



*Una universidad incluyente y comprometida con
el desarrollo integral*

DESARROLLO DE UN SISTEMA DE CONTROL DE RIEGO E IDENTIFICACIÓN DE MALEZAS MEDIANTE INTELIGENCIA ARTIFICIAL PARA CULTIVOS DE MAÍZ EN EL MUNICIPIO DE PAZ DE ARIPORO, CASANARE.

Autor

Leidy Tatiana Calderón Meza

Director

Marleny Fernández Sandoval

**INGENIERÍA ELECTRÓNICA
DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA SISTEMAS Y
TELECOMUNICACIONES
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
UNIVERSIDAD DE PAMPLONA
PAMPLONA
2021**



Autor: Leidy Tatiana Calderón Meza
Director: Marleny Fernández Sandoval

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA,
SISTEMAS Y TELECOMUNICACIONES
PROGRAMA DE INGENIERÍA ELECTRÓNICA
TRABAJO PRESENTADO PARA OPTAR POR EL TÍTULO DE
INGENIERO ELECTRÓNICO**

TEMA:

**DESARROLLO DE UN SISTEMA DE CONTROL DE RIEGO E
IDENTIFICACIÓN DE MALEZAS MEDIANTE INTELIGENCIA ARTIFICIAL
PARA CULTIVOS DE MAÍZ EN EL MUNICIPIO DE PAZ DE ARIPORO,
CASANARE.**

**FECHA DE INICIO DEL TRABAJO: 23 de marzo 2021
FECHA DE TERMINACIÓN DEL TRABAJO: 29 de noviembre 2021**

NOMBRES Y FIRMAS DE AUTORIZACIÓN PARA LA SUSTENTACIÓN:

LEIDY TATIANA CALDERÓN MEZA
AUTOR

MARLENY FERNANDEZ
SANDOVAL
DIRECTOR

YESID EUGENIO SANTAFE RAMÓN
DIRECTOR DE PROGRAMA

JURADO CALIFICADOR:

JULIO ERNESTO GÓMEZ

LUIS ALBERTO MUÑOZ BEDOYA

MARLENY FERNANDEZ SANDOVAL

**PAMPLONA NORTE DE SANTANDER
COLOMBIA
2021**



Autor: Leidy Tatiana Calderón Meza
Director: Marleny Fernández Sandoval

Para ustedes;
Yolanda Calderón y Mercedes Mesa
Eterna Gratitud



TABLA DE CONTENIDO

ÍNDICE DE FIGURAS	8
ÍNDICE DE TABLAS	9
AGRADECIMIENTO	10
RESUMEN	11
1. INTRODUCCIÓN	12
1.1 PROBLEMA	12
1.2 JUSTIFICACIÓN	13
1.3 DELIMITACIONES	14
1.3.1 OBJETIVO GENERAL	14
1.3.2 OBJETIVOS ESPECÍFICOS	14
1.4 ACOTACIONES	14
2. REFERENTES TEÓRICOS	1
2.1 ESTADO DEL ARTE	2
2.2 CARACTERIZACIÓN DEL CULTIVO	3
2.2.1 EL PROCESO DE CRECIMIENTO DE MAÍZ	4
2.2.2 MORFOLOGÍA DEL MAÍZ	5
2.2.3 CLIMAS ADECUADOS PARA EL CULTIVO	6
2.2.4 PREPARACIÓN DEL TERRENO	7
2.3 SISTEMAS DE RIEGO	7
2.3.1 POR GOTEO	7
2.4 TECNOLOGIAS DE COMUNICACIÓN Y ENVIO DE INFORMACIÓN 8	
2.4.1 ARDUINO	8
2.5 RASPBERRY PI	9
2.5.1 DESCARGA Y CONFIGURACIÓN INICIAL DEL SISTEMA OPERATIVO (SO)	10
2.5.2 INICIANDO LA RASPBERRY	11
2.5.3 ACCESO REMOTO A LA RASPBERRY PI DESDE WINDOWS 11	
2.5.4 ACTUALIZACIÓN Y DESCARGA DE PAQUETES NECESARIOS	12
2.6 CÁMARA USB	13
2.7 SENSORES DE HUMEDAD	14
2.7.1 SENSOR DE HUMEDAD DE SUELO ANTICORROSIVO-HIGRÓMETRO SNH0011	14



2.7.2 MODULO SNH0011	15
2.8 ELECTROVÁLVULA SOLENOIDE	15
2.9 SISTEMAS DE CONTROL	16
2.9.1 SISTEMAS DE CONTROL EN LAZO CERRADO	16
2.9.2 ACCIONES DE CONTROL	17
2.10 INTELIGENCIA ARTIFICIAL (IA)	17
2.10.1 NEURONA BIOLÓGICA	18
2.10.2 NEURONA ARTIFICIAL	18
2.10.3 REDES NEURONALES ARTIFICIALES (RNA).....	20
2.10.4 GOOGLE COLAB	25
2.10.5 PYTHON.....	25
3. METODOLOGÍA	26
3.1 RESUMEN DE CONTROL DE RIEGO POR GOTEO.....	27
3.1.1 CARACTERIZACIÓN DEL CULTIVO	28
3.1.2 ADQUISICIÓN	28
3.1.3 IDENTIFICACIÓN	29
3.1.4 CONTROLADOR.....	31
3.1.5 IMPLEMENTACIÓN	33
3.2 RESUMEN DE INTELIGENCIA ARTIFICIAL	35
3.2.1 ADQUISICIÓN DE LA DATA	36
3.2.2 PREPROCESAMIENTO	38
3.2.3 PROCESAMIENTO	39
3.2.4 CREACIÓN DE LA RED.....	39
3.2.5 ENTRENAMIENTO.....	41
3.2.6 EXPORTACIÓN DEL MODELO A RPI4.....	41
4. RESULTADOS.....	42
4.1 SISTEMA DE CONTROL ON-OFF POR HISTERESIS	43
4.1.1 ADQUISICIÓN DE LA VARIABLE DE TEMPERATURA	43
4.1.2 DISEÑO DE CIRCUITO DE ADQUISICIÓN	44
4.1.3 ADQUISICIÓN DE SEÑALES DEL ELEMENTO DE MEDIDA	45
4.1.4 IMPLEMENTACIÓN DEL SISTEMA DE RIEGO POR GOTEO....	47
4.2 SISTEMA DE IDENTIFICACIÓN DE MALEZA MEDIANTE IA.....	48
4.2.1 ADQUISICIÓN DE LA DATA	48
4.2.2 ENTRENAMIENTO DE LA RED NEURONAL.....	49



4.2.3 PREDICCIÓN DE IMÁGENES EN EL MODELO DE RPI4.....	51
4.2.4 PRUEBA DE NUEVAS IMÁGENES CON LA RED NEURONAL DESDE RPI4.	52
4.2.5 IDENTIFICACIÓN DE MALEZA POR ZONAS.....	54
4.3 VALIDACIÓN DEL SISTEMA POR EL EXPERTO.....	54
5. CONCLUSIONES	56
6. TRABAJOS FUTUROS	58
7. REFERENCIAS	59
8. ANEXOS	61



ÍNDICE DE FIGURAS

<i>Figura 1. Caracterización del cultivo de maíz. [7].....</i>	<i>4</i>
<i>Figura 2. Ciclo vegetativo del maíz. [8].....</i>	<i>4</i>
<i>Figura 3. Morfología del maíz. [8]</i>	<i>6</i>
<i>Figura 4. Cultivo con sistema de riego por goteo.[10]</i>	<i>7</i>
<i>Figura 5. Tarjeta Arduino UNO.[11]</i>	<i>8</i>
<i>Figura 6. Aspecto físico de tarjeta Raspberry Pi4.[12]</i>	<i>9</i>
<i>Figura 7. Cámara web. [16].....</i>	<i>14</i>
<i>Figura 8. sensor de humedad.[17].....</i>	<i>15</i>
<i>Figura 9. Convertidor del sensor de humedad. [18]</i>	<i>15</i>
<i>Figura 10. Componentes de una válvula solenoide. [19].....</i>	<i>16</i>
<i>Figura 11. Sistema de control en lazo cerrado. [20].....</i>	<i>17</i>
<i>Figura 12. Descripción gráfica control ON/OFF con histéresis.[21].....</i>	<i>17</i>
<i>Figura 13. Partes de la neurona biológica.[22]</i>	<i>18</i>
<i>Figura 14. Modelo neuronal de McCulloch-Pitts.[22].....</i>	<i>19</i>
<i>Figura 15. Topología básica de una RNA.[22]</i>	<i>20</i>
<i>Figura 16. Grafica de la función de activación RELU. [24].....</i>	<i>24</i>
<i>Figura 17. Diagrama en bloques de sistema de control de riego. [Autor]</i>	<i>27</i>
<i>Figura 18. Modelado de función de transferencia. [Autor].....</i>	<i>30</i>
<i>Figura 19. adquisición de la función de transferencia. [Autor]</i>	<i>30</i>
<i>Figura 20. Adquisición de la función de transferencia. [Autor].....</i>	<i>31</i>
<i>Figura 21. Adquisición de datos de humedad del sensor 1. [Autor].....</i>	<i>32</i>
<i>Figura 22. Desactivación de electroválvula 1. [Autor].....</i>	<i>32</i>
<i>Figura 23. Adquisición de datos de humedad del sensor 2. [Autor].....</i>	<i>33</i>
<i>Figura 24. Desactivación de electroválvula 2. [Autor].....</i>	<i>33</i>
<i>Figura 25. Circuito de implementación del sistema de control por histéresis. [Autor]. .</i>	<i>34</i>
<i>Figura 26. Diagrama lógico de programación del sistema de control. [Autor].....</i>	<i>34</i>
<i>Figura 27. Diagrama en bloques de identificación de malezas. [autor].....</i>	<i>35</i>
<i>Figura 28. Conjunto de imágenes de entrenamiento. [Autor]</i>	<i>37</i>
<i>Figura 29. conjunto de imágenes de validación. [Autor].....</i>	<i>37</i>
<i>Figura 30. muestras de las clases de imágenes con las que se entrena el modelo. [Autor].....</i>	<i>38</i>
<i>Figura 31. carga de datos para el entrenamiento y validación de la red. [Autor]</i>	<i>39</i>
<i>Figura 32. Estructura interna del entrenamiento de la red neuronal. [Autor].....</i>	<i>40</i>
<i>Figura 33. Diagrama en bloques del sistema de control de riego. [Autor]</i>	<i>43</i>
<i>Figura 34. Datos de temperatura ambiente en el cultivo. [Autor]</i>	<i>44</i>
<i>Figura 35. Diseño de PCB para el sistema de control. [Autor].....</i>	<i>44</i>
<i>Figura 36. Circuito del sistema de control de riego en pcb. [Autor]</i>	<i>45</i>
<i>Figura 37. Sistema de adquisición de señales [Autor].....</i>	<i>46</i>
<i>Figura 38. Adquisición de datos del sensor. [Autor].....</i>	<i>46</i>
<i>Figura 39. Registro de humedad de la zona del cultivo. [Autor]</i>	<i>47</i>



Figura 40. Perímetro de riego. [Autor]	47
Figura 41. Sistema de riego por goteo. [Autor]	48
Figura 42. Predicción de la data. [Autor]	49
Figura 43. Numero de épocas del entrenamiento de la red neuronal. [Autor]	50
Figura 44. Precisión de las épocas del entrenamiento de la red. [Autor]	50
Figura 45. Perdidas de las épocas del entrenamiento de la red. [Autor]	51
Figura 46. Comparación del modelo con nuevas imágenes. [Autor]	51
Figura 47. Comparación de imagen con maleza en rpi4. [Autor]	52
Figura 48. Comparación de imagen con maleza en rpi4. [Autor]	52
Figura 49. Comparación de imagen sin maleza en rpi4. [Autor]	53
Figura 50. Comparación de imagen sin maleza en rpi4. [Autor]	53
Figura 51. Toma de video por zonas. [Autor]	54
Figura 52. Comprobante de validación por experto. [Autor]	55
Figura 53. Código de visualización de gráficas de caracterización del sensor. [Autor] ..	63

ÍNDICE DE TABLAS

Tabla 1. Porcentaje de humedad con respecto a la etapa del cultivo.[9]	6
Tabla 2. Características básicas de Arduino UNO.[11]	9
Tabla 3. Características técnicas del modelo B de Raspberry Pi4.[12]	9
Tabla 4. Características técnicas de cámara USB. [16]	13
Tabla 5. Costos y presupuestos para sistema de control de riego. [Autor]	63
Tabla 6. Costos y presupuestos de sistema de identificación de malezas. [Autor]	64

INDICE DE ECUACIONES

Ecuación 1. Función de transferencia de un sistema en lazo cerrado [20]	16
Ecuación 2. Función de salida del perceptrón simple. [22]	19
Ecuación 3. Función de activación en la neurona artificial.[22]	19
Ecuación 4. Representación matemática de la función de activación softmax. [25]	23
Ecuación 5. Representación matemática de la función RELU. [24]	23
Ecuación 6. Función de transferencia del sistema de sensor 1. [Autor]	30
Ecuación 7. Función de transferencia del sistema de sensor 2. [Autor]	31



AGRADECIMIENTO

Primeramente, le doy gracias a Dios por haberme permitido culminar esta meta, ayudarme y darme las fuerzas en todo momento, a pesar de las dificultades, por darme la sabiduría, el conocimiento e inteligencia para sacar este proyecto adelante. También, quiero agradecer a mis abuelos Mercedes Mesa, Jorge Calderón y mi madre Yolanda Calderón principalmente por el esfuerzo que hicieron para hacer posible este sueño, además de apoyarme, despertar en mí el espíritu de superación y ser la fuente de inspiración para salir adelante. A mis tíos Javier Calderón, Gladys Calderón y Luz Esther Calderón por la colaboración, consideraciones y consejos. Así mismo, a mis amigos y compañeros que estuvieron conmigo en este proceso y que están conmigo en las buenas y malas, en especial a Oscar Mogotocoro y Laura Macías por brindarme una mano amiga, el apoyo moral, psicológico e incondicional y por motivarme a seguir luchando por mis sueños. Finalmente, a la ingeniera Marleny Fernández Sandoval quien me instruye, confió en mis capacidades y me dio la oportunidad de llevar a cabo el proyecto bajo su dirección, como también a cada uno de los docentes que aportaron en mi formación profesional.



RESUMEN

Los cambios climáticos han provocado desequilibrio con la naturaleza, lo cual ha afectado también la agricultura en zonas como la región Orinoquia que últimamente ha presentado dificultades para la plantación de cultivos, por tanto, se han impulsado estudios que ayuden a tratar estos impactos mediante inteligencia artificial, logrando soluciones que permiten la adaptación de las plantas. El propósito principal de este proyecto de investigación es desarrollar una solución mediante la implementación de un sistema de control de riego e identificación de malezas en cultivos de maíz de bajo costo, eficiente y fácil manejo que permita caracterizar la variable de humedad en la planta para conocer el rango de humedad que debe mantener el suelo y de este modo, abastecer la necesidad hídrica, además de monitorear el cultivo para prevenir al agricultor y tomar medidas como la fumigación a tiempo y que el cultivo se desarrolle sin inconvenientes, creando una base de datos que contiene imágenes de dos clases para realizar el entrenamiento de una red neuronal y luego validarla, la cual se encargara de identificar el momento en el que el cultivo presenta malezas e indicarle al agricultor. El sistema consta de un prototipo basado en etapas para el cumplimiento de las tareas que debe realizar, en este caso para el control de riego por goteo se tiene en cuenta cinco etapas en las que se caracteriza el cultivo, se adquieren los datos, se sintoniza la planta, se realiza la simulación y finalmente se prueba con la implementación; así mismo, para la identificación de malezas, básicamente consta de seis etapas, la adquisición de las imágenes, el preprocesamiento, el procesamiento, la creación de la red, entrenamiento y la exportación de la red neuronal.



Capítulo 1.

1. INTRODUCCIÓN

1.1 PROBLEMA

Durante los últimos quince años, las temperaturas han ido cambiando en el país de manera paulatina. Según Jeimar Tapasco, investigador del Centro Internacional de Agricultura Tropical (Ciat), para el 2030 se van a dar cambios que no van a ser buenos para cultivos en ciertas zonas del país, entre los cultivos perjudicados se encuentra el maíz [1]. Los cultivos de maíz en Colombia la mayoría han sido productivos, pero actualmente presentan inconvenientes en diferentes factores como escasez de agua, malezas, plagas, aumento de los costos de producción, entre otros, que impiden una cosecha totalmente sana, también dependen de las condiciones climáticas y el ambiente en el que se siembra, ya que la diversidad geográfica en el país implica realizar estudios del suelo donde se va a plantar la siembra, para adquirir mejores resultados en la producción y menos pérdidas económicas, postcosecha y de tiempo.



1.2 JUSTIFICACIÓN

El cultivo de maíz, aunque sea el tercero con mayor superficie de siembra en Colombia, de acuerdo a los datos más recientes, la producción cayó 1.6 mts en 2016 y se espera que desde 2018 hasta 2030 la producción crezca alrededor de 6% y 9% la demanda. Teniendo en cuenta que la dimensión social es importante para millones de colombianos, aportando el 9% del suministro diario de sus derivaciones [2]. Además, entre los departamentos con mayor producción de maíz en el país no se encuentra Casanare, debido a la falta de aplicación de tecnologías y de agricultura de precisión en los cultivos, es necesario que exista un sistema de control que permita el monitoreo del cultivo en tiempo real, lo cual es posible utilizando herramientas de inteligencia artificial para acceder de forma remota a los datos y analizarlos, además mediante el sistema tecnológico poder controlar variables como el flujo de agua, la temperatura, humedad y malezas presentes en el cultivo, haciendo uso de sensores e instrumentos que mejor se adapten a las condiciones del cultivo y también se realice los correspondientes análisis de la adquisición de señales en el software Python, de modo que tenga la capacidad de prevenir y minimizar las pérdidas en los cultivos; así mismo pueda servir como herramienta de apoyo para los agricultores al implementarse en diferentes cultivos.



1.3 DELIMITACIONES

1.3.1 OBJETIVO GENERAL

- ✓ Desarrollar un sistema de control de riego e identificar malezas mediante inteligencia artificial para cultivos de maíz en el municipio de Paz de Ariporo, Casanare.

1.3.2 OBJETIVOS ESPECÍFICOS

- ✓ Caracterizar las variables de temperatura y humedad junto con los elementos de sensorica idóneos para el sistema de control.
- ✓ Diseñar un lazo de control de acuerdo a las características del cultivo y las necesidades de respuesta del mismo.
- ✓ Aplicar técnicas de inteligencia artificial para la captación y procesamiento de imágenes del cultivo para la identificación de malezas.
- ✓ Diseñar e implementar el sistema de adquisición de señales e imágenes y ensamble físico del prototipo tecnológico integrado con el sistema de control e identificación de maleza.
- ✓ Validar el sistema mediante la producción y calidad del cultivo con apoyo de un experto.

1.4 ACOTACIONES

Teniendo en cuenta que para el análisis y la simulación de control se requiere otra herramienta computacional a parte del software Python, se utilizará Matlab como herramienta de apoyo. Además, de otras posibles herramientas para el análisis de control para comparar el rendimiento, así mismo, para que sea eficiente y de bajo costo se utilizará el sistema sensorial que mejor se adapte y a partir de esto realizar la adquisición mediante el sistema embebido Rapsberry pi4; también se implementará un sistema de identificación de maleza mediante el desplazamiento en riel de una cámara USB para realizar un mapeo, identificación y prueba en el cultivo de maíz a través de la aplicación de técnicas de inteligencia artificial como las redes neuronales; de modo que se seleccionará la más adecuada. En caso de presentarse dificultades con respecto a la tecnología de captación de imagen se considerarán otras opciones como una cámara RGB o cámaras que permitan realizar esta acción. Por otra parte, el sitio de aplicación donde se desarrollará el proyecto es en la región Orinoquia en el municipio de paz de Ariporo, Casanare, en la vereda jagüito finca San Nicolas.



Capítulo 2.

2. REFERENTES TEÓRICOS

En este capítulo se dará a conocer trabajos del estado del arte como referencia a la temática principal de esta tesis, así como también se describirán las bases teóricas necesarias para comprender conceptos de control en cultivos de maíz y los modelos de inteligencia artificial que se implementarán en el proyecto.



