



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

**ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES**

**“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO
CLASIFICADOR DE HUEVO DE GALLINA BASADO EN LAS
IMPERFECCIONES DE LA CÁSCARA APLICANDO VISIÓN
ARTIFICIAL”**

TRABAJO DE TITULACIÓN: DISPOSITIVO TECNOLÓGICO

Para optar al Grado Académico de:

**INGENIERO EN ELECTRÓNICA CONTROL Y REDES
INDUSTRIALES**

AUTORES: DAVID GEOVANNY PAGUAY DONOSO

LUIS XAVIER VALAREZO RODRÍGUEZ

TUTOR: MSC. JAVIER JOSÉ GAVILANES CARRION

Riobamba – Ecuador

2018

©2018, David Geovanny Paguay Donoso, Luis Xavier Valarezo Rodríguez

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

El Tribunal del Trabajo de Titulación certifica que: El Dispositivo Tecnológico: “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO CLASIFICADOR DE HUEVO DE GALLINA BASADO EN LAS IMPERFECCIONES DE LA CÁSCARA APLICANDO VISIÓN ARTIFICIAL”, de responsabilidad de los Señores David Geovanny Paguay Donoso y Luis Xavier Valarezo Rodríguez, ha sido minuciosamente revisado por los Miembros del Tribunal de Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE	FIRMA	FECHA
Dr. Julio Santillán. VICEDECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA	_____	_____
Ing. Freddy Chávez V. DIRECTOR DE LA ESCUELA DE INGENIERÍA EN ELECTRÓNICA CONTROL Y REDES INDUSTRIALES	_____	_____
Ing. Javier Gavilanes C. DIRECTOR DEL TRABAJO DE TITULACIÓN	_____	_____
Ing. Edwin Altamirano S. MIEMBRO DEL TRIBUNAL	_____	_____

“Nosotros **DAVID GEOVANNY PAGUAY DONOSO** y **LUIS JAVIER VALAREZO RODRÍGUEZ**, somos responsables de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual de la misma pertenecen a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO”

David Geovanny Paguay Donoso

Luis Xavier Valarezo Rodríguez

DEDICATORIA

Es mi mayor orgullo dedicar el presente trabajo de titulación primeramente a Dios por darme la fuerza y la perseverancia para poder culminar mis estudios, a mi madre del cielo la Virgen Dolorosa que desde el colegio me supo dar su bendición y protección, a mi madrecita Fanny que con su amor, cariño y apoyo me supo guiar durante todo el transcurso de mi vida por eso y muchas cosas más siempre estaré agradecido con ella, a mi papá José por sus consejos y ser un ejemplo guiándome por el camino correcto, a mi hermana Katherine por todos los momentos que vivimos juntos ayudándome cuando más la necesitaba, a mi pareja de la vida Karen por su amor y comprensión que supo darme en el transcurso de mi carrera y haberme dado el mejor regalo que fue mi hija Julieth que es la luz de mis ojos y la que me da fuerza para seguir superándome que desde el día que nació alegre la vida de todos, mis tíos Fabiola y Pedro por haberme cuidado de pequeño y ser como mis segundos padres, mis primos Fernando, Paola, Diego por haber crecido junto a mi como hermanos y como una madre más, mis tíos y primos que la vida me dio al igual que mis amigos por aquellos momentos compartidos de alegría, y a las personas que han partido hacia un lugar mejor abuelitos ; este logro es para todos ustedes

David Paguay

De manera muy especial a mis padres Herlinda y Fernando por enseñarme sus valores de respeto, humildad, sencillez y por brindarme su apoyo incondicional para la culminación de esta meta. Y a mis amigos que han estado conmigo a lo largo de mi carrera, juntos hemos avanzado en el cumplimiento de este objetivo.

Luis Valarezo

AGRADECIMIENTO

Agradecemos a Dios por habernos dado la vida y permitirnos culminar sin ningún problema nuestra vida estudiantil, bendiciéndonos en cada momento y habernos puesto en el camino con personas que nos apoyaron incondicionalmente, a nuestros padres quienes son la razón de este triunfo ya que nos apoyaron en toda nuestra vida universitaria.

El más sincero agradecimiento a la Escuela Superior Politécnica de Chimborazo por brindarnos la oportunidad de obtener una profesión para servicio de la sociedad y el país.

Agradecemos al Msc. Javier Gavilanes y al Ing. Edwin Altamirano, Director y miembro del trabajo de titulación respectivamente por permitirnos recurrir a su capacidad y experiencia para encaminar con éxito el presente trabajo.

A nuestro amigo Luis que junto a su familia supieron apoyarnos cuando más los necesitamos abriéndonos las puertas de su casa, por esa y muchas cosas más gracias amigo siempre estaremos agradecidos toda una vida.

A todos nuestros amigos y conocidos que supieron brindarnos su apoyo y aliento de motivación para no decaer en esta etapa de la vida.

DAVID Y LUIS

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	x
INDICE DE FIGURAS.....	xi
ÍNDICE DE GRÁFICOS.....	xiv
ÍNDICE DE ECUACIONES	xv
ÍNDICE DE ANEXOS	xvi
ÍNDICE DE ABREVIATURAS.....	xvii
RESUMEN	xviii
ABSTRACT.....	xix
INTRODUCCIÓN	1
CAPITULO I	
1. MARCO TEÓRICO REFERENCIAL	6
1.1 El Huevo.....	6
<i>1.1.1 Estructura del huevo</i>	<i>6</i>
<i>1.1.2 Clasificación del huevo de gallina</i>	<i>7</i>
<i>1.1.3 Defectos en la cascara del huevo</i>	<i>9</i>
<i>1.1.3.1 Grietas.....</i>	<i>9</i>
<i>1.1.3.2 Manchas de sangre.....</i>	<i>9</i>
<i>1.1.3.3 Manchas de excremento</i>	<i>10</i>
<i>1.1.3.4 Corrugaciones.....</i>	<i>10</i>
<i>1.2.1 Clasificadora de huevos Moba</i>	<i>11</i>
<i>1.2.2 Clasificadora de huevos Yemita</i>	<i>11</i>
<i>1.2.2.1 Características Técnicas de la Clasificadora de Huevos Yemita.....</i>	<i>12</i>
<i>1.2.3 Clasificadoras lineales.....</i>	<i>12</i>
1.3 Inteligencia artificial	13
<i>1.3.1 Introducción a la inteligencia artificial</i>	<i>13</i>
1.4 Visión artificial	13
<i>1.4.1 Definición de visión artificial.....</i>	<i>13</i>
<i>1.4.2 Objetivo de la visión artificial.....</i>	<i>14</i>
<i>1.4.3 Componentes de un sistema de visión artificial</i>	<i>14</i>

1.4.3.1	<i>Dispositivo de captura de imágenes</i>	14
1.4.3.2	<i>Sistema de procesamiento</i>	17
1.4.3.3	<i>Software</i>	17
1.4.4	<i>Adquisición de imágenes</i>	17
1.4.5	<i>Procesamiento de imágenes</i>	18
1.4.5.1	<i>Transformación RGB a escala de grises</i>	18
1.4.5.2	<i>Contornos</i>	19
1.4.5.3	<i>Harris corner detection</i>	19
1.4.6	<i>Dilatación</i>	20
1.5	Software empleado en el sistema	21
1.5.1	<i>SolidWorks</i>	21
1.5.2	<i>Adams</i>	22
1.5.3	<i>Arduino IDE</i>	23
1.5.4	<i>Phyton</i>	23
1.6	Raspberry PI	24
1.6.1	<i>Hardware Raspberry PI3</i>	24
1.7	Dispositivos y componentes del sistema	25
1.7.1	<i>Sensores</i>	25
1.7.1.1	<i>Sensor Qrd</i>	25
1.7.1.2	<i>Galga extensiométrica</i>	26
1.7.2	<i>Fuente de alimentación</i>	27
1.7.3	<i>Actuadores</i>	27
1.7.3.1	<i>Servomotores</i>	27
1.7.3.2	<i>Motor de pasos</i>	28

CAPITULO II

2.	MARCO METODOLÓGICO	29
2.1	Descripción del prototipo clasificador de huevos de gallina	30
2.2	Selección de elementos y dispositivos del prototipo	31
2.3	Diseño e implementación del sistema mecánico	31
2.4.1	<i>Descripción de la estructura del prototipo</i>	32
2.4.1.1	<i>Diseño de la rampa de captación de huevos</i>	33
2.4.1.2	<i>Diseño de la estructura para la captura de imágenes</i>	33
2.4.1.3	<i>Diseño del mecanismo para el traslado de huevos</i>	34
2.4.1.4	<i>Diseño del sistema de pesaje y separación</i>	35
2.4.2	<i>Implementación de los mecanismos del prototipo</i>	36
2.4	Diseño e implementación del sistema electrónico	38

2.5.1	<i>Selección de componentes para la placa electrónica</i>	39
2.5.1.1	<i>Galga extensiométrica de 5 Kg</i>	39
2.5.1.2	<i>Módulo HX-711 para galga extensiométrica</i>	39
2.5.1.3	<i>Qrd</i>	40
2.5.1.4	<i>Driver A4988</i>	40
2.5.2	<i>Elección del controlador para el prototipo</i>	41
2.5.3	<i>Esquema de conexión</i>	42
2.5.4	<i>Diseño de la placa PCB</i>	43
2.5.5	<i>Implementación de la placa electrónica</i>	43
2.5	Desarrollo e implementación del algoritmo de visión artificial	44
2.6.1	<i>Obtención de imágenes</i>	45
2.6.1.1	<i>Selección del dispositivo de captura</i>	45
2.6.2	<i>Captura de imagen</i>	47
2.6.3	<i>Pre-procesamiento</i>	48
2.6.4	<i>Umbralización</i>	49
2.6.5	<i>Detección de contornos</i>	51
2.6.6	<i>Creación de una máscara en una imagen de escala de grises</i>	52
2.6.6.1	<i>Técnicas estadísticas en imagen de escala de grises</i>	53
2.6.7	<i>Cálculo de gradiente mediante el operador Sabel</i>	56
2.6.8	<i>Detección de rupturas en la superficie de la cáscara del huevo</i>	59
2.6.9	<i>Detección de manchas en la superficie de la cáscara</i>	60
2.6	Programación del algoritmo de comparación de pesos	62
2.7	Creación de la base de datos	63
2.8.1	<i>Creación de la tabla asignando campos necesarios</i>	63
2.8.2	<i>Inserción de datos en los campos asignados</i>	64
2.8.3	<i>Visualización de datos</i>	65
2.8	Diseño e implementación de la interfaz gráfica	66

CAPITULO III

3.	MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS	69
3.1	Análisis y comparación de los resultados obtenidos del prototipo	69
3.2.1	<i>Efectividad del algoritmo de visión artificial</i>	71
3.2.2	<i>Respuesta del algoritmo de visión</i>	72
3.2.3	<i>Comparación del tiempo empleado en forma manual vs el prototipo</i>	74
3.2.4	<i>Análisis de costos del prototipo implementado</i>	74

CONCLUSIONES.....	76
RECOMENDACIONES.....	77
BIBLIOGRAFÍA	
ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-1: Tabla de clasificación de tamaños de huevos según su peso	8
Tabla 2-1: Tabla de clasificación de huevos según su grado	8
Tabla 3-1: Tabla de comparación clasificadoras lineales	13
Tabla 4-1: Características de los servos según su modelo	28
Tabla 5-2: Elementos y dispositivos que componen el prototipo	31
Tabla 1-3: Datos de las muestras	70
Tabla 2-3: Datos obtenidos de la clasificación de forma manual.....	70
Tabla 3-3: Datos obtenidos de la clasificación con el prototipo	71
Tabla 4-3: Efectividad del algoritmo de visión.....	71
Tabla 5-3: Resultados obtenidos en rupturas y manchas de la superficie.....	72
Tabla 6-3: Prueba de ruptura en la superficie que no puede captar la cámara	73
Tabla 7-3: Tiempo manual vs tiempo del prototipo.....	74
Tabla 8-2: Costos de la implementación del sistema	74

INDICE DE FIGURAS

Figura 1-1: Huevo de gallina	6
Figura 2-1: Estructura del huevo.....	7
Figura 3-1: Huevo con fisura.....	9
Figura 4-1: Huevo con mancha de sangre	9
Figura 5-1: Huevos con heces	10
Figura 6-1: Huevo corrugado.....	10
Figura 7-1: Clasificadora Moba	11
Figura 8-1: Carriles Clasificadora Yemita	12
Figura 9-1 Clasificadora Yemita	12
Figura 10-1: Espectro visible.....	15
Figura 11-1: Lentes de una cámara	15
Figura 12-1: Sensor fotográfico.....	16
Figura 13-1: Componentes de una cámara digital.....	16
Figura 14-1: Librería OpenCV	17
Figura 15-1: Descomposición de colores.....	18
Figura 16-1: Matrices que conforman una imagen a color	18
Figura 17-1: Esquina con precisión subpixel	20
Figura 18-1: Efecto de dilatación	21
Figura 19-1: Entorno gráfico de SolidWorks.....	22
Figura 20-1: Entorno de Adams.....	22
Figura 21-1: Entorno Arduino IDE	23
Figura 22-1: Entorno de Python	24
Figura 23-1: Raspberry PI	24
Figura 24-1: Partes de la Raspberry pi.....	25
Figura 25-1: Sensor qrd.....	25
Figura 26-1: Galga de peso	26
Figura 27-1: HX-711	26
Figura 28-1: Fuente de alimentación	27
Figura 29-1: Servomotor	27
Figura 30-1: Motor de pasos Nema 23.....	28
Figura 1-2: Estructura del prototipo diseñado en SolidWorks.....	32
Figura 2-2: Diseño de la rampa de captación de huevos	33
Figura 3-2: Diseño de la estructura de captura de imagen	34
Figura 4-2: Diseño de la biela empleada en el prototipo	34

Figura 5-2: Diseño del sistema rotacional y bandeja de transporte	35
Figura 6-2: Diseño del sistema de pesaje y separación del prototipo	35
Figura 7-2: Diseño del eslabón	36
Figura 8-2: Rampa de captación de huevos.....	36
Figura 9-2: Estructura para captura de imágenes.....	37
Figura 10-2: Mecanismo de transporte de huevos	37
Figura 11-2: Sistema de pesaje.....	37
Figura 12-2: Estructura mecánica del prototipo.....	38
Figura 13-2: Estructura pintada del prototipo clasificador	38
Figura 14-2: Galga extensiométrica 5 Kg	39
Figura 15-2: Módulo HX-711	40
Figura 16-2: Sensor QRD	40
Figura 17-2: Driver A4988	40
Figura 18-2: Minicomputador Raspberry PI3	41
Figura 19-2: Arduino Pro Micro.....	41
Figura 20-2: Esquema de conexión del sistema electrónico	42
Figura 21-2: Placa PCB del prototipo.....	43
Figura 22-2: Placa PCB implementada	43
Figura 23-2: Cámara web USB	45
Figura 24-2: Cámara para Raspberry PI 5MP.....	46
Figura 25-2: Ángulo de visión en función de la distancia focal.....	46
Figura 26-2: a) Cámara Genius F100, b) Cámara Raspberry PI	47
Figura 27-2: Librerías OpenCV importadas.....	47
Figura 28-2: Comandos para realizar la captura de imagen	48
Figura 29-2: Separación de la imagen en componentes primarias	48
Figura 30-2: Imagen del huevo en componentes primarias RGB	49
Figura 31-2: Técnicas de umbralización adaptativa.....	50
Figura 32-2: Imagen del huevo comando umbralizado de Otsu	50
Figura 33-2: Imagen convertida a contorno.....	51
Figura 34-2: Selección de la imagen del huevo con mayor contorno.....	52
Figura 35-2: Creación de la imagen a partir de una máscara, a) escala de grises b) filtrada	52
Figura 36-2: Comando para convertir una imagen a array	53
Figura 37-2: Umbralización en base a la media aritmética.....	54
Figura 38-2: Umbralización en base a la desviación estándar	55
Figura 39-2: Selección del canal con mayor coeficiente de variación.....	56
Figura 40-2: Obtención de la gradiente en x, y	57

Figura 41-2: Obtención de la imagen con detección de bordes	57
Figura 42-2: Sección para encontrar una imagen mediante método de Sobel.....	58
Figura 43-2: a) gradiente en x b) gradiente en y c) gradiente total	59
Figura 44-2: Comandos para procesar la imagen obtenida del gradiente	59
Figura 45-2: a) gradiente umbralizada, b) detección de contorno dilatada, c) imagen resultante	60
Figura 46-2: Rango de color establecido en base a su vector y color RGB	60
Figura 47-2: Resultados a) original b) tonalidad media oscura c) tonalidad oscuras	61
Figura 48-2: Comandos para mostrar las imágenes resultantes del procesamiento	61
Figura 49-2: Importación de las librerías para la base de datos.....	63
Figura 50-2: Creación de la tabla para asignar los campos	64
Figura 51-2: Verificación de la creación de la tabla	64
Figura 52-2: Almacenamiento de los datos.....	65
Figura 53-2: Comando para la visualización de los datos	66
Figura 54-2: Visualización de los datos almacenados en la tabla	66
Figura 55-2: Importación de la librería para la interfaz gráfica	67
Figura 56-2: Asignación de librerías.....	67
Figura 57-2: Asignación de texto en la ventana	68
Figura 58-2: Interfaz gráfica del prototipo	68

ÍNDICE DE GRÁFICOS

Gráfico 1-2: Etapas del diseño e implementación del prototipo	29
Gráfico 2-2: Descripción del proceso del prototipo	30
Gráfico 3-2: Etapas de diseño e implementación del sistema mecánico	32
Gráfico 4-2: Etapas para el diseño e implementación del sistema electrónico.....	39
Gráfico 5-2: Diagrama de las etapas del algoritmo de visión artificial	44
Gráfico 6-2: Etapas del algoritmo de comparación de pesos	62
Gráfico 7-2: Proceso para la creación de la base de datos	63
Gráfico 1-3: Efectividad media del algoritmo de visión	71

ÍNDICE DE ECUACIONES

Ecuación 1-2: Ecuación para calcular la media estadística.....	53
Ecuación 2-2: Ecuación para el cálculo de la desviación estándar	54
Ecuación 3-2: Ecuación para calcular la variación estadística	55
Ecuación 4-2: Cálculo del operador Sabel.....	56
Ecuación 5-2: Cálculo de la gradiente	58
Ecuación 1-3: Ecuación para calcular el tamaño muestral	69

ÍNDICE DE ANEXOS

Anexo A: Planos de conexión

Anexo B: Planos del prototipo

Anexo C: Planos del transporte

Anexo D: Planos de la rampa

Anexo E: Planos base motores

Anexo F: Planos de la biela

Anexo G: Diagrama de flujo del funcionamiento

Anexo H: Manual de usuario

ÍNDICE DE ABREVIATURAS

A/D:	Conversión Análogo Digital
ADC:	Attack Damage Carry
CMOS:	Complementary Metal Oxide Semiconductor
DSP:	Dispositivo Procesador de Señal Digital
GPIO:	General Purpose Input/Output
HDMI:	Interfaz Multimedia de Alta Definición
HMI:	Human Machine Interface
HSV:	Hue Saturation Value
IA:	Inteligencia Artificial
IDE:	Integrated Development Environment
INEN:	Instituto Ecuatoriano de Normalización
LCD:	Liquid Cristal Display
LED:	Light Emitting Diode
MDF:	Medium Density Fibreboard
MP:	Megapixel
NTE:	Normas Técnicas
OPENCV:	Open Source Computer Vision
PCB:	Printed Circuit Board
PLA:	Ácido Poliláctico
RAM:	Random Access Memory
RGB:	Red, Green y Blue
SD:	Secure Digital
SQL:	Structured Query Language
USB:	Universal Serial Bus

RESUMEN

Se diseñó e implementó un prototipo clasificador de huevo de gallina basado en las imperfecciones de la cáscara aplicando visión artificial. El prototipo centra su funcionamiento en un minicomputador Raspberry Pi3, que conectado a una cámara web Genius F100 adquiere imágenes de las diversas caras del huevo aplicando la técnica de ovoscopia junto con un arreglo de espejos, garantizando la cobertura total del área. Las imágenes adquiridas son procesadas dentro de la Raspberry mediante la implementación de un algoritmo escrito en el lenguaje de programación Python haciendo uso de las librerías de visión que ofrece OpenCV. Se emplean métodos estadísticos y de visión artificial para detectar las anomalías presentes en la superficie de la cáscara del huevo. Culminado con el procesamiento los huevos son transportados a través de una serie de mecanismos, donde serán clasificados en diversos carriles con ayuda de actuadores servomotores, los huevos que presenten anomalías serán los primeros en ser separados dando paso a que los huevos que no las presenten haciendo uso de galgas extensiométricas respectivamente acondicionadas, serán pesados y clasificados de acuerdo a su tamaño, mediante un algoritmo de comparación de peso programado en la tarjeta Arduino cumpliendo con lo establecido en las normas INEN. Los registros de control y clasificación son almacenados en una base de datos, mismos que pueden ser visualizados mediante un interfaz hombre máquina (HMI). A partir de las pruebas realizadas con el prototipo, se obtuvo como resultado un ahorro de tiempo aproximado del 60% con respecto al modo de clasificación manual. Se concluye que la implementación del prototipo mejora la calidad del producto y permite una mayor eficiencia del proceso llevando un control de producción diario en una base de datos almacenados por el sistema. Se recomienda seguir las instrucciones de los autores para garantizar el correcto funcionamiento del prototipo.

PALABRAS CLAVE: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <TECNOLOGÍA DEL CONTROL AUTOMÁTICO>, <VISIÓN ARTIFICIAL>, <TÉCNICA DE OVOSCOPIA>, <AUTOMATIZACIÓN DE PROCESOS>, <PROCESOS AVÍCOLAS>, <CALIDAD DE PRODUCTOS>.

ABSTRACT

A hen egg classifier prototype was designed and implemented based on the imperfections of the shell applying artificial vision. The prototype focuses its operation on a Minicomputer raspberry Pi3, which connected to a webcam Genius F100 acquires images of the various faces of the egg applying the technique of Candling along with an array of mirrors, ensuring the full coverage of Area. The acquired images are processed within the raspberry by implementing an algorithm written in the Python programming language using the Vision libraries offered by OpenCV. Statistical methods and artificial vision are used to detect the anomalies present on the surface of the egg shell. Culminate with the processing the eggs are transported through a series of mechanisms, where they will be classified in different lanes with the help of actuators servomotors, the egg that present anomalies will be the first to be separated giving way to that the eggs that do not present them making use of extensimetric gauges respectively conditioned, will be weighed and classified according to their size, by means of an algorithm of comparison of weight programmed in the Arduino card complying with the established in INEN standards. The control and classification logs are stored in a database, which can be viewed using a man-machine interface (HMI). From the tests carried out with the prototype, it was obtained as a result a time saving of approximately 60% with respect to the manual classification mode. It is concluded that the implementation of the prototype improves the quality of the product and allows a greater efficiency of the process carrying a daily production control in a database stored by the system. It is recommended to follow the instructions of the authors to guarantee the correct functioning of the prototype.

KEYWORDS: <TECHNOLOGY AND ENGINEERING SCIENCE>, <AUTOMATIC CONTROL TECHNOLOGY>, <ARTIFICIAL VISION>, <CANDLING TECHNIQUE>, <PROCESS AUTOMATION>, <AVICOLAS PROCESS>, <PRODUCT QUALITY>

INTRODUCCIÓN

ANTECEDENTES

Durante los años comprendidos entre 2000 y 2014 la producción mundial de huevos ha incrementado un 36.5%, un promedio de 2.8% por año. En 2014 se produjo casi 1320 billones de huevos en todo el mundo, equivalentes a unos 70 millones de toneladas. La producción mundial de huevos probablemente alcance los 100 millones de toneladas métricas por año en 2035. (Seleccionesavícolas, 2016)

El 20% de todos los huevos se producen en América. En el año 2010 solo cinco países de América: EUA, México, Brasil, Colombia y Argentina produjeron unos 10.8 millones de toneladas de huevos un 84% del total regional. (Elsitioavícola, 2013)

En Ecuador el año 2016, el consumo per cápita de huevos de gallina registro entre 160 y 165 unidades anuales, un valor bajo en relación al promedio de consumo en otros países de América Latina. La producción fue de 200 millones de cubetas de 30 unidades cada una en dicho año. Se estima que se obtiene una producción de alrededor de 7.3 millones de huevos del campo y de galpones aproximadamente 40 millones. (El telégrafo, 2017)

De la cantidad total de huevos de gallina producidos, un gran porcentaje se destina a la comercialización, siendo necesario que estos pasen por un proceso de clasificación ya que no todos cumplen con los requisitos para su venta. La clasificación de los huevos en las medianas y grandes empresas se realiza mediante el empleo de máquinas que los separa de acuerdo a su tamaño, mientras que en granjas se realiza este proceso aun de forma manual.

Las máquinas empleadas para la tarea de clasificación no son capaces de detectar fallos en los huevos. En países industrializados como Estados Unidos y China donde se centra la mayor producción del producto, y gracias al avance de la tecnología en el campo de visión artificial se ha iniciado con el desarrollo de máquinas que implementan técnicas y algoritmos de visión que permiten detectar huevos con anomalías y separarlos incrementando la calidad del producto.

Buscando emplear sistemas de visión para la detección de anomalías en huevos el Departamento de Ingeniería Agraria, de Università degli Studi di Milano, realizó una investigación para desarrollar un sistema de visión artificial que fuera capaz de detectar defectos en los huevos de gallina, en el estudio se observó que el procedimiento de clasificación era muy eficiente, ya que

el algoritmo fue capaz de clasificar correctamente cerca del 98 por ciento de las muestras. (LUNADEI, y otros, 2011)

Los autores Daniel Fernando Velazco Cáceres y María Luisa Pinto Salamanca del grupo de Investigación de Energía y Nuevas Tecnologías GENTE. Universidad Pedagógica y Tecnológica de Colombia. Determinaron las características del comportamiento mecánico del huevo, y pudieron establecer que existe una relación directamente proporcional entre la clasificación por peso y el tamaño del huevo.

En la Escuela Superior Politécnica de Chimborazo, mediante la investigación realizada se pudo constatar la realización de un módulo didáctico para la clasificación, empaclado e inspección de huevos. Haciendo uso de un PLC este trabajo tiene como objetivo trabajar de una forma didáctica y para aprendizaje en el modelamiento de automatización para la facultad de electrónica. (GALLEGOS LÓPEZ, y otros, 2012)

Por lo tanto, el presente trabajo de titulación presenta un prototipo clasificador de huevo de gallina basado en las imperfecciones de la cascara aplicando visión artificial, con la finalidad de brindar una nueva manera de clasificación en lugar de la forma manual, mediante un prototipo que permita además de una clasificación por pesos, la capacidad de separar los huevos malos al implementar un algoritmo de visión, así como llevar un registro en una base de datos de la producción que podrán ser visualizados mediante una interfaz gráfica.

FORMULACIÓN DEL PROBLEMA

¿Cómo ayudará la creación de un prototipo para la clasificación de huevos de gallina basado en las imperfecciones de la cáscara para evitar empaclar huevos en mal estado que afecten la salud de las personas?

SISTEMATIZACIÓN DEL PROBLEMA

¿Qué estudios se han realizado sobre los tipos de clasificadores de huevos de gallina y que métodos han ocupado para solucionar el problema?

¿Cuáles son las diferentes variables que intervienen dentro del proceso de clasificación del huevo de gallina?

¿Qué técnicas de visión artificial serán las más adecuadas para el control de calidad según las características del huevo de gallina que presenten en la superficie de la cáscara?

¿Qué sistema se podrá emplear para el manejo y traslado del huevo de gallina para evitar el daño de los mismos teniendo en cuenta el control de calidad para la clasificación del producto?

¿De qué forma se realizará una base de datos para llevar un control estadístico de la producción de huevos?

¿Cuál será el grado de aceptación que tendrá el sistema, de acuerdo a los resultados obtenidos?

JUSTIFICACIÓN TEÓRICA

La creciente producción de huevos de gallina, debido a un incremento en el consumo del mismo ha provocado que, desde los pequeños productores en galpones hasta las grandes empresas productoras de huevos intensifiquen la tarea de clasificación y control en la calidad de su producto para su posterior comercialización. Así las máquinas clasificadoras se presentan como una opción que permite facilitar este proceso, actualmente en el mercado existe una amplia gama de este tipo de máquinas cuyo objetivo es la clasificación de acuerdo al tamaño del huevo.

