



**UNIVERSIDAD TÉCNICA DEL NORTE**

**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CARRERA DE INGENIERIA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

**TEMA:**

**“SISTEMA AUTÓNOMO DE MONITOREO Y CONTROL DE CONDICIONES  
AMBIENTALES EN CULTIVOS VERTICALES HORTÍCOLAS URBANOS  
MEDIANTE UNA RED DE SENSORES”**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

**AUTOR: RUIZ GÓMEZ CRISTIAN MANUEL**

**DIRECTOR: MSC. LUIS EDILBERTO SUÁREZ ZAMBRANO**

**ASESOR: MSC. EDGAR ALBERTO MAYA OLALLA**

**ASESOR: MSC. FABIÁN GEOVANNY CUZME RODRÍGUEZ**

**Ibarra-Ecuador**

**2022**



UNIVERSIDAD TÉCNICA DEL NORTE  
BIBLIOTECA UNIVERSITARIA

ii

AUTORIZACIÓN DE USO Y PUBLICACIÓN  
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
<b>Cédula de identidad:</b>	1003659966		
<b>Apellidos y nombres:</b>	Ruiz Gómez Cristian Manuel		
<b>Dirección:</b>	Atuntaqui – Calle González Suárez y Galo Plaza		
<b>Email:</b>	<a href="mailto:cmruizg@utn.edu.ec">cmruizg@utn.edu.ec</a>		
<b>Teléfono fijo:</b>	062907354	<b>Teléfono móvil:</b>	098 811 6697

DATOS DE LA OBRA	
<b>Título:</b>	SISTEMA AUTÓNOMO DE MONITOREO Y CONTROL DE CONDICIONES AMBIENTALES EN CULTIVOS VERTICALES HORTÍCOLAS URBANOS MEDIANTE UNA RED DE SENSORES
<b>Autor:</b>	Ruiz Gómez Cristian Manuel
<b>Fecha:</b>	13/06/2022
<b>SOLO PARA TRABAJO DE GRADO</b>	
<b>Programa:</b>	Pregrado <input type="checkbox"/> X Posgrado
<b>Título por el que opta:</b>	Ingeniero en Electrónica y Redes de Comunicación
<b>Asesor/Director:</b>	MSc. Luis Suárez

## 2. CONSTANCIA

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 21 días del mes de junio de 2022

### EL AUTOR:



Cristian Manuel Ruiz Gómez

C.I. 1003659966



UNIVERSIDAD TÉCNICA DEL NORTE  
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

**CERTIFICACIÓN:**

MAGÍSTER LUIS SUÁREZ, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN  
CERTIFICA:

Que el presente trabajo de Titulación SISTEMA AUTÓNOMO DE MONITOREO Y  
CONTROL DE CONDICIONES AMBIENTALES EN CULTIVOS VERTICALES  
HORTÍCOLAS URBANOS MEDIANTE UNA RED DE SENSORES, ha sido desarrollado  
por el señor Cristian Manuel Ruiz Gómez bajo mi supervisión.

Es todo cuanto puedo certificar en honor a la verdad.

MSc. Luis Suárez Zambrano

**DIRECTOR**

## DEDICATORIA

*Dedico este trabajo, con todo el amor y cariño, a la persona que ha sido mi luz, mi inspiración y mis ganas de seguir adelante, a mi hija, Amelie.*

## AGRADECIMIENTO

*Agradezco profundamente a toda mi familia, por brindarme su apoyo incondicional en los momentos que más necesité, sin duda, ha sido lo que me impulsado como persona y como profesional, a mi madre, quien siempre se esforzó por darme lo necesario, a mi hija, por darme la fuerza e inspiración para culminar esta etapa tan importante de mi vida, a Estefanía, mi pareja, por su gran amor y apoyo en estos últimos.*

*A los amigos con los que compartí mis años de formación: Alfredo, Max, Ángel, Kevin O., Kevin E., Max, Paúl, Mateo, Johanna, Adri, Juan, Mishel, July, Caro, Emi, Ronni; sin duda alguna, no pude tener mejores compañeros para esta inolvidable etapa tan llena de emociones y grandiosos momentos, así como también a todos los amigos que han sido parte de este proceso.*

*A la Universidad Técnica del Norte y a los docentes que han sido parte de mi formación profesional, quienes, de una manera muy generosa y desinteresada, han compartido sus conocimientos conmigo, en especial a mi director, quien ha sido el principal apoyo para el desarrollo de este trabajo.*

## Índice de contenidos

<b>1</b>	<b>CAPÍTULO I. ANTECEDENTES .....</b>	<b>1</b>
1.1.	<b>Introducción .....</b>	<b>1</b>
1.2.	<b>Tema.....</b>	<b>1</b>
1.3.	<b>Problema.....</b>	<b>1</b>
1.4.	<b>Objetivos .....</b>	<b>3</b>
1.4.1.	Objetivo general.....	3
1.4.2.	Objetivos específicos .....	3
1.5.	<b>Alcance.....</b>	<b>4</b>
1.6.	<b>Justificación.....</b>	<b>5</b>
<b>2</b>	<b>CAPÍTULO II. MARCO TEÓRICO.....</b>	<b>7</b>
2.1.	<b>Introducción .....</b>	<b>7</b>
2.2.	<b>Horticultura.....</b>	<b>7</b>
2.2.1.	Cultivos hortícolas .....	8
2.2.2.	Características de los cultivos hortícolas .....	9
2.3.	<b>Sistemas de riego.....</b>	<b>10</b>
2.4.	<b>Cultivos verticales .....</b>	<b>18</b>
2.5.	<b>Variedad de hortalizas.....</b>	<b>19</b>
2.6.	<b>Agricultura de precisión.....</b>	<b>21</b>
2.6.1.	Etapas de la agricultura de precisión .....	22
2.6.2.	Herramientas para la agricultura de precisión .....	22
2.7.	<b>Machine Learning.....</b>	<b>24</b>
2.7.1.	Algoritmos de aprendizaje supervisado .....	25
2.7.2.	Métricas de desempeño.....	27
2.8.	<b>Sistemas embebidos .....</b>	<b>30</b>

2.8.1.	Características de un sistema embebido .....	30
2.8.2.	Estructura básica de un sistema embebido.....	31
2.8.3.	Microcontroladores .....	31
<b>2.9.</b>	<b>Protocolos y estándar.....</b>	<b>34</b>
2.9.1.	Modelo OSI.....	35
2.9.2.	Modelo TCP/IP .....	36
2.9.3.	IEEE 802.11 .....	37
2.9.4.	Direccionamiento IPV4 .....	44
2.9.5.	MQTT .....	46
<b>3</b>	<b>CAPÍTULO III. DESARROLLO DE LA PROPUESTA .....</b>	<b>51</b>
<b>3.1.</b>	<b>Introducción .....</b>	<b>51</b>
<b>3.2.</b>	<b>Metodología de diseño .....</b>	<b>51</b>
<b>3.3.</b>	<b>Análisis.....</b>	<b>53</b>
3.3.1.	Situación actual.....	54
3.3.2.	Propósito y descripción general del sistema .....	55
3.3.3.	Análisis de los requerimientos .....	56
3.3.4.	Alcance de la propuesta .....	61
<b>3.4.</b>	<b>Desarrollo de la propuesta .....</b>	<b>61</b>
3.4.1.	Selección del cultivo .....	62
3.4.2.	Selección del Hardware .....	65
3.4.3.	Selección de software .....	74
3.4.4.	Diagramas de flujo.....	77
3.4.5.	Arquitectura del sistema .....	82
3.4.6.	Diagrama de bloques del sistema.....	83
3.4.7.	Construcción de la estructura de agricultura vertical.....	85

3.4.8.	Cálculo de la fuente de alimentación .....	87
3.4.9.	Prototipo electrónico .....	88
3.4.10.	Diseño del circuito impreso (PCB) .....	90
3.4.11.	Calibración de los sensores .....	91
3.4.12.	Configuración del Nodo central .....	94
3.4.13.	Diseño de base de datos local .....	100
3.4.14.	Diseño de la base de datos en tiempo real .....	101
3.4.15.	Programación de los Nodos .....	104
3.4.16.	Recolección de datos de entrenamiento .....	109
3.4.17.	Tratamiento de los datos .....	113
3.4.18.	Selección del algoritmo de aprendizaje .....	117
3.4.19.	Desarrollo de la aplicación de monitoreo y control .....	127
<b>4</b>	<b>CAPÍTULO IV. IMPLEMENTACIÓN Y RESULTADOS.....</b>	<b>137</b>
<b>4.1.</b>	<b>Introducción .....</b>	<b>137</b>
<b>4.2.</b>	<b>Implementación.....</b>	<b>137</b>
4.2.1.	Implementación del cultivo.....	138
4.2.2.	Implementación Nodo Central .....	138
4.2.3.	Implementación Nodos Sensores .....	142
<b>4.3.</b>	<b>Pruebas de funcionamiento.....</b>	<b>150</b>
4.3.1.	Pruebas de funcionamiento Nodo Central .....	151
4.3.2.	Pruebas de funcionamiento del Nodo Sensor .....	156
4.3.3.	Pruebas de funcionamiento de los sensores .....	159
4.3.4.	Pruebas de funcionamiento de los actuadores .....	160
4.3.5.	Pruebas de funcionamiento aplicación móvil .....	162
<b>4.4.</b>	<b>Análisis de resultados .....</b>	<b>163</b>

4.4.1.	Análisis de la red.....	164
4.4.2.	Análisis de resultados en el ambiente no controlado .....	174
4.4.3.	Análisis de resultados en el ambiente controlado .....	177
4.4.4.	Análisis comparativo .....	179
<b>4.5.</b>	<b>Costos del sistema .....</b>	<b>181</b>
<b>4.6.</b>	<b>Discusión de resultados .....</b>	<b>184</b>

### Índice de figuras

FIGURA 1	HORTICULTURA .....	7
FIGURA 2	RIEGO POR SUPERFICIE .....	11
FIGURA 3	RIEGO TENDIDO.....	12
FIGURA 4	RIEGO POR SURCOS .....	13
FIGURA 5	RIEGO POR MELGAS.....	14
FIGURA 6	RIEGO PRESURIZADO .....	15
FIGURA 7	RIEGO POR ASPERSIÓN .....	16
FIGURA 8	RIEGO POR MICROASPERSIÓN .....	17
FIGURA 9	RIEGO POR GOTEO .....	18
FIGURA 10	MATRIZ DE CONFUSIÓN.....	28
FIGURA 11	DIAGRAMA DE BLOQUES DE UN MICROCONTROLADOR .....	32
FIGURA 12	CAPAS MODELO OSI VS TCP/IP .....	37
FIGURA 13	CABECERA SUBCAPA MAC 802.11 .....	38
FIGURA 14	TRAMA PLCP 802.11.....	40
FIGURA 15	CANALES Y FRECUENCIAS DEL PROTOCOLO 802.11B .....	40
FIGURA 16	MODULACIONES 802.11G.....	42
FIGURA 17	MCS 802.11N .....	44
FIGURA 18	CABECERA PAQUETE IP .....	45
FIGURA 19	EJEMPLO COMUNICACIÓN MQTT .....	47
FIGURA 20	UBICACIÓN DE MQTT EN EL MODELO OSI.....	48
FIGURA 21	CABECERA FIJA MQTT.....	48
FIGURA 22	TIPOS DE MENSAJES MQTT .....	49
FIGURA 23	PAQUETES DE CONTROL DE MQTT .....	50
FIGURA 24	PAQUETES DE CONTROL QUE CONTIENEN CARGA ÚTIL .....	50
FIGURA 25	METODOLOGÍA INVESTIGACIÓN ACCIÓN .....	52
FIGURA 26	DIAGRAMA DE BLOQUES DE LA PROPUESTA .....	56
FIGURA 27	RASBERRY PI III.....	66
FIGURA 28	NODE MCU .....	67
FIGURA 29	DHT 11 SENSOR DE TEMPERATURA Y HUMEDAD .....	68

FIGURA 30 SENSOR FC 28.....	69
FIGURA 31 SENSOR DE LUMINOSIDAD LDR .....	71
FIGURA 32 LÁMPARAS DE CRECIMIENTO SINJIALIGHT 300W .....	72
FIGURA 33 ELECTROVÁLVULA 110V AC.....	73
FIGURA 34 VENTILADOR 5V DC.....	73
FIGURA 35 INTERFAZ R STUDIO.....	76
FIGURA 36 DIAGRAMA DE FLUJO GLOBAL.....	78
FIGURA 37 DIAGRAMA DE FLUJO DE LOS NODOS .....	79
FIGURA 38 DIAGRAMA DEL NODO CENTRAL .....	80
FIGURA 39 DIAGRAMA DE FLUJO DE LA APLICACIÓN MÓVIL .....	81
FIGURA 40 ARQUITECTURA DEL SISTEMA .....	83
FIGURA 41 DIAGRAMA DE BLOQUES GENERAL DEL SISTEMA .....	84
FIGURA 42 DISEÑO DE INVERNADERO .....	86
FIGURA 43 MEDICIÓN DE LA CONDUCTIVIDAD ELÉCTRICA DEL CULTIVO .....	87
FIGURA 44 DIAGRAMA FÍSICO DEL HARDWARE DE LOS NODOS .....	88
FIGURA 45 DISEÑO PCB .....	90
FIGURA 46 SVG UTILIZADA PARA EL GRABADO DE LA BAQUELITA.....	91
FIGURA 47 RESULTADO DEL DISEÑO DE HARDWARE.....	91
FIGURA 48 MEDICIÓN SUELO BAJO EN HUMEDAD .....	92
FIGURA 49 MEDICIÓN SUELO CON SUFICIENTE HUMEDAD.....	92
FIGURA 50 MEDICIÓN SUELO CON EXCESO DE HUMEDAD.....	93
FIGURA 51 GRÁFICO DE REGRESIÓN EXPONENCIAL LDR VS LUXES .....	94
FIGURA 52 NODO CENTRAL.....	95
FIGURA 53 INSTALACIÓN DE RASPBIAN .....	95
FIGURA 54 ESCRITORIO DE RASPBIAN .....	96
FIGURA 55 ASIGNACIÓN DE DIRECCIÓN IP FIJA AL NODO CENTRAL.....	96
FIGURA 56 INSTALACIÓN BROKER MQTT .....	97
FIGURA 57 VERIFICACIÓN DEL SERVICIO MOSQUITTO .....	97
FIGURA 58 CONFIGURACIÓN DE MOSQUITTO .....	98
FIGURA 59 INSTALACIÓN DEL SERVIDOR DE BASE DE DATOS .....	98
FIGURA 60 COMPROBACIÓN DEL SERVIDOR DE BASE DE DATOS.....	98
FIGURA 61 CAMBIO DE CONTRASEÑA ROOT DE MYSQL .....	99
FIGURA 62 INSTALACIÓN DE NODE-RED.....	99
FIGURA 63 VERIFICACIÓN DEL SERVICIO DE NODE-RED.....	100
FIGURA 64 MODELO ENTIDAD RELACIÓN DE LA BASE DE DATOS MYSQL .....	101
FIGURA 65 CREACIÓN DEL PROYECTO EN FIREBASE.....	102
FIGURA 66 CREACIÓN BASE DE DATOS FIREBASE REALTIME .....	102
FIGURA 67 MODELO DE BASE DE DATOS FIREBASE REALTIME EN FORMATO JSON .....	103
FIGURA 68 IMPORTACIÓN DE ARCHIVO JSON EN FIREBASE REALTIME .....	104
FIGURA 69 PROGRAMACIÓN DE LOS NODOS - LIBRERÍAS .....	105
FIGURA 70 PROGRAMACIÓN DE LOS NODOS – VARIABLES GLOBALES.....	105
FIGURA 71 PROGRAMACIÓN DE LOS NODOS – SETUP .....	106
FIGURA 72 PROGRAMACIÓN DE LOS NODOS – CONEXIÓN WIFI.....	106

FIGURA 73 PROGRAMACIÓN DE LOS NODOS - FUNCIÓN A EJECUTAR AL LLEGAR UN MENSAJE MQTT .....	107
FIGURA 74 PROGRAMACIÓN DE LOS NODOS – FUNCIÓN DE BUCLE DEL NODO.....	108
FIGURA 75 PROGRAMACIÓN DE LOS NODOS – CONEXIÓN Y SUSCRIPCIÓN MQTT.....	108
FIGURA 76 INSTALACIÓN DEL NODO PARA CONEXIÓN CON BASE DE DATOS MYSQL.....	109
FIGURA 77 CONFIGURACIÓN DEL NODO PARA ESTABLECER CONEXIÓN CON MYSQL.....	109
FIGURA 78 CONFIGURACIÓN DEL TOPIC DE RECOLECCIÓN DE DATOS.....	110
FIGURA 79 CONFIGURACIÓN DEL BROKER DE NODE-RED .....	110
FIGURA 80 FLUJO DE RECOLECCIÓN DE DATOS EN NODE-RED.....	111
FIGURA 81 INTEGRACIÓN DEL NODO PARA LA RECOLECCIÓN DE DATOS .....	112
FIGURA 82 LECTURA DE LA BASE DE DATOS DE ENTRENAMIENTO EN RSTUDIO .....	113
FIGURA 83 ANÁLISIS Y REDUCCIÓN DE LA BASE DE DATOS EN RSTUDIO .....	116
FIGURA 84 GRÁFICO MULTIVARIABLE DE LA BASE DE DATOS REDUCIDA .....	117
FIGURA 85 DIAGRAMA DE FLUJO PARA LA SELECCIÓN DEL ALGORITMO DE APRENDIZAJE .....	118
FIGURA 86 FLUJO PARA LA CREACIÓN DEL DATASET .....	121
FIGURA 87 CONFIGURACIÓN PARA LA CREACIÓN DEL DATASET.....	121
FIGURA 88 FLUJOS DE ENTRENAMIENTO KNN Y DT .....	122
FIGURA 89 CONFIGURACIÓN DEL NODO “LOAD DATASET”.....	122
FIGURA 90 CONFIGURACIÓN DEL NODO “K NEIGHBORS CLASSIFIER” PARA KNN-3.....	123
FIGURA 91 CONFIGURACIÓN DEL NODO “K NEIGHBORS CLASSIFIER” PARA KNN-5.....	123
FIGURA 92 CONFIGURACIÓN DEL NODO “DECISION TREE CLASSIFIER”.....	124
FIGURA 93 FLUJO DE EVALUACIÓN DE LOS ALGORITMOS.....	125
FIGURA 94 CONFIGURACIÓN DE LOS NODOS DE PREDICCIÓN.....	125
FIGURA 95 CONFIGURACIÓN DEL NODO “ASSESSMENT” PARA OBTENER PORCENTAJE DE EXACTITUD .....	125
FIGURA 96 CONFIGURACIÓN DEL NODO “ASSESSMENT” PARA OBTENER MATRIZ DE CONFUSIÓN .....	126
FIGURA 97 SELECCIÓN DEL FRAMEWORK DE DESARROLLO.....	129
FIGURA 98 CONFIGURACIÓN DEL SDK PROPORCIONADA POR LA CONSOLA DE FIREBASE .....	129
FIGURA 99 CONFIGURACIÓN DE IONIC PARA LA COMUNICACIÓN CON FIREBASE .....	130
FIGURA 100 FUNCIONES PARA LA LECTURA Y ESCRITURA DE DATOS EN FIREBASE.....	130
FIGURA 101 INICIO/PORTADA DE LA APLICACIÓN.....	132
FIGURA 102 MENÚ DE LA APLICACIÓN.....	132
FIGURA 103 PANTALLA DE MONITOREO DE LA APLICACIÓN.....	133
FIGURA 104 PANTALLA DE CONTROL DE LA APLICACIÓN.....	133
FIGURA 105 INSTALACIÓN DE NODOS PARA LA COMUNICACIÓN CON FIREBASE.....	134
FIGURA 106 FLUJO PARA RECOLECCIÓN DE DATOS Y MONITOREO .....	135
FIGURA 107 CONFIGURACIÓN DEL NODO DE TEMPERATURA.....	135
FIGURA 108 FLUJO PARA LA SINCRONIZACIÓN CON FIREBASE .....	136
FIGURA 109 IMPLEMENTACIÓN DEL CULTIVO .....	138
FIGURA 110 INTEGRACIÓN DEL FLUJO DE CLASIFICACIÓN.....	139
FIGURA 111 CONFIGURACIÓN NODO DE CLASIFICACIÓN .....	140
FIGURA 112 FLUJO DE ACTIVACIÓN DE ACTUADORES .....	141
FIGURA 113 SCRIPT DE ACTIVACIÓN DE ACTUADORES .....	142

FIGURA 114 UBICACIÓN DEL NODO SENSOR.....	143
FIGURA 115 UBICACIÓN DEL SENSOR DHT11.....	143
FIGURA 116 UBICACIÓN DEL SENSOR FC-28.....	144
FIGURA 117 UBICACIÓN DEL SENSOR LDR.....	144
FIGURA 118 VISTA SUPERIOR SISTEMA VERTICAL.....	146
FIGURA 119 VISTA LATERAL SISTEMA VERTICAL.....	146
FIGURA 120 UBICACIÓN ARNÉS EN EL SISTEMA VERTICAL.....	148
FIGURA 121 UBICACIÓN VENTILADORES.....	148
FIGURA 122 INSTALACIÓN SISTEMA DE RIEGO.....	149
FIGURA 123 UBICACIÓN DUCTOS DE RIEGO.....	150
FIGURA 124 SISTEMA FINAL.....	150
FIGURA 125 VERIFICACIÓN DIRECCIÓN IP NODO CENTRAL.....	152
FIGURA 126 VERIFICACIÓN DE CONECTIVIDAD A INTERNET NODO CENTRAL.....	152
FIGURA 127 PRUEBA DE SUSCRIPCIÓN MQTT.....	153
FIGURA 128 PRUEBA DE PUBLICACIÓN MQTT.....	153
FIGURA 129 VERIFICACIÓN DE RECEPCIÓN MQTT.....	153
FIGURA 130 PRUEBA DE CONEXIÓN A LA BASE DE DATOS MYSQL.....	154
FIGURA 131 PRUEBA DE COMANDO SQL.....	155
FIGURA 132 PRUEBA DE RECEPCIÓN DE MENSAJES MQTT EN NODE-RED.....	156
FIGURA 133 PRUEBA DE CLASIFICACIÓN DE DATOS NODE-RED.....	156
FIGURA 134 PRUEBA DE CONECTIVIDAD WIFI NODO SENSOR.....	157
FIGURA 135 PRUEBA DE CONECTIVIDAD DE RED NODO SENSOR.....	158
FIGURA 136 PRUEBA DE CONECTIVIDAD MQTT NODO SENSOR.....	158
FIGURA 137 PRUEBA DE ENVÍO DE MENSAJE MQTT AL NODO SENSOR.....	159
FIGURA 138 PRUEBA DE RECEPCIÓN DE MENSAJE MQTT NODO SENSOR.....	159
FIGURA 139 PRUEBA DE LECTURA DE SENSORES DESDE NODE-RED.....	160
FIGURA 140 FLUJO DE PRUEBA PARA ACTIVACIÓN DE ACTUADORES NODE-RED.....	161
FIGURA 141 VERIFICACIÓN DE ACTIVACIÓN DE SISTEMA DE RIEGO.....	161
FIGURA 142 VERIFICACIÓN DE ACTIVACIÓN DE SISTEMA DE VENTILACIÓN.....	161
FIGURA 143 VERIFICACIÓN DE ACTIVACIÓN DE SISTEMA DE ILUMINACIÓN.....	162
FIGURA 144 VERIFICACIÓN DE LECTURA DE DATOS DESDE LA APLICACIÓN MÓVIL.....	163
FIGURA 145 ILUSTRACIÓN DE DISTRIBUCIÓN DE PLANTAS EN CADA NIVEL.....	164
FIGURA 146 CABECERAS MQTT - WIRESHARK.....	165
FIGURA 147 CABECERAS TCP - WIRESHARK.....	166
FIGURA 148 CABECERAS IPV4 - WIRESHARK.....	168
FIGURA 149 CABECERA 802.11 LLC - WIRESHARK.....	169
FIGURA 150 CABECERA 802.11 MAC - WIRESHARK.....	169
FIGURA 151 CABECERAS 802.11 CAPA PHY - WIRESHARK.....	171
FIGURA 152 DURACIÓN DE LA TRAMA - WIRESHARK.....	172
FIGURA 153 SIMULACIÓN DE LA SEÑAL 802.11G.....	173
FIGURA 154 MEDICIÓN LONGITUD DE LA HOJA EN EL AMBIENTE NO CONTROLADO.....	174
FIGURA 155 PLANTAS EN EL AMBIENTE NO CONTROLADO.....	174
FIGURA 156 CRECIMIENTO DE LAS HOJAS EN EL AMBIENTE NO CONTROLADO.....	175
FIGURA 157 FRONDOSIDAD EN EL AMBIENTE NO CONTROLADO.....	176

FIGURA 158 DEBILIDAD DE LAS HOJAS DEBIDO A LA TEMPERATURA DEL AMBIENTE NO CONTROLADO.....	177
FIGURA 159 TAMAÑO DE UNA DE LAS HOJAS DEL AMBIENTE CONTROLADO .....	177
FIGURA 160 CRECIMIENTO DE LAS HOJAS EN EL AMBIENTE CONTROLADO.....	178
FIGURA 161 FRONDOSIDAD EN EL AMBIENTE CONTROLADO .....	179
FIGURA 162 CARACTERÍSTICAS DEL NODEMCU ESP8266 .....	204
FIGURA 163 CONSUMO DE ENERGÍA NODEMCU ESP8266.....	204
FIGURA 164 DEFINICIÓN DE PINES NODEMCU ESP8266 .....	205
FIGURA 165 PARÁMETROS NODEMCU ESP8266.....	206
FIGURA 166 ESPECIFICACIONES GENERALES DHT11 .....	207
FIGURA 167 CONFIGURACIÓN DE PINES DHT11 .....	207
FIGURA 168 ESPECIFICACIONES DETALLADAS DHT11 .....	207
FIGURA 169 PINES DEL MÓDULO FC-28.....	208
FIGURA 170 ESPECIFICACIONES Y CARACTERÍSTICAS MODULO FC-28.....	208
FIGURA 171 CARACTERÍSTICAS DE LAS LÁMPARAS SINJIALIGHT .....	209
FIGURA 172 ESPECTRO DE ILUMINACIÓN DE LAS LÁMPARAS SINJIALIGHT .....	209
FIGURA 173 ÁREA DE ILUMINACIÓN DE LAS LÁMPARAS SINJIALIGHT .....	210
FIGURA 174 PLÁNTULAS DE ACELGA .....	216
FIGURA 175 SEMBRÍO DE CULTIVOS EN EL SISTEMA VERTICAL .....	216
FIGURA 176 MEDICIÓN DE TAMAÑO DE HOJAS EN LA SEMANA 1.....	216
FIGURA 177 IMPLEMENTACIÓN DEL SISTEMA DE RIEGO .....	217
FIGURA 178 PCB NODO SENSOR DENTRO DE CAJA PLÁSTICA .....	217
FIGURA 179 RESULTADOS.....	217

### Índice de tablas

TABLA 1 LISTA DE PARTICIPANTES .....	57
TABLA 2 NOMENCLATURA DE REQUERIMIENTOS .....	57
TABLA 3 REQUERIMIENTOS DE STAKEHOLDERS .....	58
TABLA 4 REQUERIMIENTOS FUNCIONALES DEL SISTEMA .....	59
TABLA 5 REQUERIMIENTOS DE ARQUITECTURA .....	60
TABLA 6 TAXONOMÍA DE LOS CULTIVOS DE HORTALIZAS .....	62
TABLA 7 FICHA TÉCNICA DE LA ACELGA .....	64
TABLA 8 NOMENCLATURA DE EVALUACIÓN DE REQUERIMIENTOS.....	65
TABLA 9 ANÁLISIS COMPARATIVO DEL NODO DE PROCESAMIENTO.....	66
TABLA 10 ANÁLISIS COMPARATIVO DE LOS NODOS SENSORES .....	66
TABLA 11 ANÁLISIS COMPARATIVO DE SENSORES DE HUMEDAD RELATIVA Y TEMPERATURA ..	68
TABLA 12 ANÁLISIS COMPARATIVO DE SENSORES DE HUMEDAD DE SUELO .....	69
TABLA 13 ANÁLISIS COMPARATIVO DE SENSORES DE LUMINOSIDAD .....	70
TABLA 14 ANÁLISIS COMPARATIVO DE LÁMPARAS DE CRECIMIENTO.....	71
TABLA 15 ANÁLISIS COMPARATIVO ELECTROVÁLVULAS .....	72
TABLA 16 ANÁLISIS COMPARATIVO VENTILADORES .....	73
TABLA 17 SELECCIÓN DE SOFTWARE .....	75

TABLA 18 SOFTWARE PARA LA VISUALIZACIÓN DE RESULTADOS .....	76
TABLA 19 FICHA TÉCNICA DE LA ESTRUCTURA DEL INVERNADERO .....	85
TABLA 20 LISTADO DE ELEMENTOS CON ALIMENTACIÓN DE 5V .....	88
TABLA 21 VALORES DEL SENSOR LDR VS MEDICIÓN EN LUXES.....	94
TABLA 22 TABLA DE ETIQUETAS INDIVIDUALES DE LAS VARIABLES AMBIENTALES .....	114
TABLA 23 TABLA DE ETIQUETAS DEL SISTEMA .....	115
TABLA 24 NOMENCLATURA DE CRITERIOS DE EVALUACIÓN .....	118
TABLA 25 RESTRICCIONES PARA LOS CRITERIOS DE EVALUACIÓN .....	119
TABLA 26 RANGOS DE CALIFICACIONES PARA LOS CRITERIOS DE EVALUACIÓN.....	119
TABLA 27 VALORES DE LAS MÉTRICAS PARA CADA ALGORITMO .....	126
TABLA 28 PUNTUACIÓN POR ALGORITMO DE APRENDIZAJE .....	127
TABLA 29 MEDIDAS DEL SISTEMA VERTICAL .....	145
TABLA 30 NOMENCLATURA PARA MENCIÓN DE PLANTAS Y AMBIENTES .....	164
TABLA 31 CÁLCULO DE CABECERAS DE MQTT.....	166
TABLA 32 CÁLCULO DE CABECERAS TCP.....	167
TABLA 33 CÁLCULO DE CABECERAS IPV4.....	168
TABLA 34 CÁLCULO DE CABECERAS 802.11 LLC.....	169
TABLA 35 CÁLCULO DE CABECERAS 802.11 - MAC .....	170
TABLA 36 CÁLCULO DE LA LONGITUD DE LA TRAMA .....	171
TABLA 37 PRODUCCIÓN TOTAL EN GRAMOS DE CADA SUBSISTEMA .....	180
TABLA 38 FRONDOSIDAD PROMEDIO DE CADA SUBSISTEMA .....	180
TABLA 39 TIEMPO DE COSECHA PROMEDIO POR CADA SUBSISTEMA .....	181
TABLA 40 RESUMEN DE COSTOS DE IMPLEMENTACIÓN DEL SISTEMA.....	181
TABLA 41 RESUMEN DE COSTOS DE IMPLEMENTACIÓN DE UN SISTEMA TRADICIONAL.....	182
TABLA 42 COSTOS POR CONSUMO ELÉCTRICO .....	183
TABLA 43 TABULACIÓN DE CRECIMIENTO DE LAS PLANTAS DEL NIVEL 1 .....	211
TABLA 44 TABULACIÓN DE CRECIMIENTO DE LAS PLANTAS DEL NIVEL 2 .....	211
TABLA 45 TABULACIÓN DE FRONDOSIDAD DE LAS PLANTAS DEL NIVEL 1 .....	212
TABLA 46 TABULACIÓN DE FRONDOSIDAD DE LAS PLANTAS DEL NIVEL 2 .....	212
TABLA 47 MATRIZ DE CONFUSIÓN Y CÁLCULO DE MÉTRICAS DE KNN-3.....	213
TABLA 48 MATRIZ DE CONFUSIÓN Y CÁLCULO DE MÉTRICAS DE KNN-5.....	214
TABLA 49 MATRIZ DE CONFUSIÓN Y CÁLCULO DE MÉTRICAS DE ÁRBOLES DE DECISIÓN .....	215
TABLA 50 COSTOS DE IMPLEMENTACIÓN DE LA ESTRUCTURA VERTICAL.....	218
TABLA 51 COSTOS DE IMPLEMENTACIÓN DE LOS NODOS SENSORES.....	218
TABLA 52 COSTOS DE IMPLEMENTACIÓN DEL NODO CENTRAL .....	218
TABLA 53 COSTOS DE IMPLEMENTACIÓN DE SENSORES Y ACTUADORES .....	219
TABLA 54 COSTOS DE IMPLEMENTACIÓN DE LA APLICACIÓN MÓVIL.....	219
TABLA 55 COSTOS DE IMPLEMENTACIÓN DEL SISTEMA ELÉCTRICO .....	219

## **Capítulo I. Antecedentes**

### **1.1. Introducción**

En este capítulo se encuentran detalladas las bases para el desarrollo del presente trabajo de titulación, siendo éstos: el tema, la problemática, los objetivos, el alcance y la justificación, con la finalidad de expresar la importancia de la realización de este proyecto.

### **1.2. Tema**

“SISTEMA AUTÓNOMO DE MONITOREO Y CONTROL DE CONDICIONES AMBIENTALES EN CULTIVOS VERTICALES HORTÍCOLAS URBANOS MEDIANTE UNA RED DE SENSORES”

### **1.3. Problema**

La temática de agricultura es una de las actividades que desde la antigüedad realiza el hombre dedicado a la producción de alimentos para la familia, mayoritaria mente con la implementación de técnicas empíricas que causa en algunos casos disminución en la cantidad de frutos cosechados, con el crecimiento y desarrollo de la sociedad surgen nuevos métodos y técnicas para mejorar el proceso. De manera análoga se enfrentan a problemas de degradación de los suelos que afecta la función de la actividad agrícola y de manera consecuente la producción alimenticia de exportación (Cortes & Vargas, 2020).

Lo mencionado, se acerca a una realidad que en los ámbitos rurales se vive día a día, mientras en los sectores urbanos la actividad agrícola es muy limitada por falta de espacios verdes y terrenos adecuados para los sembríos, por tanto, se puede observar el frecuente uso de diversos elementos para sembrar hortalizas que generalmente son de consumo familiar. Sin embargo, la modernidad en diferentes campos brinda la posibilidad implementar sistemas que mejoran el proceso de la agricultura.

Así la horticultura se genera en las ciudades como parte la solución a las problemáticas en la agricultura, permite realizar prácticas como el uso de técnicas de cultivo vertical como estrategia ante la masiva población mundial a la que se necesita alimentar (Carballo, 2021). Así existe el interés del investigador para desarrollar un sistema embebido automatizado que permita monitorear y controlar las condiciones ambientales de un cultivo vertical, en base a una red de sensores, para mejorar la producción de cultivos hortícolas.

En países como Argentina se analiza el acelerado avance de la urbe y la crisis que ahonda los problemas de la horticultura, para determinar las trasformaciones socio espaciales que también carecen de planificación estratégica, porque es evidente la extinción de las huertas de la urbe, de lo cual también se denota el interés personal de personas particulares, que puede conducir a aspectos de economía para agravar a la consecución de actividades agrícolas (Molpeceres et al., 2019).

En definitiva la implementación de un sistema de monitoreo y control de variables ambientales es posible mitigar casi en su totalidad la afección de las estaciones y los cambios climáticos en las plantaciones, en este caso hortícolas, en este sentido se proporciona el ambiente necesario para el desarrollo de los cultivos, obteniendo beneficios importantes como la producción de alimentos fuera de temporada, evasión de uso de pesticidas, productos más saludables, eliminación del error humano en el suministro de insumos, reducción de intermediarios para la comercialización de productos, entre otros. Mientras que con la agricultura vertical se logra optimizar espacios de cultivo con menor cantidad de suelo y aumentar el rendimiento por unidad de área en la producción.

## 1.4. Objetivos

### 1.4.1. Objetivo general

Desarrollar un sistema embebido automatizado que permita monitorear y controlar las condiciones ambientales de un cultivo vertical, en base a una red de sensores, para mejorar la producción de cultivos hortícolas.

### 1.4.2. Objetivos específicos

- Seleccionar un cultivo hortícola de ciclo corto considerando los fundamentos teóricos y proponiendo un caso de estudio eficiente en tiempo para las diferentes etapas de la implementación del sistema.
- Determinar el hardware y software a utilizar, mediante la realización de un BenchMark basado en los requerimientos del proyecto, seleccionando las herramientas que mejor se adapten a las necesidades del entorno.
- Esquematizar un sistema embebido, basado en redes de sensores y la integración de los requerimientos de hardware y software establecidos, para implementarlo sobre un modelo de cultivo vertical de productos hortícolas que permita la adquisición y manipulación de las variables ambientales con un buen compromiso de resolución y precisión.
- Diseñar un enfoque multicriterio evaluando el desempeño y costos computacionales entre varios algoritmos de aprendizaje, de tal manera que se facilite la automatización y control de las variables ambientales de interés, eficiente y eficazmente, en base a las necesidades del cultivo seleccionado.
- Evaluar el sistema propuesto realizando pruebas y análisis comparativos entre productos obtenidos a través de la agricultura convencional y del mismo, concluyendo con los beneficios y la factibilidad del proyecto.

## 1.5. Alcance

El presente trabajo de titulación tiene como objetivo diseñar e implementar un sistema embebido automatizado de control y monitoreo de variables ambientales en un modelo de agricultura vertical de productos hortícolas, basado en redes de sensores y aprendizaje de máquina, con la finalidad de mejorar la producción dentro y fuera de temporada, y multiplicar el rendimiento por unidad de área en comparación con la agricultura convencional utilizando la metodología Action-Research.

Como paso inicial al desarrollo del proyecto, se realizará una revisión bibliográfica acerca de, el ciclo de crecimiento de productos de horticultura, cuáles son las variables que tienen mayor influencia y los valores recomendados para un desarrollo óptimo de la misma, redes de sensores, sistemas de agricultura de ambiente controlado y las herramientas necesarias para el diseño y desarrollo del sistema. De esta base teórica se elegirá un cultivo hortícola de ciclo corto para las etapas de implementación del sistema, con el objetivo de optimizar la duración de las etapas de pruebas y entrenamiento. Además, se realizará un Benchmark con la finalidad de determinar las herramientas de hardware y software que mejor se acoplen a los requerimientos del entorno y que brinden mejores beneficios de adquisición.

Se planteará un esquema de recolección de datos de las variables ambientales más importantes en la horticultura, a través de una red de sensores y la integración de los requerimientos, los cuáles serán almacenados en una base de datos de software libre, que estará diseñada acorde a las necesidades del sistema y alojada el nodo central de la red de sensores. La implementación del sistema deberá cubrir el control de variables indispensables para el crecimiento de la planta como humedad, temperatura y luminosidad, siendo esta última de gran interés ya que al ser aplicado en un ambiente cerrado será el sistema quien suministre la cantidad, intensidad y tipo de luz necesaria, basándose en estudios previamente realizados y controlando casi en su totalidad, el fotoperiodo del vegetal.

Al obtener una base de datos que, luego de ser sometida a un modelo representativo de sistemas multivariantes, pueda ofrecer un conjunto óptimo de datos para abordar los distintos posibles estados de cada una de las variables, el sistema pasaría a la etapa de entrenamiento, en donde se pretende estudiar varios algoritmos de aprendizaje de máquina y elegir el que mejor se adapte a el entorno o de ser necesario, hacer un híbrido entre los que presenten mejor desenvolvimiento sin perder de vista el coste computacional, con el fin de optimizar el uso de recursos tanto de hardware como de software y por ende el consumo energético.

De la implementación del sistema, se deberá lograr automatización para el cultivo de ciertos productos hortícolas, obteniendo productos de mejor calidad que los que ofrece la agricultura convencional, para esto se realizará un análisis comparativo de los vegetales resultantes de este último y el modelo de agricultura vertical en el que se implementará el sistema embebido.

## **1.6. Justificación**

El acelerado crecimiento de los sectores urbanos propicia la implementación de sistemas tecnológicos para optimizar el uso del suelo, con el objetivo de fomentar el aumento de productividad agrícola dentro de los ambientes controlados o no. En este contexto, se denota que en futuros años la población mundial tendrá un incremento del 80%, mientras en Ecuador se ha generado aumento poblacional que se asienta en sectores urbanos por la concentración del sistema agrario en las ciudades (Bardi, 2021). Por tanto, la práctica de horticultura se fortalece para solventar la demanda de alimentos de las grandes urbes y mitiga el problema de la reducción de suelo apto para el cultivo.

Así se origina el uso frecuente de técnicas de agricultura vertical para brindar productos alimenticios, eliminando de este modo la preocupación de los cambios climáticos o la temporada de ciertos productos, en este sentido se ha multiplicado la productividad y

facilidad de comercialización que a la vez genera el beneficio en la economía de las familias. En consecuencia, la importancia del presente estudio radica en desarrollar un sistema embebido automatizado que permita monitorear y controlar las condiciones ambientales de un cultivo vertical, en base a una red de sensores, para mejorar la producción de cultivos hortícolas.

Lo mencionado fundamenta su factibilidad en el favorecimiento expresado por la Secretaría Nacional de Planificación sobre la aplicación y uso de nuevas tecnologías para incrementar la productividad (Senplades, 2012). Que análogamente permiten el continuo avance de procesos productivos de las funciones agrícolas, mediante el control y uso de técnicas que tienen el fin de reducir costos para los productores en busca de mejorar utilidad de inversión, menorar el consumo de agua de manera desmedida, limitar el uso de fertilizantes o pesticidas y la optimización y uso adecuado de la luz.

## Capítulo II. Marco Teórico

### 2.1. Introducción

El capítulo II se describe sobre la Horticultura, el cultivo y las características principales, para luego abordar sobre el sistema de riego y los distintos tipos de riego que los agricultores usan en sus sembradíos, se habla sobre los cultivos verticales, sus variedades, se indica también sobre la agricultura de presión, etapas y las herramientas de uso en este proceso, a continuación se trata sobre el aprendizaje supervisado y no supervisado, sistemas embebidos las características, estructuras y finalmente se habla sobre los sensores y como los sensores trabajan con la agricultura de precisión.

### 2.2. Horticultura

La horticultura (Figura 1) se presenta como una rama de la agricultura que es la encargada del estudio y a su vez del cultivo de las plantas hortícolas, por tanto, requiere de bastos conocimientos técnicos y de ser constante y meticuloso al abordar esta práctica (Bacópulos, 2001). Por tanto, se encarga del estudio de la producción y aprovechamiento de las frutas hortalizas y plantas ornamentales, ya que esta práctica ha sido usada para abastecimiento humano porque es una fuente de alimento para el mundo y se puede generar diversas combinaciones alimentarias a partir de los productos primarios.

**Figura 1**  
*Horticultura*



Fuente: (Repetto, 2020)

Esta práctica es conocida y usada desde algún tiempo atrás, para usar esta técnica se requiere de varias características, suelos ricos en minerales, nutrientes, debe estar el cultivo en constante riego, tener luz solar y además debe estar protegido de los factores ambientales que se presentan como granizo, lluvia, exceso de sol, granizo, vientos fuertes, por eso debe permanecer cubierto.

### **2.2.1. Cultivos hortícolas**

La Sociedad Internacional de Ciencias Hortícolas es la encargada de determinar los tipos de cultivos en la rama, estos se determinan según el tipo de producto del que se encargan, así encontramos la floricultura que se dedica a la temática ornamental con el uso de las flores, adicional la horticultura se centra en la producción de hortalizas y sus variedades, la Fruticultura como su nombre lo indica se responsabiliza de las frutas, y finalmente las especies aromáticas o medicinales es una de las ramas que se encarga de la producción y comercialización de los productos indicados (Magrama, 2013).

En esta perspectiva, los cultivos de horticultura se clasifican en tres tipos, el primero por rigor científico, seguido de orden práctico y de tipo biológico, dichos cultivos se diferencian por la extensión de las raíces (Manrique & Sanabria, 2020). Adicional, los cultivos demuestran gran avance en países como Ecuador, concentrándose en una distribución por regiones que toma en cuenta las características climáticas de cada sector, que se constituye en uno de los condicionantes para el tipo de cultivo a realizar (D. Sánchez, 2017).

En consecuencia, los cultivos implementados en el sector urbano se realizan en la actualidad con más frecuencia, por lo que se hace necesario el cuidado de manera técnica que es donde se integra a la actividad el uso de los mecanismos, creados hace algunos años pero que potencian su uso en la actualidad y se fundamentan en las políticas públicas de naciones,

esto con el fin de develar la idea moderna del campo y la ciudad; por tanto, los tipos de cultivos hortícolas se centran en la base de las grandes ciudades como campo de acción, que conlleve a producir diversos elementos que benefician la alimentación, buen uso de espacios, manejos adecuados de recursos como agua y energía, bajo la premisa de una actividad multifuncional (Gomes et al., 2019).

### **2.2.2. Características de los cultivos hortícolas**

Los cultivos hortícolas tienen un sin número de características que las hacen distintivas en el mercado ya que presentan alta demanda en la dieta diaria de alimentos en los pobladores, además los productos tienen un alto valor nutricional. La técnica de la horticultura usa distintos tipos de tecnologías que es empleada para sembrar, proteger el cultivo, cosechar, para la distribución de productos y el consumo.

Para sembrar se puede usar diferentes tipos de terrenos tanto grandes como pequeños, las personas que se dedican a esta actividad son conocidas como horticultores los mismos que están capacitados para realizar esta práctica que demanda de paciencia, conocimiento y habilidad en el manejo de estos cultivos. Se usa abonos especiales, fumigadores y un sistema de riego durante el ciclo de cultivo y cosecha, los sembradíos pueden ser de distintas especies; se requiere de mucha agua para que tenga éxito, con el uso de la tecnología se reducen los tiempos de siembra y de producción.

Por lo anteriormente mencionado se puede decir que Ecuador presenta condiciones aptas tanto en el suelo como en su clima para la actividad agrícola, por tanto, la sierra norte y centro se dedican a la horticultura, en este sentido las hortalizas producidas muestran calidad en el producto, es así como al tener esta característica han logrado ingresar a mercados seleccionados de nivel internacional. En este contexto menciona Zúñiga (2014) que se han exportado hortalizas frescas y congeladas en el año 2004-2005, crecieron de 12857 TM con

un costo de \$10239000 hasta 83424 TM con un costo de \$41440000 respectivamente, y con una tasa de crecimiento anual de 34.7% para el volumen y del 20.4% para valores FOB (pág. 3).

### **2.3. Sistemas de riego**

Desde la antigüedad la necesidad de regar los sembríos propició el surgimiento del riego, que al igual que la agricultura se realizaba de manera empírica o de manera básica, entre los sistemas implementados podemos encontrar pequeños agujajes, presas de mampostería y utilización de aguas freáticas (Serrano, 1999). En contraste a la actualidad el cambio climático implica riegos frecuentes para el crecimiento y desarrollo del cultivo, por lo que se hace necesario implementar sistemas que permitan el cuidado de los recursos de agua y luz, a la vez que brindan mejores resultados de producción y comercialización.

En este sentido, los sistemas de riego permiten que en los cultivos se mantenga la humedad adecuada y la correcta medición de absorción para mantener una buena producción, análogamente se busca aumentar la operatividad de los sistemas en los países en vías de desarrollo, en los cuales se evidencia el uso indiscriminado de recursos hídricos para regar cultivos donde el traslado del líquido causa desperdicio (Y. Hernández et al., 2020). Lo descrito responde a las necesidades de los sectores con clima cálido, mientras que en otros lugares implica el uso de tecnologías modernas para controlar sistemas de riego que poseen diversidad de estrategias.

En conclusión, el uso de sistemas modernos en las actividades de agricultura fortalece las medidas en la eficiencia de controles automáticos que operan la distribución de agua en los canales, implementados con estructuras matemáticas por la caracterización de las dinámicas en la operatividad, dichos procesos matemáticos emplean ecuaciones lineales de Saint-Venant que resultan ser ecuaciones no lineales y de tipo hiperbólicas, finalmente tienen

el objetivo del conteo de balance de masas que en otros términos se refiere al caudal y área (Y. Hernández et al., 2020).

- Riego por superficie

La escasa o nula necesidad de utilizar energía en este tipo de riego hace que sea uno de los más usados alrededor del mundo, emplea un sistema de agua unidimensional de manera vertical para garantizar la infiltración del líquido en el suelo y una infiltración acumulada que permita el adecuado crecimiento de los sembríos (Figura2), suele calcularse mediante la ecuación de Richards, revisiones teóricas y aplicaciones mediante soluciones analíticas (Boumediene, 2017). De este modo se detecta una de las ventajas por presentar costos bajos de inversión pese a demostrar pequeñas pérdidas de agua en la infiltración se ha incrementado a través de nuevos estudios la eficiencia tecnológica.

**Figura 2**  
*Riego por superficie*



Fuente: (Garnica, 2016)

Existen varias formas de regar por superficie: por surco, por melga y tendido, este último es el más ineficiente, muchos fundamentos se explican que sirven tanto para riego por surco como para riego por melga, estos riegos mantienen la característica de realizarse por gravedad, operado por un método adecuado que permite dispensar el agua sin el uso de tuberías, así esta se constituye en la principal razón para la existencia hasta la actualidad (Mañueco, 2020). En consecuencia, es vital en este tipo de riego las habilidades del regador

para asegurar el riego a toda la plantación, a la vez que de manera cuidadosa elimina las pérdidas en la conducción y distribución del agua.

- Riego tendido

Se constituye como un sistema de riego utilizado en épocas pasadas y tiene la desventaja de ocasionar elevado gasto del líquido en la distribución, debido a que se realiza surcos en el lote que deben tener mínimas pendientes para efectivizar el proceso de riego, por tanto, existe casos en que sectores o plantas del cultivo no alcanza a ser filtradas o en caso contrario produce exceso de líquido (Figura 3). Sin embargo, en la actualidad existen estudios sobre la optimización de este tipo de proceso a fin de diseñar de manera correcta el riego de forma que se evite el infructuoso gasto, así mediante el modelo de la onda cinemática validado por el modelo de Saint – Venant y Richards se disminuyó el gasto en aproximadamente 2000 m<sup>3</sup> (Chávez & Fuentes, 2018).

### **Figura 3**

*Riego tendido*



Fuente: (Garnica, 2016)

En consecuencia, la aplicación de estudios fortalece el uso de riego por tendido ya que se demuestra beneficios de reducir tiempo y cantidad de líquido utilizada, esto depende de la longitud, pendientes construidas, y las propiedades del suelo como textura, humedad y

porosidad (Chávez & López, 2019). Para llevar a cabo un tipo de estudio riguroso y de calidad depende de la aplicación de los modelos adecuados y de manera fundamental del estudio de campo que se realiza en el sector para conocer características propias del lugar.

- Riego por surcos

La exigencia de un sistema en surcos se fundamenta en el diseño, manejo y las mismas operaciones en el campo de cultivo, se caracteriza por el uso de canales que permiten que el agua avance manteniendo siempre los niveles de caudales que menoran las pérdidas y el arrastre de sustancias (Figura 4), de este modo, se utiliza el modelo en la investigación de infiltración Kostiakov - Lewis que ayudan a corroborar el perímetro de humedad y alcance de filtración (Pérez & Jiménez, 2020). Por tanto, es posible adaptar este tipo de riego a sembríos lineales de horticultura en todas las variedades por su destacado valor de eficiencia.

**Figura 4**  
*Riego por surcos*



Fuente: (Garnica, 2016)

Por otra parte, este tipo de riego depende del ancho y caída del canal para obtener un número menor en pérdida de precipitaciones, para así mantener el suelo con la cantidad de agua necesaria, además se hace indispensable la implementación de energía en mayor cantidad (Madrigal et al., 2019). En este sentido existen ahora investigaciones que definen las características apropiadas de cada modelo para mejorar el proceso, a la vez que facilita la

toma de medidas correctivas a fin de perfeccionar modelos y métodos que beneficien al productor.

- Riego por melgas

El uso de franjas de suelo a modo de canal permite el riego por melgas que consiste en la filtración del agua desde un sector al que se denomina cabecera hasta el pie del sembradío, se adapta a los cultivos de forraje como hiervas, granos, maíz y sorgo con la característica de baja infiltración (Figura 5), en esta época se utiliza el modelo WinSRFR para predecir el volumen de riego y escurrido del líquido en el cultivo (Puppo et al., 2018). En este sentido la utilidad del riego reconoce que se debe verificar la adecuada velocidad y filtración en el suelo de manera lenta que mejoraría la función de la actividad agrícola.

**Figura 5**  
*Riego por melgas*



Fuente: (R. López et al., 2015)

Adicional, el proceso debe contar con la previsión de analizar de manera frecuente la pendiente transversal para que los recursos hídricos descendan a la par, por tanto, también la optimización siempre debe ser ajustada a lo largo de la melga. Por otra parte, en algunos países se evalúa según los resultados preliminares del modelo WinSRFR con diseño que busca optimización para mitigar impactos ambientales y sociales, obteniendo como resultado mejora en el desarrollo del cultivo y la productividad (Santacruz et al., 2019).