2.1 ESTADO DEL ARTE

En diferentes estudios que se han realizado, aplicar un tratamiento selectivo para controlar las malas hierbas en un cultivo de maíz favorece la competitividad, ya que, en las etapas iniciales del desarrollo del maíz aparece la maleza que perjudican el cultivo, con el fin de evitar y eliminar infestaciones, el grupo de investigación ISCAR de Madrid, España desarrolló a través del proyecto **“SISTEMA DE VISIÓN PARA LA AGRICULTURA DE PRECISIÓN: IDENTIFICACIÓN EN TIEMPO REAL DE LÍNEAS DE CULTIVO Y MALAS HIERBAS EN CAMPOS DE MAÍZ”** donde la utilización de sistemas de visión artificial facilitan las tareas y son capaces de identificar en tiempo real las líneas de cultivo de maíz, junto con un control preciso de malas hierbas, además de controlar el guiado del tractor y solapamiento de las zonas tratadas[1]. Por otra parte, mediante el proyecto **“REDES NEURONALES APLICADAS AL CONTROL DE RIEGO USANDO INSTRUMENTACIÓN Y ANÁLISIS DE IMÁGENES PARA UN MICROINVERNADERO APLICADO AL CULTIVO DE ALBAHACA”**, se diseñó e implementó un control inteligente con redes neuronales multicapa de retro-propagación en un microinvernadero de plantas de albahaca, de modo que, mantiene un riego eficiente sin realizar cambios manualmente, además, según el estado de humedad de la planta, suministra la cantidad de agua. También, mediante el procesamiento de imagen conoce el estado actual de la planta de forma visual y así mismo, guarda el historial de las lecturas de datos de los sensores de temperatura y humedad en una tabla de Excel que al mismo tiempo se visualiza en la interfaz. [2].

El recurso hídrico no es inagotable, por tanto, es necesario mitigar el desperdicio con sistemas de riego que proporcionen la cantidad de agua que se debe utilizar en los cultivos para su óptimo desarrollo, de modo que en la hacienda Tarapacá de la ciudad de Cali, se desarrolló mediante el proyecto **“DISEÑO DE UN SISTEMA DE RIEGO EN LA GRANJA TARAPACÁ UBICADA EN LA CIUDAD DE CALI”** un sistema de riego por goteo donde se ubican los módulos de riego y tuberías, también mediante el software Epanet se simuló el sistema de riego, se determinó las velocidades, presión y caudales requeridos. Esto con el fin de no recurrir a métodos tradicionales o artesanales del riego y para el correcto funcionamiento del sistema y goteros [3]. En las instalaciones de la Universidad Católica de Colombia, se llevó a cabo el proyecto **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN DE MALEZAS EN CULTIVOS CUNDIBOYACENSES”** el cual diseñó un sistema automático orientado hacia la agricultura de precisión que ayuda a la detección de agentes invasores basados en técnicas de visión artificial, haciendo capturas en diferentes zonas de la región cundiboyacense para determinar si el cultivo es óptimo o si existe la presencia de malezas para diferenciarlo con el cultivo, la cual se procesó mediante el software Python-OpenCv y Matlab, además de la utilización de clasificadores de datos como máquinas de soporte vectorial, redes neuronales y bayesiano y procesamiento digital de imágenes. [4].



Mediante el proyecto **“TECNIFICACIÓN PARA EL CONTROL DE PRESIÓN Y EMULACIÓN DE RIEGO USANDO EL CONTROL PLC Y SOFTWARE AIT”** donde con sistemas automatizados aplicados en cultivos se puede controlar variables como el riego para mantener una siembra de calidad, este proceso se lleva a cabo mediante un controlador lógico programable (PLC) utilizando el software TIA y controlado por variadores de frecuencia, además de llevar a cabo los métodos de Ziegler y Nichols para ajustar los parámetros PID y determinar la mejor respuesta del sistema implementado en una planta piloto que consta de dos tanques de almacenamiento de líquidos, una motobomba para el traspaso del líquido, un transductor para la lectura de presión de fluido y finalmente poner en funcionamiento el sistema de riego [5]. La propuesta planteada en el proyecto **“USO DE DRONES PARA EL ANÁLISIS DE IMÁGENES MULTIESPECTRALES EN AGRICULTURA DE PRECISIÓN”** se desarrolló en los campos de Cundinamarca, tomando como producto de estudio las plantaciones de papa. El objetivo principal es demostrar que las nuevas tecnologías representan un beneficio en el mundo de la agricultura, como lo es el caso específico de la implementación de drones para el análisis de imágenes multiespectrales, por medio de estas imágenes es posible monitorear los cultivos en tiempo real, disminuyendo costos, esfuerzo entre otros. También es importante mencionar que el uso de estas herramientas permite disminuir el uso de agentes químicos ya que las plagas son identificadas a tiempo [6].

2.2 CARACTERIZACIÓN DEL CULTIVO

A raíz de los cambios climáticos se refleja la necesidad de crear e implementar estrategias para la sostenibilidad de la agricultura que se adapte a la variabilidad del clima y que por ende la producción sea beneficiada, teniendo en cuenta el comportamiento y la estructura de las plantas del cultivo de maíz, ya que de siembra a cosecha se toma 170 a 180 días, donde las exigencias climáticas son temperaturas de 25°C a 30°C y para la germinación de la semilla temperaturas entre 15 a 20°C, otro factor a considerar es abastecer las necesidades hídricas del cultivo debido a que al día exige 5mm de agua y dependiendo de la fase de crecimiento se hidrata, es decir, el maíz requiere buena humedad durante los primeros 100 a 120 días de desarrollo del cultivo. En cuanto al suelo se adapta mejor a los que tienen un PH de 6 o 7, profundos y ricos en materia orgánica. [7]



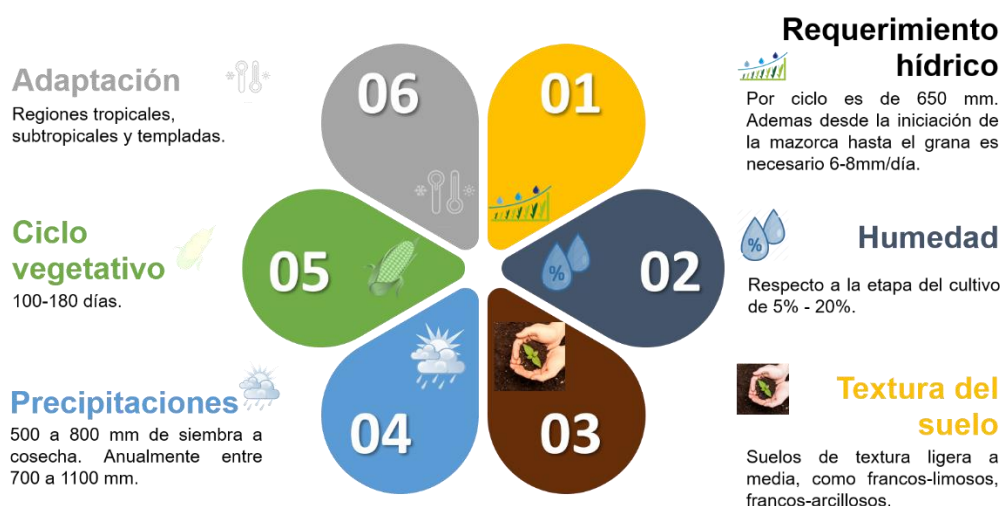


Figura 1. Caracterización del cultivo de maíz. [7]

2.2.1 EL PROCESO DE CRECIMIENTO DE MAÍZ

Las plantas de maíz tienen el mismo desarrollo que requiere ciertas unidades de crecimiento para alcanzar la maduración que son importantes para predecir cuándo ocurrirán cambios importantes, sin embargo, el tiempo de crecimiento se da en diferentes etapas dependiendo del tipo de maíz. Normalmente, la semilla de maíz germina entre los 5 a 7 días después de la siembra, donde su fisiología involucra dos grandes etapas: la vegetativa y reproductiva, las cuales, se pueden utilizar para ayudar a los productores a realizar aplicaciones oportunas de herbicidas y fungicidas. En la etapa vegetativa, comienzan las reacciones metabólicas dentro de la semilla con la humedad adecuada y la temperatura, la radícula comenzará a alargarse de la semilla del maíz. En la etapa reproductiva, se establece el número y tamaño de los granos, en esta etapa se debe tener en cuenta el estrés por humedad y calor porque puede causar reducción de rendimiento, pérdidas de mazorcas enteras o matar prematuramente la planta. [8]

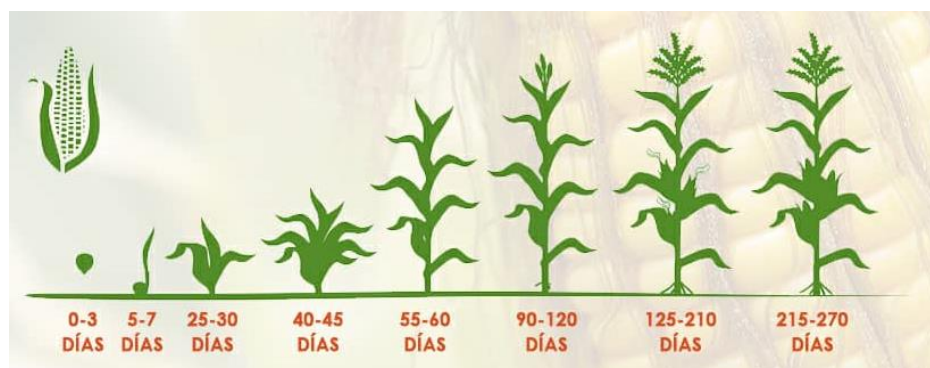


Figura 2. Ciclo vegetativo del maíz. [8]



2.2.2 MORFOLOGÍA DEL MAÍZ

La planta de maíz se conforma principalmente por:

- La raíz: presenta dos tipos: las primarias y fibrosas que van bajo tierra, y las adventicias que brotan del primer nudo de la planta y son superficiales. Ambas permiten que se mantenga erguido el largo tallo.
- El tallo: puede alcanzar hasta los 4 metros de altura y a su vez se compone de tres capas, una epidermis impermeable y transparente, una pared vegetal por la que circula la savia y una médula de tejido esponjoso y blanco en donde se almacenan los azúcares.
- Hojas: Generalmente lanceoladas, largas y finas, presentan vellosidades con extremos filosos y cortantes, alcanzan los 120 centímetros de longitud y los 9 centímetros de ancho.
- Fruto: Cada grano se denomina cariósipide y está constituido por el pericarpio que está fundido con la testa de la semilla formando la pared del fruto, el cual, es el 5 a 6 % de peso total del grano, la aleurona en torno al 2 o 3 %, el embrión alrededor del 12-13%, y el endospermo, mayoritario, presenta unos valores en torno al 80-85%. El resto lo constituye la piloriza que es una pequeña estructura cónica encargada junto con el pedicelo de unir el grano a la espiga.
- Inflorescencia: Es monoica, es decir, con inflorescencia femenina y masculina dentro de la misma planta. En cuanto a la masculina consiste en un eje central y ramas laterales, en donde brotan florecillas que poseen tres estambres cada una, en donde se produce el polen necesario para fecundar a las hembras. En cambio, la inflorescencia femenina son espigas cilíndricas dotadas de flores en hileras paralelas, provistas de ovarios en los que el polen germina, produciéndose así las semillas o granos que, al estar maduros, adquieren una textura bulbosa y coloración uniforme, siendo comestibles.[8]



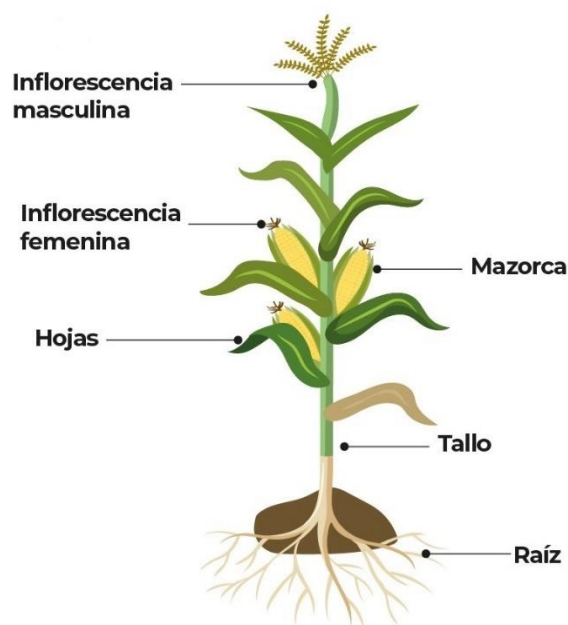


Figura 3. Morfología del maíz. [8]

2.2.3 CLIMAS ADECUADOS PARA EL CULTIVO

El maíz se cultiva mejor en climas cálidos, por lo que requiere una temperatura entre los 25 °C a 30 °C y bastante incidencia de luz solar. En climas húmedos o de temperaturas excesivas su rendimiento es más bajo ya que pueden tener problemas debido a la mala absorción de nutrientes, minerales y agua. Para la fructificación se requieren temperaturas de 20 a 32°C. [8]

Tabla 1. Porcentaje de humedad con respecto a la etapa del cultivo.[9]

Periodo	Semana	Lamina Agua [mm/día]	Días	Humedad [%]
Siembra	0	1	7	5
Brote	1-2	1.5	7	7,5
4-5 hojas	3-4	2	7	10
6-7 hojas	5-7	3	7	15
9-10 hojas	8-9	4	7	20
Floración	10-13	7	7	35



Llenado	14-18	5,5	7	27,5
Madurez	18-21	4	7	20
Secado	22-24	0	7	0

2.2.4 PREPARACIÓN DEL TERRENO

Para la preparación del terreno se recomienda un suelo arado para que el terreno quede suelto, esponjoso y tenga la capacidad de drenar el agua sin encharcamientos, además de que sea rico en materia orgánica; de lo contrario las semillas no germinarían adecuadamente. Aunque el maíz se adapta con facilidad a diferentes tipos de suelo, lo recomendable es mantener un pH de 6 0 7.[8]

2.3 SISTEMAS DE RIEGO

En los cultivos, una de las variables más importantes es el requerimiento hídrico, para proveer su punto ideal de humedad de la tierra, de modo que se observan así sistemas de riego por aspersión, localizado, inundación, entre otros.

2.3.1 POR GOTEO

También conocido como riego localizado, es una técnica de riego donde el agua se entrega a uno o varios puntos determinados, generalmente lo más cerca de las raíces de las plantas, teniendo como fin que el aprovechamiento sea máximo con las menores pérdidas, manteniendo una zona de terreno llamada bulbo en constante humedad. [10]



Figura 4. Cultivo con sistema de riego por goteo.[10]



2.4 TECNOLOGÍAS DE COMUNICACIÓN Y ENVÍO DE INFORMACIÓN

Para el montaje del prototipo se necesita de dispositivos electrónicos como sensores, sistemas embebidos como la Raspberry pi4, Arduino para la implementación. Además de una electroválvula para el riego y cámara USB para la detección de imágenes y video.

2.4.1 ARDUINO

Es una compañía de desarrollo de software y hardware libres, de código abierto que diseña y manufactura placas de desarrollo de hardware aplicado principalmente en herramientas de aprendizaje para el desarrollo de sistemas automáticos. La placa de Arduino se encuentra en diferentes encapsulados y los comúnmente utilizados son: arduino nano, arduino uno y arduino mega. Además, se caracterizan por: [11]

- El entorno de desarrollo gráfico se basa en lenguaje de programación processing/wiring,
- El hardware está compuesto por una placa con entradas y salidas, analógicas y digitales, contiene un microprocesador de la familia Atmega de bajo costo.
- La comunicación serial es a través de USB y drivers FTDI que proporcionan un puerto COM virtual con el software en el computador.
- Las Entradas analógicas del arduino son de 10 bits, por lo que entregan valores entre 0 y 1023. Con un rango de voltaje entre 0 y 5 voltios.

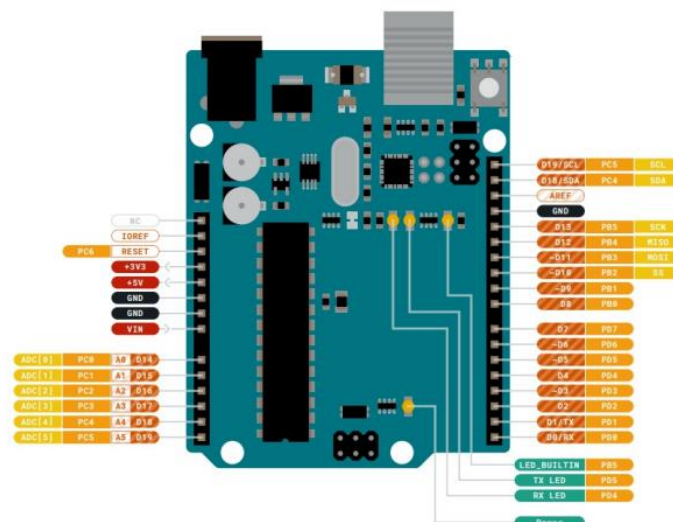


Figura 5. Tarjeta Arduino UNO.[11]



Tabla 2. Características básicas de Arduino UNO.[11]

Características	Descripción
Voltaje de operación	5v
Pines digitales	14,6 con salida PWM
Pines analógicos	6
Corriente DC	40mA
Corriente continua CC	3.3 V pin 50 mA
Memoria flash	32 KBytes (Atmega 328)
SRAM	2 KBytes (Atmega 328)
EEPROM	1 KBytes (Atmega 328)
Velocidad del reloj	16 MHz

2.5 RASPBERRY PI

En la Tabla 2. Se puede observar las características técnicas de la versión B de Raspberry Pi4.[12]

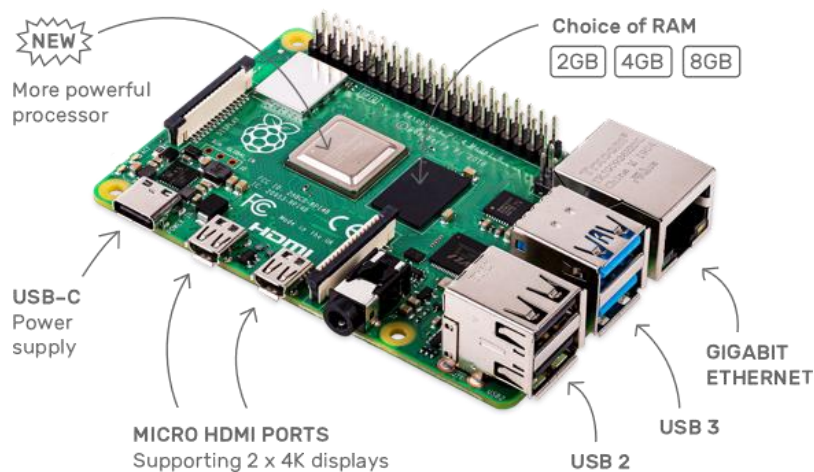


Figura 6. Aspecto físico de tarjeta Raspberry Pi4.[12]

Tabla 3. Características técnicas del modelo B de Raspberry Pi4.[12]

Características	Especificaciones
PROCESADOR	BROADCOM BCM2711B0, QUAD-CORE Cortex-A72
GPU	VIDEOCORE VI 500MHz
CPU	PROCESADOR DE CUATRO NÚCLEOS A 1,5 GHZ CON BRAZO CORTEX-A72



MEMORIA RAM	4 GB LPDDR4-3200
PUERTOS	2 X USB 3.0, 2 X USB 2.0
GPIO	40 PINES (SPI, I2C UART)
VIDEO Y SONIDO	2 x puertos micro-HDMI que admiten pantallas de 4K@60Hz a través de HDMI 2.0, puerto de pantalla MIPI DSI, puerto de cámara MIPI CSI, salida estéreo de 4 polos y puerto de vídeo compuesto.
ALMACENAMIENTO	MEMORIA SD (mínimo 4G)
ETHERNET	Gigabit Ethernet
ALIMENTACIÓN	5V/3A VÍA USB-C, 5V VÍA CABEZAL GPIO
SISTEMAS OPERATIVO	Raspberry Pi OS (Raspbian)

2.5.1 DESCARGA Y CONFIGURACIÓN INICIAL DEL SISTEMA OPERATIVO (SO)

Para el uso adecuado de la tarjeta Rapsberry pi4, se requieren de los siguientes accesorios:

- Una memoria SD (mínimo de 8GB, clase 4 0 10).
- Monitor con entrada HDMI o un televisor con entrada de video.
- Teclado y mouse USB.
- Cargador con entrada micro USB tipo C.
- Cable HDMI

Además, para la configuración del SO, se deben tener en cuenta los siguientes pasos:

1. Entrar a la página oficial de Rapsberry pi [12] y descargar la imagen del sistema operativo que queremos utilizar. Raspberry Pi OS (anteriormente llamado Raspbian) es ahora el sistema operativo oficialmente compatible para la mayoría de proyectos.
2. Antes de descomprimir y montar la imagen en la memoria micro SD, se procede a formatear
3. Descargado el archivo con extensión .zip, se procede a descomprimir y montar la imagen contenida allí en una memoria SD para conectarla a la Raspberry PI y ser ejecutada. Para ello, debemos insertar la memoria SD a un ordenador y con la ayuda del programa balenaEtcher [13], cargar la imagen en la memoria SD desde Windows.
4. Ejecutado el programa BALENAETCHER se direccionará la imagen .img del sistema operativo Raspbian, además, la unidad de destino de la tarjeta SD donde se va a cargar la imagen.



