



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS
ELECTRÓNICA E INDUSTRIAL
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES

TEMA:

**SISTEMA ELECTRÓNICO PARA EL CONTROL DE CALIDAD DE
HUEVOS DE GALLINA MEDIANTE PROCESAMIENTO DE
IMÁGENES.**

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Electrónica y Comunicaciones.

LÍNEAS DE INVESTIGACIÓN: Sistemas de Control

AUTOR: Ángel Patricio Fernández Soria

TUTOR: Ing. Marco Antonio Jurado Lozada Mg.

Ambato - Ecuador

Octubre 2018

APROBACIÓN DEL TUTOR

En mi calidad de tutor del Trabajo de Investigación sobre el tema: SISTEMA ELECTRÓNICO PARA EL CONTROL DE CALIDAD DE HUEVOS DE GALLINA MEDIANTE PROCESAMIENTO DE IMÁGENES, del señor ÁNGEL PATRICIO FERNÁNDEZ SORIA, estudiante de la Carrera de Ingeniería en ELECTRÓNICA Y COMUNICACIONES, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato, octubre de 2018

EL TUTOR

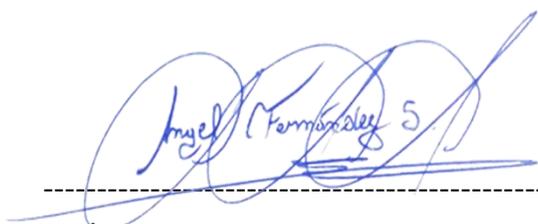


Ing. Marco Antonio Jurado Lozada, Mg.

AUTORÍA

El presente Proyecto de Investigación titulado: **SISTEMA ELECTRÓNICO PARA EL CONTROL DE CALIDAD DE HUEVOS DE GALLINA MEDIANTE PROCESAMIENTO DE IMÁGENES**, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, octubre de 2018



Ángel Patricio Fernández Soria

CC: 120637844-8

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, octubre de 2018



Ángel Patricio Fernández Soria

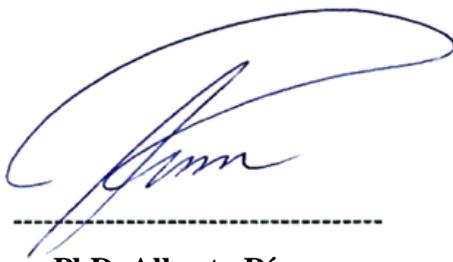
CC: 120637844-8

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes PhD. Alberto Ríos y PhD. Fabián Salazar, revisó y aprobó el Informe Final del trabajo de graduación titulado “SISTEMA ELECTRÓNICO PARA EL CONTROL DE CALIDAD DE HUEVOS DE GALLINA MEDIANTE PROCESAMIENTO DE IMÁGENES”, presentado por el señor Ángel Patricio Fernández Soria de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.



Ing. Elsa Pilar Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL



PhD. Alberto Ríos
DOCENTE CALIFICADOR



PhD. Fabián Salazar
DOCENTE CALIFICADOR

DEDICATORIA

El presente trabajo, desarrollado con tanta pasión, está dedicado principalmente a Dios y a mis padres que son la fuerza que me motiva a seguir luchando día a día, cumpliendo mis objetivos y, sobre todo, anteponiendo aquellos valores que desde siempre han sabido inculcarme para mantener mi frente en alto, valorando cada éxito alcanzado por mis propios méritos sin ofender a los demás.

Dedico también este trabajo a mi familia y mis verdaderos amigos que han estado pendientes de mí, brindándome su apoyo incondicional en todo momento y aquella motivación complementaria.

Ángel Patricio Fernández Soria.

AGRADECIMIENTO

Empiezo agradeciendo a Dios, por la vida y todas las bendiciones que me brinda día con día.

Agradezco de todo corazón a mis padres, Nancy Soria Reinoso y Ángel Fernández Escobar, por ser mi pilar fundamental, motivo de honra e inspiración y sobre todo por el sacrificio que han hecho para que yo pueda alcanzar muchos éxitos.

A mi familia y amigos que siempre han confiado en mí.

Un agradecimiento cordial a esta noble institución que me abrió las puertas para formarme profesionalmente, la Universidad Técnica de Ambato por intermedio de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, que, además, me ha permitido ser su representante a nivel nacional e internacional junto con el Club de Robótica al cual he liderado con gran honor en estos últimos años, ya que todo el tiempo invertido allí ha complementado mi formación académica.

Finalmente agradezco a todos aquellos docentes que han sabido impartir realmente su conocimiento sin malicia y sobre todo a mi tutor en este trabajo de investigación, Ing. Marco Jurado Lozada, por guiarme hasta el final con acierto y cognición.

Ángel Patricio Fernández Soria.

ÍNDICE DE CONTENIDO

APROBACIÓN DEL TUTOR.....	II
AUTORÍA.....	III
DERECHOS DE AUTOR.....	IV
APROBACIÓN DE LA COMISIÓN CALIFICADORA	V
DEDICATORIA	VI
AGRADECIMIENTO.....	VII
RESUMEN.....	XIV
ABSTRACT.....	XV
INTRODUCCIÓN	XVI
CAPÍTULO I.....	1
EL PROBLEMA	1
1.1 TEMA	1
1.2 PLANTEAMIENTO DEL PROBLEMA.....	1
1.3 DELIMITACIÓN DE CONTENIDOS	3
1.4 JUSTIFICACIÓN	4
1.5 OBJETIVOS	4
1.5.1. OBJETIVO GENERAL.....	4
1.5.2. OBJETIVOS ESPECÍFICOS	5
CAPÍTULO II	6
MARCO TEÓRICO.....	6
2.1 ANTECEDENTES INVESTIGATIVOS.....	6
2.2 FUNDAMENTACIÓN TEÓRICA.....	8
2.2.1. INCUBACIÓN	8
2.2.2. PARÁMETROS DE VERIFICACIÓN PREVIO A LA INCUBACIÓN DE HUEVOS DE GALLINA.....	9
2.2.3. FACTORES QUE INFLUYEN EN LA INCUBACIÓN Y ECLOSIÓN EXITOSA DE HUEVOS DE GALLINA.....	11
2.2.4. SELECCIÓN DE HUEVOS DE GALLINA APTOS PARA INCUBAR.....	13
2.2.5. OVOSCOPIA	14
a) HUEVO FÉRTIL	15
b) HUEVO CON GRIETAS LINEARES (HAIRLINE CRACKS).....	16
c) HUEVO CON GRIETAS TIPO ESTRELLA (STAR CRACKS).....	16
d) HUEVO EXCESIVAMENTE POROSO.....	16
e) HUEVO CON LOMBRICES	17
f) HUEVO CON CÁSCARA RUGOSA, FINA O MOTEADA	17
g) HUEVO CORREGIDO EN ÚTERO (BODY-CHECK)	17
h) HUEVO CON RUPTURA PUNTUAL	18

i)	HUEVO CON MARCA DE JAULA	18
2.2.6.	SISTEMAS DE CONTROL	18
2.2.7.	MICROCONTROLADORES	19
2.2.8.	SISTEMAS EMBEBIDOS	20
2.2.9.	IMÁGENES DIGITALES	20
a)	ELEMENTOS PRINCIPALES DE LAS IMÁGENES DIGITALES	20
	PIXELES	20
	RESOLUCIÓN DE IMÁGENES	21
	RELACIÓN DE ASPECTO	21
2.2.10.	PROCESAMIENTO DIGITAL DE SEÑALES	21
a)	PROCESAMIENTO DE IMÁGENES	21
b)	HERRAMIENTAS PARA EL PROCESAMIENTO DE IMÁGENES	22
	PYTHON	22
	TKINTER	22
	OPENCV	22
2.2.11.	BASES DE DATOS	23
a)	HERRAMIENTAS PARA EL MANEJO DE BASES DE DATOS	23
	MYSQL	23
	MARIADB	23
	PHPMYADMIN	23
2.3	PROPUESTA DE SOLUCIÓN	24
CAPÍTULO III		25
METODOLOGÍA		25
3.1	MODALIDAD DE LA INVESTIGACIÓN	25
3.2	POBLACIÓN Y MUESTRA	25
3.3	RECOLECCIÓN DE INFORMACIÓN	26
3.4	PROCESAMIENTO Y ANÁLISIS DE DATOS	26
3.5	DESARROLLO DEL PROYECTO	26
CAPÍTULO IV		29
DESARROLLO DE LA PROPUESTA		29
4.1	ANÁLISIS DE FACTIBILIDAD	30
4.1.1.	FACTIBILIDAD TÉCNICA	30
4.1.2.	FACTIBILIDAD ECONÓMICA	30
4.1.3.	FACTIBILIDAD BIBLIOGRÁFICA	30
4.2	REQUERIMIENTOS DEL PROTOTIPO	30
4.2.1.	HARDWARE	31
4.2.2.	SOFTWARE	32
4.3	DISEÑO DEL PROTOTIPO	32
4.3.1.	ESTRUCTURA DEL PROTOTIPO	33

a)	DISEÑO DE LA ESTRUCTURA DEL PROTOTIPO.....	33
4.3.2.	SISTEMA DE ILUMINACIÓN.....	36
a)	DISEÑO DEL SISTEMA DE ILUMINACIÓN.....	36
4.3.3.	SISTEMA PARA EL PROCESAMIENTO DE IMÁGENES.....	42
a)	SELECCIÓN DEL SISTEMA MICROCONTROLADO PARA EL PROCESAMIENTO DE IMÁGENES.....	42
4.3.4.	SISTEMA PARA LA ADQUISICIÓN DE IMÁGENES.....	45
a)	SELECCIÓN DEL DISPOSITIVO PARA ADQUISICIÓN DE IMÁGENES.....	45
b)	DISEÑO DEL SISTEMA ELECTROMECAÁNICO PARA EL DESPLAZAMIENTO DEL DISPOSITIVO DE ADQUISICIÓN DE IMÁGENES.....	47
c)	SELECCIÓN DE ACTUADORES PARA EL DESPLAZAMIENTO DEL DISPOSITIVO DE ADQUISICIÓN DE IMÁGENES.....	50
d)	DESARROLLO DEL SOFTWARE DE CONTROL DE MOVIMIENTO.....	52
4.3.5.	DESARROLLO DEL SOFTWARE DE PROCESAMIENTO DE IMÁGENES.....	53
a)	CAPTURA DE IMÁGENES.....	54
b)	CONVERSIÓN DE IMÁGENES A ESCALA DE GRISES.....	54
c)	REDUCCIÓN DE RUIDO EN IMÁGENES MEDIANTE FILTRO GAUSSIANO.....	55
d)	DETECCIÓN DE CONTORNOS.....	55
e)	SEGMENTACIÓN DE LAS IMÁGENES ADQUIRIDAS.....	56
f)	COMPARACIÓN DE DESCRIPTORES DE FORMA.....	57
4.3.6.	BASE DE DATOS.....	57
a)	DESARROLLO DE LA BASE DE DATOS.....	57
	SERVIDOR APACHE.....	57
	MANEJO DE BASES DE DATOS MYSQL / MARIADB.....	58
	PHP Y PHPMyAdmin.....	59
b)	DESARROLLO DEL SOFTWARE PARA VINCULAR LA BASE DE DATOS.....	61
4.3.7.	INTERFAZ HOMBRE - MÁQUINA.....	62
a)	DESARROLLO DE LA INTERFAZ HOMBRE - MÁQUINA.....	62
4.4	IMPLEMENTACIÓN Y VALIDACIÓN DEL PROTOTIPO.....	64
4.4.1.	POSICIONAMIENTO DE LA CÁMARA PARA ADQUISICIÓN DE IMÁGENES.....	65
4.4.2.	TIEMPOS DE RETARDO DEL MECANISMO DE DESPLAZAMIENTO.....	67
4.4.3.	INTERFAZ DE USUARIO.....	70
4.5	PRESUPUESTO.....	79
	CAPÍTULO IV.....	82
	CONCLUSIONES Y RECOMENDACIONES.....	82
5.1	CONCLUSIONES.....	82
5.2	RECOMENDACIONES.....	83
	BIBLIOGRAFÍA.....	84
	ANEXOS.....	93

ÍNDICE DE FIGURAS

Fig. 2.1. Clasificación de huevos de gallina aptos para incubar.	14
Fig. 2.2. Ovoscopio comercial empleando luz led para el análisis individual de parámetros internos de huevos de gallina.	15
Fig. 2.3. Mesa para ovoscopía con capacidad para bandejas de hasta 150 huevos de gallina.	15
Fig. 2.4. Huevo de gallina fresco visto con ovoscopio.	16
Fig. 2.5. Huevo de gallina con grietas lineares visto con ovoscopio.	16
Fig. 2.6. Huevo de gallina con grietas tipo estrella visto con ovoscopio.	16
Fig. 2.7. Huevo de gallina excesivamente poroso visto con ovoscopio.	17
Fig. 2.8. Huevo de gallina con lombrices visto con ovoscopio.	17
Fig. 2.9. Huevo de gallina con cáscara rugosa visto con ovoscopio.	17
Fig. 2.10. Huevo de gallina corregido en útero visto con ovoscopio.	18
Fig. 2.11. Huevo de gallina con ruptura puntual visto con ovoscopio.	18
Fig. 2.12. Huevo de gallina con marcas de jaula visto con ovoscopio.	18
Fig. 2.13. Diagrama de un Sistema de Control.	19
Fig. 2.14. Esquema general de la arquitectura de un microcontrolador (Arquitectura Harvard).	20
Fig. 3.1. Diagrama de flujo de las actividades del proyecto.	28
Fig. 4.1. Diagrama de bloques del sistema de control de calidad de huevos de gallina.	32
Fig. 4.2. Esquema de conexiones del prototipo de control de calidad de huevos de gallina.	33
Fig. 4.3. Esquema del sistema estructural del prototipo (base móvil).	35
Fig. 4.4. Esquema completo de la estructura del prototipo.	36
Fig. 4.5. Simulación del circuito base para el sistema de iluminación.	38
Fig. 4.6. Simulación del sistema de iluminación.	40
Fig. 4.7. Ubicación de componentes del sistema de iluminación sobre placa electrónica.	41
Fig. 4.8. Diseño de placa electrónica del sistema de iluminación.	42
Fig. 4.9. Sistemas Operativos Alternativos para Raspberry.	45
Fig. 4.10. Vectores de eje de coordenadas 3D.	47
Fig. 4.11. Bosquejo del sistema de desplazamiento CoreXY.	48
Fig. 4.12. Diseño mecánico del sistema de desplazamiento CoreXY (Vista superior).	49
Fig. 4.13. Diseño mecánico del sistema de desplazamiento CoreXY (Vista inferior).	50
Fig. 4.14. Diagrama de conexión de motores paso a paso controlados desde un sistema microcontrolado a través del driver DRV8825.	51
Fig. 4.15. Distribución física de pines Raspberry Pi 3 modelo B y sus funciones.	52
Fig. 4.16. Distribución lógica GPIO Raspberry Pi 3 modelo B y sus funciones.	53
Fig. 4.17. Página de verificación del Servidor Apache en Raspberry.	58
Fig. 4.18. Activación de MariaDB por consola Raspberry.	59
Fig. 4.19. Página de verificación del lenguaje PHP en Raspberry.	60
Fig. 4.20. Interfaz de administración y gestión de bases de datos phpMyadmin en Raspberry.	61

Fig. 4.21. Proceso de la interfaz gráfica del sistema de control de calidad.	64
Fig. 4.22. Implementación y validación práctica del sistema de control de calidad de huevos de gallina.	65
Fig. 4.23. Interfaz gráfica del sistema de control de calidad – Pantalla Principal.	70
Fig. 4.24. Interfaz gráfica del sistema de control de calidad – Método Manual.....	71
Fig. 4.25. Interfaz gráfica del sistema de control de calidad – Resultados Método Manual.	71
Fig. 4.26. Interfaz gráfica del sistema de control de calidad – Método automático.	72
Fig. 4.27. Interfaz gráfica del sistema de control de calidad – Resultados método automático.	72
Fig. 4.28. Contorno base de un huevo de gallina para comparación de parámetros.....	73
Fig. 4.29. Características y contornos de huevos analizados por el prototipo.	73
Fig. 4.30. Huevos de gallina analizados por el prototipo en base a las características que dificultan su incubación.	74
Fig. 4.31. Resultados de las comparaciones de los huevos de gallina analizados.	78

ÍNDICE DE TABLAS

Tabla 4.1 Características de distintos tipos de bombillas para iluminación.	37
Tabla 4.2 Características técnicas de dispositivos de control.	43
Tabla 4.3 Características técnicas de dispositivos de adquisición de imágenes (cámaras digitales). ...	46
Tabla 4.4 Características técnicas de distintos tipos de motores.	50
Tabla 4.5 Sentido de giro y desplazamiento del sistema electromecánico encargado de desplazar la cámara sobre los huevos de gallina a analizar.	66
Tabla 4.6 Características de detección de cámaras en función de la distancia.	66
Tabla 4.7 Parámetros para el movimiento del sistema electromecánico encargado de desplazar la cámara sobre los huevos de gallina a analizar - Eje X.	67
Tabla 4.8 Parámetros para el movimiento del sistema electromecánico encargado de desplazar la cámara sobre los huevos de gallina a analizar - Eje Y.	69
Tabla 4.9 Resultados de detección de huevos según características.	74
Tabla 4.10 Resultados de las comparaciones de cada huevo de gallina analizado.	75
Tabla 4.11 Resultados del consumo de memoria RAM del software desarrollado.	78
Tabla 4.12 Resultados del tiempo de respuesta empleado por los métodos de análisis para el control de calidad de huevos de gallina.	78
Tabla 4.13 Costo de Diseño del Sistema de Control de Calidad de Huevos de Gallina.	79
Tabla 4.14 Costo de Materiales del Sistema de Control de Calidad de Huevos de Gallina.	79
Tabla 4.15 Costo Total del Sistema de Control de Calidad de Huevos de Gallina.	81

RESUMEN

En la industria avícola se genera una fuerte necesidad de incrementar los controles de calidad de sus productos, por lo que es preciso automatizar procesos para evitar pérdidas económicas debido al derroche de tiempo y errores de producción.

Por lo descrito, se presenta este trabajo de investigación con el fin de implementar un prototipo de sistema electrónico para el control de calidad de huevos de gallina mediante procesamiento de imágenes, identificando huevos idóneos para incubación con el uso de hardware y software libre.

El prototipo tiene una estructura metálica con una cubeta especializada que aloja 36 huevos de gallina, posicionándolos sobre un panel de LEDs para su iluminación. Se emplea un mecanismo de desplazamiento en 2 dimensiones, basado en impresoras 3D, para hacer un barrido de adquisición de imágenes con una cámara digital. Las imágenes son procesadas mediante la aplicación de filtros, segmentación y comparación de descriptores para verificar los parámetros de calidad; los resultados son almacenados en una base de datos y se genera reportes en cada análisis.

El procesamiento de imágenes tarda 61.92 segundos, es decir, 1,72 segundos por cada huevo de gallina. El prototipo ha sido probado con huevos distribuidos para el consumo humano, obteniendo como resultado, de 108 huevos analizados, el 21% aptos y el 79% no aptos, determinando como problemas principales: fisuras, porosidad excesiva y cáscaras muy finas.

Los resultados reflejan que los huevos empleados en el consumo humano no son los más adecuados para incubación. El prototipo identifica problemas en la composición de los huevos de gallina, pero, no es posible afirmar que el 100% de los huevos seleccionados como idóneos originen pollos, porque también influyen las condiciones biológicas, manipulación e incubadora que se emplee. La eficiencia del procesamiento de imágenes está limitada por el sistema de iluminación y la cámara digital.

Palabras clave: Visión Artificial, Industria Avícola, OpenCV, Filtro Gaussiano, Canny, LAMP.

ABSTRACT

In the poultry industry there is a strong need to increase the quality controls of their products, so it is necessary to automate processes to avoid economic losses due to waste of time and production errors.

For the described, this research work is presented in order to implement a prototype electronic system for the quality control of chicken eggs through image processing, identifying eggs suitable for incubation with the use of hardware and free software.

The prototype has a metal structure with a specialized bucket that houses 36 chicken eggs, positioning them on a panel of LEDs for lighting. A displacement mechanism in 2 dimensions, based on 3D printers, is used to scan the images with the digital camera. The images are processed through the application of filters, segmentation and comparison of descriptors to verify the quality parameters; the results are stored in a database and reports are generated in each analysis.

The image processing takes 61.92 seconds, that is, 1.72 seconds for each chicken egg. The prototype has been tested with eggs distributed for human consumption, obtaining as a result, of 108 eggs analyzed, 21% suitable and 79% unsuitable, determining as main problems: cracks, excessive porosity and very thin eggshells.

The results show that the eggs used in human consumption are not the most suitable for incubation. The prototype identifies problems in the composition of chicken eggs, but, it is not possible to affirm that 100% of the eggs selected as ideal originate chickens, because also there are influence of the biological conditions, manipulation and incubator that is used. The efficiency of image processing is limited by the lighting system and the digital camera.

Keywords: Artificial Vision, Poultry Industry, OpenCV, Gaussian Filter, Canny, LAMP.

INTRODUCCIÓN

El presente trabajo de investigación es un aporte al sector avícola del país, diseñando e implementando un sistema automatizado de bajo costo gracias al manejo de hardware y software libre. Este sistema es un complemento a los controles de calidad estándar manejados en las empresas dedicadas a la producción de pollos, mediante la incubación de huevos de gallina, permitiendo verificar las condiciones físicas internas y externas previas a la incubación que a menudo no son perceptibles, empleando un sistema de iluminación especial y adquisición de imágenes a través de una cámara digital para su respectivo análisis.

Para el desarrollo de la investigación se realizó un análisis de la situación actual del manejo y controles de calidad de huevos de gallina, comúnmente aplicados en las empresas avícolas a nivel nacional, planteando la necesidad del diseño de este sistema de control de calidad automático. Se emplea un sistema electromecánico para desplazar la cámara sobre la cubeta, diseñada especialmente para el análisis de huevos de gallina, además, a través de una interfaz amigable con el usuario se permite seleccionar dos tipos de métodos dentro del mismo prototipo; el método manual y el método automático, todo esto a través del software específicamente desarrollado, que permite también almacenar los resultados de cada análisis en una base de datos disponible a través de servidores web para el fácil acceso a esta información.

En el primer capítulo se detallan los problemas de producción avícolas, específicamente de pollos, ya que al no disponer de sistemas automatizados ni antecedentes de análisis no se ha podido mejorar la producción frente a la demanda de la población que crece continuamente. En este capítulo se plantea la solución a esta problemática en base a los objetivos cumplidos al término del proyecto.

El capítulo dos presenta todos los antecedentes investigativos tomados como base en el tema desarrollado y la fundamentación teórica referente a los elementos empleados en el proyecto.

En el capítulo tres se plantea toda la base investigativa con la cual se ha procedido como la modalidad de investigación, tipos de investigación aplicados, herramientas de procesamiento y análisis de la información y los pasos para el desarrollo del proyecto.

El cuarto capítulo describe el desarrollo del proyecto como solución al problema planteado en base a las necesidades del sector avícola, detallando todos los parámetros técnicos empleados en el diseño y construcción del sistema. Se justifica el hardware y software empleado para el funcionamiento del prototipo y se presentan las pruebas de funcionamiento que verifican la validez del diseño determinando, por supuesto, la utilidad en todo el sector de producción avícola.

Finalmente se presentan las conclusiones y recomendaciones del proyecto en el capítulo cinco.

CAPÍTULO I

EL PROBLEMA

1.1 TEMA

SISTEMA ELECTRÓNICO PARA EL CONTROL DE CALIDAD DE HUEVOS DE GALLINA MEDIANTE PROCESAMIENTO DE IMÁGENES.

1.2 PLANTEAMIENTO DEL PROBLEMA

En el Ecuador, sus habitantes tienen una preferencia por el consumo de carne de pollo, cerdo y res. De acuerdo a la Asociación Nacional de Fabricantes de Alimentos y Bebidas, se tiene un porcentaje aproximado de 59,26% en el consumo de carne de pollo per cápita anual, del total de consumo de carnes. De las provincias que conforman el país, Tungurahua es una de las provincias con mayor consumo de carne de pollo registrado a nivel nacional, considerando que Ambato tiene un consumo significativo al ser el cantón más poblado de esta provincia. De acuerdo con la Corporación Nacional de Avicultores del Ecuador, el consumo de carne de pollo ofrece mayores beneficios nutritivos en comparación con la carne de ganado bovino y ovino, puesto que contiene menos colesterol, calorías y grasa, además, provee mayores cantidades de proteína. Solo la carne de pavo y avestruz supera los beneficios de la carne de pollo, pero tienen un costo más elevado. [1], [2]

El consumo de pollos per cápita a nivel nacional registra un valor anual de 35 Kg. Los pollos tienen un peso promedio de 2,5 Kg, ostentando un consumo de 14 pollos por persona anualmente. Según el INEC, se tiene un promedio de 4 personas por familia a nivel nacional, por tanto, se estima que el consumo anual de pollos es de 50'692.246,5 en Ecuador, 1'766.040,5 en Tungurahua y 1'154.496 en la ciudad de Ambato, referente al total de familias. [3], [4], [5]

Según el Ing. Elías Chiriboga Lozada, basado en fuentes de Bioalimentar, se proyecta que el Ecuador llegue a niveles de consumo de pollo per cápita, similares a Estados Unidos o Brasil. En base a los registros de la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO), el consumo de pollos per cápita anual en Estados Unidos es de 54,3 Kg mientras que en Brasil es de 58,1 Kg. Al 15 de noviembre del 2013 se registra una producción anual de 200 millones de pollos en el Ecuador según Andrés Pérez, representante de Pronaca y directivo de la Corporación Nacional de Avicultores del Ecuador (CONAVE). [6], [7], [8]

Considerando los datos presentados, en el país se consume 202'768.986 pollos al año, si se aproxima al consumo de Estados Unidos y Brasil debería consumirse 314'581.598,3 o 336'596.516,8 respectivamente lo que evidencia un déficit de producción de 136'596.516,8 pollos afectando directamente a ambas partes, consumidor y productor. [6], [7], [8]

En el Ecuador se producen 55'427.095 de Huevos de Gallina semanalmente, según los datos del Instituto Nacional de Estadísticas y Censos, destacando que la provincia de Tungurahua es la mayor productora de huevos de gallina a nivel nacional con un porcentaje de 39,986%. Adicionalmente, el 91,84% de la producción nacional se destina a las ventas, el 8,04% a otros destinos que involucra la incubación y el 0,12% al autoconsumo. [9]

Según el Ing. Bolívar Montenegro, director de la Escuela de Ingeniería Zootecnia de la Universidad Técnica Estatal de Quevedo, en una incubadora artesanal el rango fluctúa entre 75 al 85% de fertilidad; mientras que en una incubadora industrial varía del 85 al 95% dependiendo del origen de la fabricación de la incubadora así por ejemplo si es de fabricación nacional o importada y si es automática o manual. [10]

Según J. Abad, J. Castelló, E. Carbajo, P. Casanovas, A. Dalmau, E García, R. Lera y R. Martínez; en condiciones normales de almacenamiento y controles de temperatura adecuados previo a la incubación de los huevos de gallina se obtienen porcentajes de entre 64 y 78 % de eclosión, adicionando el método de volteo durante la incubación se alcanza un porcentaje máximo de nacimiento del 95%, sin tener registros de controles de calidad adicionales. [11]

Típicamente los controles de calidad que se hacen a través de ovoscopios y balanzas, son manuales, evidenciando un problema en las mediciones por los errores humanos, además no existen registros publicados de los procesos automatizados de selección de huevos de gallina aptos para incubación, ni porcentajes de producción en las empresas dedicadas a esta actividad, a nivel mundial y menos a nivel local, dificultando la precisión de la información estadística o su vez los cálculos en los diseños de nuevos sistemas mejorados para la automatización de estos procesos productivos.

En base a un promedio registrado entre los valores máximos y mínimos de eclosión de los huevos de gallina, posterior a los procesos de incubación, se tiene un 79,5% de nacimientos, lo cual se puede mejorar teniendo controles más robustos de calidad de los parámetros intrínsecos del producto previo a la incubación, aparte de los controles de temperatura e higiene que se manejan tanto en las plantas como en las máquinas avícolas. [10], [11]

Es evidente que el mercado ecuatoriano requiere aumentar la producción de pollos, eclosionando un porcentaje más elevado de huevos, automatizando las actividades que implican los controles de calidad previos, con el fin de evitar la pérdida de recursos que se tiene durante el proceso de incubación en las distintas empresas dedicadas a esta actividad avícola, otorgando, la factibilidad de aumentar la nutrición de los consumidores tanto en carne como huevos de gallina.

1.3 DELIMITACIÓN DE CONTENIDOS

ÁREA ACADÉMICA: Física y Electrónica.

LÍNEA DE INVESTIGACIÓN: Sistemas de Control.

SUBLÍNEA DE INVESTIGACIÓN: Sistemas Embebidos.

DELIMITACIÓN ESPACIAL

La presente investigación se realizó en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

DELIMITACIÓN TEMPORAL

La presente investigación se efectuó en el periodo julio 2017 - agosto 2018 de acuerdo a lo estipulado en el Reglamento de Graduación para Obtener el Título Terminal de Tercer Nivel de la Universidad Técnica de Ambato.

1.4 JUSTIFICACIÓN

El presente proyecto se enfoca en optimizar los procesos de producción precautelando los recursos empleados en las empresas dedicadas a incubación de huevos de gallina y comercialización de pollos para consumo nacional e internacional, controlando la calidad del producto mediante un sistema de automatización para separar los huevos que puedan eclosionar y tener otros posibles usos, como el consumo humano, con aquellos que no sean aptos para incubar, realizando una detección temprana de problemas y aumentando la cantidad de pollos producidos luego del proceso de incubación.

Al desarrollar este sistema los beneficiarios inmediatos son los consumidores, debido a que al tener mayor producción se alcanza un mayor consumo de carne de pollo y huevos para su nutrición, a la par, son beneficiados los productores de la provincia de Tungurahua obteniendo mayor cantidad de ventas sin tener pérdidas económicas y materiales, es decir se optimizan los recursos de sus empresas.

La Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, cuenta con el espacio físico, biblioteca y laboratorios necesarios para el desarrollo del proyecto, facilitando el acceso a la información y recursos pertinentes. El sistema propuesto puede replicarse a nivel nacional en cuyo mercado se pueden adquirir fácilmente los elementos. Además, este proyecto es rentable y factible puesto que no implica una gran inversión económica considerando que se trabajó a nivel de software y hardware libre.

1.5 OBJETIVOS

1.5.1. OBJETIVO GENERAL

Implementar un prototipo de sistema electrónico para el control de calidad de huevos de gallina mediante procesamiento de imágenes.

1.5.2. OBJETIVOS ESPECÍFICOS

- Analizar las técnicas y métodos empleados en el sector avícola para la detección de huevos de gallina aptos para incubación.
- Determinar las características de hardware y software adecuadas para el sistema de control de calidad de huevos de gallina.
- Diseñar un prototipo de sistema electrónico para el control de calidad de huevos de gallina mediante procesamiento de imágenes.

CAPÍTULO II

MARCO TEÓRICO

2.1 ANTECEDENTES INVESTIGATIVOS

En base a tesis y publicaciones científicas se obtienen los trabajos de investigación presentados a continuación:

En el año 2013, Cristiane de Fátima dos Santos, Deivid do Vale Nascimento y Jucelino Cardoso Marciano dos Santos, Brasil, en base a su investigación presentada en el artículo científico denominado: “A software for selection of eggs using digital image processing with customization between profits and quality”, mediante el uso de una cámara de video digital y el entorno de Matlab, plantean que las iteraciones y los umbrales utilizados en los procesos se pueden ajustar según las necesidades locales donde se despliega el sistema, aumentando la segmentación de iteraciones del método, el software es más sensible y pasa a reconocer defectos más pequeños. Otro aspecto que debe también ser ajustado de acuerdo con las necesidades del usuario es una aplicación de base de datos, lo que puede ir más allá de la propuesta y llevar a cabo el control sobre otra información, por ejemplo, procedente de un lote que se está procesando, el tiempo necesario para procesar los lotes, etc. En trabajos futuros se podrán emplear descriptores de imagen y se ha de establecer una comparativa. [12]

En el año 2014, Michael José Medina Jiménez y José Bolívar Oñate Rosado de la Universidad Autónoma Del Caribe, Colombia, realizaron la tesis titulada: “Diseño e implementación de un prototipo que clasifique huevos de gallina según sus características”, en la cual emplean hardware libre (Arduino, cámara y sensores) para la adquisición de datos y software de pago (Matlab), determinando que para este tipo de sistema se requiere de un diseño único. Al emplear un sistema de visión artificial y un diseño estructural, afirman que es indispensable un sistema de iluminación

controlada para obtener imágenes de calidad alcanzando porcentajes de error menores al 5 %. [13]

En el año 2014, María Cristina Navas Lema y Ramiro Sebastián Vargas Cruz de la Universidad de las Fuerzas Armadas extensión Latacunga, Ecuador, realizaron el estudio “Diseño e implementación de un sistema de ovoscopia con visión artificial para la detección de huevos fértiles para Incubandina”, al estructurar la parte mecánica y de control del sistema con hardware libre, procesando los datos con software de pago (Matlab), plantean que, si bien la visión artificial no detecta en un 100% los errores debido a las condiciones bajo las cuales se realiza la Ovoscopia, puede reconocer cierto número de imperfecciones gracias a lo cual cataloga a un huevo como fértil e infértil resultados claramente visibles en el número de eclosiones satisfactorias. [14]

En el año 2015, el trabajo de titulación bajo el tema: “Diseño, Construcción y Automatización de una máquina clasificadora para huevos por peso, en diferentes tamaños para 2000 huevos por hora de capacidad” realizado por Diego Fernando Enríquez Zurita en la Universidad Politécnica Salesiana Sede Quito, Ecuador, empleó un PLC de Siemens basado en lenguaje ladder, desarrollado en el software Logo Comfort, mediante el cual, Enríquez afirma que los programas paramétricos de simulación son una herramienta fundamental en la verificación de los cálculos sobre todo cuando se emplean elementos irregulares en el diseño, ya que la máquina debe ser capaz de soportar su propio peso, la carga de los huevos e incluso peso extra. Además, este tipo de máquinas mejoran la producción y se alcanza mayor capacidad de ovoproductos clasificados que la propuesta planteada. [15]

En el año 2016, Luis Felipe Gómez Aguilar y Javier Villasánchez Rosas del Instituto Politécnico Nacional, México, en base a su tesis titulada: “Propuesta de un sistema de visión artificial aplicada al proceso ovoscópico en la vacunación in ovo”, afirman que Labview facilita el procesamiento de imágenes con sus diferentes métodos de visión, mostrando diversas opciones de filtrado, manipulación, morfología, binarización, detección de umbral, extracción de imágenes en RGB, etc. Además, se pueden realizar HMI's básicas dentro del mismo programa, esto con la finalidad de hacer más sencilla la interacción. [16]

En el año 2017, Ryota Shimizu, Shusuke Yanagawa, Toru Shimizu, Mototsugu Hamada y Tadahiro Kuroda de la Universidad Keio, Japón, publicaron el artículo denominado: “Convolutional neural network for industrial egg classification”, donde emplean un método de redes neuronales entrenado con 2000 imágenes, inspeccionando a mayor detalle la calidad de productos industriales, como es el caso de los huevos de gallina, procesando las imágenes a través de convoluciones sucesivas y un método estocástico de optimización. La precisión de la clasificación mejoró 3,3%, adquiriendo imágenes de 4 ángulos en lugar de imágenes de 1 solo ángulo, por lo que confirman que este proceso es efectivo para la clasificación de objetos tridimensionales. [17]

En el año 2018, A. V. Vasileva; E. V. Gorbunova; A. S. Vasilev; V. S. Peretyagin; A. N. Chertov y V. V. Korotaev, de la Universidad ITMO, Rusia, publicaron el artículo científico “Assessing exterior egg quality indicators using machine vision”, donde describen su investigación basada en visión artificial bajo el software de pago LabVIEW. El mecanismo desarrollado por los autores consiste en 2 sistemas de iluminación, con difusores de luz para garantizar uniformidad; superior e inferior, cuyo brillo es manejado por un microcontrolador de 8 bits, eliminando las sombras causadas por los huevos de gallina sobre la superficie de base. En la parte superior, en medio del sistema de iluminación, se ubica la cámara digital con sensor de color CMOS, encargada de adquirir la imagen a procesar, permitiendo el cálculo automático de parámetros geométricos a partir de huevos de gallina, y una estimación numérica de las características estructurales de la cáscara con tiempos significativamente reducidos. [18]

2.2 FUNDAMENTACIÓN TEÓRICA

2.2.1. INCUBACIÓN

La Incubación es el proceso de desarrollo del embrión dentro de los huevos puestos por las aves, en condiciones de calor específicas ya sean naturales o artificiales, a continuación, se describe cada una de ellas:

- La Incubación Natural se refiere a las acciones que realizan las aves para que sus huevos eclosionen, echándose sobre ellos, considerando el número de huevos que pueden cubrir con su cuerpo. Durante el período de incubación (21 días en las gallinas) el ave sufre alteraciones hormonales, aumentando el nivel de la

prolactina, además, el ave pierde parte de sus plumas en la región pectoral y abdominal con el fin de tener mayor transferencia de calor hacia los huevos, en contacto directo con la piel del ave.

