
# self.botoncap.setMaximumWidth(100)
# self.botoncap.setMinimumHeight(30)
vbox.addWidget(self.etiqimagen,alignment=Qt.AlignCenter) #se agregan los objetos dentro del
contenedor
#vbox.addWidget(self.etiqimagen2,alignment=Qt.AlignCenter) #y se establece una alineación
centrada
hboxenc.addWidget(self.etiqlogo,alignment=Qt.AlignCenter)
hboxenc.addWidget(self.etiqtema,alignment=Qt.AlignCenter)

```

```

hbox2.addWidget(self.etiqimagen2,alignment=Qt.AlignCenter)
hbox2.addWidget(self.etiqimagen3,alignment=Qt.AlignCenter)
vbox1.addLayout(hboxenc)
#   vbox1.addWidget(self.botoncap,alignment=Qt.AlignCenter)
vbox1.addLayout(hbox2)
hbox.addLayout(vbox1) #se agrega un contenedor dentro de otro contenedor
hbox.addLayout(vbox) #se agrega un contenedor dentro de otro contenedor
#se establecen los contenedores como objetos dentro de la ventana principal
self.setLayout(hbox)
#   self.botoncap.clicked.connect(self.comparar) #código para detectar cuando se presiona el botón
y accionar la función
    self.vivo = video() #se llama a la clase video para su ejecución
    self.vivo.varpixmap.connect(self.actual_image) #se conectan las señales de las clases
    self.vivo.start() #se inicia el hilo de funciones

def closeEvent(self, evento): #controlador de eventos para cerrar la aplicación
    respuesta = QMessageBox.question(self, 'Cerrar aplicación', '¿Seguro que deseas cerrar la
aplicación?',
        QMessageBox.Yes, QMessageBox.No) #Se crea un cuadro de diálogo con 2 botones: Yes y
No por defecto
    if respuesta == QMessageBox.Yes: #Se Determina si el botón Yes es presionado
        self.vivo.stop() #se detiene el hilo de funciones
        evento.accept() #se acepta el evento la función de finalización
    else: #caso contrario
        evento.ignore() #se ignora el evento, se cierra el el cuadro de diálogo y se sigue ejecutando el
programa

#   def comparar(self): #función para activar las variables auxiliares cuando se presiona el botón
#       self.definir = True
#       self.activaraux = True
#       print (self.definir)

def esperar(self): #función para establecer un tiempo de espera entre capturas de imagen
    if self.activaraux:

```

```

for i in range(5):
    time.sleep(1.5)
    self.definir2 = True
else:
    #mientras no se active la señal o se presione el botón, se ignora
    #la función y se continua ejecutando el resto del código
    pass

@pyqtSlot(np.ndarray) #se emplea para conectar el arreglo matricial de la clase anterior
def actual_image(self, cv_img):
    rotada=cv2.rotate(cv_img, cv2.ROTATE_90_COUNTERCLOCKWISE)
    #corte=rotada[610:3260,180:2160] #3840x2160
    corte=rotada[305:1630,90:1080] #1920x1080
    qt_img = self.convertir(corte) #se almacena la imagen convertida de opencv a QPixmap
compatible con la interfaz
    self.etiqimagen.setPixmap(qt_img) #el método setPixmap se emplea para cargar la imagen en la
etiqueta
    GPIO.setmode(GPIO.BOARD) #Configuración de la numeración de pines en el orden de
distribución de la placa
    GPIO.setup(18,GPIO.IN) #Configuración de pin como entrada
    if GPIO.input(18) == GPIO.HIGH: #detección del estado del pin conectado al sensor
        #función para activar las variables auxiliares cuando detecta el sensor
        self.definir = True
        self.activaraux = True
    else:
        pass
    while (self.definir): #ciclo para capturar la primera imagen cuando se active el sensor
        qt_img2 = self.convertir2(corte)
        self.etiqimagen2.setPixmap(qt_img2)
        self.definir = False #cambio de estado de variable para cerrar el ciclo después de capturar la
imagen
    while (self.definir2): #ciclo para capturar la segunda imagen después de un tiempo
        qt_img3 = self.convertir3(corte)
        self.etiqimagen3.setPixmap(qt_img3)

```



```

0,0,0,0,
0,0,0,0,
0,0,0,0,
0,0,0,0,
0,0,0,0,
0,0,0,0,
0,0,0,0]

#FILTRADO IMAGEN ORIGINAL
original = cv2.imread("fechastr.jpg")
gris0 = cv2.cvtColor(original, cv2.COLOR_BGR2GRAY)
gauss0 = cv2.GaussianBlur(gris0, (11,11), 0)
canny0 = cv2.Canny(gauss0, 25, 10) #25,15

#FILTRADO IMAGENES BASE
compx = cv2.imread("cfis.jpg")
grisx = cv2.cvtColor(compx, cv2.COLOR_BGR2GRAY)
gaussx = cv2.GaussianBlur(grisx, (5,5), 0)
cannyx = cv2.Canny(gaussx, 48, 18) #48,18
compx1 = cv2.imread("cman.jpg")
grisx1 = cv2.cvtColor(compx1, cv2.COLOR_BGR2GRAY)
gaussx1 = cv2.GaussianBlur(grisx1, (5,5), 0)
cannyx1 = cv2.Canny(gaussx1, 40, 15) #40,15
compx2 = cv2.imread("cnormal.jpg")
grisx2 = cv2.cvtColor(compx2, cv2.COLOR_BGR2GRAY)
gaussx2 = cv2.GaussianBlur(grisx2, (5,5), 0)
cannyx2 = cv2.Canny(gaussx2, 100, 25) #100,25
compx3 = cv2.imread("cporos.jpg")
grisx3 = cv2.cvtColor(compx3, cv2.COLOR_BGR2GRAY)
gaussx3 = cv2.GaussianBlur(grisx3, (15,15), 0)
cannyx3 = cv2.Canny(gaussx3, 24, 10) #24,10

#COMPARACION INDIVIDUAL DE CADA IMAGEN
resvector1[0]=cv2.matchShapes(cannyx,canny0[20:290,0:335],1,0.0)#1
resvector1[1]=cv2.matchShapes(cannyx1,canny0[20:290,0:335],1,0.0)
resvector1[2]=cv2.matchShapes(cannyx2,canny0[20:290,0:335],1,0.0)
resvector1[3]=cv2.matchShapes(cannyx3,canny0[20:290,0:335],1,0.0)

```

```

#cv2.imshow("test1", canny0[20:290,0:335])
resvector1[4]=cv2.matchShapes(cannyx,canny0[20:290,335:660],1,0.0)#2
resvector1[5]=cv2.matchShapes(cannyx1,canny0[20:290,335:660],1,0.0)
resvector1[6]=cv2.matchShapes(cannyx2,canny0[20:290,335:660],1,0.0)
resvector1[7]=cv2.matchShapes(cannyx3,canny0[20:290,335:660],1,0.0)
#cv2.imshow("test2", canny0[20:290,335:660])
resvector1[8]=cv2.matchShapes(cannyx,canny0[20:290,660:990],1,0.0)#3
resvector1[9]=cv2.matchShapes(cannyx1,canny0[20:290,660:990],1,0.0)
resvector1[10]=cv2.matchShapes(cannyx2,canny0[20:290,660:990],1,0.0)
resvector1[11]=cv2.matchShapes(cannyx3,canny0[20:290,660:990],1,0.0)
#cv2.imshow("test3", canny0[20:290,660:990])
resvector1[12]=cv2.matchShapes(cannyx,canny0[290:530,0:335],1,0.0)#4
resvector1[13]=cv2.matchShapes(cannyx1,canny0[290:530,0:335],1,0.0)
resvector1[14]=cv2.matchShapes(cannyx2,canny0[290:530,0:335],1,0.0)
resvector1[15]=cv2.matchShapes(cannyx3,canny0[290:530,0:335],1,0.0)
#cv2.imshow("test4", canny0[290:530,0:335])
resvector1[16]=cv2.matchShapes(cannyx,canny0[290:530,335:660],1,0.0)#5
resvector1[17]=cv2.matchShapes(cannyx1,canny0[290:530,335:660],1,0.0)
resvector1[18]=cv2.matchShapes(cannyx2,canny0[290:530,335:660],1,0.0)
resvector1[19]=cv2.matchShapes(cannyx3,canny0[290:530,335:660],1,0.0)
#cv2.imshow("test5", canny0[290:530,335:660])
resvector1[20]=cv2.matchShapes(cannyx,canny0[290:530,660:990],1,0.0)#6
resvector1[21]=cv2.matchShapes(cannyx1,canny0[290:530,660:990],1,0.0)
resvector1[22]=cv2.matchShapes(cannyx2,canny0[290:530,660:990],1,0.0)
resvector1[23]=cv2.matchShapes(cannyx3,canny0[290:530,660:990],1,0.0)
#cv2.imshow("test6", canny0[290:530,660:990])
resvector1[24]=cv2.matchShapes(cannyx,canny0[530:790,0:335],1,0.0)#7
resvector1[25]=cv2.matchShapes(cannyx1,canny0[530:790,0:335],1,0.0)
resvector1[26]=cv2.matchShapes(cannyx2,canny0[530:790,0:335],1,0.0)
resvector1[27]=cv2.matchShapes(cannyx3,canny0[530:790,0:335],1,0.0)
#cv2.imshow("test7", canny0[530:790,0:335])
resvector1[28]=cv2.matchShapes(cannyx,canny0[530:790,335:660],1,0.0)#8
resvector1[29]=cv2.matchShapes(cannyx1,canny0[530:790,335:660],1,0.0)
resvector1[30]=cv2.matchShapes(cannyx2,canny0[530:790,335:660],1,0.0)

```

```

resvector1[31]=cv2.matchShapes(cannyx3,canny0[530:790,335:660],1,0.0)
#cv2.imshow("test8", canny0[530:790,335:660])
resvector1[32]=cv2.matchShapes(cannyx,canny0[530:790,660:990],1,0.0)#9
resvector1[33]=cv2.matchShapes(cannyx1,canny0[530:790,660:990],1,0.0)
resvector1[34]=cv2.matchShapes(cannyx2,canny0[530:790,660:990],1,0.0)
resvector1[35]=cv2.matchShapes(cannyx3,canny0[530:790,660:990],1,0.0)
#cv2.imshow("test9", canny0[530:790,660:990])
resvector1[36]=cv2.matchShapes(cannyx,canny0[790:1040,0:335],1,0.0)#10
resvector1[37]=cv2.matchShapes(cannyx1,canny0[790:1040,0:335],1,0.0)
resvector1[38]=cv2.matchShapes(cannyx2,canny0[790:1040,0:335],1,0.0)
resvector1[39]=cv2.matchShapes(cannyx3,canny0[790:1040,0:335],1,0.0)
#cv2.imshow("test10", canny0[790:1040,0:335])
resvector1[40]=cv2.matchShapes(cannyx,canny0[790:1040,335:660],1,0.0)#11
resvector1[41]=cv2.matchShapes(cannyx1,canny0[790:1040,335:660],1,0.0)
resvector1[42]=cv2.matchShapes(cannyx2,canny0[790:1040,335:660],1,0.0)
resvector1[43]=cv2.matchShapes(cannyx3,canny0[790:1040,335:660],1,0.0)
#cv2.imshow("test11", canny0[790:1040,335:660])
resvector1[44]=cv2.matchShapes(cannyx,canny0[790:1040,660:990],1,0.0)#12
resvector1[45]=cv2.matchShapes(cannyx1,canny0[790:1040,660:990],1,0.0)
resvector1[46]=cv2.matchShapes(cannyx2,canny0[790:1040,660:990],1,0.0)
resvector1[47]=cv2.matchShapes(cannyx3,canny0[790:1040,660:990],1,0.0)
#cv2.imshow("test12", canny0[790:1040,660:990])
resvector1[48]=cv2.matchShapes(cannyx,canny0[1040:1325,0:335],1,0.0)#13
resvector1[49]=cv2.matchShapes(cannyx1,canny0[1040:1325,0:335],1,0.0)
resvector1[50]=cv2.matchShapes(cannyx2,canny0[1040:1325,0:335],1,0.0)
resvector1[51]=cv2.matchShapes(cannyx3,canny0[1040:1325,0:335],1,0.0)
#cv2.imshow("test13", canny0[1040:1325,0:335])
resvector1[52]=cv2.matchShapes(cannyx,canny0[1040:1325,335:660],1,0.0)#14
resvector1[53]=cv2.matchShapes(cannyx1,canny0[1040:1325,335:660],1,0.0)
resvector1[54]=cv2.matchShapes(cannyx2,canny0[1040:1325,335:660],1,0.0)
resvector1[55]=cv2.matchShapes(cannyx3,canny0[1040:1325,335:660],1,0.0)
#cv2.imshow("test14", canny0[1040:1325,335:660])
resvector1[56]=cv2.matchShapes(cannyx,canny0[1040:1325,660:990],1,0.0)#15
resvector1[57]=cv2.matchShapes(cannyx1,canny0[1040:1325,660:990],1,0.0)

```