- Riego presurizado

El presente sistema se realiza por la conducción del agua por tuberías que mediante la presión filtra de manera directa a las plantas (Figura 6). Así el riego deja de perder líquido y mantiene la planta húmeda, tiene la finalidad de reducir el consumo de energía que cuida el ambiente y lo adapta a los cambios de clima de la actualidad con el objetivo de minimizar la emisión de dióxido de carbono, el sistema se maneja bajo emisores que distribuyen el agua a través de bombes de manera equitativa (Ascencios, 2021).

**Figura 6**  
*Riego presurizado*



Fuente: (Garnica, 2016)

Las características de los tipos de riego presurizado requieren del uso de energía que debe ser manejada para el cuidado del ambiente, adicional minimiza el uso de agua para evitar el mal consumo, por los recursos requeridos la inversión inicial económica es alta, además presenta un costo operativo elevado, entre los beneficios puede destacar mayor eficiencia mediante la utilización de los recursos, el uso de tecnología es también muy frecuente lo cual amplía la gama de planificaciones (Quinde & Zhagui, 2019).

- Riego por aspersión

Es la aplicación de agua mediante un rociador que esparce pequeñas gotas de agua, tratando de imitar el proceso natural de la lluvia, en el sistema el agua es conducida por tuberías que sobre presión se reparte en una regadera de gotas que humedece las plantas (Figura 7). En este caso el viento juega un rol fundamental para conducir la lluvia de agua a las plantas, en este sentido se puede ocasionar pérdidas al no cubrir el cultivo y direccionar el riego a otro lugar en lo cual se evaporaría el líquido (Salazar, 2019).

**Figura 7**  
*Riego por aspersión*



Fuente: (Garnica, 2016)

Es un sistema en el que el uso de la energía se eleva con mecanismo tecnológicos como sistema de aspersión fija, portátil, cañón, pivot central y avance frontal, también las precauciones a tomar en la velocidad de riego del goteo a fin de evitar el escurrimiento del líquido en el suelo, del mismo modo se debe tomar en cuenta las desventajas y beneficios como buenos resultados en las ventas de los productos, que generará rentabilidad para el producto (Pazos et al., 2019).

- Riego por microaspersión

Se ha realizado investigaciones sobre la microaspersión implementado según el método diagonal que se realiza por medio de goteo (Figura 8), los estudios tienen su base en los parámetros hidráulicos, entre los cuales emergen los procesos de presión, caudal e

intensidad, lo mencionado con el fin de generar el contenido de humedad del suelo, en este sentido también se encuentra las principales razones de las que depende el correcto riego como: textura del suelo y el contenido de la materia (Alaniz & Cruz, 2018).

### **Figura 8**

#### *Riego por microaspersión*



Fuente: (Garnica, 2016)

Es válido mencionar que, en la producción agrícola mediante buenos sistemas de aspersión, genera competitividad para los productores, al cubrir las cualidades hídricas que permitan cobijar el cultivo mediante el correcto desarrollo, adicional, el diseño de dichos sistemas aumenta la producción de frutos o elementos naturales, ayuda a mejorar la calidad de producto, a mantener los cultivos con el adecuado consumo hídrico de las plantas en los veranos a la vez que identifica las fuentes de energía para contribuir al riego (Á. Hernández, 2019).

- Riego por goteo

Diferentes estudios demuestran que es mayor el rendimiento de los cultivos cuando se implementa sistemas de riego por goteo, en el cual se utilizan microtubos y un tanque elevado (Figura 9), se obtiene como resultado mayor rendimiento promedio (Benavides et al., 2017).

En este sentido parte de la historia de los sistemas cuenta que el uso de goteo era utilizado en Egipto para regar sus campos, por eso se ha constituido en la época actual en uno de los sistemas más utilizados.

**Figura 9**  
*Riego por goteo*



Fuente: (Garnica, 2016)

Entre los sistemas que se utiliza encontramos los clásicos que tiene válvula ajustable para medir la cantidad de agua según el tipo de planta, otro es un kit muy versátil incluye mangueras, grifos, boquilla y una junta tridireccional que puede ahorrar hasta más de la mitad del 100% y finalmente existe un sistema de riego adaptativo (Ascencios, 2020). En el mercado se encuentra sistemas de goteo que han evolucionado y así mejoran en gran medida las características de los terrenos productores de alimentos.

#### **2.4. Cultivos verticales**

El cultivo vertical se encuentra enfocado en el sembradío que se pueden realizar dentro de edificios, así como también en rascacielos, entonces entre mayor tamaño tenga se puede almacenar más cultivo, por tanto, la estructura funciona como un invernadero, en tal sentido es considerable acoplar las variables de ambiente frente a la optimización en el crecimiento del sembradío y el incremento de la producción.

En este contexto se presentan cuatro aspectos a tomarse en consideración para aumentar la viabilidad en la agricultura de precisión con el fin de producir más productos por cada metro cuadrado como: 1) el diseño físico, 2) iluminación, 3) medio de crecimiento y 4) características de sostenibilidad (Soria, 2018). Por consiguiente, la producción se la realiza con el uso de una estructura que presenta forma de rascacielos y para la generación del ambiente natural se combina la luz natural con la artificial, el uso de este tipo de cultivo presenta un ahorro del 90% del agua.

El uso de este tipo de invernaderos extiende el entorno y está bajo ambiente controlado, la utilización de los sensores electrónicos permite que el cultivo reciba la porción adecuada de luz, así como también nutrientes y calor. En este sentido no necesitan de mucho espacio para la producción de alimentos, utiliza tecnología LED, los productos son orgánicos y amigables con el medio ambiente.

## **2.5. Variedad de hortalizas**

Las hortalizas se encuentran variadas para el consumo humano, además que aportan un sin número de vitaminas y minerales por lo tanto son empleadas para la alimentación, entre las principales se puede mencionar según Haro (Haro, 2012a) las siguientes.

- Cilantro. – el cultivo de la planta es anual, llega a crecer de 40 a 60 centímetros, sus ramificaciones son delgadas, sus flores son pequeñas, blancas o tenuemente rosadas, presentan un olor agradable y suave, pero de degustación fuerte.
- Acelga. - la siembra de las acelgas tiene la característica de ser una planta bianual de ciclo largo que no desarrolla raíz o a su vez no da un fruto comestible más bien es comestible sus hojas.
- Coliflor. – planta anual que presenta un tallo reducido y no ramifica, sus hojas son de aspecto liso, largas y con de color verde claro, la parte comestible está en los pequeños

árboles que salen del tronco que son de color blanco crema, pero la tonalidad de las flores es amarilla del que se desprenden cuatro pétalos.

- Zanahoria amarilla. – el cultivo de esta planta es de ciclo anual, en el tiempo de su desarrollo se forma una roseta con escasas hojas, en la segunda estación de evolución luego del descanso se presenta un tallo corto en el mismo que se forman flores.
- Lechuga. - es una planta de cosecha anual que presenta raíz pivotante de rápido crecimiento, de pendiendo del tipo sus hojas pueden ser lisas, crespas de color verde amarillenta hasta llegar a una tonalidad morada.
- Rábano. – planta anual o bienal, presenta un color rojo, rosado, blanco, pardo oscuro, las hojas son basales con pequeños pelos incrustado, de borde irregularmente dentado, el fruto se presenta de 3 a 10 centímetros de longitud, de contextura esponjosa, indehiscente.
- Pimiento. – el ciclo de esta planta es anual, presenta hojas enteras lampiñas y lanceoladas, las flores surgen solitarias en los nudos del tallo, el fruto es de color verde, rojo, amarillo, tomate, blanco, negro, violeta dependiendo de la variedad.
- Espinaca. –Presenta superficialmente su sistema radicular, llega a medir el producto de 80cm de largo a 30 cm de ancho, las hojas es la parte comestible las mismas que son lisas de color verde y el crecimiento es en roseta, además tiene un tallo corto.
- Cebolla colorada. – es una planta bienal, presenta un tallo reducido, el color de la piel muestra un rojo púrpura, el tamaño del producto comestible es mediano y grande la producción de estas se encuentran durante todo el año.
- Col. – la planta de este producto se presenta bianual, siendo así que su repollo es una gran yema terminal, las hojas son suaves al tacto, pero otras son crocantes, su forma es achatada, en otras ocasiones redonda o a su vez puntiaguda.

## 2.6. Agricultura de precisión

La agricultura de precisión se basa en suministrar una cantidad de insumos en el lugar y momento exacto, siendo así que se usa la tecnología de información para conocer y adecuar el manejo de suelo, así como también de los cultivos dentro de un lote. Por tanto, involucra la utilización de un sistema de posicionamiento global con la fusión de otros recursos electrónicos con el fin de obtener datos precisos del cultivo, además satisface una necesidad presente en la agricultura moderna que es el manejo adecuado de grandes extensiones de tierra. La principal ventaja es que dentro de un lote se puede realizar análisis de resultados de los ensayos, y de esta forma se puede adecuar el manejo diferencial en el contexto mismo.

La agricultura de precisión es usada para trabajo de campo el mismo que usa un sin número de tecnología como: satélites, sensores, imágenes y datos geográficos toda esta información recabada se fusiona para comprender la variación del suelo y también de los cultivos (E. García & Flego, 2018). Por tanto, con el uso de la agricultura de precisión ayuda a los cultivadores a tomar decisiones acertadas en cuanto a las semillas que se van a usar en el cultivo, determinar el terreno más apropiado para trabajar, también se puede estar en la capacidad de pronosticar el rendimiento de la cosecha, por tanto, se presenta preciso el trabajo a realizar.

Por consiguiente, el fin de la agricultura de presión es incrementar la producción en el campo, pero con un bajo consumo de recursos en tal sentido se logrará un ahorro y reflejará mayor ganancia, la aplicación de esta tecnología en el área agrícola permite la explotación del suelo de forma más eficiente, es decir conseguir más productos con menos agua, gasolina, herbicidas y amigables al medio ambiente.

### **2.6.1. Etapas de la agricultura de precisión**

La agricultura de precisión permite la gestión de los cultivos por medio de la observación, midiendo y tomando decisiones frente a la variabilidad de los factores que pueden jugar en contra al cultivo (Quiroga et al., 2016), siendo así que dentro de las etapas de la agricultura de precisión se presentan:

- La captación de los datos es un mecanismo tecnológico que necesita que este preinstalada en el suelo a trabajar y con la incorporación de una red de sensores se estará en la capacidad de recabar la información.
- El análisis de resultados que se realiza cuando han sido registrados los datos para luego ser interpretados matemáticamente o a base de la estadística con el mapeo, gráficos y clasificación de datos
- La toma de decisiones una vez que se tienen las principales conclusiones que fueron emitidas sobre la base de los datos recabados se tomará en consideración la decisión más apropiada la misma que la hará un ingeniero agrónomo o técnico agrícola y por último la evaluación o a su vez conocida como la etapa de monitorización del rendimiento siendo así que una vez que termina el cultivo el agricultor analizará el contexto en el cual se desarrolló la producción, por ende en este punto se puede conocer sobre los aciertos y falencias presentadas para luego aprender de los errores

### **2.6.2. Herramientas para la agricultura de precisión**

Con el uso e implementación de la agricultura de precisión pueden abordarse los problemas con facilidad, pero siempre y cuando sean abordada la práctica adecuadamente, los agricultores deberán definir las necesidades para determinar las herramientas necesarias a ser usadas, por cuanto el uso de estos equipos permite ser eficiente, formular registros concisos y precisos así como también facilita la administración en el sector agrícola, por ende es

recomendable reconocer las necesidades antes de invertir en software o demás equipos, en este sentido presenta a continuación Hernández (2021) los siguientes herramientas a ser tomadas en consideración con la técnica.

**Equipo de autoguía.** - este tipo de equipo tecnológico logra que los agricultores cubran un área de manera rápida y con precisión, en tal sentido se evita que zonas olvidadas no sean cubiertas, con el uso de esta herramienta reduce tiempo, dinero y evita accidentes.

**Tecnología de raios variables.** - el uso de esta tecnología es beneficiosa en el cultivo por cuanto permite combatir las plagas, enfermedades y regular el nivel del pH del suelo ya que en diferentes sectores se presenta alta o baja.

**Tecnología de sensores de proximidad.** – en este tipo de tecnología se presentan sensores de humedad y temperatura entre otros, el uso y aplicación de estos sensores permite al agricultor detectar a tiempo algún problema que pueda atravesar el cultivo, en tal sentido permite detectar las falencias a tiempo con menos recursos.

**Sistema de posicionamiento global y sistemas de información geográficas.** – el sistema de posicionamiento global es un satélite que permite localizar al equipo de campo que se encuentra dentro del entorno del cultivo, presentando una precisión de un metro en tiempo real. En cambio, el sistema de información geográfico es un software que realiza el registro, recupera, almacena, muestra y analiza la información geográfica plasmada como un mapa, la importancia de encontrar con rapidez radica en:

- La comparación es rápida tanto los resultados de laboratorio con la localización de la muestra de suelo con el uso de un mapa del terreno.
- Se puede aplicar con precisión tanto los pesticidas como los fertilizantes con el fin de cubrir las necesidades del suelo.
- El cultivo se puede ajustar si las condiciones no son favorables en el suelo

- Mientras se realiza el monitoreo los agricultores pueden guardar los datos de la siembra mientras se cubre el terreno.

**Sensores remotos.** – los sensores se presentan aéreos y satelitales, los mismos que muestran variación en el color del suelo que se puede presumir modificaciones en la distribución y tipo de suelo, indican el desarrollo de la siembra, así como también se puede conocer los límites del terreno.

## 2.7. Machine Learning

La transformación de las formas de trabajo de las máquinas mediante procedimientos sistémicos, en la actualidad da origen al aprendizaje automático que desarrolla una metodología para predecir patrones o inclinaciones básicas de los ambientes de cultivos, detectados mediante algoritmos. Por tanto, la selección de un sistema de algoritmos adecuados para obtener resultados de calidad permite regular, administrar controlar los aspectos fundamentales para obtener una buena producción (J. García et al., 2020).

De este modo, también se logra la optimización en tiempo, trabajo de mano de obra que finalmente minimiza los costos que aporta a los beneficios empresariales, seguramente en los siguientes años los algoritmos con los mecanismos inteligentes incursionarán en cualquier campo de los que se desenvuelve el hombre, en este sentido, Machine Learning se ha desarrollado como una de las ramas de la inteligencia artificial.

Así el agro a nivel nacional e internacional ha experimentado de manera notable un crecimiento económico, donde se toma como objetivo la evolución y modernización de sectores de las naciones para el beneficio común. Por lo que el rendimiento de los cultivos se observa de manera creciente, al punto de provocar en temporadas de exceso de producción y en otras el desabastecimiento de las mismas.

De acuerdo con los objetivos que se plantea conseguir con los datos de los cultivos se selecciona el tipo de estudio a realizar, este puede ser aprendizaje supervisado o no supervisado y se definen según el tipo de agrupación, además que se constituyen en una de las estrategias de ataque a cualquier tipo de imprevisto en las plantas.

### **2.7.1. Algoritmos de aprendizaje supervisado**

El método se implementa en algunos estudios como parte de la solución para determinar enfermedades en los cultivos, en el proceso se implementa maquinaria automatizada para detectar padecimientos en base a los datos que se obtiene de terreno, de este modo los productores tienen el objetivo de descubrir a tiempo afecciones y así implementar las medidas necesarias para evitar pérdidas en grandes volúmenes, el método se emplea y posterior no necesita ser monitoreado en todo momento, lo cual es una de las ventajas imperativas para la optimización de tiempo.

Lo mencionado difiere mucho a épocas pasadas donde la observación, análisis y estudios era un proceso realizado por el hombre para detectar de manera empírica las enfermedades de los cultivos, en cambio la avanzada transformación de la tecnología brinda la posibilidad de implementar procesamiento digital y el aprendizaje automático de las computadoras que, mediante cámaras para obtener imágenes y sensores que terminan los patrones de la afección, determinan el tipo y grado de enfermedad para mejorar cuidados en el proceso de crecimiento y posterior la calidad en el cultivo de los productos, esto de manera independiente al tipo de cultivo (Aristizabal, 2017).

De este modo, el desarrollo de las nuevas tecnologías aúna las actividades de la función agrícola y el proceso de algoritmos que utilizan datos etiquetados para poder controlar las variables de entrada y salida que dan respuesta a los procesos de clasificación y

regresión que cumplen con la función de identificar, diagnosticar o detectar y predecir expectativas de vida y crecimiento respectivamente.

### **2.7.1.1. Árboles de decisión**

Se presentan los árboles de decisión como gráficos que permiten visualizar posibles soluciones a una decisión que se forja con ciertas características, siendo así que es uno de los más usados dentro del algoritmo de aprendizaje supervisado ya que posibilita la realización de tareas de clasificación, el manejo y comprensión de la forma del funcionamiento se presenta simple pero las estimaciones son altamente confiables

El funcionamiento del árbol de decisión no es complejo, por tanto, para conseguir un árbol confiable y valorar cada subdivisión presentan de entre todos los árboles presentados el nodo raíz de los sucesivos, el algoritmo deberá medir de alguna manera las predicciones logradas y valorarlas para comparar de entre todas y obtener la mejor. Para medir y valorar, utiliza diversas funciones, siendo las más conocidas y usadas los “Índice gini” y “Ganancia de información” que utiliza la denominada “entropía “. La división de nodos continuará hasta que llegar a la profundidad máxima posible del árbol o se limiten los nodos a una cantidad mínima de muestras en cada hoja.

(Bagnato, 2018)

Por consiguiente, en los últimos tiempos los algoritmos que se presentan mediante árboles son expuesto como uno de los mejores, ya que ayudan grandemente a la predicción, estabilidad y facilita la interpretación de los datos, se adaptan fácilmente a la resolución de cualquier tipo de inconveniente (Brenes, 2020).

### **2.7.1.2. K-NN (*k* -nearest neighbor).**

El algoritmo K-NN es un método que se presenta para la búsqueda en la observación de los datos que se encuentran más cerca, misma que tiene como finalidad predecir y

clasificar el lugar de interés para lo cual se basa en la estimación de los datos que lo envuelven, siendo así que es supervisado por cuanto se presentan los datos etiquetados, se basa en instancia, además se presenta el funcionamiento fácil, es usado en la resolución de problemas tales como: reconocimiento, búsqueda semántica y localización de problema (Mezquita et al., 2021).

Así como también el KNN es un algoritmo de clasificación que es utilizado en los cultivos para detectar patrones iguales o diferentes, es de ayuda para los cuidadores por cuanto el correcto diagnóstico permite implementar estrategias a tiempo para prevenir enfermedades y pérdidas en económicas. El sistema trabaja mediante sensores que tiene la función de verificar el grado de humedad y la temperatura del ambiente a fin de tomar las medidas necesarias que se enfoquen en mantener y mejorar el cultivo y su producción (Pinos, 2021).

### **2.7.2. Métricas de desempeño**

Para medir el desempeño de los algoritmos de aprendizaje para un determinado modelo de datos se ha de usar diferentes métricas como las que se describe a continuación.

#### **2.7.2.1. Matriz de confusión**

La matriz de confusión es una medida de rendimiento para el problema de clasificación de aprendizaje automático donde la salida puede ser de dos o más clases. Es una tabla con 4 combinaciones diferentes de valores previstos y reales.(Narkhede, 2018)

En la Figura 10 se ilustra la composición de la matriz de confusión para un modelo de datos con dos clases.

**Figura 10**  
*Matriz de confusión*

Actual	Positive	TP	FN
	Negative	FP	TN
		Positive	Negative
		Predicted	

Fuente: (MLee, 2021)

El uso de la matriz de confusión es muy importante ya que a través de esta se obtiene varias métricas que permiten determinar el rendimiento de un algoritmo de clasificación. Está compuesta por el número de aciertos: TP (Verdaderos Positivos) y TN (Verdaderos Negativos); así como los errores: FN (Falsos Negativos) y FP (Falsos Positivos) determinados por el algoritmo.

- TP (Verdaderos Positivos): El algoritmo determinó un caso positivo y acertó.
- TN (Verdaderos Negativos): El algoritmo determinó un caso negativo y acertó.
- FN (Falsos Negativos): El algoritmo determinó un caso negativo, cuando realmente era positivo.
- FP (Falsos Positivos): El algoritmo determinó un caso positivo, cuando realmente era negativo.

#### 2.7.2.2. *Exactitud (Accuracy)*

La exactitud es la métrica más común en los algoritmos de clasificación que está definida por el cociente de los aciertos del algoritmo, entre el número total de evaluaciones que este realizó. Matemáticamente, se define como se puede observar a continuación en la Ecuación 1:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

La exactitud por sí sola, no da una estimación completa cuando se trabaja con un conjunto de datos de clase desequilibrada, donde existe una disparidad significativa entre la cantidad de etiquetas positivas y negativas. (Google Developers, 2020)

### 2.7.2.3. *Precisión (Precision)*

La precisión cuantifica la cantidad de predicciones positivas, que fueron acertadas y se define mediante la expresión matemática que se muestra en la Ecuación 2.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

### 2.7.2.4. *Sensibilidad (Recall)*

Cuantifica el número de predicciones positivas reales con respecto a todos los elementos positivos del conjunto de datos. La Ecuación 3 representa la definición matemática de la Sensibilidad

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

### 2.7.2.5. *Puntuación F1 (F1 Score)*

La puntuación F1 permite cuantificar un balance entre “precisión” y “sensibilidad”, esto es meramente útil ya que se toma en cuenta tanto las clasificaciones “Falsas Negativas” así como las “Falsas Positivas”, quiere decir que, se da la misma importancia a la capacidad de distinguir un “Positivo Real” de un “Positivo Falso”, así como el porcentaje de distinción de un “Positivo Real”, entre todos los “Positivos” de un conjunto de datos (Ecuación 4).

$$F1 = 2 * \frac{\text{Presicion} * \text{Recall}}{\text{Presicion} + \text{Recall}} \quad (4)$$

## 2.8. Sistemas embebidos

El sistema automático a base de sensores permite humedecer el suelo para brindar agua a las plantas, utilizando un proceso de mecanismos construidos con materiales como PVC y funciona con un sistema de llenado de tinaco, en el cual la bomba cesa el nivel de agua, en otro contexto situado en las ciudades algunas personas cultivan platas que necesitan riego en diferentes tiempos de acuerdo con el tipo de planta, por lo que el dueño necesita la inversión y planificación para el cuidado de las misma (Valdez & Villanueva, 2020).

Por lo mencionado se implementa sistemas embebidos que con el uso de hardware y software sistémicos que permiten cumplir de manera automática con las actividades del ser humano ante la necesidad de cuidar los cultivos. El sistema implementa circuitos que se presentan en tamaños pequeños para huertos o para grandes cultivos con fines comerciales. También es importante reconocer la flexibilidad el momento y programar el sistema de acuerdo con las necesidades de los productores y como lo requiera el cultivo (Alonso et al., 2017).

### 2.8.1. Características de un sistema embebido

La red de sensores inalámbricos posee particularidades que lo hacen diferentes de los demás sistemas, se basa en la computación para dar solución a los problemas de la sociedad y la actividad agrícola con el fin de proporcionar funcionalidad y desarrollo (Álvarez et al., 2018). En consecuencia, las características que se destacan son algoritmos complejos que se utilizan paso a paso para lograr controlar motores o microprocesadores, interfaz de usuario donde se implementa el menú, adicional el trabajo en tiempo real es primordial, mientras la

multifrecuencia soporta la presión de actividades y finalmente los costos componentes computacionales, materiales y los dispositivos (Velasquez, 2016).

En consecuencia, la seguridad de la informática que se incrusta en el ROM incorpora singularidad de características de la lógica de sistemas secuenciales y digitales para ocasionar nuevas funciones que beneficien al campo de la horticultura y agricultura para facilitar el manejo de riego desde un computador, que permitan robustecer la funcionalidad para mitigar los defectos (Martínez, 2018).

### **2.8.2. Estructura básica de un sistema embebido**

Internamente cuenta con componentes conformados por un microprocesador, adjunto a un software, una memoria RAM o ROM y una serie de salidas y entradas, en consecuencia, las actividades a realizar no necesitan de procesadores complejos por lo que se utiliza de 8 a 16 bits solamente, el programa que alberga la memoria desarrolla las aplicaciones para el manejo del sistema (Carpio, 2018).

En detalle también se cumple el proceso de la siguiente manera, el sensor transforma la cantidad física en energía que puede ser leída por un instrumento electrónico, adicional un convertidor digitaliza la señal para luego ser procesados y almacenados en la memoria, finalmente los datos se convierten en análogos y son compartidos al convertidor.

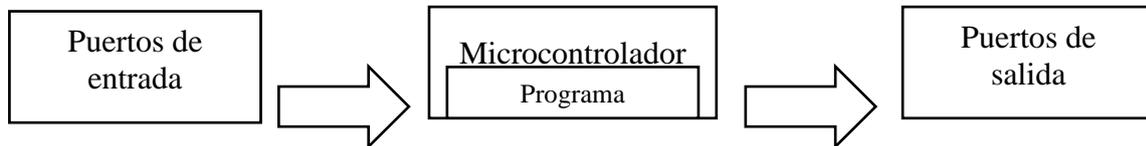
### **2.8.3. Microcontroladores**

Los microcontroladores se presentan como circuitos integrados, siendo este un complemento principal en una aplicación embebida, por lo tanto, se pueden ver como una pequeña computadora la cual tiene incluida sistemas que le permiten controlar los elementos tanto de entrada como de salida. Además, muestra en sus componentes un procesador, memoria que permiten guardar, así como también su funcionamiento se asemeja a un minicomputador, es aplicable en los productos que requiere un proceso automático

condicionado a las entradas (Marmolejo, 2017). Para ejemplificar mejor las líneas anteriores obsérvese la Figura 11.

**Figura 11**

*Diagrama de bloques de un Microcontrolador*



Fuente:(Pichucho, 2007)

### **2.8.3.1. NodeMCU**

Los microcontroladores NodeMCU se encuentran en el mercado con un bajo costo, fue desarrollada en base a Arduino, por lo tanto, es compatible con el lenguaje que es usado normalmente en programación, incorpora dentro de su sistema un módulo ESP8266 el mismo que le sirve para poderse conectar mediante vía WIFI. En tal sentido se puede decir que “Node MCU es una plataforma de desarrollo y un firmware de códigos abiertos escritos en LUA y con una API de red estilo Node JS, todos los componentes mencionados le ayudan para presentar una gran flexibilidad”. (Villarreal, 2020)

### **2.8.3.2. ESP8266**

Las redes inalámbricas en las nuevas tecnologías incorporaron el uso de microcontrolador ESP8266 es un chip para brindar WIFI, compatible con el protocolo TCP IP., el mismo que es asociado con el internet de las cosas (Torres, 2019). Presentan ventajas por el bajo costo que representa adquirirlo y permite incorporar el internet a los trabajos que se los realice con este módulo.

### **2.8.3.3. Placa Raspberry PI**

La placa raspberry PI se presenta en el mercado con un bajo costo, es un ordenador que presenta un tamaño pequeño, siendo así que cabe en la mano, sin embargo, su usabilidad puede adaptarse a cualquier equipo y funciona como un computador, es empleado en

proyectos eléctricos, presenta una buena conectividad y conexión GPIO que admite un sin número de proyectos educativos (Sole, 2021).

#### **2.8.3.4. Sensores**

Los sensores poseen la sensibilidad y capacidad de transformar las magnitudes en señal eléctrica, a fin de hacer factible la comunicación de un ambiente computacional de medición y el espacio físico del ser humano, esto se puede realizar gracias a la avanzada industria y los elementos que la constituyen para abordar propósitos que monitoreen sistemas informáticos en todos los ámbitos en los que se desenvuelve el ser humano (B. Sánchez, 2019).

Existen sensores que en la praxis de la agricultura son necesarios para obtener información del cultivo y sus necesidades. A continuación, se presenta los tipos de sensores.

- **Sensores de humedad de suelo.** - permite obtener datos a tiempo real que se presenten a los usuarios (González, 2019) admite conocer el nivel de agua en la tierra para tomar las medidas preventivas en los sembríos, entre los principales se puede nombrar sensores termales infrarrojos, sensores N, P, K, sensores de PH.
- **Sensor de humedad relativa.** - mide el vapor de agua que en cantidades mayores provoca humedad considerable (Alarcón et al., 2017). Los sensores más usados son: mecánicos, higroscópicas, de conductividad, capacitivos, resistivos, infrarrojos.
- **Sensor temperatura ambiente.** - es la temperatura procesada a través del sensor que permite obtener un valor de los cambios mediante señales eléctricas (Pacheco, 2017). Se puede encontrar en el mercado para este fin sensores: RTD, bimetálicos, sensores de temperatura por dilatación fluida, por cambio de estado entre otros.

- **Sensor de temperatura del suelo.** -el sensor mide valores de temperatura del suelo, para conocer las diferentes variaciones térmicas (Pacheco, 2017). Se presentan para el efecto: sensores krohne, womaster, tekbox, strelingsensors
- **Sensor de PH.** -este mide el valor entre lo ácido o lo alcalino, todo esto para determinar si el cultivo absorbe los nutrientes necesarios (Alarcón et al., 2017). Para el tipo de uso se encuentran: inalámbricos, manuales y portátiles, medidores de lones, electrodos de PH.
- **Sensor de Emisiones de CO2.** - es un sensor que mide los niveles de CO2, ya que es un gas de efecto invernadero. Determina si el compost es apto para utilizarlo en el cultivo (L. López et al., 2020).

## 2.9. Protocolos y estándar

Los protocolos son conocidos como un conjunto de elementos que están reglados para lograr comunicarse entre dos o más sistemas que a la vez transfieren información. Es decir, son protocolos de lenguaje de comunicación que están estructurados en base a: sintaxis, semántica, sincronización y métodos para recuperación de errores, dando a conocer claramente la manera para intercambiar los datos, todo esto con el fin de que exista una comunicación eficaz y a la vez ordenada, su implementación se la realiza a través de hardware o software o también con la combinación de los dos (L. Fernández, 2020). Las reglas de los protocolos deben seguirse con precisión, si no el PC no lograría comunicarse con los otros dispositivos, en este contexto se presentan protocolos directos y breves, en algunos casos son complejos y profundos, por lo tanto, consienten en que los dispositivos entiendan las señales electrónicas que se intercambian entre sí por medio de una conexión de red.

### 2.9.1. Modelo OSI

El modelo OSI, es un modelo de referencia para Redes de Comunicación, propuesta por la ISO (Organización Internacional de Normas), con la finalidad de estandarizar las comunicaciones. Las siglas OSI significan, Interconexión de Sistemas Abiertos, y está conformado por siete capas, las cuales se explica a continuación. (Tanenbaum & Wetherall, 2012)

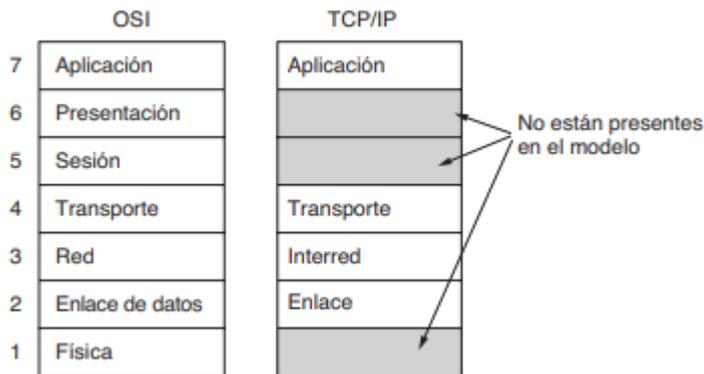
- **Capa física.** – Está relacionada con los aspectos físicos de la comunicación, engloba características eléctricas, mecánicas y de temporización para envío y recepción de bits puros a través de un canal.
- **Capa enlace de datos.** – Se encarga de la detección de errores y de orquestar la comunicación entre dos terminales de tal manera que un receptor rápido no inunde de información a un receptor lento.
- **Capa de red.** – Se encarga de determinar de como los paquetes viajan a través de la red, es decir, las rutas que estos toman para llegar desde un origen hasta un destino.
- **Capa de transporte.** – Se encarga de fragmentar, si es necesario, los datos recibidos por capas superiores, además de mantener una comunicación de extremo a extremo gracias a los campos en sus cabeceras.
- **Capa de sesión.** – Permite establecer sesiones entre diferentes máquinas, además de la sincronización y el manejo de del diálogo para impedir que dos estaciones realicen una misma tarea al mismo tiempo.
- **Capa de presentación.** – Define la manera en como se representan los datos, es decir, la semántica y la sintaxis que representa cada bit, dependiendo del tipo de aplicación.

- **Capa aplicación.** – Esta capa, es la que se encarga de proporcionar servicios de red a los programas que utiliza el usuario.(Tanenbaum & Wetherall, 2012)

### 2.9.2. Modelo TCP/IP

Mientras en el modelo OSI, se define la lógica de referencia, describiendo cada una de las funciones de un sistema de comunicación, con la finalidad de fragmentar los procesos para llevar a cabo la comunicación en componentes pequeños, el modelo TCP/IP, fue diseñado y desarrollado por el Departamento de Defensa (DoD) en la década de 1960 y se basa en protocolos y estándares para cubrir las siete capas del modelo OSI en solo cuatro capas. La Figura 12 muestra las capas correspondientes entre el modelo OSI y el modelo TCP/IP, siendo las de este último:

- **Capa de acceso a la red.** – Es correspondiente con la unión de la capa física y la capa de enlace de datos del modelo OSI, por lo que se encarga de la gestión de control de acceso al medio, detección de errores, transmisión de bits, entre otros.
- **Capa de internet.** – Cumple las funciones de la capa de red del modelo OSI, por ende, en esta capa se definen los protocolos de direccionamiento y enrutamiento siendo el principal de este modelo, el protocolo IP.
- **Capa de transporte.** – Es responsable de la comunicación de extremo a extremos, corresponde a la capa con su mismo nombre en el modelo OSI, siendo TCP el protocolo principal.
- **Capa aplicación.** – La capa aplicación del modelo TCP/IP encapsula las tres capas superiores del modelo OSI dentro de sí, y se encarga de definir las interfaces de comunicación de las aplicaciones del usuario.(Tanenbaum & Wetherall, 2012)

**Figura 12***Capas modelo OSI vs TCP/IP*

Fuente: (Tanenbaum &amp; Wetherall, 2012)

### 2.9.3. IEEE 802.11

Desde que se empezó a usar equipos portátiles apareció el aspecto de la movilidad y con esto la necesidad de tener conexión dentro de un mismo contexto, pero sin perderla, por lo tanto, se proveyó a los equipos portátiles y a los departamentos transmisores y receptores de onda corta con el fin de que siga la comunicación fluida. Por lo tanto, se creó el LANs inalámbrico, con ayuda de la IEEE siendo el resultado de la creación del estándar 802.11, que es compatible con Ethernet (Alvarez, 2019).

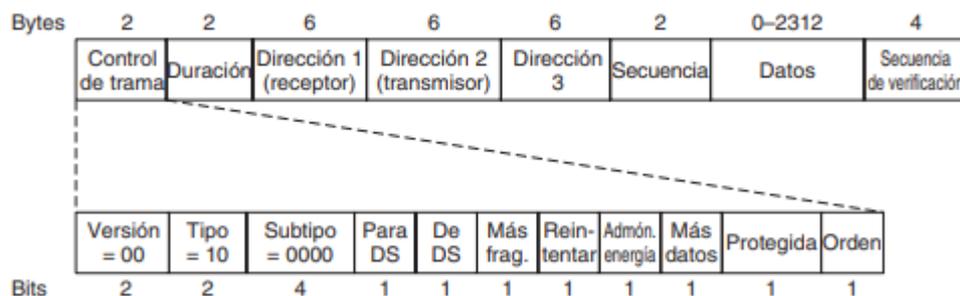
En este contexto el proyecto IEEE 802.11 ha cambiado la manera de conectividad, el Wi-Fi, definido por una familia de IEEE estándares 802.11, es la tecnología inalámbrica más popular utilizado para la transmisión de datos Wi-Fi, ya que comunica a más de la mitad del tráfico de usuarios y trabaja en la capa de acceso a la red del modelo TCP/IP (Khorov et al., 2020).

## Capa MAC

La subcapa MAC 802.11 proporciona una abstracción de la capa física a la subcapa de control de enlace lógico y las capas superiores de la red OSI. Es responsable de encapsular tramas y describir formatos de trama. En la Figura 13 se representa los campos principales de una trama WLAN según lo establecido por IEEE 802.11 y estos son:

- **Frame Control:** es un campo inicial de 2 bytes compuesto por 11 subcampos. Contiene información de control del marco.
- **Duración:** es un campo de 2 bytes que especifica el período de tiempo durante el cual la trama y su reconocimiento ocupan el canal.
- **Campos de dirección:** hay tres campos de dirección de 6 bytes que contienen direcciones de origen, destino inmediato y punto de conexión final, respectivamente.
- **Secuencia:** es un campo de 2 bytes que almacena los números de cuadro.
- **Datos:** este es un campo de tamaño variable que lleva los datos de las capas superiores. El tamaño máximo del campo de datos es de 2312 bytes.
- **Secuencia de verificación:** es un campo de 4 bytes que contiene información de detección de errores.

**Figura 13**  
*Cabecera subcapa MAC 802.11*



Fuente: (Tanenbaum & Wetherall, 2012)

## Capa PHY

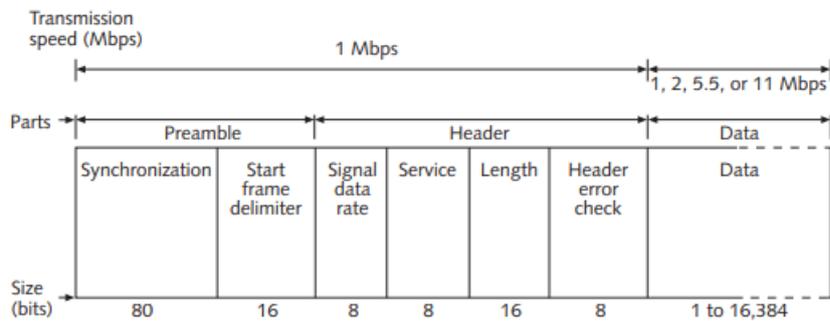
La capa física de 802.11, con la finalidad de dividir sus funcionalidades, se divide en dos subcapas denominada PLCP (Physical Layer Convergence Procedure) y PMD (Physical Medium Dependent).

La subcapa dependiente del medio físico (PMD) constituye los estándares para las características del medio inalámbrico y define el método para transmitir y recibir datos a través de ese medio. La segunda subcapa de la capa PHY es la subcapa de procedimiento de convergencia de capas (PLCP). La subcapa PLCP realiza dos funciones básicas: reformatea los datos recibidos de la capa MAC (al transmitir) en una trama que la subcapa PMD puede transmitir, y "escucha" el medio para determinar cuándo se pueden enviar los datos.

Los detalles de las subcapas PMD y PLCP para IEEE WLAN se pueden dividir en aquellos que se aplican a los diferentes estándares IEEE: 802.11b, 802.11g y 802.11n WLAN.(Ciampa, 2013)

### 1) 802.11b

Los estándares PLCP para 802.11b se basan en DSSS (Direct Sequence Spread Spectrum). La subcapa PLCP debe reformatear los datos recibidos de la subcapa MAC (al transmitir) en una trama que la subcapa PMD puede transmitir. En la Figura 14 se muestra una trama PLCP.

**Figura 14***Trama PLCP 802.11*

Fuente: (Ciampa, 2013)

Una vez que el PCLP ha creado la trama, luego lo pasa a la subcapa PMD de la capa PHY. El trabajo del PMD es traducir los 1 y 0 binarios de la trama en señales de radio que se pueden utilizar para la transmisión. PMD puede transmitir los datos a 11, 5,5, 2 o 1 Mbps.

El estándar 802.11b utiliza la banda Industrial, Científica y Médica (ISM) para sus transmisiones. El estándar 802.11b especifica 14 frecuencias que se pueden usar, comenzando en 2.412 GHz y aumentando en 0,005 GHz (excepto para el canal 14). Estos se enumeran en la Figura 15.(Ciampa, 2013)

**Figura 15***Canales y frecuencias del protocolo 802.11b*

Channel Number	Lower Frequency	Center Frequency	Upper Frequency
1	2.401	2.412	2.423
2	2.406	2.417	2.428
3	2.411	2.422	2.433
4	2.416	2.427	2.438
5	2.421	2.432	2.443
6	2.426	2.437	2.448
7	2.431	2.442	2.453
8	2.436	2.447	2.458
9	2.441	2.452	2.463
10	2.446	2.457	2.468
11	2.451	2.462	2.473
12	2.456	2.467	2.478
13	2.461	2.472	2.483
14	2.473	2.484	2.495

Fuente:(Ciampa, 2013)

El estándar 802.11b especifica dos tipos diferentes de modulación a utilizar.

- Para transmisiones a 1Mbps - DBPSK
- Para transmisiones a 2, 5,5 y 11 Mbps - DQPSK

El estándar 802.11b también describe el tipo de codificación DSSS que se utilizará.

- Para transmisiones a 1 o 2 Mbps - código Barker
- Transmisiones mayores a 2 Mbps - clave de código complementario (CCK)

## 2) 802.11g

El estándar 802.11g utiliza la frecuencia ISM de 2,4 GHz en lugar de la banda UNII utilizada por 802.11a. El estándar describe dos modos obligatorios junto con un modo opcional. El primer modo obligatorio es el mismo modo CCK utilizado por 802.11b a 11 y 5,5 Mbps. El segundo modo obligatorio para los 54 Mbps más rápidos, conocidos como PHY de tasa extendida (ERP), utiliza OFDM como el estándar 802.11a pero aún en la frecuencia de 2,4 GHz. el opcional puede transmitir a 22 Mbps y se conoce como PBCC-22 (Packet Binary Convolutional Coding), que tiene velocidades entre 6 y 54 Mbps. Los estándares 802.11g se resumen en la Figura 16.

El estándar 802.11g proporciona un mayor rendimiento que las redes 802.11b y cubre un área más amplia que las redes 802.11a mientras mantiene la compatibilidad con dispositivos 802.11b. Sin embargo, 802.11g tiene algunas desventajas. En primer lugar, ofrece solo tres canales que no se superponen, en comparación con los 23 canales de 802.11a. En segundo lugar, en un entorno donde ambos dispositivos 802.11b y 802.11g están transmitiendo, todos los dispositivos 802.11g se reducirán a solo Velocidades de 11Mbps.

**Figura 16**  
*Modulaciones 802.11g*

Transmission Speed (Mbps)	Modulation	DSSS Coding Technique
1	Differential binary phase shift keying (DBPSK)	Barker code
2	Differential quadrature phase shift keying (DQPSK)	Barker code
5.5	Differential quadrature phase shift keying (DQPSK)	Complementary code keying (CCK)
6 (mandatory speed)	OFDM	n/a
11	Differential quadrature phase shift keying (DQPSK)	Complementary code keying (CCK)
12 (mandatory speed)	OFDM	n/a
18 (optional speed)	OFDM	n/a
22 (optional speed)	PBCC-22	n/a
24 (mandatory speed)	OFDM	n/a
36 (optional speed)	OFDM	n/a
48 (optional speed)	OFDM	n/a
54 (optional speed)	OFDM	n/a

Fuente: (Ciampa, 2013)

### 3) 802.11n

Las mejoras en IEEE 802.11n PHY contribuyen a aumentar la velocidad y el alcance de redes 802.11n. Estas mejoras incluyen canales de 40 MHz y variable intervalo de guardia. Los resultados se pueden categorizar en los esquemas de modulación y codificación.

El ancho de banda del canal determina la velocidad de transmisión y es una medida de la eficiencia de la radio. Esta eficiencia, conocida como eficiencia espectral, se mide en el número de bits por Hertz. IEEE 802.11 y 802.11b usando DSSS tiene un espaciado de canal de 22 MHz de ancho mientras que 802.11a/g usando OFDM tiene Canales de 20 MHz de ancho. La eficiencia espectral de 802.11b es de 0,5 bits por Hertz (o 11 Mbps en 22 MHz), mientras que 802.11a y 802.11g tienen una mayor eficiencia espectral, tanto como 2,7 bits por Hertz a 54 Mbps.

Las WLAN IEEE 802.11n pueden usar canales de 20 MHz o 40 MHz de ancho. Los canales de 40 MHz son en realidad dos canales adyacentes de 20 MHz que están "unidos". Los dos canales se conocen como el canal primario y el canal secundario. Sin embargo, en

cambio de simplemente duplicar la tasa de datos, el uso de un canal de 40 MHz en realidad puede aumentar la tasa aún más. Cada canal de 20 MHz tiene una pequeña cantidad de espacio reservado en la parte superior y la inferior de la frecuencia para reducir la interferencia de otros canales. Debido a que la unión de canales no necesita el espacio entre los dos canales unidos, en su lugar se puede usar para aumentar el tamaño del canal aún más.

Dos medidas de seguridad pueden proteger contra la interferencia causada por la unión de canales. Primero, se requieren redes WLAN 802.11n que utilicen canales de 40 MHz para escuchar otros dispositivos inalámbricos. El AP 802.11n puede moverse automáticamente a otro canal o cambiar a una operación de 20 MHz si otro AP comienza a operar en cualquiera de las dos mitades del canal designado de 40 MHz. Esto se conoce como selección de frecuencia dinámica (DFS). En otra opción, llamada Phased Operación de coexistencia (PCO), un AP 802.11n alterna entre el uso de 20MHz y 40-canales de MHz.

Para las redes IEEE 802.11n, hay una gran cantidad de opciones para transmitir. Aunque continúa usando OFDM y un símbolo de cuatro microsegundos, similar a 802.11a/g, 802.11n aumenta el número de subportadoras en cada canal de 20 MHz de 48 a 52. El estándar IEEE 802.11n permite una selección de ocho tasas de datos para un transmisor a utilizar y también aumenta el número de transmisores permitidos a cuatro. Cuando se utilizan canales de 40 MHz, 802.11n aumenta el número de subportadoras disponibles a 108. De manera similar, se proporcionan ocho velocidades de datos para cada transmisor para el canal de 40 MHz. En total, hay 77 combinaciones posibles de estos factores: modulación, tasa de codificación convolucional, intervalo de guarda, ancho de canal y número de flujos espaciales: para 802.11n. El estándar IEEE define un esquema de modulación y codificación (MCS) que describe las diferentes combinaciones y asigna un número de índice a cada esquema. En la Figura 17 se muestra una tabla MCS parcial.

**Figura 17**  
MCS 802.11n

MCS Index	Spatial Streams	Modulation Type	Data Rate (Mbps) 20-MHz Channel		Data Rate (Mbps) 40-MHz Channel	
			800 ns GI	400 ns GI	800 ns GI	400 ns GI
0	1	BPSK	6.5	7.20	13.50	15.00
1	1	QPSK	13.00	14.40	27.00	30.00
2	1	QPSK	19.50	21.70	40.50	45.00
3	1	16-QAM	26.00	28.90	54.00	60.00
4	1	16-QAM	39.00	43.30	81.00	90.00
5	1	64-QAM	52.00	57.80	108.00	120.00
6	1	64-QAM	58.50	65.00	121.50	135.00
7	1	64-QAM	65.00	72.20	135.00	150.00
8	2	BPSK	13.00	14.40	27.00	30.00
9	2	QPSK	26.00	28.90	54.00	60.00
10	2	QPSK	39.00	43.30	81.00	90.00
11	2	16-QAM	52.00	57.80	108.00	120.00
12	2	16-QAM	78.00	86.70	162.00	180.00
13	2	64-QAM	104.00	115.60	216.00	240.00
14	2	64-QAM	117.00	130.00	243.00	270.00
15	2	64-QAM	130.00	144.40	270.00	300.00
16	3	BPSK	19.5	21.70	40.50	45.00
...	...	...	...	...	...	...
31	4	64-QAM	260.00	288.90	540.00	600.00

Fuente: (Ciampa, 2013)

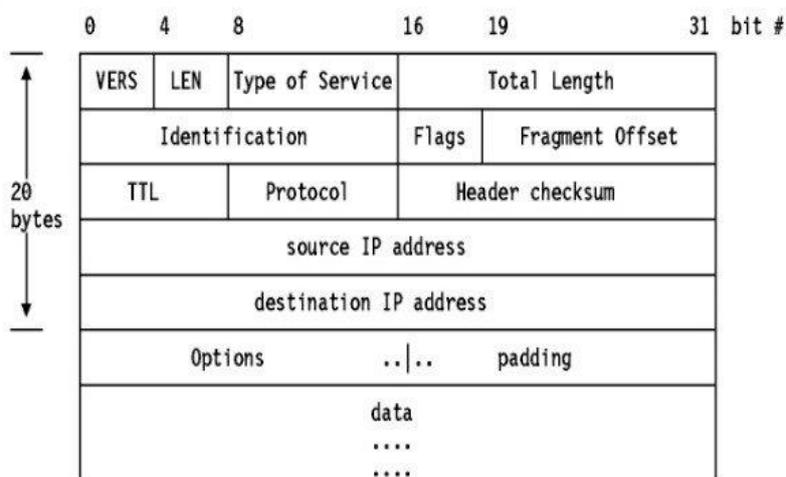
#### 2.9.4. Direccionamiento IPV4

IPv4 es un protocolo sin conexión utilizado para redes de conmutación de paquetes. Opera con un modelo de entrega de mejor esfuerzo, en el que no se garantiza la entrega, ni se asegura la secuencia adecuada o la prevención de la entrega duplicada. El Protocolo de Internet Versión 4 (IPv4) es la cuarta revisión del Protocolo de Internet y un protocolo ampliamente utilizado en la comunicación de datos a través de diferentes tipos de redes. IPv4 es un protocolo sin conexión utilizado en redes de capa de conmutación de paquetes, como Ethernet. Proporciona una conexión lógica entre dispositivos de red proporcionando identificación para cada dispositivo. Hay muchas formas de configurar IPv4 con todo tipo de dispositivos, incluidas configuraciones manuales y automáticas, según el tipo de red.

IPv4 se define y especifica en la publicación RFC 791 de IETF. IPv4 utiliza direcciones de 32 bits para la comunicación Ethernet en cinco clases: A, B, C, D y E. Las clases A, B y C tienen una longitud de bits diferente para dirigirse al host de la red. Las direcciones de clase D están reservadas para fines militares, mientras que las direcciones de clase E están reservadas para uso futuro.

IPv4 usa direccionamiento de 32 bits (4 bytes), lo que da  $2^{32}$  direcciones. Las direcciones IPv4 se escriben en notación decimal de puntos, que consta de cuatro octetos de la dirección expresados individualmente en decimal y separados por puntos, por ejemplo, 192.168.1.5. En la Figura 18 se representa la cabecera de un paquete IPV4.

**Figura 18**  
*Cabecera paquete IP*



Fuente: (altercmdweb, 2018)

- HLEN: longitud del encabezado IP (4 bits), que es el número de palabras de 32 bits en el encabezado. El valor mínimo para este campo es 5 y el máximo es 15.
- Tipo de servicio: bajo retraso, alto rendimiento, confiabilidad (8 bits)
- Longitud Total: Longitud de cabecera + Datos (16 bits), que tiene un valor mínimo de 20 bytes y el máximo de 65.535 bytes.

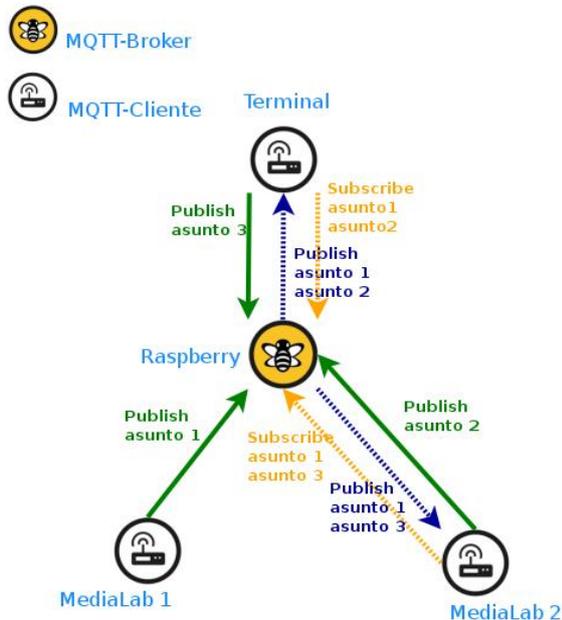
- Identificación: ID de paquete único para identificar el grupo de fragmentos de un solo datagrama IP (16 bits)
- Indicadores: 3 indicadores de 1 bit cada uno: bit reservado (debe ser cero), indicador de no fragmentar, indicador de más fragmentos (mismo orden)
- Desplazamiento de fragmento: representa el número de bytes de datos por delante del fragmento en particular en el datagrama en particular. Especificado en términos de número de 8 bytes, que tiene el valor máximo de 65.528 bytes.
- Tiempo de vida: la vida útil del datagrama (8 bits). Evita que el datagrama circule a través de la red al restringir la cantidad de saltos que toma un paquete antes de entregarlo al destino.
- Protocolo: Nombre del protocolo al que se van a pasar los datos (8 bits)
- Suma de verificación de encabezado: suma de verificación de encabezado de 16 bits para verificar errores en el encabezado del datagrama
- Dirección IP de origen: dirección IP de 32 bits del remitente
- Dirección IP de destino: dirección IP de 32 bits del receptor
- Opción: Información opcional como ruta de origen, ruta de registro. Utilizado por el administrador de red para comprobar si una ruta funciona o no.