Al desarrollar una clasificadora de huevos de gallina basada en las imperfecciones de la cáscara, se busca como finalidad presentar un prototipo de bajo costo, que mediante el empleo de técnicas de visión artificial puede detectar las imperfecciones en la superficie de la cascara como son: roturas, manchas de sangre o desechos, para su posterior separación dando lugar a la clasificación de acuerdo al peso de los huevos que se encuentren en condiciones normales, siendo esta una ventaja sobre las clasificadoras presentes en el mercado.

El prototipo busca solventar una necesidad presente en los pequeños hasta los grandes productores de huevos de gallina, como es el control en la calidad y registro de cantidad de producción. Mediante la implementación de un sistema funcional que se presente como una opción económicamente rentable en comparación con las clasificadoras existentes, ya que incorpora la separación de huevos con fallas, la clasificación de acuerdo a su peso y el registro de producción en una base de datos.

La realización de este trabajo será de gran beneficio para poder tener un control de higiene y calidad de los huevos que son llevados al consumidor y cuidar de mejor manera la proliferación de enfermedades causadas por la mala manipulación de los huevos de granja, que en ocasiones son puestos en las cubetas junto con huevos rotos o que tienen desechos fecales de gallinas que pueden traer enfermedades como la salmonela.

JUSTIFICACIÓN APLICATIVA

Con la implementación del prototipo clasificador de huevos de gallina se ayudará a verificar el estado y la calidad del producto que será llevado al consumidor. El proceso de detección de fallos en la cáscara, estará basado en una cámara de buena resolución empleada como sensor que capturará y analizará las imágenes del huevo, mediante un procesamiento y la aplicación de un algoritmo de visión se detectan los huevos defectuosos como son: cascara rota, huevos con sangre o heces, deformidad en la cascara.

Una vez detectado los huevos malos un sistema mecánico y electrónico se encargará del transporte y clasificación de los mismos acordes a su peso mediante el uso de galgas extensiométricas. Servomotores empleados como actuadores serán encargados de empujar a los huevos hacia carriles separados.

Se llevará a cabo una base de datos con la información obtenida de la cantidad y calidad de los huevos, esto ayudará a medir la producción de la productora avícola para tener un registro diario de la cantidad de huevos que se va obteniendo y dar un resultado factible al productor donde podrá verificar el estado en el que se encuentra la producción.

OBJETIVOS

OBJETIVO GENERAL

Diseñar e implementar un prototipo clasificador de huevo de gallina basado en las imperfecciones de la cáscara aplicando visión artificial

OBJETIVOS ESPECÍFICOS

- Realizar el estudio del arte sobre los distintos tipos de clasificadores y las estrategias que ocupan.
- Analizar las posibles soluciones, métodos y estrategias que podrían ser aplicables a la solución del problema y elegir el mejor método.
- Programar un algoritmo de comparación para el procesamiento de control de calidad para la clasificación del huevo de gallina según su volumen e imperfecciones en la cáscara.
- Realizar una base de datos con la información obtenida de la clasificación y control de calidad para llevar un registro de la producción.
- Diseñar e implementar los sistemas: mecánicos, eléctricos, electrónicos y de control para la clasificación del huevo de gallina.
- Comprobar el funcionamiento y verificar los resultados obtenidos.

CAPITULO I

1. MARCO TEÓRICO REFERENCIAL

En este capítulo se realiza una descripción del huevo de gallina y las diferentes anomalías que pueden existir en la superficie de su cáscara. Se observa el procedimiento empleado para la clasificación de forma manual que pueden ser empleados por visión artificial y por el sistema de comparación de pesos para separar los huevos por su tamaño.

Se presenta información para la realización del prototipo clasificador de huevo de gallina basado en las imperfecciones de la cáscara aplicando visión artificial, así como los componentes necesarios tanto software como hardware para desarrollar el presente trabajo de titulación.

1.1 El Huevo

El huevo es uno de los productos de origen animal que posee un gran valor nutricional es empleado para un sin número de preparaciones por su fácil consumo, figura 1-1. todos estos compuestos son propios para el desarrollo del embrión y por tal motivo dispone de estas proteínas, su estructura permite proteger la yema y clara que dará origen a una nueva gallina.



Figura 1-1: Huevo de gallina

Realizado por: Paguay, David; Valarezo, Luis; 2018

1.1.1 Estructura del huevo

La estructura del huevo está compuesta principalmente de tres partes que son: la clara, la yema y la cáscara como se puede ver en la figura 2-1, cada una tiene una función específica que permitirá el desarrollo del embrión. La estructura del huevo es por la naturaleza para brindar

protección de microorganismos que afecten su valor nutricional, esta línea de defensa la componen la cáscara y las membranas que mantienen su integridad. (Huevos el majadal, 2012)

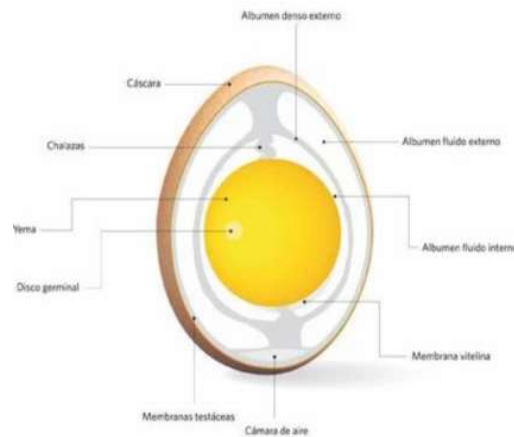


Figura 2-1: Estructura del huevo

Fuente: <https://goo.gl/TeXnSc>

Las partes principales que constituyen la estructura del huevo son:

- **Yema:** está formada de grasas, minerales y proteínas, contiene la mayor parte de nutrientes.
- **Clara:** es la parte transparente que rodea la yema, está constituida principalmente en su mayoría por agua y brinda proteínas que no son grasas.
- **Cáscara:** brinda seguridad a los nutrientes internos de los huevos, evita contaminación de enfermedades que puedan afectar al consumidor, su parte externa es mucho más dura que la interna permitiendo salir al pollo con facilidad.

1.1.2 Clasificación del huevo de gallina

Para la clasificación de los huevos de gallina, a nivel nacional las granjas avícolas se rigen a las normativas del artículo NTE INEN 1973 para los huevos comerciales y ovoproductos teniendo como requisitos las siguientes características en función de su masa (peso) unitario, en la cual se admite una tolerancia máxima de ± 1 gramo por huevo para todos los tamaños y tipos. (Normalización, 2015)

La tabla de clasificación de huevos acorde a su tamaño y en función de su peso, se puede observar en la tabla 1-1.

Tabla 1-1: Tabla de clasificación de tamaños de huevos según su peso

Tipo (Tamaño)	Masa unitaria		Masa por docena		Masa por 30 huevos		
	(g)		(g)		(g)		
	Mínimo (≥)	Máximo (<)	Mínimo (≥)	Máximo (<)	Mínimo (≥)	Máximo (<)	
I	Súper gigante	76	-	912	-	2280	-
II	Gigante	70	76	840	912	2100	2280
III	Extra grande	64	70	768	840	1920	2100
IV	Grande	58	64	696	768	1740	1920
V	Mediano	50	58	600	696	1500	1740
VI	Pequeño	46	50	552	600	1380	1500
VII	Inicial	-	46	-	552	-	1380

Fuente: <https://ia801603.us.archive.org/29/items/ec.nte.1973.2011/ec.nte.1973.2011.pdf>

Realizado por: Paguay, David; Valarezo, Luis; 2018

En base a su calidad, el huevo de gallina también se puede clasificar según la tabla 2-1, en esta tabla la tolerancia que se aceptara es del 10% entre el grado A y el grado B.

Tabla 2-1: Tabla de clasificación de huevos según su grado

Características	Grado A	Grado B
Cascarón y cutícula	Normal, intacta, limpia.	Normal e intacta, manchas mínimas no propias del producto.
Cámara de aire	Su altura no excederá de los 9mm, inmóvil.	Su altura no excederá de los 15mm, inmóvil.
Clara	Transparente, limpia de consistencia gelatinosa, exenta de cuerpos extraños.	Transparente, limpia de consistencia gelatinosa. Se admiten manchas de sangre y/O carne hasta 3mm.
Yema	Visible al trasluz, bajo de sombra solamente, sin contorno aparente, no separándose sensiblemente de la posición central en caso de rotación del huevo. Exenta de cuerpos extraños.	Visible al trasluz, bajo de sombra solamente; pequeña separación en caso de rotación del huevo. Se admiten manchas de sangre de hasta 3 mm.
Olor y sabor	Exento de olores y sabores extraños.	Exento de olores y sabores extraños.

Fuente: <https://ia801603.us.archive.org/29/items/ec.nte.1973.2011/ec.nte.1973.2011.pdf>

Realizado por: Paguay, David; Valarezo, Luis; 2018

1.1.3 Defectos en la cascara del huevo

La cáscara es una fuente de información donde se puede observar las diferentes anomalías que se presentan sobre su superficie, algunos de estos defectos de calidad pueden ser observados a simple vista y otros no, entre las principales alteraciones en la cáscara que podemos encontrar son:

1.1.3.1 Grietas

Las fisuras, figura 3-1, pueden ser ocasionadas por golpes leves o graves, puede haber grietas tan finas como un pelo y a veces son por causas naturales como calor excesivo, gallinas viejas, falta de calcio en la cáscara o infecciones micóticas. (veterinariadigital, 2015)

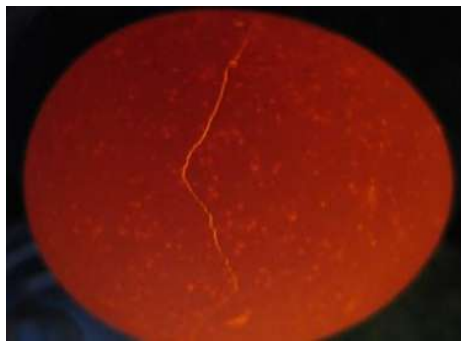


Figura 3-1: Huevo con fisura

Fuente: <https://goo.gl/pd38YV>

1.1.3.2 Manchas de sangre

Pueden ser ocasionadas por lesiones al expulsar el huevo, estos se pueden manchar por una cloaca prolapsada y también por gallinas que estén comenzando a poner huevos, figura 4-1.



Figura 4-1: Huevo con mancha de sangre

Fuente: <https://goo.gl/kk9gcc>

1.1.3.3 Manchas de excremento

Si el huevo presenta manchas de heces, se puede presentar por una alimentación incorrecta o por problemas intestinales que manchan el huevo al salir esto puede ocasionar salmonella, como se observa en la figura 5-1.



Figura 5-1: Huevos con heces

Fuente: <https://goo.gl/582L7k>

1.1.3.4 Corrugaciones

Los huevos corrugados pueden ser uno de los defectos de calidad que resulta más extraño, puede darse por enfermedades como Newcastle y bronquitis infecciosa, figura 6-1. (gallosdeguerra, 2016)



Figura 6-1: Huevo corrugado

Fuente: <https://goo.gl/UPaBpU>

1.2 Clasificadoras de huevo

Las clasificadoras más comerciales que se puede encontrar son las de tipo rotatorio y las lineales, cada una de estas tiene sus beneficios y sus defectos en cuanto al precio y los mecanismos de clasificación.

1.2.1 Clasificadora de huevos Moba

Moba basa la clasificación de los huevos con una manipulación individual de acuerdo a su peso y a la calidad, contando con clasificadoras automáticas con capacidad de (20.000 - 180.000) huevos/hora, en el proceso se retira los huevos de grado inferior también conocidos como “huevos industriales” que cumplen otros propósitos de acuerdo al estado o calidad de estos.

En la clasificadora de huevos Moba existe dos formas de retirar los huevos no deseados, la primera es manualmente en una cabina donde se examina su calidad o la segunda forma que es totalmente automática que cuenta con equipos especializados para su detección, posteriormente son pesados y transportados para ser empaquetados automáticamente en envases de consumo. (nieser, 2017)

Las contantes mejoras de algoritmos de clasificación para una buena eficiencia garantizando la trazabilidad exacta al mismo tiempo, figura 7-1.



Figura 7-1: Clasificadora Moba

Fuente: <https://goo.gl/sbqugy>

1.2.2 Clasificadora de huevos Yemita

La máquina más popular y aceptada por medianos y pequeños consumidores por ser eficiencia y fiabilidad es de tipo rotativa con cuatro sectores la cual selecciona los huevos de acuerdo al peso y cuenta con parámetros de clasificación modificables.

La clasificadora Yemita siempre tiene disponible el plato donde se depositan los huevos a pesar que siempre se mantiene llena las cucharas que los sostienen y cuenta con una capacidad de seleccionar (3.600 – 4.200) huevos/ hora, figura 8-1.



Figura 8-1: Carriles Clasificadora Yemita

Fuente: <https://goo.gl/q8C1hA>

1.2.2.1 Características Técnicas de la Clasificadora de Huevos Yemita

La máquina clasificadora funciona con un motor eléctrico de 220W a una velocidad de 1500RPM y una potencia de 1,5Hp.

Entre los materiales que se usa para fabricar esta clasificadora tenemos el aluminio, hierro, plástico inyectado y fibra de vidrio el plato donde se deposita los huevos es pintado con EXPOXI esto ayuda a que sea más resistente al desgaste. (clasificadoradehuevos, 2010), como se puede ver en la figura 9-1



Figura 9-1 Clasificadora Yemita

Fuente: <https://goo.gl/P1h48M>

1.2.3 Clasificadoras lineales

En la siguiente Tabla 3-1, podemos realizar una comparación de las clasificadoras lineales y la cantidad de huevos/hora que pueden clasificar

Tabla 3-1: Tabla de comparación clasificadoras lineales

MÁQUINA	CARACTERÍSTICAS	CANTIDAD (huevos/hora)
CLASIFICADORA DE HUEVOS STAALKAT	Serie: Alph Capacidad: Media Fabricante: INGENIERIA AVICOLA S.L Es una embaladora robusta y segura que es capaz de acoplarse a las altas exigencias de producción.	25000-45000
CLASIFICADORAS RECOGIDA MANUAL	Serie: Manual Capacidad: Media Fabricante: Avi Service Es moderna desde su diseño que cuenta con detalles y ajustes de acuerdo con la tecnología.	2220-15000
CLASIFICADORA DE HUEVOS MANUALES	Serie: Moba 68 Capacidad: media Fabricante: JM Alcantara S.L Categoría: 4-7 Cuenta con un espejo candling el cual permite hasta dos operarios al mismo tiempo.	2220-15000

Fuente: <http://avicultura.poultry.com/categorias/huevos/clasificadoras-de-huevos-de-pequena-capacidad>

Realizado por: Paguay, David; Valarezo, Luis; 2018

1.3 Inteligencia artificial

1.3.1 Introducción a la inteligencia artificial

Según Pedro Ponce Cruz “La IA es una rama de las ciencias computacionales que se ocupa de los símbolos y métodos no algorítmicos para la resolución de problemas.” En cierta parte este concepto puede darnos una idea de que la inteligencia artificial puede abarcar algunos campos de la rama computacional como puede ser visión artificial, en la que utiliza ciertas técnicas de la IA para realizar el procesamiento de imágenes digitales controladas por un computador. (PONCE CRUZ, 2010)

1.4 Visión artificial

1.4.1 Definición de visión artificial

Si se hiciera un análisis de cuál sería el órgano o sistema más complejo del cuerpo de humano obviamente el primero en la lista sería el cerebro, y el segundo sería la visión uno el órgano tan

complejo que tomo millones años para perfeccionarse y dar una percepción única del mundo, más que los otros sentidos, es fundamental en el aprendizaje, al distinguir colores y formas.

Se puede definir visión artificial como a un conjunto de técnicas y métodos fundamentados en principios matemáticos para emular el funcionamiento del sistema de visión que poseen los seres humanos. No solo en lo que es la adquisición de imágenes sino en el procesamiento e interpretación de las mismas. (elai, 2005)

La visión artificial puede aplicarse en todo proceso en donde se necesite la interpretación de información derivada de imágenes, por lo que se puede aplicar en el compa industrial, medicina, la agricultura, por lo que se puede decir que sus aplicaciones son muy variadas.

1.4.2 Objetivo de la visión artificial

Los objetivos que se plantea son abrir el espacio necesario para la incorporación de nuevas tecnologías como la inteligencia artificial dentro del campo industrial y en la robótica principalmente para tareas que para los humanos resulte muy peligrosa o muy difícil de realizar.

1.4.3 Componentes de un sistema de visión artificial

Un sistema de visión está compuesto por un dispositivo de captura de imagen es decir una cámara, la luminosidad como componente medio ambiental que influye en la tonalidad y el color de una imagen, un sistema de conversión que transforma las señales analógicas captadas por la cámara en datos que un computador pueda procesar. (elai, 2005)

1.4.3.1 Dispositivo de captura de imágenes

Es unos de los componentes más importantes por no decir el más fundamental, es el encargado de captar una parte del espectro de luz y todo aquello en lo que refleje llamado así también una imagen como se puede observar en la figura 10-1.

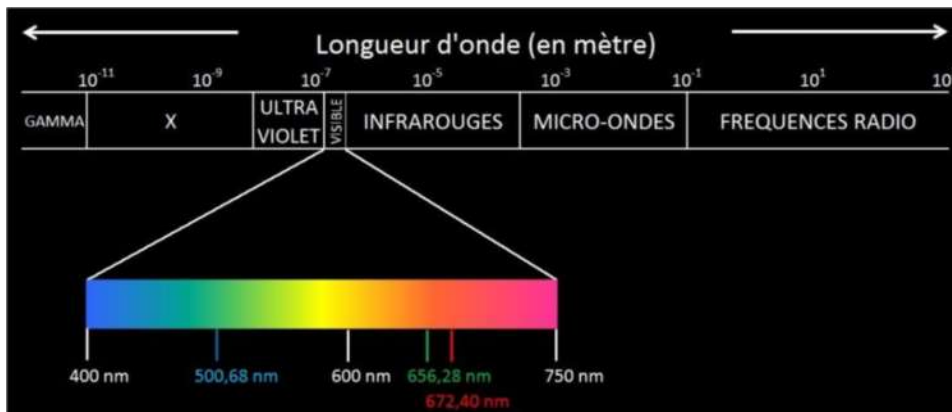


Figura 10-1: Espectro visible

Fuente: http://www.astrophysic.org/spectre_electromagnetique.jpg

En la actualidad existen una gran variedad de cámaras, pero lo que tiene en común todas es que son digitales pero el principio por el que se sustenta su funcionamiento se asemeja al de una cámara analógica y estas usaban carretes de películas de 35 mm, exceptuando que estas cámaras no tienen una pantalla LCD, lo que facilita a la persona que esté haciendo uso de la misma el enfoque, encuadre y la visualización. Además de esta diferencia los componentes internos entre una cámara digital y analógica son muy notables, ya las digitales poseen componentes electrónicos que hacen que una imagen sea procesada por un computador.

El componente más básico que conforman una cámara digital son:

- **El objetivo fotográfico:** está conformado por un conjunto de lentes con las que se pueden hacer una convergencia o divergencia para el enfoque de una imagen, estos están cubiertos en su superficie por una membrana anti reflejante de color azul, esto se realiza para que la luz entrante no se refleje en los lentes y provoquen una gran mancha de luz reflejada en la imagen resultante, figura 11-1. Existen dos tipos de objetivos fotográficos uno es fijo con una distancia focal ya definida o con un mecanismo que hace que esta distancia vaya variando también llamado zoom óptico.



Figura 11-1: Lentes de una cámara

Fuente: <https://www.decamaras.com/imagen/articulos/objetivos/canon24f14.jpg>

- **Sensor fotográfico:** es un dispositivo que esta acoplado en prácticamente todas las cámaras digitales y esto se debe a la función que desempeña ya que en su superficie son captados los fotones de la imagen cuando la luz pasa a través de los lentes convirtiendo en impulsos eléctricos derivando en una corriente de naturaleza alterna y todo lo hace un integrado por lo general tipo CMOS semiconductor y su superficie determina la calidad de la imagen obtenida como se observa en la figura 12-1. (fayerwayer, 2011)

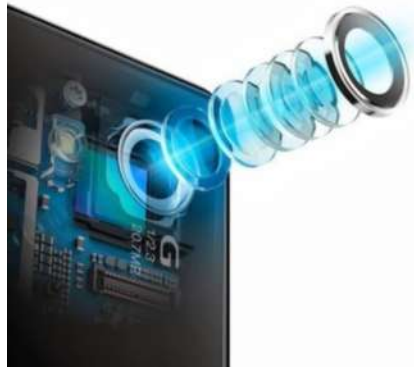


Figura 12-1: Sensor fotográfico

Fuente: <https://goo.gl/sxUbSB>

- **Dispositivo de conversión ADC:** convertidor analógica digital ya como se mencionó anteriormente las señales que produce son analógicas lo que es incompatible con las señales con las que funciona una computadora o cualquier componente electrónico. Lo que hace este dispositivo es transformar las señales analógicas en un código binario compuesto únicamente por unos y ceros.
- **Dispositivo procesador de Señal Digital (DSP):** su función es de regular y controlar todo el sistema electrónico de la cámara y también es capaz de procesar las imágenes para enviarlas a un dispositivo de almacenamiento también llamado memoria como se muestra en la figura 13-1. (asifunciona, 2008)



Figura 13-1: Componentes de una cámara digital

Fuente: <https://goo.gl/sxUbSB>

1.4.3.2 Sistema de procesamiento

El sistema de procesamiento es aquel que toma la imagen del dispositivo de captura digital, y procesarla puede ser un computador o un sistema embebido. El sistema puede hacer procesamiento de imágenes, así como también técnicas de visión artificial y la velocidad con la que haga estas tareas dependerá directamente de las características del sistema y del lenguaje donde se haga la programación, así como la complejidad de lo que el sistema que se tenga.

1.4.3.3 Software

Con el sistema de procesamiento listo se necesita de un programa que tenga acceso a unas librerías de visión artificial y de esa forma acceder a un conjunto de funciones que permitan hacer varias aplicaciones de visión artificial, en la actualidad existen muchos programas en los que se puede hacer visión artificial como Matlab y Labview cuya programación es de alto nivel es mucho más fácil y eficiente también existen software libre como las librerías como OpenCV, figura 14-1. (Unpocodejava, 2013)



Figura 14-1: Librería OpenCV

Fuente: <https://goo.gl/EMwiw3>

1.4.4 Adquisición de imágenes

Las imágenes que se obtiene del dispositivo de captura pueden ser transformadas y datos que pueden ser procesados de manera mucho más sencilla y esta forma es representar una imagen en un arreglo de 2 dimensiones también llamado matriz, ya la imagen es un conjunto de valores indexados en una coordenada i, j al ser una imagen de color ese se descompone en tres matrices o matrices que juntas las diversas tonalidades, como se muestra en la figura 15-1.

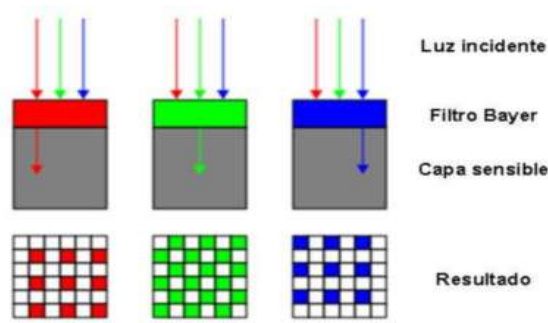


Figura 15-1: Descomposición de colores

Fuente: <https://goo.gl/EMwiw3>

1.4.5 Procesamiento de imágenes

1.4.5.1 Transformación RGB a escala de grises

El modelo RGB es modelo más usado en procesamiento de imagen por su facilidad de implementar y entender su funcionamiento y la combinación de los tres colores primarios que lo componen (rojo, verde, azul), figura 16-1, forman una amplia gama de colores pero la imagen se descompone en tres matrices asociadas a estos tres colores hace muy difícil el procesamiento de estas, por lo hay que fusionar estas tres matrices en una sola esto se hace mediante un método matemático el cual multiplica cada canal por un valor calculado que convierte cada canal que constituye en una de escala de grises cuyo rango varia de 0 a 255. (alereimondo, 2005)

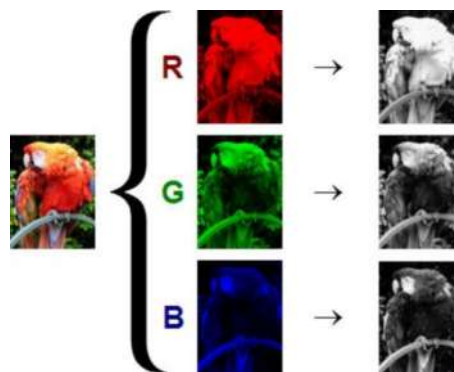


Figura 16-1: Matrices que conforman una imagen a color

Fuente: <https://goo.gl/SzacZo>

Una vez hecho esto se calcula un valor producto de una constante con cada valor que se encuentra en una coordenada específica dentro de su propia matriz el promedio de los tres valores forma una imagen resultante es una imagen en escala de grises los valores que se multiplica son 0.3 para el canal rojo, 0.59 para el canal verde y 0.11 para el canal azul.

La importancia de esta operación es que facilita el procesamiento de imágenes y el uso de técnicas de visión artificial.

1.4.5.2 *Contornos*

Los contornos son puntos continuos que definen el límite de la imagen conservando su color o intensidad facilitando el reconocimiento de objetos para su análisis posterior.

- Es recomendable el uso de imágenes binarias por ser las más precisas siendo sometidas al umbral y a la detección de bordes falsos.
- Al modificar la imagen original con la función `findContours`, también es posible almacenar en otras variables la imagen de origen antes de encontrar su contorno.
- OpenCV realiza el algoritmo para hallar contornos de un objeto blanco en un fondo negro.

Para el uso de la función `findContours` se debe obtener el argumento de la imagen de origen, el modo de recuperación de contorno y el método de aproximación del contorno. Esta función genera una lista Python donde se encuentran los contornos y la jerarquía, donde cada contorno representa una matriz Numpy (x, y) como puntos límites del objeto. (hetpro-store, 2018)

Para visualizar los contornos se usa la función `cv2.drawContours`, tiene la capacidad de dibujar diferentes formas siempre que cuenten con sus puntos límites. Para dibujar el objeto debe tener el argumento de su imagen fuente, los contornos que tendrán una lista Python con el índice de contornos con su respectivo color y grosor.

1.4.5.3 *Harris corner detection*

Es un método que detecta las esquinas de un objeto, usado usualmente en algoritmos de visión, se basa en tomar en cuenta un diferencial del puntaje de esquina con respecto a la dirección, asegurando precisión al distinguir un borde de una esquina. (aurova, 2002)

Al usar este método se obtiene una imagen en escala de grises con estos puntos, dichos puntos son derivadas de la imagen en direcciones x e y dando lugar a una puntuación para saber si tiene esquina o no.

➤ **Detector de esquina Harris en OpenCV**

Este programa tiene la función `cv2.cornerHarris()` con cuatro argumentos la imagen a escala de grises tipo flotante(`img`), el tamaño del espacio para detectar la esquina(`blockSize`), parámetro utilizado inicial del derivado Sobel (`ksize`), parámetro libre d Harris (`k`). Los resultados al usar esta función se muestran en la siguiente figura 17-1: (programacion, 2015)

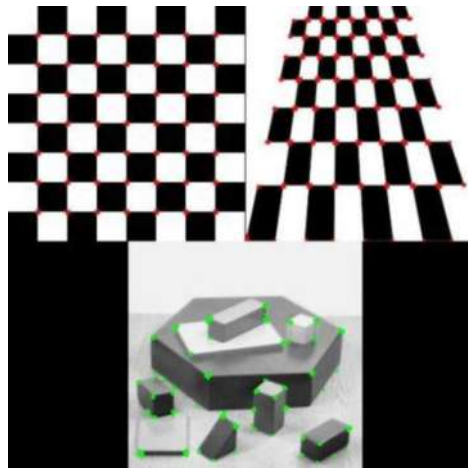


Figura 17-1: Esquina con precisión subpixel

Fuente: <https://goo.gl/ZnrWc3>

1.4.6 Dilatación

Es un operador básico de la morfología matemática que consta de la teoría de conjuntos donde se simplifica la imagen conservando sus características principales como la forma del objeto este es usado como un pre procesamiento de la imagen donde es eliminado los ruidos, y es posible obtener una descripción detallada del objeto.

Usualmente la dilatación es aplicada a imágenes binarias y no tan usual con imágenes en escala de grises, su efecto en imágenes binarias consiste en ampliar gradualmente las áreas de pixeles en primer plano y reduciendo el tamaño de los agujeros.