```
resvector1[58]=cv2.matchShapes(cannyx2,canny0[1040:1325,660:990],1,0.0)
resvector1[59]=cv2.matchShapes(cannyx3,canny0[1040:1325,660:990],1,0.0)
#cv2.imshow("test15",canny0[1040:1325,660:990])
#BUSCAR VALOR MENOR
menor=resvector1[0]
menor1=resvector1[4]
menor2=resvector1[8]
menor3=resvector1[12]
menor4=resvector1[16]
menor5=resvector1[20]
menor6=resvector1[24]
menor7=resvector1[28]
menor8=resvector1[32]
menor9=resvector1[36]
menor10=resvector1[40]
menor11=resvector1[44]
menor12=resvector1[48]
menor13=resvector1[52]
menor14=resvector1[56]
posicion=0
posicion1=0
posicion2=0
posicion3=0
posicion4=0
posicion5=0
posicion6=0
posicion7=0
posicion8=0
posicion9=0
posicion10=0
posicion11=0
posicion12=0
posicion13=0
posicion14=0
```

```

contf=0
contm=0
contn=0
contp=0
for x in range(0,4):
    if (resvector1[x]<menor):
        menor=resvector1[x]
        posicion=x
    if posicion==0:
        cv2.putText(original,'F',(100,120),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contf=contf+1
    if posicion==1:
        cv2.putText(original,'M',(100,120),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contm=contm+1
    if posicion==2:
        cv2.putText(original,'N',(100,120),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contn=contn+1
    if posicion==3:
        cv2.putText(original,'P',(100,120),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contp=contp+1
for x in range(4,8):
    if (resvector1[x]<menor1):
        menor1=resvector1[x]
        posicion1=x-4
    if posicion1==0:
        cv2.putText(original,'F',(435,120),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contf=contf+1
    if posicion1==1:

```

```

        cv2.putText(original,'M',(435,120),
3,(255,255,255),8,cv2.LINE_AA
        contm=contm+1
    if posicion1==2:
        cv2.putText(original,'N',(435,120),
3,(255,255,255),8,cv2.LINE_AA
        contn=contn+1
    if posicion1==3:
        cv2.putText(original,'P',(435,120),
3,(255,255,255),8,cv2.LINE_AA
        contp=contp+1
    for x in range(8,12):
        if (resvector1[x]<menor2):
            menor2=resvector1[x]
            posicion2=x-8
    if posicion2==0:
        cv2.putText(original,'F',(760,120),
3,(255,255,255),8,cv2.LINE_AA
        conf=conf+1
    if posicion2==1:
        cv2.putText(original,'M',(760,120),
3,(255,255,255),8,cv2.LINE_AA
        contm=contm+1
    if posicion2==2:
        cv2.putText(original,'N',(760,120),
3,(255,255,255),8,cv2.LINE_AA
        contn=contn+1
    if posicion2==3:
        cv2.putText(original,'P',(760,120),
3,(255,255,255),8,cv2.LINE_AA
        contp=contp+1
    for x in range(12,16):
        if (resvector1[x]<menor3):
            menor3=resvector1[x]

```

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

```

    posicion3=x-12
if posicion3==0:
    cv2.putText(original,'F',(100,390),
3,(255,255,255),8,cv2.LINE_AA)
    cv2.FONT_HERSHEY_SIMPLEX,
    contf=contf+1
if posicion3==1:
    cv2.putText(original,'M',(100,390),
3,(255,255,255),8,cv2.LINE_AA)
    cv2.FONT_HERSHEY_SIMPLEX,
    contm=contm+1
if posicion3==2:
    cv2.putText(original,'N',(100,390),
3,(255,255,255),8,cv2.LINE_AA)
    cv2.FONT_HERSHEY_SIMPLEX,
    contn=contn+1
if posicion3==3:
    cv2.putText(original,'P',(100,390),
3,(255,255,255),8,cv2.LINE_AA)
    cv2.FONT_HERSHEY_SIMPLEX,
    contp=contp+1
for x in range(16,20):
    if (resvector1[x]<menor4):
        menor4=resvector1[x]
        posicion4=x-16
if posicion4==0:
    cv2.putText(original,'F',(435,390),
3,(255,255,255),8,cv2.LINE_AA)
    cv2.FONT_HERSHEY_SIMPLEX,
    contf=contf+1
if posicion4==1:
    cv2.putText(original,'M',(435,390),
3,(255,255,255),8,cv2.LINE_AA)
    cv2.FONT_HERSHEY_SIMPLEX,
    contm=contm+1
if posicion4==2:
    cv2.putText(original,'N',(435,390),
3,(255,255,255),8,cv2.LINE_AA)
    cv2.FONT_HERSHEY_SIMPLEX,
    contn=contn+1
if posicion4==3:

```

```

        cv2.putText(original,'P',(435,390),
3,(255,255,255),8,cv2.LINE_AA)
        contp=contp+1
        for x in range(20,24):
            if (resvector1[x]<menor5):
                menor5=resvector1[x]
                posicion5=x-20
        if posicion5==0:
            cv2.putText(original,'F',(760,390),
3,(255,255,255),8,cv2.LINE_AA)
            conf=conf+1
        if posicion5==1:
            cv2.putText(original,'M',(760,390),
3,(255,255,255),8,cv2.LINE_AA)
            contm=contm+1
        if posicion5==2:
            cv2.putText(original,'N',(760,390),
3,(255,255,255),8,cv2.LINE_AA)
            contn=contn+1
        if posicion5==3:
            cv2.putText(original,'P',(760,390),
3,(255,255,255),8,cv2.LINE_AA)
            contp=contp+1
        for x in range(24,28):
            if (resvector1[x]<menor6):
                menor6=resvector1[x]
                posicion6=x-24
        if posicion6==0:
            cv2.putText(original,'F',(100,630),
3,(255,255,255),8,cv2.LINE_AA)
            conf=conf+1
        if posicion6==1:
            cv2.putText(original,'M',(100,630),
3,(255,255,255),8,cv2.LINE_AA)

```

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

```

        contm=contm+1
    if posicion6==2:
        cv2.putText(original,'N',(100,630),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contn=contn+1
    if posicion6==3:
        cv2.putText(original,'P',(100,630),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contp=contp+1
    for x in range(28,32):
        if (resvector1[x]<menor7):
            menor7=resvector1[x]
            posicion7=x-28
        if posicion7==0:
            cv2.putText(original,'F',(435,630),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contf=contf+1
        if posicion7==1:
            cv2.putText(original,'M',(435,630),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contm=contm+1
        if posicion7==2:
            cv2.putText(original,'N',(435,630),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contn=contn+1
        if posicion7==3:
            cv2.putText(original,'P',(435,630),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contp=contp+1
    for x in range(32,36):
        if (resvector1[x]<menor8):
            menor8=resvector1[x]
            posicion8=x-32
        if posicion8==0:

```

```

        cv2.putText(original,'F',(760,630),
3,(255,255,255),8,cv2.LINE_AA)
        contf=contf+1
    if posicion8==1:
        cv2.putText(original,'M',(760,630),
3,(255,255,255),8,cv2.LINE_AA)
        contm=contm+1
    if posicion8==2:
        cv2.putText(original,'N',(760,630),
3,(255,255,255),8,cv2.LINE_AA)
        contn=contn+1
    if posicion8==3:
        cv2.putText(original,'P',(760,630),
3,(255,255,255),8,cv2.LINE_AA)
        contp=contp+1
    for x in range(36,40):
        if (resvector1[x]<menor9):
            menor9=resvector1[x]
            posicion9=x-36
    if posicion9==0:
        cv2.putText(original,'F',(100,890),
3,(255,255,255),8,cv2.LINE_AA)
        contf=contf+1
    if posicion9==1:
        cv2.putText(original,'M',(100,890),
3,(255,255,255),8,cv2.LINE_AA)
        contm=contm+1
    if posicion9==2:
        cv2.putText(original,'N',(100,890),
3,(255,255,255),8,cv2.LINE_AA)
        contn=contn+1
    if posicion9==3:
        cv2.putText(original,'P',(100,890),
3,(255,255,255),8,cv2.LINE_AA)

```

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

```

    contp=contp+1
for x in range(40,44):
    if (resvector1[x]<menor10):
        menor10=resvector1[x]
        posicion10=x-40
    if posicion10==0:
        cv2.putText(original,'F',(435,890),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contf=contf+1
    if posicion10==1:
        cv2.putText(original,'M',(435,890),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contm=contm+1
    if posicion10==2:
        cv2.putText(original,'N',(435,890),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contn=contn+1
    if posicion10==3:
        cv2.putText(original,'P',(435,890),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contp=contp+1
for x in range(44,48):
    if (resvector1[x]<menor11):
        menor11=resvector1[x]
        posicion11=x-44
    if posicion11==0:
        cv2.putText(original,'F',(760,890),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contf=contf+1
    if posicion11==1:
        cv2.putText(original,'M',(760,890),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contm=contm+1
    if posicion11==2:

```

```

        cv2.putText(original,'N',(760,890),
3,(255,255,255),8,cv2.LINE_AA)
        contn=contn+1
        if posicion11==3:
            cv2.putText(original,'P',(760,890),
3,(255,255,255),8,cv2.LINE_AA)
            contp=contp+1
        for x in range(48,52):
            if (resvector1[x]<menor12):
                menor12=resvector1[x]
                posicion12=x-48
            if posicion12==0:
                cv2.putText(original,'F',(100,1140),
3,(255,255,255),8,cv2.LINE_AA)
                contf=contf+1
            if posicion12==1:
                cv2.putText(original,'M',(100,1140),
3,(255,255,255),8,cv2.LINE_AA)
                contm=contm+1
            if posicion12==2:
                cv2.putText(original,'N',(100,1140),
3,(255,255,255),8,cv2.LINE_AA)
                contn=contn+1
            if posicion12==3:
                cv2.putText(original,'P',(100,1140),
3,(255,255,255),8,cv2.LINE_AA)
                contp=contp+1
        for x in range(52,56):
            if (resvector1[x]<menor13):
                menor13=resvector1[x]
                posicion13=x-52
            if posicion13==0:
                cv2.putText(original,'F',(435,1140),
3,(255,255,255),8,cv2.LINE_AA)

```

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

```

    conf=conf+1
    if posicion13==1:
        cv2.putText(original,'M',(435,1140),
3,(255,255,255),8,cv2.LINE_AA)
        contm=contm+1
    if posicion13==2:
        cv2.putText(original,'N',(435,1140),
3,(255,255,255),8,cv2.LINE_AA)
        contn=contn+1
    if posicion13==3:
        cv2.putText(original,'P',(435,1140),
3,(255,255,255),8,cv2.LINE_AA)
        contp=contp+1
    for x in range(56,60):
        if (resvector1[x]<menor14):
            menor14=resvector1[x]
            posicion14=x-56
    if posicion14==0:
        cv2.putText(original,'F',(760,1140),
3,(255,255,255),8,cv2.LINE_AA)
        conf=conf+1
    if posicion14==1:
        cv2.putText(original,'M',(760,1140),
3,(255,255,255),8,cv2.LINE_AA)
        contm=contm+1
    if posicion14==2:
        cv2.putText(original,'N',(760,1140),
3,(255,255,255),8,cv2.LINE_AA)
        contn=contn+1
    if posicion14==3:
        cv2.putText(original,'P',(760,1140),
3,(255,255,255),8,cv2.LINE_AA)
        contp=contp+1
    #cv2.imshow("procesado", cv2.resize(original,(480,640)))

```