### 2.9.5. MQTT

MQTT es un protocolo de comunicación M2M Machine to Machine, que es muy popular y usado para la comunicación entre dispositivos de IoT, las conexiones se encuentran abiertas y pueden ser reutilizadas. Su funcionamiento se lo realiza desde un servicio de mensajería push con patrón, la conexión del cliente la realizan desde el servidor central llamado bróker (Llamas, 2019). El MQTT es un protocolo que puede ser incorporado para usar en Arduino o Raspberry Pi para conseguir comunicarse con los diferentes sensores, dando un mayor resultado de usabilidad en IoT. Entre las principales versiones que presenta

MQTT están: MQTT v3.1.0-, MQTT v3.1.1- es común, MQTT v5 presenta un uso limitado y MQTT-SN (Dinculeană & Cheng, 2019).

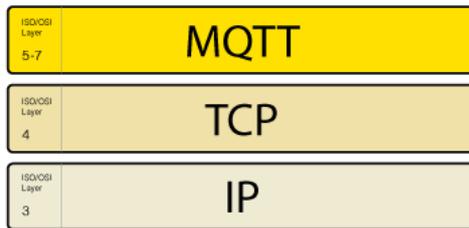
**Figura 19**  
*Ejemplo comunicación MQTT*



Fuente: (Cubides & Pulido, 2016)

La Figura 19 ilustra un ejemplo de comunicación MQTT, en donde se cuenta con un Broker MQTT (Raspberry) y tres clientes (Terminal, MediaLab1 y MediaLab2), para que exista una comunicación a través de MQTT, el Raspberry debe contar con el software necesario para ofrecer el servicio de Broker, de la misma manera los clientes, pero en el caso de estos últimos debe ser el software de cliente MQTT. Una vez que el cliente establece una conexión con el bróker, estos serán capaces de publicar información a través de “topics” o suscribirse a estos últimos para recibir la información correspondiente a un determinado asunto.

MQTT está basado en el protocolo TCP/IP, y trabaja a nivel de capa aplicación, por defecto en el puerto 1883, en la Figura 20 se muestra una representación de la ubicación del protocolo MQTT sobre las capas del modelo OSI, acompañado de los protocolos TCP e IP en las capas inferiores.

**Figura 20***Ubicación de Mqtt en el modelo OSI*

Fuente: (jecrespom, 2018)

Para que la comunicación sea posible entre dos estaciones, MQTT intercambia diferentes tipos de mensajes de control y la estructura de cada uno de estos, cuenta con tres partes importantes que son:

- **Cabecera fija.** – Es la parte del paquete MQTT en donde se indica: el “tipo de paquete de control MQTT” que se está enviando (bit 7-4), las “banderas” correspondientes a este paquete de control (bit 3-0) y el “tamaño restante”, que es un campo entero variable, hasta de cuatro bytes, que indica el tamaño del resto del paquete MQTT, después de la cabecera fija. La Figura 21 y 22, muestran la estructura de la cabecera fija y los tipos de paquetes de control, respectivamente, incluyendo en esta última el sentido de cada tipo de paquete.

**Figura 21***Cabecera Fija MQTT*

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT Control Packet type				Flags specific to each MQTT Control Packet type			
byte 2...	Remaining Length							

Fuente: Adaptado de (OASIS, 2019)

**Figura 22***Tipos de mensajes MQTT*

Name	Value	Direction of flow
Reserved	0	Forbidden
CONNECT	1	Client to Server
CONNACK	2	Server to Client
PUBLISH	3	Client to Server or Server to Client
PUBACK	4	Client to Server or Server to Client
PUBREC	5	Client to Server or
PUBREL	6	Client to Server or Server to Client
PUBCOMP	7	Client to Server or Server to Client
SUBSCRIBE	8	Client to Server
SUBACK	9	Server to Client
UNSUBSCRIBE	10	Client to Server
UNSUBACK	11	Server to Client
PINGREQ	12	Client to Server
PINGRESP	13	Server to Client
DISCONNECT	14	Client to Server or Server to Client
AUTH	15	Client to Server or Server to Client

Fuente: Adaptado de (OASIS, 2019)

- Cabecera variable.** – Algunos paquetes MQTT incluyen una cabecera variable, dentro de la cual se pueden encontrar dos campos específicos, que son: “identificador de paquete”, el cual es un entero de dos bytes (sin signo), y el campo de “propiedades”, el cual depende específicamente del tipo de paquete transmitido. En la Figura 23 se muestra la lista de tipos de paquete que usan el campo “identificador de paquete”.

**Figura 23***Paquetes de control de MQTT*

MQTT Control Packet	Packet Identifier field
CONNECT	NO
CONNACK	NO
PUBLISH	YES (If QoS > 0)
PUBACK	YES
PUBREC	YES
PUBREL	YES
PUBCOMP	YES
SUBSCRIBE	YES
SUBACK	YES
UNSUBSCRIBE	YES
UNSUBACK	YES
PINGREQ	NO
PINGRESP	NO
DISCONNECT	NO
AUTH	NO

Fuente: Adaptado de (OASIS, 2019)

- **Carga útil.** – Así mismo, algunos paquetes MQTT cuentan con una carga útil o “payload”, como el final del paquete MQTT, y en el caso específico del tipo de paquete “PUBLISH”, este contiene el mensaje de la aplicación. En la Figura 24 se muestran los tipos de paquete que utilizan carga útil.

**Figura 24***Paquetes de control que contienen carga útil*

MQTT Control Packet	Payload
CONNECT	Required
CONNACK	None
PUBLISH	Optional
PUBACK	None
PUBREC	None
PUBREL	None
PUBCOMP	None
SUBSCRIBE	Required
SUBACK	Required
UNSUBSCRIBE	Required
UNSUBACK	Required
PINGREQ	None
PINGRESP	None
DISCONNECT	None
AUTH	None

Fuente: Adaptado de (OASIS, 2019)

## **Capítulo III. DESARROLLO DE LA PROPUESTA**

### **3.1. Introducción**

En el desarrollo del presente capítulo se presenta una breve descripción de la metodología utilizada para el desarrollo del proyecto, el análisis Stakeholders que en el campo de la gestión del trabajo de investigación se constituyen como las partes afectantes de manera directa o indirecta. Adicionalmente se aplicó la técnica cualitativa Benchmark interno con el objetivo de medir la calidad del proceso productivo desarrollado, estableciendo mejoras, resultados e impacto (Calderón & Sánchez, 2016); culminando con el diseño y desarrollo del sistema en general, basados en la metodología Investigación Acción.

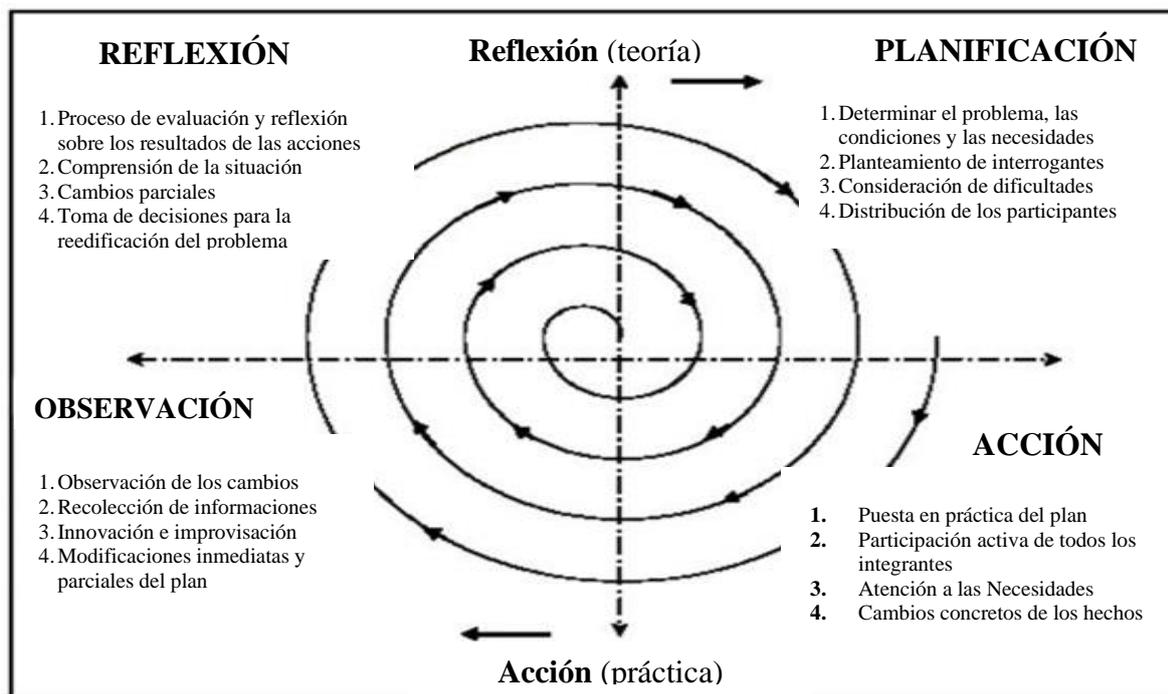
### **3.2. Metodología de diseño**

Se aplicó la metodología Investigación Acción que orienta el trabajo realizado por el proceso de participación del investigador, cuyo objetivo es obtener resultados de investigación de manera activa, enfocado en el manejo de los procesos que involucran en el cuidado del ambiente, tiene como características generales buscar efectiva solución del problema, alto grado de reflexión por parte del investigador y promueve de manera participativa la interacción entre las personas involucradas (Guevara et al., 2020).

Entre las ventajas que la Investigación Acción genera en el presente trabajo es la ejecución de la metodología en un proceso de ciclos sobre el desarrollo de un sistema embebido automatizado que permita monitorear y controlar las condiciones ambientales de un cultivo vertical, en base a una red de sensores, para mejorar la producción de cultivos hortícolas. Por tanto, el proceso metodológico sigue las etapas de reflexión, planificación, actuación y observación (Botella & Ramos, 2019).

La Figura 25, muestra una representación de las etapas de la Metodología Investigación Acción.

**Figura 25**  
*Metodología Investigación Acción*



Fuente: Adaptado de (Botella & Ramos, 2019).

Así el proceso en espiral de la investigación se inicia por la etapa de planificación para desarrollar un sistema autónomo de monitoreo y control de condiciones ambientales en cultivos verticales hortícolas, con el uso de un sistema embebido automatizado que permite monitorear las condiciones del ambiente en el que crece el cultivo, utilizando una red de sensores para mejorar la producción, donde se describe la selección del cultivo a estudiar con las características del ambiente y los requerimientos para ser implementado. La segunda etapa de identificación de los hechos se centra en el diseño de la propuesta y la selección del hardware y software a utilizar, mediante la realización de un BenchMark basado en los requerimientos del proyecto, seleccionando las herramientas que mejor se adapten a las necesidades del entorno.

La etapa de análisis se desarrolla en la esquematización de un sistema embebido, basado en redes de sensores y la integración de los requerimientos de hardware y software establecidos, para implementarlo sobre un modelo de cultivo vertical de productos hortícolas

que permita la adquisición y manipulación de las variables ambientales con un buen compromiso de resolución y precisión, adicional, el diseño de un enfoque multicriterio evaluando el desempeño y costos computacionales entre varios algoritmos de aprendizaje, de tal manera que se facilite la automatización y control de las variables ambientales de interés, eficiente y eficazmente, en base a las necesidades del cultivo seleccionado, también la etapa de implementación del software, el trabajo de campo y recolección de datos se enfoca en las variables de temperatura y humedad relativa del ambiente, humedad del suelo y luminosidad, con la finalidad de etiquetar los resultados, y finalmente en la evaluación del sistema se realiza pruebas y análisis comparativos entre productos obtenidos a través de la agricultura convencional y del mismo para determinar los beneficios y la factibilidad del proyecto.

En conclusión, la metodología permite aprender haciendo, de manera fundamental con la observación, comunicación y evaluación del proceso involucrado, que tuvo como finalidad transformar las condiciones que existían sobre el tema planteado, para beneficio de los cultivos y la inversión de las personas que los producen, donde el escenario implementa maquinaria tecnológica avanzada y saberes de la actividad horticultural para ser transformados.

### **3.3. Análisis**

Como se menciona en el apartado anterior la propuesta se desarrolla en base a cuatro etapas definidas en el presente apartado como, situación actual, propósito del sistema, análisis de requerimientos para reflexionar sobre las necesidades del proyecto a implementar además del alcance que tendrá el proyecto en donde se incluye los riesgos y restricciones a los cuales estará ligado.

### 3.3.1. Situación actual

Los diversos desafíos y problemas en la agricultura tradicional son escasez de comida, inseguridad, crecimiento poblacional, lugares de cultivo ubicados únicamente en zonas rurales y clima impredecible. Una alternativa a la agricultura convencional, para aumentar la producción sostenible es la agricultura de precisión, mediante la cual se controlan las variables del ambiente y por ende se facilite resolver varios de estos problemas.

El sistema propuesto en el presente documento es eficiente, utiliza agua y nutrientes para el crecimiento óptimo de la planta. La agricultura de precisión y la agricultura vertical se ha expandido a nivel mundial en los últimos años como un modelo sostenible. El avance en la tecnología ha ganado importancia para automatizar la agricultura sistema y utilizar eficientemente recursos como el agua, fertilizantes en lugar de los métodos de cultivo convencionales, los agricultores urbanos pueden adoptar la agricultura con métodos de ambiente controlado sin pesticidas y Organismos Genéticamente Modificados.

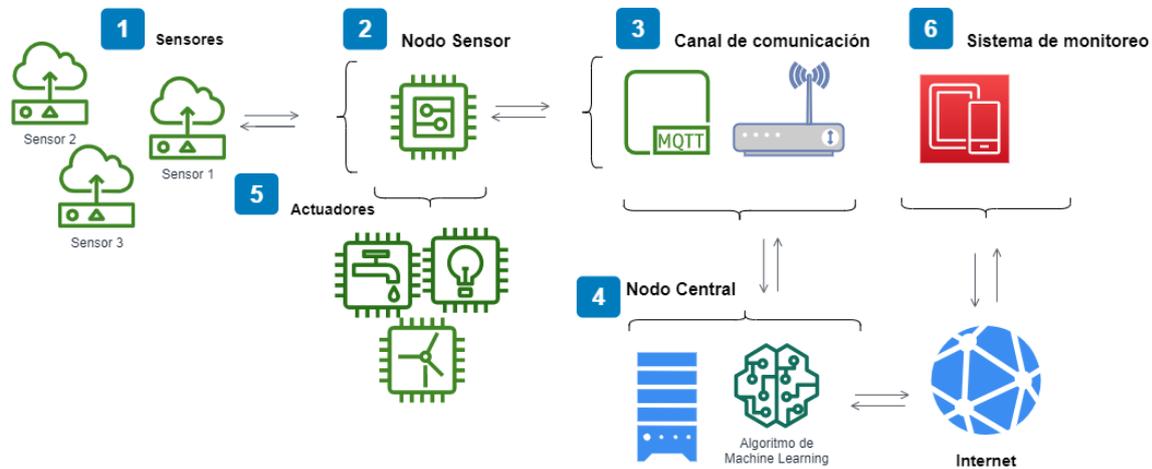
En este contexto el presente apartado describe las razones esenciales por las que se desarrolló la propuesta con las bases y lineamientos del entorno que se involucra, así se encuentra los requerimientos humanos y materiales aunados con el software y hardware para la construcción del sistema, siguiendo la secuencia de los pasos planteados en la metodología y con el propósito de producir alimentos de mejor calidad, dentro y fuera de temporada, para abastecer la demanda alimentaria ocasionada por el crecimiento poblacional, reducir los costos en la producción y comercialización de productos agrícolas, brindar alimentos para una dieta más saludable, optimizar el uso del suelo a través de diferentes técnicas de cultivo como la agricultura vertical, y minimizar el uso de agua, fertilizantes y pesticidas en la agricultura.

### 3.3.2. Propósito y descripción general del sistema

El propósito general del presente proyecto de investigación es desarrollar un sistema embebido automatizado que permita monitorear y controlar las condiciones ambientales de un cultivo vertical, en base a una red de sensores, para lo cual se pretende seleccionar un cultivo hortícola de ciclo corto y construir una estructura vertical de dos niveles de ambiente controlado “indoor”, en el que se tomó en cuenta las variables ambientales básicas para el desarrollo de una planta como son: luminosidad, humedad del suelo, temperatura y humedad relativa.

El sistema permite medir y monitorear las variables ambientales antes mencionadas, a través de una red de sensores y un algoritmo basado en computadora, para que este último posterior a una etapa de aprendizaje, permita controlar y automatizar las medidas correctivas, con la finalidad de optimizar la producción de los cultivos en una comparativa a la agricultura tradicional.

En la Figura 26, se muestra una descripción general del sistema basada en bloques, donde se puede apreciar se ha contemplado dos nodos a los cuales se encuentran conectados los sensores que permitirán medir las variables ambientales de cada nivel dentro de la estructura de agricultura vertical, así como también los actuadores que modificarán dichas variables como medida correctiva al estado de cada uno de los niveles. A través de una red de área local estos intercambiarán información con el nodo central, el cual se encargará de procesar la información recibida por parte de los sensores, solicitar medidas correctivas en caso de haberlas, almacenar la información en una base de datos local y publicar esta información a través de Internet con la finalidad de monitorear y controlar el sistema a través de una aplicación móvil.

**Figura 26***Diagrama de bloques de la propuesta***Sistema autónomo de monitoreo y control de variables ambientales**

Fuente:(Pichucho, 2007)

**3.3.3. Análisis de los requerimientos**

En el presente apartado se determina los diversos requerimientos para la correcta ejecución de la propuesta, se constituyen por Stakeholders, requisitos de diseño, nomenclatura de requerimientos, requerimientos de Stakeholders, requerimientos funcionales del sistema y requerimientos de arquitectura.

**3.3.3.1. Determinación de los Stakeholders**

Los Stakeholders se encuentran inmersos de forma directa e indirecta ya que son parte de un proyecto. Por tal razón es importante dar a conocer los requerimientos necesarios en cuanto a usuarios como se muestra en la Tabla 1. Por cuanto serán las personas que están trabajando directamente y serán los pilares fundamentales en la investigación ya que ayudarán a llevar a buen término el estudio.

**Tabla 1***Lista de participantes*

<b>N°</b>	<b>Descripción</b>
1	Ingeniero Agrónomo experto en horticultura Clemente Alvarado
2	Director del proyecto Msc. Luis Suárez
3	Autor de la investigación Sr. Cristian Ruiz
4	Asesor1 Msc. Fabián Cuzme
5	Asesor2 Msc. Edgar Maya

**3.3.3.2. Requisitos del diseño**

En el presente apartado se describe tres puntos importantes de los requerimientos de Stakeholders, sistema y diseño con la finalidad de presentar el análisis de los datos realizado desde la situación actual, el cumplimiento de los objetivos como el alcance, hasta finalizar con datos reales y actuales al contexto de desarrollo. De este modo la información presentada beneficiará a los lectores con ideas consolidadas para mantener cultivos productivos y generadores de otras fuentes necesarias para la sociedad.

**3.3.3.3. Nomenclatura de requerimientos**

El manejo de datos complejos atribuye el requerimiento de nomencladores que sintetice la extensión de estos para mayor facilidad de entendimiento, así la Tabla 2 muestra requerimientos de función del Stakeholders, sistema y arquitectura con el objetivo de mejorar en los lectores su comprensión.

**Tabla 2***Nomenclatura de requerimientos*

<b>Descripción</b>	<b>Abreviaturas</b>
Requerimientos de Stakeholders	StSR
Requerimiento Funcional del sistema	SySR
Requerimiento de Arquitectura	SRSR

### 3.3.3.4. *Requerimientos de Stakeholders*

Existen requisitos que se debe cumplir en los requerimientos de Stakeholders, por tal razón se indica en la Tabla 3 un diseño para cubrir las demandas, ya que serán los encargados de la evaluación del dispositivo luego de haber sido implementado.

**Tabla 3**  
*Requerimientos de Stakeholders*

		<b>StSR</b>		
#	<b>Requerimiento</b>	<b>Prioridad</b>		
		<b>Alta</b>	<b>Media</b>	<b>Baja</b>
<b>Requerimientos operacionales</b>				
<b>StSR 1</b>	El sistema debe implementarse en un cultivo vertical	X		
<b>StSR 2</b>	El entorno del sistema debe albergar actuadores para regular variables ambientales.	X		
<b>StSR 3</b>	El sistema debe ser eficiente energicamente			X
<b>StSR 4</b>	El sistema debe recolectar datos de las variables ambientales	X		
<b>StSR 5</b>	El nodo central debe contar con conexión a Internet	X		
<b>StSR 6</b>	El sistema de iluminación debe contar con una PPFD de al menos 600 $\mu\text{mol} / \text{m}^2 / \text{s}$	X		
<b>StSR 7</b>	El espectro del sistema de iluminación debe contener las componentes de luz fundamentales para el crecimiento de la planta.	X		
<b>StSR 8</b>	El sistema de riego debe alcanzar a cada una de las plantas.	X		
<b>Requerimientos de usuarios</b>				
<b>StSR 9</b>	Monitoreo de datos del sistema	X		
<b>StSR 10</b>	Presentación clara y precisa de los datos y de fácil interacción	X		
<b>StSR 11</b>	La implementación del sistema debe ser económicamente eficiente	X		
<b>StSR 12</b>	Los dispositivos de hardware deben ser de fácil adquisición		X	

### 3.3.3.5. Requisitos funcionales del sistema

Los sistemas implementados hacen necesario el uso de funciones exactas que especifican las formas de comportamiento de los patrones estudiados, en la Tabla 4 se describe en detalle de diseño de un sistema electrónico basado en redes de sensores inalámbricos, entre los estudios realizados detecto formas y patrones en los cultivos.

**Tabla 4**  
*Requerimientos funcionales del sistema*

<b>SySR</b>				
#	<b>Requerimientos Funcionales del Sistema</b> <b>Requerimiento</b>	<b>Prioridad</b>		
		Alta	Media	Baja
<b>Requerimiento de uso</b>				
<b>SySR 1</b>	Botones on/off y reset de los nodos			X
<b>SySR 2</b>	El sistema debe precautelar la desactivación de todos los actuadores, en caso de daños en los nodos que lo controlan.	X		
<b>Requerimientos de performance</b>				
<b>SySR 3</b>	Los nodos deben poder conectarse al servidor a través de una red Wifi	X		
<b>SySR 4</b>	Los nodos deben enviar los datos recolectados al nodo central.	X		
<b>SySR 5</b>	El nodo central debe recibir los datos enviados por los nodos y a su vez, publicarlos para un monitoreo constante dentro y fuera de la red privada.	X		
<b>Requerimientos de interfaces</b>				
<b>SySR 6</b>	Se debe garantizar la integración de todos los sensores y actuadores a los nodos.	X		
<b>SySR 7</b>	El Microcontrolador debe ser capaz de interpretar los datos tomados a través de los sensores.	X		
<b>SySR 8</b>	Los componentes de acople de los sensores y actuadores deben ser compatibles con las características del microcontrolador.	X		
<b>Requerimientos físicos</b>				
<b>SySR 9</b>	Los nodos deben tener un tamaño reducido	X		
<b>SySR 10</b>	Los nodos deben ser impermeabilizados			X

<b>SySR 11</b>	Los sensores deberán ser ubicados en lugares estratégicos	X
----------------	-----------------------------------------------------------	---

### 3.3.3.6. *Requerimientos de arquitectura*

Se hace referencia al subsistema electrónico con el objetivo de dar elección al hardware necesario, que en su construcción requiere de los componentes de un software hardware, por tanto, la Tabla 5 enlista los requisitos arquitectónicos necesarios.

**Tabla 5**  
*Requerimientos de arquitectura*

		<b>SRSH</b>		
#	Requerimientos de Arquitectura	Prioridad		
		Alta	Media	Baja
<b>Requerimientos lógicos</b>				
<b>SRSH 1</b>	Los nodos deben contar con al menos, una entrada analógica y cinco entradas/salidas digitales		X	
<b>SRSH 2</b>	Los nodos deben ser capaces de enviar los datos recolectados al nodo central.		X	
<b>SRSH 3</b>	Adaptación y calibración de los sensores al sistema embebido	X		
<b>SRSH 4</b>	Adaptación de los actuadores al sistema embebido	X		
<b>Requerimientos de diseño</b>				
<b>SRSH 5</b>	Integración de los diferentes elementos en una placa compacta		X	
<b>SRSH 6</b>	Protección de los circuitos ante las variables ambientales	X		
<b>SRSH 7</b>	Los nodos deberían enviar los datos a través del protocolo 802.11		X	
<b>SRSH8</b>	El hardware y software del sistema deben ser de libre distribución		X	
<b>Requerimientos de software</b>				
<b>SRSH 9</b>	Lenguajes de programación a utilizar deben ser de alto nivel		X	
<b>SRSH 10</b>	Las librerías de los sensores y/o plugins de los sensores deben ser compatibles con los microcontroladores	X		
<b>SRSH 11</b>	Aplicación de monitoreo, de fácil uso e implementación		X	
<b>Requerimientos de hardware</b>				
<b>SRSH 12</b>	Los nodos deben contar con una capacidad de procesamiento adecuado, de acuerdo con las tareas a realizar		X	
<b>SRSH 13</b>	Módulo para conexión inalámbrica integrado en la placa de desarrollo	X		
<b>Requerimientos eléctricos</b>				
<b>SRSH 14</b>	Se debe contar con toma eléctrica de 110 AC y fuentes de alimentación de 5V DC	X		

### **3.3.4. Alcance de la propuesta**

A continuación, se listan las restricciones y los riesgos a los cuales está sujeto el proyecto, de manera general, con la final de exponer los temas no considerados o que no están al alcance del desarrollo del mismo.

#### **3.3.4.1. Restricciones**

- El estudio se limita al estudio de las variables ambientales previamente mencionadas: luminosidad, humedad del suelo, temperatura y humedad relativa.
- El estudio se limita únicamente al cultivo que se seleccionará en la etapa de planeación.
- Para mejores resultados posiblemente sea necesario tomar en cuenta factores extra que no están dentro de este estudio.
- El desarrollo y la implementación del sistema serán realizados en una pequeña escala.
- Únicamente el desarrollador puede hacer modificaciones en los comportamientos del sistema.
- Para su correcto funcionamiento el sistema deberá permanecer encendido todo el tiempo.

#### **3.3.4.2. Riesgos**

- Fallos de energía eléctrica.
- Pérdida de conexión a Internet.

### **3.4. Desarrollo de la propuesta**

En la primera fase de la investigación del proyecto se definen las metas para desarrollar un sistema embebido automatizado que permita monitorear y controlar las condiciones ambientales de un cultivo vertical, en base a una red de sensores, para mejorar la producción de cultivos hortícolas, por lo que se presenta las fases a continuación:

### 3.4.1. Selección del cultivo

Se toma en cuenta para la selección del cultivo la taxonomía de algunas hortalizas, de las cuales define la familia, género y el tiempo de cosecha, para seleccionar el cultivo adecuado para la investigación.

**Tabla 6**

*Taxonomía de los cultivos de hortalizas*

<b>Hortaliza</b>	<b>Familia</b>	<b>Género</b>	<b>Tiempo (días)</b>
<b>Cilantro</b>	Apiaceae	Coriandrum	60 a 70
<b>Acelga</b>	Chenopodiaceae	Beta	50 a 60
<b>Coliflor</b>	Brassicaceae	Brassica	90 a 120
<b>Zanahoria</b>	Apiaceae	Daucus	60 a 70
<b>Lechuga</b>	Asteraceae	Lactuca	60 a 80
<b>Rábano</b>	Brassicaceae	Raphanus	50 a 70
<b>Pimiento</b>	Solanaceae	Capsicum	50 a 70
<b>Espinaca</b>	Chenopodiaceae	Spinacia	60 a 80
<b>Cebolla</b>	Brassicaceae	Brassica	160 a 200
<b>Col</b>	Brassicaceae	Brassica	75 a 88

Fuente: (Haro, 2012b)

En base a los datos mostrados en la Tabla 6, y en específico a la columna que nos muestra el tiempo de cosecha en días, se selecciona la Acelga, como caso de estudio para la implementación del Sistema de Agricultura Vertical, debido a que su tiempo de cosecha tiende a ser más eficiente en comparación a las otras variedades de hortalizas.

La acelga, *beta vulgaris* var. *cycla*/chenopodiaceae, pertenece a la familia de las Quenopodiáceas (Molina, 2000). Se caracteriza por tener hojas grandes de color verde, carnosas con pencas blancas, se utiliza para el consumo humano. Este tipo de hortaliza presente beneficios medicinales y alimenticios, durante el cultivo hay que tomar en consideración los ciclos de crecimiento para que no se pase el tiempo de cosecha porque de lo contrario saldrá flor y esto impedirá su comercialización por que se encuentra madura.

La acelga se produce en clima templado, sin embargo, es perjudicial para el sembradío los cambios drásticos de temperatura, detiene su desarrollo con temperaturas menor a 5°C que

están bajo cero, no es necesario disponer de mucha luz siendo así que el exceso le perjudica, en cultivos de invernadero la humedad oscila entre 60 y 90% (Costa, 2015). Presenta una raíz larga que se adentra en la tierra además es fibrosa, sus hojas presentan forma oval, muestra un peciolo ancho y a la vez largo que también contribuye una parte para ser comestible, su color varía dependiendo de la variedad, pero va de verde oscuro marcado a verde claro. La semilla se presenta poligérmica siendo así que pueden nacer de ella 2 o hasta 3 plantas, su ciclo germinativo es de 4 años.

La siembra de las acelgas tiene la característica de ser una planta bianual de ciclo largo que no desarrolla raíz o a su vez no da un fruto comestible más bien es comestible sus hojas. La variedad de acelgas es un poco escasa, por tanto, menciona Hidalgo, Heredia y Yanez (Hidalgo et al., 2018) que la clasificación se da según el color, tamaño, hojas, peciolos, grosor de la penca, y recuperación rápida luego del corte de sus hojas.

Adicional se selecciona la lechuga que es conocida científicamente como *Lactuca sativa*, es una planta que pertenece al grupo herbáceo (De Grazia et al., 2001). Es un alimento que comúnmente se encuentra en la mesa de los consumidores, además es fácil de encontrar, está disponible en todos lados y aun precio accesible, siendo así que está en constante siembra y cosecha para cubrir la demanda de esta hortaliza, no requiere de mucho cuidado y de inversiones grandes.

Para sembrar la lechuga se necesita de un clima fresco para progresar, su cosecha se la realiza durante todo el año, se presenta como una planta fuerte y parca por cuanto no necesita de muchos cuidados ni tampoco se ve afectada por las plagas, por tanto, es un cultivo adaptable a cualquier tipo de terreno y a climas variados. Durante el proceso de crecimiento se requiere de riego constante hasta que salga el cogollo, sin embargo, puede resistir moderadamente a la falta de agua, pero afecta la calidad del producto final, también necesita

de suelo ricos que tengan en sus componentes nutrientes y que a la vez se encuentren bien drenados que no presenten exceso de arcilla, así también debe tener un pH que oscile entre 6 y 6.8.

A continuación, se describe el diseño, iluminación, medio de crecimiento y características de sostenibilidad de los cultivos que pueden crecer bajo ambiente controlado y en estructuras de niveles. En consecuencia, la planta herbácea acelga para ser cultivada bajo el proceso monitoreado de crecimiento por las características que se presentan en las fichas técnicas de la Tabla 7.

**Tabla 7**

*Ficha Técnica de la acelga*

<b>Ficha Técnica de la Acelga</b>		
<b>Nombre científico:</b>	Beta vulgaris	Imagen de la acelga
<b>Nombre del cultivo:</b>	Acelga	
<b>Descripción:</b>	El crecimiento de esta planta es bianual, emite el botón floral hasta el segundo año de siembra, el cultivo se lo hace para aprovechar sus hojas que se presentan agradable al paladar y que además presentan vitamina A, B1, B2 y C.	
<b>Temperatura:</b>	Germinación 19 a 24°C Desarrollo vegetativo. 15 25°C Soporte: min de 6°C a max 33°C	
<b>Humedad relativa:</b>	60% a 90%	
<b>Luminosidad</b>	10,000 a 40,000 luxes incluyendo plástico de invernadero para reducir la intensidad. Cultivo de días largos, 12 horas de iluminación.	
<b>Días de cosecha:</b>	Entre 50 y 70 días	
<b>Solución nutritiva:</b>	Se coloca cada 3 días	
<b>Es conveniente la asociación con:</b>	Lechuga y escarola	
<b>Cosecha:</b>	Cuando sus hojas miden entre 25cm aproximadamente, se realiza el primer corte y se esto sucede a los 50 días desde el momento de la siembra, los cortes se lo realizan cada 8 a 12 días.	
<b>Información alimentaria:</b>	<b>Aportes:</b>	Energía: 29,70 kcal Proteína: 1,88gr. Grasa total: 0,20gr Agua: 95,1 gr. Fibra: 1,20gr.
	<b>Vitaminas:</b>	Vitamina B1: 0,05 mg

		Vitamina B2: 0,05 mg Vitamina B6: 0,11 mg Vitamina B12: 0,00mg Vitamina A: 335, 17 Ug.
	<b>Minerales:</b>	Calcio: 105,00mg. Hierro: 3,30mg. Yodo: 39,06mg Magnesio: 76,00mg Zinc: 0,03mg Sodio: 150,00mg Potasio: 380,00mg Fósforo: 0,00mg

Fuente: Hernández y Sarmiento (2017, p. 182)

### 3.4.2. Selección del Hardware

El BenchMark es el proceso que guió la elección del hardware y software a utilizados basado en los requerimientos del proyecto, se seleccionó las herramientas que mejor se adaptaron a las necesidades del entorno estudiado. En este sentido, se aplicó una tabla comparativa con la valoración de 1 para si cumple y 0 para el sistema que no posee los requerimientos como se presenta en la tabla, de este modo se inició con la selección del hardware y software del nodo de procesamiento de los sensores.

**Tabla 8**

*Nomenclatura de evaluación de requerimientos*

Descripción	Valor
Si cumple	1
No cumple	0

- **Selección del nodo de procesamiento**

En una preselección se obtiene una lista de dos sistemas como: Orange Pi y Raspberry Pi a los cuales se aplica la comparación de los BenchMark con el uso de la tabla comparativa 8, a fin de seleccionar el sistema óptimo para el ambiente en el que se estudia el cultivo hortícola.

**Tabla 9***Análisis comparativo del nodo de procesamiento*

Hardware	StSR	StSR	StSR	StSR	SySR	SRSR	SRSR	Valoración
	3	5	11	13	9	8	12	
Raspberry Pi	1	1	1	1	1	1	1	7
Orange Pi	1	1	0	0	1	1	1	5

**Figura 27***Raspberry Pi III*

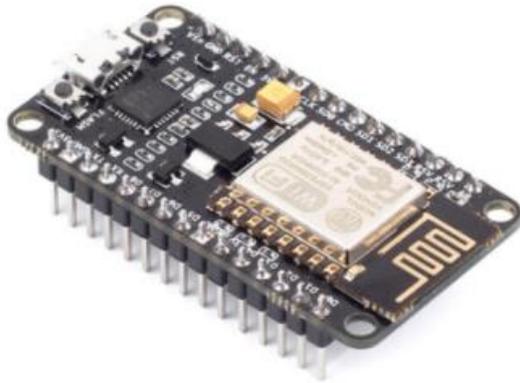
Fuente: Pastor, (2018)

En este contexto, de acuerdo con los resultados de la tabla comparativa 9 se Raspberry Pi III, cuya placa se ilustra en la Figura 27 que cumple la función de una plataforma de código abierto, cuyas ventajas principales en comparativa con Orange Pi, son los costos y facilidad de adquisición en el mercado local.

**Tabla 10***Análisis comparativo de los nodos sensores*

Hardware	Requerimientos							Valoración
	StSR	SRHS	SRSR	SRSR	StSR	StSR	StSR	
	5	10	8	13	14	15	16	
Node MCU	1	1	1	0	1	1	1	6
Arduino Mega	1	1	0	0	1	1	0	4

**Figura 28**  
*Node MCU*



Fuente: Pastor, (2018)

En base a la Tabla 10, se seleccionó a NodeMCU, como placa de desarrollo para los nodos sensores, debido a su principal ventaja en cuanto a costos, e integración del módulo ESP8266 para conexión WiFi. En la Figura 28, se muestra una fotografía de la placa de desarrollo Node MCU.

- **Selección de sensores**

A continuación, se seleccionó los sensores de acuerdo con la tabla comparación de sistemas que brinda el mercado, por lo cual se determinó las variables que se analizó en el proyecto como: humedad relativa, humedad de suelo, temperatura y luminosidad. El análisis Benchmark se centra en los criterios de disponibilidad, fiabilidad, estabilidad, tiempo de respuesta y costo en el mercado

- **Sensor de humedad relativa y temperatura**

Se define una lista de tres sensores que se adecúan a los criterios enmarcados para la selección y los requerimientos implantados en el proyecto, estos pertenecen al conjunto de sensores digitales de la biblioteca de MicroPython. Entre las características esenciales tenemos medidor de humedad relativa y temperatura a la vez, porque cuenta con un termistor

y un sensor capacitivo, en el chip recepta los datos análogamente y los envía por cable a la base de datos. Por tanto, según sus características para el presente estudio se seleccionó DHT 11 y DHT 22 por los rangos de medición, resolución y tiempos de lectura presentados en la Tabla 11 y los posibles criterios es beneficio del estudio.

**Tabla 11**

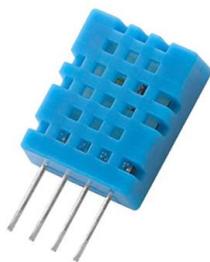
*Análisis comparativo de sensores de humedad relativa y temperatura*

<b>Hardware</b>	<b>StSR</b>	<b>StSR</b>	<b>StSR</b>	<b>SRSR</b>	<b>StSR</b>	<b>Valoración</b>
	<b>4</b>	<b>11</b>	<b>12</b>	<b>3</b>	<b>10</b>	
DHT11	1	1	1	1	1	6
DHT22	1	0	1	1	1	5

Por consecuencia, el sensor de humedad relativa y temperatura DHT 11 (Figura 29) es seleccionado para ser implementado en el cultivo hortícola por los criterios de buen funcionamiento, posee el requerimiento de rápida respuesta y mantiene características óptimas que permite al investigador realizar el proyecto de manera factible.

**Figura 29**

*DHT 11 Sensor de temperatura y humedad*



**Fuente: Hernández (2018)**

- **Sensor de humedad de suelo**

Para la selección se estable los sensores FC 28 y YL 69 por ser los más destacados en las características de mercado, entre lo relevante se encontró que los resultados de los dos

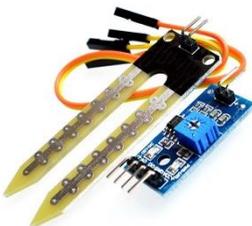
sistemas son presentados de 0 a 1023, sin embargo, los dos presentan características que difieren el uno del otro.

**Tabla 12**  
*Análisis comparativo de sensores de humedad de suelo*

Hardware	StSR	StSR	StSR	SRSR	StSR	Valoración
	4	11	12	3	10	
FC28	1	1	1	1	1	6
YL69	1	0	0	1	1	4

De este modo, las características de FC 28 (Figura 30), permite al investigador la factibilidad para ser utilizado por lo que fue seleccionado. Presenta compatibilidad con Arduino y Node MCU, el nombre comercial es Higómetro FC 28 (Figura 30) entre las características más relevantes presenta potencialidad para ajustar el valor umbral, que permite controlar la humedad del suelo y la activación de bombeo de manera eficiente. Las especificaciones muestran un chip comparador LM393, corriente 35mA, superficie de electrodo en estaño, el voltaje de alimentación (VCC): 3.3V-5V, la señal analógica 0V hasta 5V y la señal de salida 3.3V/5V (TTL).

**Figura 30**  
*Sensor FC 28*



Fuente: Llamas (2021)

Por las dimensiones son fáciles de transportar con medidas mínimas de panel PCB con 30mmx15mm y el sensor de 60mmx20mm. Mientras los pines son VCC (5V), GND, interfaz

salida digital (0 y 1) e interfaz de salida analógica AO. En el trabajo de campo la medición estándar permite al investigador el monitoreo con el valor de 0 si es suelo se encuentra sumergido en agua y 1023 si está muy seco, mientras el valor intermedio se fija de 600 a 700. En consecuencia, el potenciómetro permitió ajustar la medida para recibir una señal LOW cuando los niveles de humedad son adecuados y HIGH cuando el exceso de humedad está presente.

- **Sensor de luminosidad**

Se considera los sensores LDR y TSL 2560 para la selección por ser los dispositivos que miden la intensidad de luz en el espacio del cultivo y reflejan los datos necesarios para el estudio, estos se encuentran a disposición de las tiendas del sector por lo que hacen factible el uso y son ajustables a las especificaciones requeridas por el investigador.

**Tabla 13**  
*Análisis comparativo de sensores de luminosidad*

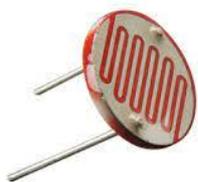
Hardware	StSR	StSR	StSR	SRSH	SRSH	StSR	Valoración
	6	11	12	8	13	14	
LDR	1	1	1	1	1	1	6
TSL2560	1	0	0	1	1	1	4

En concordancia a los resultados obtenidos se selecciona el sensor LDR por la susceptibilidad a los espectros de luz, por las características óptimas de voltaje de funcionamiento de 3.3 – 5 V con una corriente de entrada de 60mA y un potenciómetro ajustable a los requerimientos del cultivo. Posee conexión de 3 pines como el voltaje de alimentación VCC, conecta el cátodo de potencia GND y envía señal digital de salida TTL 3.3V llamado DO, el ambiente permite una temperatura de -25°C – 80°C, mientras el

almacenamiento es de  $-65^{\circ}\text{C}$  –  $150^{\circ}\text{C}$  y finalmente el tamaño estándar es de  $3.20 \times 1.40 \text{cm} / 1.26 \times 0.55''$

**Figura 31**

*Sensor de luminosidad LDR*



Fuente: Villarruel (2019)

- **Selección de actuadores**
- **Selección de lámparas**

En la Tabla 14, se presenta el análisis comparativo para la selección de las lámparas de iluminación artificial, este es uno de los componentes más importantes del sistema ya que dicha iluminación es la que las plantas aprovecharán para realizar el proceso de la fotosíntesis, por lo que es de suma importancia seleccionar adecuadamente, este actuador

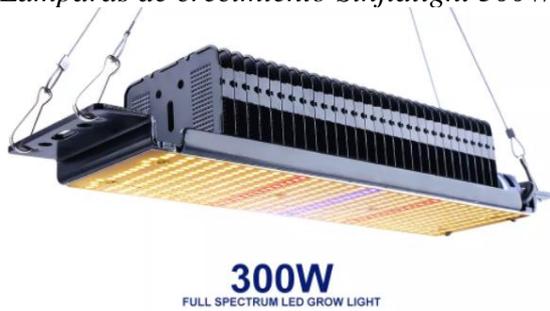
**Tabla 14**

*Análisis comparativo de lámparas de crecimiento*

Hardware	StSR	StSR	StSR	SRSH	SRSH	StSR	Valoración
	6	11	12	8	13	14	
Sinjialight 300W	1	1	1	1	1	1	6
HF-GLP1000W	0	1	0	1	1	1	4

En base a los requerimientos del sistema, se selecciona la lámpara de crecimiento Sinjialight de 300W (Figura 32), ya que en el análisis comparativo obtiene la mayor puntuación.

**Figura 32**  
Lámparas de crecimiento Sinjialight 300W



Fuente: (Tian, 2020)

- **Selección de electroválvulas**

La electroválvula es el actuador que permitirá la activación del regadío cuando los cultivos de cada nivel, así lo requieran. Por tal motivo, en la Tabla 15, se reúnen los requerimientos que involucran a este dispositivo y se realiza el análisis comparativo respectivo.

**Tabla 15**  
*Análisis comparativo electroválvulas*

Hardware	StSR	StSR	StSR	SRSH	SRSH	StSR	Valoración
	4	11	12	3	14	10	
Electroválvula	1	1	1	1	1	1	7
110V AC							
Electroválvula	1	1	1	1	0	1	6
12V DC							

Teniendo una mejor adaptación a los requerimientos, se elige a la electroválvula de 110V AC, cuya imagen se muestra en la Figura 33.

**Figura 33**  
*Electroválvula 110V AC*



Fuente: (Tekmikro, 2021)

- **Selección de los ventiladores**

A través de la activación de los ventiladores, el sistema automatizado, proporcionará una temperatura dentro de los rangos óptimos del cultivo seleccionado, por lo que es importante seleccionar adecuadamente estos dispositivos, y por ello se ha elaborado la Tabla 16, en donde se realiza una comparativa para una óptima selección.

**Tabla 16**  
*Análisis comparativo ventiladores*

Hardware	StSR	StSR	StSR	SRSR	SRSR	StSR	Valoración
	<b>4</b>	<b>11</b>	<b>12</b>	<b>3</b>	<b>14</b>	<b>10</b>	
Ventilador 5V	1	1	1	1	1	1	7
Ventilador 12V	1	1	1	1	0	1	6

Gracias a su mejor adaptación con el sistema a desarrollar, se ha elegido el ventilador de 5V DC, el cual tiene la apariencia de la Figura 34.

**Figura 34**  
*Ventilador 5V DC*



Fuente: (Electronics, 2021)

### **3.4.3. Selección de software**

Uno de los componentes esenciales de la tecnología está constituido por el software y las características determinantes de cada uno sobre instrucciones, estructuras y descripción de información e los programas (Colina, 2019). Por tanto, en el presente apartado se realiza la selección de los softwares utilizados en el proceso de la investigación.

- **Software de programación NodeMCU**

Para la construcción de la Unidad Microcontroladora de los Nodos se considera los softwares LUA y Arduino de los cuales se establece Arduino el software a utilizar por ser un lenguaje de programación óptimo, adicional contiene dos funciones básicas setup y loop para lo cual se utiliza processing y sketch que finalmente permitirán compilar la información en la placa de Arduino. Por tanto, el código de programación para obtener los datos que arrojan los diferentes sensores se encuentra en el anexo ...

- **Software para el almacenamiento de datos**

Para evitar sobrecarga en el almacenamiento de los datos se toma en cuenta las necesidades o requerimientos para el desarrollo del proyecto, por lo que se define MySQL y PostgreSQL para realizar la preselección, a continuación, se analiza los requerimientos para presentación clara y precisa de los datos y de fácil interacción, lenguajes de programación a

utilizar deben de ser de alto nivel y de código abierto, los microcontroladores deben contar con pines de salida para activar/desactivar los actuadores del sistema y los nodos deben contar con una capacidad de procesamiento adecuado.

De este modo la Tabla 17 compila los datos de los requerimientos del sistema y de interfases del software.

**Tabla 17**  
*Selección de software*

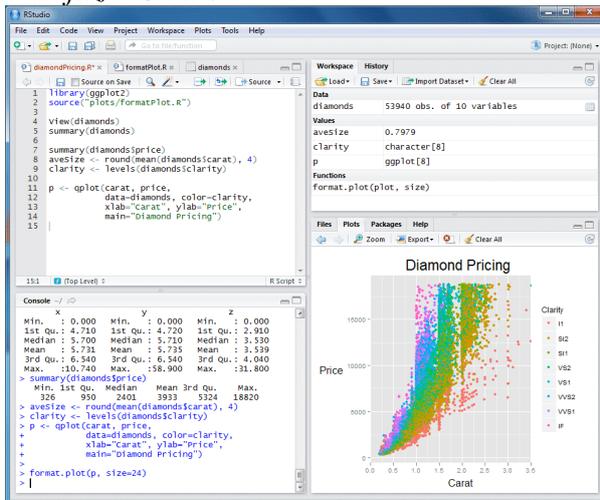
Software	Requerimientos				Valoración total
	StSR 6	StSR 7	SRSR 9	SRSR 12	
MySQL	1	0	1	0	2
PostgreSQL	0	1	0	0	1

Con la valoración total de 2 MySQL cumple con los requerimientos de software para almacenar los datos, posee licencia de código abierto con el objetivo de mejorar el desarrollo de la web y almacenar los datos.

- **Software para la minería de datos**

Para el presente apartado se selecciona el programa estadístico RStudio por poseer código abierto y trabaja bajo un lenguaje de programación en cuatro paneles, adicional permite la programación de scripts, también cuenta con un manual de guía en los archivos para su fácil desarrollo y alta calidad de los cheatsheets. Los paneles presentan al lado inferior izquierdo la ejecución de códigos y resultados, en la parte superior del panel antes mencionado se encuentra los ficheros en pestañas y el editor de los códigos a ejecutar, a la derecha en la parte superior se encuentra el historial de comandos ejecutados y finalmente en la parte inferior derecha se encuentra archivos, gráficas y la página de ayuda. En la Figura 35 se muestra la interfaz de R Studio.

**Figura 35**  
*Interfaz R Studio*



**Fuente: Calvo (2018)**

- **Software para el monitoreo del sistema**

Para la selección del software en el apartado de visualización de datos se toma en cuenta Aws, Azure y Firebase. Con el análisis previamente de los requerimientos que se presentan en la Tabla 18, de lo cual la investigación deja como el software óptimo Firebase para la apreciación de los resultados, considerando características de fácil manejo que permitan al investigador interpretar los datos de manera sencilla.

**Tabla 18**  
*Software para la visualización de resultados*

Software	Requerimientos				Valoración total
	StSR 6	StSR 7	SRSR 9	SRSR 12	
Aws	1	0	1	0	2
Azure	0	1	1	0	1
Firestore	1	1	1	0	3

- **Software para controlar el sistema**

Para el control del sistema se selecciona Node red a fin de controlar raspberry por ser una de las herramientas basada en flujo inventada en 1970, fue utilizado para definir patrones de control de la aplicación utilizada en el sistema autónomo de monitoreo y control de condiciones ambientales en cultivos verticales hortícolas urbanos mediante una red de sensores. La programación de los flujos se analiza mediante la red de cajas negras o nodos y en tiempos de ejecución basados en Node.js.

- **Software para el diseño de la aplicación**

Ionic es un kit de desarrollo de software que se instala son necesarios para desarrollar las aplicaciones de Android, se selecciona por ser de código abierto y facilitará el desarrollo de aplicaciones móviles para lograr el objetivo propuesto en el sistema hortícola, se han diseñado desde el origen de Ionic diferentes versiones, sin embargo, se ha seleccionado específicamente porque permite creación de aplicaciones de escritorio y en la web que a la vez se permite la instalación de estas en otros dispositivos.

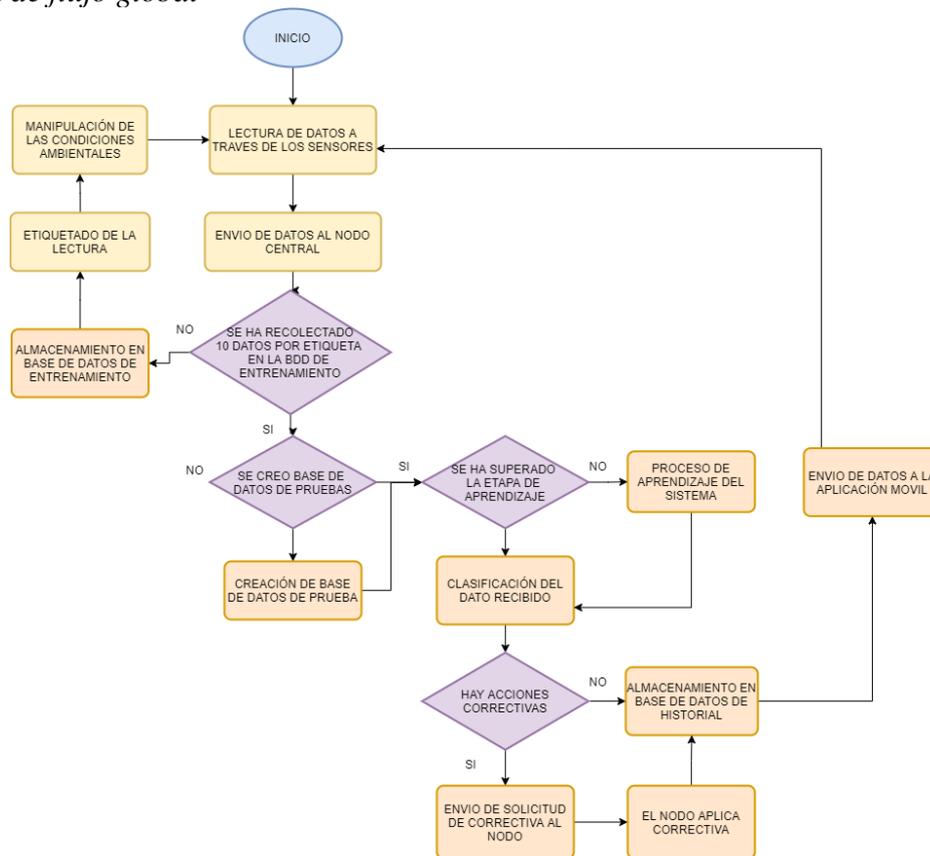
### **3.4.4. Diagramas de flujo**

En el presente apartado se muestran los diagramas de flujo que rigen al sistema, tanto de manera global, como por cada uno de los componentes de este.