La estructura del operador de dilatación consta de dos entradas, la imagen que se va a dilatar y el conjunto de puntos de coordenada conocido como (kernel), para determinar el efecto preciso de dilatación.

Para una imagen binaria la definición matemática consta de un conjunto de coordenadas euclidianas (X) que es la imagen binaria de entrada y un conjunto de coordenadas para el elemento de estructuración (K), entonces la dilatación entre X por K es el conjunto de puntos de (x) tal que Kx (origen en x) con X sea diferente del vacío.

La dilatación de imágenes en escala de grises es similar a una imagen binaria con la diferencia que tiene un conjunto de elementos en 3D.

Al realizar la operación de dilatación se espera una imagen binaria con pixeles de primer plano de 255 por 0 pixeles de fondo y es posible obtenerla desde una imagen en escala de grises al usar el umbral, figura 18-1, es importante una correcta configuración en su entrada para obtener resultados más factibles y precisos. (alojamientos, 2009)

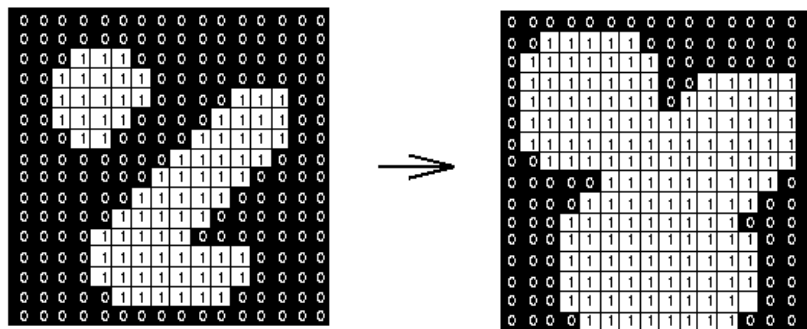


Figura 18-1: Efecto de dilatación

Fuente: <https://goo.gl/ZnrWc3>

La estructura cuadrada de 3x3 es la más común en la operación de dilatación, mientras más grande sea el elemento estructurado su efecto de dilatación también será extremo, es posible lograr el mismo efecto al usar un elemento estructurado más pequeño pero repetidas veces la operación de dilatación.

Para la detección de bordes a base de la operación de dilatación es necesario realizar una resta a la imagen original, marcando los pixeles en el borde del objeto que se agregaron al realizar la dilatación.

1.5 Software empleado en el sistema

Se realiza una breve descripción de los distintos softwares que se empleó para la realización del prototipo clasificador de huevo de gallina aplicando visión artificial, entre los cuales se tiene los siguientes:

1.5.1 *SolidWorks*

El software SolidWorks es utilizado para diseñar mecanismos y estructuras de carácter ingenieril ya que posee un entorno potente que brinda herramientas muy útiles en el diseño, así

como simulaciones de los mecanismos diseñados. El entorno gráfico de SolidWorks es muy amigable a la vista como se puede ver en la figura 19-1.

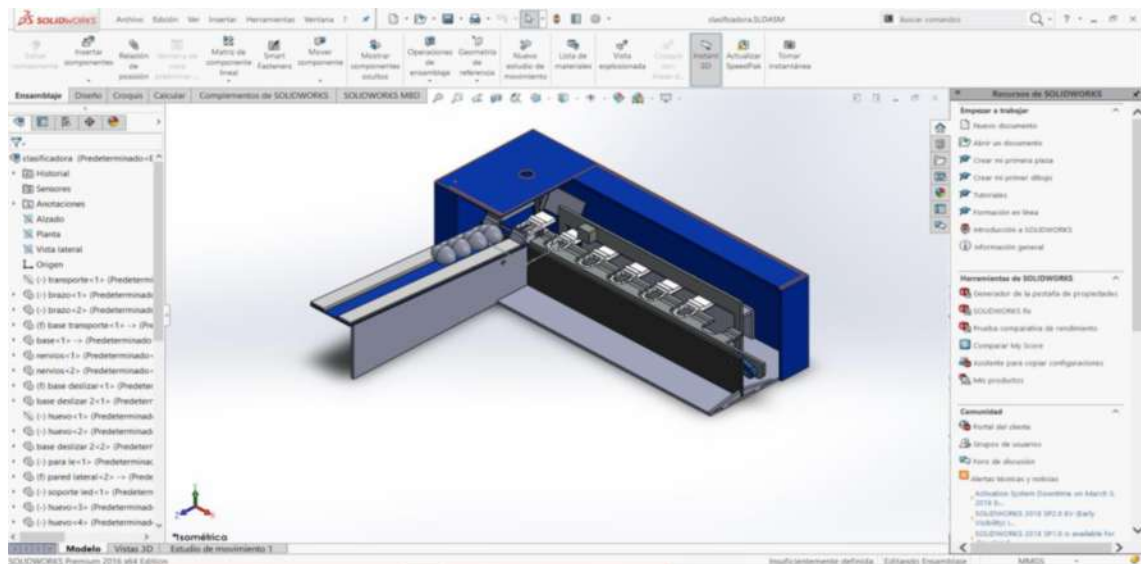


Figura 19-1: Entorno gráfico de SolidWorks

Realizado por: Paguay, David; Valarezo, Luis; 2018

1.5.2 Adams

Adams es uno de los programas que ayuda a simular la dinámica de piezas mecánicas que pueden moverse, también calcula las cargas y fuerzas que se distribuyen en dichos sistemas, con este programa se pueden simular dichos procesos para evitar cambios en los proyectos y adelantarnos a daños futuros en las estructuras realizadas, la visualización del software lo podemos observar en la figura 20-1.

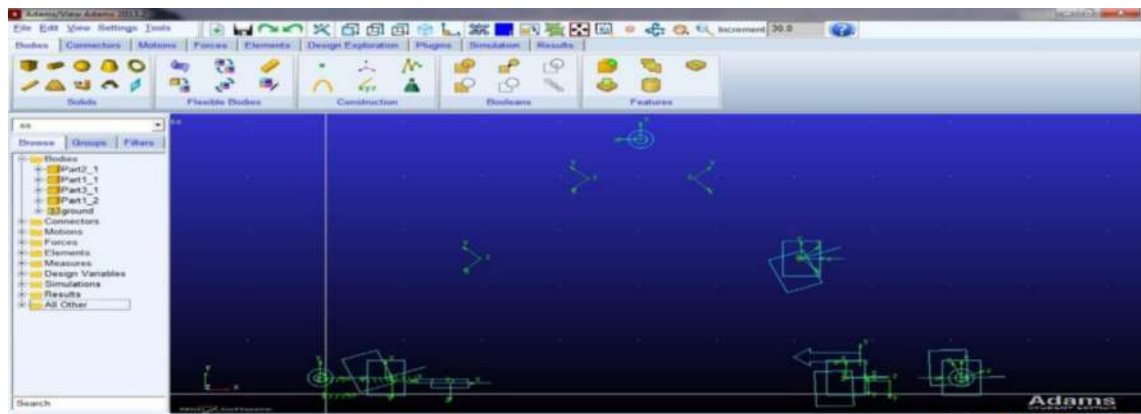


Figura 20-1: Entorno de Adams

Realizado por: Paguay, David; Valarezo, Luis; 2018

1.5.3 Arduino IDE

El entorno de programación de Arduino es de código abierto basándose en un lenguaje de programación C++, es compactible con cualquier sistema operativo y su estructura es muy simple permitiendo que cualquier persona sea capaz de entender la programación. Cuenta con librerías que facilitan la interacción con el hardware y el manejo de funciones para el desarrollo de las operaciones, figura 21-1. (aprendiendoarduino, 2017)



```
motor_paso
Archivo Editar Programa Herramientas Ayuda

motor_paso

//const int stepPin = 2; // declaramos el pin 10 del arduino como step
//const int dirPin = 3; // declaramos el pin 12 del arduino con dir
const int tiempo = 5; // declaramos la variable tiempo como 5 milisegundos
const int an1 = A1;
int leer1;
int leer2;
void setup() {

  pinMode(stepPin,OUTPUT); // declaramos como salida
  pinMode(dirPin,OUTPUT); // declaramos como salida
  Serial.begin(9600);
}

void loop() {

  digitalWrite(stepPin,HIGH); // ponemos en alto a 1 logico a pin dir para que gire hacia un sentido

  for(int n = 0; n < 200; n++) // usamos un loop for . en nuestro caso el motor es de 200 pasos, para
  digitalWrite(stepPin,HIGH); // las instrucciones que se encuentran dentro del loop
  delay(tiempo); // con las que paso que nuestro motor se mueva .es la parte donde van
  digitalWrite(stepPin,LOW); // sucesion para que el motor se mueva
  delay(tiempo);
  leer1 = analogRead(an1);
  leer2 = analogRead(an1);
  Serial.println(leer2);
  delay(500);
  //Serial.println(leer2);
}
}

```

Figura 21-1: Entorno Arduino IDE

Realizado por: Paguay, David; Valarezo, Luis; 2018

1.5.4 Python

Phyton se podría considerar como un lenguaje de programación de alto nivel, permite realizar muchas funciones por su fácil comprensión y síntesis en la programación, su estructura difiere de los otros lenguajes como C++ ya que debe seguir un orden de rango en la programación, el cual puede dar ciertos errores en sintaxis al momento de compilar, posee palabras propias y librerías que mejoran la codificación con palabras escritas en inglés.

Es compactible con cualquier sistema operativo ya que es un lenguaje de programación libre, su estructura difiere del tipo de versión que se tenga instalada cambiando el uso de caracteres y el tamaño de su sangría, su respuesta en cuanto a programación es lenta por lo que es un lenguaje de alto nivel, figura 22-1. (es.gizmodo, 2013)



Figura 22-1: Entorno de Python

Realizado por: Paguay, David; Valarezo, Luis; 2018

1.6 Raspberry PI

La Raspberry es considerada como un pequeño computador, trabaja bajo el sistema operativo raspbian propio de la fundación Raspberry pi, existen varios modelos siendo el último en salir el modelo Raspberry pi 3b que a diferencia de los otros posee un procesador de cuatro núcleos conjuntamente con 1Gb en RAM y ranuras para conexiones USB, hdmi, Ethernet e interfaces para cámara y pantalla como puertos para audio. Figura 23-1

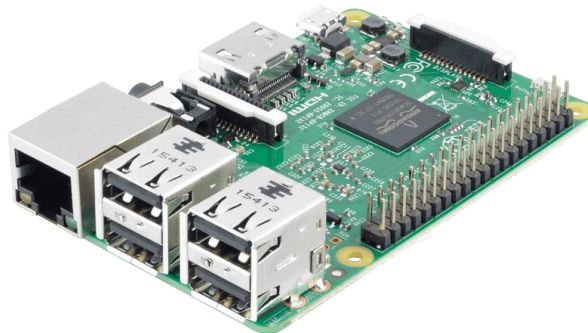


Figura 23-1: Raspberry PI

Fuente: <https://goo.gl/SfG5VM>

1.6.1 Hardware Raspberry PI3

La Raspberry pi consta de un procesador de 1.2 GHz con 64bit ARMv8 de cuatro núcleos, es uno de los computadores más pequeños y potentes que puede haber, permite realizar proyectos en los que se inicie en la programación con lo que se puede usar cualquiera de los 40 GPIO y conectarse de forma inalámbrica a la red, consta de una ranura para tarjetas micro SD que guarda el sistema operativo y todos los archivos creados, figura 24-1. (tuelectronica, 2016)

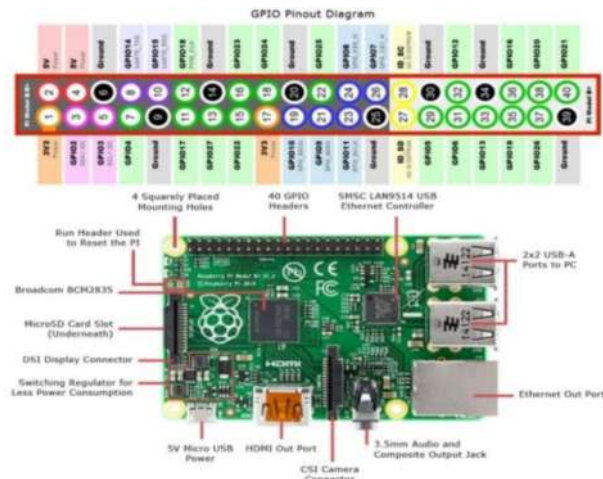


Figura 24-1: Partes de la Raspberry pi

Fuente: <https://goo.gl/JBh8e6>

1.7 Dispositivos y componentes del sistema

1.7.1 Sensores

Los sensores son elementos capaces de recibir magnitudes de información del exterior y transformarlas en otras magnitudes por lo general son eléctricas, estas señales pueden ser acondicionadas para medir temperatura, presión, humedad, posición entre los principales que pueden haber.

1.7.1.1 Sensor Qrd

El Qrd es un sensor óptico infrarrojo de corto alcance, el módulo que se utiliza realiza su funcionamiento por medio de emisión de luz y el detector está en la misma dirección, por su fácil funcionamiento es utilizado en algunos casos como sensor de presencia, detección de colores y su costo no es muy elevado, para el desarrollo y medición de sensores se pudo ver que los Qrd eran los más óptimos por su fácil utilización y puede trabajar de rangos de 3.3V a 5V, figura 25-1 (teslabem, 2016)

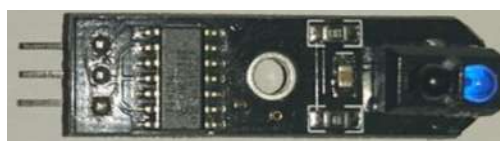


Figura 25-1: Sensor qrd

Realizado por: Paguay, David; Valarezo, Luis; 2018

1.7.1.2 Galga extensiométrica

Las galgas son sensores que miden su deformación aplicando una presión o carga sobre ellas, variando su resistencia que puede cambiar de acuerdo al material de cada elemento, este efecto se llama piezo resistivo que al aplicar un peso producirá una variación en su resistencia eléctrica. Para obtener los valores de resistencia que nos da la galga en ocasiones toca realizar un arreglo de resistencias llamado puente de wheatstone para obtener las lecturas de las variaciones su grado de tolerancia es $\pm 1\%$, figura 26-1. (es.omega, 2013)



Figura 26-1: Galga de peso

Realizado por: Paguay, David; Valarezo, Luis; 2018

Ahora en la actualidad es mucho más fácil de medir el cambio de resistencia de las galgas por medio de un acondicionador de señal que reemplaza al puente de wheatstone y permite de una manera más sencilla leer las variaciones de nuestra celda de carga, el acondicionador es una placa que va directamente conectada a la galga y es la siguiente

- Hx711: el módulo es el encargado de la lectura del puente de wheatstone, permite la conexión entre la celda de carga y el controlador, convierte la lectura analógica con su conversor de A/D a digital de 24 bits, figura 27-1.



Figura 27-1: HX-711

Realizado por: Paguay, David; Valarezo, Luis; 2018

1.7.2 Fuente de alimentación

La fuente de alimentación es un dispositivo capaz de transformar la corriente alterna en corriente continua, posee varias salidas de las que podemos decir que son de $\pm 12V$, $\pm 5V$, $+3.3V$ como se puede ver en la figura 28-1, la fuente alimentara los circuitos electrónicos del prototipo ya que no se necesita de mucha corriente para que funcionen los elementos.



Figura 28-1: Fuente de alimentación

Realizado por: Paguay, David; Valarezo, Luis; 2018

1.7.3 Actuadores

1.7.3.1 Servomotores

El servo es un dispositivo que está compuesta por un pequeño motor, un juego de engranes que reducen la velocidad, pero aumentan la fuerza, su rango de trabajo es de 0° a 180° pero con un arreglo interno del servo se puede modificar para que este trabaje de 0° a 360° como un motor estándar, figura 29-1.



Figura 29-1: Servomotor

Fuente: <https://goo.gl/Pf1BVP>

Para poder controlar un servo se debe aplicar un pulso de cierta duración y frecuencia específica, estos poseen tres cables que servirán tanto para la comunicación y la alimentación del mismo, los colores de los cables varían según las marcas como podemos ver en la tabla 4-1.

Tabla 4-1: Características de los servos según su modelo

Fabricante	Duración de pulsos [ms]			Frecuencia [Hz]	Color de los cables		
	Mínima (0°)	Neutral (90°)	Máxima (180°)		Positivo	Negativo	Control
Futaba	0.9	1.5	2.1	50	Rojo	Negro	Blanco
Hitech	0.9	1.5	2.1	50	Rojo	Negro	Amarillo
Graupner/Jr	0.8	1.5	2.2	50	Rojo	Marrón	Naranja
Multiplex	1.05	1.6	2.15	50	Rojo	Negro	Amarillo
Robbe	0.65	1.3	1.95	50	Rojo	Negro	Blanco
Simprop	1.2	1.7	2.2	50	Rojo	Azul	Negro

Fuente: <http://www2.elo.ut fsm.cl/~mineducagv/docs/ListaDetalladadeModulos/servos.pdf>

Realizado por: Paguay, David; Valarezo, Luis; 2018

1.7.3.2 Motor de pasos

Los motores de pasos son ideales para mecanismos que necesiten precisión, para poder moverlos se aplican pulsos para que recorran paso a paso una vuelta que generalmente en algunos modelos son de 200 pasos para realizar un giro completo de 360°, Figura 30-1, están constituidos por un rotor aplicados imanes a un numero de bobinas excitadoras en su estator todo esto debe ser manejada por un controlador. (robots-argentina, 2006)

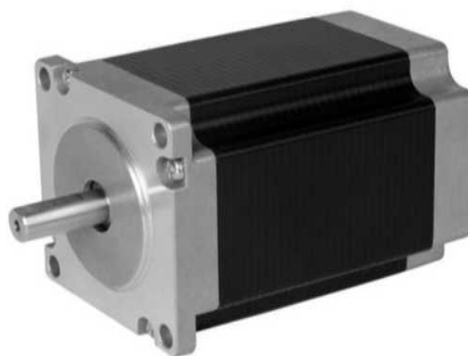


Figura 30-1: Motor de pasos Nema 23

Fuente: <https://goo.gl/zKmEbh>

CAPITULO II

2. MARCO METODOLÓGICO

En este capítulo, se presenta la descripción del prototipo, la selección de los elementos que lo componen, el diseño y la implementación de los sistemas mecánicos, electrónicos y de visión artificial, la programación del algoritmo para la detección de fallos en la cáscara de huevo y de la separación mediante la comparación de sus pesos, la creación y almacenamiento de los registros obtenidos en una base de datos, así como la interfaz gráfica que muestra dichos resultados.

Para el desarrollo del trabajo de titulación se aplica la técnica de observación y recolección de datos relacionados a la clasificación de huevos de gallina de forma manual. Se hace uso de la metodología experimental para el diseño de los diversos sistemas que conforman el prototipo. En el gráfico 1-2, se muestra las etapas, para alcanzar estos objetivos.

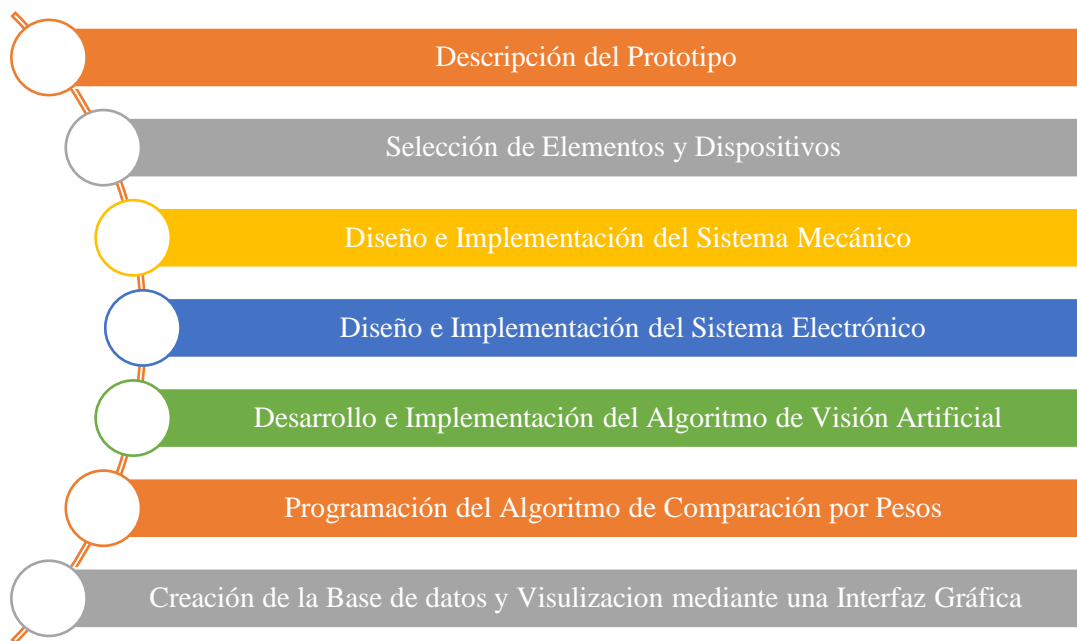


Gráfico 1-2: Etapas del diseño e implementación del prototipo

Realizado por: Paguay, David; Valarezo, Luis; 2018

Para la elaboración de tablas, cuadros y gráficos se emplea la técnica de análisis de contenido.

2.1 Descripción del prototipo clasificador de huevos de gallina

El prototipo para la clasificación de huevos de gallina basado en las imperfecciones de la cáscara aplicando visión artificial; busca mejorar el tiempo empleado en el proceso de forma manual, así como obtener un resultado de incremento en la calidad del producto mediante la valoración del mismo, y llevar un control de la producción mediante el empleo de una base de datos. El gráfico 2-2, muestra el funcionamiento del prototipo.

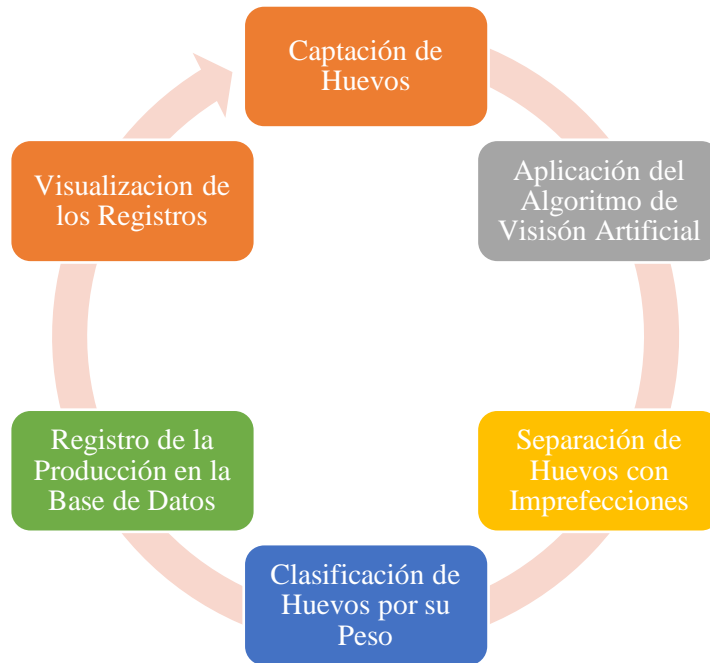


Gráfico 2-2: Descripción del proceso del prototipo

Realizado por: Paguay, David; Valarezo, Luis; 2018

El proceso de clasificación, comienza con la captación de los huevos de gallina los cuales mediante una rampa caen en forma de hilera hacia una estructura adecuada con un sistema de iluminación y arreglo de espejos óptimos para poder aplicar visión artificial. Un sensor determina la presencia del huevo, iniciando la ejecución del algoritmo de visión artificial programado en la Raspberry en busca de anomalías.

Culminado el análisis efectuado con el algoritmo, se activa un mecanismo que transporta los huevos hacia unas bases acondicionadas a la forma del huevo, en este punto se efectúa su clasificación mediante un mecanismo que los separa hacia carriles de acuerdo a su tamaño y los fallos que presentan. Los huevos detectados con algún tipo de fallo mediante el algoritmo son separados en el primer carril, dejando recorrer a los demás.

Los huevos que no presentan anomalías, son separados a través del empleo de sensores que determinan su peso y mediante la aplicación de un algoritmo de comparación de los mismos,

estos son clasificados en: huevos pequeños, huevos medianos y huevos grandes según dictan las normas INEN. Los registros de producción y calidad son almacenados en una base de datos y visualizados mediante una interfaz gráfica.

2.2 Selección de elementos y dispositivos del prototipo

A continuación, se presenta la selección de los elementos y dispositivos que se emplean para la implementación del prototipo, se aplica el método inductivo para adquirir información que nos permita establecer los elementos que se ajusten a las necesidades buscando su mejor funcionamiento. Los elementos que se emplean en cada una de las etapas del prototipo se describen en la tabla 1-2.

Tabla 5-2: Elementos y dispositivos que componen el prototipo

Elementos	Características
MOTOR DE PASOS	NEMA 23 DE 12,5 KG.CM
DRIVER A4988	8-35V, 2A PARA MOTORES DE PASO BIPOLAR
POLEA	16MM(DIÁMETRO)X16MM(ALTURA), 20 DIENTES
CÁMARA WEB	MARCA GENIUS F100, 1080P FULL HD PIXEL CMOS, 12MP
LED	LUZ LED DE ALTO BRILLO DE 9 PUNTOS
RASPBERRY PI3 B	1 GB RAM, 4 PUERTOS USB, ETHERNET, HDMI, 40 GPIO
PLACA ARDUINO	5-12V E/S ANALÓGICAS Y DIGITALES
SENSOR QRD	3,3-5, DISTANCIA DE DETECCIÓN 10MM
SERVOMOTORES	TOWER PRO MG996R ENGRANAJE METÁLICO 9,4 KG/CM
GALGAS	EXTENSIOMETRICAS DE 5KG
CONTROLADOR DE GALGAS	HX711, 2,7-5V
HMI	PANTALLA LCD 3,5" TFT 480X320 TOUCH
FUENTE	800W, 38A

Fuente: Paguay, David; Valarezo, Luis; 2018

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.3 Diseño e implementación del sistema mecánico

Para el diseño del sistema mecánico se recurre al método heurístico, que permite según una serie de pasos hallar la solución al problema, junto con técnicas adquiridas durante la carrera se diseñan los mecanismos necesarios usando el software SolidWorks para poder implementar el

prototipo de clasificación de huevos. Los mecanismos empleados en el prototipo según sus etapas se detallan en el gráfico 3-2.

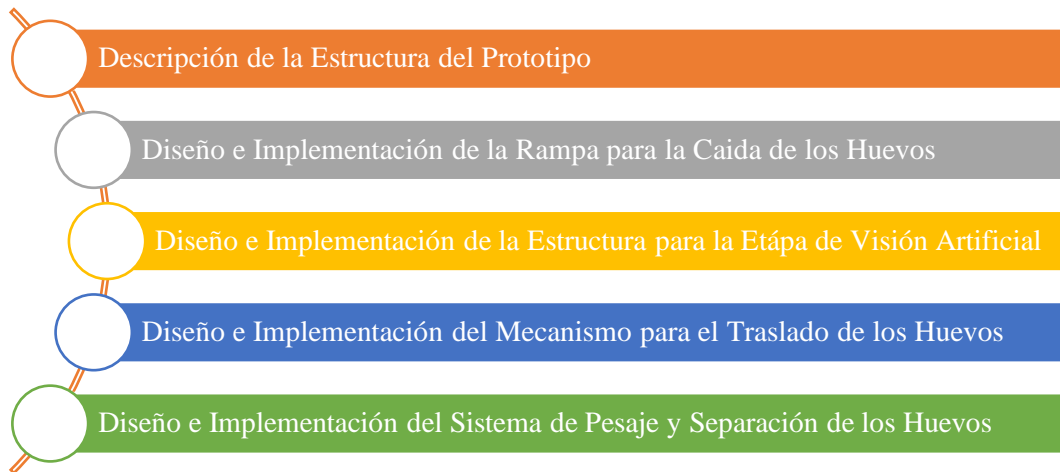


Gráfico 3-2: Etapas de diseño e implementación del sistema mecánico

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.4.1 Descripción de la estructura del prototipo

Realizado el estudio correspondiente a la clasificación de huevos de gallina en forma manual, se procede al desarrollo del diseño de la estructura que permite realizar dicho procedimiento de forma automática, figura 1-2, empleando un sistema de visión artificial para separar los huevos que presenten anomalías en la superficie de la cáscara, mismas que en ocasiones son poco visibles para el ojo humano.