5. Una vez cargada la imagen en la memoria, la tarjeta SD esta lista para ser insertada en la Raspberry Pi.
6. Finalmente, se conectan los periféricos de entrada y salida como son: el mouse, teclado, pantalla HDMI, red Wifi y se habilita el periférico de la cámara USB.

2.5.2 INICIANDO LA RASPBERRY

Para dar inicio al entorno gráfico del sistema operativo Raspbian se deben conectar; un cable HDMI desde la salida de video de la tarjeta hacia la entrada de video de una TV, además de un teclado y un mouse al puerto USB y finalmente energizar la tarjeta conectando un cargador micro USB tipo c desde el adaptador de voltaje. Energizada la Raspberry Pi, se empieza inmediatamente a cargar el sistema operativo ya instalado en la SD. Se ajusta con los datos correspondientes al país, lenguaje, zona horaria y se selecciona la red wifi. Cargados todos los directorios se debe iniciar sesión ingresando; el Login (usuario) que es 'pi' y el Password (contraseña) que es 'raspberrypi'. Echo esto, el último paso para abrir el interfaz gráfico de usuario se debe ejecutar el siguiente comando en el terminal de la tarjeta seguido de la tecla enter.

```
sudo startx
```

Para poder acceder remotamente al escritorio de la Raspberry Pi se debe instalar el servidor de escritorio remoto "Xrdp". Esta aplicación permitirá acceder desde Linux, Windows o Mac al entorno Raspbian de Raspberry, para ello se accede al LXTerminal de la tarjeta y se ejecuta el siguiente comando seguido de la tecla enter.

```
sudo apt install xrdp
```

Finalizado el proceso de instalación, se debe reiniciar la Raspberry con el comando:

```
sudo reboot
```

Finalizada la instalación, la Raspberry PI ya está lista para acceder remotamente desde cualquier sistema operativo anteriormente mencionado.

2.5.3 ACCESO REMOTO A LA RASPBERRY PI DESDE WINDOWS

Los pasos para acceder desde Windows al entorno gráfico de la Raspberry Pi son los siguientes:

1. Configurar el uso compartido de internet a través de ordenador.



2. Identificar la IP asignada a la Raspberry pi usando el programa Fing disponible en [14].
3. Con ayuda de la aplicación de Windows llamada “Conexión a Escritorio remoto”.
4. Una vez conectado accedemos al entorno grafico de la Raspberry Pi. En la cual aparecerá la aplicación Xrdp instalada anteriormente, en ella se ingresa el usuario (pi) y contraseña (raspberry).
5. Finalmente se accede al entorno del sistema operativo Raspbian de Raspberry Pi.

2.5.4 ACTUALIZACIÓN Y DESCARGA DE PAQUETES NECESARIOS

Verificamos y actualizamos los paquetes y programas instalados a la última versión disponible de Raspbian, ingresando a una terminal y ejecutando el siguiente comando:

```
sudo apt – get update  
sudo apt – get upgrade
```

2.5.4.1 INSTALACIÓN DE PAQUETES PARA PYTHON

Finalmente, se deben instalar los paquetes necesarios de Python para el Desarrollo del algoritmo de extracción de patrones que son:[12]

- Paquete manejo de gráficos (matplotlib):

```
sudo apt – get install python – pip  
python3 – pip  
sudo – H pip install matplotlib == 3.2.1
```

- Paquete manejo de ficheros (numpy):

```
sudo – H pip install numpy == 1.18.2
```

- Biblioteca para aprendizaje automático (TensorFlow):

```
sudo apt – get install libhdf5 – dev libc – ares – dev libeigen3 – dev  
sudo apt – get install libatlas – base – dev libatlas3 – base  
sudo pip3 install h5py == 2.10.0  
sudo pip3 install – U – –user six wheel mock
```



```
# download the wheel
wget https://github.com/Qengineering/Tensorflow - Raspberry
    - Pi/raw/master/tensorflow - 1.15.2 - cp37 - cp37m
    - linux_armv7l.whl
# install TensorFlow
sudo - Hpip3installtensorflow - 1.15.2 - cp37 - cp37m - linux_armv7l.whl
# and complete the installation by rebooting
sudo reboot
```

- Biblioteca de visión artificial (opencv):

```
sudo apt - get install libhdf5 - dev libhdf5 - serial - dev libatlas - base
    - dev libjasper - dev libqtgui4 libqt4 - test
pip3 install opencv - contrib - python == 4.1.0.25
```

- Biblioteca de redes neuronales (keras):

```
sudo apt - install keras == 2.3.1
```

- Modulo para navegar por el sistema operativo (os):

```
sudo apt - install os
```

- Paquete para gestión de cámara web USB (fswebcam):

```
sudo apt - install fswebcam
```

2.6 CÁMARA USB

Debido a su capacidad para captar imagen y video en tiempo real, y su accesibilidad al puerto USB de la Raspberry pi, son ideales para el mapeo de la zona del cultivo de maíz. Las cámaras toman las imágenes a razón de varios cuadros (frames) por segundo, pero esta velocidad es inversamente proporcional a su resolución o calidad. [15] En este caso, por ejemplo, se presentan las características de la cámara que se adapta en el proyecto en la tabla 4.

Tabla 4. Características técnicas de cámara USB. [16]

CARACTERÍSTICAS	DESCRIPCIÓN
Modelo	PGR-008 cámara web: 2K Quad HD
Sensor	Sensor de imagen CMOS de 1/2.8"
Resolución	2K/1080P/720P (30 fps).



Formato de video	MJPEG.
Ángulo de visión	120°
Entrada	CC 5 V 100 MA.
Puertos	USB 2.0/3.0



Figura 7. Cámara web. [16]

2.7 SENSORES DE HUMEDAD

Los sensores de humedad miden una señal eléctrica que calcula la cantidad de agua en el suelo, lo cual permite determinar el volumen de agua almacenado en éste después de un riego o una lluvia, calcular el consumo de agua por el cultivo en un día o una semana y determinar la eficiencia del riego. (sonda)

2.7.1 SENSOR DE HUMEDAD DE SUELO ANTICORROSIVO-HIGRÓMETRO SNH0011

Es un sensor que lee la cantidad de humedad presente, consiste en dos sondas para pasar corriente a través del suelo, y luego lee la resistencia para obtener el nivel de humedad. Se caracteriza por su sensibilidad ajustable, voltaje de operación de 3.3V a 12V, modo de salida dual y amplificador LM393. [17]





Figura 8. sensor de humedad.[17]

2.7.2 MODULO SNH0011

El módulo SNH0011 es un convertidor del sensor que nos entrega una señal digital y una análoga, lo cual indica el nivel de humedad presente en el suelo lo que nos permite determinar un umbral de decisión. Con el potenciómetro, se precisa el valor umbral de la salida digital dependiendo de la humedad detectada por el sensor de humedad de sonda, pero en este caso determina una salida de 1 o 0, por tanto, lo que se utiliza es la salida análoga para establecer el rango de activación para el riego según corresponda la medida de humedad del sensor.

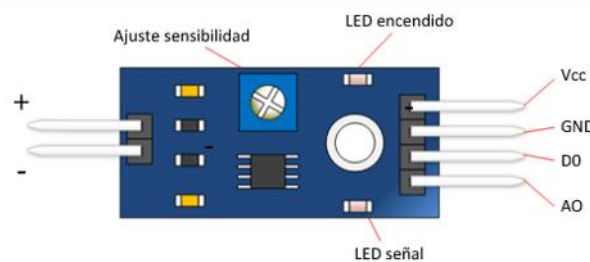


Figura 9. Convertidor del sensor de humedad. [18]

2.8 ELECTROVÁLVULA SOLENOIDE

Las electroválvulas son válvulas cuyo accionamiento es debido a un dispositivo eléctrico, interruptores eléctricos simples, termostáticos, de flotador, de baja presión, de alta presión, por reloj, o cualquier otro dispositivo que abra o cierre un circuito eléctrico. El objeto de su diseño radica en el control del flujo de agua a través de la tubería. Una electroválvula está formada por dos elementos: [19]

- **El solenoide:** es una bobina de cobre, con la forma de un cilindro largo, que al transportar una corriente se asemeja a un imán de modo que un núcleo móvil de material ferromagnético es atraído a la bobina cuando fluye una corriente.[19]
- **La válvula:** es el elemento que permite o restringe el paso del agua debido al accionamiento del solenoide.[19]



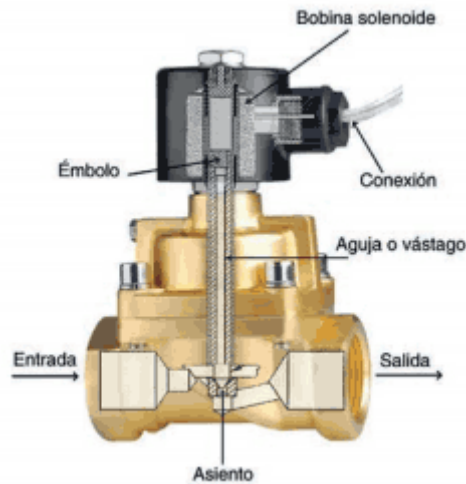


Figura 10. Componentes de una válvula solenoide. [19]

Las electroválvulas normalmente cerradas funcionan de manera que cuando se excita el solenoide este mueve su núcleo ferromagnético provocando la entrada de agua que eleva el elemento de cierre de la válvula de tal forma que se produce la apertura de la misma, permitiendo el paso de agua. [19]

2.9 SISTEMAS DE CONTROL

Son sistemas diseñados para lograr que una o varias variables se comporten de una manera deseada. La variable puede mantenerse constante o cambiar de una manera determinada. [20]

2.9.1 SISTEMAS DE CONTROL EN LAZO CERRADO

Control en el cual la salida afecta la acción de control. Es decir, se mide y se realimenta la salida para compararla con la entrada, además, el uso de la realimentación vuelve la respuesta del sistema relativamente insensible a las perturbaciones externas y a las variaciones internas en los parámetros del sistema. Es así posible usar componentes relativamente poco precisos y baratos para obtener el control adecuado de una planta determinada. Se representa mediante la ecuación 1: [20]

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

Ecuación 1. Función de transferencia de un sistema en lazo cerrado [20]



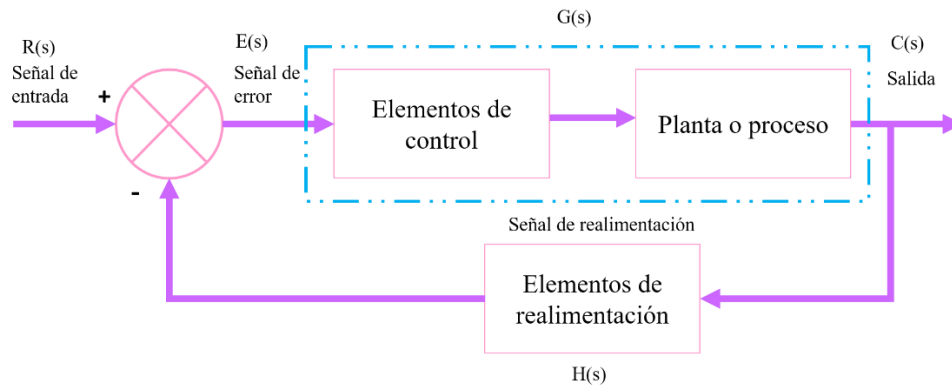


Figura 11. Sistema de control en lazo cerrado. [20]

2.9.2 ACCIONES DE CONTROL

Es la característica, modo o algoritmo, con el cual el controlador calcula la señal de salida al elemento final de control, según la evolución de la variable del proceso y de su valor deseado.

2.9.2.1 CONTROL ON-OFF CON HISTÉRESIS

El control on-off con histéresis, incorpora un retardo en el cambio de la señal para disminuir la cantidad de conmutaciones, prolongando la vida útil de los componentes. Además, la salida no cambia mientras el error (SP) está en la zona muerta. M1 se activa unos grados por arriba del SP y se desactiva unos grados por debajo. Cuanto mayor es la zona muerta ($\Delta \phi$), menor es la frecuencia de oscilación. [21]

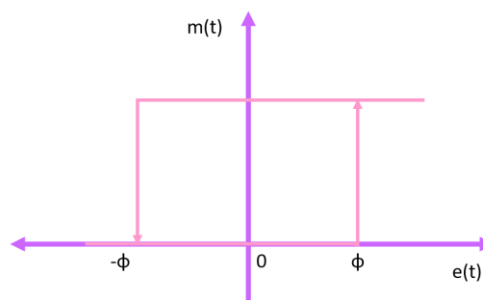


Figura 12. Descripción gráfica control ON/OFF con histéresis.[21]

2.10 INTELIGENCIA ARTIFICIAL (IA)

La IA es una rama de las ciencias computacionales que se ocupa de los símbolos y métodos no algorítmicos para la resolución de problemas. Los cuatro grandes objetivos de la IA son modelar, formalizar, programar e implementar máquinas



soporte capaces de interactuar de forma no trivial con el medio. existen varios elementos que componen la ciencia de la IA, dentro de los cuales se pueden encontrar tres grandes ramas: Lógica difusa, redes neuronales artificiales y algoritmos genéticos, estas técnicas de aprendizaje automático nos permiten establecer modelos sin necesidad de conocer los detalles matemáticos utilizando datos o experiencias pasadas, de este modo se podrán identificar ciertos patrones o regularidades en los datos y así podremos construir buenas aproximaciones al problema. [22]

2.10.1 NEURONA BIOLÓGICA

La neurona biológica es una célula que se compone de tres partes principales: el soma, las dendritas y el axón. Se caracterizan por su capacidad para comunicarse mediante señales o impulsos de otras neuronas que se reciben a través de sus dendritas y el cuerpo celular y a su vez estas señales son procesadas por el soma para luego transmitir las por el axón como un impulso eléctrico, los cuales distribuyen la información y sus salidas se conectan a otras neuronas. [22]

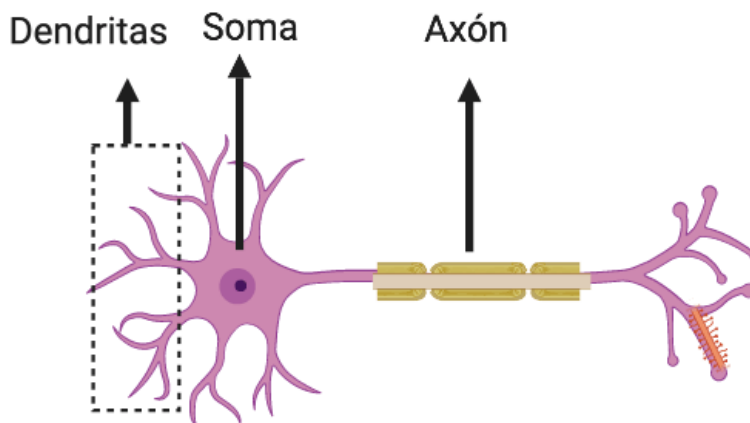


Figura 13. Partes de la neurona biológica.[22]

2.10.2 NEURONA ARTIFICIAL

Se basan en el procesamiento de información del modo que lo hace el cerebro y el modelo matemático de neurona propuesto por McCulloch y Pitts en 1943, donde cada neurona recibe un conjunto de entradas y devuelve una única salida. Además, dentro de una neurona artificial existen numerosas conexiones entre las distintas neuronas que la forman las cuales son robustas al igual que en la neurona biológica. Su similitud entre la actividad sináptica se puede considerar como entradas a la neurona las cuales representan las dendritas y que a través del parámetro denominado peso puede excitar o inhibir la neurona y en efecto, si la suma que equivale al soma es igual o mayor que el umbral de la neurona, entonces la neurona se activa representando al axón. [22]



La neurona artificial es el procesador elemental de la red, procesa el vector $(x_1, x_2, x_3, \dots, x_N)$ de entrada y genera una respuesta lógica. De la Ecuación 2 representa la salida de la neurona que no es más que una suma ponderada de todas las entradas multiplicado por un valor w más el bias que representa el punto de corte del eje vertical, luego esta suma ponderada de valores será pasada por una función de activación que en el caso de la neurona artificial es la función escalón. El aprendizaje de la neurona está dado por el valor del vector de pesos los cuales serán actualizados en una serie de épocas hasta dar con la solución del problema.[22]

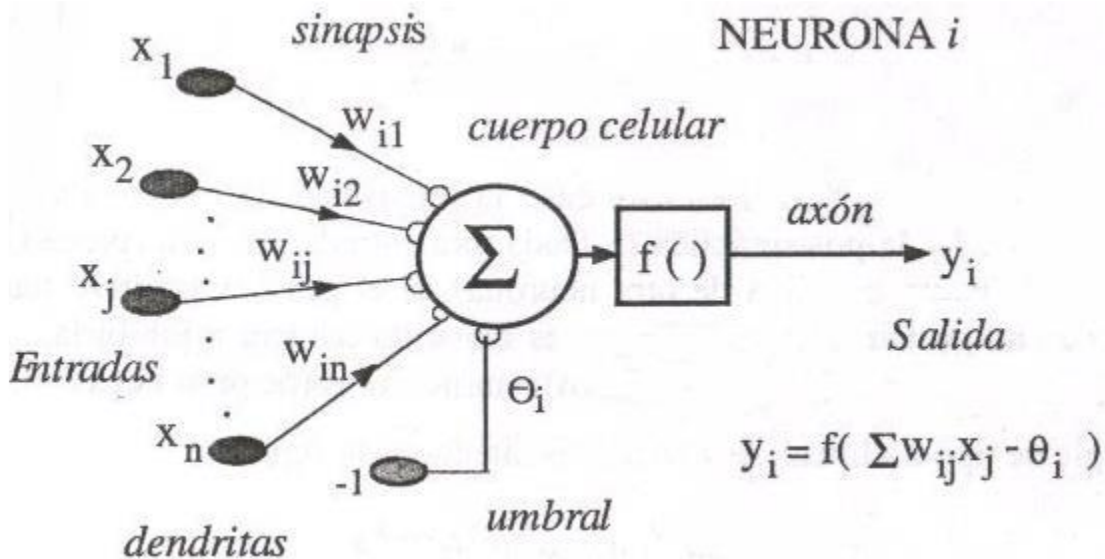


Figura 14. Modelo neuronal de McCulloch-Pitts.[22]

Cabe destacar que esta arquitectura solo admite entradas y salidas de tipo lógica y solo puede dar soluciones a problemas de tipo linealmente separables.

- y_i es la salida de la neurona que es de tipo lógica.
- f es la función de activación.
- θ_i representa la constante de bias.
- w_{ij} es el peso de la conexión.
- x_j representa el valor de entrada.

$$y = f\left(\sum_k w_k x_k + \theta\right)$$

Ecuación 2. Función de salida del perceptrón simple. [22]

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Ecuación 3. Función de activación en la neurona artificial.[22]



2.10.3 REDES NEURONALES ARTIFICIALES (RNA)

Es un sistema de mapeo no lineal que se basa en los sistemas nerviosos humanos y animales, consiste en la agrupación de neuronas artificiales conectadas entre sí. Una RNA se constituye por diferentes capas que tienen la topología de la figura 15, donde se encuentra la capa de entrada de datos, la capa oculta que puede contener una o más capas y por último tenemos la capa de salida que generalmente tiene el mismo número de neuronas que de clases. [22]

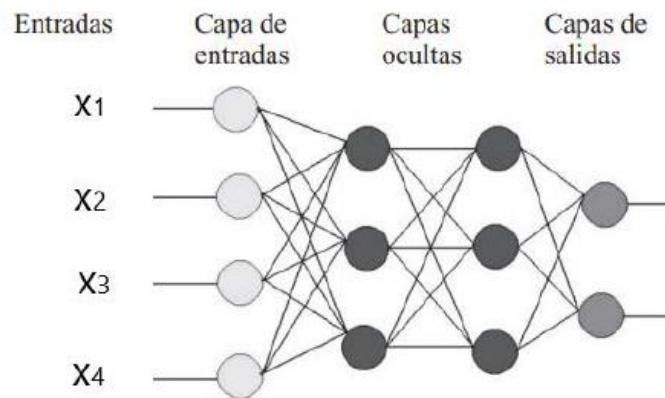


Figura 15. Topología básica de una RNA.[22]

2.10.3.1 TENSORFLOW

TensorFlow es una plataforma de código abierto que permite a los desarrolladores comenzar a utilizar el aprendizaje profundo en la nube de forma rápida y sencilla. El marco tiene un amplio respaldo en la industria y se ha convertido en una opción válida para la investigación de aprendizaje profundo y el desarrollo de aplicaciones, especialmente en ámbitos como la visión artificial, la comprensión de lenguaje natural y la traducción de voz. TensorFlow proporciona API estables de Python y C ++, así como una API compatible con versiones anteriores no garantizada para otros lenguajes.[23]

2.10.3.1.1 ¿QUÉ ES UN TENSOR?