- La Incubación Artificial reproduce los procesos naturales, realizándolos a través de máquinas con el fin de generar las condiciones idóneas para la eclosión de los huevos, utilizando al ave solo como proveedora de los huevos fértiles. Actualmente las máquinas incubadoras poseen adelantos de física, química y electrónica que superan la capacidad de incubación del ave.

En el proceso de incubación, tanto natural como artificial, la temperatura debe permanecer entre 37,5°C y 37,8°C para huevos de gallina; la humedad relativa también es importante y puede variar entre el 50% y 55%, además, debe existir aireación suficiente para liberar CO₂ del lugar de incubación y movimiento de los huevos que debe ser de 45° respecto al eje vertical entre las 2 posiciones posibles alrededor de la dimensión más corta del huevo con el fin de que la yema no se pegue a la cáscara, evitando la muerte del embrión. [19], [20], [21]

2.2.2. PARÁMETROS DE VERIFICACIÓN PREVIO A LA INCUBACIÓN DE HUEVOS DE GALLINA

En el proceso natural, las gallinas adecuan los huevos de su puesta y se posicionan sobre ellos para iniciar inmediatamente con la incubación, durante este proceso, ellas mismas desechan aquellos huevos que presenten daños considerables o notorios según su percepción.

En el proceso artificial, los huevos de gallina deben ser verificados antes de ingresarlos a las cámaras de incubación, con el objeto de clasificarlos y precautelar las mejores condiciones para que los huevos eclosionen satisfactoriamente. Los principales parámetros que deben ser verificados antes de que los huevos ingresen a ser incubados son:

- Tamaño y peso del huevo.
- Limpieza del huevo y su entorno.
- Estructura y composición del huevo.

Según el Ingeniero Zootecnista Adolfo Sánchez, docente de la UTEQ, la calidad del huevo se obtiene en base al peso, tamaño y medidas higiénicas externas que son verificadas previo a los procesos de incubación, se debe cuidar el aseo de los huevos para evitar la contaminación de otros huevos dentro de la incubadora. En las industrias, por el gran volumen de huevos que manejan, para incubación, se tiene procesos automatizados como controles de temperatura y volteo, por ejemplo, garantizando un mayor porcentaje de eclosiones. La automatización de los procesos para incrementar el porcentaje de producción, la rentabilidad económica y calidad del producto debería aplicarse no solo en las industrias sino también en las plantas de producción doméstico artesanales. [22]

La Doctora Magdalena Herrera, docente investigadora en el área avícola de la UTEQ, considera que los huevos deben estar aislados a temperaturas adecuadas bajas y humedad necesaria en los procesos previos a la incubación para poder mantener la sanidad de los mismos, la limpieza no debe ser con agua, se los puede limpiar con pequeñas esponjas retirando heces sin afectar el contenido de los huevos a través de los poros que poseen. Hay que considerar además el tamaño y peso de los huevos, seleccionarlos en base a estos parámetros, los huevos muy pequeños no son aptos ya que el resultado son pollos débiles, se debe seleccionar los huevos de 60 gr. en adelante. Los huevos considerados fértiles se emplean en la incubación y los que no, se emplean en la alimentación, los campesinos determinan la fertilidad de los huevos a través de bombillos, incluso velas que son métodos rudimentarios; en las industrias se cuenta con sensores adecuados para verificar la calidad de los huevos. Todo lo referente a técnica e innovación es un avance; partiendo de lo que existe se debe mejorar los procesos automatizados, además, los procesos automatizados permiten cubrir mayor cantidad de productos que las técnicas rudimentarias. Es una pérdida de tiempo trabajar, en los procesos de incubación, con huevos no seleccionados ya que se genera productos de muy mala calidad. [23]

El Ingeniero Bolívar Montenegro, director de la Escuela de Ingeniería Zootecnia de la UTEQ, considera que lo primero que se verifica antes de la incubación es el tamaño y peso del huevo. Típicamente los huevos se seleccionan empleando Ovoscopios, que permiten verificar la estructura y composición. Para la limpieza y desinfección de los huevos de gallina se emplea Permanganato de Potasio. La inserción de la tecnología

en base a procesos automatizados puede mejorar la producción considerando el tamaño de la empresa. [10]

2.2.3. FACTORES QUE INFLUYEN EN LA INCUBACIÓN Y ECLOSIÓN EXITOSA DE HUEVOS DE GALLINA

Para que un huevo eclosione satisfactoriamente existen varios factores a considerar durante las etapas previas a la incubación, los cuales se detallan a continuación.

- Los Factores Genéticos son características físicas, biológicas o hereditarias que se transmiten desde los productores, en el caso de estudio las gallinas, hacia los huevos y posteriormente los pollos que eclosionan. Existe grandes diferencias en los huevos de gallinas, tanto en la calidad de la cáscara como en el tamaño de los mismos, debido a una falta de selección y mejora genética de los animales, lo que implica la desigualdad de cifras encontradas en cuanto a parámetros tales como tasa de incubabilidad, porcentaje de fertilidad o peso al nacimiento, así como necesidades ambientales para el proceso de la incubación. [24]
- El Peso es una condición física importante que determina la aptitud de incubación individual de los huevos de gallina. El peso del huevo es variable, estando influenciado por factores tales como: el tamaño de la hembra, el momento del ciclo de puesta y la alimentación. El peso del huevo determina de forma clara y positiva el peso del pollo al nacimiento, aspecto importante para la vitalidad del recién nacido. Los huevos de gran tamaño producen pollos edematosos y de nacimiento tardío, debido a una falta de intercambio gaseoso y de vapor de agua. Por el contrario, los huevos excesivamente pequeños producen pollos deshidratados, de pequeño tamaño y muy débil al nacimiento, debido a la gran pérdida de agua durante el proceso de incubación. No deben incubarse huevos de peso inferior a 52 g., ni superior a 69 g. [24], [25]
- La cáscara representa un 10 % del peso total del huevo, no es lisa ni impermeable, contiene poros que permiten la oxigenación del pollo. El grosor de la cáscara varía entre 1,4 y 2,4 mm, con un valor medio entre 1,8 y 2 mm, influyendo en la cantidad de pérdida de agua durante el proceso de incubación. También existen diferencias en cuanto a la porosidad de la cáscara. Sus características dependerán especialmente de la correcta deposición de carbonato cálcico, que a su vez responde al tiempo de

permanencia del mismo en el útero. Cuanto más temprano es puesto el huevo en el día mayor es la calidad de la cáscara, y cuanto más vieja es la gallina más fina es esta cáscara. Se eliminan, para incubación, todos aquellos huevos con anomalías en la cáscara y con fisuras en la misma, ya que el riesgo de contaminación por microorganismos patógenos es muy elevado. [24], [26], [27]

- La alimentación es una necesidad biológica que consiste en adquirir nutrientes y la energía necesaria en cada ave. La alimentación de la hembra influye tanto en la calidad como en el tamaño del huevo y, consecuentemente, en la posibilidad y peso al nacimiento del pollo. El huevo debe contener todos los nutrientes que el embrión necesita cuando es puesto por la gallina. Es muy importante mantener una dieta equilibrada durante toda la época de reproducción, evitando carencias de vitaminas y minerales lo que puede ocasionar importantes alteraciones en el embrión. [24]
- La sanidad animal involucra los procesos de prevención de enfermedad y el control o erradicación de las mismas. La presencia de agentes infecciosos puede originar baja tasa de incubabilidad, elevada mortalidad embrionaria y menor peso de los pollos al nacer. Cuando la planta de incubación recibe huevos contaminados, al momento de incubarse, se dan las condiciones ideales para el crecimiento de los microorganismos, pudiendo contaminar al resto de los huevos dentro de la incubadora. Es importante la desparasitación regular de los reproductores. Ante las amenazas biológicas, la planta de incubación debe mantener un monitoreo microbiológico cuya frecuencia depende del conocimiento que se tenga de la condición sanitaria de los huevos que se reciben, pero es normal que se realice un muestro quincenal o mensual cuando se tiene un buen historial. [24], [28]
- Es importante la edad de los reproductores, generalmente las gallinas alcanzan la madurez sexual entre las 24 y 28 semanas de edad. En la primera temporada de puesta los porcentajes de fertilidad son bajos. [24], [29]
- La relación de los reproductores implica el número de machos y hembras que deben existir e influye en los resultados de fertilidad. Los mejores resultados de fertilidad se consiguen con una relación macho: hembra de 1: 2, es decir que por cada macho deben existir dos hembras dentro del plantel. [24]

- El estrés es el impedimento a la tranquilidad de los productores. Cualquier situación de estrés que sufran las aves durante la época de reproducción ocasiona una disminución en la de fertilidad y en la tasa de puesta, por lo que debe ser evitada. Cuando la reproducción no se efectúa en trío sino en grandes grupos, la presencia de machos muy dominantes que luchan constantemente es una causa de estrés hacia las hembras. Por otra parte, las gallinas son muy sensibles al estrés sónico, de tal manera que los planteles de reproducción deben situarse lo más alejados posible de las carreteras principales o de cualquier otro contaminante acústico. Además, una manipulación excesiva de los reproductores, durante la época de monta, puede ocasionar una situación de estrés crónico, pudiendo afectar negativamente a la reproducción. [24]
- El manejo del huevo fértil se refiere al uso manual o automático de los huevos aptos para incubación, implícitamente el transporte de los mismos que debe ser sumamente delicado por la fragilidad presente. La pre-incubación abarca todas aquellas prácticas de manejo efectuadas desde la puesta del huevo hasta su colocación en el interior de la incubadora. La incubación propiamente dicha engloba la eclosión o nacimiento del pollo. El manejo al que se someten los huevos es una de las principales causas de una mala incubabilidad y, además, de fácil diagnóstico. Los cambios que tienen lugar en el huevo durante la incubación se presentan ordenados y regidos por leyes naturales. Estos cambios se producen, con normalidad, solamente bajo niveles determinados de temperatura, humedad, contenido químico del aire y posiciones del huevo. Por otra parte, el mismo huevo incubado modifica el medio que lo rodea al emitir calor, gases y vapor de agua hacia el mismo. [24]

2.2.4. SELECCIÓN DE HUEVOS DE GALLINA APTOS PARA INCUBAR

En el proceso de selección de huevos de gallina se debe descartar los siguientes:

- Huevos que presenten suciedad.
- Huevos rotos, perforados por la gallina o con fisuras externas visibles e internas verificadas con ovoscopios.
- Huevos muy pequeños o muy grandes.

- Huevos con doble yema.
- Huevos con exceso de poros.
- Huevos con cáscara fina o débil.
- Huevos deformes, muy largos, redondos, arrugados o de extremo puntiagudo.

Por ende, aquellos huevos de gallina que no presenten las condiciones mencionadas son considerados aptos para el proceso de incubación. En la figura 2.1 se ilustra la clasificación detallada de los huevos de gallina. [30], [31], [32]



Fig. 2.1. Clasificación de huevos de gallina aptos para incubar. [33]

2.2.5. OVOSCOPIA

La ovoscopia es la técnica en la que se mantiene luz brillante por encima o debajo de un huevo para estudiar el crecimiento y el desarrollo de un embrión. Se realiza en una habitación a oscuras con la lámpara de examen a trasluz. La ovoscopia permite detectar los huevos saludables y descartar aquellos huevos infértiles que poseen anomalías, manchas de sangre, manchas de carne y cáscaras de huevo rotas. En general, se suele utilizar una lámpara común o fría de 40 a 60 vatios. En la figura 2.2 se muestra un sistema ovoscópico que permite visualizar los parámetros internos de los huevos, de forma individual. [34], [35]



Fig. 2.2. Ovoscopio comercial empleando luz led para el análisis individual de parámetros internos de huevos de gallina. [36]

En la figura 2.3 se muestra un sistema ovoscópico que permite visualizar los parámetros internos de los huevos, de forma colectiva.



Fig. 2.3. Mesa para ovoscopia con capacidad para bandejas de hasta 150 huevos de gallina. [37]

A través de los sistemas de ovoscopia se logran identificar huevos con las siguientes características:

a) HUEVO FÉRTIL

Un huevo fértil es aquel que tiene óptimas condiciones para eclosionar satisfactoriamente, es decir, que no presenta problemas internos o externos. En este tipo de huevos, la clara se muestra transparente y limpia, la cámara de aire menor a 6 mm e inmóvil, la yema visible en forma de sombra sin contorno en la posición central como se observa en la figura 2.4. [38]

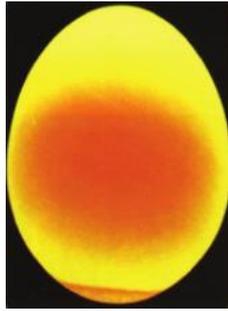


Fig. 2.4. Huevo de gallina fresco visto con ovoscopio. [38]

b) HUEVO CON GRIETAS LINEARES (HAIRLINE CRACKS)

En este tipo de huevos, las grietas son muy finas, generalmente longitudinales a lo largo de la cáscara como indica la figura 2.5. [39]

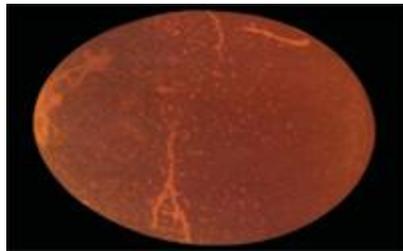


Fig. 2.5. Huevo de gallina con grietas lineares visto con ovoscopio. [39]

c) HUEVO CON GRIETAS TIPO ESTRELLA (STAR CRACKS)

Como se puede observar en la figura 2.6, en este tipo de huevos, las grietas son muy finas e irradian hacia el exterior desde un punto central de impacto. [39]

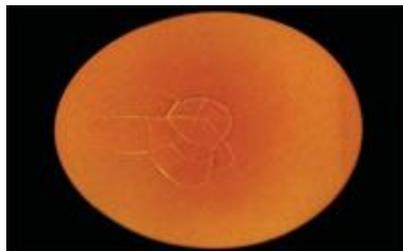


Fig. 2.6. Huevo de gallina con grietas tipo estrella visto con ovoscopio. [39]

d) HUEVO EXCESIVAMENTE POROSO

Los huevos con exceso de poros muestran gran parte de la cáscara traslúcida, como denota la figura 2.7. [39]



Fig. 2.7. Huevo de gallina excesivamente poroso visto con ovoscopio. [39]

e) HUEVO CON LOMBRICES

Este tipo de huevos se presentan cuando las lombrices se posicionan en el oviducto de la gallina, formando parte del huevo en el momento de la puesta. La incidencia es bastante rara y se puede apreciar internamente como en la figura 2.8. [39]



Fig. 2.8. Huevo de gallina con lombrices visto con ovoscopio. [39]

f) HUEVO CON CÁSCARA RUGOSA, FINA O MOTEADA

Los huevos presentan estas características cuando existen problemas en la cáscara, generalmente cuando ésta es muy frágil o fina mostrando gran parte de la cáscara de apariencia traslúcida, distinguiéndose como en la imagen 2.9. [40]

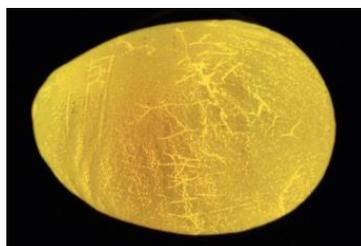


Fig. 2.9. Huevo de gallina con cáscara rugosa visto con ovoscopio. [40]

g) HUEVO CORREGIDO EN ÚTERO (BODY-CHECK)

Este tipo de huevos muestran anomalías o grietas transversales a lo largo de la cáscara, similares a las de la figura 2.10. [40]

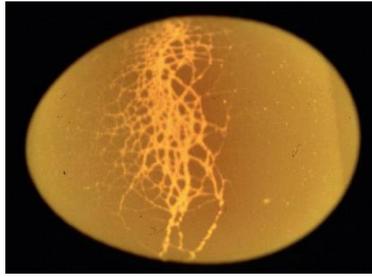


Fig. 2.10. Huevo de gallina corregido en útero visto con ovoscopio. [40]

h) HUEVO CON RUPTURA PUNTUAL

Este tipo de huevos muestran una pequeña fisura en la cáscara, generalmente por impacto con objetos agudos, como en la figura 2.11. [40]

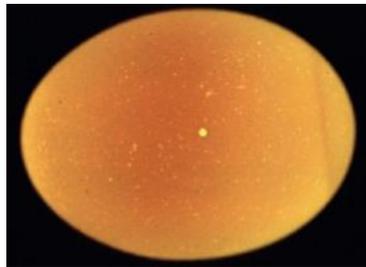


Fig. 2.11. Huevo de gallina con ruptura puntual visto con ovoscopio. [40]

i) HUEVO CON MARCA DE JAULA

Este tipo de huevos presentan marcas generadas generalmente por el contacto con alambres u otros objetos presentes en el lugar de puesta o almacenamiento y durante la recolección, se distinguen como en la figura 2.12. [40]



Fig. 2.12. Huevo de gallina con marcas de jaula visto con ovoscopio. [40]

2.2.6. SISTEMAS DE CONTROL

Los sistemas de control son procesos que tienen un conjunto de elementos relacionados entre sí para alcanzar un determinado fin; generan valores en las señales de salida en función de las señales o datos de entrada, lo que permite seleccionar la acción de

control adecuada para mejorar las respuestas previstas. La regulación y control de procesos industriales permiten la automatización de la producción, mejorando la calidad del producto, incrementando la seguridad, disminución del consumo energético, optimización del coste de producción y mantenimiento, etc. En la figura 2.13 se puede observar la generalización de los parámetros principales de un sistema de control. [41]

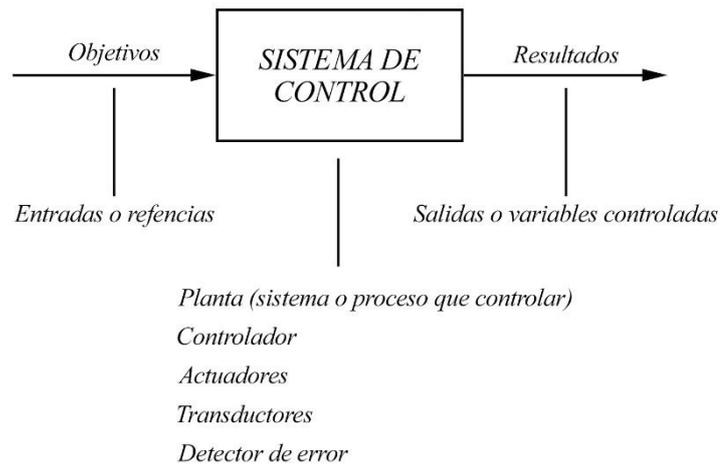


Fig. 2.13. Diagrama de un Sistema de Control. [42]

2.2.7. MICROCONTROLADORES

Un microcontrolador es un circuito integrado que posee memoria para almacenar programas, un procesador para ejecutar los programas y periféricos de entrada/salida para conexión de interrupciones, sensores o actuadores. Un microcontrolador puede reemplazar a un circuito electrónico completo, dentro de algunas de las ventajas están el aumento de la confiabilidad por el uso de menor número de componentes, simplificación de ensamblaje de sistemas, flexibilidad y adaptabilidad al hardware electrónico, rapidez en las modificaciones y desarrollo del producto mediante cambios a nivel de software. En la figura 2.14 se ilustra una configuración de microcontrolador con sus elementos principales. [43]

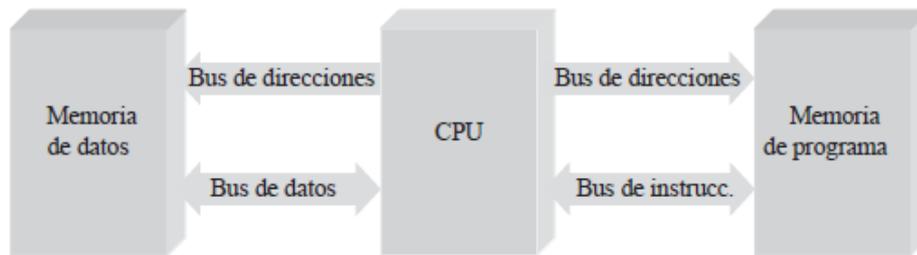


Fig. 2.14. Esquema general de la arquitectura de un microcontrolador (Arquitectura Harvard). [44]

2.2.8. SISTEMAS EMBEBIDOS

Un sistema Embebido es un dispositivo controlado por un procesador, dedicado a realizar una única tarea o una serie de ellas. Algunos sistemas embebidos son únicos en su tipo o se construyen para proyectos específicos. Por ejemplo: Embebidos en industria automotriz: navegador GPS, de la aviación: piloto automático, control de aterrizaje. Embebidos en comunicaciones: teléfonos celulares. Embebidos en el hogar: control de heladeras, microondas, robots que aspiran o cortan pasto. Etcétera. Todos están compuestos por software y hardware diseñados específicamente para la tarea que tienen que cumplir, e interactuando muy cercanamente. En los últimos años alrededor de 500 millones de microprocesadores se usaron en computadores y 10 mil millones en embebidos. Los microcontroladores se encuentran presentes en numerosos sistemas embebidos. [45], [46]

2.2.9. IMÁGENES DIGITALES

Una imagen digital es una representación visual en 2 dimensiones de un objeto, está conformada por matrices generalmente binarias con características vectoriales o netamente de bits. Una imagen digital puede almacenarse en un sistema computarizado, bajo distintos formatos. [47]

a) ELEMENTOS PRINCIPALES DE LAS IMÁGENES DIGITALES

Los elementos de las imágenes digitales son las características intrínsecas que pueden ser modificadas y se listan a continuación.

PIXELES

Un pixel es la unidad mínima de color homogéneo que se puede visualizar en una pantalla de acuerdo a la configuración de una imagen. El color de los pixeles deriva de

la combinación de 3 colores base que son rojo, verde y azul conocida como RGB por las siglas en inglés. [47]

RESOLUCIÓN DE IMÁGENES

La resolución de una imagen es la calidad o el nivel de detalle que puede apreciarse en dicha imagen digital. La resolución está ligada directamente con la cantidad de píxeles y es medida en píxeles por pulgada (ppp), es decir, el número de píxeles que caben en una pulgada por lo que, a mayor densidad de píxeles, mayor resolución. [48]

RELACIÓN DE ASPECTO

La relación de aspecto de una imagen digital o una pantalla se refiere a la proporción entre el ancho y altura de la misma.

2.2.10. PROCESAMIENTO DIGITAL DE SEÑALES

El procesamiento de señal digital (DSP) se refiere a la representación de señales en forma digital junto con el procesamiento de estas señales e información que transportan. Aunque la DSP, tal como se conoce hoy en día, comenzó a florecer en la década de 1960, algunas de las técnicas de procesamiento importantes y poderosas que se utilizan en la actualidad pueden remontarse a algoritmos numéricos que se propusieron y estudiaron hace siglos. Desde la década de 1970, cuando se introdujeron los primeros circuitos integrados DSP, el campo del procesamiento de señal digital ha evolucionado dramáticamente. Con un aumento tremendamente rápido en la velocidad de los procesadores DSP, junto con un aumento correspondiente en su sofisticación y potencia computacional, el procesamiento de señal digital se ha convertido en una parte integral de muchos productos y aplicaciones comerciales, y se está convirtiendo en un término común. [49]

a) PROCESAMIENTO DE IMÁGENES

“El procesamiento de imágenes es la disciplina orientada a obtener, mediante herramientas de programación, imágenes mejoradas o modificadas con efectos especiales para un propósito específico”. [50]

b) HERRAMIENTAS PARA EL PROCESAMIENTO DE IMÁGENES

A nivel de software se puede emplear un sinnúmero de programas orientados a distintas tareas y sobre todo crear nuevos programas en base a las necesidades de los usuarios bajo distintos lenguajes de programación, librerías o módulos.

Para procesar imágenes y crear una interfaz que permita interactuar entre el sistema computarizado y el usuario se debe definir un lenguaje de programación además de los módulos, funciones, métodos o librerías necesarias de acuerdo a los requerimientos. A continuación, se definen algunas herramientas de gran utilidad para el desarrollo de aplicaciones, orientadas al procesamiento de imágenes.

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. [51]

TKINTER

Tkinter es un paquete de herramientas, métodos o librerías y principalmente la interfaz estándar de Python. Esta interfaz como tal no es parte de Python, pero lo complementa y de acuerdo al sistema operativo puede extenderse de mejor manera como en el caso de las plataformas Unix. [51]

OPENCV

OpenCV (Open Source Computer Vision Library) es una herramienta computarizada bajo licencia de software libre, trabaja bajo interfaces que se desarrollan en lenguajes de programación como C++, C, Python y Java y tiene soporte para varios sistemas operativos como Windows, Linux, Mac OS, iOS y Android. OpenCV se enfoca en aplicaciones de tiempo real, puede aprovechar el procesamiento multi-core, habilitado con OpenCL, puede aprovechar la aceleración de hardware de las plataformas

computarizadas y se especializa en la adquisición y manipulación de la información visual de imagen o video. [52]

2.2.11. BASES DE DATOS

Una base de datos, que se abrevia como BD, es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible. Los datos pueden ser utilizados por diferentes programas y diferentes usuarios. Por lo tanto, el concepto de base de datos generalmente está relacionado con el de red, ya que se debe poder compartir esta información. A la base de datos se la conoce también como un Sistema de información para designar a la estructura global que incluye todos los mecanismos para compartir datos. [53]

a) HERRAMIENTAS PARA EL MANEJO DE BASES DE DATOS

Para la creación, administración o monitoreo de bases de datos se puede emplear varios programas como los que se definen a continuación.

MYSQL

MySQL es un intérprete de órdenes SQL que es un tipo lenguaje vinculado con las operaciones que se pueden realizar en una base de datos. MySQL posee capacidades de edición a través de comandos mediante los cuales permite crear y manipular las bases de datos pudiendo obtener resultados en forma de tablas. [54]

MARIADB

MariaDB es un servidor de bases de datos creado por los desarrolladores originales de MySQL bajo licencia de software libre. Este servidor es muy versátil, convierte los datos en información estructurada y la vincula fácilmente con sitios web. [55]

PHPMYADMIN

PhpMyAdmin es una herramienta de software libre con una interfaz gráfica escrita en PHP, que permite administrar un amplio rango de funciones de MySQL y MariaDB a través de la web. En esta interfaz se puede manipular las bases de datos de forma más intuitiva a través de tablas, índices, usuarios, permisos, entre otros. [56]

2.3 PROPUESTA DE SOLUCIÓN

La implementación del sistema automatizado, encargado del análisis y detección de los huevos de gallina aptos para incubación permite incrementar la calidad de la producción avícola optimizando tiempo y recursos.

CAPÍTULO III

METODOLOGÍA

3.1 MODALIDAD DE LA INVESTIGACIÓN

El proyecto se desarrolló con investigación aplicada para determinar los problemas que se presentan al incubar los huevos de gallina tanto en el aspecto económico, recursivo y temporal, información relevante que conllevó a proponer una solución integral basada en sistemas de control electrónicos de nueva generación. Se tomó en consideración lo siguiente:

La investigación tiene característica bibliográfica buscando soluciones al déficit de producción de pollos y pérdidas económicas que representan a través de libros, tesis previas, publicaciones científicas, fichas bibliográficas, fichas nemotécnicas e informes estadísticos.

En el proyecto se empleó la investigación de campo para obtener información relacionada a los métodos utilizados para los controles de calidad de los huevos de gallina aptos para incubación verificando la efectividad de dichos procesos en distintas empresas dedicadas a la incubación.

Se efectuó una investigación de tipo experimental para el manejo de los huevos de gallina que estuvieron sujetos a las pruebas pertinentes y la verificación de resultados en cuanto refiere a los procesos productivos de pollos de incubadora.

3.2 POBLACIÓN Y MUESTRA

Por las características de la investigación no se requiere población y muestra.

3.3 RECOLECCIÓN DE INFORMACIÓN

Al ser una investigación de campo se empleó guías de observación valorando el proceso más óptimo para el control de parámetros de los huevos de gallina aptos para incubar y el análisis de los mismos. Además, a través de la investigación bibliográfica se determinaron las técnicas más factibles para el manejo del sistema en cuanto concierne a interfaces amigables para visualización de resultados, procesamiento de señales, bases de datos y controladores.

3.4 PROCESAMIENTO Y ANÁLISIS DE DATOS

- Elaboración de un análisis exhaustivo en base a los criterios obtenidos del marco teórico, objetivos y enfoque de la investigación.
- Examinación de las posibles tecnologías para automatizar los procesos de análisis o verificación de los huevos de gallina, el sistema de comunicación referente al almacenamiento de datos, las ventajas y desventajas del sistema o tecnología a emplear a través de datos técnicos conseguidos en libros, revistas, artículos científicos y data sheets.
- Estructuración de una propuesta acertada con enfoque en la calidad de los huevos de gallina aptos para incubación, aumentando la producción, y el almacenamiento de los datos obtenidos en bases de datos teniendo registros más efectivos.
- Validación los datos presentación de resultados a través de diagramas o bloques estadísticos adecuados, indicando las diferenciaciones entre los parámetros y porcentajes obtenidos por el prototipo y los que actualmente se presentan.

3.5 DESARROLLO DEL PROYECTO

El proyecto se desarrolló en función de las siguientes actividades. En la figura 2.14 se relacionan las actividades que se describen a través de un flujograma.

- Analizar los principales parámetros que se verifican en los procesos previos a la incubación de huevos de gallina. Esta actividad se basa en el primer objetivo de análisis de técnicas y métodos, procurando la obtención de la información pertinente para el desarrollo del proyecto.
- Comparar los métodos actuales utilizados en los controles de calidad de huevos de gallina. En base al primer objetivo, se complementa la adquisición de la

información con el fin de establecer los requerimientos técnicos del proyecto a través del cumplimiento de esta actividad.

- Determinar el hardware y software a utilizar para el procesamiento de señales y la evaluación de los huevos de gallina aptos para incubación. En concordancia con la información adquirida, esta actividad se relaciona con los dos primeros objetivos debido al establecimiento de características técnicas para el funcionamiento del prototipo.
- Distinguir las ventajas y desventajas de los elementos o tecnología seleccionada para emplear en el diseño del sistema. Esta actividad se relaciona con el segundo objetivo donde se establece todos los componentes a emplear en el prototipo.
- Diseñar el prototipo del sistema de control de calidad junto con el medio de comunicación para el almacenamiento de datos en la base de datos. El cumplimiento de esta actividad está relacionado con el tercer objetivo enfocado en el funcionamiento del prototipo.
- Desarrollar los algoritmos necesarios para el control, manejo del sistema y base de datos. Esta actividad se relaciona con el tercer objetivo, basado en el control por software.
- Implementar una interfaz gráfica para el monitoreo del sistema dentro del proceso previo a la producción de pollos. Basada en el tercer objetivo, esta actividad permite al usuario tener el control total del prototipo.
- Realizar pruebas de funcionamiento del prototipo del sistema. Esta actividad se relaciona con los tres objetivos específicos planteados, donde se verifica la aplicabilidad del prototipo desarrollado.
- Desarrollar el informe escrito del proyecto de investigación. Posterior al desarrollo del prototipo, esta actividad, relacionada con todos los objetivos permite plasmar la información adquirida y generada por el presente proyecto.

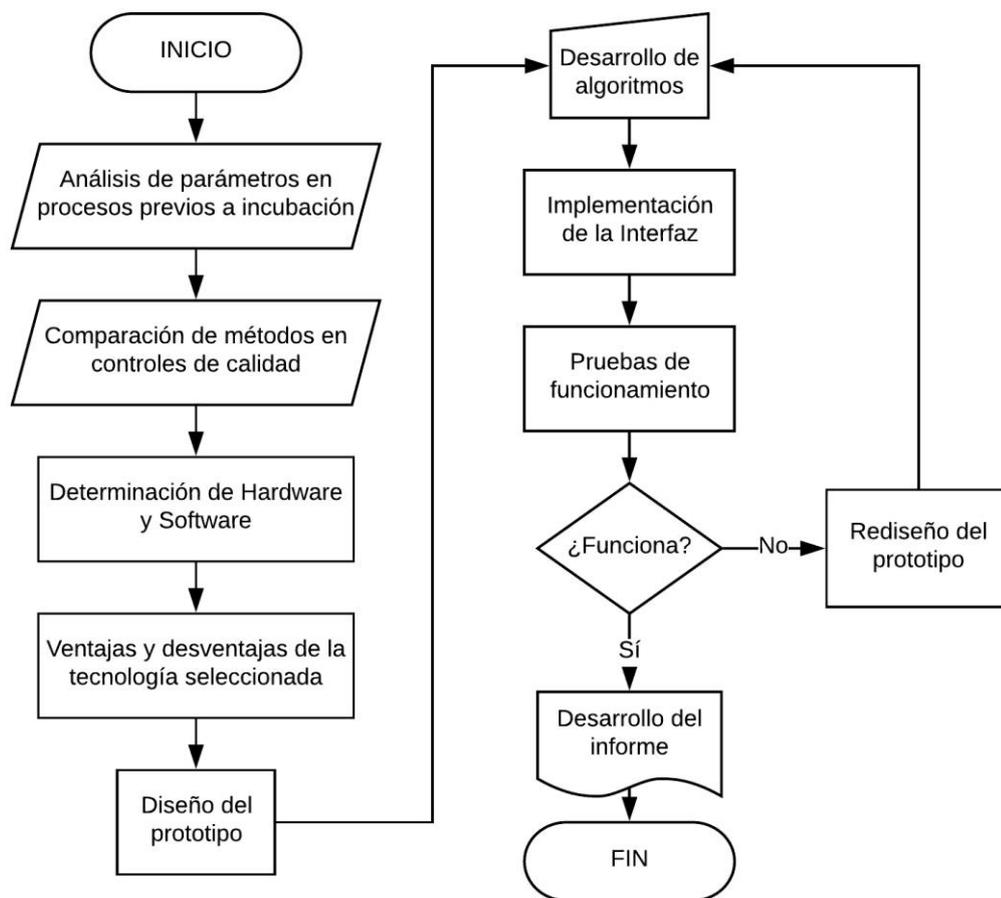


Fig. 3.1. Diagrama de flujo de las actividades del proyecto. [Investigador]

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

La Corporación Nacional de Avicultores del Ecuador afirma que el consumo de carne de pollo implica beneficios nutritivos en comparación con la carne de ganado bovino y ovino, puesto que contiene menos colesterol, calorías y grasa, además, provee mayores cantidades de proteína necesarias para el consumo humano, solo la carne de pavo y avestruz supera los beneficios de la carne de pollo, pero tiene un costo más elevado. La comunidad ecuatoriana demanda una mayor producción de pollo por los beneficios de su carne y de las proyecciones del consumo per cápita anual a nivel nacional que es de 35 Kg para que llegue a los valores estimados de países como Estados Unidos que es de 54,3 Kg o Brasil que es de 58,1 Kg que son algunos de los mayores consumidores de carne de pollo a Nivel Mundial. Por lo descrito anteriormente es necesaria la optimización de los procesos productivos. [1], [2], [6], [7], [8]

Los análisis tradicionales o controles de calidad son manuales, sobre todo en las pequeñas y medianas empresas dedicadas a la incubación de huevos de gallina, lo que puede traer problemas de precisión y otros errores humanos afectando a la cantidad y calidad de producción. Los sistemas automatizados, mejoran la calidad de los productos, permitiendo, además, clasificar y descartar huevos de gallina de mala calidad e incluso reutilizar aquellos que aún pueden servir como alimento para el consumo humano. [10]

Los sistemas de control de calidad buscan mayor robustez y al ser más confiables y eficientes, se asegura una mayor producción avícola, reduciendo así las pérdidas económicas y optimizando recursos, por lo cual, el sistema de procesamiento de imágenes o visión computarizada para el análisis de los parámetros internos de los

huevos de gallina previo a la incubación se presenta como una alternativa eficaz en los procesos productivos, que conlleva no solo al control de calidad sino también a análisis estadísticos con el fin de determinar porcentajes o índices de productividad de las empresas y adicionalmente la eficacia de los proveedores. Además, el sistema implementado se establece como una herramienta escalable que puede aplicarse y replicarse en pequeñas, medianas o grandes empresas dedicadas a la producción de pollos mediante incubación artificial, en concordancia con sus propios requerimientos.