```

0,0,0,0,
0,0,0,0]
#FILTRADO IMAGEN ORIGINAL
original = cv2.imread(fechastr2)
gris0 = cv2.cvtColor(original, cv2.COLOR_BGR2GRAY)
gauss0 = cv2.GaussianBlur(gris0, (11,11), 0)
canny0 = cv2.Canny(gauss0, 25, 10) #25,15
#FILTRADO IMAGENES BASE
compx = cv2.imread("cfis.jpg")
grisx = cv2.cvtColor(compx, cv2.COLOR_BGR2GRAY)
gaussx = cv2.GaussianBlur(grisx, (5,5), 0)
cannyx = cv2.Canny(gaussx, 48, 18) #48,18
compx1 = cv2.imread("cman.jpg")
grisx1 = cv2.cvtColor(compx1, cv2.COLOR_BGR2GRAY)
gaussx1 = cv2.GaussianBlur(grisx1, (5,5), 0)
cannyx1 = cv2.Canny(gaussx1, 40, 15) #40,15
compx2 = cv2.imread("cnormal.jpg")
grisx2 = cv2.cvtColor(compx2, cv2.COLOR_BGR2GRAY)
gaussx2 = cv2.GaussianBlur(grisx2, (5,5), 0)
cannyx2 = cv2.Canny(gaussx2, 100, 25) #100,25
compx3 = cv2.imread("cporos.jpg")
grisx3 = cv2.cvtColor(compx3, cv2.COLOR_BGR2GRAY)
gaussx3 = cv2.GaussianBlur(grisx3, (15,15), 0)
cannyx3 = cv2.Canny(gaussx3, 24, 10) #24,10
#COMPARACION INDIVIDUAL DE CADA IMAGEN
resvector1[0]=cv2.matchShapes(cannyx,canny0[20:290,0:320],1,0.0)#1
resvector1[1]=cv2.matchShapes(cannyx1,canny0[20:290,0:320],1,0.0)
resvector1[2]=cv2.matchShapes(cannyx2,canny0[20:290,0:320],1,0.0)
resvector1[3]=cv2.matchShapes(cannyx3,canny0[20:290,0:320],1,0.0)
#cv2.imshow("test1", canny0[20:290,0:320])
resvector1[4]=cv2.matchShapes(cannyx,canny0[20:290,320:645],1,0.0)#2
resvector1[5]=cv2.matchShapes(cannyx1,canny0[20:290,320:645],1,0.0)
resvector1[6]=cv2.matchShapes(cannyx2,canny0[20:290,320:645],1,0.0)
resvector1[7]=cv2.matchShapes(cannyx3,canny0[20:290,320:645],1,0.0)

```

```

#cv2.imshow("test2", canny0[20:290,320:645])
resvector1[8]=cv2.matchShapes(cannyx,canny0[20:290,645:990],1,0.0)#3
resvector1[9]=cv2.matchShapes(cannyx1,canny0[20:290,645:990],1,0.0)
resvector1[10]=cv2.matchShapes(cannyx2,canny0[20:290,645:990],1,0.0)
resvector1[11]=cv2.matchShapes(cannyx3,canny0[20:290,645:990],1,0.0)
#cv2.imshow("test3", canny0[20:290,645:990])
resvector1[12]=cv2.matchShapes(cannyx,canny0[290:530,0:320],1,0.0)#4
resvector1[13]=cv2.matchShapes(cannyx1,canny0[290:530,0:320],1,0.0)
resvector1[14]=cv2.matchShapes(cannyx2,canny0[290:530,0:320],1,0.0)
resvector1[15]=cv2.matchShapes(cannyx3,canny0[290:530,0:320],1,0.0)
#cv2.imshow("test4", canny0[290:530,0:320])
resvector1[16]=cv2.matchShapes(cannyx,canny0[290:530,320:645],1,0.0)#5
resvector1[17]=cv2.matchShapes(cannyx1,canny0[290:530,320:645],1,0.0)
resvector1[18]=cv2.matchShapes(cannyx2,canny0[290:530,320:645],1,0.0)
resvector1[19]=cv2.matchShapes(cannyx3,canny0[290:530,320:645],1,0.0)
#cv2.imshow("test5", canny0[290:530,320:645])
resvector1[20]=cv2.matchShapes(cannyx,canny0[290:530,645:990],1,0.0)#6
resvector1[21]=cv2.matchShapes(cannyx1,canny0[290:530,645:990],1,0.0)
resvector1[22]=cv2.matchShapes(cannyx2,canny0[290:530,645:990],1,0.0)
resvector1[23]=cv2.matchShapes(cannyx3,canny0[290:530,645:990],1,0.0)
#cv2.imshow("test6", canny0[290:530,645:990])
resvector1[24]=cv2.matchShapes(cannyx,canny0[530:790,0:320],1,0.0)#7
resvector1[25]=cv2.matchShapes(cannyx1,canny0[530:790,0:320],1,0.0)
resvector1[26]=cv2.matchShapes(cannyx2,canny0[530:790,0:320],1,0.0)
resvector1[27]=cv2.matchShapes(cannyx3,canny0[530:790,0:320],1,0.0)
#cv2.imshow("test7", canny0[530:790,0:320])
resvector1[28]=cv2.matchShapes(cannyx,canny0[530:790,320:645],1,0.0)#8
resvector1[29]=cv2.matchShapes(cannyx1,canny0[530:790,320:645],1,0.0)
resvector1[30]=cv2.matchShapes(cannyx2,canny0[530:790,320:645],1,0.0)
resvector1[31]=cv2.matchShapes(cannyx3,canny0[530:790,320:645],1,0.0)
#cv2.imshow("test8", canny0[530:790,320:645])
resvector1[32]=cv2.matchShapes(cannyx,canny0[530:790,645:990],1,0.0)#9
resvector1[33]=cv2.matchShapes(cannyx1,canny0[530:790,645:990],1,0.0)
resvector1[34]=cv2.matchShapes(cannyx2,canny0[530:790,645:990],1,0.0)

```

```

resvector1[35]=cv2.matchShapes(cannyx3,canny0[530:790,645:990],1,0.0)
#cv2.imshow("test9", canny0[530:790,645:990])
resvector1[36]=cv2.matchShapes(cannyx,canny0[790:1040,0:320],1,0.0)#10
resvector1[37]=cv2.matchShapes(cannyx1,canny0[790:1040,0:320],1,0.0)
resvector1[38]=cv2.matchShapes(cannyx2,canny0[790:1040,0:320],1,0.0)
resvector1[39]=cv2.matchShapes(cannyx3,canny0[790:1040,0:320],1,0.0)
#cv2.imshow("test10", canny0[790:1040,0:320])
resvector1[40]=cv2.matchShapes(cannyx,canny0[790:1040,320:645],1,0.0)#11
resvector1[41]=cv2.matchShapes(cannyx1,canny0[790:1040,320:645],1,0.0)
resvector1[42]=cv2.matchShapes(cannyx2,canny0[790:1040,320:645],1,0.0)
resvector1[43]=cv2.matchShapes(cannyx3,canny0[790:1040,320:645],1,0.0)
#cv2.imshow("test11", canny0[790:1040,320:645])
resvector1[44]=cv2.matchShapes(cannyx,canny0[790:1040,645:990],1,0.0)#12
resvector1[45]=cv2.matchShapes(cannyx1,canny0[790:1040,645:990],1,0.0)
resvector1[46]=cv2.matchShapes(cannyx2,canny0[790:1040,645:990],1,0.0)
resvector1[47]=cv2.matchShapes(cannyx3,canny0[790:1040,645:990],1,0.0)
#cv2.imshow("test12", canny0[790:1040,645:990])
resvector1[48]=cv2.matchShapes(cannyx,canny0[1040:1325,0:320],1,0.0)#13
resvector1[49]=cv2.matchShapes(cannyx1,canny0[1040:1325,0:320],1,0.0)
resvector1[50]=cv2.matchShapes(cannyx2,canny0[1040:1325,0:320],1,0.0)
resvector1[51]=cv2.matchShapes(cannyx3,canny0[1040:1325,0:320],1,0.0)
#cv2.imshow("test13", canny0[1040:1325,0:320])
resvector1[52]=cv2.matchShapes(cannyx,canny0[1040:1325,320:645],1,0.0)#14
resvector1[53]=cv2.matchShapes(cannyx1,canny0[1040:1325,320:645],1,0.0)
resvector1[54]=cv2.matchShapes(cannyx2,canny0[1040:1325,320:645],1,0.0)
resvector1[55]=cv2.matchShapes(cannyx3,canny0[1040:1325,320:645],1,0.0)
#cv2.imshow("test14", canny0[1040:1325,320:645])
resvector1[56]=cv2.matchShapes(cannyx,canny0[1040:1325,645:990],1,0.0)#15
resvector1[57]=cv2.matchShapes(cannyx1,canny0[1040:1325,645:990],1,0.0)
resvector1[58]=cv2.matchShapes(cannyx2,canny0[1040:1325,645:990],1,0.0)
resvector1[59]=cv2.matchShapes(cannyx3,canny0[1040:1325,645:990],1,0.0)
#cv2.imshow("test15", canny0[1040:1325,645:990])
#BUSCAR VALOR MENOR
menor=resvector1[0]

```

```
menor1=resvector1[4]
menor2=resvector1[8]
menor3=resvector1[12]
menor4=resvector1[16]
menor5=resvector1[20]
menor6=resvector1[24]
menor7=resvector1[28]
menor8=resvector1[32]
menor9=resvector1[36]
menor10=resvector1[40]
menor11=resvector1[44]
menor12=resvector1[48]
menor13=resvector1[52]
menor14=resvector1[56]
posicion=0
posicion1=0
posicion2=0
posicion3=0
posicion4=0
posicion5=0
posicion6=0
posicion7=0
posicion8=0
posicion9=0
posicion10=0
posicion11=0
posicion12=0
posicion13=0
posicion14=0
contf=0
contm=0
contn=0
contp=0
for x in range(0,4):
```

```

    if (resvector1[x]<menor):
        menor=resvector1[x]
        posicion=x
    if posicion==0:
        cv2.putText(original,'F',(100,120),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contf=contf+1
    if posicion==1:
        cv2.putText(original,'M',(100,120),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contm=contm+1
    if posicion==2:
        cv2.putText(original,'N',(100,120),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contn=contn+1
    if posicion==3:
        cv2.putText(original,'P',(100,120),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contp=contp+1
    for x in range(4,8):
        if (resvector1[x]<menor1):
            menor1=resvector1[x]
            posicion1=x-4
        if posicion1==0:
            cv2.putText(original,'F',(435,120),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contf=contf+1
        if posicion1==1:
            cv2.putText(original,'M',(435,120),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contm=contm+1
        if posicion1==2:
            cv2.putText(original,'N',(435,120),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,

```

```

    contn=contn+1
    if posicion1==3:
        cv2.putText(original,'P',(435,120),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contp=contp+1
    for x in range(8,12):
        if (resvector1[x]<menor2):
            menor2=resvector1[x]
            posicion2=x-8
        if posicion2==0:
            cv2.putText(original,'F',(760,120),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contf=contf+1
        if posicion2==1:
            cv2.putText(original,'M',(760,120),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contm=contm+1
        if posicion2==2:
            cv2.putText(original,'N',(760,120),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contn=contn+1
        if posicion2==3:
            cv2.putText(original,'P',(760,120),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contp=contp+1
    for x in range(12,16):
        if (resvector1[x]<menor3):
            menor3=resvector1[x]
            posicion3=x-12
        if posicion3==0:
            cv2.putText(original,'F',(100,390),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contf=contf+1
        if posicion3==1:

```

```

        cv2.putText(original,'M',(100,390),
3,(255,255,255),8,cv2.LINE_AA
        contm=contm+1
    if posicion3==2:
        cv2.putText(original,'N',(100,390),
3,(255,255,255),8,cv2.LINE_AA
        contn=contn+1
    if posicion3==3:
        cv2.putText(original,'P',(100,390),
3,(255,255,255),8,cv2.LINE_AA
        contp=contp+1
    for x in range(16,20):
        if (resvector1[x]<menor4):
            menor4=resvector1[x]
            posicion4=x-16
    if posicion4==0:
        cv2.putText(original,'F',(435,390),
3,(255,255,255),8,cv2.LINE_AA
        conf=conf+1
    if posicion4==1:
        cv2.putText(original,'M',(435,390),
3,(255,255,255),8,cv2.LINE_AA
        contm=contm+1
    if posicion4==2:
        cv2.putText(original,'N',(435,390),
3,(255,255,255),8,cv2.LINE_AA
        contn=contn+1
    if posicion4==3:
        cv2.putText(original,'P',(435,390),
3,(255,255,255),8,cv2.LINE_AA
        contp=contp+1
    for x in range(20,24):
        if (resvector1[x]<menor5):
            menor5=resvector1[x]

```

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