Los tensores son matrices multidimensionales con un tipo uniforme (llamado dtype). Todos los tensores son inmutables como los números y las cadenas de Python: nunca se puede actualizar el contenido de un tensor, solo crear uno nuevo. Se caracterizan por tener: [23]

- Forma: La longitud de cada uno de los ejes de un tensor.
- Rango: Número de ejes tensoriales. Un escalar tiene rango 0, un vector tiene rango 1, una matriz tiene rango 2.
- Eje o dimensión.



- Tamaño: el número total de elementos en el tensor, el vector de forma del producto.

2.10.3.2 MODELO KERAS

Keras es una interfaz de programación de aplicaciones (API) de alto nivel para el aprendizaje, escrito en Python y capaz de correr sobre los frameworks TensorFlow, Microsoft Cognitive Toolkit o Theano. Construido sobre TensorFlow 2.0, Keras es un marco de trabajo sólido de la industria que puede escalar a grandes grupos de GPU o un pod completo de TPU. Tiene dos modos de construir redes neuronales. El modelo secuencial que es el más simple porque permite que las capas se agreguen en secuencia donde cada capa tiene exactamente un tensor de entrada y un tensor de salida. El modelo funcional se usa para crear redes más complicadas porque cada llamada a la API de Keras creará y devolverá una instancia que se puede llamar como si fuera una función lo que permite crear gráficos más complejos. [23]

2.10.3.2.1 CAPAS

Las capas son funciones con una estructura matemática conocida que se pueden reutilizar y tienen variables entrenables. En TensorFlow, la mayoría de las implementaciones de alto nivel de las capas y modelos, son de Keras o Sonnet. *tf.keras.layers.Layer* es clase base de todas las capas de Keras, heredadas de la misma clase fundamental: *tf.Module*. además, tienen las suyas propias como, por ejemplo: [23]

- Flaten: se encargan de aplanar la entrada y no afectan el tamaño del lote, se define de la siguiente manera:

tf.keras.layers.flatten

- Dense: Es la capa densa conectada que implementa la operación: $output = activation(dot(input, kernel) + bias)$ donde *activation* se pasa la función de activación por elementos como *activation* argumento, *kernel* es una matriz de ponderaciones creada por la capa y *bias* es un vector de sesgo creado por la capa (solo aplicable si lo *use_bias* es *True*). Todos estos son atributos de Dense, se define como:

tf.keras.layers.Dense

2.10.3.2.2 COMPILACIÓN DEL MODELO



La compilación del modelo agrupa capas en un objeto con funciones de entrenamiento e inferencia. Hay dos formas de crear una instancia *Model*: Con la subclasificación del modelo, en el que se debe definir las capas e implementar el paso hacia adelante. La segunda forma es mediante la API funcional, la cual encadena llamadas para especificar el paso hacia adelante del modelo y finalmente se crea el modelo a partir de entradas y salidas. Una vez implementado el modelo se puede configurar el modelo con pérdidas y métricas, mediante: [23]

- Optimizador: Es el encargado de generar pesos cada vez mejores, su funcionamiento esencial se basa en calcular el gradiente de la función de coste (derivada parcial) por cada peso (parametro/dimension) de la red. Utilizado para el rendimiento óptimo de tensorflow, en el que se derivan varias clases; por ello es útil visualizar su rendimiento en diferentes funciones de coste, por ejemplo, el optimizador Adam que es un método de descenso de gradiente estocástico que se basa en la estimación adaptativa de momentos de primer y segundo orden. [23]
- Función de pérdida: Es donde están los valores de verdad básicos y son las predicciones del modelo. La función de pérdida debería devolver un tensor flotante, además debe tener forma, excepto en el caso de funciones de pérdida dispersa, como la entropía cruzada categórica dispersa que espera matrices enteras de forma (*tf.keras.metrics.sparse_categorical_crossentropy*). [23]
- Métricas: el modelo evaluará durante el entrenamiento y las pruebas. Cada uno de estos puede ser una cadena (nombre de una función incorporada), una función o instancia. Normalmente se utiliza la métrica de precisión: [23]

metrics = ['accuracy']

2.10.3.2.3 FUNCIONES DE ACTIVACIÓN

Tanto en las redes neuronales artificiales como biológicas, una neurona no sólo transmite la entrada que recibe. Existe un paso adicional, una función de activación, que es análoga a la tasa de potencial de acción disparando en el cerebro. La función de activación utiliza la misma suma ponderada de la entrada anterior, y la transforma una vez más como salida. Sin esta, no tendría sentido la red porque se comportaría como un modelo lineal incapaz de solucionar problemas no lineales. Actualmente existe variedad de funciones de activación,



pero hay dos tipos que tienen aceptación y respuesta en el desarrollo deeplearning. [24]

2.10.3.2.3.1 SOFTMAX

La función de activación de Softmax mapea entradas no normalizadas en un conjunto de probabilidades exponenciales y normalizadas. Se utiliza en problemas de clasificación de clases múltiples. Esta función de activación devuelve la distribución de probabilidad de cada una de las clases soportadas en el modelo. La función Softmax calcula la distribución de probabilidades del evento sobre 'n' eventos diferentes. Es decir, esta función calculará las probabilidades de cada clase objetivo sobre todas las clases objetivo posibles. Más tarde, las probabilidades calculadas serán útiles para determinar la clase objetivo para las entradas dadas. La principal ventaja de usar Softmax es el rango de probabilidades de salida. El rango será de 0 a 1, y la suma de todas las probabilidades será igual a uno. Si la función softmax utilizada para el modelo de clasificación múltiple devuelve las probabilidades de cada clase y la clase objetivo tendrá una probabilidad alta. Se define con la ecuación 4. [25]

$$\phi(x_i) = \frac{e^{x_i}}{\sum_{j=0 \dots k} e^{x_j}} \quad i = 0, 1, 2, \dots, k$$

Ecuación 4. Representación matemática de la función de activación softmax. [25]

2.10.3.2.3.2 RELU

Las unidades de lineales rectificadas (ReLU) permiten el paso de todos los valores positivos sin cambiarlos, pero asigna todos los valores negativos a 0. Aunque existen funciones de activación aún más recientes, es una de las funciones más utilizadas por la optimización en la rapidez de aprendizaje de las redes neuronales. En la ecuación 5 si la función recibe valores de entrada negativos el resultado será cero, pero si el valor de argumento es mayor igual a cero mantendrá su valor. En comparación con otras funciones de activación RELU no tiene zonas de saturación esto significa que los pesos de conexión son los óptimos ya que al no tener variación de saturación hay una mayor flexibilidad en los pesos. [24]

$$f(z) = \max(0, z) = \begin{cases} 0, & z < 0 \\ z, & z \geq 0 \end{cases}$$

Ecuación 5. Representación matemática de la función RELU. [24]



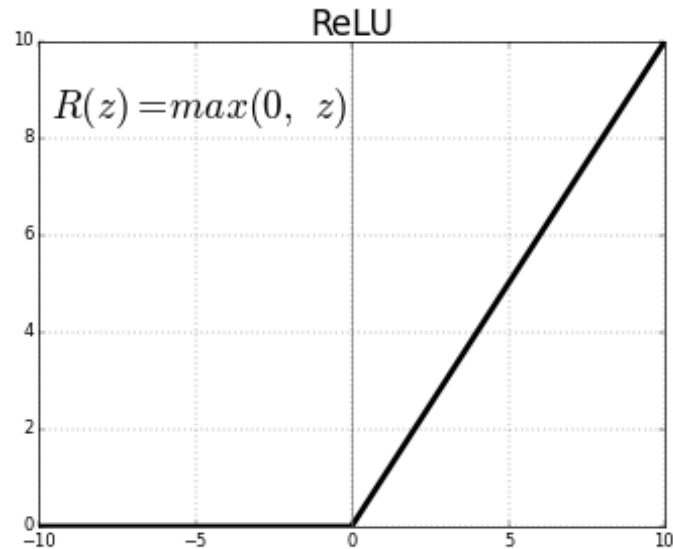


Figura 16. Gráfica de la función de activación RELU. [24]

2.10.3.3 ENTRENAMIENTO

Es el proceso de configuración de una red neuronal Para que las entradas produzcan las salidas deseadas, mediante métodos como la implantación de pesos conocidos con anterioridad, uso de técnicas de retroalimentación y patrones de aprendizaje que cambian los pesos hasta encontrar los adecuados. Se debe tener en cuenta las siguientes conceptualizaciones: [22]

- Pesos: Representan la intensidad de interacción entre cada neurona presináptica j y la neurona postsináptica. Los pesos sinápticos pueden tomar valores positivos que actúan como excitador, negativos actúan como inhibidor o cero cuando no existe comunicación.
- Bias o sesgo: Es el que controla qué tan predispuesta está la neurona a disparar un 1 o un 0 independiente de los pesos. Un sesgo alto hace que la neurona requiera una entrada más alta para generar una salida de 1. Un sesgo bajo lo hace más fácil.
- Entradas y salidas: Las entradas y salidas de una neurona pueden ser clasificadas en dos grandes grupos, binarias [1 0] o continuas [1 -1].

El proceso de aprendizaje se divide en tres grupos, el primero se conoce como supervisado cuando introducen entradas que corresponden a determinadas salidas, ya sea por un agente externo o por el mismo sistema, en el no supervisado el entrenamiento se enfoca a encontrar características estadísticas entre agrupamientos de patrones en las entradas y finalmente por refuerzo que es la combinación de los dos anteriores, sin embargo, no se le proporciona el valor de la salida esperada. Este tipo de aprendizaje es muy útil en aquellos casos en que se desconoce cuál es la salida exacta que debe proporcionar la red.[22]



2.10.4 GOOGLE COLAB

Google colaboraty es una plataforma virtual que permite escribir y ejecutar código de Python en la nube, sin necesidad de configuraciones requeridas, además de su acceso gratuito a unidades de procesamiento gráfico y tensorial (GPU y TPU) configurándolo desde el entorno de ejecución. Los notebooks de Colab te permiten combinar código ejecutable y texto enriquecido en un único documento el cual se almacena en el Google drive personal. [26]

2.10.5 PYTHON

Python es un lenguaje de programación de alto nivel multiplataforma de código abierto y gratuito, lo que permite desarrollar software de todo tipo, ya que tiene un enfoque simple de programación orientada a objetos. El lenguaje es sencillo de leer y escribir por su sintaxis dinámica, además facilita trabajar con inteligencia artificial, big data y machine learning. El intérprete de Python y la extensa biblioteca estándar están disponibles gratuitamente en formato fuente o binario para todas las plataformas principales desde el sitio web de Python. [27]

2.10.5.1 Opencv.

Open Computer Vision es una biblioteca de enlaces de Python diseñada para resolver problemas de visión por computadora y aprendizaje automático. OpenCV admite una amplia variedad de lenguajes de programación como C ++, Python, Java, Matlab, etc., y está disponible en diferentes plataformas, incluidas Windows, Linux, OS X, Android e iOS. Está escrita de forma nativa en c++, esta librería consta de más de 2500 algoritmos completamente funcionales y optimizados, estos algoritmos pueden emplearse para la detección y reconocimiento de rostros, identificar objetos, clasificar acciones humanas, rastreo de movimiento, detección de objetos. Una de sus funciones es el espacio de color *cvt.color*, lo cual hace referencia a las capas de una imagen, es decir, una codificación que permite representar a los colores de forma numérica. Por ejemplo, los espacios de color más comunes son: [28]

- RGB. Una matriz de tres dimensiones con cada matriz guardando los valores de 8 bits (0 a 255) para la intensidad. Cada matriz guarda los colores R – Rojo, G – Verde (Green) y B – Azul (Blue. En OpenCV se llama BGR, por el orden en el que se guardan cada canal.
- HSV. Es una matriz de tres dimensiones, con cada matriz guardando, saturación, intensidad y color.

Por otra parte, otra de sus funciones es el cambio del tamaño de una imagen, es decir, las dimensiones de la misma, además el aspecto de la imagen se conserva y se utiliza para mejorar la imagen. Para cambiar el tamaño de una imagen, OpenCV proporciona la función *cv2.resize ()*. [28]



3.

Metodología.

3. METODOLOGÍA

En este capítulo se describe la metodología que se implementó para el desarrollo del proyecto, donde se explica las etapas de control para la creación de la planta, como también las herramientas de inteligencia artificial para la adquisición de datos y el procesamiento de imágenes de entrenamiento.



3.1 RESUMEN DE CONTROL DE RIEGO POR GOTEO

En la Figura 17 se observa el diagrama en bloques del desarrollo del sistema de control de riego a partir de la variable de humedad detectadas por el sensor higrómetro de sonda, el sistema está constituido básicamente de cinco etapas que se explicaran brevemente a continuación.

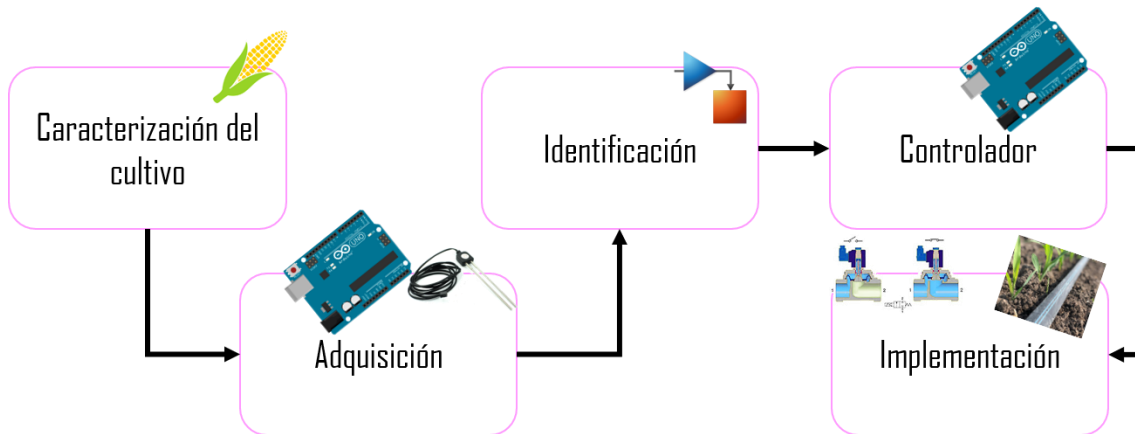


Figura 17. Diagrama en bloques de sistema de control de riego. [Autor]

La primera etapa del sistema es la caracterización del cultivo, encargada de todo lo relacionado con la plantación, desde la asesoría con el experto para llevar a cabo el proceso de preparación de la tierra, el tipo de semilla a usar para realizar la siembra, además de las condiciones edafoclimáticas para establecer los niveles de humedad, a partir de las medidas que arrojen los sensores de humedad higrómetro.

La segunda etapa del sistema es la adquisición y lectura de los datos del sensor obtenidos mediante el entorno y la tarjeta de desarrollo arduino uno, donde se determina el umbral bajo y alto del porcentaje de humedad para accionar la electroválvula.

La tercera etapa consiste en la identificación realizada en el entorno de programación Matlab con su aplicativo simulink, en el que se desarrolló la adquisición de datos de humedad proveniente del sensor puesto en la planta de maíz.

En la cuarta etapa se establece un control por histéresis para obtener las variables de entrada del sistema para implementar los umbrales de porcentaje de humedad alto, y bajo en el controlador. Luego, se desarrolla el diagrama en bloques de control por histéresis, en el cual se tiene en cuenta la variación de las medidas de humedad para determinar los valores de salida y el rango en el que la electroválvula debe mantenerse encendida para dar paso al riego de las plantas, como también el momento en que debe apagarse para evitar encharcamientos en el cultivo o que sea innecesario el riego por lluvia externa (perturbaciones). En este caso se considera 0V para cuando la válvula esta



apagada, es decir, cuando la humedad alcanza su rango máximo y 5V para cuando esta encendida y la humedad sea menor al rango mínimo.

En la última etapa de implementación, es donde se lleva a cabo el ensamble de los circuitos en PCB considerando los cálculos correspondientes, además de la conexión de los elementos del sistema en lazo cerrado y el modelado de la planta a partir de las precipitaciones o perturbaciones del ambiente.

3.1.1 CARACTERIZACIÓN DEL CULTIVO

Mediante la asesoría con el ingeniero agrónomo inicialmente se llevó a cabo la preparación de la tierra para la siembra con abonos como el cal y triple 18, además de seleccionar el tipo de semilla tecnificada que se adaptó a las condiciones climáticas para que la cosecha se recogiera en un tiempo estimado de 90 días y esta sea resistente a las temperaturas de la zona, como de otros factores (babosas, insectos, aves). Luego, se realizaron las medidas pertinentes para la adaptación del sistema de control; primeramente, se utilizaron dos sensores de humedad DHT11, los cuales en el proceso de adquisición de los datos se saturaron, además que la medida era superficial y la planta de maíz utiliza la humedad que está disponible en el cultivo 50 Cm por debajo. Por lo tanto, se optó por cambiar el elemento de medida por el sensor de sonda higrómetro para tomar los datos del sensor utilizando un circuito probado en Protoboard.

3.1.2 ADQUISICIÓN

La adquisición de los datos del sensor se realizó en código de arduino y se implementó en simulink como se muestra en la figura 18, donde primero se hace la lectura el pin análogo del módulo del sensor para luego guardarlos en la variable *tmp*, Así mismo en la variable de voltaje de salida (*vsa*) se muestra el cálculo del voltaje leído del arduino al módulo, y la conversión de bits leídos del puerto analógico lo cual nos arroja el voltaje correspondiente.



