



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

**“DISEÑO Y CONSTRUCCIÓN DE UN CLASIFICADOR 4.0**  
**BASADO EN MACHINE LEARNING PARA EL CONTROL DE**  
**CALIDAD DE PRODUCTOS DEL SECTOR AGRÍCOLA”**

**Trabajo de titulación:**

**Tipo:** Dispositivo Tecnológico

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA, CONTROL Y REDES**  
**INDUSTRIALES**

**AUTOR:** CARLOS ANDRÉS GUAILLAZACA GONZÁLEZ

**DIRECTOR:** ING. EDWIN VINICIO ALTAMIRANO SANTILLÁN

Riobamba – Ecuador  
2020

**©2020, CARLOS ANDRÉS GUAILLAZACA GONZÁLEZ**

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

El Tribunal del Trabajo de Titulación certifica que: El trabajo de titulación Tipo: Dispositivo Tecnológico, **DISEÑO Y CONSTRUCCIÓN DE UN CLASIFICADOR 4.0 BASADO EN MACHINE LEARNING PARA EL CONTROL DE CALIDAD DE PRODUCTOS DEL SECTOR AGRÍCOLA**, realizado por el señor: **CARLOS ANDRÉS GUAILLAZACA GONZÁLEZ**, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, el mismo que cumple con los requerimientos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

**FIRMA**                      **FECHA**

Ing. Washington Gilberto Luna Encalada  
**DECANO DE LA FACULTAD**  
**DE INFORMÁTICA Y ELECTRÓNICA**

\_\_\_\_\_

Ing. Edwin Vinicio Altamirano Santillán  
**DIRECTOR DEL TRABAJO DE TITULACIÓN**

\_\_\_\_\_

Ing. Jorge Luis Hernández Ambato  
**MIEMBRO DEL TRABAJO DE TITULACIÓN**

\_\_\_\_\_

Yo, Carlos Andrés Guallazaca González, declaro que el presente trabajo de titulación es de mi autoría y que los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académicas de los contenidos de este trabajo de titulación. El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 25 de Agosto del 2020



**Carlos Andrés Guallazaca González**

**C.I: 060355679-6**

## **DEDICATORIA**

El presente trabajo de titulación lo dedico con todo mi amor y cariño a mis amados padres, Carlos y Marlene, quien con su amor, paciencia y esfuerzo me han permitido llegar a cumplir hoy un sueño más, gracias por inculcar en mí el ejemplo de esfuerzo y valentía, de no temer a las adversidades porque Dios está siempre con nosotros. A mis hermanos Brandon y Elian, quienes son mi inspiración para seguir siempre adelante y ser parte ejemplar en su formación ejemplar, a Daniela por ser una figura importante en mi vida.

**Carlos**

## **AGRADECIMIENTO**

Expreso mi gratitud a Dios, quien con su bendición me ha permitido culminar esta etapa muy importante en mi vida. A la Virgencita de Guadalupe, por guiarme día a día en esta aventura de la vida.

Mi profundo agradecimiento a Daniela, quien fue un pilar fundamental en mi formación, a mis amigos William, Robinson y Cristhian, gracias por apoyarme y extender su mano en los momentos duros de este proceso.

Finalmente quiero expresar mi más grande y sincero agradecimiento a mis tutores, Ingeniero Edwin Altamirano y Doctor Jorge Hernández, principales colaboradores durante todo este proceso, quienes con su dirección, conocimiento, enseñanza y colaboración permitió el desarrollo de este trabajo.

**Carlos**

## TABLA DE CONTENIDO

ÍNDICE DE TABLAS .....	xi
ÍNDICE DE FIGURAS .....	xiii
ÍNDICE DE GRÁFICOS.....	xvii
ÍNDICE DE ABREVIATURAS .....	xviii
ÍNDICE DE ANEXOS .....	xix
RESUMEN .....	xx
ABSTRACT.....	xxi
INTRODUCCIÓN .....	1
<b>CAPÍTULO I</b>	
<b>1. MARCO TEÓRICO REFERENCIAL .....</b>	<b>4</b>
<b>1.1. Inteligencia Artificial .....</b>	<b>4</b>
<b>1.1.1. <i>Categorización de la Inteligencia Artificial</i> .....</b>	<b>5</b>
<b>1.1.2. <i>Importancia de la Inteligencia Artificial</i>.....</b>	<b>6</b>
<b>1.2. Agentes Inteligentes.....</b>	<b>7</b>
<b>1.2.1. <i>Características de un Agente Inteligente</i> .....</b>	<b>7</b>
<b>1.2.2. <i>Ejemplo Agente Inteligente</i> .....</b>	<b>8</b>
<b>1.3. Técnicas Computacionales.....</b>	<b>9</b>
<b>1.3.1. <i>Hard Computing</i>.....</b>	<b>10</b>
<b>1.3.2. <i>Soft Computing</i>.....</b>	<b>10</b>
<b>1.4. Lógica Difusa. ....</b>	<b>11</b>
<b>1.4.1. <i>Funciones de Activación RELU</i>.....</b>	<b>13</b>
<b>1.4.2. <i>Controlador Difuso</i>.....</b>	<b>19</b>
<b>1.4.3. <i>Ventajas y Desventajas del controlador difuso</i> .....</b>	<b>20</b>
<b>1.5. Algoritmos Genéticos. ....</b>	<b>20</b>
<b>1.5.1. <i>Operaciones Biológicas</i> .....</b>	<b>21</b>

1.5.2.	<i>Diagrama de flujo de un algoritmo genético.</i>	22
1.5.3.	<i>Ventajas y Desventajas de los GA.</i>	23
1.6.	<b>Redes Neuronales Artificiales.</b>	24
1.6.1.	<i>Neurona</i>	24
1.6.2.	<i>Combinación de Neuronas.</i>	27
1.6.3.	<i>Redes Neuronales Multicapa</i>	28
1.7.	<b>Protocolo MQTT</b>	30
1.7.1.	<i>¿Por qué es importante MQTT?</i>	31
1.7.2.	<i>MQTT en la Industria</i>	31

## CAPITULO II

2.	<b>PROPUESTA Y DISEÑO DE PROTOTIPO</b>	33
2.1.	<b>Requerimientos Generales.</b>	33
2.1.1.	<i>Definición de Variables.</i>	35
2.2.	<b>Selección del Hardware</b>	35
2.2.1.	<i>Dispositivo de Procesamiento control nivel actuadores</i>	36
2.2.2.	<i>Etapas distribución y protección energética</i>	37
2.2.3.	<i>Etapas de Dispositivos de Adquisición de Datos</i>	40
2.2.3.1.	<i>Sensor de presencia Digital</i>	40
2.2.4.	<i>Etapas de Control de Actuadores</i>	42
2.2.4.1.	<i>Driver Motor</i>	42
2.2.4.2.	<i>Servo Motor</i>	43
2.2.5.	<i>Fuente de Voltaje 110v AC - 24v DC</i>	44
2.3.	<b>Implementación del Software</b>	45
2.3.1.	<i>Interfaz Web.</i>	45
2.3.1.1.	<i>Pantalla Inicio o Home</i>	48
2.3.1.2.	<i>Selección del Algoritmo Machine Learning KNN de la Librería ML5.JS</i>	50
2.3.1.3.	<i>Implementación de un clasificador de imágenes con Redes Neuronales Convolucionales (CNN)</i>	54



2.3.1.4.	<i>Generar y Cargar el modelo en JavaScript</i> .....	61
<b>2.4.</b>	<b>MQTT</b> .....	65
<b>2.5.</b>	<b>CAD Estructural</b> .....	68

### CAPÍTULO III

<b>3.</b>	<b>VALIDACIÓN DE PROTOTIPO</b> .....	74
<b>3.1.</b>	<b>Implementación del sistema prototipo</b> .....	74
<b>3.2.</b>	<b>Pruebas del Sistema - Etapa de Entrenamiento.</b> .....	76
<b>3.2.1.</b>	<b><i>Etapa Entrenamiento para el Producto 1 Bananas</i></b> .....	76
3.2.1.1.	<i>Entrenamiento de la calidad buena para el Producto 1 Bananas</i> .....	76
3.2.1.2.	<i>Entrenamiento de la calidad regular Producto 1 Bananas</i> .....	78
3.2.1.3.	<i>Entrenamiento de la calidad mala Producto 1 Bananas.</i> .....	79
<b>3.2.2.</b>	<b><i>Etapa Entrenamiento para el Producto 2 Naranjas.</i></b> .....	80
3.2.2.1.	<i>Entrenamiento de la calidad buena para el Producto 2 Naranjas</i> .....	80
3.2.2.2.	<i>Entrenamiento de la calidad regular para el Producto 2 Naranjas.</i> .....	81
3.2.2.3.	<i>Entrenamiento de la calidad malo para el Producto 2 Naranjas.</i> .....	82
<b>3.2.3.</b>	<b><i>Etapa Entrenamiento para el Producto 3 Plátano Verde.</i></b> .....	83
3.2.3.1.	<i>Entrenamiento de la calidad buena para el Producto 3 Plátano Verde</i> .....	83
3.2.3.2.	<i>Entrenamiento de la calidad regular para el Producto 3 Plátano Verde</i> .....	84
3.2.3.3.	<i>Entrenamiento de la calidad malo para el Producto 3 Plátano Verde</i> .....	86
<b>3.2.4.</b>	<b><i>Etapa Entrenamiento para el Producto 4 Manzanas</i></b> .....	87
3.2.4.1.	<i>Entrenamiento de la calidad bueno Producto 4 Manzanas</i> .....	87
3.2.4.2.	<i>Entrenamiento de la calidad regular Producto 4 Manzanas</i> .....	88
3.2.4.3.	<i>Entrenamiento de la calidad mala Producto 4 Manzanas.</i> .....	89
<b>3.3.</b>	<b>Resumen - Etapa de Entrenamiento.</b> .....	91
<b>3.4.</b>	<b>Evaluación final del Sistema - Etapa de Validación.</b> .....	92
<b>3.4.1.</b>	<b><i>Etapa de Evaluación para el Producto 1 Bananas.</i></b> .....	92
<b>3.4.2.</b>	<b><i>Etapa de Evaluación para el Producto 2 Naranjas.</i></b> .....	93
<b>3.4.3.</b>	<b><i>Etapa de Evaluación para el Producto 3 Plátanos Verdes.</i></b> .....	94

3.4.4.	<i>Etapa de Evaluación para el Producto 4 Manzanas</i> .....	95
3.5.	<b>Resumen - Etapa de Validación</b> .....	96
3.6.	<b>Pruebas de comunicación</b> .....	97

## CAPÍTULO IV

4.	<b>EVALUACIÓN ECONÓMICA</b> .....	103
4.1.	<b>Costos</b> .....	103
4.1.1.	<i>Costos directos</i> .....	103
4.1.1.1.	<i>Costos implementación eléctrica</i> .....	103
4.1.1.2.	<i>Costos implementación estructural</i> .....	103
4.1.2.	<i>Costos directos totales</i> .....	104
4.1.3.	<b>Costos Indirectos</b> .....	104
4.1.4.	<i>Costo Total</i> .....	104

	<b>CONCLUSIONES</b> .....	105
--	---------------------------	-----

	<b>RECOMENDACIONES</b> .....	106
--	------------------------------	-----

## GLOSARIO

## BIBLIOGRAFÍA

## ANEXOS

## ÍNDICE DE TABLAS

<b>Tabla 1-1:</b>	Representación de las funciones de la aspiradora figura 5-1.....	9
<b>Tabla 2-1:</b>	Diferencias entre Soft computing y Hard computing. ....	11
<b>Tabla 1-2:</b>	Variables del sistema. ....	35
<b>Tabla 2-2:</b>	Datos Técnicos del ESP8266 .....	37
<b>Tabla 3-2:</b>	Descripción de los terminales del TO263-5L.....	39
<b>Tabla 4-2:</b>	Características Sharp GP2y0d810z0f.....	41
<b>Tabla 5-2:</b>	Datos Técnicos Convertidor dc-dc Buck.....	43
<b>Tabla 6-2:</b>	Datos Técnicos Tower Pro MG995.....	44
<b>Tabla 7-2:</b>	Lista de materiales para ensamble estructural. ....	71
<b>Tabla 1-3:</b>	Tabla número de prueba y el número de imágenes capturadas. ....	75
<b>Tabla 2-3:</b>	Evaluación del sistema en la etapa de entrenamiento: producto bueno – bananas	77
<b>Tabla 3-3:</b>	Evaluación del sistema etapa de entrenamiento, producto regular bananas.....	78
<b>Tabla 4-3:</b>	Evaluación del sistema etapa de entrenamiento, producto malo bananas .....	79
<b>Tabla 5-3:</b>	Evaluación del sistema etapa de entrenamiento, producto bueno naranjas .....	80
<b>Tabla 6-3:</b>	Evaluación del sistema etapa de entrenamiento, producto regular naranjas .....	81
<b>Tabla 7-3:</b>	Evaluación del sistema etapa de entrenamiento, producto malo naranjas .....	82
<b>Tabla 8-3:</b>	Evaluación del sistema etapa de entrenamiento, producto bueno plátano verde. .	84
<b>Tabla 9-3:</b>	Evaluación del sistema etapa de entrenamiento, producto regular plátano verde. .	85
<b>Tabla 10-3:</b>	Evaluación del sistema etapa de entrenamiento, producto malo plátano verde. ..	86
<b>Tabla 11-3:</b>	Evaluación del sistema etapa de entrenamiento, producto bueno manzanas. ....	87
<b>Tabla 12-3:</b>	Evaluación del sistema etapa de entrenamiento, producto regular manzanas. ....	88
<b>Tabla 13-3:</b>	Evaluación del sistema etapa de entrenamiento, producto malo manzanas. ....	90
<b>Tabla 14-3:</b>	Resumen etapa entrenamiento.....	91
<b>Tabla 15-3:</b>	Etapa de validación producto 1 bananas. ....	93
<b>Tabla 16-3:</b>	Etapa de validación producto 2 naranjas. ....	94

<b>Tabla 17-3:</b>	Etapa de validación producto 3 plátanos verdes.....	95
<b>Tabla 18-3:</b>	Etapa de validación producto 4 manzanas.....	96
<b>Tabla 1-4:</b>	Costos eléctricos .....	103
<b>Tabla 2-4:</b>	Costos estructurales .....	103
<b>Tabla 3-4:</b>	Costos directos totales .....	104
<b>Tabla 4-4:</b>	Costos Indirectos .....	104
<b>Tabla 5-4:</b>	Costos Totales.....	104

## ÍNDICE DE FIGURAS

<b>Figura 1-1:</b>	Interacción de los Agentes Inteligentes con el Medio.....	7
<b>Figura 2-1:</b>	Interpretación entorno de trabajo de una aspirada, escenario A y B. ....	9
<b>Figura 3-1:</b>	Conjuntos certeros vs Conjuntos difusos.....	12
<b>Figura 4-1:</b>	Nivel de pertenencia vs Universo de discurso. ....	13
<b>Figura 5-1:</b>	Características función de membresía. ....	13
<b>Figura 6-1:</b>	Funciones de membresía Normal - Subnormal.....	14
<b>Figura 7-1:</b>	Funciones de membresía Simétricas – No simétricas. ....	14
<b>Figura 8-1:</b>	a. Abierta a la derecha b. Abierta a la Izquierda c. Cerrado.....	15
<b>Figura 9-1:</b>	Función de membresía Singleton .....	15
<b>Figura 10-1:</b>	Función de membresía GAMMA. ....	16
<b>Figura 11-1:</b>	Función de membresía Triangular. ....	16
<b>Figura 12-1:</b>	Función de membresía Trapezoidal.....	17
<b>Figura 13-1:</b>	Función de membresía Gaussiana. ....	17
<b>Figura 14-1:</b>	Función de membresía Campana Generalizada. ....	18
<b>Figura 15-1:</b>	Función de membresía Sigmoidal. ....	18
<b>Figura 17-1:</b>	Operación Cruce.....	21
<b>Figura 18-1:</b>	Operación Mutación. ....	22
<b>Figura 19-1:</b>	Algoritmo General de los GA.....	23
<b>Figura 20-1:</b>	Representación de una Neurona Biológica. ....	24
<b>Figura 21-1:</b>	Representación de una Neurona Artificial. ....	25
<b>Figura 22-1:</b>	Representación de una Neurona Artificial. ....	25
<b>Figura 23-1:</b>	Representación binaria ejemplo de la función de una neurona artificial.....	26
<b>Figura 24-1:</b>	Representación binaria ejemplo de la función de una neurona artificial. ....	27
<b>Figura 25-1:</b>	Representación Recta de regresión lineal del ejemplo 24-1.....	27
<b>Figura 26-1:</b>	Representación Recta de regresión lineal operación XOR. ....	28

<b>Figura 27-1:</b>	Combinación de neuronas para dar solución a XOR. ....	28
<b>Figura 28-1:</b>	Estructura básica ANN multicapa. ....	29
<b>Figura 29-1:</b>	Conjunto de datos MNIST: Caracteres escritura a mano. ....	29
<b>Figura 30-1:</b>	Desfragmentación de la imagen en matriz de datos. ....	29
<b>Figura 31-1:</b>	Red Neuronal con una capa oculta. ....	30
<b>Figura 32-1:</b>	Lógica del protocolo MQTT.....	30
<b>Figura 33-1:</b>	Implementación del protocolo MQTT en la industria. ....	32
<b>Figura 1-2:</b>	Metodología. ....	33
<b>Figura 2-2:</b>	Arquitectura General del Sistema.....	34
<b>Figura 3-2:</b>	Esquema del funcionamiento y aprendizaje. ....	35
<b>Figura 4-2:</b>	NodeMCU ESP8266. ....	37
<b>Figura 5-2:</b>	Convertidor DC-DC Tipo Buck .....	38
<b>Figura 6-2:</b>	Diagrama interno del circuito integrado XL4005 con empaquetado TO263-5L. ....	39
<b>Figura 7-2:</b>	Distribución de los terminales del TO263-5L.....	39
<b>Figura 8-2:</b>	Convertidor DC-DC tipo Buck basado en el circuito integrado XL4005.....	40
<b>Figura 9-2:</b>	Sensor Sharp GP2y0d810z0f.....	41
<b>Figura 10-2:</b>	Característica de medición del sensor Sharp GP2y0d810z0f.....	42
<b>Figura 11-2:</b>	Esquemático Sharp GP2y0d810z0f. ....	42
<b>Figura 12-2:</b>	Convertidor DC-DC Tipo Buck XL4005 .....	43
<b>Figura 13-2:</b>	Tower Pro-MG 995 .....	44
<b>Figura 15-2:</b>	Editor de Texto Atom IDE GitHub.....	45
<b>Figura 16-2:</b>	Paquete Atom-Live-Server. ....	47
<b>Figura 17-2:</b>	Paquete Atom-ternjs. ....	47
<b>Figura 18-2:</b>	Paquete Minimap.....	47
<b>Figura 19-2:</b>	Paquete Atom-beautify.....	47
<b>Figura 21-2:</b>	Home página web parte introductoria.....	50
<b>Figura 22-2:</b>	Puntos de datos similares suelen existir cerca uno del otro. ....	51
<b>Figura 23-2:</b>	Cálculo de las distancias desde x hasta todos sus vecinos. ....	52
<b>Figura 24-2:</b>	Analogía de la lógica KNN .....	53

<b>Figura 25-2:</b>	MI5 por Tensorflow. ....	53
<b>Figura 26-2:</b>	Metodología para implementar Tensorflow web. ....	54
<b>Figura 27-2:</b>	Representación general de la red neuronal clasificador de imágenes.....	55
<b>Figura 28-2:</b>	Anaconda Navigator.....	56
<b>Figura 29-2:</b>	Instalación de paquetes desde el terminal. ....	56
<b>Figura 30-2:</b>	Inicialización del algoritmo para el entrenamiento. ....	56
<b>Figura 31-2:</b>	Variables selección del Data Set local. ....	57
<b>Figura 32-2:</b>	Declaración de los Parámetros .....	57
<b>Figura 33-2:</b>	Parámetros cv2.resize y cv2.cvtColor.....	58
<b>Figura 34-2:</b>	Parámetros cv2.resize y cv2.cvtColor.....	58
<b>Figura 35-2:</b>	Algoritmo que genera la gráfica 35-2.....	58
<b>Figura 36-2:</b>	Algoritmo que genera las etiquetas.....	59
<b>Figura 37-2:</b>	Representación de las imágenes pre procesadas. ....	59
<b>Figura 38-2:</b>	Algoritmo ceración de CNN.....	59
<b>Figura 39-2:</b>	Algoritmo ceración de CNN.....	60
<b>Figura 40-2:</b>	Ejecutar el entrenamiento. ....	60
<b>Figura 41-2:</b>	Entrenamiento interfaz web.....	61
<b>Figura 42-2:</b>	Librerías dentro de neurona.html.....	62
<b>Figura 43-2:</b>	Librerías dentro de neurona.html.....	62
<b>Figura 44-2:</b>	Configuración inserción del video en la ventana web.....	62
<b>Figura 45-2:</b>	Botones en la ventana.....	63
<b>Figura 46-2:</b>	Estándar de los botones utilizando selectAll.....	63
<b>Figura 47-2:</b>	Depurador del navegador. ....	64
<b>Figura 48-2:</b>	Capturar el nombre del botón dentro de HTML. ....	64
<b>Figura 49-2:</b>	Capturar nombre del botón con el depurador del navegador. ....	64
<b>Figura 50-2:</b>	Extraer el modelo MobileNet. ....	65
<b>Figura 51-2:</b>	Extraer el modelo MobileNet .....	65
<b>Figura 52-2:</b>	Broker shiftr.io .....	65
<b>Figura 53-2:</b>	Parámetros de validación mqtt. ....	65

<b>Figura 54-2:</b> Login del broker.....	66
<b>Figura 55-2:</b> Configuración inicial para establecer la conexión .....	66
<b>Figura 56-2:</b> Dashboard del enlace app web-mqtt-shiftr desde Ipad OS. ....	67
<b>Figura 57-2:</b> Dashboard mqtt desde el navegador teléfono Android. ....	67
<b>Figura 58-2:</b> Croquis V-Slot 20mmx20mm. ....	68
<b>Figura 59-2:</b> Croquis V-Slot 20mmx40mm. ....	68
<b>Figura 60-2:</b> Chumacera 20mm de diámetro.....	69
<b>Figura 61-2:</b> Rodillos transmisión dinámica del motor a la banda. ....	69
<b>Figura 62-2:</b> Cámara y case. ....	69
<b>Figura 63-2:</b> Motor.....	70
<b>Figura 64-2:</b> Servo Motor Tower Pro MG-995. ....	70
<b>Figura 65-2:</b> Ensamble estructura. ....	70
<b>Figura 66-2:</b> Ensamblaje real 1. ....	71
<b>Figura 67-2:</b> Visualización del sistema completo, envío de datos a diferentes dispositivos. ...	72
<b>Figura 69-2:</b> Ensamble completo. ....	72
<b>Figura 70-2:</b> Visualización de datos entorno móvil.....	73
<b>Figura 1-3:</b> Pestaña para en el entrenamiento del sistema .....	75
<b>Figura 2-3:</b> Interfaz web, configuración del fondo.....	97
<b>Figura 3-3:</b> Interfaz web, configuración del fondo.....	98
<b>Figura 4-3:</b> Interfaz web, configuración del fondo.....	99
<b>Figura 5-3:</b> Interfaz web, Bananas producto regular. ....	99
<b>Figura 6-3:</b> Shiftr bróker, Bananas producto regular, visualización del flujo de datos 1. ....	100
<b>Figura 7-3:</b> Shiftr bróker, Bananas producto regular visualización del flujo de datos 2. ....	100
<b>Figura 8-3:</b> Interfaz web, Bananas producto malo.....	101
<b>Figura 9-1:</b> Interfaz web, Bananas producto malo.....	101



## ÍNDICE DE GRÁFICOS

<b>Gráfico 1-2:</b>	Proceso para el desarrollo de software del proyecto. ....	45
<b>Gráfico 2-2:</b>	Diagrama de flujo de la aplicación web para la interfaz del proyecto. ....	46
<b>Gráfico 3-2:</b>	Diagrama de flujo funcionamiento del algoritmo knn.....	52
<b>Gráfico 1-3:</b>	Error relativo vs número de imágenes, producto bueno – bananas .....	77
<b>Gráfico 2-3:</b>	Error relativo vs número de imágenes, producto regular - bananas.....	78
<b>Gráfico 3-3:</b>	Error relativo vs número de imágenes, producto malo- bananas.....	79
<b>Gráfico 4-3:</b>	Error relativo vs número de imágenes, producto bueno - Naranjas.....	81
<b>Gráfico 5-3:</b>	Error relativo vs número de imágenes, producto regular- Naranjas .....	82
<b>Gráfico 6-3:</b>	Error relativo vs número de imágenes, producto malo- naranjas .....	83
<b>Gráfico 7-3:</b>	Error relativo vs número de imágenes, producto Bueno- Plátano Verde.....	84
<b>Gráfico 8-3:</b>	Error relativo vs número de imágenes, producto regular-plátano verde .....	85
<b>Gráfico 9-3:</b>	Error relativo vs número de imágenes, producto malo-plátano verde .....	86
<b>Gráfico 10-3:</b>	Error relativo vs número de imágenes, producto bueno-manzanas .....	88
<b>Gráfico 11-3:</b>	Error relativo vs número de imágenes, producto regular-manzanas.....	89
<b>Gráfico 12-3:</b>	Error relativo vs número de imágenes, producto malo-manzanas .....	90
<b>Gráfico 13-3:</b>	Resumen etapa entrenamiento exitoso. ....	91
<b>Gráfico 14-3:</b>	Errores del sistema según el producto en la etapa validación.....	97

## ÍNDICE DE ABREVIATURAS

- CAD:** Diseño asistido por computadora.
- KNN:** K-nearest neighbors.
- GA:** Algoritmo Genético.
- CNN:** Red Neuronal Convolutacional.
- ADN:** Ácido Desoxirribonucleico.
- ANN:** Artificial Neural Network.
- MNIST:** Modified National Institute of Standards and Technology database.
- RGB:** Rojo – Verde – Azul.
- MQTT:** Message Queue Telemetry Transport.
- PWM:** Modulación por ancho de pulsos.
- PCB:** Print Circuit Board.
- VOUT:** Voltaje de salida.
- VIN:** Voltaje de entrada.
- GND:** Groud o tierra.
- IOT:** Internet de las cosas
- M2M:** Machine to machine
- API:** Interfaz de programación de aplicaciones.

## **ÍNDICE DE ANEXOS**

**ANEXO A:** Boad PCB

**ANEXO B:** Diseño Esquemático PCB

**ANEXO C:** CAD Ensamblaje

**ANEXO D:** CAD Vista Lateral

**ANEXO E:** Código del script que se integra al contenedor HTML

## RESUMEN

El presente trabajo de titulación tuvo como objetivo la implementación de un prototipo electrónico con la capacidad de supervisar, clasificar y controlar la calidad de productos del sector agrícola, mediante la aplicación de técnicas de soft computing y algoritmos de machine learning que contribuyeron a la identificación de imágenes en tiempo real. Para la realización de este sistema se emplearon los métodos: deductivo, heurístico y experimental. El desarrollo del trabajo de titulación constó de las siguientes fases: Primero, se realizó el diseño asistido por computador (CAD) de la delineación estructural de la banda transportadora y de los actuadores. Segundo, se planificó e implementó el interfaz web utilizando la librería P5.js de JavaScript que permitió registrar imágenes en tiempo real mediante una webcam. Posteriormente, se creó un modelo de gestión de aprendizaje automático utilizando la librería ml5.js de JavaScript para inteligencia artificial que facilitó la implementación del algoritmo “*k-vecinos más cercanos*” para crear un modelo.js mediante una etapa de entrenamiento. Para el establecimiento de la comunicación y envío de datos de control procesados, desde la neurona hasta el microcontrolador ESP8266, se configuró un servidor central para el transporte de mensajes (broker) MQTT. Finalmente, se realizaron 10 pruebas para cada una de las variables (Producto Bueno, Regular y Descarte). Se concluyó que el promedio de la eficiencia del sistema es de un 90% con una base de datos de 45 a 50 imágenes por cada variable. Es recomendable realizar un buen proceso de entrenamiento considerando que el éxito de la operación es directamente proporcional al número de imágenes capturadas para un mejor aprendizaje.

**Palabras clave:** <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INTELIGENCIA ARTIFICIAL>, <INTERNET DE LAS COSAS (IOT)>, <APRENDIZAJE COMPUTACIONAL>, <AGRO TECNOLOGIA>, <AUTOMATIZACIÓN DE PROCESOS>, <SISTEMA INTELIGENTE>.



0012-DBRAI-UPT-2020

## **ABSTRACT**

The objective of this degree work was the implementation of an electronic prototype with the capacity to supervise, classify and control the quality of products in the agricultural sector, through the application of soft computing techniques and machine learning algorithms that contributed to the identification of images in real time. The following methods were used to implement this system: deductive, heuristic, and experimental. The development of the degree work consisted of the following phases: First, the computer-aided design (CAD) of the structural delineation of the conveyor belt and the actuators was carried out. Second, the web interface was planned and implemented using the P5.js JavaScript library that allowed for real-time image recording via a webcam. Then, an automatic learning management model was created using the JavaScript library ml5.js for artificial intelligence that facilitated the implementation of the "k nearest neighbors" algorithm to create a model through a training stage. For the establishment of communication and sending of processed control data, from the neuron to the ESP8266 microcontroller, a central server was configured for the transport of MQTT messages (broker). Finally, 10 tests were performed for each of the variables (Good, Regular, and Discarded Product). It was concluded that the average efficiency of the system is 90% with a database of 45 to 50 images for each variable. It is recommended to make a good training process considering that the success of the operation is directly proportional to the number of images captured for better learning.

**Keywords:** <TECNOLOGY AND SCIENCES OF ENGINEERING>, <ARTIFICIAL INTELLIGENCE>, <INTERNET OF THINGS (IOT)>, <COMPUTATIONAL LEARNING>, <AGRICULTURAL TECHNOLOGY>, <PROCESS AUTOMATION>, <INTELLIGENT SYSTEM>.

## INTRODUCCIÓN

Se debe reconocer que las PYMES dentro del tejido empresarial ecuatoriano tienen un peso significativo, donde alrededor del 80% de establecimientos comerciales son micros y medianas empresas. Y en términos de generación de empleo involucra alrededor de un 65% a nivel nacional (Carrillo, 2019), para lo cual este es un sector muy significativo en la base económica del país y se debe prestar la mayor atención para que sean actores principales que ayuden a dinamizar la economía ecuatoriana.

En la actualidad las empresas miden su rendimiento competitivo en base a su capacidad de producción y a los estándares de calidad de sus productos manufacturados, donde predominan en estas variables la tecnología empleada para una producción exitosamente rentable. Las pequeñas y medianas empresas ecuatorianas viven una desventaja muy grande en comparación a la capacidad de producción de empresas multinacionales, en la gran mayoría, sus procesos se caracterizan por ser manuales y usar conocimientos empíricos, la maquinaria y tecnología empleada es casi nula, su cadena de producción es lenta y no llega a satisfacer en su totalidad la demanda y requerimientos de su grupo objetivo de clientes.

Para una producción exitosa es indispensable proveer a estas empresas vulnerables con tecnología al alcance de sus necesidades y posibilidades económicas ya que el principal problema representa los altos costos de inversión en maquinaria y soluciones de hardware y software.

El nivel de automatización en los procesos de producción de las PYMES es sumamente bajo (King, 2013), lo cual genera un gran problema con la calidad de los productos y tiempos de producción no adecuados para satisfacer la demanda interna, lo que la exportación de sus productos se convierte en una utopía inalcanzable.

Actualmente en el Ecuador se evidencia que existe en mayor porcentaje el trabajo manual y la implementación de maquinaria es escasa por esta razón no existe una clasificadora inteligente que se adapte a las necesidades agroindustriales en nuestro medio, es por eso que se decidió plantear la investigación para diseñar y construir un prototipo de un clasificador 4.0 basado en machine learning para el control de calidad productos dirigido a pequeñas y medianas empresas del sector Agropecuario, aquellas que son la base fundamental para dinamizar de nuestra economía. (Ardila, 2011)

La transformación de la matriz productiva según el objetivo número 10 del Plan Nacional del Buen Vivir, “supone una interacción con la frontera científico-técnica, en la que se producen cambios estructurales que direccionan las formas tradicionales del proceso y la estructura productiva actual, hacia nuevas formas de producir que promueven la diversificación productiva en nuevos sectores, con mayor intensidad en conocimientos y consideraciones de asimetrías tecnológicas” (SEMPLADES, 2017). La implementación e integración de sistemas con inteligencia artificial para el control y supervisión de productos en una cadena de producción o clasificación de productos agrícolas se realiza con la finalidad de ofrecer una alternativa más económica y eficiente para un mejor control de calidad en el proceso de manufactura.

## **FORMULACIÓN DEL PROBLEMA**

*¿Es posible diseñar y construir un clasificador 4.0 basado en machine learning para el control de calidad de productos del sector agrícola?*

### **Sistematización Del Problema**

- ¿Existe en el mercado equipos que cumplan los requerimientos necesarios para ejecutar una clasificación de productos de manera automática?
- ¿Cuáles son los requisitos de hardware y software necesarios para la implementación del sistema basado en Inteligencia Artificial?
- ¿Qué parámetros serán necesarios tomar en cuenta para el entrenamiento supervisado de la red neuronal?
- ¿Qué pruebas serán necesarias para el correcto funcionamiento del sistema a implementarse?

## **OBJETIVOS**

### **OBJETIVO GENERAL**

Diseñar y construir de un clasificador 4.0 basado en machine learning para el control de calidad de productos del sector agrícola.

### **OBJETIVOS ESPECIFICOS**

- Analizar diferentes métodos y técnicas de inteligencia artificial y entre ellas escoger la más adecuada para la clasificación de imágenes.

- Determinar los requisitos de hardware necesarios para su posterior diseño.
- Recolectar información para el entrenamiento supervisado de la red neuronal.
- Realizar el análisis y resultados de las pruebas correspondientes de la implementación del sistema y corrección de estas.



# CAPÍTULO I

## 1. MARCO TEÓRICO REFERENCIAL

En el presente capítulo se realiza la descripción de la inteligencia artificial y sus diferentes técnicas de aplicabilidad, obteniéndose la más idónea para la clasificación en productos agrícolas, también se especifica el detallan los datos de productos clasificados y se los expone en la nube, utilizando doctrinas de la industria 4.0.

### 1.1. Inteligencia Artificial

El estudio de la inteligencia ha despertado interés en distintas áreas del conocimiento tanto filosófico, psicológico y educativo para ayudar a entender el comportamiento de ciertos grupos de seres vivos.(Ardila, 2011)

La descripción propia de inteligencia se la conceptualiza como la capacidad de solucionar problemas, encontrar respuestas a incertidumbres, tener la facultad de razonar, adaptarse al entorno y aprender de las experiencias por medio de la intuición; estas características han llevado a los seres vivos a evolucionar de manera constante para sobrevivir y mejorar la capacidad de adaptación frente a nuevas experiencias.

La inteligencia artificial es una amplia rama de la informática que se ocupa de desarrollar máquinas inteligentes capaces de realizar tareas que generalmente requieren inteligencia humana. La IA es una ciencia interdisciplinaria con múltiples enfoques, pero los avances en el aprendizaje automático y el aprendizaje profundo están creando un cambio de paradigma en prácticamente todos los sectores de la industria tecnológica.

Un claro ejemplo de los inicios de la inteligencia artificial nace en la Segunda Guerra Mundial, cuando un equipo de matemáticos británicos que trabajaban en Bletchley Park logró romper el código militar alemán, lo que permitió a los Aliados descifrar las comunicaciones de los militares alemanes. Estas comunicaciones estaban basadas en el lenguaje de la criptografía, que es la ciencia de crear y decodificar mensajes. El mensaje que se desea enviar se llama texto sin formato y el mensaje codificado texto cifrado, la decodificación del texto cifrado generalmente depende que el emisor y el receptor compartan una información privilegiada llamada clave. (Morales, 2013)

Fundamentalmente la Inteligencia artificial es una rama de la informática que tiene como objetivo responder afirmativamente la prueba de Turing, la misma que hace un esfuerzo de replicar o simular la inteligencia humana en máquinas.

La principal limitación para definir la IA como simplemente “construir máquinas inteligentes” es que en realidad no explica qué es la inteligencia artificial ¿Qué hace que una máquina sea inteligente?

Abordando el libro *Inteligencia Artificial: un enfoque moderno* de los autores Stuart Russell y Peter Norving nos plantean la unificación de su trabajo con relación a los agentes inteligentes en las máquinas. Considerando esto, la definición para Inteligencia artificial es “el estudio de agentes que reciben percepciones del entorno y realizan acciones” (Russell et al., 2011)

Mediante la exploración bibliográfica realizada por Russell y Norving plantearon cuatro enfoques diferentes que cronológicamente han definido a la Inteligencia artificial (Russell et al., 2011):

- Pensando humanamente.
- Pensando racionalmente.
- Actuando humanamente.
- Actuando racionalmente.

Los dos primeros puntos hacen referencia a los procesos del pensamiento y razonamiento, mientras que los otros plantean la conducta. Russell y Norving particularmente se centran en los agentes inteligentes o racionales que actúan en un proceso para lograr un resultado exitoso, citando “todas las habilidades necesarias para el Test de Turing también permiten que un agente actúe racionalmente” (Russell et al., 2011).

Patrick Henry, científico informático estadounidense del Instituto de Tecnología de Massachusetts, define “El objetivo principal de la Inteligencia Artificial es hacer que las máquinas sean más inteligentes, y hacer que las máquinas sean más útiles” (Winston, 1990).

### ***1.1.1. Categorización de la Inteligencia Artificial***

En el uso relevante que se le da a la Inteligencia Artificial lo podemos clasificar en dos grupos:

- ✓ **Inteligencia Artificial Débil:** también denominada IA débil, este tipo de inteligencia artificial se la encuentra operando dentro de un contexto o una actividad limitada. La IA débil a menudo

se centra en realizar una tarea en específica, está a lo largo del tiempo ha experimentado numerosos avances en la última década.

Ejemplos de uso cotidiano de Inteligencia Artificial:

- Asistentes inteligentes como (*google assistant, Siri o Alexa*)
- Recomendaciones de tratamientos de salud personalizados.
- Bots conversacionales para marketing y servicio al cliente.
- Detección de spam.
- Herramientas de monitoreo de redes sociales para contenido peligroso o noticias falsas.
- Recomendaciones de contenido en plataformas de entretenimiento como *Netflix y Spotify*.

✓ **Inteligencia Artificial Fuerte:** es el tipo de inteligencia que llevan a cabo las tareas consideradas como humanas. Estos tienden a ser sistemas más complejos. Sus algoritmos pueden manejar situaciones en las que se les puede requerir que resuelvan problemas sin que una persona supervise o tome una decisión en particular. Este tipo de sistemas son evidentes en autos autónomos o en quirófanos de hospitales inteligentes (Garbey et al., 2015).

En la práctica la búsqueda del Santo Grial “algoritmo universal para aprender y actuar en cualquier entorno” (Russell et al., 2011) es casi imposible.

### ***1.1.2. Importancia de la Inteligencia Artificial***

La Inteligencia Artificial aprovecha al máximo los datos, cuando los algoritmos son de autoaprendizaje, es importante retroalimentar al sistema con la mayor cantidad de datos posibles para que los algoritmos pueden generar nuevo conocimiento o información relevante que puede crear una ventaja competitiva en la industria, incluso si toda la competencia está aplicando técnicas similares, los mejores datos ganarán.

La IA logra una precisión increíble a través de las redes neuronales. Por ejemplo, interactuar con los asistentes virtuales *Google assistant, Alexa, Siri* o mismo buscador de *Google* y servicios como *Google Photos*, mientras más los usemos su precisión mejora proporcionalmente. En el campo de la medicina, las técnicas de Inteligencia Artificial han ayudado a una clasificación de imágenes y el reconocimiento de patrones con la misma precisión que los radiólogos altamente capacitados.