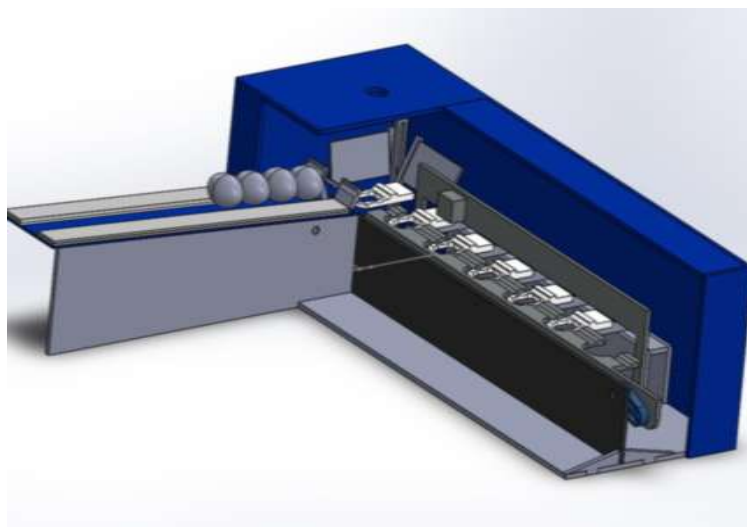


Figura 1-2: Estructura del prototipo diseñado en SolidWorks

Realizado por: Paguay, David; Valarezo, Luis; 2018

La estructura del prototipo está conformada por varias etapas mismas que son diseñadas para realizar una función determinada, entre las cuales se tiene:

- Rampa de captación de huevos.
- Estructura para la captura de imágenes.
- Mecanismo para el transporte de los huevos.
- Sistema de pesaje y separación.

2.4.1.1 *Diseño de la rampa de captación de huevos*

Para el pre procesamiento de los huevos se implementa una estructura en forma de rampa, figura 2-2, donde el operador va ubicando los huevos para que se deslicen en forma de hilera; se realizan pruebas en el software Adams para calcular el ángulo de inclinación que debe poseer la rampa para evitar que los huevos sufran fisuras ocasionadas por el diseño del prototipo.

Con el ángulo de inclinación adecuado se consigue una velocidad de traslado moderada. Al final de la rampa se ubica un pequeño mecanismo que deja pasar un huevo a la vez, mientras los demás quedan en espera el tiempo que se desarrolla el análisis mediante el algoritmo de visión artificial en busca de fallos.

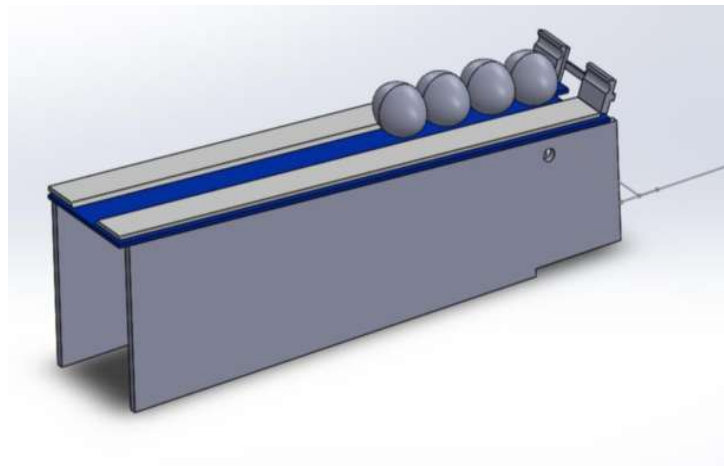


Figura 2-2: Diseño de la rampa de captación de huevos

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.4.1.2 *Diseño de la estructura para la captura de imágenes*

Para la captura y procesamiento de las imágenes se diseña una estructura adecuada a la forma del huevo como se observa en la figura 3-2, donde este repose mientras se ejecuta el algoritmo de visión. Se emplea un arreglo de espejos para conseguir la captura de imágenes de la superficie del huevo. En la base donde reposa el huevo se introduce un led de alto brillo para de esta manera visualizar el contorno de la cáscara, tal como se ejecuta en la técnica de ovoscopia.

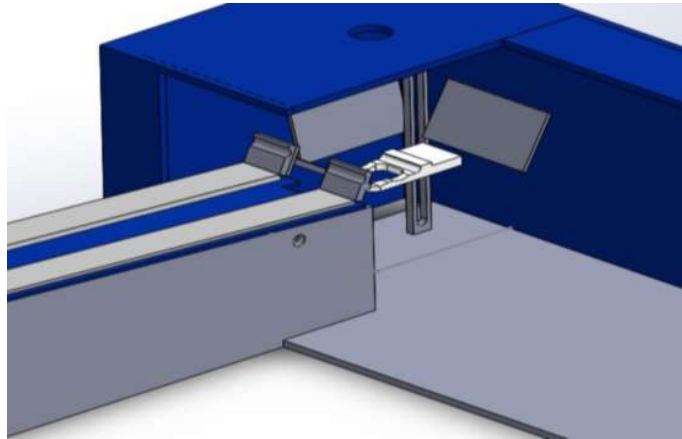


Figura 3-2: Diseño de la estructura de captura de imagen

Realizado por: Paguay, David; Valarezo, Luis; 2018

La cámara se ubica a una altura de 21cm con respecto a la base, esto permite tener un mejor campo de visión para la captura de imágenes que serán procesadas, evitando tener cuerpos extraños en las capturas. Internamente la estructura de captura de imágenes es de color negro, creando un ambiente óptimo para la adquisición de las imágenes evitando las reflexiones en las paredes internas de la estructura.

2.4.1.3 *Diseño del mecanismo para el traslado de huevos*

Se optó por un mecanismo que hace uso de una banda accionada por un motor de pasos que, en conjunto con acoples y bielas, figura 4-2, generan un movimiento rotacional que hace posible el traslado de los huevos hacia bases sin que estos sufran daños al momento del movimiento realizado.



Figura 4-2: Diseño de la biela empleada en el prototipo

Realizado por: Paguay, David; Valarezo, Luis; 2018

El sistema de movimiento rotacional emplea una bandeja de transporte con capacidad de trasladar un número de 6 huevos hacia las bases donde se procede a su pesado y posterior separación, por cada giro que se efectuó, figura 5-2.

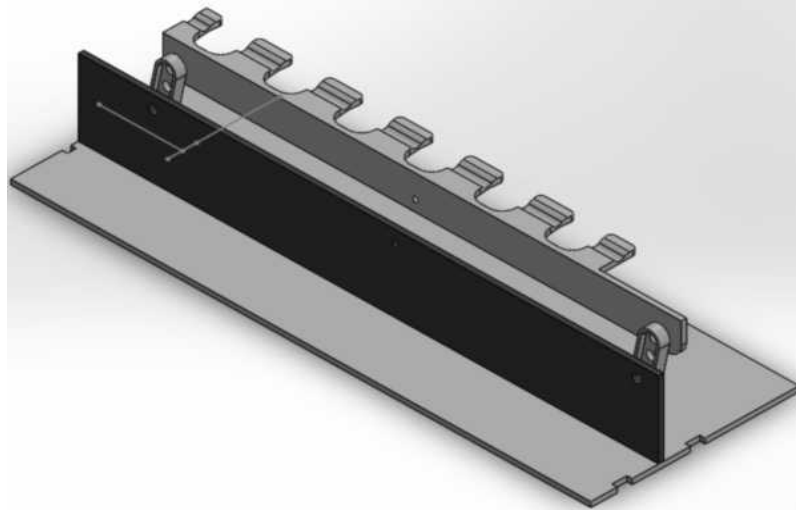


Figura 5-2: Diseño del sistema rotacional y bandeja de transporte

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.4.1.4 *Diseño del sistema de pesaje y separación*

Para efectuar el pesado y separación se diseñan bases que se adaptan a la forma de los huevos, sirviendo de soporte para efectuar el proceso de pesaje, figura 6-2. Las bases se encuentran separadas a una distancia equitativa permitiendo el paso de la bandeja que traslada los huevos.

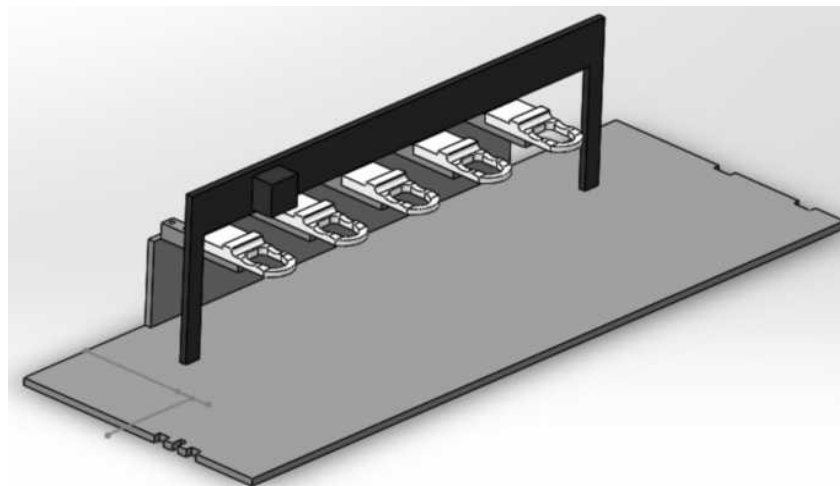


Figura 6-2: Diseño del sistema de pesaje y separación del prototipo

Realizado por: Paguay, David; Valarezo, Luis; 2018

Para efectuar la separación se emplean actuadores que tienen acoplados un eslabón en forma de garra impresa en 3D, figura 7-2, mismos que se encuentran ubicados en la parte superior de las bases.

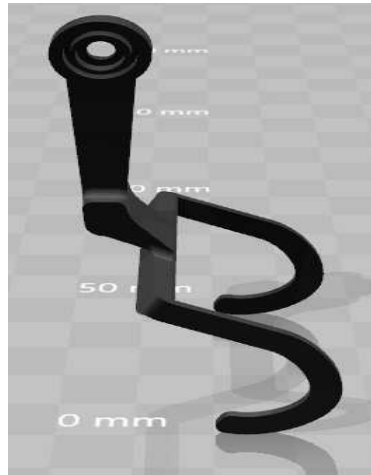


Figura 7-2: Diseño del eslabón

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.4.2 Implementación de los mecanismos del prototipo

Culminada la etapa de diseño de los distintos mecanismos y estructuras que conforman el prototipo se procede a la implementación. Empleando tecnologías como son el corte laser de MDF y la impresión 3D en PLA se pudo materializar los diversos mecanismos y estructuras.

La implementación de la rampa por donde son captados los huevos de gallina, se realiza empleando material MDF de 5.5 mm de espesor. Mediante la separación en partes de la rampa, estas son cortadas mediante laser para su posterior ensamble, teniendo como resultado el que se aprecia en la figura 8-2.



Figura 8-2: Rampa de captación de huevos

Realizado por: Paguay, David; Valarezo, Luis; 2018

La estructura para la captura de imágenes se implementa con material MDF, figura 9-2, se ubican en la base los soportes que sirven de apoyo para los espejos, de soporte para el huevo mientras se ejecuta el algoritmo de visión. Además, se ubica una en la base de soporte la luz led empleada para poder desarrollar el proceso de visión.



Figura 9-2: Estructura para captura de imágenes

Realizado por: Paguay, David; Valarezo, Luis; 2018

Para la implementación del mecanismo de traslado de huevos se emplea una banda dentada, que en conjunto con un acople adherido al motor de pasos logran un movimiento circular de la bandeja transportadora, como se observa en la figura 10-2.



Figura 10-2: Mecanismo de transporte de huevos

Realizado por: Paguay, David; Valarezo, Luis; 2018

El sistema de pesaje y clasificación se implementa para que este sea adecuado a la forma del huevo y no sufra daño alguno al momento de realizar estos procesos. El resultado de la implementación se aprecia en la figura 11-2.

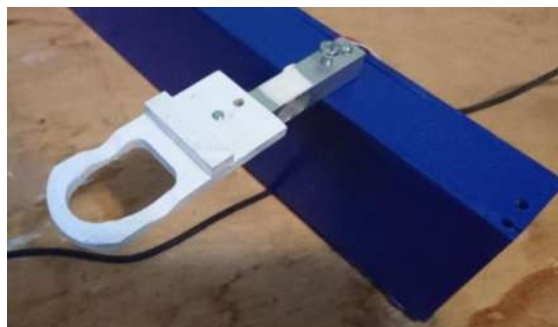


Figura 11-2: Sistema de pesaje

Realizado por: Paguay, David; Valarezo, Luis; 2018

Implementadas todas las estructuras y todos los mecanismos que conforman el prototipo clasificador, se presenta el resultado del mismo en la figura 12-2.



Figura 12-2: Estructura mecánica del prototipo

Realizado por: Paguay, David; Valarezo, Luis; 2018

Finalmente se procede a pintar el prototipo haciendo empleo de pintura acrílica, para dar una mejor presentación como se muestra en la figura 13-2.

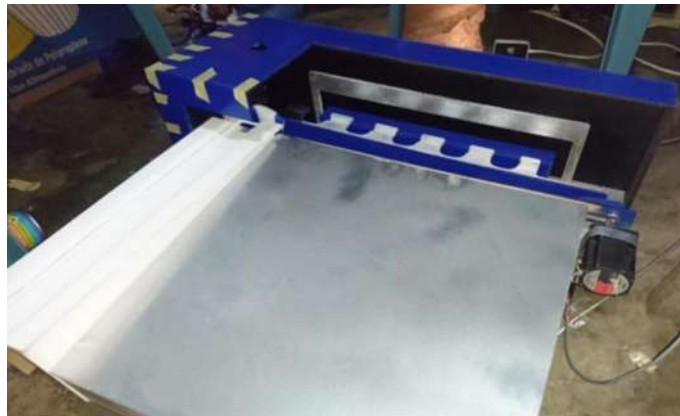


Figura 13-2: Estructura pintada del prototipo clasificador

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.4 Diseño e implementación del sistema electrónico

El prototipo clasificador de huevos cuenta con una placa electrónica que realiza las tareas de monitoreo y control de los sensores y actuadores. Previa su implementación se realiza la selección del controlador y elementos, se elabora el esquema de conexión y el diseño del PCB. En el gráfico 4-2, se enlista el procedimiento empleando para la elaboración de la placa electrónica.



Gráfico 4-2: Etapas para el diseño e implementación del sistema electrónico

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.5.1 Selección de componentes para la placa electrónica

Para efectuar el proceso de clasificación, el prototipo hace empleo de los siguientes componentes electrónicos mismos que conforman la placa PCB.

2.5.1.1 Galga extensiométrica de 5 Kg

Sensor que realiza el pesaje de los huevos mediante la medición del cambio de la resistencia eléctrica, producida por la deformación resultante de la aplicación de una fuerza sobre ella. Se optó por este sensor debido a que su diseño, figura 14-2, se adapta a la base donde se realiza el pesaje de los huevos, presenta compatibilidad con diversos controladores y una fácil calibración mediante software a través de un módulo.



Figura 14-2: Galga extensiométrica 5 Kg

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.5.1.2 Módulo HX-711 para galga extensiométrica

Se emplea como un medio de conexión entre la galga y el controlador, facilita la lectura del peso mediante la conversión de la lectura analógica a digital procedente de la galga. Se emplea este módulo por la compatibilidad que presenta con la galga extensiométrica seleccionada, además de que su tamaño es compacto y de fácil implementación, figura 15-2.



Figura 15-2: Módulo HX-711

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.5.1.3 Qrd

Sensor óptico, figura 16-2, que mide el nivel de luz. Se optó por seleccionar este sensor ya que cuenta con un filtro de luz que mitiga las perturbaciones de la luz de día. Dentro del prototipo se usa para la detección de la presencia del huevo e iniciar el análisis mediante visión artificial, así como la función de final de carrera óptico para el mecanismo de transporte de los huevos.



Figura 16-2: Sensor QRD

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.5.1.4 Driver A4988

La figura 17-2 muestra el driver empleado para el control del motor de pasos Nema 23 empleado en el mecanismo de transporte de los huevos del sistema mecánico. Se emplea debido a que permite el manejo de motores de pasos con una corriente de hasta 2 A y el control que efectúa mediante dos líneas de control.



Figura 17-2: Driver A4988

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.5.2 Elección del controlador para el prototipo

Para la realización del trabajo de titulación, se opta por el empleo del minicomputador Raspberry PI3, figura 18-2, como controlador principal del prototipo clasificador de huevos de gallina debido a que se presenta como un sistema embebido capaz de: permitir la aplicación de visión artificial, programar una interfaz gráfica, crear una base de datos.



Figura 18-2: Minicomputador Raspberry PI3

Realizado por: Paguay, David; Valarezo, Luis; 2018

Además, dispone de la cantidad suficiente de GPIOs (pines) de conexión para realizar el monitoreo de sensores y el control de los actuadores. La Raspberry se encarga de las siguientes funciones:

- Ejecución del algoritmo de visión artificial, para la detección de anomalías en los huevos de gallina.
- Separación de los huevos defectuosos mediante el accionamiento de un actuador.
- Control del movimiento del motor, para el traslado de los huevos.
- Creación de la base de datos con los registros arrojados por el prototipo.
- Visualización de los datos almacenados mediante una interfaz gráfica.

Si bien la Raspberry se encarga de la mayoría de funciones del prototipo, esta presenta la limitante de no poseer pines analógicos negando la posibilidad de efectuar las lecturas de las galgas extensiométricas. Siendo necesario el empleo de la tarjeta Arduino, figura -2, para acondicionar la señal de las galgas y realizar la clasificación de los huevos mediante el accionamiento de actuadores servomotores.

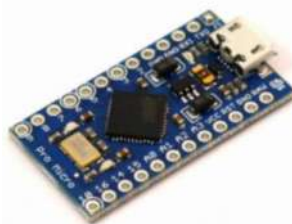


Figura 19-2: Arduino Pro Micro

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.5.3 Esquema de conexión

La figura 20-2, presenta el esquema de conexión de los elementos electrónicos que conforman la placa electrónica del sistema.

El minicomputador Raspberry PI3 se presenta como el elemento controlador principal, encargado de la lectura de los sensores qrd que son empleados para dar inicio al proceso de visión y verificar la ubicación de la plataforma de transporte de huevos. Se encarga además del control del servomotor que separa los huevos con anomalías y el motor de pasos que conforma el mecanismo de transporte.

La pantalla táctil conectada a la Raspberry, se encarga de la visualización de los datos registrados en la base de datos, resultantes del proceso de clasificación.

La etapa de clasificación se realiza mediante el acondicionamiento de las galgas de peso, mediante el empleo de las tarjetas Arduino, mismas que también se encargan del accionamiento de servomotores que realizan la clasificación acorde a un algoritmo de comparación de pesos.

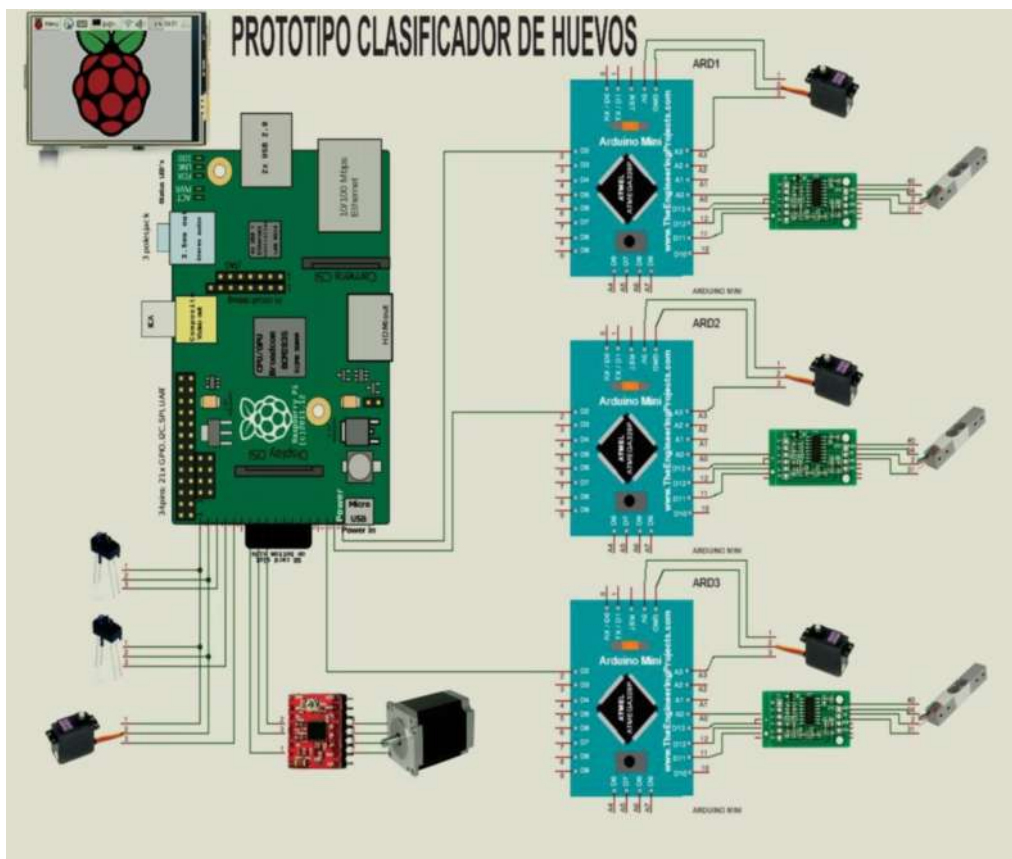


Figura 20-2: Esquema de conexión del sistema electrónico

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.5.4 Diseño de la placa PCB

El diseño de la placa electrónica del prototipo, se efectúa mediante el software Eagle donde se diagraman las conexiones de los distintos componentes en Eagle Schematic de acuerdo al esquema de conexión. En Eagle Board se crea el circuito PCB impreso, debido a que permite realizar un ruteo profesional hasta llegar al resultado final que se muestra en la figura 21-2.

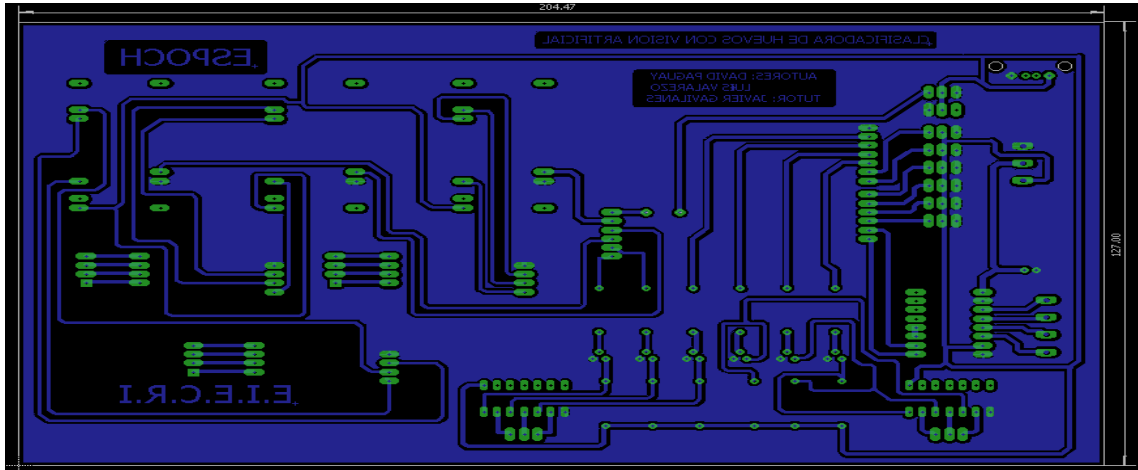


Figura 21-2: Placa PCB del prototipo

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.5.5 Implementación de la placa electrónica

Finalizado con el proceso de diseño se procede a la implementación de la placa PCB, aplicando la técnica de transferencia de calor se traspa el diseño impreso en papel couché hacia la baquela, mediante el ataque con ácido férrico las pistas quedan corroídas para finalmente proceder a la perforación, ubicación y posterior soldado de los componentes como se muestra en la figura 22-2.

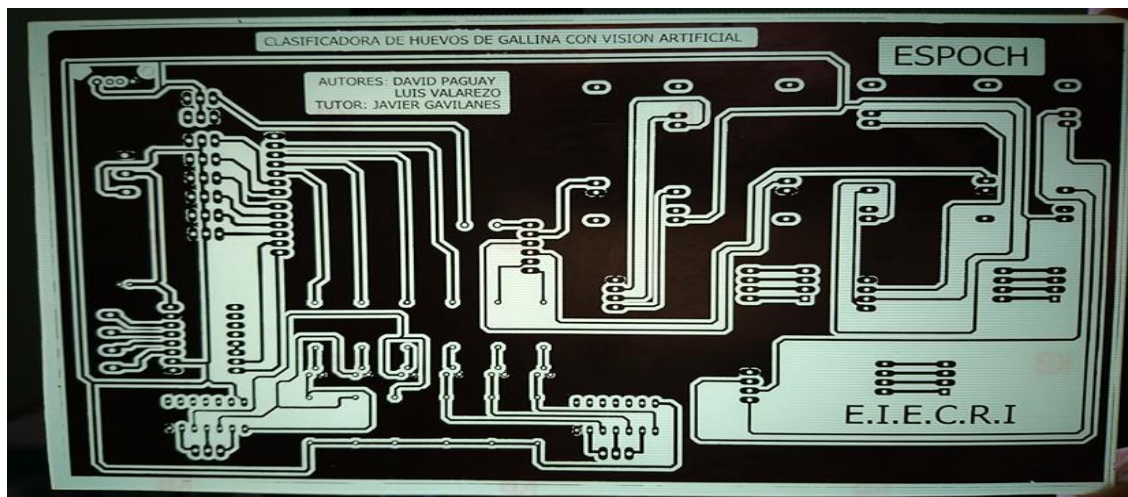


Figura 22-2: Placa PCB implementada

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.5 Desarrollo e implementación del algoritmo de visión artificial

El gráfico 5-2, muestra el diagrama de las etapas empleadas para el desarrollo y posterior implementación del algoritmo de visión para la detección de los fallos en la superficie de la cáscara de los huevos.

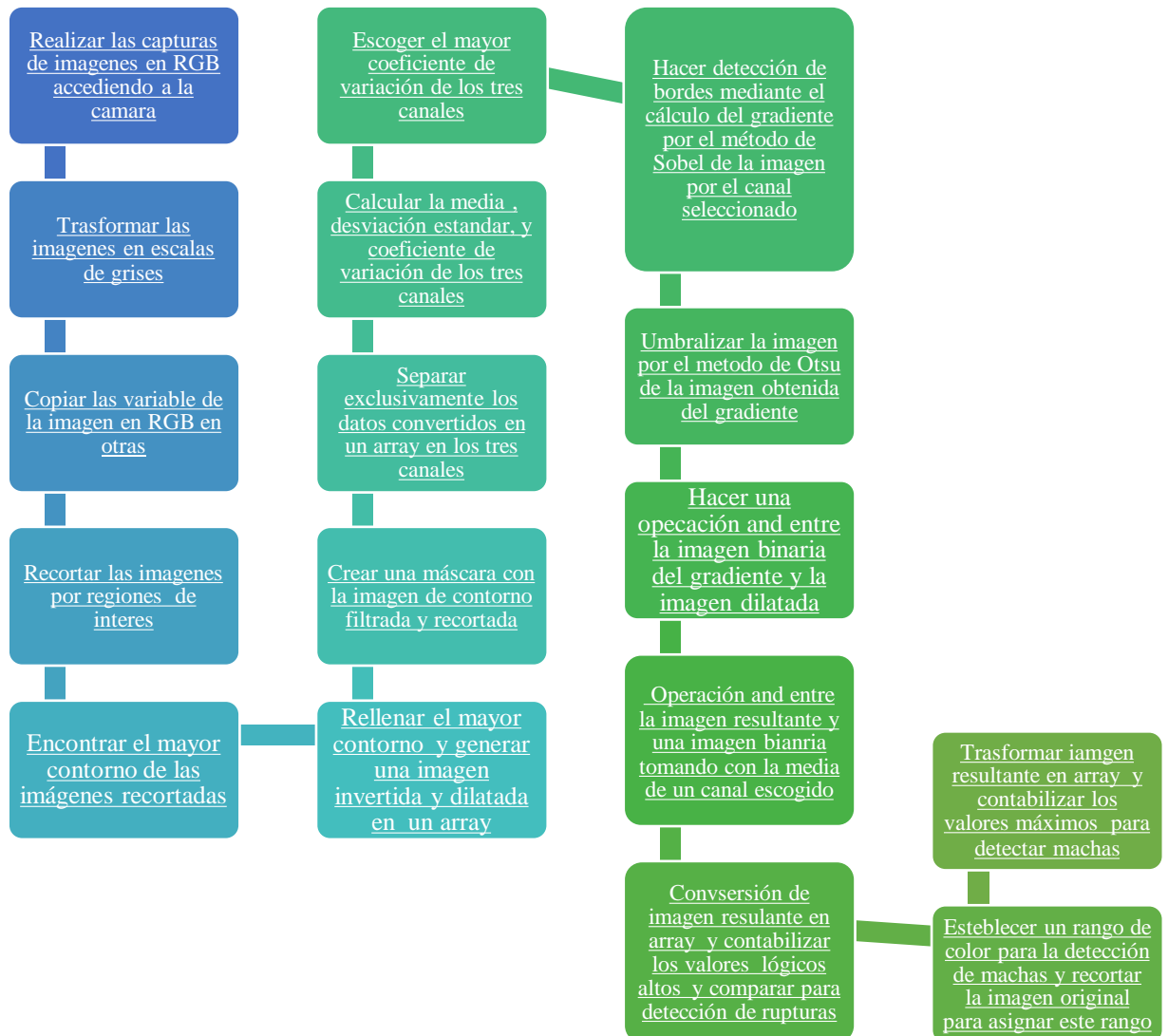


Gráfico 5-2: Diagrama de las etapas del algoritmo de visión artificial

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.6.1 Obtención de imágenes

El sistema cuenta como elemento importante a una cámara que permite adquirir imágenes de forma constante, de tal manera que no se genere ningún error en el código al acceder a la cámara, siendo uno de los parámetros importantes el establecer el id de dicho dispositivo ya que no debe cambiar al apagar y encender el prototipo.