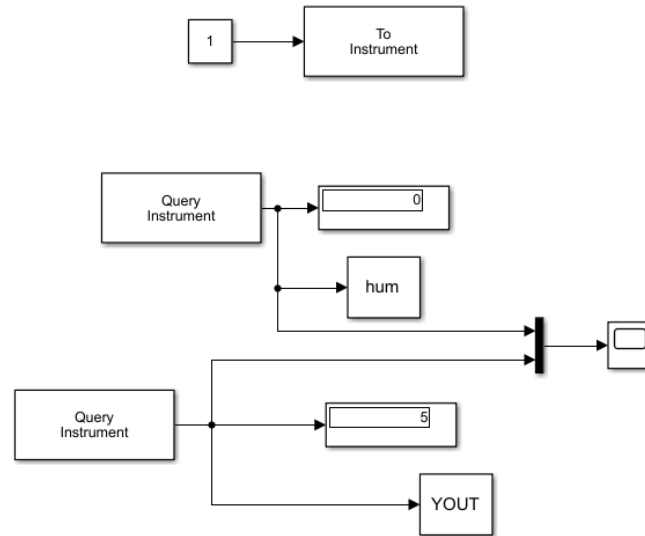


Figura 18. Adquisición de datos del sensor. [Autor]

finalmente, en la variable de humedad (*hum*) se mostrará el porcentaje de humedad del suelo, el cual se tiene en cuenta a partir de la figura 19 con la relación entre la humedad y el voltaje de salida, cuanto más cercano este a 5V más seco se encuentra el terreno y cuando este más cercano a 0V se encontrará más húmedo el nivel del suelo.

```
tmp = analogRead(moistureA0);
vsal = (5 * tmp)/1023.00;
hum = 100 - (vsal * 20);
```

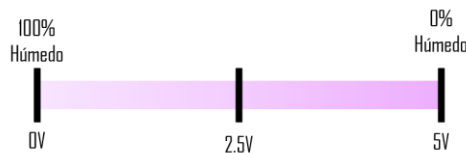


Figura 19. Relación de la cantidad de humedad y el voltaje de salida. [Autor]

3.1.3 IDENTIFICACIÓN

Se desarrolló la identificación de la planta luego de que se hayan adquirido los datos del sensor (Figura 18), así pues, se guardan las variables de humedad y salida para conocer la función de transferencia del sistema que corresponde ecuación 6 y su modelado mostrado en la figura 20. Cabe aclarar que para obtener la función de transferencia se debe realizar en el command window de Matlab con la función *ident*, lo cual genera la ventana de la figura 21 y primero se importaron los datos en el dominio del tiempo, luego se digitan las variables



de entrada y salida y se importan como se muestra en la figura 22 y finalmente el identificador de sistemas estimó la función de transferencia, asignando el número de polos y ceros, 4 polos y 3 ceros respectivamente. Logrando una precisión de identificación mayor al 90%.

$$\frac{y_{sal}}{x_{entr}} = \frac{-0.003881 s^3 - 5.028e - 05 s^2 - 1.031e - 07 s - 8.004e - 13}{s^4 + 0.03819 s^3 + 0.0003487 s^2 + 6.819e - 07 s + 4.878e - 11}$$

Ecuación 6. Función de transferencia del sistema de sensor 1. [Autor]

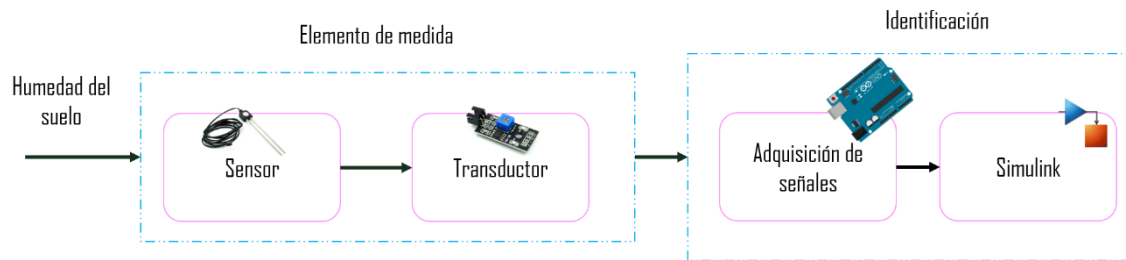


Figura 18. Modelado de función de transferencia. [Autor]

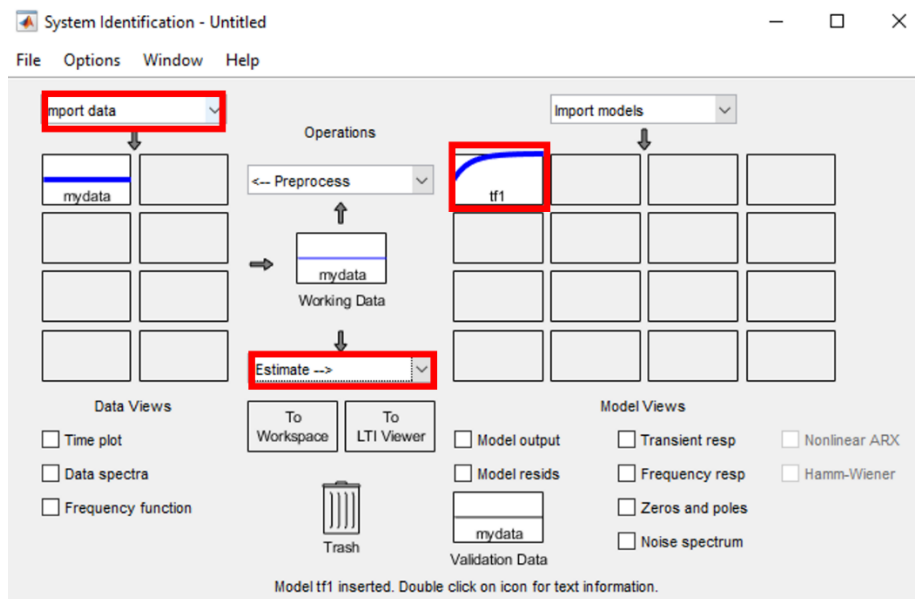


Figura 19. adquisición de la función de transferencia. [Autor]



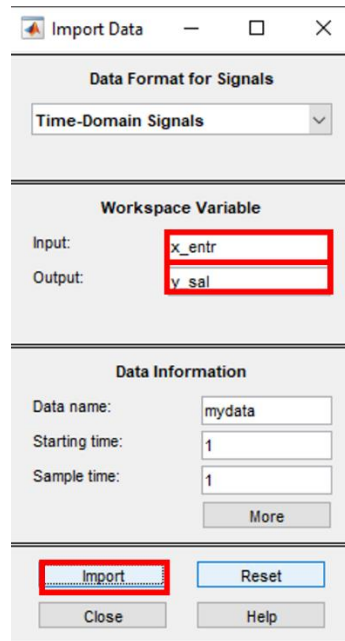


Figura 20. Adquisición de la función de transferencia. [Autor]

Igualmente, para la sintonización del segundo sensor se realizó el mismo procedimiento, teniendo en cuenta que los datos adquiridos varían y, por tanto, su función de transferencia como se ve en la ecuación 7.

$$\frac{y_{sal2}}{x_{entr2}} = \frac{0.1618 s^3 + 6.096e - 05 s^2 + 6.079e - 07 s + 3.015e - 12}{s^4 + 1.212 s^3 + 0.00167 s^2 + 2.264e - 06 s + 2.857e - 09}$$

Ecuación 7. Función de transferencia del sistema de sensor 2. [Autor]

3.1.4 CONTROLADOR

Por lo que se refiere al controlador se compone de la tarjeta arduino junto con la planta, cuando la humedad está por debajo del umbral del 40%, (figura 23) la electroválvula se activa como se visualiza en la figura 24, en efecto sucede y se desactiva cuando alcanza el 70% de la humedad.



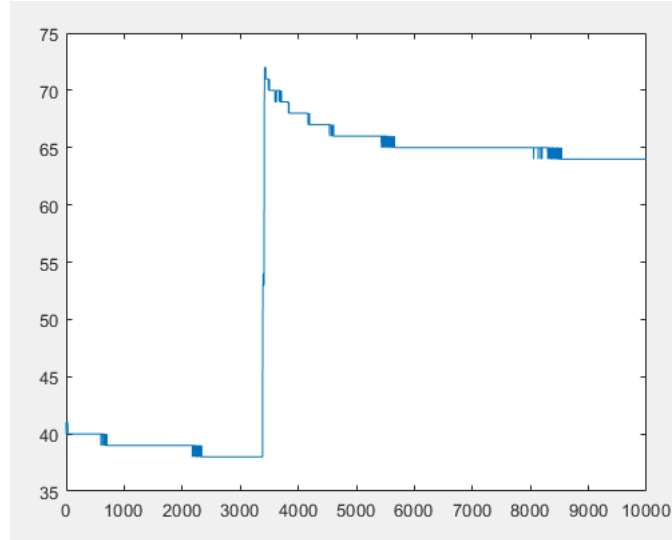


Figura 21. Adquisición de datos de humedad del sensor 1. [Autor]

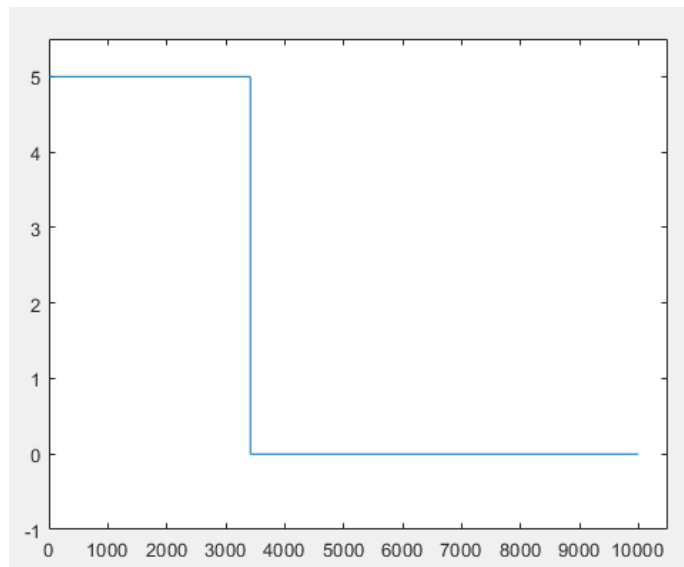


Figura 22. Desactivación de electroválvula 1. [Autor]



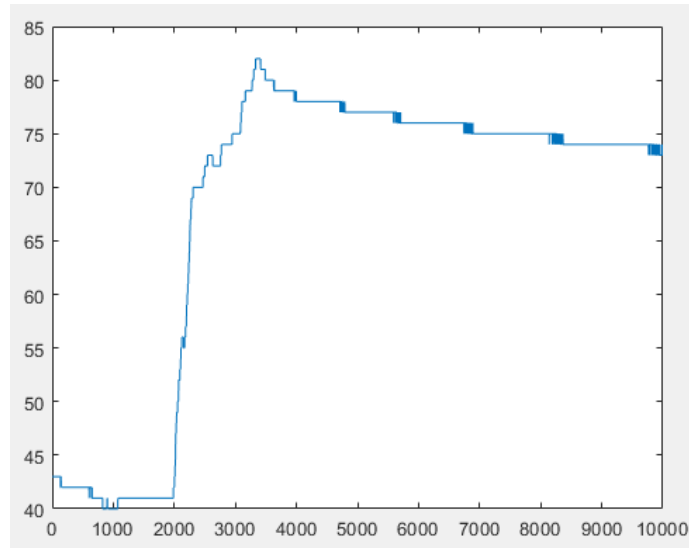


Figura 23. Adquisición de datos de humedad del sensor 2. [Autor]

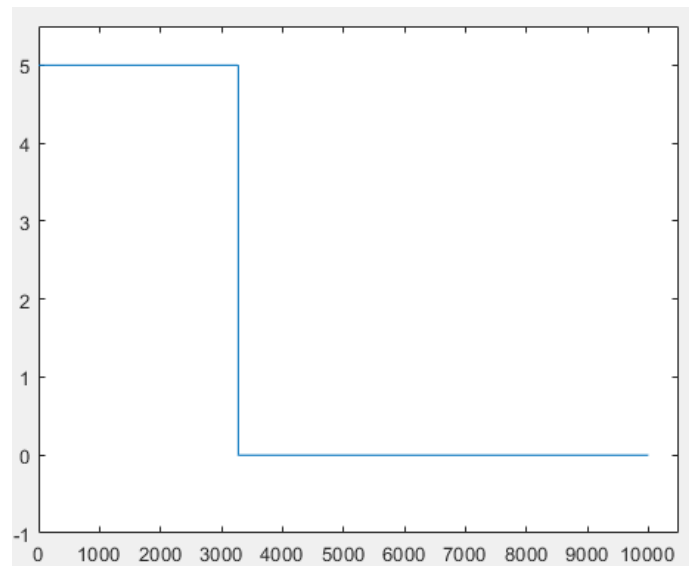


Figura 24. Desactivación de electroválvula 2. [Autor]

3.1.5 IMPLEMENTACIÓN

Con respecto a la implementación se llevó a cabo con el ensamble de cada elemento que compone el lazo cerrado, en donde se realizó el diagrama del circuito en proteus que se va a implementar en la PCB indicada en la figura 25, para ello, es necesario tener en cuenta los cálculos para activar la electroválvula y aislar las salidas del arduino de las corrientes de encendido de las electroválvulas, además del código de los dos sensores mostrado en la figura 26.



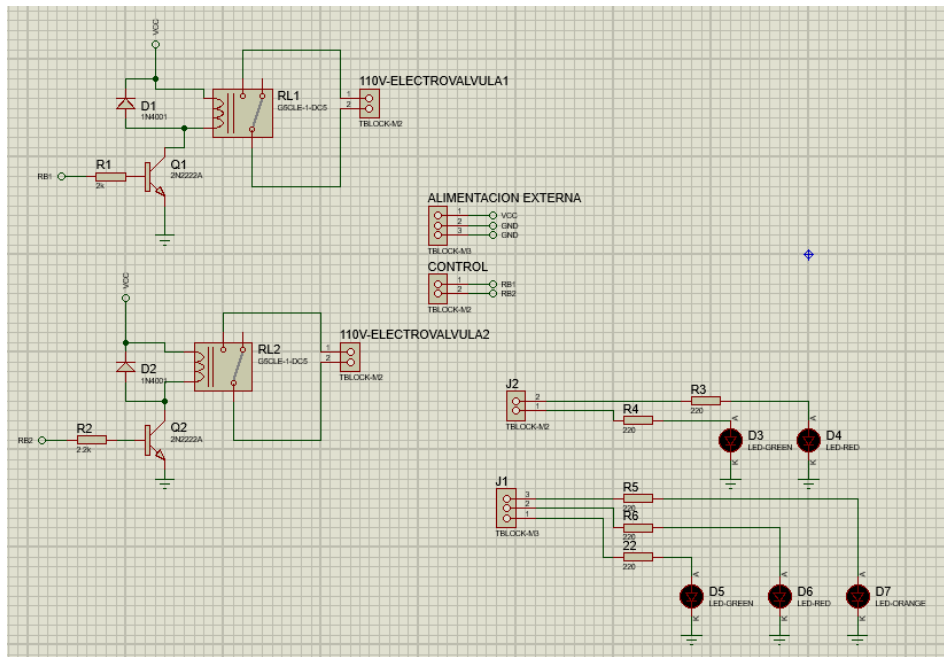


Figura 25. Circuito de implementación del sistema de control por histéresis. [Autor].

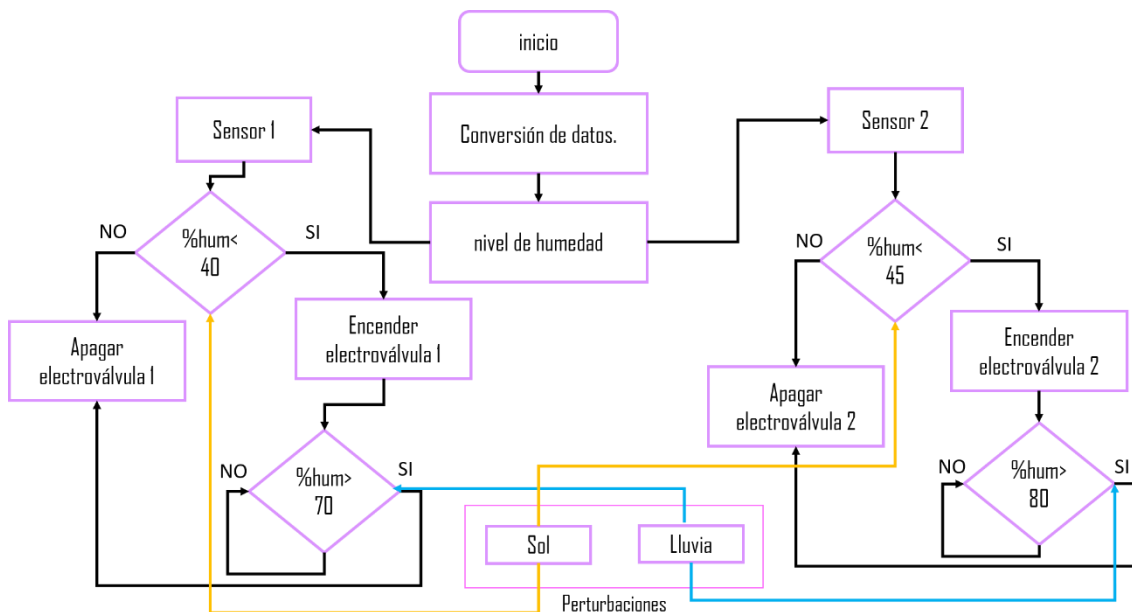


Figura 26. Diagrama lógico de programación del sistema de control. [Autor]

Los cálculos para la activación de la electroválvula se muestran a continuación:

$$I_B = \frac{I_C}{\beta_1}$$

$$R_B = \frac{V_{sal} - V_{BEON}}{I_B}$$

$$R_B = \frac{V_{sal} - V_{BEON}}{\frac{I_C}{\beta_1}}$$



El relé necesita una corriente de 85 mA para ser activado, garantizamos la corriente de colector superior al 10% de la corriente de consumo del relé, es decir

$$I_C = 85 + 85 * 10\% [mA]$$

$$I_C = 93,5 [mA]$$

Para una corriente de 93,5 mA, el Datasheet nos indica un β de 50

$$R_B = \frac{5V - 0,7V}{\frac{93,5 mA}{50}}$$

$$R_B = \frac{4,3V}{1,87 mA}$$

$$R_B = 2,2 K\Omega$$

$$R_B = 2 K\Omega$$

3.2 RESUMEN DE INTELIGENCIA ARTIFICIAL

En la Figura 27 se observa el diagrama en bloques del desarrollo del sistema de identificación a partir del entrenamiento de imágenes en una red neuronal, el sistema se constituye de seis etapas que se explicaran brevemente a continuación.

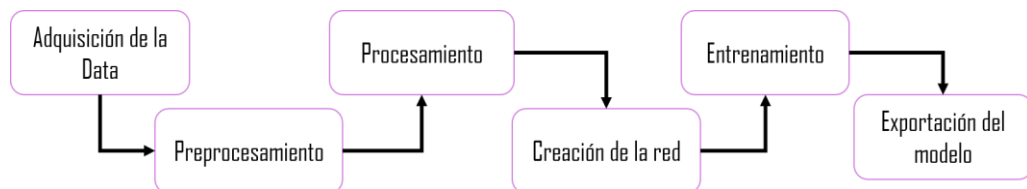


Figura 27. Diagrama en bloques de identificación de malezas. [autor]

En la primera etapa se desarrolla la adquisición de la data, en este caso se recolectan las imágenes que se utilizan previamente para el entrenamiento y validación de la red neuronal, donde se utilizan dos tipos de imágenes, maíz con maleza que contiene 70 imágenes para el entrenamiento y 20 para la validación, en el caso de las imágenes de maíz



sin maleza cuenta con 200 imágenes para el entrenamiento y 80 para validación.

En la segunda etapa de preprocesamiento, se crean cada una de las clases de imágenes, las cuales se suben al Google Drive y se adjuntan en el código, donde se van identificar mediante etiquetas, es decir, se asigna 0 para las imágenes correspondientes al maíz con maleza y 1 para las plantas sin maleza, luego de realizar el etiquetado de las imágenes, se concatenan para que todas las imágenes se guarden en una variable llamada *images*.

La tercera etapa consiste en el procesamiento, donde se aplican las técnicas para el tratamiento de las imágenes, como el espacio de color RGB que se adapta con el espacio que trabaja opencv, además de realizar un redimensionamiento a las imágenes para mejorar la calidad de las mismas.

La cuarta etapa se trata de la creación de la red, donde se trabaja con el modelo secuencial y se utilizan dos tipos de capas, *flatten* en la cual se usan tres canales de dimensión 150x150 y *dense* para asignar las funciones de activación *softmax* y *relu*. Finalmente, se compila la red teniendo en cuenta que se mide la precisión en el entrenamiento y validación posteriormente, además de sus pérdidas por entropía cruzada de escasa categoría por las dos clases que se utilizan en la red y el rendimiento y optimización *Adam* la cual estima pesos cada vez mejores.

La quinta etapa de entrenamiento del modelo, se considera el número de épocas en el que la red se entrena y encuentra la precisión en la que esta, así mismo, mediante estos datos se comparan con la validación que se prueba con diferentes imágenes a las del entrenamiento para verificar si la red aprendió y mantiene, mejora o disminuye su precisión.

La etapa de exportación del modelo consiste en guardar los pesos y el modelo para luego encriptarlos en la tarjeta rpi4 y trabajar desde allí el modelo para la identificación de maleza en el cultivo a través de la captación de imágenes dada por la cámara USB.

3.2.1 ADQUISICIÓN DE LA DATA

Para la recolección de la data se tomaron imágenes de dos cultivos diferentes, en el cual se diferenciaban por la presencia y no presencia de maleza, como se muestra en la figura 28 se utilizaron 270 imágenes para el entrenamiento, donde 70 pertenecen a imágenes con maleza y 200 sin maleza, además en la figura 29 se muestra también el conjunto de imágenes adquiridas para la validación.