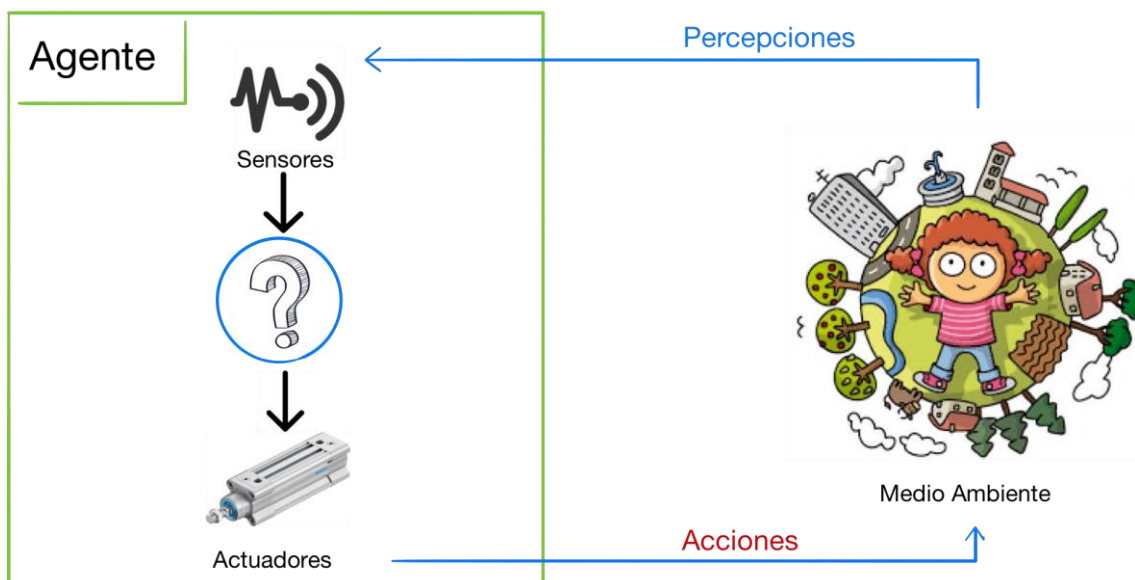
La Inteligencia Artificial puede adaptarse a través de modelos de aprendizaje mediante algoritmos lógicos que permiten que los datos hagan la programación según las necesidades del entorno. La

IA encuentra estructura y regularidades en los datos para que el algoritmo adquiera una habilidad, el mismo que puede orientarse a un clasificador o un predictor.

## 1.2. Agentes Inteligentes

“Un agente es una entidad con la capacidad de discernir su entorno o medio ambiente mediante sensores y actuar ante él utilizando actuadores”(Russell et al., 2011).

En la figura 1-1 se ilustra como interactúa una gente inteligente ante un entorno o medio ambiente.



**Figura 1-1:** Interacción de los Agentes Inteligentes con el Medio

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

Los agentes inteligentes son sistemas tecnológicos que poseen la capacidad de aprender sin la supervisión o integración humana, estos agentes pueden adaptarse al medio ambiente y a entornos complejos, normalmente se los asignan para realizar funciones particulares. Por ejemplo, un agente puede monitorear la temperatura de un paciente específico y unirse a otros agentes inteligentes para proporcionar una inteligencia colectiva que controla todos los signos vitales del mismo paciente y emitir un diagnóstico del estado de salud general.

### 1.2.1. Características de un Agente Inteligente

Un agente inteligente debe cumplir ciertas características para denotar cualidades importantes que ayuden a resolver problemas. Para ello se plantea una nueva definición, “Un agente es un sistema

ya sea software o hardware, capaz de actuar con el entorno de una manera flexible y autónoma”(Wooldridge y Jennings, 1995), las características son:

- ✓ **Reactivo**, el agente ejerce su función de acuerdo con los sucesos producidos en su entorno.
- ✓ **Proactivo**, el agente es capaz de establecer metas tomando la iniciativa propia.
- ✓ **Social**, el agente tiene la capacidad de comunicación entre otros agentes y/o humanos (usuarios).
- ✓ **Autónomo**, el agente posee la capacidad de operar o realizar sus funciones sin la intervención directa de un humano, no es necesario una supervisión.

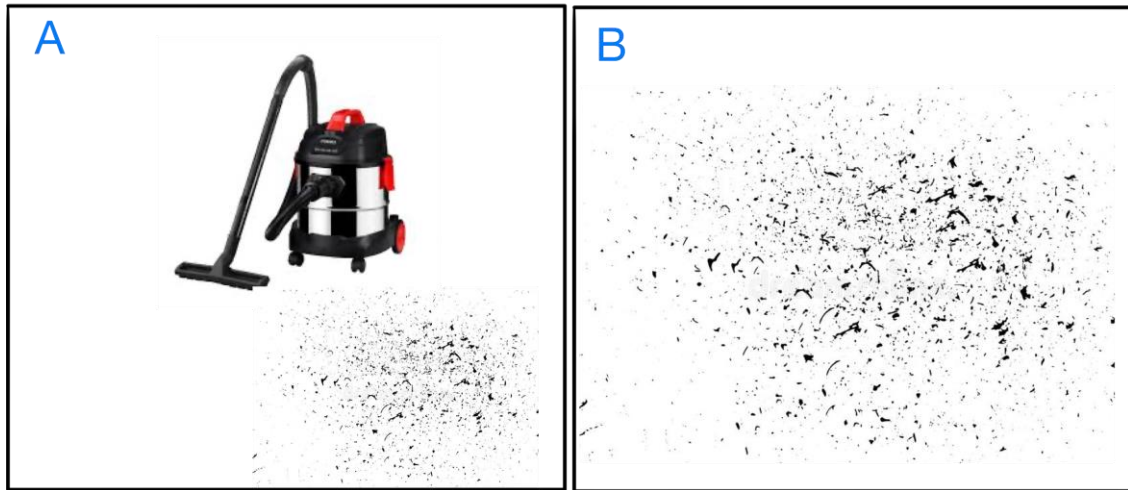
Según (Wooldridge y Jennings, 1995) para calificar a una máquina como un agente inteligente debe validarse las características anteriores. Pero también los agentes se les puede atribuir otras características para capacidad de resolver problemas como nos plantea (Franklin y Graesser, 1997):

- **Continuidad temporal**, el agente tiene la capacidad de ejecutar su función continuamente en un intervalo de tiempo que el usuario lo requiera.
- **Racionalidad**, el agente siempre realiza una función correcta bajo estrictas normas establecidas por el usuario, depende de los datos adquiridos de su entorno.
- **Veracidad**, el agente no transmite información falsa.
- **Adaptabilidad**, el agente tiene la capacidad de cambiar su comportamiento o función según las necesidades que exija su entorno.

### ***1.2.2. Ejemplo Agente Inteligente***

Para una mejor comprensión del concepto lo relacionamos con el funcionamiento de una aspiradora como lo plantea en (Russell et al., 2011) y se lo representa en la figura 2-1, donde se tiene dos escenarios A y B. La aspiradora automáticamente distingue en que escenario se encuentra y si existe suciedad en él.

El agente o la aspiradora elige la función si se mueve a la derecha o izquierda, aspira o simplemente no hace nada. Para que el agente tome la decisión adecuada según el escenario él se pregunta, ¿Existe suciedad en A o en B?, si es así aspirar el entorno que se encuentra sucio, caso contrario moverse al otro entorno. Parcialmente la función del agente inteligente se lo plantea en la tabla 1-1.



**Figura 2-1:** Interpretación entorno de trabajo de una aspiradora, escenario A y B.  
**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

**Tabla 1-1:** Representación de las funciones de la aspiradora figura 5-1.

SECUANCIA DE PERCEPCIONES	FUNCIÓN
[A, Limpio]	Moverse a la Derecha
[A, Sucio]	Aspirar
[B, Limpio]	Moverse a la Izquierda
[B, Sucio]	Aspirar

**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

Como se puede observar en la tabla 1-1, las funciones del agente quedan a criterio del programador de las rutinas, mientras más funciones se empleen, mejoramos la capacidad de reacción con el entorno y la tarea que realiza es inteligente según los parámetros establecidos.

### 1.3. Técnicas Computacionales.

La computación se ha convertido en una herramienta imprescindible para resolver y colaborar con varios procesos en general pero ¿Qué es la computación?, es un proceso que realiza una función en particular con la ayuda de un dispositivo informático o electrónico, el mismo que tiene como objetivo proporcionar soluciones precisas de una manera correcta y flexible, normalmente estos procesos se encuentran fundamentados en teoremas y funciones matemáticas que ayudan a obtener una solución aproximada o exacta de un problema en particular.

A lo largo del tiempo se ha desarrollado varios métodos computacionales utilizados en la informática, pero los mismos se los puede clasificar e identificar en dos metodologías como es el Soft computing y el Hard computing. Estos métodos se basan principalmente en como resuelven un problema y su manera de interpretarlos, el Hard computing usa una metodología analítica más precisa, determinista y exacta, mientras que el Soft computing tiene un enfoque moderno basado en una verdad parcial, aproximación y estocástico.

### ***1.3.1. Hard Computing.***

Hard computing tradicionalmente se ha enfocado en un modelo analítico establecido y caracterizado con precisión. Esta metodología se basa en modelos matemáticos y algoritmos exactos para resolver problemas secuenciales ante una entrada de datos binarios y por ende se justifica una respuesta binaria.

Considerando un ejemplo, si en un escenario planteamos la siguiente pregunta, ¿El día de hoy lloverá?, la respuesta debería ser sí o no, esto significa que las posibilidades son estrictamente limitadas a un análisis determinista y obtener una respuesta binaria.

### ***1.3.2. Soft Computing***

A principios de la década 1990, Soft computing se convirtió en un área formal de estudio dentro de las Ciencias de la Computación, el Dr. Lotfi Zadeh Ingeniero Electrónico y pionero de la lógica difusa e inteligencia artificial (Dzitac, Filip y Manolescu, 2017), interpretó el mundo real lleno de ambigüedades a una representación matemática, según él, Soft computing es un enfoque que intenta imitar la mente humana para razonar y aprender en un entorno de incertidumbre.

Soft computing nació de la necesidad de resolver problemas no lineales como, por ejemplo, un hombre va a una barbería y pide al barbero que le corte un poco el cabello, el profesional juzga fácilmente cuánto es un poco. Pedirle al cliente que especifique una longitud precisa es poco razonable (Hagerty, 2017).

Métodos computacionales como Hard computing pueden modelar y analizar con precisión solo sistemas relativamente simples, pero existe sistemas más complejos que surgen en diferentes áreas como en biología, la medicina, política, economía y campos similares. El Soft computing trata con la imprecisión, la incertidumbre, la verdad parcial y la aproximación para lograr una respuesta predictiva muy robusta (Gupta y Kulkarni, 2013).

En la tabla 2-1 se resume las diferencias entre Soft computing y Hard computing.

**Tabla 2-1:** Diferencias entre Soft computing y Hard computing.

<b>HARD COMPUTING</b>	<b>SOFT COMPUTING</b>
Modelo analítico basado en la precisión.	Modelo basado en la aproximación, verdad parcial y la incertidumbre.
Lógica Binaria	Lógica Difusa.
Naturaleza Determinista.	Naturaleza Estocástica.
Trabaja solo con datos de entrada exactos.	Trabaja con datos ambiguos.
Cálculo únicamente secuencial.	Puede realizar cálculos paralelos.
Resultados precisos.	Resultados aproximados.

**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

Las técnicas que usa Soft computing se asemejan a procesos biológicos y estos son:

- Lógica Difusa.
- Algoritmos Genéticos.
- Redes Neuronales.

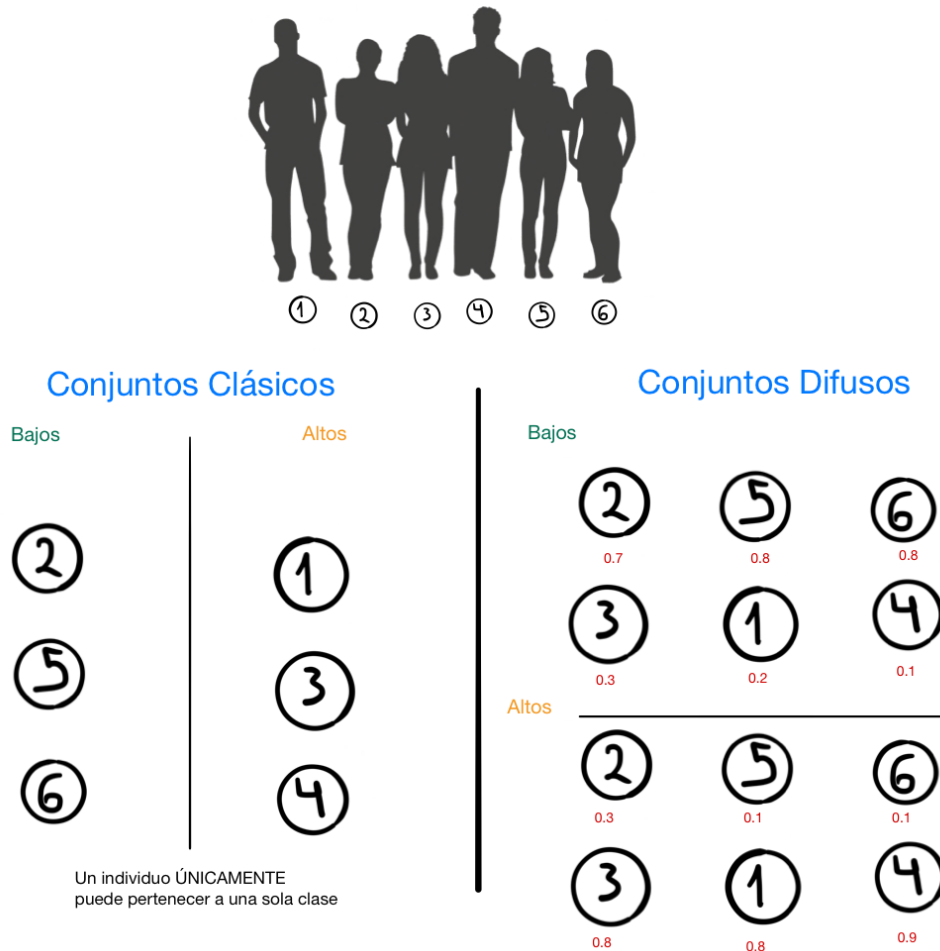
#### **1.4. Lógica Difusa.**

El término conjunto difuso fue propuesto por primera vez en el año de 1965 por el Dr. Lotfi Zadeh en su artículo científico “Fuzzy Sets” (Zadeh, 1965) donde nos explica que la mayoría de veces, las clases de objetos que se encuentran en el mundo físico no tienen criterios de membresía definidos con precisión, por ejemplo si nombramos a una clase “Hombres altos” o “Mujeres hermosas” estas clases no tienen sentido matemático habitual. Sin embargo, el hecho es que tales clases definidas de manera imprecisa juegan un papel importante en el pensamiento humano.



### Diferencia entre Conjuntos Certeros (clásicos) y Conjunto Difusos.

Realizando un ejemplo para una mejor diferenciación, dentro de un grupo de personas de diferente estatura, existe la necesidad de clasificarlos según su tamaño.

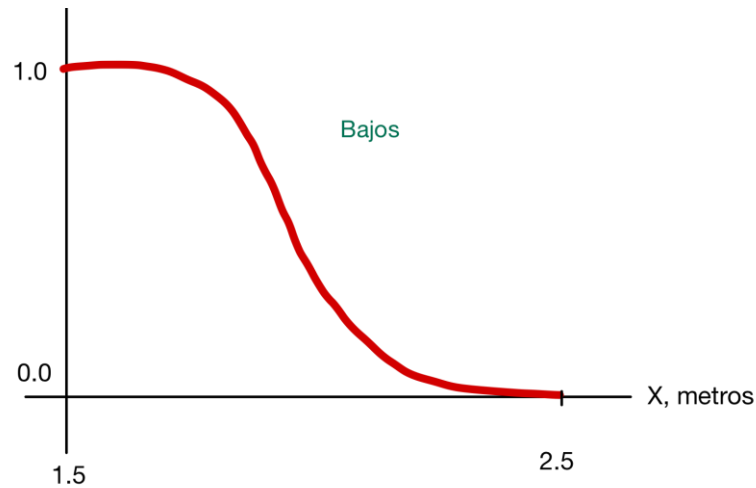


**Figura 3-1:** Conjuntos certeros vs Conjuntos difusos  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

En la figura 3-1, para la clasificación de las personas según su tamaño en los conjuntos certeros se establece un criterio previo, una estatura determinada, si sobrepasan dicho rango es alto caso contrario es bajo. Para la clasificación en conjuntos difusos es diferente, todas las personas del grupo pueden pertenecer en cierta medida a ambos conjuntos, los números rojos indican el valor de pertenencia que normalmente su rango es  $[0, 1]$  donde el 1 quiere decir alta pertenencia y los números que tienden a 0 es de baja pertenencia. La diferencia con los conjuntos certeros, un individuo si puede pertenecer a más de una clase.

Existen dos conceptos fundamentales para la manipulación de conjuntos difusos. El primero es el **Universo de Discurso**, que es la totalidad de los elementos que se está clasificando, en este caso

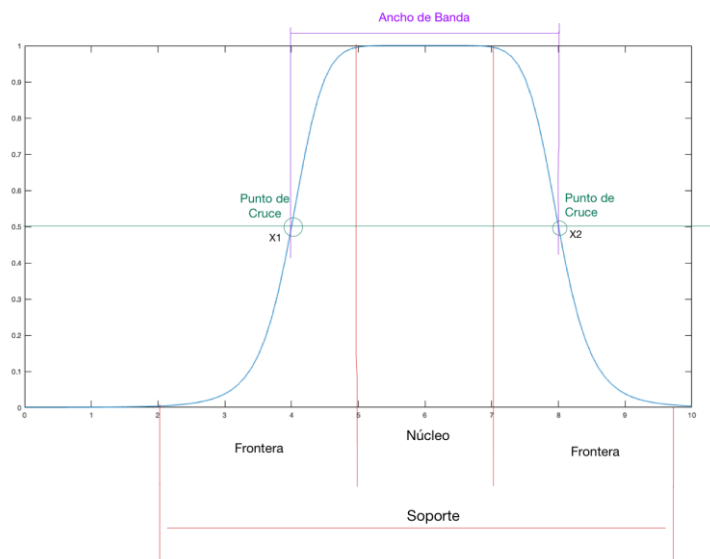
son 6 personas que pertenecen a nuestro universo de discurso, otra forma para representar nuestro universo de discurso es de la siguiente manera  $X \in [1.0, 2.5]$  metros. El segundo concepto importante es la **Función de Membresía**  $\mu(x)$  que es la medida de pertenencia de un elemento al conjunto, aquella función de membresía tiene un Dominio = universo de discurso  $x \in X$  y la Imagen  $\mu \in [0,1]$  en la figura 4-1 son los números rojos.



**Figura 4-1:** Nivel de pertenencia vs Universo de discurso.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

#### 1.4.1. Funciones de Activación RELU.

Las funciones de membresía son fundamentales para el diseño de controladores difusos para ello empezaremos primero describiendo sus características:



**Figura 5-1:** Características función de membresía.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

$$\text{Núcleo } (A) = \{x | \mu_A(x) = 1\} \quad (1-6)$$

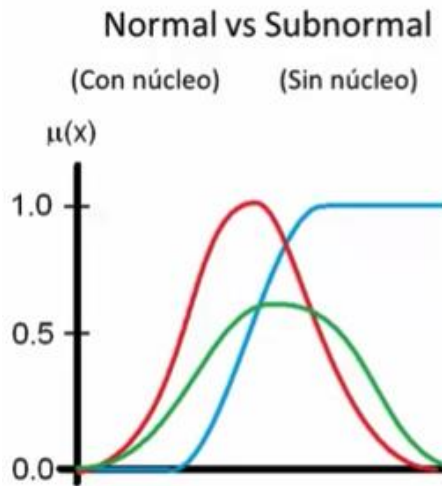
$$\text{Fronteras}(A) = \{x | 0 < \mu_A(x) < 1\} \quad (1-7)$$

$$\text{Soporte}(A) = \{x | \mu_A(x) > 0\} \quad (1-8)$$

$$\text{Cruce}(A) = \{x | \mu_A(x) = 0.5\} \quad (1-9)$$

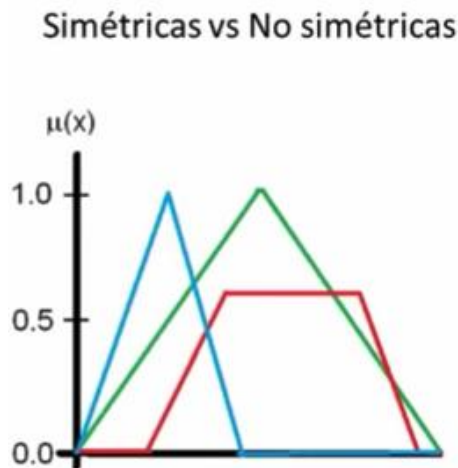
$$\text{Ancho}(A) = |x_2 - x_1| \quad (1-10)$$

Para poder identificar de una mejor manera las funciones de membresía las podemos clasificar en tres grupos, el primero Normal-Subnormal figura 6-1, el segundo Simétricas-No simétricas figura 7-1 y el tercero abierta por (derecha-izquierda) figura 8-1.



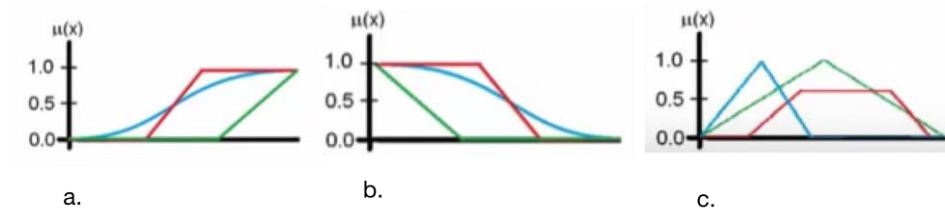
**Figura 6-1:** Funciones de membresía Normal - Subnormal  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

Las funciones normales siempre o el menos en algún punto dentro del universo de discurso toman el valor de membresía máximo o igual a 1, mientras que las funciones subnormales no lo hacen y se consideran funciones sin núcleo como es el caso de la función verde de la figura 6-1.



**Figura 7-1:** Funciones de membresía Simétricas – No simétricas.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

En el ejemplo de la figura 7-1 la función verde se identifica como una función de membresía simétrica ya que alrededor de su punto central se encuentran la simetría perfecta en su mitad del universo de discurso.

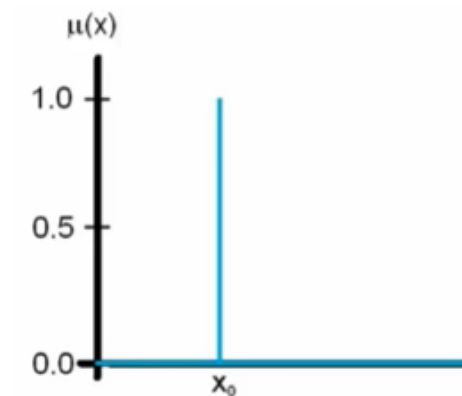


**Figura 8-1:** Funciones de membresía a. Abierta a la derecha b. Abierta a la Izquierda c. Cerrado  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

**Funciones de membresía especiales.**

**1) Función Singleton. -**

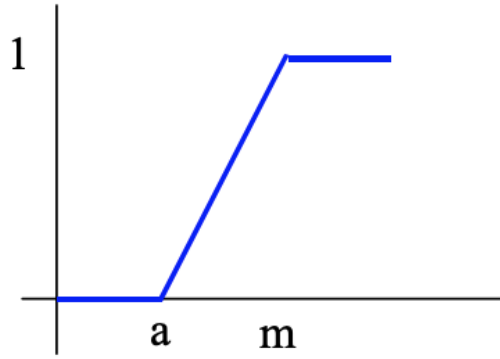
$$\mu_A(x_0) = \begin{cases} 1 & x = x_0 \\ 0 & x \neq x_0 \end{cases} \quad (1-11)$$



**Figura 9-1:** Función de membresía Singleton  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

**2) Función GAMMA. -**

$$\mu_A(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{m-a} & a < x < m \\ 1 & x \geq m \end{cases} \quad (1-12)$$

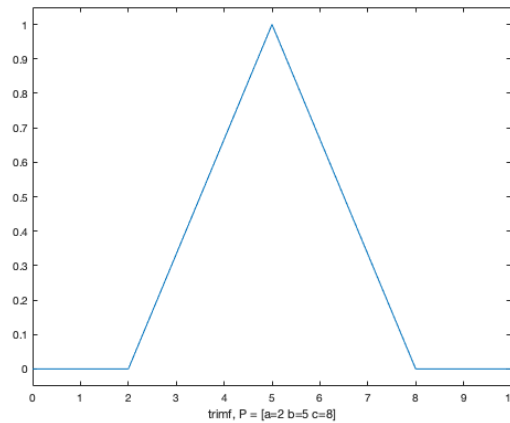


**Figura 10-1:** Función de membresía GAMMA.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

**3) Función Triangular. –**

$$f(x; a, b, c) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & x \geq c \end{cases} \quad (1-13)$$

$$y = \text{trimf}(x, [a \ b \ c]) \Rightarrow a \leq b \leq c$$

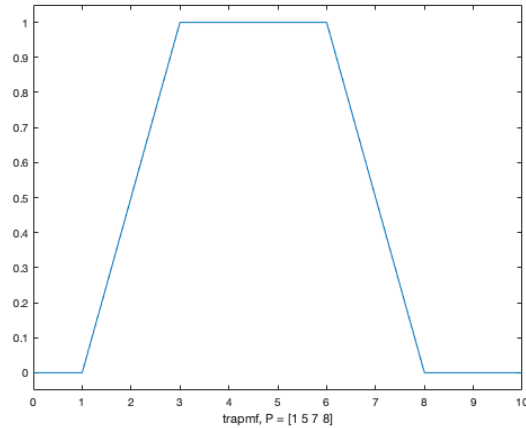


**Figura 11-1:** Función de membresía Triangular.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

**4) Función Trapezoidal. -**

$$f(x; a, b, c, d) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & x \geq d \end{cases} \quad (1-14)$$

$$y = (x, [a \ b \ c \ d]) \Rightarrow a \leq b \leq c \leq d$$

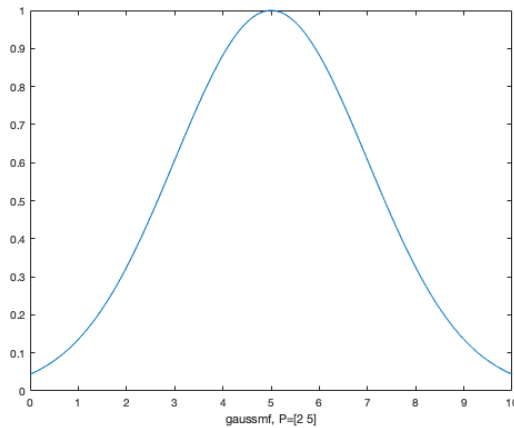


**Figura 12-1:** Función de membresía Trapezoidal.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

**5) Función Gaussiana. -**

$$f(x; \sigma, x_0) = e^{-\frac{1}{2}\left(\frac{x-x_0}{\sigma}\right)^2} \quad (1-15)$$

*σ determina el ancho*  
*x<sub>0</sub> fija el centro*  
*y = gaussmf(x[sig x<sub>0</sub>])*

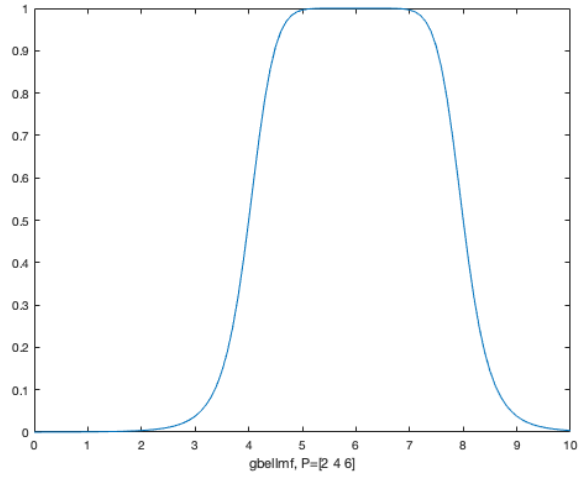


**Figura 13-1:** Función de membresía Gaussiana.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

**6) Función Campana Generalizada. -**

$$f(x; a, b, x_0) = \frac{1}{1 + \left|\frac{x-x_0}{a}\right|^{2b}} \quad (1-16)$$

*a determina el ancho*  
*b determina la pendiente*  
*x<sub>0</sub> fija el centro*  
*y = gblmf(x,[a b x<sub>0</sub>])*

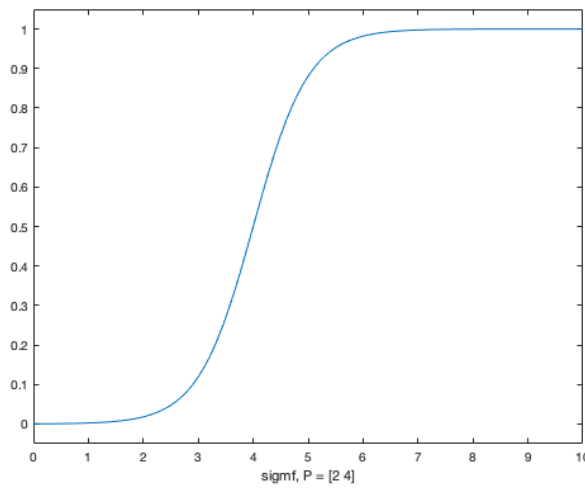


**Figura 14-1:** Función de membresía Campana Generalizada.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés. 2020

**7) Función Sigmoidal. -**

$$f(x, a, x_0) = \frac{1}{1 + e^{-a(x-x_0)}} \quad (1-17)$$

*a* determina la pendiente  
*x*<sub>0</sub> fija el punto de cruce  
*y* = sigmf(*x*, [*a* *x*<sub>0</sub>])

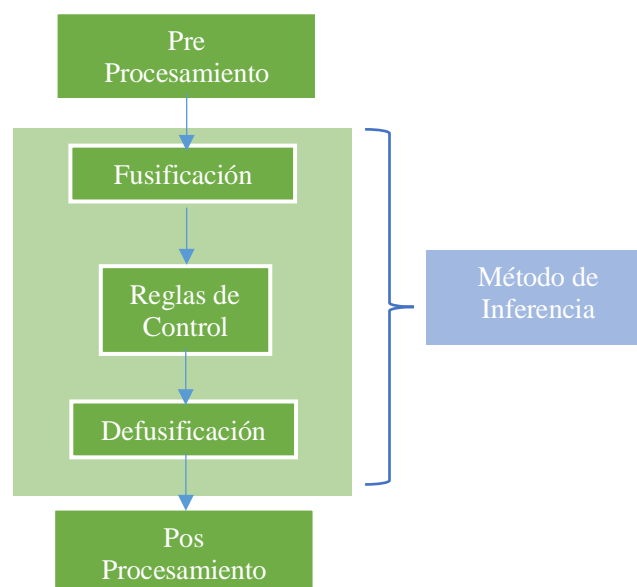


**Figura 15-1:** Función de membresía Sigmoidal.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés. 2020

### 1.4.2. Controlador Difuso

Para establecer un controlador difuso se debe emplear 6 etapas las cuales son:

1. **Pre-Procesamiento.** - Etapa en la cual se acondicionan las señales antes de ser introducidas al procesador, aquí se aplican etapas de filtrado, amplificación de señales, etc.
2. **Fusificación.** - Se realiza la conversión de la señal en valores difusos, para obtener el valor de pertenencia que tiene en cada uno de los valores lingüísticos.
3. **Reglas de control.** - Es el conjunto de reglas lingüísticas (if – them) para definir el control del proceso para el sistema.
4. **Método de inferencia.** - Es el algoritmo que infiere a una conclusión a partir de las señales de entrada y las reglas de control.
5. **Defusificación.** - Convierte el conjunto difuso resultante de la inferencia en una cantidad certera para generar la señal de control.
6. **PosProcesamiento.** - Etapa final en la cual genera una señal de salida acondicionada para el actuador.



**Gráfico 1-1:** Proceso para implementar un controlador difuso.  
Realizado por: GUALLAZACA GONZÁLEZ, Carlos Andrés. 2020



### ***1.4.3. Ventajas y Desventajas del controlador difuso***

- **Ventajas.**

- Razonamiento similar a la lógica humana.
- Alta precisión.
- El razonamiento y conocimiento humano representado en funciones y en reglas de membresía.
- Basado en un modelo lingüístico.

- **Desventajas.**

- Para mayor precisión, se necesita aumentar las etiquetas o calificaciones difusas, lo que incrementa exponencialmente el proceso de las reglas de membresía.
- Tiempo de ejecución del sistema es extenso.
- Falta de respuesta en tiempo real.
- Número restringido del uso de las variables de entrada.

### **1.5. Algoritmos Genéticos.**

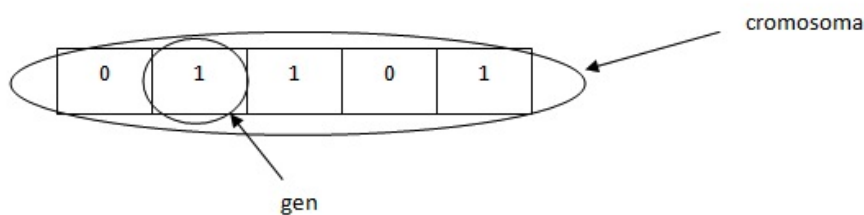
Existen varios problemas creados en la ciencia, en ingeniería y hasta en la vida diaria que pueden entenderse como la búsqueda de un máximo o un mínimo de una función matemática, llamados problemas de optimización. Ejemplo, si una función es el tiempo que tarda un individuo en trasladarse de un punto A a un punto B, y el deseo es minimizar el tiempo que tarda en esta actividad. Las variables se definen como las trayectorias del recorrido, para resolver este problema de optimización y se lo puede realizar con diferentes métodos matemáticos como análisis o álgebra, pero también existen métodos que se basan en la naturaleza, ya que existen diferentes fenómenos que pueden adaptarse a un entorno determinado y esta adaptación se puede entender como una forma de optimización.

Los GAs es un método inspirado en la teoría de la evolución escrita por Darwin, que reside su capacidad de simular la adaptación biológica, de esta manera se pueden explorar diferentes soluciones para escoger la mejor y aplicarse a una gran variedad de situaciones. Existen varios sistemas complejos con problemas de optimización que no se pueden resolver por los métodos matemáticos tradicionales, como por ejemplo funciones aleatorias, discontinuas o problemas que no se los puede definir mediante una función que es propio a un escenario que se enfrenta la IA (Ojeda Ch., 1998).

En síntesis, los algoritmos genéticos parten de una población definida de la cual se selecciona a los individuos óptimos o fuertes para una reproducción futura o mutación, que generará una población de individuos mejor capacitados que la generación anterior.

### 1.5.1. Operaciones Biológicas

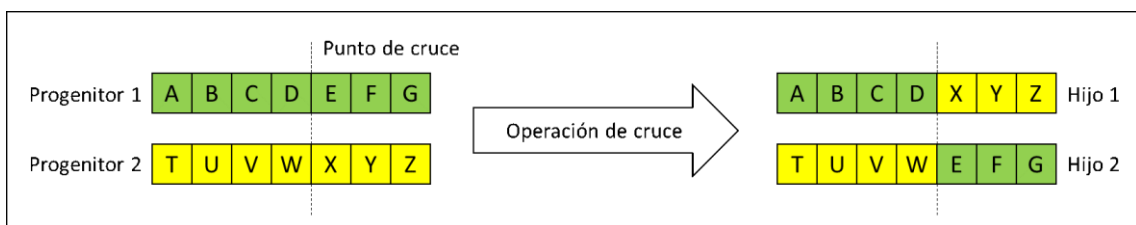
La célula es la unidad básica de todo ser vivo, por lo tanto, en cada una de las células, existe el mismo conjunto de cromosomas. Los cromosomas son cadenas de ADN, y a los cromosomas se los representa en cadenas de números binarios de 0 y 1.



**Figura 16-1:** Representación Cromosoma y Gen en una cadena Binaria.

Realizado por: GUAILLAZACA GONZALEZ, Carlos Andrés.

- **Selección:** Escoge a los cromosomas o individuos entre la población general para una futura reproducción. Mientras mejor esté ajustada la función fitness, más veces será seleccionado para reproducirse.
- **Cruce o Crossover:** Esta operación permite generar nuevos individuos a partir de padres que se escogen aleatoriamente en una cadena de cromosomas.



**Figura 17-1:** Operación Cruce.

Realizado por: GUAILLAZACA GONZALEZ, Carlos Andrés.

- **Mutación:** Es la operación realiza un cambio aleatorio de un gen que permite al programa no caer en un bucle infinito.



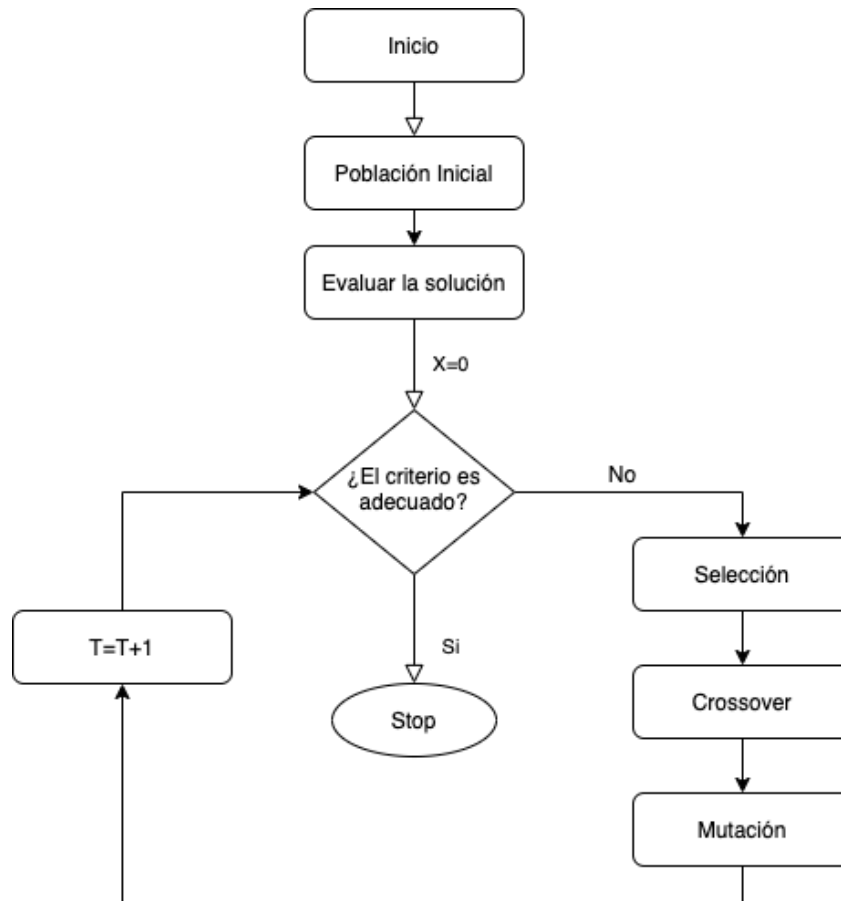
**Figura 18-1:** Operación Mutación.

Realizado por: GUAILLAZACA GONZALEZ, Carlos Andrés.

### 1.5.2. Diagrama de flujo de un algoritmo genético.

En la figura 19-1 refleja el diagrama de flujo que sigue el algoritmo genético para dar solución a un problema de optimización.

- [inicio]: Generar aleatoriamente “n” cromosomas (posibles soluciones para el problema).
- [Fitness Función]: Función de ajuste, ayuda a evaluar  $f(x)$  para cada cromosoma  $x$  en la población inicial.
- [descendencia]: Da origen a una nueva generación de individuos.
  - [selección]: Selección de cromosomas.
  - [crossover]: Cruza material genético entre cromosomas.
  - [mutación]: Mutación de un gen.
- [reemplazo]: Reemplaza a la población anterior por la nueva generación creada por los operadores.
- [test]: Verifica el nivel de satisfacción de la nueva descendencia.
- [loop]: Retorna al paso 2.



**Figura 19-1:** Algoritmo General de los GA.  
 Realizado por: GUAILLAZACA GONZALEZ, Carlos Andrés.

### 1.5.3. Ventajas y Desventajas de los GA.

- **Ventajas.**

- Puede encontrar soluciones idóneas en menor tiempo, todo depende de la heurística definida.
- La mutación aleatoria garantiza una amplia gama de soluciones.
- Optimiza funciones continuas y discretas.
- Útil cuando la población inicial de datos y el espacio de búsqueda es muy grande y hay una gran cantidad de parámetros.

- **Desventajas.**

- Método complejo al encontrar una buena heurística que refleje la función del algoritmo.
- Puede encontrar soluciones idóneas en menor tiempo, todo depende de la heurística definida.
- Al ser un método estocástico, no hay garantía sobre la optimización o la calidad de la solución.
- Si no se implementa correctamente, el GA puede no converger a la solución óptima.

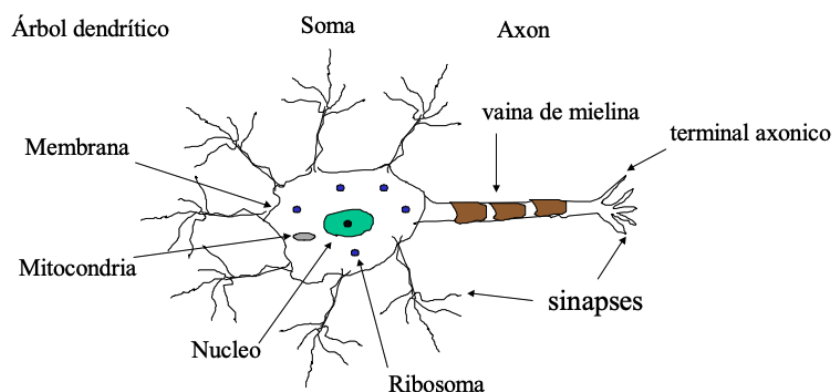
## 1.6. Redes Neuronales Artificiales.