#### ***3.4.4.1. Diagrama de flujo global***

El diagrama de flujo global del sistema describe el comportamiento global del sistema en las diferentes etapas de implementación del sistema, como se puede observar en la Figura 36, se condiciona los comportamientos, sobre todo del nodo central, dependiendo si se encuentra en la etapa de recolección de datos, entrenamiento o producción.

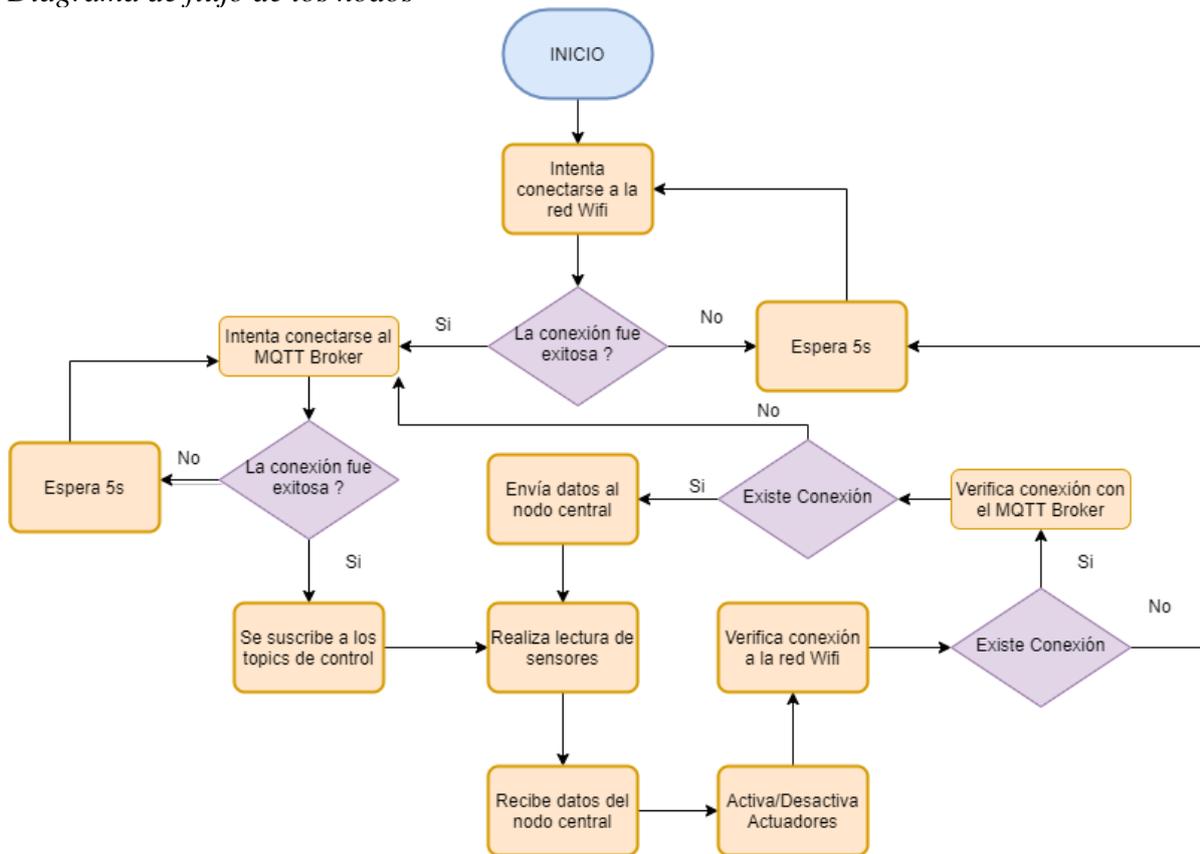
**Figura 36**  
*Diagrama de flujo global*



#### 3.4.4.2. Diagrama de flujo de los nodos

En la Figura 37 se describe el flujo de cada uno de los procesos que debe ejecutar cada uno de los nodos con el objetivo de cumplir con las tareas que le corresponde, validando las conexiones correspondientes, asegurando así el canal de comunicación para que estos puedan llegar al nodo central.

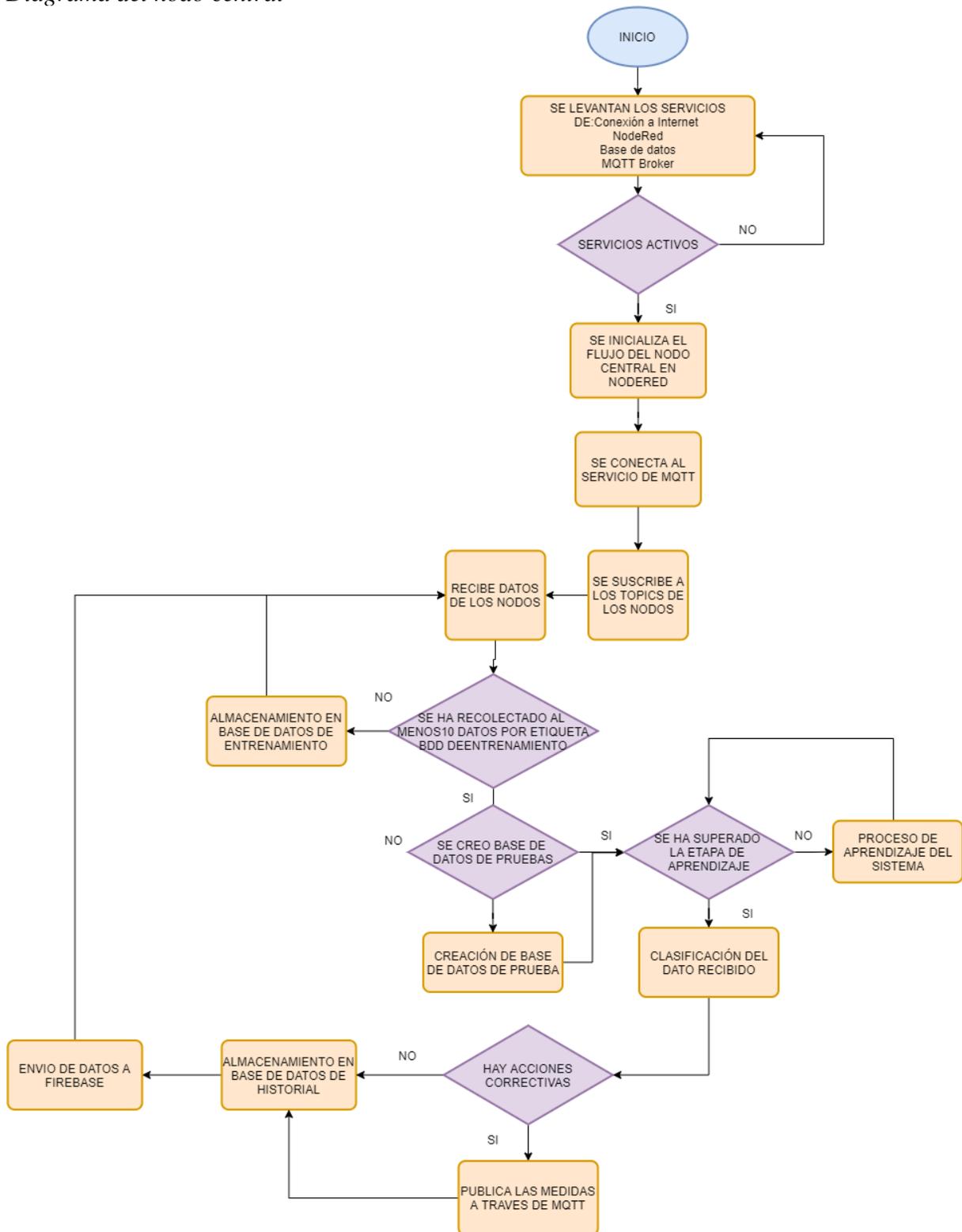
**Figura 37**  
Diagrama de flujo de los nodos



#### 3.4.4.3. Diagrama del nodo central

Es importante recalcar que el nodo central es quien se encarga de tomar las decisiones de aplicar medidas correctivas, a través del algoritmo de aprendizaje aplicado, además de ser quien se encarga del almacenamiento de la información en la base de datos y publica esta información, para que sea posible el monitoreo a través de la aplicación móvil, esto se puede apreciar en la Figura 38.

**Figura 38**  
*Diagrama del nodo central*

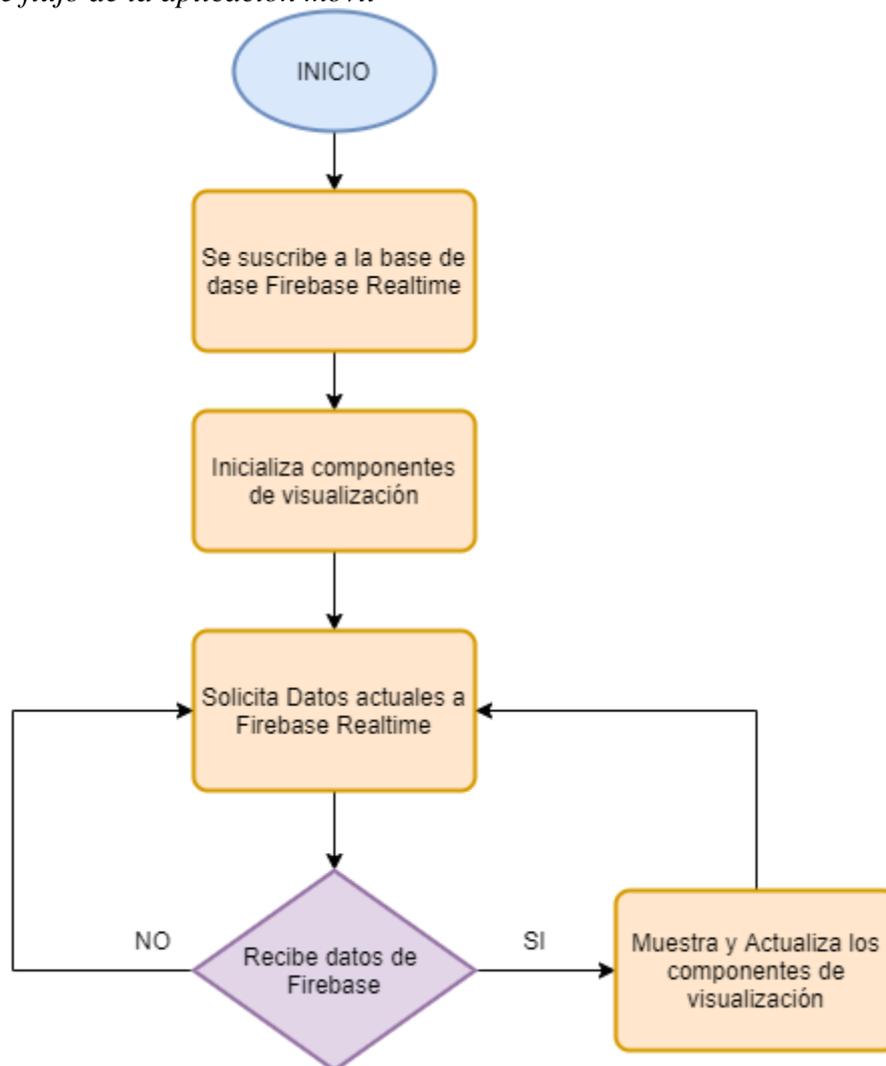


#### 3.4.4.4. Diagrama de la aplicación móvil

Cómo se mencionó en el apartado de la selección de software, para que el monitoreo sea posible, se hará uso de una base de datos en tiempo real, denominada Firebase Realtime, por lo que la tarea de la aplicación será estar a la escucha de las actualizaciones de dicha base de datos y mostrar esta información en una interfaz gráfica, esto se describe en la Figura 39.

**Figura 39**

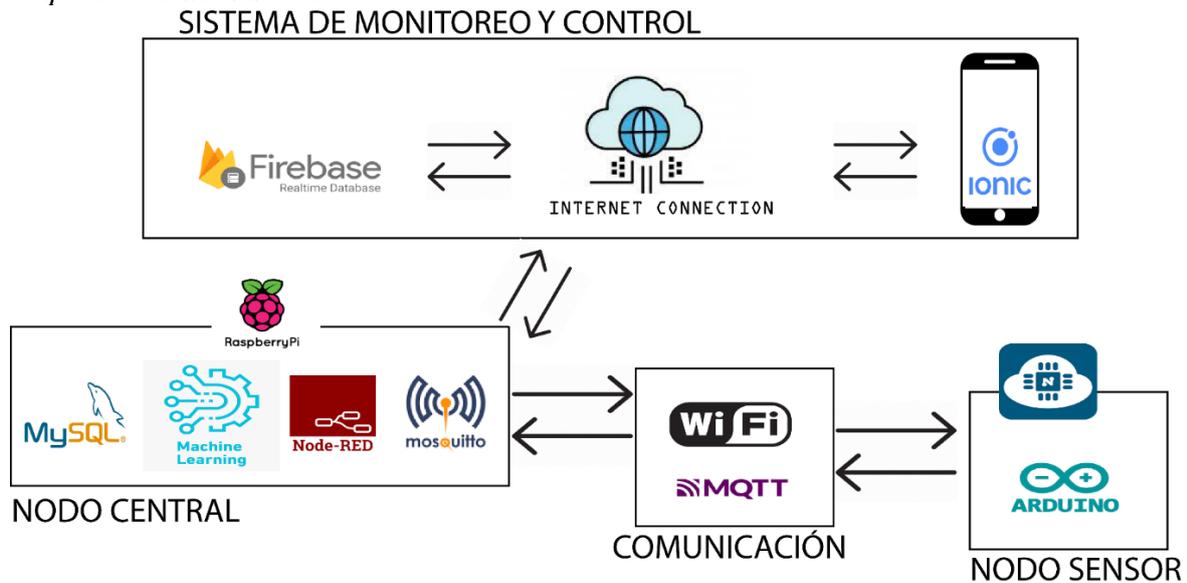
*Diagrama de flujo de la aplicación móvil*



### 3.4.5. Arquitectura del sistema

En la Figura 40, se muestra una representación de la arquitectura del sistema, con sus diferentes componentes:

- **Nodo Central.** - Está conformado por una placa RaspBerry Pi III, en cuyo sistema operativo se encuentran alojados los servicios de: Base de Datos a través de MySQL, Broker Mqtt a través de Mosquitto, El control de Flujos del sistema a través de Node-RED y el algoritmo de Machine Learning encargado de tomar las decisiones para la automatización. El conjunto de estas herramientas permite el correcto funcionamiento del Nodo Central quien a su vez debe establecer una comunicación directa con los otros componentes.
- **Nodo Sensor.** – Está conformado por una placa NodeMCU ESP8266 en donde el IDE de Arduino es la herramienta fundamental para poder programar las acciones que deba ejecutar este componente, tanto para la lectura de datos de los sensores como para el control de los actuadores del sistema.
- **Sistema de monitoreo y control.** – Este bloque cuenta de varios subsistemas: Comunicación en tiempo real a través de Firebase Realtime , un SmartPhone cuya función es alojar la aplicación móvil construida con el Framework Ionic para mostrar los datos en tiempo real y por último la conexión a Internet, que es a través de quien se intercambia la información entre Firebase Realtime y la aplicación Ionic.
- **Comunicación.** – El bloque de comunicación embarcan los protocolos a través de los cuales se establece la comunicación entre el Nodo Central y los Nodos Sensores, para el intercambio de información, de los datos ambientales o del control manual o automático de los actuadores del sistema.

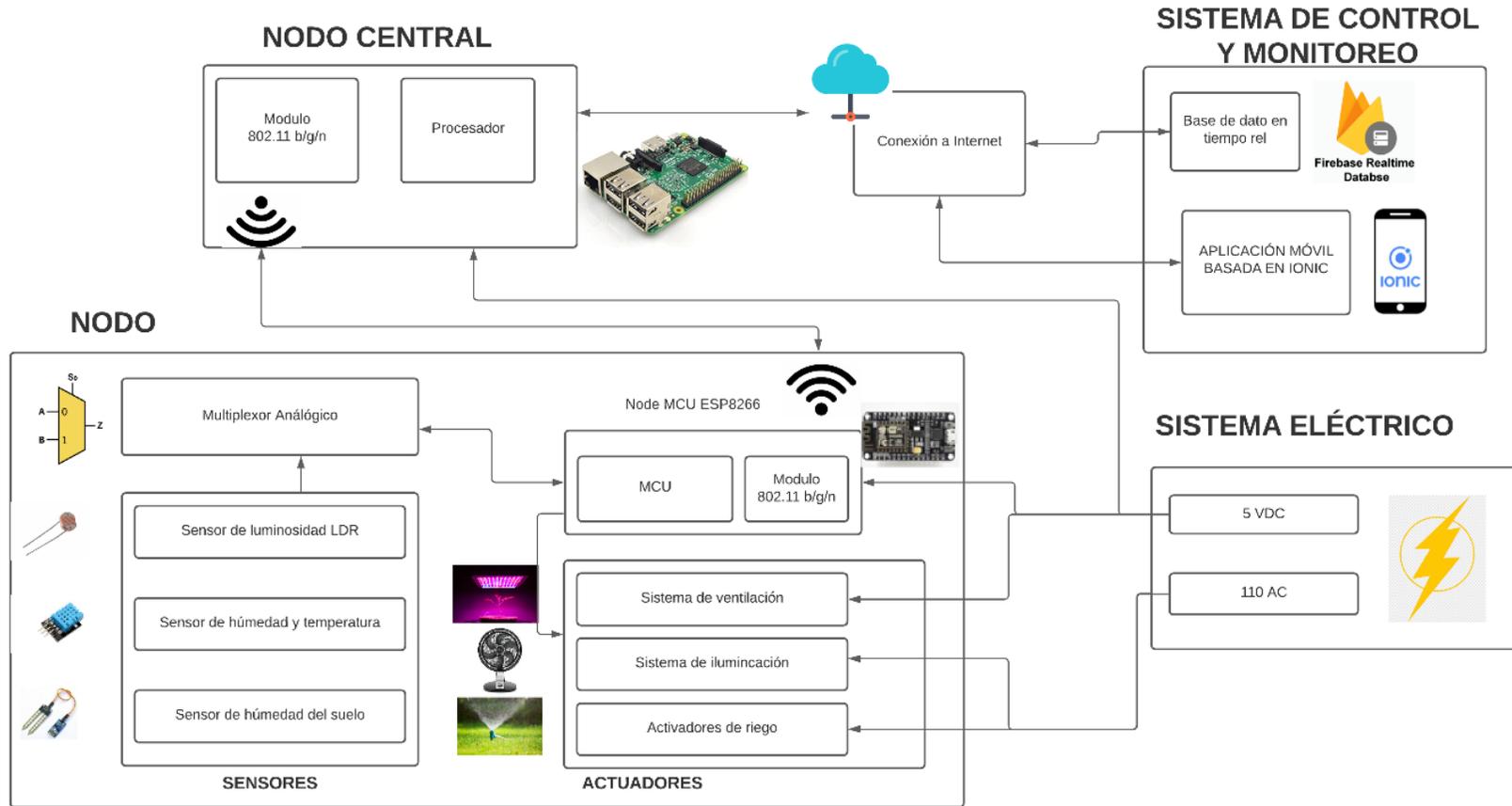
**Figura 40***Arquitectura del sistema*

### 3.4.6. Diagrama de bloques del sistema

Para determinar el diseño se inicia con 4 sensores los cuales controlan humedad de suelo, humedad relativa, luminosidad y temperatura, por lo que se propone desarrollar un sistema de cultivo con el ambiente controlado de las variables antes mencionadas, de este modo el dispositivo implementado fue de apoyo para que el investigador tome decisiones en el tiempo correcto sobre las medidas necesarias que mantengan o mejoren el cultivo hortícola.

El sistema embebido propuesto Figura 41, hará uso de diferentes tecnologías de las que se hablará más a detalle en las siguientes secciones, con el único propósito de obtener recetas de cultivo en un ambiente cerrado, lo que significa que el sistema tendrá una etapa de acondicionamiento o entrenamiento, en la cual mediante la toma de datos aprenderá la manera adecuada de controlar de manera eficiente las variables ambientales y lumínicas de un entorno cerrado, ubicado en un lugar determinado, con el fin de ofrecer el mejor ambiente posible a un determinado producto hortícola de ciclo de crecimiento corto.

**Figura 41**  
*Diagrama de bloques general del sistema*



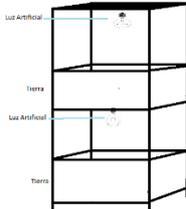
Así, la estructura arquitectónica del sistema presenta el sistema compuesto por nodos que controlarán las variables propuestas, que con un procesamiento de datos a través de redes de sensores que permiten monitorear el ambiente del cultivo a tiempo real como se muestra en la Figura 42, donde existe un sistema central conectado a sensores que enviaron los datos de la plantación del cultivo a un sistema de procesamiento que mediante el internet envió los datos a la nube para ser almacenados.

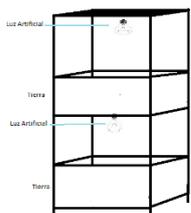
### 3.4.7. Construcción de la estructura de agricultura vertical

Es válido mencionar aspectos importantes que se realizó para el desarrollo del cultivo, en el cual se diseña la estructura con las características de la Tabla 19, para obtener una estructura como se presenta en la Figura 42 con una profundidad de cada nivel con 30 centímetros para el óptimo desarrollo de las plantas.

**Tabla 19**

*Ficha técnica de la estructura del invernadero*

<b>Ficha Técnica de la estructura de invernadero</b>		
<b>Nombre del invernadero</b>	Invernadero vertical	<b>Imagen</b>
<b>Descripción:</b>	Los invernaderos en altura o verticales son esencialmente prácticos para ser utilizados en zonas urbanas, para aprovechar el espacio de las terrazas, a la vez que se monitorea el aspecto le luz, humedad y temperatura.	
<b>Material para usar:</b>	Metal Tríplex Plástico	
<b>Ventajas</b>	Estructura práctica para colocar en espacios de poca dimensión. Se puede aprovechar la altura de las estructuras Facilidad para ser movilizados. En el mercado se encuentran de diferentes materiales y medidas.	
<b>Desventajas</b>	Limitada cantidad de plantas para cultivar Solamente se aplica con cultivos de tamaño mediano o pequeño. La madera tiende a corroerse debido a la humedad.	
<b>Instalación</b>		
<b>Detalles</b>	La base que posee compara con su altura es muy pequeña, por lo que es recomendable repartir el peso del cultivo de manera adecuada. Al sembrar el cultivo es recomendable colocar el mayor peso en la parte baja para que en el segundo nivel sea más ligero, la estructura siempre debe estar situada junto a la pared para no tener inconvenientes. Las medidas de la estructura están diseñadas de acorde a las necesidades del cultivo seleccionado, en este caso, las dimensiones de la caja que contendrá las plantas son: 30cm de alto, 100cm de	

	ancho y 50 cm de profundidad, con lo que, según la ficha técnica del cultivo, tendríamos espacio para abordar 10 plantas distribuidas en dos filas en su etapa madura. El espacio libre entre las cajas contenedoras es de 50 cm, espacio suficiente para colocar las lámparas de crecimiento, para que estas iluminen toda el área de cultivo.
<b>Construcción</b>	
<b>Material de construcción:</b>	Se selecciona el trípex por su facilidad de trabajo en la estructura y el metal por la fortaleza para mantener la estructura en pie con el cultivo
<b>Material de cubierta:</b>	El material para cubrir el cultivo es plástico con lo cual se pretende mantener la temperatura y hermetización de las plantas.
<b>Medidas y diseño propuesto</b>	
	
En cuanto al tamaño la base tendrá una altura de 10 centímetros, las camas poseen una profundidad de 30 centímetros, mientras los espacios entre cada nivel mantienen 40 centímetros de separación.	

Fuente: Hernández y Sarmiento (2017, p. 182)

## Figura 42

### *Diseño de invernadero*



Una vez construida la estructura, se procede con la preparación del suelo y con la finalidad de cumplir con los requerimientos de Stackholders, se debe proporcionar a los cultivos con un sustrato rico en nutrientes, que posea al menos una conductividad eléctrica de

1200ppm, en la Figura 43 se muestra la medición de las propiedades del sustrato adquirido con la asesoría del experto en agricultura.

**Figura 43**

*Medición de la conductividad eléctrica del cultivo*



Como conclusión de esta medición se obtiene que el sustrato no cuenta con los niveles adecuados de nutrientes para este cultivo, por lo que se procede a abonar la tierra, con suficiente cantidad de abono orgánico hasta obtener un sustrato con las propiedades indicadas para el desarrollo de las plantas.

### **3.4.8. Cálculo de la fuente de alimentación**

Tomando en cuenta el hardware seleccionado en el apartado 3.4.2, para la construcción del Nodo Sensor, es fundamental tener en cuenta la demanda de corriente que esto tendrá una vez se haya diseñado, por tal motivo en este apartado se ha tomado los datos necesarios de la hoja del fabricante de cada elemento para realizar el cálculo de dicha demanda.

En la Tabla 20 se muestra una lista de los elementos que demandan un corriente de la misma fuente de alimentación que el Nodo Sensor, la cantidad de estos elementos por cada nodo, el valor de corriente demandado y en la última columna el total de corriente demandada por la cantidad de elementos que se conectarían a un Nodo Sensor.

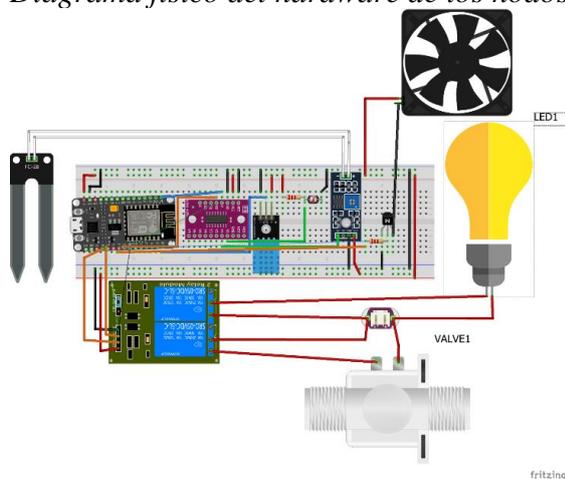
**Tabla 20***Listado de elementos con alimentación de 5V*

Item	Cantidad	Corriente(mA)	Corriente Total (mA)
Ventiladores	2	200	400
Modulo Rele	1	90	90
Node Mcu	1	80	80
Sensor Ldr	1	2	2
Sensor FC28	2	35	70
Sensor DHT11	1	0.5	0.5
<b>Total</b>			<b>642.5</b>

La sumatoria de corrientes mostrada al final de la Tabla 20, sugiere que la fuente de alimentación debe tener una capacidad mayor a 642.5 miliamperios, por lo que basados en este dato se elige una fuente de alimentación para cada Nodo Sensor, con un voltaje y corriente de salida de 5 voltios y 1 amperio, respectivamente.

### 3.4.9. Prototipo electrónico

En la Figura 44, se muestra el diagrama de conexión físico de los diferentes elementos de hardware que están involucrados en el bloque de los nodos, los cuáles han sido seleccionados realizando un Benchmark en base los requerimientos del sistema especificados anteriormente.

**Figura 44***Diagrama físico del hardware de los nodos*

- **Multiplexor Analógico**

En un primer bloque se tiene la conexión del multiplexor analógico hacia la alimentación del sistema, y los pines de control a pines digitales del NodeMCU, a su vez su salida se encuentra conectada a la única entrada analógica que el nodo posee.

- **Sensor de luminosidad LDR**

Este sensor se encuentra conectado en serie con una resistencia de 2.2KOhms, en donde la caída de voltaje de este hacia una de las entradas del multiplexor analógico.

- **Sensor de humedad del suelo**

De la misma manera que el anterior la salida de datos de este, está conectada hacia una de las entradas del multiplexor analógico.

- **Sensor de Temperatura y humedad relativa**

Este sensor por el hecho de ser digital necesita una mínima calibración y se encuentra conectado a una de las entradas digitales del Nodo.

- **Lámparas de iluminación**

Estas lámparas debido a su alimentación serán activadas a través de un relé de 5VDC – 110AC, el cual estará conectado a una de las salidas digitales del Nodo.

- **Electroválvula**

De la misma manera que las lámparas, debido a su alimentación será activada a través de un relé de 5VDC – 110AC y una salida digital.

- **Ventiladores**

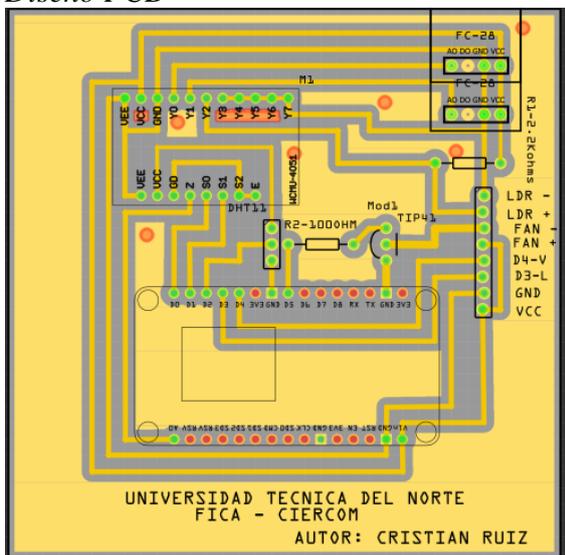
Estos debido a su demanda de corriente serán activados a través de una salida digital del Nodo, cuya señal atravesará un transistor configurado en saturación para proporcionar la corriente necesaria a este elemento.

### 3.4.10. Diseño del circuito impreso (PCB)

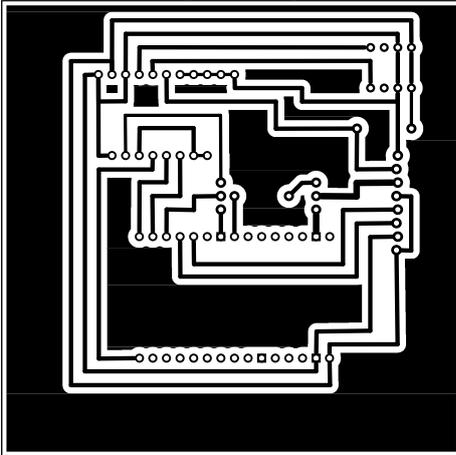
En base a lo indicado en el apartado anterior, se procede con la creación de un circuito impreso PCB, de tal manera de contar con un diseño lo más compacto posible e integrarlo en el sistema de agricultura vertical, sin que este sea afectado por las variables ambientales y tampoco afecte el espacio de las plantas.

Con la ayuda de la herramienta de software para desarrollo de hardware Fritzing, también se desarrolló el diseño del circuito impreso del sistema, como se puede observar en la Figura 45.

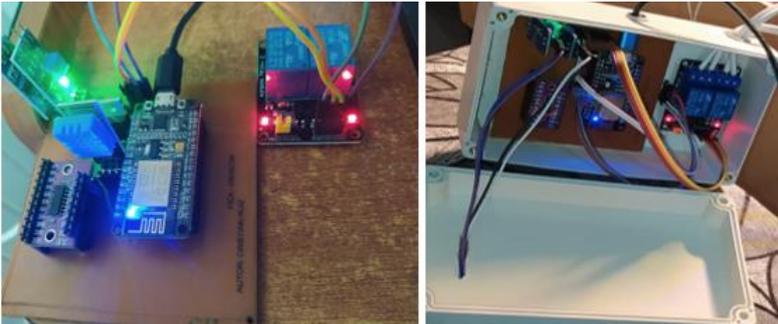
**Figura 45**  
*Diseño PCB*



A continuación, en la Figura 46, se muestra el resultado a imprimir para su posterior grabado en baquelita y por último, se culmina con el soldado de los componentes.

**Figura 46***SVG utilizada para el grabado de la baquelita*

Las medidas de la PCB son de 86mmX86mm, sin embargo, es importante recalcar que el módulo relé utilizado para activar los actuadores, tiene conexión externa, de la misma manera que el resto de los sensores, las medidas del módulo relé son de 38mmX48mm, de esta manera se ha considerado adquirir una caja plástica de medidas 150mmX90mm, con la finalidad aislar los componentes electrónicos de la humedad del ambiente y de posibles salpicaduras de agua debido al riego automático, esto se puede apreciar en la Figura 47.

**Figura 47***Resultado del diseño de hardware*

### 3.4.11. Calibración de los sensores

Una vez que se ha obtenido el circuito impreso con los componentes del sistema de hardware, es importante considerar una “calibración de los sensores”, para el presente proyecto, como ya se ha mencionado, se utilizarán 3 sensores específicos, de los cuales se explica a continuación.

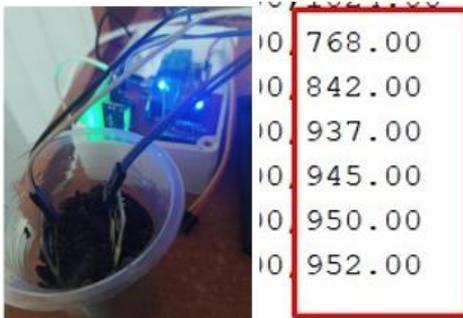
- **Sensor de humedad del suelo FC-28**

Para la calibración del FC-28 Se ha realizado tres diferentes mediciones: suelo relativamente seco, húmedo y con exceso de agua.

En la Figura 48, se puede apreciar el primer caso, suelo húmedo, donde se puede apreciar que los sensores arrojan valores mayores a 700, para el caso específico de este sensor, no se utilizará unidades de medida, sino simplemente el valor de la entrada analógica.

**Figura 48**

*Medición suelo bajo en humedad*



Así mismo, en la Figura 49 y 50, se puede apreciar que los valores de medición de los sensores para un suelo húmedo y con exceso de agua, oscilan entre 400 – 700 y menores a 400, respectivamente.

**Figura 49**

*Medición suelo con suficiente humedad*



**Figura 50***Medición suelo con exceso de humedad*

De esta manera se toma en consideración dichos valores para etiquetar los datos recolectados, en la etapa correspondiente.

- **Sensor de humedad y temperatura DHT11**

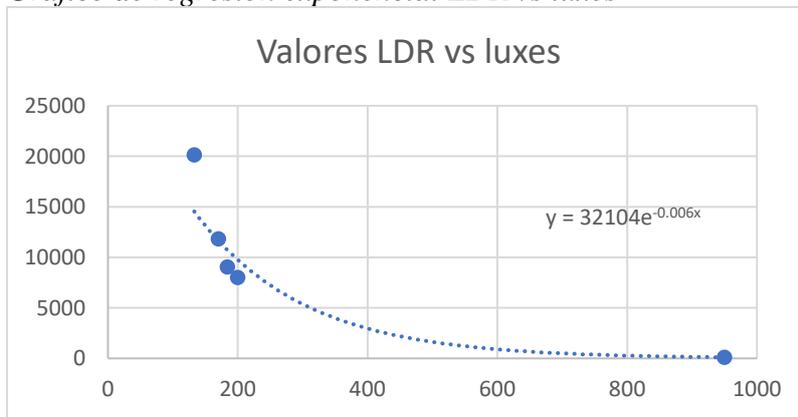
El sensor DHT11 trabaja de enviando sus datos de forma digital, porque si existiera que realizar una calibración deberá que hacerse de la misma manera, en este caso se ha comprobado el correcto funcionamiento de los mismos en base a otro sensor de temperatura ambiental y se ha comprobado el correcto funcionamiento del mismo, como se observa en la Figura 51.

- **Sensor de luminosidad LDR**

El sensor de luminosidad LDR trabaja de manera analógica, y como unidad de medida para la intensidad luminosa según el sistema internacional de unidades, es el lux, por lo que con la ayuda de una aplicación en Android denominada “Light Meter” y los valores analógicos obtenidos del sensor, se ha realizado una regresión exponencial con el fin de obtener una ecuación que aproxime con mayor precisión la transformación de una entrada analógica en luxes, en la Tabla 21, se puede apreciar los datos considerados para dicha regresión y en la Figura 51 el gráfico de dichos valores y con línea punteada la curva de la regresión exponencial calculada por Excel.

**Tabla 21***Valores del sensor LDR vs medición en luxes*

<b>Entrada Analógica</b>	<b>Medición en Luxes</b>
<b>200</b>	8011
<b>184</b>	9062
<b>170</b>	11830
<b>133</b>	20142
<b>950</b>	115

**Figura 51***Gráfico de regresión exponencial LDR vs luxes*

De esta manera se ha podido obtener la ecuación 1, la cual es una expresión matemática que representa la curva de la regresión exponencial ya mencionada, donde “x” es el valor de la entrada analógica de 0 – 1024 e “y” el valor de dicha entrada transformado en luxes.

$$y = 32104e^{-0.006x} \quad ( 5 )$$

### 3.4.12. Configuración del Nodo central

El proyecto necesitará de un nodo central que se encargue de procesar la información enviada por los nodos, en este caso dicho nodo será una placa Raspberry PI 3, el cual se lo puede apreciar en la Figura 52 dentro de su case de protección.

**Figura 52**  
*Nodo Central*



## Instalación del Sistema Operativo en la placa Raspberry

El nodo central requiere de un sistema operativo y un dispositivo de almacenamiento para albergar dicho sistema como cualquier otro servidor. Dada la popularidad de estas placas, en la actualidad cuentan con una herramienta llamada RaspBerry Pi Imager, la cual no es más que un programa que facilita la instalación de un sistema operativo compatible con la placa dentro de un dispositivo de almacenamiento, en este caso una tarjeta SD con capacidad de 32GB y todo esto en unos pocos pasos, se debe:

- Descargar la herramienta del sitio oficial
- Seleccionar el SO a instalar (En este caso Raspbian de 32 bits)
- Seleccionar el dispositivo de almacenamiento
- Clic en el botón Escribir o Write

En la Figura 53 se puede apreciar la interfaz de la herramienta en la cual se pueden ejecutar los pasos mencionados con facilidad.

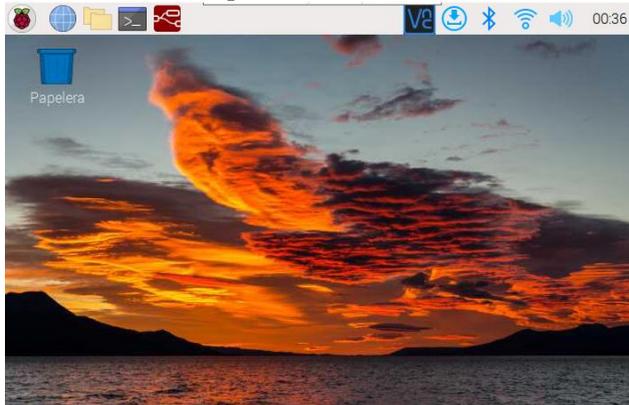
**Figura 53**  
*Instalación de Raspbian*



Una vez instalado el Sistema Operativo, basta con montar el dispositivo de almacenamiento en la ranura SD de la placa y energizarla, y conectándole una pantalla a

través de HDMI se podrá visualizar el escritorio del sistema operativo instalado, como se muestra en la Figura 54.

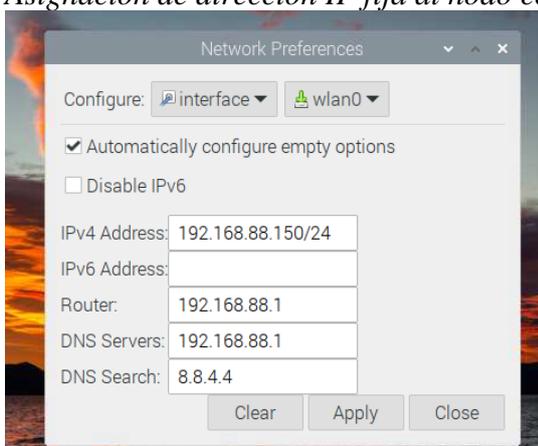
**Figura 54**  
*Escritorio de Raspbian*



### Configuración de IP fija

En la Figura 55 se puede observar la asignación de una dirección IP fija a la interfaz inalámbrica, esto servirá para que los nodos que se conecten a la red sepan a qué dirección enviar los datos a través de la dirección IP asignada y el protocolo MQTT. En este caso se le ha asignado la dirección IP 192.168.88.150/24.

**Figura 55**  
*Asignación de dirección IP fija al nodo central*



### Instalación de Broker MQTT

Para empezar la instalación de cualquier programa en distribuciones Linux, siempre es recomendable, actualizar los repositorios del sistema operativo, para que pueda descargar las

versiones más actuales y estables. Esto se lo puede lograr de diferentes maneras en este caso, se ha utilizado el comando “sudo apt-get update”.

Una vez que ha culminado con el proceso de actualización, para instalar el Broker MQTT, simplemente se ha utilizado el comando “sudo apt-get install mosquitto mosquitto-clients”, esto se puede apreciar en la Figura 55 y una vez instalado el software, se ha verificado que el servicio se haya levantado correctamente con el comando “sudo service mosquitto status”, la Figura 56 muestra que el servicio se encuentra corriendo.

**Figura 56**

*Instalación Broker MQTT*

```
pi@raspberrypi:~ $
pi@raspberrypi:~ $ sudo apt-get install mosquitto mosquitto-clients
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
```

**Figura 57**

*Verificación del servicio Mosquitto*

```
pi@raspberrypi:~ $
pi@raspberrypi:~ $ sudo service mosquitto status
mosquitto.service - Mosquitto MQTT Broker
Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor pre>
Active: active (running) since Fri 2021-12-10 00:22:39 -05; 26min ago
Docs: man:mosquitto.conf(5)
      man:mosquitto(8)
Process: 511 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=exi>
Process: 520 ExecStartPre=/bin/chown mosquitto /var/log/mosquitto (code=exi>
Process: 524 ExecStartPre=/bin/mkdir -m 740 -p /run/mosquitto (code=exited,>
Process: 527 ExecStartPre=/bin/chown mosquitto /run/mosquitto (code=exited,>
Main PID: 529 (mosquitto)
Tasks: 1 (limit: 1597)
CPU: 2.048s
CGroup: /system.slice/mosquitto.service
        └─529 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf
```

Por último, es necesario editar el archivo de configuración de mosquitto ubicado en “/etc/mosquitto/mosquitto.conf”, para permitir la conexión anónima de los nodos conectados a la misma red y de manera anónima como se muestra en la Figura 58.

**Figura 58***Configuración de mosquitto*

```

mosquitto.conf /etc/mosquitto/mosquitto.conf
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

listener 1883 0.0.0.0
allow_anonymous true
pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d

```

## Instalación de Mysql

Para la instalación del servidor de base de datos Mysql, simplemente es necesario ejecutar el comando “sudo apt-get install mariadb-server”, así como se muestra en la Figura 59. Una vez instalado el servidor verificamos con “sudo service mariadb status”, la Figura 60 nos muestra que el servidor ha sido instalado y se encuentra corriendo sin problemas.

**Figura 59***Instalación del servidor de base de datos*

```

pi@raspberrypi:~$ sudo apt-get install mariadb-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho

```

**Figura 60***Comprobación del servidor de base de datos*

```

pi@raspberrypi:~$ sudo service mariadb status
● mariadb.service - MariaDB 10.5.12 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese
   Active: active (running) since Fri 2021-12-10 00:22:52 -05; 35min ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 510 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/
   Process: 519 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_STA
   Process: 530 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && V
   Process: 944 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_ST
   Process: 946 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/S
   Main PID: 601 (mariabd)
   Status: "Taking your SQL requests now..."

```

Para culminar con la configuración de mysql, simplemente ejecutamos el comando “mysql\_secure\_installation”, como se muestra en la Figura 61, esto permitirá cambiar la contraseña root de la base de datos, simplemente se debe seguir los pasos del programa de consola.

**Figura 61***Cambio de contraseña root de mysql*

```

pi@raspberrypi:~ $ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] █

```

**Instalación de Node-RED**

Dada la popularidad que ha tomado la instalación de Node-RED, como software de control en placas RaspBerry, se puede encontrar en el sitio oficial, un comando, que lo que hace es descargar y ejecutar un archivo bash, que contiene todas las instrucciones para instalar automáticamente todas las dependencias de Node-RED para después de esto simplemente utilizarlo, Figura 62.

**Figura 62***Instalación de Node-RED*

```

pi@raspberrypi:~ $ bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)

This script checks the version of node.js installed is 12 or greater. It will tr
y to
install node 14 if none is found. It can optionally install node 12, 14 or 16 LT
S for you.

```

Una vez que ha finalizado el proceso, se ha ejecutado el comando “sudo systemctl enable nodered.service”, con la finalidad de levantar Node-RED automáticamente al encender la placa, como cualquier otro servicio y el comando sudo systemctl status nodered.service” para verificar el servicio (Figura 63).

**Figura 63***Verificación del servicio de Node-RED*

```

pi@raspberrypi:~ $
pi@raspberrypi:~ $ sudo systemctl enable nodered.service
pi@raspberrypi:~ $ sudo systemctl status nodered.service
● nodered.service - Node-RED graphical event wiring tool
   Loaded: loaded (/lib/systemd/system/nodered.service; enabled;
   Active: active (running) since Fri 2021-12-10 00:22:36 -05; 46
     Docs: http://nodered.org/docs/hardware/raspberrypi.html
   Main PID: 401 (node-red)
     Tasks: 63 (limit: 1597)
    CPU: 3min 25.082s
   CGroup: /system.slice/nodered.service
             └─ 401 node-red

```

### 3.4.13. Diseño de base de datos local

Para el diseño de la base de datos se ha tomado en cuenta las diferentes etapas que se llevará a cabo en el desarrollo del proyecto, quiere decir que se crearán tablas con propósitos específicos en cada etapa.

En primer lugar, se ha considerado crear una Tabla estática llamada “niveles”, la cual hace referencia a cada nivel de la estructura vertical, de esta manera se separarán los datos que provengan del nodo ubicado en el nivel 1 y en el nivel 2.

Luego se ha diseñado la Tabla en donde se recolectarán los datos utilizados en la etapa de entrenamiento del sistema llamada “entrenamiento” y cada una de sus columnas que son:

- Id (Primary Key)
- Luz (Valor en luxes del sensor LDR)
- Horas\_luz (Tiempo de horas luz que ha recibido el cultivo hasta ese momento)
- Temperatura (Valor de temperatura en grados centígrados)
- Humedad (Porcentaje de humedad relativa)
- Humedad\_suelo (Valor promedio de los sensores de humedad del suelo)
- Fecha (Fecha y hora de inserción del dato)
- Nivel (Nivel al que pertenecen los datos)

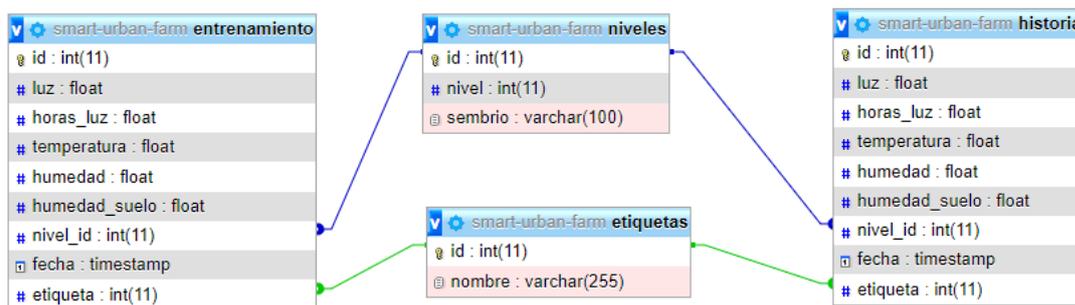
- Etiqueta (Clase a la que pertenece el conjunto de datos)

Para la etapa de producción del sistema se ha considerado clonar la Tabla de entrenamiento, para así, no mezclar los datos que se etiquetarán de manera supervisada, con los datos que etiquetará algoritmo de aprendizaje, a esta Tabla se le ha llamado “historial” y a la vez una Tabla que guardará la descripción de cada etiqueta y su identificador numérico llamado “etiquetas”.

En la Figura 64, se puede apreciar el diagrama entidad relación generado a través del cliente de bases de datos MySQL, PhpMyAdmin

**Figura 64**

*Modelo entidad relación de la base de datos MySQL*



Nota: El archivo SQL de generación de la base de datos se puede observar en el Anexo

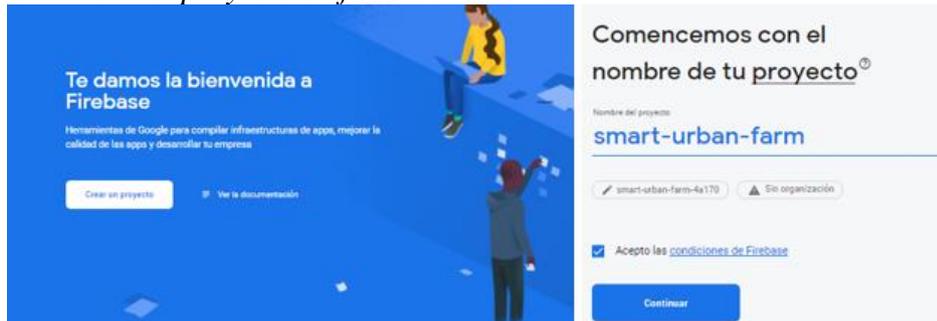
### 3.4.14. Diseño de la base de datos en tiempo real

La base de datos que se utilizará para el monitoreo y control del sistema es Firebase Realtime, la cual es una base de datos NoSQL que se maneja a través del formato JSON y nos permite una comunicación en tiempo real de manera rápida y sencilla, para tener acceso a esta plataforma es necesario contar con una cuenta de Google y dirigirse a la consola de Firebase disponible en el link:

<https://console.firebase.google.com/u/1/>, desde aquí se debe crear un nuevo proyecto, lo cual se realiza con tan solo un clic en el botón “Crear Proyecto”, se proporciona un nombre para el mismo (en este caso se lo ha llamado “smart-urban-farm”) y se acepta los términos y condiciones de la plataforma, Figura 65.

**Figura 65**

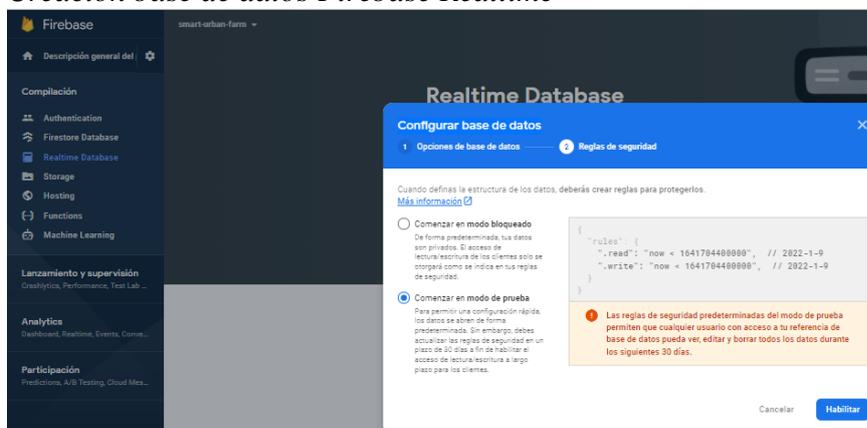
*Creación del proyecto en firebase*



Luego de dar clic en continuar y aceptar los términos y condiciones de Google Analytics, se empezará a crear el proyecto y en el panel de administración del proyecto se selecciona Firebase Realtime, en donde daremos clic en crear base de datos, y en la ventana modal, seleccionaremos empezar en modo de prueba, con la finalidad de no solicitar autenticación para la lectura y escritura de datos, mientras se desarrolla el proyecto, Figura 66.

**Figura 66**

*Creación base de datos Firebase Realtime*



Una vez creado el ambiente de la base de datos, podemos importar el modelo de nuestra base de datos, para esto se ha creado un archivo JSON, Figura 67, con dicho modelo, el cual cuenta con dos nodos principales: monitoreo y control. Dentro de cada uno de estos nodos, se encuentran las variables ambientales, junto con el estado (etiqueta) y los actuadores del sistema, respectivamente, y dentro de cada uno de estos el nivel al que corresponde, es decir, nivel 1 o nivel 2.