4.1 ANÁLISIS DE FACTIBILIDAD

4.1.1. FACTIBILIDAD TÉCNICA

El presente proyecto de investigación es técnicamente factible debido a que los equipos, elementos electrónicos y mecánicos requeridos son comerciales y se pueden adquirir fácilmente dentro del país. Además, se emplea tecnología basada en hardware y software libre.

4.1.2. FACTIBILIDAD ECONÓMICA

El desarrollo del presente proyecto fue factible económicamente debido a que el investigador cubrió todos los gastos del mismo.

4.1.3. FACTIBILIDAD BIBLIOGRÁFICA

La información requerida para este proyecto es de libre acceso y se obtuvo de libros, publicaciones científicas, repositorios de la Universidad Técnica de Ambato, tesis e internet.

4.2 REQUERIMIENTOS DEL PROTOTIPO

Para llevar a cabo este proyecto fue necesario que el sistema sea modular y escalable de acuerdo con las características fisiológicas que presentan los huevos de gallina. El enfoque del sistema es seleccionar los huevos de gallina idóneos para incubación a través de la observación a trasluz, dicho de otro modo, colocando los objetos de estudio sobre un sistema de iluminación especial con el fin de verificar sus parámetros internos o condiciones externas inadecuadas como son:

- Suciedad.
- Fisuras u orificios.
- Doble yema.

- Exceso de poros.
- Cáscara fina.
- Tamaño y fisionomía irregulares.

Este análisis, emplea técnicas para la adquisición de imágenes a través de una cámara digital, lo que implica un sistema automatizado que puede desplazarse sobre las matrices de huevos de gallina que se forman dentro del prototipo, el procesamiento de esta información se lleva a cabo a través de una microcomputadora, encargada de determinar los huevos idóneos, almacenar la información y controlar el hardware del sistema.

4.2.1. HARDWARE

Para la instalación del sistema de control de calidad de los huevos de gallina, se requiere principalmente los siguientes componentes:

- La Cámara Digital es la encargada de adquirir y transmitir las imágenes para el análisis en la microcomputadora.
- La Microcomputadora es la parte principal del sistema donde se debe ejecutar los algoritmos para la selección de los huevos de gallina y alojar la información en una base de datos para análisis estadísticos.
- La Pantalla es el dispositivo encargado de la interacción hombre-máquina a través de una interfaz que permita brindar la información que se requiera.
- La Estructura es el soporte principal que debe contener en su conjunto cada uno de los componentes del sistema y, además, alojar a los elementos de análisis.
- El Sistema de Iluminación es el encargado de generar los haces de luz adecuados para la visualización de los parámetros de los objetos de estudio.
- El Sistema de Desplazamiento permite trasladar la cámara sobre los objetos de estudio, que deben ser colocados a contraluz, en base a la configuración de las matrices planteadas para recolección de información.

En la figura 4.1 se muestra un bosquejo de la interacción entre los elementos principales del prototipo tanto en hardware como en software.

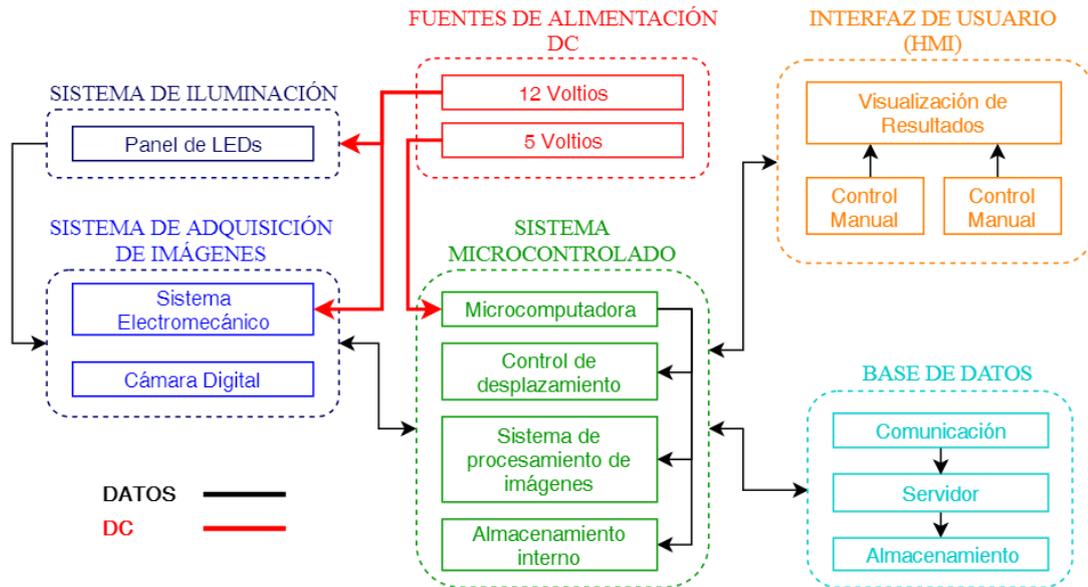


Fig. 4.1. Diagrama de bloques del sistema de control de calidad de huevos de gallina. [Investigador]

4.2.2. SOFTWARE

El presente proyecto requirió un programa que disponga de herramientas matemáticas de alto nivel, para realizar toda la codificación de los algoritmos implícitos en el procesamiento digital de imágenes permitiendo identificar los parámetros de los huevos de gallina especificados anteriormente. Además, el sistema informático permite almacenar información, interactuar con el usuario y configurar entradas y salidas de propósito general para control de los actuadores como que son los motores encargados de desplazar la cámara dentro del prototipo.

4.3 DISEÑO DEL PROTOTIPO

Este sistema se basa en las siguientes etapas o subsistemas complementarios, cada uno de los subsistemas requiere varios componentes, análisis y procesos para su desarrollo, los mismos que se detallan a continuación. En la figura 4.2 se presenta la interconexión de los dispositivos necesarios para el funcionamiento del prototipo, la selección de estos componentes se justifica en los siguientes puntos.

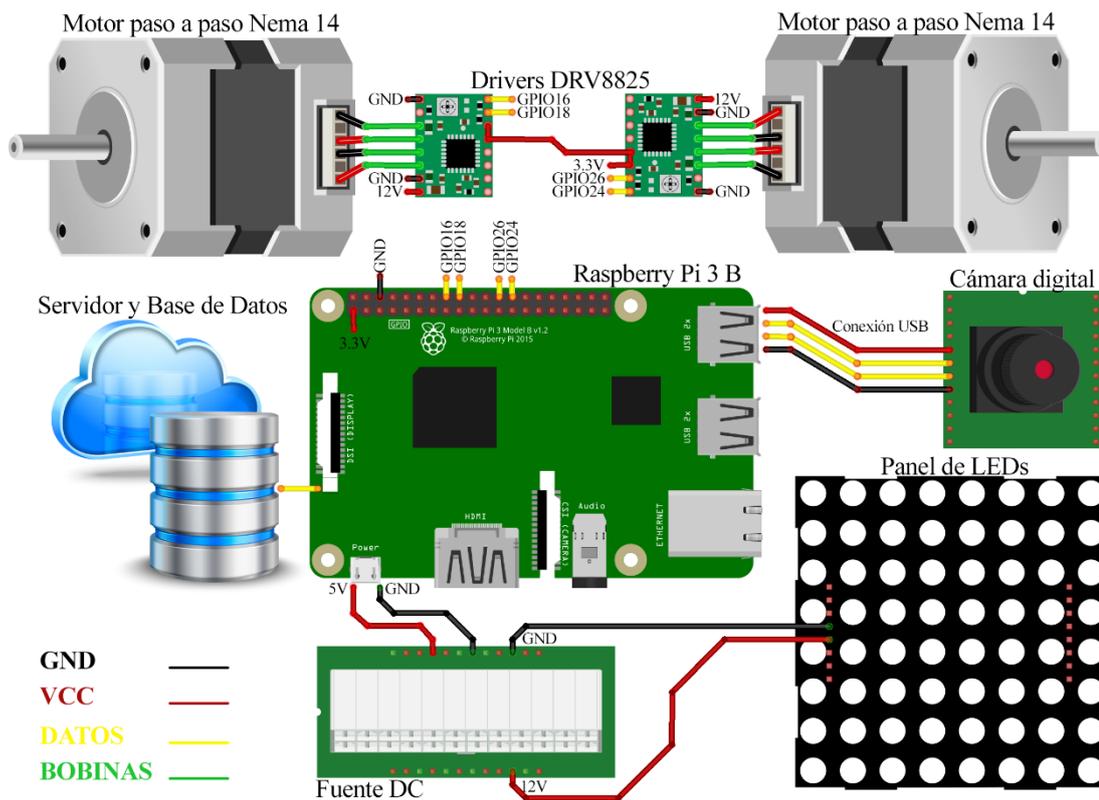


Fig. 4.2. Esquema de conexiones del prototipo de control de calidad de huevos de gallina.
[Investigador]

4.3.1. ESTRUCTURA DEL PROTOTIPO

La Estructura es la base metálica que aloja cada uno de los subsistemas y los productos que deben ser analizados. Esta estructura se divide en 2 partes: una base fija que alojará los elementos electrónicos, encargados de realizar el análisis y una base móvil que alojará los huevos de gallina que son el objeto de estudio. La base móvil, similar a una cubeta tradicional pero adaptada para que los huevos queden a trasluz, podrá insertarse y retirarse con el fin de cambiar los productos para los análisis de calidad. Los demás componentes, como el sistema de adquisición de imágenes y toda la electrónica que conlleva el prototipo está fijado en la base principal procurando que todo el sistema sea compacto y adaptable.

a) DISEÑO DE LA ESTRUCTURA DEL PROTOTIPO

En la figura 4.3 se muestra el bosquejo de la base del prototipo enfocada principalmente en una distribución matricial de los huevos de gallina para el respectivo análisis de acuerdo con los parámetros que se detalla a continuación. La base móvil toma como referencia el tamaño promedio de los huevos de gallina para incubación.

Además, para el diseño, se realiza el cálculo de las dimensiones físicas en cm en base a la cantidad de huevos que se emplearán por análisis en una matriz de 6 x 6 que puede subdividirse en 4 matrices de 3 x 3. Entonces, considerando las dimensiones promedio de los huevos de gallina se obtiene lo siguiente:

D1 = Diámetro mayor de un huevo de gallina = 6 cm

D2 = Diámetro menor de un huevo de gallina = 4,5 cm

Al considerar la matriz de 6 x 6 establecida, se obtiene las siguientes dimensiones:

B = Dimensión horizontal de la base, Ecuación 4.1

$$B = D1 * 6 \quad (4.1)$$

$$B = 6 \text{ cm} * 6$$

$$\mathbf{B = 36 \text{ cm}}$$

H = Dimensión vertical de la base, Ecuación 4.2

$$H = D2 * 6 \quad (4.2)$$

$$H = 4,5 \text{ cm} * 6$$

$$\mathbf{H = 27 \text{ cm}}$$

Adicionando al menos 1,5 cm entre cada huevo de gallina y 3 cm hacia los bordes, implica 5 espacios de 1,5 cm y 2 espacios de 3 cm:

B1 = Dimensión horizontal de la base con espaciamiento, Ecuación 4.3

$$B1 = B + (1,5 \text{ cm} * 5) + (3 \text{ cm} * 2) \quad (4.3)$$

$$B1 = 36 \text{ cm} + (7,5 \text{ cm}) + (6 \text{ cm})$$

$$\mathbf{B1 = 49,5 \text{ cm}}$$

H1 = Dimensión vertical de la base con espaciamiento, Ecuación 4.4

$$H1 = H + (1,5 \text{ cm} * 5) + (3 \text{ cm} * 2) \quad (4.4)$$

$$H1 = 27 \text{ cm} + (7,5 \text{ cm}) + (6 \text{ cm})$$

$$\mathbf{H1 = 40,5 \text{ cm}}$$

Al obtener las dimensiones físicas de la base móvil, sobre la cual se desplaza la cámara, éstas permitieron establecer las dimensiones de la base fija principal y del sistema de desplazamiento. Además, para evitar el contacto directo de los huevos con el metal, se adicionó una lámina de Foamy cuyos orificios se superponen a los de la base metálica sobresaliendo 3mm del borde hacia el interior con el fin de sujetar los huevos al colocarlos.

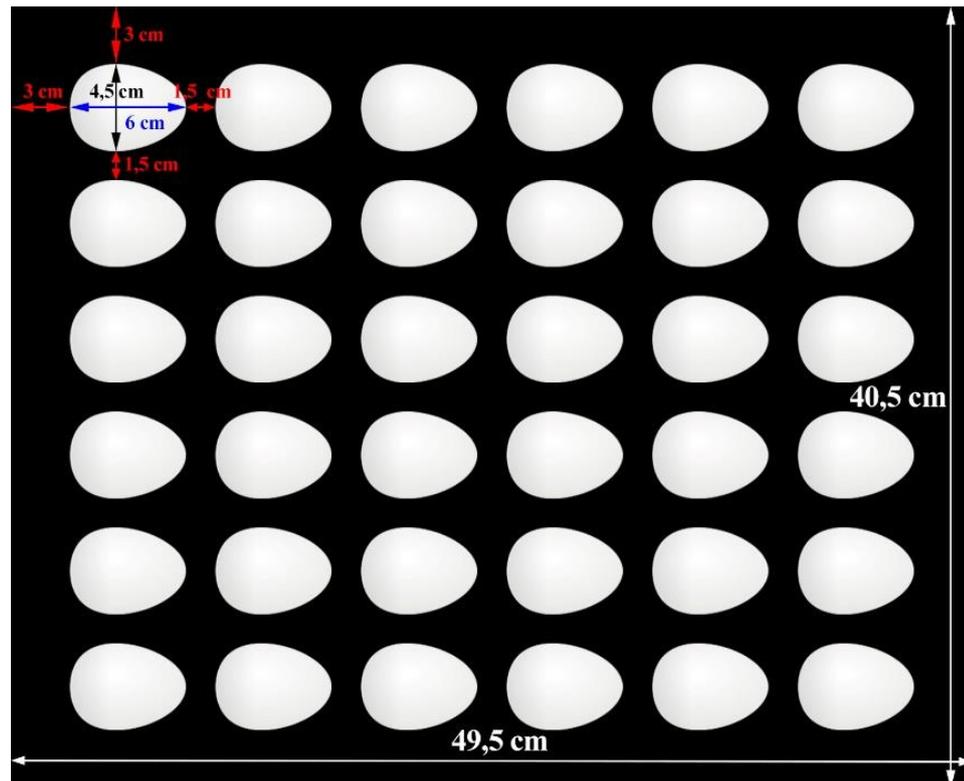


Fig. 4.3. Esquema del sistema estructural del prototipo (base móvil). [Investigador]

En la base fija se sitúa el sistema de iluminación, quedando posicionado por debajo de la cubeta, y también el sistema de desplazamiento los cuales se detallan más adelante. En esta base se adicionan 2 centímetros a las dimensiones de la base móvil para robustecer la estructura y para que la inserción de la base móvil sea sencilla y compacta como se ilustra en la figura 4.4. Además de lo indicado se acopló orificios a distintas alturas para poder regular manualmente mediante sujeción con pernos y tuercas, el sistema de desplazamiento, lo que permite adecuar la cámara que se emplee en la mejor posición para adquirir las imágenes.

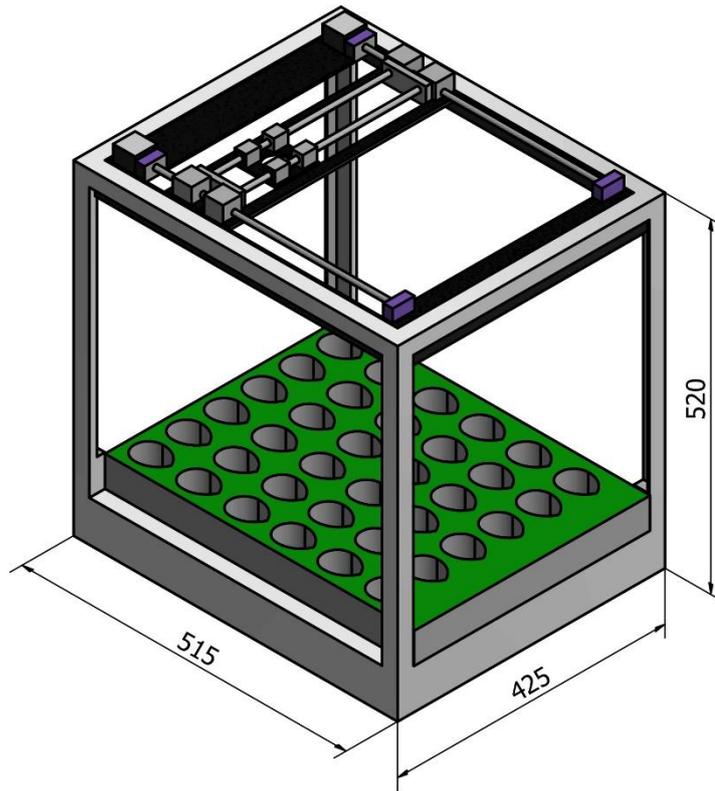


Fig. 4.4. Esquema completo de la estructura del prototipo. [Investigador]

4.3.2. SISTEMA DE ILUMINACIÓN

El Sistema de Iluminación es una base que emite energía lumínica y se coloca en la parte inferior de la base móvil de la estructura debido a que se debe iluminar los objetos de estudio, debido a su translucidez, para así poder observar sus parámetros internos mediante el sistema de adquisición de imágenes. Esta energía lumínica no debe afectar al proceso de incubación de los huevos de gallina y, además, no debe generar calor evitando así problemas internos en los productos previo a su incubación.

a) DISEÑO DEL SISTEMA DE ILUMINACIÓN

En la tabla 4.1 se detallan las características de las bombillas comerciales de iluminación habituales en el mercado. Generalmente se emplean bombillas comunes en sistemas de ovoscopia, un ovoscopio comercial genera alrededor de 500 lúmenes por lo cual, se toma como referencia para el análisis y selección, la iluminación equivalente a 450 y 800 lúmenes que son comercializados de forma estándar. [35]

Tabla 4.1 Características de distintos tipos de bombillas para iluminación. [57]

450 lúmenes				
	Bombilla Incandescente	Bombilla Halógena	Lámpara Fluorecente Compacta	Bombilla LED
Potencia (Vatios)	40	29	9	8
Vida útil (Años)	1	1 – 3	6 – 10	15 – 25
Emisión de calor	Alta	Alta	Media	Baja
Costo Aproximado (\$)	1,5	6	2	3
800 lúmenes				
	Bombilla Incandescente	Bombilla Halógena	Lámpara Fluorecente Compacta	Bombilla LED
Potencia (Vatios)	60	43	14	13
Vida útil (Años)	1	1 – 3	6 – 10	15 – 25
Emisión de calor	Alta	Alta	Media	Baja
Costo Aproximado (\$)	2	7,5	3	4

En base a las características de los dispositivos y tecnologías empleadas para iluminación, se selecciona la Luz LED para el análisis de los huevos de gallina a trasluz debido a su costo relativamente bajo con ventajas notables como mayor vida útil, bajo consumo de energía y baja emisión de calor, evitando alteraciones en el proceso de incubación.

Una vez seleccionada la tecnología LED se desarrolla la placa para ubicar los elementos. Se considera conexiones en serie y en paralelo para un mejor manejo de la potencia de consumo del sistema, reduciendo la disipación de calor en la base con el fin de no interferir en el proceso de incubación.

En el presente proyecto se emplean LEDs de color blanco por su mayor intensidad lumínica. Se plantea un circuito base que se muestra en la figura 4.5, considerando los siguientes parámetros y aplicando la Ley de Ohm se calcula la resistencia requerida:

$$V_A = \text{Voltaje de Alimentación} = 12 \text{ V}$$

$V_{LED} = \text{Voltaje de LED (Blanco)} = 3 \text{ V}$

$I_{LED} = \text{Intensidad de LED (Blanco)} = 20 \text{ mA}$

$R = \text{Resistencia Requerida en Ohmios, Ecuación 4.5}$

Debido a las caídas de tensión sobre los leds y la resistencia para limitar la corriente, se colocan 3 leds en serie, formando una rama del circuito total.

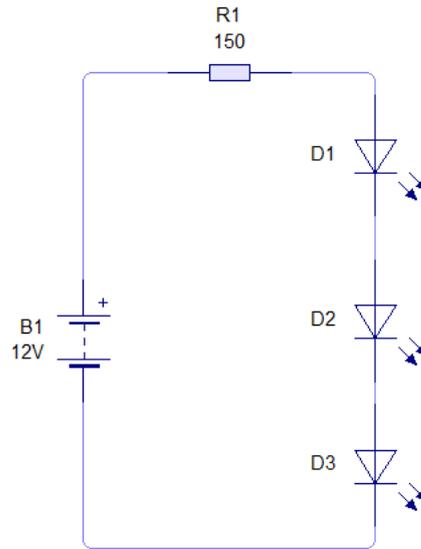


Fig. 4.5. Simulación del circuito base para el sistema de iluminación. [Investigador]

$$R = \frac{V_A(V) - 3 * V_{LED}(V)}{I_{LED}(A)} \quad (4.5)$$

$$R = \frac{12(V) - 3 * 3(V)}{0,02(A)}$$

$$R = 150 \Omega$$

$P = \text{Potencia disipada por la resistencia, Ecuación 4.6}$

$$P = [I_{LED}(A)]^2 * R(\Omega) \quad (4.6)$$

$$P = [0,02(A)]^2 * 150(\Omega)$$

$$P = 0,06 \text{ W}$$

La resistencia calculada coincide con un valor de resistencia comercial y debido a la potencia también calculada, es suficiente emplear elementos de $\frac{1}{4} \text{ W}$.

De acuerdo con las dimensiones del prototipo, a una distancia promedio de 1,5 cm entre LEDs de 5 mm estándar, se calcula 216 elementos de emisión de luz distribuidos de forma tal que se ubiquen 6 elementos bajo cada huevo, empleando 72 resistencias limitadoras, es decir, 1 resistencia por cada 3 LEDs en serie, lo que indica 72 ramas en paralelo en el circuito.

I_R = Corriente en cada rama

X = Número de ramas

I_T = Corriente total del circuito, Ecuación 4.7

$$I_T = I_R(A) * X \quad (4.7)$$

$$I_T = 0,02(A) * 72$$

$$I_T = 1,44 (A)$$

Fuente Requerida:

Se establece una fuente de alimentación de **12 V** que pueda entregar al menos **1,44 A** que es la corriente total a través del circuito de iluminación o en su defecto 1,5 A, que se puede encontrar en fuentes comerciales.

Lúmenes a generar por el sistema de iluminación:

Cada led, de acuerdo con los parámetros calculados, genera entre 3 y 5 lúmenes, por lo tanto, el sistema de iluminación debe generar entre **648 y 1080 lúmenes**. [58]

En la figura 4.6 se ilustra una simulación de los ramales en serie y en paralelo del circuito determinado.

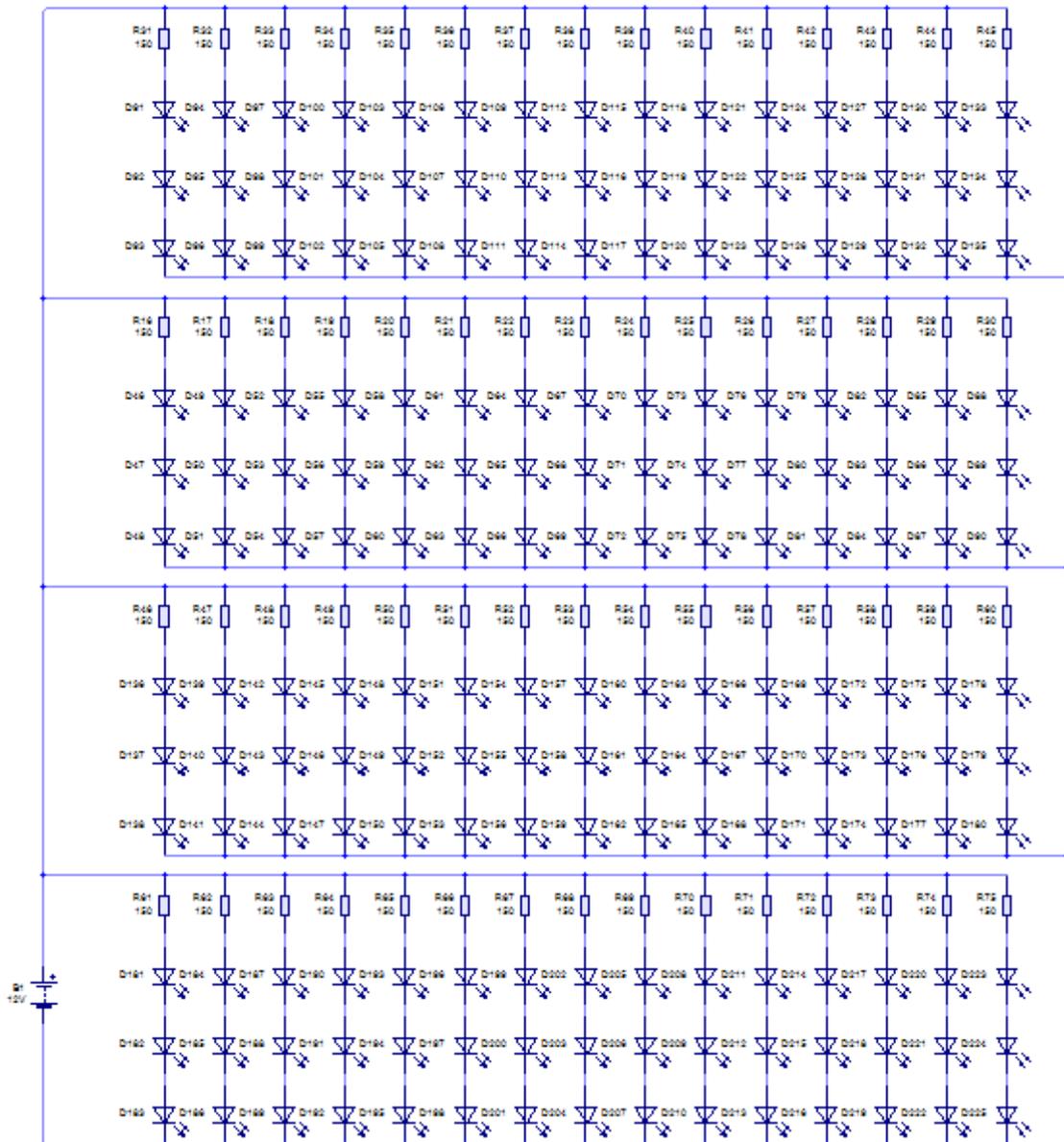


Fig. 4.6. Simulación del sistema de iluminación. [Investigador]

En la figura 4.7 se muestra la ubicación de los componentes en la placa electrónica para la generación de luz necesaria para el sistema, distribuidos de acuerdo con la matriz planteada.

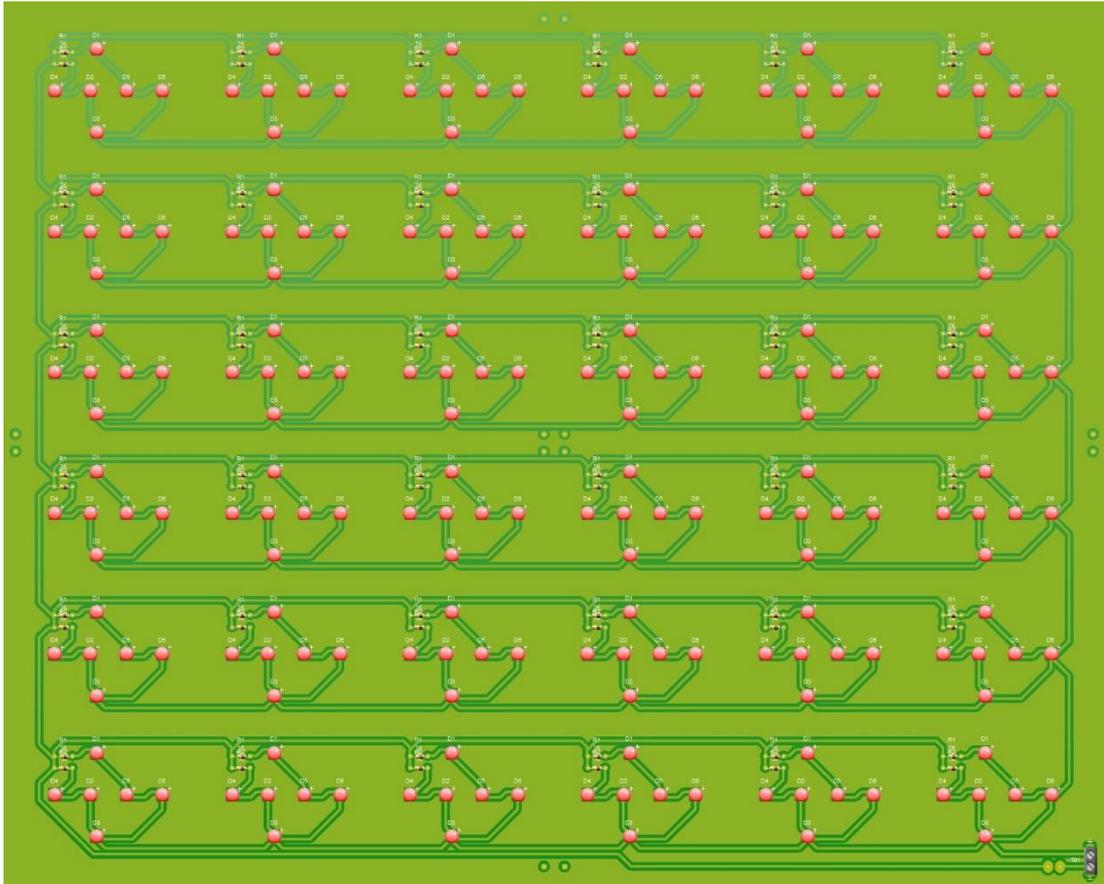


Fig. 4.7. Ubicación de componentes del sistema de iluminación sobre placa electrónica. [Investigador]

En la figura 4.8 se muestra el diseño del circuito impreso para la elaboración de la placa electrónica del sistema de iluminación.

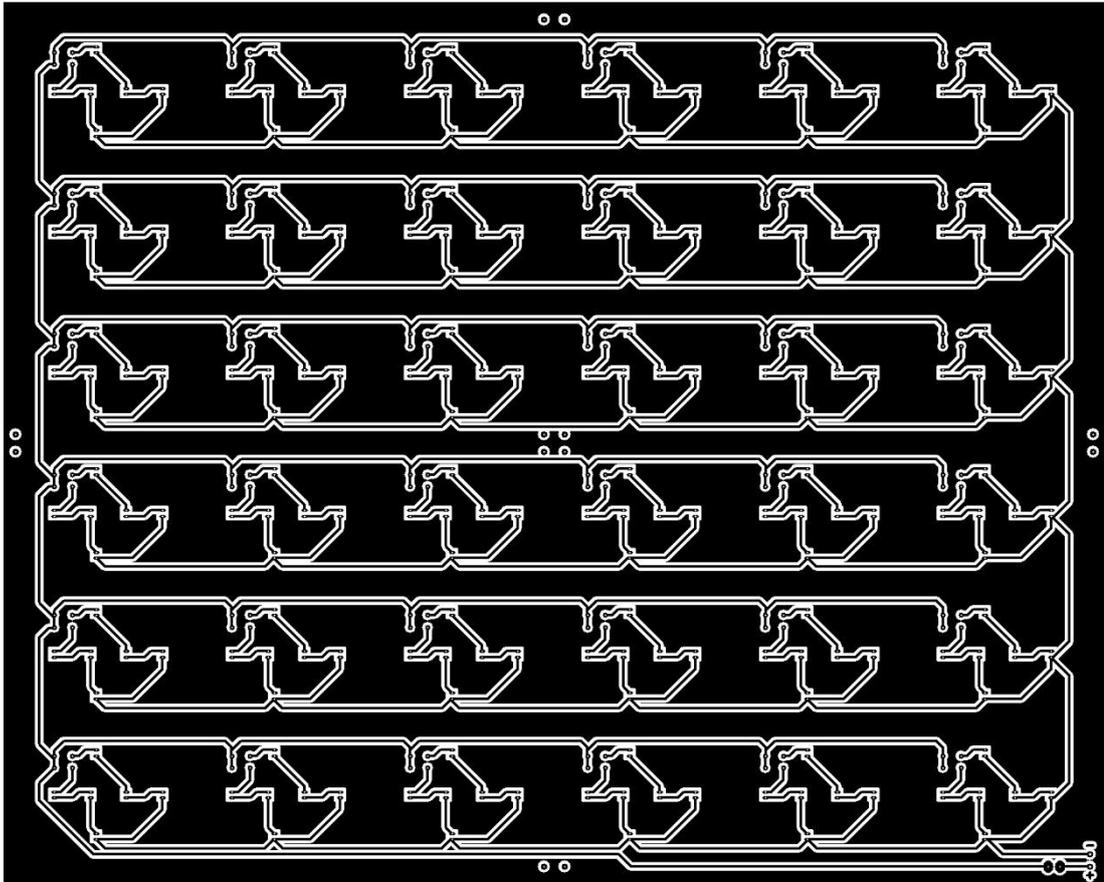


Fig. 4.8. Diseño de placa electrónica del sistema de iluminación. [Investigador]

4.3.3. SISTEMA PARA EL PROCESAMIENTO DE IMÁGENES

El Sistema para el Procesamiento de Imágenes es un dispositivo microcontrolado o computarizado que se encarga de analizar las imágenes que adquiera la cámara; en esta sección del sistema, se realiza la selección de los huevos idóneos para el proceso de incubación en base a los parámetros y algoritmos que se determinen y ejecuten, además, este sistema se encarga del control electrónico, brindar la información requerida a la base de datos e interactuar con el usuario a través de la interfaz creada.

a) SELECCIÓN DEL SISTEMA MICROCONTROLADO PARA EL PROCESAMIENTO DE IMÁGENES

En la tabla 4.2 se detallan las características de placas de desarrollo a nivel micro computarizado, comunes en el mercado.

Tabla 4.2 Características técnicas de dispositivos de control. [59], [60], [61]

Dispositivo Especificación	Raspberry Pi	BeagleBone	pcDuino
Modelo	3 B	Black	3 B
Tamaño	85 x 56 mm	86.36 x 53.34 mm	121 x 65mm
Procesador	Broadcom BCM2387 Quad- Core ARM Cortex- A53 de 64 bits y cuatro núcleos	Sitara AM3359 ARM Cortex A8	AllWinner A20 SoC, ARM Cortex A7 Dual Core
Velocidad de Reloj	1.2 GHz	1 Ghz	1GHz
Memoria RAM	1 GB LPDDR2	512MB DDR3	1 GB
Almacenamiento Interno	No	4 GB Flash eMMC	4GB Flash
GPIO	40 pines	69 pines	14 pines
Comunicación Inalámbrica	LAN inalámbrica 802.11 b/g/n - 2.4 Ghz Bluetooth 4.1 Low Energy (BLE)	No	Built-in WiFi
GPU	VideoCore IV de doble núcleo 400 Mhz – Gráficos 3D	3D graphics accelerator	OpenGL ES2.0, OpenVG 1.1, Mali 400 Dual Core
Ethernet	1 puerto Ethernet 10/100	1 puerto Ethernet 10/100	Ethernet 10M/100Mbps s/1Gbps
Micro SD	Clase 10, de al menos 8 GB	Puerto disponible para expansión de memoria	4GB Flash, microSD card (TF) hasta 32GB

Puertos USB	4 puertos 2.0	4 puertos 2.0	1 x USB host, 1xUSB OTG
Salida de Audio	Jack, 3.5 mm	HDMI	Jack, 3.5mm I2S stereo digital audio interface
Salida de Video	Jack, HDMI	Jack, microHDMI	HDMI 1.4 with HDCP support
Sistemas Operativos Soportados	Compatible con todas las últimas distribuciones de ARM GNU/Linux, Android y Windows 10 IoT	Debian, Android, Ubuntu, Cloud9 IDE on Node.js w/ BoneScript library plus much more	Ubuntu 12.04 Android 4.2
Consumo	2.5 A / 5 V	210-460 mA / 5V	2000mA / 5V
Costo aproximado	\$60	\$105	\$60

En base a las características de las microcomputadoras analizadas, se seleccionó la Raspberry Pi 3 B para el desarrollo del prototipo, debido a su mejor procesador y velocidad de reloj, la facilidad de conexión de varios elementos electrónicos, además de ser un sistema informático completo y contar con conexiones de red inalámbrica y ethernet, compatibilidad con mayor cantidad de sistemas operativos y mayor acceso a información como tutoriales de apoyo para proyectos. Además de los elementos descritos, es una de las placas de desarrollo más potente, accesible en el mercado ecuatoriano y sobre todo económica.