Las Redes Neuronales Artificiales, es una técnica de Softcomputing que permite desarrollar los algoritmos más importantes del machine learning, el cual como modelo computacional existe ya desde mediados del siglo pasado, pero con la mejora exponencial de la tecnología especialmente del hardware se ha empezado a utilizar de una manera constante dando soluciones a problemáticas en diferentes áreas del conocimiento. Por ejemplo: el reconocimiento de caracteres, de imágenes, predicciones bursátiles, traducción de idiomas, generación de texto, predicción de fraude en sistemas bancarios, conducción autónoma, análisis genético, diagnóstico de enfermedades y clasificación de productos en el sector industrial, las redes neuronales es un sistema de algoritmos muy potentes con los que podemos modelar comportamientos de inteligencia racional humana.

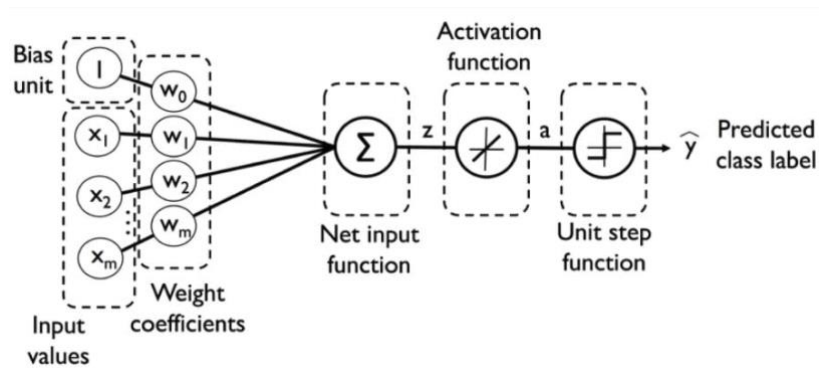
En la mayoría de las estructuras computacionales avanzadas o sistemas electrónicos complejos, la complejidad de estos sistemas emerge de la interacción de partes simples que cooperan entre sí, en redes neuronales a esta parte fundamental se la conoce como Neurona.

### 1.6.1. Neurona.

Una neurona es la unidad básica del procesamiento dentro de una red neuronal, similar a una neurona biológica, estas tienen conexiones de entrada a las que llegan los estímulos externos y conexiones de salida que es la reacción a la señal de entrada. La figura 20-1 es la representación de una neurona biológica, lo que inspiró a desarrollar un concepto de Neurona Artificial presentado en la figura 21-1.



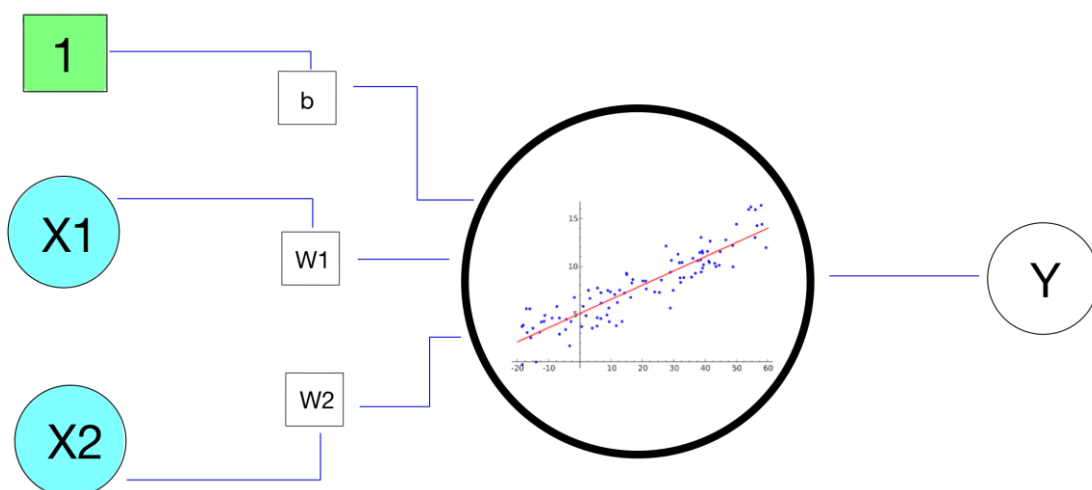
**Figura 20-1:** Representación de una Neurona Biológica.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés. 2020



**Figura 21-1:** Representación de una Neurona Artificial.  
Fuente: <https://www.simplilearn.com/multilayer-artificial-neural-network-tutorial>

En la figura 21-1, el  $f(x)$  que se encuentra dentro de la neurona, representa una suma ponderada de cada una de las entradas por el peso  $w$  que se les asigna a las conexiones de entrada la cual ayuda a definir la intensidad de la variable de entrada que afecta a el comportamiento de la neurona.

Los pesos  $w$  presentes en la figura 22-1 son los parámetros de entrada de los modelos, y matemáticamente se representa como los modelos de regresión lineal lo cual se modifican los valores de una recta o plano para alterar las características propias de la función original, en la regresión lineal se cuenta con un término independiente que mueve a la recta verticalmente, a este valor se le denomina como sesgo (BIAS) que se representa básicamente como otra conexión a la neurona pero la variable siempre se le asigna 1 y se manipula a la función alterando el valor del parámetro del sesgo  $b$ .



**Figura 22-1:** Representación de una Neurona Artificial.  
Realizado por: GUALLAZACA GONZÁLEZ, Carlos Andrés. 2020

$$Y_1 = w_1x_1 + w_2x_2 + b \quad (1-1)$$

Para una mejor comprensión de la función que realiza una neurona para codificar información se hace la analogía con el siguiente ejemplo de la figura 23-1:















**Figura 23-1:** Representación binaria ejemplo de la función de una neurona artificial.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés. 2020

Utilizando la neurona de la figura 22-1 se modela el ejemplo de la figura 23-1, en este caso las variables de entrada  $X_1$  y  $X_2$  serán variables binarias, esto indica, una persona que hace ejercicio y lleva una dieta adecuada y  $Y_1$  variable de salida indica que si aquella persona lleva una vida saludable o no. Las neuronas artificiales como ya se mencionó están basadas en los modelos de regresión lineal y su resultado siempre será un dato numérico continuo y no una variable binaria, para resolver esto se evalúa el resultado de la regresión lineal y si el valor supera un cierto UMBRAL la salida será 1 caso contrario 0 como se muestra en la ecuación 2-1, en la neurona se cuenta con un dato que es el sesgo o BIAS se le asigna un valor opuesto al del umbral  $BIAS = -UMBRAL$  lo que se reescribe en la ecuación 3-1 lo que determina que el cómputo del valor de la neurona es mayor o menor que 0.

$$\begin{cases} W_x \leq UMBRAL & \rightarrow Y = 0 \\ W_x > UMBRAL & \rightarrow Y = 1 \end{cases} \quad (2-1)$$

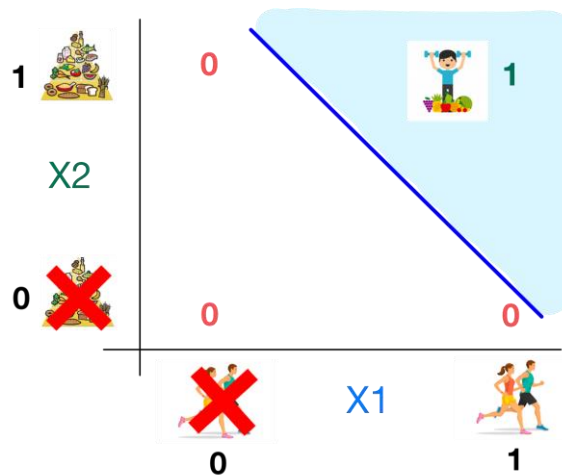
$$\begin{cases} W_x + b \leq 0 & \rightarrow Y = 0 \\ W_x + b > 0 & \rightarrow Y = 1 \end{cases} \quad (3-1)$$

En la figura 24-1, para obtener los valores reales continuos de Y se les asignan valores reales a las variables de entrada, con la combinación de  $Y_1 = 4x_1 + 5x_2 + -6$  solo es posible obtener una respuesta positiva (valor de 1) en la salida cuando ambas variables de entrada están activadas.

X1	X2	Y	Out
		-6	
		-2	
		-1	
		3	

**Figura 24-1:** Representación binaria ejemplo de la función de una neurona artificial.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés. 2020

Visualizando la gráfica anterior se puede representar en un plano cada eje que simboliza a las variables de entrada de la neurona, si uno de los elementos es igual a 0 o faltante el resultado no será el esperado, esto quiere decir que computacionalmente es el modelo de una **compuerta lógica AND**. En la figura 25-1, se identifica la representación de la combinación de parámetros que trazan una recta creando una frontera que divide a dos grupos diferentes, esto ayudará a resolver problemas de clasificación en diferentes áreas.



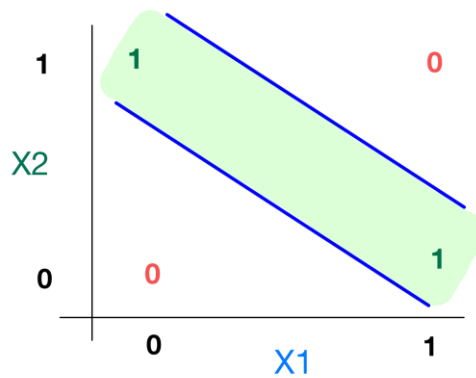
**Figura 25-1:** Representación Recta de regresión lineal del ejemplo 24-1  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés. 2020

### 1.6.2. Combinación de Neuronas.

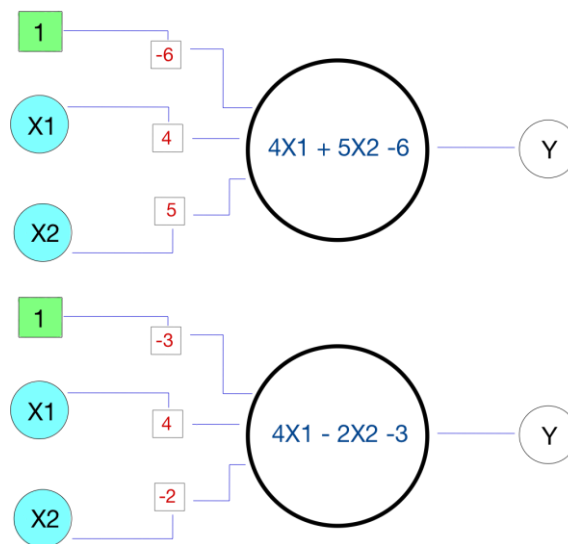
En el ejemplo anterior se explica que utilizando una compuerta AND para dividir el plano en dos fronteras y discriminar la solución es sencillo, pero que ocurre si en la necesidad para obtener un resultado que obligue a utilizar la lógica de la compuerta XOR figura 26-1. Por esta razón es



imposible, resolverlo utilizando una sola neurona para separar linealmente la zona afirmativa y negativa, para ello es necesario utilizar 2 neuronas que cada una creará una recta fronteriza en el plano y así poder resolver el problema planteado, figura 26-1 se representa la creación de la frontera afirmativa (Mehta et al., 2018). Figura 27-1 representa la implementación de dos neuronas.



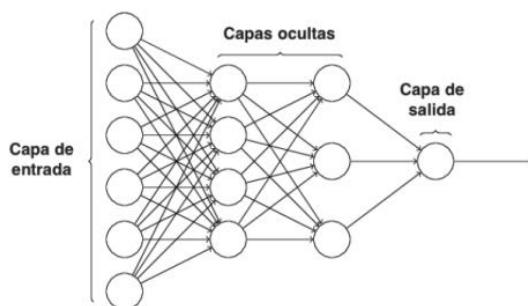
**Figura 26-1:** Representación Recta de regresión lineal operación XOR.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés. 2020



**Figura 27-1:** Combinación de neuronas para dar solución a XOR.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés.

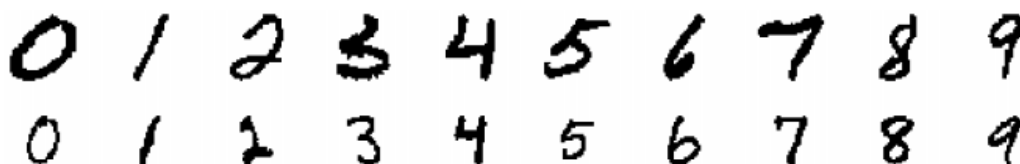
### 1.6.3. Redes Neuronales Multicapa

Las redes neuronales multicapa comúnmente son empleadas para resolver problemas de clasificación para conjuntos no lineales, mediante la adición de capas de neuronas ocultas como se representa en la figura 28-2. Las neuronas no están conectadas directamente a la salida. Las capas ocultas se pueden interpretar geométricamente como hiperplanos adicionales, que mejora la capacidad de separación de la red. (Aizenberg et al., 2008)



**Figura 28-1:** Estructura básica ANN multicapa.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés.

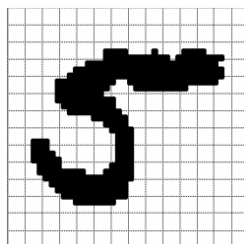
Las capas ocultas desfragmentan el problema y facilitan a la red neuronal a clasificar el problema o patrón. Usando un ejemplo clásico tomado del libro “Reconocimiento de dígitos escritos a mano” (Jain y Zongker, 1997). En la figura 29-1, se representa un conjunto de datos con caracteres de escritura de números a mano los mismos que tienen un tamaño de 784 píxeles pertenecientes a un conjunto de datos MNIST.



**Figura 29-1:** Conjunto de datos MNIST: Caracteres escritura a mano.  
Fuente: (Jain y Zongker 1997)

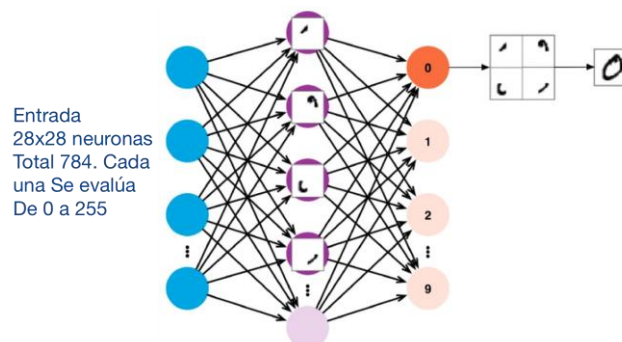
El MNIST contiene más de 70 000 ejemplos de dígitos escritos a mano, esto permite realizar el desarrollo, entrenamiento y pruebas de algoritmos basados en redes neuronales artificiales convolucionales.

Cada uno de los dígitos tiene  $28 \times 28 = 784$  píxeles. Cada píxel toma un valor entre 0 y 255 que corresponde a los códigos de colores RGB, 0 blanco y 255 negro.



**Figura 30-1:** Desfragmentación de la imagen en matriz de datos.  
Fuente: (Jain y Zongker, 1997)

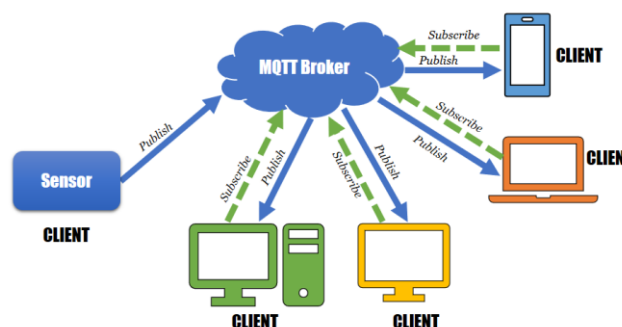
Un sistema informático realmente no puede entender el concepto de ver imágenes o analizar un dígito como lo hacemos los humanos, sin embargo, lo que hace es diseccionar la imagen en una matriz de 784 números como [0, 0, 255, 180, 220, ... .., 2, 180, 0], entonces se puede introducir la información a una red neuronal. **La computadora no entiende una imagen al verla, pero puede entender y analizar los números de pixeles que representa una imagen.** En la figura 31-1 se expresa la metodología representativa de una capa oculta para determinar una selección de una adecuada respuesta.



**Figura 31-1:** Red Neuronal con una capa oculta.  
**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés.

### 1.7. Protocolo MQTT

MQTT es un protocolo de comunicación utilizado comúnmente en el desarrollo IOT por los recursos que ofrece. Este se divide en cuatro etapas: Conexión, autenticación, comunicación y finalización. El bróker MQTT como se observa en la figura 32-1 es el eje central del modelo publicación – suscripción, cuando un cliente (dispositivo) quiere enviar datos a un servidor o intermediario se denomina publicación. Cando la operación se invierte, se llama suscripción. («MQTT», 2014)



**Figura 32-1:** Lógica del protocolo MQTT.  
**Fuente:** <https://morioh.com/p/93ba2353480e>.

### ***1.7.1. ¿Por qué es importante MQTT?***

MQTT fue diseñado para ser compatible con IOT, M2M, y en aplicaciones móviles, tiene las siguientes características:

- Es ligero para que sea posible transmitir grandes volúmenes de datos sin grandes gastos en recursos.
- Emite fácilmente datos de un cliente a muchos.
- Permite ver eventos cada vez que suceden (arquitectura orientada a eventos).
- Funciona muy bien con dispositivos alimentados por batería, el consumo es mínimo.

### ***1.7.2. MQTT en la Industria***

La industria 4.0 es una tendencia actual en la automatización industrial, cuyo objetivo principal es modernizar la forma de fabricación y trabajo de producción al facilitar la integración de dispositivos y mejorar la comunicación entre todos los módulos de control.

El protocolo MQTT en soluciones de Iot se ha popularizado en ventaja al protocolo HTTP (Protocolo de transferencia de hipertexto) estándar propuesto por la IETF cuya deficiencia a soluciones para el internet de las cosas no es efectivo por presentar una latencia alta debido a la saturación del protocolo con los mensajes empaquetados. Por este motivo, MQTT es utilizado como un estándar en M2M (machine to machine) que invoca a un enrutamiento basado en nombres y mitiga el enrutamiento por direcciones IP para flujos de tráfico de Iot (Yokotani y Sasaki, 2016).

La empresa KEB dentro de las soluciones tecnológicas brindadas al campo industrial, cuenta con un enrutador 4G para el control a distancia de PLCs y sensores que se encuentran en campo. El Enrutador C6 está diseñado para aplicaciones industriales robustas ya que es de montaje riel Din y funciona con 24VCD. La particularidad de este enrutador es la capacidad de admitir al protocolo MQTT para mensajería de publicación y suscripción a un bróker personalizado por el usuario final. Esto ayuda a que varios clientes puedan suscribirse al mismo tema y a cada uno puede usar los datos publicados de manera diferente, según lo requiera la aplicación. («MQTT - modern data transmission for Industry 4.0» 2018).



**Figura 33-1:** Implementación del protocolo MQTT en la industria.

Fuente: <https://morioh.com/p/93ba2353480e>.

La seguridad es una consideración importante para dispositivos Iot en un entorno industrial y de aplicaciones de electrónica de consumo. Mqtt es capaz de implementar características de autenticación y seguridad que permiten la integridad de los datos.

KEB con el enrutador C6 es un ejemplo de implementación a una solución integral en el campo industrial prestando las seguridades necesarias con la configuración de TLS y SSL que se realiza en la capa de trasmisión.

Existen muchos usos comerciales del protocolo mqtt para el desarrollo Iot, pero para la implementación industrial llamada Industria 4.0, el uso de MQTT es una alternativa intuitiva que llevarán a los clientes a una implementación de control y trasmisión de datos más personalizadas a un bajo costo de diseño e implementación, esto conlleva a una transformación industrial moderna con la interacción de varios actores que toman decisiones en la producción.

## CAPITULO II

### 2. PROPUESTA Y DISEÑO DE PROTOTIPO

En el presente capítulo se detalla cómo ha sido desarrollado el prototipo clasificador 4.0 basado en machine learning para el control de calidad de productos agrícolas, por lo tanto, se adopta la siguiente metodología:



**Figura 1-2:** Metodología.

Realizado por: GUAILLAZACA GONZALEZ, Carlos Andrés. 2020

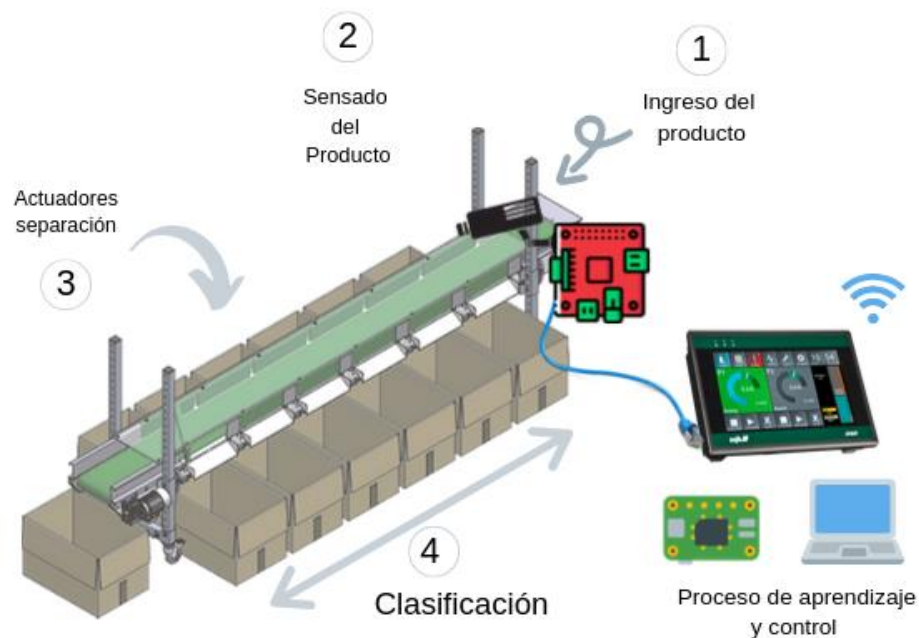
La metodología planteada se orienta al cumplimiento de metas y objetivos que determinan el avance del proyecto para posteriormente plantear nuevos objetivos de cada etapa.

#### 2.1. Requerimientos Generales.

En esta investigación se pretende diseñar y construir un sistema capaz de supervisar, controlar y clasificar cualquier producto manufacturado, objeto o materia prima procedente del agro. Previo al proceso de control y supervisión de calidad, al prototipo se le induce a una etapa de entrenamiento supervisado para que el sistema reconozca y diferencie los productos malos y buenos de una manera intuitiva para que dicha información se registre en la nube y pueda ser monitoreado remotamente por agentes encargados de producción y clientes. Los requisitos que se busca reflejar en dicho prototipo de sistema son:

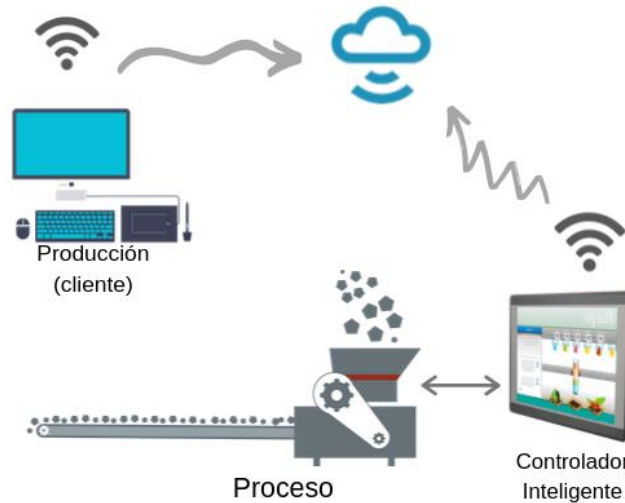
- Capacidad de clasificar productos buenos, regulares y malos.
- Versatilidad para la migración a otros productos.
- Interfaz humano-máquina intuitivo, flexible, sencillo y fácil de interactuar.
- Supervisión de proceso y de los productos clasificados de manera remota.
- Fácil instalación.
- Flexibilidad para la utilización de del sistema mediante una aplicación web.
- Mantener una comunicación fiable entre el clasificador y un Dashboard.

En la figura 2-2 se muestra el esquema de funcionamiento y aprendizaje donde el clasificador está montado sobre la línea de producción acorde a las necesidades del emprendimiento objetivo. El dispositivo procesa los datos para posteriormente determinar el control y clasificación de los productos. Previo a esto se desarrolló una prueba con el objetivo de retroalimentar al sistema con todos los datos necesarios para una clasificación inteligente.



**Figura 2-2:** Arquitectura General del Sistema.  
Realizado por: GUAÍLLAZACA GONZÁLEZ, Carlos Andrés. 2020

El dispositivo debe tener la capacidad de enviar reportes a la nube para que el departamento de producción tenga acceso a datos importantes del proceso. Por lo tanto, se puede tomar decisiones importantes de la empresa con los datos recolectados sobre la producción. Otro aspecto importante que el cliente también tiene a disposición un resumen de estos reportes como se representa en la figura 3-2.



**Figura 3-2:** Esquema del funcionamiento y aprendizaje.  
**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés. 2020

### 2.1.1. Definición de Variables.

Las acciones objetivas del sistema están limitadas al accionamiento de 3 actuadores. Estos se interpretan como las variables de control, en consecuencia, serán los encargados de clasificar y seleccionar productos buenos, regulares y malos.

A continuación, en la tabla 1-2 se representa las variables del sistema.

**Tabla 1-2:** Variables del sistema.

VARIABLES DE CONTROL	VARIABLES CONTROLADAS
Actuador 1	Producto Bueno
Actuador 2	Producto Regular
Actuador 3	Producto Malo o Descarte.

**Realizado por:** GUAILLAZACA GONZALEZ, Carlos Andrés, 2020

### 2.2. Selección del Hardware

Para el correcto funcionamiento del dispositivo clasificador se pensó en la selección de varias tarjetas electrónicas como la tarjeta de desarrollo Raspberry Pi. Las prestaciones de esta tarjeta son adecuadas para el desarrollo de proyectos de visión artificial y aprendizaje automático. Además, cabe mencionar que esta plataforma de procesamiento permite implementar cierto grado



de autonomía y encapsulamiento de hardware debido que se trata de una arquitectura del tipo “Computer-On-Board” muy versátil para la implementación de sistemas embebidos.

Por otro lado, se debe considerar las exigencias de un adecuado trabajo de ejecución en paralelo de las tareas de control y procesamiento de datos mediante técnicas de machine learning e IoT orientado a la industria. Estas tareas exigen simplicidad y flexibilidad en el proceso productivo. Por tal motivo, los algoritmos de entrenamiento y procesamiento de todo el sistema se lo desarrollan en un entorno web con el uso de librerías de KNN mediante la librería ml5.js implementada en JavaScript. Mayores detalles de esta librería se describe en la sección de software.

Por medio de los periféricos de comunicación embebidos en la tarjeta Raspberry es posible implementar vías para el envío y recepción de datos, como por ejemplo a través del puerto Ethernet. Esto ayuda a tener un procesamiento de la información directamente en la nube, y todos los datos de control viajarán hasta un servidor web MQTT, el cual se enlaza a la tarjeta de desarrollo Iot ESP8266. Esta última emite las órdenes a los diferentes actuadores empleados en el proceso.

### ***2.2.1. Dispositivo de Procesamiento control nivel actuadores***

El microprocesador ESP8266 en sus inicios fue utilizado como un módulo extra para los microprocesadores de arduino, su función era enlazar los procesos de arduino con la web. Posteriormente, gracias al desarrollo de la comunidad de hardware libre, se logró un avance fuerte llegando al punto de poder programarlos en C o Python, independizándolo de cualquier otro microprocesador para realizar un procesamiento.

NodeMCU es la tarjeta de desarrollo que se usó en el proyecto ya que cuenta con el procesador ESP8266 y un puerto de comunicación serial para su programación. Adicionalmente, esta tarjeta cuenta con un periférico de wifi, por lo que no necesita de un módulo extra para la comunicación. En la figura 4-2 se presenta el esquema de la distribución del NodeMCU y en la tabla 2-2 una comparativa con otras tarjetas.

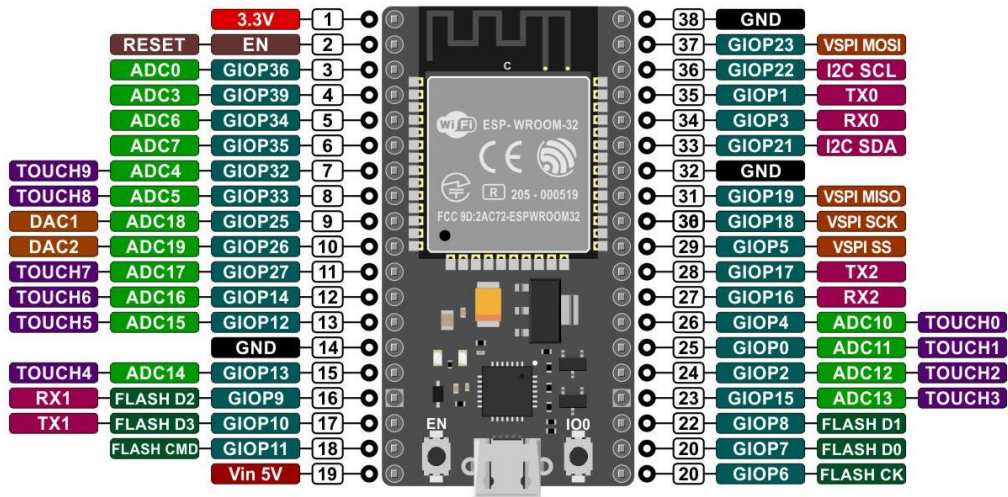


Figura 4-2: NodeMCU ESP8266.

Fuente: <https://forum.fritzing.org/t/esp32s-hiletgo-dev-board-with-pinout-template/5357>

Tabla 2-2: Datos Técnicos del ESP8266

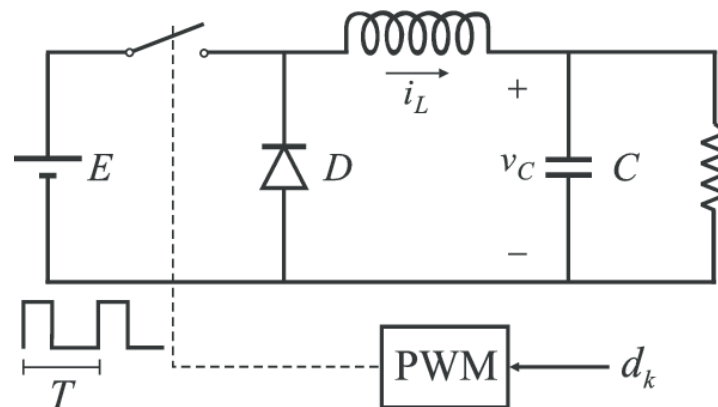
	ESP32	ESP8266	ARDUINO UNO R3
Núcleos	2	1	1
Arquitectura	32 bits	32 bits	8 bits
Wifi	Incorporado	Incorporado	No
RAM	512Kb	160Kb	2Kb
FLASH	16Mb	16Mb	32Kb
GPIO	36	17	14
Interfaces	SPI, I2C, UART, I2S.	SPI, I2C, UART, I2S.	SPI, I2C, UART.
ADC	18	1	6
DAC	2	0	0

Realizado por: GUAILLAZACA GONZALEZ, Carlos Andrés, 2020

### 2.2.2. Etapa distribución y protección energética

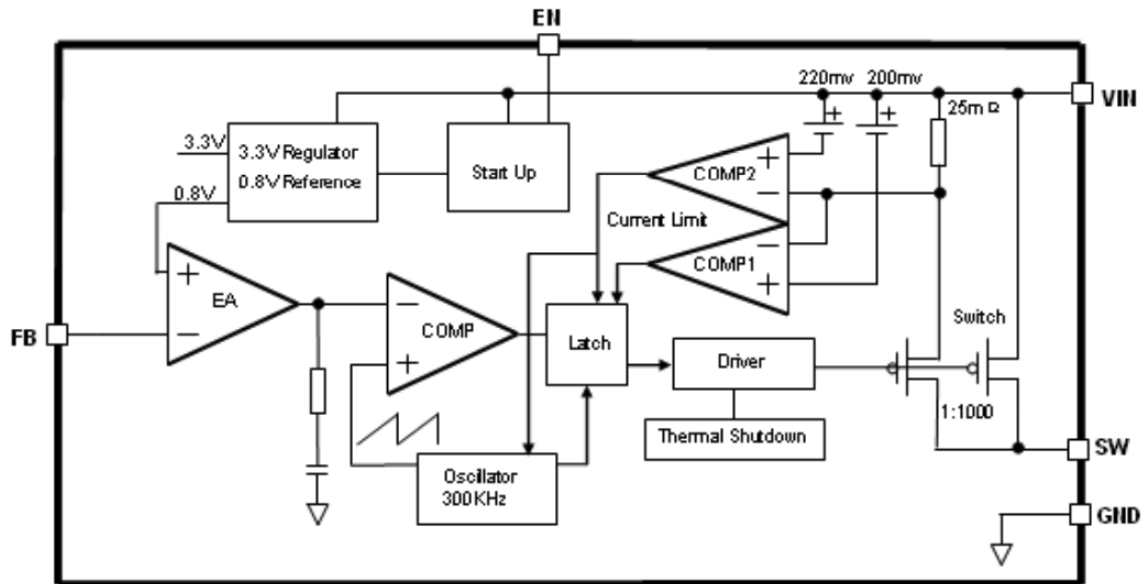
Para la etapa de alimentación energética se hizo un análisis de acuerdo con el requerimiento energético en conjunto de toda la planta. Para ello se ha dividido en 2 necesidades a solventar. La primera es la demanda de la energía del motor principal de la banda transportadora y los actuadores para la clasificación que demandan de una alimentación de 12V a 1.2A (actuadores de separación) y 3A (motor para la trasmisión de la banda transportadora), y, por último, la energía para la etapa de control, comunicación y sensores que necesitan una alimentación de 5V a 1A.

Tomando en consideración este requerimiento en general, se elige una fuente de voltaje de 12V a 15A 180W. La razón de esta sobredimensión en corriente es para solventar las necesidades de cambios y adición de cargas en el transcurso de la ejecución del proyecto. En consecuencia. Para la segunda etapa (control) se requiere reducir el voltaje a 5V de una manera eficiente, para lo cual se descartan los reguladores de voltaje lineales basados en el LM7805, cuyo grado de eficiencia no llega al 70% y la energía que disipa en calor representa un costo energético innecesario. Por esta razón, se optó por un convertidor de voltaje DC-DC tipo BUCK, como el que se presenta en la figura 5-2.



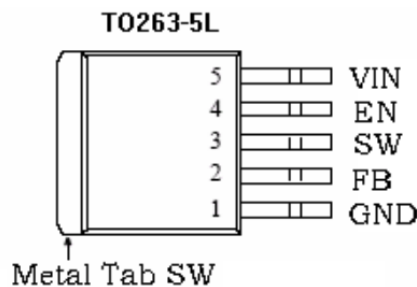
**Figura 5-2:** Convertidor DC-DC Tipo Buck  
Fuente: GUAILLAZACA GONZALEZ, Carlos Andrés, 2020

La importancia de un convertidor DC-DC es mantener un nivel de  $V_{out}$  adecuado y constante, manteniendo un nivel de eficiencia alto en la conversión. Por lo tanto, su correcta operación depende directamente de una minuciosa selección de los componentes electrónicos para su configuración. Es importante mencionar que en el proyecto no fue necesario diseñar e implementar un convertidor que se ajuste a los criterios indicados debido a que se optó por un módulo comercial convertidor Step Down basado en el circuito integrado (IC) XL4005. En la figura 6-2 ilustra el esquema de componentes y conexión interna del IC XL4005 referente a un empaquetado TO263-5L.



**Figura 6-2:** Diagrama interno del circuito integrado XL4005 con empaquetado TO263-5L.  
**Fuente:** <http://www.xlsemi.com/datasheet/XL4005%20datasheet.pdf>

En la figura 7-2 y la tabla 3-2 se explica la configuración de cada uno de los pines de salida del integrado XL4005 con empaquetado TO263-5L y su función, respectivamente. Para la adaptación y conexión del circuito convertidor tipo Buck mostrado en la figura 6-2, es necesario reemplazar el switch de conmutación con el IC XL4005.



**Figura 7-2:** Distribución de los terminales del TO263-5L.  
**Fuente:** <http://www.xlsemi.com/datasheet/XL4005%20datasheet.pdf>

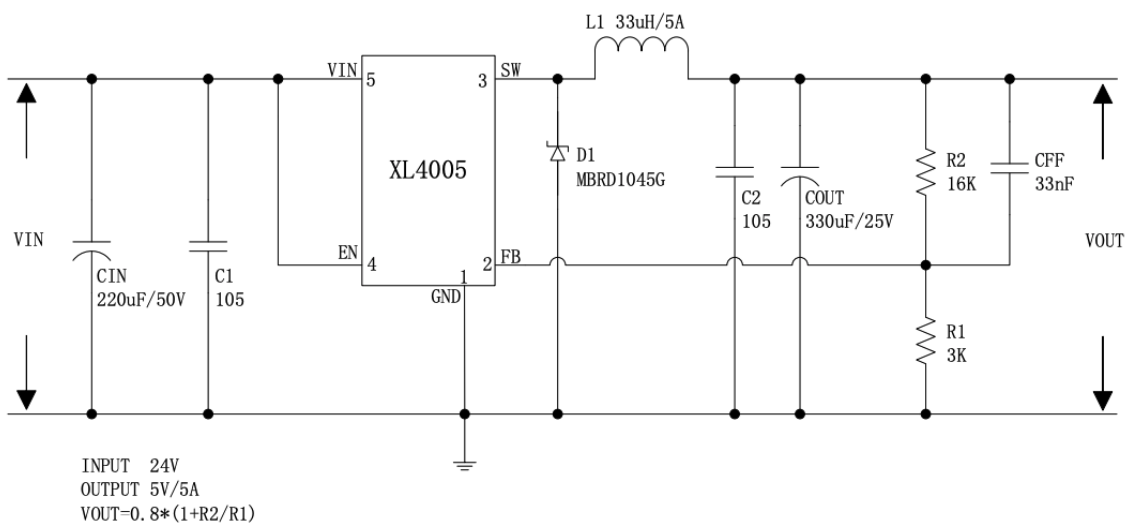
**Tabla 3-2:** Descripción de los terminales del TO263-5L

Pin #	Pin	Descripción
1	GND	Pin de tierra. Se debe tener cuidado en el diseño ya que se debe colocar fuera del diodo Schottky para evitar que los picos de corriente de conmutación induzcan ruido de voltaje.
2	FB	Pin Feedback. Mediante un divisor de voltaje externo el FB detecta el voltaje de salida y lo regula. El voltaje del umbral de retroalimentación es 0.8V
3	SW	Pin de salida al nodo del interruptor.
4	EN	Pin activador. Detecta el estado del dispositivo 1 enciende y 0 apaga.

5	VIN	Pin entrada de voltaje, 5v a 32v DC, importante eliminar el ruido con la selección de un condensador.
---	-----	---

Realizado por: GUAILLAZACA GONZALEZ, Carlos Andrés, 2020

Una vez reemplazando la etapa de conmutación en la configuración general del convertidor tipo Buck por el integrado XL4005, es necesario atenuar el ruido generado por la fuente de entrada con dos capacitores en paralelo, uno electrolítico de 220uF a 50V y otro cerámico de 1uF. Para tener una señal de retroalimentación se utiliza un divisor de voltaje resistivo para sensar el voltaje de salida que retroalimenta al XL4005. Un esquemático que detalla de mejor manera las conexiones explicadas se presenta en la figura 8-2.



**Figura 8-2:** Esquema final del convertidor DC-DC tipo Buck basado en el circuito integrado XL4005.

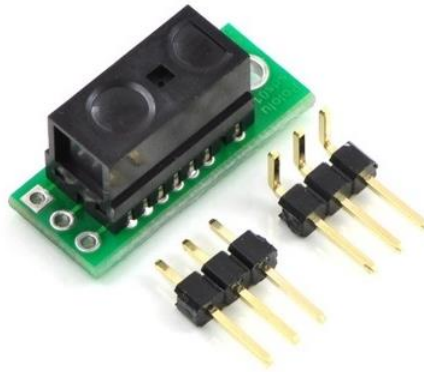
Fuente: GUAILLAZACA GONZALEZ, Carlos Andrés, 2020

### 2.2.3. Etapa de Dispositivos de Adquisición de Datos

En esta etapa se describe los componentes necesarios que retroalimentan de información al microcontrolador *ESP8266* que recibe las órdenes desde el broker *shiftr.io* y controla a los actuadores de separación. Para tener un mejor control de existencia de producto en la banda transportadora y la ubicación de este se implementan sensores digitales que ayudan a minimizar el error y existencia de perturbaciones en la ejecución de la tarea de automatizar la clasificación.

#### 2.2.3.1. Sensor de presencia Digital

Para llevar un control del desplazamiento del producto sobre la banda transportadora, y el lugar donde se depositó el mismo, se utiliza sensores de presencia digitales Sharp GP2y0d810z0f, como el que se ilustra en la figura 9-2.