2.6.1.1 Selección del dispositivo de captura

Actualmente existen una amplia gama de dispositivos de captura de imágenes, mismos que cuentan con características y funcionalidades que permiten una adquisición óptima de imágenes, como una alta resolución, visión nocturna, visión de infrarroja, visión de espectro completo, etc.

Al emplear un sistema embebido como la Raspberry PI3 como centro de control y monitoreo se cuenta con dos alternativas para el uso de la cámara, una de ellas es un dispositivo con un conector USB y la otra es una cámara que es propia para Raspberry que tiene un conector CSI.

Las ventajas de usar una cámara con conector USB, figura 23-2, es que no se necesita una configuración previa antes de usar el dispositivo simplemente al conectarla a la Raspberry esta le asigna una id que se puede llamar directamente del lenguaje de programación, pero su desventaja es que al ser de tipo USB esta propenso a parámetros como la humedad, calor, polvo que pueden ocasionar inaccesibilidad al dispositivo.



Figura 23-2: Cámara web USB

Fuente: <https://pictures.content4us.com/350px/CMP-WEBCAM21.JPG>

Al utilizar una cámara con conector CSI, figura 24-2, se tiene una mayor confiabilidad y accesibilidad en el dispositivo además su conector tiene una mayor protección a agentes externos, ya que en la Raspberry el sócalo donde se conecta tiene un seguro y sus terminales están protegidos, además al contar con un tamaño mucho menor que las cámaras

convencionales facilita su implementación e instalación. Además, se puede acceder de forma remota al dispositivo y hacer capturas de imágenes mediante un servidor web.



Figura 24-2: Cámara para Raspberry PI 5MP

Fuente: <http://www.devlopez.com/wp-content/uploads/2015/11/camara-raspberry-pi.jpg>

Los parámetros antes mencionados son generalmente más considerables, pero no siempre los más imprescindibles, para este prototipo se debe considerar a la par con la resolución de la imagen que ofrece el dispositivo el ángulo de visión de la cámara y este depende directamente de la distancia focal que tiene la misma.

La distancia focal que es la distancia entre el centro del lente y el punto donde se concentran todo el rayo óptico. Cuando se cuenta con distancia focal pequeña esto se traduce a un ángulo de visión mucho mayor, y viceversa como se muestra en la figura 25-2.

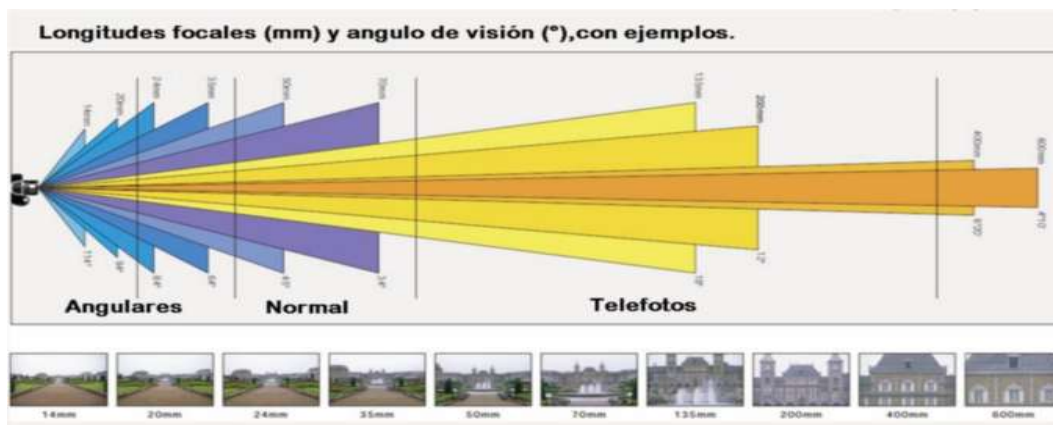


Figura 25-2: Ángulo de visión en función de la distancia focal

Fuente: <http://4.bp.blogspot.com/-RsvrkjtDgLY/ViZO88TnFqI/AAAAAAAAAtzc/2vJRXYtQ4vY/s1600/longitud%2Bfocal.jpg>

Finalmente, en base a las pruebas realizadas y a la información obtenida se decidió usar una cámara con un ángulo que cumplía con las expectativas requeridas. El modelo de cámara es la Genius F100, consta de las siguientes características una resolución de 12 MP: 1920 x 1080, 1280 x 720, 640 x 480 píxeles cuyas dimensiones son 150 x 49 x 48 mm. Al hacer las pruebas

se obtuvieron los resultados esperados ya que no se debe colorar a una altura mayor para captar el campo de visión requerido a comparación con una cámara Raspberry pi, figura 26-2.

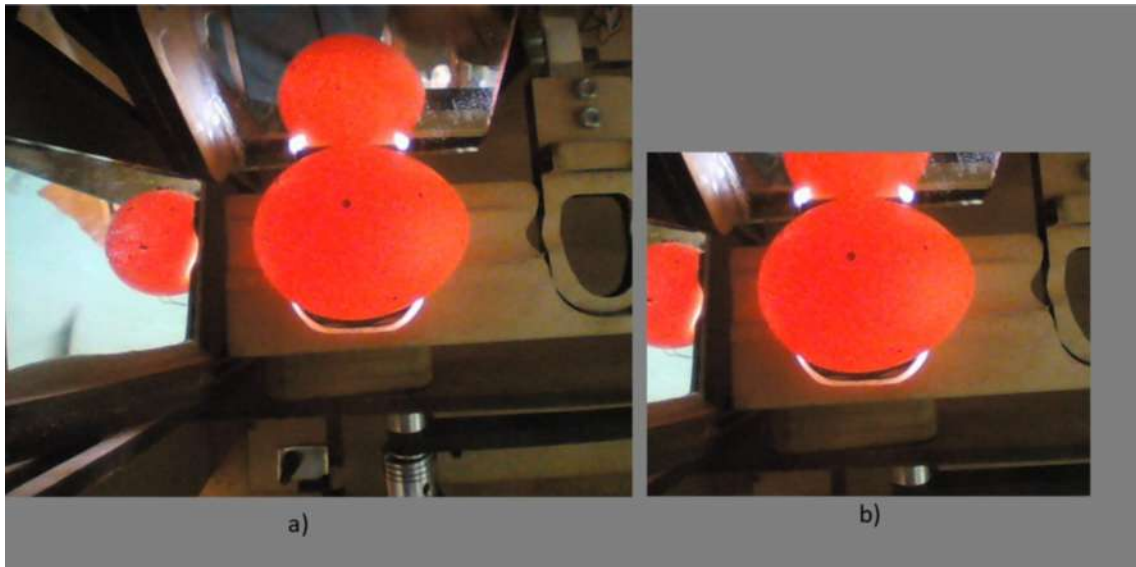


Figura 26-2: a) Cámara Genius F100, b) Cámara Raspberry PI

Realizado por: Paguay, David; Valarezo, Luis; 2018

La distancia a la que se hizo estas pruebas fue de 210 mm se puede evidenciar efectivamente la diferencia entre que existe en ambas cámaras.

2.6.2 Captura de imagen

El programa fue implementado en Python versión 3.4.2 y para hacer uso de técnicas de procesamiento de imágenes y realizar el algoritmo de visión artificial se instaló las librerías de OpenCV versión 3.2 y otras librerías adicionales para facilitar la implementación de algoritmo; como se muestra en la figura 27-2.

```
import numpy as np
import cv2
from matplotlib import pyplot as plt
```

Figura 27-2: Librerías OpenCV importadas

Realizado por: Paguay, David; Valarezo, Luis; 2018

La librería numpy permite llamar a un conjunto de funciones que permiten hacer operaciones entre vectores y matrices como transforma una imagen en un arreglo tanto de una dimensión o de 2 dimensiones. La librería cv2 permite acceder a las funciones de que permiten la adquisición y procesamiento de imágenes, y por último la librería pyplot que permite graficar las imágenes para su respectivo análisis.

Para realizar la captura de las imágenes primero se asigna la id de la cámara esta puede partir desde 0 hasta n valores y se lo hace llamando al comando Videocapture este se almacena en una variable que posteriormente se hace para el almacenamiento de la imagen a procesar en una variable secundaria.

```
captura=cv2.VideoCapture(0)
ret, img=captura.read()
img=cv2.imread("prueba_1.jpg",1)
```

Figura 28-2: Comandos para realizar la captura de imagen

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.6.3 Pre-procesamiento

Una vez obtenida la captura de la imagen y almacenada en una variable el siguiente paso es hacer el procesamiento, esto se hace con el objetivo de transformar la imagen en variables que se puedan procesar con mayor facilidad.

Cuando se hace la captura de la imagen esta en color que puede estar en un formato ya definido como RGB o HSV, el primer formato es el más usado, así como más conocido debido a su facilidad para procesar la imagen, sus siglas significan: Red(Rojo), Blue(Azul), Green(verde), que son colores primarios al conjugarse en combinación que van de 0 a 255 forma una gama de colores.

También se hizo uso del formato HSV por sus siglas Hue (matriz), Saturation (Saturación), Value (Valor), así mismo como el formato anterior la combinación de estos generan un color definido, pero se escogió en primer formato debido a que es conocido y se cuenta con más fuente de información que ayudan a entender su implementación.

```
azul = img[:, :, 0]
verde = img[:, :, 1]
rojo = img[:, :, 2]
img1=img.copy()
img2=img.copy()
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
vsh=cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
v_c=vsh[:, :, 0]
s_c=vsh[:, :, 1]
h_c=vsh[:, :, 2]
rojo1=rojo.copy()
azul1=azul.copy()
verde1=verde.copy()
gray1 = gray.copy()
gray2 = gray.copy()
```

Figura 29-2: Separación de la imagen en componentes primarias

Realizado por: Paguay, David; Valarezo, Luis; 2018

Como se puede ver también se convertido la imagen en una a escalas de grises que no es más que el promedio de los valores de cada una de las componentes RGB con un valor cuyo rango es de 0 a 255 tomando como 0 como un negro, 255 como blanco y valores intermedios como tonalidades de grises. Pero separar la imagen en sus componentes primarias facilito el procesamiento posterior, figura 30-2.

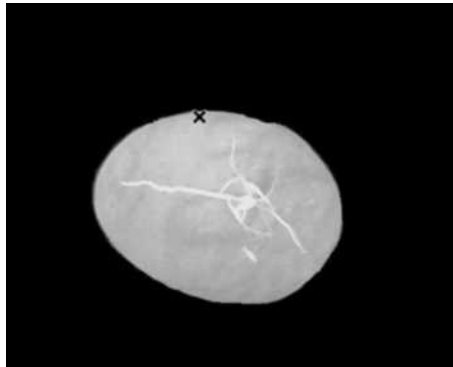


Figura 30-2: Imagen del huevo en componentes primarias RGB

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.6.4 Umbralización

Este método permite convertir la imagen en escalas de grises a una binaria, esta es compuesta por dos únicos valores 0 o 255 tomando el color de negro o blanco, este cambio se controla con un valor llamado umbral que estable un límite para asignar los 2 valores que tomara la imagen de las técnicas de umbralización.

- `cv2.THRESH_BINARY`
- `cv2.THRESH_BINARY_INV`
- `cv2.THRESH_TRUNC`
- `cv2.THRESH_TOZERO`
- `cv2.THRESH_TOZERO_INV`

Para el desarrollo de programa se usó técnicas de umbralización adaptativa que no requieren un valor de umbral fijo se basan en técnicas que aíslan en pequeñas regiones y obtienen unos varios valores y dan mejores resultados en imágenes que tiene una luminosidad estas son las siguientes.

- `cv2.ADAPTIVE_THRESH_MEAN_C`: el valor del umbral es la media del área del vecindario.

- `cv2.ADAPTIVE_THRESH_GAUSSIAN_C`: el valor del umbral es la suma ponderada de los valores del barrio donde los pesos son una ventana gaussiana.

La técnica que dio mejor resultados fue el Método de Otsu que usa técnicas estadísticas específicamente utiliza la varianza, la media de dispersión de datos y calcula un valor de umbral de la dispersión de cada segmento diferentes, calcula el cociente entre ambas variancias y encuentra para la coincidencia del valor sea más eficiente, figura 31-2 .

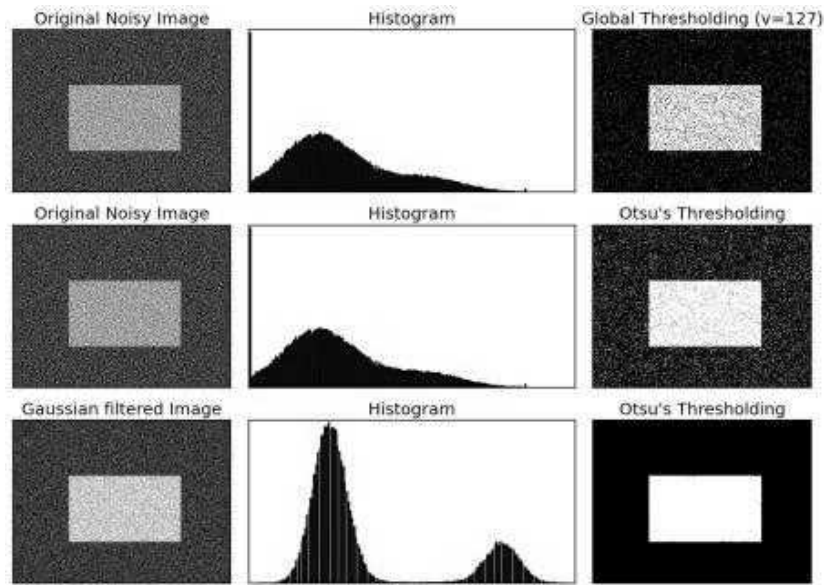


Figura 31-2: Técnicas de umbralización adaptativa

Fuente: <https://docs.opencv.org/3.3.0/otsu.jpg>

El comando que realiza el método de Otsu tiene parámetros ya definidos como la imagen en escalas de grises los valores que ya son establecidos entre 0-255 y el método a usarse, se utilizó una imagen en escalas de grises que es suficiente para obtener un buena umbralización ya que solo se necesita que se binarice toda superficie de la cascara, figura 32-2.

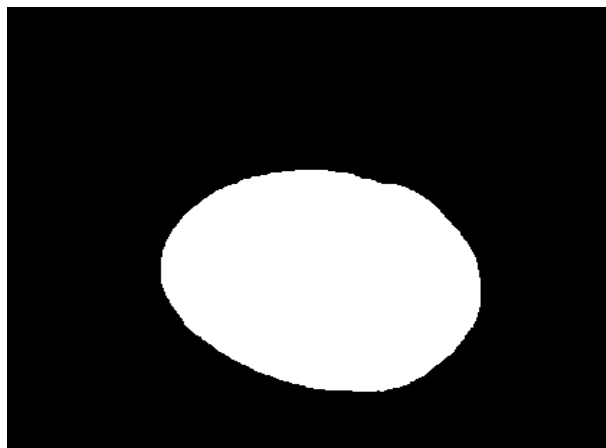


Figura 32-2: Imagen del huevo comando umbralizado de Otsu

Realizado por: Paguay, David; Valarezo, Luis; 2018

Se menciona que al umbralizar la imagen y convertirla en binaria no se separa completamente la superficie de la cascara habrá objetos que tomen el valor de 255 con un color blanco como reflejos de luz o objetos con colores claros para evitar estos inconvenientes hace resaltar el objeto se hace uso de técnicas de iluminación por contrastes, además de hacer una cámara oscura la cual es una estructura pintada de un color que absorba la luz para contrastar el objeto a analizar.

2.6.5 *Detección de contornos*

Para separar el objeto a analizar de lo que haya pasado por la umbralización, se llama a una función que permite convertir los objetos binarizados a contornos, figura 33-2, esto se hace con el objetivo de encontrar el mayor contorno y ese contorno será el de la superficie que se requiere para hacer el análisis.

La función a emplear es `findContours` los parámetros con los que cuentas una variable de entrada que es la imagen y dos métodos `cv2.RETR_TREE`, `cv2.CHAIN_APPROX_SIMPLE` el primero es la recuperación del contorno que obtiene los datos en coordenadas de cada pixel en un valor de 255 tomando el color blanco el segundo es una aproximación de contorno para detectar todos los puntos i, j del contorno exterior.

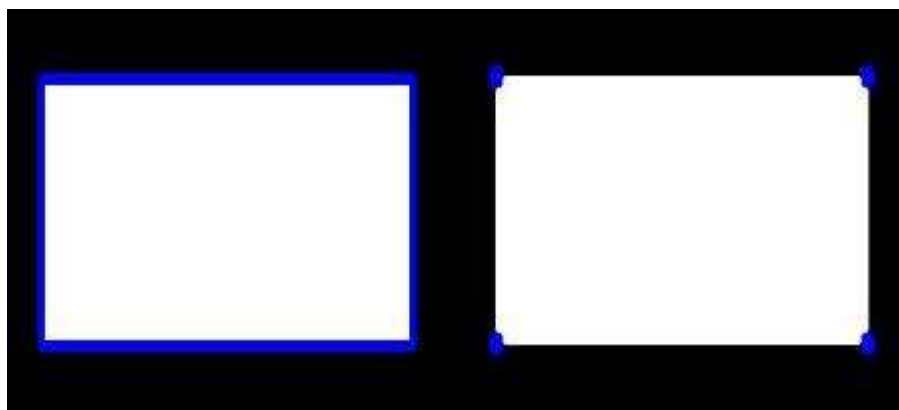


Figura 33-2: Imagen convertida a contorno

Fuente: <https://docs.opencv.org/3.3.1/none.jpg>

Luego de obtener los contornos se hace un ordenamiento ascendiente y se escoge el mayor de todos, ya que esta será la región que se quiere procesar. Este proceso se hace mediante la función llamada `shoret` la cual ordena los contornos antes mencionados o también con la función `max` que encuentra únicamente el contorno mayor, debe tomarse en cuenta que estos valores son coordenadas en ambos ejes de la imagen, pero lo que se necesita es la imagen de la superficie aislada.

Para obtener esto se dibuja las coordenadas obtenidas de los contornos, mediante el uso de una función llamada drawContours sus parámetros son una imagen inicial de color negro como base el contorno el cual está almacenado en una variable y este es mayor el color del contorno por lo general es uno que resalte del color de la imagen base y el ultimo parámetro es ancho de la línea en el que se dibuja el contorno el resultado es una imagen con un contorno dibujado pero sin relleno, para hacer esto se llena el contorno llamando la función fill que rellena el contorno tomando como basa una imagen de color uniforme obteniendo así la cascara del huevo totalmente asilada y binarizada, figura 34-2.

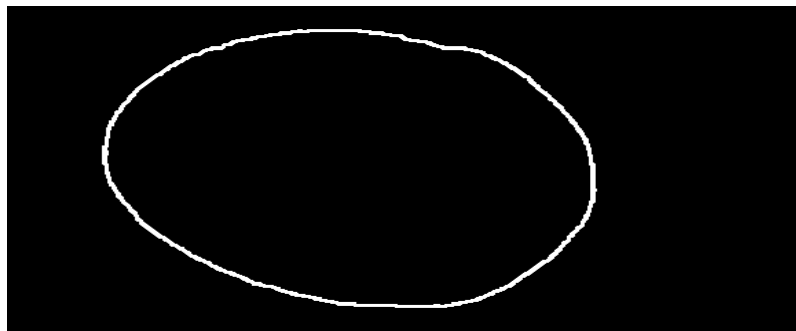


Figura 34-2: Selección de la imagen del huevo con mayor contorno

Realizado por: Paguay, David; Valarezo, Luis; 2018

Al aislar y binarizar la cascara del huevo permite crear una máscara que permitirá aislarla, pero en una imagen en escalas de grises y hasta en una imagen en color.

2.6.6 Creación de una máscara en una imagen de escala de grises

La máscara viene dada por una imagen base que es la imagen que se obtiene hasta el momento que ya es la imagen en escalas de grises la que se va analizar para obtener un buen resultado el comando usado es bitwise operación and entre la imagen en escalas de grises y la imagen binaria resultante como se muestra en la figura 35-2.

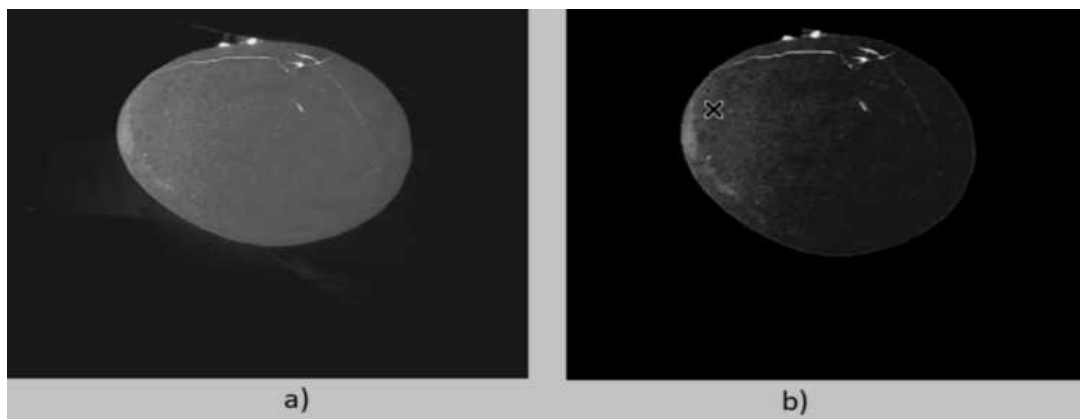


Figura 35-2: Creación de la imagen a partir de una máscara, a) escala de grises b) filtrada

Realizado por: Paguay, David; Valarezo, Luis; 2018

Otra forma de obtener estos datos es usar el valor que adopta como la imagen binarizada el cual es 255 y se condiciona en los datos de cada uno de los canales RGB así tomando los datos que están dentro del área de interés y otra ventaja de esto sería que al aislar los datos se mejora el tiempo de compilación de algoritmo ya que solo se analiza los datos que se necesitan.

Para facilitar el análisis de la imagen esta debe ser convertida a datos de tipo array es decir en un vector o en una matriz con una longitud n esta operación es muy simple y se realiza mediante el comando np. asarray que una función de la librería numpy que permite trabajar con arreglos, así que transforma la imagen en datos que permitan hacer operaciones de manera más rápida como la figura 36-2.

```
# Filtrado RGB y VSH
mascara_binaria=np.asarray(imagen_filtrada)
convesion_rojo_arrray_1= np.asarray(canal_rojo_filtrada)
convesion_azul_arrray_1= np.asarray(canal_azul_filtrada)
convesion_verde_arrray_1= np.asarray(canal_verde_filtrada)
# VSH

convesion_v_arrray_1= np.asarray(canal_v_filtrada)
convesion_s_arrray_1= np.asarray(canal_s_filtrada)
convesion_h_arrray_1= np.asarray(canal_h_filtrada)
```

Figura 36-2: Comando para convertir una imagen a array

Realizado por: Paguay, David; Valarezo, Luis; 2018

Una vez hecho esto se compara con una condicionante cuyo valor será el máximo posible de la imagen binarizada ya pre procesada si los valores toman un valor de 255 (blanco) se almacena el valor en escala de grises en un vector de lo contrario no se almacena los mismo se hará para los tres canales RGB.

2.6.6.1 Técnicas estadísticas en imagen de escala de grises

➤ Media Estadística

Con el vector de los tres canales que componen la imagen se puede hacer uso de un técnico estadístico llamado la media como primer paso. La media en estadística permite encontrar el valor promedio de conjunto de datos, aplicando la ecuación 1-2.

Ecuación 1-2: Ecuación para calcular la media estadística

$$\bar{x} = \frac{\sum fx}{N}$$

Donde:

$\sum fx$ = a media de los valores

N = número de elementos para cálculo de la media

Para comprobar este valor se umbralizó a partir del valor obtenido de la media y ya definido se pudo constatar que con este valor se pudo abarcar un área considerable ya que la función utilizada toma un rango que sea igual o mayor a al valor calculado, figura 37-2.

```
>>> ret, binaria_ordinal=cv2.threshold(gray, valor-_medio, cv2.THRESH_BINARY)
```



Figura 37-2: Umbralización en base a la media aritmética

Realizado por: Paguay, David; Valarezo, Luis; 2018

El cálculo de la media se hace mediante la librería numpy y una función llamada np. mean, fácil de implementar ya que solo requiere un solo conjunto de datos.

➤ **Desviación Estándar**

Otro parámetro importante es el cálculo de desviación estándar, que es una medida de dispersión para valores con una razón de cambio, ecuación 2-2.

Ecuación 2-2: Ecuación para el cálculo de la desviación estándar

$$s = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$$

Donde:

s = desviación estándar

n = número de muestras

x_i = valor determinado de la muestra

\bar{x} = media de la muestra

El comando destinado a esta operación en np.std que también brinda el uso de la librería numpy la operación tanto de la media como la desviación se realizan de forma muy rápida tomando en cuenta que la longitud de cada vector borde los 40000. Para comprar este valor en el procesamiento de imágenes también se hizo el proceso de la binarización en base al valor calculado, figura 38-2.

```
>>>ret, binaria_oginal_std=cv2.threshold(gray, desviación_estandar, cv2.THRESH_BINARY)
```



Figura 38-2: Umbralización en base a la desviación estándar

Realizado por: Paguay, David; Valarezo, Luis; 2018

➤ Coeficiente de Variación Estadística

El objetivo de tener tanto la media y la desviación es calcular el coeficiente de variación. Esta herramienta permite calcular un coeficiente porcentual cuya interpretación muestra que tan dispersos están los datos con respecto a la media si ese valor es menor al 25% los datos son homogéneos es decir que en el conjunto de vectores no hay datos anómalos, si sobre pasa dicho valor existe una diferencia notable entre los valores y la media como la ecuación 3-2.

Ecuación 3-2: Ecuación para calcular la variación estadística

$$CV = \frac{\sigma}{\mu} \cdot 100\%$$

Donde:

CV = coeficiente de variación

σ = desviación estándar

μ = media estadística

Al calcularse el coeficiente de variación para los tres canales no se hace la verificación de cada una de las variables si son menores al 25%, lo que se hizo fue escoger el mayor valor de los tres coeficientes por lo general de los tres canales, el rojo es el que tiene el menor coeficiente a los demás canales.

La imagen seleccionada bajo esta condición es aquella en donde más resalta cualquier tipo de anomalía como rupturas, machas y deformaciones, ya que al usar la técnica de iluminación de reflexión permite ver estas anomalías, figura 39-2.