```

    posicion5=x-20
    if posicion5==0:
        cv2.putText(original,'F',(760,390),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contf=contf+1
    if posicion5==1:
        cv2.putText(original,'M',(760,390),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contm=contm+1
    if posicion5==2:
        cv2.putText(original,'N',(760,390),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contn=contn+1
    if posicion5==3:
        cv2.putText(original,'P',(760,390),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contp=contp+1
    for x in range(24,28):
        if (resvector1[x]<menor6):
            menor6=resvector1[x]
            posicion6=x-24
        if posicion6==0:
            cv2.putText(original,'F',(100,630),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contf=contf+1
        if posicion6==1:
            cv2.putText(original,'M',(100,630),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contm=contm+1
        if posicion6==2:
            cv2.putText(original,'N',(100,630),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contn=contn+1
        if posicion6==3:

```

```

        cv2.putText(original,'P',(100,630),
3,(255,255,255),8,cv2.LINE_AA
        contp=contp+1
    for x in range(28,32):
        if (resvector1[x]<menor7):
            menor7=resvector1[x]
            posicion7=x-28
    if posicion7==0:
        cv2.putText(original,'F',(435,630),
3,(255,255,255),8,cv2.LINE_AA
        conf=conf+1
    if posicion7==1:
        cv2.putText(original,'M',(435,630),
3,(255,255,255),8,cv2.LINE_AA
        contm=contm+1
    if posicion7==2:
        cv2.putText(original,'N',(435,630),
3,(255,255,255),8,cv2.LINE_AA
        contn=contn+1
    if posicion7==3:
        cv2.putText(original,'P',(435,630),
3,(255,255,255),8,cv2.LINE_AA
        contp=contp+1
    for x in range(32,36):
        if (resvector1[x]<menor8):
            menor8=resvector1[x]
            posicion8=x-32
    if posicion8==0:
        cv2.putText(original,'F',(760,630),
3,(255,255,255),8,cv2.LINE_AA
        conf=conf+1
    if posicion8==1:
        cv2.putText(original,'M',(760,630),
3,(255,255,255),8,cv2.LINE_AA

```

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

```

        contm=contm+1
    if posicion8==2:
        cv2.putText(original,'N',(760,630),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contn=contn+1
    if posicion8==3:
        cv2.putText(original,'P',(760,630),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contp=contp+1
    for x in range(36,40):
        if (resvector1[x]<menor9):
            menor9=resvector1[x]
            posicion9=x-36
        if posicion9==0:
            cv2.putText(original,'F',(100,890),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contf=contf+1
        if posicion9==1:
            cv2.putText(original,'M',(100,890),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contm=contm+1
        if posicion9==2:
            cv2.putText(original,'N',(100,890),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contn=contn+1
        if posicion9==3:
            cv2.putText(original,'P',(100,890),
3,(255,255,255),8,cv2.LINE_AA)
            cv2.FONT_HERSHEY_SIMPLEX,
            contp=contp+1
    for x in range(40,44):
        if (resvector1[x]<menor10):
            menor10=resvector1[x]
            posicion10=x-40
        if posicion10==0:

```

```

        cv2.putText(original,'F',(435,890),
3,(255,255,255),8,cv2.LINE_AA)
        contf=contf+1
    if posicion10==1:
        cv2.putText(original,'M',(435,890),
3,(255,255,255),8,cv2.LINE_AA)
        contm=contm+1
    if posicion10==2:
        cv2.putText(original,'N',(435,890),
3,(255,255,255),8,cv2.LINE_AA)
        contn=contn+1
    if posicion10==3:
        cv2.putText(original,'P',(435,890),
3,(255,255,255),8,cv2.LINE_AA)
        contp=contp+1
    for x in range(44,48):
        if (resvector1[x]<menor11):
            menor11=resvector1[x]
            posicion11=x-44
    if posicion11==0:
        cv2.putText(original,'F',(760,890),
3,(255,255,255),8,cv2.LINE_AA)
        contf=contf+1
    if posicion11==1:
        cv2.putText(original,'M',(760,890),
3,(255,255,255),8,cv2.LINE_AA)
        contm=contm+1
    if posicion11==2:
        cv2.putText(original,'N',(760,890),
3,(255,255,255),8,cv2.LINE_AA)
        contn=contn+1
    if posicion11==3:
        cv2.putText(original,'P',(760,890),
3,(255,255,255),8,cv2.LINE_AA)

```

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

cv2.FONT_HERSHEY_SIMPLEX,

```

    contp=contp+1
for x in range(48,52):
    if (resvector1[x]<menor12):
        menor12=resvector1[x]
        posicion12=x-48
    if posicion12==0:
        cv2.putText(original,'F',(100,1140),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contf=contf+1
    if posicion12==1:
        cv2.putText(original,'M',(100,1140),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contm=contm+1
    if posicion12==2:
        cv2.putText(original,'N',(100,1140),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contn=contn+1
    if posicion12==3:
        cv2.putText(original,'P',(100,1140),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contp=contp+1
for x in range(52,56):
    if (resvector1[x]<menor13):
        menor13=resvector1[x]
        posicion13=x-52
    if posicion13==0:
        cv2.putText(original,'F',(435,1140),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contf=contf+1
    if posicion13==1:
        cv2.putText(original,'M',(435,1140),
3,(255,255,255),8,cv2.LINE_AA)
        cv2.FONT_HERSHEY_SIMPLEX,
        contm=contm+1
    if posicion13==2:

```

```

        cv2.putText(original,'N',(435,1140),
3,(255,255,255),8,cv2.LINE_AA)
        contn=contn+1
        if posicion13==3:
            cv2.putText(original,'P',(435,1140),
3,(255,255,255),8,cv2.LINE_AA)
            contp=contp+1
        for x in range(56,60):
            if (resvector1[x]<menor14):
                menor14=resvector1[x]
                posicion14=x-56
            if posicion14==0:
                cv2.putText(original,'F',(760,1140),
3,(255,255,255),8,cv2.LINE_AA)
                contf=contf+1
            if posicion14==1:
                cv2.putText(original,'M',(760,1140),
3,(255,255,255),8,cv2.LINE_AA)
                contm=contm+1
            if posicion14==2:
                cv2.putText(original,'N',(760,1140),
3,(255,255,255),8,cv2.LINE_AA)
                contn=contn+1
            if posicion14==3:
                cv2.putText(original,'P',(760,1140),
3,(255,255,255),8,cv2.LINE_AA)
                contp=contp+1
        #cv2.imshow("procesado", cv2.resize(original,(480,640)))
        self.basefisuras=self.basefisuras+contf
        self.basemanchas=self.basemanchas+contm
        self.basenormal=self.basenormal+contn
        self.baseporos=self.baseporos+contp
        #print(contf," ",contm," ",contn," ",contp)
        #print(self.basefisuras," ",self.basemanchas," ",self.basenormal," ",self.baseporos)

```

```

#####
#####BASE DE DATOS#####
conn = pymysql.connect(
    host="localhost", user="admin",
    passwd="12345", db="clasificar"
)
mycursor = conn.cursor()
sql = "INSERT INTO analisis (NORMALES, POROSOS, FISURADOS, MARCAS) VALUES (%s, %s, %s,
%s)"
val = (self.basenormal, self.baseporos, self.basefisuras, self.basemanchas)
mycursor.execute(sql, val)
conn.commit()
conn.close()
#####
self.basefisuras=0
self.basemanchas=0
self.basenormal=0
self.baseporos=0
cv2.imwrite("proc"+fechastr2,original)
rgb_imagen3 = cv2.cvtColor(original, cv2.COLOR_BGR2RGB)
al3, an3, bypp3 = rgb_imagen3.shape
bytes_por_linea3 = bypp3 * an3
im_cvaqt2 = QtGui.QImage(rgb_imagen3.data, an3, al3, bytes_por_linea3,
QtGui.QImage.Format_RGB888)
vis3 = im_cvaqt2.scaled((self.size()/1.35), Qt.KeepAspectRatio)
return QPixmap.fromImage(vis3)

if __name__=="__main__": #bloque para ejecutar solo los métodos que sean llamados
    app = QApplication(sys.argv) #se inicializa la aplicación con todos los argumentos requeridos por el
módulo Qt
    a = aplicacion() #se llama y ejecuta la clase aplicacion
    a.showMaximized() #se inicia la ventana maximizada de la clase aplicacion
    sys.exit(app.exec_()) #indica que el sistema se cerrará en función de los eventos establecidos

```

BASE DE DATOS

```

import pymysql

def basedatos():

    conn = pymysql.connect(

        host="localhost", user="root",

        passwd="ROOT ", db="control"

    )

    mycursor = conn.cursor()

    sql = "INSERT INTO metodos (Normal, Poros, Fisuras, Manchas) VALUES (%s, %s, %s, %s)"

    val = ("14", "6", "2", "8")

    mycursor.execute(sql, val)

    conn.commit()

    conn.close()

```

MÉTRICAS DE COMPARACIÓN

```

import cv2 #librería opencv requerida para las comparaciones y procesamiento de imágenes

resvector=[0,0,0,0] #vector para alojar los resultados

original = cv2.imread("poros1.jpg") #se carga la imagen

#en cada proceso a continuación se agrega la imagen previa procesada sucesivamente

gris0 = cv2.cvtColor(original, cv2.COLOR_BGR2GRAY) #se convierte la imagen a escala de grises

#se aplica un filtro gaussiano para eliminar ruido don (5,5) es el kernel de atenuación, es decir

#los píxeles que se agrupan para difuminar en cada imagen y el 0 es un valor por defecto

gauss0 = cv2.GaussianBlur(gris0, (5,5), 0)

#el método Canny consiste en detercar bordes y características de las imágenes que puede

#considerarse también como filtro complementario que elimina características despreciables

#o no eliminado en el proceso previo de acuerdo a los umbrales (máximo,mínimo), estos umbrales

#se calibran de acuerdo a los análisis que se realizan hasta establecer un valor estándar

```

```

canny0 = cv2.Canny(gauss0, 50, 18)

#los parámetros de las imágenes contenidas a continuación son fijos, calibrados para obtener las
#características más precisas de las cáscaras de acuerdo a los requerimientos

compx = cv2.imread("cfis.jpg")

grisx = cv2.cvtColor(compx, cv2.COLOR_BGR2GRAY)

gaussx = cv2.GaussianBlur(grisx, (5,5), 0)

cannyx = cv2.Canny(gaussx, 48, 18)

compx1 = cv2.imread("cman.jpg")

grisx1 = cv2.cvtColor(compx1, cv2.COLOR_BGR2GRAY)

gaussx1 = cv2.GaussianBlur(grisx1, (5,5), 0)

cannyx1 = cv2.Canny(gaussx1, 40, 15)

compx2 = cv2.imread("cnormal.jpg")

grisx2 = cv2.cvtColor(compx2, cv2.COLOR_BGR2GRAY)

gaussx2 = cv2.GaussianBlur(grisx2, (5,5), 0)

cannyx2 = cv2.Canny(gaussx2, 100, 25)

compx3 = cv2.imread("cporos.jpg")

grisx3 = cv2.cvtColor(compx3, cv2.COLOR_BGR2GRAY)

gaussx3 = cv2.GaussianBlur(grisx3, (5,5), 0)

cannyx3 = cv2.Canny(gaussx3, 24, 10)

#a continuación se comparan 2 imágenes procesadas después de pasar por el filtro canny
#con el método matchShapes asignando valores por defecto y únicamente considerando o
#cambiando las imágenes a comparar

fisura=cv2.matchShapes(cannyx,canny0,1,0.0)

mancha=cv2.matchShapes(cannyx1,canny0,1,0.0)

normal=cv2.matchShapes(cannyx2,canny0,1,0.0)

```

```

poros=cv2.matchShapes(cannyx3,canny0,1,0.0)