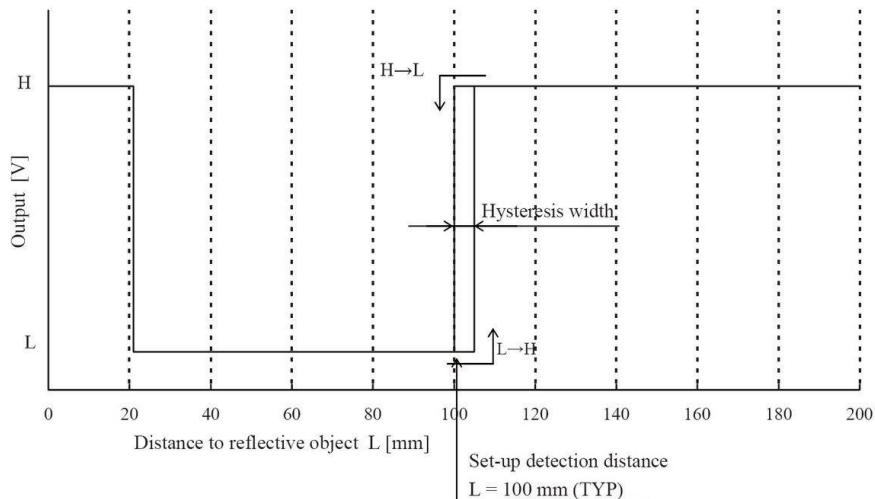
**Figura 9-2:** Sensor Sharp GP2y0d810z0f.  
**Fuente:** <https://www.pololu.com/product/1134>

En la tabla 4-2 se describen las características del sensor Sharp GP2y0d810z0f. Además, en la figura 10-2 se plasma la gráfica característica de los rangos de trabajo y la histéresis en la curva del umbral operacional. En la figura 11-2 se muestra el diagrama electrónico del pcb del sensor donde se identifican los terminales de  $V_{out}$ ,  $V_{in}$  y  $GND$ . Adicionalmente, cuenta con una configuración de activación de señal visual que enciende un led rojo según el estado del sensor. Los capacitores  $C_1 = 0.1 \mu F$ ,  $C_3 = 4.7 \mu F$  cumplen la función de filtrar señales de ruido sobre la línea de  $V_{cc}$  y afectar al microcontrolador U1.

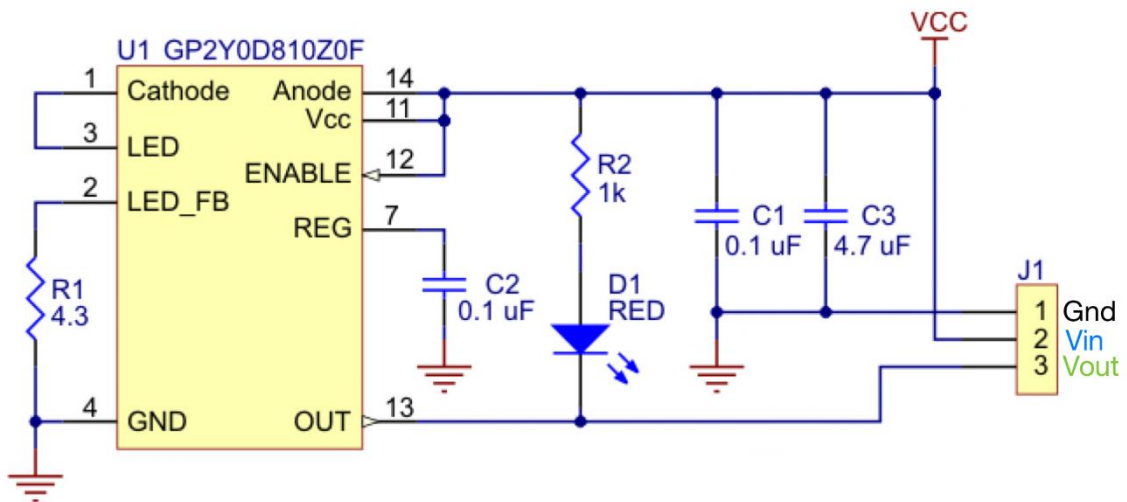
**Tabla 4-2:** Características Sharp GP2y0d810z0f.

CARACTERÍSTICA	DESCRIPCIÓN
Voltaje $V_{in}$	2.7v – 6.2v
Corriente de consumo	5mA
Rango Distancia	0.5cm – 5cm
Tipo de señal	Señal digital (0 cuando se detecta un objeto y 1 en caso contrario).
Periodo de actualización de estado	2.56 ms

**Realizado por:** GUAILLAZACA GONZALEZ, Carlos Andrés, 2020



**Figura 10-2:** Característica de medición del sensor Sharp GP2y0d810z0f.  
**Fuente:** <https://www.pololu.com/product/1134>



**Figura 11-2:** Esquemático Sharp GP2y0d810z0f.  
**Fuente:** <https://www.pololu.com/product/1134>

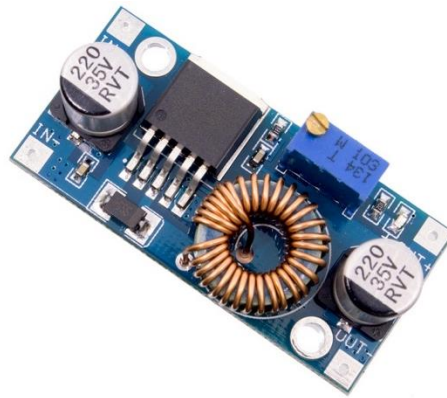
#### 2.2.4. Etapa de Control de Actuadores

Para tener un control secuencial del movimiento de los productos desde la banda transportadora hasta las canastas contenedoras es necesario emplear una etapa de control de los actuadores que reciben la señal desde el microcontrolador ESP8266 que previamente procesa la información en la nube.

##### 2.2.4.1. Driver Motor

Para la transmisión de la banda transportadora se utilizó un motor de 12V a 3A según las pruebas de rendimiento de la banda transportadora con 2kg de carga. Se determinó que no es necesario

utilizar un Driver Motor Puente H porque no se necesita inversión de giro del motor. Por otra parte, se planteó utilizar el accionamiento del motor mediante contactos de un relé, pero ese está ligado a una sola velocidad. Por tal motivo se utilizó otro convertidor DC-DC tipo Buck para poder controlar la velocidad del motor y soportar los picos de corriente elevados que llegan hasta los 4.46A. La etapa de alimentación energética del motor se presenta en la figura 12-2.



**Figura 12-2:** Convertidor DC-DC Tipo Buck XL4005  
**Fuente:** [https://electropro.pe/index.php?route=product/product&product\\_id=605](https://electropro.pe/index.php?route=product/product&product_id=605)

En la tabla 5-2 se detallan todos los datos y características técnicas del convertidor XL4005.

**Tabla 5-2:** Datos Técnicos Convertidor dc-dc Buck.

Característica	Detalle
Voltaje de entrada	5 - 32 VDC
Voltaje de salida	0.8 – 30 VDC
Corriente de salida	5A (corrientes > 3A requiere disipador de calor)
Potencia de salida	75 W
Frecuencia de funcionamiento	300 KHz
Temperatura de funcionamiento	-400 °C a 80 °C

**Fuente:** [https://electropro.pe/index.php?route=product/product&product\\_id=605](https://electropro.pe/index.php?route=product/product&product_id=605)

#### 2.2.4.2. Servo Motor

Para la configuración del prototipo en la parte de la clasificación de separación del producto se instalaron 2 servomotores que accionaran un sistema de compuertas individuales que guían al producto regular y malo. Los servomotores empleados son de la marca Tower Pro-serie Mg995 con una capacidad de torque de 11kg/cm. En la figura 13-2 se ilustra el Servo motor y en la tabla 6-2 se detallan los datos técnicos por los cuales se ha decidido utilizarlos.





**Figura 13-2:** Tower Pro-MG 995  
**Fuente:** <http://www.denontek.com.pk/TOWERPRO-MG995>

**Tabla 6-2:** Datos Técnicos Tower Pro MG995.

CARACTERÍSTICA	DETALLE
Voltaje de operación	4.8 V a 6.6 V DC
Velocidad de operación	0.20 s/60° (4.8 V), 0.16 s/60° (6 V)
Torque	9.4 kgf/cm (4.8 V), 11 kgf/cm (6 V)
Ángulo de Rotación	360°
Dimensiones	Largo 40.7 mm, ancho 19.7 mm, altura 42.9 mm aprox.

**Fuente:** <http://www.denontek.com.pk/TOWERPRO-MG995>

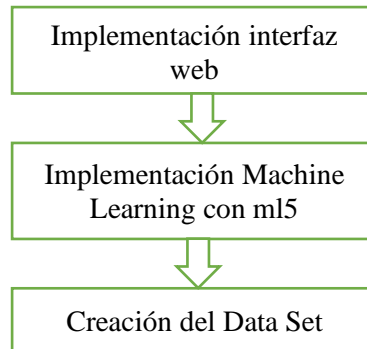
### 2.2.5. Fuente de Voltaje 110v AC - 24v DC

El diseño de un prototipo se debe considerar que puede ser propenso a cambios o adición de elementos electrónicos o actuadores. La selección de la fuente de voltaje para la distribución energética debe tener la capacidad de soportar la demanda de todo el sistema en conjunto. Por tal motivo es adecuado sobredimensionar la demanda de las cargas antes mencionadas. Para ello se ha seleccionado una fuente de voltaje conmutada utilizada en impresoras 3D o CNC open source como se detalla en la figura 14-2. Esta fuente conmutada de voltaje es de 180W 110v AC de entrada y el voltaje de salida es de 12V DC a 15A.



**Figura 14-2:** Fuente conmutada 110v ac – 12v dc 15A 180W.  
**Fuente:** <http://www.denontek.com.pk/TOWERPRO-MG995>

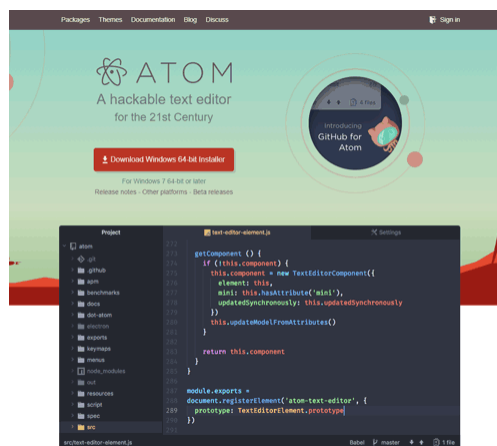
## 2.3. Implementación del Software



**Gráfico 1-2:** Proceso para el desarrollo de software del proyecto.  
Realizado por: GUALLAZACA GONZÁLEZ, Carlos Andrés, 2020.

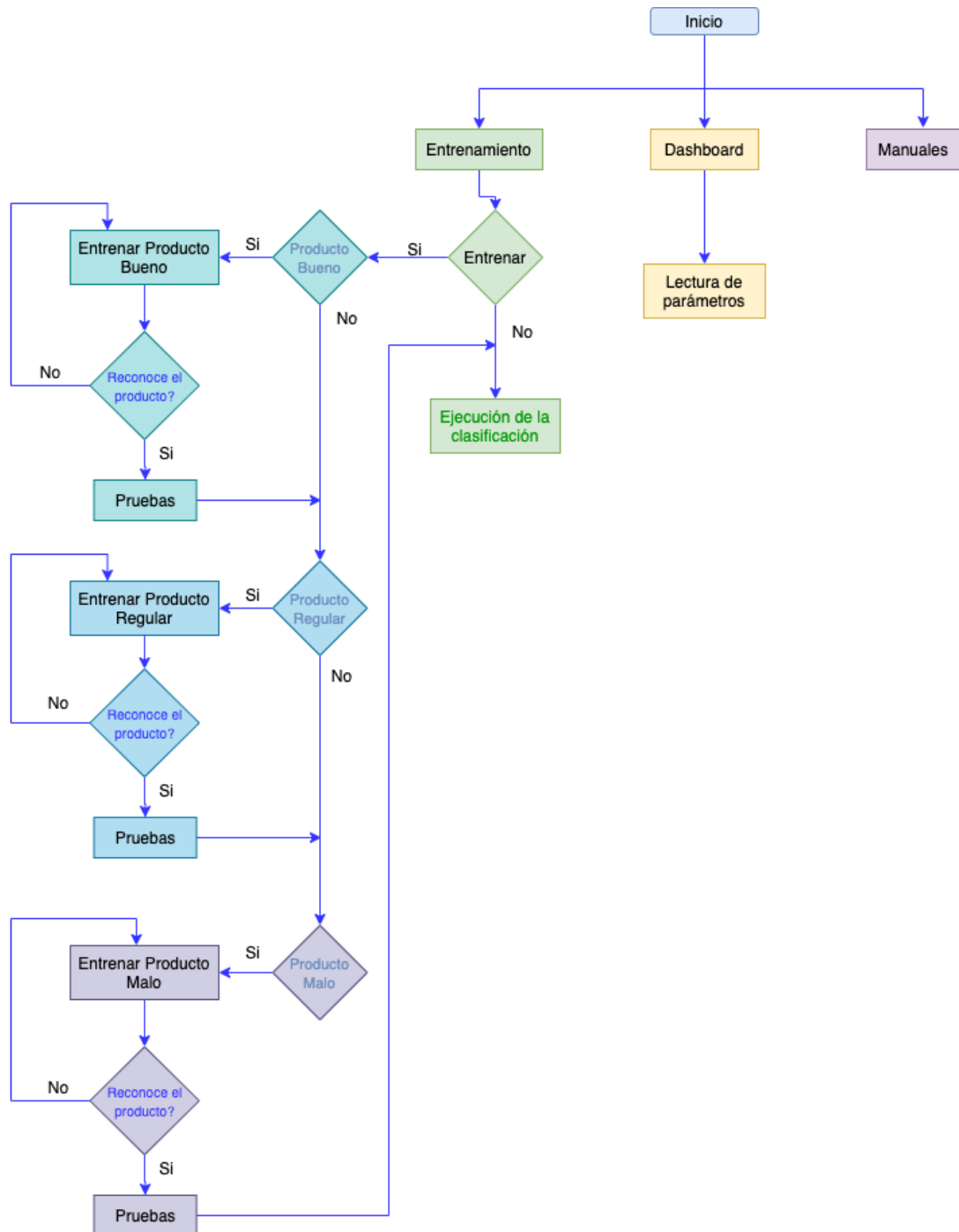
### 2.3.1. Interfaz Web.

Detalla el procedimiento que se realizó para la implementación de la interfaz web y la estructura donde se empleó el algoritmo de la red neuronal. Para el desarrollo de la interfaz se utilizó Atom IDE de GitHub. Este es un editor de código muy versátil en el cual es posible programar en HTML, CSS y JavaScript, en un solo entorno con la facilidad de agregar varios paquetes de manera gratuita para su correcta configuración.



**Figura 15-2:** Editor de Texto Atom IDE GitHub.  
Fuente: <https://atom.io/>

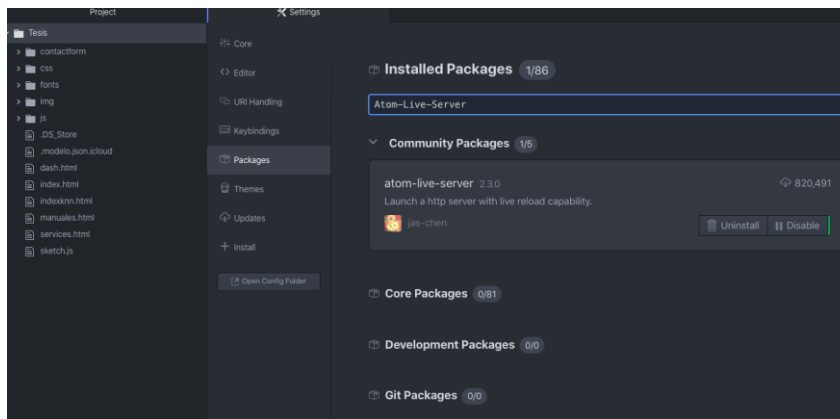
En el gráfico 2-2 se esquematiza el diagrama de flujo con la lógica del funcionamiento de la gestión del sistema, la cual consta de 3 pestañas: Servicios, Dashboard y Manuales. Es importante indicar que dentro del entorno de Servicios se encuentran todas las operaciones para realizar el entrenamiento y puesta en marcha del sistema.



**Gráfico 2-2:** Diagrama de flujo de la aplicación web para la interfaz del proyecto.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

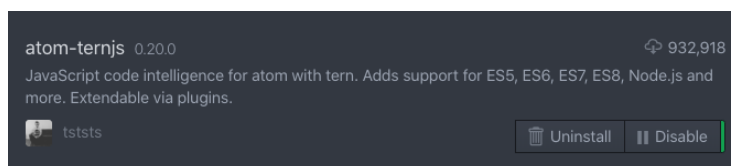
Para el desarrollo de la parte visual del proyecto es necesario instalar unos packages extras dentro de Atom IDE como:

- **Atom-Live-Server.** – Ayuda a iniciar un servidor http de desarrollo simple con capacidad de actualización instantánea. En la figura 16-2 representa a la pantalla de configuración de paquetes y la selección del Atom-Live-Server.



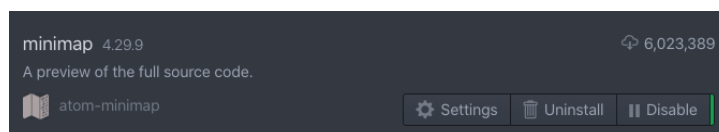
**Figura 16-2:** Paquete Atom-Live-Server.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

- **Atom-ternjs.** - Agrega inteligencia al código y la edición pasa por alto errores de sintaxis, el programa corre sin problema ante errores mínimos figura 17-2.



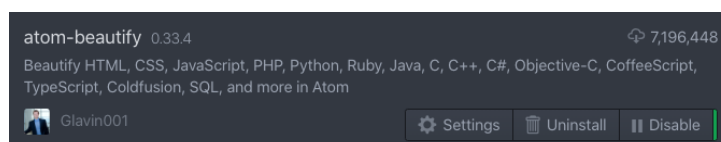
**Figura 17-2:** Paquete Atom-ternjs.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

- **Minimap.** - Es un paquete de ayuda al programador que permite visualizar una ventana de todo el código en miniatura para poder moverse en diferentes partes de este y facilita la localización de elementos en el código figura 18-2.



**Figura 18-2:** Paquete Minimap.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

- **Atom-Beautify.** – un paquete de configuración inicial que normalmente se lo utiliza para estilizar el código fuente de colores y resaltar varias estructuras importantes según el lenguaje de programación que se esté utilizando figura 19-2.



**Figura 19-2:** Paquete Atom-beautify  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

### 2.3.1.1. Pantalla Inicio o Home

El ambiente de bienvenida es importante en un proyecto de aplicación Web. Para esto, se creó una interfaz amigable para el usuario final, y a través de la cual conozca todas las ventajas y limitaciones que tiene el sistema.

Inicialmente se debe crear un archivo index.html que define el contenido de hipertexto o contenido visual marcado dentro de la página web inicial, a continuación, se explica las partes importantes de la página web inicial:

#### Head o cabecera

En este apartado se inicializa declarando todas las sentencias y formatos dados desde estilos de CSS. Se utilizó el framework Bootstrap para crear una plantilla ordenada para todas las pestañas o páginas de la página web y se adaptan a diferentes pantallas sin dificultad. Dicho código se presenta a continuación:

```
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Clasificador Inteligente</title>

  <!-- Bootstrap -->
  <link href="css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="css/animate.css"> // plantilla animaciones
  <link rel="stylesheet" href="css/font-awesome.min.css"> // plantilla letras
  <link rel="stylesheet" href="css/jquery.bxslider.css">
  <link rel="stylesheet" type="text/css" href="css/normalize.css" />
  <link rel="stylesheet" type="text/css" href="css/set1.css" />
  <link href="css/overwrite.css" rel="stylesheet">
  <link href="css/style.css" rel="stylesheet"> // marcar formatos
  <link rel="shortcut icon" href="img/Tesis.png" type="image/x-icon"> // icono
</head>
```

#### Barra de navegación

En esta sección se crea una barra de navegación donde se tiene acceso a las diferentes pantallas de navegación o pestañas como Inicio (home), Entrenamiento, Dashboard y Manuales. Dicho código se presenta a continuación:

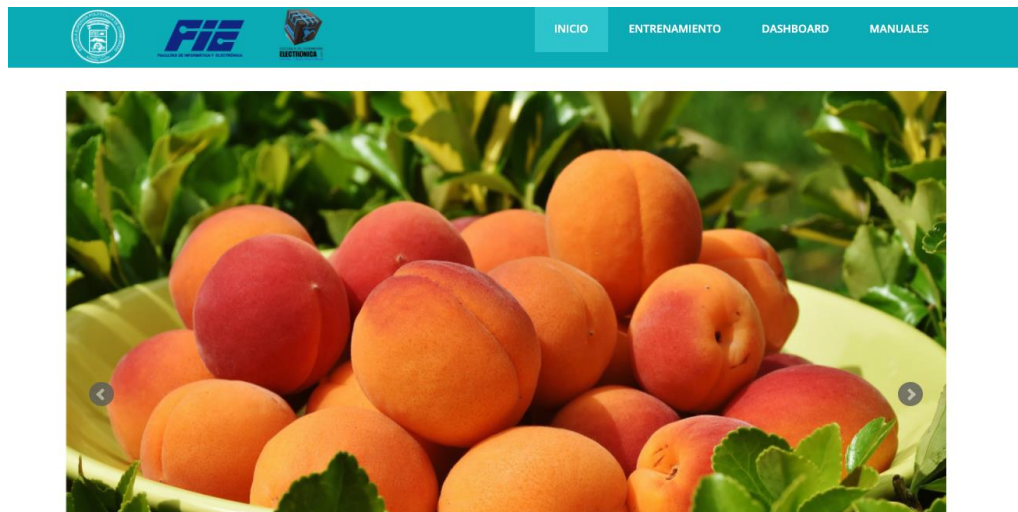
```
<div class="navbar-header">
  <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
target=".navbar-collapse.collapse">
  <span class="sr-only">Toggle navigation</span>
  <span class="icon-bar"></span>
  <span class="icon-bar"></span>
  <span class="icon-bar"></span>
</button>
```

```

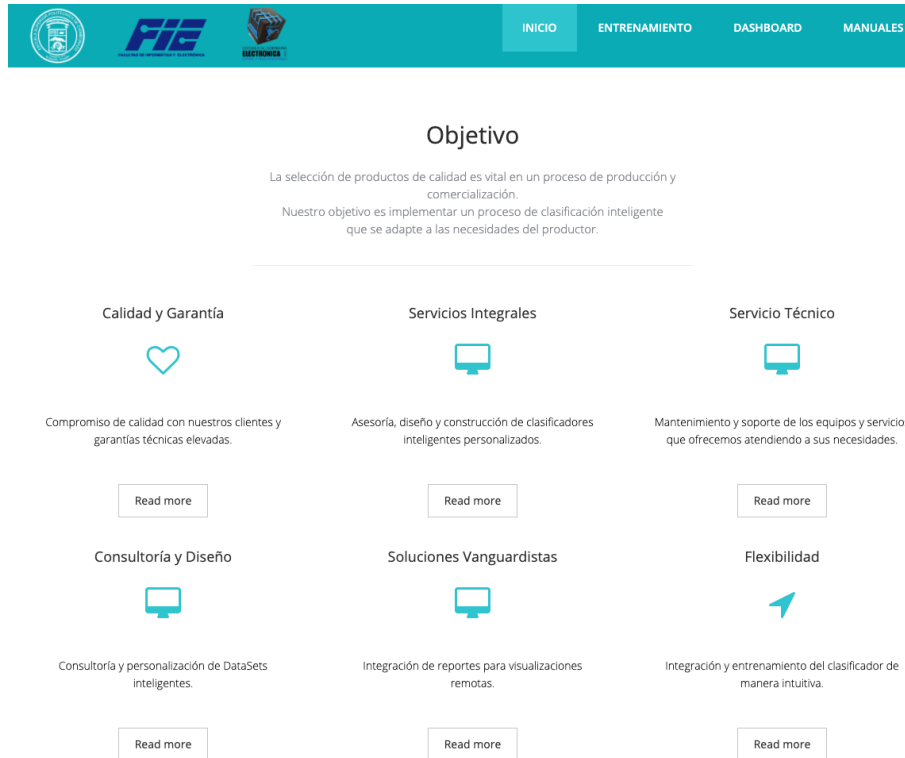
<div class="logo">
  
  
  
</div>
</div>
<ul class="nav nav-tabs" role="tablist">
<li role="presentation" class="active"><a href="index.html">Inicio</a></li>
<li role="presentation"><a href="neurona.html">Entrenamiento</a></li>
<li role="presentation"><a href="dash.html">DashBoard</a></li>
<li role="presentation"><a href="manuales.html">Manuales</a></li>
</ul>

```

Las dos secciones anteriores marcan una plantilla fundamental en la página web ya que en las 3 pestañas se replica los recursos antes mencionados, pero con diferentes contenidos. En la figura 20-2 y 21-2 se presenta la página e información inicial.



**Figura 20-2:** Home página web  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020



**Figura 21-2:** Home página web parte introductoria.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

### 2.3.1.2. Selección del Algoritmo Machine Learning KNN de la Librería ML5.JS

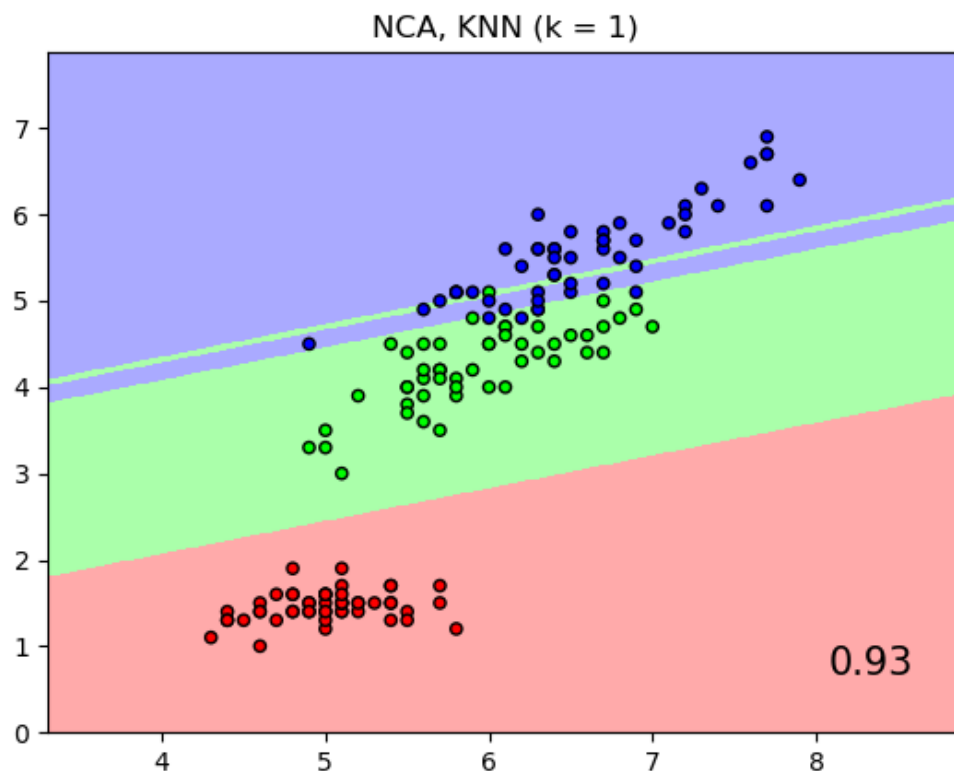
En el primer capítulo se trató acerca de la introducción a la inteligencia artificial, agentes inteligentes, desarrollo del Softcomputing, lógica difusa, algoritmos genéticos y redes neuronales artificiales. Para el proyecto de clasificación de productos, primeramente, se debe tener una manera adecuada de adquirir los datos o censar el producto al cual se lo va a clasificar. Para ello se utiliza una cámara que permite al algoritmo visualizar e identificar rasgos característicos para guardar en su universo de discurso o Data Sets. Para que el sistema sea más robusto, el proceso de entrenamiento es directamente proporcional al éxito y números de aciertos.

Para el proyecto se necesita un método para clasificación de imágenes, para eso se implementa el algoritmo KNN (k vecinos más cercanos). Este es un fuerte algoritmo para el aprendizaje automático supervisado (machine learning). La base principal de su entrenamiento son los datos de entrada etiquetados previamente según requiera el usuario para aprender y el mismo produce una salida apropiada de datos después del entrenamiento cuando se le proporciona datos sin etiquetar.

Si se quiere transmitir a un sistema el concepto de naturaleza se podría emplear técnicas gráficas que imiten la metodología de aprendizaje similar a la de un niño que empieza a reconocer nuevos objetos a través de gráficos que representen entornos naturales.

Similar a este proceso es cómo funciona el machine learning supervisado con el algoritmo KNN, mismo que fue implementado en el sistema de clasificación para que la metodología que use un operador sea similar a la analogía citada anteriormente.

El algoritmo KNN interpreta puntos similares cercanos uno de otro, esto ayuda a una clasificación por grupos según su similitud como se identifica en la figura 22-2.

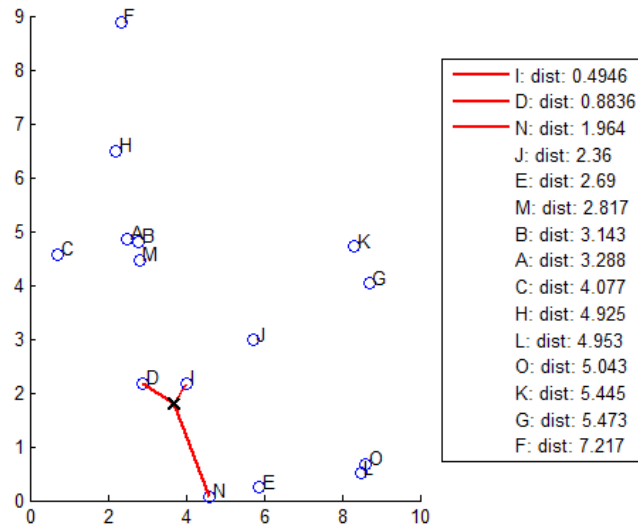


**Figura 22-2:** Imagen muestra cómo los puntos de datos similares suelen existir cerca uno del otro.

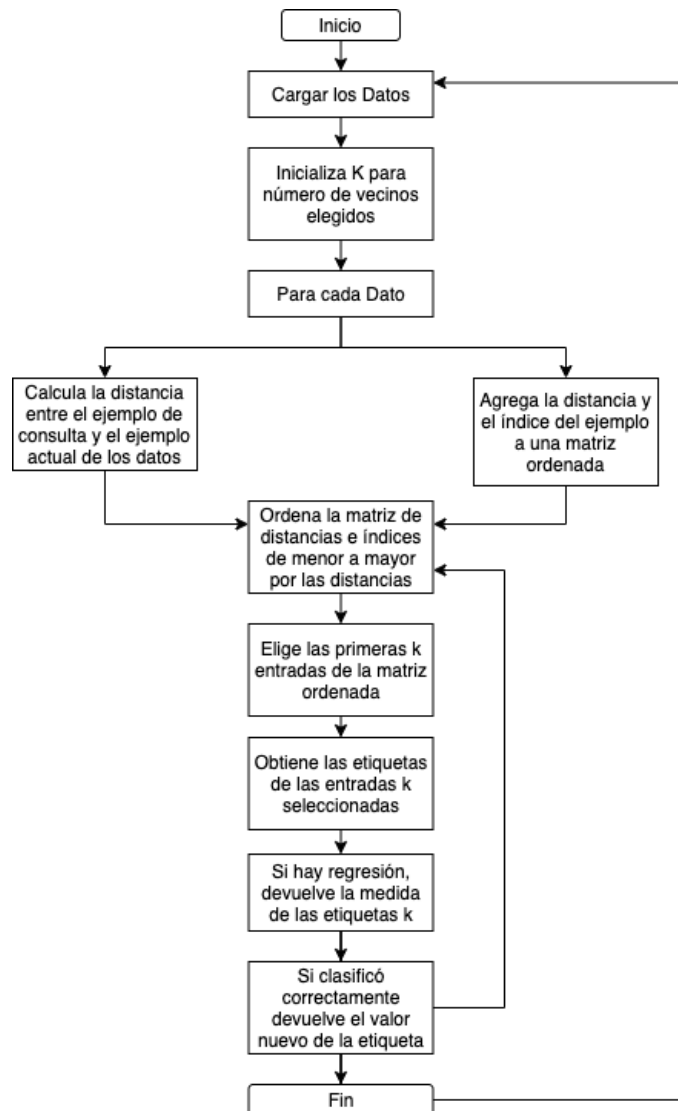
Fuente: <https://scikit-learn.org/stable/modules/neighbors.html>

En la figura 22-2 se observa que, a mayor escala, los datos (puntos) similares están separados por grupos según sus características. El algoritmo KNN captura una característica de cada dato para comparar la cercanía que tiene con sus vecinos, que es la distancia entre dos puntos, tal y como se muestra en la figura 23-2.



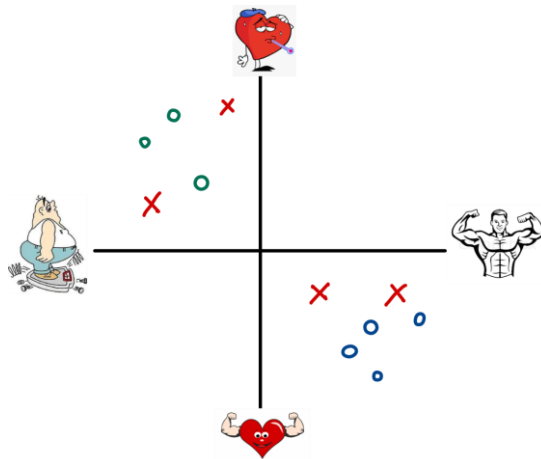


**Figura 23-2:** Cálculo de las distancias desde x hasta todos sus vecinos.  
**Fuente:** («Clasificación utilizando vecinos más cercanos - MATLAB & Simulink - MathWorks América Latina», 2018)



**Gráfico 3-2:** Diagrama de flujo funcionamiento del algoritmo knn.  
**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

Otro caso para ejemplificar la aplicación del algoritmo KNN es el que se presenta en la figura 24-2. En este ejemplo se hace una analogía en la cual, el eje X representa a una persona saludable y una no tan sana, y en el eje Y la probabilidad en contraer una enfermedad cardiovascular. Los círculos azules representan nuevos datos etiquetados que indican llevar una vida saludable, y los círculos verdes a personas que se acercan a tener complicaciones. Ahora las X de color rojo indican nuevas personas o datos no etiquetados que según sus características podemos deducir su estado de salud.



**Figura 24-2:** Analogía de la lógica KNN  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

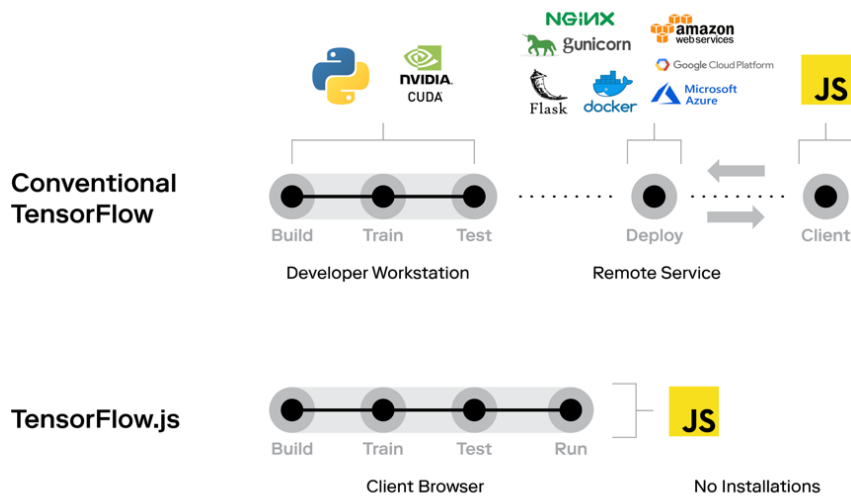
El M15.js es una librería de JavaScript que implementa algoritmos de aprendizaje automático de máquinas. La ideología de dicha librería es la simplicidad, por tal motivo, en el proyecto clasificador inteligente se lo implementa para que la etapa de entrenamiento lo pueda ejercer cualquier persona sin importar el nivel de conocimiento experto ya que está pensado en el usuario final.



**Figura 25-2:** M15 por Tensorflow.  
Fuente: [www.tensorflow.com](http://www.tensorflow.com)

La flexibilidad y la ventaja de ml5.js es que nos permite programar un modelo en JavaScript como lo hace Keras en Python («Keras | TensorFlow Core» 2019). La diferencia radica en que Keras necesita un servicio adicional para trasladar el modelo a plataformas Web como se observa en la figura

26-2, mientras que ml5.js está desarrollado para cubrir esa necesidad sin utilizar más recursos o servicios de terceros.

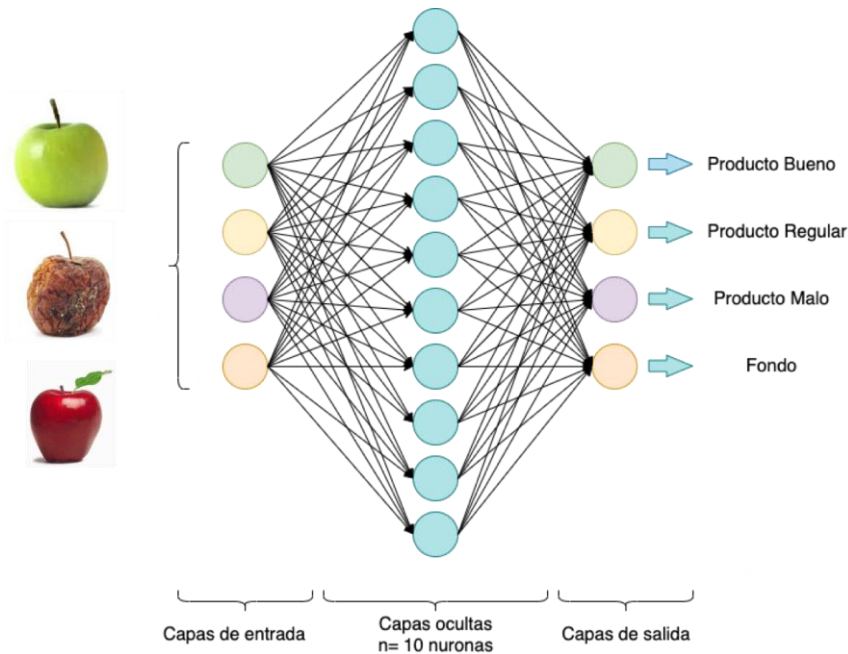


**Figura 26-2:** Metodología para implementar Tensorflow web.

Fuente: <https://www.analyticsvidhya.com/blog/2019/06/build-machine-learning-model-in-your-browser-tensorflow-js-deeplearn-js/>

### 2.3.1.3. Implementación de un clasificador de imágenes con Redes Neuronales Convolucionales (CNN)

Entendido el concepto de redes neuronales explicadas en el capítulo anterior y el algoritmo KNN de Tensorflow, a continuación, se explica la metodología que se utiliza en el trasfondo cualquier algoritmo de clasificación de imágenes utilizando técnicas de Softcomputing. En la imagen 27-2 se ilustra el contexto general de las operaciones que realiza una red neuronal con las imágenes.



**Figura 27-2:** Representación general de la red neuronal clasificador de imágenes.  
 Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

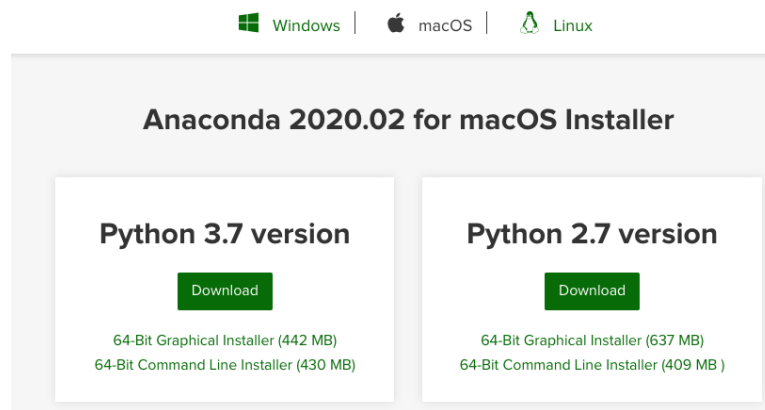
Las redes neuronales convolucionales son muy similares a las redes neuronales comunes del capítulo anterior: están formadas por neuronas que tienen pesos y sesgos aprendibles. Cada neurona recibe algunas entradas, realiza un producto de puntos y, opcionalmente, lo sigue con una no linealidad. Toda la red aún expresa una única función de puntuación diferenciable: desde los píxeles de una imagen sin procesar y una función de pérdida en la última capa.

Lo que se diferencia de una red neuronal tradicional es que las arquitecturas convolucionales tienen como entradas estrictamente imágenes. Esto hace que la función de reenvío sea más eficiente de implementar y reduce enormemente la cantidad de parámetros en la red.