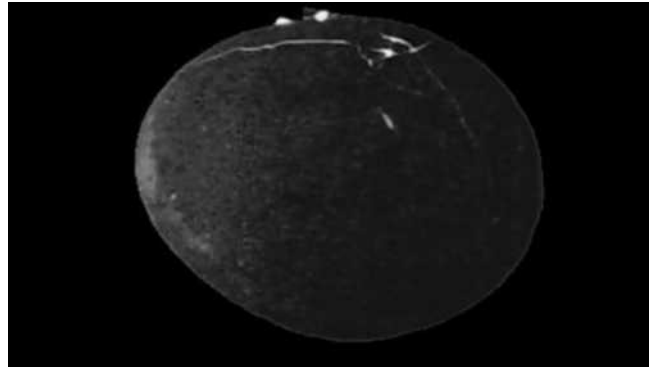


Figura 39-2: Selección del canal con mayor coeficiente de variación

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.6.7 Cálculo de gradiente mediante el operador Sobel

El operador Sobel es un método matemático que permite detectar bordes en una imagen, se basa en obtener el gradiente de la intensidad de una imagen en cada pixel de la imagen y busca una magnitud del mayor cambio posible la dirección y sentido desde un oscuro a un blanco, con la ecuación 4-2.

Ecuación 4-2: Cálculo del operador Sobel

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A$$

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

Donde:

A = imagen a procesar

Gx = matriz de gradiente eje x

Gy = matriz de gradiente eje y

G = magnitud del gradiente

θ = ángulo de gradiente

Los comandos utilizados son el cv2.sobel sus parámetros de entrada son la imagen que va a ser analizada las dimensiones de la matriz la escala y función para detección de bordes para sacar el gradiente en los ejes x, y respectivamente como se muestra en la figura 40-2.

```
# GRADIENTE
# Gradiente-X
grad_x = cv2.Sobel(recorte_1,ddepth,1,0,ksize = 3, scale = scale, delta = delta,borderType = cv2.BORDER_DEFAULT)
# Gradiente-Y
grad_y = cv2.Sobel(recorte_1,ddepth,0,1,ksize = 3, scale = scale, delta = delta, borderType = cv2.BORDER_DEFAULT)
```

Figura 40-2: Obtención de la gradiente en x, y

Realizado por: Paguay, David; Valarezo, Luis; 2018

Para obtener la imagen se utilizan 3 cv2.convertScaleAbs esta función sirve para convertir los valores obtenidos anteriormente en datos de 8 bits que van desde 0 a 255 por lo que valores negativos asumen como el máximo, resultado es una captura de bordes en donde haiga un cambio significativo en los valores de escalas de grises, figura 41-2.

```
#cambio de escala a 8 bits
abs_grad_x = cv2.convertScaleAbs(grad_x)
abs_grad_y = cv2.convertScaleAbs(grad_y)
gradiente_total= cv2.addWeighted(abs_grad_x,0.75,abs_grad_y,0.75,0)
```

Figura 41-2: Obtención de la imagen con detección de bordes

Realizado por: Paguay, David; Valarezo, Luis; 2018

Para asegurarse de detectar exclusivamente de detectar las rupturas se condiciona tanto el gradiente resultante en x y en y para que sea mayor a 96, ya que las rupturas en la superficie de la cascara superan dicho valor y al ser un vector la operación de esta condición es mucho más sencilla. Hay que resaltar que al usar el método de Sobel, la imagen resultante en el gradiente Gx y Gy.

Para obtener la imagen resultante del gradiente se introduce las imágenes anteriormente mencionadas las dos componentes de gradiente el siguiente paso es sumar las dos imágenes, ecuación 5-2, al ser una en escala de grises se puede describir como una función de dos

variables que son la representación de las coordinas de cada pixel y su valor resultante $I(x, y)$ se evalúa dentro de un rango ya definido.

Ecuación 5-2: Cálculo de la gradiente

$$g(x) = (1 - \alpha)f_0(x) + \alpha f_1(x)$$

$$dts = \alpha f_0(x) + \beta f_1(x) + \gamma$$

Donde:

dts = imagen de salida

α = coeficiente de transparencia

$\beta = (1-\alpha)$ complemento de coeficiente

γ = valor de traslado de función asignación, por defecto igual 0

La función para realizar toda esta operación en cv2. `addWeighted` los parámetros de salida son la imagen producto de la suma los parámetros de entrada son las 2 imágenes con sus coeficientes que indican el nivel de transparencia de cada imagen y el ultimo es un escalón que se asigna cero ya que no se quiero alterar demasiado los valores de cada imagen respectivamente, figura 42-2.

```
grad_x = cv2.Sobel(recorte_1,ddepth,1,0,ksize = 3, scale = scale, delta = delta,borderType = cv2.BORDER_DEFAULT)
# Gradiente-Y
grad_y = cv2.Sobel(recorte_1,ddepth,0,1,ksize = 3, scale = scale, delta = delta, borderType = cv2.BORDER_DEFAULT)
grad_x=np.where(grad_x<96,0,grad_x)
grad_y=np.where(grad_y<96,0,grad_y)
#np.savetxt('gradiente',grad_x)
#cambio de escala a 8 bits
abs_grad_x = cv2.convertScaleAbs(grad_x)
abs_grad_y = cv2.convertScaleAbs(grad_y)
gradiente_total= cv2.addWeighted(abs_grad_x,0.3,abs_grad_y,0.3,0)
```

Figura 42-2: Sección para encontrar una imagen mediante método de Sobel

Realizado por: Paguay, David; Valarezo, Luis; 2018

Los resultados obtenidos con esta sección de programa se muestran en la figura 43-2.

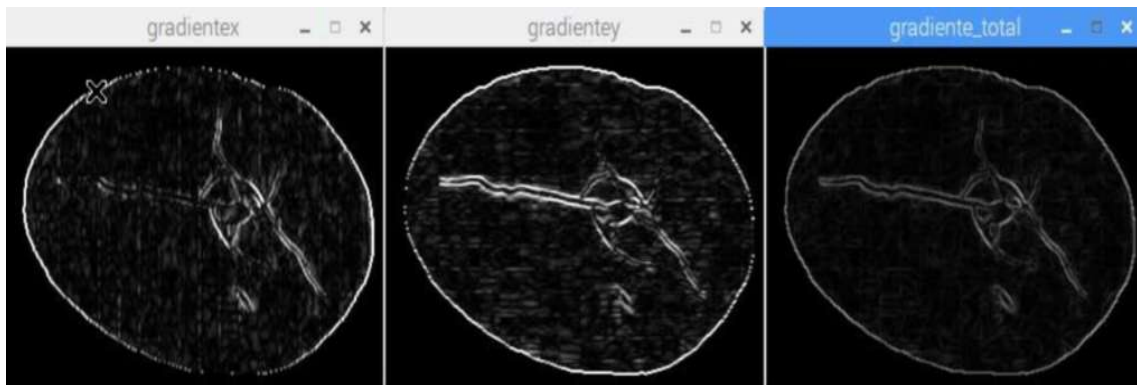


Figura 43-2: a) gradiente en x b) gradiente en y c) gradiente total

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.6.8 *Detección de rupturas en la superficie de la cáscara del huevo*

Después de obtener los bordes en la imagen usando método del gradiente de Sobel el siguiente paso es aislar únicamente la ruptura, para eso se hace una umbralización adaptativa con el método de Otsu y este sirve para hacer una operación and entre esta imagen y la imagen de contorno previamente invertida el objetivo de esto es eliminar el borde de la cáscara con el fondo ya que este provocara una mala interpretación.

Se emplea el comando `cv2.threshold`, siendo los parámetros de entrada la imagen de entrada el rango para la bancarización y el método a utilizar, figura 44-2.

```
ret2,gradiente_binaria = cv2.threshold(gradiente_total,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
```

```
erocion_gradiente=cv2.bitwise_and(gradiente_binaria,imagen_filtrada_p)
```

Figura 44-2: Comandos para procesar la imagen obtenida del gradiente

Realizado por: Paguay, David; Valarezo, Luis; 2018

Los resultados obtenidos, figura 45-2, una vez que se ejecuta la operación and entre las dos imágenes, la binaria gradiente binaria que es resultado de umbralizar la imagen de bordes mediante el uso del gradiente y la imagen con el borde de la superficie de la cascara almacenada en la variable `imagen_filtarda_p`, se muestran en la figura 45-2.

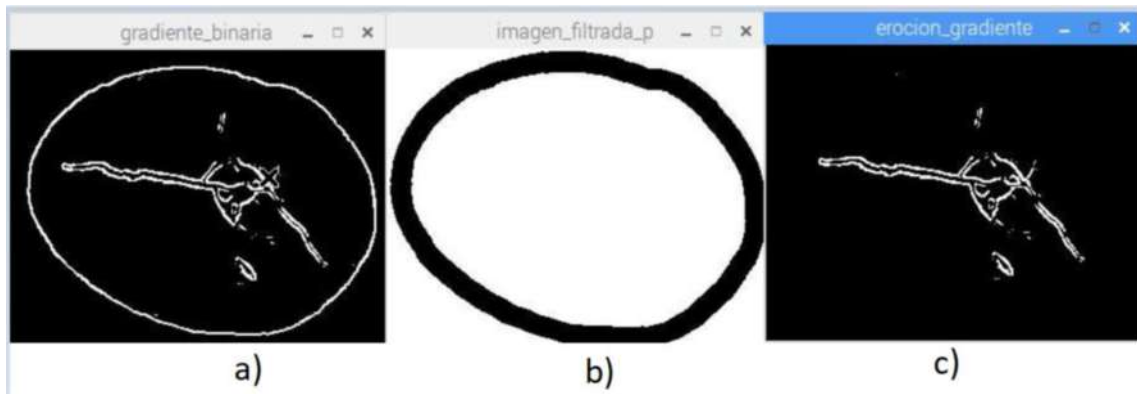


Figura 45-2: a) gradiente umbralizada, b) detección de contorno dilatada, c) imagen resultante

Realizado por: Paguay, David; Valarezo, Luis; 2018

El último paso es implementar un último filtro con una imagen que se umbraliza a un valor que es valor medio de la imagen que puede ser uno de los tres canales dependiendo de cuál sea el valor máximo de coeficiente de variación, esa imagen sirve para eliminar la posible detección de manchas que también pasan con tonalidades claras en el método del gradiente con esto se obtiene la detección de rupturas en la casaca del huevo.

2.6.9 *Detección de manchas en la superficie de la cáscara*

Para la detección de manchas en la superficie, el método más viable y fácil de implementar es el experimental ya que se tomó el numero suficientes de pruebas y se obtuvo un rango de color que abarco tonalidades parcialmente y medianamente oscuras este rango se fijó para los tres canales RGB de una imagen a color.

Para realizar esto se establece un rango almacenado en un vector con tres valores de acuerdo al número de componentes (RGB) de la imagen, figura 46-2.

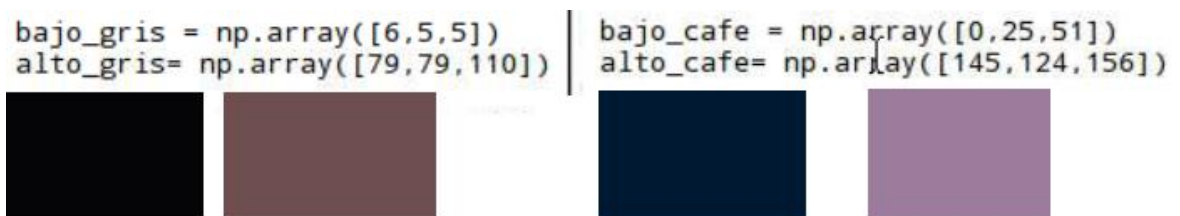


Figura 46-2: Rango de color establecido en base a su vector y color RGB

Realizado por: Paguay, David; Valarezo, Luis; 2018

Posterior a esto se estable una máscara binaria es decir una imagen binaria con un valor umbral establecido en el rango del vector tanto los valores que establecen un rango mínimo y máximo, la función para obtener esto es cv2.Range sus parámetros de entrada son la imagen que se va a

procesar los vectores cada uno con sus tres valores tomando un rango de color definido los resultados de esto son los siguientes, figura 47-2.

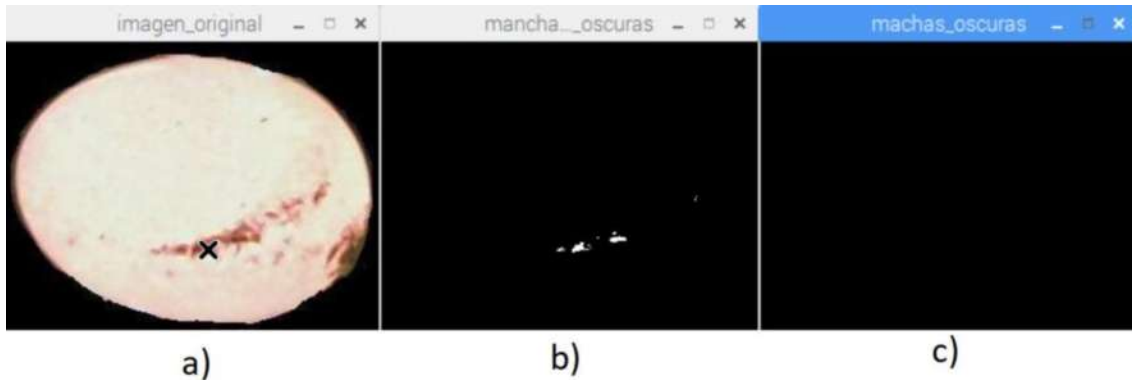


Figura 47-2: Resultados a) original b) tonalidad media oscura c) tonalidad oscuras

Realizado por: Paguay, David; Valarezo, Luis; 2018

Se denota que para las imágenes procedentes de las heces su rango esta en las tonalidades medias oscuras. Pero esto depende de las diferentes tonalidades de la cascara del huevo, que debido al nivel de calcio que es un parámetro muy variante se selección este rango de color.

Pero al detectarse las machas como estas tiene un rango de color ya defino así que sin importar la tonalidad de cada prueba que se haga.

También se debe mencionar que se debe aislar los datos que se hay al borde del huevo con el fondo ya que pueden provocar una decisión errónea, así que se debe hacer una operación and entre la imagen que se obtiene del rango de color con la imagen y la que tiene el contorno de la superficie con una dilatación para eliminar dichos valores.

Después de esto se utilizan comandos de visualización, así como el comando `cv2.imshow` los parámetros de entrada son una descripción en comentario y la imagen que desea mostrar, como se muestra en la figura 48-2.

```
#print(numero_ruptura)
cv2.imshow("imagen_original",recorte_original_rgb)
cv2.imshow("machas_oscuros",mascara_gris_f)
cv2.imshow("manchas_tonalidad_media_oscuros",mascara_cafe_f)
```

Figura 48-2: Comandos para mostrar las imágenes resultantes del procesamiento

Realizado por: Paguay, David; Valarezo, Luis; 2018

El último paso es el cierre de las ventanas anteriormente mencionadas y cortar el bucle para la captura y el procesamiento de las imágenes.

2.6 Programación del algoritmo de comparación de pesos

El algoritmo de comparación de pesos para realizar el proceso de clasificación de los huevos acorde a su tamaño basado en su peso se programa en cada placa Arduino, y sigue las siguientes etapas como se muestra en el gráfico 6-2.



Gráfico 6-2: Etapas del algoritmo de comparación de pesos

Realizado por: Paguay, David; Valarezo, Luis; 2018

La estructura básica del algoritmo es la misma hasta la etapa de toma de decisión, en esta se actúa mediante el criterio de clasificación de pesos normados en la tabla 1-1. El algoritmo inicia con la lectura del peso de los huevos mediante el empleo de galgas extensiométricas, este sensor al medir la deformación que se produce genera lecturas que van desde cero hasta el peso del objeto, haciendo necesario estabilizar las lecturas antes de entrar en la toma de decisión.

Una vez que se tiene una lectura estable se aplica una toma de decisión mediante una estructura if, para así poder verificar que el peso del huevo se encuentre en un rango definido para el peso de huevos pequeños, medianos y grandes. El resultado de la condición es el accionamiento o no del actuador, mismo que es empleado para clasificar los huevos mediante un eslabón, cuando estos estén dentro del rango establecido.

Finalizada la toma de decisión, se establece la comunicación con el controlador Raspberry mediante un pulso que es leído por una Gpio, este pulso sirve de contador y así de esta manera poder ir almacenando los registros de la clasificación y por ende de la producción en la base de datos del prototipo.

2.7 Creación de la base de datos

La librería para generar la base de datos es SQLite, misma que no necesita ser instalada por ya viene por defecto en Python, permite acceder a la base utilizando una variación estándar de SQL. Esta biblioteca tiene una gama de funciones que permiten el almacenamiento interno de datos dentro de la Raspberry. Entre las funciones que permite la librería se tiene crear, ingresar, hacer consultar modificar valores dentro de un registro.

En el gráfico 7-2, se muestra el proceso para la creación de base de datos con los registros de la producción y calidad que arroja el prototipo.



Gráfico 7-2: Proceso para la creación de la base de datos

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.8.1 Creación de la tabla asignando campos necesarios

Para la creación de la base se necesita la librería que permite acceder al conjunto de funciones que permitan su creación, figura 49-2. También se necesita la librería Time para guardar la fecha y hora de cada registro.

```
import sqlite3,datetime

#Conectar a la base de datos
conexion = sqlite3.connect("test.sqlite3")
```

Figura 49-2: Importación de las librerías para la base de datos

Realizado por: Paguay, David; Valarezo, Luis; 2018

Con las funciones ya habilitadas, se procede a crear la base de datos asignándole un nombre con extensión .sqlite, el nombre que se le designo es test. Una vez hecho esto se crea una variable llamada conexión, con esa variable se accede a una función cursor (), esta instancia permite utilizar 2 métodos, el primero es el execute. Para esto se toma la variable almacenada previamente y se llama al método, esto sirve para crear la una tabla en donde se guardarán un registro y tiene los siguientes campos huevos con rupturas, huevos manchados, huevos grandes,

huevos medianos, huevos pequeños, huevos muy pequeños, el campo clave que es el ID de cada uno de los registros, figura 50-2.

```
#Seleccionar el cursor para realizar la consulta
consulta = conexion.cursor()
#String con la consulta para crear la tabla
sql = """CREATE TABLE IF NOT EXISTS test
(id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
H_ROTOS INTEGER NOT NULL,
H_MANCHADOS INTEGER NOT NULL,
H_GRANDE INETGER NOT NULL,
H_MEDIANO INETGER NOT NULL,
H_PEQUEÑO INETGER NOT NULL,
H_MUY_PEQUEÑO INETGER NOT NULL,
fecha DATE NOT NULL)
```

Figura 50-2: Creación de la tabla para asignar los campos

Realizado por: Paguay, David; Valarezo, Luis; 2018

Para garantizar la correcta creación de la tabla en donde se almacenarán los registros se hace una variación mediante una condición if, la variable de control se consulta previamente almacenada en conexión que es donde se crea la base de datos y se llama al método execute, dentro de este método se inserta la variable sql que contiene todos los campos a almacenar.

Después de haber pasado esta condición if satisfactoriamente, lo siguiente es la consulta de la tabla guardar los cambios, guararnos los cambios realizados y cerramos la variable conexión donde esta almacenada la variable para crear la base de datos, como se puede observar en la figura 51-2.

```
#Ejecutamos la consulta
if (consulta.execute(sql)):
    print("Tabla creada con éxito")
else: print("Ha ocurrido un error al crear la tabla")
#Cerramos la consulta
consulta.close()
#Guardamos los cambios
conexion.commit()
#Cerramos la conexión
conexion.close()
```

Figura 51-2: Verificación de la creación de la tabla

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.8.2 *Inserción de datos en los campos asignados*

Para esta operación se realiza los mismos pasos anteriormente mencionados, se establece la variable conexión para abrir la base ya creada, con esa variable se llama al método cursor y esto se almacena en otra variable llamada consulta.

Para almacenar un registro con datos en sus respectivos campos se crea la variable argumentos que en obtiene los valores cada en correcto orden con el campo siendo el ultimo el año, mes, día, hora minutos y segundos.

Para indexar los valores también se necesitan llamar a los campos respectivos dentro de una variable sql estas se asignan con comentarios anteponiendo el comando INSERT INTO test. Al finalizar esto se hace una verificación para asegurarse de los datos se han guardado de forma de correcta con el comando if se llama al método execute dentro de este se insertan las dos variables sql donde se detallan el orden de los campos y argumentos que son los datos que se almacenas en esos campos, figura 52-2.

```
#Valor de los argumentos
argumentos = (0.0,0.0,0.0,0.0,datetime.datetime.now())

#consulta SQL con argumentos ?, ?, ?, ?
sql = """INSERT INTO test(H_ROTOS,H_MANCHADOS,H_GRANDE,H_MEDIANO,H_PEQUEÑO,H_MUY_PEQUEÑO, fecha)
VALUES (?, ?, ?, ?, ?, ?, ?)"""
#Realizar la consulta
if (consulta.execute(sql, argumentos)):
    print("Registro guardado con éxito")
else: print("Ha ocurrido un error al guardar los datos")
```

Figura 52-2: Almacenamiento de los datos

Realizado por: Paguay, David; Valarezo, Luis; 2018

Como se mencionó anteriormente también se debe cerrar la variable de consulta se guardar cambios y se cierra la conexión con la base de datos.

2.8.3 Visualización de datos

Para la visualización de los datos almacenados se efectúan el mismo número de pasos anteriormente mencionados, se llama la variable consulta con el método execute en donde se detallan el número de campos que se crearon.

Para imprimir los datos se emplea el comando print de esa forma se imprimen todos los registros guardados; otra forma de hacerlo es usando una sentencia de programación for que sirve para generar un bucle controlado por la variable consulta cuya inicialización empieza con la variable i, definida en un rango con saltos de escalonados de una unidad, figura 53-2.

```

consulta.execute('SELECT id, H_ROTOS,H_DEFORMADOS,H_MANCHADOS,H_SANGRE,H_PEQUENO,H_CALCIO,fecha from test')
print(consulta)
for i in consulta:
| a=1

print("ID=",i[0])
print("H_ROTOS",i[1])
print("H_DEFORMADOS",i[2])
print("H_MANCHADOS",i[3])
print("H_SANGRE",i[4])
print("H_PEQUENO",i[5])
print("H_CALCIO",i[6])
print("fecha",i[7],"\n")
h_rotos=i[1]

conexion.close()

```

Figura 53-2: Comando para la visualización de los datos

Realizado por: Paguay, David; Valarezo, Luis; 2018

El resultado obtenido muestra primero la impresión de todos los registros almacenados y segundo la impresión detallando el campo donde se almacena el registro, como se puede observar en la figura 54-2.

```

>>>
**** Programa para insertar datos en bases de datos sqlite3 ****
(1, 0, 0, 0, 0, 0, 0, '2018-02-27')
(2, 0, 0, 0, 0, 0, 0, '2018-02-27')
(3, 0, 0, 0, 0, 0, 0, '2018-02-27 13:06:42.812997')
(4, 0, 0, 0, 0, 0, 0, '2018-02-27 13:06:47.071934')
(5, 0, 0, 0, 0, 0, 0, '2018-02-27 13:06:51.746150')
(6, 0, 0, 0, 0, 0, 0, '2018-02-27 13:06:55.102244')
(7, 0, 0, 0, 0, 0, 0, '2018-02-27 13:06:59.041716')
(8, 0, 0, 0, 0, 0, 0, '2018-02-27 13:07:04.517233')
(9, 0, 0, 0, 0, 0, 0, '2018-02-27 13:40:04.808700')
ID= 9
H_ROTOS 0
H_DEFORMADOS 0
H_MANCHADOS 0
H_SANGRE 0
H_PEQUENO 0
H_CALCIO 0
fecha 2018-02-27 13:40:04.808700

```

Figura 54-2: Visualización de los datos almacenados en la tabla

Realizado por: Paguay, David; Valarezo, Luis; 2018

2.8 Diseño e implementación de la interfaz gráfica

Para la interfaz HMI se desarrolla una interfaz gráfica programada en Python. La librería que permite hacer esto tiene como nombre tkinter, figura 55-2, esta librería ofrece muchas opciones para la programación orientada a objetos como insertar texto, botones, slider, gráficos y menús.

Al llamar a esta librería se debe crear un constructor donde estarán todos los objetos que se inserten, así como la creación de la ventana sus respectivas dimensiones, así como el título de la ventana principal.

```
import tkinter
from tkinter import |
ventana = Tk()
ventana.geometry("1000x1000+0+0")
ventana.title("PROTIPO CLASIFICADOR DE HUEVO")
```

Figura 55-2: Importación de la librería para la interfaz gráfica

Realizado por: Paguay, David; Valarezo, Luis; 2018

Después de esto se la creación de las funciones que formaran parte del programa como la de la ventana principal donde se coloca la dimensiones el color de fondo dentro de esta ventana estarán datos que se mostraran, figura 56-2.

```
def informacion():
    ventanaInfo=Tk()
    ventanaInfo.geometry("350x300+100+100")
    ventanaInfo.title("Informacion del Prototipo ")
    lblInfo=Label(ventanaInfo, text="ESTE PROTIPO HACE LA CLASIFICACION EN BASE A AL VIOSN ARTIFICIAL", relief="sunken", borderwidth=3, font=("Agency FB", 10))
    ventanaInfo.mainloop()

def leer():
    ventanaLeer=Tk()
    ventanaLeer.geometry("350x300+100+100")
    ventanaLeer.title("Guia de Usuario")
    lblLeer=Label(ventanaLeer, text="ldfkldkfldfk", relief="sunken", borderwidth=3, font=("Agency FB", 10)).place(x=10, y=10)
    ventanaLeer.mainloop()
```

Figura 56-2: Asignación de librerías

Realizado por: Paguay, David; Valarezo, Luis; 2018

Después de asignar las funciones se inserta datos como el título, la universidad nombre de la facultad y nombre de los autores esto se lo realiza con el comando label sus parámetros son el constructor de la ventana, el texto que se desea ingresa, el color del texto, color del fondo la posición, cabe recalcar que la posición debe estar dentro del rango de la ventana principal sus coordinas.

También se debe resaltar donde se va a ingresar estos textos en la ventana principal o en una ventana que se abre con el menú principal el código se insertan antes de ingresar los textos, figura 57-2.

```

# para que se emuestre la barra de menus en la ventana
ventana.config(menu=barraMenu)
boton_inicio=tkinter.Button(ventana,text="INICIO",bg="blue")
titulo=Label(ventana,text="ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO",bg="blue",font=("Algerian",22,"bold"),fg="white").place(x=200,y=200)
titulo1=Label(ventana,text="FACULTAD DE INFORMÁTICA Y ELECTRÓNICA",bg="blue",font=("Algerian",16,"bold"),fg="aquamarine").place(x=200,y=210)
titulo1a=Label(ventana,text="ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES",bg="blue",font=("Algerian",12,"bold"),fg="aquamarine").place(x=200,y=220)
titulo2=Label(ventana,text="DETECCION DE LA IMPERFECCIONES DE LA CASCARA ",bg="blue",font=("helvetica",14,"bold"),fg="black").place(x=200,y=230)
titulo4=Label(ventana,text="AUTORES: ",bg="blue",font=("helvetica",10,"bold"),fg="black").place(x=200,y=240)
titulo5=Label(ventana,text="VALAREZO RODRIGUEZ LUIS ",bg="green",font=("helvetica cursiva",8),fg="black").place(x=200,y=250)
titulo6=Label(ventana,text="PAGUAY DONOSO DAVID ",bg="green",font=("helvetica cursiva",8),fg="black").place(x=200,y=260)

```

Figura 57-2: Asignación de texto en la ventana

Realizado por: Paguay, David; Valarezo, Luis; 2018

El prototipo se controla mediante la asignación de 2 botones inicio y paro para arrancar la secuencia del motor paso a paso y para cuando se produzca algún error. También esta enlazada con la base datos para poder visualizar los registros de producción, figura 58-2.