Una vez seleccionado la microcomputadora Raspberry Pi 3 Modelo B se procede a instalar el sistema operativo en una tarjeta de memoria micro SD debido a que el dispositivo no cuenta con memoria interna. Se considera el sistema operativo oficial “Raspbian”, de la tarjeta de control debido a que cuenta con los elementos de software necesarios para su manipulación.

En la figura 4.9 se muestra otros sistemas operativos disponibles que pueden ejecutarse en Raspberry.

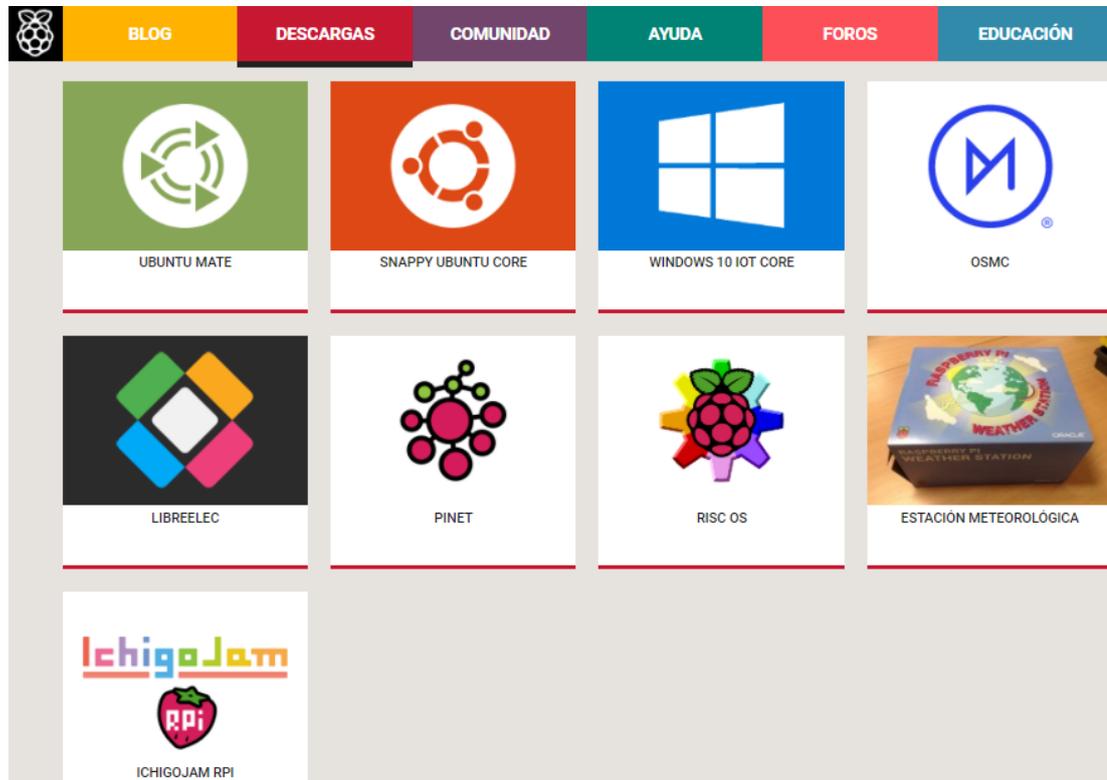


Fig. 4.9. Sistemas Operativos Alternativos para Raspberry. [62]

Posterior a la instalación del Sistema Operativo en la microcomputadora, se procede a instalar las herramientas de software para el control de movimiento, procesamiento de imágenes, interfaz y base de datos, que, por las características del sistema, deberá ser de código libre.

4.3.4. SISTEMA PARA LA ADQUISICIÓN DE IMÁGENES

El Sistema para la Adquisición de Imágenes se establece como una base controlada que se posicionará sobre los objetos de estudio, una vez insertados en la base principal. Este sistema será capaz de desplazarse en 2 dimensiones, permitiendo un mejor control y alcance del dispositivo de adquisición de imágenes.

a) SELECCIÓN DEL DISPOSITIVO PARA ADQUISICIÓN DE IMÁGENES

El dispositivo de adquisición de imágenes se basa en una cámara digital que es la encargada transmitir la información al sistema microcontrolado. En la tabla 4.3 se

detallan características generales de cámaras digitales que pueden vincularse con la microcomputadora seleccionada.

Tabla 4.3 Características técnicas de dispositivos de adquisición de imágenes (cámaras digitales).
[63], [64], [65]

	Raspberry Pi Camera Module v2	Cámara Web Genius FaceCam 1000x	Cámara Web Ins IS-W811
Tamaño Aproximado	25 × 24 × 9 mm	20 x 22 x 60 mm	59.3 x 24.6 x 41.1 mm
Peso	3 gramos	50 gramos	58 gramos
Resolución	8 megapíxeles	1 megapíxel	12 megapíxeles
Ángulo de Visión	62.2°	60°	30°
Fotogramas por segundo	30, 60 o 90	30	30
Interfaz de programación de aplicaciones	OpenMAX IL y otros disponibles	No disponible.	No disponible.
Voltaje de entrada	5V	5V	5V
Sensor	Sony IMX219	CMOS	CMOS
Zoom óptico	No disponible	Regulable	Extendido
Extensión del cable de conexión	0,15 m	1,5 m	1 m
Tipo de Conexión	CSI	USB	USB
Precio Aproximado	\$50	\$18	\$10

En base a las características de las cámaras analizadas, se selecciona la Cámara Web Genius FaceCam 1000x debido a su fácil conexión con la microcomputadora, gran ángulo de visión, fácil manipulación al ser Plug&Play, su tamaño y peso reducidos que permiten adaptarla de manera adecuada al prototipo, su versatilidad para emplearla en todo tipo de aplicaciones y un precio accesible. Además, la cámara seleccionada, cuenta con un cable más extenso que permite que la cámara se desplace sin problemas

a lo largo del prototipo. La calidad de las imágenes depende, a más de la resolución en píxeles, del procesamiento que se le dé a dicha información adquirida y sobre todo se considera la capacidad del sistema microcontrolado que se emplea debido a que, a mayor resolución en píxeles, mayor consumo de recursos al procesar las imágenes.

b) DISEÑO DEL SISTEMA ELECTROMECAÁNICO PARA EL DESPLAZAMIENTO DEL DISPOSITIVO DE ADQUISICIÓN DE IMÁGENES

El sistema electrónico para el desplazamiento de la cámara digital, que se encargará de adquirir las imágenes, es un sistema compuesto de engranajes y actuadores (motores de corriente continua), que deberán desplazar la cámara digital sobre la base móvil donde se ubicarán los objetos de estudio (huevos de gallina), con el fin de adquirir las imágenes a procesar dentro del sistema informático de acuerdo con las matrices generadas.

Para el desarrollo de este sistema se toma como punto de partida los sistemas de impresión 3D, puesto que basan su funcionamiento en el movimiento del extrusor para distribuir el filamento, en el caso del prototipo, el enfoque es el movimiento de la cámara. Debido a que se requiere únicamente un desplazamiento en 2 dimensiones, se excluye el eje de desplazamiento en coordenadas “z” del sistema cartesiano tridimensional que se muestra a continuación en la figura 4.10.

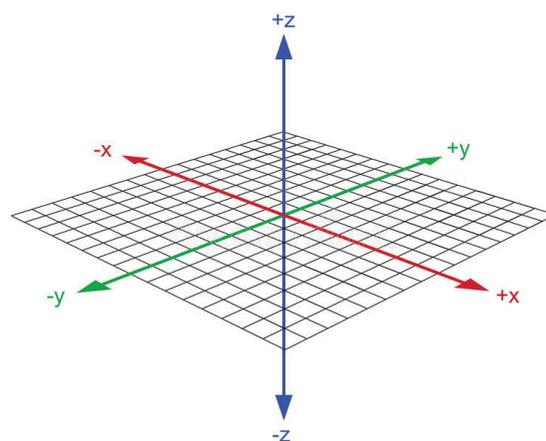


Fig. 4.10. Vectores de eje de coordenadas 3D. [66]

El desplazamiento en el eje z se puede calibrar manualmente y mantenerlo fijo sobre la base del prototipo, en concordancia con las pruebas pertinentes de funcionamiento

que determinen la distancia ideal de la cámara hacia las matrices generadas sobre la base móvil con los objetos de estudio.

Considerando las bases, se procede al desarrollo del sistema mecánico para el desplazamiento requerido tomando como referencia el sistema CoreXY.

El sistema CoreXY permite mover una plataforma en los ejes x e y manteniendo fijos los motores en la estructura con un diseño sencillo, adaptable y de rápido desplazamiento como se muestra en la figura 4.11. Este sistema se basa en poleas y correas que se cruzan fuera del área de trabajo, evitando problemas u obstáculos indeseados en el prototipo. [67]

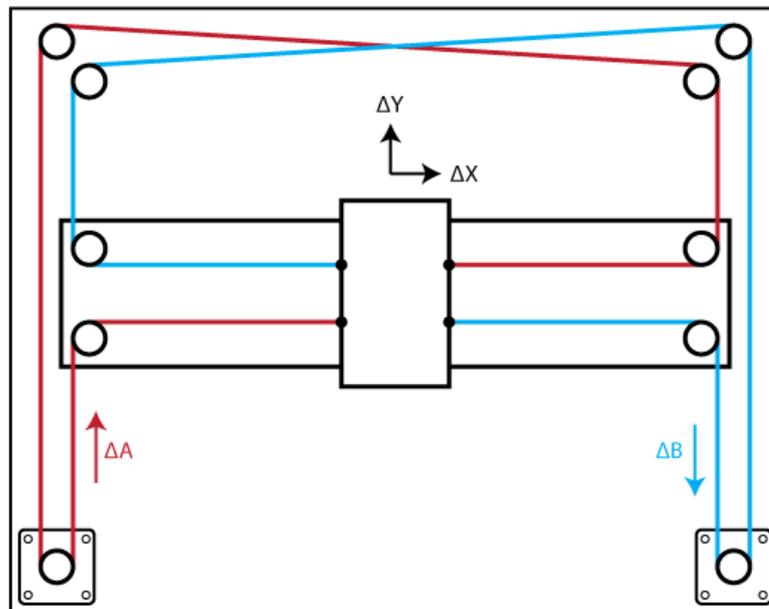


Fig. 4.11. Bosquejo del sistema de desplazamiento CoreXY. [67]

Las ecuaciones que se plantean a continuación definen los movimientos del sistema en los ejes coordenados.

Desplazamiento en el eje x, Ecuación 4.8.

$$\Delta X = \frac{1}{2}(\Delta A + \Delta B) \quad (4.8)$$

Desplazamiento en el eje y, Ecuación 4.9.

$$\Delta Y = \frac{1}{2}(\Delta A - \Delta B) \quad (4.9)$$

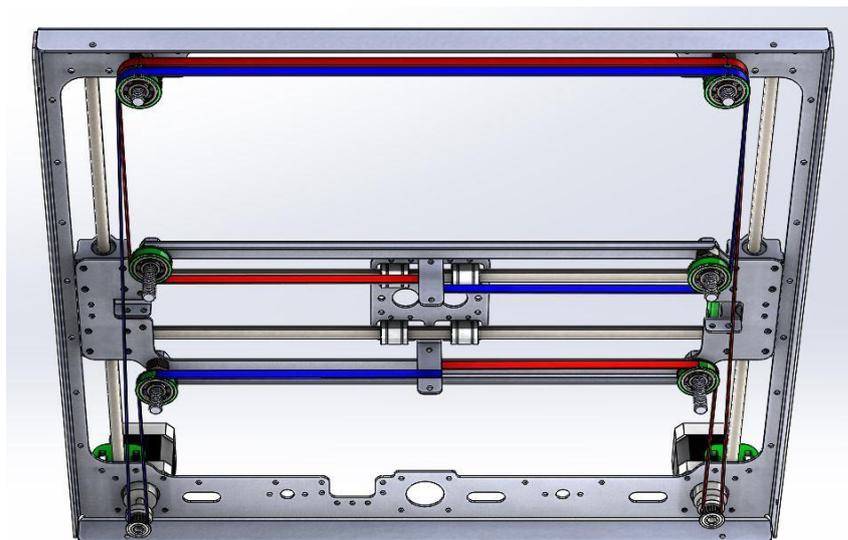


Fig. 4.13. Diseño mecánico del sistema de desplazamiento CoreXY (Vista inferior). [67]

Para el desarrollo del prototipo se ha modificado parcialmente el diseño presentado, posicionando los motores sobre la base o eje con menor sección. Los planos del prototipo se muestran en el Anexo 1.

c) SELECCIÓN DE ACTUADORES PARA EL DESPLAZAMIENTO DEL DISPOSITIVO DE ADQUISICIÓN DE IMÁGENES

El dispositivo de adquisición de imágenes deberá desplazarse. En la tabla 4.3 se detallan características generales de cámaras digitales que pueden vincularse con la microcomputadora seleccionada.

Tabla 4.4 Características técnicas de distintos tipos de motores. [68], [69], [70]

	Motor DC Pololu 4.4:1 Metal Gearmotor 25Dx48L mm HP	Servo Motor Futaba S3003	Motor Paso a Paso Nema 14
Ángulo de Rotación	360°	180°	360°
Tipo de control	Driver adicional y Software	Software	Driver adicional y Software
Precisión del control de rotación	Baja	Alta	Alta

Requerimientos de la placa de control por motor	1 GPIO PWM 2 GPIO Digitales	1 GPIO Digital	2 - 4 GPIO Digitales
Torque	1,44 Kg-cm	3,2 Kg-cm	0,65 Kg-cm
Corriente	6,5 A	-	0,28 A / fase
Voltaje	6 V	4,8 – 6 V	7,4 V
Peso	82 g	37,2 g	130 g
Precio Aproximado	\$25	\$13	\$20
Material	Metal	Plástico	Metal

En base a las características de los motores analizados, se selecciona los motores paso a paso debido a su mayor precisión para realizar la labor deseada, es decir, mover y posicionar la cámara, su fácil control, rotación continua, fuerza necesaria y un precio accesible. Además, no se requiere más aditamentos que el driver propio para mover este tipo de motores y los pines GPIO de la microcomputadora.

En la figura 4.14 se muestra un esquema de conexión de los motores paso a paso, cada uno de ellos emplearán 2 puertos GPIO de la Raspberry para poder funcionar y mover el sistema de desplazamiento. Se emplea el driver DRV8825 que se comercializa comúnmente como driver de control para este tipo de motores paso a paso.

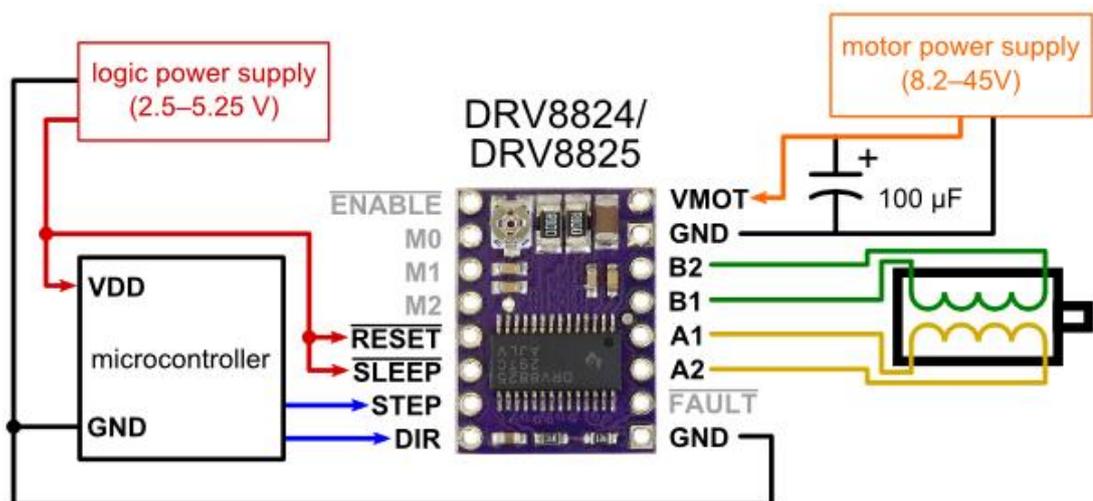


Fig. 4.14. Diagrama de conexión de motores paso a paso controlados desde un sistema microcontrolado a través del driver DRV8825. [71]

En este tipo de motores, el fabricante recomienda excitar dos bobinas cada vez para tener un máximo par motor y buena velocidad por lo que se emplea el modo de paso completo, aunque esto genera alto consumo de energía. [71]

d) DESARROLLO DEL SOFTWARE DE CONTROL DE MOVIMIENTO

El software de control del prototipo se basa en el lenguaje de programación Python que maneja Raspberry Pi bajo el sistema operativo Raspbian. Dentro de las características más potentes de esta microcomputadora se encuentra una disposición de Pines de propósito general, para aplicaciones electrónicas, que se los conoce como GPIO cuya numeración no sigue un formato específico y deben ser configurados de acuerdo a los requerimientos como entradas o salidas. Los Pines GPIO pueden generar pulsos digitales cuyos niveles lógicos son: Alto o 1 Lógico de 3,3 Voltios y Bajo o 0 Lógico de 0 Voltios, además, junto con esta distribución de pines se encuentran otros que generan energía en corriente continua para alimentar distintos elementos electrónicos, entregando 5 Voltios o 3,3 Voltios, entre otros pines para aplicaciones específicas como se indica en la figura 4.15. [72]

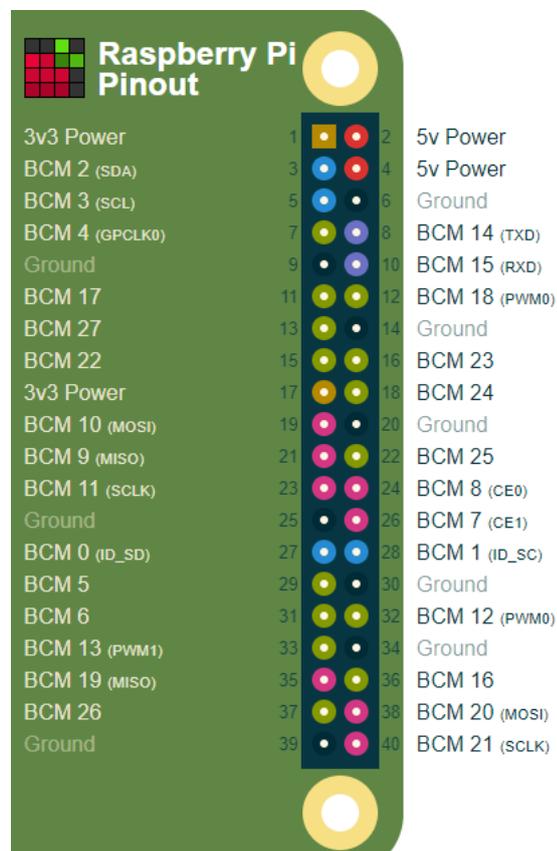


Fig. 4.15. Distribución física de pines Raspberry Pi 3 modelo B y sus funciones. [73]

En la figura 4.16 se indica la distribución lógica de los Pines GPIO de Raspberry que deben ser consideradas para las configuraciones por software.

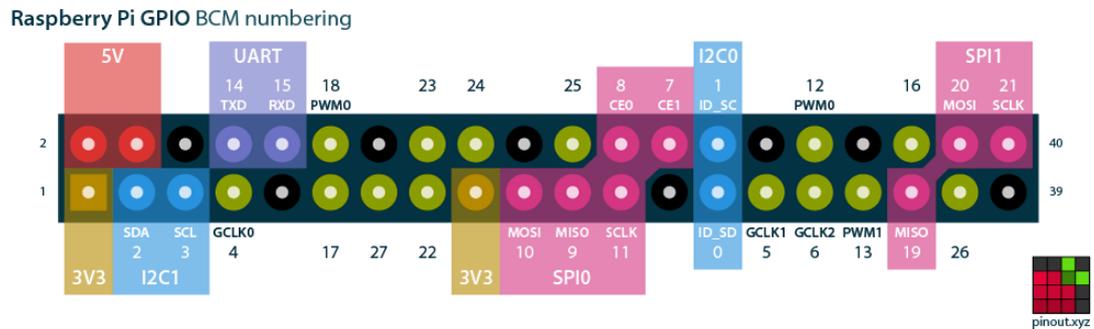


Fig. 4.16. Distribución lógica GPIO Raspberry Pi 3 modelo B y sus funciones. [73]

Para proceder con el control del desplazamiento del sistema de adquisición de imágenes se instala el paquete RPI.GPIO, para manejar los pines de Raspberry, con la siguiente instrucción en la línea de comandos del sistema:

```
sudo apt-get install python-dev python-rpi.gpio
```

Una vez instalado el paquete requerido se crea un fichero y se le asigna algún nombre, en este caso motorpasos.py bajo la siguiente instrucción:

```
nano motorpasos.py
```

Dentro del fichero que se crea, se alojan todas las instrucciones para el control GPIO y para ejecutar dichas funciones se llama a los métodos a través de la línea de comandos del sistema. Las instrucciones empleadas se detallan en el Anexo 2.

4.3.5. DESARROLLO DEL SOFTWARE DE PROCESAMIENTO DE IMÁGENES

Una de las herramientas de software más robustas para el procesamiento de imágenes y visión artificial es OpenCV, que como su nombre lo indica es de código abierto, acoplándose perfectamente con los requerimientos del prototipo a diferencia de Matlab que también es un software robusto, con características similares, pero bajo licencia de pago.

Para instalar OpenCV en su versión más reciente se ha tomado como referencia la guía tutorial de Adrian Rosebrock publicada el 4 de septiembre del 2017, donde se detalla paso a paso el procedimiento a seguir. [74]

Posterior a la instalación de OpenCV se procede a analizar las imágenes que se obtenga en el proceso de adquisición de datos de forma automática como se plantea a continuación.

a) CAPTURA DE IMÁGENES

La captura de imágenes es la adquisición de la información visual a través del dispositivo, es decir, la cámara digital. En base a la librería OpenCV instalada en la microcomputadora y el lenguaje de programación Python en su versión 3.5 se emplea el siguiente código para capturar, visualizar y almacenar imágenes.

Primero se asigna una variable encargada de habilitar el dispositivo:

```
camera = cv2.VideoCapture(0)
```

Posteriormente se lee en tiempo real la información visual que genera la cámara digital. Este código puede insertarse dentro de un ciclo para que la lectura sea continua de acuerdo con la capacidad y rapidez de la cámara:

```
retval, cam = camera.read()
```

Para guardar la imagen captada en un determinado momento, se emplea el siguiente comando donde se especifica la ruta de almacenamiento junto con el nombre de la imagen y el dispositivo que está habilitado:

```
cv2.imwrite("/home/pi/analisis.png",cam)
```

Si se desea visualizar una imagen almacenada, en primera instancia se lee dicha imagen a través de una variable, especificando la ruta de almacenamiento y posterior a ello se muestra la imagen:

```
original = cv2.imread("/home/pi/analisis.png")
```

```
cv2.imshow(original)
```

b) CONVERSIÓN DE IMÁGENES A ESCALA DE GRISES

Para la manipulación de imágenes es necesario convertirlas a escala de grises, lo que implica convertirlas en imágenes de tipo binario asignando a cada pixel un solo valor, simplificando y agilizando el procesamiento. A través del siguiente comando es posible realizar la conversión de imágenes a escala de grises a través de OpenCV:

```
gris = cv2.cvtColor(original, cv2.COLOR_BGR2GRAY)
```

Este comando recibe como parámetro la imagen almacenada o capturada en ese instante y la instrucción de conversión. Además, la imagen puede ser leída directamente en escala de grises de la siguiente manera, donde, como segundo parámetro se indica esta instrucción al colocar `cv2.IMREAD_GRAYSCALE` o simplemente el número 0:

```
gris = cv2.imread('/home/pi/analisis.png', cv2.IMREAD_GRAYSCALE)
```

```
gris = cv2.imread('/home/pi/analisis.png',0)
```

c) REDUCCIÓN DE RUIDO EN IMÁGENES MEDIANTE FILTRO GAUSSIANO

Existen varios filtros que pueden emplearse en una imagen con distintos objetivos, pero el filtro gaussiano es uno de los más aplicados y robustos ya que sigue leyes probabilísticas a través de desviaciones estándar que permiten descartar los factores externos que influyen en una imagen. En el comando que se emplea para aplicar este filtro se debe insertar como parámetros, la imagen original o a escala de grises, el ancho y altura del kernel que es un entorno dentro del sistema operativo y evita que se saturate realizando aquí todas las operaciones y, además, como último parámetro se debe especificar la desviación estándar de la siguiente manera:

```
gauss = cv2.GaussianBlur(gris,(5,5),0)
```

d) DETECCIÓN DE CONTORNOS

Existen varios métodos que permiten detectar los contornos o bordes que rodean a los objetos dentro de una imagen como en el caso del presente proyecto, la detección de los huevos de gallina. Para la aplicación de estos métodos es necesario que la imagen sea de tipo binario, permitiendo de esta manera analizar posteriormente las características o comparar los parámetros en base a los valores umbrales que se establezcan sobre la imagen procesada.

A continuación, se presenta una de las funciones empleadas para detectar bordes en la imagen, en la cual se debe considerar como parámetros la imagen procesada con la aplicación del filtro respectivo, y los valores mínimo y máximo de las características

de los contornos, mientras más bajos sean estos valores se detectará, a más de los bordes otras características de los huevos de gallina como poros, fisuras, entre otras. En este caso se debe probar con distintos valores entre 0 y 255 hasta establecer el valor más óptimo donde se descarten perturbaciones como brillo de la imagen que modifica el valor de los pixeles pero que se mantengan los detalles intrínsecos de los huevos de gallina, este método crea una nueva imagen que puede ser almacenada en una variable temporal como la siguiente:

```
canny = cv2.Canny(gauss, 0, 24)
```

Para delimitar únicamente los bordes externos que rodean al huevo de gallina o cualquier otro objeto se emplea la siguiente función donde los parámetros son la imagen temporal de características como una copia, la especificación de bordes únicamente externos, y las coordenadas de los pixeles en ese orden:

```
(_, contornos,_) = cv2.findContours(canny.copy(), cv2.RETR_EXTERNAL,  
cv2.CHAIN_APPROX_SIMPLE)
```

Los resultados del análisis de contornos se almacenan en la variable contornos y pueden ser dibujados sobre la imagen original de la siguiente manera:

```
cont=cv2.drawContours(original,contornos,-1,(0,255,0), 2)
```

En el último caso se requiere como parámetros la imagen base sobre la cual se va a dibujar los contornos, los contornos obtenidos, el índice de contorno como en este caso -1 para dibujar todos y al final color RGB y grosor de los dibujos.

e) SEGMENTACIÓN DE LAS IMÁGENES ADQUIRIDAS

La segmentación de una imagen consiste en dividirla en partes de interés. En el caso del prototipo cada imagen abarca 9 huevos, estos deben ser analizados de manera independiente, por lo que se realiza cortes sucesivos a la imagen para separarlos, estableciendo sus coordenadas en ancho como en alto de la siguiente manera:

```
canny = canny[0:170, 15:225]
```

Cada objeto debe tener coordenadas diferentes y puede almacenarse en alguna variable o analizarse directamente, el primer parámetro son las coordenadas en el eje y el segundo las coordenadas en el eje x.

f) COMPARACIÓN DE DESCRIPTORES DE FORMA

Para realizar la comparación de características o descriptores de los huevos de gallina se emplea una imagen base procesada bajo los mismos parámetros. Cada corte que se realiza como parte de la segmentación de la imagen, es decir cada huevo de gallina, es comparado con la imagen principal que a través del siguiente método devuelve un valor de similitud el cual puede ser almacenado en una matriz en concordancia con la posición del huevo sobre la cubeta:

```
matrizauto[0][0] = cv2.matchShapes(cannybase,canny[0:170, 15:225],1,0.0)
```

El primer parámetro es la imagen base, el segundo la imagen seccionada que es objeto de análisis, el tercero una de las 3 formas que emplea el método para la comparación y el cuarto un valor flotante específico del método por lo que se inicializa como 0.0.

Cuanto menor sea el valor, mayor será la similitud, es decir que el huevo estará en mejores condiciones para ser incubado.

4.3.6. BASE DE DATOS

La base de datos es el repositorio de la información adquirida por el prototipo de control de calidad, en el cual se detalla la cantidad de huevos analizados tanto idóneos como no idóneos para el proceso de incubación, hora y fecha de cada análisis y el tipo de análisis efectuado. La información mencionada se encuentra disponible para análisis estadísticos de acuerdo a los requerimientos del usuario.

a) DESARROLLO DE LA BASE DE DATOS

Para el desarrollo de la base de datos se emplea PHP para la administración de la información y MySQL para la gestión de la base de datos.

Al contar con el sistema operativo Raspbian que es una distribución de GNU/Linux, manteniendo el principio del proyecto de emplear Software Libre, se procede con la instalación de los sistemas de gestión y almacenamiento, que son los siguiente:

SERVIDOR APACHE

El servidor Apache es un servidor web que se emplea como ruta de acceso a la base de datos. El proceso de instalación de este servidor se presenta a continuación, actualizando inicialmente Raspberry y el sistema operativo:

```
sudo rpi-update
```

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install apache2
```

Posteriormente se verifica la correcta instalación del servidor, ingresando desde un navegador dentro de la misma microcomputadora y colocando localhost o 127.0.0.1 en la barra de direcciones. Es posible verificar también la instalación del servidor desde otro dispositivo, ingresando la dirección IP de la microcomputadora en un navegador. En cualquier de los casos mencionados se visualizará la información como en la figura 4.17.

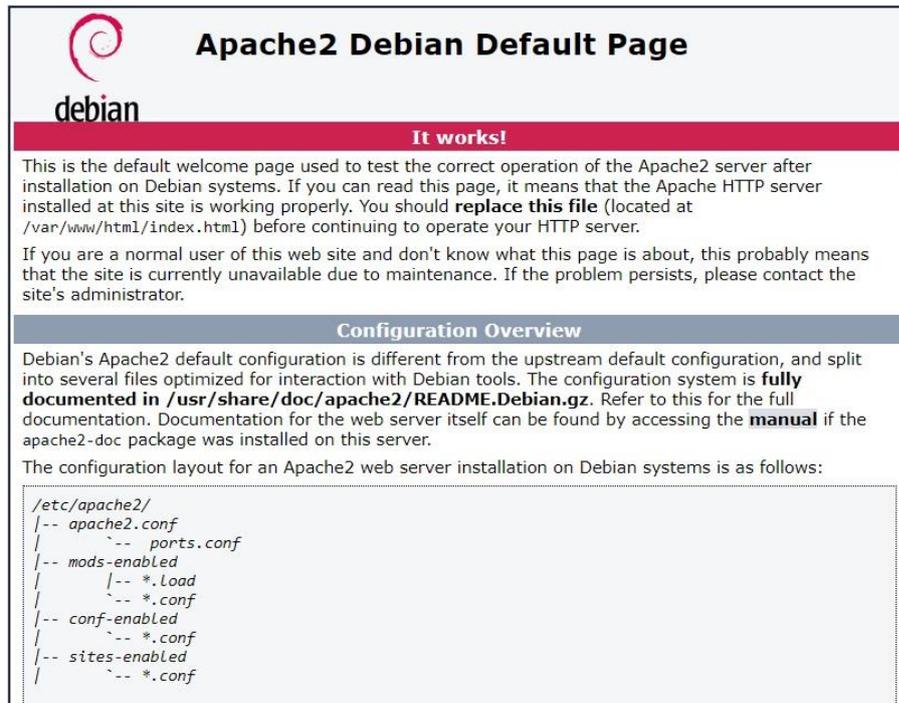


Fig. 4.17. Página de verificación del Servidor Apache en Raspberry. [Investigador]

MANEJO DE BASES DE DATOS MYSQL / MARIADB

Para manejar la base de datos se emplea MySQL que es un sistema gestor de bases de datos, del cual deriva MariaDB con licencia de código abierto y se instala de la siguiente manera:

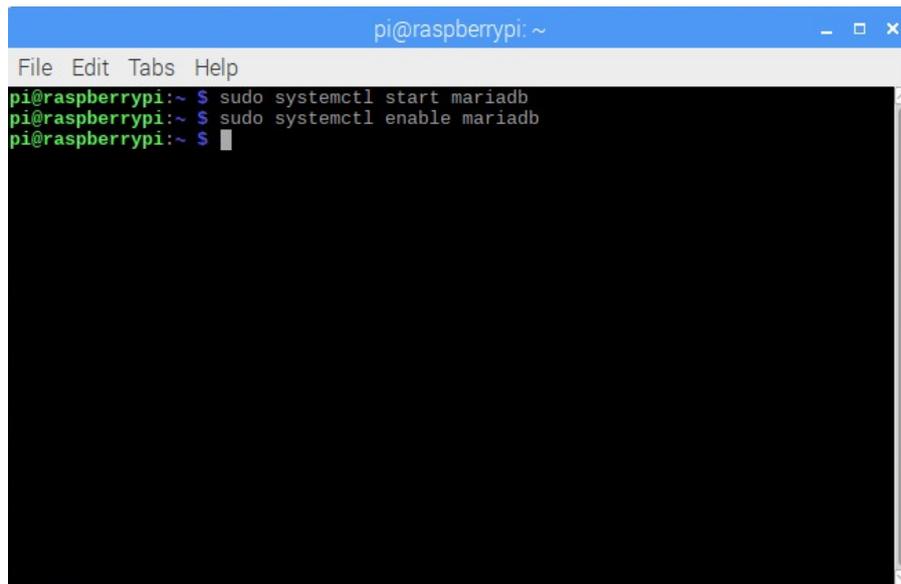
```
sudo apt-get install php5-mysql mysql-server mysql-client
```

Para iniciar el servicio se procede mediante los siguientes comandos:

```
sudo systemctl start mariadb
```

```
sudo systemctl enable mariadb
```

Si el procedimiento se realiza adecuadamente, no se mostrará ningún mensaje en la consola como se muestra en la figura 4.18.

A screenshot of a terminal window titled 'pi@raspberrypi: ~'. The window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The terminal output shows three lines of commands and their execution: 'pi@raspberrypi:~ \$ sudo systemctl start mariadb', 'pi@raspberrypi:~ \$ sudo systemctl enable mariadb', and 'pi@raspberrypi:~ \$' followed by a cursor. The rest of the terminal is black, indicating no output was shown for these commands.