Una CNN funciona mediante la extracción de características de las imágenes. Esto elimina la necesidad de extracción manual de las funciones de activación. La CNN aprende mientras la red entrena con un conjunto de imágenes en este caso un conjunto de imágenes de frutas en diferente estado. Esto hace que los modelos de aprendizaje profundo sean más precisos para las tareas de visión por computadora. Para esto se utilizó un algoritmo de Tensor Flow y las librerías de Keras como se detalla a continuación:

**Paso N.º 1**

Como primer paso se debe configurar el computador con el ambiente de desarrollo Python, para esto se debe acceder al sitio web [www.anaconda.org](http://www.anaconda.org) en el cual encontramos la siguiente pantalla figura 28-2.



**Figura 28-2:** Anaconda Navigator.  
**Fuente:** <https://www.anaconda.com/distribution/>

### Paso N.º 2

Para utilizar el estándar actual en la codificación de Python se debe seleccionar la versión 3.7 ya que la 2.7.X se encuentra descontinuada. Para ejecutar algoritmos de machine learning es necesario instalar los siguientes paquetes: Tensorflow, Keras y Numpy. Este proceso se lo realiza desde el terminal. Un ejemplo de esta tarea se muestra en la figura 29-2.

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) Carloss-MacBook-Pro:~ carlossguillazaca$ pip install tensorflow
```

**Figura 29-2:** Instalación de paquetes desde el terminal.  
**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

### Paso N.º 3

Una vez instalado todo lo necesario, se abre el entorno de desarrollo Spyder que está integrado en la plataforma de Anaconda-Navigator y se inicia declarando lo siguiente:

```
1 import sys
2 import os
3 from tensorflow.python.keras.preprocessing.image import ImageDataGenerator
4 from tensorflow.python.keras import optimizers
5 from tensorflow.python.keras.models import Sequential
6 from tensorflow.python.keras.layers import Dropout, Flatten, Dense, Activation
7 from tensorflow.python.keras.layers import Convolution2D, MaxPooling2D
8 from tensorflow.python.keras import backend as K
9
```

**Figura 30-2:** Inicialización del algoritmo para el entrenamiento.  
**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

El código se lo realiza con Tensorflow y dentro de este se tiene la posibilidad de llamar a una API de Keras.

- En la primera línea sys y os son librerías que nos permite movernos dentro de las carpetas del sistema operativo.
- En la tercera línea el ImportDataGenerator se encarga de pre procesar las imágenes que entran al algoritmo.
- En la cuarta línea el Optimizers es un optimizador adjunto del algoritmo.

- En la quinta línea el Sequential es una librería que permite crear redes neuronales secuenciales, es decir que cada una de las capas están en orden.
- El Convolution2D y MaxPooling2D son las capas donde se harán las convoluciones.
- En la octava línea importamos el backeSignum.nd de keras, este ayuda a que cuando haya una sesión de keras que está corriendo en segundo plano la elimina y optimiza recursos físicos del computador para empezar el entrenamiento fresco del nuevo algoritmo.
- En la figura 31-2 se declaran las variables que contienen los directorios de las carpetas de los archivos de las imágenes data\_entrenamiento (imágenes para entrenar) y data\_validación (imágenes para comprobar el funcionamiento).

```
data_entrenamiento = './data/entrenamiento'
data_validacion = './data/validacion'
```

**Figura 31-2:** Variables selección del Data Set local.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

#### Paso N.º 4

En el paso número 4 se declara los parámetros que se expresan en la figura 32-2 y se explican a continuación:

```
20 epocas=20
21 longitud, altura = 150, 150
22 batch_size = 32
23 pasos = 1000
24 validation_steps = 300
25 filtrosConv1 = 32
26 filtrosConv2 = 64
27 tamaño_filtro1 = (3, 3)
28 tamaño_filtro2 = (2, 2)
29 tamaño_pool = (2, 2)
30 clases = 3
31 lr = 0.0004
```

**Figura 32-2:** Declaración de los Parámetros

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

- Épocas, de la línea 20 variable que indica el número de veces que se va a iterar sobre todo el Data Set del entrenamiento.
- La altura y longitud son variables que van a delimitar el tamaño uniforme de las imágenes.
- El batch\_size es el número de imágenes que se va a enviar al computador a procesar en cada uno de los pasos.
- Pasos, variable que indica el número de veces que se va a procesar la información en cada una de las épocas.
- Validation\_step, son los pasos de validación que al final de cada una de las épocas se van a ejecutar 200 pasos con el Data Set de validación antes mencionado.

- FiltrosConv1 y FiltrosConv2, son filtros que se aplica en cada convolución en la primera convolución agrega una profundidad de 32 y en la segunda convolución una profundidad de 64.
- Tamano\_filtro1 y tamano\_filtro2, son tamaños de los filtros que se agregan a cada convolución (altura, longitud).
- Tamano\_pool, este es el tamaño del filtro que se va a utilizar en Maxpooling2D.
- Clases, son las salidas de la red neuronal o el conjunto de especies que se va a clasificar.
- Lr, este es el learning rate, indican el tamaño de los ajustes de la red neuronal para acercarse a una solución lógica.

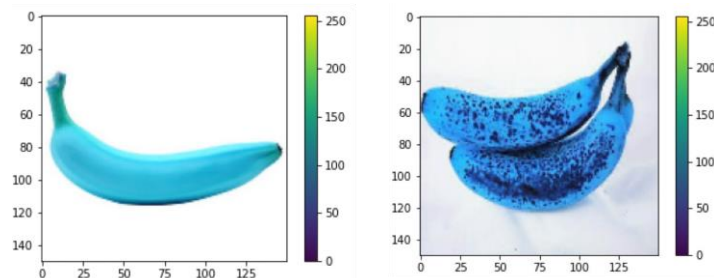
### Paso N.º 5

En este paso se realiza el preprocesamiento de las imágenes de entrada y unificar las características como se observa en las figuras 34-2 con el siguiente código de la figura 33-2.

```
import cv2
import numpy as np
import os
import matplotlib.pyplot as plt

producto1_folder_path="/content/BananaMadura"
Producto1=[]
img_size=150
for img in os.listdir(producto1_folder_path):
    img = cv2.imread(os.path.join(producto1_folder_path,img))
    #img_gray= cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    img_resize= cv2.resize(img,(img_size,img_size))
    Producto1.append(img_resize)
```

**Figura 33-2:** Parámetros cv2.resize y cv2.cvtColor  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020



**Figura 34-2:** Parámetros cv2.resize y cv2.cvtColor  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

Para generar la concatenación de las imágenes y enlazar las matrices que generan las imágenes con las clases creadas que se utilizó se muestra en la figura 35-2:

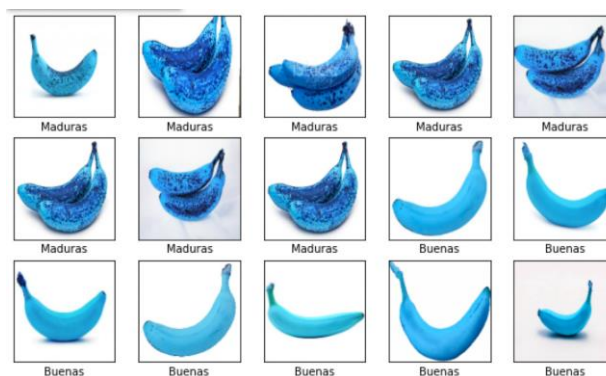
```
plt.figure(figsize=(10,10))
for i in range(20):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(Images[i])
    #, cmap=plt.cm.binary
    plt.xlabel(class_names[Labels[i]])
plt.show()
```

**Figura 35-2:** Algoritmo que genera la gráfica 35-2  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

- **Class\_names** de la figura 35-2, representa a **class\_names=['Maduras', 'Buenas']**.
- **Labels**, representa a la concatenación de las etiquetas generadas de cada producto con el código: **Labels =np.concatenate([etiqueta\_maduras , etiqueta\_buenas])**.
- Las etiquetas se decidieron de la siguiente manera: 1 para productos buenos y 0 para productos maduros. En la figura 36-2 se observa la etiqueta para el producto bueno y el 9 representa la cantidad de imágenes en la carpeta de productos buenos.

```
etiquetas_Producto1 = np.repeat(1,9)
print(len(etiquetas_Producto1))
print(etiquetas_Producto1)
```

**Figura 36-2:** Algoritmo que genera las etiquetas  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020



**Figura 37-2:** Representación de las imágenes pre procesadas.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

## Paso N.º 6

En este punto se crea la Red Neuronal Convolutiva CNN de la siguiente manera reflejada en la figura 38-2:

```
56 cnn = Sequential()
57 cnn.add(Convolution2D(filtrosConv1, tamaño_filtro1, padding = "same", input_shape=(longitud, altura, 3), activation='relu'))
58 cnn.add(MaxPooling2D(pool_size=tamaño_pool))
```

**Figura 38-2:** Algoritmo ceración de CNN.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

- Cnn, es la variable que genera una red secuencial, es decir, que son varias capas apiladas entre ellas.
- En la línea de código 57 de la figura 38-2, describe que la primera capa hará una convolución con los números de filtros mencionados en el paso 4, el padding configura las esquinas, el input\_shape define las variables del paso 4 y por último se tiene una función de activación mencionada en el primer capítulo que se habló de las RELU.



- La línea 58 se agrega una capa de pooling. Tomar en cuenta que se puede agregar más capas de cnn repitiendo el código de la figura 38-2 para una mejor predicción.

Después de la estructura anterior en la figura 39-2 se detalla el algoritmo para la clasificación:

```

62
63     cnn.add(Flatten())
64     cnn.add(Dense(256, activation='relu'))
65     cnn.add(Dropout(0.5))
66     cnn.add(Dense(clases, activation='softmax'))
67
68     cnn.compile(loss='categorical_crossentropy',
69               optimizer=optimizers.Adam(lr=lr),
70               metrics=['accuracy'])
71

```

**Figura 39-2:** Algoritmo ceración de CNN.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

- En la línea 63, el Flatten ayuda a que las imágenes profundas se conviertan a planas, es decir, que solo se va a tener una sola dimensión.
- La línea 64, se declara Dense con 256 neuronas que reciben la información de la línea anterior.
- El Dropout se declaró el 0.5, esto quiere decir que durante el entrenamiento se apaga el 50% de las neuronas cada paso, con esto se evita que el algoritmo aprenda un solo camino para una clasificación de un producto en específico y genere alternativas cuando se enfrente a nueva información.
- La última capa se declara en la línea 66 con una clase de 3 neuronas (variable declarada en el paso 4), softmax determina el porcentaje de cada una de las salidas y el número mayor es el que predomina en la clasificación.
- En la línea 68, explica que durante el entrenamiento la función de perdida determina el estado porcentual del éxito en dicha operación.

## Paso N.º 7

En este paso ya se declara el algoritmo que ejecuta ya el entrenamiento de la red neuronal convolucional antes creada como se observa en la figura 40-2:

```

75     cnn.fit_generator(
76         entrenamiento_generador,
77         steps_per_epoch=pasos,
78         epochs=epocas,
79         validation_data=validacion_generador,
80         validation_steps=validation_steps)
81
82     target_dir = './modelo/'
83     if not os.path.exists(target_dir):
84         os.mkdir(target_dir)
85     cnn.save('./modelo/modelo.h5')
86     cnn.save_weights('./modelo/pesos.h5')
87

```

**Figura 40-2:** Ejecutar el entrenamiento.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

- El `cnn.fit` quiere decir que se va a entrenar la red neuronal con las imágenes pre procesadas, se va a ejecutar cada 1000 pasos, cada época o el número de iteraciones es igual a 20.
- La línea 82 es muy importante, ya que se describe como se genera un modelo ya entrenado que puede ser usado posteriormente y evitar entrenar cada vez que se desea utilizar la clasificación.

### 2.3.1.4. Generar y Cargar el modelo en JavaScript

En este apartado se crea el interfaz web de la pestaña del entrenamiento y supervisión en tiempo real del proceso. Este tiene como objetivo mostrar datos como el estado del modelo, corrección y entrenamiento de productos para enriquecer el data set, cargar un nuevo parámetro y cargar o descargar modelos previamente entrenados como se observa en la figura 41-2.



**Figura 41-2:** Entrenamiento interfaz web.  
**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

Para implementar lo anterior se siguió la siguiente metodología:

#### Paso N.º 1

En el mismo archivo que se creó para realizar la pantalla de inicio, se creó un nuevo HTML que se llama `neurona.html`, dentro de este se declara el `sketch.js` (código de la parte lógica del entrenamiento) e instala la librería de `p5.js` con las siguientes librerías:

```

<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.7.3/p5.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.7.3/addons/p5.dom.min.js"></script>
<script src="https://unpkg.com/ml5@0.2.1/dist/ml5.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.js"
  type="text/javascript"></script>
<script src="sketch.js"></script>

```

**Figura 42-2:** Librerías dentro de neurona.html.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

## Paso N.º 2

En este paso se crea el archivo sketch.js que contiene la parte lógica del entrenamiento y supervisión, empezando con la declaración de las variables iniciales de los parámetros como se observa en la figura 43-2:

```

1  var Camara;
2  var BotonesEntrenar;
3  var knn;
4  var modelo;
5  var Texto;
6  var Clasificando = false;
7  var InputTextbox;
8  var BotonTextBox;

```

**Figura 43-2:** Librerías dentro de neurona.html.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

- En la línea 1 se crea un objeto Cámara con las configuraciones que se ilustra en la figura 44-2. Antes de llamar a la sentencia createCapture(VIDEO), se ha creado un canvas o lienzo que es la base donde contendrá la ventana del video con un tamaño (320, 240) px. Las variables “x” y “y” posicionan la ubicación del lienzo dentro del interfaz general.

```

41
42  function setup() {
43
44    var canvas=createCanvas(320, 240);
45    var x=(windowWidth - width)/2;
46    var y=(windowHeight - height)/2;
47    canvas.position(x,y);
48    background(0, 0, 0);
49    Camara = createCapture(VIDEO);
50    Camara.size(320, 240);
51    Camara.hide();

```

**Figura 44-2:** Configuración inserción del video en la ventana web.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

- Var BotonesEntrenar de la línea 2 del código, se crea una clase para el estándar de los botones que se utilizan en el interfaz como se observa en la figura 45-2 y el estándar es ilustra en la figura 46-2 con la utilización de selectAll ().

```

var Boton1 = createButton("Producto Bueno");
Boton1.class("BotonEntrenar");
Boton1.position(150,350);

var Boton2 = createButton("Producto Regular");
Boton2.class("BotonEntrenar");
Boton2.position(150,400);

var Boton3 = createButton("Producto Malo");
Boton3.class("BotonEntrenar");
Boton3.position(150,450);

var BotonNada = createButton("Fondo");
BotonNada.class("BotonEntrenar");
BotonNada.position(180,500);

```

**Figura 45-2:** Botones en la ventana.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

```

BotonesEntrenar = selectAll(".BotonEntrenar");

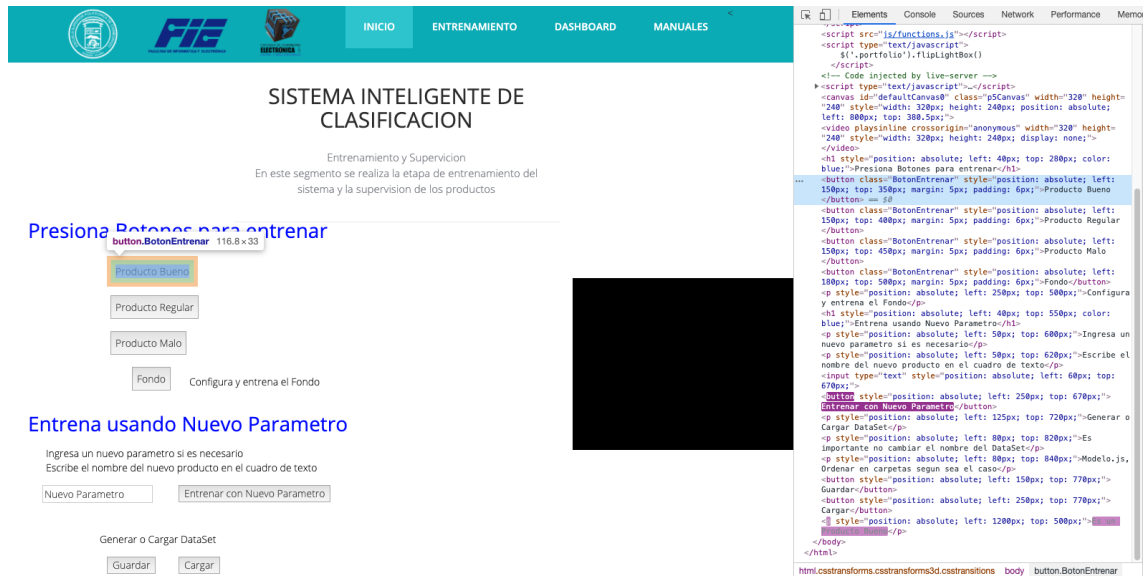
for (var B = 0; B < BotonesEntrenar.length; B++) {
  BotonesEntrenar[B].style("margin", "5px");
  BotonesEntrenar[B].style("padding", "6px");
  BotonesEntrenar[B].mousePressed(PresionandoBoton);
}

```

**Figura 46-2:** Estándar de los botones utilizando selectAll.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

- SelectAll busca dentro de todo un HTML los recursos que contengan la misma clase declarada en la sintaxis de la figura 46-2 y se lo comprobó ingresando a la página web. Utilizando el depurador del navegador como se muestra en la figura 47-2 se selecciona todos los botones para aplicar un solo parámetro de configuración. Este paso es importante ya que posteriormente se crea una función por botón que manipulará los datos que se van al bróker MQTT.



**Figura 47-2:** Depurador del navegador.  
**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

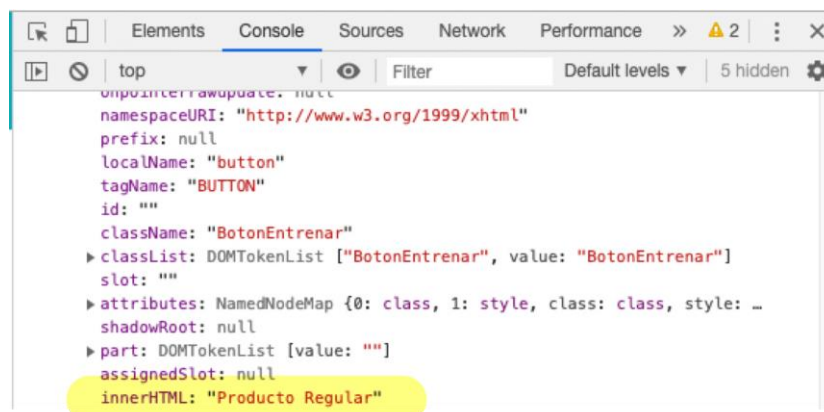
- Para obtener el nombre del botón figura 48-2 necesario para adquirir los datos que se van a clasificar y a enviar al mqtt, nuevamente se ha usado el depurador figura 49-2.

```

122
123 function PresionandoBoton() {
124     var NombreBoton = this.elt.innerHTML;
125     console.log("Entrenando con " + NombreBoton);
126     EntrenarKnn(NombreBoton);
127 }

```

**Figura 48-2:** Capturar el nombre del botón dentro de HTML.  
**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020



**Figura 49-2:** Capturar nombre del botón con el depurador del navegador.  
**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

### Paso N.º 3

Para modelar ml5.js dentro de JavaScript se ha cargado la librería ya mencionada en figura 41-2 dentro de la neurona.html, luego, dentro del archivo sketch.js en la figura 40-2 se ha creado una variable knn que permite extraer el modelo de MobileNet detallado en la figura 50-2.

```

53  modelo = ml5.featureExtractor('MobileNet', ModeloListo);
54  knn = ml5.KNNClassifier();

```

**Figura 50-2:** Extraer el modelo MobileNet.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

Dentro de la sintaxis de la línea 53 del código se ha declarado la función ModeloListo que lo único que hace es enviar texto con la descripción “Modelo Listo”.

```

129 function EntrenarKnn(ObjetoEntrenar) {
130   const Imagen = modelo.infer(Camara);
131   knn.addExample(Imagen, ObjetoEntrenar);
132 }

```

**Figura 51-2:** Extraer el modelo MobileNet.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

La función EntrenarKnn de la figura 51-2, crea una matriz de números a partir de una imagen que representan al modelo que ya se trató en el punto del entrenamiento con redes neuronales convolucionales.

## 2.4. MQTT

Para configurar el servicio del protocolo de comunicación mqtt, se ha utilizado el bróker shiftr.io ilustrado en la figura 52-2.



**Figura 52-2:** Broker shiftr.io

Fuente: <https://shiftr.io/>

Es importante declarar las siguientes variables en el código sketch.js:

```

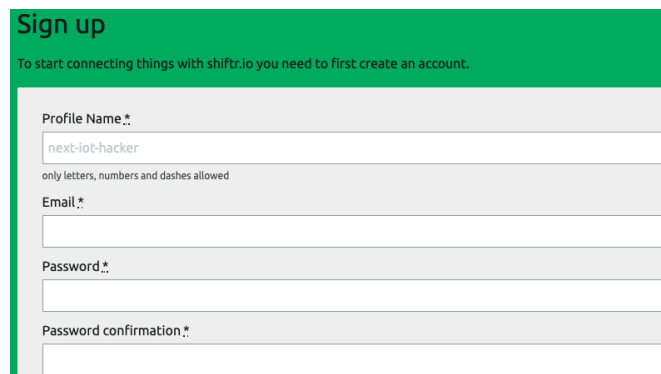
10 let BrokerMQTT = 'broker.shiftr.io';
11 let PuertoMQTT = 80;
12 let ClienteIDMQTT = "MQTT-P5";
13 let UsuarioMQTT = "carlosguillazaca";
14 let ContraseñaMQTT = "nomeacuerdo2402";

```

**Figura 53-2:** Parámetros de validación mqtt.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

Los parámetros de la figura 53-2 se crean en la pantalla de registro de la figura 54-2 proporcionada desde shiftr.io.



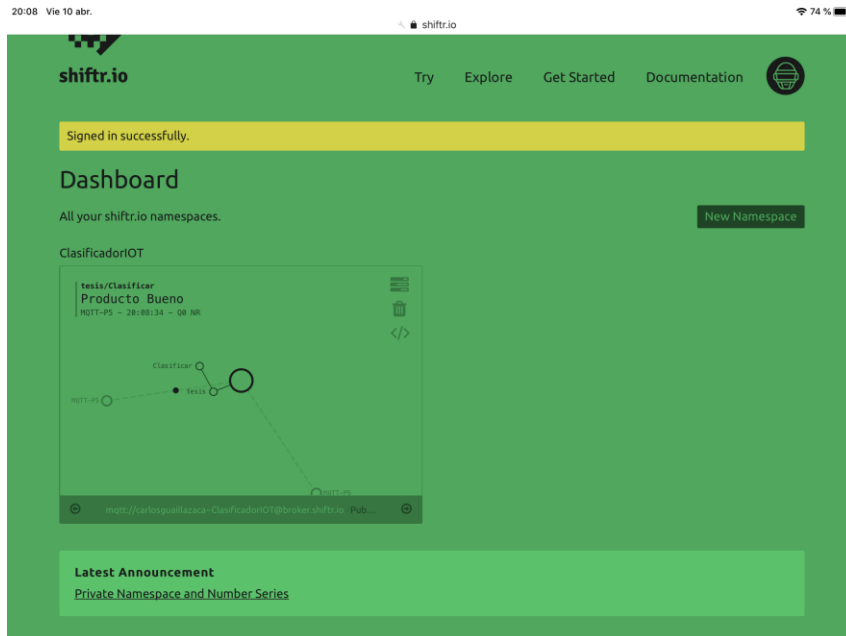
**Figura 54-2:** Login del broker.  
**Fuente:** <https://shiftr.io/>

Para establecer la conexión por defecto el bróker sugiere agregar el código descrito en la figura 55-2.

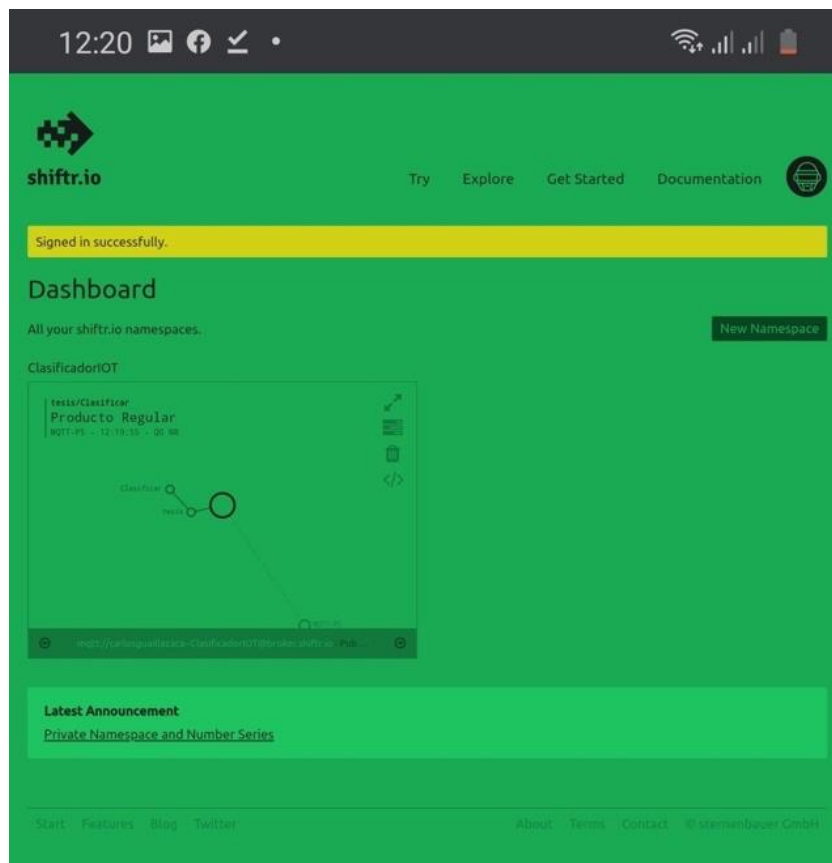
```
16 client = new Paho.MQTT.Client(BrokerMQTT, PuertoMQTT, ClienteIDMQTT);
17
18 // Mensajes de callback
19 client.onConnectionLost = MQTTPerder;
20 client.onMessageArrived = MQTTMensaje;
21
22 client.connect({
23     onSuccess: CuandoConectadoMQTT,
24     userName: UsuarioMQTT,
25     password: ContraseñaMQTT
26 });
```

**Figura 55-2:** Configuración inicial para establecer la conexión.  
**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

Cuando ya se ha establecido la comunicación en el Dashboard de Shiftr.io, se crea una ventana con el proyecto puesto en marcha que se ilustra en la figura 56-2. Desde allí se supervisa en tiempo real el estado del flujo de los datos desde la aplicación web hasta el bróker que posteriormente se puede agregar varios clientes que accedan a esta información y poder utilizarla, como por ejemplo para supervisarla desde un teléfono como se muestra en la figura 57-2.



**Figura 56-2:** Dashboard del enlace app web-mqtt-shiftr desde Ipad OS.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020.



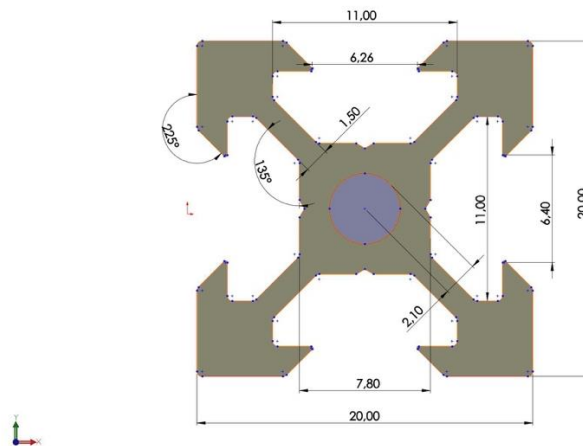
**Figura 57-2:** Dashboard mqtt desde el navegador teléfono Android.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020.



## 2.5. CAD Estructural.

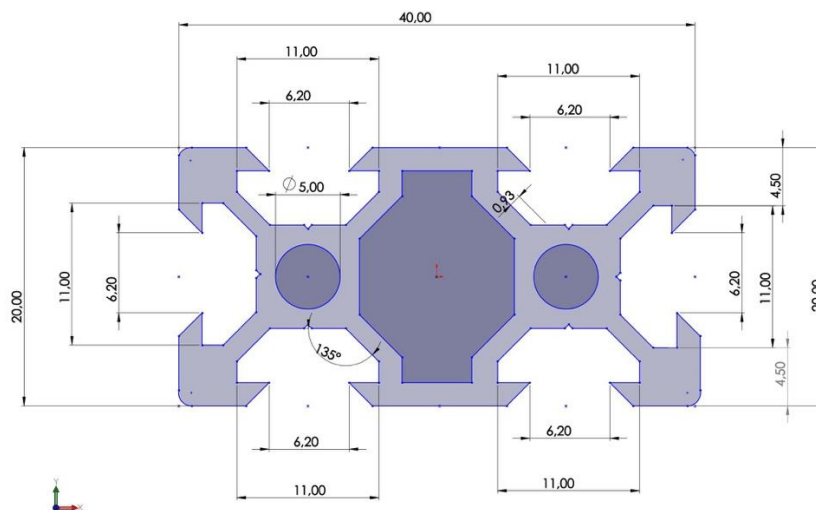
El diseño CAD estructural del clasificador, se lo realizó utilizando el software CAD SolidWorks y se consideró aspectos como: longitud óptima y útil para realizar la tarea de clasificación a nivel de prototipo, material a utilizar, ensamblaje sencillo y accesible.

El alma estructural de la base del clasificador que se seleccionó el perfil V-Slot 20x40mm y 20x20mm. V-Slot es un perfil de aluminio que facilita el ensamblaje de maquinaria industrial de precisión tales como CNC o Impresoras 3D. Se escogió este material porque posee muchas ventajas, pese a ser aluminio un material no muy rígido la forma de la fabricación de los perfiles es muy resistente y liviano a la vez lo que significa que es más barato en comparación de un cuerpo sólido. La figura58-2 representa el croquis del V-Slot 20mmx20mm el cual se utilizó 8x100mm, además se usó V-Slot de 20mmx40mm 2x1000mm el croquis se representa en la figura 59-2.



**Figura 58-2:** Croquis V-Slot 20mmx20mm.

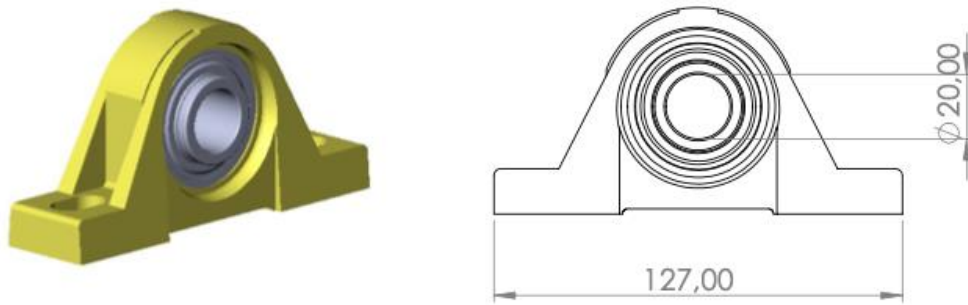
Realizado por: GUALLAZACA GONZÁLEZ, Carlos Andrés, 2020



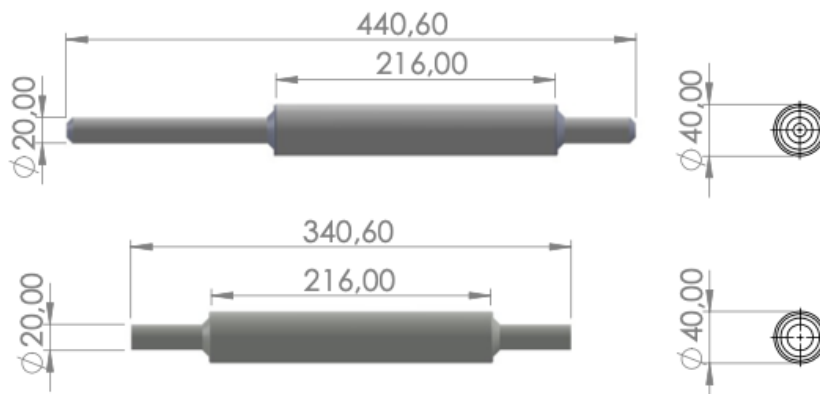
**Figura 59-2:** Croquis V-Slot 20mmx40mm.

Realizado por: GUALLAZACA GONZÁLEZ, Carlos Andrés, 2020

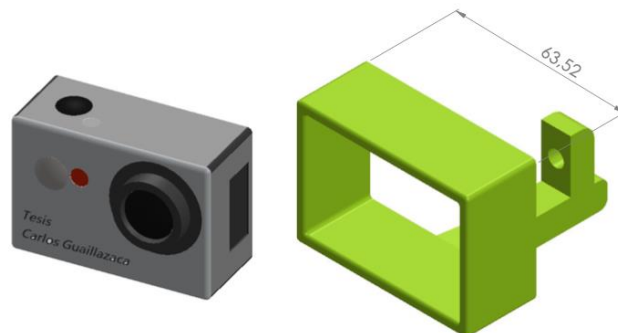
A continuación, se detallan las figuras 60-2, 61-2, 63-2 y 64-2 las cuales pertenecen al ensamble representado en la figura 65-2.



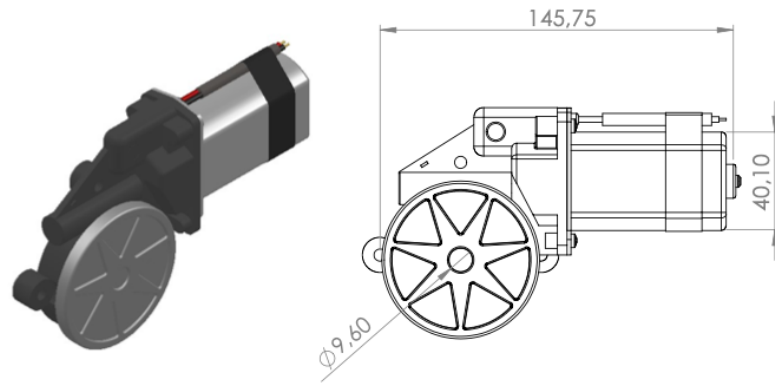
**Figura 60-2:** Chumacera 20mm de diámetro.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020



**Figura 61-2:** Rodillos transmisión dinámica del motor a la banda.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

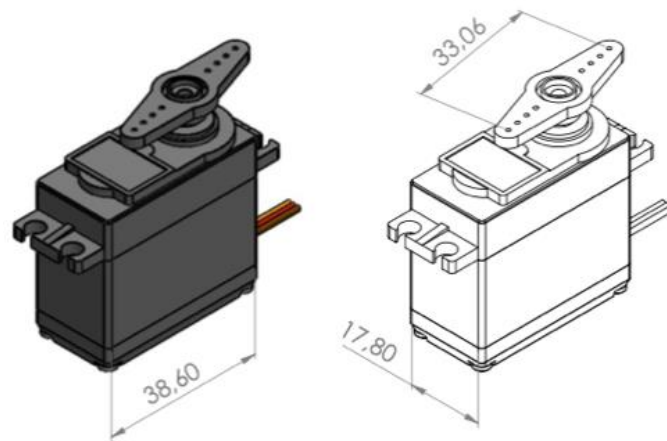


**Figura 62-2:** Cámara y case.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020



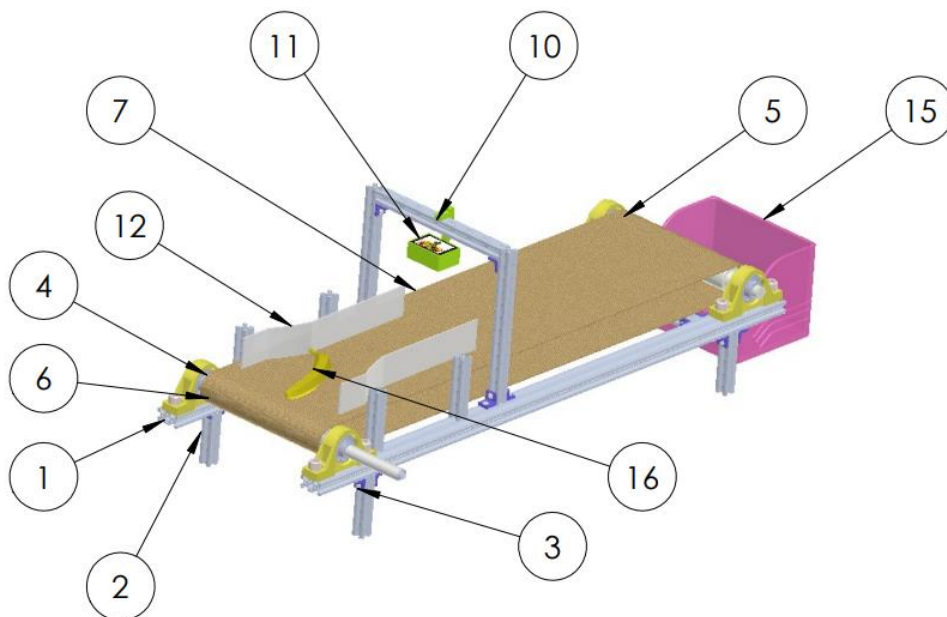
**Figura 63-2:** Motor.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020



**Figura 64-2:** Servo Motor Tower Pro MG-995.

Realizado por: GUAILLAZACA, Carlos.



**Figura 65-2:** Ensamble estructura.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

En la figura 65-2 muestra las piezas que conforman el ensamble del prototipo y en la tabla 7-2 se presenta el detalle de cada una.

**Tabla 7-2:** Lista de materiales para ensamble estructural.

N.º DE ELEMENTO	ELEMENTO	CANTIDAD
1	Vslot 20 x 40 x 1000mm	2
2	Vslot 20 x 20 x 250	8
3	Angulo Simple serie 20	22
4	Chumacera 20mm ¾"	4
5	Eje Teflón diámetro 40mm	1
6	Eje Teflón principal 40mm	1
7	Banda sintética PVC 190mm	1
8	Tornillo 5 x 0.8 x 10	44
9	Tornillo 5 x 0.8 x 5	1
10	Vslot 20 x 20 x 100mm	5
11	Case Cámara	1
12	Guías de acrílico	2
15	Caja producto 1	1
16	Fruta	1

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020.



**Figura 66-2:** Ensamblaje real 1.

Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020.

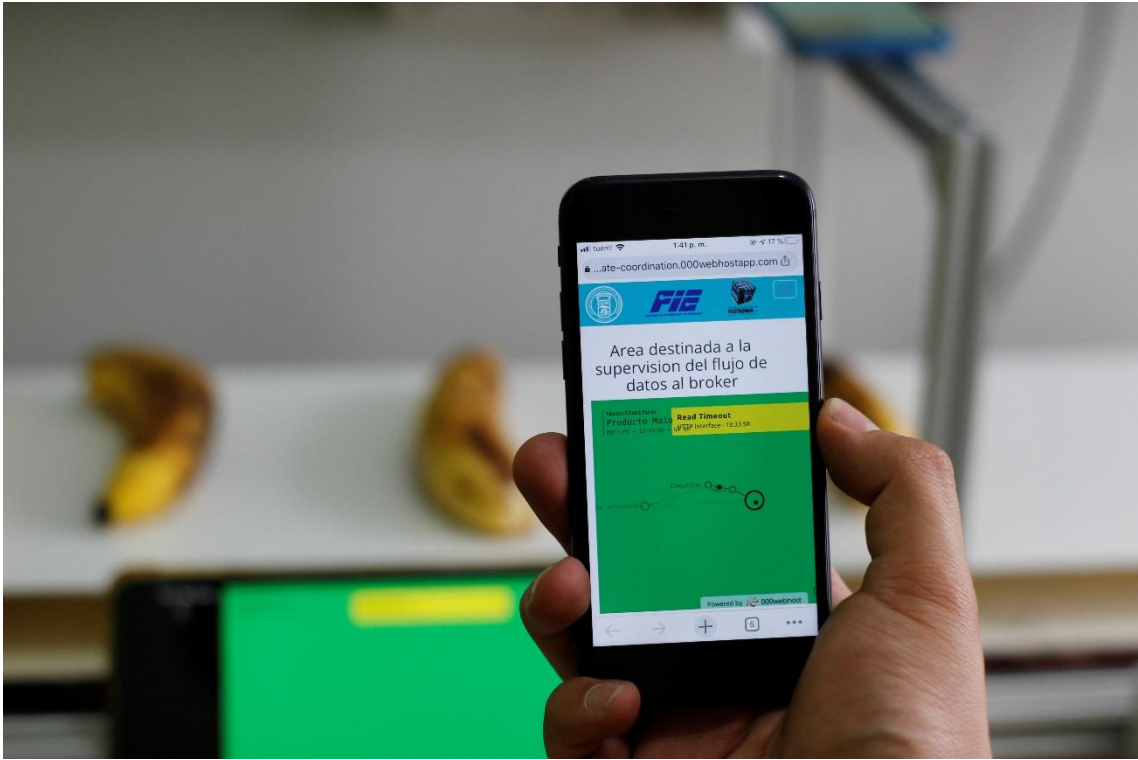


**Figura 67-2:** Visualización del sistema completo, envío de datos a diferentes dispositivos.  
**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020



**Figuras 69-2:** Ensamble completo.  
**Realizado por:** GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020





**Figura 70-2:** Visualización de datos entorno móvil.  
Realizado por: GUAILLAZACA GONZÁLEZ, Carlos Andrés, 2020

## CAPÍTULO III

### 3. VALIDACIÓN DE PROTOTIPO

En este capítulo se presentan las pruebas realizadas y los resultados obtenidos de todo el sistema en conjunto con la finalidad de comprobar el correcto funcionamiento y el cumplimiento de los objetivos.

Para la evaluación del prototipo se utilizaron los criterios de validación analítica, validación experta, validación observacional y validación experimental explicada por (Cobos, 2017).

#### 3.1. Implementación del sistema prototipo

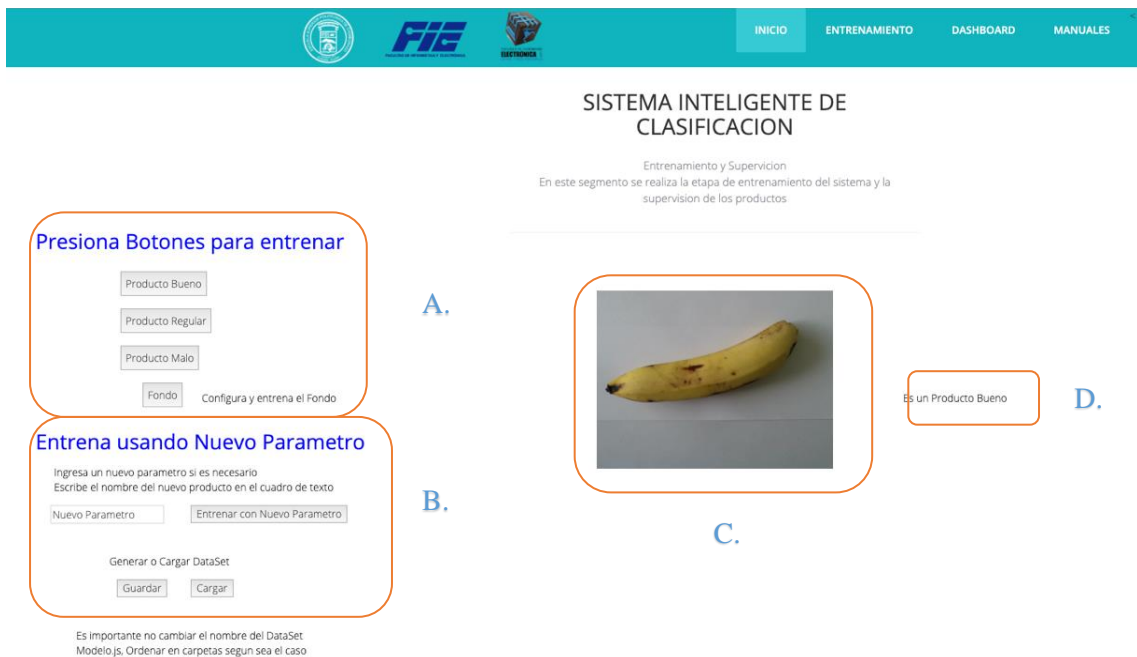
Para ejecutar la clasificación del producto fue necesario previamente dedicar tiempo para el entrenamiento del sistema y de esta manera obtener un modelo para cada una de las pruebas realizadas que se detallan en la tabla 1-3. Posteriormente, el modelo obtenido sirve como base para la etapa de validación, esto se lo realizó con un producto diferente al que se utilizó en la etapa de entrenamiento, procedimiento de validación se detalla más adelante.

La tabla 1-3, representa al número de pruebas que se realizaron en la etapa de entrenamiento, con la finalidad de escoger un modelo óptimo para la etapa de validación, este modelo se lo caracteriza de acuerdo con el número de la prueba realizada. Cada prueba consta con un número de imágenes capturadas que sirven de retroalimentación al algoritmo de entrenamiento. En la prueba 1 se capturan 5 imágenes, en la prueba 2 10 imágenes, en la prueba 3 15 imágenes, y así sucesivamente las imágenes son acumulativas según las pruebas que se realicen. El objetivo de realizar las pruebas según las imágenes capturadas es para identificar el nivel de efectividad del modelo y no saturar al algoritmo con entrenamiento innecesario.

**Tabla 1-3:** Tabla número de prueba y el número de imágenes capturadas.

PRUEBA	NUMERO DE IMÁGENES CAPTURADAS
PRUEBA 1	5
PRUEBA 2	10
PRUEBA 3	15
PRUEBA 4	20
PRUEBA 5	25
PRUEBA 6	30
PRUEBA 7	35
PRUEBA 8	40
PRUEBA 9	45
PRUEBA 10	50

**Realizado por:** Guallazaca González, Carlos Andrés, 2020



**Figura 1-3:** Pestaña para en el entrenamiento del sistema.

**Realizado por:** Guallazaca González, Carlos Andrés, 2020

En la figura 1-3, se estableció un panel frontal para la interacción directa con el usuario. Esta actividad permite que el manejo y manipulación de las variables sean de manera simple, flexible e intuitiva. Además, en dicho panel se distinguen los siguientes bloques:

- Bloque A de la figura 1-3; muestra una selección de botones para que el usuario según su necesidad tenga la capacidad de enriquecer la base de datos del sistema.
- Bloque B de la figura 1-3; se establece un cuadro de texto para ingresar un nuevo producto o parámetro que no se encuentre en los botones principales previamente entrenados.



- Bloque C de la figura 1-3; se observa en tiempo real la imagen del producto para supervisar el correcto ingreso de información al sistema.
- Bloque D de la figura 1-3; se muestra un cuadro de texto con la variable de salida según el producto que se visualiza en el bloque C.

### **3.2. Pruebas del Sistema - Etapa de Entrenamiento.**

Para esta etapa se utilizaron 4 productos agrícolas: Bananas, Naranjas, Plátano verde y manzanas. La selección de estos productos tuvo como finalidad probar el rendimiento de KNN con varios escenarios identificando colores y formas. Cada producto se evaluó de acuerdo con tres variables (Producto Bueno, Producto Regular y Producto Malo).

Para la creación de un data set o conjunto de datos que validan las características de cada producto según la imagen obtenida del mismo, fue imprescindible plantear la siguiente pregunta: ¿Cuántas capturas de imágenes son necesarias para minimizar el error? Antes de dar una respuesta a esta interrogante, es importante entender la metodología utilizada para las dos etapas mencionadas.

#### ***3.2.1. Etapa Entrenamiento para el Producto 1 Bananas***

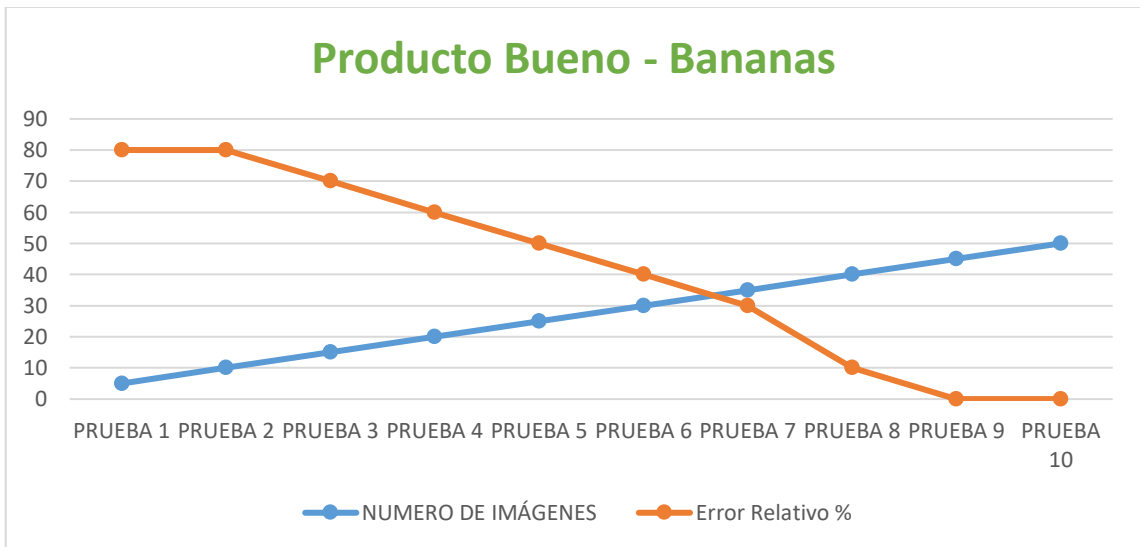
##### ***3.2.1.1. Entrenamiento de la calidad buena para el Producto 1 Bananas***

Para el producto 1 (bananas) en la etapa del entrenamiento se obtuvo los resultados presentados en la tabla 2-3 y gráfico 1-3, siendo esto el resumen del rendimiento de las pruebas realizadas en la etapa de entrenamiento para calidad buena del producto 1.

**Tabla 2-3:** Evaluación del sistema en la etapa de entrenamiento: producto bueno – bananas

# PRUEBA	# IMÁGENES	ACIERTOS	FRACASOS	MUESTRA TOTAL	Error Absoluto	Error Relativo %
PRUEBA 1	5	2	8	10	8	80
PRUEBA 2	10	2	8	10	8	80
PRUEBA 3	15	3	7	10	7	70
PRUEBA 4	20	4	6	10	6	60
PRUEBA 5	25	5	5	10	5	50
PRUEBA 6	30	6	4	10	4	40
PRUEBA 7	35	7	3	10	3	30
PRUEBA 8	40	9	1	10	1	10
PRUEBA 9	45	10	0	10	0	0
PRUEBA 10	50	10	0	10	0	0

Realizado por: Guallazaca González, Carlos Andrés, 2020



**Gráfico 1-3.** Error relativo vs número de imágenes, producto bueno – bananas

Realizado por: Guallazaca González, Carlos Andrés, 2020

El gráfico 1-3 manifiesta la relación proporcional que tiene el resultado según el volumen del conjunto de datos. A partir de la prueba 7 el error absoluto se minimiza a un 30% llegando a un error 0 en la prueba 9 y 10 con un data set de 45 a 50 imágenes de un producto bueno.

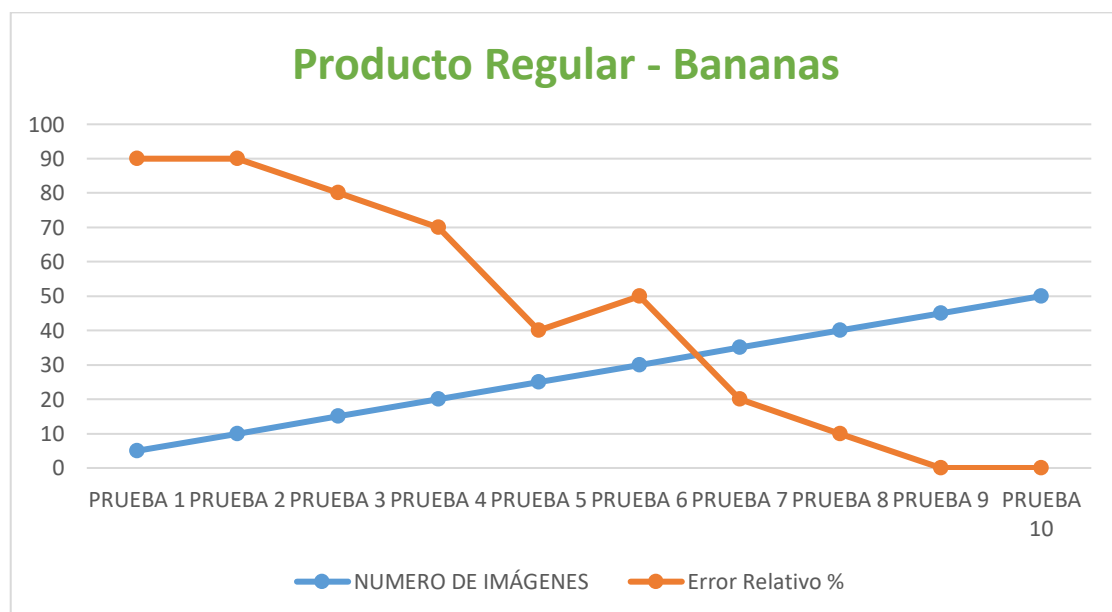
### 3.2.1.2. Entrenamiento de la calidad regular Producto 1 Bananas

Para el producto 1 bananas calidad regular se sigue el mismo procedimiento obteniendo los siguientes resultados, tabla 3-3 y gráfico 2-3.

**Tabla 3-3:** Evaluación del sistema etapa de entrenamiento, producto regular bananas.

# PRUEBA	# IMÁGENES	ACIERTOS	FRACASOS	MUESTRA TOTAL	Error Absoluto	Error Relativo %
PRUEBA 1	5	1	9	10	9	90
PRUEBA 2	10	1	9	10	9	90
PRUEBA 3	15	2	8	10	8	80
PRUEBA 4	20	3	7	10	7	70
PRUEBA 5	25	6	4	10	4	40
PRUEBA 6	30	5	5	10	5	50
PRUEBA 7	35	8	2	10	2	20
PRUEBA 8	40	9	1	10	1	10
PRUEBA 9	45	10	0	10	0	0
PRUEBA 10	50	10	0	10	0	0

Realizado por: Guailazaca González, Carlos Andrés, 2020



**Gráfico 2-3.** Error relativo vs número de imágenes, producto regular - bananas

Realizado por: Guailazaca González, Carlos Andrés, 2020

El gráfico 2-3 representa el error relativo vs número de imágenes capturadas interpretados en la curva que desciende a un error relativo del 20% en la prueba 7 y un error 0 en la prueba 9 y 10 con un margen de 45 a 50 imágenes capturadas.

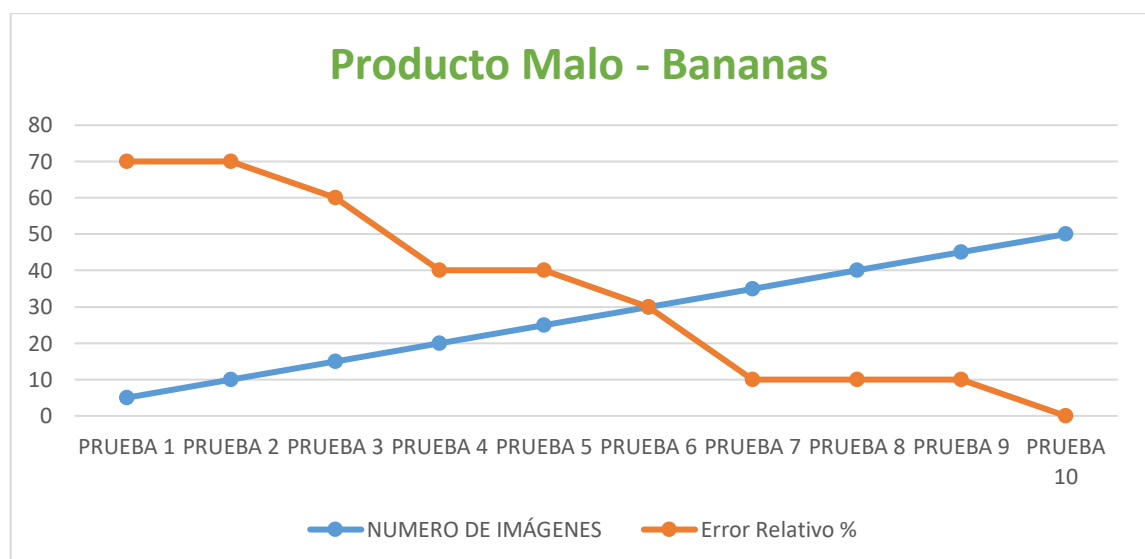
### 3.2.1.3. Entrenamiento de la calidad mala Producto 1 Bananas.

Siguiendo el procedimiento de las dos variables anteriores, ahora con una nueva calidad de producto se obtuvo la siguiente información, tabla 4-3 y gráfico 3-3:

**Tabla 4-3:** Evaluación del sistema etapa de entrenamiento, producto malo bananas

# PRUEBA	# IMÁGENES	ACIERTOS	FRACASOS	MUESTRA TOTAL	Error Absoluto	Error Relativo %
PRUEBA 1	5	3	7	10	7	70
PRUEBA 2	10	3	7	10	7	70
PRUEBA 3	15	4	6	10	6	60
PRUEBA 4	20	6	4	10	4	40
PRUEBA 5	25	6	4	10	4	40
PRUEBA 6	30	7	3	10	3	30
PRUEBA 7	35	9	1	10	1	10
PRUEBA 8	40	9	1	10	1	10
PRUEBA 9	45	9	1	10	1	10
PRUEBA 10	50	10	0	10	0	0

Realizado por: Guallazaca González, Carlos Andrés, 2020



**Gráfico 3-3.** Error relativo vs número de imágenes, producto malo- bananas

Realizado por: Guallazaca González, Carlos Andrés, 2020

El error relativo vs número de imágenes reflejado en el gráfico 3-3 interpretan una eficiencia del 10% desde la prueba 7 y un error relativo de 0% a partir de la prueba 10 con 50 imágenes en el conjunto de datos.

### 3.2.2. Etapa Entrenamiento para el Producto 2 Naranjas.

Para tener una mejor perspectiva se creó un nuevo conjunto de datos con un producto diferente que es el producto 2 (naranjas), para comprobar las estadísticas con relación al producto anterior (bananas).

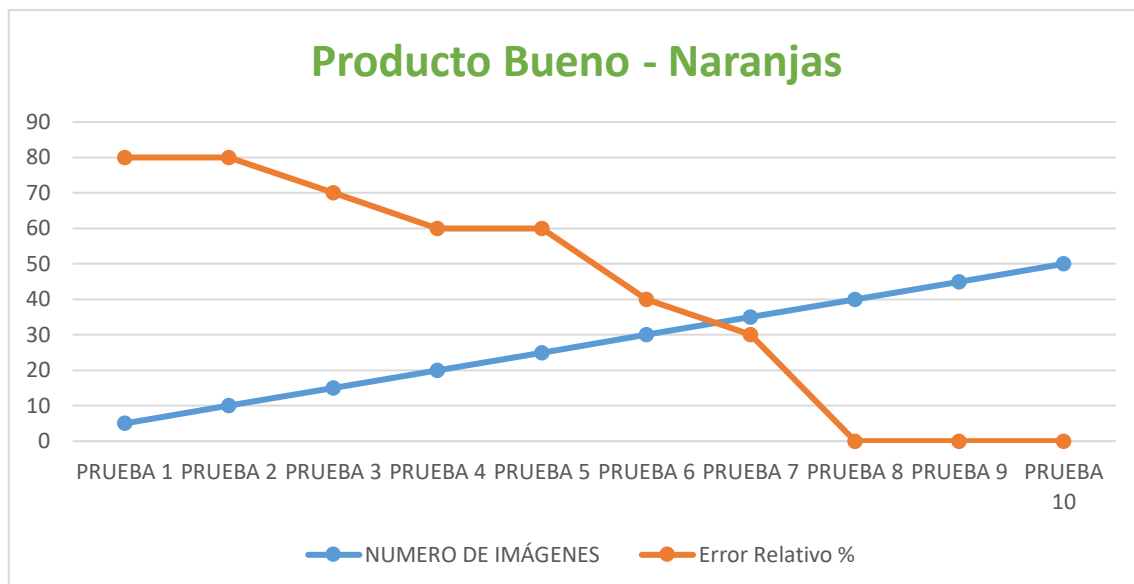
#### 3.2.2.1. Entrenamiento de la calidad buena para el Producto 2 Naranjas

Siguiendo la metodología preliminar para la verificación de un análisis de resultado, se recrean las 10 pruebas con naranjas, obteniendo la siguiente información:

**Tabla 5-3:** Evaluación del sistema etapa de entrenamiento, producto bueno naranjas

# PRUEBA	# IMÁGENES	ACIERTOS	FRACASOS	MUESTRA TOTAL	Error Absoluto	Error Relativo %
PRUEBA 1	5	2	8	10	8	80
PRUEBA 2	10	2	8	10	8	80
PRUEBA 3	15	3	7	10	7	70
PRUEBA 4	20	4	6	10	6	60
PRUEBA 5	25	4	6	10	6	60
PRUEBA 6	30	6	4	10	4	40
PRUEBA 7	35	7	3	10	3	30
PRUEBA 8	40	10	0	10	0	0
PRUEBA 9	45	10	0	10	0	0
PRUEBA 10	50	10	0	10	0	0

Realizado por: Guallazaca González, Carlos Andrés, 2020



**Gráfico 4-3.** Error relativo vs número de imágenes, producto bueno - Naranjas  
**Realizado por:** Guallazaca González, Carlos Andrés, 2020

El gráfico 4-3 muestra el descenso de la curva del error relativo con un valor del 30% en la prueba 7 y una estabilidad del 0% en la prueba 8, 9 y 10, que determina una eficiencia a partir de las 40 imágenes capturadas.

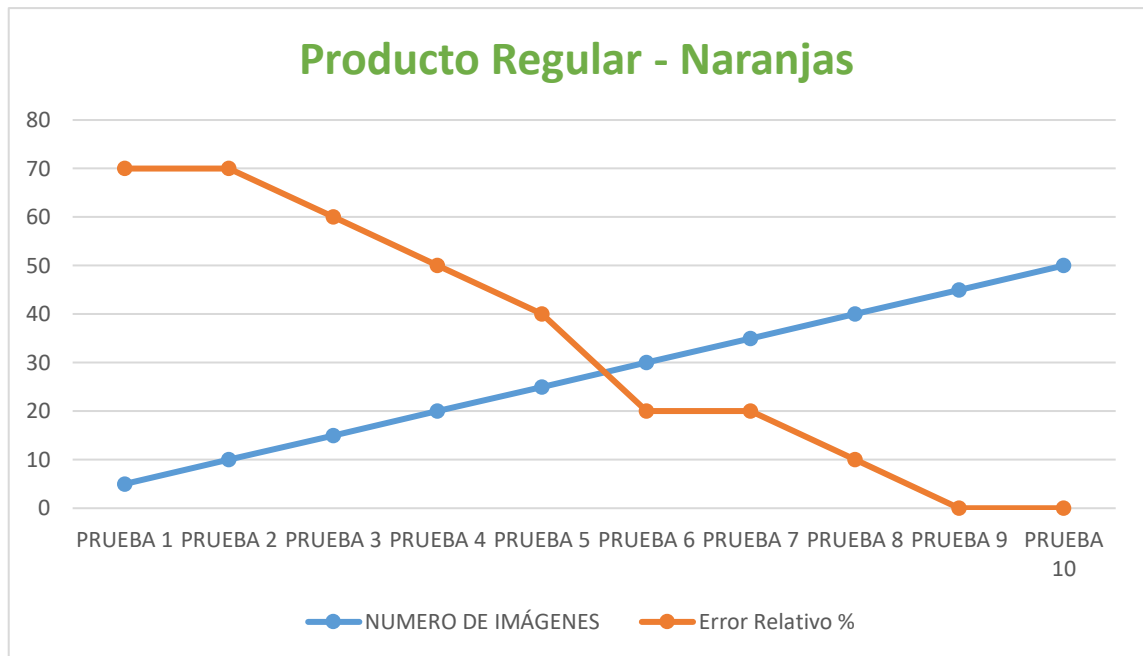
### 3.2.2.2. Entrenamiento de la calidad regular para el Producto 2 Naranjas.

Para el caso de las naranjas con calidad regular se muestra la información ilustrada en la tabla 6-3 y el gráfico 5-3:

**Tabla 6-3:** Evaluación del sistema etapa de entrenamiento, producto regular naranjas

# PRUEBA	# IMÁGENES	ACIERTOS	FRACASOS	MUESTRA TOTAL	Error Absoluto	Error Relativo %
PRUEBA 1	5	3	7	10	7	70
PRUEBA 2	10	3	7	10	7	70
PRUEBA 3	15	4	6	10	6	60
PRUEBA 4	20	5	5	10	5	50
PRUEBA 5	25	6	4	10	4	40
PRUEBA 6	30	8	2	10	2	20
PRUEBA 7	35	8	2	10	2	20
PRUEBA 8	40	9	1	10	1	10
PRUEBA 9	45	10	0	10	0	0
PRUEBA 10	50	10	0	10	0	0

**Realizado por:** Guallazaca González, Carlos Andrés, 2020



**Gráfico 5-3.** Error relativo vs número de imágenes, producto regular- Naranjas

Realizado por: Guallazaca González, Carlos Andrés, 2020

El gráfico 5-3 interpreta una eficiencia desde la prueba 8 con un 10% de error relativo y 0% en las pruebas 9 (45 imágenes capturadas) y prueba 10 (50 imágenes capturadas).

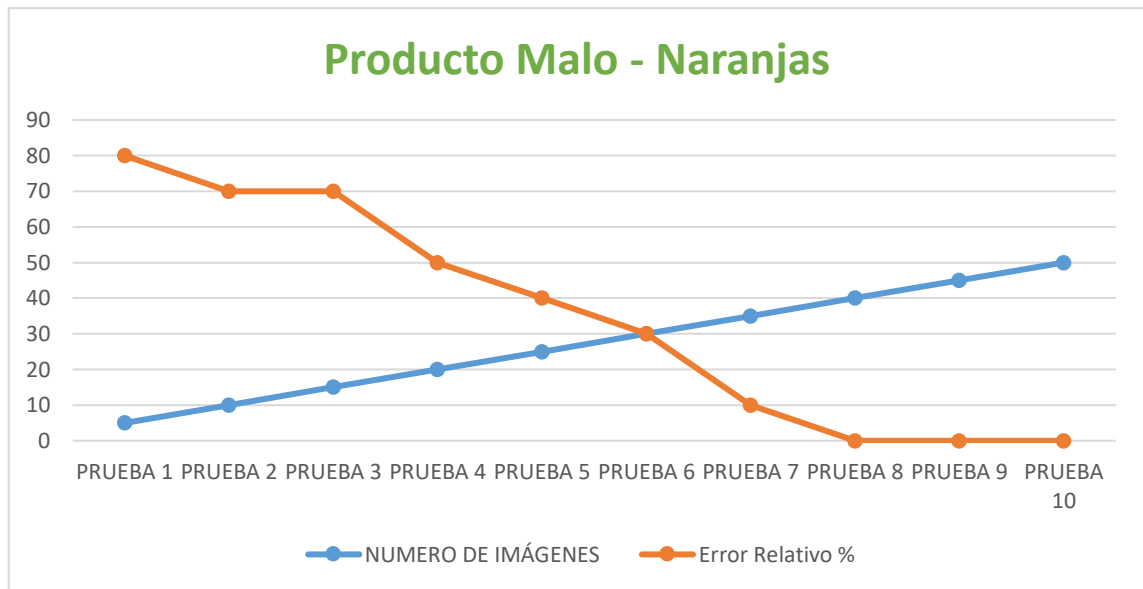
### 3.2.2.3. Entrenamiento de la calidad malo para el Producto 2 Naranjas.

En la prueba realizada del segundo producto (Naranjas) para la calidad mala se determinó la tabla 7-3 y el gráfico 6-3

**Tabla 7-3:** Evaluación del sistema etapa de entrenamiento, producto malo naranjas

# PRUEBA	# IMÁGENES	ACIERTOS	FRACASOS	MUESTRA TOTAL	Error Absoluto	Error Relativo %
PRUEBA 1	5	2	8	10	8	80
PRUEBA 2	10	3	7	10	7	70
PRUEBA 3	15	3	7	10	7	70
PRUEBA 4	20	5	5	10	5	50
PRUEBA 5	25	6	4	10	4	40
PRUEBA 6	30	7	3	10	3	30
PRUEBA 7	35	9	1	10	1	10
PRUEBA 8	40	10	0	10	0	0
PRUEBA 9	45	10	0	10	0	0
PRUEBA 10	50	10	0	10	0	0

Realizado por: Guallazaca González, Carlos Andrés, 2020



**Gráfico 6-3.** Error relativo vs número de imágenes, producto malo- naranjas  
**Realizado por:** Guailazaca González, Carlos Andrés, 2020

El gráfico 6-3 al igual que las pruebas anteriores, determina un error relativo del 10% en la prueba 7 y un error de 0 para las pruebas 8, 9 y 10, demostrando la proporcionalidad del éxito según el número de imágenes capturadas del producto.

### 3.2.3. Etapa Entrenamiento para el Producto 3 Plátano Verde.

En esta sección se realiza un contraste en los productos utilizados para evaluar el rendimiento de KNN. En las secciones anteriores se resolvió el entrenamiento con productos de un color similar, pero con formas diferente, para esta tarea se utilizó plátano verde sección 3.2.3 y manzanas sección 3.2.4, y se obtuvieron los siguientes resultados para el producto 3:

#### 3.2.3.1. Entrenamiento de la calidad buena para el Producto 3 Plátano Verde

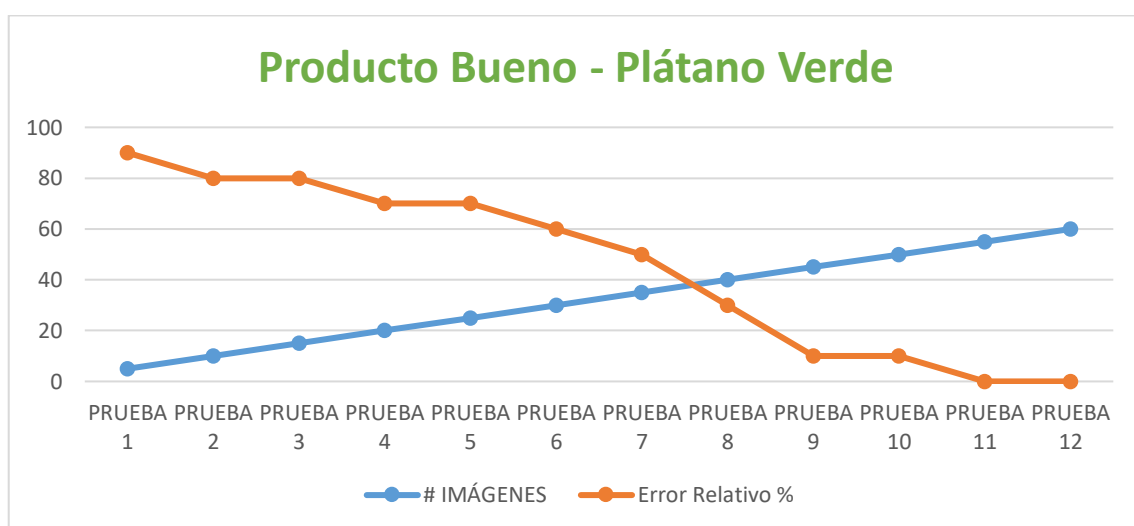
En la prueba realizada del tercer producto (plátano verde) para la calidad buena se determinó la tabla 8-3 y el gráfico 7-3:



**Tabla 8-3:** Evaluación del sistema etapa de entrenamiento, producto bueno plátano verde.

# PRUEBA	# IMÁGENES	ACIERTOS	FRACASOS	MUESTRA TOTAL	Error Absoluto	Error Relativo %
PRUEBA 1	5	1	9	10	9	90
PRUEBA 2	10	2	8	10	8	80
PRUEBA 3	15	2	8	10	8	80
PRUEBA 4	20	3	7	10	7	70
PRUEBA 5	25	3	7	10	7	70
PRUEBA 6	30	4	6	10	6	60
PRUEBA 7	35	5	5	10	5	50
PRUEBA 8	40	7	3	10	3	30
PRUEBA 9	45	9	1	10	1	10
PRUEBA 10	50	9	1	10	1	10
PRUEBA 11	55	10	0	10	0	0
PRUEBA 12	60	10	0	10	0	0

Realizado por: Guallazaca González, Carlos Andrés, 2020



**Gráfico 7-3.** Error relativo vs número de imágenes, producto Bueno- Plátano Verde

Realizado por: Guallazaca González, Carlos Andrés, 2020

Para la etapa de entrenamiento del producto 3 calidad bueno (plátano verde) se tuvo que agregar 2 pruebas más, con un total de 60 imágenes para obtener un error relativo del 10% en la prueba número 9 y 10, demostrando de igual manera la proporcionalidad del éxito para obtener un modelo capaz de resolver incertidumbres y no recaiga en un proceso aprendido o repetitivo.

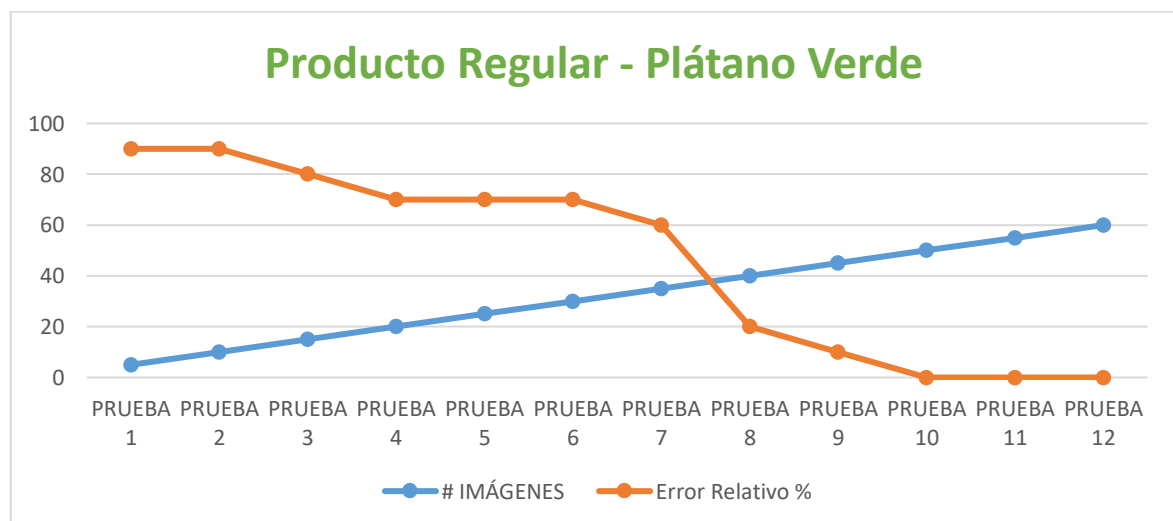
### 3.2.3.2. Entrenamiento de la calidad regular para el Producto 3 Plátano Verde

De igual manera en esta sección se consideran las 12 pruebas realizadas al inicio con el producto bueno para tener un nivel de homogeneidad en la toma de datos y retroalimentación para el sistema. Para ello se sigue la misma metodología obteniendo los siguientes resultados reflejados en la tabla 9-3 y gráfico 8-3.

**Tabla 9-3:** Evaluación del sistema etapa de entrenamiento, producto regular plátano verde.

# PRUEBA	# IMÁGENES	ACIERTOS	FRACASOS	CANTIDAD PRUEBA	Error Absoluto	Error Relativo %
PRUEBA 1	5	1	9	10	9	90
PRUEBA 2	10	1	9	10	9	90
PRUEBA 3	15	2	8	10	8	80
PRUEBA 4	20	3	7	10	7	70
PRUEBA 5	25	3	7	10	7	70
PRUEBA 6	30	3	7	10	7	70
PRUEBA 7	35	4	6	10	6	60
PRUEBA 8	40	8	2	10	2	20
PRUEBA 9	45	9	1	10	1	10
PRUEBA 10	50	10	0	10	0	0
PRUEBA 11	55	10	0	10	0	0
PRUEBA 12	60	10	0	10	0	0

Realizado por: Guallazaca González, Carlos Andrés, 2020



**Gráfico 8-3.** Error relativo vs número de imágenes, producto regular-plátano verde

Realizado por: Guallazaca González, Carlos Andrés, 2020

Se hace una observación para el comportamiento de estos dos últimos gráficos 7-3 y 8-3, la calidad buena y regular para el producto 3 plátano verde tiene un gran porcentaje de similitud y

el sistema requiere de más imágenes para que KNN pueda diferenciar y clasificar los puntos similares para cada calidad respectivamente.

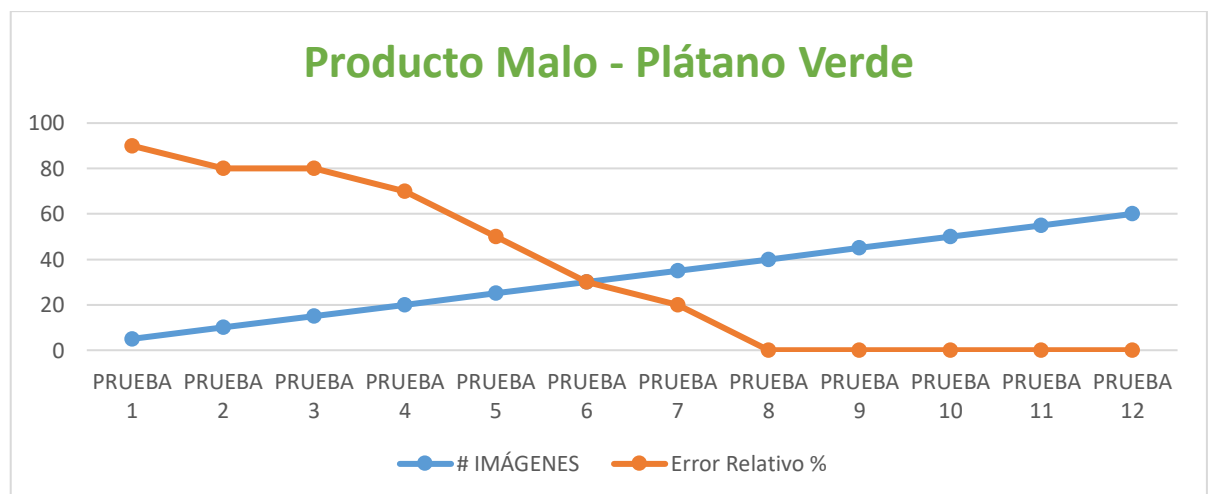
### 3.2.3.3. Entrenamiento de la calidad malo para el Producto 3 Plátano Verde

Siguiendo la misma metodología en la etapa de entrenamiento para del tercer producto plátano verde para la calidad mala se determinó la tabla 10-3 y el gráfico 9-3:

**Tabla 10-3:** Evaluación del sistema etapa de entrenamiento, producto malo plátano verde.

# PRUEBA	# IMÁGENES	ACIERTOS	FRACASOS	CANTIDAD PRUEBA	Error Absoluto	Error Relativo %
PRUEBA 1	5	1	9	10	9	90
PRUEBA 2	10	2	8	10	8	80
PRUEBA 3	15	2	8	10	8	80
PRUEBA 4	20	3	7	10	7	70
PRUEBA 5	25	5	5	10	5	50
PRUEBA 6	30	7	3	10	3	30
PRUEBA 7	35	8	2	10	2	20
PRUEBA 8	40	10	0	10	0	0
PRUEBA 9	45	10	0	10	0	0
PRUEBA 10	50	10	0	10	0	0
PRUEBA 11	55	10	0	10	0	0
PRUEBA 12	60	10	0	10	0	0

Realizado por: Guallazaca González, Carlos Andrés, 2020



**Gráfico 9-3.** Error relativo vs número de imágenes, producto malo-plátano verde

Realizado por: Guallazaca González, Carlos Andrés, 2020

En el gráfico 9-3 se observa un descenso en la curva a un error relativo del 20% en la prueba 7 con 35 imágenes, esto indica que en las pruebas posteriores se minimizará aún más el error que

es palpable desde la prueba 8 en adelante. La diferencia del gráfico de la calidad mala en comparación a los gráficos de calidad buena y regular es significativa, ya que el producto plátano verde calidad mala pertenece a otro subproducto derivado del mismo, y no entraría a la clasificación del plátano verde, el cambio de un estado al otro es muy evidente en su coloración, esto facilita el procesamiento de knn para este producto.

### 3.2.4. Etapa Entrenamiento para el Producto 4 Manzanas.

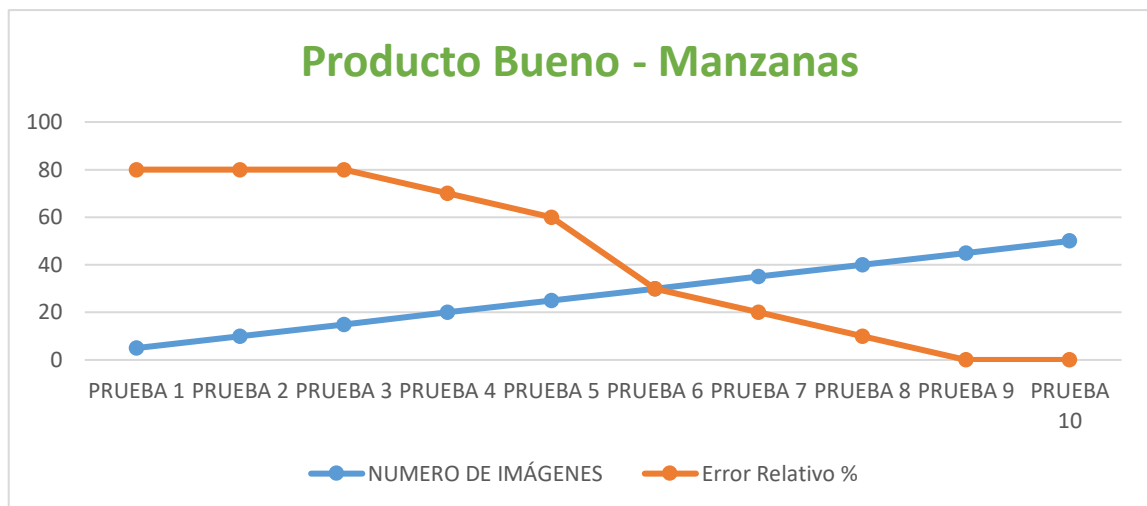
#### 3.2.4.1. Entrenamiento de la calidad bueno Producto 4 Manzanas

Siguiendo la misma metodología para los productos anteriores, finalmente, se realizan las pruebas con el cuarto producto correspondiente a manzanas. Los resultados se exponen en la tabla 11-3 y el gráfico 10-3 a continuación:

**Tabla 11-3:** Evaluación del sistema etapa de entrenamiento, producto bueno manzanas.

# PRUEBA	# IMÁGENES	ACIERTOS	FRACASOS	CANTIDAD PRUEBA	Error Absoluto	Error Relativo %
PRUEBA 1	5	2	8	10	8	80
PRUEBA 2	10	2	8	10	8	80
PRUEBA 3	15	2	8	10	8	80
PRUEBA 4	20	3	7	10	7	70
PRUEBA 5	25	4	6	10	6	60
PRUEBA 6	30	7	3	10	3	30
PRUEBA 7	35	8	2	10	2	20
PRUEBA 8	40	9	1	10	1	10
PRUEBA 9	45	10	0	10	0	0
PRUEBA 10	50	10	0	10	0	0

**Realizado por:** Guallazaca González, Carlos Andrés, 2020



**Gráfico 10-3.** Error relativo vs número de imágenes, producto bueno-manzanas

Realizado por: Guallazaca González, Carlos Andrés, 2020

En el gráfico 10-3 se observa a un error relativo del 10% en la prueba 8 con 40 imágenes y una proporcionalidad de éxito en las pruebas 9 y 10 con 45 y 50 imágenes.

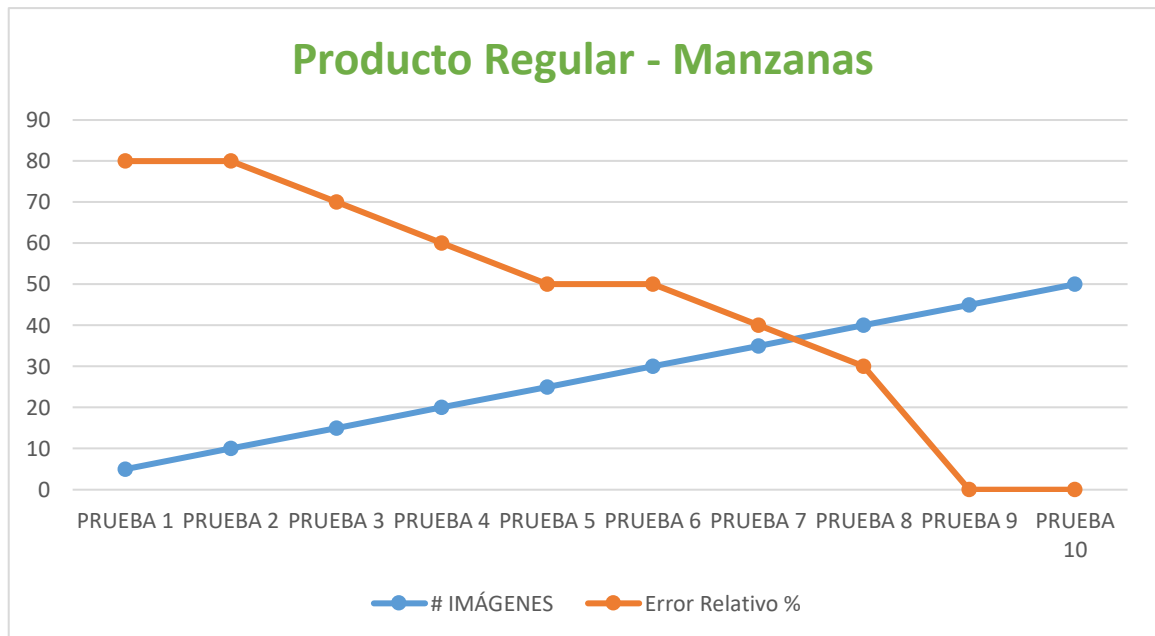
### 3.2.4.2. Entrenamiento de la calidad regular Producto 4 Manzanas

Para el producto 4 regular se sigue el mismo procedimiento obteniendo los siguientes resultados, tabla 12-3 y gráfico 11-3.

**Tabla 12-3:** Evaluación del sistema etapa de entrenamiento, producto regular manzanas.

# PRUEBA	# IMÁGENES	ACIERTOS	FRACASOS	CANTIDAD PRUEBA	Error Absoluto	Error Relativo %
PRUEBA 1	5	2	8	10	8	80
PRUEBA 2	10	2	8	10	8	80
PRUEBA 3	15	3	7	10	7	70
PRUEBA 4	20	4	6	10	6	60
PRUEBA 5	25	5	5	10	5	50
PRUEBA 6	30	5	5	10	5	50
PRUEBA 7	35	6	4	10	4	40
PRUEBA 8	40	7	3	10	3	30
PRUEBA 9	45	10	0	10	0	0
PRUEBA 10	50	10	0	10	0	0

Realizado por: Guallazaca González, Carlos Andrés, 2020



**Gráfico 11-3.** Error relativo vs número de imágenes, producto regular-manzanas  
**Realizado por:** Guallazaca González, Carlos Andrés, 2020

El gráfico 10-3 representa el error relativo vs número de imágenes capturadas interpretados en la curva que desciende a un error relativo del 30% en la prueba 8 y un error 0 en la prueba 9 y 10 con un margen de 45 a 50 imágenes capturadas.

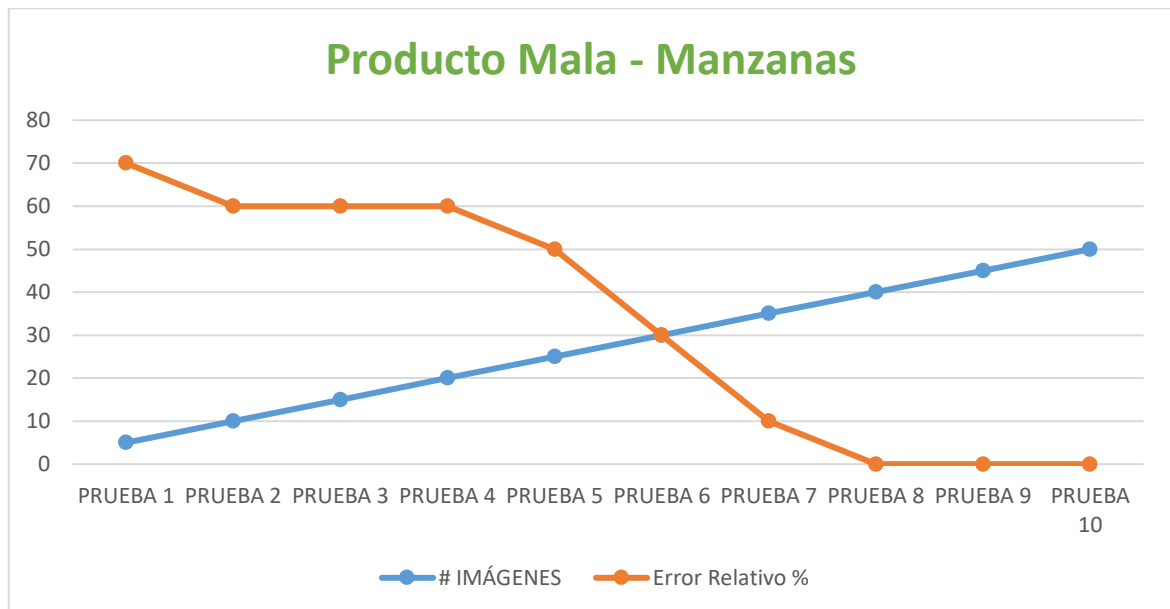
#### 3.2.4.3. Entrenamiento de la calidad mala Producto 4 Manzanas.

Siguiendo con la metodología, ahora con una nueva calidad de producto se obtuvo la siguiente información, tabla 13-3 y gráfico 12-3:

**Tabla 13-3:** Evaluación del sistema etapa de entrenamiento, producto malo manzanas.

# PRUEBA	# IMÁGENES	ACIERTOS	FRACASOS	CANTIDAD PRUEBA	Error Absoluto	Error Relativo %
PRUEBA 1	5	3	7	10	7	70
PRUEBA 2	10	4	6	10	6	60
PRUEBA 3	15	4	6	10	6	60
PRUEBA 4	20	4	6	10	6	60
PRUEBA 5	25	5	5	10	5	50
PRUEBA 6	30	7	3	10	3	30
PRUEBA 7	35	9	1	10	1	10
PRUEBA 8	40	10	0	10	0	0
PRUEBA 9	45	10	0	10	0	0
PRUEBA 10	50	10	0	10	0	0

**Realizado por:** Guallazaca González, Carlos Andrés, 2020



**Gráfico 12-3.** Error relativo vs número de imágenes, producto malo-manzanas

**Realizado por:** Guallazaca González, Carlos Andrés, 2020

El error relativo vs número de imágenes, que se refleja en el gráfico 11-3 interpretan una eficiencia del 90% desde la prueba 7 con un error relativo del 10% y un error relativo de 0% a partir de la prueba 8 con 40 imágenes en el conjunto de datos.

### 3.3. Resumen - Etapa de Entrenamiento.

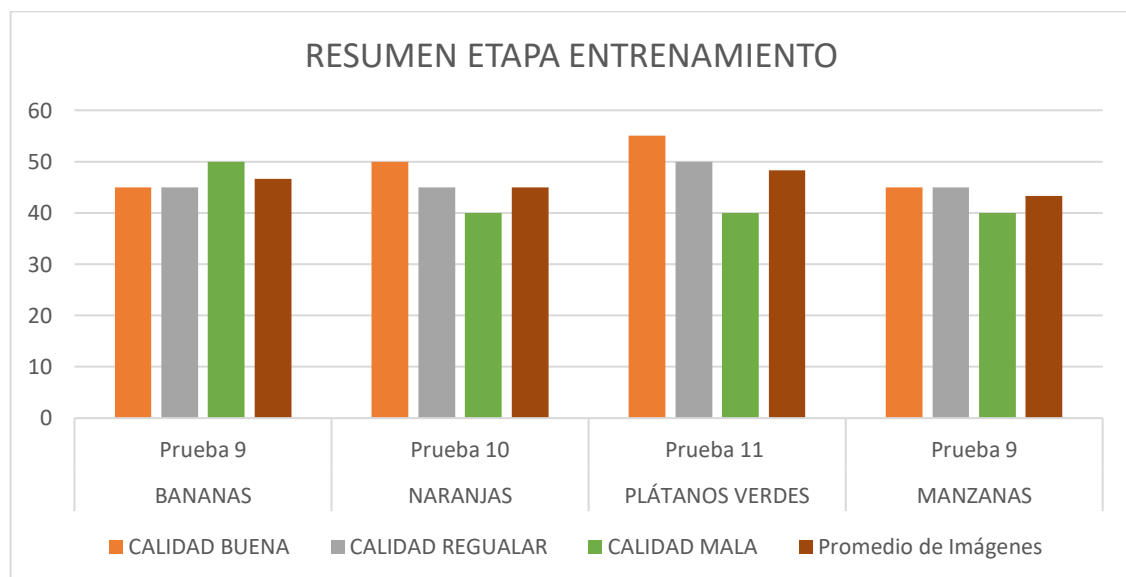
Terminada la etapa del entrenamiento se obtuvieron los resultados expuestos en la tabla 14-3. Los valores escogidos representan a las pruebas exitosas de acuerdo con el error mínimo (0%) en el proceso y el número de imágenes capturadas correspondientes.

**Tabla 14-3:** Resumen etapa entrenamiento.

PRODUCTOS	CALIDADES						Total, Imágenes
	CALIDAD BUENA	# IMÁGENES	CALIDAD REGULAR	# IMÁGENES	CALIDAD MALA	# IMÁGENES	
BANANAS	Prueba 9	45	Prueba 9	45	Prueba 10	50	140
NARANJAS	Prueba 10	50	Prueba 9	45	Prueba 8	40	135
PLÁTANOS VERDES	Prueba 11	55	Prueba 10	50	Prueba 8	40	145
MANZANAS	Prueba 9	45	Prueba 9	45	Prueba 8	40	130
Imágenes Procesadas							550

**Realizado por:** Guallazaca González, Carlos Andrés, 2020

Respondiendo a la interrogante planteada en la sección 3.2, mediante el gráfico 12-3, el sistema para la etapa de entrenamiento fue necesario utilizar un promedio entre 40 y 55 imágenes para llegar a un resultado satisfactorio y generar un modelo flexible que se lo utilizará en la etapa de evaluación.