**Figura 67**

*Modelo de base de datos Firebase Realtime en formato JSON*

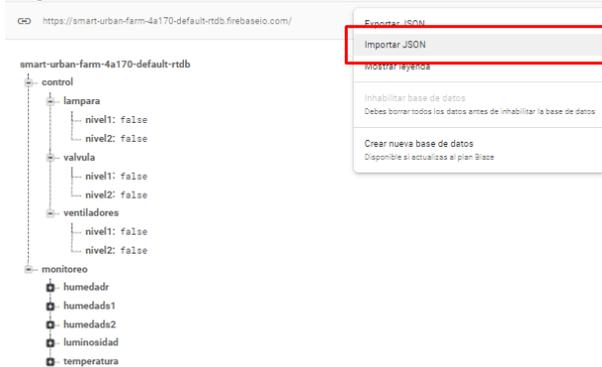
```

1  {
2    "control" : {
3      "lampara" : {
4        "nivel1" : false,
5        "nivel2" : false
6      },
7      "valvula" : {
8        "nivel1" : false,
9        "nivel2" : false
10     },
11     "ventiladores" : {
12       "nivel1" : false,
13       "nivel2" : false
14     }
15   },
16   "monitoreo" : {
17     "estado" : {
18       "nivel1" : "",
19       "nivel2" : ""
20     },
21     "horas_luz" : {
22       "nivel1" : 0,
23       "nivel2" : 0
24     },
25     "humedadr" : {
26       "nivel1" : 0,
27       "nivel2" : 0
28     },
29     "humedads" : {
30       "nivel1" : 0,
31       "nivel2" : 0
32     },
33     "luminosidad" : {
34       "nivel1" : 0,
35       "nivel2" : 0
36     },
37     "temperatura" : {
38       "nivel1" : 0,
39       "nivel2" : 0
40     }
41   }
42 }
43

```

Con este archivo guardado en el ordenador, se procede a hacer una importación dentro de la consola, y quedando así lista la base de datos de monitoreo y control en tiempo real, así como se aprecia en la Figura 68.

**Figura 68**  
**Importación de archivo JSON en Firebase Realtime**



Nota: El código completo del archivo JSON se encuentra en el Anexo

### 3.4.15. Programación de los Nodos

Para la programación de los nodos, se ha hecho uso del IDE de Arduino, pero dada la situación de que las placas ESP, no son versiones oficiales de Arduino, es necesario agregar en el menú “Archivo/Preferencias”, opción “Gestor de URLs adicionales de tarjetas”, la siguiente URL, para el soporte del Node MCU ESP8266:

[https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json), [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)

Una vez hecho esto se procede a instalar la librería para ESP8266, a través del Gestor de tarjetas del IDE, y una vez instalada se puede conectar la placa a través de USB y seleccionar el puerto COM correspondiente a esta.

Para empezar con la programación se han incluido las librerías a utilizar en la cabecera del proyecto (Figura 69), que en este caso son:

- ESP8266WiFi (Conexión Wifi)
- PubSubClient (Publicación y suscripción de mensajes MQTT)
- DHT (Gestión del sensor DHT11)

**Figura 69***Programación de los nodos - Librerías*

```

1 #include <ESP8266WiFi.h>
2 #include <PubSubClient.h>
3 #include "DHT.h"

```

Luego se han definido las variables globales (Figura 70):

- S0 y S1 (Definir pines para la activación de la entrada del multiplexor)
- DHTPin, lamp, valve y fan (Definir pines para el sensor DHT y actuadores)
- Now, lastMeasure (Variables de tiempo auxiliares)
- Ssid, password (Credenciales red Wifi)
- Mqtt\_server (Dirección IP del Nodo Central)
- Dht (Gestion del sensor DHT11)
- Espclient (Gestión de la conexión Wifi)
- Client (Gestión del cliente MQTT)

**Figura 70***Programación de los nodos – Variables globales*

```

4
5 // 4051 S0 - GPIO 16 = D0 on ESP-12E NodeMCU board
6 const int s0 = 16;
7 // 4051 S1 - GPIO 5 = D1 on ESP-12E NodeMCU board
8 const int s1 = 5;
9 // DHT Sensor - GPIO 4 = D2 on ESP-12E NodeMCU board
10 const int DHTPin = 4;
11 // Lamp - GPIO 0 = D3 on ESP-12E NodeMCU board
12 const int lamp = 0;
13 // Valve - GPIO 2 = D4 on ESP-12E NodeMCU board
14 const int valve = 2;
15 // Fan - GPIO 14 = D5 on ESP-12E NodeMCU board
16 const int fan = 14;
17
18 // Timers auxiliar variables
19 long now = millis();
20 long lastMeasure = 0;
21 #define DHTTYPE DHT11 // DHT 11
22 // Credenciales para conectarse a la Red Wifi
23 const char* ssid = "xxxxx";
24 const char* password = "xxxxxxx";
25 // MQTT broker IP Address
26 const char* mqtt_server = "192.168.88.150";
27 DHT dht(DHTPin, DHTTYPE);
28 WiFiClient espClient;
29 PubSubClient client(espClient);

```

La primera función en ejecutarse cuando el Node MCU enciende, es la función

“setup” (Figura 71), en esta se ha realizado lo siguiente:

- Configuración de los pines de entrada y salida.
- Inicialización del sensor DHT11
- Inicialización de la comunicación serial
- Llamado a la configuración para la conexión Wifi
- Configuración del Broker MQTT del nodo
- Configuración de la función a ejecutar cuando el cliente recibe un mensaje a través de un topic MQTT suscrito.

### Figura 71

#### Programación de los nodos – Setup

```

5 // Configuración de callback para los suscritos
6 void setup() {
7   pinMode(s0, OUTPUT);
8   pinMode(s1, OUTPUT);
9   pinMode(lamp, OUTPUT);
0   pinMode(valve, OUTPUT);
1   pinMode(fan, OUTPUT);
2   dht.begin();
3   Serial.begin(115200);
4   setup_wifi();
5   client.setServer(mqtt_server, 1883);
6   client.setCallback(callback);
7 }
8

```

La función “setup\_wifi()” (Figura 72), simplemente es un loop, que termina cuando se ha establecido la conexión a la red Wifi de manera correcta

### Figura 72

#### Programación de los nodos – Conexión Wifi

```

void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("WiFi connected - ESP IP address: ");
  Serial.println(WiFi.localIP());
}

```

La función “callback” (Figura 73), se ejecuta cuando el nodo recibe un topic al que se ha suscrito, en este caso, se suscribirá solo a los topics de control, los cuales darán la orden de activar o desactivar los actuadores, según órdenes del nodo central.

### Figura 73

#### Programación de los nodos - Función a ejecutar al llegar un mensaje MQTT

```

5 void callback(String topic, byte* message, unsigned int length) {
7   Serial.print("Ha llegado un mensaje del topic: ");
3   Serial.print(topic);
3   Serial.print(". Mensaje: ");
3   String messageTemp;
1   for (int i = 0; i < length; i++) {
2     Serial.print((char)message[i]);
3     messageTemp += (char)message[i];
1   }
5   Serial.println();
5   // Enciende o apaga la lámpara, si el topic es levell/lamp
7   if(topic=="levell/lamp"){
3     Serial.print("Cambiando Lampara a: ");
3     if(messageTemp == "on"){
4       digitalWrite(lamp, HIGH);
1     Serial.print("On");
2     }
3     else if(messageTemp == "off"){
4       digitalWrite(lamp, LOW);
5     Serial.print("Off");
5     }
7   }
3   // Enciende o apaga la electroválvula, si el topic es levell/valv

```

Una vez que se ha terminado de ejecutar la función “setup”, el nodo empezará a ejecutar repetida e indefinidamente la función “loop” (Figura 74), en esta función se realizan los siguientes procesos.

- Verificar que existe una conexión con el Broker MQTT y en caso de que no exista, llamar a la función “reconnect”
- Si existe conexión el intérprete escapa de la función “reconnect” y continua con las siguientes líneas de código, en donde se captura el tiempo en milisegundos, y se evalúa que hayan transcurrido más de 15s a partir de la última medición.
- Si la condición anterior se cumple el nodo activará cada una de las entradas del multiplexor analógico para leer los datos de los sensores conectados al mismo (FC-28 y LDR) y luego pasa a leer el sensor DHT11 para computar la

temperatura y humedad relativa del ambiente, una vez leídos todos los sensores, crea la trama de datos y la envía al Broker. Caso contrario la función termina y se vuelve a ejecutar nuevamente desde el primer paso.

### Figura 74

#### Programación de los nodos – Función de bucle del nodo

```

139 void loop() {
140   if (!client.connected())
141     reconnect();
142   if (!client.loop())
143     client.connect("ESP8266123");
144   now = millis();
145   // Publica las lecturas de los sensores cada 15s.
146   if (now - lastMeasure > 1500) {
147     lastMeasure = now;
148     //Se establece s0 y s1 en LOW para obtener la entrada Y0 - Sensor FC28-1
149     digitalWrite(s0, LOW);
150     digitalWrite(s1, LOW);
151     float fc28_1 = (float)analogRead(0);
152     //Espera 250 ms para realizar una siguiente lectura
153     delay(250);
154     //Se establece s0 en HIGH y s1 en LOW para obtener la entrada Y1 - Sensor FC28-2
155     digitalWrite(s0, HIGH);
156     digitalWrite(s1, LOW);
157     float fc28_2 = (float)analogRead(0);

```

La función “reconnect” (Figura 75), evalúa repetidamente cada 5s la conexión del cliente, con el Broker, si determina que existe conexión, suscribe al cliente a los topics de control de los actuadores y el intérprete escapa de esta función.

### Figura 75

#### Programación de los nodos – Conexión y suscripción MQTT

```

99 // Esta función reconecta el ESP8266 a el MQTT broker
100 void reconnect() {
101   // Loop until we're reconnected
102   while (!client.connected()) {
103     Serial.print("Attempting MQTT connection...");
104     // Intentando conectar
105     if (client.connect("ESP8266123")) {
106       Serial.println("connected");
107       // Subscribe o resubscribe a un topic
108       client.subscribe("levell/lamp");
109       client.subscribe("levell/valve");
110       client.subscribe("levell/fan");
111     } else {
112       //Imprime el codigo de error de la conexión
113       Serial.print("failed, rc=");
114       Serial.print(client.state());
115       Serial.println(" try again in 5 seconds");
116       // Espera 5s antes de volver a intentar
117       delay(5000);
118     }
119   }
120 }

```

Nota: El código completo de los nodos, debidamente comentado, se encuentra en el Anexo

### 3.4.16. Recolección de datos de entrenamiento

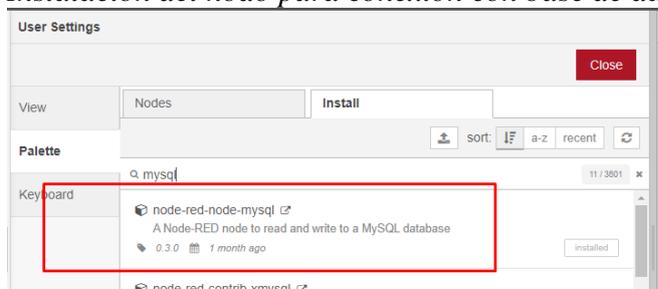
A continuación, se muestra la implementación de software necesaria, para crear un modelo en el cuál, el nodo central sea capaz de almacenar en la base de datos local y la integración del hardware dentro del sistema de agricultura vertical, con la finalidad de conseguir una base de datos con suficientes muestras para su posterior entrenamiento.

#### Creación del flujo de recolección de datos

Para la recolección de datos de entrenamiento, es preciso hacer uso de Node-RED, ya que este se encargará a través de un flujo determinado, recibir los datos enviados por los sensores y persistirlos en la base de datos local. Para esto ha sido necesario la instalación del nodo “node-red-node-mysql” a través del “Manage Palette”, el mismo que permite establecer una conexión con la base de datos de MySQL. En la Figura 76 y 77, se muestra la instalación y configuración de dicho nodo, respectivamente.

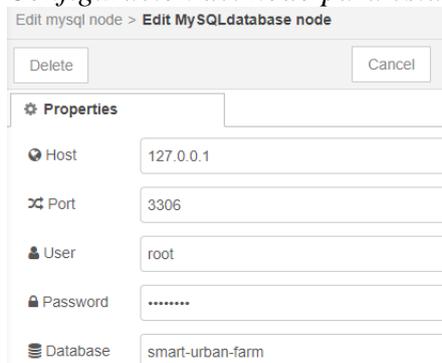
#### Figura 76

*Instalación del nodo para conexión con base de datos MySQL*



#### Figura 77

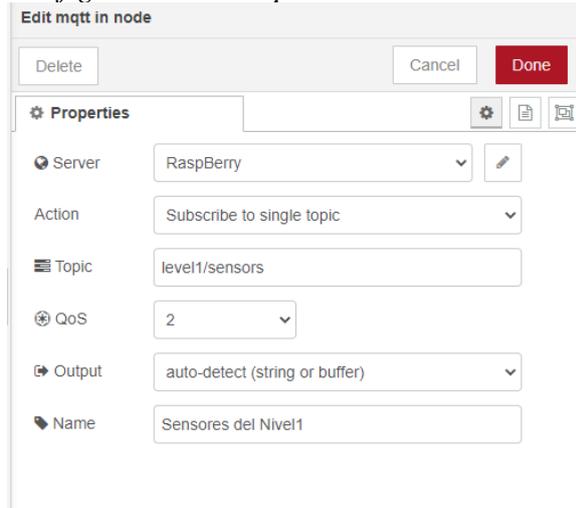
*Configuración del nodo para establecer conexión con MySQL*



Lo siguiente que se necesitará es un nodo que permita suscribirse (Figura 78) a los mensajes MQTT enviados por los nodos, en donde estará la trama de datos con las lecturas de los sensores. La Figura 79, muestra la configuración para la conexión con el Broker.

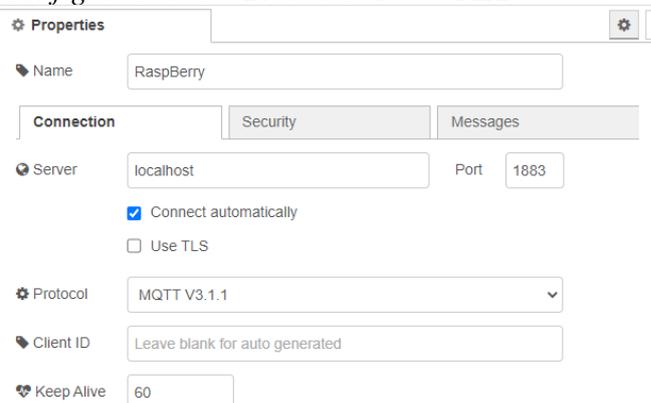
**Figura 78**

*Configuración del topic de recolección de datos*



**Figura 79**

*Configuración del Broker de Node-RED*



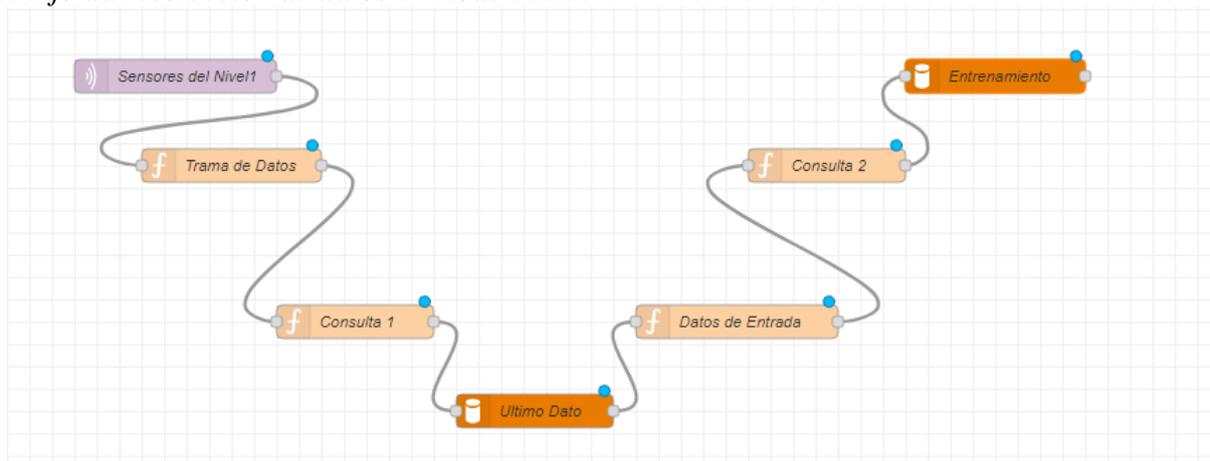
Una vez configuradas las dos conexiones necesarias, para recibir las lecturas realizadas por los sensores, y persistirlas en la base de datos, se ha creado un flujo, el cual se muestra en la Figura 80, en donde se realizan los siguientes procesos:

- El flujo recibe los datos a través de MQTT

- Pasa estos datos al siguiente nodo el cuál es una función que convierte la trama de datos en un array de valores numéricos y prepara una sentencia SQL para traer de la base de datos local, el último dato del día.
- El nodo llamado “Consulta 1”, ejecuta la sentencia preparada por el nodo anterior y devuelve los datos al siguiente nodo.
- El nodo “Datos de Entrada”, contiene un script que, basado en los valores de luminosidad del último dato insertado en el día actual y la diferencia de tiempos entre la hora del último dato y el dato actual, calcula el número de horas que lleva iluminado el cultivo y pasa todos estos valores al siguiente nodo.
- El nodo “Consulta 2” prepara la sentencia para almacenar los datos de los sensores y las horas de luz, en la base de datos.
- Por último, el nodo entrenamiento ejecuta la sentencia preparada en el nodo anterior, por lo que almacenara en los datos en la Tabla de “entrenamiento”.

### Figura 80

#### Flujo de recolección de datos en Node-RED



Nota: Todos los flujos y su programación se encuentran en el repositorio indicado en el Anexo 10.

## Integración del hardware dentro del sistema de agricultura vertical

El objetivo de esta etapa será integrar el hardware diseñado, con cada uno de los sensores, con la finalidad de que los nodos envíen datos reales de las lecturas de los sensores hacia el nodo central, por tal motivo se procede con la preparación del entorno siguiendo los siguientes pasos:

- Siembra de acelga dentro de la estructura vertical
- Instalación de las lámparas de crecimiento
- Instalación de sistema de ventilación
- Integración del nodo y sus sensores

Es importante recalcar que la manipulación de los actuadores en este punto se la realizará de manera manual, con la finalidad de generar una base de datos amplia de tal manera que abarque suficientes datos para las posibles etiquetas que se pueden dar dentro del ambiente en donde se encuentran las plantas. En la Figura 81 se muestra el resultado de estos procedimientos.

### Figura 81

*Integración del nodo para la recolección de datos*



### 3.4.17. Tratamiento de los datos

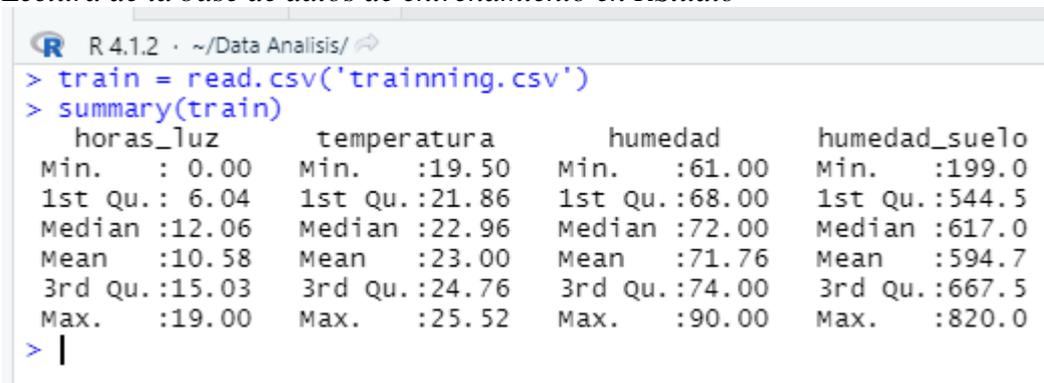
Luego de haber conseguido almacenar 13060 datos con las diferentes posibilidades de las variables ambientales, en el sistema “indoor”, se procede con el tratamiento de los datos, en donde como primera instancia se realizará el etiquetado de los mismo y subsecuentemente la reducción de esta base de datos con la finalidad de eliminar datos redundantes y que solo consumirían recursos de hardware del nodo central, esto es muy importante ya que estos últimos, son limitados.

#### Etiquetado de los datos

Para etiquetar los datos, lo primero que se ha de definir, son las diferentes etiquetas, para las posibles combinaciones de valores que se han obtenido gracias a los sensores y el nodo. Estas etiquetas se han considerado en base a la ficha técnica de los valores ambientales en los cuales la acelga, tiene crecimiento óptimo. Para esto, lo primero que se ha realizado es la exportación de la Tabla “entrenamiento”, en un archivo CSV con la ayuda de PhpMyAdmin. Una vez que se ha obtenido dicho archivo, se hace uso de la herramienta RStudio para un primer análisis de la base de datos recolectada. En la Figura 82, se muestra la lectura del archivo CSV y un resumen de este.

#### Figura 82

*Lectura de la base de datos de entrenamiento en RStudio*



```

R 4.1.2 · ~/Data Analysis/
> train = read.csv('training.csv')
> summary(train)
  horas_luz      temperatura      humedad      humedad_suelo
Min.   : 0.00    Min.   :19.50    Min.   :61.00    Min.   :199.0
1st Qu.: 6.04    1st Qu.:21.86    1st Qu.:68.00    1st Qu.:544.5
Median :12.06    Median :22.96    Median :72.00    Median :617.0
Mean   :10.58    Mean   :23.00    Mean   :71.76    Mean   :594.7
3rd Qu.:15.03    3rd Qu.:24.76    3rd Qu.:74.00    3rd Qu.:667.5
Max.   :19.00    Max.   :25.52    Max.   :90.00    Max.   :820.0
> |

```

A continuación, se exponen las conclusiones de estos resultados.

- Para el caso de este proyecto, solo existen dos posibles etiquetas para las horas de iluminación, falta de horas y suficientes horas.
- La temperatura tiende a sobrepasar de los rangos óptimos de crecimiento de la acelga por lo que se tomara en cuenta los tres posibles estados, temperatura baja, optima o alta.
- Para el etiquetado de los datos es que se podría descartar las etiquetas que impliquen a la humedad relativa ya que sus valores dentro del ambiente siempre se encuentran dentro del rango de crecimiento óptimo de la acelga según su ficha técnica. Además de que está de la mano con la temperatura ambiental.
- La humedad del suelo varía, puede variar en todo el rango de los valores del sensor FC-28, por lo que será preciso asignarle tres posibles estados, humedad baja, suficiente o en exceso.

En la Tabla 22, se muestran las etiquetas individuales de cada una de las variables ambientales a tomar en cuenta para el algoritmo de aprendizaje.

**Tabla 22**

*Tabla de etiquetas individuales de las variables ambientales*

<b>Variable</b>	<b>Rango</b>	<b>Etiqueta</b>
<b>Horas de luz (horas)</b>	< 12	FALTA DE HORAS LUZ
	>12	SUFICIENTES HORAS LUZ
<b>Temperatura (°C)</b>	<19	TEMPERATURA BAJA
	19 - 24	TEMPERATURA OPTIMA
	>24	TEMPERATURA ALTA
<b>Humedad del suelo</b>	<500	BAJA HUMEDAD DEL SUELO
	500 - 700	SUFICIENTE HUMEDAD DEL SUELO
	>700	EXCESO DE HUMEDAD DEL SUELO

Dado que el algoritmo de aprendizaje deberá clasificar el conjunto de datos como tal, se crea la combinación de etiquetas, obtenido así un total de 19 posibles estados del sistema,

para los cuales el mismo deberá manipularlos a través de los actuadores, dichas etiquetas se muestran en la Tabla 23.

**Tabla 23**

*Tabla de etiquetas del sistema*

ID	Nombre
1	TEMPERATURA BAJA - FALTA DE HORAS LUZ - BAJA HUMEDAD DEL SUELO
2	TEMPERATURA BAJA - FALTA DE HORAS LUZ - SUFICIENTE HUMEDAD DEL SUELO
3	TEMPERATURA BAJA - FALTA DE HORAS LUZ - EXCESO DE HUMEDAD DEL SUELO
4	TEMPERATURA BAJA - SUFICIENTES HORAS LUZ - BAJA HUMEDAD DEL SUELO
5	TEMPERATURA BAJA - SUFICIENTES HORAS LUZ - SUFICIENTE HUMEDAD DEL SUELO
6	TEMPERATURA BAJA - SUFICIENTES HORAS LUZ - EXCESO DE HUMEDAD DEL SUELO
7	TEMPERATURA OPTIMA - FALTA DE HORAS LUZ - BAJA HUMEDAD DEL SUELO
8	TEMPERATURA OPTIMA - FALTA DE HORAS LUZ - SUFICIENTE HUMEDAD DEL SUELO
9	TEMPERATURA OPTIMA - FALTA DE HORAS LUZ - EXCESO DE HUMEDAD DEL SUELO
10	TEMPERATURA OPTIMA - SUFICIENTES HORAS LUZ - BAJA HUMEDAD DEL SUELO
11	TEMPERATURA OPTIMA - SUFICIENTES HORAS LUZ - SUFICIENTE HUMEDAD DEL SUELO
12	TEMPERATURA OPTIMA - SUFICIENTES HORAS LUZ - EXCESO DE HUMEDAD DEL SUELO
13	TEMPERATURA ALTA - FALTA DE HORAS LUZ - BAJA HUMEDAD DEL SUELO
14	TEMPERATURA ALTA - FALTA DE HORAS LUZ - SUFICIENTE HUMEDAD DEL SUELO
15	TEMPERATURA ALTA - FALTA DE HORAS LUZ - EXCESO DE HUMEDAD DEL SUELO
16	TEMPERATURA ALTA - SUFICIENTES HORAS LUZ - BAJA HUMEDAD DEL SUELO
17	TEMPERATURA ALTA - SUFICIENTES HORAS LUZ - SUFICIENTE HUMEDAD DEL SUELO
18	TEMPERATURA ALTA - SUFICIENTES HORAS LUZ - EXCESO DE HUMEDAD DEL SUELO
19	CONDICIONES OPTIMAS

Una vez definidas las etiquetas del sistema, se procede con el etiquetado de los datos, para lo cual se ha generado un script de PHP, que lo haga de manera automática el cuál se lo puede observar en el Anexo x.

### **Reducción de la base de datos**

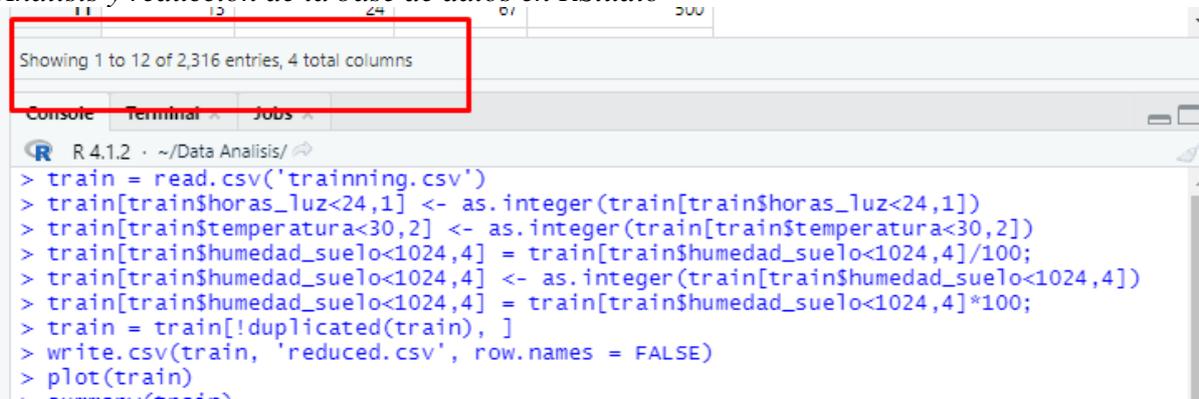
El objetivo de la reducción de la base de datos es reducir el consumo de recursos de hardware al eliminar datos redundantes dejando así los más significativos para la construcción de un modelo eficiente y eficaz. Para esto se ha hecho uso de RStudio, en donde previamente ya se ha cargado el archivo CSV como un análisis preliminar de los valores de las variables.

En la Figura 83, se muestra como se ha modificado las variables significativas del sistema, realizando los siguientes pasos:

- Se ha discretizado a valores enteros, los valores de las variables de horas luz y temperatura.
- Se ha dividido para 100 los valores de la humedad del suelo y posteriormente se discretizado a valores enteros.
- Se eliminan los datos repetidos y se vuelve a multiplicar por 100 los valores de la humedad del suelo.
- Se muestra un resumen de los datos obtenidos, juntamente con un gráfico de estos y se exporta el dataframe de R a un nuevo archivo CSV.

**Figura 83**

*Análisis y reducción de la base de datos en RStudio*



```

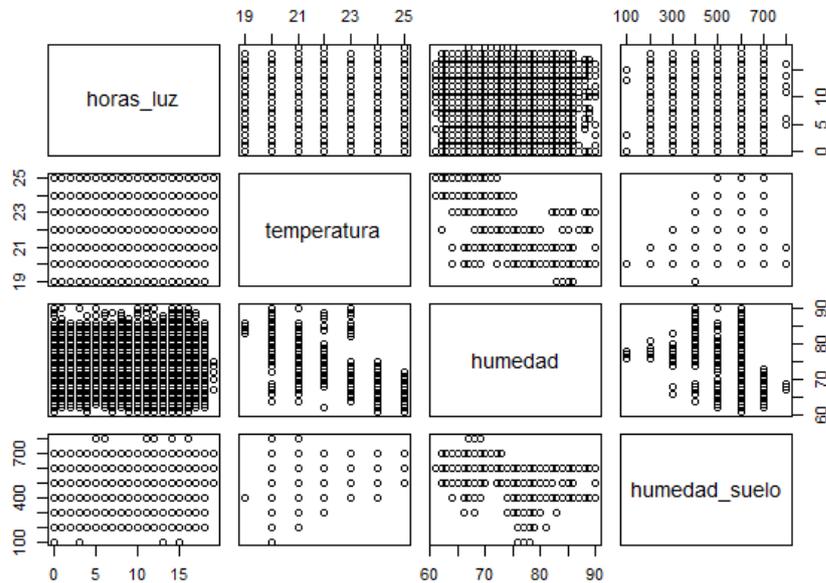
> train = read.csv('training.csv')
> train[train$horas_luz<24,1] <- as.integer(train[train$horas_luz<24,1])
> train[train$temperatura<30,2] <- as.integer(train[train$temperatura<30,2])
> train[train$humedad_suelo<1024,4] = train[train$humedad_suelo<1024,4]/100;
> train[train$humedad_suelo<1024,4] <- as.integer(train[train$humedad_suelo<1024,4])
> train[train$humedad_suelo<1024,4] = train[train$humedad_suelo<1024,4]*100;
> train = train[!duplicated(train), ]
> write.csv(train, 'reduced.csv', row.names = FALSE)
> plot(train)
< summary(train)

```

Con estos pasos se ha logrado reducir de un total de 13059 filas, a un total 2315, es decir se ha logrado reducir más de un 80% con respecto a la base de datos original, sin perder de vista los rangos a los que pertenecen cada variable. En la Figura 84 se muestra un gráfico multivariable de la nueva y reducida base de datos.

**Figura 84**

*Gráfico Multivariable de la base de datos reducida*



### 3.4.18. Selección del algoritmo de aprendizaje

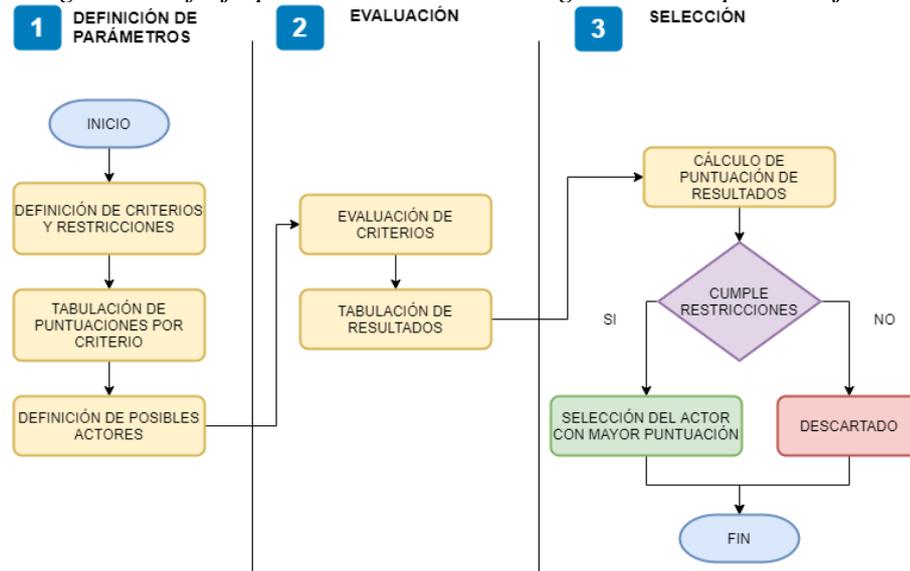
En el presente apartado, se aborda la selección de un algoritmo de aprendizaje supervisado, para lo cual se ha diseñado un enfoque multicriterio con la finalidad de optimizar los procesos de clasificación dentro de lo que respecta a la automatización y control del sistema de agricultura vertical urbano.

Se ha definido un proceso de selección de tres etapas, dentro de las cuales se ha definido subprocesos, con la finalidad de abordar el problema de una manera más sencilla. La Figura 85 muestra el flujograma que define el enfoque establecido, en este se puede apreciar las diferentes etapas divididas por una línea vertical, las cuales son:

- Definición de parámetros.
- Evaluación.
- Selección.

**Figura 85**

*Diagrama de flujo para la selección del algoritmo de aprendizaje*



A continuación, se procede con el cumplimiento de cada uno de los subprocesos definidos en el flujograma de selección.

### **Definición de criterios y restricciones**

Los criterios y restricciones estarán basadas en el cumplimiento de lograr una optimización de recursos computacionales, logrando a la vez una clasificación altamente eficaz por parte del algoritmo seleccionado. Es importante mencionar que las métricas de desempeño de los algoritmos de aprendizaje son fundamentales para la definición de criterios de desempeño, en la Tabla 24 se muestran los criterios a evaluar para dicha selección, seguidamente en la Tabla 25 se muestran las restricciones y por último en la Tabla 26 se tiene los rangos de calificaciones para la evaluación de dichos criterios.

**Tabla 24**

*Nomenclatura de criterios de evaluación*

<b>Criterio</b>	<b>Código</b>
<b>Tiempo de Respuesta</b>	TR
<b>Exactitud</b>	E
<b>Sensibilidad</b>	RC
<b>Precisión</b>	P
<b>Valor F</b>	F1

**Tabla 25**  
*Restricciones para los criterios de evaluación*

<b>Restricción</b>	<b>Código</b>
<b>TR&lt;1000ms</b>	R1
<b>E&gt;0.9</b>	R2
<b>F1&gt;0.85</b>	R3

**Tabla 26**  
*Rangos de calificaciones para los criterios de evaluación*

<b>Calificación</b>	<b>Puntaje</b>	<b>Rango</b>
<b>EXCELENTE</b>	4	0.95-1.00
<b>ALTO</b>	3	0.90-0.95
<b>MEDIO</b>	2	0.85-0.90
<b>BAJO</b>	1	0.80-0.85
<b>DEFICIENTE</b>	0	<0.80

### **Definición de actores**

Es importante evaluar el desempeño entre varios algoritmos de aprendizaje, con la finalidad de seleccionar el que mejor se adapte al modelo de datos. Para la selección del algoritmo de aprendizaje, se ha tomado en cuenta a dos candidatos, por su alta confiabilidad, bajo consumo de recursos computacionales y por su facilidad de integración con Node-RED, estos son:

- KNN - 3
- KNN - 5
- Árboles de decisión

Si bien es cierto que KNN - 3 y KNN - 5, se rigen por el mismo algoritmo, los costos computacionales y el desempeño de los dos será diferente, y también se tiene el algoritmo de Árboles de decisión, estos algoritmos son de los más conocidos y populares en Machine Learning, y otra de sus ventajas es que son de fácil interpretación.

## **Evaluación de criterios**

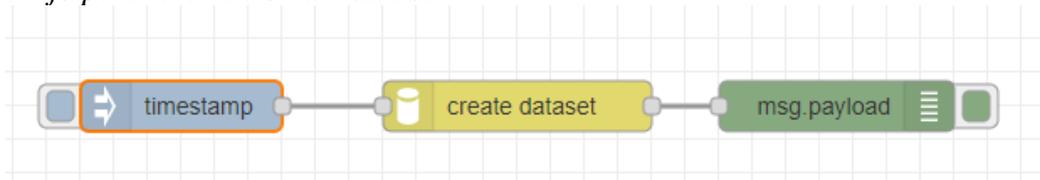
La evaluación de estos algoritmos se la ha realizado en el entorno de Node-RED, con la ayuda de la librería “node-red-contri-machine-learning” en su versión 1.0.8. Esta librería trabaja en conjunto con otras dependencias fundamentales que trabajan bajo Python, estas dependencias son:

- Python 3.6.4 o superior
- NumPy
- Pandas
- SciKit-Learn
- Y opcionalmente Tensorflow

Una vez que se ha resuelto dichas dependencias, se procede a crear los flujos en Node-RED, de tal manera de evaluar el desempeño de los algoritmos mencionados, teniendo como entrada la base de datos obtenida en el anterior apartado.

## **Creación del dataset**

En la Figura 86 presenta el primer flujo implementado, al insertar el timestamp, el nodo “créate dataset”, tomara el archivo de base de datos proporcionado en las configuraciones establecidas en la Figura 87, entre estas también encontramos las columnas de la base de datos que serán utilizadas como entrada y cuál de ellas será la salida, el porcentaje de la base de datos asignado para entrenamiento y para prueba.

**Figura 86***Flujo para la creación del dataset***Figura 87 Configuración para la creación del dataset**

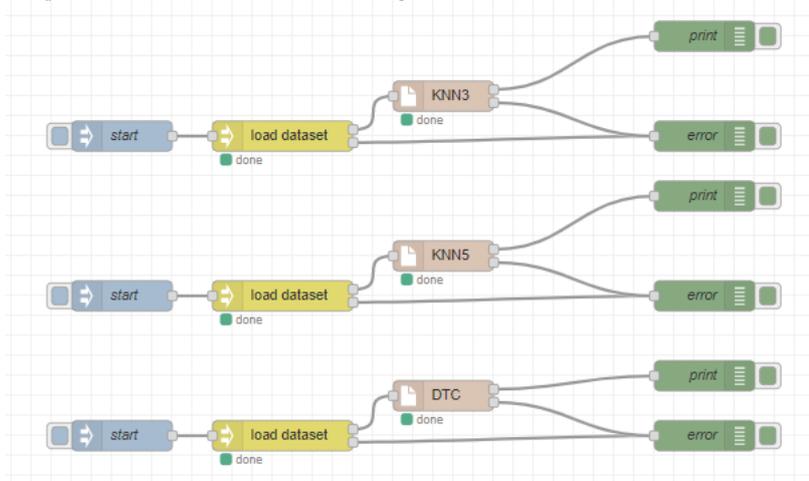
La imagen muestra la interfaz de configuración de un nodo. El título es 'Properties'. Los campos de configuración son:

- Path: test/reducida.csv
- Save folder: test/datasets
- Save name: sufR
- Input: 0,1,2,3
- Output: 4
- Training partition: 80
- Shuffle dataset:
- Seed: 0

Una vez que este nodo se ejecute se podrá observar en las rutas especificadas los archivos train y test, con los que posteriormente se construye el modelo según el algoritmo.

Para crear el modelo de cada uno de los algoritmos, se utiliza el flujo mostrado en la Figura 88, en donde el nodo “load dataset”, cargará un dataset específico en base a las configuraciones mostradas en la Figura 89, en donde se puede apreciar, se está cargando la partición de “Training”, en base al dataset creado anteriormente.

**Figura 88**  
Flujos de entrenamiento KNN y DT



**Figura 89**  
Configuración del nodo “load dataset”

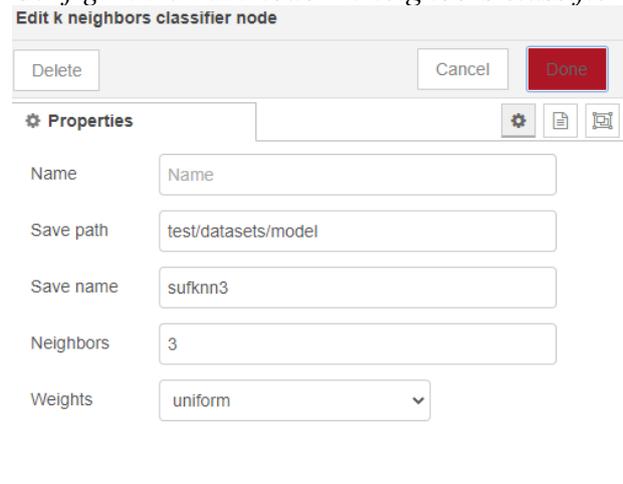
The screenshot shows a configuration window titled 'Edit load dataset node'. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below is a 'Properties' section with several fields and checkboxes:

- Name:** A text input field containing 'Name'.
- Dataset folder:** A text input field containing 'test/datasets'.
- Dataset name:** A text input field containing 'sufR'.
- Partition:** A dropdown menu with 'Training' selected.
- Load input:** A checkbox that is checked.
- Load output:** A checkbox that is checked.

Las configuraciones mostradas en la Figura 90, corresponden al nodo “k neighbors classifier”, se estableció el parámetro “Neighbors” (el cual representa el número de vecinos utilizado por el algoritmo), en un valor de 3. El path utilizado para almacenar el modelo será “test/datasets/model” y el nombre del archivo en este caso será “sufknn3”.

**Figura 90**

*Configuración del nodo “k neighbors classifier” para KNN-3*



Delete Cancel Done

⚙️ 📄 🔄

Name Name

Save path test/datasets/model

Save name sufknn3

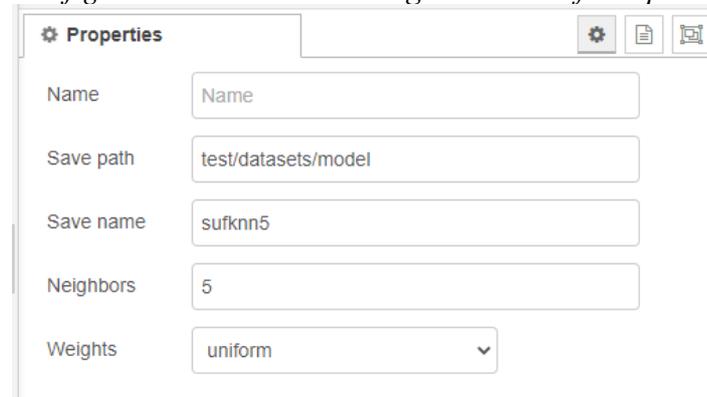
Neighbors 3

Weights uniform

Para el caso de KNN-5 se establece el número de vecinos a un valor de 5 y el nombre con el que se almacenará el modelo del algoritmo será “sufknn5”, como se observa en la figura 91.

**Figura 91**

*Configuración del nodo “k neighbors classifier” para KNN-5*



⚙️ 📄 🔄

Name Name

Save path test/datasets/model

Save name sufknn5

Neighbors 5

Weights uniform

Para la generación del modelo del algoritmo de Árboles de decisión, prácticamente solo se cambia el nodo de “k neighbors classifier”, por el nodo “decision tree classifier”, en la Figura 92 se puede apreciar la configuración del nodo, en donde se carga el mismo dataset que en los casos anteriores, pero el modelo generado se almacenará con el nombre “sufdtc”.

**Figura 92**  
 Configuración del nodo “*decision tree classifier*”

Property	Value
Name	Name
Save path	test/datasets/model
Save name	sufdtc
Max depth	Max depth of the tree.
Criterion	gini
Splitter	best

Luego de que se ha entrenado cada uno de los algoritmos, lo siguiente será probar cada uno de los modelos obtenidos, para esto se hará uso del nodo “predictor”, este nodo recibe en su entrada un dataset de casos a analizar, en base a un modelo, el cual se especifica en sus configuraciones, para obtener en su salida las predicciones de los datos de entrada recibidos. Por otra parte, se tiene el nodo “assessment” este nodo recibe en su entrada un dataset, el cual puede tener una “bandera”, indicando si los datos son reales o predichos y cada vez que recibe datos predichos hace el análisis comparativo, en base a los últimos datos reales que recibió en su entrada, la salida de este depende de la configuración de este. Para este caso se utilizará para obtener la precisión de cada uno de los algoritmos, así como su matriz de confusión.

En la Figura 93, se ha creado un flujo en donde se ha conectado un nodo “load dataset” tomando la partición de “Testing” del dataset creado en un inicio, a la entrada de un nodo “assessment” juntamente con tres diferentes nodos del tipo “predictor”, estos últimos corresponden a las predicciones de cada uno de los algoritmos a evaluar. La configuración de cada uno de estos nodos se puede apreciar en la Figura 94.



**Figura 96**

*Configuración del nodo “assessment” para obtener matriz de confusión*

The image shows a configuration window for a node. At the top, there is a 'Properties' tab. Below it, there is a 'Name' field with the placeholder text 'Name'. Underneath, there is a 'Type' dropdown menu with 'confusion matrix' selected.

Para comprobar estos flujos es necesario hacer un “Deploy” de los cambios realizados y luego, los pasos a seguir son:

- Crear el dataset, dando un clic en el nodo “timestamp” del flujo de creación del dataset.
- Generar los modelos a partir del dataset creado, se debe dar un clic en cada uno de los nodos “timestamp” del flujo de entrenamiento.
- Evaluar cada uno de los algoritmos, empezando por dar un clic en el primer nodo “timestamp” del flujo de evaluación, luego lo mismo en cada uno de los subsecuentes. La inserción del “timestamp”, también permite determinar el tiempo de respuesta de cada uno de los modelos al final de cada flujo.

**Tabulación de resultados**

En el Anexo 6 se adjunta la tabulación de la matriz de confusión de cada uno de los algoritmos de aprendizaje, con el cálculo de cada uno de los criterios evaluados, de los cuales se obtiene el resultado final, mostrado en la Tabla 27.

**Tabla 27**

*Valores de las métricas para cada algoritmo*

Algoritmo\Criterio	TR (ms)	TR (normalizado)	E	P	RC	F1
<b>KNN3</b>	378	0.8488	0.9490	0.8365	0.9540	0.8531
<b>KNN5</b>	439	0.8244	0.9407	0.8265	0.9523	0.8435
<b>DT</b>	103	0.9588	0.9915	0.9947	0.9946	0.9946

Cabe mencionar que para la normalizar los valores del criterio “TR”, se ha utilizado el valor máximo permitido por la restricción “R1”, siendo el valor de “TR normalizado”:

$$TR(normalizado) = (2500(ms) - TR(ms))/2500(ms)$$

### Cálculo de puntuación de resultados

En la siguiente Tabla se puede observar los puntajes obtenidos en cada criterio, por cada uno de los algoritmos.

**Tabla 28**

*Puntuación por algoritmo de aprendizaje*

Algoritmo\Criterio	TR (normalizado)	E	P	RC	F1	Total
<b>KNN3</b>	2	3	1	4	2	<b>12</b>
<b>KNN5</b>	2	3	1	4	1	<b>11</b>
<b>DT</b>	4	4	4	4	4	<b>20</b>

### Selección

De la Tabla 28, se puede concluir que KNN5 no cumple con la restricción “R3”, de tal manera que, siguiendo el flujograma de selección, se determina a este actor como descartado y como último paso, se tiene la selección del algoritmo con mejor puntuación, se ha obtenido los resultados mostrados en la Tabla 27, en donde se puede observar que la mejor puntuación en base a la evaluación de los múltiples criterios realizada, es el presentado por el algoritmo de Árbol de decisión con un puntaje total de 20. Por tal motivo se selecciona a este como el algoritmo de aprendizaje para la implementación del “Sistema autónomo de monitoreo y control de condiciones ambientales en cultivos verticales hortícolas urbanos mediante una Red de sensores”.

#### 3.4.19. Desarrollo de la aplicación de monitoreo y control

Para el desarrollo de la aplicación móvil de monitoreo y control, se ha hecho uso de varias herramientas como los son:

- Firebase Realtime
- Ionic + Angular
- Node-RED

Por lo que a continuación se da una explicación breve, de los pasos realizados para el cumplimiento de este objetivo.

### **Creación del proyecto con Ionic**

Para poder empezar a trabajar con el framework de Ionic, primeramente, es necesario tener en cuenta que se debe contar con ciertas dependencias, estas son:

- Node Js
- Npm

Si el sistema cuenta con estas dependencias, se procede con la instalación global de Ionic en el ordenador, y esto se logra desde una terminal, ejecutando el comando “npm install -g @ionic/cli”, Npm descargará los archivos fuentes de Ionic para que pueda ser utilizado de manera local.

Una vez que se cuenta Ionic instalado en la máquina, se procede con la creación del proyecto, Ionic cuenta con algunas plantillas de proyectos, sin embargo, se ha optado por la creación de un proyecto en blanco, con el comando “ionic start smart-urban-farm blank”, Ionic creará un nuevo proyecto en el directorio desde el que se haya emitido el comando, no sin antes proporcionarle el framework de javascript que se utilizará en el proyecto de Ionic, con las flechas del teclado se ubica en la opción de Angular y se presiona la tecla “Enter” (Figura 97).

**Figura 97***Selección del framework de desarrollo*

```
D:\UTN\10mo_2daPARTE\CODIGO>ionic start smart-urban-farm blank

Pick a framework!

Please select the JavaScript framework to use for your new app.
--type option.

? Framework:
> Angular | https://angular.io
  React   | https://reactjs.org
  Vue     | https://vuejs.org
```

Al culminar la creación se podrá observar una nueva carpeta con el nombre “smart-urban-farm”, el cual es el nombre que se le proporcionó a Ionic para la creación del proyecto.

### Conexión con Firebase Realtime

Para la conexión con Firebase Realtime, es necesario dirigirse a la carpeta del proyecto creada anteriormente y a través de Npm instalar “firebase” y “angular/fire”, esto se logra a través del comando “npm install -save firebase @angular/fire”.

En un apartado anterior se mostró la creación del proyecto Firebase, para la creación de la base de datos Realtime, en este punto es necesario regresar a la consola de firebase y agregar una nueva app web, proporcionamos el nombre y se copia la configuración para el SDK de Firebase, que será proporcionado por la consola (Figura 98).

**Figura 98***Configuración del SDK proporcionada por la consola de Firebase*

```
const firebaseConfig = {
  apiKey: "AIzaSyf",
  authDomain: "sm",
  databaseURL: "ht",
  projectId: "sma",
  storageBucket: "",
  messagingSenderId: "",
  appId: "1:32184",
  measurementId: ""
};
```

Una vez que se tenga estos datos en el portapapeles, hay que dirigirse a la ruta “app/src” dentro de la carpeta del proyecto y modificar el archivo “app.module.ts”, en donde se insertan los datos de configuración para el SDK de Firebase y se inserta la las líneas

señaladas con flecha en la Figura 99, con la finalidad de importar e inicializar los módulos correspondientes.

**Figura 99**

*Configuración de Ionic para la comunicación con Firebase*

```
import { AngularFireModule } from '@angular/fire';
import { AngularFireDatabaseModule } from '@angular/fire/database';

const firebaseConfig = {
  apiKey: 'AIza...',
  authDomain: '...',
  projectId: '...',
  storageBucket: '...',
  messagingSenderId: '...',
  appId: '1:22:...',
  measurementId: '...',
  databaseURL: '...'
};

@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [
    BrowserModule, IonicModule.forRoot(),
    AppRoutingModule,
    AngularFireModule.initializeApp(firebaseConfig),
    AngularFireDatabaseModule,
  ],
  providers: [{ provide: RouteReuseStrategy, useClass: IonicRouteStrategy }],
  bootstrap: [AppComponent],
})
```

Para realizar las consultas a la base de datos, se ha definido dos funciones que permitirán leer datos y modificarlos, en base a una ruta específica del JSON que la representa, su código se puede apreciar en la Figura 100. Esto se ha implementado dentro de un servicio de angular, con la finalidad de poder llamarlas a lo largo de toda la aplicación.

**Figura 100**

*Funciones para la lectura y escritura de datos en Firebase*

```
getData(path: string){
  return this.db.list(path).snapshotChanges()
    .pipe(map((actions)=>{
      const obj = {};
      actions.map(
        (data: any)=>{
          const key = data.key;
          obj[key]= data.payload.val();
          return;
        }
      );
      return obj;
    }));
}

updateData(path: string, data: any){
  this.db.database.ref(path).set(data).then().catch();
}
```

## Creación de pantallas

Para la interfaz gráfica de la aplicación se ha estimado la implementación de 4 pantallas, y estas son:

- Pantalla de inicio o portada
- Menú de la aplicación
- Monitoreo
- Control

Usando un terminal de línea de comandos, ubicado dentro de la carpeta del proyecto se crea cada una de las pantallas mencionadas y respectivamente con los siguientes comandos.