```
pi@raspberrypi:~ $ sudo systemctl start mariadb
pi@raspberrypi:~ $ sudo systemctl enable mariadb
pi@raspberrypi:~ $
```

Fig. 4.18. Activación de MariaDB por consola Raspberry. [Investigador]

PHP Y PHPMYAdmin

PHP es un lenguaje de programación de código abierto orientado al desarrollo web.

PHPMyAdmin es una herramienta de código abierto escrita en PHP que concentra la administración y gestión de bases de datos MySQL mediante una interfaz gráfica.

Para instalar PHP se emplean los siguientes comandos:

```
sudo apt-get install php5 libapache2-mod-php5
```

A continuación, se reinicia el servidor web:

```
sudo systemctl restart apache2
```

Para verificar el correcto funcionamiento de PHP se debe generar un archivo con el siguiente comando:

```
sudo nano phpinfo.php
```

El archivo generado debe alojarse bajo la ruta /var/www/html y debe ser editado con los siguientes datos:

```
<?php
phpinfo();
?>
```

Posteriormente se verifica la correcta instalación de PHP, ingresando desde un navegador dentro de la misma microcomputadora y colocando localhost/phpinfo.php o 127.0.0.1/phpinfo.php en la barra de direcciones. Es posible verificar también la instalación del lenguaje desde otro dispositivo, ingresando la dirección IP/phpinfo.php de la microcomputadora en un navegador. En cualquier de los casos mencionados se visualizará la información como en la figura 4.19.

PHP Version 7.0.27-0+deb9u1 	
System	Linux raspberrypi 4.14.43-v7+ #1115 SMP Fri May 25 14:27:48 BST 2018 armv7l
Build Date	Jan 5 2018 13:51:52
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-mysqld.ini, /etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/15-xml.ini, /etc/php/7.0/apache2/conf.d/20-bz2.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-curl.ini, /etc/php/7.0/apache2/conf.d/20-dom.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gd.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-mbstring.ini, /etc/php/7.0/apache2/conf.d/20-mysqli.ini, /etc/php/7.0/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-simplexml.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini, /etc/php/7.0/apache2/conf.d/20-wddx.ini, /etc/php/7.0/apache2/conf.d/20-xmlreader.ini, /etc/php/7.0/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.0/apache2/conf.d/20-xsl.ini, /etc/php/7.0/apache2/conf.d/20-zip.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012.NTS
PHP Extension Build	API20151012.NTS
Debug Build	no
Thread Safety	disabled

Fig. 4.19. Página de verificación del lenguaje PHP en Raspberry. [Investigador]

Para la instalación de PHPMyAdmin se procede de la siguiente manera:

```
sudo apt-get install phpmyadmin
```

Al finalizar la instalación es necesario vincular phpMyAdmin con el servidor web Apache, agregando la siguiente instrucción al final del archivo:

```
sudo nano /etc/apache2/apache2.conf
Include /etc/phpmyadmin/apache.conf
```

Se procede a reiniciar el servidor:

```
sudo systemctl restart apache2
```

Finalmente se verifica el correcto funcionamiento de la interfaz PHPMyAdmin, ingresando desde un navegador dentro de la misma microcomputadora y colocando localhost/phpmyadmin o 127.0.0.1/phpmyadmin en la barra de direcciones. Es posible verificar también la instalación desde otro dispositivo, ingresando la dirección IP/phpmyadmin de la microcomputadora en un navegador. En cualquier de los casos mencionados se visualizará la información como en la figura 4.20.

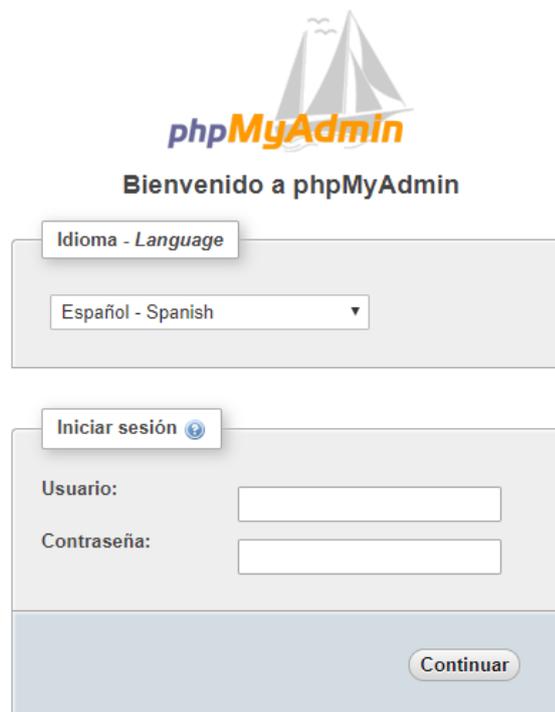


Fig. 4.20. Interfaz de administración y gestión de bases de datos phpMyadmin en Raspberry.

[Investigador]

El usuario y contraseña para la manipulación de las bases de datos se crea en MariaDB bajo el siguiente comando:

```
sudo mysql -u root -p
```

b) DESARROLLO DEL SOFTWARE PARA VINCULAR LA BASE DE DATOS

Posterior a la creación de la base de datos se requiere vincularla con el software desarrollado para almacenar los resultados de cada análisis y poder acceder fácilmente

a ellos a través de la web. Para escribir los resultados en la base de datos a través de Python, se crea el siguiente método:

```
def baseauto(r,s):  
  
    conn = pymysql.connect(  
  
        host="localhost", user="root", passwd="ROOT_AFS26", db="control")  
  
    mycursor = conn.cursor()  
  
    sql = "INSERT INTO metodos (Aptos, No_Aptos, Tipo) VALUES (%s, %s, %s)"  
  
    val = (str(r), str(s), "Auto")  
  
    mycursor.execute(sql, val)  
  
    conn.commit()  
  