**Gráfico 13-3.** Resumen etapa entrenamiento exitoso.

**Realizado por:** Guallazaca González, Carlos Andrés, 2020



### **3.4. Evaluación final del Sistema - Etapa de Validación.**

En esta sección se ejecuta una metodología diferente a la anterior, cabe recalcar que para la etapa de entrenamiento se utilizó pruebas de validación para escoger y determinar el data set o modelo.js adecuado que se ajusta a cada producto mediante capturas de imágenes, es decir con la analítica anterior se genera una base del conocimiento para el sistema, y de esta manera evaluar la etapa de validación con productos diferentes y calidades aleatorias.

Para esta etapa, se utilizó un conjunto de productos reservados que no fueron utilizados en la sección 3.2, etapa de entrenamiento para el sistema. Es decir, el producto es nuevo y no hay registro de este.

La metodología que se utilizó en esta etapa es la siguiente: se reservó 9 bananas (3 de calidad buena, 3 de calidad regular y 3 de calidad mala), 9 naranjas (3 de calidad buena, 3 de calidad regular y 3 de calidad mala), 9 plátanos verdes (3 de calidad buena, 3 de calidad regular y 3 de calidad mala) y 9 manzanas (3 de calidad buena, 3 de calidad regular y 3 de calidad mala) cabe recalcar que estos productos no fueron utilizados anteriormente, el sistema no los conoce pero sabe diferenciarlos. Las calidades se evaluaron aleatoriamente por cinco ocasiones, y se obtuvieron los siguientes resultados:

#### ***3.4.1. Etapa de Evaluación para el Producto 1 Bananas.***

Para la etapa de evaluación con el primer producto (bananas) se obtuvo un resultado exitoso, el sistema reconoció sin dificultad las diferentes calidades del producto como refleja en la tabla 14-3 ilustrada a continuación:

**Tabla 15-3:** Etapa de validación producto 1 bananas.

Pruebas de validación	Etapa Preclasificación			Etapa Post clasificación			Total, Productos	Productos Erróneos	Error %
	Calidad Buena	Calidad Regular	Calidad Mala	Calidad Buena	Calidad Regular	Calidad Mala			
P. Validación 1	3	3	3	3	3	3	9	0	0
P. Validación 2	3	3	3	3	3	3	9	0	0
P. Validación 3	3	3	3	3	3	3	9	0	0
P. Validación 4	3	3	3	3	3	3	9	0	0
P. Validación 5	3	3	3	3	3	3	9	0	0
								Promedio de Error	0

**Realizado por:** Guallazaca González, Carlos Andrés, 2020

Con el modelo generado según la tabla 13-3, respondemos a la interrogante de la sección 3.2, el sistema cumple satisfactoriamente la identificación de las calidades del primer producto con un modelo generado a partir de 140 imágenes capturadas en diferentes posiciones de la fruta.

### ***3.4.2. Etapa de Evaluación para el Producto 2 Naranjas.***

En la etapa de evaluación para el producto 2 naranjas se evidenció 2 errores en la identificación, específicamente en la prueba de validación 1 y 3 como se observa en la tabla 15-3 ilustrada a continuación:

**Tabla 16-3:** Etapa de validación producto 2 naranjas.

Pruebas de validación	Etapa Preclasificación			Etapa Post clasificación			Total, Productos	Productos Erróneos	Error %
	Calidad Buena	Calidad Regular	Calidad Mala	Calidad Buena	Calidad Regular	Calidad Mala			
P. Validación 1	3	3	3	3	4	2	9	1	11,1
P. Validación 2	3	3	3	3	3	3	9	0	0
P. Validación 3	3	3	3	4	2	3	9	1	11,1
P. Validación 4	3	3	3	3	3	3	9	0	0
P. Validación 5	3	3	3	3	3	3	9	0	0
								Promedio de Error	4,44

**Realizado por:** Guallazaca González, Carlos Andrés, 2020

En la tabla 15-3 refleja el porcentaje de error de acuerdo con el número de aciertos que se tuvo al momento de la identificación, obteniendo un error del 11.11% con un error en la prueba de validación 1 con la identificación en la calidad regular y en la prueba de validación 3 con el mismo porcentaje de error y el fallo ocurrió con la calidad buena. El promedio del error en esta prueba de validación fue el 4,44%.

### ***3.4.3. Etapa de Evaluación para el Producto 3 Plátanos Verdes.***

Siguiendo con la misma metodología, para el producto 3 (plátanos verdes) se obtiene la tabla 16-3. El sistema con este producto tuvo 3 errores en las pruebas de validación 1, 3 y 4, erróneamente identificó un producto regular siendo de la calidad buena, esto representa un error del 11,11% en cada prueba de validación referente a las cantidades evaluadas, con un promedio en toda la actividad realizada del 6.6% de error.

**Tabla 17-3:** Etapa de validación producto 3 plátanos verdes.

Pruebas de validación	Etapa Preclasificación			Etapa Post clasificación			Total, Productos	Productos Erróneos	Error %
	Calidad Buena	Calidad Regular	Calidad Mala	Calidad Buena	Calidad Regular	Calidad Mala			
P. Validación 1	3	3	3	2	4	3	9	0	11,1
P. Validación 2	3	3	3	3	3	3	9	0	0
P. Validación 3	3	3	3	2	4	3	9	0	11,1
P. Validación 4	3	3	3	2	4	3	9	0	11,1
P. Validación 5	3	3	3	3	3	3	9	0	0
								Promedio de Error	6,66

**Realizado por:** Guallazaca González, Carlos Andrés, 2020

#### ***3.4.4. Etapa de Evaluación para el Producto 4 Manzanas.***

En esta etapa de evaluación con el cuarto producto (manzanas) satisfactoriamente no se evidenció errores en la identificación, como se refleja en la tabla 17-3 ilustrada a continuación:

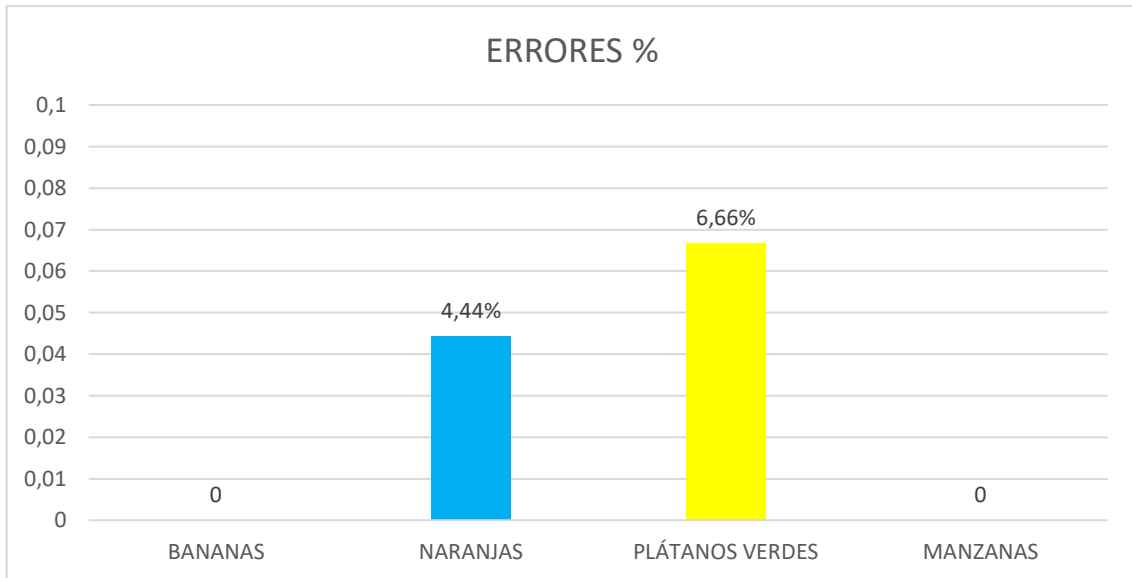
**Tabla 18-3:** Etapa de validación producto 4 manzanas.

Pruebas de validación	Etapa Preclasificación			Etapa Post clasificación			Total, Productos	Productos Erróneos	Error %
	Calidad Buena	Calidad Regular	Calidad Mala	Calidad Buena	Calidad Regular	Calidad Mala			
P. Validación 1	3	3	3	3	3	3	9	0	0
P. Validación 2	3	3	3	3	3	3	9	0	0
P. Validación 3	3	3	3	3	3	3	9	0	0
P. Validación 4	3	3	3	3	3	3	9	0	0
P. Validación 5	3	3	3	3	3	3	9	0	0
								Promedio de Error	0

**Realizado por:** Guallazaca González, Carlos Andrés, 2020

### 3.5. Resumen - Etapa de Validación.

En el gráfico 13-3 que se ilustra a continuación, se evidencia la eficiencia en la identificación de las calidades de cada uno de los productos con respecto a los errores presentados. Considerando el número de pruebas realizadas, que corresponden a 5 pruebas por cada producto con un total de 3 frutas por calidad que equivalen a 36 frutas aisladas a la etapa de entrenamiento como se explica en la introducción de la sección 3.4 que fueron utilizadas para evaluar el sistema como variables nuevas que KNN no conocía, pero fue capaz de identificarlas.



**Gráfico 14-3.** errores del sistema según el producto en la etapa validación.

**Realizado por:** Guallazaca González, Carlos Andrés, 2020

### 3.6. Pruebas de comunicación.

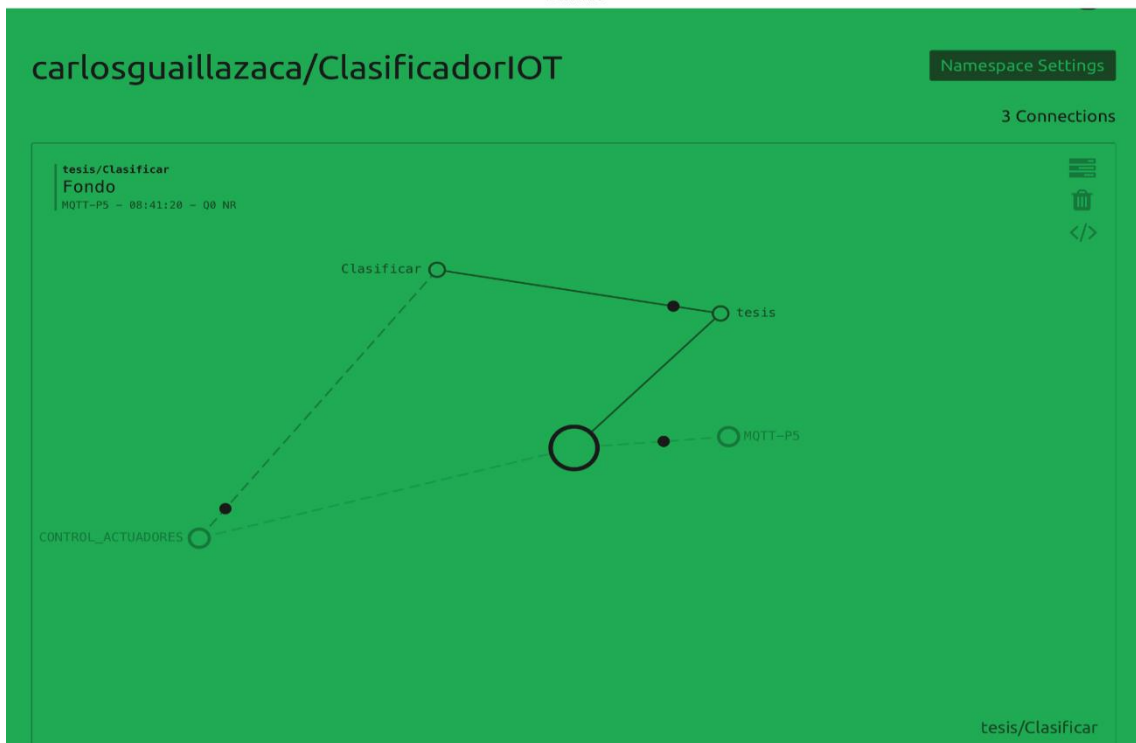
El objetivo de esta sección fue validar el correcto flujo de datos saturando el canal de comunicación, desde MQTT-P5 que es el interfaz web que recibe la información de cada producto mediante la cámara, este se enlazó con el bróker shiftr que es el intermediario entre los siguientes clientes: Control-Actuadores (ESP8266), navegador google Chrome desde el computador y el navegador safari del iPad, estos dos últimos se los utilizó para supervisar y ver la simulación del tráfico de datos junto con el monitor serial del bróker para constatar la llegada del dato correcto.

Utilizando la metodología de la sección anterior de la etapa de validación para el producto bananas, se configuró el fondo que se muestra en la figura 2-3.



**Figura 2-3.** Interfaz web, configuración del fondo.

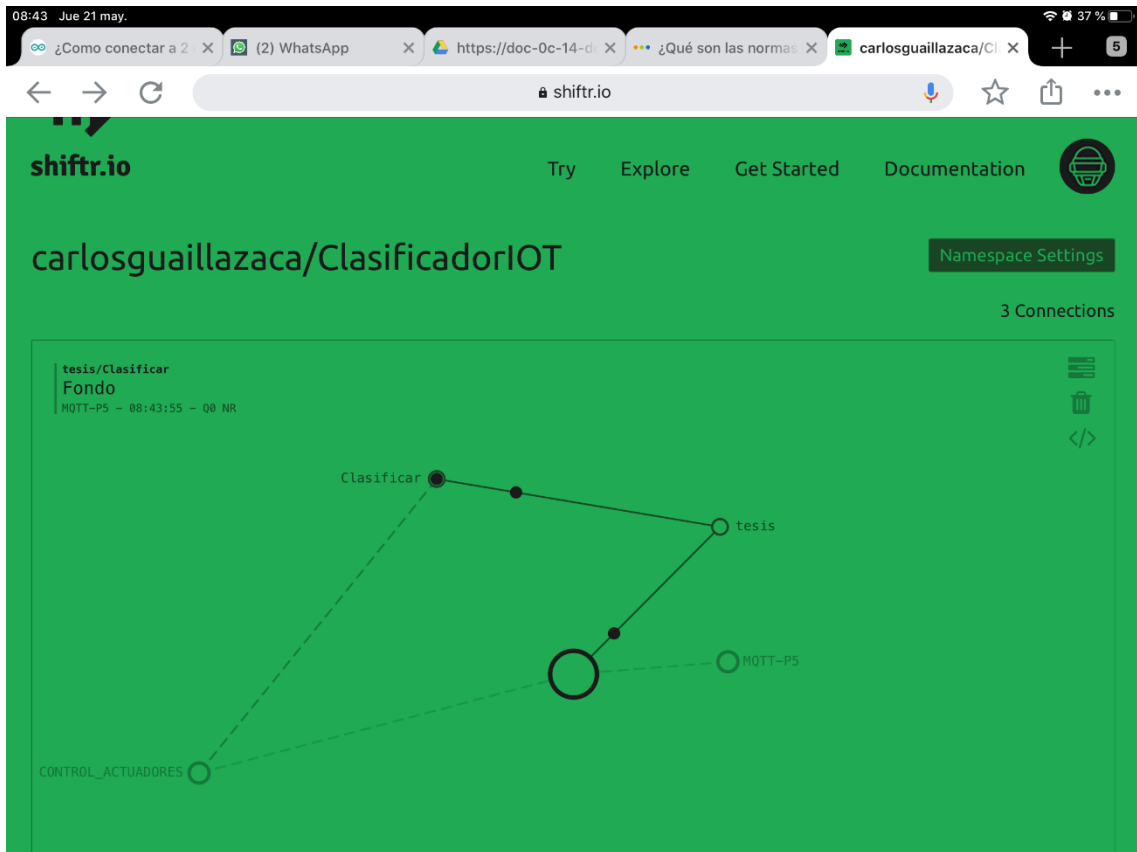
**Realizado por:** Guallazaca González, Carlos Andrés, 2020



**Figura 3-3.** Interfaz web, configuración del fondo.

**Realizado por:** Guallazaca González, Carlos Andrés, 2020

En las figuras 3-3 y 4-3 se observa la simulación del flujo de datos que se envían desde MQTT-P5 que es la aplicación web hasta el control de los actuadores, etiqueta que se le dio al controlador ESP8266. En la parte superior izquierda del script de shiftr se ubica el monitor serial donde llega el identificador de las variables enviadas desde el navegador web, este dato se refleja en el Dashboard del siguiente link del proyecto <https://agnate-coordination.000webhostapp.com/dash.html>



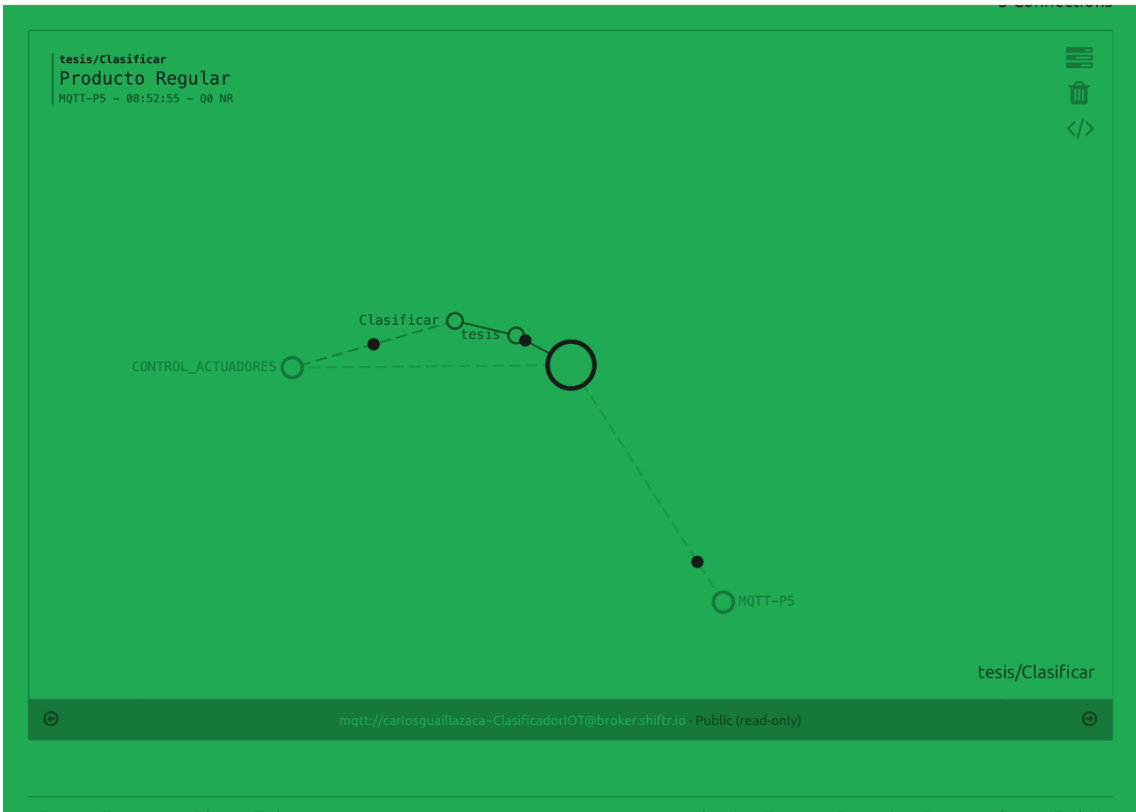
**Figura 4-3.** Interfaz web, configuración del fondo.  
**Realizado por:** Guallazaca González, Carlos Andrés, 2020

En la figura 5-3, se muestra el control de calidad para el producto banana, en sección de visualización pasa el producto bananas con calidad regular y KNN lo etiquetó como calidad regular, en las figuras 6-3 y 7-3 se registran el flujo de datos en el bróker al mismo tiempo.

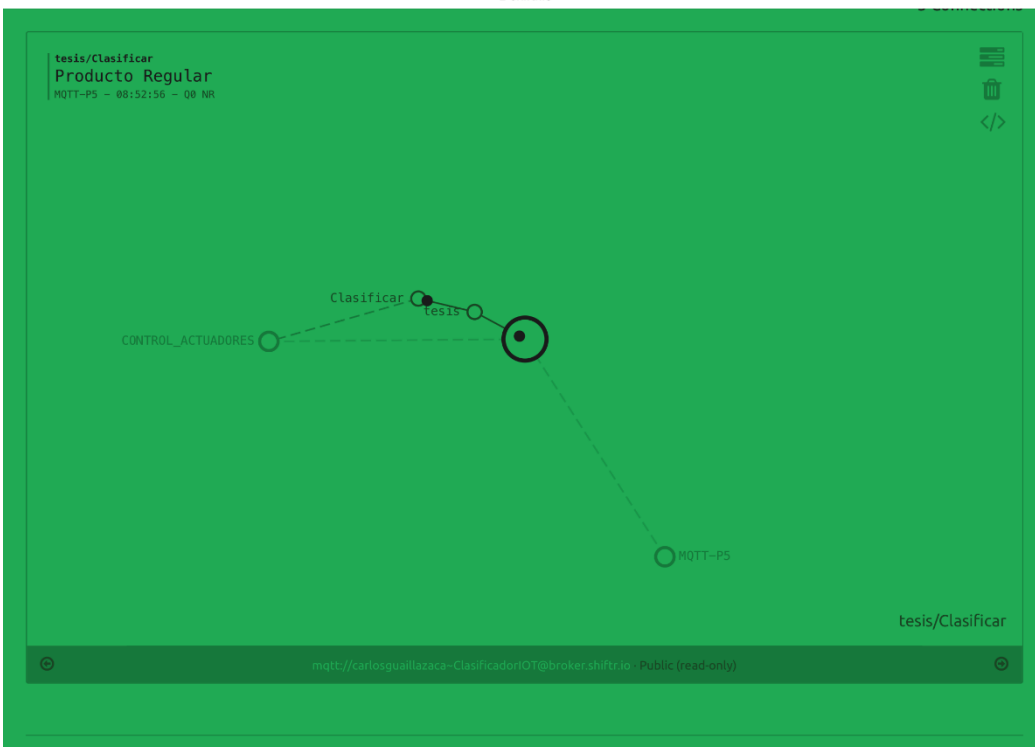


**Figura 5-3.** Interfaz web, Bananas producto regular.  
**Realizado por:** Guallazaca González, Carlos Andrés, 2020





**Figura 6-3.** Shiftr bróker, Bananas producto regular, visualización del flujo de datos 1.  
**Realizado por:** Guallazaca González, Carlos Andrés, 2020

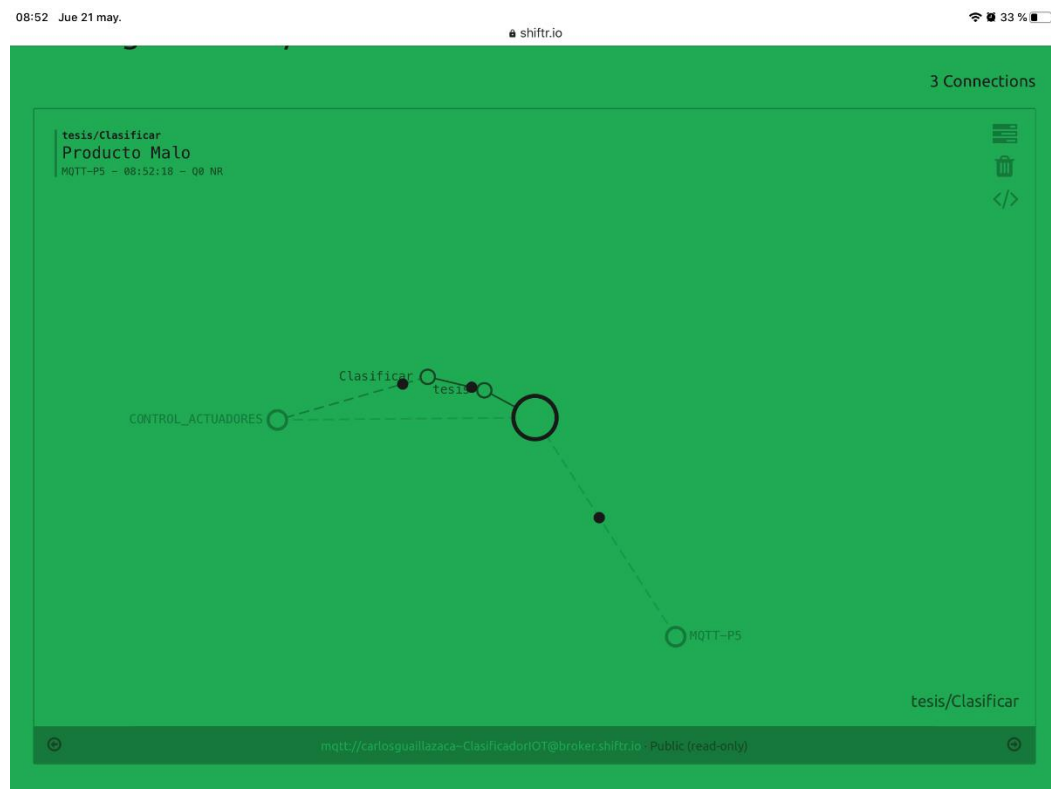


**Figura 7-3.** Shiftr bróker, Bananas producto regular visualización del flujo de datos 2.  
**Realizado por:** Guallazaca González, Carlos Andrés, 2020

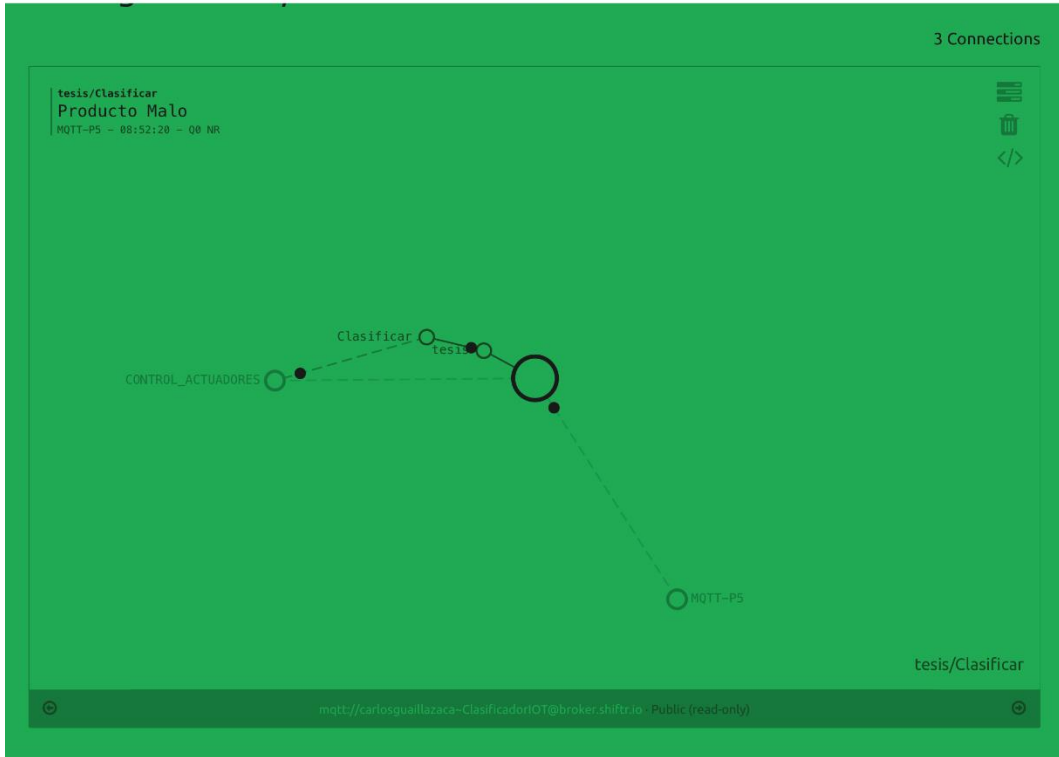
De igual manera en la figura 8-3, con producto bananas y calidad mala, knn etiquetó correctamente la calidad y el dato lo trasporta al bróker ilustrado en las figuras 9-3 y 10-3.



**Figura 8-3.** Interfaz web, Bananas producto malo.  
**Realizado por:** Guailazaca González, Carlos Andrés, 2020



**Figura 9-3.** Interfaz web, Bananas producto malo.  
**Realizado por:** Guailazaca González, Carlos Andrés, 2020



**Figura 10-3.** Interfaz web, Bananas producto malo.  
**Realizado por:** Guillazaca González, Carlos Andrés, 2020

## CAPÍTULO IV

### 4. EVALUACIÓN ECONÓMICA

#### 4.1. Costos

Se realiza un análisis general de los costos efectuados en el diseño e implementación del clasificador 4.0 basado en machine learning.

##### 4.1.1. Costos directos

Son los que se relacionan directamente para la realización o fabricación del modelo prototipo.

##### 4.1.1.1. Costos implementación eléctrica

**Tabla 1-4:** Costos eléctricos

CANTIDAD	ELEMENTO	V. unitario (USD)	V. Total (USD)
1	Computador	800	800
2	NodeMCU ESP8266	12	24
3	Sensores Digitales	30	90
1	Módulo de 1 Relés	18	18
5m	Cable gemelo 14 AWG	0,80	4
1	Motor elevavidrios	50	50
<b>TOTAL</b>			<b>986</b>

Realizado por: Guailazaca González, Carlos Andrés, 2020

##### 4.1.1.2. Costos implementación estructural

**Tabla 2-4:** Costos estructurales

CANTIDAD	ELEMENTO	V. unitario (USD)	V. Total (USD)
2	Vslot 20 x 40 x 1000mm	14.28	28.57
8	Vslot 20 x 20 x 250	7.46	59.68
22	Angulo Simple serie 20	0.90	19.8
4	Chumacera 20mm ¾"	9.00	36
1	Eje Teflón diámetro 40mm	40.00	40.00
1	Eje Teflón principal 40mm	40.00	40.00
1	Banda sintética PVC 190mm	60.00	60.00
44	Tornillo 5 x 0.8 x 10	0.10	4.4
1	Tornillo 5 x 0.8 x 5	0.10	0.10
5	Vslot 20 x 20 x 100mm	7.00	35.00
1	Case Cámara	40.00	40.00

2	Guías de acrílico	10.00	20.00
1	Caja producto 1	5.00	5.00
1	Caja producto 2	5.00	5.00
1	Caja producto 3	5.00	5.00
<b>TOTAL</b>			443.68

**Realizado por:** Guallazaca González, Carlos Andrés, 2020

#### 4.1.2. Costos directos totales

**Tabla 3-4:** Costos directos totales

Descripción	V. Total
Costos eléctricos	986
Costos mecánicos	443.68
<b>Costo directo total</b>	<b>1429.68</b>

**Realizado por:** Guallazaca González, Carlos Andrés, 2020

#### 4.1.3. Costos Indirectos

**Tabla 4-4:** Costos Indirectos

N <sup>a</sup>	Descripción	V. unitario (USD)	V. Total (USD)
2	Imprevistos	50	50
		Total	250

**Realizado por:** Guallazaca González, Carlos Andrés, 2020

#### 4.1.4. Costo Total

**Tabla 5-4:** Costos Totales

N <sup>a</sup>	Descripción	V. unitario (USD)	V. Total (USD)
1	Costos directos totales	1133,00	1429,68
2	Costos indirectos totales	200,00	200,00
		<b>Costo Total</b>	<b>2629,68</b>

**Realizado por:** Guallazaca González, Carlos Andrés, 2020

## CONCLUSIONES

- Del análisis del estado del arte referente a los sistemas actuales disponibles en la industria moderna, se determinó que para la clasificación de productos es factible utilizar las técnicas de machine learning.
- Se analizó los diferentes métodos y técnicas de inteligencia artificial, llegando a la conclusión de utilizar algoritmos de redes neuronales convolucionales para la clasificación de imágenes.
- Se establecieron los requerimientos de diseño de hardware y software para que el sistema de control de calidad de productos funcione de una manera adecuada para el sector agrícola.
- La metodología aplicada en este trabajo de titulación encaminó a que el prototipo cumpla con los parámetros establecidos que contribuyeron de manera óptima con los requerimientos del clasificador.
- El diseño del interfaz de usuario se implementó mediante JavaScript en una aplicación web y el procesamiento de la información en un microcontrolador ESP8266 mediante Iot que se enlazaron correctamente mediante el protocolo MQTT.
- Los resultados obtenidos permitieron determinar que durante el entrenamiento el producto mejor identificado fueron las naranjas, requiriendo solamente 41.67 imágenes para entrenar su respectivo modelo. Mientras que, por otro lado, el producto más difícil de identificar fue el plátano verde, que requirió 48.33 imágenes. Finalmente, los mejores modelos de identificación fueron validados mediante 5 pruebas y un nuevo conjunto de 9 imágenes por cada producto.
- Los resultados obtenidos determinaron que los modelos de identificación para los productos bananas y manzanas no presentaron errores en la validación, mientras que los modelos de identificación de los productos naranja y plátano verde presentaron errores de 4.44% y 6.67%, respectivamente.
- A través del estudio se concluye que el algoritmo KNN se ajustó a la identificación de formas y colores, con una adecuada etapa de entrenamiento supervisado, lo cual lo hace flexible para utilizarlo en aplicaciones en tiempo real.

## RECOMENDACIONES

- Se recomienda incorporar una etapa de preprocesamiento de imágenes en el sistema ya implementado, con el objetivo de evaluar productos con parámetros objetivos como: la clasificación según niveles de maduración, niveles de afectaciones por placa, o detectar enfermedades en las frutas.
- Para una mejor versatilidad de la aplicación para diferentes productos se recomienda descargar el modelo corregido y supervisado desde la plataforma web – pestaña entrenamiento, cuando el sistema esté funcionando correctamente. Este archivo modelo.js guardar en carpetas etiquetadas según el producto sin alterar el nombre original del archivo, esto ayudará a cargar lo aprendido al sistema para no recurrir a un reentrenamiento y minimizar los niveles de errores.
- Se recomienda evitar abrir el archivo “modelo.js” directamente por qué puede generar lentitud dependiendo de las características del computador colapsando el sistema. Su utilización se lo hace desde el apartado cargar modelo y descargar modelo.
- Para mejorar la velocidad de clasificación de los productos, se debe utilizar una conexión a internet con una velocidad superior a los 10 Mbps para el modo Iot, o también se la puede realizar mediante una conexión de cable directo desde el computador hasta el controlador ESP que coordina los actuadores.

## **GLOSARIO**

**Inteligencia Artificial:** La inteligencia artificial es un conjunto de algoritmos que hacen posible que las máquinas aprendan de la experiencia, se ajusten a las nuevas entradas de datos y realicen tareas similares a las de los humanos, tomando como referencia un criterio analítico.

**Redes Neuronales Artificiales:** Las redes neuronales artificiales, es una tecnología basada en estudios del cerebro y del sistema nervioso biológico. Estas redes emulan una red neuronal real utilizando un conjunto reducido de expresiones matemáticas.

**Tensorflow:** Plataforma de código abierto para el aprendizaje automático. Cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos comunitarios que permite a los investigadores impulsar el estado del arte en machine learning y a desarrolladores a implementar aplicaciones de machine learning.

**Industria 4.0:** Es la nueva fase en la revolución industrial que se ha descrito a lo largo del tiempo en la transformación de técnicas de un proceso productivo industrial. El concepto de industria 4.0 conceptualiza a la interconectividad, la automatización, el aprendizaje automático y el reporte de datos en tiempo real. Industria 4.0 es el equivalente a fabricación inteligente.



## BIBLIOGRAFÍA

**AIZENBERG, I., PALIY, D.V., ZURADA, J.M. y ASTOLA, J.T.**, Blur Identification by Multilayer Neural Network Based on Multivalued Neurons. *IEEE Transactions on Neural Networks*, 2008. vol. 19, no. 5, pp. 883-898. ISSN 1045-9227, 1941-0093. DOI 10.1109/TNN.2007.914158.

**ARDILA, R.**, INTELIGENCIA. ¿QUÉ SABEMOS Y QUÉ NOS FALTA POR INVESTIGAR? *Revista de la Academia Colombiana de Ciencias Exactas, Físicas y Naturales*, 2011. vol. 35, no. 134, pp. 97-103. ISSN 0370-3908.

**CARRILLO, S.**, ¿Qué son las PYMES? *Grupo Enroke* [en línea]. [Consulta: 2 septiembre 2019]. 2019. Disponible en: <http://grupoenroke.com/que-son-las-pymes/%20>.

Clasificación utilizando vecinos más cercanos - MATLAB & Simulink - MathWorks América Latina. [en línea], [2017]. [Consulta: 5 febrero 2020]. Disponible en: <https://la.mathworks.com/help/stats/classification-using-nearest-neighbors.html>.

COBOS, F.S., 2017. *Proyectos de productos editoriales multimedia. ARGN0110*. S.l.: IC Editorial. ISBN 978-84-17224-53-0.

**DZITAC, I., FILIP, F.G. y MANOLESCU, M.-J.**, Fuzzy Logic Is Not Fuzzy: World-renowned Computer Scientist Lotfi A. Zadeh. *International Journal of Computers Communications & Control*, 2017. vol. 12, no. 6, pp. 748. ISSN 1841-9836, 1841-9836. DOI 10.15837/ijccc.2017.6.3111.

**FRANKLIN, S. y GRAESSER, A.**, Is It an agent, or just a programación: A taxonomy for autonomous agents. En: J.P. MÜLLER, M.J. WOOLDRIDGE y N.R. JENNINGS (eds.), *Intelligent Agents III Agent Theories, Architectures, and Languages*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997. pp. 21-35. ISBN 978-3-540-68057-4.

**GARBAY, M., JOERGER, G., HUANG, A., SALMON, R., KIM, J., SHERMAN, V., DUNKIN, B. y BASS, B.**, An intelligent hospital operating room to improve patient health care. *Journal of Computational Surgery*, 2015. vol. 2, no. 1, pp. 3. ISSN 2194-3990. DOI 10.1186/s40244-015-0016-7.

**GUPTA, P. y KULKARNI, N.,** An Introduction of Soft Computing Approach over Hard Computing. *International Journal of Latest Trends in Engineering and Technology*, 2013. vol. 3, no. 1, pp. 5.

**HAGERTY, J.R.,** Lotfi Zadeh Tried to Adapt Math and Computers to a World of Ambiguity. *Wall Street Journal* [en línea]. 22 septiembre 2017. [Consulta: 31 enero 2020]. ISSN 0099-9660. 2017. Disponible en: <https://www.wsj.com/articles/lotfi-zadeh-tried-to-adapt-math-and-computers-to-a-world-of-ambiguity-1506088800>.

**JÁCOME ESTRELLA, H. y KING, K.,** *Estudios industriales de la micro, pequeña y mediana empresa*. 2013. 1. ed. Quito, Ecuador: FLACSO Ecuador. ISBN 978-9978-67-396-6.

**JAIN, A.K. y ZONGKER, D.,** Representation and recognition of handwritten digits using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997. vol. 19, no. 12, pp. 1386-1390. ISSN 01628828. DOI 10.1109/34.643899.

**Keras | TensorFlow Core. TensorFlow** [en línea], 2019. [Consulta: 17 mayo 2020]. Disponible en: <https://www.tensorflow.org/guide/keras?hl=es>.

**MEHTA, D., ZHAO, X., BERNAL, E.A. y WALES, D.J.,** Loss surface of XOR artificial neural networks. 2018. 97, no. 5, pp. 052307. ISSN 2470-0045, 2470-0053. DOI 10.1103/PhysRevE.97.052307.

**MORALES, G.** El heroe Alan Turing: Aportaciones de la Criptología a la victoria aliada en la Segunda Guerra Mundial.,2013, pp. 11.

MQTT. [en línea], 2014. [Consulta: 6 abril 2020]. Disponible en: <http://mqtt.org/>.

**MQTT - modern data transmission for Industry 4.0. KEB** [en línea], 2018. [Consulta: 17 mayo 2020]. Disponible en: <https://www.kebamerica.com/blog/mqtt-industry-4-0/>.

**OJEDA CH., L.R.,** Los algoritmos genéticos y el método de generación y prueba. *Ingeniería e Investigación*, 1998. no. 40, pp. 8.

**RUSSELL, S., NORVIG, P., CORCHADO RODRÍGUEZ, J.M. y JOYANES AGUILAR, L.,** *Inteligencia artificial: un enfoque moderno*. Madrid: Pearson Educación. ISBN 978-84-205-4003-0. 2011.

**SEMPLADES**, *Plan Nacional Buen Vivir 2013-2017* [en línea]. 2017. S.l.: s.n. Disponible en: [https://www.unicef.org/ecuador/Plan\\_Nacional\\_Buen\\_Vivir\\_2013-2017.pdf](https://www.unicef.org/ecuador/Plan_Nacional_Buen_Vivir_2013-2017.pdf).

**WINSTON, P.H.**, Artificial intelligence: a perspective. [en línea]. 1990. S.l.: s.n., pp. 11. Disponible en: <https://www.semanticscholar.org/paper/Artificial-intelligence%3A-a-perspective-Winston/75d03d764d147b751de9fb835fb5935aa08e0013>.

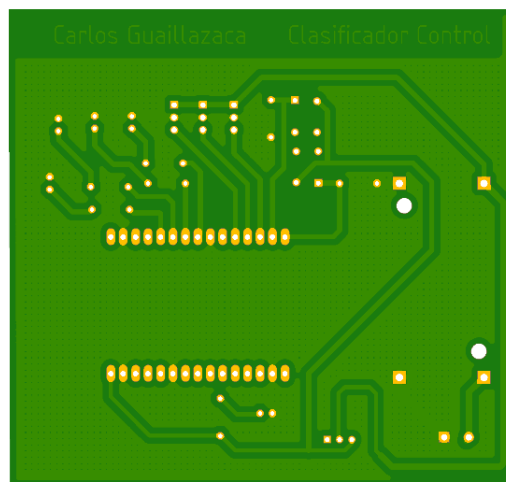
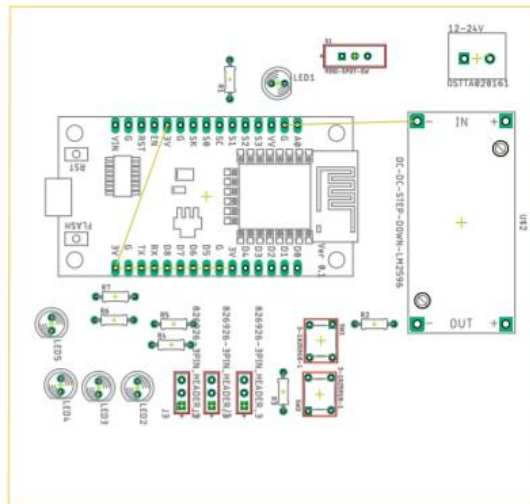
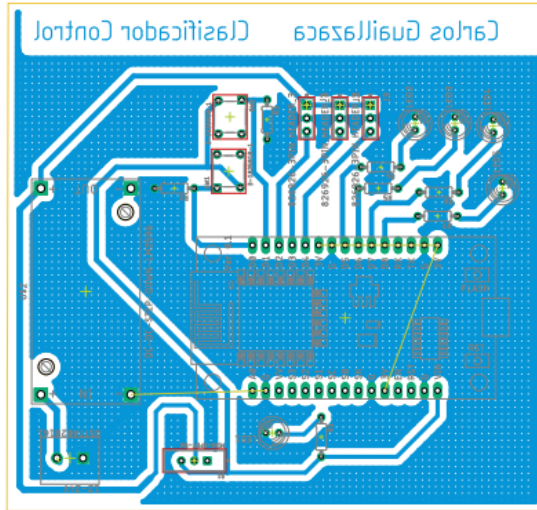
**WOOLDRIDGE, M. y JENNINGS, N.R.**, Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 1995, vol. 10, no. 2, pp. 115-152. ISSN 0269-8889, 1469-8005. DOI 10.1017/S0269888900008122.

**YOKOTANI, T. y SASAKI, Y.**, Comparison with HTTP and MQTT on required network resources for IoT. *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)* [en línea]. Bandung, Indonesia: IEEE, 2016. pp. 1-6. [Consulta: 16 mayo 2020]. ISBN 978-1-5090-0744-8. DOI 10.1109/ICCEREC.2016.7814989. Disponible en: <http://ieeexplore.ieee.org/document/7814989/>.

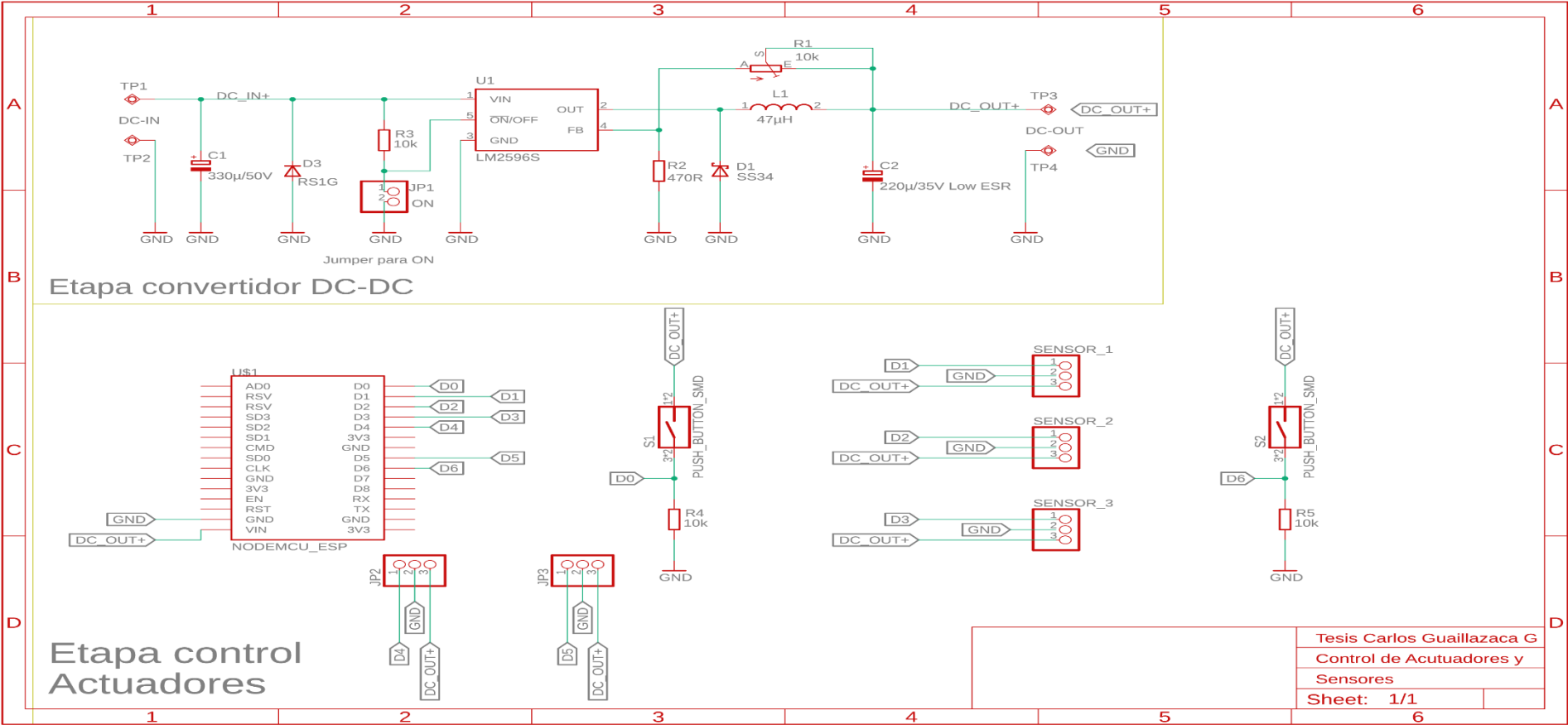
**ZADEH, L.A.**, Fuzzy sets. *Information and Control*, 1965. vol. 8, no. 3, pp. 338-353. ISSN 0019-9958. DOI 10.1016/S0019-9958(65)90241-X.

# ANEXOS

## ANEXO A: Boad PCB



# ANEXO B: Diseño Esquemático PCB

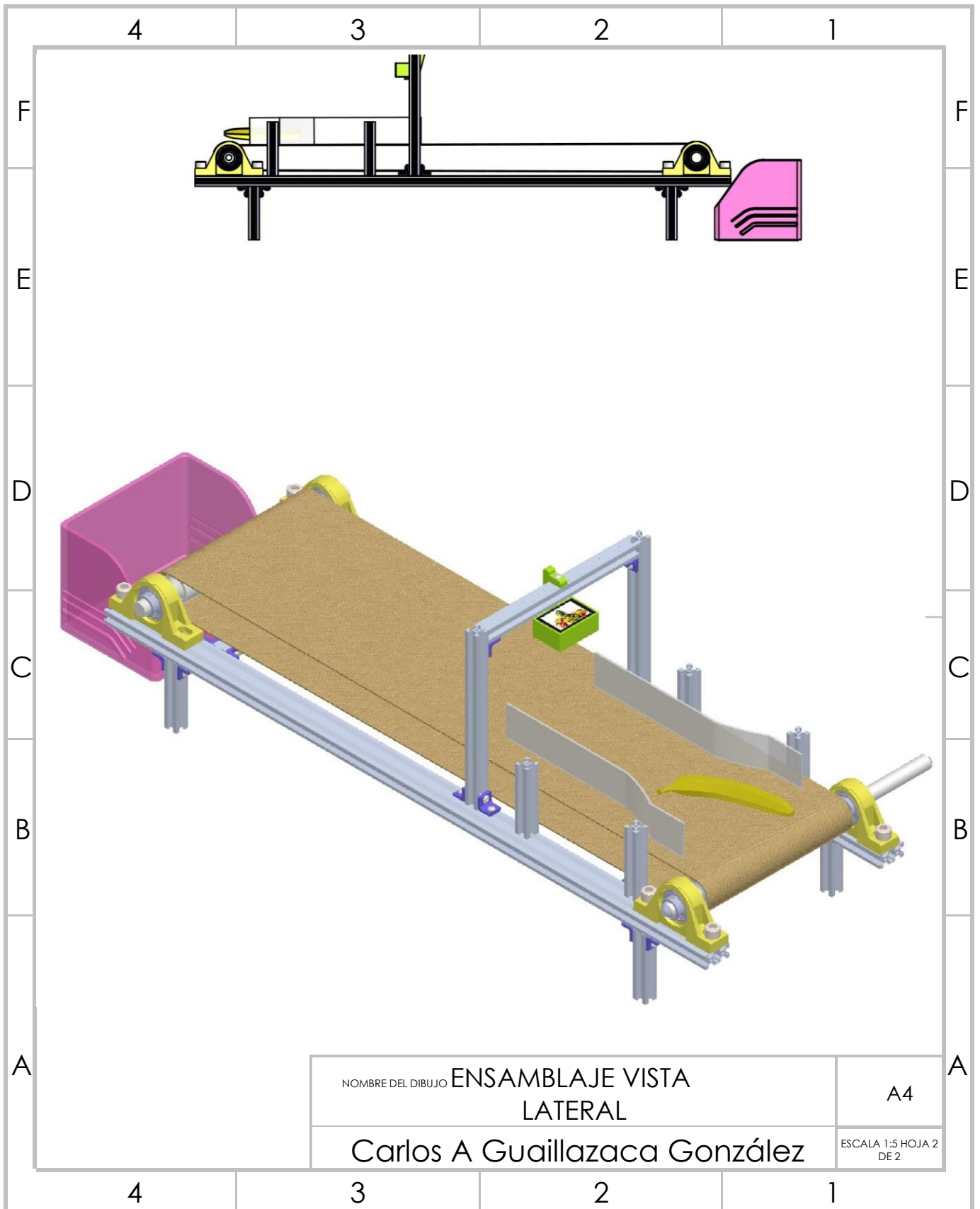


Tesis Carlos Guallazaca G  
 Control de Actuadores y  
 Sensores  
 Sheet: 1/1

## ANEXO C: CAD Partes del ensamblaje

4	3	2	1																																																						
F			F																																																						
E			E																																																						
D			D																																																						
C			C																																																						
B			B																																																						
A			A																																																						
<table border="1"> <thead> <tr> <th>N.º DE ELEMENTO</th> <th>N.º DE PIEZA</th> <th>CANTIDAD</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Vslot 20x40x1000</td> <td>2</td> </tr> <tr> <td>2</td> <td>Vslot 20x20x250</td> <td>8</td> </tr> <tr> <td>3</td> <td>AnguloSimple 20</td> <td>22</td> </tr> <tr> <td>4</td> <td>Chumacera 20mm</td> <td>4</td> </tr> <tr> <td>5</td> <td>rodillo02</td> <td>1</td> </tr> <tr> <td>6</td> <td>rodillo01</td> <td>1</td> </tr> <tr> <td>7</td> <td>Correa3-2^EnsamblajeTesis001</td> <td>1</td> </tr> <tr> <td>8</td> <td>B18.3.4M - 5 x 0.8 x 10 SBHCS --N</td> <td>44</td> </tr> <tr> <td>9</td> <td>B18.3.4M - 5 x 0.8 x 5 SBHCS --N</td> <td>1</td> </tr> <tr> <td>10</td> <td>Vslot 20x20x100</td> <td>5</td> </tr> <tr> <td>11</td> <td>Assem1</td> <td>1</td> </tr> <tr> <td>12</td> <td>Guias</td> <td>2</td> </tr> <tr> <td>13</td> <td>Box 002</td> <td>1</td> </tr> <tr> <td>14</td> <td>Box amarillo</td> <td>1</td> </tr> <tr> <td>15</td> <td>Box Centro</td> <td>1</td> </tr> <tr> <td>16</td> <td>banana</td> <td>1</td> </tr> <tr> <td>17</td> <td>HX-SHCS 0.5-13x0.5x0.5-N</td> <td>7</td> </tr> </tbody> </table>			N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD	1	Vslot 20x40x1000	2	2	Vslot 20x20x250	8	3	AnguloSimple 20	22	4	Chumacera 20mm	4	5	rodillo02	1	6	rodillo01	1	7	Correa3-2^EnsamblajeTesis001	1	8	B18.3.4M - 5 x 0.8 x 10 SBHCS --N	44	9	B18.3.4M - 5 x 0.8 x 5 SBHCS --N	1	10	Vslot 20x20x100	5	11	Assem1	1	12	Guias	2	13	Box 002	1	14	Box amarillo	1	15	Box Centro	1	16	banana	1	17	HX-SHCS 0.5-13x0.5x0.5-N	7	
N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD																																																							
1	Vslot 20x40x1000	2																																																							
2	Vslot 20x20x250	8																																																							
3	AnguloSimple 20	22																																																							
4	Chumacera 20mm	4																																																							
5	rodillo02	1																																																							
6	rodillo01	1																																																							
7	Correa3-2^EnsamblajeTesis001	1																																																							
8	B18.3.4M - 5 x 0.8 x 10 SBHCS --N	44																																																							
9	B18.3.4M - 5 x 0.8 x 5 SBHCS --N	1																																																							
10	Vslot 20x20x100	5																																																							
11	Assem1	1																																																							
12	Guias	2																																																							
13	Box 002	1																																																							
14	Box amarillo	1																																																							
15	Box Centro	1																																																							
16	banana	1																																																							
17	HX-SHCS 0.5-13x0.5x0.5-N	7																																																							
<table border="1"> <tr> <td style="text-align: center;">NOMBRE DEL DIBUJO</td> <td style="text-align: center;">ENSAMBLAJE</td> <td style="text-align: center;">A4</td> </tr> <tr> <td colspan="2" style="text-align: center;">Carlos A Guillazaca González</td> <td style="text-align: center;">ESCALA 1:5 HOJA 1 DE 2</td> </tr> </table>			NOMBRE DEL DIBUJO	ENSAMBLAJE	A4	Carlos A Guillazaca González		ESCALA 1:5 HOJA 1 DE 2																																																	
NOMBRE DEL DIBUJO	ENSAMBLAJE	A4																																																							
Carlos A Guillazaca González		ESCALA 1:5 HOJA 1 DE 2																																																							
4	3	2	1																																																						

ANEXO D: CAD Vista Lateral



NOMBRE DEL DIBUJO	ENSAMBLAJE VISTA LATERAL	A4
Carlos A Guallazaca González		ESCALA 1:5 HOJA 2 DE 2

## ANEXO E: Código del script que se integra al contenedor HTML