- `ionic g page intro`
- `ionic g page menu`
- `ionic g page monitoring`
- `ionic g page control`

Una vez realizado esto se genera dentro de la ruta “src/app” una carpeta por cada una de estas pantallas, y las cuales cuentan con un archivo html, css y ts, los cuales permiten generar la estructura, apariencia y comportamiento de cada una de las pantallas.

En las figuras 101, 102, 103 y 104, se muestra la apariencia de cada una de las pantallas creadas anteriormente luego de agregar algo de contenidos a los archivos generados en cada una de las carpetas correspondientes a cada pantalla.

**Figura 101***Inicio/portada de la aplicación***Figura 102***Menú de la aplicación*

**Figura 103***Pantalla de monitoreo de la aplicación***Figura 104***Pantalla de control de la aplicación*

Se ha hecho uso del servicio antes mencionado, que aloja las funciones que se comunican con Firebase Realtime, en el caso de control y para el gráfico de los datos adicionalmente se ha hecho uso de la librería de Javascript Chart.js, en el caso de monitoreo.

Se ha programado el comportamiento de la aplicación, de tal manera que se encuentre sincronizada con la base de datos en tiempo real y de esta manera poder saber, cuando sea y de donde sea, que es lo que ocurre en el sistema.

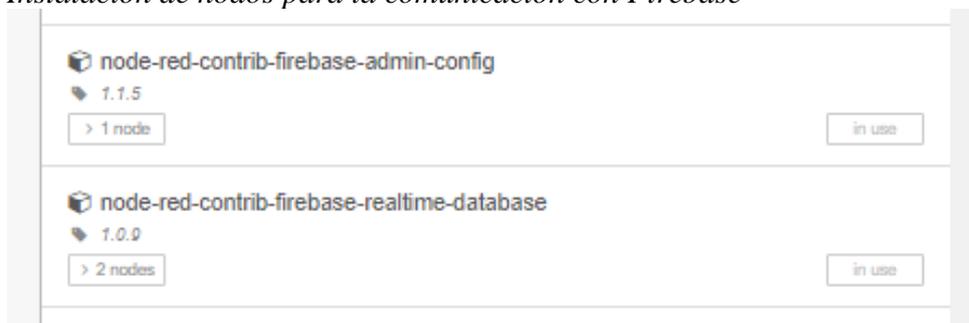
Dado que el enfoque de este proyecto no es el diseño de aplicaciones móviles, no se entra en detalle al respecto, sin embargo, el código fuente de la aplicación, está disponible en github: <https://github.com/cmruizg777/smart-urban-farm>.

## Flujos en Node-RED

Para culminar de enlazar los sistemas, es necesario crear un flujo, dentro de Node-RED, de tal manera que envíe los datos recibidos por los sensores hacia Firebase, y este a su vez los enviará a la aplicación. Para esto es preciso hacer uso de una librería más, disponible en la paleta de administración de Node-RED, llamada “node-red-contrib-firebase-realtime-database”, y a su vez “node-red-contrib-firebase-admin-config”, ya que la primera depende de esta última. Estas librerías agregarán nuevos nodos que permitirán la comunicación con Firebase. En la Figura 105, se puede observar estas librerías una vez instaladas, desde la paleta de administración.

### Figura 105

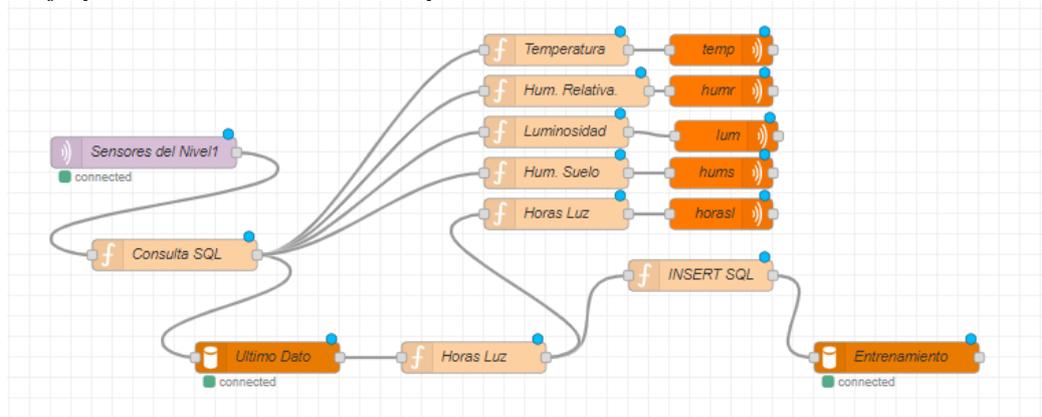
*Instalación de nodos para la comunicación con Firebase*



Para monitorear las variables ambientales de cada nivel, solo es necesario realizar una ligera modificación al flujo de recolección de datos, en donde paralelamente a calcular las horas de iluminación del cultivo, se agregará una función por cada variable ambiental recolectada y la salida de cada una de estas un nodo de Firebase, con la finalidad de que solo modifique el dato correspondiente de esa variable en la base de datos, esto se puede apreciar de manera más clara en la Figura 106 y en la Figura 107 una de las configuraciones de estos nodos.

**Figura 106**

*Flujo para recolección de datos y monitoreo*



**Figura 107**

*Configuración del nodo de temperatura.*

⚙️ Properties

**Firestore database communication**

**Firestore Admin**  
 Firebase smart urban farm

**Node Name**  
 temp

**Write operation**  
 Set

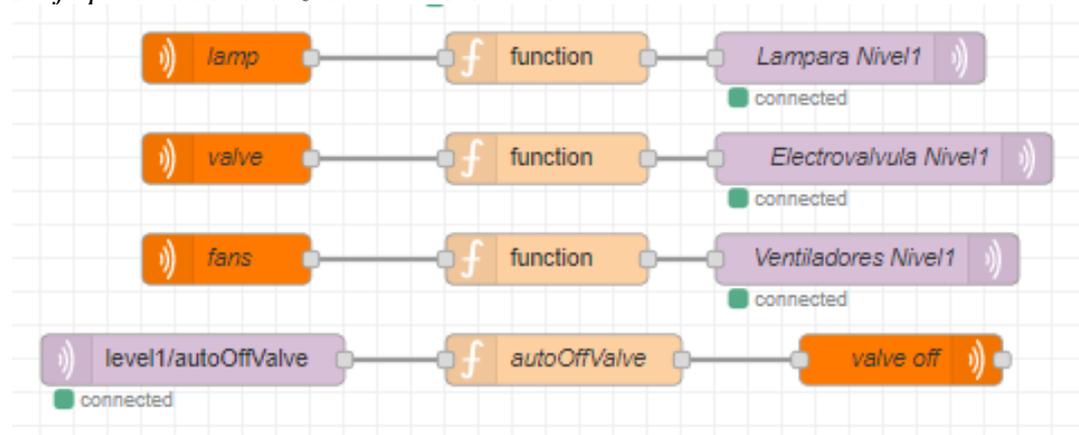
**Database Ref**  
 temperatura/luminosidad/nivel1

Por último, para la parte de control, simplemente se ha usado el nodo de Firebase, para traer los datos especificados en las configuraciones de cada uno de estos nodos, para

posteriormente publicarlos a través de MQTT. En la Figura 108 se muestra el flujo establecido para el control de los actuadores. De tal manera que si la app, envía una solicitud de encender la lámpara del nivel 1, este flujo recibirá estos datos y los hará llegar hasta los nodos y estos procesarán la orden.

### Figura 108

*Flujo para la sincronización con Firebase*



## **Capítulo IV. Implementación y resultados**

### **4.1. Introducción**

Hasta el momento, siguiendo la metodología Investigación Acción, se ha realizado un proceso de reflexión de la situación actual de la producción de alimentos, y su posible demanda en base a la inevitable tasa de crecimiento poblacional, además se ha se ha definido la problemática del presente proyecto, el planteamiento de objetivos y se ha establecido un alcance, como primera acción del proyecto se ha realizado una investigación de los fundamentos teóricos necesarios para llevar a cabo los objetivos planteados. Con esto se ha culminado un primer ciclo dentro de la metodología Investigación Acción, subsecuentemente se ha vuelto a realizar observaciones.

El presente capítulo aborda la implementación final del sistema, en donde se integra el diseño de cada uno de los componentes descritos en el Capítulo III, para la producción del cultivo de acelga (*beta vulgaris*), se definen también los flujos para determinar los comandos a enviar por el Nodo Central hacia los sensores después de la clasificación de un conjunto de datos de variables ambientales recibidas, el siguiente apartado muestra las pruebas de funcionamiento del sistema en donde se ha verificado el desempeño de los sensores, y se ha definido acciones en caso de lecturas erróneas, para con esto poner en marcha la producción de los cultivos. En el apartado de Análisis de resultados, se documenta el desarrollo de los cultivos dentro del sistema y fuera de él, los acontecimientos a lo largo del tiempo, y una comparativa. Para finalizar este capítulo, se concluye con un análisis de costo y beneficio del proyecto en base a los resultados obtenidos.

### **4.2. Implementación**

En el presente apartado se muestra la integración de los componentes del sistema, partiendo del modelo implementado para la recolección de datos se ha realizado

#### 4.2.1. Implementación del cultivo

A continuación, se detalla el procedimiento realizado para el sembrío de los cultivos de acelga en la estructura vertical.

**Figura 109**

*Implementación del cultivo*



- Como primera fase de implementación, se ha considerado la preparación del suelo y la siembra de los cultivos, bajo la asesoría del experto Agrónomo Ing. Clemente Alvarado, para la cuál se ha considerado abonar el sustrato de tal manera que cuente con los nutrientes necesarios.
- Se ha llenado cada una de las cajas de cada nivel, con el sustrato abonado, hasta una altura de 25 cm.
- Se ha procedido a la siembra de las plántulas en la estructura vertical, cavando pequeños agujeros de 2cm en el sustrato y dejando un espacio de 20 cm entre cada una de ellas.

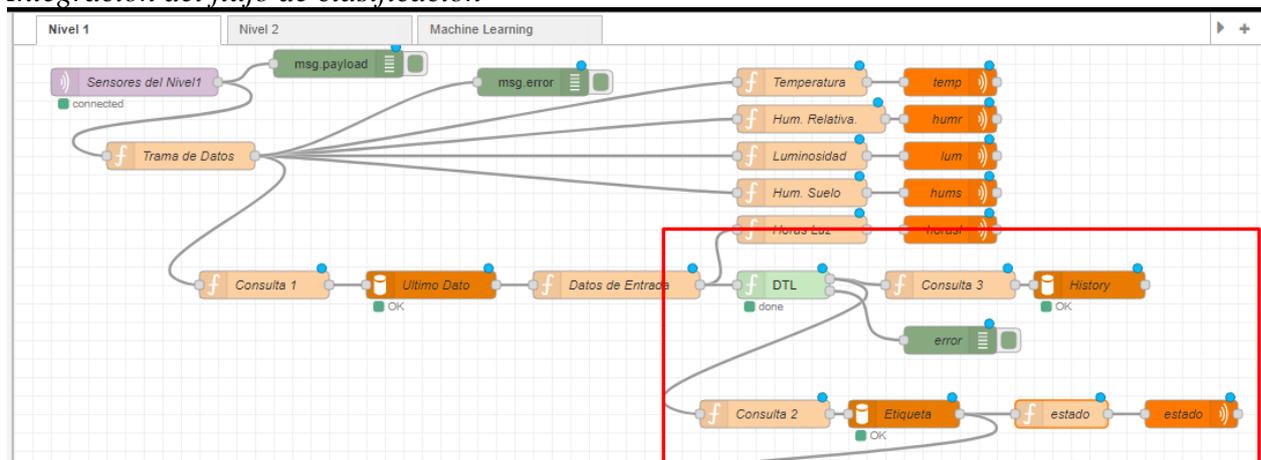
#### 4.2.2. Implementación Nodo Central

Al igual que en la etapa de recolección de datos, el Nodo Central recibe la información enviada por los nodos sensores, con la diferencia de que ahora también la clasifica los datos, emite comandos a los Nodos Sensores, con la finalidad de controlar las variables ambientales en base a dicha clasificación, y también envía los datos a Firebas

Realtime, por ello ha sido necesario realizar ciertas modificaciones en los flujos de Node-RED las cuales se detallan a continuación.

La Figura 110 muestra la primera etapa del flujo de Node-RED, en donde la primera modificación, se puede observar, está en la salida del nodo “Trama de Datos”, el cuál como se explicó toma el “string” de datos recibidos a través de MQTT y mediante líneas de código de Javascript los convierte en un “array” de valores numéricos, se ha conectado esta salida a la entrada de cuatro funciones, las cuales toman cada uno de los datos recibidos por los sensores, es decir: temperatura, humedad relativa, luminosidad y humedad del suelo, estas funciones, envían este dato a su salida para la actualización del campo correspondiente en Firebase Realtime.

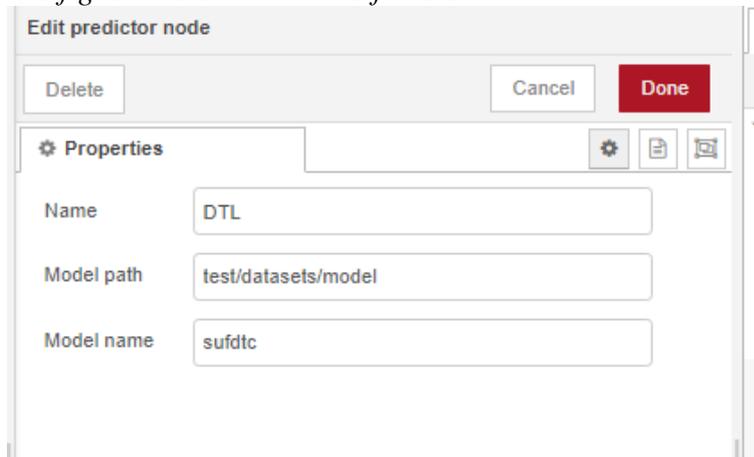
**Figura 110**  
*Integración del flujo de clasificación*



También se ha agregado a la salida del nodo “Datos de Entrada”, que es en donde se calculan las horas de iluminación que ha tenido el nivel correspondiente y se reúnen todos los datos ambientales para que el modelo entrenado de Árboles de decisión los clasifique con un número de etiqueta, el nodo que se encarga de la clasificación es “DTL”, el cuál corresponde a un “predictor node”, de la librería “firebase-contrib-machine-learning” y su configuración se observa en la Figura 111, la cual únicamente recibe como parámetros, un nombre para el

nodo, el path donde se encuentra ubicado el modelo ya entrenado y el nombre de archivo del modelo.

**Figura 111**  
*Configuración nodo de clasificación*



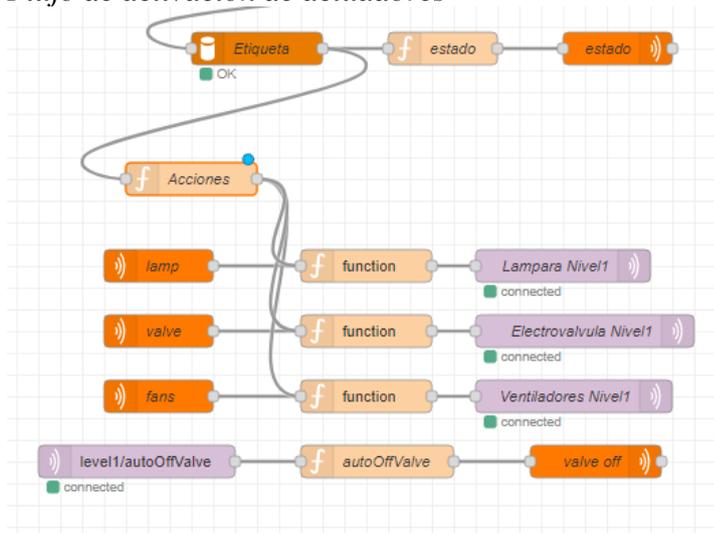
El nodo “DTL”, cuenta con dos diferentes salidas, una correspondiente a las etiquetas asignadas para los datos de entrada y la segunda que cuenta con los errores ocasionados en un determinado momento. La primera salida del nodo “DTL”, se ha conectado a una función de Javascript a la cual se la ha llamado “Consulta 2”, esta función obtendrá los datos numéricos del nodo de predicción y genera la sentencia SQL para obtener la descripción completa de la etiqueta. También se conecta a la función “Consulta 3”, que igualmente genera una sentencia SQL, pero para almacenar los datos en la Tabla “historial” en la base de datos local.

En la Figura 112 se muestra la interconexión de los diferentes nodos una vez que se obtiene la descripción o nombre de la etiqueta en el nodo “Etiqueta”, a la salida de este se ha conectado dos funciones: una función llamada “etiqueta”, cuya función es enviar la descripción de la etiqueta obtenida por nodo de predicción a Firebase Realtime, y también a una función a la cual se la ha llamado “Acciones”, en esta función se genera un “array” de valores booleanos de 3 posiciones, en donde en cada posición se indica:

- Posición 0. – Si la descripción de la etiqueta contiene la palabra “TEMPERATURA ALTA”, con valor “true”, el comando de activación de los ventiladores.
- Posición 1. – Si la descripción de la etiqueta contiene la palabra “FALTA DE HORAS LUZ”, con valor “true”, el comando de activación de las luces de crecimiento (lámparas).
- Posición 2. – Si la descripción de la etiqueta contiene la palabra “BAJA HUMEDAD DEL SUELO”, con valor “true”, el comando de activación de la electroválvula.

**Figura 112**

*Flujo de activación de actuadores*



En todos los casos, si la condición no se cumple, cada posición tendrá un valor por defecto de “false”, con lo cual se enviará el comando de apagar el actuador correspondiente. A la salida de esta función se conectan tres diferentes funciones, donde cada una de estas toma una posición del “array” (Figura 113), para publicar el mensaje correspondiente al Nodo Sensor a través de MQTT.

**Figura 113***Script de activación de actuadores*

```

1- if(Array.isArray(msg.payload)){
2   let etiqueta = msg.payload[0].nombre;
3   msg.payload = etiqueta;
4-  if(etiqueta!="CONDICIONES OPTIMAS"){
5     const d = new Date();
6     let state = etiqueta.split(" - ");
7     let actions = [false,false,false];
8     if(state[1]=="FALTA DE HORAS LUZ" && d.getHours() >= 5) actions[0] = true;
9     if(state[2]=="BAJA HUMEDAD DEL SUELO") actions[1] = true;
10    if(state[0]=="TEMPERATURA ALTA") actions[2] = true;
11    msg.payload = actions;
12  }
13 }
14 return msg;

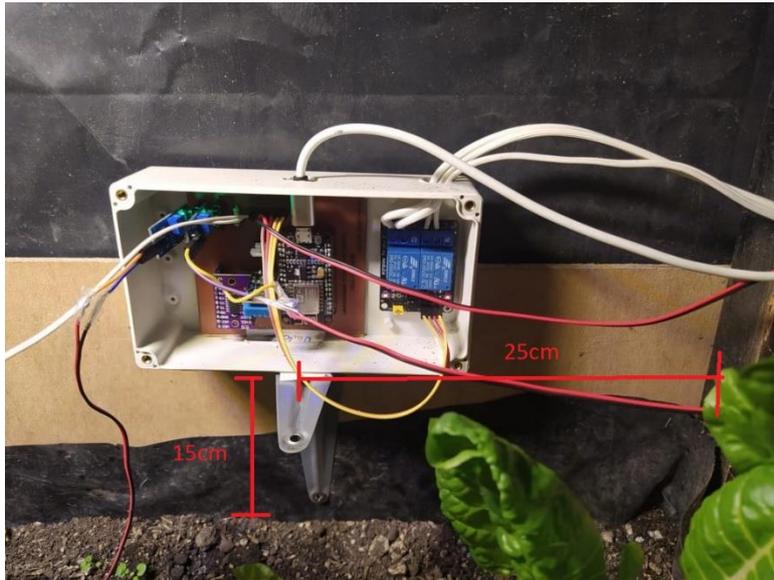
```

### 4.2.3. Implementación Nodos Sensores

En la etapa de recolección de datos, la única funcionalidad de los Nodos Sensores era la de conectarse a la red MQTT y enviar la lectura de los sensores al Nodo Central, en esta nueva etapa del sistema, los Nodos Sensores también se encargarán de activar o desactivar los actuadores del sistema, esto se realiza cada vez que reciben los comandos desde el Nodo Central a través de MQTT. En la única ocasión que el Nodo Sensor actúa por si solo, es para desactivar la electroválvula, esto con la finalidad de evitar inundar los cultivos. Por tal motivo, se ha integrado el nodo, los sensores y actuadores, en lugares estratégicos, de tal manera que cada componente cumpla con su funcionalidad sin ser afectado por los demás.

#### 4.2.3.1. Instalación del Nodo

Como se explicó anteriormente, se ha integrado todos los elementos del Nodo Sensor en una placa PCB y estos dentro de una caja plástica. Se ha considerado la ubicación del Nodo Sensor, de tal manera que no interfiera con el crecimiento de las plantas y tampoco sea afectado por la humedad, ni el sistema de riego, esto se aprecia en la Figura 114. Se ha instalado el nodo a una distancia de 25cm desde la pared lateral derecha de cada nivel, a una altura de 15cm desde el nivel del sustrato, con la ayuda de un “pie de amigo” y cinta faz.

**Figura 114***Ubicación del Nodo Sensor***4.2.3.2. Sensor DHT11**

Debido a que el nodo se ha ubicado dentro del cultivo, el sensor de temperatura y humedad relativa se ha mantenido sobre la placa PCB, como se observa en la Figura 115, de esta manera se facilita esta instalación sin que esto influya negativamente a la toma de datos.

**Figura 115***Ubicación del sensor DHT11***4.2.3.3. Sensor FC-28.**

Se ha utilizado dos sensores de humedad del suelo FC28, para cada nivel, se ha optado por ubicarlos así, el primero entre las plantas 1 y 2, y el segundo entre las plantas 9 y

10, cada uno a una distancia de 10cm de cada planta y enterrándolos a su máxima profundidad posible, para asegurar la humedad de las raíces. Esto se muestra en la Figura 116.

**Figura 116**

*Ubicación del sensor FC-28*

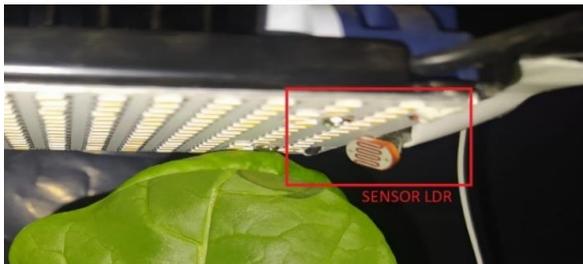


#### **4.2.3.4. Sensor LDR.**

La principal función del sensor LDR, es medir la intensidad de luz que están recibiendo las plantas, con la finalidad de controlar y calcular la cantidad de horas de iluminación que éstas han recibido a lo largo del día, es indispensable que no sea obstruido por ningún obstáculo, por tal motivo y basados en la observación del crecimiento de las hojas del cultivo, se ha determinado ubicar el sensor lo más cercano de la lámpara de tal manera que no sea cubierto por las grandes hojas de acelga. En la Figura 117 se puede observar una hoja de acelga que ha crecido hasta la altura de la lámpara y también la ubicación del sensor LDR junto a esta última.

**Figura 117**

*Ubicación del sensor LDR*



La disposición de los actuadores es fundamental para que estos puedan regular y afectar positivamente a todo el nivel de cultivo, por tal motivo a continuación se detallan las consideraciones tomadas para la instalación de los mismo.

#### **4.2.3.5. Instalación Lámparas de crecimiento**

Es importante que las plantas reciban la mejor iluminación posible de las lámparas de crecimiento, dado que el sistema diseñado se encuentra al interior de una vivienda, sin luz solar, esto quiere decir que estas lámparas serán totalmente responsables de que las plantas puedan realizar el proceso de la fotosíntesis. La ubicación de las lámparas deberá cubrir la zona de cultivo, por lo que se las ha centrado desde una vista superior y se ha tomado en cuenta la hoja de datos del fabricante, donde se especifica que, para una altura de 40cm las lámparas cubrirán un área de iluminación de 100cmx70cm lo suficiente para cubrir a un nivel de plantación.

Para instalar adecuadamente estas lámparas se ha realizado los cálculos correspondientes en base a las dimensiones de los materiales, las cuales se muestran en la Tabla 29.

**Tabla 29**  
*Medidas del sistema vertical*

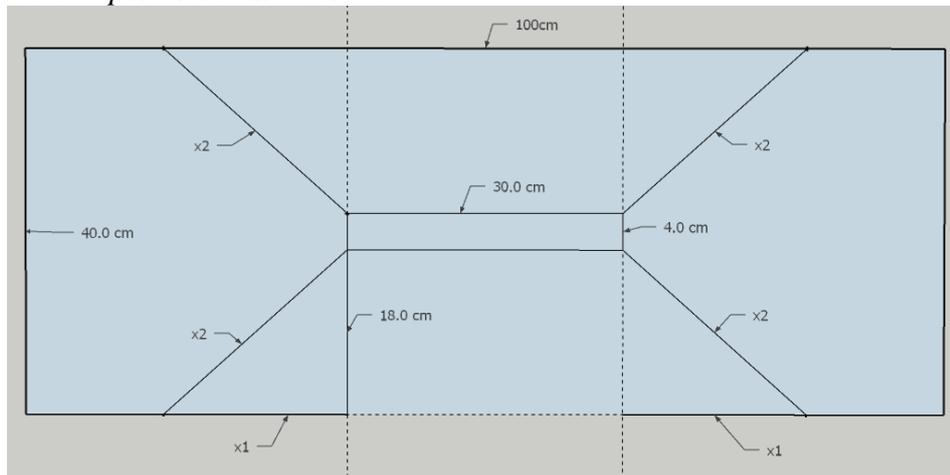
<b>COMPONENTE</b>	<b>LONGITUD</b>
<b>Espacio Entre Niveles</b>	55cm
<b>Ancho Nivel</b>	100cm
<b>Fondo Nivel</b>	40cm
<b>Ancho Lámparas</b>	30cm
<b>Alto Lámparas</b>	4cm
<b>Fondo Lámparas</b>	7cm
<b>Margen Hasta el agujero para arnés</b>	1.5cm

En base a estas medidas se ha trazado la Figura 117 que muestra una vista frontal de las medidas de un nivel centrando la lámpara a 35cm de cada uno de los lados del nivel, la

Figura 118 una vista superior y la Figura 119 un triángulo rectángulo imaginario que se forma de la proyección de una de las cuerdas del arnés, sobre el plano del límite superior del nivel, donde dicha cuerda, es la hipotenusa.

**Figura 118**

*Vista superior sistema vertical*

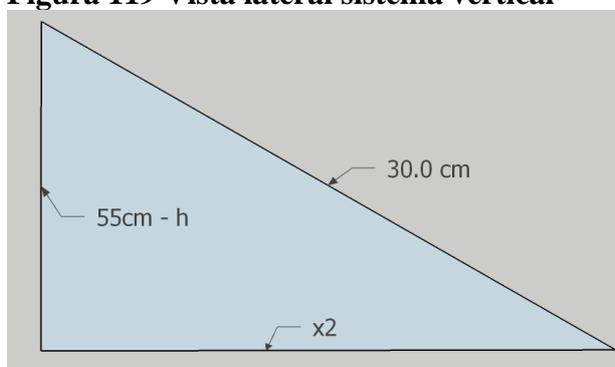


De esta Figura se obtiene la ecuación 5

$$x_2^2 = x_1^2 + 18^2 \quad (6)$$

Donde “x2” es la proyección de la cuerda del arnés sobre el plano superior del nivel y “x1” la distancia desde el un extremo del arnés hasta el punto donde está ubicada la lámpara sobre ese mismo plano.

**Figura 119 Vista lateral sistema vertical**



De la Figura 119 se obtiene la ecuación 6

$$x_2^2 = 30^2 - (55 - h)^2 \quad ( 7$$

Donde “h” es la altura desde el sustrato de cultivo, hasta la ubicación de las lámparas.

Restando la ecuación 6 de la ecuación 5, y despejando “x1” se obtiene la ecuación, y con esto se sabe que se debe ubicar cada extremo del arnés a una distancia “35cm - x1”, del lado más cercano del nivel, para ubicar la lámpara a una altura “h”.

$$x_1 = \sqrt{30^2 - 18^2 - (55 - h)^2} [cm] = \sqrt{576 - (55 - h)^2} [cm] \quad ( 8$$

Como la hoja del fabricante menciona que se debe ubicar la lámpara a una altura de 40cm para cubrir un área de 100cmx70cm, entonces reemplazando “h” en la ecuación 8 se obtiene:

$$x_1 = \sqrt{576 - (55 - 40)^2} = \sqrt{576 - 225} = \sqrt{351} = 18.73 [cm] \quad ( 9$$

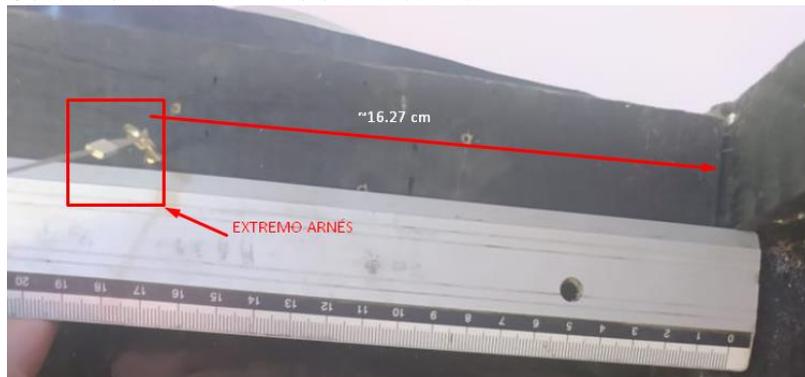
Por lo tanto, se debe ubicar cada extremo del arnés a:

$$35 - x_1 [cm] = 35 - 18.73 [cm] = 16.27 [cm] \quad ( 10$$

Desde los laterales del nivel.

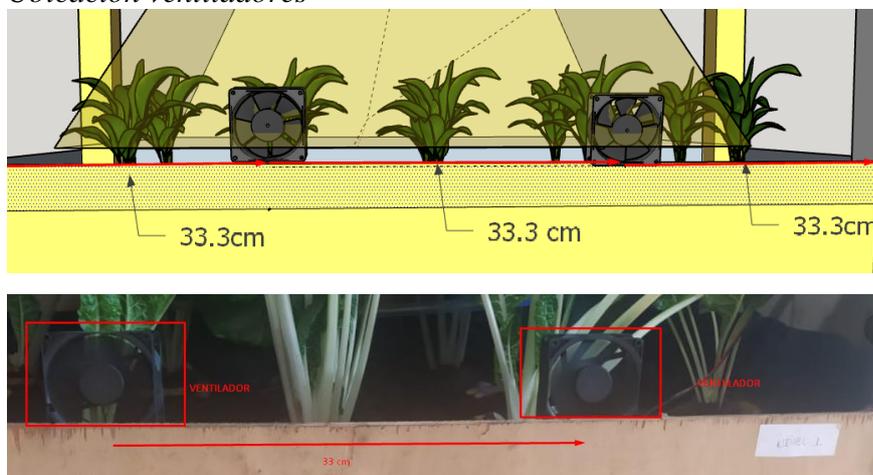
En la Figura 120, se muestra la ubicación de uno de los extremos del arnés en el nivel

1.

**Figura 120***Ubicación arnés en el sistema vertical*

#### 4.2.3.6. Instalación Ventiladores

Como ya se ha mencionado, los ventiladores, cumplirán la función de intentar mantener la temperatura ambiental en los niveles, dentro del rango óptimo de crecimiento de la acelga, basadas en las decisiones tomadas por el Nodo Central. Para ello es importante ubicarlos de manera distribuida, por esto se los ha ubicado de manera equidistante entre sí y entre los laterales del nivel, al lado opuesto de la ubicación del sensor de temperatura, de esta manera se pretende que los ventiladores muevan todo el aire que envuelve las plantas para mantenerlas frescas y también obtener lecturas más acertadas del sensor. En la Figura 121 se muestra como se planifico la instalación de los ventiladores y su ubicación real en el sistema construido.

**Figura 121***Ubicación ventiladores*

#### 4.2.3.7. *Instalación Sistema de Riego*

El sistema de riego es fundamental para el desarrollo de las plantas, por tal razón se ha instalado y elegido un método de riego, de fácil implementación, pero asegurando que cada planta reciba un suministro de agua adecuado. A continuación, se detalla el procedimiento.

Se ha preparado un tramo de manguera de 3 metros, cuyos extremos se conectaron al grifo de agua, el cual se mantiene abierto todo el tiempo, y el otro extremo a un acople de 1/2 pulgada en forma de “T” hembra (observar Figura 122), al conectaron cada una de las electroválvulas. Al otro extremo de cada electroválvula se ha conectado otro tramo de manguera de longitud de 2 metros, cauterizando el extremo final de la misma y haciendo pequeños orificios por los cuales se drena el agua que permite pasar la electroválvula, a presión.

**Figura 122**

*Instalación sistema de riego*



Pensando en la instalación del riego se ha construido las tapas laterales de cada nivel, 10 centímetros mas altas que la tapa frontal, con la finalidad de que sirvan de soporte a la manguera que atraviesa el nivel, por la mitad de este, desde una vista lateral y que parte conectada desde la electroválvula, esto debido a que, si se la atraviesa a ras de suelo, se corre el riesgo de tapar los orificios de drenaje con tierra. Los orificios se han ubicado cada 20 cm en la parte inferior de como se instalo la manguera y se ha sujetado está última con grapas de media pulgada para evitar se desorienten los orificios que idealmente apuntan hacía un surco

que se ha formado con la tierra paralelamente a la manguera de riego, esto se observa en la Figura 123.

### Figura 123

*Ubicación ductos de riego*



### 4.3. Pruebas de funcionamiento

Luego de haber culminado con la implementación del sistema y obtener el diseño final del mismo, cuya estructura de agricultura vertical y la integración de los Nodos Sensores, juntamente con los sensores y actuadores se puede apreciar en la Figura 124, con la finalidad de garantizar el buen funcionamiento de los principales componentes del sistema, en el presente apartado se ha realizado las pruebas de funcionamiento de estos. Para ellos se determina la secuencia de pasos a seguir para realizar la prueba de cada componente y se concluye el buen o mal funcionamiento de estos, a partir de los resultados obtenidos.

### Figura 124

*Sistema Final*



### 4.3.1. Pruebas de funcionamiento Nodo Central

El nodo central cumple con las siguientes funciones:

- Intercambio de información con los Nodos Sensores y con Firebase Realtime
- Gestión de la comunicación a través de MQTT
- Almacenamiento de datos localmente
- Control de flujos del sistema

Por tal motivo a continuación se detallan las pruebas de funcionamiento de cada uno de los componentes que hacen posible que estas tareas se cumplan.

#### 4.3.1.1. Pruebas de funcionamiento de Red

El Nodo Central utiliza la red local, para el intercambio de información con los Nodos Sensores, también es importante que este cuente con salida a Internet, para la comunicación Firebase Realtime. Las acciones realizadas para verificar el funcionamiento de red del Nodo Central han sido las siguientes:

- 1) Verificación de la IP estática asignada a la interfaz inalámbrica en el apartado 3.4.11.
- 2) Verificación de la conexión a Internet.

En la Figura 125 se muestra la verificación del paso 1, en una terminal de línea de comandos del Nodo Central, emitiendo el comando “ifconfig wlan0”, la segunda línea de salida de este comando indica la dirección IP “192.168.88.150”, por lo que dicha verificación ha sido exitosa. Así mismo, en la Figura 126, se muestra la verificación de conexión a Internet a través de la herramienta “ping”, haciendo un test sobre el dominio de Google “www.google.com”, la respuesta es exitosa teniendo una respuesta a un tiempo de aproximadamente 80ms, por lo cual se concluye el correcto funcionamiento de Red en el Nodo Central.

**Figura 125***Verificación dirección IP Nodo Central*

```

pi@raspberrypi:~$ ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.88.150 netmask 255.255.255.0 broadcast 192.168.88.255
    inet6 fe80::1b36:9166:af52:ff2e prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:e9:47:05 txqueuelen 1000 (Ethernet)
    RX packets 105473 bytes 29950746 (28.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 175862 bytes 40966180 (39.0 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

**Figura 126***Verificación de conectividad a Internet Nodo Central*

```

pi@raspberrypi:~$ ping www.google.com
PING www.google.com (142.250.78.132) 56(84) bytes of data:
64 bytes from bog02s18-in-f4.1e100.net (142.250.78.132): icmp_seq=1 ttl=115 time=80.3 ms
64 bytes from bog02s18-in-f4.1e100.net (142.250.78.132): icmp_seq=2 ttl=115 time=80.2 ms
64 bytes from bog02s18-in-f4.1e100.net (142.250.78.132): icmp_seq=3 ttl=115 time=80.2 ms
64 bytes from bog02s18-in-f4.1e100.net (142.250.78.132): icmp_seq=4 ttl=115 time=80.1 ms
64 bytes from bog02s18-in-f4.1e100.net (142.250.78.132): icmp_seq=5 ttl=115 time=80.4 ms

```

#### 4.3.1.2. Pruebas de funcionamiento Broker MQTT

A nivel de aplicación, el Nodo Central, intercambia la información con los Nodos Sensores a través del protocolo MQTT, para esto es fundamental que se cuente con un Broker MQTT, y como se ha mencionado al inicio de este apartado, también este Broker, es un servicio del mismo Nodo Central. Las verificaciones realizadas han sido:

- 1) Suscripción a un topic de prueba
- 2) Publicación de mensajes
- 3) Recepción de mensajes

Para la suscripción de mensajes MQTT a través de topics, se ha utilizado la terminal de línea de comandos del Nodo Central y el comando “mosquitto\_sub -d -t testTopic”, el cuál hace uso de la herramienta “mosquitto\_clients” el parámetro “-t”, permiten especificar el nombre del topic al que se suscribe, en la Figura 127 se puede observar la salida que proporciona este comando, y como se puede observar en la última línea nos muestra el mensaje de “Suscribed”, y la terminal queda a la escucha de los mensajes gracias al parámetro “-d”.

**Figura 127***Prueba de suscripción MQTT*

```

pi@raspberrypi:~ $ mosquitto_sub -d -t testTopic
Client (null) sending CONNECT
Client (null) received CONNACK (0)
Client (null) sending SUBSCRIBE (Mid: 1, Topic: testTopic, QoS: 0, Options: 0x00)
Client (null) received SUBACK
Subscribed (mid: 1): 0

```

Abriendo una nueva pestaña en la terminal de línea de comandos se ha emitido el comando “mosquitto\_pub -d -t testTopic -m ‘Hello World!’”, este comando publica el mensaje “Hello World!” en el topic “testTopic”, la salida de este comando se aprecia en la Figura 128.

**Figura 128***Prueba de publicación MQTT*

```

pi@raspberrypi:~ $ mosquitto_pub -d -t testTopic -m "Hello world!"
Client (null) sending CONNECT
Client (null) received CONNACK (0)
Client (null) sending PUBLISH (d0, q0, r0, m1, 'testTopic', ... (12 bytes))
Client (null) sending DISCONNECT
pi@raspberrypi:~ $

```

La Figura 129 muestra el contenido de la primera pestaña de la terminal de la línea de comandos, después de haber publicado el mensaje de prueba, se puede apreciar dicho mensaje en la última línea, precedido de un mensaje que indica que ha recibido un mensaje del cliente “null”.

**Figura 129***Verificación de recepción MQTT*

```

pi@raspberrypi:~ $ mosquitto_sub -d -t testTopic
Client (null) sending CONNECT
Client (null) received CONNACK (0)
Client (null) sending SUBSCRIBE (Mid: 1, Topic: testTopic, QoS: 0, Options: 0x00)
Client (null) received SUBACK
Subscribed (mid: 1): 0
Client (null) received PUBLISH (d0, q0, r0, m0, 'testTopic', ... (12 bytes))
Hello world!

```

Por tanto, se ha verificado el funcionamiento adecuado del Broker MQTT.

#### 4.3.1.3. Pruebas de funcionamiento de la Base de Datos local

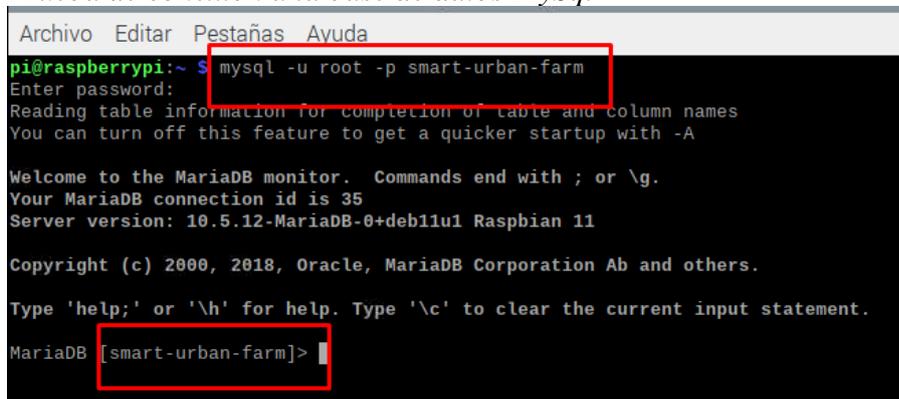
Cuando el nodo central reciba los datos de los Nodos Sensores, los almacenará en la Base de Datos MySQL, para verificar el funcionamiento de este servicio se ha realizado los siguientes pasos:

- 1) Conexión a la base de datos a través del cliente MySQL instalado en el Nodo Central.
- 2) Ejecución de consulta simple, que muestre las tablas de la Base de Datos.

Para la conexión a la base de datos se ha utilizado el comando “mysql -u root -p smart-urban-farm”, la bandera “-u”, permite especificar el usuario con el que se conectará a la base de datos indicada después de la bandera “-p”, luego se proporciona la contraseña ligada al usuario especificado, y se muestra una salida como en la Figura 130.

**Figura 130**

*Prueba de conexión a la base de datos MySQL*



```

Archivo  Editar  Pestañas  Ayuda
pi@raspberrypi:~$ mysql -u root -p smart-urban-farm
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 35
Server version: 10.5.12-MariaDB-0+deb11u1 Raspbian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [smart-urban-farm]>

```

Una vez que MySQL ha aprobado la conexión a la base de datos indicada, se ha emitido la consulta “SHOW TABLES;”, esto muestra las tablas existentes, así como se observa en la Figura 131.

**Figura 131***Prueba de comando SQL*

```
MariaDB [smart-urban-farm]> SHOW TABLES;
+-----+
| Tables_in_smart-urban-farm |
+-----+
| entrenamiento                |
| etiquetas                    |
| historial                   |
| niveles                     |
+-----+
4 rows in set (0.003 sec)
```

Basándose en los resultados, sin haber tenido ningún inconveniente y obteniendo la salida esperada de la consulta, se concluye un correcto funcionamiento de la Base de Datos local.

#### **4.3.1.4. Pruebas de funcionamiento Node-RED**

La lógica de procesos del sistema en general es controlada por el Nodo Central a través de Node-RED, en donde previamente se han definido dichos procesos a través de flujos, por tal motivo, Node-RED se convierte en el núcleo de funcionamiento del sistema, los pasos realizados para verificar el funcionamiento de esta herramienta son:

- 1) Verificación de la comunicación MQTT.
- 2) Verificación de la comunicación con la Base de Datos.
- 3) Verificación del algoritmo de clasificación.
- 4) Verificación de la comunicación con Firebase Realtime,

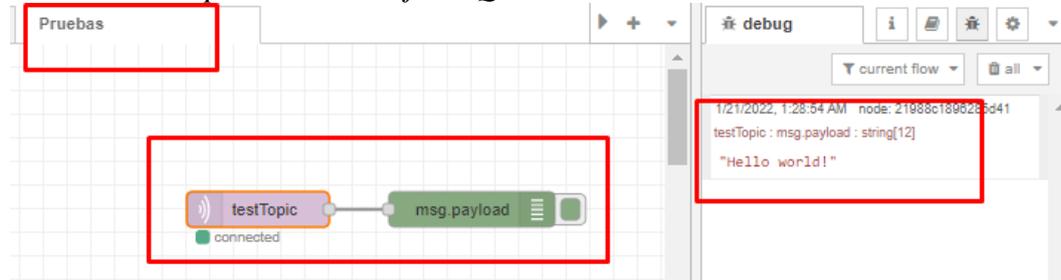
Para la verificar el funcionamiento se ha agregado un nuevo flujo, al que se ha llamado pruebas, a este se le ha agregado simplemente los nodos de interés, conectados a un nodo “debug”, para que permita visualizar los resultados.

En la Figura 132 se ha agregado un nodo MQTT de entrada, el cual se ha suscrito al topic “testTopic”, y un nodo “debug”, interconectados entre sí. Paralelamente se ha emitido el

comando utilizado en el apartado 4.3.1.2, para enviar un mensaje de prueba, y en la ventana de “debug” de Node-RED, se puede observar dicho mensaje impreso.

**Figura 132**

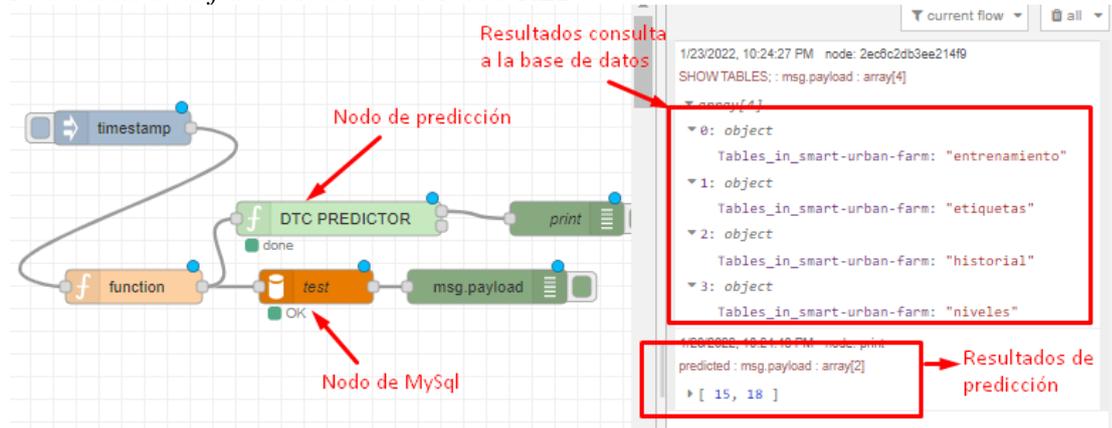
*Prueba de recepción de mensajes MQTT en Node-RED*



Para comprobar la base de datos se agrega un nodo “timestamp”, que ejecute una consulta a la base de datos local, a la vez que coloca un conjunto de datos ambientales de prueba, a la entrada del nodo de clasificación, este flujo y sus resultados se aprecian en la Figura 133.

**Figura 133**

*Prueba de clasificación de datos Node-RED*



#### 4.3.2. Pruebas de funcionamiento del Nodo Sensor

La principal función del Nodo Sensor es la de enviar los datos tomados por los sensores a el Nodo Central y de recibir una respuesta de este último, después de que haya clasificado estos datos, por lo tanto, se puede decir que la principal función de este

componente es la de enviar y recibir datos. El Nodo Sensor envía y recibe esta información a través de la red Wifi y el protocolo MQTT, por este motivo a continuación se han realizado las pruebas de funcionamiento de estos protocolos.

#### 4.3.2.1. Pruebas de funcionamiento conectividad Wifi

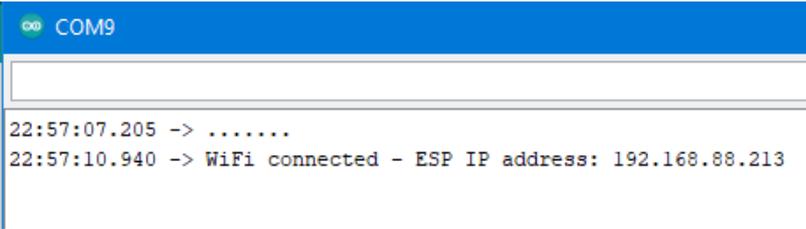
Los pasos seguidos para verificar el funcionamiento de la conectividad Wifi en el Nodo Sensor han sido los siguientes:

- 1) Conexión del Nodo Sensor al IDE de Arduino.
- 2) Imprimir la dirección IP obtenida de conexión a la red Wifi, a través del monitor serial.
- 3) Test de comunicación desde PC.

Al conectar el Nodo Sensor al IDE de Arduino a través del puerto COM asignado por la PC, se puede visualizar, como en la Figura 134, que mientras el Nodo Sensor realiza la conexión con el AP Wifi imprime una secuencia de puntos, una vez que establece el enlace imprime un mensaje con la dirección IP que ha obtenido.

#### Figura 134

##### Prueba de conectividad WiFi Nodo Sensor



```
COM9
22:57:07.205 -> .....
22:57:10.940 -> WiFi connected - ESP IP address: 192.168.88.213
```

Utilizando la herramienta “ping” en una terminal de comandos y haciendo el test a la dirección IP obtenida, se ha obtenido una respuesta exitosa, como se puede observar en la Figura 135.

**Figura 135***Prueba de conectividad de red Nodo Sensor*

```

C:\Users\Admin>ping 192.168.88.213

Haciendo ping a 192.168.88.213 con 32 bytes de datos:
Respuesta desde 192.168.88.213: bytes=32 tiempo=12ms TTL=255
Respuesta desde 192.168.88.213: bytes=32 tiempo=436ms TTL=255
Respuesta desde 192.168.88.213: bytes=32 tiempo=277ms TTL=255
Respuesta desde 192.168.88.213: bytes=32 tiempo=34ms TTL=255

Estadísticas de ping para 192.168.88.213:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
            (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 12ms, Máximo = 436ms, Media = 189ms

C:\Users\Admin>

```

#### 4.3.2.2. Pruebas de funcionamiento de MQTT

Pasos para la verificación del funcionamiento del protocolo MQTT:

- 1) Conexión del Nodo Sensor al IDE de Arduino.
- 2) Imprimir mensaje de conexión, a través del monitor serial.
- 3) Imprimir a través del monitor serial, cuando se reciba un mensaje MQTT.

De la misma manera que en el anterior apartado, se ha impreso en el monitor serial, una vez que se ha establecido la conexión con el Broker MQTT, un mensaje dando aviso de esto, Figura 135.

**Figura 136***Prueba de conectividad MQTT Nodo Sensor*

```

23:04:36.420 -> .....
23:04:40.144 -> wifi connected - ESP IP address: 192.168.88.213
23:04:40.144 -> Attempting MQTT connection...connected

```

Nuevamente en la Figura 137 se ha hecho uso del comando “mosquitto\_pub”, desde la terminal del Nodo Central, y a modo de ejemplo, ¡se ha publicado el mensaje “Hello World!” en el topic “level2/lamp” que corresponde a uno de los topics que está suscrito el Nodo Sensor del nivel 2, y en la Figura 138, se puede apreciar como el IDE de Arduino imprime en el monitor serial dicho mensaje.

**Figura 137***Prueba de envío de mensaje MQTT al Nodo Sensor*

```

pi@raspberrypi:~ $ mosquitto_pub -d -t level2/lamp -m "Hello world!"
Client (null) sending CONNECT
Client (null) received CONNACK (0)
Client (null) sending PUBLISH (d0, q0, r0, m1, 'level2/lamp', ... (12 bytes))
Client (null) sending DISCONNECT
pi@raspberrypi:~ $ █

```

**Figura 138***Prueba de recepción de mensaje MQTT Nodo Sensor*