#se cargan los resultados de cada comparación

resvector[0]=fisura

resvector[1]=mancha

resvector[2]=normal

resvector[3]=poros

print(resvector) #imprime por consola los valores contenidos en el vector

#la comparación que arroje un valor más cercano a 0 corresponde a la característica detectada

cv2.imshow("original", canny0)

cv2.imshow("fisura", cannyx)

cv2.imshow("mancha", cannyx1)

cv2.imshow("normal", cannyx2)

cv2.imshow("poros", cannyx3)

cv2.waitKey(0)

cv2.destroyAllWindows()

```

ALINEACIÓN

```

import cv2

# test=cv2.imread('02.jpg') #segmentar 1980x2650

#   segm   =   cv2.line(test,   (625,0),   (625,2650),   (0,   255,   0),   2)
#primera(670,0),(670,2650)segunda(625,0),(625,2650)

#   segm1  =   cv2.line(segm,   (1280,0),   (1280,2650),   (0,   255,   0),   2)
#primera(1320,0),(1320,2650)segunda(1280,0),(1280,2650)

# segm2 = cv2.line(segm1, (0,580), (1980,580), (0, 255, 0), 2) #todo(0,580),(1980,580)

# segm3 = cv2.line(segm2, (0,1060), (1980,1060), (0, 255, 0), 2) #todo(0,1060),(1980,1060)

# segm4 = cv2.line(segm3, (0,1580), (1980,1580), (0, 255, 0), 2) #todo(0,1580),(1980,1580)

```

```

# segm5 = cv2.line(segm4, (0,2080), (1980,2080), (0, 255, 0), 2) #todo(0,2080),(1980,2080)

test=cv2.imread('07092022_171959.jpg') #segmentar 990x1325

segm      =      cv2.line(test,      (320,0),      (320,1325),      (0,      255,      0),      2)
#primera(335,0),(335,1325)segunda(320,0),(320,1325)

segm1     =     cv2.line(segm,     (645,0),     (645,1325),     (0,     255,     0),     2)
#primera(660,0),(660,1325)segunda(645,0),(645,1325)

segm2 = cv2.line(segm1, (0,290), (990,290), (0, 255, 0), 2) #todo(0,290), (990,290)

segm3 = cv2.line(segm2, (0,530), (990,530), (0, 255, 0), 2) #todo(0,530), (990,530)

segm4 = cv2.line(segm3, (0,790), (990,790), (0, 255, 0), 2) #todo(0,790), (990,790)

segm5 = cv2.line(segm4, (0,1040), (990,1040), (0, 255, 0), 2) #todo(0,1040), (990,1040)

salida=cv2.resize(segm1, (480, 640))

cv2.imshow('ok',salida)

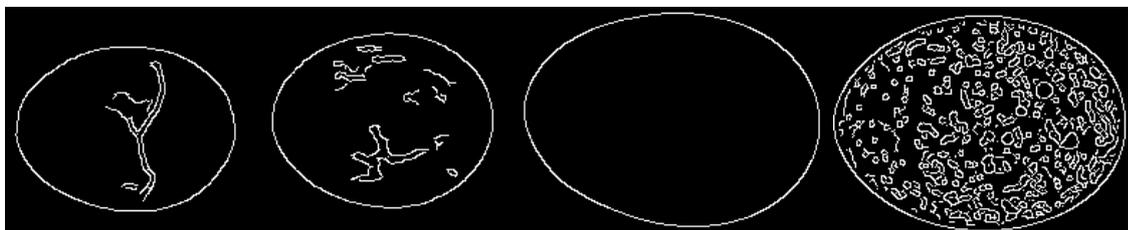
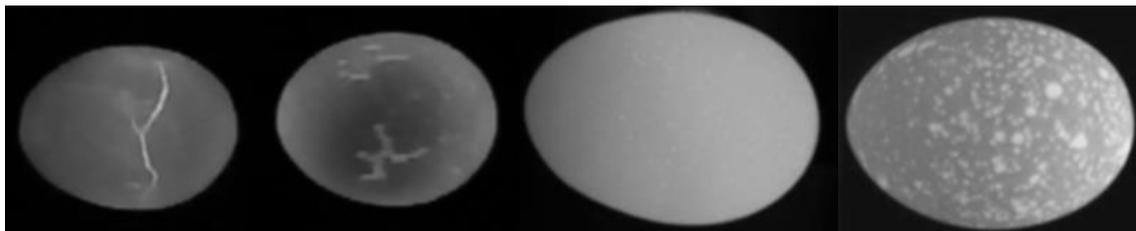
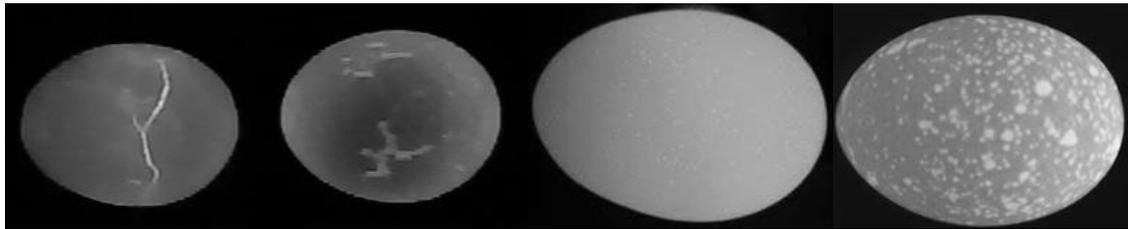
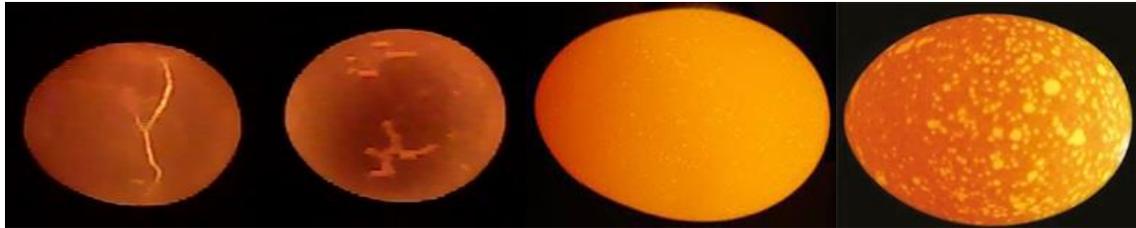
cv2.waitKey(0)