```
1  var Camara;
2  var BotonesEntrenar;
3  var knn;
4  var modelo;
5  var Texto;
6  var Clasificando = false;
7  var InputTextbox;
8  var BotonTextBox;
9
10 let BrokerMQTT = 'broker.shiftr.io';
11 let PuertoMQTT = 80;
12 let ClienteIDMQTT = "MQTT-P5";
13 let UsuarioMQTT = "carlosguallazaca";
14 let ContraseñaMQTT = "nomeacuerdo2402";
15
16 client = new Paho.MQTT.Client(BrokerMQTT, PuertoMQTT,
  * ClienteIDMQTT);
17
18 // set callback handlers
19 client.onConnectionLost = MQTTPerder;
20 client.onMessageArrived = MQTTMensaje;
21
22 client.connect({
23   onSuccess: CuandoConectadoMQTT,
24   userName: UsuarioMQTT,
25   password: ContraseñaMQTT
26 });
27
28 function MQTTPerder(responseObject) {
29   if (responseObject.errorCode !== 0) {
30     console.log("MQTT Perdio coneccion Error:" +
  *   responseObject.errorMessage);
31   }
32 }
33
34 function MQTTMensaje(message) {
35   console.log("Mensaje recibido:" + message.payloadString);
36 }
37
38 function CuandoConectadoMQTT() {
39   console.log("MQTT Conectado");
40 }
41
42 function setup() {
43
44   var canvas=createCanvas(320, 240);
45   var x=(windowWidth-width+200)/2;
46   var y=(windowHeight -height+300)/2;
47   canvas.position(x,y);
48   background(0, 0, 0);
```



```

49 Camara = createCapture(VIDEO);
50 Camara.size(320, 240);
51 Camara.hide();
52
53 modelo = ml5.featureExtractor('MobileNet', ModeloListo);
54 knn = ml5.KNNClassifier();
55
56 h1=createElement('h1','Presiona Botones para entrenar');
57 h1.position(40,280);
58 h1.style('color:blue');
59
60 var Boton1 = createButton("Producto Bueno");
61 Boton1.class("BotonEntrenar");
62 Boton1.position(150,350);
63
64 var Boton2 = createButton("Producto Regular");
65 Boton2.class("BotonEntrenar");
66 Boton2.position(150,400);
67
68 var Boton3 = createButton("Producto Malo");
69 Boton3.class("BotonEntrenar");
70 Boton3.position(150,450);
71
72 var BotonNada = createButton("Fondo");
73 BotonNada.class("BotonEntrenar");
74 BotonNada.position(180,500);
75
76 aviso0 = createP('Configura y entrena el Fondo')
77 aviso0.position(250,500);
78
79 aviso1=createElement('h1',"Entrena usando Nuevo Parametro")
80 aviso1.position(40,550);
81 aviso1.style('color:blue');
82
83 aviso2 = createP('Ingresa un nuevo parametro si es necesario')
84 aviso2.position(50,600);
85
86 aviso3 = createP('Escribe el nombre del nuevo producto en el
• cuadro de texto')
87 aviso3.position(50,620);
88
89 InputTextbox = createInput("Nuevo Parametro");
90 InputTextbox.position(60,670);
91
92 BotonTextBox = createButton("Entrenar con " + InputTextbox.value())
93 BotonTextBox.mousePressed(EntrenarTextBox);
94 BotonTextBox.position(250,670);
95
96 aviso4=createP("Generar o Cargar DataSet");
97 aviso4.position(125,720);

```

```

97     aviso5.position(400,720);
98
99     aviso5=createP("Es importante no cambiar el nombre del DataSet");
100     aviso5.position(80,820);
101     aviso5=createP("Modelo.js, Ordenar en carpetas segun sea el
    * caso");
102     aviso5.position(80,840);
103
104     var BotonGuardar = createButton("Guardar");
105     BotonGuardar.mousePressed(GuardarNeurona);
106     BotonGuardar.position(150,770);
107     var BotonCargar = createButton("Cargar");
108     BotonCargar.mousePressed(CargarNeurona);
109     BotonCargar.position(250,770);
110
111     Texto = createP("Modelo no Listo, esperando");
112     Texto.position(900,400);
113
114     BotonesEntrenar = selectAll(".BotonEntrenar");
115
116     for (var B = 0; B < BotonesEntrenar.length; B++) {
117         BotonesEntrenar[B].style("margin", "5px");
118         BotonesEntrenar[B].style("padding", "6px");
119         BotonesEntrenar[B].mousePressed(PresionandoBoton);
120     }
121 }
122
123 function PresionandoBoton() {
124     var NombreBoton = this.elt.innerHTML;
125     console.log("Entrenando con " + NombreBoton);
126     EntrenarKnn(NombreBoton);
127 }
128
129 function EntrenarKnn(ObjetoEntrenar) {
130     const Imagen = modelo.infer(Camara);
131     knn.addExample(Imagen, ObjetoEntrenar);
132 }
133
134 function ModeloListo() {
135     console.log("Modelo Listo");
136     Texto.html("Modelo Listo");
137     Texto.position(900,500);
138
139 }
140
141 function clasificar() {
142     const Imagen = modelo.infer(Camara);
143     knn.classify(Imagen, function(error, result) {
144         if (error) {
145             console.error();
146         }
147     });

```

```

146     } else {
147         Texto.html("Es un " + result.label); // aqui escribe el
        • resultado de lo q entreno
148         message = new Paho.MQTT.Message(result.label); // ponemos aqui
149         message.destinationName = "tesis/Clasificar";
150         client.send(message);
151         //clasificar();
152     }
153 })
154 }
155
156 function EntrenarTextBox() {
157     const Imagen = modelo.infer(Camara);
158     knn.addExample(Imagen, InputTextbox.value());
159 }
160
161 function GuardarNeurona() {
162     if (Clasificando) {
163         save(knn, "modelo.json");
164     }
165 }
166
167 function CargarNeurona() {
168     console.log("Cargando una Neurona");
169     knn.load("./modelo.json", function() {
170         console.log("Neurona Cargada knn");
171         Texto.html("Neurona cargada de archivo");
172     })
173 }
174
175 function draw() {
176     image(Camara, 0, 0, 320, 240);
177     BotonTextBox.html("Entrenar con " + InputTextbox.value());
178     if (knn.getNumLabels() > 0 && !Clasificando) {
179         //clasificar();
180         setInterval(clasificar, 500);
181         Clasificando = true;
182     }
183 }
184
185 // Temporary save code until m5 version 0.2.2
186 const save = (knn, name) => {
187     const dataset = knn.knnClassifier.getClassifierDataset();
188     if (knn.mapStringToIndex.length > 0) {
189         Object.keys(dataset).forEach(key => {
190             if (knn.mapStringToIndex[key]) {
191                 dataset[key].label = knn.mapStringToIndex[key];
192             }
193         });
194     }

```

```
195     const tensors = Object.keys(dataset).map(key => {
196         const t = dataset[key];
197         if (t) {
198             return t.dataSync();
199         }
200         return null;
201     });
202     let fileName = 'myKNN.json';
203     if (name) {
204         fileName = name.endsWith('.json') ? name : `${name}.json`;
205     }
206     saveFile(fileName, JSON.stringify({
207         dataset,
208         tensors
209     }));
210 };
211
212 const saveFile = (name, data) => {
213     const downloadElt = document.createElement('a');
214     const blob = new Blob([data], {
215         type: 'octet/stream'
216     });
217     const url = URL.createObjectURL(blob);
218     downloadElt.setAttribute('href', url);
219     downloadElt.setAttribute('download', name);
220     downloadElt.style.display = 'none';
221     document.body.appendChild(downloadElt);
222     downloadElt.click();
223     document.body.removeChild(downloadElt);
224     URL.revokeObjectURL(url);
225 };
226
```

## ANEXO F: Monitor Serial del ESP8266

The image displays the Arduino IDE interface for a project named 'MQTT\_3Actuadore' on an Arduino 1.8.7 board. The serial monitor is connected to COM3 and shows a continuous stream of data. The sketch code in the editor includes:

```
void setup() {
  Serial.begin(115200);
  pinMode(Actuador1, OUTPUT);
  pinMode(Actuador2, OUTPUT);
  pinMode(Actuador3, OUTPUT);
  digitalWrite(Actuador1, 1);
  Serial.println("Iniciando Wifi");
  WiFi.mode(WIFI_STA); //Cambiar modo del Wi-Fi
  delay(100);
  wifiMulti.addAP(ssid1, pass1);
  wifiMulti.addAP(ssid2, pass2);
  wifiMulti.addAP(ssid3, pass3);

  client.begin("broker.shiftr.io", net);
  client.onMessage(RecibirMQTT);

  Conectar();
}

void loop() {
  client.loop();
  delay(10);

  if (!client.connected()) {
    Conectar();
  }
}
```

The serial monitor output shows a sequence of messages: 'Paquete Recibido: /tesis/Clasificar - Producto Regular', 'Seleccion Producto Regular', and 'Paquete Recibido: /tesis/Clasificar - Producto Malo'. The output ends with 'Sin seleccion'. At the bottom of the IDE, a status bar indicates 'SSL ciphers (most compatible), 4MB (FS:2MB OTA~1019KB), v2, Lower Memory, Disabled, None, Only Sketch, 115200 en COM3'.

## ANEXO G: Código ESP8266

```
#ifndef ARDUINO_ARCH_ESP32
#include <WiFi.h>
#include <WiFiMulti.h>
WiFiMulti wifiMulti;
#else
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
ESP8266WiFiMulti wifiMulti;
#endif

#include <MQTT.h>

const char ssid1[] = "Red Krlos";
const char pass1[] = "Nomeacuerdo2402";
const char ssid2[] = "NETLIFE - MARLENITA";
const char pass2[] = "CBGM280271";
const char ssid3[] = "ssid"; // PARA AGREGAR OTRA RED SI SE DESEA
const char pass3[] = "pass";

WiFiClient net;
MQTTClient client;

unsigned long lastMillis = 0;
int Actuador1 = 5;
int Actuador2 = 16;
int Actuador3 = 4;

void Conectar() {
  Serial.print("Conectando a Wifi...");
  while (wifiMulti.run() != WL_CONNECTED) {
    Serial.print(".");
    delay(1000);
  }
  Serial.print("\nConectado a MQTT...");

  while (!client.connect("CONTROL_ACTUADORES", "carlosguillazaca",
"nomeacuerdo2402")) {
    Serial.print(".");
    delay(1000);
  }

  Serial.println("\nConectado MQTT!");
  Serial.println("\niNICIALIZANDO CLASIFICACION....");

  client.subscribe("/tesis/Clasificar");
}
```

```

void RecibirMQTT(String &topic, String &payload) {
  Serial.println("Paquete_Recibido: " + topic + " - " + payload);

  // Control para Actudor 1.....
  if (payload == "Producto Bueno") {
    Serial.println("Seleccion Producto Bueno");
    digitalWrite(Actuador1, HIGH);
    digitalWrite(Actuador2, LOW);
    digitalWrite(Actuador3, LOW);
  } else if (payload == "Fondo") {
    Serial.println("Sin seleccion");
    digitalWrite(Actuador1, LOW);
    digitalWrite(Actuador2, LOW);
    digitalWrite(Actuador3, LOW);
  }
}

// Control para Actudador 2.....
if (payload == "Producto Regular") {
  Serial.println("Seleccion Producto Regular");
  digitalWrite(Actuador2, HIGH);
  digitalWrite(Actuador1, LOW);
  digitalWrite(Actuador3, LOW);
} else if (payload == "Fondo") {
  Serial.println("Sin Seleccion");
  digitalWrite(Actuador1, LOW);
  digitalWrite(Actuador2, LOW);
  digitalWrite(Actuador3, LOW);
}
}

// Control para Actuador 3.....
if (payload == "Producto Malo") {
  Serial.println("Seleccion Producto Malo");
  digitalWrite(Actuador3, HIGH);
  digitalWrite(Actuador1, LOW);
  digitalWrite(Actuador2, LOW);
} else if (payload == "Fondo") {
  Serial.println("Sin Seleccion.");
  digitalWrite(Actuador1, LOW);
  digitalWrite(Actuador2, LOW);
  digitalWrite(Actuador3, LOW);
}
}
}

```

```
void setup() {
  Serial.begin(115200);
  pinMode(Actuador1, OUTPUT);
  pinMode(Actuador2, OUTPUT);
  pinMode(Actuador3, OUTPUT);
  digitalWrite(Actuador1, 1);
  Serial.println("Iniciando Wifi");
  WiFi.mode(WIFI_STA); //Cambiar modo del Wi-Fi
  delay(100);
  wifiMulti.addAP(ssid1, pass1);
  wifiMulti.addAP(ssid2, pass2);
  wifiMulti.addAP(ssid3, pass3);

  client.begin("broker.shiftr.io", net);
  client.onMessage(RecibirMQTT);

  Conectar();
}

void loop() {
  client.loop();
  delay(10);

  if (!client.connected()) {
    Conectar();
  }
}
```