```

23:04:30.420 > .....
23:04:40.144 -> WiFi connected - ESP IP address: 192.168.88.213
23:04:40.144 -> Attempting MQTT connection - connected
23:22:19.199 -> Ha llegado un mensaje del topic: level2/lamp. Mensaje: Hello world!

```

### 4.3.3. Pruebas de funcionamiento de los sensores

Los sensores toman los valores de las variables ambientales en tiempo real, de tal manera que aportan con la información cuantitativa del estado actual del sistema, para que el Nodo Central tome estos valores y los clasifique de una manera cualitativa, si los valores de los sensores están distorsionados, es muy probable que el Nodo Central solicite acciones incorrectas con respecto a los actuadores, por tal motivo es sumamente importante verificar que los valores que el Nodo de cada nivel lee de los sensores, sea coherente con el estado real del sistema. Para verificar el estado de los sensores se ha realizado los siguientes pasos:

- 1) Puesta en funcionamiento de los Nodos Sensores.
- 2) Envío de datos leídos al Nodo Central.
- 3) Impresión de valores recibidos en el Nodo Central.
- 4) Verificación de valores.

Para facilitar el proceso de verificación, se ha hecho uso del Nodo Central, debido a que con los sensores instalados dentro de la estructura base, se torna un poco complicado la conexión directa del Nodo Sensor a la PC.

En la Figura 139 se muestra un flujo que consta de dos nodos, uno correspondiente a MQTT de suscripción, y a la salida de este un nodo de “debug”, al nodo MQTT se lo ha

suscrito al topic “level2/sensors”, el corresponde al topic sobre el cual el Nodo del nivel 2, enviara la trama de datos de los sensores, y en la parte derecha se muestra la impresión de los valores recibidos separados por el carácter.

**Figura 139**

*Prueba de lectura de sensores desde Node-RED*



#### 4.3.4. Pruebas de funcionamiento de los actuadores

Como ya se ha mencionado en apartados anteriores, los actuadores son parte fundamental del sistema, ya que estos permitirán modificar en cierto grado el estado de las variables ambientales del entorno, tendiendo así a mantener un ambiente automatizado y controlado gracias a la clasificación de los datos del algoritmo de aprendizaje, la verificación del funcionamiento se la ha realizado con los siguientes pasos:

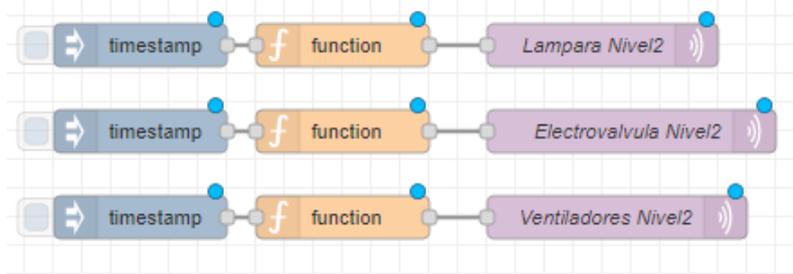
- 1) Puesta en funcionamiento de los Nodos Sensores.
- 2) Envío de activación de lámparas desde el Nodo Central.
- 3) Envío de activación de ventiladores desde el Nodo Central.
- 4) Envío de activación de electroválvula desde el Nodo Central.

Para enviar la orden de activación de cada uno de los actuadores, desde el Nodo Central, se ha definido tres pequeños flujos, los cuales se observan en la Figura 140, que envían cada uno el comando de activación de un actuador y dentro del código de implementación de los Nodos Sensores, se ha suscrito a un topic correspondiente a cada uno

de los actuadores, y cuando reciba un mensaje enviará la señal correspondiente, a sus pines digitales de salida.

**Figura 140**

*Flujo de prueba para activación de actuadores Node-RED*



En las Figuras 141, 142 y 143 se puede apreciar la manguera suspendida en la mitad del nivel, señalado con una flecha el orificio a través del cual se drena el agua una vez activada la electroválvula y el chorro de agua que se desprende en ese momento, así también se aprecian los ventiladores en rotación y las lámparas una vez que han sido encendidas, respectivamente.

**Figura 141**

*Verificación de activación de sistema de riego*



**Figura 142**

*Verificación de activación de sistema de ventilación*



**Figura 143***Verificación de activación de sistema de iluminación*

#### 4.3.5. Pruebas de funcionamiento aplicación móvil

La función principal de la aplicación móvil es la de permitir al usuario final interactuar con el sistema, de tal manera que él pueda leer de manera fácil y sencilla los datos que se están obteniendo de los sensores en tiempo real y también interactuar directamente con los actuadores, en caso de necesitarlo. Se han realizado los siguientes pasos de verificación:

- 1) Puesta en funcionamiento de los Nodos Sensores.
- 2) Puesta en funcionamiento de los flujos de Node-RED
- 3) Navegación entre pantallas.
- 4) Verificación de gráficas y estado actual del sistema.
- 5) Activación/Desactivación remota de los actuadores

En la Figura 144, se muestra una captura de la pantalla de monitoreo de luminosidad, cuyos valores se ha constatado paralelamente con los valores que llegan a Node-RED, obteniendo un resultado satisfactorio. Se puede observar en la parte superior, en color verde el último valor leído, correspondiente al nivel 1 y en color azul el correspondiente al nivel 2, seguidamente se encuentra una gráfica con los últimos valores que han llegado hasta la aplicación, con su respectiva marca de tiempo y por último la descripción de la etiqueta asignada por el algoritmo de aprendizaje.

**Figura 144**

*Verificación de lectura de datos desde la aplicación móvil*



#### 4.4. Análisis de resultados

En el presente apartado se detallan los resultados del análisis de la red Wifi obtenidos del Sistema autónomo de monitoreo y control de condiciones ambientales, los resultados de producción en este ambiente controlado, así como los obtenidos al realizar el cultivo de las mismas hortalizas en un ambiente no controlado, se ha realizado un análisis minucioso a nivel productivo de las plantas, y para concluir se realiza un análisis comparativo entre las dos metodologías, en virtud de cumplir el objetivo final del presente proyecto. Por recomendación del experto en agricultura, el Ing. agrónomo Clemente Alvarado, se ha determinado realizar la cosecha de la acelga mediante la recolección de hojas, para esto se ha realizado mediciones semanales del crecimiento de las hojas y la variación de la frondosidad de las plantas a lo largo del tiempo.

También es importante mencionar que para el día 1, todas las plantas cuentan con un tamaño promedio de 4cm y una frondosidad de 4 a 5, por planta.

En la Tabla 30 se muestra la nomenclatura utilizada para cada zona de cultivo, con las que se trabaja en los siguientes subapartados.

**Tabla 30***Nomenclatura para mención de plantas y ambientes*

<b>Código</b>	<b>Referencia</b>
<b>N0-X</b>	Planta N° X ambiente no controlado
<b>N1-X</b>	Planta N° X ambiente controlado nivel 1
<b>N2-X</b>	Planta N° X ambiente controlado nivel 2

También en la Figura 145, se ilustra una vista superior de uno de los niveles de la estructura vertical, y la numeración utilizada para cada una de las plantas.

**Figura 145***Ilustración de distribución de plantas en cada nivel*

#### 4.4.1. Análisis de la red

Para este análisis, se ha tomado en cuenta el tamaño de las cabeceras de cada uno de los protocolos de red utilizados, esto con la finalidad de calcular el tamaño total de los paquetes y tomando en cuenta los parámetros de capa física, también el tiempo que permanecen en el medio, en este caso el aire. También se ha realizado capturas con un sniffer de red, en este caso Wireshark, el cuál nos ha permitido capturar el tráfico inalámbrico del sistema, configurando el adaptador de red Wifi en modo monitor. A continuación, se detalla el análisis por cada uno de los protocolos utilizados y concluyendo al final con los cálculos correspondientes.

- **Cabeceras MQTT**

Como se ha revisado en el Capítulo II, se sabe que MQTT cuenta de 3 partes fundamentales que componen su cabecera, estas son:

- Cabecera Fija (Fixed Header)
- Cabecera Variable (Variable Header)
- Carga útil (Payload)

La cabecera fija de MQTT puede contener de uno a cuatro bytes, dependiendo del campo “tamaño restante”, la cabecera variable podrá contener dos campos: “Topic Name” y “Packet Identifier”. Y la carga útil transportará el contenido del mensaje que se envía a través de MQTT.

Para el caso de estudio, se ha capturado uno de los mensajes transmitidos por uno de los nodos sensores, el cual contiene los datos adquiridos por los sensores en ese momento. En la Figura 146, se puede apreciar dicho mensaje en la interfaz del sniffer de red, en este se indica las diferentes partes de las cabeceras MQTT que se mencionó anteriormente.

**Figura 146**  
*Cabeceras MQTT - Wireshark*

No.	Time	Source	Destination	Protocol	Length	Info
10117	27.778302	192.168.88.150	192.168.88.212	MQTT	32	Ping Response
10127	39.930070	192.168.88.212	192.168.88.150	MQTT	141	Publish Message [level1/sensors]
10167	40.045450	192.168.88.150	192.168.88.212	MQTT	106	Publish Message [level1/fan]

```

> Frame 10127: 141 bytes on wire (1128 bits), 141 bytes captured (1128 bits)
> IEEE 802.11 Data, Flags: .p.....T
> Logical-Link Control
> Internet Protocol Version 4, Src: 192.168.88.212, Dst: 192.168.88.150
> Transmission Control Protocol, Src Port: 63302, Dst Port: 1883, Seq: 81, Ack: 22, Len: 53
MQ Telemetry Transport Protocol, Publish Message
  Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    0011 .... = Message Type: Publish Message (3)
    .... 0... = DUP Flag: Not set
    .... .00. = QoS Level: At most once delivery (Fire and Forget) (0)
    .... ...0 = Retain: Not set
  Msg Len: 51
  Topic Length: 14
  Topic: level1/sensors
  Message: 32342e32353b39342e30303b313032342e30303b31303234...
  
```

En este caso se tienen los siguientes valores para los datos de MQTT.

**Tabla 31**  
*Cálculo de cabeceras de MQTT*

Sección	Tamaño (Bytes)
<b>Cabecera Fija</b>	4
<b>Cabecera Variable</b>	14
<b>Carga útil</b>	35
<b>Total</b>	<b>53</b>

Como se puede observar tanto en la Figura 146 y Tabla 31, se ha tomado cada uno de los tamaños correspondientes a cada sección de la cabecera MQTT, teniéndose un total para este caso, de 53 bytes.

- **Cabeceras TCP**

Generalmente la cabecera del protocolo TCP, tiene un tamaño de veinte bytes, a no ser que se utilice el campo “OPTIONS”, con el cual puede llegar a un tamaño máximo de sesenta bytes. En la Figura siguiente, se muestra los campos de la cabecera del segmento TCP, para el mismo mensaje anteriormente analizado en MQTT.

**Figura 147**  
*Cabeceras TCP - Wireshark*

No.	Time	Source	Destination	Protocol	Length	Info
8317	27.778562	192.168.88.150	192.168.88.212	MQTT	92	Ping Response
10127	39.930070	192.168.88.212	192.168.88.150	MQTT	141	Publish Message [level1/sensors]
10167	40.045459	192.168.88.150	192.168.88.212	MQTT	106	Publish Message [level1/fan]

Internet Protocol Version 4, Src: 192.168.88.212, Dst: 192.168.88.150	
Transmission Control Protocol, Src Port: 63302, Dst Port: 1883, Seq: 81, Ack: 22, Len: 53	
Source Port:	63302
Destination Port:	1883
[Stream index:	0]
[TCP Segment Len:	53]
Sequence number:	81 (relative sequence number)
Sequence number (raw):	6590
[Next sequence number:	134 (relative sequence number)]
Acknowledgment number:	22 (relative ack number)
Acknowledgment number (raw):	1920634730
0101 .... = Header Length:	20 bytes (5)
Flags:	0x018 (PSH, ACK)
Window size value:	2123
[Calculated window size:	2123]
[Window size scaling factor:	-2 (no window scaling used)]
Checksum:	0xedd4 [unverified]
[Checksum Status:	Unverified]
Urgent pointer:	0
[SEQ/ACK analysis]	
[Timestamps]	
TCP payload (53 bytes)	
[PDU Size:	53]
MQ Telemetry Transport Protocol, Publish Message	
Header: Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)	

De la figura anterior se puede rescatar los puertos de origen y destino, que como en este caso se está enviando los datos MQTT, desde el Nodo Sensor, hasta el Broker, el puerto de destino es el configurado por el protocolo, por defecto, es decir, el puerto “1883”, por su parte el puerto de origen es el “63302”, el cual prácticamente es un aleatorio, seleccionado por el Nodo Sensor. También se puede observar como el campo “Header Length”, nos indica el tamaño de la cabecera TCP, el cual muestra claramente, es de 20 bytes.

En la Tabla 32, se observa el tamaño total del segmento TCP, sumando los datos de MQTT.

**Tabla 32**  
*Cálculo de cabeceras TCP*

Sección	Tamaño (Bytes)
<b>Cabecera TCP</b>	20
<b>Datos MQTT</b>	53
<b>Total</b>	<b>73</b>

- **Cabecera IPV4**

Al igual que en el caso de TCP, generalmente la cabecera IPV4 tiene un tamaño de 20 Bytes, a no ser que se utilicen los campos de “OPTIONS”. En la Figura 148 se muestra los campos de la cabecera IP, en donde se pueden destacar las IP de origen y destino, así como el protocolo de capa superior (en este caso TCP), así como el tamaño total del paquete IP.

**Figura 148**  
*Cabeceras IPV4 - Wireshark*

No.	Time	Source	Destination	Protocol	Length	Info
8517	27.778302	192.168.88.150	192.168.88.212	MQTT	92	Ping Response
10127	39.930070	192.168.88.212	192.168.88.150	MQTT	141	Publish Message [level1/sensors]
10167	40.045450	192.168.88.150	192.168.88.212	MQTT	106	Publish Message [level1/fun]

```

> Frame 10127: 141 bytes on wire (1128 bits), 141 bytes captured (1128 bits)
> IEEE 802.11 Data, Flags: .p....T
> Logical-Link Control
> Internet Protocol Version 4, Src: 192.168.88.212, Dst: 192.168.88.150
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 93
    Identification: 0x0022 (34)
  > Flags: 0x0000
    Fragment offset: 0
    Time to live: 255
    Protocol: TCP (6)
    Header checksum: 0x88bd [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.88.212
    Destination: 192.168.88.150
  > Transmission Control Protocol, Src Port: 63302, Dst Port: 1883, Seq: 81, Ack: 22, Len: 53
  > MQ Telemetry Transport Protocol, Publish Message
  
```

Y a continuación se muestra la Tabla 33, que contiene el tamaño total del paquete IP, sumando la cabecera de IPV4 y el segmento TCP.

**Tabla 33**  
*Cálculo de cabeceras IPV4*

Sección	Tamaño (Bytes)
Cabecera IPV4	20
Segmento TCP	73
<b>Total</b>	<b>93</b>

- **Cabeceras 802.11g**

El protocolo 802.11 define las dos capas mas bajas del modelo OSI, y a su vez divide estas capas en subcapas, para el caso de la capa enlace de datos, define las subcapas: LLC y MAC; en cuanto para la capa física o también conocida como PHY, define: PLCP y PMD.

Para el caso de la capa LLC, 802.11 define una cabecera con un tamaño de 8 bytes, con los campos que permitirán gestionar la lógica de control de acceso al medio, en la Figura 149, se muestran los datos de dicha cabecera.

**Figura 149**  
Cabecera 802.11 LLC - Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
8317	27.778562	192.168.88.150	192.168.88.212	MQTT	92	Ping Response
10127	39.930070	192.168.88.212	192.168.88.150	MQTT	141	Publish Message [level1/sensors]
10167	40.045459	192.168.88.150	192.168.88.212	MQTT	106	Publish Message [level1/fan]

```

> Frame 10127: 141 bytes on wire (1128 bits), 141 bytes captured (1128 bits)
on interface eth0
IEEE 802.11 Data, Flags: .p....T
  Logical-Link Control
    DSAP: SNAP (0xaa)
    SSAP: SNAP (0xaa)
    Control field: U, func=UI (0x03)
      000.00.. = Command: Unnumbered Information (0x00)
      ....11 = Frame type: Unnumbered frame (0x3)
      Organization Code: 00:00:00 (Officially Xerox, but
      Type: IPv4 (0x0800)
    Internet Protocol Version 4, Src: 192.168.88.212, Dst: 192.168.88.150
    Transmission Control Protocol, Src Port: 63302, Dst Port: 1883, Seq: 81, Ack: 22, Len: 53
    MQTT Telemetry Transport Protocol, Publish Message
  
```

**Tabla 34**  
Cálculo de cabeceras 802.11 LLC

Sección	Tamaño (Bytes)
Paquete IP	93
Cabecera LLC	8
<b>Total</b>	<b>101</b>

La siguiente cabecera por analizar es la cabecera MAC de 802.11, la cual como se ha mencionado anteriormente, tiene diferentes campos los cuales se pueden apreciar en la Figura 150.

**Figura 150 Cabecera 802.11 MAC - Wireshark**