    conn.close()
```

En el método se especifican los datos de acceso como dirección, usuario, contraseña y nombre de la base de datos, además, se toma como parámetros la cantidad de huevos aptos y no aptos analizados por el prototipo y se adicionan en la tabla dentro de la base de datos posicionándolos a través de un cursor en el lugar correspondiente. El identificador o número de análisis, la fecha y el tipo de método empleado se adicionan automáticamente en la base de datos.

4.3.7. INTERFAZ HOMBRE - MÁQUINA

La Interfaz Hombre – Máquina es el software que se encarga de la comunicación entre el usuario y el prototipo de control de calidad de huevos de gallina para incubación, brindando la información que se requiere visualizar, tanto en la adquisición y el procesamiento de las imágenes como la información que se almacena en la base de datos luego del proceso.

a) DESARROLLO DE LA INTERFAZ HOMBRE - MÁQUINA

Para el desarrollo de la Interfaz se empleó la herramienta Tkinter que es una librería o módulo de interfaz gráfica bajo el lenguaje de programación Python cuyo entorno viene precargado en Raspberry, además de adicionar cada parte de código requerido para el funcionamiento del prototipo.

Se creó una ventana principal con los datos del prototipo, en la cual se habilitan 2 botones, para seleccionar el método a emplear de los 2 métodos con los que trabaja el software desarrollado.

Al seleccionar el método manual se cambia la ventana principal por la ventana del método donde se visualiza una matriz de elementos de 6x6 para seleccionar, a criterio del usuario, los huevos de gallina aptos para incubación en base a las posiciones en la cubeta, además, en un espacio de la ventana se visualiza en tiempo real la transmisión de video de la cámara que muestra matrices de huevos de 3x3 a la vez. El método manual controla el desplazamiento de la cámara en las distintas posiciones sobre los cuadrantes establecidos y la selección de los huevos de gallina, a través de botones con varias opciones de acuerdo a la posición en la que se encuentre la cámara.

Al seleccionar el método automático, el programa se encarga de adquirir directamente las imágenes y procesarlas en concordancia con los parámetros detallados anteriormente, desplazando la cámara en forma secuencial sobre los distintos cuadrantes, culminando con la visualización de resultados.

La información o resultados adquiridos por ambos métodos se almacenan directamente en la base de datos. En la figura 4.21 se detalla el proceso de la interfaz mediante un diagrama de flujo.

El código completo del programa para el funcionamiento del prototipo de control de calidad de huevos de gallina con su respectiva interfaz se muestra en el Anexo 3.

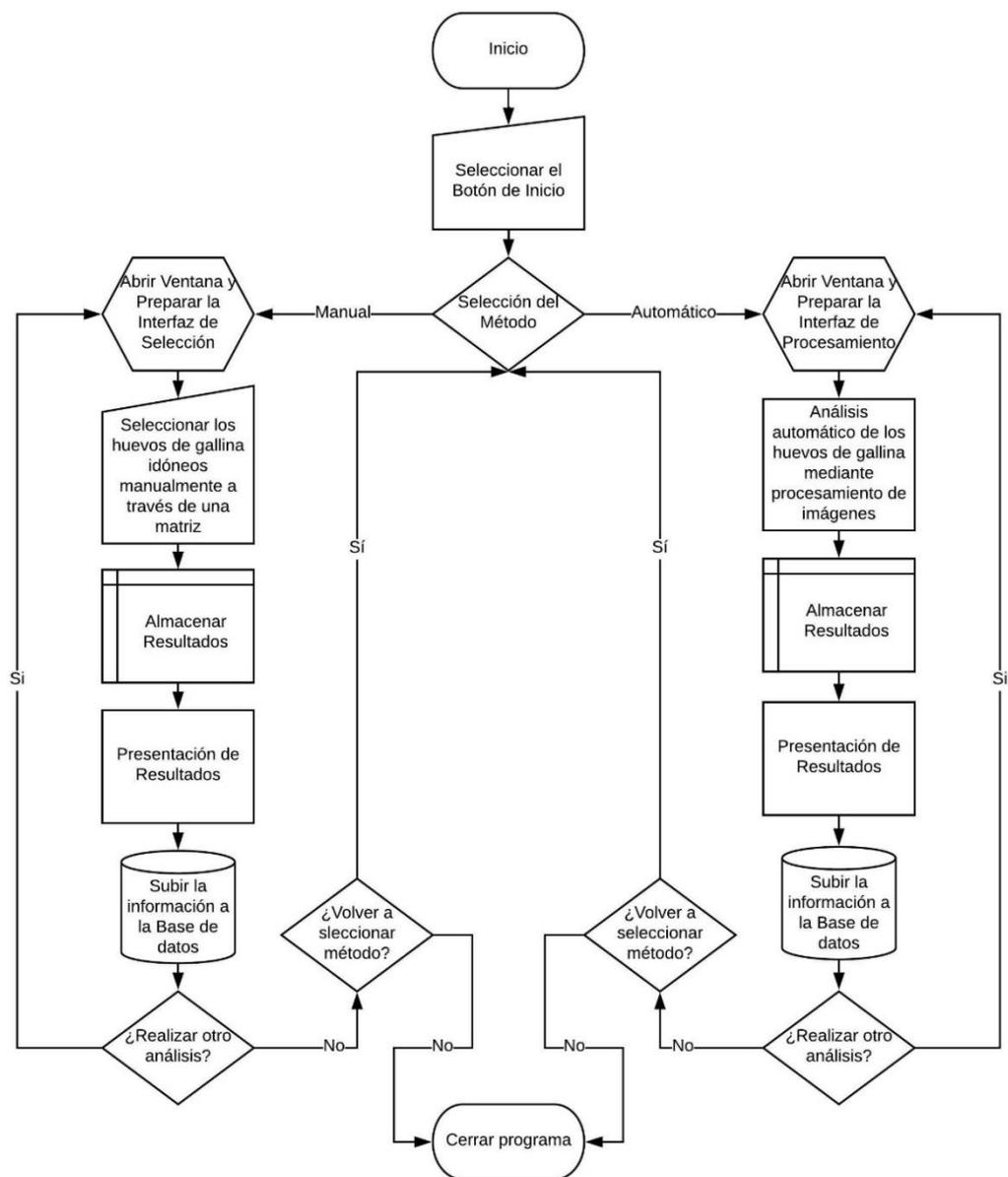


Fig. 4.21. Proceso de la interfaz gráfica del sistema de control de calidad. [Investigador]

4.4 IMPLEMENTACIÓN Y VALIDACIÓN DEL PROTOTIPO

El sistema de control de calidad de huevos de gallina mediante procesamiento de imágenes ha sido probado con huevos de gallina distribuidos para el consumo humano en tiendas, locales comerciales y el campo, determinando su aptitud para incubación. En la figura 4.22 se muestra el proceso de implementación y validación del prototipo.

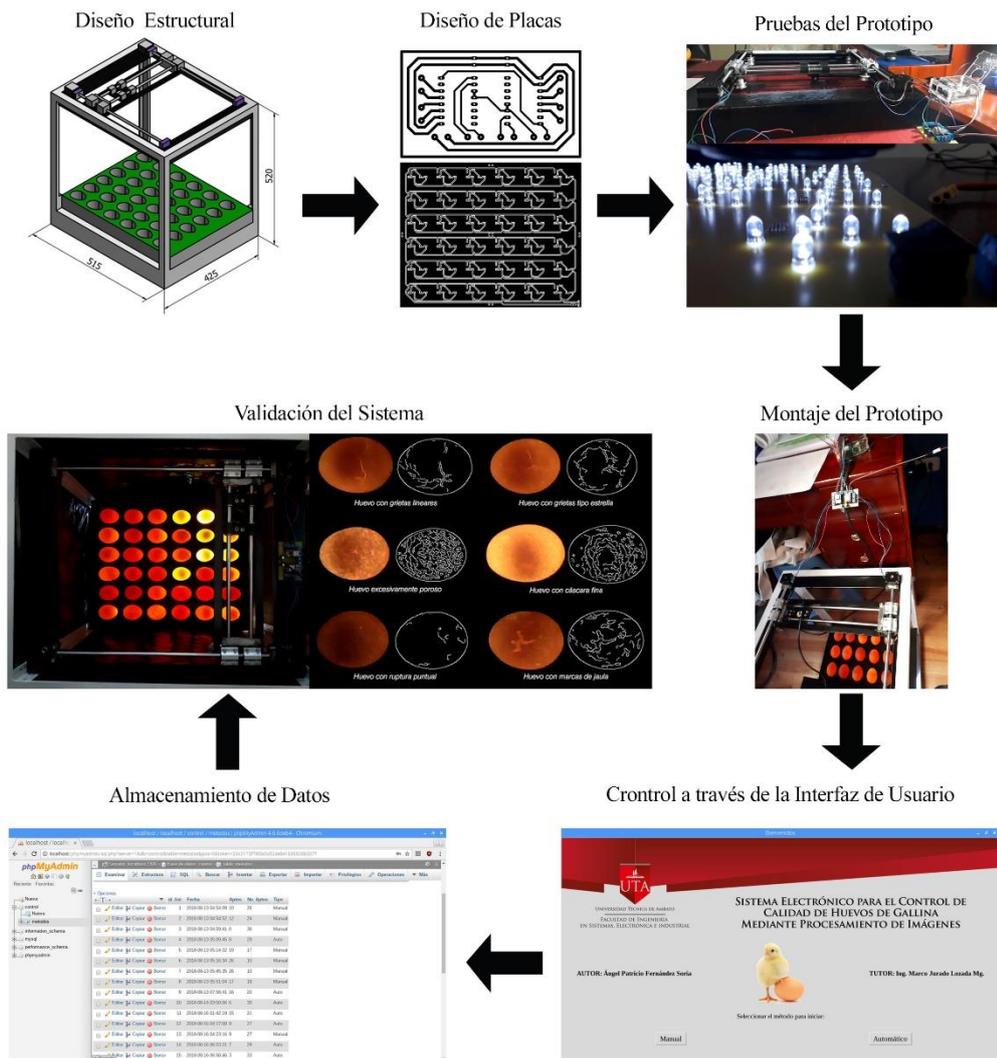


Fig. 4.22. Implementación y validación práctica del sistema de control de calidad de huevos de gallina. [Investigador]

4.4.1. POSICIONAMIENTO DE LA CÁMARA PARA ADQUISICIÓN DE IMÁGENES

Como primer factor se ha analizado el posicionamiento de la cámara, sobre las 2 dimensiones establecidas, eje x con la sección más extensa y eje y con la sección más corta. En las Tablas 4.5 y 4.6 se detallan las pruebas realizadas que permitieron a la cámara ubicarse en las posiciones adecuadas para adquirir las imágenes. La tabla 4.5 muestra el sentido de giro de los motores y el desplazamiento resultante sobre los ejes cartesianos.

Tabla 4.5 Sentido de giro y desplazamiento del sistema electromecánico encargado de desplazar la cámara sobre los huevos de gallina a analizar. [Investigador]

Configuración de giro de los motores para la orientación del desplazamiento de la cámara			
Nº Prueba	Sentido de giro Motor A	Sentido de giro Motor B	Dirección de desplazamiento
1	Horario	Horario	Eje Y Positivo
2	Horario	Antihorario	Eje X Positivo
3	Antihorario	Horario	Eje X Negativo
4	Antihorario	Antihorario	Eje Y Negativo

El sentido de giro y desplazamiento del sistema electromecánico permite posicionar a la cámara sobre los cuadrantes creados en la cubeta de huevos o base móvil.

Para posicionar el sistema se determinó el alcance de la cámara, realizando pruebas con 2 de las cámaras descritas en la tabla 4.3 verificando efectivamente que la Cámara Genius FaceCam 1000x fue la mejor opción debido a que la Cámara Ins IS-W811 al tener un acercamiento o zoom extendido puede visualizar menor cantidad de huevos de gallina de acuerdo al detalle de la tabla 4.6.

Tabla 4.6 Características de detección de cámaras en función de la distancia. [Investigador]

Resultados de la capacidad de detección de huevos de gallina a través de la cámara			
Nº Prueba	Cámara	Distancia Eje Z	Cantidad de huevos visualizados
1	Ins IS-W811	37,5 cm	4
2	Genius FaceCam 1000x	37,5 cm	> 12
3	Genius FaceCam 1000x	35,5 cm	12
4	Genius FaceCam 1000x	33,5 cm	> 9
5	Genius FaceCam 1000x	31,5 cm	9
6	Genius FaceCam 1000x	29,5 cm	> 6

Como se ha descrito anteriormente, la regulación de la altura en el eje z se realiza de forma manual, sujetando el sistema electromecánico a la base fija. En base a las pruebas realizadas con la cámara seleccionada se determinó que la mejor altura desde este dispositivo hacia la cubeta o base móvil es de 31,5 cm abarcando 9 huevos para la adquisición y procesamiento de imágenes lo que implica que para el análisis de los 36 huevos se divide la cubeta en 4 matrices simétricas o cuadrantes. No se consideró la ubicación en la que se abarca 12 huevos debido a que, por la configuración de la cubeta, no se obtendrían matrices simétricas y se repetirían los análisis de varios

huevos que no sería óptimo, además, a mayor distancia se pierde el detalle de la imagen en cada uno de los huevos que sean analizados.

4.4.2. TIEMPOS DE RETARDO DEL MECANISMO DE DESPLAZAMIENTO

La tabla 4.7 muestra las pruebas realizadas sobre el eje x para el desplazamiento y posicionamiento de la cámara para adquisición de imágenes en los respectivos cuadrantes. Los motores seleccionados requieren, al emplear el modo de paso completo, pulsos sincronizados 1 pulso en alto y 1 pulso en bajo por cada paso.

Tabla 4.7 Parámetros para el movimiento del sistema electromecánico encargado de desplazar la cámara sobre los huevos de gallina a analizar - Eje X. [Investigador]

EJE X						
N.º Prueba	Ajuste Banda	Micro pausa entre pulsos	Número de pasos de los motores	Tiempo Calculado	Tiempo Medido	Observaciones
1	Alto	0,005 s	900	9 s	9,37 s	Atascos en el sistema y posicionamiento incorrecto de cámara
2	Alto	0,006 s	1000	12 s	12,41 s	Atascos en el sistema y posicionamiento incorrecto de cámara
3	Bajo	0,006 s	1000	12 s	12,37 s	Sobreposición de banda y posicionamiento incorrecto de cámara
4	Medio	0,006 s	1000	12 s	12,32 s	Desplazamiento normal, baja velocidad y posicionamiento incorrecto de cámara
5	Medio	0,006 s	1100	13,2 s	13,54 s	Desplazamiento normal, baja velocidad y posicionamiento incorrecto de cámara

6	Medio	0,006 s	1130	13,56 s	13,91 s	Desplazamiento normal, baja velocidad y posicionamiento incorrecto de cámara
7	Medio	0,006 s	1150	13,8 s	14,15 s	Desplazamiento normal, baja velocidad y posicionamiento aproximado de cámara
8	Medio	0,006 s	1170	14,04 s	14,39 s	Desplazamiento normal, baja velocidad y posicionamiento incorrecto de cámara
9	Medio	0,006 s	1160	13,92 s	14,27 s	Desplazamiento normal, baja velocidad y posicionamiento adecuado de cámara
10	Medio	0,005 s	1160	11,6 s	11,89 s	Desplazamiento normal, velocidad normal y posicionamiento adecuado de cámara
11	Medio	0,004 s	1160	9,28 s	9,51 s	Desplazamiento normal, velocidad aceptable y posicionamiento adecuado de cámara
12	Medio	0,003 s	1160	6,96 s	7,13 s	Desplazamiento con cambios bruscos en los motores, velocidad rápida y posicionamiento incorrecto de cámara

La tabla 4.8 muestra las pruebas realizadas sobre el eje y para el desplazamiento y posicionamiento de la cámara para adquisición de imágenes. Se consideró los

resultados obtenidos sobre el eje x y se modificó únicamente el número de pasos que dan los motores para posicionar la cámara en los respectivos cuadrantes.

Tabla 4.8 Parámetros para el movimiento del sistema electromecánico encargado de desplazar la cámara sobre los huevos de gallina a analizar - Eje Y. [Investigador]

EJE Y						
N.º Prueba	Ajuste Banda	Micro pausa entre pulsos	Número de pasos de los motores	Tiempo Calculado	Tiempo Medido	Observaciones
1	Medio	0,004 s	800	6,4 s	6,56 s	Desplazamiento normal, velocidad aceptable y posicionamiento incorrecto de cámara
2	Medio	0,004 s	850	6,8 s	6,97 s	Desplazamiento normal, velocidad aceptable y posicionamiento incorrecto de cámara
3	Medio	0,004 s	900	7,2 s	7,38 s	Desplazamiento normal, velocidad aceptable y posicionamiento incorrecto de cámara
4	Medio	0,004 s	950	7,6 s	7,79 s	Desplazamiento normal, velocidad aceptable y posicionamiento incorrecto de cámara
5	Medio	0,004 s	930	7,44 s	7,62 s	Desplazamiento normal, velocidad aceptable y posicionamiento incorrecto de cámara
6	Medio	0,004 s	926	7,408 s	7,594 s	Desplazamiento normal, velocidad aceptable y posicionamiento incorrecto de cámara

7	Medio	0,004 s	922	7,376 s	7,562 s	Desplazamiento normal, velocidad aceptable y posicionamiento aceptable de cámara
8	Medio	0,004 s	918	7,344 s	7,53 s	Desplazamiento normal, velocidad aceptable y posicionamiento adecuado de cámara

En base a los resultados obtenidos se determinó que el desplazamiento en el eje x tarda 7,53 segundos y el desplazamiento en el eje y tarda 9,51 segundos en el prototipo desarrollado, en un solo sentido y con un ajuste medio de la banda dentada empleada en el prototipo. Para que la cámara recorra los 4 cuadrantes y regrese al origen se requiere 4 desplazamientos, 2 desplazamientos sobre el eje x y 2 desplazamientos sobre el eje y, es decir, el tiempo total que emplea el sistema de desplazamiento es de 34,08 segundos.

4.4.3. INTERFAZ DE USUARIO

La interfaz de usuario del sistema de control de calidad de huevos de gallina fue desarrollada considerando 2 métodos de uso principales. Al abrir el programa se muestra una pantalla principal como se indica en la figura 4.23 donde se debe seleccionar el método a emplear a través de 2 botones.



Fig. 4.23. Interfaz gráfica del sistema de control de calidad – Pantalla Principal. [Investigador]

El primer método es manual, donde se ha creado una matriz con botones de selección y el usuario controla el desplazamiento de la cámara sobre los cuadrantes para determinar a criterio personal los huevos que están aptos para incubar y los que no. En la figura 4.24 se puede visualizar la ventana que se ejecuta al seleccionar el método manual.

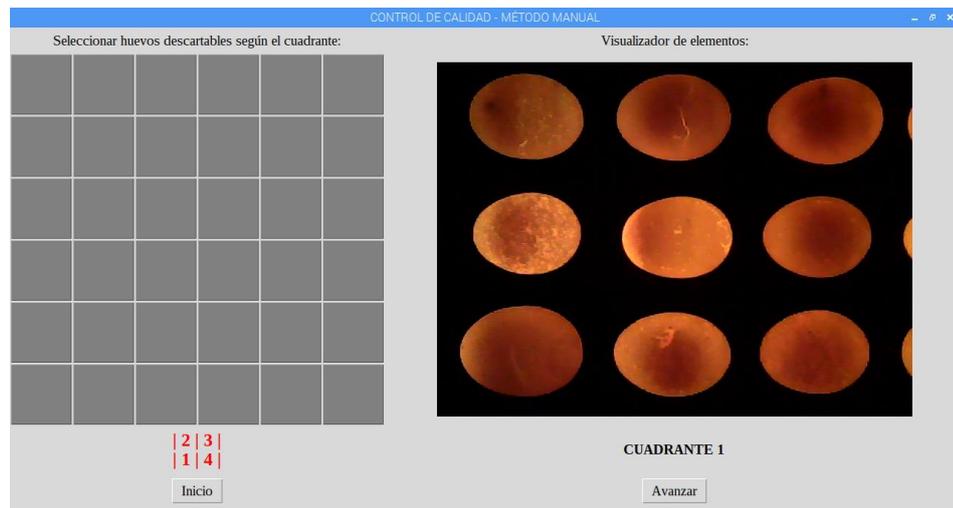


Fig. 4.24. Interfaz gráfica del sistema de control de calidad – Método Manual. [Investigador]

En la figura 4.25 se visualiza los resultados, de acuerdo al método manual, de una de las pruebas realizadas con el sistema, estos resultados se almacenan directamente en la base de datos al finalizar. Posterior a la selección de los huevos de gallina se presentan las opciones para volver al inicio, realizar otro análisis bajo el mismo método o salir completamente del programa.

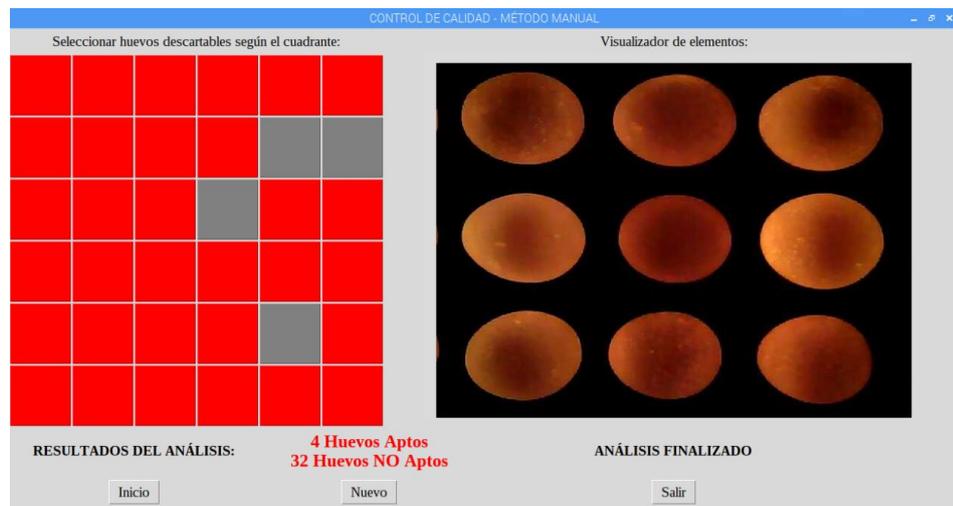


Fig. 4.25. Interfaz gráfica del sistema de control de calidad – Resultados Método Manual. [Investigador]

El segundo método es automático, donde se inicia una primera ventana que espera que el usuario dé la orden de ejecución para empezar o retorne a la ventana principal como se muestra en la figura 4.26.



Fig. 4.26. Interfaz gráfica del sistema de control de calidad – Método automático. [Investigador]

Al dar la orden de ejecución, se abrirá una segunda ventana en blanco sin opciones que esperará que culmine el análisis de los 36 huevos de gallina que abarca la cubeta. Al finalizar el análisis se presentan los resultados como en la figura 4.27 y al igual que en el método manual se almacenan directamente en la base de datos. Los huevos que sean seleccionados como No Aptos se marcan con una “X”.

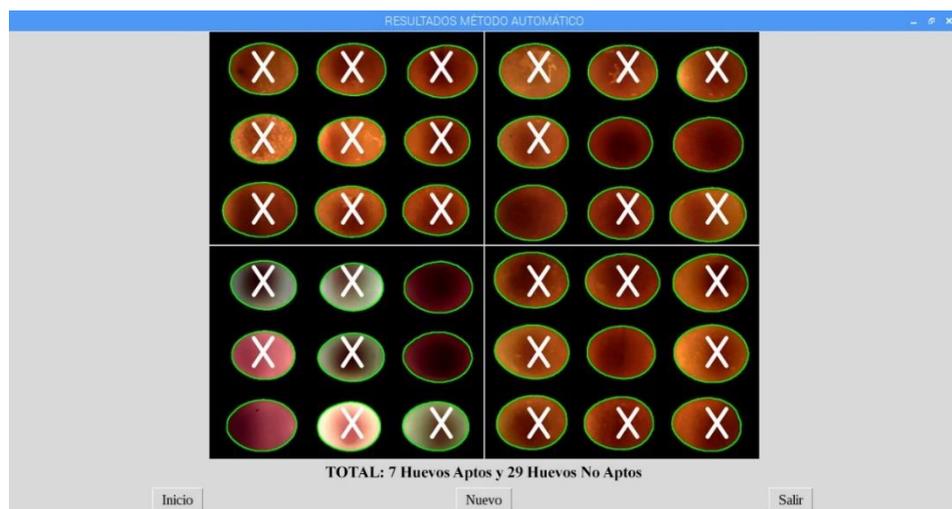


Fig. 4.27. Interfaz gráfica del sistema de control de calidad – Resultados método automático.

[Investigador]

En el método automático, para emplear la función de descriptores de forma, se toma como referencia una imagen base de un huevo en buen estado. La imagen base se compara con cada uno de los segmentos de las imágenes adquiridas, es decir, con cada uno de los huevos empleados en el análisis, en base a sus contornos. En la figura 4.28 se muestra la imagen base empleada para las comparaciones, que se aproxima a una elipse de ecuación: $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ con relación 4:3 de a con respecto a b.

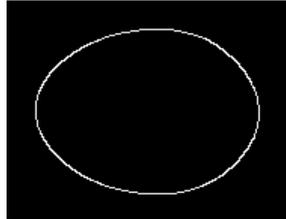


Fig. 4.28. Contorno base de un huevo de gallina para comparación de parámetros. [Investigador]

Al comparar los contornos se obtiene una métrica de similitud, mientras este resultado sea menor hay mayor coincidencia, cuanto mayor es la coincidencia se indica que el huevo analizado es apto para incubar. En la figura 4.29 se visualiza una de las imágenes adquiridas que, a través del procesamiento, genera los contornos para comparación de los huevos analizados.

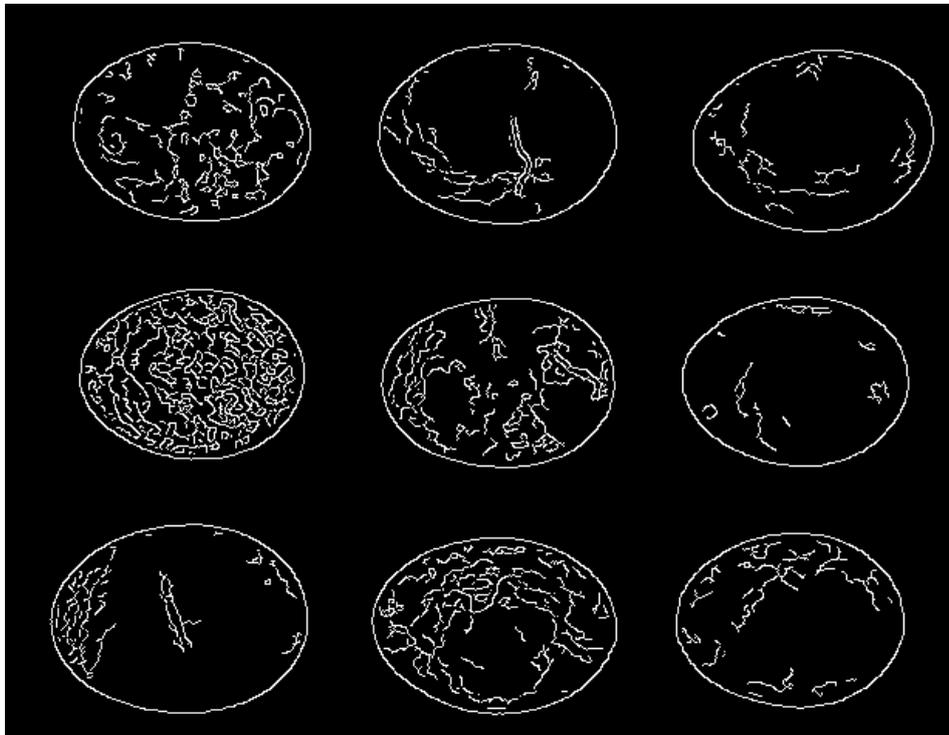


Fig. 4.29. Características y contornos de huevos analizados por el prototipo. [Investigador]

En base a los contornos se puede identificar fisuras, exceso de poros, entre otros parámetros indicados anteriormente. En la tabla 4.9 se presenta los resultados obtenidos frente a las distintas características de los huevos analizados.

Tabla 4.9 Resultados de detección de huevos según características. [Investigador]

Resultados de la detección de huevos de gallina de acuerdo a las características que dificultan su incubación		
Detalle	Detección	Funcionamiento
Huevo con grietas lineares	Detectado	Correcto
Huevo con grietas tipo estrella	Detectado	Correcto
Huevo excesivamente poroso	Detectado	Correcto
Huevo con lombrices	No Existente	-
Huevo con cáscara rugosa, fina o moteada	Detectado	Correcto
Huevo corregido en útero	No Existente	-
Huevo con ruptura puntual	Detectado	Correcto
Huevo con marca de jaula	Detectado	Correcto

A continuación, en la figura 4.30, se presenta una muestra de los huevos analizados y sus características problemáticas detectadas por el prototipo de control de calidad.

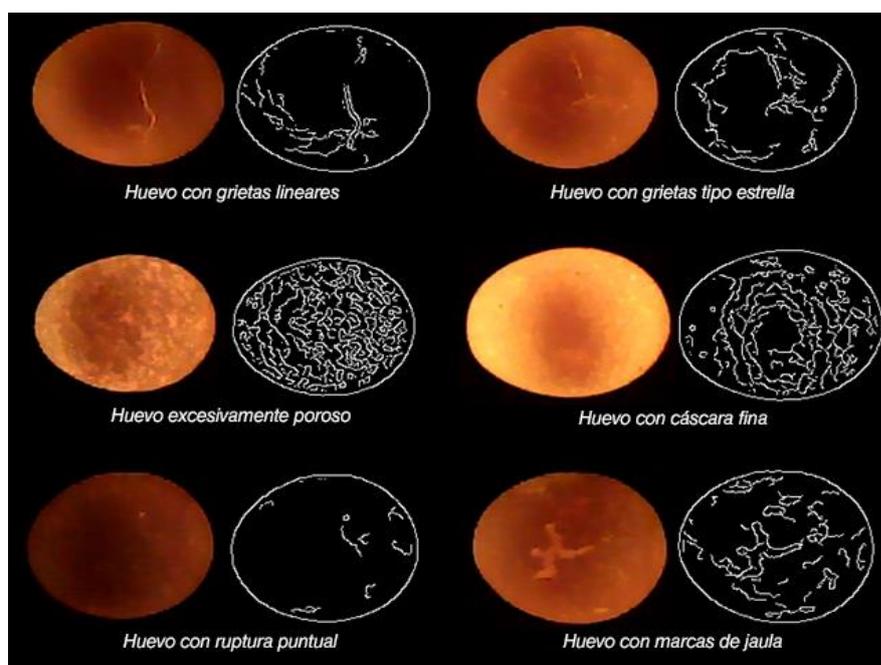


Fig. 4.30. Huevos de gallina analizados por el prototipo en base a las características que dificultan su incubación. [Investigador]

Para el análisis de 108 huevos de gallina se obtuvo un valor umbral de la métrica de comparación, dicho valor es de 0,1, aquellos huevos que están por debajo del valor determinado son aptos para incubación y todos aquellos huevos que están por encima de dicho valor deben descartarse de los procesos de incubación. En la tabla 10 se muestran los resultados de los análisis realizados automáticamente.

Tabla 4.10 Resultados de las comparaciones de cada huevo de gallina analizado. [Investigador]

Resultados obtenidos de la comparación de parámetros de los huevos de gallina analizados		
N.º Huevo	Valor Medido	Resultado
1	0,23293664	No Apto
2	0,17023643	No Apto
3	0,16508804	No Apto
4	0,29118159	No Apto
5	0,23091644	No Apto
6	0,14124969	No Apto
7	0,15733086	No Apto
8	0,23634959	No Apto
9	0,16926613	No Apto
10	0,27122207	No Apto
11	0,20505683	No Apto
12	0,11615769	No Apto
13	0,12013687	No Apto
14	0,1960595	No Apto
15	0,17213319	No Apto
16	0,1169968	No Apto
17	0,13704495	No Apto
18	0,23247796	No Apto
19	0,24170493	No Apto
20	0,21812536	No Apto
21	0,18741668	No Apto
22	0,21056632	No Apto
23	0,09733955	Apto
24	0,06028675	Apto
25	0,13515506	No Apto
26	0,1316065	No Apto
27	0,18500784	No Apto
28	0,21633667	No Apto
29	0,15516106	No Apto

30	0,17273734	No Apto
31	0,13514092	No Apto
32	0,0391915	Apto
33	0,21297818	No Apto
34	0,18173388	No Apto
35	0,20522748	No Apto
36	0,20659259	No Apto
37	0,12934983	No Apto
38	0,12360293	No Apto
39	0,01028777	Apto
40	0,28904835	No Apto
41	0,20091643	No Apto
42	0,06646956	Apto
43	0,02646196	Apto
44	0,23554951	No Apto
45	0,18926615	No Apto
46	0,26155207	No Apto
47	0,15015683	No Apto
48	0,14615763	No Apto
49	0,17013687	No Apto
50	0,1690595	No Apto
51	0,17913332	No Apto
52	0,13690682	No Apto
53	0,16705493	No Apto
54	0,18247791	No Apto
55	0,13170491	No Apto
56	0,12815534	No Apto
57	0,04111887	Apto
58	0,20956631	No Apto
59	0,19733915	No Apto
60	0,02728684	Apto
61	0,14515506	No Apto
62	0,24326066	No Apto
63	0,13950784	No Apto
64	0,12603662	No Apto
65	0,09759913	Apto
66	0,11673633	No Apto
67	0,13304091	No Apto
68	0,08375326	Apto
69	0,12997818	No Apto
70	0,18673383	No Apto
71	0,12622748	No Apto

72	0,11259259	No Apto
73	0,14293464	No Apto
74	0,07202641	Apto
75	0,14508404	No Apto
76	0,12055459	No Apto
77	0,14091633	No Apto
78	0,01424969	Apto
79	0,01573086	Apto
80	0,13634556	No Apto
81	0,11926615	No Apto
82	0,12812207	No Apto
83	0,12005683	No Apto
84	0,09622269	Apto
85	0,12013687	No Apto
86	0,13960595	No Apto
87	0,01071038	Apto
88	0,1169968	No Apto
89	0,13704495	No Apto
90	0,05248795	Apto
91	0,14170493	No Apto
92	0,11982536	No Apto
93	0,14871668	No Apto
94	0,10956632	No Apto
95	0,09037056	Apto
96	0,09029665	Apto
97	0,13515506	No Apto
98	0,02571805	Apto
99	0,15800784	No Apto
100	0,14633667	No Apto
101	0,01551805	Apto
102	0,11273734	No Apto
103	0,13540192	No Apto
104	0,01391915	Apto
105	0,1293818	No Apto
106	0,13173388	No Apto
107	0,08522748	Apto
108	0,09659259	Apto

En la figura 4.31 se muestran los resultados porcentuales de los análisis realizados mediante el método automático.

RESULTADOS DE ANÁLISIS REALIZADOS A HUEVOS DE GALLINA PARA INCUBACIÓN

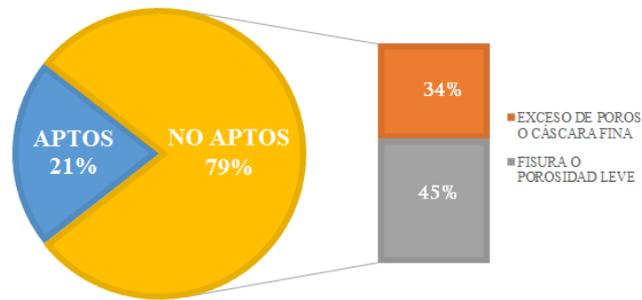


Fig. 4.31. Resultados de las comparaciones de los huevos de gallina analizados. [Investigador]

Se ha analizado el consumo de memoria RAM en el dispositivo al emplear el software desarrollado, en sus distintas etapas como se indica en la tabla 4.11.

Tabla 4.11 Resultados del consumo de memoria RAM del software desarrollado. [Investigador]

Memoria RAM empleada por el software desarrollado		
Proceso	Consumo Máximo de Memoria RAM	Porcentaje Máximo
Ventana Principal	195 MB	19%
Método Manual durante selección	440 MB	43%
Método Automático durante selección	72 MB	7%
Desplazamiento de la cámara	51 MB	5%

En la tabla 4.12 se muestra los resultados de los tiempos de ejecución de cada método incluido el método de verificación tradicional por ovoscopia individual de cada huevo de gallina. En los métodos manuales el tiempo puede incrementarse de acuerdo a la persona que realice los análisis.

Tabla 4.12 Resultados del tiempo de respuesta empleado por los métodos de análisis para el control de calidad de huevos de gallina. [Investigador]

Tiempo empleado por los métodos de análisis		
Tipo de Análisis	Tiempo de Respuesta	Cantidad de huevos por proceso
Método Automático por software desarrollado	96 s	36
Método Manual por software desarrollado	> 180 s	36
Método Manual Tradicional	> 360 s	36

El tiempo promedio que toma analizar un huevo de gallina mediante el prototipo desarrollado es de 1,72 segundos descontando los 34,08 segundos que tarda el desplazamiento de la cámara.

4.5 PRESUPUESTO

El presupuesto total para el desarrollo del prototipo, considera todos los recursos empleados como material de oficina, servicios y elementos propios del sistema de control de calidad en lo que respecta a construcción y diseño. Para determinar el costo de diseño, se considera el tiempo invertido en el mismo, analizando el salario de un Ingeniero en Electrónica y Comunicaciones establecido por el Ministerio de Trabajo en 858 dólares mensuales, considerando las 8 horas diarias laborales establecidas en Ecuador, en la tabla 4.13 se muestra del costo de Diseño del Proyecto.

Tabla 4.13 Costo de Diseño del Sistema de Control de Calidad de Huevos de Gallina. [Investigador]

Diseño del Prototipo					
Ítem	Detalle	Cantidad	Unidad	Costo Unitario	Costo Total
1	Tiempo dedicado al diseño del prototipo	90	Hora	5,36	482,63
COSTO TOTAL DEL DISEÑO					482,63

En la tabla 4.14 se detalla el costo de todos los materiales y demás insumos empleados para la elaboración del prototipo.

Tabla 4.14 Costo de Materiales del Sistema de Control de Calidad de Huevos de Gallina.

[Investigador]

Elementos para el desarrollo del prototipo					
Ítem	Detalle	Cantidad	Unidad	Costo Unitario	Costo Total
1	Raspberry Pi 3 Modelo B	1	c/u	55,9	55,9
2	Cargador 5V 3A para Raspberry	1	c/u	10	10
3	Disipadores de calor para drivers Raspberry	2	c/u	0,81	1,62
4	Case Acrílico Raspberry + Ventilador	1	c/u	6	6
5	Motor paso a paso Nema 14	2	c/u	19,25	38,5
6	Driver motor a pasos drv8825	2	c/u	3,5	7

7	Teclado + mouse inalámbrico	1	c/u	20	20
8	Micro SD 32Gb Samsung	1	c/u	30	30
9	Base metálica del prototipo	1	c/u	75	75
10	Cubeta metálica para soporte de huevos de gallina	1	c/u	75	75
11	Estructura metálica para desplazamiento de la cámara en los ejes XY	1	c/u	100	100
12	Cámara web	1	c/u	16	16
13	PCB fibra de vidrio 20x30 cm	4	c/u	7,5	30
14	Fundas ácido	8	c/u	0,6	4,8
15	Leds Blancos alto brillo	230	c/u	0,07	16,1
16	resistencias, 150 Ohms - 1/4 W	80	c/u	0,03	2,4
17	Estaño 100 gr 0,8mm	1	c/u	4,6	4,6
18	Banda dentada	4	metros	6,5	26
19	Polea GT2 para motor	2	c/u	4,3	8,6
20	Rodamiento Lineal 8mm SC8UU	4	c/u	5	20
21	Rodamientos Lineal LM8UU	4	c/u	2,5	10
22	Rodamiento 696ZZ FBJ	8	c/u	1	8
23	Varilla 8mm	20	decímetros	0,75	15
24	Borneras	10	c/u	0,2	2
25	Regleta espadines hembra 40 pines	1	c/u	0,6	0,6
26	Impresiones en papel couché	5	c/u	0,3	1,5
27	Impresiones en papel bond A4	400	c/u	0,05	20
28	Internet	12	mes	5	60
29	Foamy Pi Plus LIS NEG 60X115X2	1	c/u	1,25	1,25
COSTO TOTAL DE ELEMENTOS					665,87

Finalmente, en la Tabla 4.15 se muestra el costo total del proyecto Sumando los 2 valores principales detallados.

Tabla 4.15 Costo Total del Sistema de Control de Calidad de Huevos de Gallina. [Investigador]

Presupuesto empleado en el prototipo					
Ítem	Detalle	Cantidad	Unidad	Costo Unitario	Costo Total
1	Costo total del diseño	1	c/u	482,63	482,63
2	Costo total de elementos	1	c/u	665,87	665,87
COSTO TOTAL DEL PROYECTO					1148,5

El costo final del Sistema Electrónico para el Control de Calidad de Huevos de Gallina mediante Procesamiento de Imágenes es de \$1148,5 dólares, el cual puede ser reducido si se realiza una producción mayor, considerando que el costo de diseño se dividiría y los materiales tendrían un costo menor por adquisición al por mayor.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- El prototipo desarrollado para el control de calidad de huevos de gallina para incubación a través de métodos de procesamiento de imágenes identifica correctamente los problemas que pueden suscitarse en su composición física, pero, no es posible afirmar que el 100% de los huevos seleccionados como idóneos puedan originar pollos después del proceso de incubación ya que también depende de las condiciones biológicas, manipulación, movilización y sobre todo de la incubadora que se emplee. Este prototipo permite que los procesos sean más robustos, sobre todo en las pequeñas y medianas empresas donde se genera mayor pérdida, evitando el desperdicio de recursos debido a que los huevos seleccionados como no aptos para incubación pueden ser empleados para consumo humano, excepto aquellos que presenten fisuras y optimizando tiempo ya que se tiene un promedio de análisis de 1,72 segundos por huevo de gallina empleando el presente sistema.
- La eficiencia del procesamiento de imágenes del prototipo está limitada por 2 elementos principales aparte del software empleado, estos son: el sistema de iluminación, debido a que mientras mayor sea la cantidad de luz que genere, mayor será el nivel de detalle que pueda apreciarse en los huevos de gallina y el otro elemento es la cámara encargada de adquirir las imágenes puesto que en base a su calidad se obtiene mayor nitidez que no está ligada directamente a la cantidad de megapíxeles sino al sensor y el pre procesamiento que pueda realizar.
- El sistema de desplazamiento del prototipo, para posicionar la cámara sobre la cubeta de huevos de gallina, cuenta con 2 motores fijos que mueven un mecanismo

basado en poleas mediante bandas dentadas, tomando como referencia una impresora 3D, por lo que la fluidez y rapidez del desplazamiento depende de las características de los motores empleados y de la construcción de dicho mecanismo; el presente sistema emplea 34,08 segundos en desplazarse sobre los 4 cuadrantes determinados para el análisis de los 36 huevos de gallina abarcados. Además, este prototipo presenta una enorme ventaja en la relación costo beneficio frente otros sistemas desarrollados, debido a que su diseño es portable con dimensiones externas de 42,5 x 51,5 x 52 cm y un peso aproximado de 30 Kg, empleando en su totalidad Hardware y Software libres por lo que posterior a su adquisición o desarrollo, el usuario no tendrá gastos adicionales aparte de los mantenimientos preventivos o correctivos que se presentan en cualquier máquina.

5.2 RECOMENDACIONES

- Utilizar huevos de gallina frescos, de preferencia con pocas horas o hasta un día de almacenamiento, para tener resultados más óptimos e incrementar la probabilidad de eclosión satisfactoria, aplicando el presente sistema después de haber sometido a los huevos a procesos de limpieza, desinfección y filtrado de acuerdo al tamaño y peso adecuados para optimizar la detección de aquellos huevos aptos para incubación.
- Procurar que el prototipo cuente con iluminación adecuada, evitando filtraciones de luz externa e interna, cubriendo totalmente el prototipo y los espacios vacíos en caso de que no se inserten los 36 huevos de gallina que abarca la cubeta para incrementar la confiabilidad de los análisis, captando de esta manera mayor detalle de los huevos de gallina y reduciendo errores en el procesamiento de las imágenes.
- Ubicar el prototipo en lugares frescos y secos para evitar que alguna fuente de calor u otros factores externos influyan de alguna manera en las características de los huevos de gallina previo a su incubación, a la par, manipular el sistema cuidadosamente evitando sobre todo atascos en las bandas del mecanismo. En caso de fallos de energía, volver a posicionar el mecanismo en su origen debido a que la Raspberry Pi guarda la información procesada pero no los procesos que se estén ejecutando al apagarse de manera fortuita.

BIBLIOGRAFÍA

- [1] Revista Líderes, Diario El Comercio. “En ocho provincias se concentra el mayor consumo de cárnicos”. 15 de marzo de 2015 15:48. [En línea]. Disponible en: <http://www.revistalideres.ec/lideres/consumo-carnicos-ecuador.html>
- [2] J. Orellana. CONAVE. “El Gremio Avícola Nacional sus acciones, incidencias de las mismas y la necesidad del fortalecimiento gremial”. [En línea]. Disponible en: http://amevea-ecuador.org/web_antigua/datos/AMEVEA_2007___ING._JOSE_ORELLANA.PDF
- [3] Diario El Universo, “Consumo de pollo subió cinco veces más frente a 1990”. 12 de mayo, 2014. [En línea]. Disponible en: <https://www.eluniverso.com/noticias/2014/05/12/nota/2951971/consumo-pollo-subio-cinco-veces-mas-frente-1990>
- [4] A. H. Silva Bastidas, “Consumo voluntario y rendimiento a la canal en pollos de engorde alimentados con residuos pos cosecha de Theobroma Cacao L.”. Tesis. Universidad Técnica de Ambato. Ambato, 2016. [En línea]. Disponible en: <http://repositorio.uta.edu.ec/bitstream/123456789/23701/1/tesis%20003%20Ingenier%20C3%ADa%20Agropecuaria%20-%20Alberto%20Silva%20-%20cd%20002.pdf>
- [5] INEC, “Población y Demografía. Resultados Censo de Población 2010”, 2010. [En línea]. Disponible en: <http://www.ecuadorencifras.gob.ec/resultados/>
- [6] P. E. Chiriboga Lozada, “Evaluación de tres Balanceados Energéticos-Proteicos”. Tesis. Universidad Central del Ecuador. Quito, 2015. [En línea]. Disponible en: <http://www.dspace.uce.edu.ec/bitstream/25000/3240/1/T-UCE-0004-04.pdf>
- [7] Federación Nacional de Avicultores de Colombia, “Consumo Per Cápita De Pollo En El Mundo (Kilos/persona/año)”. Organización de las Naciones Unidas para la Alimentación, 2012.
- [8] El Telégrafo, “Ecuador produce 200 millones de pollos al año”, Diario El Telégrafo, 15 de noviembre 2013. [En línea]. Disponible en:

<https://www.eltelegrafo.com.ec/noticias/economia/4/ecuador-produce-200-millones-de-pollos-al-ano>

[9] INEC, “Encuesta de Superficie y Producción Agropecuaria Continua (ESPAC)”. Ecuador, 2014.

[10] B. Montenegro. Dirección Escuela de Ingeniería Zootecnia, Universidad Técnica Estatal de Quevedo. [Entrevista]. Quevedo, marzo del 2017.

[11] J. Abad, J. Castelló, E. Carbajo, P. Casanovas, A. Dalmau, E. García, R. Lera y R. Martínez. “Reproducción E Incubación en Avicultura”. Real Escuela de Avicultura. IMGESA. 1ª Ed. Barcelona, España, noviembre 2003, pp. 290 – 310.

[12] C. de Fatima dos Santos, D. do Vale Nascimento and J. Cardoso Marciano dos Santos, "A software for selection of eggs using digital image processing with customization between profits and quality," 2013 8th Iberian Conference on Information Systems and Technologies (CISTI), Lisboa, 2013, pp. 1-7. [En línea]. Disponible en: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6615873&isnumber=6615699>

[13] M. Medina Jiménez; J. Oñate Rosado. “Diseño e implementación de un prototipo que clasifique huevos de gallina según sus características”. Tesis. Facultad De Ingeniería, Universidad Autónoma Del Caribe. Barranquilla, Colombia, 2014. [En línea]. Disponible en: <http://repositorio.uac.edu.co/bitstream/handle/11619/1377/TMCT%200017C.pdf?sequence=1&isAllowed=y>

[14] M. Navas Lema; R. Vargas Cruz. “Diseño e implementación de un sistema de ovoscopía con visión artificial para la detección de huevos fértiles para INCUBANDINA”. Tesis. Departamento de Energía y Mecánica. Universidad de las Fuerzas Armadas ESPE, Latacunga, noviembre 2014. [En línea]. Disponible en: <https://repositorio.espe.edu.ec/bitstream/21000/9149/1/AC-ESPEL-MEC-0030.pdf>

[15] D. Enríquez Z. “Diseño, construcción y automatización de una máquina clasificadora para huevos por peso, en diferentes tamaños para 2000 huevos por hora de capacidad”. Tesis. Universidad Politécnica Salesiana Sede Quito. Quito, abril 2015.

[En línea]. Disponible en: <http://dspace.ups.edu.ec/bitstream/123456789/9091/6/UPS-KT01124.pdf>

[16] L. Gómez A.; J. Villasánchez R. “Propuesta de un sistema de visión artificial aplicada al proceso ovoscópico en la vacunación in ovo”. Tesis. Escuela Superior de Ingeniería Mecánica y Eléctrica. Instituto Politécnico Nacional. México, 2016. [En línea]. Disponible en: <http://tesis.ipn.mx/jspui/bitstream/123456789/17343/1/PROPUESTA%20DE%20UN%20SISTEMA%20DE%20VISI%C3%93N%20ARTIFICIAL%20APLICADA%20AL%20PROCESO%20OVOSC%C3%93PICO%20EN%20LA%20VACUNACI%C3%93N%20IN%20O.pdf>

[17] R. Shimizu, S. Yanagawa, T. Shimizu, M. Hamada and T. Kuroda, "Convolutional neural network for industrial egg classification," 2017 International SoC Design Conference (ISOCC), Seoul, 2017, pp. 67-68. Doi: 10.1109/ISOCC.2017.8368830. [En línea]. Disponible en: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8368830&isnumber=8368771>

[18] A. V. Vasileva, E. V. Gorbunova, A. S. Vasilev, V. S. Peretyagin, A. N. Chertov & V. V. Korotaev (2018): Assessing exterior egg quality indicators using machine vision, British Poultry Science. [En línea]. Disponible en: <https://doi.org/10.1080/00071668.2018.1523535>

[19] L. Vaca. “Producción Avícola”. Editorial Universidad Estatal a Distancia. Costa Rica, pp. 93 – 94.

[20] F. Caravaca, J. Castel, J. Guzmán, M. Delgado, Y. Mena, M. Alcalde y P. Gonzáles. “Bases de la Producción Animal”. 1ª Ed. Universidad de Sevilla. España, 2003.

[21] A Callejo. “Producción Animal: Manejo del huevo en la incubadora”. [En línea]. Universidad Politécnica de Madrid. España. Disponible en: http://ocw.upm.es/produccion-animal/produccion-avicola/contenidos/TEMA_7._INCUBACION/7-2-manejo-del-huevo-en-la-incubadora/view

- [22] A. Sánchez. Docente Facultad de Ciencia Pecuarias, Universidad Técnica Estatal de Quevedo. [Entrevista]. Quevedo, agosto del 2017.
- [23] M. Herrera. Docente Investigadora Facultad de Ciencia Pecuarias, Universidad Técnica Estatal de Quevedo. [Entrevista]. Quevedo, agosto del 2017.
- [24] S. Ricaurte Galindo. M.V.Z. “Análisis de control de calidad en incubación de huevos”. 07 de agosto de 2006. [En línea]. Disponible en: http://www.engormix.com/avicultura/articulos/analisis-control-calidad-incubacion-t26501.htm#_=_
- [25] A. Callejo. “Manejo del huevo fértil antes de la incubación”. Universidad Politécnica de Madrid. [En línea]. Disponible en: http://ocw.upm.es/produccion-animal/produccion-avicola/contenidos/TEMA_7._INCUBACION/7-1-manejo-del-huevo-fertil-antes-de-la-incubacion/view
- [26] L. Puig. “Calidad del huevo en reproductoras”. [En línea]. Disponible en: <https://agrinews.es/wp-content/uploads/2013/12/calidad-huevo-reproductoras.pdf>
- [27] “Cáscara de Huevo”. [En línea]. Disponible en: <http://www.botanical-online.com/cascaradehuevo.htm>
- [28] O. Velásquez. “Factores que afectan la productividad en la planta de incubación”. Guatemala, 06 de noviembre de 2008. [En línea]. Disponible en: <https://www.engormix.com/avicultura/articulos/planta-de-incubacion-factores-afectan-a-su-productividad-t27664.htm>
- [29] M. Romero. “Producción avícola a pequeña escala”. Secretaría De Agricultura, Ganadería, Desarrollo Rural Pesca y Alimentación. Subsecretaría de Desarrollo Rural Dirección General de Apoyos para el Desarrollo Rural. Facultad de Medicina Veterinaria y Zootecnia UAM-Xochimilco. México. [En línea]. Disponible en: <http://www.sagarpa.gob.mx/desarrolloRural/Documents/fichasaapt/Producci%C3%B3n%20Av%C3%ADcola.pdf>
- [30] X. Asensio. “La limpieza del huevo para incubar, un factor de calidad para el pollito”. AVIAGEN. Sevilla - España, abril 2014. [En línea]. Disponible en: <http://seleccionesavicolas.com/avicultura/2014/5/la-limpieza-del-huevo-para-incubar-un-factor-de-calidad-para-el-pollito>

- [31] A. M. Ariza. “Visita de Campo. Acondesa S. A: Planta de Incubación.” Fundación Universitaria San Martín. Facultad de Medicina Veterinaria y Zootecnia. 2012. [En línea]. Disponible en: <https://es.slideshare.net/annieariza/incubacion2>
- [32] “Guía de Manejo de la planta Incubadora”. Avícola Metrenco E.I.R.L. Diprodal Distribuidora y Productora Avícola Ltda. Chile. [En línea]. Disponible en: <http://www.avicolametrenco.cl/Guia%20de%20Planta%20de%20Incubacion.pdf>
- [33] Aviagen Turkeys. “Qué es un huevo incubable de buena calidad”. [En línea]. Disponible en: <http://www.aviagenturkeys.us/uploads/2015/12/18/AviagenHatchingEggPoster%20Spanish.pdf>
- [34] C. Aguinaldo. “¿Qué es la ovoscopia?”. [En línea]. Disponible en: http://www.ehowenespanol.com/ovoscopia-info_207152/
- [35] J. Souza, A. Peixoto, F. de Toledo. “Enciclopédia agrícola brasileira: N-R”. Edusp, 1995.
- [36] Amazon. “Magicfly Bright Cool LED Light Egg Candler Tester - Incubator Warehouse Exclusive”. [En línea]. Disponible en: https://www.amazon.com/Magicfly-Bright-Light-Candler-Tester/dp/B00KCKNYCY/ref=sr_1_5?s=lawn-garden&ie=UTF8&qid=1504585410&sr=1-5&keywords=Egg+Candling+Light+Candler+Tester
- [37] Fornari. “Ovoscópio: como usar e quais os benefícios”. 30 de junio de 2016. [En línea]. Disponible en: <http://www.fornariindustria.com.br/classificacao-de-ovos/ovoscopio-como-usar-e-quais-os-beneficios/>
- [38] J. Camarero. “Manual Didáctico de Cocina - Tomo I”. España, 2006. [En línea]. Disponible en: <https://books.google.com.ec/books?id=IRrmLiWACP8C&pg=PA219&dq=ovoscopia&hl=es-419&sa=X&ved=0ahUKEwjC16G5uozWAhXKOiYKHZ1PCxEQ6AEIPTAE#v=onepage&q=ovoscopia&f=false>

- [39] The Poultry Site. “Optimum Egg Quality: A Practical Approach”. The State of Queensland, Australia (through its Department of Primary Industries and Fisheries) and DSM Nutritional Products Ltd., 2007. [En línea]. Disponible en: <http://www.thepoultrysite.com/publications/1/optimum-egg-quality-a-practical-approach/>
- [40] J. Arango. “Calidad Externa e Interna del Huevo”. Avicol. Hy-Line. Bogotá – Colombia, marzo 2013. [En línea]. Disponible en: <http://avicol.co/descargas2/CalidadExternaInternaHuevo.pdf>
- [41] C. Angulo; C. Raya. “Tecnología de sistemas de control”. Universidad Politécnica de Cataluña. 1ª Ed. Ediciones UPC. Barcelona - España, septiembre de 2004.
- [42] S. Gomáriz, D. Biel, J. Matas, M. Reyes. “Teoría de Control: Diseño Electrónico”. 2ª Ed. Univ. Politèc. de Catalunya. España, 2000. pp. 15. [En línea]. Disponible en: https://books.google.com.ec/books?id=Jro3rHU_urMC&printsec=frontcover&dq=sistema+de+control+electronico&hl=es-419&sa=X&ved=0ahUKEwi6keD-1ajWAhUIWSYKHUFxDwsQ6AEILjAC#v=onepage&q&f=false
- [43] P. Lascorz Salazar. “Robótica Y Automática en El Aula: Trabajando Con Microcontroladores Picaxe”. 1ªEd. Noviembre 4, 2009. pp. 6-7.
- [44] T. Floyd. “Fundamentos de Sistemas Digitales”. 9ª Ed. PEARSON EDUCACIÓN S.A., Madrid, 2006.
- [45] D. Benchimol “Proyectos con microcontroladores aprenda a desarrollar sus propias aplicaciones”. M.P. Ediciones. Argentina.
- [46] S. Pedre. “Sistemas embebidos”. Laboratorio de Robótica y Sistemas Embebidos Departamento de Computación - FCEN – UBA. 12 de octubre de 2012. [En línea]. Disponible en: http://www-2.dc.uba.ar/charladeborrachos/presentaciones/charla_2012-10-12.pdf.
- [47] P. Aguilera López, H. Valencia Sánchez. “Imagen digital fija (Tratamiento informático de la información)”. Editex. España, 2017.

- [48] Y. Lubián Serrano. “Digitalización y captura de imágenes fotográficas. ARPG0110”. IC Editorial. España, 2013.
- [49] W. Tomasi. “Sistemas de Comunicaciones Electrónicas”. 4ª Ed. Pearson Education. México, 2003.
- [50] S. Adri Martin. “Procesamiento Digital de Imágenes”. EAE, 2011. [En línea]. Disponible en: <https://books.google.com.ec/books?id=KweppwAACAAJ&dq=%E2%80%9CProcesamiento+Digital+de+Im%C3%A1genes%E2%80%9D&hl=es-419&sa=X&ved=0ahUKEwiv-NPlkZDWAhVQySYKHSG7DiEQ6AEIMjAC>
- [51] “Python”. Copyright ©2001-2018, Python team. [En línea]. Disponible en: <https://www.python.org/>
- [52] “OpenCV”. © Copyright 2018, OpenCV team. [En línea]. Disponible en: <https://opencv.org/>
- [53] C. Vialfa. “Introducción a las bases de datos”. Abril 2017. [En línea]. Disponible en: <http://es.ccm.net/contents/66-introduccion-a-las-bases-de-datos>
- [54] “4.5.1 mysql — The MySQL Command-Line Tool”. © 2018, Oracle Corporation and/or its affiliates. [En línea]. Disponible en: <https://dev.mysql.com/doc/refman/8.0/en/mysql.html>
- [55] “About MariaDB”. 2018 Copyright MariaDB Foundation. [En línea]. Disponible en: <https://mariadb.org/about/>
- [56] “PhpMyAdmin Bringing MySQL to the web”. Copyright © 2003 - 2018 phpMyAdmin contributors. [En línea]. Disponible en: <https://www.phpmyadmin.net/>
- [57] “Energy Consumption And ‘Going Green’”. VIP LIGHTING. [En línea]. Disponible en: <http://www.viplighting.com.au/led-lighting/>
- [58] L. Kessler. “Todo sobre tiras de LEDs”. Copyright: 2007 - Afinidadelectrica.com. [En línea]. Disponible en: <http://www.afinidadelectrica.com.ar/articulo.php?IdArticulo=277>

- [59] “Raspberry Pi 3 Model B”. Raspberry Pi Foundation UK Registered Charity 1129409. [En línea]. Disponible en: <https://www.raspberrypi.org/products/raspberrypi-3-model-b/>
- [60] “BeagleBone Black”. BeagleBoard.org and the BeagleBoard.org Foundation. [En línea]. Disponible en: <https://beagleboard.org/black>
- [61] “LinkSprite pcDuino3”. LinkSprite. [En línea]. Disponible en: <http://www.linksprite.com/linksprite-pcduino3/>
- [62] “Downloads”. Raspberry Pi Foundation UK Registered Charity 1129409. [En línea]. Disponible en: <https://www.raspberrypi.org/downloads/>
- [63] “Camera Module”. Raspberry Pi Foundation UK Registered Charity 1129409. [En línea]. Disponible en: <https://www.raspberrypi.org/documentation/hardware/camera/>
- [64] “720p HD webcam FaceCam 1000X”. © 2018 KYE System Corp. [En línea]. Disponible en: <http://pe.geniusnet.com/product/facecam-1000x>
- [65] “Cámaras Web”. Copyright © 2018 INS CORP. All Rights Reserved. [En línea]. Disponible en: http://ins.com.ec/index.php?option=com_content&view=article&id=5&Itemid=18
- [66] “Royalty-Free Vector. Image of 3d coordinate axis on white”. Copyright © 2000-2018 Dreamstime. All rights reserved. [En línea]. Disponible en: <https://www.dreamstime.com/stock-illustration-d-coordinate-axis-vector-image-white-image67829314>
- [67] J. Frax. “CoreXY”. © 2018 Frax3D. [En línea]. Disponible en: <https://www.frax3d.com/corexy>
- [68] “4.4:1 Metal Gearmotor 25Dx48L mm HP 6V”. © 2001–2018 Pololu Corporation. [En línea]. Disponible en: <https://www.pololu.com/product/1570/specs>
- [69] “Especificaciones Técnicas del Servomotor Futaba”. NoMADA®. 2015. [En línea]. Disponible en: <http://nomada-e.com/descargas/datasheet/13-Servomotor%20FUTABA%20%5BS3003%5D.pdf>

[70] “Stepper Motor: Bipolar, 200 Steps/Rev, 35×26mm, 7.4V, 0.28 A/Phase”. © 2001–2018 Pololu Corporation. [En línea]. Disponible en: <https://www.pololu.com/product/1207>

[71] “DRV8825 Stepper Motor Driver Carrier, High Current”. © 2001–2018 Pololu Corporation. [En línea]. Disponible en: <https://www.pololu.com/product/2133>

[72] “GPIO”. Raspberry Pi Foundation UK Registered Charity 1129409. [En línea]. Disponible en: <https://www.raspberrypi.org/documentation/usage/gpio/README.md>

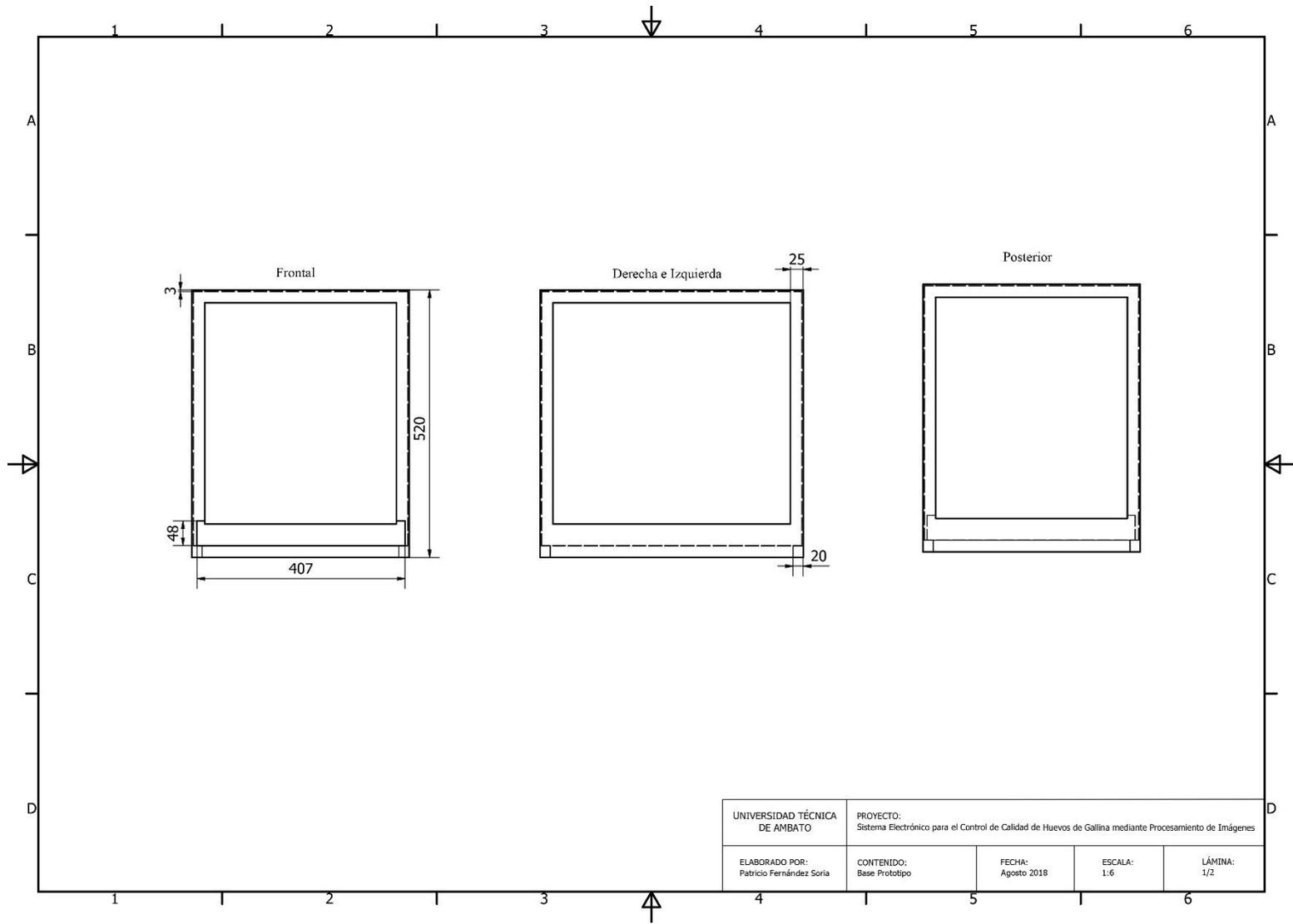
[73] “¡Asignación de pines! La guía detallada sobre la asignación de pines GPIO para Raspberry Pi”. Pinout.xyz. [En línea]. Disponible en: <https://pinout.xyz/>

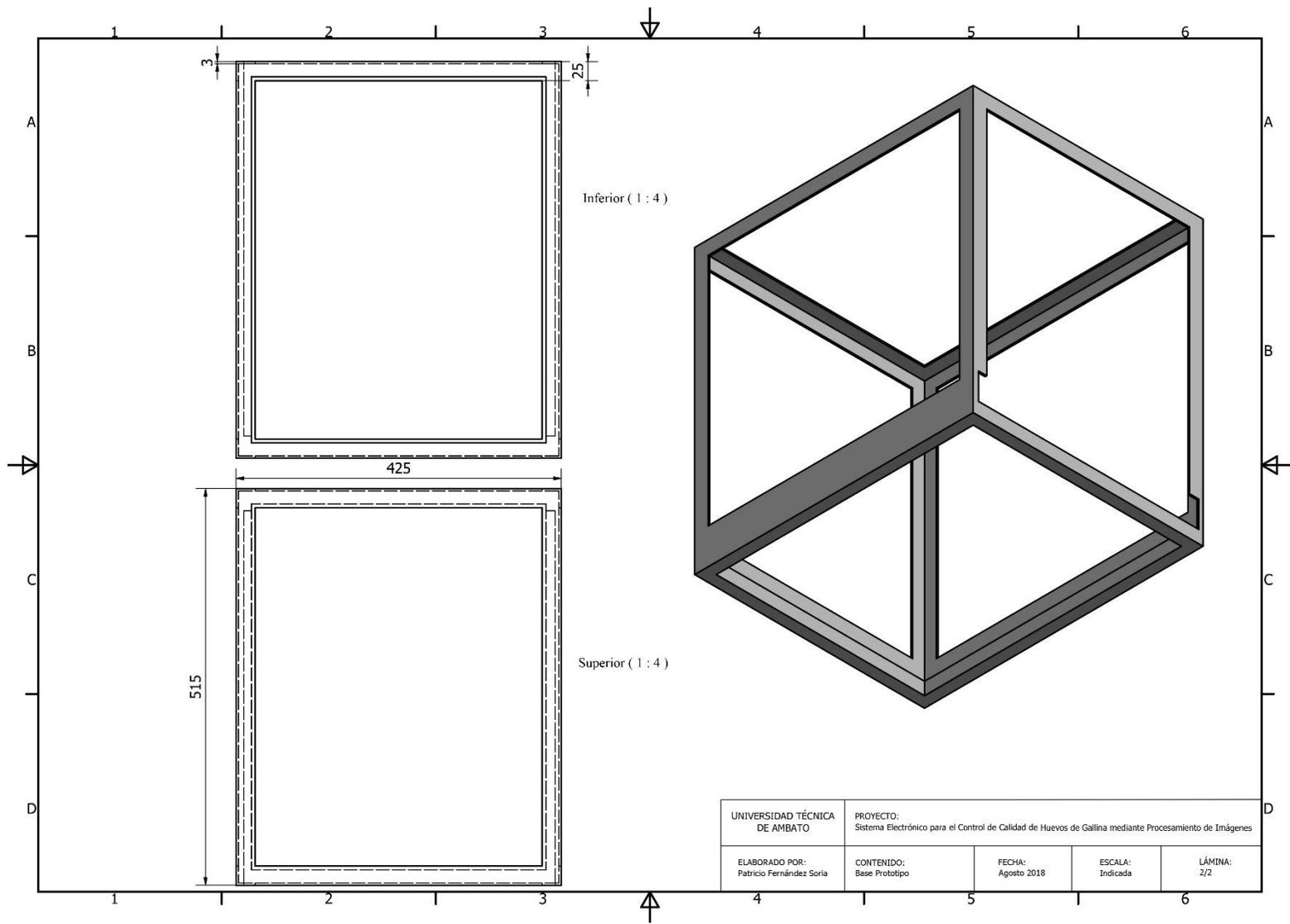
[74] A. Rosebrock. “Raspbian Stretch: Install OpenCV 3 + Python on your Raspberry Pi”. Septiembre 4, 2017. [En línea]. Disponible en: <https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/>

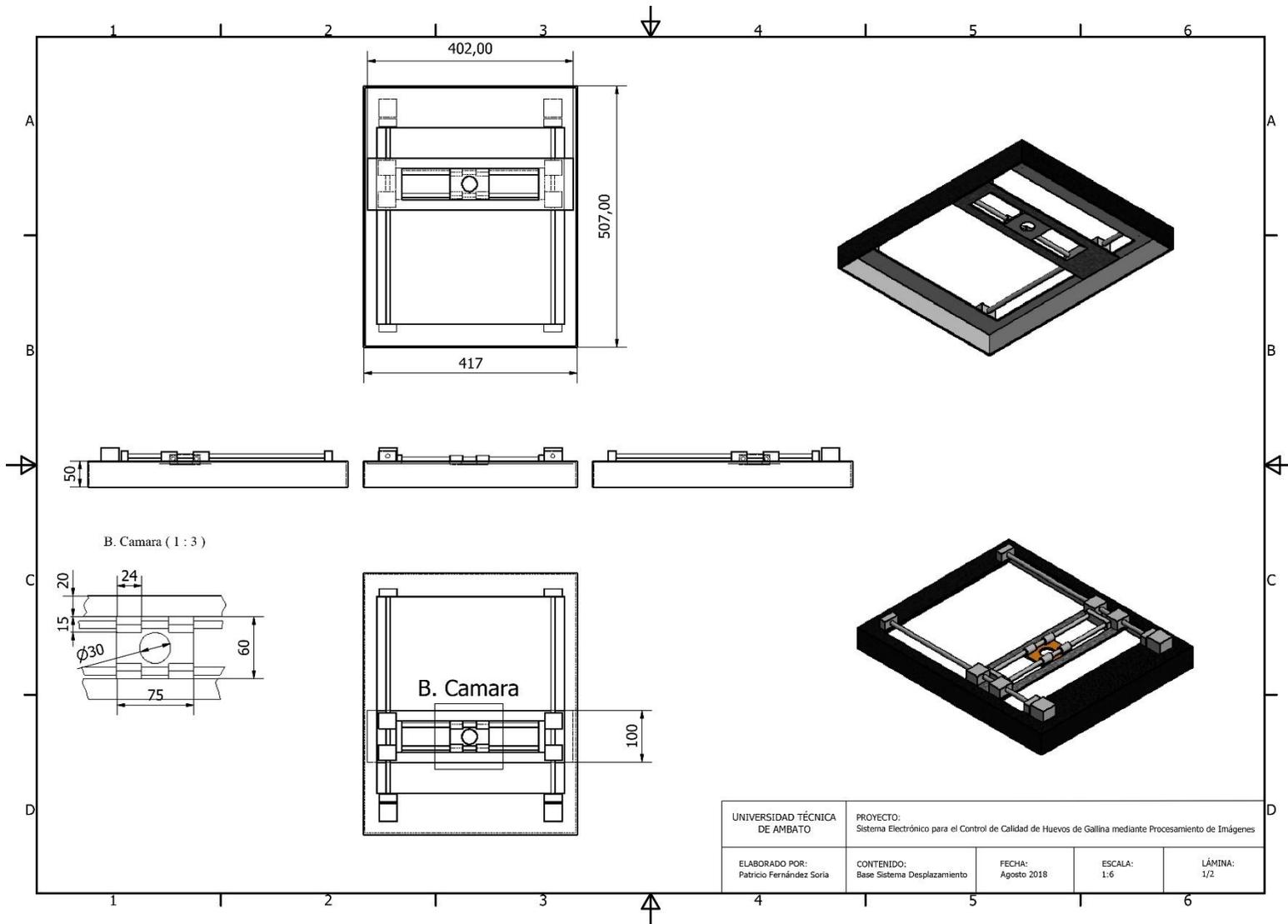
ANEXOS

Anexo 1: Planos del prototipo desarrollado.

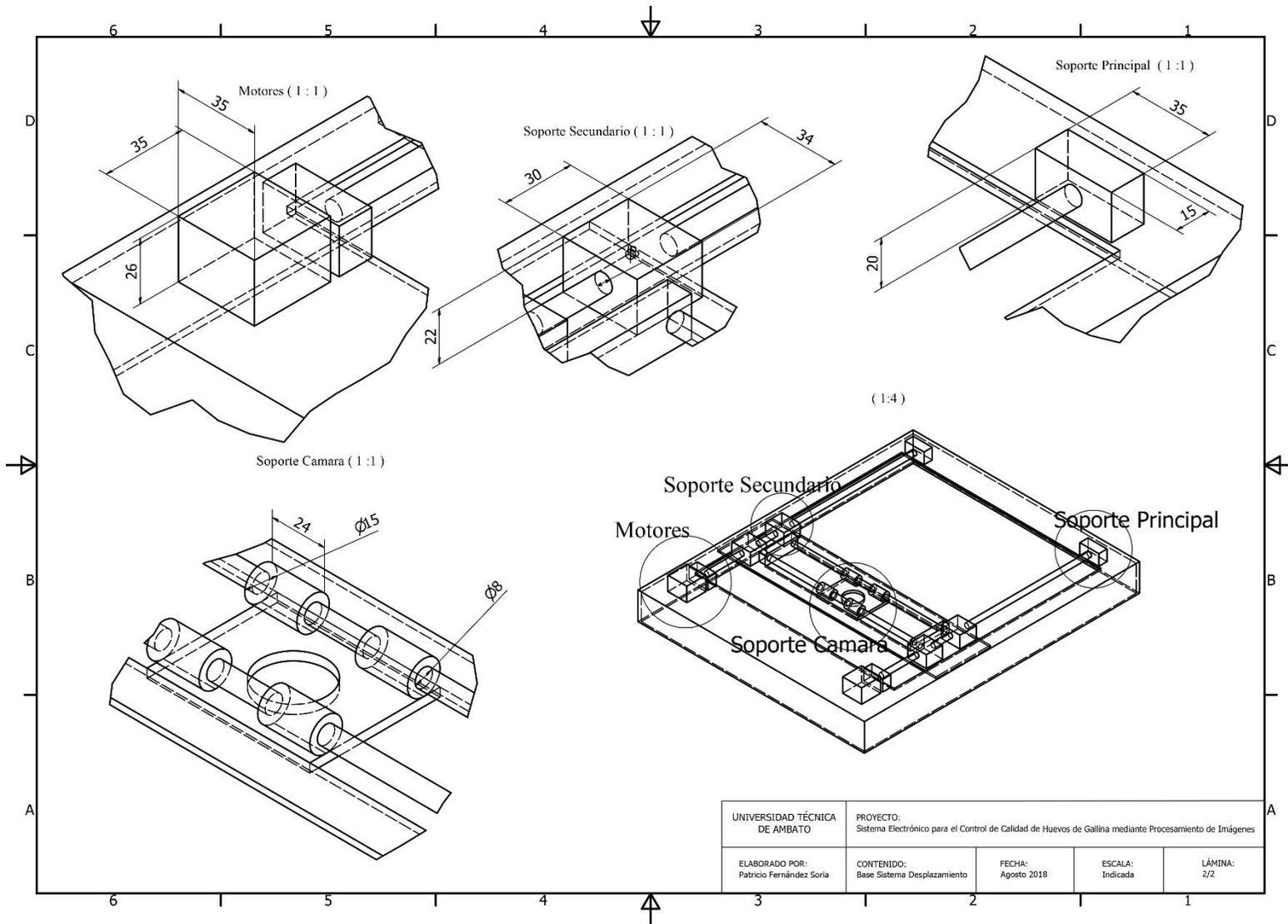
A continuación, se presentan los planos desarrollados para cada una de las bases que conforman la estructura del prototipo.



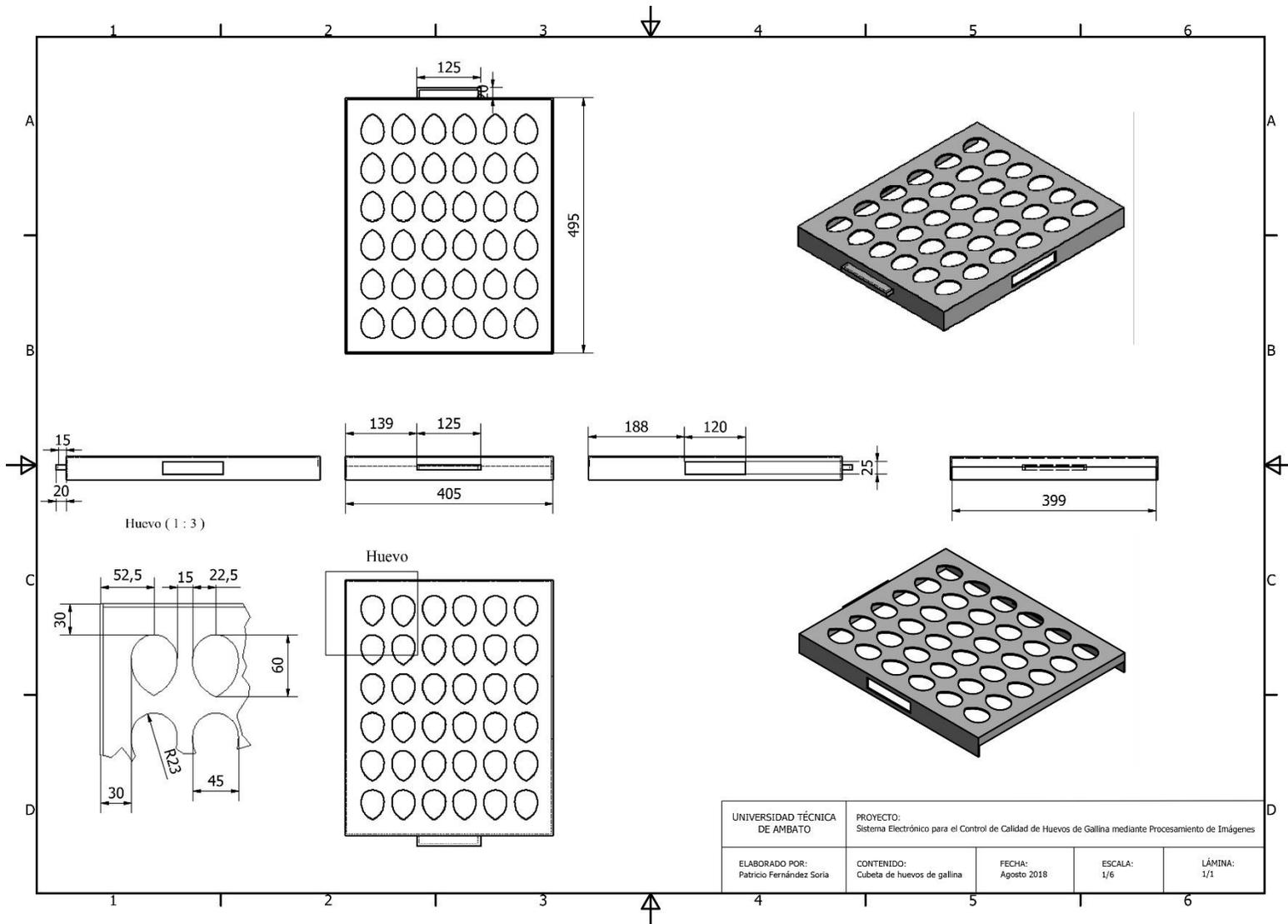




UNIVERSIDAD TÉCNICA DE AMBATO	PROYECTO: Sistema Electrónico para el Control de Calidad de Huevos de Gallina mediante Procesamiento de Imágenes			
ELABORADO POR: Patricio Fernández Soria	CONTENIDO: Base Sistema Desplazamiento	FECHA: Agosto 2018	ESCALA: 1:6	LÁMINA: 1/2



UNIVERSIDAD TÉCNICA DE AMBATO	PROYECTO: Sistema Electrónico para el Control de Calidad de Huevos de Gallina mediante Procesamiento de Imágenes			
ELABORADO POR: Patricio Fernández Soria	CONTENIDO: Base Sistema Desplazamiento	FECHA: Agosto 2018	ESCALA: Indicada	LÁMINA: 2/2



UNIVERSIDAD TÉCNICA DE AMBATO	PROYECTO: Sistema Electrónico para el Control de Calidad de Huevos de Gallina mediante Procesamiento de Imágenes			
ELABORADO POR: Patricio Fernández Soria	CONTENIDO: Cubeta de huevos de gallina	FECHA: Agosto 2018	ESCALA: 1/6	LÁMINA: 1/1

Anexo 2: Código de la microcomputadora para mover motores paso a paso con una configuración de paso completo.

El software fue desarrollado bajo lenguaje de programación Python 3.5 importando los módulos RPi.GPIO y time para el manejo de pines y micro pausas necesarias para el control de los motores.