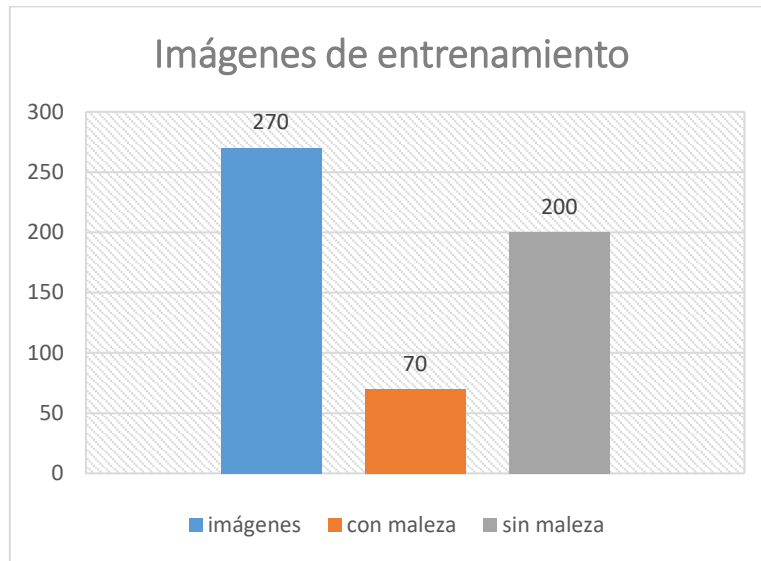


Figura 28. Conjunto de imágenes de entrenamiento. [Autor]

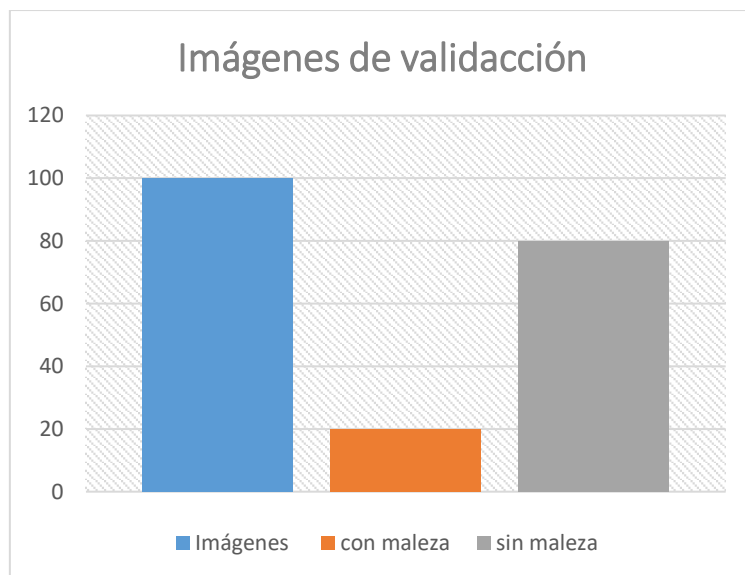


Figura 29. conjunto de imágenes de validación. [Autor]

La figura 30 contiene imágenes de muestras con los que se entrenaron el modelo de inteligencia artificial y como se puede evidenciar son imágenes de maíz con maleza y sin maleza.





Figura 30. muestras de las clases de imágenes con las que se entrena el modelo. [Autor]

3.2.2 PREPROCESAMIENTO

En esta etapa de preprocesamiento se da inicio a la creación de las clases a partir de la data adquirida y posteriormente subida en Google drive para incorporarlas en el código de Google colab y de este modo realizar el entrenamiento y validación de la red neuronal como se presenta en la figura 31. Luego para guardar las imágenes en una lista se etiquetan las dos clases de cada una de las imágenes, esto para que nos permita la identificación del tipo que pertenece la imagen, cuando es 0 a imágenes sin maleza y 1 a las imágenes con maleza; finalmente, se concatenaron todas las 270 imágenes



en la variable *images* esto con el fin de que tensorflow las reconozca como entradas de la red neuronal.

```
#CARGAR LOS DATOS DEL DRIVE
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[ ] !cp -r "/content/drive/MyDrive/DATA/entrenamiento" "/content/entrenamiento"

[ ] !cp -r "/content/drive/MyDrive/DATA/validacion" "/content/validacion"
```

Figura 31. carga de datos para el entrenamiento y validación de la red. [Autor]

```
etiquetas_SM = np.repeat(0,200)
etiquetas_CM = np.repeat(1,70)
images = np.concatenate([SM,CM])
```

3.2.3 PROCESAMIENTO

Por otra parte, en el procesamiento se realiza el tratamiento de las imágenes, donde se tiene en cuenta el espacio de color RGB que se les aplico a cada una de las imágenes debido a que OpenCv trabaja con el espacio BGR, además de realizar un redimensionamiento de las imágenes de 150 pixeles de ancho por 150 pixeles de alto, y se guarda en la ultima posición de la lista al procesar cada imagen.

```
for img in os.listdir(SM_folder_path):
    img = cv2.imread(os.path.join(SM_folder_path,img))
    img_color= cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
    img_resize= cv2.resize(img_color,(img_size,img_size))
    SM.append(img_resize)
```

3.2.4 CREACIÓN DE LA RED

En cuanto a la creación de la red neuronal convolucional, se tiene en cuenta la implementación del modelo keras donde primero se definió la estructura de la red con la función `model = keras.Sequential`, con esto se está indicando que el número de capas van de manera jerárquica que en este caso son de dos tipos `keras.layers.Flatten` y `keras.layers.Dense`. En cada capa la función de activación se seleccionó un tipo diferente, en *dense* es tipo RELU ya que tiene una rápida convergencia en el entrenamiento al no tener zonas de saturación en comparación con otras funciones de activación. Para *flatten* es de tipo SOFTMAX la cual define la capa de salida que consta de 2 neuronas equivalente al número total de clases, para lograr obtener resultados porcentuales, la función de



activación va a seleccionar como salida la neurona con mayor porcentaje de activación.

```
model = keras.Sequential([keras.layers.Flatten(input_shape=(150, 150, 3)),keras.layers.Dense(128, activation='relu'),keras.layers.Dense(2, activation='softmax')])
```

En la compilación se configura el proceso de aprendizaje, primero se implementa el algoritmo de optimización, es decir, el método de Adam (*Adaptive moment estimation*), el cual, es adecuado para grandes parámetros y se adapta dependiendo de los resultados de la predicción y función de pérdida, en otras palabras, calcula cuanto necesita para reducir la pérdida entre la entrada y la predicción. Luego se determina la función de pérdida, debido a que se tiene dos clases se hace uso del `sparse_categorical_crossentropy` el cual calcula la pérdida entre la etiqueta y la predicción. Por último, para la medición del aprendizaje del modelo se define `accuracy` que permite ver la precisión del modelo.

```
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

En la *Figura 32* se aprecia la estructura del modelo implementado como el número de parámetros que serán ajustados en el entrenamiento del modelo. En la columna uno se especifica el nombre de cada capa creada, la columna dos especifica el tamaño de los tensores de salida y en la columna tres detalla el número de parámetros por bloque.

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 67500)	0
dense (Dense)	(None, 128)	8640128
dense_1 (Dense)	(None, 2)	258

Total params: 8,640,386
Trainable params: 8,640,386
Non-trainable params: 0

Figura 32. Estructura interna del entrenamiento de la red neuronal. [Autor]



3.2.5 ENTRENAMIENTO

Para el entrenamiento del modelo, se realizó con la función *fit()* el cual recibe como parámetro las imágenes, etiquetas para el entrenamiento, el número de épocas, y para conocer la precisión con la que la red fue entrenada se acude a la función *tensorboard*, se guardan los datos en el directorio para lanzar la aplicación desde Google colab y visualizar las gráficas del entrenamiento de la red al transcurrir las épocas .

```
from tensorflow.keras.callbacks import TensorBoard

tensorboardcnn = TensorBoard(log_dir='logs/model')

model.fit(Images, Labels, epochs=10, callbacks=[tensorboardcnn])
```

3.2.6 EXPORTACIÓN DEL MODELO A RPI4

Finalmente, para la última etapa del sistema de identificación de malezas, primero guardamos el modelo y los pesos en el directorio de trabajo, luego se realizó la exportación del modelo entrenado y sus pesos adquiridos en Google colab, descargando en el sistema embebido, para verificar la validación desde su entorno raspbian para adaptarlo y monitorizar el cultivo.

```
model.save('./modelo/modelo.h5')

model.save_weights('./modelo/pesos.h5')
```



4.

Resultados.

4. RESULTADOS

En este capítulo se dará a conocer los resultados obtenidos a lo largo del proceso del desarrollo del proyecto, en primera instancia se hablará sobre la implementación del sistema de control de riego por goteo, luego del sistema de identificación de maleza con inteligencia artificial, donde se muestra en efecto la adquisición de la data, el entrenamiento y validación alcanzadas por la red neuronal.



4.1 SISTEMA DE CONTROL ON-OFF POR HISTÉRESIS

En la figura 33 se muestra el resultado del sistema de control on-off por histéresis para el riego por goteo, donde se visualiza el paso a paso del desarrollo para el prototipo, en el cual el rango de control es el rango del porcentaje de humedad para el sensor 1, para el caso del sensor 2 el rango del porcentaje de humedad es de 45 a 80%, esto a partir del cumplimiento de las condiciones y teniendo en cuenta las perturbaciones. También se tiene en cuenta que a partir del nivel de agua que mantenga el tanque permitirá el riego, en caso de no estarlo la electroválvula no se acciona.

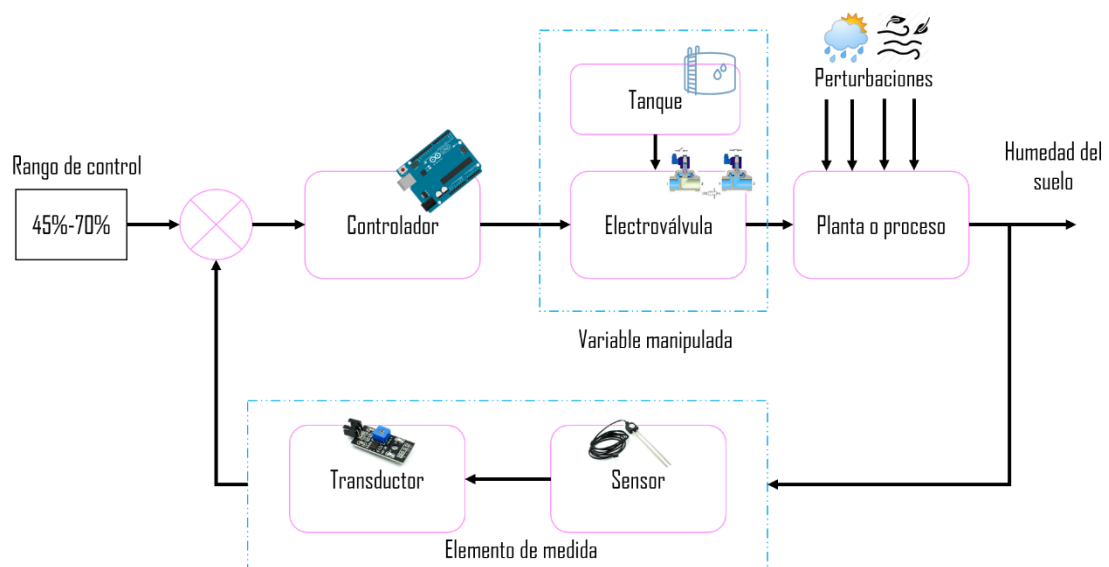


Figura 33. Diagrama en bloques del sistema de control de riego. [Autor]

4.1.1 ADQUISICIÓN DE LA VARIABLE DE TEMPERATURA

En la figura 34 se visualiza la adquisición de la variable de temperatura, donde se obtuvo los datos que caracterizan la temperatura en un intervalo de tiempo de 20 segundos.



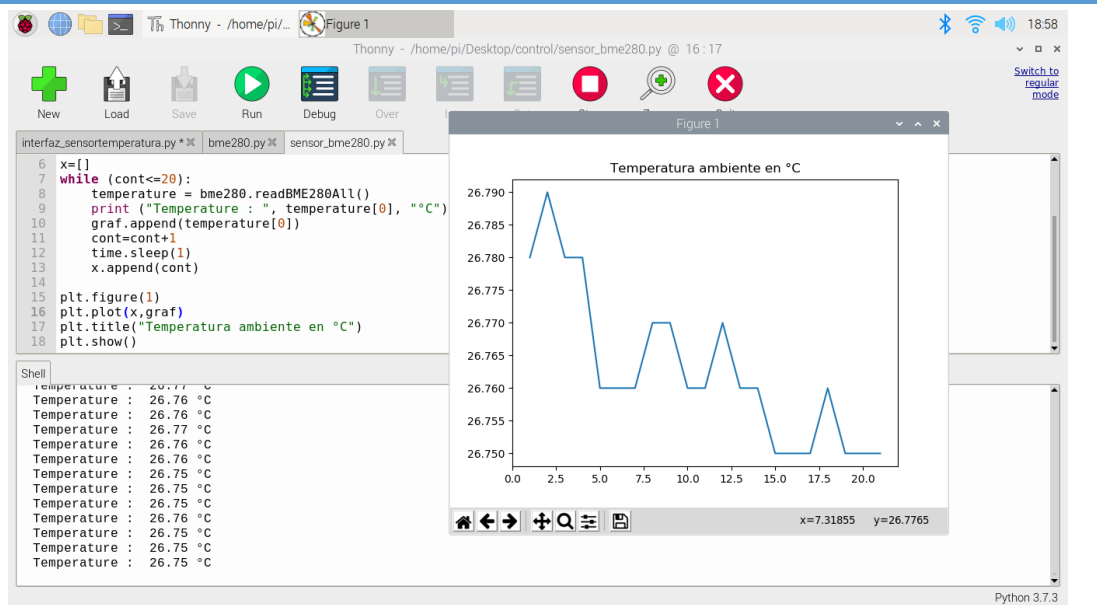


Figura 34. Datos de temperatura ambiente en el cultivo. [Autor]

4.1.2 DISEÑO DE CIRCUITO DE ADQUISICIÓN

Para lograr la adaptación del prototipo en el área del cultivo se necesitaba del circuito impreso, el cual se diseñó en proteus y a partir de ello se realizó la pcb de implementación para la adquisición de los datos y ejecución de la planta.

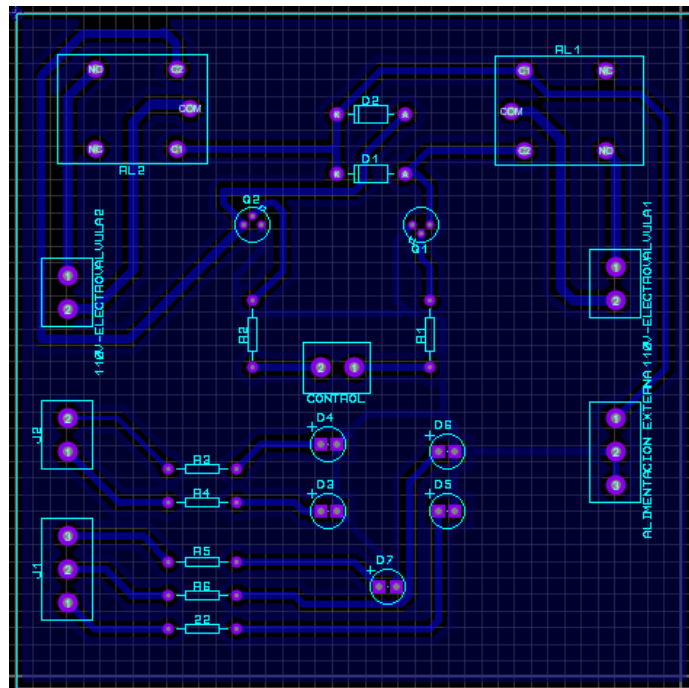


Figura 35. Diseño de PCB para el sistema de control. [Autor]



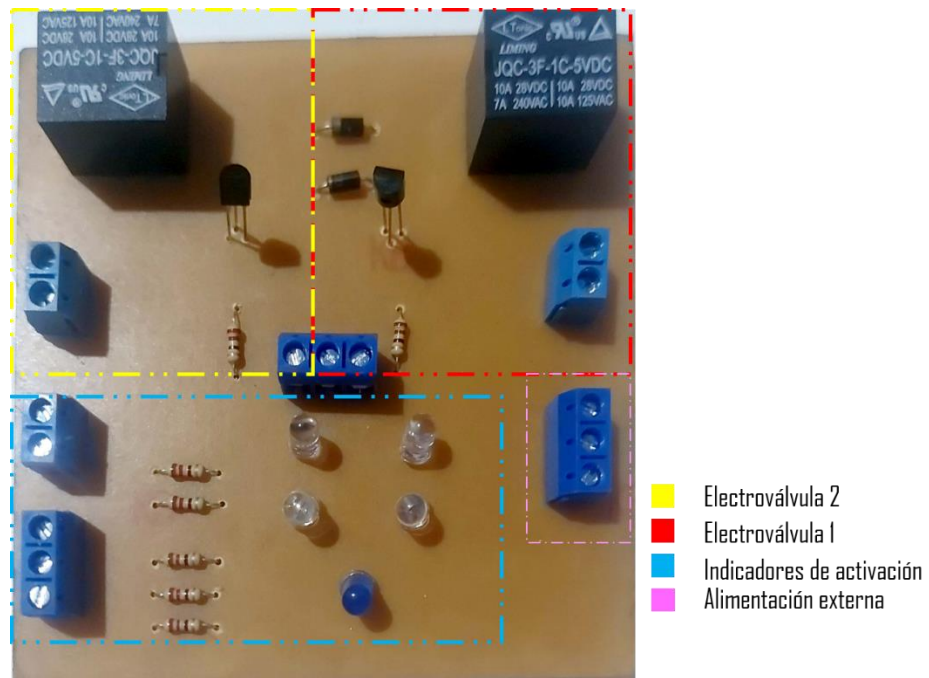


Figura 36. Circuito del sistema de control de riego en pcb. [Autor]

De la figura anterior se puede observar, que la tarjeta se compone de pines de entrada y salida para la conexión de los cables con el controlador, elemento de medida y las electroválvulas, además cuenta con indicadores led de encendido cuando los leds verdes están activos las electroválvulas se encuentran activas, pasa lo contrario cuando están encendidos los leds rojos, además el led azul indicara que el nivel de agua del tanque es bajo y por tanto la electroválvula no va a funcionar.

4.1.3 ADQUISICIÓN DE SEÑALES DEL ELEMENTO DE MEDIDA

Implementando el elemento de medida (figura 36), primero se realizó la adquisición ilustrada en la figura 37 y 38, además se realiza la sintonización de los datos para determinar las condiciones con los que la electroválvula trabaja.



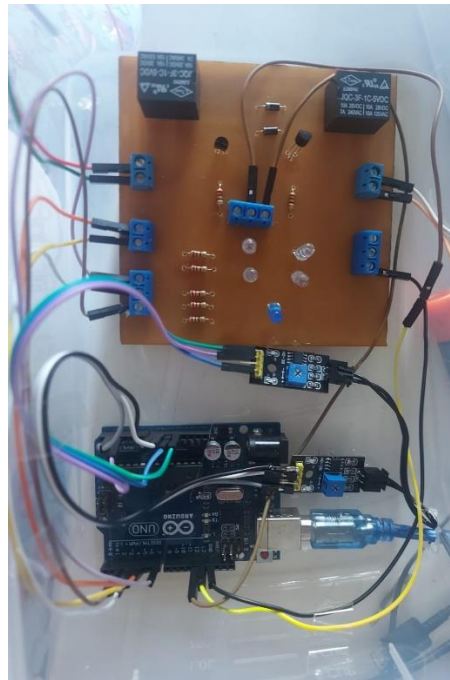


Figura 37. Sistema de adquisición de señales [Autor]



Figura 38. Adquisición de datos del sensor. [Autor]



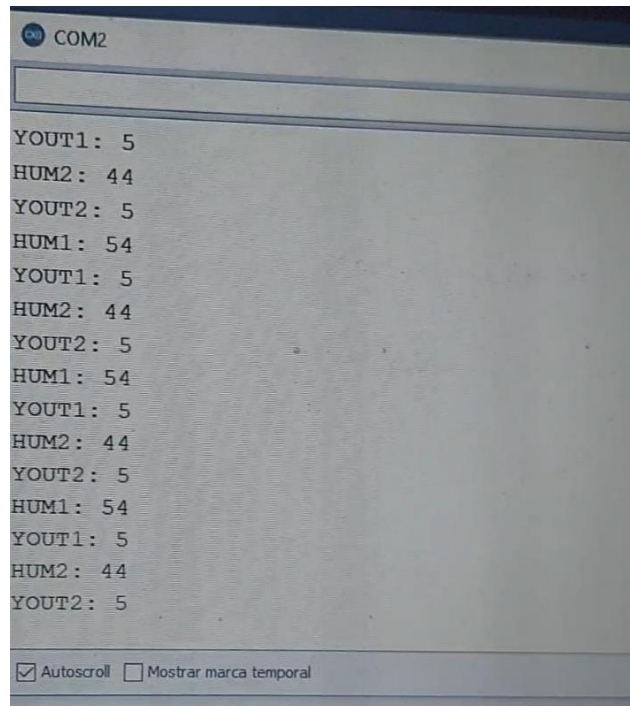


Figura 39. Registro de humedad de la zona del cultivo. [Autor]

4.1.4 IMPLEMENTACIÓN DEL SISTEMA DE RIEGO POR GOTEO

En la figura 39 se ilustra el tendido de la manguera en el cultivo por medio de la cual se realiza el riego. Además, la implementación del sistema de riego por goteo según el requerimiento hídrico.



Figura 40. Perímetro de riego. [Autor]





Figura 41. Sistema de riego por goteo. [Autor]

4.2 SISTEMA DE IDENTIFICACIÓN DE MALEZA MEDIANTE IA

Para el sistema de identificación de maleza se tuvo en cuenta la adquisición de la data, entrenamiento de la red neuronal donde se tiene las gráficas de aprendizaje, además de la comparación con el modelo que se realiza en la exportación a la Raspberry pi4 y la manera en cómo se comporta en tiempo real tomada por sectores en el área de cultivo.

4.2.1 ADQUISICIÓN DE LA DATA

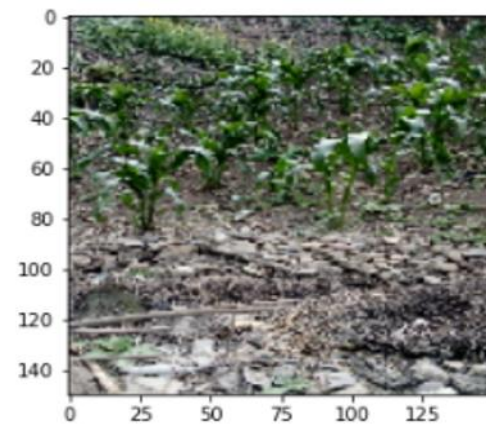
Mediante la adquisición de la data se desarrolló la predicción de la red neuronal con nuevas imágenes, en la figura 37 se muestra la predicción en Google colab de una nueva imagen a la red, hay que tener en cuenta que debe usarse el mismo tratamiento a las imágenes a comparar y el resultado fue que tiene buen aprendizaje por su precisión con las imágenes que no conoce.