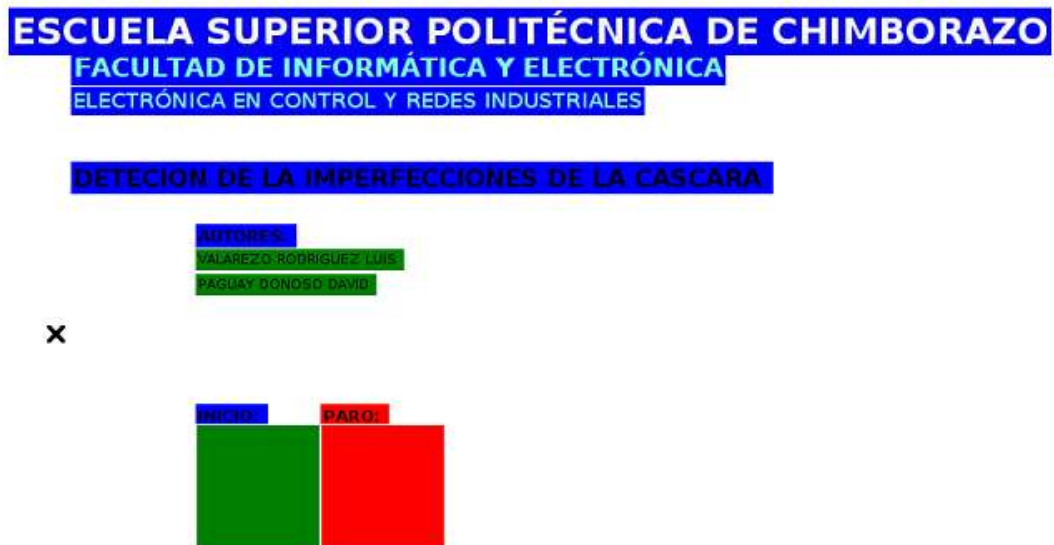


Figura 58-2: Interfaz gráfica del prototipo

Realizado por: Paguay, David; Valarezo, Luis; 2018

La programación se efectúa de una forma simple y sencilla, ya que al compilar con la programación basada en objetos su tiempo de compilación se adiciona al tiempo de la demás programación.

CAPITULO III

3. MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS

En este capítulo se presentan las pruebas y resultados obtenidos del prototipo clasificador de huevos de gallina. Para realizar las pruebas se procede a recaudar información durante cinco días en la granja avícola San Fernando, con el registro de producción sacamos una muestra que nos servirá para comprobar el funcionamiento de nuestro prototipo; se realiza una comparación entre la clasificación de forma manual y con el prototipo, así como un análisis de precios con otras clasificadoras existentes en el mercado.

3.1 Análisis y comparación de los resultados obtenidos del prototipo

En este apartado se realiza una evaluación de la forma que se clasifica el huevo de gallina de forma manual, el procedimiento se realiza teniendo en cuenta la cantidad de cubetas diarias que salen de la producción, posterior a esto se procede a clasificar pesando cada uno de los huevos para separarlos según su peso, a medida que se va realizando el proceso se separa cualquier tipo de defecto que exista en la cáscara y que pueda influenciar en la calidad del producto.

La duración de la clasificación varía según el número de cubetas que salen por día y de la cantidad de personas que ayuden en el proceso, para recaudar información y obtener los resultados una persona especialista y preparada se encarga de verificar el peso y estado de los huevos separándolos por sus tamaños según las normas establecidas, para calcular el número de muestras que debemos tomar aplicamos la siguiente fórmula estadística para el tamaño de una muestra para datos globales.

Ecuación 1-3: Ecuación para calcular el tamaño muestral

$$n = \frac{Z^2 N p q}{e^2 (N - 1) + Z^2 p q}$$

Donde:

N = Población

Z = Nivel de confianza

n = Tamaño de la muestra

p = Probabilidad de acierto

q = Probabilidad de error

e = Error muestral deseado

Para el nivel de confianza se realiza las pruebas con un 95% que corresponde a $Z=1.96$, y con el 5% de error muestral con estos datos se obtiene la siguiente tabla 1-3 de los cinco días de prueba.

Tabla 1-3: Datos de las muestras

	Día 1	Día 2	Día 3	Día 4	Día 5
Total de cubetas	210	215	209	213	211
Total de huevos de producción	6300	6450	6270	6390	6330
Total de huevos de muestra	362	363	362	362	362

Fuente: Paguay, David; Valarezo, Luis; 2018

Realizado por: Paguay, David; Valarezo, Luis; 2018

Teniendo las muestras se realiza el proceso de clasificación de forma manual, conjuntamente con una balanza digital la persona especialista separa los huevos acordes a sus características, los resultados obtenidos se pueden muestran en la tabla 2-3.

Tabla 2-3: Datos obtenidos de la clasificación de forma manual

	Día 1	Día 2	Día 3	Día 4	Día 5
Huevos grandes	257	265	264	265	258
Huevos medianos	58	54	46	51	59
Huevos pequeños	23	29	25	22	17
Huevos con rupturas	14	8	17	11	21
Huevos con heces	4	3	7	9	5
Huevos con sangre	6	4	3	4	2

Fuente: Paguay, David; Valarezo, Luis; 2018

Realizado por: Paguay, David; Valarezo, Luis; 2018

Con la implementación del prototipo, para el mismo número de muestras clasificadas de forma manual se obtuvieron los siguientes resultados expuestos en la tabla 3-3.

Tabla 3-3: Datos obtenidos de la clasificación con el prototipo

	Día 1	Día 2	Día 3	Día 4	Día 5
Huevos grandes	258	266	265	268	259
Huevos medianos	62	56	47	54	62
Huevos pequeños	25	30	27	23	24
Huevos con rupturas	12	7	15	8	18
Huevos con heces	3	3	5	7	3
Huevos con sangre	4	3	3	2	2

Fuente: Paguay, David; Valarezo, Luis; 2018

Realizado por: Paguay, David; Valarezo, Luis; 2018

3.2.1 Efectividad del algoritmo de visión artificial

Con los datos obtenidos de forma manual y mediante el prototipo, se realizó un análisis de la efectividad de la aplicación del algoritmo de visión para la detección de imperfecciones en la cáscara del huevo de gallina. Los resultados obtenidos se presentan en la tabla 4-3.

Tabla 4-3: Efectividad del algoritmo de visión

Efectividad %	Día 1	Día 2	Día 3	Día 4	Día 5	Media
Huevos con rupturas	85.71 %	87.5 %	88.23 %	72.73 %	85.71 %	83.97 %
Huevos con heces	75 %	100 %	71.43 %	77.78 %	60 %	76.84 %
Huevos con sangre	66.67 %	75 %	100 %	50 %	100 %	78.33 %
					Promedio	79.71%

Fuente: Paguay, David; Valarezo, Luis; 2018

Realizado por: Paguay, David; Valarezo, Luis; 2018

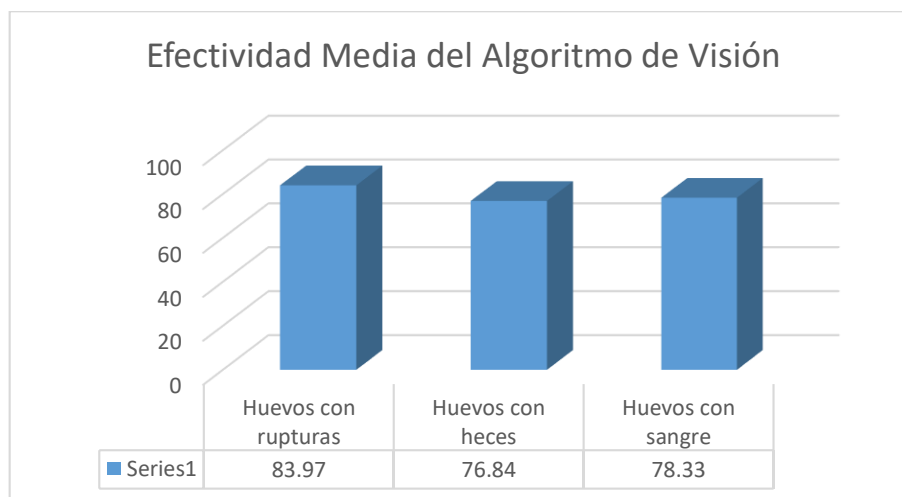


Gráfico 1-3: Efectividad media del algoritmo de visión






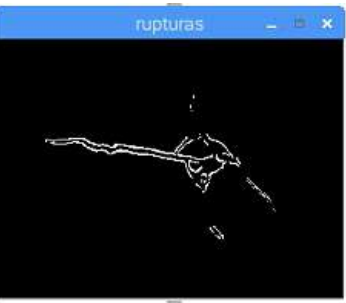


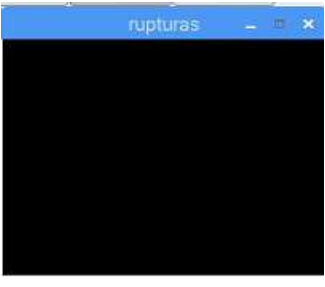
Realizado por: Paguay, David; Valarezo, Luis; 2018

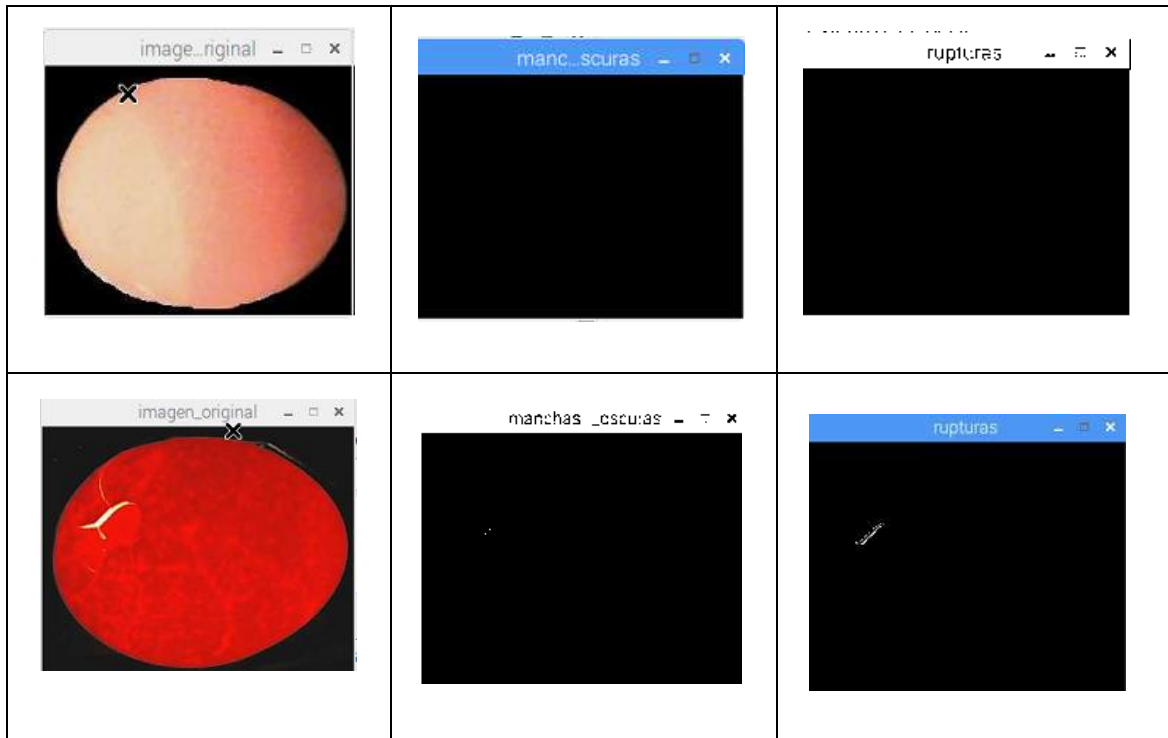
Mediante el cálculo del promedio de las efectividades medias resultantes de la aplicación del algoritmo de visión se obtuvo que el prototipo tiene una efectividad promedio total de 79.71%, para encontrar huevos con anomalías en la superficie de la cáscara. Se puede evidenciar un mejor desempeño para encontrar fallos producidos por rupturas en la cáscara del huevo.

3.2.2 Respuesta del algoritmo de visión

Los resultados se obtuvieron en base a las imágenes que fueron seleccionadas de una muestra más grande, es decir las imágenes que contengan imperfecciones en la cáscara más importantes como machas y rupturas. Los resultados se muestran en la tabla 5-3.

Tabla 5-3: Resultados obtenidos en rupturas y manchas de la superficie

Imagen en color	Detección de manchas	Detección de rupturas
		
		
		



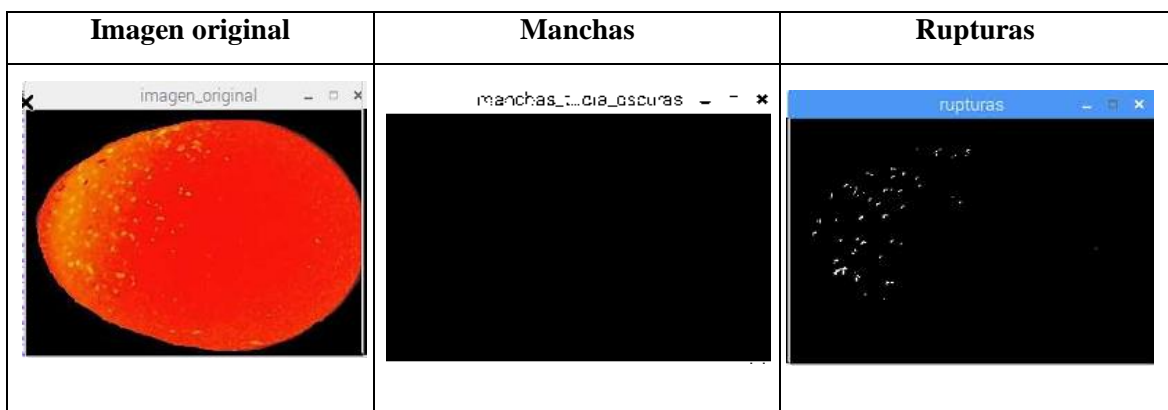
Fuente: Paguay, David; Valarezo, Luis; 2018

Realizado por: Paguay, David; Valarezo, Luis; 2018

El resultado se muestra como una imagen binaria cuyo valor esta entre 0 y 255 tanto para la imagen de machas como para la imagen de rupturas.

Un caso excepcional es al probar una ruptura en un punto en que la cámara lo enfocaba directamente sino al lado opuesto la ruptura genero una grieta de luz que hizo que el algoritmo la tome efectivamente como una ruptura. Es decir, en caso de darse esta situación el algoritmo responde de manera eficiente como se muestra en la tabla 6-3.

Tabla 6-3: Prueba de ruptura en la superficie que no puede captar la cámara



Fuente: Paguay, David; Valarezo, Luis; 2018

Realizado por: Paguay, David; Valarezo, Luis; 2018

3.2.3 Comparación del tiempo empleado en forma manual vs el prototipo

Con el mismo número de muestras con las que se obtuvo la eficiencia del algoritmo de visión, y haciendo uso de un cronometro se registró el tiempo empleado para la clasificación de los huevos, en forma manual y mediante el prototipo obteniendo los tiempos expuestos en la tabla 7-3.

Tabla 7-3: Tiempo manual vs tiempo del prototipo

Tiempo	Día 1	Día 2	Día 3	Día 4	Día 5	Promedio
Manual	38 min	42 min	39 min	41 min	40 min	40 min
Prototipo	17 min	15 min	17 min	14 min	17 min	16 min

Fuente: Paguay, David; Valarezo, Luis; 2018

Realizado por: Paguay, David; Valarezo, Luis; 2018

Con los resultados del tiempo promedio tanto en forma manual y con el empleo del prototipo, se calculó el porcentaje de diferencia entre los mismos aplicando la ecuación 2-3:

Ecuación 2-3: Ecuación para calcular el porcentaje de diferencia

$$\%diferencia = \frac{\text{valor anterior} - \text{valor actual}}{\text{valor anterior}} * 100$$

$$\%diferencia = \frac{40 \text{ min} - 16 \text{ min}}{40 \text{ min}} * 100$$

$$\%diferencia = 60\%$$

El resultado obtenido fue un ahorro en el tiempo del 60%, en la realización del proceso de separación y clasificación de huevos de gallina basados en las imperfecciones de la cáscara y el peso, mediante la implementación del prototipo.

3.2.4 Análisis de costos del prototipo implementado

Tabla 8-2: Costos de la implementación del sistema

MATERIA PRIMA	CANTIDAD	V. UNITARIO	V. TOTAL
Raspberry PI3	1	80,00	80,00
Pantalla 3.5 pulg para Raspberry	1	35,00	35,00
Arduino Pro Micro	3	9,00	27,00
Galgas extensiométricas 5 Kg	3	7,00	21,00
Módulo HX-711	3	4,00	12,00
Qrd	3	5,00	15,00

Motor de pasos Nema 23, 12 Kg-f	1	25,00	25,00
Driver A4988	1	4,50	4,50
Led de alto brillo	1	1,50	1,50
Servomotores de 9 Kg	5	13,00	65,00
Cámara web Genius F100	1	75,00	75,00
Fuente de 800 W	1	15,00	15,00
Plancha de MDF 5.5 mm	1	20,00	20,00
Espejos	1	3,00	3,00
Acoples y banda dentada	1	20,00	20,00
Pernos con tornillos	25	0,30	7,50
Pegamento	1	10,00	10,00
Pintura Látex	10	3,00	30,00
Otros	1	20,00	20,00
TOTAL:			486,50
MANO DE OBRA			
Impresión 3D	1	40,00	40,00
Corte MDF	1	50,00	50,00
TOTAL:			90,00
COSTO TOTAL IMPLEMENTACIÓN DEL SISTEMA:			576,50

Fuente: Paguay, David; Valarezo, Luis; 2018

Realizado por: Paguay, David; Valarezo, Luis; 2018

CONCLUSIONES

- Con el prototipo implementado la automatización de la clasificación de huevos se pudo comparar con la de otros clasificadores que realizan el mismo trabajo, el resultado de la investigación fue que no hay clasificadores que ocupen visión artificial y guarden sus registros en una base de datos a un bajo costo.
- El prototipo clasificador de huevo de gallina utiliza la técnica de ovoscopia en conjunto a un arreglo de espejos y las paredes pintadas de color negro, esto permitió realizar las pruebas y comprobar que se obtiene una mayor cobertura de las caras del huevo permitiendo visualizar las anomalías de la cáscara del huevo.
- Con el algoritmo de comparación de pesos programado se obtuvo un proceso de separación más eficiente logrando reducir el tiempo empleado para el proceso en forma manual, con el algoritmo de visión artificial se incrementa la calidad del producto separando los huevos que presenten anomalías que pueden llegar a causar daño en los consumidores.
- La incorporación de una base de datos en el prototipo, se presenta como una herramienta de ayuda para el productor ya que se registra la información de producción y control de calidad del producto, información que puede ser sujeta a un análisis.
- Los sistemas diseñados y posteriormente implementados en el prototipo facilitan el proceso de clasificación del huevo de gallina. El sistema mecánico permite el transporte del producto durante todo el proceso sin que sufra daño, el sistema electrónico permite el control de los motores y permite el control de la clasificación de los huevos.
- Las pruebas realizadas en el prototipo obtuvieron un porcentaje del 100% en la separación de huevos por peso, en cuanto que el algoritmo de visión posee una efectividad promedio total del 79,71% para encontrar huevos con anomalías debido a factores externos.
- La comparación de resultados entre el proceso en forma manual y con el uso del prototipo presenta una reducción promedio del tiempo del 60% para realizar el proceso de clasificación.

RECOMENDACIONES

- Se debe tener cuidado de poner algún peso sobre las galgas antes de que comience el proceso de clasificación para que la calibración no resulte afectada en el algoritmo de comparación de pesos.
- Para poder clasificar los huevos es recomendable ponerlos desde el inicio de la rampa para evitar que sufran daños al momento de descender y choquen entre si evitando rupturas ocasionadas por la máquina.
- Tener cuidado con los mecanismos que transportan los huevos al momento de realizar el mantenimiento evitando que la banda se descuadre y provoque vibraciones provocando caídas.
- Para tener respuestas más rápidas en el procesamiento de imágenes se podría reemplazar la Raspberry por un computador Core I7 con una mayor memoria RAM de por lo menos 4Gb, mejorando la velocidad del algoritmo y aumentando la capacidad de almacenaje de datos.
- Se debe tener en cuenta el voltaje con el que funcionan los GPIO que es de 3.3V de la Raspberry para evitar daños en el procesador, cuando se realiza trabajos con otras placas de desarrollo como Arduino que trabajan con 5V.

BIBLIOGRAFÍA

ALEREIMONDO. *Procesamiento global de imágenes.* [En línea] 2005.

[Citado el: 12 de febrero de 2018.] Disponible en:

<http://alereimondo.no-ip.org/OpenCV/uploads/41/tema2.pdf>.

Imágenes binarias. [En línea] 2009. [Citado el: 20 de diciembre de 2017.]

Disponible en: <http://alojamientos.us.es/gtocoma/pid/tema5-1.pdf>.

AUROVA. *Detección de esquinas y vértices.* [En línea] 2002. [Citado el: 02 de enero de 2018.]

Disponible en: <http://www.aurova.ua.es/previo/tavarca/ponencias/esquinas.pdf>.

Captura de imagen. [En línea] 2011. [Citado el: 12 de febrero de 2018.] Disponible en:

<https://www.fayerwayer.com/2011/05/captura-de-imagen-en-una-camara-digital-chwonders/>.

Clasificadora de huevos. [En línea] 2010. [Citado el: 05 de diciembre de 2017.] Disponible en:

<http://www.clasificadoradehuevos.com/>.

Composición y estructura del huevo. [En línea] 2012. [Citado el: 27 de enero de 2018.]

Disponible en: <https://huevoselmajadal.com/2012/05/24/composicion-y-estructura-del-huevo/>.

Defectos de la cáscara del huevo. [En línea] 2015. [Citado el: 09 de enero de 2018.]

Disponible en: <http://www.veternariadigital.com/articulos/defectos-de-la-cascara-del-huevo>.

Defectos en la calidad del huevo. [En línea] 2016. [Citado el: 06 de febrero de 2018.]

Disponible en:

<http://gallosdeguerra.net/crianza/defectos-en-la-calidad-de-los-huevos-de-las-gallinas-de-raza/>.

Diferencias entre la Raspberry PI 3. [En línea] 2016. [Citado el: 23 de diciembre de 2017.]

Disponible en: <https://tuelectronica.es/que-diferencias-tiene-raspberry-pi-3-b/>.

EL TELÉGRAFO. *Datos de producción de huevos.* [En línea] 2017.

[Citado el: 23 de enero de 2018.] Disponible en:

<https://www.eltelegrafo.com.ec/noticias/economia/8/el-ecuatoriano-consume-165-huevos-al-ano>.

ELAI. *Visión artificial.* [En línea] 2005. [Citado el: 09 de febrero de 2018.] Disponible en:
<http://www.elai.upm.es/webantigua/spain/Asignaturas/Robotica/ApuntesVA/cap1IntroVA.pdf>.

ELAI. *La naturaleza de la luz.* [En línea] 2005. [Citado el: 11 de febrero de 2018.]
Disponible en:
http://www.elai.upm.es/webantigua/spain/Asignaturas/MIP_VisionArtificial/ApuntesVA/cap2VAFormImagv1.pdf.

GALLEGOS LÓPEZ, Cristian Fernando y JARRÍN VIVAR, Jorge Darío. *Diseño e implementación de un módulo didáctico para clasificación, empaquetado e inspección de huevos aplicando plc.* *dspace.esPOCH.edu.ec.* [En línea] (Tesis). (Pregrado) Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, 2012 [Citado el: 22 de enero de 2018.]
Disponible en: <http://dspace.esPOCH.edu.ec/bitstream/123456789/2852/1/108T0031.pdf>.

HETPRO. *OpenCV encontrar contornos y áreas.* [En línea] 2018.
[Citado el: 01 de marzo de 2018.] Disponible en:
<https://hetpro-store.com/TUTORIALES/opencv-findcontours/>.

Imágenes en escala de grises. [En línea] 2015. [Citado el: 28 de diciembre de 2017.]
Disponible en:
http://programacion.net/articulo/mostrar_imagenes_en_escala_de_grises_con_los_filtros_css_1134.

LUNADEI, L, y otros. *oa.upm.es.* [En línea] 2011. [Citado el: 25 de enero de 2018.]
Disponible en: http://oa.upm.es/9709/1/CIGR2011_Full_Paper_uova.pdf.

Motor de pasos. [En línea] 2006. [Citado el: 18 de diciembre de 2017.] Disponible en:
http://robots-argentina.com.ar/MotorPP_basico.htm.

NIESER. *Prima 2000.* [En línea] 2017. [Citado el: 06 de enero de 2018.] Disponible en:
<http://www.nieser.com.ar/es/equipamiento/procesamiento-de-huevos/clasificacion/clasificadoras-de-huevos/67-prima-2000-es>.

NORMALIZACION. *Clasificación de huevos por su peso.* [En línea] 2015.
[Citado el: 08 de enero de 2018.] Disponible en:
http://www.normalizacion.gob.ec/wp-content/uploads/downloads/2015/07/nte_inen_2877.pdf.

OMEGA. *Galgas extensiométricas.* [En línea] 2013. [Citado el: 24 de enero de 2018.]
Disponible en: <https://es.omega.com/prodinfo/galgas-extensiometricas.html>.

Partes principales de una cámara digital. [En línea] 2008. [Citado el: 13 de febrero de 2018.]
Disponible en: http://www.asifunciona.com/practico/pr_camara_digital/camdig_4.htm.

Phyton. [En línea] 2013. [Citado el: 19 de diciembre de 2017.] Disponible en:
<https://es.gizmodo.com/que-lenguaje-de-programacion-deberia-aprender-para-emp-1479554075>.

PONCE CRUZ, Pedro. *Inteligencia Artificial con aplicaciones a la ingeniería.*
Mexico: Alfaomega, 2010. ISBN 978-607-7854-83-8, pp. 1.

Producción mundial de huevos. [En línea] 2016. [Citado el: 20 de enero de 2018.]
Disponible en:
<http://seleccionesavicolas.com/avicultura/2016/01/la-produccion-mundial-de-huevos-situacion-actual-y-previsiones>.

Programación arduino. [En línea] 2017. [Citado el: 18 de febrero de 2018.] Disponible en:
<https://aprendiendoarduino.wordpress.com/2017/01/23/programacion-arduino-5/>.

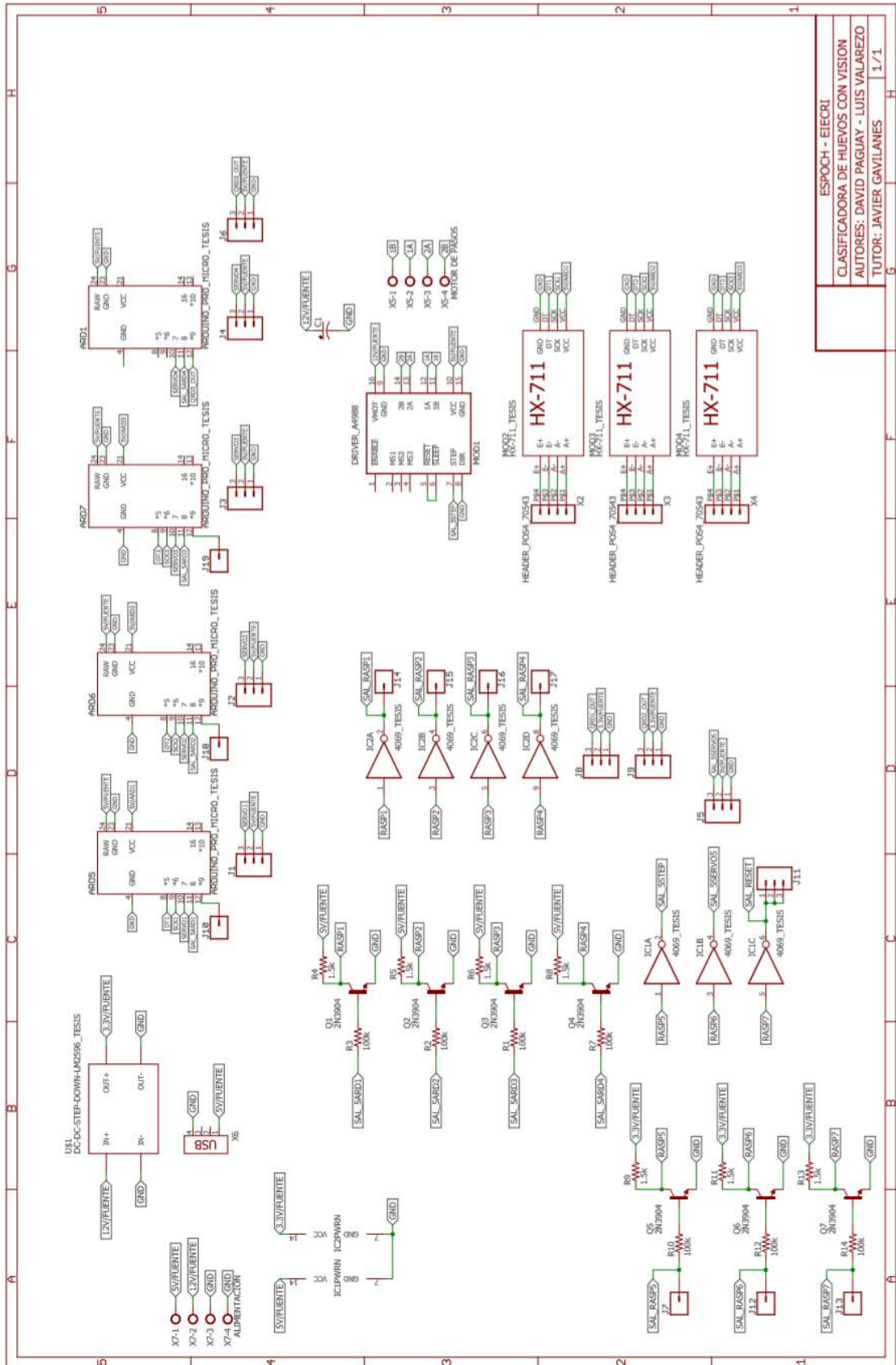
Tendencias avícolas mundiales. [En línea] 2013. [Citado el: 20 de enero de 2018.] Disponible en:
<http://www.elsitioavicola.com/articles/2310/tendencias-avacolas-mundiales-2012-racord-mundial-de-huevos-a-pesar-de-crecimiento-mas-lento/>.