```
import time                                     #Para las pausas
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)                       #Usar la numeración de pines de la placa

pinDir = 24                                    #Pin DIR
pinStep = 26                                   #Pin Step
pinDir1 = 16                                   #Pin DIR
pinStep1 = 18                                  #Pin Step
numSteps = 918                                 #Número de pasos del motor
numStepsf = 1160                              #Número de pasos del motor
microPausa = 0.004                            #Número de segundos de pausa

GPIO.setup(pinDir,GPIO.OUT)
GPIO.setup(pinStep,GPIO.OUT)
GPIO.setup(pinDir1,GPIO.OUT)
GPIO.setup(pinStep1,GPIO.OUT)

while True:
    time.sleep(4)                               #esperar x segundos antes de moverse
    GPIO.output(pinDir,0)                       #mover al cuadrante 2
    GPIO.output(pinDir1,0)
    for x in range(0,numSteps):
        GPIO.output(pinStep, True)
        GPIO.output(pinStep1, True)
        time.sleep(microPausa)
        GPIO.output(pinStep, False)
        GPIO.output(pinStep1, False)
        time.sleep(microPausa)
    time.sleep(4)

    GPIO.output(pinDir, 1)                     #mover al cuadrante 3
    GPIO.output(pinDir1, 0)
    for x in range(0,numStepsf):
        GPIO.output(pinStep, True)
        GPIO.output(pinStep1, True)
        time.sleep(microPausa)
        GPIO.output(pinStep, False)
        GPIO.output(pinStep1, False)
        time.sleep(microPausa)
    time.sleep(4)

    GPIO.output(pinDir,1)                       #mover al cuadrante 4
    GPIO.output(pinDir1,1)
    for x in range(0,numSteps):
        GPIO.output(pinStep, True)
        GPIO.output(pinStep1, True)
        time.sleep(microPausa)
        GPIO.output(pinStep, False)
        GPIO.output(pinStep1, False)
        time.sleep(microPausa)
    time.sleep(4)

    GPIO.output(pinDir, 0)                     #Volver al origen
    GPIO.output(pinDir1, 1)
```

```
    for x in range(0,numStepsf):
        GPIO.output(pinStep, True)
        GPIO.output(pinStep1, True)
        time.sleep(microPausa)
        GPIO.output(pinStep, False)
        GPIO.output(pinStep1, False)
        time.sleep(microPausa)
    time.sleep(4)
GPIO.cleanup()          #Para acabar correctamente
```

Anexo 3: Código de la microcomputadora para el sistema de control desarrollado.

El software fue desarrollado bajo lenguaje de programación Python 3.5 importando los módulos requeridos que contienen las definiciones y declaraciones de los distintos métodos o funciones. Para el manejo de objetos o widgets de la interfaz gráfica se emplea el módulo tkinter y PIL, para el manejo de los pines de propósito general de la microcomputadora se emplea el módulo RPi.GPIO, para el procesamiento de imágenes se importa el módulo cv2 de opencv y para las funciones propias del sistema como tiempo o concatenación de archivos se maneja los módulos sys y time.

```
import sys
import numpy as np
import cv2
import time
import RPi.GPIO as GPIO
import pymysql
from tkinter import *
from PIL import Image, ImageTk
from tkinter import messagebox

root = Tk()
root.title("Bienvenidos")
root.attributes('-zoomed', True)
img = Image.open('portada.png')
root.update()
AnchuraMaxima=root.winfo_width()
w,h = img.size
AlturaMaxima=(h*AnchuraMaxima)/w
img.thumbnail((AnchuraMaxima,AlturaMaxima), Image.ANTIALIAS)
tkimage = ImageTk.PhotoImage(img)
label=Label(root, image=tkimage, width=AnchuraMaxima,
height=AlturaMaxima).grid(row=0, column=0, columnspan=6)
img1 = Image.open('pollo.png')
tkimage1 = ImageTk.PhotoImage(img1)
label1=Label(root, image=tkimage1).grid(row=1, column=2, columnspan=2)
label2=Label(root, text="AUTOR: Ángel Patricio Fernández Soria",
font=('Times New Roman', '14','bold')).grid(row=1, column=0, columnspan=2)
label3=Label(root, text="TUTOR: Ing. Marco Jurado Lozada Mg.", font=('Times
New Roman', '14','bold')).grid(row=1, column=4, columnspan=2)
label3=Label(root, text="Seleccionar el método para iniciar:", font=('Times
New Roman', '14')).grid(row=2, column=0, columnspan=6, pady=20)
def accesomanual():
    window.deiconify()
    root.withdraw()
def accesoauto():
    win2.deiconify()
    root.withdraw()
m = Button(root, text="Manual", command=accesomanual, font=('Times New
Roman', '16'))
m.grid(row=3, column=0, columnspan=3, pady=15)
a = Button(root, text="Automático", command=accesoauto, font=('Times New
Roman', '16'))
a.grid(row=3, column=3, columnspan=3, pady=15)

GPIO.setmode(GPIO.BOARD)          #Usar la numeración de pines de la placa
GPIO.setwarnings(False)
```

```

pinDir = 24 #Pin DIR
pinStep = 26 #Pin Step
pinDir1 = 16 #Pin DIR
pinStep1 = 18 #Pin Step
numSteps = 918 #Número de pasos del motor
numStepsf = 1160 #Número de pasos del motor
microPausa = 0.004 #Número de segundos de pausa
GPIO.setup(pinDir,GPIO.OUT)
GPIO.setup(pinStep,GPIO.OUT)
GPIO.setup(pinDir1,GPIO.OUT)
GPIO.setup(pinStep1,GPIO.OUT)
window = Toplevel(root) #Crea una ventana
window.wm_title("CONTROL DE CALIDAD - MÉTODO MANUAL")
window.attributes('-zoomed', True)
window.withdraw()
def nocerrar():
    messagebox.showwarning("Advertencia","Culminar procesos antes de
cerrar")
window.protocol('WM_DELETE_WINDOW', nocerrar)
matriz = [
    [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, 0, 0],
    [0, 0, 0, 0, 0, 0]
]
matrizauto = [[0, 0, 0],
    [0, 0, 0],
    [0, 0, 0]]
matrizauto2 = [[0, 0, 0],
    [0, 0, 0],
    [0, 0, 0]]
matrizauto3 = [[0, 0, 0],
    [0, 0, 0],
    [0, 0, 0]]
matrizauto4 = [[0, 0, 0],
    [0, 0, 0],
    [0, 0, 0]]
contador = 0
all_buttons = []
imageFrame = Frame(window)
imageFrame.grid(row=1, column=6, rowspan=6, columnspan=6, padx=35)
lmain = Label(imageFrame)
lmain.grid(row=1, column=6, rowspan=6, columnspan=6, padx=35)
cap = cv2.VideoCapture(0)
m1=Label(window, text="Seleccionar huevos descartables según el
cuadrante:", font=('Times New Roman', '14'))
m1.grid(row=0, column=0, columnspan=6, pady=5)
m2=Label(window, text="Visualizador de elementos:", font=('Times New Roman',
'14'))
m2.grid(row=0, column=6, columnspan=6)
c=Label(window, text="CUADRANTE 1", font=('Times New Roman', '14','bold'))
c.grid(row=7, column=8, columnspan=2, pady=5)
def show_frame():
    _, frame = cap.read()
    cv2image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGBA)
    img = Image.fromarray(cv2image)
    imgtk = ImageTk.PhotoImage(image=img)
    lmain.imgtk = imgtk
    lmain.configure(image=imgtk)
    lmain.after(10, show_frame)
def creaMatriz():
    for y, row in enumerate(matriz):
        buttons_row = []
        for x, element in enumerate(row):

```

```

        boton = Button(window,bg="gray", width=7, height=5,
command=lambda a=x,b=y: onPressed(a,b))
        boton.grid(row=y+1, column=x)
        buttons_row.append(boton)
        all_buttons.append(buttons_row)
def onPressed(x, y):
    if all_buttons[y][x]['bg'] == 'red':
        all_buttons[y][x]['bg'] = 'gray'
        matriz[y][x] = 0
    else:
        all_buttons[y][x]['bg'] = 'red'
        matriz[y][x] = 1
def basemanual(u,v):
    conn = pymysql.connect(
        host="localhost", user="root",
        passwd="ROOT_AFS26", db="control"
    )
    mycursor = conn.cursor()
    sql = "INSERT INTO metodos (Aptos, No_Aptos, Tipo) VALUES (%s, %s, %s)"
    val = (str(u), str(v), "Manual")
    mycursor.execute(sql, val)
    conn.commit()
    conn.close()
def baseauto(r,s):
    conn = pymysql.connect(
        host="localhost", user="root",
        passwd="ROOT_AFS26", db="control"
    )
    mycursor = conn.cursor()
    sql = "INSERT INTO metodos (Aptos, No_Aptos, Tipo) VALUES (%s, %s, %s)"
    val = (str(r), str(s), "Auto")
    mycursor.execute(sql, val)
    conn.commit()
    conn.close()
def inicio():
    root.deiconify()
    window.withdraw()
    for k in range (0, 6):
        for l in range (0, 6):
            matriz[k][l] = 0
    for i in range (0, 6):
        for j in range (0, 6):
            all_buttons[i][j]['bg'] = 'gray'
def volver2():
    a2.grid_remove()
    v2.grid_remove()
    al.grid()
    ini.grid()
    a2['state'] = 'disabled'
    v2['state'] = 'disabled'
    al['state'] = 'disabled'
    ini['state'] = 'disabled'
    GPIO.output(pinDir,1) #mover al cuadrante 1 (1 Dir - 1 Dir1)
    GPIO.output(pinDir1,1)
    for x in range(0,numSteps):
        GPIO.output(pinStep, True)
        GPIO.output(pinStep1, True)
        time.sleep(microPausa)
        GPIO.output(pinStep, False)
        GPIO.output(pinStep1, False)
        time.sleep(microPausa)
    al['state'] = 'normal'
    ini['state'] = 'normal'
    c.configure(text='CUADRANTE 1')
def volver3():
    a3.grid_remove()
    v3.grid_remove()

```

```

a2.grid()
v2.grid()
a3['state'] = 'disabled'
v3['state'] = 'disabled'
a2['state'] = 'disabled'
v2['state'] = 'disabled'
GPIO.output(pinDir, 0)           #mover al cuadrante 2 (0 Dir - 1 Dir1)
GPIO.output(pinDir1, 1)
for x in range(0,numStepsf):
    GPIO.output(pinStep, True)
    GPIO.output(pinStep1, True)
    time.sleep(microPausa)
    GPIO.output(pinStep, False)
    GPIO.output(pinStep1, False)
    time.sleep(microPausa)
a2['state'] = 'normal'
v2['state'] = 'normal'
c.configure(text='CUADRANTE 2')
def volver4():
    fin.grid_remove()
    v4.grid_remove()
    a3.grid()
    v3.grid()
    fin['state'] = 'disabled'
    v4['state'] = 'disabled'
    a3['state'] = 'disabled'
    v3['state'] = 'disabled'
    GPIO.output(pinDir,0)         #mover al cuadrante 3 (0 Dir - 0 Dir1)
    GPIO.output(pinDir1,0)
    for x in range(0,numSteps):
        GPIO.output(pinStep, True)
        GPIO.output(pinStep1, True)
        time.sleep(microPausa)
        GPIO.output(pinStep, False)
        GPIO.output(pinStep1, False)
        time.sleep(microPausa)
    a3['state'] = 'normal'
    v3['state'] = 'normal'
    c.configure(text='CUADRANTE 3')
def avanzar1():
    a1.grid_remove()
    ini.grid_remove()
    a2.grid()
    v2.grid()
    a1['state'] = 'disabled'
    ini['state'] = 'disabled'
    a2['state'] = 'disabled'
    v2['state'] = 'disabled'
    GPIO.output(pinDir,0)         #mover al cuadrante 2 (0 Dir - 0 Dir1)
    GPIO.output(pinDir1,0)
    for x in range(0,numSteps):
        GPIO.output(pinStep, True)
        GPIO.output(pinStep1, True)
        time.sleep(microPausa)
        GPIO.output(pinStep, False)
        GPIO.output(pinStep1, False)
        time.sleep(microPausa)
    a2['state'] = 'normal'
    v2['state'] = 'normal'
    c.configure(text='CUADRANTE 2')
def avanzar2():
    a2.grid_remove()
    v2.grid_remove()
    a3.grid()
    v3.grid()
    a2['state'] = 'disabled'
    v2['state'] = 'disabled'

```

```

a3['state'] = 'disabled'
v3['state'] = 'disabled'
GPIO.output(pinDir, 1) #mover al cuadrante 3 (1 Dir - 0 Dir1)
GPIO.output(pinDir1, 0)
for x in range(0,numStepsf):
    GPIO.output(pinStep, True)
    GPIO.output(pinStep1, True)
    time.sleep(microPausa)
    GPIO.output(pinStep, False)
    GPIO.output(pinStep1, False)
    time.sleep(microPausa)
a3['state'] = 'normal'
v3['state'] = 'normal'
c.configure(text='CUADRANTE 3')
def avanzar3():
    a3.grid_remove()
    v3.grid_remove()
    fin.grid()
    v4.grid()
    a3['state'] = 'disabled'
    v3['state'] = 'disabled'
    fin['state'] = 'disabled'
    v4['state'] = 'disabled'
    GPIO.output(pinDir,1) #mover al cuadrante 4 (1 Dir - 1 Dir1)
    GPIO.output(pinDir1,1)
    for x in range(0,numSteps):
        GPIO.output(pinStep, True)
        GPIO.output(pinStep1, True)
        time.sleep(microPausa)
        GPIO.output(pinStep, False)
        GPIO.output(pinStep1, False)
        time.sleep(microPausa)
    fin['state'] = 'normal'
    v4['state'] = 'normal'
    c.configure(text='CUADRANTE 4')
def finalizar():
    positivos = np.sum(matriz)
    r1.configure(text="{ } Huevos Aptos".format(36-positivos)+'\n'+"{ } Huevos
NO Aptos".format(positivos))
    for i in range (0, 6):
        for j in range (0, 6):
            all_buttons[i][j]['state'] = 'disabled'
    v4.grid_remove()
    fin.grid_remove()
    in2.grid()
    new.grid()
    ex.grid()
    r2.grid_remove()
    v4['state'] = 'disabled'
    fin['state'] = 'disabled'
    in2['state'] = 'disabled'
    new['state'] = 'disabled'
    ex['state'] = 'disabled'
    r.grid()
    r1.grid()
    in2['state'] = 'normal'
    new['state'] = 'normal'
    ex['state'] = 'normal'
    basemanual((36-positivos), (positivos))
    c.configure(text='ANÁLISIS FINALIZADO')
def salir():
    GPIO.output(pinDir, 0) #Volver al origen (0 Dir - 1 Dir1)
    GPIO.output(pinDir1, 1)
    for x in range(0,numStepsf):
        GPIO.output(pinStep, True)
        GPIO.output(pinStep1, True)
        time.sleep(microPausa)

```

```

        GPIO.output(pinStep, False)
        GPIO.output(pinStep1, False)
        time.sleep(microPausa)
    root.destroy()
def reinicio():
    ex.grid_remove()
    in2.grid_remove()
    new.grid_remove()
    al.grid()
    ini.grid()
    r2.grid()
    r.grid_remove()
    r1.grid_remove()
    ex['state'] = 'disabled'
    in2['state'] = 'disabled'
    new['state'] = 'disabled'
    al['state'] = 'disabled'
    ini['state'] = 'disabled'
    for k in range(0, 6):
        for l in range(0, 6):
            matriz[k][l] = 0
    for i in range(0, 6):
        for j in range(0, 6):
            all_buttons[i][j]['bg'] = 'gray'
    GPIO.output(pinDir, 0)           #Volver al origen (0 Dir - 1 Dir1)
    GPIO.output(pinDir1, 1)
    for x in range(0,numStepsf):
        GPIO.output(pinStep, True)
        GPIO.output(pinStep1, True)
        time.sleep(microPausa)
        GPIO.output(pinStep, False)
        GPIO.output(pinStep1, False)
        time.sleep(microPausa)
    al['state'] = 'normal'
    ini['state'] = 'normal'
    c.configure(text='CUADRANTE 1')
    for i in range(0, 6):
        for j in range(0, 6):
            all_buttons[i][j]['state'] = 'normal'
def inicio2():
    ex.grid_remove()
    in2.grid_remove()
    new.grid_remove()
    al.grid()
    ini.grid()
    r2.grid()
    r.grid_remove()
    r1.grid_remove()
    ex['state'] = 'disabled'
    in2['state'] = 'disabled'
    new['state'] = 'disabled'
    al['state'] = 'disabled'
    ini['state'] = 'disabled'
    for k in range(0, 6):
        for l in range(0, 6):
            matriz[k][l] = 0
    for i in range(0, 6):
        for j in range(0, 6):
            all_buttons[i][j]['bg'] = 'gray'
    GPIO.output(pinDir, 0)           #Volver al origen (0 Dir - 1 Dir1)
    GPIO.output(pinDir1, 1)
    for x in range(0,numStepsf):
        GPIO.output(pinStep, True)
        GPIO.output(pinStep1, True)
        time.sleep(microPausa)
        GPIO.output(pinStep, False)
        GPIO.output(pinStep1, False)

```

```

        time.sleep(microPausa)
        al['state'] = 'normal'
        ini['state'] = 'normal'
        c.configure(text='CUADRANTE 1')
        for i in range (0, 6):
            for j in range (0, 6):
                all_buttons[i][j]['state'] = 'normal'
        root.deiconify()
        window.withdraw()
        ini = Button(window, text="Inicio", command=inicio, font=('Times New Roman',
        '14'))
        ini.grid(row=8, column=0, columnspan=6, pady=5)
        v2 = Button(window, text="Volver", command=volver2, font=('Times New Roman',
        '14'))
        v2.grid(row=8, column=0, columnspan=6, pady=5)
        v3 = Button(window, text="Volver", command=volver3, font=('Times New Roman',
        '14'))
        v3.grid(row=8, column=0, columnspan=6, pady=5)
        v4 = Button(window, text="Volver", command=volver4, font=('Times New Roman',
        '14'))
        v4.grid(row=8, column=0, columnspan=6, pady=5)
        a1 = Button(window, text="Avanzar", command=avanzar1, font=('Times New
        Roman', '14'))
        a1.grid(row=8, column=6, columnspan=6, pady=5)
        a2 = Button(window, text="Avanzar", command=avanzar2, font=('Times New
        Roman', '14'))
        a2.grid(row=8, column=6, columnspan=6, pady=5)
        a3 = Button(window, text="Avanzar", command=avanzar3, font=('Times New
        Roman', '14'))
        a3.grid(row=8, column=6, columnspan=6, pady=5)
        fin = Button(window, text="Finalizar", command=finalizar, font=('Times New
        Roman', '14'))
        fin.grid(row=8, column=6, columnspan=6, pady=5)
        ex = Button(window, text="Salir", command=salir, font=('Times New Roman',
        '14'))
        ex.grid(row=8, column=7, columnspan=4, pady=5)
        new = Button(window, text="Nuevo", command=reinicio, font=('Times New
        Roman', '14'))
        new.grid(row=8, column=4, columnspan=3, pady=5)
        in2 = Button(window, text="Inicio", command=inicio2, font=('Times New
        Roman', '14'))
        in2.grid(row=8, column=0, columnspan=4, pady=5)
        r=Label(window, text="RESULTADOS DEL ANÁLISIS:", font=('Times New Roman',
        '14','bold'))
        r.grid(row=7, column=0, columnspan=4, pady=5)
        r1=Label(window, font=('Times New Roman', '18','bold'), fg='red')
        r1.grid(row=7, column=4, columnspan=3, pady=5)
        r2=Label(window, text="| 2 | 3 |"+'\n'+"| 1 | 4 |", font=('Times New Roman',
        '18','bold'), fg='red')
        r2.grid(row=7, column=0, columnspan=6, pady=5)
        #show_frame()
        creaMatriz()
        v2.grid_remove()
        v3.grid_remove()
        v4.grid_remove()
        a2.grid_remove()
        a3.grid_remove()
        fin.grid_remove()
        ex.grid_remove()
        in2.grid_remove()
        new.grid_remove()
        r.grid_remove()
        r1.grid_remove()

win2 = Toplevel(root) #Crea una ventana
win2.wm_title("CONTROL DE CALIDAD - MÉTODO AUTOMÁTICO")
win2.attributes('-zoomed', True)

```