cv2.destroyAllWindows()

```

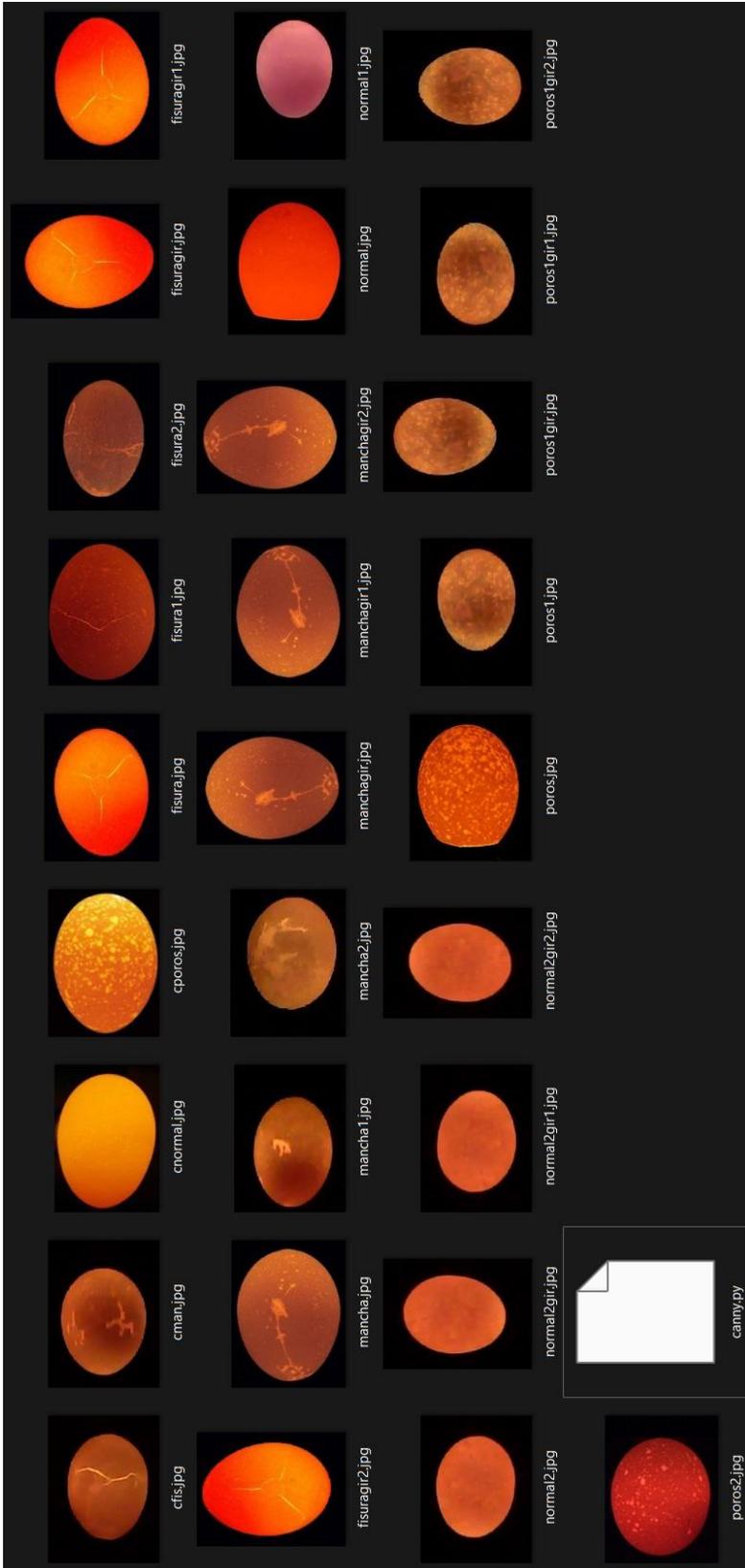
ANEXO 4

IMÁGENES BASE TOMADAS PARA ADQUISICIÓN DE CARACTERÍSTICAS DE LA CÁSCARA DE LOS HUEVOS DE GALLINA



ANEXO 5

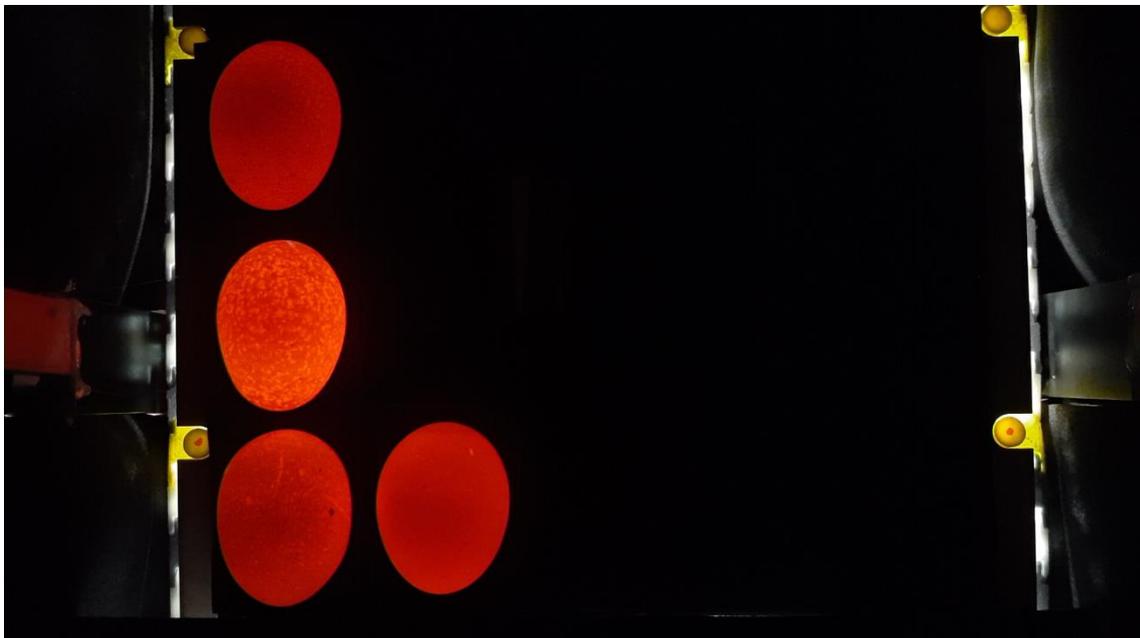
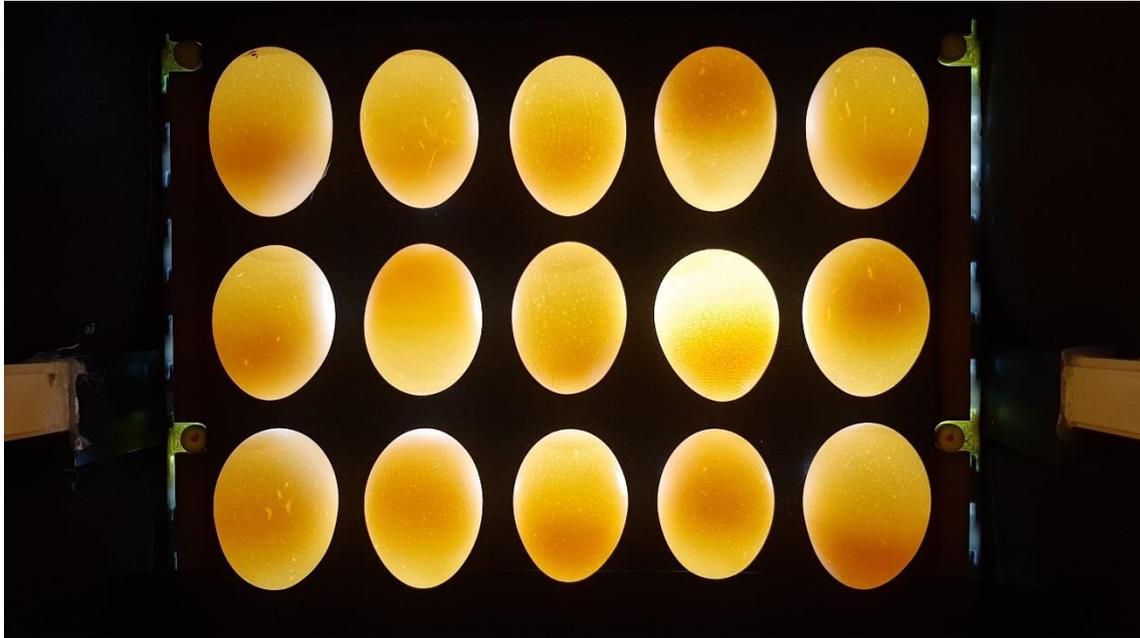
SET DE IMÁGENES BASE TOMADAS PARA TESTEAR, CONFIGURACIÓN DE PARÁMETROS Y ANÁLISIS DE MÉTRICAS DE COMPARACIÓN



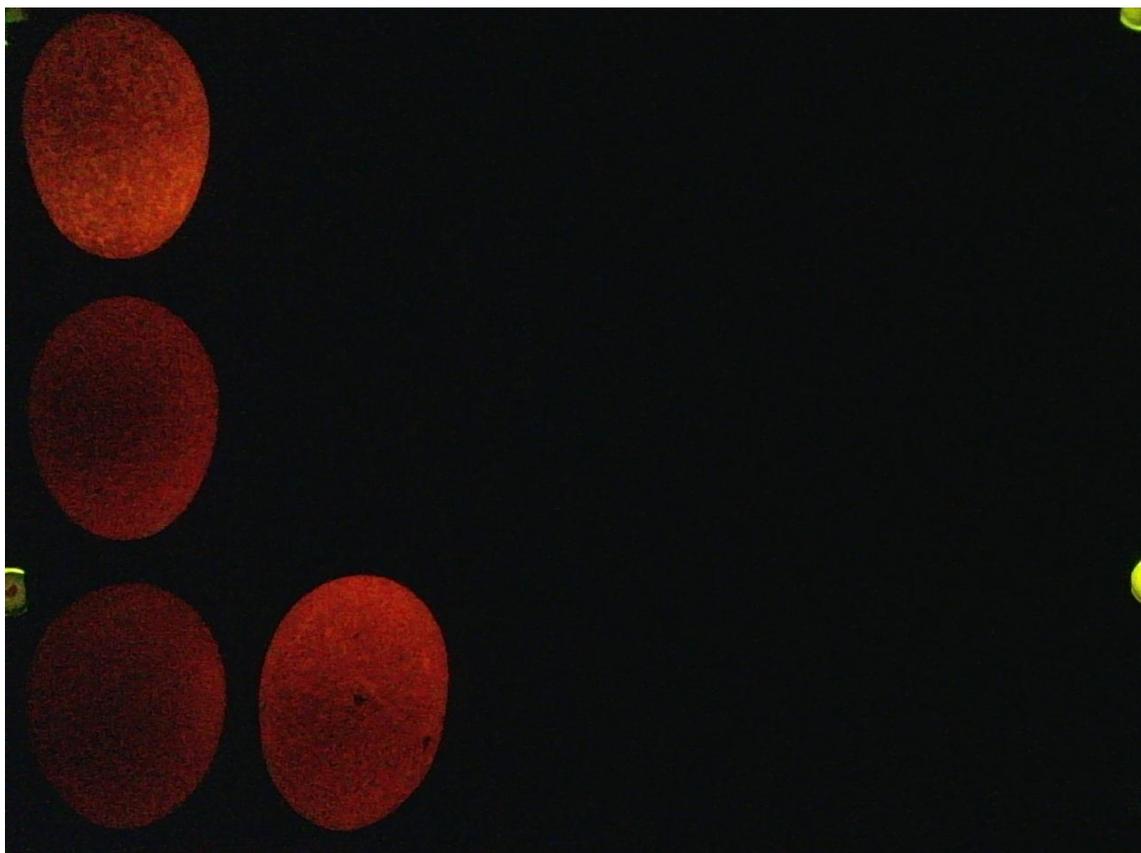
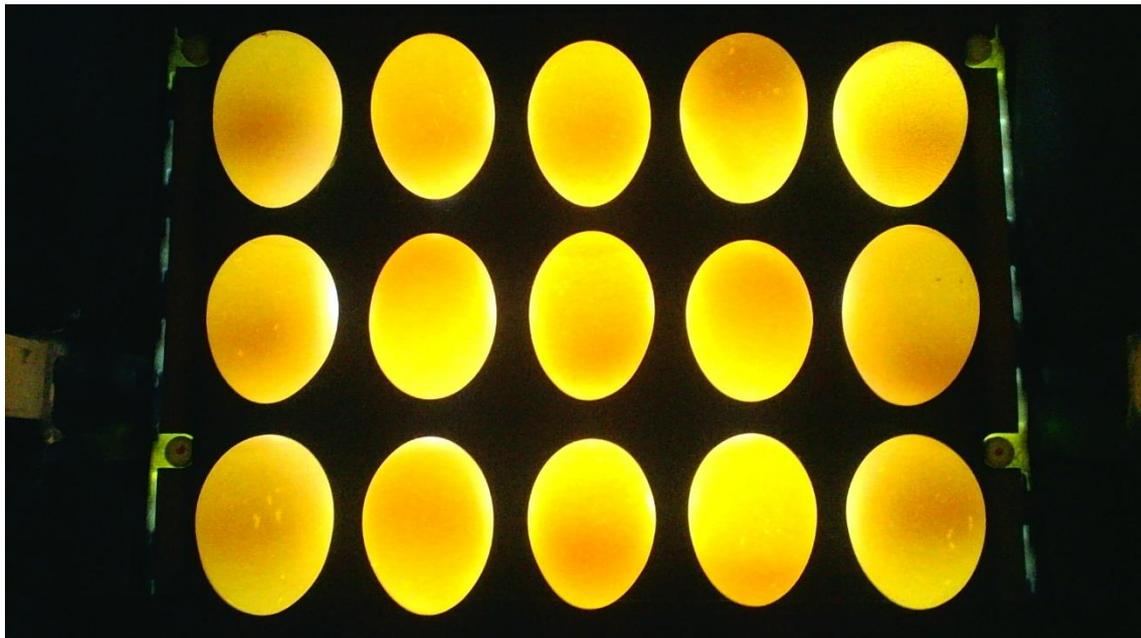
ANEXO 6

**CAPTURAS REALIZADAS CON DISTINTAS CÁMARAS DIGITALES EMPLEANDO EL SOFTWARE
DESARROLLADO**

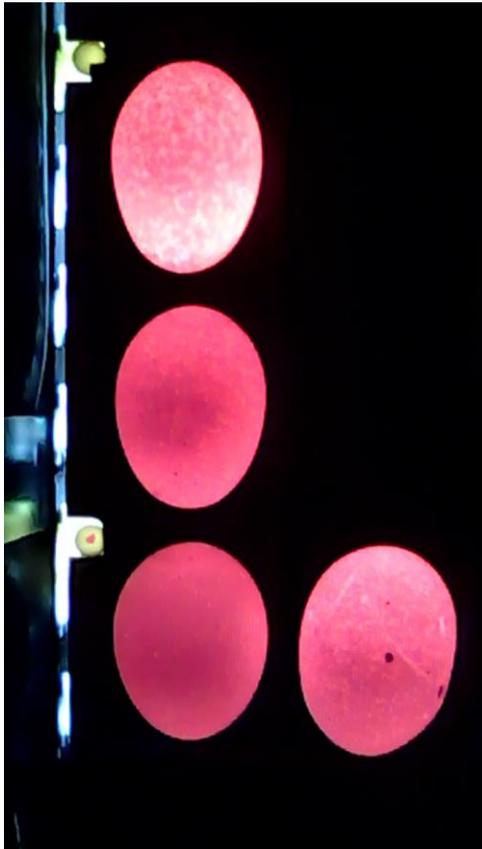
SAMSUNG GALAXY A50



GENÉRICA FULL HD 1080P

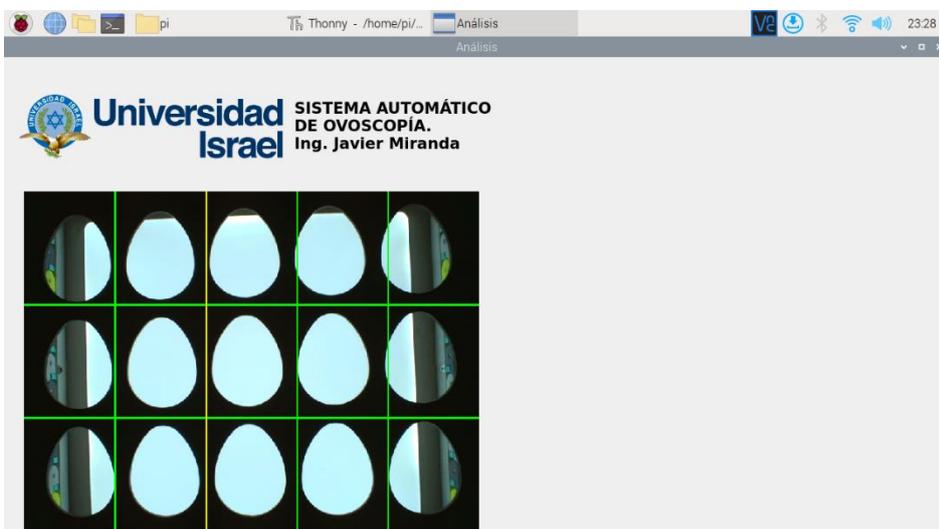
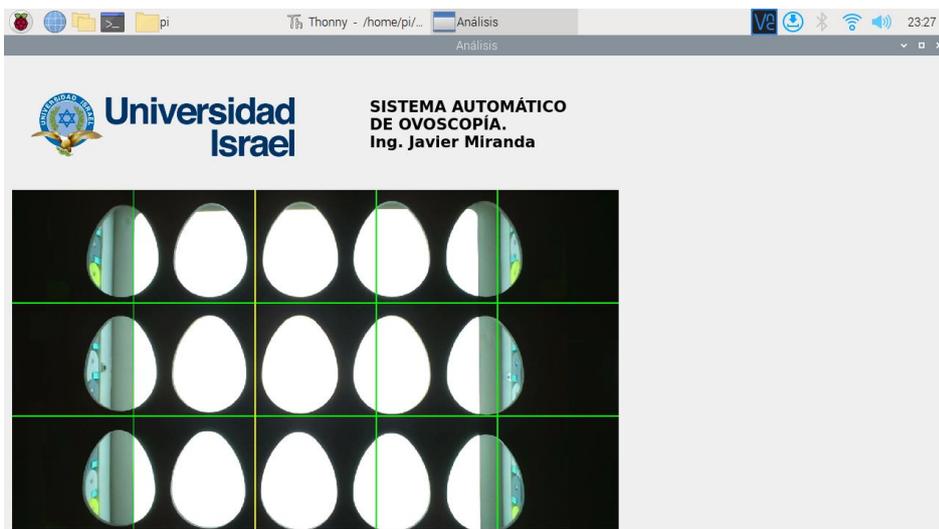


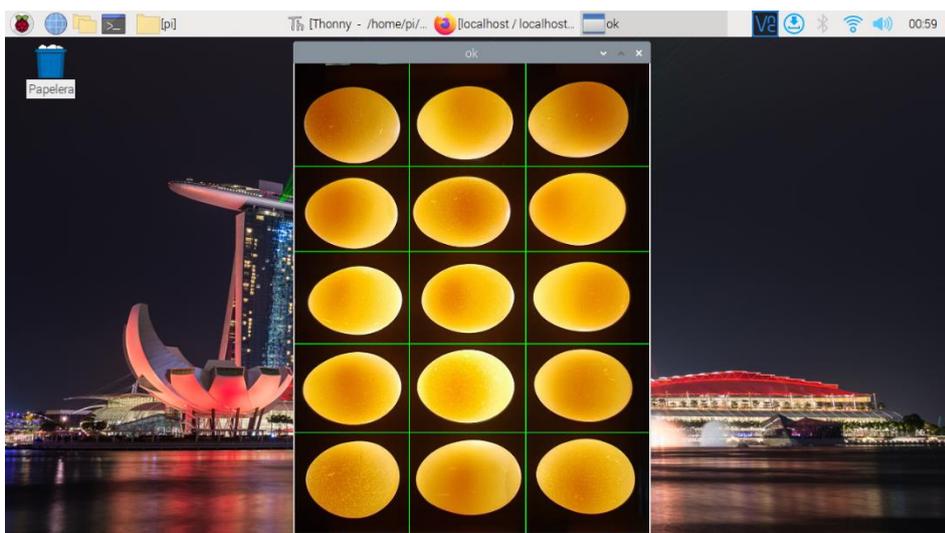
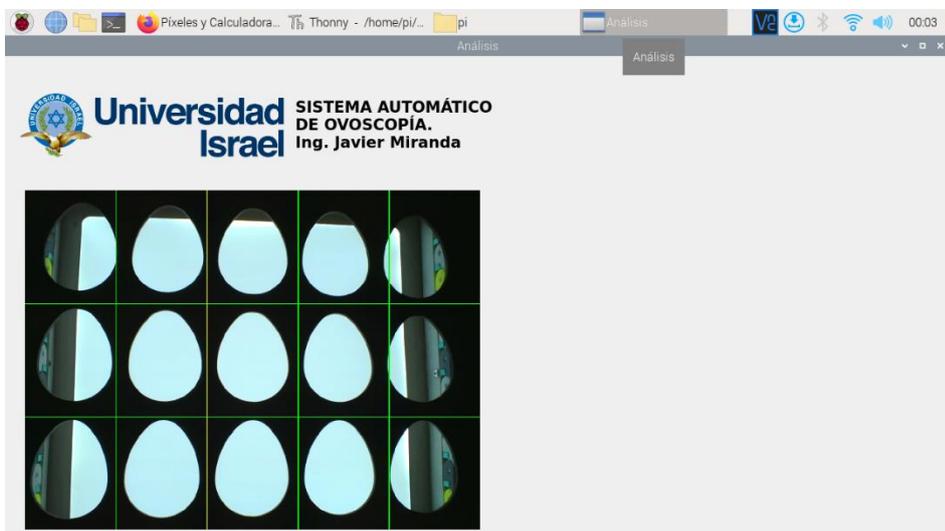
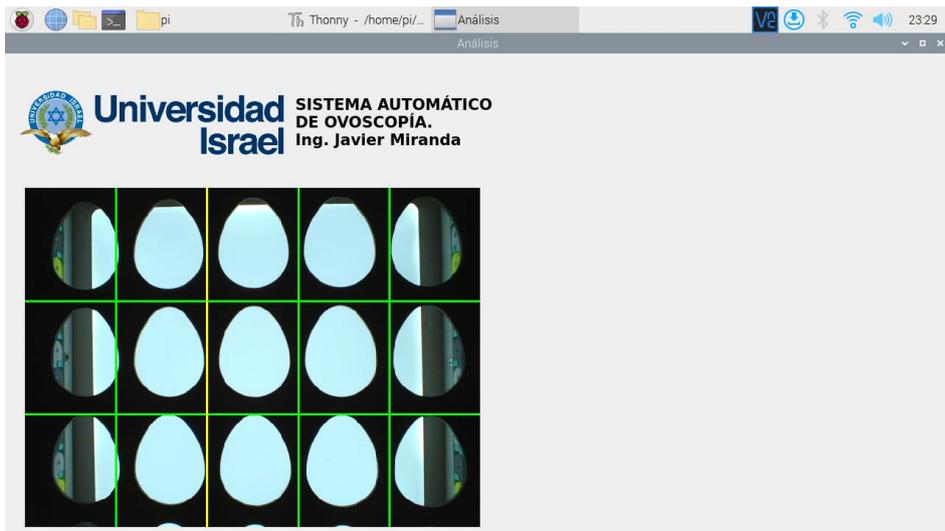
GENIUS FACECAM 1000X



ANEXO 7

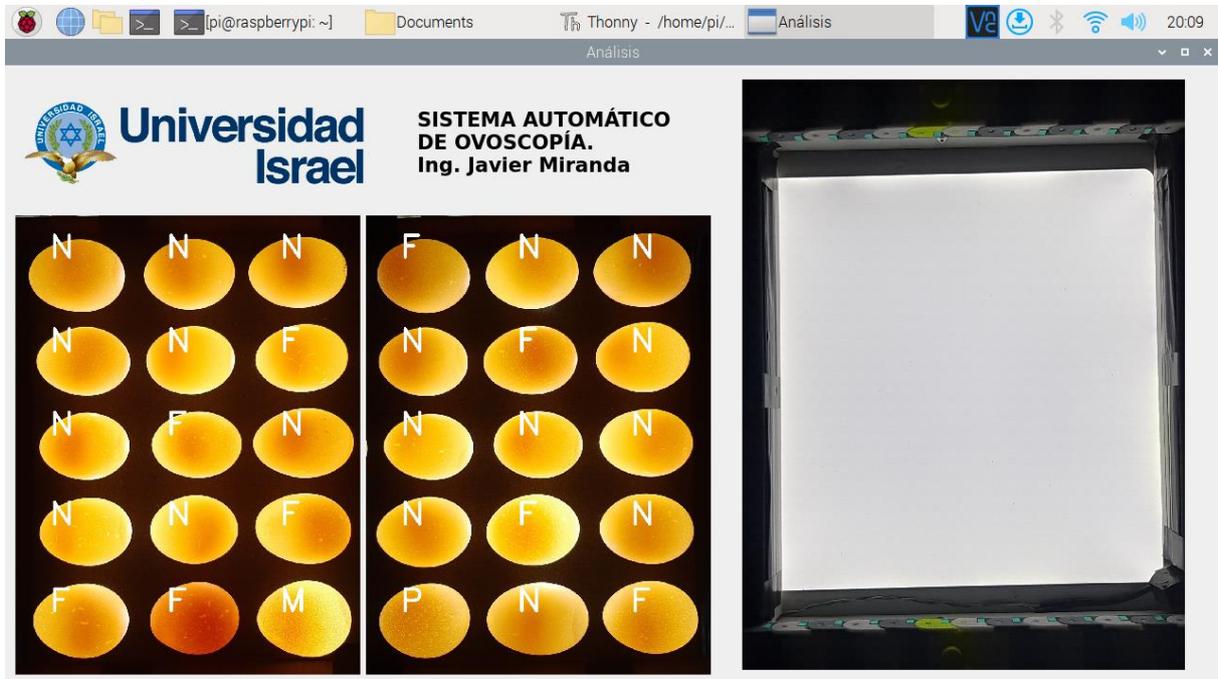
ALINEACIÓN DE CÁMARA





ANEXO 8

REPETICIONES DE ANÁLISIS REALIZADOS DE UNA MISMA CUBETA CON VISUALIZACIÓN EN LA INTERFAZ DESARROLLADA E IMÁGENES ALMACENADAS CUYO NOMBRE ES LA FECHA Y HORA DE CAPTURA



[pi@raspberrypi: ~] pi Thonny - /home/pi/... Análisis 20:17

Universidad Israel **SISTEMA AUTOMÁTICO DE OVOSCOPIA.**
Ing. Javier Miranda

N	N	N	N	N	N
N	N	F	N	F	N
N	F	N	N	N	N
N	N	F	N	F	N
F	F	M	P	N	F

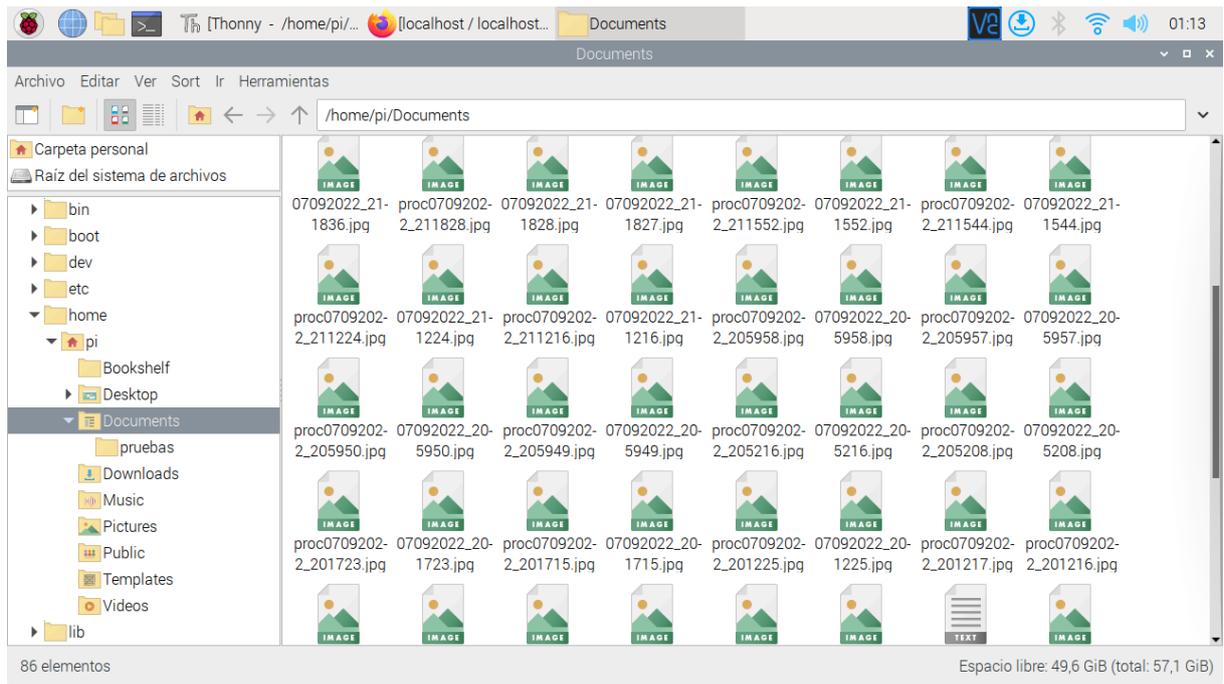


Documents Thonny - /home/pi/... Análisis 20:52

Universidad Israel **SISTEMA AUTOMÁTICO DE OVOSCOPIA.**
Ing. Javier Miranda

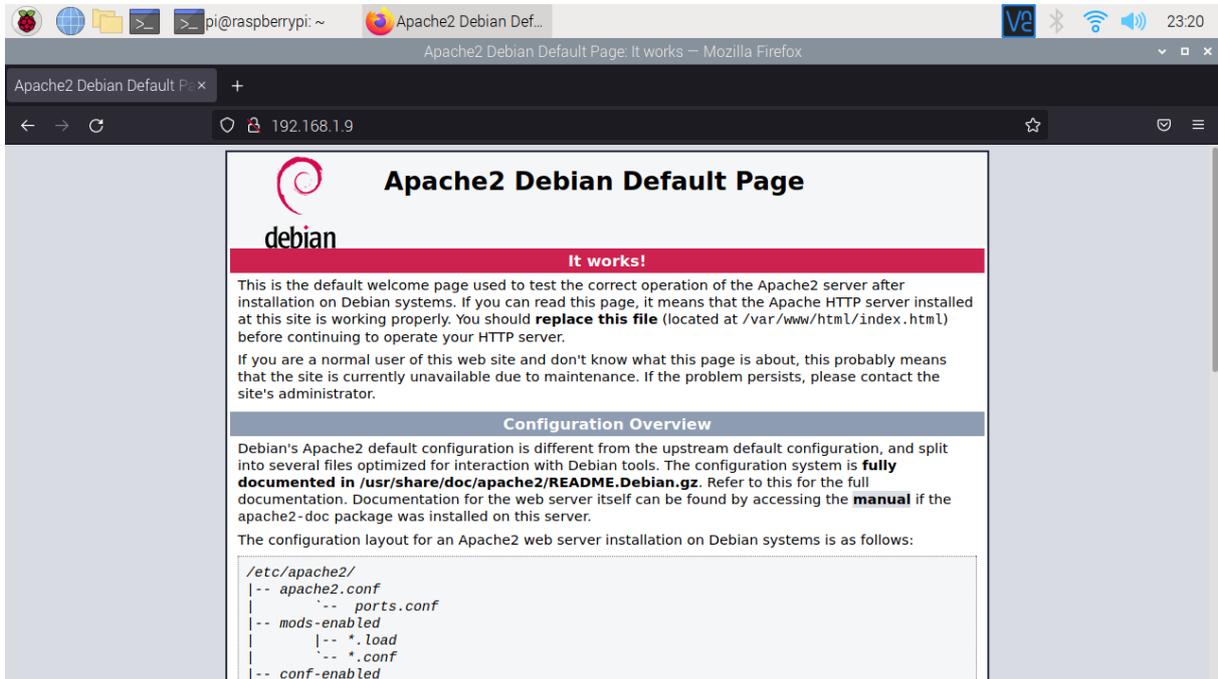
N	N	N	N	N	N
N	N	F	N	F	N
N	F	N	N	N	N
N	N	F	N	F	N
F	F	M	P	N	F



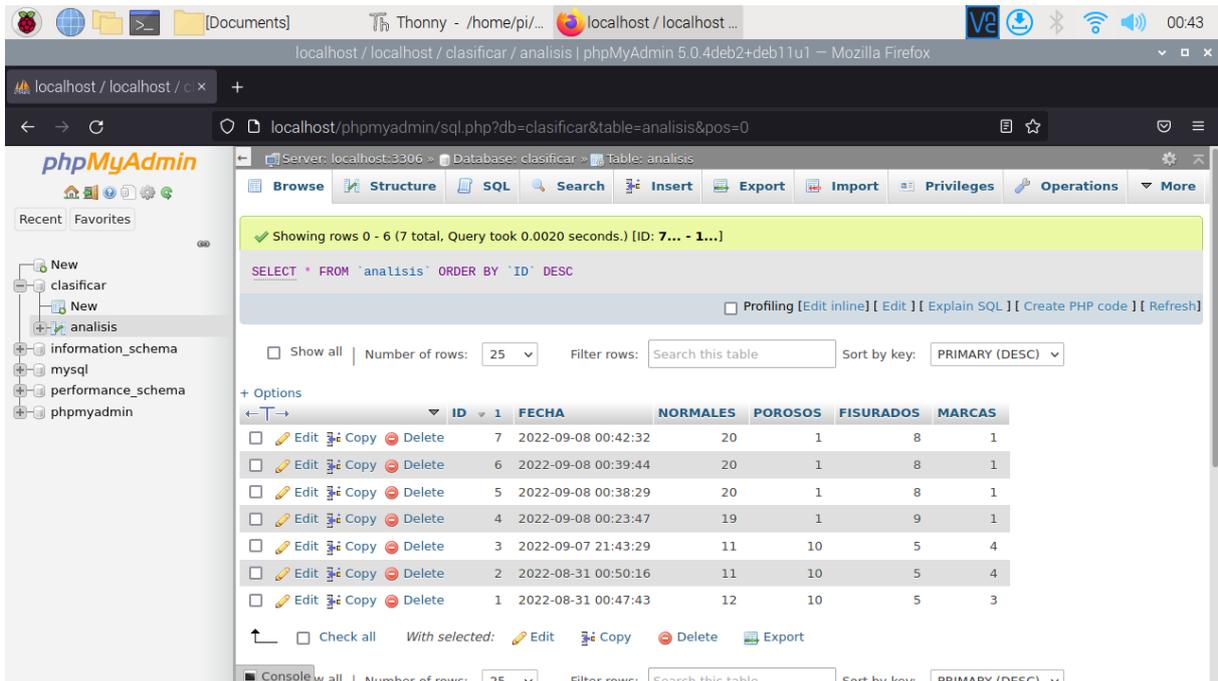


ANEXO 9

SERVIDOR Y BASE DE DATOS



The screenshot shows a web browser window displaying the Apache2 Debian Default Page. The page title is "Apache2 Debian Default Page" and it features the Debian logo. A red banner with the text "It works!" is prominent. Below this, there is a paragraph explaining that the page is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. It mentions that if the page is readable, the Apache HTTP server