```
img = new[79]
plt.imshow(img)
plt.show()
img2= cv2.resize(img,(img_size,img_size))
print(img2.shape)
img2 = (np.expand_dims(img2,0))
print(img2.shape)

print(img.shape)
img = (np.expand_dims(img,0))
print(img2.shape)

predictions_single = model.predict(img)
print(predictions_single)
print(np.sum(predictions_single))
print(np.argmax(predictions_single))
print(class_names[np.argmax(predictions_single)])
```



```
(150, 150, 3)
(1, 150, 150, 3)
(150, 150, 3)
(1, 150, 150, 3)
[[1. 0.]]
1.0
0
sin maleza
```

Figura 42. Predicción de la data. [Autor]

4.2.2 ENTRENAMIENTO DE LA RED NEURONAL

Se obtuvo que el tiempo de entrenamiento no fue mayor de 30 minutos ya que se trata de una red neuronal sencilla, el modelo logró una precisión de entrenamiento y validación del 100% en la época 5.



```
Train on 270 samples
Epoch 1/10
270/270 [=====] - 1s 3ms/sample - loss: 2963.1514 - acc: 0.6741
Epoch 2/10
270/270 [=====] - 1s 2ms/sample - loss: 556.2077 - acc: 0.8852
Epoch 3/10
270/270 [=====] - 1s 2ms/sample - loss: 70.3059 - acc: 0.9593
Epoch 4/10
270/270 [=====] - 1s 2ms/sample - loss: 3.9744 - acc: 0.9963
Epoch 5/10
270/270 [=====] - 1s 2ms/sample - loss: 0.0000e+00 - acc: 1.0000
Epoch 6/10
270/270 [=====] - 1s 2ms/sample - loss: 0.0000e+00 - acc: 1.0000
Epoch 7/10
270/270 [=====] - 1s 2ms/sample - loss: 0.0000e+00 - acc: 1.0000
Epoch 8/10
270/270 [=====] - 1s 2ms/sample - loss: 0.0000e+00 - acc: 1.0000
Epoch 9/10
270/270 [=====] - 1s 2ms/sample - loss: 0.0000e+00 - acc: 1.0000
Epoch 10/10
<tensorflow.python.keras.callbacks.History at 0x7f28055f42d0>
```

Figura 43. Numero de épocas del entrenamiento de la red neuronal. [Autor]

En las figuras 39 y 40 se visualizan las gráficas de aprendizaje de la red neuronal, en la figura 39 se observa el aprendizaje de la red al transcurrir cada época del entrenamiento; Así mismo, en la figura 40 segunda se aprecia la disminución de perdida al transcurrir cada época del entrenamiento. Por tanto, la red neuronal aprendió a diferenciar imágenes con maleza de imágenes sin maleza.

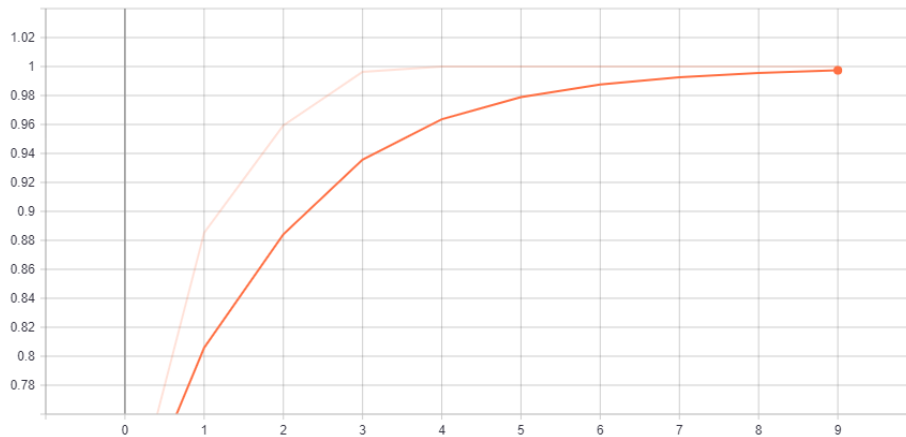


Figura 44. Precisión de las épocas del entrenamiento de la red. [Autor]



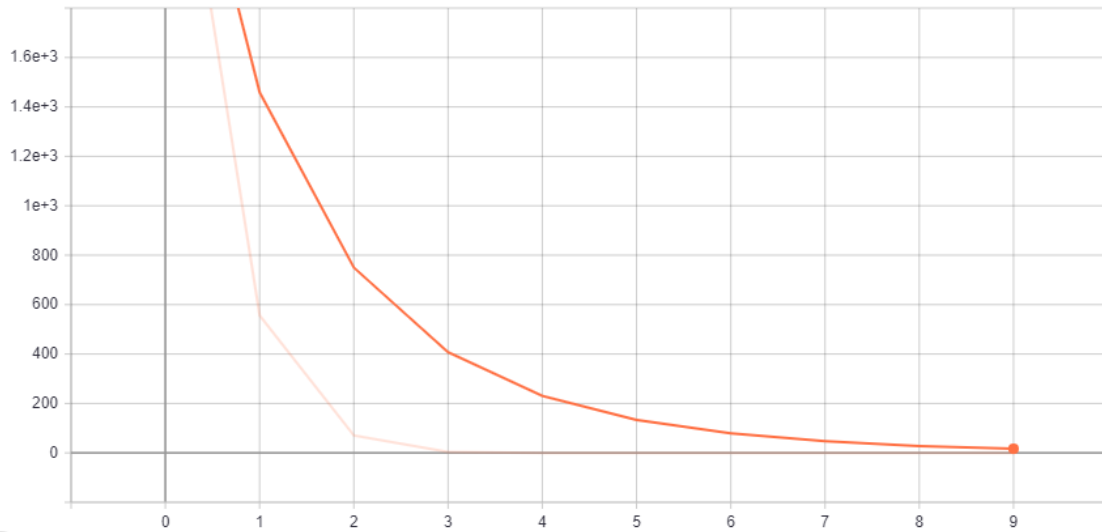


Figura 45. Perdidas de las épocas del entrenamiento de la red. [Autor]

4.2.3 PREDICCIÓN DE IMÁGENES EN EL MODELO DE RPI4.

En la figura 41 se muestra la comparación de una imagen con maleza con el modelo entrenado y exportado en Rapsberry pi4. Donde se comprueba que la red predice nuevas imágenes, se imprimen sus dimensiones, a la clase que pertenece y su precisión.

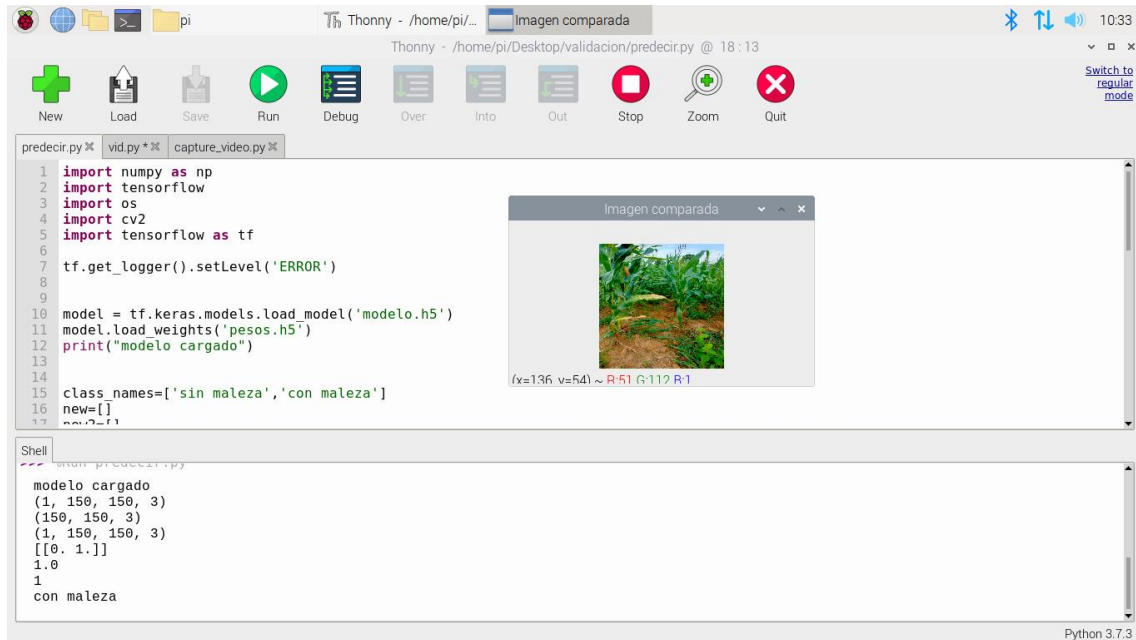


Figura 46. Comparación del modelo con nuevas imágenes. [Autor]



4.2.4 PRUEBA DE NUEVAS IMÁGENES CON LA RED NEURONAL DESDE RPI4.

En la figura 42 y 43 se muestra la predicción de imágenes con maleza con el modelo entrenado y exportado en Raspberry pi4. Donde se verifica la predicción de nuevas imágenes en la red neuronal, y se muestran las características de la misma, sus dimensiones, a la clase que pertenece y su precisión.

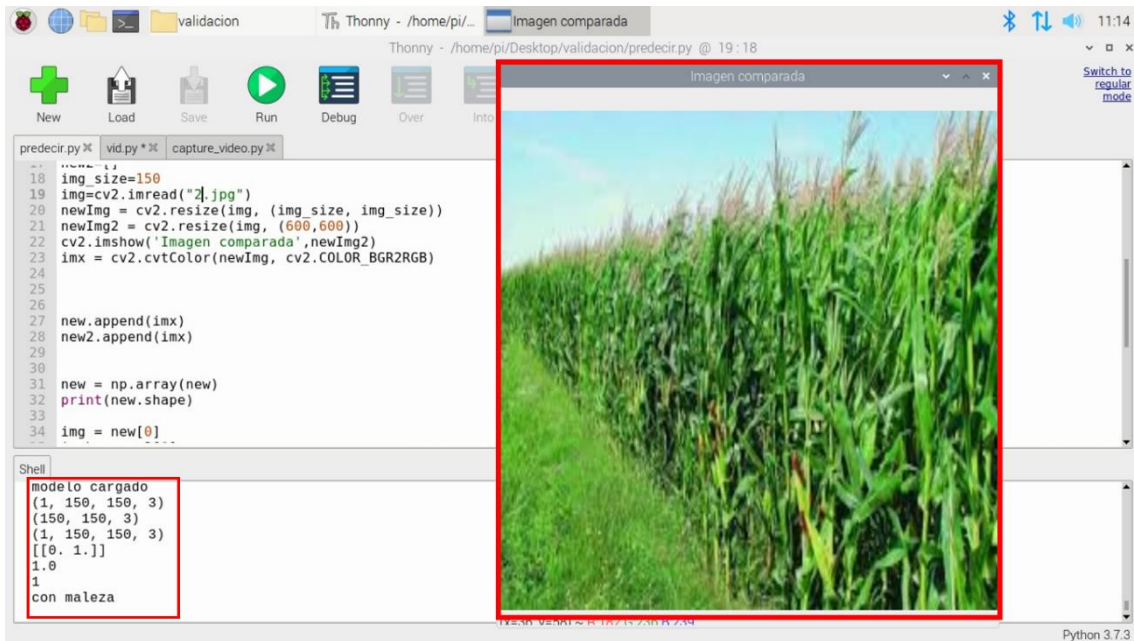


Figura 47. Comparación de imagen con maleza en rpi4. [Autor]

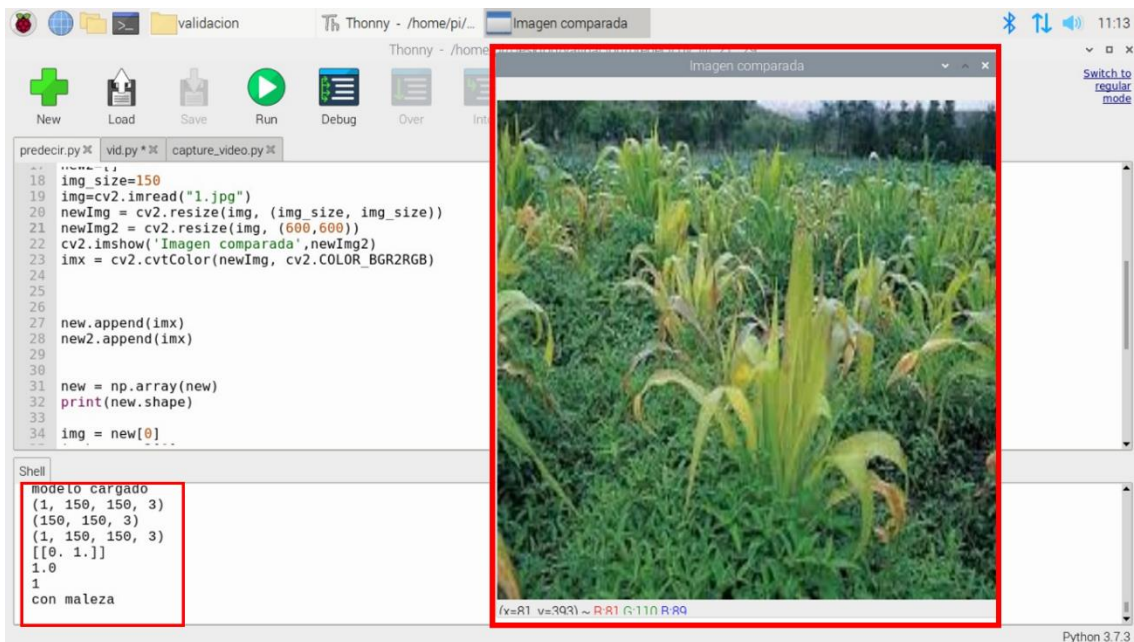


Figura 48. Comparación de imagen con maleza en rpi4. [Autor]



En la figura 44 y 45 se muestra la predicción de imágenes sin maleza con el modelo entrenado y exportado en Raspberry pi4. Donde se verifica la predicción de nuevas imágenes en la red neuronal, y se muestran las características de la misma, sus dimensiones, a la clase que pertenece y su precisión.

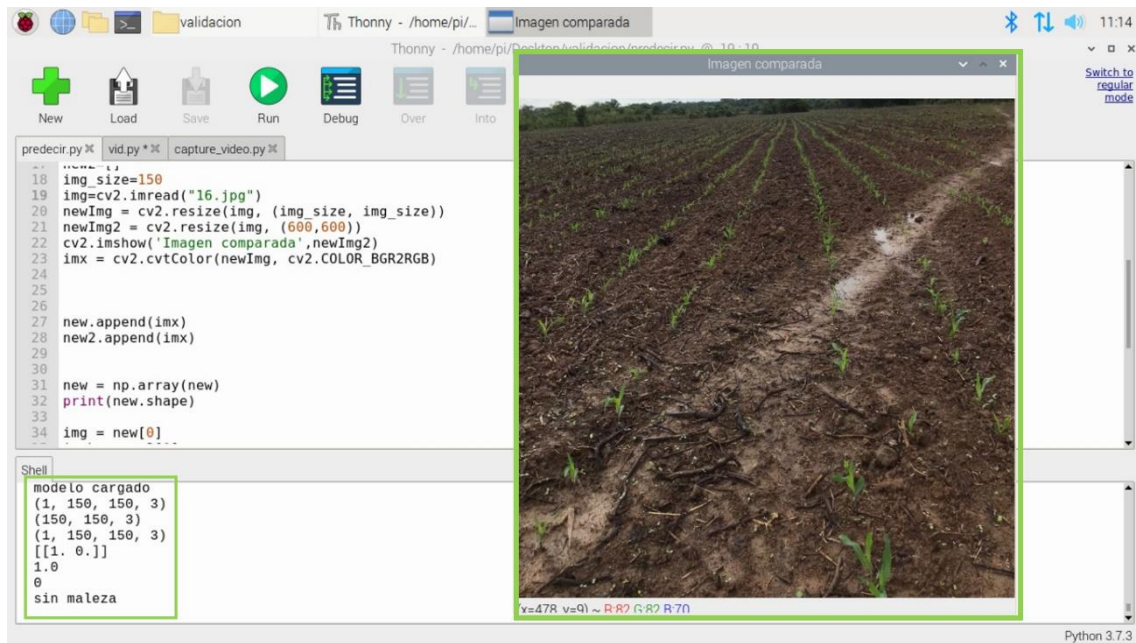


Figura 49. Comparación de imagen sin maleza en rpi4. [Autor]

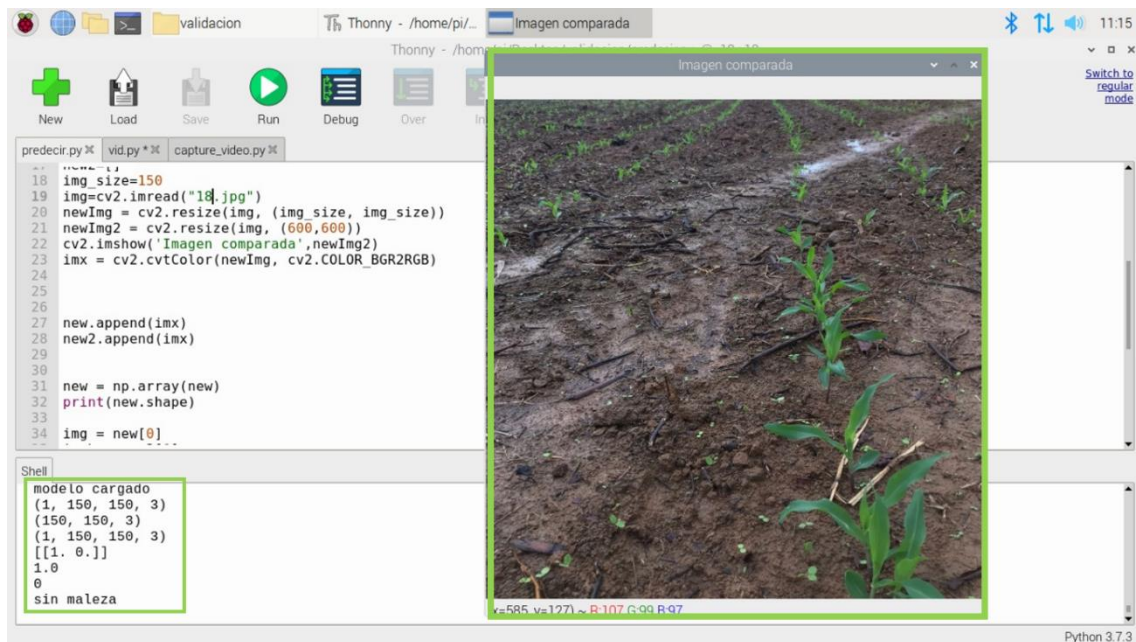
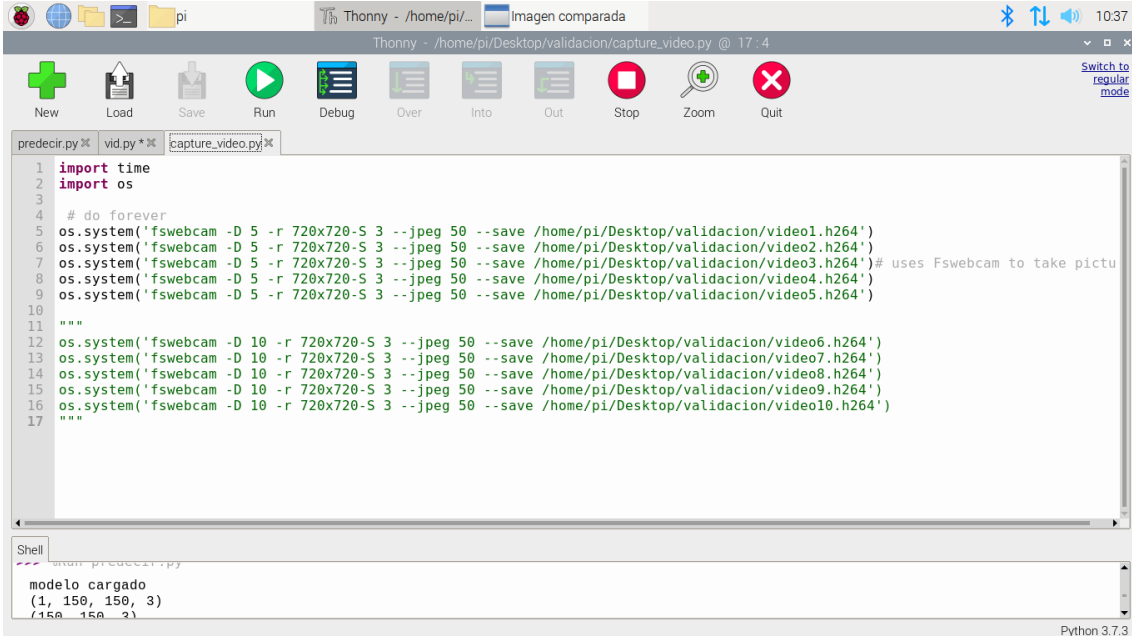


Figura 50. Comparación de imagen sin maleza en rpi4. [Autor]



4.2.5 IDENTIFICACIÓN DE MALEZA POR ZONAS.

En la figura 46 se visualiza el código para tomar 10 videos de 10 segundos de duración por sectores con la cámara USB, para luego cargar los videos y predecir donde se puede encontrar posibles malezas en el cultivo de maíz.