```

IEEE 802.11 Data, Flags: .p....T
  Type/Subtype: Data (0x0020)
  > Frame Control Field: 0x0841
    .000 0000 0011 1100 = Duration: 60 microseconds
    Receiver address: Routerbo_80:b5:db (74:4d:28:80:b5:db)
    Transmitter address: Espressi_c7:90:61 (a4:cf:12:c7:90:61)
    Destination address: Raspberr_db:6e:76 (b8:27:eb:db:6e:76)
    Source address: Espressi_c7:90:61 (a4:cf:12:c7:90:61)
    BSS Id: Routerbo_80:b5:db (74:4d:28:80:b5:db)
    STA address: Espressi_c7:90:61 (a4:cf:12:c7:90:61)
    .....0000 = Fragment number: 0
    1010 1011 1001 .... = Sequence number: 2745
  > CCMP parameters ← encriptación
  Logical-Link Control
  Internet Protocol Version 4, Src: 192.168.88.212, Dst: 192.168.88.150
  Transmission Control Protocol, Src Port: 63302, Dst Port: 1883, Seq: 81, Ack: 22, Len: 53
  
```

Es importante mencionar que la cabecera MAC, tiene un tamaño de 30 bytes, no obstante, la suma total entre los datos, y dicha cabecera, suman un total de 145 bytes, esto

debido a la encriptación de los datos, mediante el protocolo 802.11i y el campo FCS, que se agrega al final de la PSDU.

**Tabla 35**  
*Cálculo de cabeceras 802.11 - MAC*

<b>Sección</b>	<b>Tamaño (Bytes)</b>
Datos encriptados	111
Cabecera MAC	30
FCS	4
<b>Total</b>	<b>145</b>

En la Figura 151, se muestra una captura de pantalla de Wireshark, después de haber aplicado un filtro para obtener información de los paquetes capturados del primer nodo, entre los cuales se ha podido encontrar las tramas que contienen información de las cabeceras PLCP. Dentro de esta información se pueden encontrar diferentes parámetros importantes como lo son:

- El protocolo de capa de PHY: 802.11g
- El canal utilizado, juntamente con la frecuencia central: Canal 1 y 2412Mhz
- La tasa de transferencia utilizada: 6Mbps
- La modulación utilizada: OFDM
- La potencia de la señal: -9dbm
- El tamaño de la cabecera: 26 Bytes
- Banderas de información extra

**Figura 151**  
*Cabeceras 802.11 capa PHY - Wireshark*

No.	Time	Source	Destination	Protocol	Length	Info
1395	10.120113	192.168.88.86	192.168.88.150	MQTT	144	Publish Message [level1/actuators/lamp]
1898	11.004586	192.168.88.86	192.168.88.150	MQTT	171	Publish Message [level1/sensors]

> Frame 1898: 171 bytes on wire (1368 bits), 171 bytes captured (1368 bits)  
 > Radiotap Header v0, Length 26  
   Header revision: 0  
   Header pad: 0  
   Header length: 26  
   > Present flags  
     MAC timestamp: 572338777  
   > Flags: 0x10  
     Data Rate: 6.0 Mb/s  
     Channel frequency: 2412 [BG 1]  
     Channel flags: 0x00c0, Orthogonal Frequency-Division Multiplexing (OFDM), 2 GHz spectrum  
     Antenna signal: -9dBm  
     Antenna: 1  
   > RX flags: 0x0000  
   > 802.11 radio information  
     PHY type: 802.11g (ERP) (6)  
     Short preamble: False  
     Proprietary mode: None (0)  
     Data rate: 6.0 Mb/s  
     Channel: 1  
     Frequency: 2412MHz  
     Signal strength (dBm): -9dBm  
     TSF timestamp: 572338777  
   > [Duration: 220µs]  
   > IEEE 802.11 Data, Flags: .p....TC  
   > Logical-Link Control

**Tabla 36**  
*Cálculo de la longitud de la trama*

Sección	Tamaño (Bytes)
PSDU	145
Tail	0.75
Cabecera PLCP	5
<b>Total</b>	<b>150.75</b>

- **Cálculo del tiempo de las tramas en el aire**

Para una tasa de datos de 6Mbps, 802.11g hará uso de BPSK y OFDM, teniéndose una tasa de codificación de 1/2. Se ha tenido un total de 150.75 bytes, lo que equivale a un total de 1206 bits. Sabiendo que cada símbolo OFDM a una tasa de datos de 6Mbps, codifica 24 bits de datos por símbolo, se tendría un total de:

$$\#Simbolos_{OFDM} = 1206/24 \cong 51 \quad ( 11 )$$

Según (Soto, 2011), la duración de cada símbolo OFDM es de 4 $\mu$ s, sabiendo que también que la duración del preámbulo es de 16 $\mu$ s, se puede calcular el tiempo de la trama en el medio (el aire), es:

$$TiempoAire = 4 * 51 + 16 [\mu s] \cong 220[\mu s] \quad ( 12 )$$

Valor que también coincide con los datos que Wireshark muestra en la Figura 152.

**Figura 152**

*Duración de la trama - Wireshark*

No.	Time	Source	Destination	Protocol	Length	Info
1395	10.120113	192.168.88.86	192.168.88.150	MQTT	144	Publish Message [level1/actuators]
1898	11.004586	192.168.88.86	192.168.88.150	MQTT	171	Publish Message [level1/sensors]

Antenna: 1  
 RX flags: 0x0000  
 ... .. = Bad PLCP: False  
 802.11 radio information  
 PHY type: 802.11g (ERP) (6)  
 Short preamble: False  
 Proprietary mode: None (0)  
 Data rate: 6.0 Mb/s  
 Channel: 1  
 Frequency: 2412MHz  
 Signal strength (dBm): -9dBm  
 TSF timestamp: 572338777  
 [Duration: 220µs]  
 [Preamble: 20µs]  
 [IFS: 3546µs]  
 [Start: 57233857µs]

- **Cálculo del Throughput**

Se entiende que el Throughput, no es mas que la relación entre los datos útiles y el tiempo de transmisión, para el caso de estudio, los datos intercambiados, son relativamente cortos en comparación con las cabeceras que se insertan para poder hacer posible la comunicación, sin embargo, es de mucho valor tener en cuenta dicho parámetro como una buena práctica en el análisis de redes.

Anteriormente se ha calculado, el tiempo que utiliza uno de los Nodos Sensores, al enviar los datos tomados del ambiente controlado, hasta el Nodo Central, se ha tomado este

ejemplo, dado que es la mayor cantidad de información que se transmite de un Nodo, hacia el otro. Teniendo en cuenta que el tiempo de transmisión ha sido, de 220us y se considera carga útil al campo “Payload” y “Topic Name”, del protocolo MQTT, se tiene una carga útil con un total de:

$$CargaUtil = Payload + Topic = 35 + 14 [bytes] = 49 [bytes] \quad ( 13 )$$

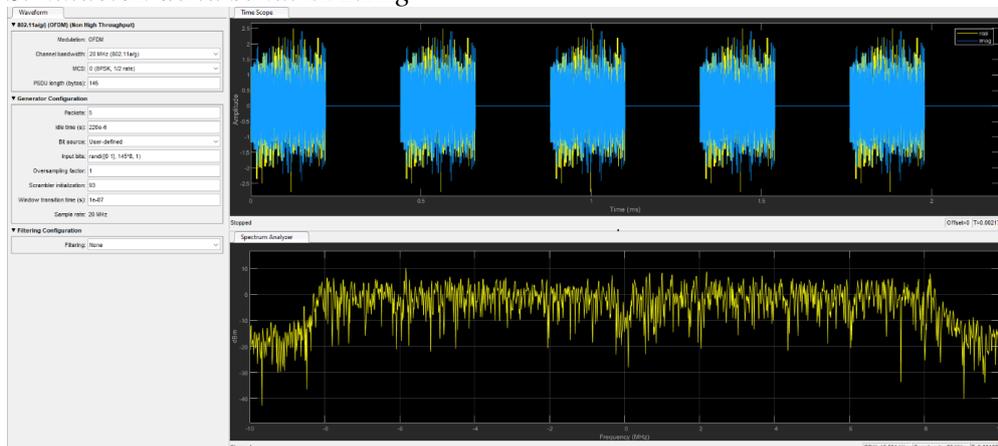
Por lo tanto, el valor del Throughput, para este caso, será de:

$$Throughput = 49[bytes] * 8[bits/byte]/(220[us]) \cong 1.78[Mbps] \quad ( 14 )$$

- **Verificación en el software de simulación MATLAB**

Con la ayuda del software de simulación, se ha verificado el cálculo previamente realizado, en el cual se ha determinado que el tiempo que ocupan las tramas en el aire, llevando la información de uno de los Nodos Sensores, hacia el Nodo Central, a través del protocolo 802.11g con una tasa de transmisión de 6Mbps, es de aproximadamente 220us, la herramienta utilizada para dicha simulación dentro de Matlab, ha sido “wlan waveform generator”, a la cuál se le ha introducido los parámetros mencionados, esto se puede apreciar en la Figura 153.

**Figura 153**  
*Simulación de la señal 802.11g*



#### 4.4.2. Análisis de resultados en el ambiente no controlado

Para lograr resultados aceptables en el ambiente no controlado, se ha tenido que esperar un lapso de aproximadamente 70 días de crecimiento de los cultivos y poder cosecharlos para el consumo. En la Figura 152 se muestra una fotografía del tamaño de una de las hojas del cultivo en el ambiente no controlado, después de aproximadamente 9 semanas de crecimiento, del cual se ha obtenido una medición de aproximadamente 23cm de largo y 13 cm de ancho.

**Figura 154**

*Medición longitud de la hoja en el ambiente no controlado*



**Figura 155**

*Plantas en el ambiente no controlado*



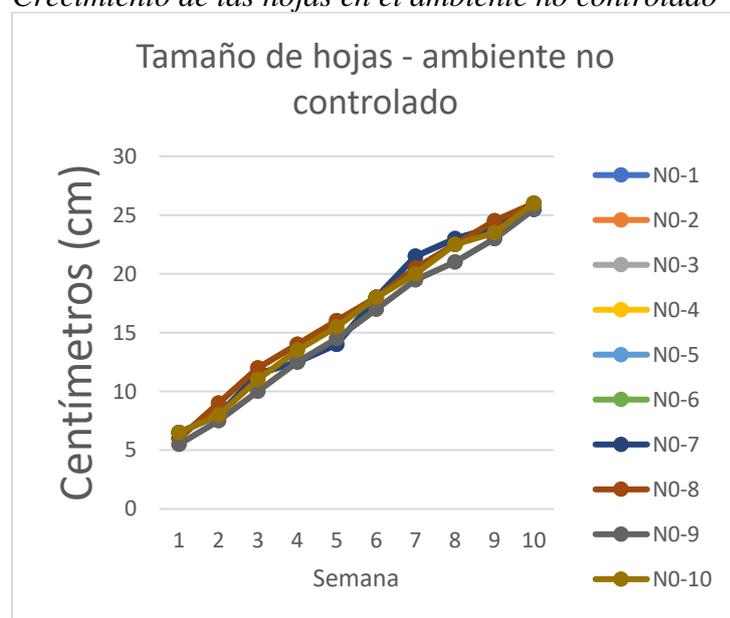
De la anterior Figura se puede apreciar que el producto crece en buenas condiciones, sus hojas tienen un color verde oscuro que dan la sensación de frescura del vegetal, además de pequeñas ondulaciones características de la acelga.

#### 4.4.2.1. Tamaño

En la Figura 156, se muestra el gráfico resultante de la tabulación de los resultados obtenidos de las mediciones semanales del tamaño de las hojas de las plantas cultivadas a través de una metodología tradicional, es decir un ambiente no controlado, el eje horizontal se tienen enumeradas del 1 al 10, las semanas transcurridas durante el análisis de crecimiento, por tanto, en el eje vertical se tiene la longitud alcanzada, en centímetros, por las hojas de las plantas, correspondiente a cada semana.

**Figura 156**

*Crecimiento de las hojas en el ambiente no controlado*



En el lado derecho del gráfico anterior se puede observar las etiquetas de cada una de las plantas cultivadas en el ambiente no controlado, cada una con el color que la representa en la gráfica de crecimiento, de esta última se puede observar a simple vista se puede observar un crecimiento uniforme entre las plantas, además de que tiene una tendencia lineal.

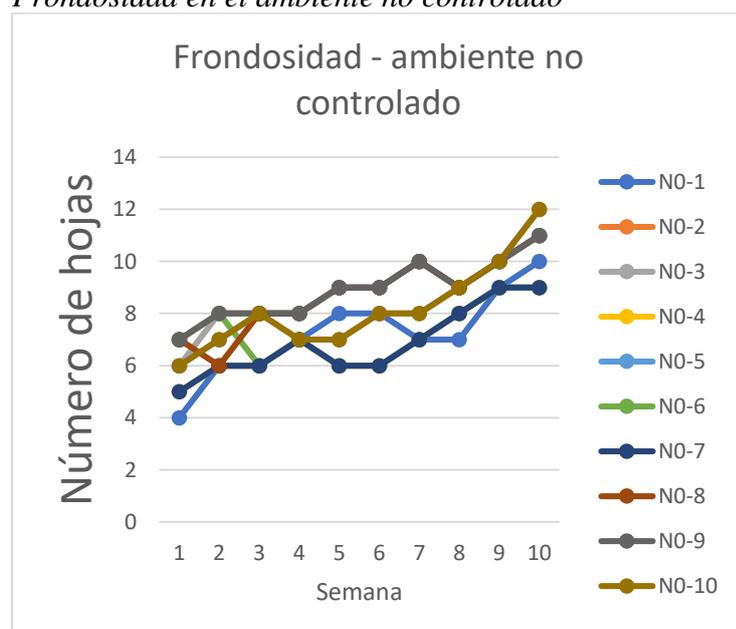
#### 4.4.2.2. Frondosidad

De la misma manera que en el análisis del tamaño de las hojas de las plantas, se ha verificado semanalmente, el número de hojas que brotan de las mismas, a este parámetro se

lo conoce como frondosidad y en el eje vertical de la Figura 157 se muestra esta variación con respecto al tiempo, en semanas, por cada una de las plantas. Debido a que, durante el transcurso de las diez semanas, no se ha tenido cosechas intermedias, la variación de la frondosidad no es muy pronunciada, siendo las bajas notables de este parámetro, debido a la muerte de algunas hojas, ocasionada por las condiciones climáticas a las que se expone el cultivo, haciendo énfasis en el exceso de luminosidad y las altas temperaturas, la humedad del suelo también es afectada a causa de estas dos anteriores, sin embargo es un parámetro que es posible regular manualmente, con riegos más frecuentes para mantener esta variable en un estado aceptable para el desarrollo del cultivo.

**Figura 157**

*Frondosidad en el ambiente no controlado*



En la Figura 158, se muestra la fotografía de una de las plantas del ambiente no controlado, después de haber recibido varias horas sol directo, acompañadas de una alta temperatura ambiental (aproximadamente 28°C). Claramente se puede observar la debilidad en sus hojas al colgar de sus tallos, a diferencia de como se las muestra en la Figura 156 y 158 en condiciones menos adversas.

**Figura 158**

*Debilidad de las hojas debido a la temperatura del ambiente no controlado*

**4.4.3. Análisis de resultados en el ambiente controlado**

A continuación, se documentan los resultados obtenidos dentro del ambiente controlado, tomando en cuenta las mismas variables que en el subapartado anterior, además de la nomenclatura mencionada, en cuanto a los niveles y la ubicación de cada planta. Como primera observación a estos resultados, en la Figura 159, se muestra una de las hojas ya cosechadas después de las primeras 6 semanas de haber sembrado las plántulas dentro del sistema, la cual tiene una longitud de aproximadamente 28 y 15 centímetros, de largo y ancho respectivamente.

**Figura 159**

*Tamaño de una de las hojas del ambiente controlado*

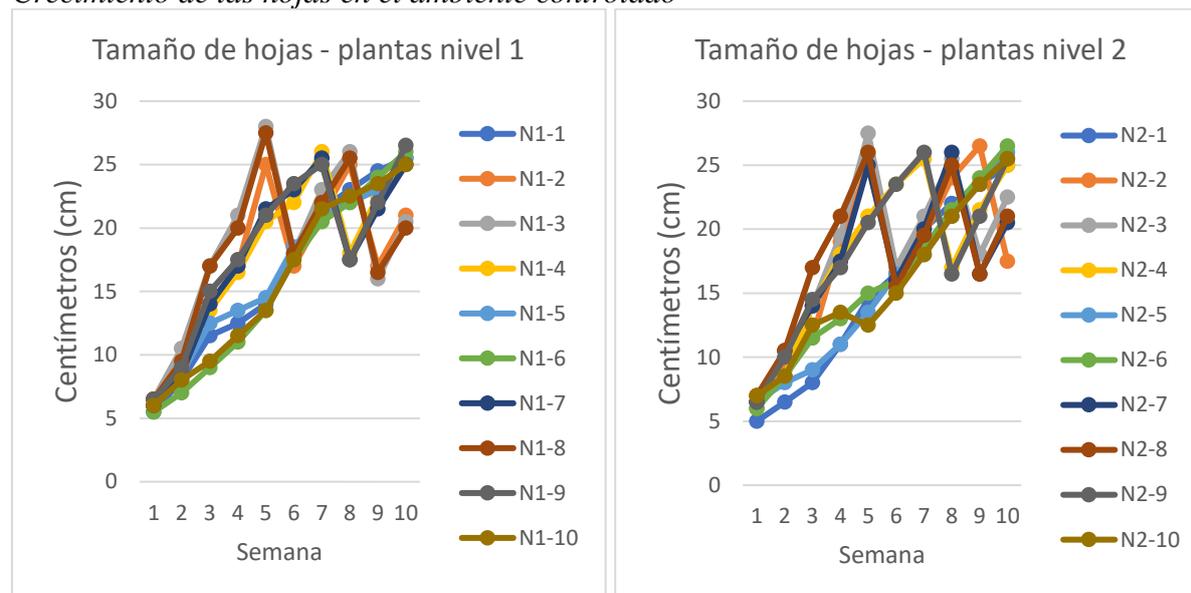


#### 4.4.3.1. Tamaño

La Figura 160, contiene los datos que, corresponde a una zona de cultivo diferente, en el lado izquierdo se tiene la gráfica de crecimiento de las plantas del nivel 1, de la estructura vertical y en el lado derecho, la del nivel 2.

**Figura 160**

*Crecimiento de las hojas en el ambiente controlado*



Se puede observar en cada uno de los gráficos una baja abrupta del tamaño de las hojas, una vez que este a superado los 25cm, esto se debe a que, en este punto, se ha realizado la recolección de hojas para el consumo, también se puede apreciar que las plantas que se encuentran en las esquinas, cuya numeración es: 1,5,6 y 10; tienen un tiempo de desarrollo mucho más extendido que las plantas más centrales: 2,3,4,7,8 y 9; esto se debe a que existe mayor concentración de luminosidad en la zona central, que en las zonas laterales, además que debido a esta misma razón, la diferencia de iluminación entre estas zonas es cada vez más amplia, ya que las plantas centrales empiezan a hacer un efecto de visera, sobre las demás, sin embargo el cosechar las hojas de las plantas centrales apenas estas lleguen al punto de madurez mencionado, se favorece a las zonas laterales permitiendo pasar mayor

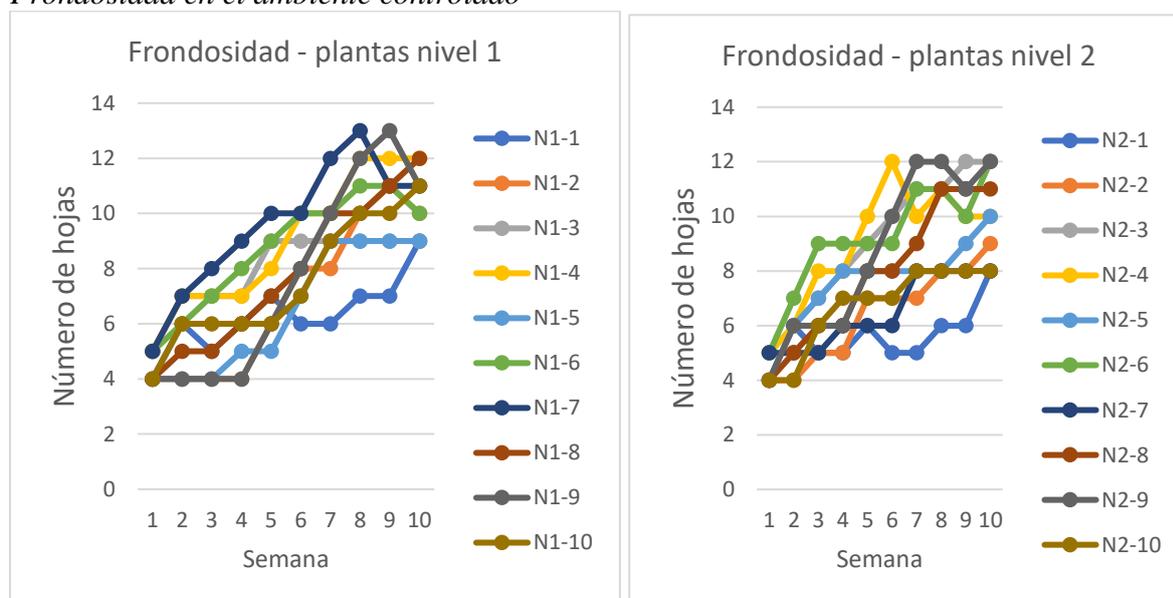
luminosidad y por ende provocando un mejor proceso fotosintético, lo cual se traduce en un mejor crecimiento de la planta.

#### 4.4.3.2. Frondosidad

La frondosidad de las plantas en el ambiente controlado se muestra en los gráficos de la Figura 161, se pueden observar variaciones bastante pronunciadas y esto se debe a que al igual que en N0-X, mueren y se cosechan hojas, reduciendo el número de hojas, pero a la vez se tiene el aumento de estas y se debe al nacimiento de nuevas hojas en el transcurso del tiempo.

**Figura 161**

*Frondosidad en el ambiente controlado*



**Nota:** En el anexo 5 se encuentran las tabulaciones de las mediciones semanales realizadas de cada una de las plantas.

#### 4.4.4. Análisis comparativo

Para realizar la comparativa de producción obtenida de los dos ambientes, se toma en cuenta las variables: crecimiento del tamaño y frondosidad; realizando un cálculo promedio con el fin de obtener un estimado de estas variables, pero en unidades de densidad, es decir

crecimiento y frondosidad por cada planta; esto debido a que la cantidad de los cultivos ha sido mayor en el caso del ambiente controlado.

En la Tabla 37 se muestra la producción total por cada ambiente, y se calcula el promedio de producción por cada planta, durante un periodo de 10 semanas.

**Tabla 37**  
*Producción total en gramos de cada subsistema*

<b>Ambiente</b>	<b>Producción Total (gr)</b>	<b>Número de plantas</b>	<b>Promedio</b>	<b>Porcentaje diferencial</b>
<b>N0</b>	725	10	72.5	0%
<b>N1</b>	1420	10	142	95.86%
<b>N2</b>	1380	10	138	90.34%

Teniéndose los valores de la Tabla 38, en donde se ha realizado un cálculo promedio de la frondosidad de cada ambiente, hasta la primera cosecha, que para el caso de los ambientes controlados es aproximadamente 6 semanas y en el ambiente no controlado 10 semanas. Adicionalmente se calcula el porcentaje de diferencia entre la frondosidad del ambiente no controlado y los ambientes controlados.

**Tabla 38**  
*Frondosidad promedio de cada subsistema*

<b>Ambiente</b>	<b>Frondosidad Promedio</b>	<b>Porcentaje diferencial</b>
<b>N0</b>	7.825	0%
<b>N1</b>	8.3	6.07%
<b>N2</b>	8.2	4.79%

De la Tabla anterior se puede concluir que se mantiene un mayor número de hojas durante la primera cosecha en el ambiente controlado, que en el ambiente no controlado.

La Tabla 39 muestra el tiempo de cosecha promedio, calculado entre todas las plantas de cada ambiente, y así como en el caso anterior se calcula el porcentaje diferencial entre los ambientes controlados con respecto al ambiente controlado.

**Tabla 39**  
*Tiempo de cosecha promedio por cada subsistema*

<b>Ambiente</b>	<b>Tiempo Promedio</b>	<b>Porcentaje diferencial</b>
<b>N0</b>	10	0%
<b>N1</b>	8	20%
<b>N2</b>	8.3	17%

De esto se puede concluir que el ambiente controlado demandó de un promedio mínimo del 17% menos de tiempo para su primera cosecha, que en el caso del ambiente no controlado.

#### **4.5. Costos del sistema**

A continuación, se menciona los costos de implementación del proyecto, así como también se realiza cálculos estimados del costo de producción en ambientes tradicionales o no controlados, concluyendo con los beneficios que brinda la implementación del Sistema autónomo de monitoreo y control de condiciones ambientales en cultivos verticales hortícolas urbanos mediante una red de sensores.

- **Costos de implementación del proyecto**

En la Tabla 40, se muestra el costo de cada uno de los subsistemas contemplados para la implementación y desarrollo del proyecto, y al final de esta se muestra la suma total de estos, lo que viene a representar el costo de implementación total de este proyecto.

**Tabla 40**  
*Resumen de costos de implementación del sistema*

<b>Item</b>	<b>Valor</b>
Costo Estructura de Agricultura Vertical	59.5
Costo Nodo Sensor	53.78
Costo Nodo Central	119.5
Costo Sensores y Actuadores	177.2
Costo Sistema de Monitoreo	0
Costo Sistema Eléctrico	31.25
<b>Total</b>	<b>441.23</b>

**Nota:** El detalle de costos de cada uno de los subsistemas mencionados en la Tabla 40, se presenta en el anexo 8.

A continuación, en la Tabla 41 se muestra un resumen de costos en cuanto a la implementación de un sistema de agricultura tradicional, es decir un ambiente no controlado, para lo cual se ha tomado en cuenta el uso de insecticidas debido a que un ambiente abierto, existen ciertas plagas que atacan a la planta.

**Tabla 41**

*Resumen de costos de implementación de un sistema tradicional*

<b>Item</b>	<b>Valor</b>
Costo Estructura de Agricultura Vertical	59.5
Insecticida	12.00
<b>Total</b>	<b>71.50</b>

- **Costo por consumo eléctrico**

Un parámetro importante para tomar en cuenta es el consumo energético del sistema, el cual, debido al uso de lámparas de iluminación artificial, se vuelve relativamente alto, a pesar de que la tecnología utilizada por estas optimiza dicho consumo.

Tomando en cuenta el valor actual del kilovatio hora en Ecuador, el cual es de USD 0,1047 centavos por kilovatio hora (el Universo, 2021) , se ha realizado la siguiente Tabla, en donde se detalla el consumo de potencia por cada uno de los componentes del sistema electrónico, la cantidad de estos, y el tiempo estimado que estos se mantienen encendidos durante un día completo. Con estos datos se ha calculado el costo mensual generado por cada uno de los componentes, y al final de la Tabla, se tiene la suma total de estos, la cual representa el valor de consumo eléctrico por mes, generado por todo el sistema, el cual ha resultado un valor aproximado de USD 4.95.

**Tabla 42**  
*Costos por consumo eléctrico*

Item	Voltaje	Corriente (mA)	Cantidad	Potencia (W)	Tiempo/día (horas)	Costo mes
<b>Electroválvula</b>	110	30	2	6.6	0.000115	0.00
<b>Lámparas</b>	110	430	2	94.6	12	3.57
<b>Ventiladores</b>	5	200	4	4	24	0.30
<b>Modulo Rele</b>	5	90	2	0.9	24	0.07
<b>Node Mcu</b>	5	80	2	0.8	24	0.06
<b>Sensor Ldr</b>	5	2	2	0.02	24	0.00
<b>Sensor FC28</b>	5	35	4	0.7	24	0.05
<b>Sensor DHT11</b>	5	0.5	2	0.005	24	0.00
<b>RaspBerry Pi</b>	5	1200	2	12	24	0.90
<b>Total</b>						4.95

- **Producción**

En el mercado nacional, el precio de la acelga es fluctuante, por tal motivo se ha tomado un precio referencial de acuerdo con el precio actual del producto hortícola en los supermercados nacionales, que al momento oscila entre 0.75 y 0.90 centavos de dólar por cada 400 gramos del producto, por lo que haciendo un cálculo simple se obtiene, que la producción económica obtenida por el ambiente controlado, durante las 10 semanas de estudio es de USD 6.30, Ecuación 14 y 15.

$$pac = (produccio\_total[gr] * precio[USD])/400[gr] \quad ( 15)$$

$$pac = (2800[gr] * 0.90[USD])/400[gr] = 6.30 [USD] \quad ( 16)$$

Mientras que la producción del ambiente no controlado ha sido de USD 1.63, Ecuación 16.

$$pac = (725[gr] * 0.90[USD])/400[gr] = 1.63 [USD] \quad ( 17)$$

#### 4.6. Discusión de resultados

En este apartado se pretende consolidar acerca de los resultados obtenidos durante el desarrollo de la investigación, así como volver a hacer un énfasis de los objetivos y enfoque del proyecto, para esto se ha tomado las diferentes variables que afectan positiva o negativamente la implementación de este.

Como primer punto, según los datos obtenidos en el apartado 4.4.4, es bastante notoria la superioridad de producción al implementar sistemas inteligentes y automatizados en cultivos hortícolas de ciclo corto, lo que ha permitido dotar al cultivo objeto de este estudio, de un ambiente dentro de los parámetros óptimos, y los resultados respaldan dicha afirmación, como se observó:

- a) Se ha obtenido para el nivel 1, una producción del 90.34% superior al cultivo tradicional y para el nivel 2 una producción del 95.86 % superior al mismo.
- b) una frondosidad, en el nivel 1, del 4.79% superior al cultivo tradicional y un 6.07 % en el nivel 2, superior al cultivo tradicional.

- c) y un tiempo de producción de para el nivel 1 del 17% y para el nivel 2 del 20 % menos, que el cultivo tradicional.

También se puede apreciar en el anexo 7 de este documento la coloración y aspecto de las plantas, las cuales dentro del sistema autónomo se aprecian mucho más frescas y saludables, gracias al ambiente controlado.

Sin embargo, dado que el enfoque de este proyecto ha sido principalmente a la implementación de huertos urbanos en interiores (dentro de domicilios), no se ha dimensionado los gastos de implementación de todo el proyecto, en un enfoque comercial. Es por esta misma razón que los requerimientos en cuanto al uso de tecnologías de comunicación para los componentes del sistema, han sido las tecnologías que la mayoría de las personas, en la actualidad, tienen en casa, como lo es 802.11. Si se quisiera implementar esta investigación en un enfoque comercial, muchos de los requerimientos utilizados no serían los mismos, sobre todo en el área de comunicaciones ya que existen tecnologías y soluciones que se adaptarían muchísimo mejor para estos entornos, que permitirían optimizar los recursos de implementación.

Dado que las tecnologías utilizadas para la implementación requieren un alto consumo energético, en especial la iluminación artificial, teniendo muy en cuenta que los cultivos necesitan una iluminación continua y prolongada, los costos de consumo eléctrico son relativamente altos con respecto a la agricultura tradicional. No obstante, es muy posible realizar mejoras para optimizar estos aspectos, los cuales se detallan en el apartado de Trabajos futuros.

Por último, se ha realizado una lista de ventajas y desventajas de la implementación del sistema en cuanto a la producción del producto hortícola seleccionado para este estudio, el cual ha sido “acelga beta vulgaris”.

**Ventajas**

1. Menor tiempo de espera hasta la primera cosecha.
2. Hojas más robustas y frescas durante el desarrollo del cultivo.
3. Baja intervención manual.
4. Mayor cantidad de hojas durante el desarrollo de la planta.
5. No hay necesidad del uso de químicos. Productos 100% orgánicos.
6. Monitoreo y control continuo de los cultivos de forma remota.
7. Aprovechamiento del espacio superficial

**Desventajas**

1. Mayor costo de implementación.
2. Alto consumo de energía.
3. Dependencia de la red de energía eléctrica.
4. Posible desconexión o fallo de los nodos.

## TRABAJOS FUTUROS

- El principal aspecto en el cual, se considera trabajar, es en la optimización de recursos para la implementación de esta solución, esta investigación se ha centrado específicamente en obtener una mejor producción de los cultivos, sin embargo, ciertos recursos, como energéticos y económicos, hacen que de momento esta solución, sea costosa de implementar. Una mejora se podría obtener al implementar esta solución bajo un esquema de alimentación autosustentable.
- Es importante dotar de un sustrato o una solución con los nutrientes necesarios para un óptimo crecimiento de cualquier cultivo, por lo que un aspecto importante a tomar en cuenta para mejorar el presente estudio podría ser la automatización de proporción de nutrientes a través del sistema de riego en un sistema hidropónico, lo cual a su vez también optimizaría el uso de espacio, como de recursos hídricos.
- Realizar estudio en un enfoque comercial, con este o diferentes cultivos hortícolas, utilizando otras tecnologías que se adapten de mejor manera especialmente a la extensión de terreno que se puede dar en un ambiente de este tipo, en el cual también sería factible implementar un sistema de iluminación híbrida entre la luz natural y la luz artificial, de acorde a las necesidades de las plantas.

## CONCLUSIONES

- Se ha seleccionado como caso de estudio al cultivo “acelga beta vulgaris”, dado su corto periodo de desarrollo y resistencia a condiciones adversas, el cual ha respondido de manera positiva, dando resultados de desarrollo excelentes, dentro del sistema autónomo implementado.
- Las herramientas y componentes adquiridos se han evaluado realizando un Benchmark, acorde a la oferta del mercado nacional, así como a los requerimientos establecidos, los cuales se han adaptado correctamente a las necesidades y condiciones del entorno.
- Se ha diseñado e implementado, un sistema embebido que ha permitido la toma de datos de las condiciones ambientales, en un sistema de agricultura vertical, y que, a su vez con la ayuda de actuadores, modifica las condiciones ambientales de estudio, cuando así lo determina el Nodo de procesamiento central, dada la clasificación del estado del sistema por el algoritmo de aprendizaje utilizado.
- Se ha diseñado un enfoque multicriterio para la selección del algoritmo de aprendizaje a utilizar, basado en el desempeño de cada uno, para lo cual se ha tomado en cuenta diferentes métricas típicamente utilizadas para la evaluación de algoritmos en Machine Learning, esto entre dos muy comunes y de bajo coste computacional, teniendo como resultado la elección del algoritmo de Árboles de decisión.
- Se ha realizado un análisis comparativo de los cultivos producidos, por el sistema autónomo desarrollado y los producidos por un sistema de agricultura convencional, de donde se concluye la superioridad de producción del cultivo en el ambiente controlado, se ha realizado una discusión de dichos resultados, destacando las ventajas y desventajas de la implementación de este proyecto.

## RECOMENDACIONES

- Se debe proporcionar a los cultivos de un sustrato rico en nutrientes para que puedan desarrollarse de una manera adecuada, para medir esto se lo puede realizar a través de un sensor NPK o en su defecto un sensor de conductividad eléctrica, realizando una mezcla dos a uno, entre agua de baja conductividad y el sustrato utilizado, si el sustrato no cuenta con los nutrientes necesarios, se debe dotar de abono a la tierra, de preferencia orgánico.
- El Nodo Sensor, hace uso de relevadores para activar los actuadores del sistema, y así modificar el estado de las variables ambientales, es importante tomar en cuenta los posibles fallos de los sensores de humedad del suelo, así como también realizar las conexiones adecuadamente, para evitar que el sistema de riego inunde de agua a los cultivos.
- Después que se ha instalado los Nodos Sensores dentro del sistema vertical, es algo complicado manipular o volver a programar los nodos, por lo que se recomienda agregarles la funcionalidad, de ser reprogramados remotamente a través de la red Wifi a través de la librería “ArduinoOTA”.
- Al desarrollar mejoras en el prototipo, o nuevos proyectos basados en este estudio, se debe tomar en cuenta los gastos económicos que implican y procurar optimizar recursos antes de su implementación, para que así se genere tecnología y proyectos más asequibles para la comunidad.

## Referencias

- Alaniz, D., & Cruz, J. (2018). *Evaluación del patrón de humedecimiento del agua en el sistema de riego por micro aspersión en la finca experimental el Plantel (UNA). 2016.* (Tesis de Grado). Universidad Nacional Agraria .
- Alarcón, A., Arias, G., Díaz, C., & Sotto, J. (2017). Diseño de un sistema de control y automatización de temperatura, humedad del suelo y humedad relativa para optimizar el rendimiento de cultivos bajo cubierta en CORHUILA. *4to. Congreso Internacional AMITIC*, 48–53.
- Alonso, J., Ferrari, H., Sambrana, I., & Irrazábal, E. (2017). Desarrollo de un Planificador Embebido para la Gestión de Riego Automático aplicado a invernáculos. Implementación en INTA El Sombrero . *XXIII Congreso Argentino de Ciencias de La Computación* , 944–953.
- altercmdweb. (2018, April 28). *IPv4*. <https://pc-solucion.es/2018/04/28/ipv4/>
- Álvarez, A., Montoya, J., Arrieta, C., Pinzón, D., & Reyes, D. (2018). Sistema de riego autónomo con capacidad de medición de variables ambientales usando software libre . *Encuentro Nacional de Educación En La Ingeniería* .
- Alvarez, D. (2019). *Estudio comparativo teórico-práctico entre los estándares Wlan 802.11N y 802.11AC.* (Tesis de Grado). Universidad de las Fuerzas Armadas.
- Aristizabal, J. (2017). *Evaluación de un método de aprendizaje supervisado para la detección de las enfermedades, antracnosis y phytophthora Infestans en cultivos de fruta de Risaralda.* (Tesis de Grado). Universidad Tecnológica de Pereira .
- Ascencios, D. (2020). Calibración, validación y automatización del sistema de riego por goteo subterráneo usando un microcontrolador arduino. *Revista de Investigaciones*

*Altoandinas - Journal of High Andean Research*, 22(1), 95–105.

<https://doi.org/10.18271/ria.2020.540>

Ascencios, D. (2021). *Eficiencia hídrica, energética y emisiones CO2 utilizando sistemas de riego presurizado en las áreas verdes de la UNALM*. (Tesis de Posgrado). Universidad Nacional Agraria La Molina.

Bacópulos, E. (2001). *Olericultura* .

Bagnato, J. (2018). *Árbol de Decisión en Python: Clasificación y predicción*. APRENDE MACHINE LEARNING. <https://www.aprendemachinlearning.com/arbol-de-decision-en-python-clasificacion-y-prediccion/>

Bardi, U. (2021). *Los límites del crecimiento retomados* (Los Libros De La Catarata, Ed.).

Benavides, O., Barraza, F., & Navia, J. (2017). Efecto del riego por goteo y exudación sobre el rendimiento de hortalizas en clima frío. *Revista de Ciencias Agrícolas*, 34(1), 108. <https://doi.org/10.22267/rcia.173401.67>

Botella, A., & Ramos, P. (2019). Investigación-acción y aprendizaje basado en proyectos. Una revisión bibliográfica . *Perfiles Educativos* , 41(163), 127–141.

Boumediene, S. (2017). *Soluciones para la infiltración y la redistribución en el riego por superficie: aproximaciones simplificadas y modelizadas*. (Tesis de Grado). Universidad Politécnica de Madrid .

Brenes, A. (2020). *Predicción del caudal promedio horario de la estación hidrológica palmar, utilizando modelos de machine learning basados en árboles de decisión* . (Tesis de Grado). Universidad de Costa Rica .

- Calderón, S., & Sánchez, E. (2016). *La gestión de stakeholders en proyectos. Identificación y evaluación de los stakeholders clave en un proyecto ecoturístico en la laguna de Huamanpata- Región Amazonas.*
- Carballo, O. (2021). Dinámicas espaciales de la horticultura en los oasis norte y centro de Mendoza, Argentina. *Anales de Geografía de La Universidad Complutense*, 41(1), 39–58. <https://doi.org/https://dx.doi.org/10.5209/aguc.76721>
- Carpio, D. (2018). *Diseño e implementación de un control PID con ganancias programadas utilizando un sistema embebido Launchpad MPS 430 para planta balón y viga .* (Tesis de Grado). Universidad Politécnica Salesiana sede Guayaquil .
- Chávez, C., & Fuentes, C. (2018). Optimización del riego por surcos mediante una fórmula analítica y su impacto en la reducción del agua aplicada. *Agrociencia*, 52(4), 483–496.
- Chávez, C., & López, J. (2019). Diseño y evaluación del riego por gravedad en los distritos de riego 011, 023 y 085. *COMEII*.
- Ciampa, M. (2013). *CWNA Guide to Wireless LANs* (3rd ed.).
- Colina, A. (2019). *Ingeniería de Software I*.
- Cortes, V., & Vargas, M. (2020). *Diseño e implementación de un sistema de riego automatizado y monitoreo de variables ambientales mediante IOT en los cultivos urbanos de la fundación mujeres empresarias Marie Poussedin.* (Tesis de Grado). Universidad Católica de Colombia .
- Costa, T. (2015). *Uso de estiércol caprino y bocashi en el cultivo de acelga (Beta vulgaris var. cicla Pers). en el colegio de Bachillerato Puyango de la parroquia Alamor.* (Tesina). Universidad de Loja.

Cubides, J., & Pulido, C. (2016). 4. *MQTT — documentación de mediaLab - 1*.

<https://docsmedialab.readthedocs.io/es/latest/quickReference/mqtt.html>

De Grazia, J., Tittonell, P., & Chiesa, Á. (2001). Efectos de la época de siembra, radiación y nutrición nitrogenada sobre el patrón de crecimiento y el rendimiento del cultivo de lechuga (*Lactuca sativa* L.). *Agr.: Prod. Prot. Veg*, 16(3), 355–365.

Dinculeană, D., & Cheng, X. (2019). Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices. *Applied Sciences*, 9(5), 848.

<https://doi.org/10.3390/app9050848>

el Universo. (2021). *Ecuador, sexto en ranking de tarifas eléctricas residenciales, por debajo de Colombia y Perú | Economía | Noticias | El Universo*.

<https://www.eluniverso.com/noticias/economia/tarifas-electricas-de-ecuador-son-competitivas-segun-colegio-de-ingenieros-electricos-y-la-arc-nota/>

Electronics. (2021). *Ventilador 5v 4010 40x40x10mm Nuevo Electronics | MercadoLibre*.

[https://articulo.mercadolibre.com.ec/MEC-514010791-ventilador-5v-4010-40x40x10mm-nuevo-electronics-\\_JM#position=17&search\\_layout=stack&type=item&tracking\\_id=01d6e0fb-beb9-4f1f-b418-2da8e12033a2](https://articulo.mercadolibre.com.ec/MEC-514010791-ventilador-5v-4010-40x40x10mm-nuevo-electronics-_JM#position=17&search_layout=stack&type=item&tracking_id=01d6e0fb-beb9-4f1f-b418-2da8e12033a2)

Fernández, L. (2020). *Protocolos de redes: la guía completa con todos los protocolos básicos*. <https://www.redeszone.net/tutoriales/internet/protocolos-basicos-redes/>

García, E., & Flego, F. (2018). Agricultura de Precisión. *Tecnología Agropecuaria* .

García, J., Zambrano, J., Alcivar, R., & Zambrano, W. (2020). Predicción del rendimiento de cultivos agrícolas usando aprendizaje automático. *Revista Arbitrada Interdisciplinaria Koinonía*, 5(2), 144. <https://doi.org/10.35381/r.k.v5i2.1013>

- Garnica, L. (2016). *¿Cómo elegir el sistema de riego más adecuado?* Portal Frutícola .  
<https://www.portalfruticola.com/noticias/2016/07/23/como-elegir-el-sistema-de-riego-mas-adeecuado/>
- Gomes, J., Gomes, R., & Souza, A. (2019). The multifunctionality of urban horticulture and its integration with the city ecosystem: a brief review of concepts and the case of São Luís. *Horticultura Brasileira*, 37(3), 252–259. <https://doi.org/10.1590/s0102-053620190301>
- González, C. (2019). *Evaluación de un nuevo sensor de humedad de suelo inalámbrico de bajo costo*. (Trabajo de Grado). Universidad de Talca .
- Google Developers. (2020, February 10). *Classification: Accuracy | Machine Learning Crash Course | Google Developers*. [https://developers.google.com/machine-learning/crash-course/classification/accuracy?hl=es\\_419](https://developers.google.com/machine-learning/crash-course/classification/accuracy?hl=es_419)
- Guevara, G., Verdesoto, A., & Castro, N. (2020). Metodología de investigación educativa (descriptivas, experimentales, participativas, y de investigación-acción). *Revista Científica Mundo de La Investigación y El Conocimiento*, 4(3), 163–173.
- Haro, P. (2012a). *Diagnóstico de cultivo hortícolas que realizan las personas con capacidades especiales en el cantón Alausí, provincia de Chimborazo*. (Tesis de Grado). Universidad Estatal de Bolívar .
- Haro, P. (2012b). *Diagnóstico de cultivos hortícolas que realizan las personas con capacidades especiales en el cantón Alausí, provincia de Chimborazo*. (Tesis de Grado). Universidad Estatal de Bolívar.

- Hernández, Á. (2019). *Sistema de riego automatizado por micro aspersión para el cultivo de cacao en el Municipio de Cabuyaro, Meta*. (Tesis de Grado). Universidad Distrital Francisco José de Caldas .
- Hernández, R. (2021). La agricultura de precisión. Una necesidad actual. *Revista Ingeniería Agrícola*, 11(1), 67–74. <https://doi.org/10.17013/risti.24.39-56>
- Hernández, Y., Rivas, R., & Feliu, V. (2020). Control automático de la distribución de agua en sistemas de riego: revisión y retos . *Revista de Ingeniería Electrónica, Automática y Comunicaciones* , 41(2), 80–97.
- Hidalgo, G., Heredia, S., & Yanez, P. (2018). Estudio de aceptabilidad de acelgas rojas y amarillas (*Beta vulgaris* variedad cicla) en preparaciones de autor en Ecuador. *Espíritu Emprendedor TES*, 2(1), 53–66. <https://doi.org/10.33970/eetes.v2.n1.2018.37>
- jecrespom. (2018). *MQTT / Aprendiendo Arduino*.  
<https://aprendiendoarduino.wordpress.com/2018/11/19/mqtt/>
- Khorov, E., Levitsky, I., & Akyildiz, I. (2020). Current Status and Directions of IEEE 802.11be, the Future Wi-Fi 7. *IEEE Access*, 8.
- Llamas, L. (2019). *¿Qué es MQTT? Su importancia como protocolo IoT*. Ingeniería, Informática y Diseño. <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>
- López, L., Monterroso, A., & Gómez, J. (2020). Diseño de calibración para cuantificar emisiones de Co2 (respiración del suelo) durante intervalos diurnos. *Agrociencias* , 54(6), 731–745.
- López, R., Jiménez, J., Hernández, L., & Parieti, M. (2015). *Riego por melgas-surcos para incrementar el uso eficiente del agua y la productividad del arroz*.

<https://docplayer.es/80204383-Riego-por-melgas-surcos-para-incrementar-el-uso-eficiente-del-agua-y-la-productividad-del-arroz.html>

- Madrigal, O., Brown, O., Pelier, L., Gallardo, Y., & Castellanos, L. (2019). Análisis del consumo de agua y energía en el riego por surcos. *Revista de Ciencia y Tecnología: RECyT*, 32(1), 16–21.
- Magrama. (2013). *Resultados técnico-económicos de Cultivos Hortícolas 2011*. Subdirección General de Análisis, Prospectiva y Coordinación, Subsecretaría. Ministerio de Agricultura, Alimentación y Medio Ambiente.
- Manrique, J., & Sanabria, N. (2020). Planeación y administración en los cultivos hortícolas (2013-2018). *Revista Estrategia Organizacional*, 9(2).  
<https://doi.org/10.22490/25392786.4043>
- Mañueco, M. (2020). Evaluación del riego por superficie mediante el uso de sensores en el Alto Valle del Río Negro y Neuquén. In *XII Congreso de AgroInformática (CAI 2020)- JAIIO 49 (Modalidad Virtual)*.
- Marmolejo, R. (2017). *Microcontrolador - qué es y para que sirve* . HETPRO. <https://hetpro-store.com/TUTORIALES/microcontrolador/>
- Martínez, H. (2018). *Optimización del recurso hídrico en el proceso de riego de un cultivo hidropónico de fresa en la parroquia Huachi del Cantón Ambato en base a sistemas embebidos open source*. (Tesis de Grado). Escuela Superior Politécnica de Chimborazo.
- Mezquita, Y., Alonso, R., Casado, R., Prieto, J., & Corchado, J. (2021). *A Review of k-NN Algorithm Based on Classical and Quantum Machine Learning* (pp. 189–198).  
[https://doi.org/10.1007/978-3-030-53829-3\\_20](https://doi.org/10.1007/978-3-030-53829-3_20)

MLee. (2021, February 22). *Visual Guide to the Confusion Matrix*.

<https://towardsdatascience.com/visual-guide-to-the-confusion-matrix-bb63730c8eba>

Molina, U. (2000). *Programa de hortalizas* . Lamolina.Edu.

[http://www.lamolina.edu.pe/hortalizas/Publicaciones/Datos básicos/3-p1 a p14 \(de acelga a apio\).pdf](http://www.lamolina.edu.pe/hortalizas/Publicaciones/Datos_basicos/3-p1_a_p14_(de_aceelga_a_apio).pdf)

Molpeceres, C., Spontón, E., & Livraghi, E. (2019). Urbanización y horticultura en la Patagonia Austral. Dinámica de las transformación en la ciudad de Ushuaia. *Revista de Estudios Marítimos y Sociales*, 14, 76–101.

Narkhede, S. (2018, May 9). *Understanding Confusion Matrix*.

<https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>

OASIS. (2019). *MQTT Version 5.0*. <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>

Pacheco, M. (2017). *Desarrollo de sensores fosforescentes en estado líquido a temperatura ambiente. Aplicación al análisis de cumarinas*. (Tesis Doctoral). Universidad Nacional de la Plata.

Pazos, M., Gutiérrez, X., & Díaz, O. (2019). El diámetro económico y su uso óptimo para un sistema de riego por aspersión . *Opuntia Brava*, 11(4), 332–338.

Pérez, R., & Jiménez, E. (2020). Evaluación de modelos de infiltración en riego por surcos en suelos oscuros plásticos. *Ingeniería Hidráulica y Ambiental* , 41(1), 43–55.

Pichucho, J. (2007). *Módulo para verificar el funcionamiento de los programas grabados en el PIC 16F84A*. (Tesis de Tecnología). Escuela Politécnica Nacional de Tecnólogos .

Pinos, V. (2021). *Predicción del valor de concentración letal media, LC50 y del nivel de toxicidad de compuestos orgánicos para Daphnia Mgna usando algoritmos de*

- aprendizaje automático supervisado* . (Tesis de Grado). Universidad Abierta de Cataluña.
- Puppo, L., Aguerre, M., Camio, G., Hayashi, R., & Morales, P. (2018). Evaluación del riego por melgas en los suelos del sur del Uruguay. Uso del modelo WinSRFR, resultados preliminares. *Agrociencia Uruguay*, 22(2). <https://doi.org/10.31285/AGRO.22.2.12>
- Quinde, P., & Zhagui, J. (2019). *Diseño de un sistema de riego presurizado para el sector Pungoloma de la comunidad de Quilloac del cantón Cañar provincia de Cañar* . (Tesis de Grado). Universidad del Azuay.
- Quiroga, E., Jaramillo, S., Campo, W., & Chanchi, G. (2016). Propuesta de una arquitectura para agricultura de precisión soportada en I o T. *Revista Ibérica de Sistemas y Tecnología de Información*, 24(10), 39–56. <https://doi.org/10.17013/risti.24.39-56>
- Repetto, J. (2020). *La pandemia impulsa cambios en la horticultura*. Sobre La Tierra. <http://sobrelatierra.agro.uba.ar/la-pandemia-impulsa-cambios-en-la-horticultura/>
- Salazar, R. (2019). *Diseño de un sistema de riego por aspersión aplicado al modelo productivo de agrícola industrial Sukia S.A.* . (Tesis de Grado). Instituto Tecnológico de Costa.
- Sánchez, B. (2019). *Sistema embebido para la gestión energética basado en tecnología IoT y servicios cloud*. (Tesis de Grado). Universidad de Alicante.
- Sánchez, D. (2017). *Caracterización de la sustentabilidad, eficiencia energética y rentabilidad económica de los sistemas de producción hortícola de la parroquia San Joaquín-Azuay-Ecuador*. (Trabajo de Grado). Universidad de Cuenca .

- Santacruz, L., Rosero, G., Sierra, M., & Cristancho, J. (2019). Optimización del riego por superficie para incrementar la productividad del cultivo de palma de aceite en los Llanos Orientales. *Revista Palmas*, 40(3), 11–19.
- Senplades. (2012). *Normas para la inclusión de programas y proyectos en los planes de inversión pública*. SENPLADES. [https://www.obraspublicas.gob.ec/wp-content/uploads/downloads/2012/09/2012\\_Planificacion\\_Guias\\_Metodologicas\\_inclusion\\_Programas\\_Proyectos-en\\_Planes\\_Inversion-Publica1.pdf](https://www.obraspublicas.gob.ec/wp-content/uploads/downloads/2012/09/2012_Planificacion_Guias_Metodologicas_inclusion_Programas_Proyectos-en_Planes_Inversion-Publica1.pdf)
- Serrano, J. (1999). Los sistemas de riego y el desarrollo agrícola en Aguascalientes durante el siglo XIX. *Revista Semestral de Ciencias Sociales y Humanidades* , 5, 81–107.
- Sole, R. (2021). *Raspberry Pi: Crea proyectos DIY por muy poco dinero* . Profesional Review. <https://www.profesionalreview.com/2021/07/18/que-es-raspberry-pi/>
- Soria, L. (2018). *El futuro de la alimentación se llama agricultura vertical* . Compromiso Empresarial. La Revista En Innovación Social . <https://www.compromisoempresarial.com/rsc/2018/08/el-futuro-de-la-alimentacion-se-llama-agricultura-vertical/>
- Soto, O. (2011). *Comparación de la eficiencia volumétrica entre redes inalámbricas WiFi y WiMAX*.
- Tanenbaum, A., & Wetherall, D. (2012). *Redes de computadoras* (5ta ed.).
- Tekmikro. (2021). *(1) Valvulas Electricas Electrovalvulas 12v 110v 220v Agua Aire* / MercadoLibre. [https://articulo.mercadolibre.com.ec/MEC-512720543-valvulas-electricas-electrovalvulas-12v-110v-220v-agua-aire-\\_JM#position=2&search\\_layout=stack&type=item&tracking\\_id=ff78457e-0821-4ff0-9581-16cb8bfe164d](https://articulo.mercadolibre.com.ec/MEC-512720543-valvulas-electricas-electrovalvulas-12v-110v-220v-agua-aire-_JM#position=2&search_layout=stack&type=item&tracking_id=ff78457e-0821-4ff0-9581-16cb8bfe164d)

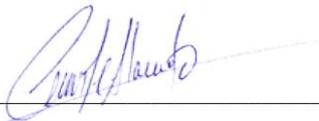
- Tian, H. (2020). *Sinjialight-luz Led De Espectro Completo Para Cultivo De Plantas De Interior,300w - Buy Full Spectrum Led Grow Light,300w Led Grow Light,Warm Led Grow Light Product on Alibaba.com*. <https://spanish.alibaba.com/product-detail/SINJIAlight-300W-Warm-Full-Spectrum-Led-1600202822981.html>
- Torres, C. (2019). *ESP8266 . Micro Crontroladores .*  
<https://microcontroladores.com/esp8266/>
- Valdez, J., & Villanueva, J. (2020). *Sistema automático de riego basado en sistemas embebidos de bajo costo. 1.*
- Velasquez, R. (2016). *Diseño de un sistema Embebido para el control de un invernadero.* (Tesis de Grado). Universidad Mayor de San Andrés .
- Villarreal, S. (2020). *¿Qué es el microcontrolador nodeMCU? - Código IoT. CODIGOIoT.*  
<https://www.codigoiot.com/que-es-el-microcontrolador-nodemcu/>
- Zúñiga, P. (2014). *Manual de producción orgánica de semillas de hortalizas.* (Tesis de Gado). Universidad Central del Ecuador .

## ANEXOS

### Anexo 1: Validación de Requerimientos

<b>P</b> RESENTACIÓN DE REQUERIMIENTOS A LA MESA DE TRABAJO	
<p>ELEMENTOS PARA TENER EN CUENTA EN LA PREPARACIÓN DE REQUERIMIENTOS PARA EL PROYECTO DE TITULACIÓN:</p> <ul style="list-style-type: none"> <li>• Para garantizar la más amplia participación en las conversaciones entre los involucrados en el desarrollo del proyecto “SISTEMA AUTÓNOMO DE MONITOREO Y CONTROL DE CONDICIONES AMBIENTALES EN CULTIVOS VERTICALES HORTÍCOLAS URBANOS MEDIANTE UNA RED DE SENSORES” el Sr. Cristian Ruiz e Ing. Suarez Luis, presentan las propuestas de los requerimientos del sistema a la Mesa de trabajo por medio electrónico, para la terminación de conflictos y la construcción de los objetivos planteados.</li> <li>• Las propuestas deberán enfocarse en los ejes temáticos definidos para la Mesa de Trabajo que son:             <ol style="list-style-type: none"> <li>1. Requerimientos operacionales y de usuario,</li> <li>2. Requerimientos del sistema y</li> <li>3. Requerimientos de arquitectura del sistema; que deberán brindar soluciones para el óptimo funcionamiento del proyecto.</li> </ol> </li> <li>• Las siguientes preguntas pueden orientar al momento de preparar los requerimientos:             <ol style="list-style-type: none"> <li>a) ¿Qué requerimientos operacionales necesita para el sistema?</li> <li>b) ¿Qué requerimientos necesita el usuario para el sistema?</li> <li>c) ¿Qué requerimientos de interfaz se necesita para el sistema?</li> <li>d) ¿Qué requerimientos de uso se necesita para el sistema?</li> <li>e) ¿Qué requerimientos de estado se necesita para el sistema?</li> <li>f) ¿Qué requerimientos físicos se necesita para el sistema?</li> <li>g) ¿Qué requerimientos de seguridad se necesita para el sistema?</li> <li>h) ¿Qué requerimientos de software se necesita para el sistema?</li> <li>i) ¿Qué requerimientos de hardware se necesita para el sistema?</li> <li>j) ¿Qué requerimientos de eléctricos se necesita para el sistema?</li> <li>k) ¿Qué requerimientos de diseño se necesita para el sistema?</li> </ol> </li> </ul>	
Tema	<p>“SISTEMA AUTÓNOMO DE MONITOREO Y CONTROL DE CONDICIONES AMBIENTALES EN CULTIVOS VERTICALES HORTÍCOLAS URBANOS MEDIANTE UNA RED DE SENSORES”</p>

Datos de los involucrados	<p>Ing. Luis Suarez (Tutor de tesis)</p> <p>Ing. Clemnte Alvarado (Ing. Agrónomo experto en horticultura)</p> <p>Sr. Ronni Andrade (Tesista)</p>
¿Qué requerimientos operacionales necesita para el sistema?	<ul style="list-style-type: none"> <li>• El sistema debe implementarse en un cultivo vertical</li> <li>• El entorno del sistema debe albergar actuadores para regular variables ambientales.</li> <li>• El sistema debe ser eficiente energéticamente</li> <li>• El sistema debe recolectar datos de las variables ambientales</li> <li>• El nodo central debe contar con conexión a Internet</li> <li>• El sistema de iluminación debe contar con una PPFD de al menos <math>600 \mu\text{mol} / \text{m}^2 / \text{s}</math></li> <li>• El espectro del sistema de iluminación debe contener las componentes de luz fundamentales para el crecimiento de la planta.</li> <li>• El sistema de riego debe alcanzar a cada una de las plantas.</li> </ul>
¿Qué requerimientos necesita el usuario para el sistema?	<ul style="list-style-type: none"> <li>• Monitoreo de datos del sistema</li> <li>• Presentación clara y precisa de los datos y de fácil interacción</li> <li>• La implementación del sistema debe ser económicamente eficiente</li> <li>• Los dispositivos de hardware deben ser de fácil adquisición</li> </ul>
¿Qué requerimientos de interfaz se necesita para el sistema?	<ul style="list-style-type: none"> <li>• Se debe garantizar la integración de todos los sensores y actuadores a los nodos.</li> <li>• El Microcontrolador debe ser capaz de interpretar los datos tomados a través de los sensores.</li> <li>• Los componentes de acople de los sensores y actuadores deben ser compatibles con las características del microcontrolador.</li> </ul>
¿Qué requerimientos de uso se necesita para el sistema?	<ul style="list-style-type: none"> <li>• Botones on/off y reset de los nodos</li> <li>• El sistema debe precautelar la desactivación de todos los actuadores, en caso de daños en los nodos que lo controlan.</li> </ul>
¿Qué requerimientos físicos se necesita para el sistema?	<ul style="list-style-type: none"> <li>• Los nodos deben tener un tamaño reducido</li> <li>• Los nodos deben ser impermeabilizados</li> <li>• Los sensores deberán ser ubicados en lugares estratégicos</li> </ul>
¿Qué requerimientos de software se necesita para el sistema?	<ul style="list-style-type: none"> <li>• Lenguajes de programación a utilizar deben ser de alto nivel</li> <li>• Las librerías de los sensores y/o plugins de los sensores deben ser compatibles con los microcontroladores</li> <li>• Aplicación de monitoreo, de fácil uso e implementación</li> </ul>

<p>¿Qué requerimientos de hardware se necesita para el sistema?</p>	<ul style="list-style-type: none"> <li>• Los nodos deben contar con una capacidad de procesamiento adecuado, de acuerdo con las tareas a realizar</li> <li>• Módulo para conexión inalámbrica integrado en la placa de desarrollo</li> <li>• Sensor capaz de detectar la luminosidad.</li> <li>• Sensor capaz de detectar la temperatura y humedad relativa del ambiente.</li> <li>• Sensor capaz de detectar el nivel de humedad del suelo.</li> <li>• Actuador capaz de proporcionar iluminación adecuada a los cultivos.</li> <li>• Actuador capaz de habilitar un riego controlado.</li> <li>• Actuador capaz de reducir la excesiva temperatura ambiental.</li> </ul>
<p>¿Qué requerimientos de eléctricos se necesita para el sistema?</p>	<ul style="list-style-type: none"> <li>• Se debe contar con toma eléctrica de 110 AC y fuentes de alimentación de 5V DC.</li> </ul>
<p>¿Qué requerimientos de diseño se necesita para el sistema?</p>	<ul style="list-style-type: none"> <li>• Para la red de sensores se debe diseñar una caja que permita la protección ante agentes de climáticos.</li> </ul>
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  <hr style="width: 100px; margin: 5px auto;"/> <p>Ing. Luis Suarez (Tutor de tesis)</p> </div> <div style="text-align: center;">  <hr style="width: 100px; margin: 5px auto;"/> <p>Ing. Clemente Alvarado (Experto en horticultura)</p> </div> </div> <div style="text-align: center; margin-top: 20px;">  <hr style="width: 100px; border-top: 1px dotted black; margin: 5px auto;"/> <p>Sr. Cristian Ruiz (Tesista)</p> </div>	

## Anexo 2: Especificaciones técnicas del NodeMCU ESP8266

### Figura 162

#### Características del NodeMCU ESP8266

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- WiFi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4s guard interval
- Deep sleep power <10uA, Power down leakage current < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20 dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C
- FCC, CE, TELEC, WiFi Alliance, and SRRC certified

### Figura 163

#### Consumo de energía NodeMCU ESP8266

Parameters	Min	Typical	Max	Unit
Tx802.11b, CCK 11Mbps, P OUT=+17dBm		170		mA
Tx 802.11g, OFDM 54Mbps, P OUT =+15dBm		140		mA
Tx 802.11n, MCS7, P OUT =+13dBm		120		mA
Rx 802.11b, 1024 bytes packet length, -80dBm		50		mA
Rx 802.11g, 1024 bytes packet length, -70dBm		56		mA
Rx 802.11n, 1024 bytes packet length, -65dBm		56		mA
Modem-Sleep①		15		mA
Light-Sleep②		0.9		mA
Deep-Sleep③		10		uA
Power Off		0.5		uA

**Figura 164**  
Definición de pines NodeMCU ESP8266

Pin	Name	Type	Function
1	VDDA	P	Analog Power 3.0–3.6V
2	LNA	I/O	RF Antenna Interface. Chip Output Impedance=50Ω No matching required but we recommend that the n-type matching network is retained.
3	VDD3P3	P	Amplifier Power 3.0–3.6V
4	VDD3P3	P	Amplifier Power 3.0–3.6V
5	VDD_RTC	P	NC (1.1V)
6	TOUT	I	ADC Pin (note: an internal pin of the chip) can be used to check the power voltage of VDD3P3 (Pin 3 and Pin4) or the input voltage of TOUT (Pin 6). These two functions cannot be used simultaneously.
7	CHIP_EN	I	Chip Enable. High: On, chip works properly; Low: Off, small current
8	XPD_DCDC	I/O	Deep-Sleep Wakeup; GPIO16
9	MTMS	I/O	GPIO14; HSPI_CLK
10	MTDI	I/O	GPIO12; HSPI_MISO
11	VDDPST	P	Digital/IO Power Supply (1.8V–3.3V)
12	MTCK	I/O	GPIO13; HSPI_MOSI; UART0_CTS
13	MTDO	I/O	GPIO15; HSPI_CS; UART0_RTS
14	GPIO2	I/O	UART Tx during flash programming; GPIO2
15	GPIO0	I/O	GPIO0; SPI_CS2
16	GPIO4	I/O	GPIO4
17	VDDPST	P	Digital/IO Power Supply (1.8V–3.3V)
18	SDIO_DATA_2	I/O	Connect to SD_D2 (Series R: 200Ω); SPIHD; HSPIHD; GPIO9
19	SDIO_DATA_3	I/O	Connect to SD_D3 (Series R: 200Ω); SPIWP; HSPIWP; GPIO10
20	SDIO_CMD	I/O	Connect to SD_CMD (Series R: 200Ω); SPI_CS0; GPIO11
21	SDIO_CLK	I/O	Connect to SD_CLK (Series R: 200Ω); SPI_CLK; GPIO6
22	SDIO_DATA_0	I/O	Connect to SD_D0 (Series R: 200Ω); SPI_MSIO; GPIO7
23	SDIO_DATA_1	I/O	Connect to SD_D1 (Series R: 200Ω); SPI_MOSI; GPIO8
24	GPIO5	I/O	GPIO5
25	U0RXD	I/O	UART Rx during flash programming; GPIO3
26	U0TXD	I/O	UART Tx during flash programming; GPIO1; SPI_CS1
27	XTAL_OUT	I/O	Connect to crystal oscillator output, can be used to provide BT clock input
28	XTAL_IN	I/O	Connect to crystal oscillator input
29	VDDD	P	Analog Power 3.0V–3.6V
30	VDDA	P	Analog Power 3.0V–3.6V
31	RES12K	I	Serial connection with a 12 kΩ resistor and connect to the ground
32	EXT_RSTB	I	External reset signal (Low voltage level: Active)

**Figura 165**  
*Parámetros NodeMCU ESP8266*

Categories	Items	Values
<b>WiFi Parameters</b>	Certificates	FCC/CE/TELEC/SRRC
	WiFi Protocols	802.11 b/g/n
	Frequency Range	2.4G-2.5G (2400M-2483.5M)
	Tx Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
		802.11 g: -75 dbm (54 Mbps)
802.11 n: -72 dbm (MCS7)		
Types of Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip	
<b>Hardware Parameters</b>	Peripheral Bus	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/PWM
	Operating Voltage	3.0-3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40°-125°
	Ambient Temperature Range	Normal temperature
	Package Size	5x5mm
	External Interface	N/A
<b>Software Parameters</b>	WiFi mode	station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP

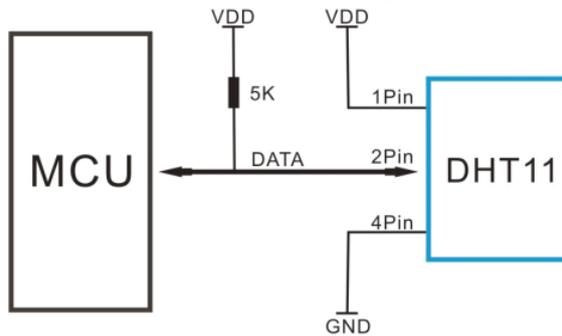
## Anexo 2: Especificaciones técnicas Sensor DHT11

**Figura 166**

*Especificaciones generales DHT11*

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 °C	±5%RH	±2°C	1	4 Pin Single Row

**Figura 167 Configuración de pines DHT11**



**Figura 168**

*Especificaciones detalladas DHT11*

Parameters	Conditions	Minimum	Typical	Maximum
<b>Humidity</b>				
Resolution		1%RH	1%RH	1%RH
Repeatability			±1%RH	
Accuracy	25°C		±4%RH	
	0-50°C			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0°C	30%RH		90%RH
	25°C	20%RH		90%RH
	50°C	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25°C, 1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
<b>Temperature</b>				
Resolution		1°C	1°C	1°C
		8 Bit	8 Bit	8 Bit
Repeatability			±1°C	
Accuracy		±1°C		±2°C
Measurement Range		0°C		50°C
Response Time (Seconds)	1/e(63%)	6 S		30 S

### Anexo 3: Especificaciones técnicas del sensor FC-28

**Figura 169**

*Pines del módulo FC-28*



**Figura 170**

*Especificaciones y características modulo FC-28*

#### ESPECIFICACIÓN Y CARACTERÍSTICAS

- Modelo: HW-103
- Chip Comparador: LM393.
- Corriente: 35 mA
- Superficie de electrodo: Estaño
- Voltaje:
  - De Alimentación (VCC): 3.3V – 5V
  - En la señal de salida analógico (AO): 0V hasta 5V
  - En la señal de salida digital (DO): 3.3V / 5V (TTL)
- Dimensiones
  - PCB Panel : 30 mm x 15 mm
  - Sensor : 60 mm x 20 mm
- Pines:
  - VCC (5V)
  - GND
  - Interfaz de salida digital (0 y 1)
  - Interfaz de salida analógica AO

## Anexo 4: Especificaciones técnicas Lámparas Sinjialight 300W

**Figura 171**

*Características de las Lámparas Sinjialight*

Picture	300W Full Spectrum LED Grow Light	
	<b>Input voltage</b>	AC 100V-240V
	<b>Rated power</b>	300W
	<b>Power consumption</b>	47W+/-3%
	<b>Base</b>	With Plug
	<b>LED quantity</b>	465pcs
	<b>Light color</b>	30red+15blue +420warmwhite
	<b>Material</b>	Aluminum
	<b>LED Type</b>	SMD2835
	<b>Dimension</b>	300*70*52mm
	<b>Weight(Volume)</b>	550g

**Figura 172**

*Espectro de iluminación de las Lámparas Sinjialight*

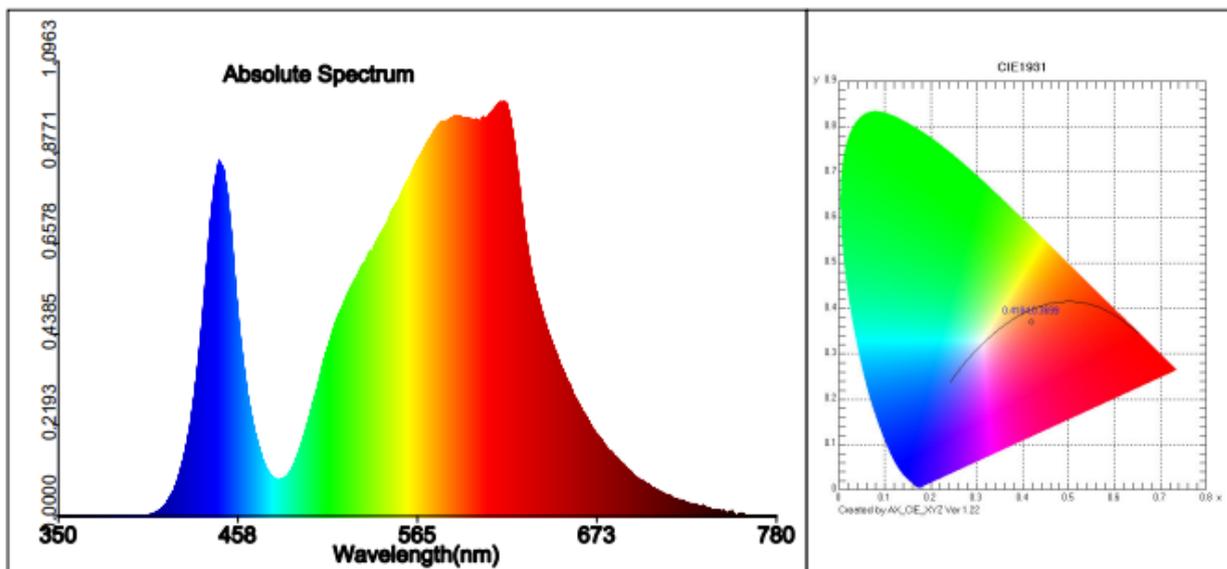


Figura 173

Área de iluminación de las Lámparas Sinjalight



# 300W

FULL SPECTRUM LED GROW LIGHT

Hight PPF: 783  $\mu\text{mol}/(\text{m}\cdot\text{s})$



## Anexo 5: Tabulaciones de resultados

**Tabla 43**

*Tabulación de crecimiento de las plantas del nivel 1*

PLANTA	SEMANA									
	1	2	3	4	5	6	7	8	9	10
<b>N1-1</b>	5.5	8	11.5	12.5	14	18	21.5	23	24.5	25.5
<b>N1-2</b>	6	9	14	17	25	17	21	25	17	21
<b>N1-3</b>	6.5	10.5	17	21	28	18	23	26	16	20.5
<b>N1-4</b>	6.5	9.5	13.5	16.5	20.5	22	26	18	22.5	25.5
<b>N1-5</b>	6	8.5	12.5	13.5	14.5	18.5	21	22	23	25.5
<b>N1-6</b>	5.5	7	9	11	13.5	17.5	20.5	22	24	26
<b>N1-7</b>	6	8.5	14	17	21.5	23	25.5	17.5	21.5	25
<b>N1-8</b>	6.5	9.5	17	20	27.5	18	22	25.5	16.5	20
<b>N1-9</b>	6.5	9	15	17.5	21	23.5	25	17.5	22	26.5
<b>N1-10</b>	6	8	9.5	11.5	13.5	17.5	21.5	22.5	23.5	25

**Tabla 44**

*Tabulación de crecimiento de las plantas del nivel 2*

PLANTA	SEMANA									
	1	2	3	4	5	6	7	8	9	10
<b>N2-1</b>	5	6.5	8	11	14.5	16.5	20	22	23.5	25
<b>N2-2</b>	6	9	11.5	19	25	16.5	19.5	24	26.5	17.5
<b>N2-3</b>	6.5	8.5	14	19.5	27.5	17	21	25.5	18	22.5
<b>N2-4</b>	7	8.5	14	18	21	23.5	25.5	17	21.5	25
<b>N2-5</b>	6.5	8	9	11	13.5	16.5	18	21	23.5	26
<b>N2-6</b>	6	8.5	11.5	13	15	16	18.5	21.5	24	26.5
<b>N2-7</b>	6.5	10.5	14	17.5	25	16	20	26	16.5	20.5
<b>N2-8</b>	7	10.5	17	21	26	15.5	19.5	25	16.5	21
<b>N2-9</b>	6.5	10	14.5	17	20.5	23.5	26	16.5	21	25.5
<b>N2-10</b>	7	8.5	12.5	13.5	12.5	15	18	21	23.5	25.5

**Tabla 45**  
*Tabulación de frondosidad de las plantas del nivel 1*

PLANTA	SEMANA									
	1	2	3	4	5	6	7	8	9	10
<b>N1-1</b>	4	6	5	6	7	6	6	7	7	9
<b>N1-2</b>	4	4	4	4	6	8	8	10	11	10
<b>N1-3</b>	5	7	7	7	9	9	9	9	9	9
<b>N1-4</b>	5	7	7	7	8	10	10	12	12	12
<b>N1-5</b>	4	4	4	5	5	7	9	9	9	9
<b>N1-6</b>	5	6	7	8	9	10	10	11	11	10
<b>N1-7</b>	5	7	8	9	10	10	12	13	11	11
<b>N1-8</b>	4	5	5	6	7	8	10	10	11	12
<b>N1-9</b>	4	4	4	4	6	8	10	12	13	11
<b>N1-10</b>	4	6	6	6	6	7	9	10	10	11

**Tabla 46**  
*Tabulación de frondosidad de las plantas del nivel 2*

PLANTA	SEMANA									
	1	2	3	4	5	6	7	8	9	10
<b>N2-1</b>	4	6	5	5	6	5	5	6	6	8
<b>N2-2</b>	4	4	5	5	7	7	7	8	8	9
<b>N2-3</b>	5	6	7	8	9	10	11	11	12	12
<b>N2-4</b>	5	6	8	8	10	12	10	11	10	10
<b>N2-5</b>	4	6	7	8	8	8	8	8	9	10
<b>N2-6</b>	5	7	9	9	9	9	11	11	10	12
<b>N2-7</b>	5	5	5	6	6	6	8	8	8	8
<b>N2-8</b>	4	5	6	6	8	8	9	11	11	11
<b>N2-9</b>	4	6	6	6	8	10	12	12	11	12
<b>N2-10</b>	4	4	6	7	7	7	8	8	8	8

## Anexo 6: Cálculo de parámetros de desempeño para la selección del algoritmo de aprendizaje

**Tabla 47**

*Matriz de confusión y cálculo de métricas de KNN-3*

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	TP	FP	TN	FN+TP	FN	Accuracy	Precision	Recall	F1
<b>C1</b>	18	0	0	1	0	0	0	0	0	0	0	0	18	1	429	21	3	0.9490	0.9474	0.8571	0.90
<b>C2</b>	0	44	0	0	0	0	3	0	0	0	0	0	44	3	403	47	3	0.9490	0.9362	0.9362	0.94
<b>C3</b>	0	0	80	0	0	0	0	0	0	0	0	0	80	0	367	86	6	0.9490	1.0000	0.9302	0.96
<b>C4</b>	0	0	0	7	0	0	0	0	0	0	0	0	7	0	440	8	1	0.9490	1.0000	0.8750	0.93
<b>C5</b>	0	0	2	0	44	0	0	0	0	0	0	0	44	2	403	47	3	0.9490	0.9565	0.9362	0.95
<b>C6</b>	3	0	0	0	0	21	0	0	0	0	0	0	21	3	426	21	0	0.9490	0.8750	1.0000	0.93
<b>C7</b>	0	2	0	0	0	0	108	0	0	1	0	0	108	3	339	113	5	0.9490	0.9730	0.9558	0.96
<b>C8</b>	0	0	4	0	0	0	0	1	0	0	0	0	1	4	446	1	0	0.9490	0.2000	1.0000	0.33
<b>C9</b>	0	0	0	0	0	0	0	0	17	0	0	0	17	0	430	17	0	0.9490	1.0000	1.0000	1.00
<b>C10</b>	0	0	0	0	0	0	2	0	0	67	0	0	67	2	380	70	3	0.9490	0.9710	0.9571	0.96
<b>C11</b>	0	0	0	0	3	0	0	0	0	0	1	0	1	3	446	1	0	0.9490	0.2500	1.0000	0.40
<b>C12</b>	0	1	0	0	0	0	0	0	0	2	0	39	39	3	408	39	0	0.9490	0.9286	1.0000	0.96
<b>FN+TP</b>	21	47	86	8	47	21	113	1	17	70	1	39			<b>MACRO</b>			0.9490	0.8365	0.9540	0.8531

**Tabla 48***Matriz de confusión y cálculo de métricas de KNN-5*

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	TP	FP	TN	FN+TP	FN	Accuracy	Precision	Recall	F1
<b>C1</b>	18	0	0	1	0	0	0	0	0	0	0	0	18	1	426	21	3	0.9407	0.9474	0.8571	0.90
<b>C2</b>	0	41	0	0	0	0	6	0	0	0	0	0	41	6	403	42	1	0.9407	0.8723	0.9762	0.92
<b>C3</b>	0	0	80	0	0	0	0	0	0	0	0	0	80	0	364	86	6	0.9407	1.0000	0.9302	0.96
<b>C4</b>	0	0	0	7	0	0	0	0	0	0	0	0	7	0	437	8	1	0.9407	1.0000	0.8750	0.93
<b>C5</b>	0	0	2	0	44	0	0	0	0	0	0	0	44	2	400	48	4	0.9407	0.9565	0.9167	0.94
<b>C6</b>	3	0	0	0	0	21	0	0	0	0	0	0	21	3	423	21	0	0.9407	0.8750	1.0000	0.93
<b>C7</b>	0	1	0	0	0	0	109	0	0	1	0	0	109	2	335	119	10	0.9407	0.9820	0.9160	0.95
<b>C8</b>	0	0	4	0	0	0	0	1	0	0	0	0	1	4	443	1	0	0.9407	0.2000	1.0000	0.33
<b>C9</b>	0	0	0	0	0	0	0	0	17	0	0	0	17	0	427	17	0	0.9407	1.0000	1.0000	1.00
<b>C10</b>	0	0	0	0	0	0	3	0	0	66	0	0	66	3	378	69	3	0.9407	0.9565	0.9565	0.96
<b>C11</b>	0	0	0	0	4	0	0	0	0	0	1	0	1	4	443	1	0	0.9407	0.2000	1.0000	0.33
<b>C12</b>	0	0	0	0	0	0	1	0	0	2	0	39	39	3	405	39	0	0.9407	0.9286	1.0000	0.96
<b>FN+TP</b>	21	42	86	8	48	21	119	1	17	69	1	39			<b>MACRO</b>			0.9407	0.8265	0.9523	0.8435

**Tabla 49***Matriz de confusión y cálculo de métricas de Árboles de decisión*

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	TP	FP	TN	FN+TP	FN	Accuracy	Precision	Recall	F1
<b>C1</b>	19	0	0	0	0	0	0	0	0	0	0	0	19	0	447	19	0	0.9915	1.0000	1.0000	1.00
<b>C2</b>	0	46	0	0	0	0	1	0	0	0	0	0	46	1	420	48	2	0.9915	0.9787	0.9583	0.97
<b>C3</b>	0	0	80	0	0	0	0	0	0	0	0	0	80	0	386	80	0	0.9915	1.0000	1.0000	1.00
<b>C4</b>	0	0	0	7	0	0	0	0	0	0	0	0	7	0	459	7	0	0.9915	1.0000	1.0000	1.00
<b>C5</b>	0	0	0	0	46	0	0	0	0	0	0	0	46	0	420	46	0	0.9915	1.0000	1.0000	1.00
<b>C6</b>	0	0	0	0	0	24	0	0	0	0	0	0	24	0	442	24	0	0.9915	1.0000	1.0000	1.00
<b>C7</b>	0	2	0	0	0	0	109	0	0	0	0	0	109	2	357	110	1	0.9915	0.9820	0.9909	0.99
<b>C8</b>	0	0	0	0	0	0	0	4	0	0	0	0	4	0	462	4	0	0.9915	1.0000	1.0000	1.00
<b>C9</b>	0	0	0	0	0	0	0	0	17	0	0	0	17	0	449	17	0	0.9915	1.0000	1.0000	1.00
<b>C10</b>	0	0	0	0	0	0	0	0	0	69	0	0	69	0	397	70	1	0.9915	1.0000	0.9857	0.99
<b>C11</b>	0	0	0	0	0	0	0	0	0	0	4	0	4	0	462	4	0	0.9915	1.0000	1.0000	1.00
<b>C12</b>	0	0	0	0	0	0	0	0	0	1	0	41	41	1	425	41	0	0.9915	0.9762	1.0000	0.99
<b>FN+TP</b>	19	48	80	7	46	24	110	4	17	70	4	41			<b>MACRO</b>			0.9915	0.9947	0.9946	0.9946

## Anexo 7: Implementación y resultados

**Figura 174**

*Plántulas de acelga*



**Figura 175**

*Sembrío de cultivos en el sistema vertical*



**Figura 176**

*Medición de tamaño de hojas en la semana 1*



**Figura 177**  
*Implementación del sistema de riego*



**Figura 178 PCB**  
*Nodo Sensor dentro de caja plástica*



**Figura 179**  
*Resultados*



**Anexo 8: Detalle de costos del proyecto****Tabla 50***Costos de implementación de la estructura vertical*

<b>Ítem</b>	<b>Cantidad</b>	<b>Valor</b>	<b>Total</b>
Plancha Tríplex 9mm 240x122	1	17.5	17.5
Tiras de Madera 5x5x200	10	1	10
Tornillos 2 pulgadas	50	0.04	2
Metro plástico negro	4	1.5	6
Plántula Acelga	30	0.05	1.5
Bulto de tierra abonada	4	5	20
Libra Abono orgánico	5	0.5	2.5
	<b>Total</b>		<b>59.5</b>

**Tabla 51***Costos de implementación de los Nodos Sensores*

<b>Item</b>	<b>Cantidad</b>	<b>Valor</b>	<b>Total</b>
<b>Hardware</b>			
Node MCU ESP8266	2	7.14	14.28
Modulo Rele x2	2	2.5	5
Cables Conectores	20	0.08	1.6
Resistencia 100 ohms	2	0.1	0.2
Resistencia 2.2 kohms	2	0.1	0.2
Transistor TIP41	2	0.5	1
Barra de Espadines Macho	1	0.5	0.5
Barra de espadines Hembra	2	0.5	1
Impresión PCB	2	10	20
Caja Plástica 156x86	2	5	10
<b>Software</b>			
Arduino IDE	1	0	0
Fritzing	1	0	0
	<b>Total</b>		<b>53.78</b>

**Tabla 52***Costos de implementación del Nodo Central*

<b>Item</b>	<b>Cantidad</b>	<b>Valor</b>	<b>Total</b>
<b>Hardware</b>			
Raspberry PI3B	1	119.5	119.5
<b>Software</b>			
Node-RED	1	0	0
MySQL Server	1	0	0
Mosquitto MQTT Broker	1	0	0
			<b>119.5</b>

**Tabla 53**  
*Costos de implementación de sensores y actuadores*

<b>Item</b>	<b>Cantidad</b>	<b>Valor</b>	<b>Total</b>
Sensor DHT11	2	3.5	7
Sensor FC28	4	2	8
Sensor LDR	2	0.5	1
Ventilador 5v 0.25A	4	2.5	10
Lámparas Growing 300W	2	60	120
Electrovalvula 110v	2	10	20
Metro manguera 1/2 pulgada	6	0.5	3
Abrazadera	4	0.5	2
Rollo Teflón	1	0.5	0.5
T 1/2 pulgada	1	0.5	0.5
Conector Manguera	4	1	4
Sujetador Manguera 1/2	10	0.1	1
Tornillos 1 pulgada	10	0.02	0.2
	<b>Total</b>		177.2

**Tabla 54**  
*Costos de implementación de la aplicación móvil*

<b>Item</b>	<b>Cantidad</b>	<b>Valor</b>	<b>Total</b>
Firebase Realtime 1GB/360MBxdia	1	0	0
Ionic Framework	1	0	0
Chart JS	1	0	0
Aplicación Móvil	1	0	0
	<b>Total</b>		0

**Tabla 55**  
*Costos de implementación del sistema eléctrico*

<b>Item</b>	<b>Cantidad</b>	<b>Valor</b>	<b>Total</b>
Metro cable gemelo 16 awg	8	0.4	3.2
Enchufe eléctrico	4	0.5	2
Toma Corriente	1	0.6	0.6
Metro cable de audio	8	0.2	1.6
Cable USB	2	1.5	3
Fuente 5V USB 1A	2	3	6
Consumo energético/mes	3	4.95	14.85
	<b>Total</b>		31.25

### Anexo 9: Script de Matlab para el cálculo de TXTIME de 802.11b/g

```

%% 802.11 B
clc
% longitud del preambulo (largo = 144, corto =72 )
PreambleLength = 144 ;
% tiempo de la cabecera DSSS (largo = 48, corto = 24)
PLCPHeaderTime = 48 ;
% Tamaño de la PSDU
LENGTH = 145 ;
% Modulación PBCC = 1 , CCK u otro = 0
PBCC = 0;
% Mbps del DataRate
DATARATE = 1;
TXTIME = PreambleLength + PLCPHeaderTime + ceil(((LENGTH+PBCC)*8) /
DATARATE)

%% 802.11 G
clc
% Tiempo del preámbulo
TPREAMBLE = 16;
% Tiempo de la señal OFDM 8bits
TSIGNAL = 4;
% Tiempo de un símbolo OFDM
TSYM = 4;
% Tamaño de la PSDU
LENGTH = 145;
% Numero de bits por simfolo OFDM | Data Rate
(Mbps)
% MCS Modulation Coding Rate (NBPSC) (NCBPS) (NDBPS) 20 MHz 10 MHz 5
MHz
% 0 BPSK 1/2 1 48 24 6 3
1.5
% 1 BPSK 3/4 1 48 36 9 4.5
2.25
% 2 QPSK 1/2 2 96 48 12 6
3
% 3 QPSK 3/4 2 96 72 18 9
4.5
% 4 16QAM 1/2 4 192 96 24 12
6
% 5 16QAM 3/4 4 192 144 36 18
9
% 6 64QAM 2/3 6 288 192 48 24
12
% 7 64QAM 3/4 6 288 216 54 27
13.5
NDBPS = 24;
TXTIME = TPREAMBLE + TSIGNAL + TSYM * ceil ((16 + 8 * LENGTH + 6)/NDBPS)

```

## **Anexo 10: Repositorios de GitHub del código del proyecto**

- **Repositorio de los flujos de Node-RED**

<https://github.com/cmruizg777/tesis-node-red.git>

- **Repositorio del código fuente de la aplicación móvil**

<https://github.com/cmruizg777/tesis-aplicacion-movil.git>

- **Repositorio del código de los Nodos Sensores**

<https://github.com/cmruizg777/tesis-nodos-sensores.git>

- **Repositorio de las bases de datos**

<https://github.com/cmruizg777/tesis-bases-de-datos.git>