is working properly. A section titled "Configuration Overview" follows, detailing that the Debian Apache2 default configuration is different from the upstream default and is split into several files. It lists the configuration files: `/etc/apache2/`, `apache2.conf`, `ports.conf`, `mods-enabled`, `load`, `conf`, and `conf-enabled`.

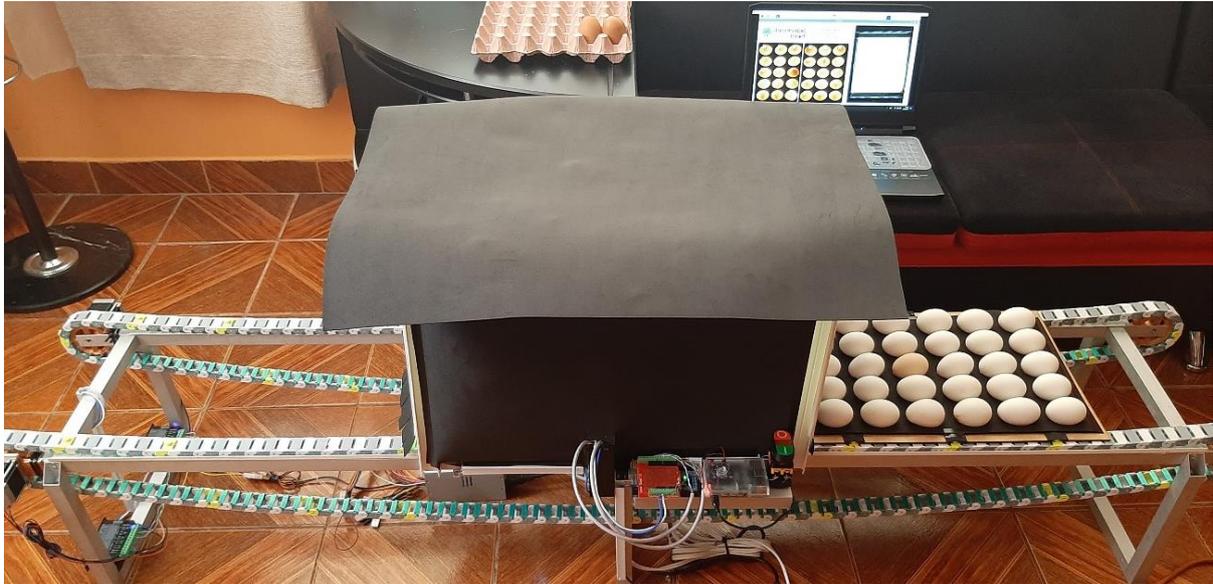


The screenshot shows the phpMyAdmin interface. The browser address bar indicates the URL `localhost/phpmyadmin/sql.php?db=clasificar&table= analisis&pos=0`. The interface displays a table query result for the `analisis` table. The query is `SELECT * FROM `analisis` ORDER BY `ID` DESC`. The table has 7 rows, and the query took 0.0020 seconds. The table structure is as follows:

ID	FECHA	NORMALES	POROSOS	FISURADOS	MARCAS
7	2022-09-08 00:42:32	20	1	8	1
6	2022-09-08 00:39:44	20	1	8	1
5	2022-09-08 00:38:29	20	1	8	1
4	2022-09-08 00:23:47	19	1	9	1
3	2022-09-07 21:43:29	11	10	5	4
2	2022-08-31 00:50:16	11	10	5	4
1	2022-08-31 00:47:43	12	10	5	3

ANEXO 10

IMÁGENES DEL PROTOTIPO FINAL



ANEXO 11

CERTIFICADOS DE VALIDACIÓN DE LA PROPUESTA



Yo, **Ángel Oliverio Fernández Escobar**, con C.I **1710374834**, en mi calidad de validador de la propuesta del proyecto titulado: **SISTEMA AUTOMÁTICO DE OVOSCOPIA PARA ANÁLISIS DE CALIDAD DE HUEVOS DE GALLINA MEDIANTE PROCESAMIENTO DIGITAL DE IMÁGENES**.

Elaborado por el Ing. **Roberto Javier Miranda Coronel**, con C.I **1804467338**, estudiante de la Maestría en Electrónica y Automatización, de la Universidad Tecnológica Israel (UISRAEL), como parte de los requisitos para obtener el Título de Magister, me permito declarar haber revisado el proyecto y realizado la evaluación de criterios.

Quito D.M., 08 de septiembre de 2022

ANGEL
OLIVERIO
FERNANDEZ
ESCOBAR

Firmado digitalmente
por ANGEL OLIVERIO
FERNANDEZ ESCOBAR
Fecha: 2022.09.08
22:01:47 -05'00'

Ing. M.Sc. Ángel Oliverio Fernández Escobar

C.I.: 1710374834

Registro SENESCYT: 1018-2017-1823632