```
1 import time
2 import os
3
4 # do forever
5 os.system('fswebcam -D 5 -r 720x720-S 3 --jpeg 50 --save /home/pi/Desktop/validacion/video1.h264')
6 os.system('fswebcam -D 5 -r 720x720-S 3 --jpeg 50 --save /home/pi/Desktop/validacion/video2.h264')
7 os.system('fswebcam -D 5 -r 720x720-S 3 --jpeg 50 --save /home/pi/Desktop/validacion/video3.h264') # uses Fswebcam to take pictu
8 os.system('fswebcam -D 5 -r 720x720-S 3 --jpeg 50 --save /home/pi/Desktop/validacion/video4.h264')
9 os.system('fswebcam -D 5 -r 720x720-S 3 --jpeg 50 --save /home/pi/Desktop/validacion/video5.h264')
10
11 """
12 os.system('fswebcam -D 10 -r 720x720-S 3 --jpeg 50 --save /home/pi/Desktop/validacion/video6.h264')
13 os.system('fswebcam -D 10 -r 720x720-S 3 --jpeg 50 --save /home/pi/Desktop/validacion/video7.h264')
14 os.system('fswebcam -D 10 -r 720x720-S 3 --jpeg 50 --save /home/pi/Desktop/validacion/video8.h264')
15 os.system('fswebcam -D 10 -r 720x720-S 3 --jpeg 50 --save /home/pi/Desktop/validacion/video9.h264')
16 os.system('fswebcam -D 10 -r 720x720-S 3 --jpeg 50 --save /home/pi/Desktop/validacion/video10.h264')
17 """
```

```
Shell
~/bin/validacion
$ python3 predecir.py
modelo cargado
(1, 150, 150, 3)
(150, 150, 2)
```

Figura 51. Toma de video por zonas. [Autor]

4.3 VALIDACIÓN DEL SISTEMA POR EL EXPERTO



Paz de Ariporo, Casanare.

15 de diciembre de 2021

Declaración de cumplimiento

Jonathan Farney Carvajal Barrios, ingeniero agrónomo identificado con matrícula profesional MP 121019-0527923, me permito comunicar que valore el proyecto DESARROLLO DE UN SISTEMA DE CONTROL DE RIEGO E IDENTIFICACIÓN DE MALEZAS MEDIANTE INTELIGENCIA ARTIFICIAL EN EL MUNICIPIO DE PAZ DE ARIPORO, CASANARE, realizado por Leidy Tatiana Calderón Meza, estudiante de ingeniería electrónica de la universidad de pamplona; validando la implementación del prototipo de bajo costo donde el sistema de riego por goteo es una alternativa de ahorro al agricultor y que preserva el recurso hídrico, así como el sistema de inteligencia artificial ayuda en el diagnóstico de detección de maleza en un cultivo de maíz, además, certifico que se establecieron asesorías para determinar los porcentajes de humedad que se deben cuidar en el cultivo, así como una revisión y clasificación de las imágenes correspondientes a maíz sin maleza y con maleza.

Cordialmente,



Jonathan Farney Carvajal Barrios
M.P. 121019-0527923
C.C. 1005078425

Figura 52. Comprobante de validación por experto. [Autor]



5.

Conclusiones.

5. CONCLUSIONES

- En la caracterización de la variable de humedad se consideró el elemento de sensorica de sonda higrómetro, el cual se adaptó a las necesidades de respuesta para el área 10x10 metros del cultivo, se tomaron los datos de humedad a una profundidad de 50cm bajo tierra, que es la distancia en que la raíz de la planta absorbe el agua, además, cuando se realizó la adquisición no se visualizaron fluctuaciones en el porcentaje de humedad y no fue necesario implementar una etapa de filtrado porque no hay señales externas que modifiquen la respuesta del sensor, era de esperarse porque el convertidor envía datos digitales a la tarjeta y la transmisión de datos esta blindada ante posibles señales analógicas cercanas al sistema. Por otra parte, se consideró la caracterización de la variable de temperatura la cual se tomó los datos de la temperatura ambiente en el área del cultivo, pero debido a que la semilla al ser tecnificada se adapta a las condiciones climáticas que la zona mantiene para el cultivo de maíz no fue necesario aplicar la variable en el lazo de control.
- Considerando las respuestas del sistema se implementó un lazo de control cerrado a partir de un control ON-OFF por histéresis debido a que este respondió de manera satisfactoria en un tiempo estimado de 20 a 25 minutos para alcanzar la estabilidad del sistema para el control de riego por goteo, ya que se pudo determinar el rango de humedad en el que la electroválvula se acciona y se consideraron las perturbaciones como la lluvia y el sol. Además, el sistema de control protege la electroválvula porque no se enciende y se apaga de inmediato, al conocer el porcentaje de humedad en el que se debe mantener para activarse o desactivarse y esto se debe a los datos del elemento de medida ya que es un sistema retroalimentado donde la salida influye sobre la entrada.



- Mediante la aplicación de técnicas de IA como las redes neuronales y técnicas para el preprocesamiento de imágenes como visión artificial proporcionadas por OpenCv se obtuvo el 100% de la clasificación. Con la ayuda de la red neuronal se monitoriza el cultivo de maíz mediante un acompañamiento al agricultor sin necesidad de supervisión constante, en el que a través de una alarma que se genera al ser detectada maleza el agricultor se dirige al cultivo a regar el fungicida. Además, no solamente se adapta en cultivos de maíz, también se puede adaptar a otros tipos de cultivos al que se quiera adecuar el sistema en el que se pueden implementar estrategias como la detección de enfermedades de la planta, la detección de hongos, la detección de bacterias y/o babosas.
- Mediante el prototipo tecnológico se beneficia el agricultor y el cultivo con ventajas como la preservación del medio ambiente por medio de la acción de los dispositivos sensoriales que dieron a conocer cuando la planta tiene necesidad hídrica y abastecerla en justa medida con el ahorro del agua, ya que por sistemas de control de riego por goteo se disminuye en gran medida el desperdicio del agua sin necesidad de un riego permanente. Además, mediante el mapeo de la cámara USB dirigida por IA se ayudó a proteger el cultivo ya que cubre la zona deseada que en este caso es 10x10 metros.
- El sistema de identificación de malezas se validó mediante asesorías con un ingeniero agrónomo quien dio el aval del funcionamiento en el que se ensayó a través de sectores de un metro, indicando así el lugar donde se encuentra la maleza, también se debe tener en cuenta que este sistema para trabajos futuros se puede implementar para saber dónde se encuentra determinada enfermedad y mediante visión artificial realizar un monitoreo del nivel de humedad, además con el calendario establecer puntos de humedad, es decir, los días en que se debe mantener un porcentaje de humedad ajustado, en este caso además de las medidas adquiridas por el sensor, se tuvo en cuenta las condiciones climáticas dadas por el ideam. Por otra parte, se debe tener en cuenta que para lograr este tipo de sistemas se utilizan los datos de caracterización, adquisición, sintonización, procesamiento, entrenamiento y validación; dependiendo del cultivo al que se quiere adaptar un sistema de control e identificación de malezas.



6. TRABAJOS FUTUROS

Son muchas las aplicaciones que puede tener un sistema de riego e identificación de malezas en los cultivos no solo de maíz, sino también de verduras, frutas y otros vegetales, en cuanto al área de la agricultura la implementación de sistemas de IA ayuda al agricultor y disminuye la mano de obra, además aporta beneficios con la monitorización de los cultivos como por ejemplo, ayuda a la detección y prevención de enfermedades, babosas, insectos y/o aves que afecten el progreso del cultivo para que se tomen medidas inteligentes; con la utilización de sensores, drones, tractores que realicen las tareas para mantener un cultivo sano. También tener un control de riego ayuda a mantener las plantaciones en mejores condiciones cuando requieren abastecimiento hídrico y se presentan épocas inesperadas de verano. Para ello es importante realizar la caracterización del cultivo al que se va aplicar sistemas de riego por goteo para controlar la distribución del agua y evitar los desperdicios, ya que son sistemas amigables con el medio ambiente porque el riego va directamente a la planta sin necesidad de malgastar el recurso hídrico.



7. REFERENCIAS

- [1] J. G. Huilca Salcedo, P. F. Sichique Velecela, “Diseño e implementación de un sistema embebido de monitoreo de las variables climáticas para plantaciones de maíz”, Proyecto de grado, Fac. Ingeniería, Prog. Ingeniería Electrónica, Univ. Politécnica salesiana, Cuenca, Ecuador, 2019.
- [2] M.G. Vásquez Rueda, F.G. Flórez García, H.A. Moreno Casillas, “Redes neuronales aplicadas al control de riego usando instrumentación y análisis de imágenes para un microinvernadero aplicado al cultivo de Albahaca”, *Research in Computing Science*, vol. 5 N°. 5, pp. 93-103, Abr 14, 2018. Doi: 10.13053/rcs-147-5-7.
- [3] P.F. Martín Gómez, J.O. Montoya Gómez, G.E. Muñoz Serrano, “Sistema de riego autónomo para pequeños cultivos basado en medición de temperatura y humedad”, *Revista politécnica*, Vol. 13 N° 25 pp. 65-74, Jul-Dic 2017. [En línea], Disponible en: <https://revistas.elpoli.edu.co/index.php/pol/article/view/1141/1476>.
- [4] A. F. Montenegro Bermúdez, C.D. Parada Rojas, “Diseño e implementación de un sistema de detección de malezas en cultivos Cundiboyacenses”, *Proyecto de grado*, Fac. de Ingeniería, programa de electrónica y telecomunicaciones, Bogotá, Cundinamarca, Colombia. 2015, p. 74. [En línea]. Disponible en: <https://repositorioslatinoamericanos.uchile.cl/handle/2250/88562>
- [5] J.A. Villabona, C.L. Sandoval, “Tecnificación para el control de presión y emulación de riego usando el control lógico programable y software Totally Integrated Automation”, *Revista Colombiana de Tecnologías de Avanzada*, Vol. 2 N° 32, May-Jun 2018. DOI: <https://doi.org/10.24054/16927257.v32.n32.2018.3036>.
- [6] V.A. Berrio Meneses, J. Mosquera Téllez, D.F. Álzate Velásquez, “Uso de drones para el análisis de imágenes multiespectrales en agricultura de precisión”, *@LIMENTECH CIENCIA Y TECNOLOGÍA ALIMENTARIA*, Vol. 13 N° 1 pp. 28-40, Abr-May 2015. DOI: <https://doi.org/10.24054/16927125.v1.n1.2015.1647>.
- [7] Infoagro Systems, S.L. (2020). [Internet]. Disponible en: <https://www.infoagro.com/herbaceos/cereales/maiz2.htm>.
- [8] L. Manga (12, Dic, 2020). “Planta tu huerto”. [Internet]. Disponible en: <https://plantatuhuerto.com/etapas-de-crecimiento-del-maiz-y-unidades-de-grado-de-crecimiento/>
- [9] P. F. Martin Gomez, J. O. Montoya Gomez, G. E. Muñoz Serrano “SISTEMA DE RIEGO AUTÓNOMO PARA PEQUEÑOS CULTIVOS BASADO EN MEDICIÓN DE TEMPERATURA Y HUMEDAD”, *revista politécnica*, Vol 13, n°. 25, p. 65-74, Jul-Dic, 2017.
- [10] D. Carvajal Tabares, J.S. Hincapie Ruiz, “PROTOTIPO DE SISTEMA DE CONTROL DE RIEGO PARA UN CULTIVO DE SOYA CON SUPERVISIÓN REMOTA DE HUMEDAD”, proyecto de grado. Fac. de ingeniería, Dpto. de ingeniería electrónica, Univ. San Buenaventura. Santiago de Cali, 2016.
- [11] Información de ARDUINO disponible en su página oficial: <https://docs.arduino.cc/hardware/uno-rev3>



[12] Información de RASPBERRY PI disponible en su página oficial:

<https://www.raspberrypi.com/>

[13] Información de BALENAETCHER disponible en su página oficial:

<https://www.balena.io/etcher/>

[14] Información de FING disponible en su página oficial:

<https://www.fing.com/>

[15] Redacción el tiempo (02/may/2000). “Qué son y cómo funcionan las cámaras web”. [Internet]. Disponible en: <https://www.eltiempo.com/archivo/documento/MAM-1285331>.

[16] información de CAMARA WEB 2K PRO AUTOENFOQUE ULTRA HD MICROFONO TRIPODE disponible en su página oficial: <https://www.mercadolibre.com.co/>

[17] información de SENSOR DE HUMEDAD DE SUELO ANTICORROSIVO – HIGRÓMETRO disponible en su página oficial: <https://electronilab.co/>

[18] Cómo utilizar un sensor de humedad de suelo con Arduino (2021). [Internet]. Disponible en: Sensor de humedad del suelo Módulo de detección de higrómetro de suelo (2013). [internet]. Disponible en: <https://electron-space.blogspot.com/2013/09/soil-moisture-sensor-soil-hygrometer.html>

[19] ELECTRO INDUSTRIA. ¿Cómo funciona una válvula solenoide? [Sitio web]. Providencia, Chile. [Consultado: 5 de mayo de 2021]. Disponible en: <http://www.emb.cl/electroindustria/articulo.mvc?xid=3514&ni=como-funciona-una-valvula-solenoide>

[20] KATSUHIKO OGATA. Ingeniería de control moderna. 5ta ed. Madrid, España. PEARSON EDUCACIÓN, S.A., 2010. 909 p. ISBN: 978-84-8322-660-5

[21] Universidad nacional de Tucumán, Catedra. Acciones de control [Sitio web]. Argentina; [Consultado: 25 de mayo de 2021]. Disponible en: https://catedras.facet.unt.edu.ar/sistemasdecontrol/wpcontent/uploads/sites/101/2016/04/5_Acciones-de-control_2016.pdf

[22] Ponce Cruz, Pedro. Inteligencia artificial: con aplicaciones a la ingeniería 1 ed. México: Alfaomega Grupo Editor, S.A. de C.V., 2010, 378 p. ISBN: 978-607-7854-83-8.

[23] Información de TENSORFLOW disponible en su página oficial:

<https://www.tensorflow.org/>

[24] Redes neuronales (s.f.). “<https://github.com/>”. [sitio web]. Disponible en: https://ml4a.github.io/ml4a/es/neural_networks/

[25] <https://empresas.blogthinkbig.com/las-matematicas-del-machine-learning-funciones-de-activacion/>

[26] Disponible en su página oficial de GOOGLE COLABORATY:

<https://colab.research.google.com/#scrollTo=ISrWNR3MuFUS>

[27] Disponible en su página oficial de PYTHON:



<https://www.python.org/>

[28] Disponible en su página oficial de OPENCV:

https://docs.opencv.org/4.x/d0/de3/tutorial_py_intro.html

8. ANEXOS

A continuación, se muestra el código para la simulación de la planta del sistema de control.

```
const int moistureA0 = A3;

const int ledPin = 13;

int A0 = 0;

int tmp = 0;

int umb_bajo = 0;

int umb_alto = 0;

int YOUT = 0;

int hum = 0;

float vsal = 0 ;

void setup () {
  Serial.begin(9600);
  pinMode(moistureA0, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop () {
  tmp=analogRead( moistureA0 );
  A0=tmp;
  vsal=(5*tmp)/1023.00;
  hum=100-(vsal*20);
  umb_bajo = 45;
  umb_alto = 80;
```



```
if (hum <= umb_bajo) {
  YOUT = 5;
  digitalWrite(ledPin, HIGH);
  Serial.println(hum);
  Serial.println(YOUT);
  delay(100);
}

Serial.println(hum);
Serial.println(YOUT);
delay(100);

if (hum >= umb_alto) {
  while (true){
    YOUT = 0;
    digitalWrite(ledPin, LOW);
    Serial.println(hum);
    Serial.println(YOUT);
    delay(100);
    tmp=analogRead( moistureAO );
    AO=tmp;
    vsal=(5*tmp)/1023.00;
    hum=100-(vsal*20);

    if (hum <= umb_bajo) {
      digitalWrite(ledPin, HIGH);
      Serial.println(hum);
      Serial.println(YOUT);
      delay (100);
      break;
    }
  }
}
```



```
}
}
}
}
```

En la figura 47 se muestra el código en Matlab para obtener las gráficas de la caracterización del sensor 1 y de la misma manera para el sensor 2.

```
clc; clear all; close all;

load('hum.mat');
load('YOUT.mat');

xin = hum;
yout = YOUT;

x1 = xin.signals;
x_entr = x1.values;

y1 = yout.signals;
y_sal = y1.values;

figure (1),
plot(x_entr);

figure (2),
plot(y_sal);
axis([0 10500 -1 5.5]);
```

Figura 53. Código de visualización de gráficas de caracterización del sensor. [Autor]

Tabla 5. Costos y presupuestos para sistema de control de riego. [Autor]

Cantidad	Artículo	Valor Unitario	Valor total
1	Manguera de 100m	68000	68000
20	Unión tipo T	1500	30000
1	Arduino UNO	35000	35000
20	Tapones	500	10000
2	Sensores de sonda higómetro	28000	46000
1	Electroválvula	71100	71100
1	Llave de lavadora	7500	7500



Autor: Leidy Tatiana Calderón Meza
Director: Marleny Fernández Sandoval

DESARROLLO DE UN SISTEMA DE CONTROL DE RIEGO E IDENTIFICACIÓN DE MALEZAS MEDIANTE INTELIGENCIA ARTIFICIAL PARA CULTIVOS DE MAÍZ EN EL MUNICIPIO DE PAZ DE ARIPORO, CASANARE.

1	Teflón	1000	1000
1	Impresión de placas PCB (Relés de activación, transistores, diodos, resistencias, borneras, leds)	30.000	30.000
1	Conexión externa	50.000	50.000
1	Carcasa de protección	5000	5000
TOTAL			353.600

Tabla 6. Costos y presupuestos de sistema de identificación de malezas. [Autor]

Cantidad	Artículo	Valor Unitario	Valor total
1	Raspberry con accesorios (ventilador, cable HDMI, fuente de alimentación, teclado y mouse)	450000	450000
1	Cámara USB	120000	120000
1	Riel de desplazamiento	3000	3000
2	Motores eléctricos	2500	5000
-	Varios (pistola de silicona, estaño, cautín)	20000	20000
TOTAL			598000