TESLABEM. *Sensor infrarrojo.* [En línea] 2016. [Citado el: 27 de enero de 2018.] Disponible en: <http://teslabem.com/qrd1114-sensor-infrarrojo.html>.

¿Qué es OpenCV?. [En línea] 2013. [Citado el: 05 de enero de 2018.] Disponible en:
<https://unpocodejava.com/2013/10/09/que-es-opencv/>.

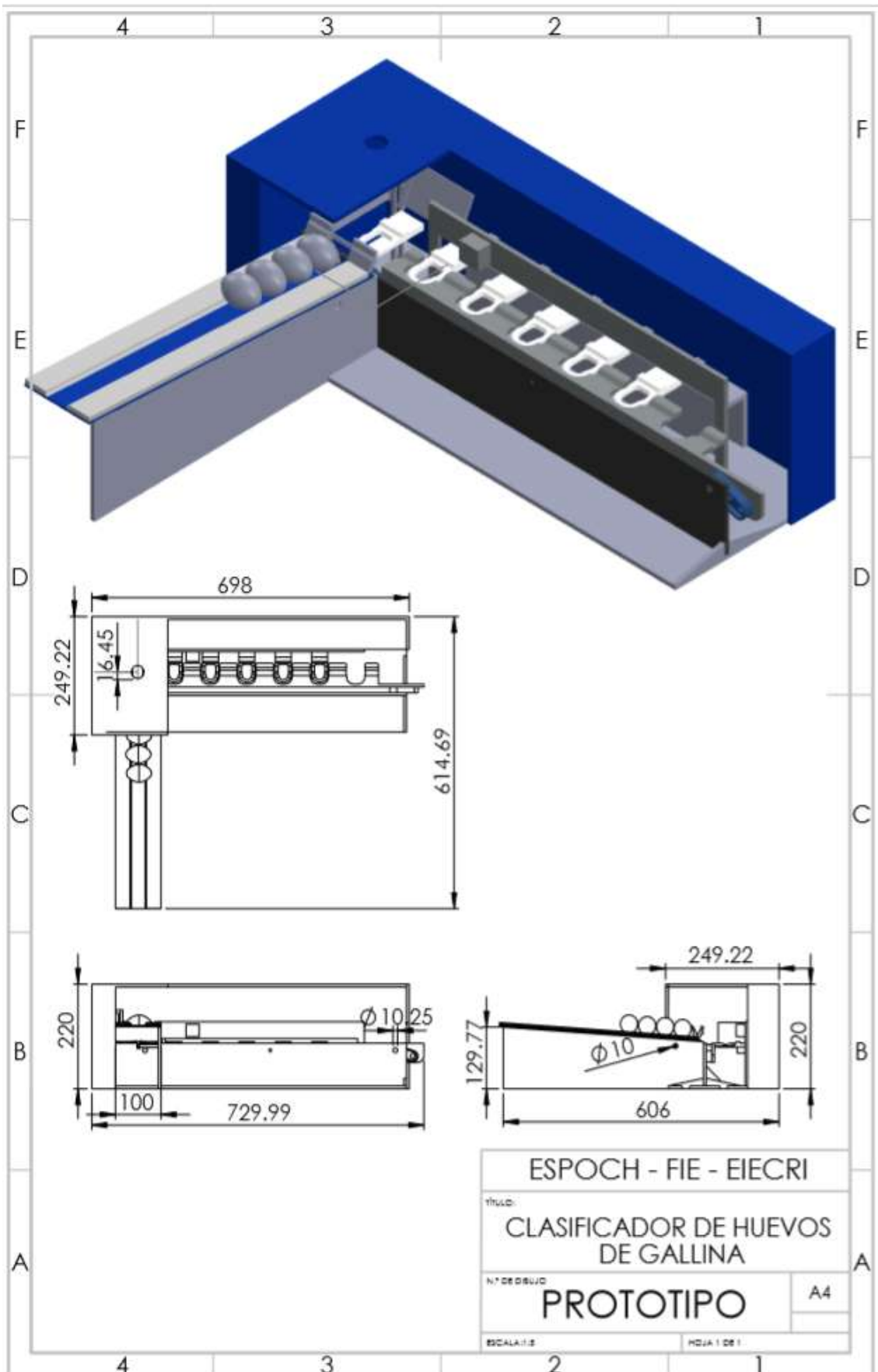
ANEXOS

Anexo A: Planos de conexión

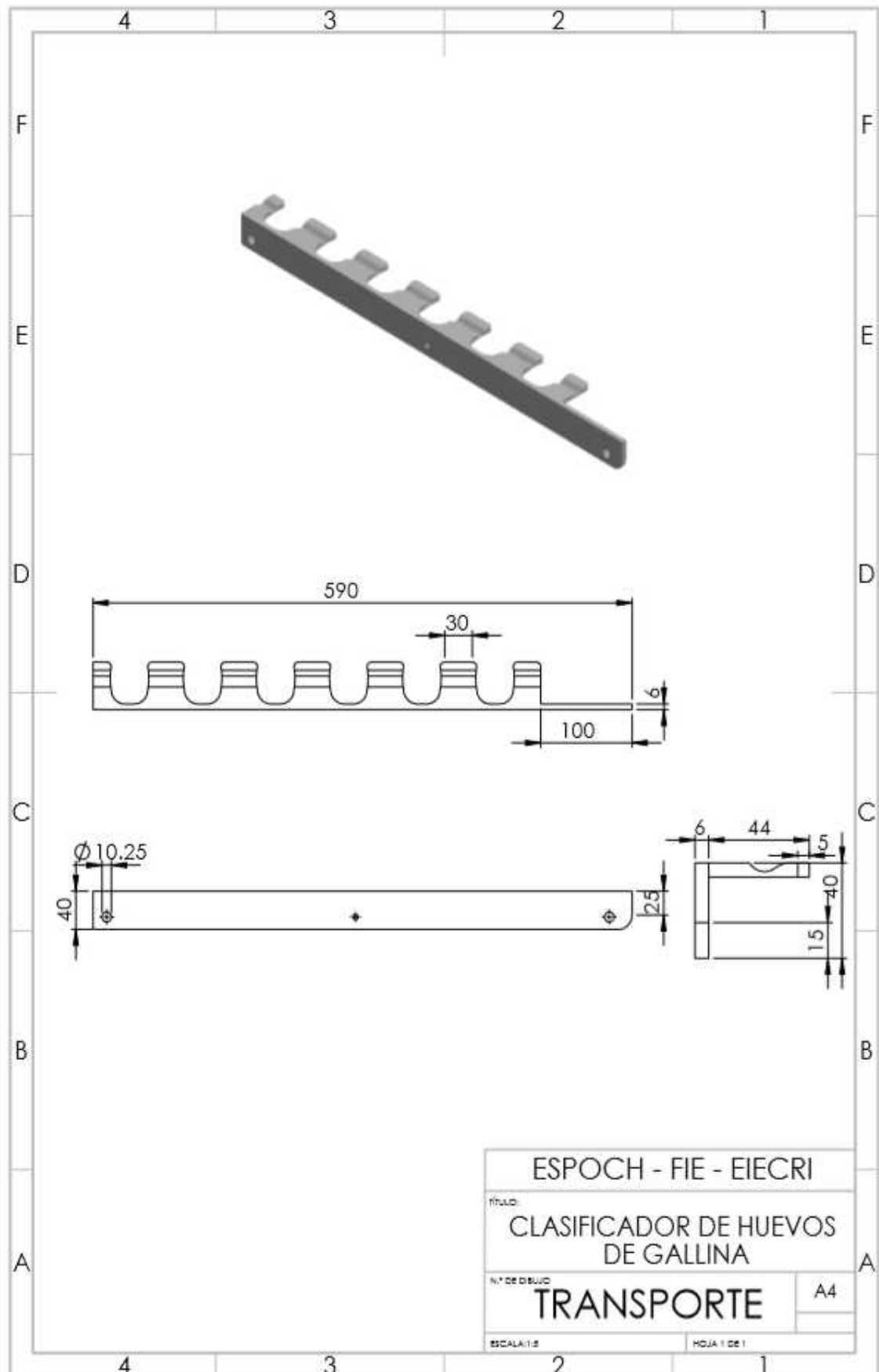


ESPOCH - EIECRI
 CLASIFICADORA DE HUEVOS CON VISION
 AUTORES: DAVID PAGUAY - LUIS VALAREZO
 TUTOR: JAVIER GAVILANES 1/1

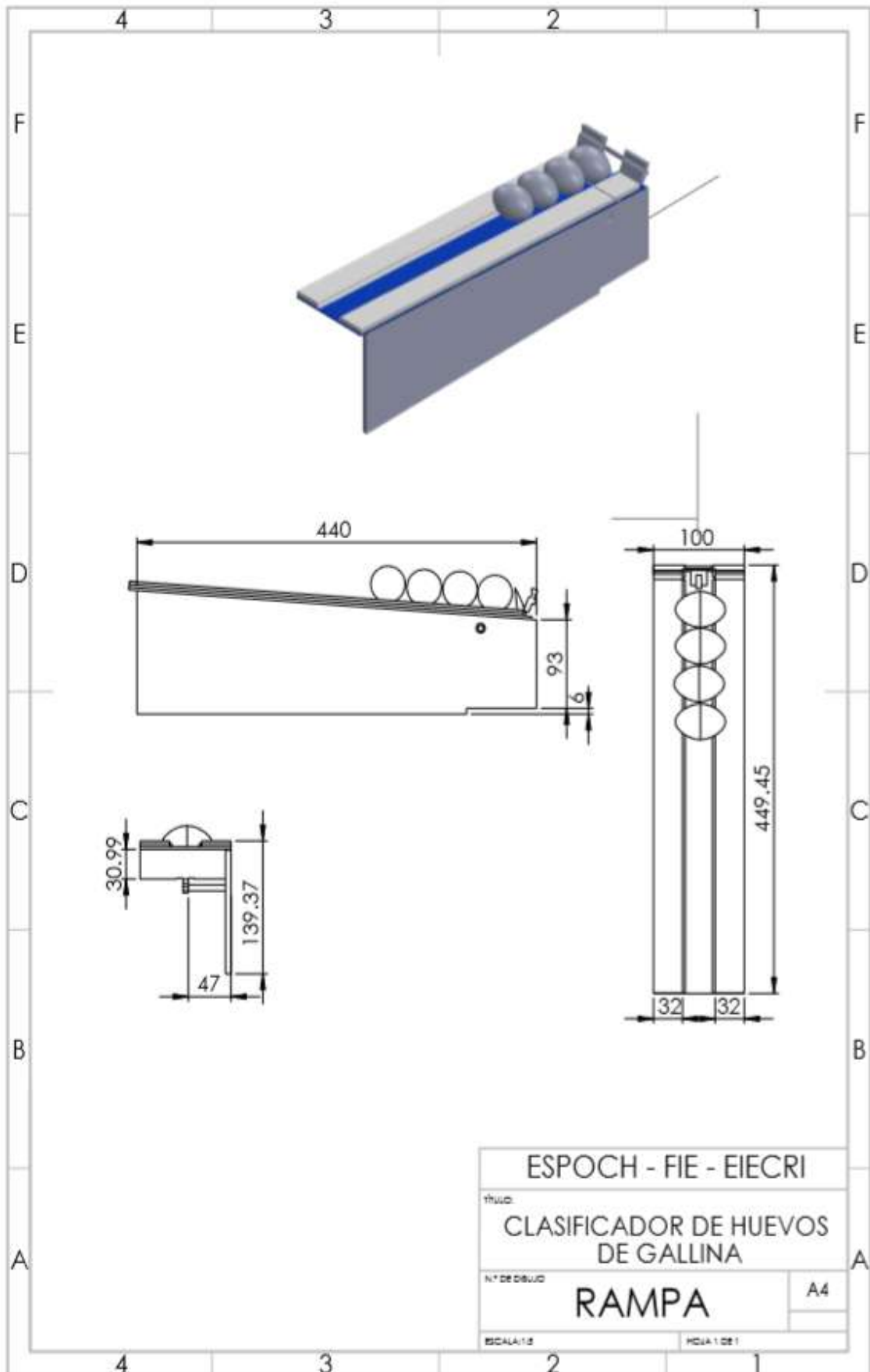
Anexo B: Planos del prototipo



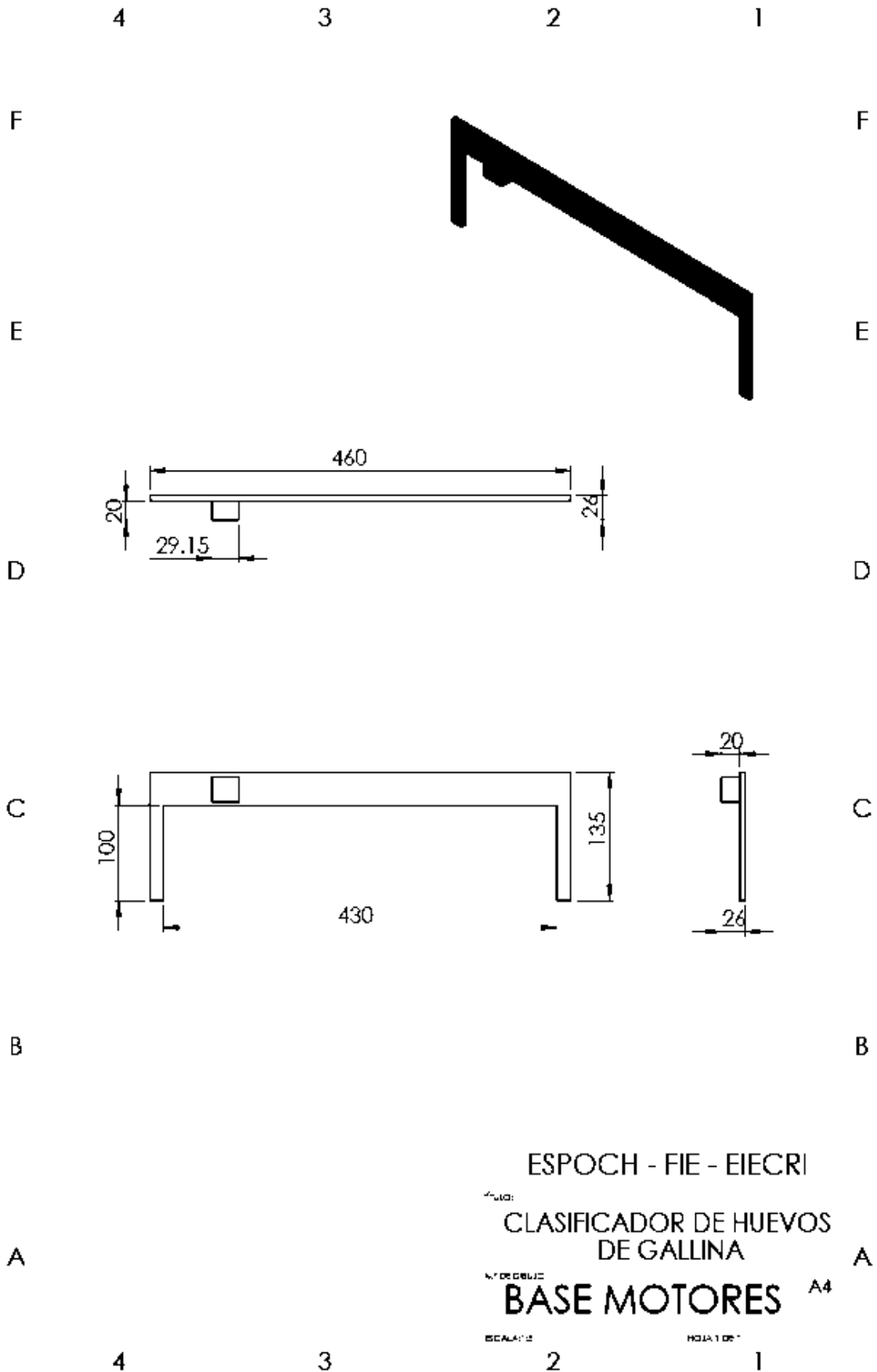
Anexo C: Planos del transporte



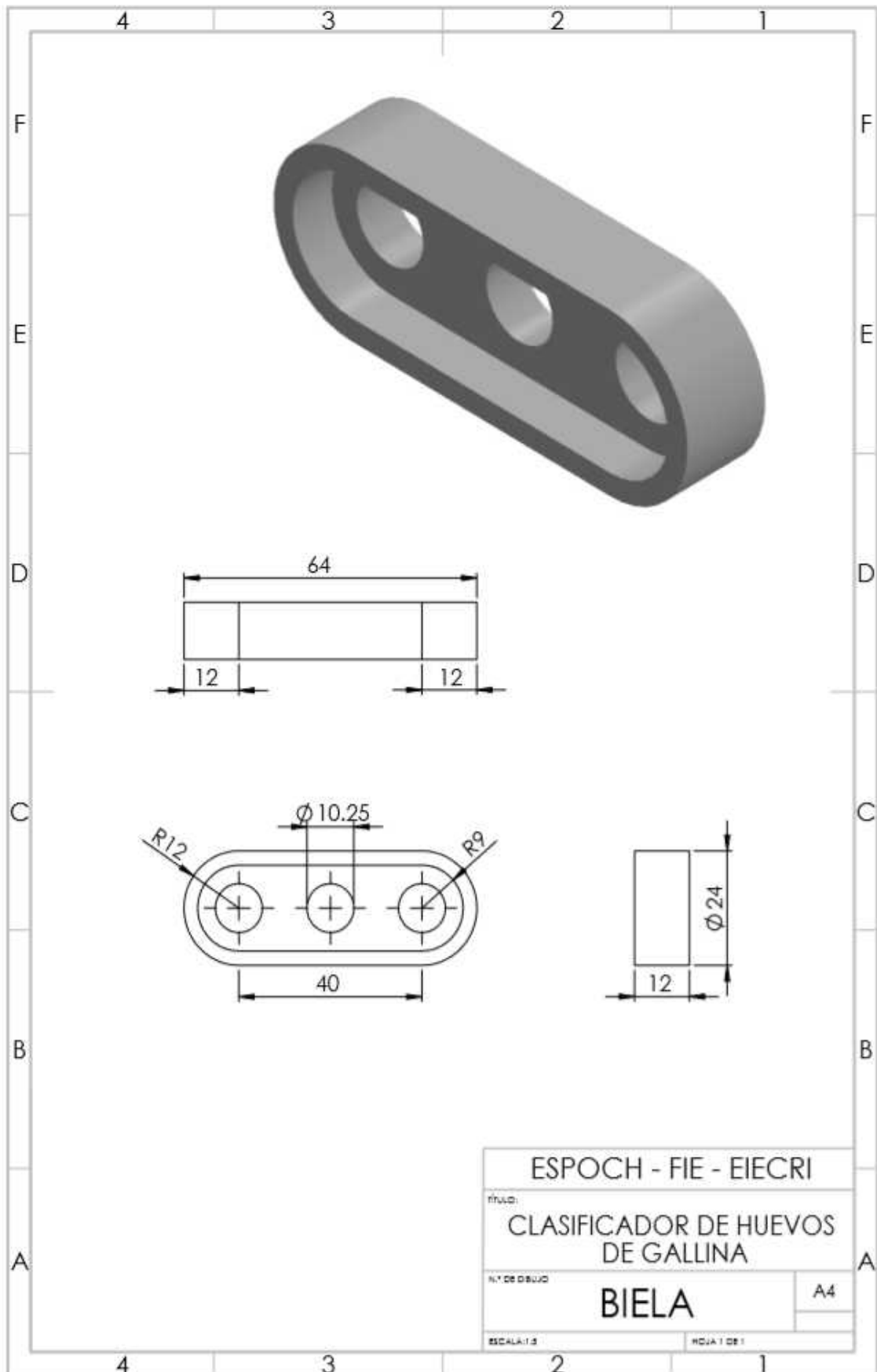
Anexo D: Planos de la rampa



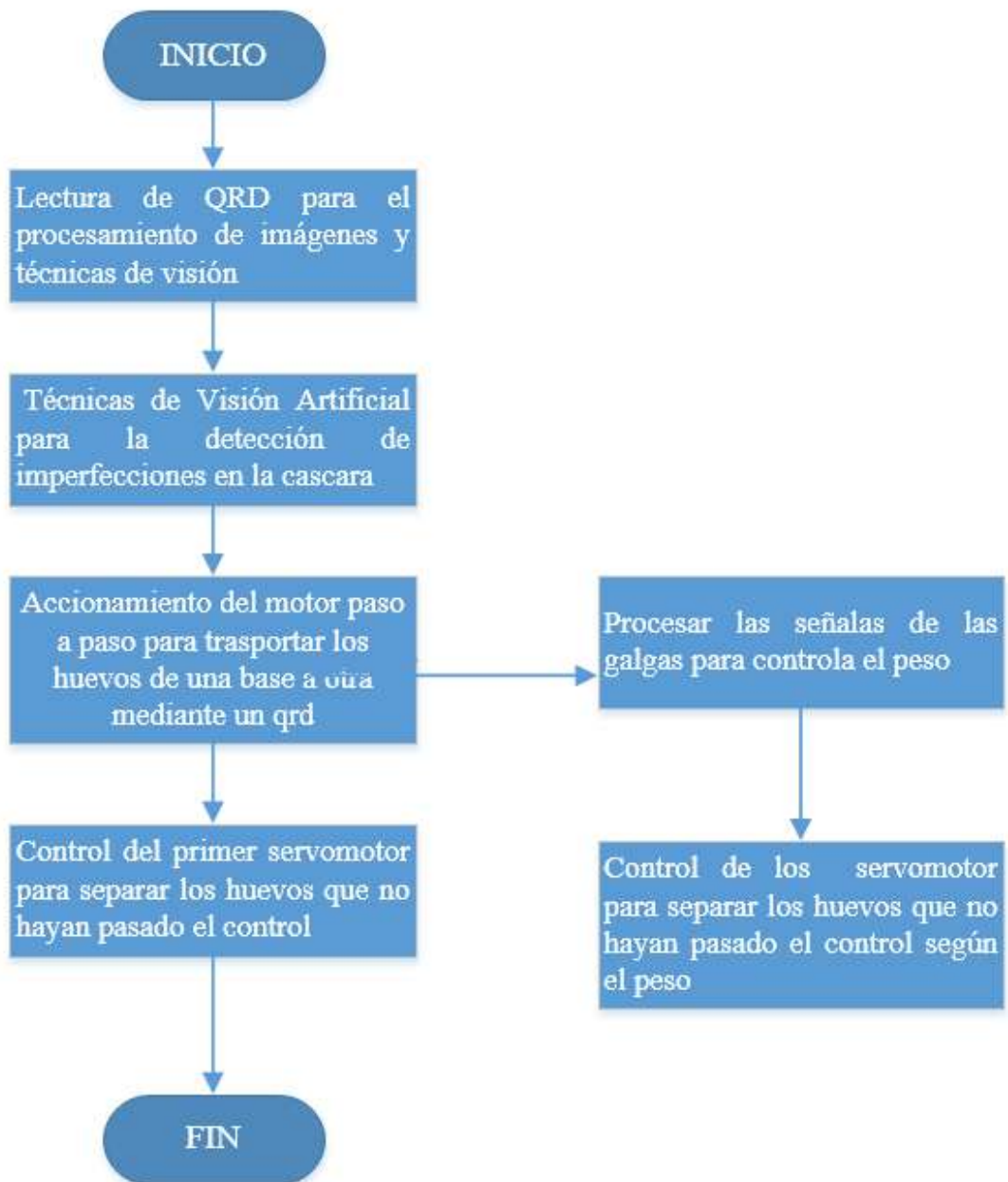
Anexo E: Planos base motores



Anexo F: Planos de la biela



Anexo G: Diagrama de flujo del funcionamiento



Anexo H: Manual de usuario

Manual de Usuario

Para el uso adecuado de la maquina se debe seguir una seria de pasos.

1. Encender el dispositivo conectándolo a una toma de 110 AC voltaje nominal, proveyendo de voltaje a la Raspberry y los controladores respectivos.
2. Al colocar los huevos se lo debe de hacer llenando progresivamente hasta inicio de la rampa.
3. Tampoco colocar algún peso en las galgas antes de que la camina comience a funcionar.
4. Esperar que la Raspberry inicie el sistema con sus sistema operativo y reinicie la HMI
5. En el entorno de la HMI se puede se puede ver Los datos de identificación como faculta y escuela respectiva.

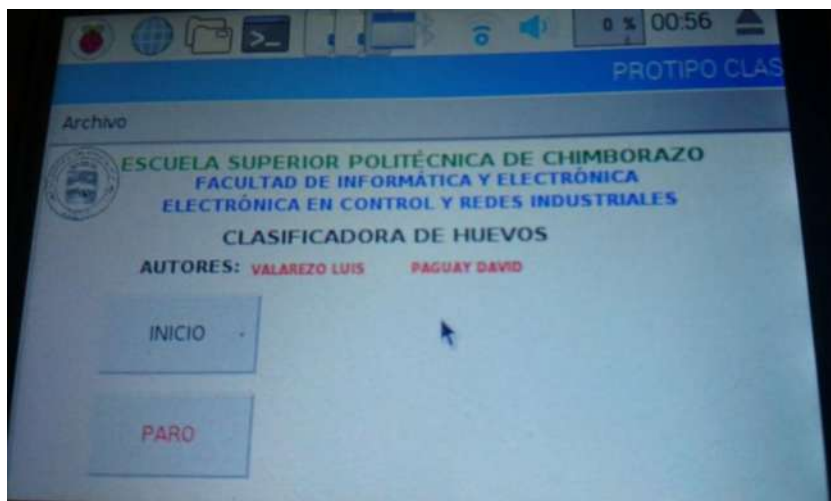


Fig. 1-1: Entorno Grafico

6. El sistema consta de 2 botones uno de inicio de y de paro.
7. AL presionar el botón inicio la maquina comenzara a funcionar y hacer el procesamiento respectivo como las técnicas de visión, captar las señales de los sensores y el moviente del motor.

- Al presionar el botón Pause se detendrá el motor y proceso de visión máquina, esta opción existe en caso de producirse una falla.



Fig. 1-2: Entorno Grafico

- Para el conteo de los se debe observar cada uno de los recuadros que se encuentra a lado de botones Inicio y Pausa
- Para acceder a la base de datos se debe ir a un menú se desplaza un submenú donde y se selecciona base de datos un nueva venta con botón actualizar y se verá el almacenamiento con año mes día hora minutos y segundo. El orden en el que está dispuesto es huevos rotos, huevos manchados, huevos grandes, huevos pequeños y huevos muy pequeños.

	NUMBEFIRST	MSURC	CINI	SEIS	SIEI	OCHO
1	0	0	0	0	0	2018-02-27
2	0	0	0	0	0	2018-02-27
3	0	0	0	0	0	2018-02-27 13:06:42.812997
4	0	0	0	0	0	2018-02-27 13:06:47.071934
5	0	0	0	0	0	2018-02-27 13:06:51.746150
6	0	0	0	0	0	2018-02-27 13:06:55.102244
7	0	0	0	0	0	2018-02-27 13:06:59.041716
8	0	0	0	0	0	2018-02-27 13:07:04.517233
9	0	0	0	0	0	2018-02-27 13:40:04.808700

ACTUALIZAR

Fig. 1-3: Base de Datos

- Para salir del HMI cuenta con un botón salir donde saldrá de la aplicación al finalizar con la clasificación de los huevos.