Yo, **Martín Armando González Vélez**, con C.I **1203099740**, en mi calidad de validador de la propuesta del proyecto titulado: **SISTEMA AUTOMÁTICO DE OVOSCOPIA PARA ANÁLISIS DE CALIDAD DE HUEVOS DE GALLINA MEDIANTE PROCESAMIENTO DIGITAL DE IMÁGENES**.

Elaborado por el Ing. **Roberto Javier Miranda Coronel**, con C.I **1804467338**, estudiante de la Maestría en Electrónica y Automatización, de la Universidad Tecnológica Israel (UISRAEL), como parte de los requisitos para obtener el Título de Magister, me permito declarar haber revisado el proyecto y realizado la evaluación de criterios.

Quito D.M., 08 de septiembre de 2022



PhD. Martín Armando González Vélez

C.I.: 1203099740

Registro SENESCYT: 7241116261



Yo, **Franklin Wilfrido Salazar Logroño**, con C.I **0603039967**, en mi calidad de validador de la propuesta del proyecto titulado: **SISTEMA AUTOMÁTICO DE OVOSCOPIA PARA ANÁLISIS DE CALIDAD DE HUEVOS DE GALLINA MEDIANTE PROCESAMIENTO DIGITAL DE IMÁGENES.**

Elaborado por el Ing. **Roberto Javier Miranda Coronel**, con C.I **1804467338**, estudiante de la Maestría en Electrónica y Automatización, de la Universidad Tecnológica Israel (UISRAEL), como parte de los requisitos para obtener el Título de Magister, me permito declarar haber revisado el proyecto y realizado la evaluación de criterios.

Quito D.M., 09 de septiembre de 2022



Ing. Mg. Franklin Wilfrido Salazar Logroño

C.I.: 0603039967

Registro SENESCYT: 724187348



Yo, **José Ezequiel Naranjo Robalino**, con C.I **1804710463**, en mi calidad de validador de la propuesta del proyecto titulado: **SISTEMA AUTOMÁTICO DE OVOSCOPIA PARA ANÁLISIS DE CALIDAD DE HUEVOS DE GALLINA MEDIANTE PROCESAMIENTO DIGITAL DE IMÁGENES**.

Elaborado por el Ing. **Roberto Javier Miranda Coronel**, con C.I **1804467338**, estudiante de la Maestría en Electrónica y Automatización, de la Universidad Tecnológica Israel (UISRAEL), como parte de los requisitos para obtener el Título de Magister, me permito declarar haber revisado el proyecto y realizado la evaluación de criterios.

Quito D.M., 08 de septiembre de 2022



Ing. Mg. José Ezequiel Naranjo Robalino

C.I.: 1804710463

Registro SENESCYT: 1010-2021-2276938