```

win2.withdraw()
def salir2():
    root.destroy()
win2.protocol('WM_DELETE_WINDOW', salir2)
def ini2():
    root.deiconify()
    win2.withdraw()
def analizar():
    win3.deiconify()
    win2.withdraw()
    clasificar()
def clasificar():
    file = "/home/pi/analisis1.png"
    file1 = "/home/pi/analisis1_1.png"
    file2 = "/home/pi/analisis1_2.png"
    file3 = "/home/pi/analisis1_3.png"
    file4 = "/home/pi/analisis2.png"
    file5 = "/home/pi/analisis2_1.png"
    file6 = "/home/pi/analisis2_2.png"
    file7 = "/home/pi/analisis2_3.png"
    file8 = "/home/pi/analisis3.png"
    file9 = "/home/pi/analisis3_1.png"
    file10 = "/home/pi/analisis3_2.png"
    file11 = "/home/pi/analisis3_3.png"
    file12 = "/home/pi/analisis4.png"
    file13 = "/home/pi/analisis4_1.png"
    file14 = "/home/pi/analisis4_2.png"
    file15 = "/home/pi/analisis4_3.png"
    filev = "/home/pi/analisis1_4.png"
    filex = "/home/pi/analisis2_4.png"
    filey = "/home/pi/analisis3_4.png"
    filez = "/home/pi/analisis4_4.png"
    #####ANÁLISIS 1#####
    camera = cv2.VideoCapture(0)
    end_time = time.time() + 15
    while(time.time() < end_time):
        retval, cam = camera.read()
    emLoop= True
    while(emLoop):
        cv2.imwrite(file,cam)
        original = cv2.imread("/home/pi/analisis1.png")
        gris = cv2.cvtColor(original, cv2.COLOR_BGR2GRAY)
        gauss = cv2.GaussianBlur(gris, (5,5), 0)
        cv2.imwrite(file1,gauss)
        canny = cv2.Canny(gauss, 0, 24)
        ccanny=canny[0:480, 10:625]
        cv2.imwrite(file2,ccanny)
        coriginal=original[0:480, 10:625]
        (_, contornos,_) = cv2.findContours(ccanny.copy(),
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        cont=cv2.drawContours(coriginal,contornos,-1,(0,255,0), 2)
        cv2.imwrite(file3,cont)
        cantidad=format(len(contornos))
        #print(cantidad)
        compx = cv2.imread("/home/pi/comparar.png")
        grisex = cv2.cvtColor(compx, cv2.COLOR_BGR2GRAY)
        gaussx = cv2.GaussianBlur(grisex, (5,5), 0)
        cannyx = cv2.Canny(gaussx, 100, 240)
        matrizauto[0][0] = cv2.matchShapes(cannyx,ccanny[0:170,
15:225],1,0.0)
        matrizauto[0][1] = cv2.matchShapes(cannyx,ccanny[0:170,
210:425],1,0.0)
        matrizauto[0][2] = cv2.matchShapes(cannyx,ccanny[0:170,
410:625],1,0.0)
        matrizauto[1][0] = cv2.matchShapes(cannyx,ccanny[160:330,
15:225],1,0.0)

```

```

        matrizauto[1][1] = cv2.matchShapes(cannyx, ccanny[160:330,
210:425], 1, 0.0)
        matrizauto[1][2] = cv2.matchShapes(cannyx, ccanny[160:330,
410:625], 1, 0.0)
        matrizauto[2][0] = cv2.matchShapes(cannyx, ccanny[320:480,
15:225], 1, 0.0)
        matrizauto[2][1] = cv2.matchShapes(cannyx, ccanny[320:480,
210:425], 1, 0.0)
        matrizauto[2][2] = cv2.matchShapes(cannyx, ccanny[320:480,
410:625], 1, 0.0)
        for i in range(0, 3):
            for j in range(0, 3):
                if matrizauto[j][i] > 0.1:
                    contador = contador+1
                    font = cv2.FONT_HERSHEY_SIMPLEX

                    cv2.putText(cont, 'X', ((i*200+90), (j*160+110)), font,
3, (255, 255, 255), 8, cv2.LINE_AA)
                    cv2.imwrite(filev, cont)
                    a = np.array(matrizauto)
                    #print(a)
                    camera.release()
                    emLoop= False
#####MOVER 2#####
GPIO.output(pinDir, 0)                #mover al cuadrante 2 (0 Dir - 0 Dir1)
GPIO.output(pinDir1, 0)
for x in range(0, numSteps):
    GPIO.output(pinStep, True)
    GPIO.output(pinStep1, True)
    time.sleep(microPausa)
    GPIO.output(pinStep, False)
    GPIO.output(pinStep1, False)
    time.sleep(microPausa)
#####ANÁLISIS 2#####
camera2 = cv2.VideoCapture(0)
end_time = time.time() + 15
while(time.time() < end_time):
    retval2, cam2 = camera2.read()
emLoop2= True
while(emLoop2):
    cv2.imwrite(file4, cam2)
    original2 = cv2.imread("/home/pi/analisis2.png")
    gris2 = cv2.cvtColor(original2, cv2.COLOR_BGR2GRAY)
    gauss2 = cv2.GaussianBlur(gris2, (5,5), 0)
    cv2.imwrite(file5, gauss2)
    canny2 = cv2.Canny(gauss2, 0, 24)
    ccanny2=canny2[0:480, 10:625]
    cv2.imwrite(file6, ccanny2)
    coriginal2=original2[0:480, 10:625]
    (_, contornos2, _) = cv2.findContours(ccanny2.copy(),
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    cont2=cv2.drawContours(coriginal2, contornos2, -1, (0, 255, 0), 2)
    cv2.imwrite(file7, cont2)
    cantidad2=format(len(contornos2))
    #print(cantidad2)
    compx = cv2.imread("/home/pi/comparar.png")
    grisx = cv2.cvtColor(compx, cv2.COLOR_BGR2GRAY)
    gaussx = cv2.GaussianBlur(grisx, (5,5), 0)
    cannyx = cv2.Canny(gaussx, 100, 240)
    matrizauto2[0][0] = cv2.matchShapes(cannyx, ccanny2[0:170,
15:225], 1, 0.0)
    matrizauto2[0][1] = cv2.matchShapes(cannyx, ccanny2[0:170,
210:425], 1, 0.0)
    matrizauto2[0][2] = cv2.matchShapes(cannyx, ccanny2[0:170,
410:625], 1, 0.0)
    matrizauto2[1][0] = cv2.matchShapes(cannyx, ccanny2[160:330,
15:225], 1, 0.0)

```

```

        matrizauto2[1][1] = cv2.matchShapes(cannyx, ccanny2[160:330,
210:425],1,0.0)
        matrizauto2[1][2] = cv2.matchShapes(cannyx, ccanny2[160:330,
410:625],1,0.0)
        matrizauto2[2][0] = cv2.matchShapes(cannyx, ccanny2[320:480,
15:225],1,0.0)
        matrizauto2[2][1] = cv2.matchShapes(cannyx, ccanny2[320:480,
210:425],1,0.0)
        matrizauto2[2][2] = cv2.matchShapes(cannyx, ccanny2[320:480,
410:625],1,0.0)
        for i in range(0,3):
            for j in range(0,3):
                if matrizauto2[j][i] > 0.1:
                    contador = contador+1
                    font = cv2.FONT_HERSHEY_SIMPLEX
                    cv2.putText(cont2, 'X', ((i*200+90), (j*160+110)), font,
3, (255,255,255), 8, cv2.LINE_AA)
                    cv2.imwrite(filex, cont2)
                    a2 = np.array(matrizauto2)
                    #print(a2)
                    camera2.release()
                    emLoop2= False
#####MOVER 3#####
GPIO.output(pinDir, 1)                #mover al cuadrante 3 (1 Dir - 0 Dir1)
GPIO.output(pinDir1, 0)
for x in range(0, numStepsf):
    GPIO.output(pinStep, True)
    GPIO.output(pinStep1, True)
    time.sleep(microPausa)
    GPIO.output(pinStep, False)
    GPIO.output(pinStep1, False)
    time.sleep(microPausa)
#####ANÁLISIS 3#####
camera3 = cv2.VideoCapture(0)
end_time = time.time() + 15
while(time.time() < end_time):
    retval3, cam3 = camera3.read()
emLoop3= True
while(emLoop3):
    cv2.imwrite(file8, cam3)
    original3 = cv2.imread("/home/pi/analisis3.png")
    gris3 = cv2.cvtColor(original3, cv2.COLOR_BGR2GRAY)
    gauss3 = cv2.GaussianBlur(gris3, (5,5), 0)
    cv2.imwrite(file9, gauss3)
    canny3 = cv2.Canny(gauss3, 0, 24)
    ccanny3=canny3[0:480, 10:625]
    cv2.imwrite(file10, ccanny3)
    coriginal3=original3[0:480, 10:625]
    (_, contornos3, _) = cv2.findContours(ccanny3.copy(),
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    cont3=cv2.drawContours(coriginal3, contornos3, -1, (0,255,0), 2)
    cv2.imwrite(file11, cont3)
    cantidad3=format(len(contornos3))
    #print(cantidad3)
    compx = cv2.imread("/home/pi/comparar.png")
    grissx = cv2.cvtColor(compx, cv2.COLOR_BGR2GRAY)
    gaussx = cv2.GaussianBlur(grissx, (5,5), 0)
    cannyx = cv2.Canny(gaussx, 100, 240)
    matrizauto3[0][0] = cv2.matchShapes(cannyx, ccanny3[0:170,
15:225],1,0.0)
    matrizauto3[0][1] = cv2.matchShapes(cannyx, ccanny3[0:170,
210:425],1,0.0)
    matrizauto3[0][2] = cv2.matchShapes(cannyx, ccanny3[0:170,
410:625],1,0.0)
    matrizauto3[1][0] = cv2.matchShapes(cannyx, ccanny3[160:330,
15:225],1,0.0)

```

```

        matrizauto3[1][1] = cv2.matchShapes(cannyx, ccanny3[160:330,
210:425], 1, 0.0)
        matrizauto3[1][2] = cv2.matchShapes(cannyx, ccanny3[160:330,
410:625], 1, 0.0)
        matrizauto3[2][0] = cv2.matchShapes(cannyx, ccanny3[320:480,
15:225], 1, 0.0)
        matrizauto3[2][1] = cv2.matchShapes(cannyx, ccanny3[320:480,
210:425], 1, 0.0)
        matrizauto3[2][2] = cv2.matchShapes(cannyx, ccanny3[320:480,
410:625], 1, 0.0)
        for i in range(0, 3):
            for j in range(0, 3):
                if matrizauto3[j][i] > 0.1:
                    contador = contador+1
                    font = cv2.FONT_HERSHEY_SIMPLEX
                    cv2.putText(cont3, 'X', ((i*200+90), (j*160+110)), font,
3, (255, 255, 255), 8, cv2.LINE_AA)
                    cv2.imwrite(filey, cont3)
                    a3 = np.array(matrizauto3)
                    #print(a3)
                    camera3.release()
                    emLoop3= False
#####MOVER 4#####
GPIO.output(pinDir, 1)                #mover al cuadrante 4 (1 Dir - 1 Dir1)
GPIO.output(pinDir1, 1)
for x in range(0, numSteps):
    GPIO.output(pinStep, True)
    GPIO.output(pinStep1, True)
    time.sleep(microPausa)
    GPIO.output(pinStep, False)
    GPIO.output(pinStep1, False)
    time.sleep(microPausa)
#####ANÁLISIS 4#####
camera4 = cv2.VideoCapture(0)
end_time = time.time() + 15
while(time.time() < end_time):
    retval4, cam4 = camera4.read()
emLoop4= True
while(emLoop4):
    cv2.imwrite(file12, cam4)
    original4 = cv2.imread("/home/pi/analisis4.png")
    gris4 = cv2.cvtColor(original4, cv2.COLOR_BGR2GRAY)
    gauss4 = cv2.GaussianBlur(gris4, (5,5), 0)
    cv2.imwrite(file13, gauss4)
    canny4 = cv2.Canny(gauss4, 0, 24)
    ccanny4=canny4[0:480, 10:625]
    cv2.imwrite(file14, ccanny4)
    coriginal4=original4[0:480, 10:625]
    (_, contornos4, _) = cv2.findContours(ccanny4.copy(),
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    cont4=cv2.drawContours(coriginal4, contornos4, -1, (0,255,0), 2)
    cv2.imwrite(file15, cont4)
    cantidad4=format(len(contornos4))
    #print(cantidad4)
    compx = cv2.imread("/home/pi/comparar.png")
    grisex = cv2.cvtColor(compx, cv2.COLOR_BGR2GRAY)
    gaussx = cv2.GaussianBlur(grisex, (5,5), 0)
    cannyx = cv2.Canny(gaussx, 100, 240)
    matrizauto4[0][0] = cv2.matchShapes(cannyx, ccanny4[0:170,
15:225], 1, 0.0)
    matrizauto4[0][1] = cv2.matchShapes(cannyx, ccanny4[0:170,
210:425], 1, 0.0)
    matrizauto4[0][2] = cv2.matchShapes(cannyx, ccanny4[0:170,
410:625], 1, 0.0)
    matrizauto4[1][0] = cv2.matchShapes(cannyx, ccanny4[160:330,
15:225], 1, 0.0)

```

```

        matrizauto4[1][1] = cv2.matchShapes(cannyx, ccanny4[160:330,
210:425], 1, 0.0)
        matrizauto4[1][2] = cv2.matchShapes(cannyx, ccanny4[160:330,
410:625], 1, 0.0)
        matrizauto4[2][0] = cv2.matchShapes(cannyx, ccanny4[320:480,
15:225], 1, 0.0)
        matrizauto4[2][1] = cv2.matchShapes(cannyx, ccanny4[320:480,
210:425], 1, 0.0)
        matrizauto4[2][2] = cv2.matchShapes(cannyx, ccanny4[320:480,
410:625], 1, 0.0)
        for i in range(0, 3):
            for j in range(0, 3):
                if matrizauto4[j][i] > 0.1:
                    contador = contador+1
                    font = cv2.FONT_HERSHEY_SIMPLEX
                    cv2.putText(cont4, 'X', ((i*200+90), (j*160+110)), font,
3, (255, 255, 255), 8, cv2.LINE_AA)
                    cv2.imwrite(filez, cont4)
                    a4 = np.array(matrizauto4)
                    #print(a4)
                    camera4.release()
                    emLoop4= False
                    baseauto((36-contador), (contador))
                    #####MOVER 1#####
                    GPIO.output(pinDir, 0)           #Volver al origen (0 Dir - 1 Dir1)
                    GPIO.output(pinDir1, 1)
                    for x in range(0, numStepsf):
                        GPIO.output(pinStep, True)
                        GPIO.output(pinStep1, True)
                        time.sleep(microPausa)
                        GPIO.output(pinStep, False)
                        GPIO.output(pinStep1, False)
                        time.sleep(microPausa)
                    inia = Button(win2, text="Inicio", command=ini2, font=('Times New Roman',
'14'))
                    inia.grid(row=2, column=0)
                    aemp = Button(win2, text="Analizar", command=analizar, font=('Times New
Roman', '14'))
                    aemp.grid(row=2, column=1)
                    esp=Label(win2, text="Esperando orden...", font=('Times New Roman',
'16', 'bold'))
                    esp.grid(row=0, column=0, columnspan=2)
                    win2.columnconfigure(0, weight=1)
                    win2.columnconfigure(1, weight=1)
                    win2.rowconfigure(0, weight=1)
                    win2.rowconfigure(1, weight=1)
                    win2.rowconfigure(2, weight=1)
                    imga = Image.open('pollo.png')
                    tkimagea = ImageTk.PhotoImage(imga)
                    labela=Label(win2, image=tkimagea).grid(row=1, column=0, columnspan=2)

                    win3 = Toplevel(root) #Crea una ventana
                    win3.wm_title("RESULTADOS MÉTODO AUTOMÁTICO")
                    win3.attributes('-zoomed', True)
                    win3.withdraw()
                    def salir3():
                        root.destroy()
                    win3.protocol('WM_DELETE_WINDOW', salir3)
                    def ini3():
                        root.deiconify()
                        win3.withdraw()
                    def rein3():
                        win2.deiconify()
                        win3.withdraw()
                    imgal = Image.open(' analisis1_4.png')
                    imgal.thumbnail((384, 288), Image.ANTIALIAS)
                    tkimagea1 = ImageTk.PhotoImage(imgal)

```

```

label31=Label(win3, image=tkimagea1).grid(row=1, column=1, columnspan=2)
imga2 = Image.open(' analisis2_4.png')
imga2.thumbnail((384,288), Image.ANTIALIAS)
tkimagea2 = ImageTk.PhotoImage(imga2)
label32=Label(win3, image=tkimagea2).grid(row=0, column=1, columnspan=2)
imga3 = Image.open(' analisis3_4.png')
imga3.thumbnail((384,288), Image.ANTIALIAS)
tkimagea3 = ImageTk.PhotoImage(imga3)
label33=Label(win3, image=tkimagea3).grid(row=0, column=3, columnspan=2)
imga4 = Image.open(' analisis4_4.png')
imga4.thumbnail((384,288), Image.ANTIALIAS)
tkimagea4 = ImageTk.PhotoImage(imga4)
label34=Label(win3, image=tkimagea4).grid(row=1, column=3, columnspan=2)
ini3 = Button(win3, text="Inicio", command=ini3, font=('Times New Roman',
'14'))
ini3.grid(row=3, column=0, columnspan=2, pady=3)
asal = Button(win3, text="Nuevo", command=rein3, font=('Times New Roman',
'14'))
asal.grid(row=3, column=2, columnspan=2, pady=3)
arein = Button(win3, text="Salir", command=salir3, font=('Times New Roman',
'14'))
arein.grid(row=3, column=4, columnspan=2, pady=3)
ar=Label(win3, text="TOTAL: {} Huevos Aptos".format(36-contador)+'\n'+ " y {}
Huevos No Aptos".format(contador), font=('Times New Roman', '16','bold'))
ar.grid(row=2, column=1, columnspan=4, pady=3)
win3.columnconfigure(5, weight=1)
win3.columnconfigure(0, weight=1)

root.mainloop()

```

Anexo 4: Imágenes de los análisis efectuados por el prototipo.

A continuación, se presenta una muestra de las imágenes adquiridas por el prototipo para el procesamiento y análisis de los huevos de gallina aptos para incubación.

En la ilustración 1 se presenta una de las imágenes adquiridas por el prototipo para ser procesada.

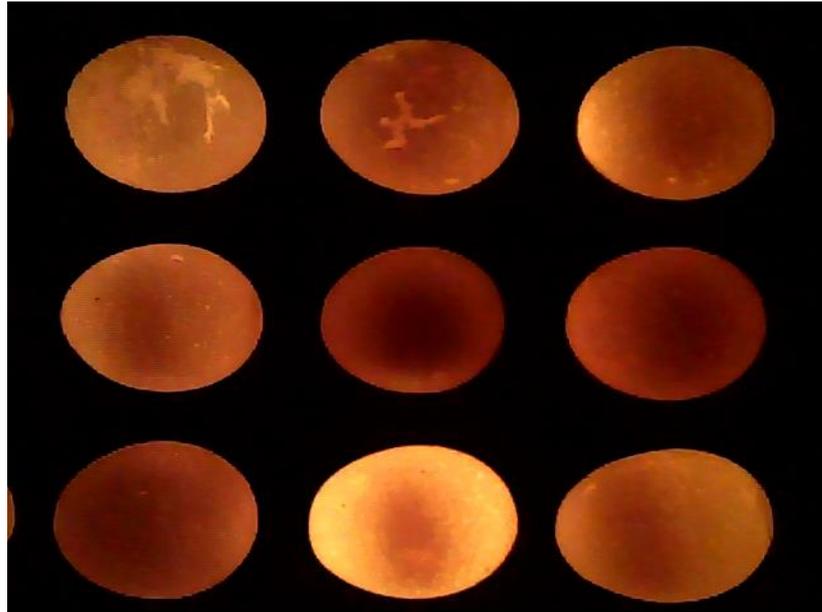


Ilustración 1. Imagen normal, de uno de los cuadrantes de la cubeta, adquirida por la cámara. [Investigador]

En la ilustración 2 se muestra la imagen resultante de la conversión a escala de grises y aplicación de filtro gaussiano para la reducción de ruido en la imagen adquirida.

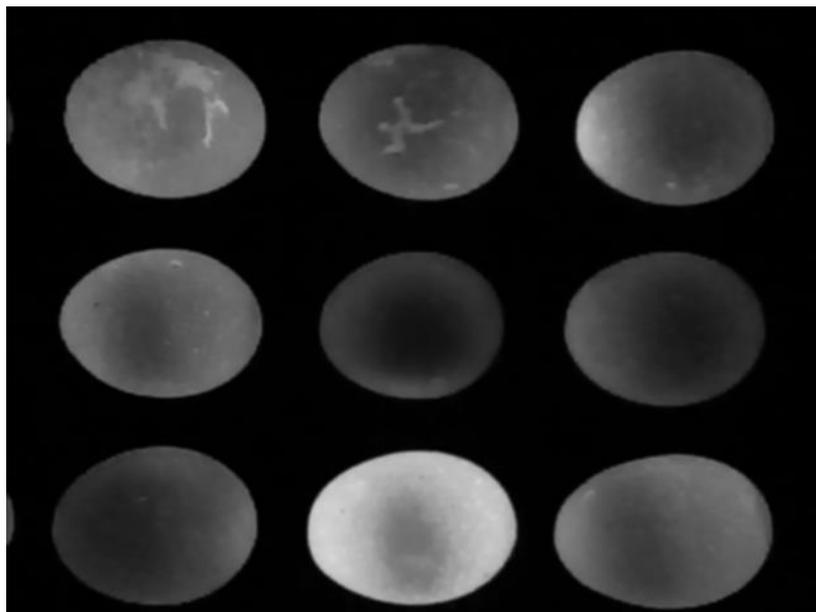


Ilustración 2. Conversión a escala de grises y aplicación de filtro gaussiano en la imagen adquirida por la cámara. [Investigador]

Posterior a la aplicación del filtro gaussiano se obtienen las características de los huevos de gallina como se muestra en la ilustración 3.

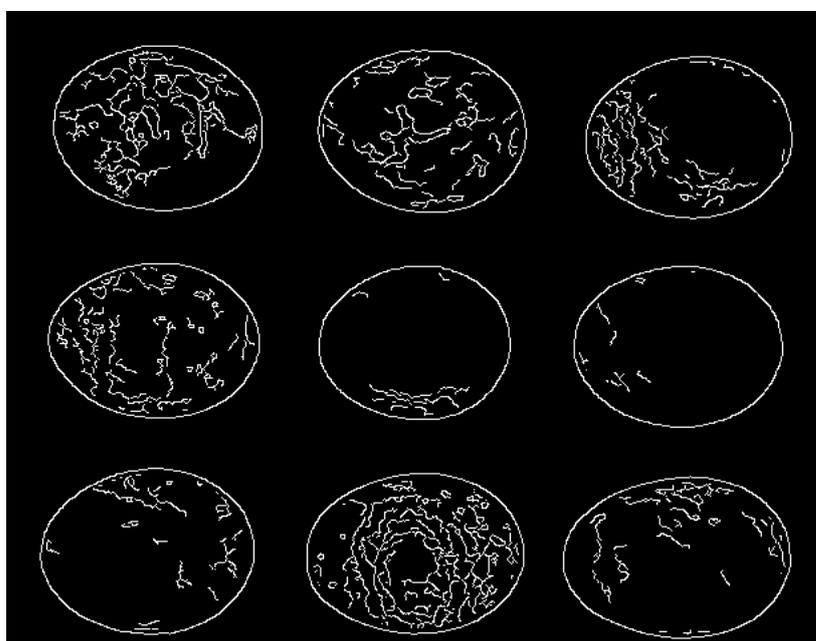


Ilustración 3. Determinación de las características intrínsecas de los huevos de gallina analizados. [Investigador]

En base a las características obtenidas se determinan los bordes de los huevos analizados para realizar un conteo de los mismos verificando que efectivamente se

encuentran colocados los 9 elementos en el presente cuadrante de la cubeta como se muestra en la ilustración 4.

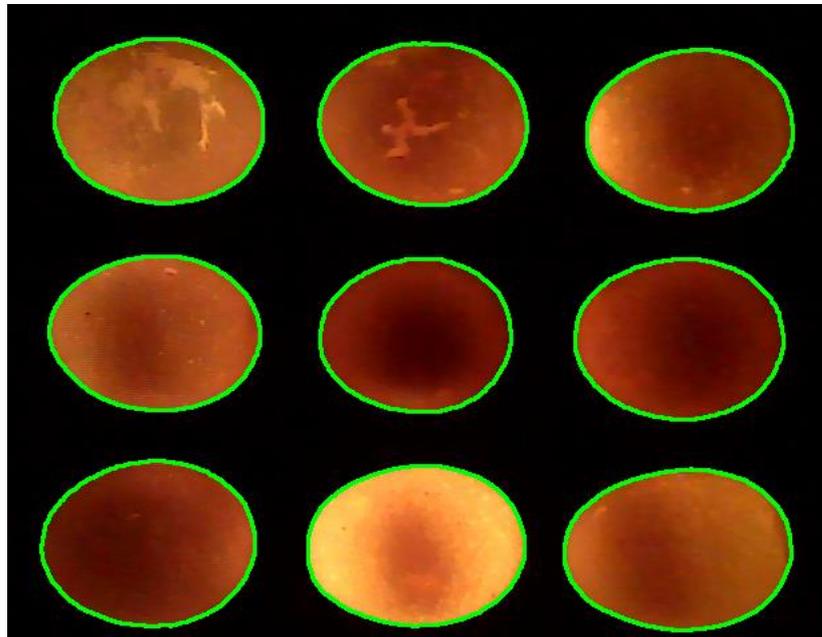


Ilustración 4. Determinación de bordes de los huevos de gallina analizados para su respectivo conteo. [Investigador]

Continuando con la verificación de los parámetros establecidos, se realiza una segmentación de la imagen resultante del análisis de características como se muestra en la ilustración 5.

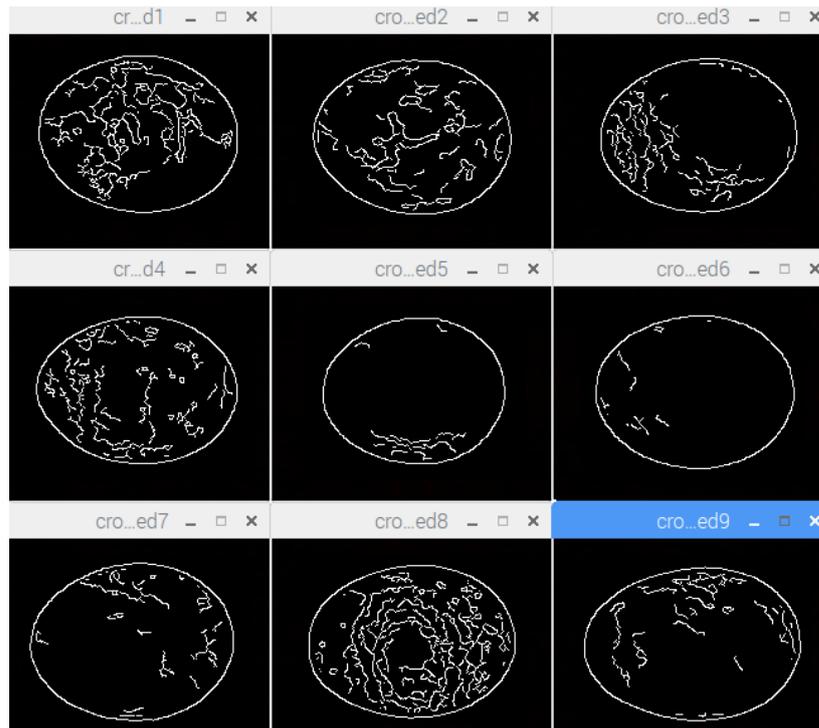


Ilustración 5. Segmentación de la imagen procesada con las características de cada uno de los huevos de gallina analizados. [Investigador]

Cada uno de los segmentos de la imagen principal, es decir cada huevo de gallina analizado, es comparado con la imagen base mostrada anteriormente determinando un valor resultante de esta comparación y mostrando visualmente aquellos huevos aptos y no aptos para su respectiva incubación de acuerdo al concerniente cuadrante sobre la cubeta del prototipo como se muestra en la ilustración 6.

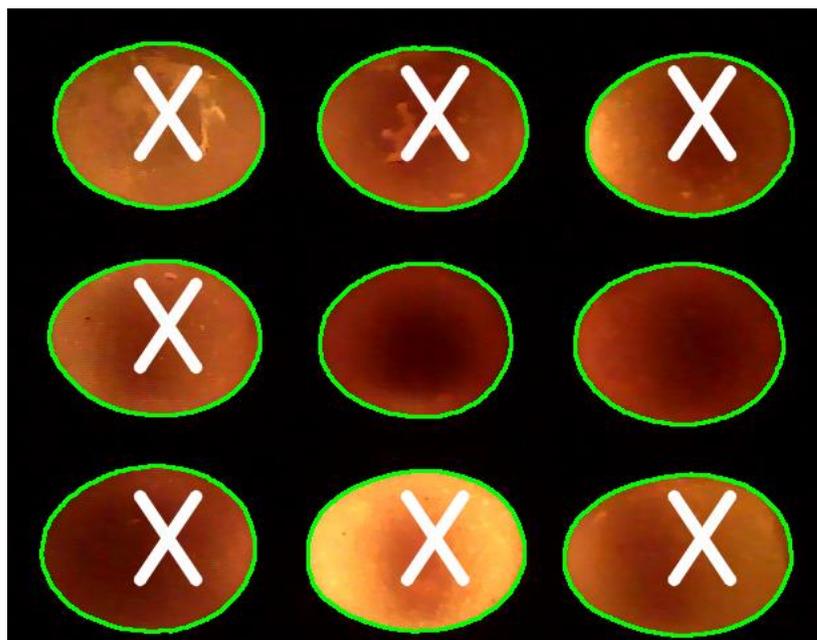


Ilustración 6. Resultados del procesamiento de imágenes de los huevos de gallina analizados. [Investigador]

Además, en los análisis, se pueden presentar errores al momento de adquirir y procesar las imágenes si no se coloca en su totalidad los huevos de gallina sobre la cubeta, ya que por la configuración de la misma pasa directamente la luz del sistema de iluminación afectando claramente a la cámara como se indica en la ilustración 7.

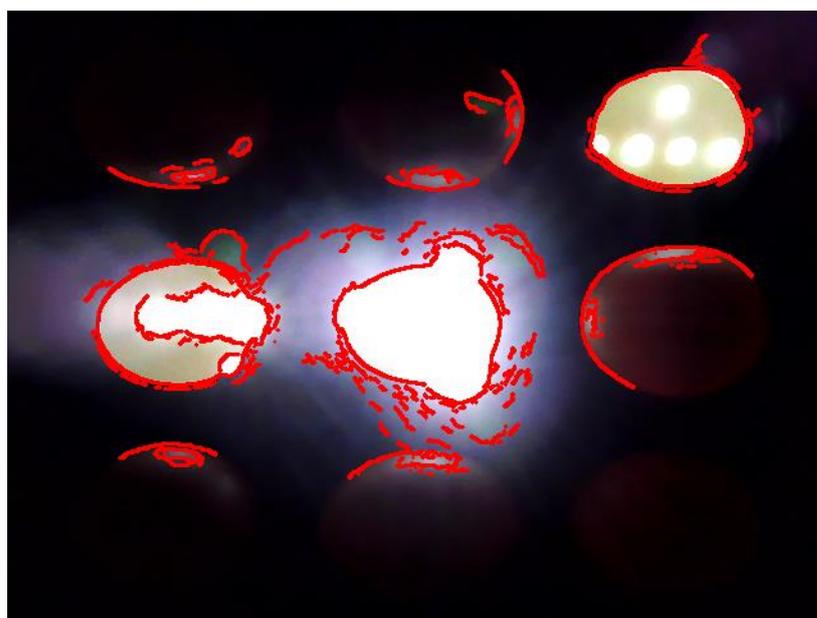
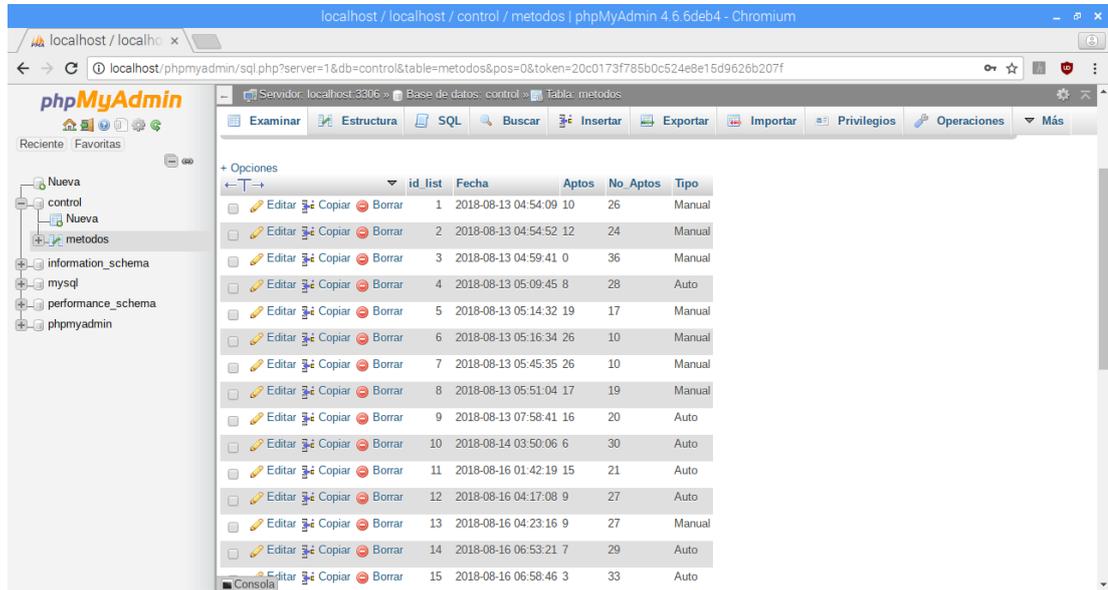


Ilustración 7. Análisis erróneo por falta de colocación de huevos de gallina en la cubeta del prototipo. [Investigador]

En la ilustración 8 se presenta la base de datos con información de varios análisis realizados.



	id_list	Fecha	Aptos	No. Aptos	Tipo
<input type="checkbox"/>	1	2018-08-13 04:54:09	10	26	Manual
<input type="checkbox"/>	2	2018-08-13 04:54:52	12	24	Manual
<input type="checkbox"/>	3	2018-08-13 04:59:41	0	36	Manual
<input type="checkbox"/>	4	2018-08-13 05:09:45	8	28	Auto
<input type="checkbox"/>	5	2018-08-13 05:14:32	19	17	Manual
<input type="checkbox"/>	6	2018-08-13 05:16:34	26	10	Manual
<input type="checkbox"/>	7	2018-08-13 05:45:35	26	10	Manual
<input type="checkbox"/>	8	2018-08-13 05:51:04	17	19	Manual
<input type="checkbox"/>	9	2018-08-13 07:58:41	16	20	Auto
<input type="checkbox"/>	10	2018-08-14 03:50:06	6	30	Auto
<input type="checkbox"/>	11	2018-08-16 01:42:19	15	21	Auto
<input type="checkbox"/>	12	2018-08-16 04:17:08	9	27	Auto
<input type="checkbox"/>	13	2018-08-16 04:23:16	9	27	Manual
<input type="checkbox"/>	14	2018-08-16 06:53:21	7	29	Auto
<input type="checkbox"/>	15	2018-08-16 06:58:46	3	33	Auto

Ilustración 8. Funcionamiento del prototipo. [Investigador]

En la ilustración 9 se presenta el prototipo en funcionamiento.

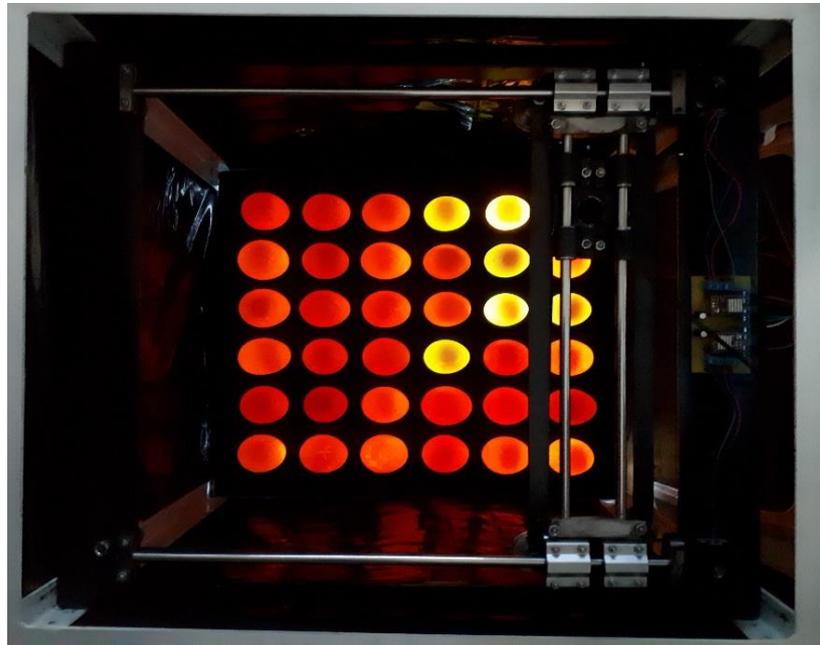


Ilustración 9. Funcionamiento del prototipo. [Investigador]