



**UNIVERSIDAD NACIONAL DE CHIMBORAZO  
FACULTAD DE INGENIERÍA**

**CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES**

**“Trabajo de grado previo a la obtención del Título de Ingeniero en  
Electrónica y Telecomunicaciones”**

**TRABAJO DE GRADUACIÓN**

**Título del proyecto:**

**“IMPLEMENTACIÓN DE UN SISTEMA DE ALMACENAMIENTO DE  
LA INFORMACIÓN, MONITOREO Y CONTROL APLICANDO EL  
INTERNET DE LAS COSAS, PARA LA AUTOMATIZACIÓN DE UN  
INVERNADERO”.**

**Autores:**

**Nelson Joaquín Melo Paredes  
Walter Abdón Valverde Macao**

**Director:**

**Ing. Alfonso Gunsha**

**RIOBAMBA-ECUADOR**

**Año 2017**

Los miembros del Tribunal de Graduación del proyecto de investigación de título: IMPLEMENTACIÓN DE UN SISTEMA DE ALMACENAMIENTO DE LA INFORMACIÓN, MONITOREO Y CONTROL APLICANDO EL INTERNET DE LAS COSAS, PARA LA AUTOMATIZACIÓN DE UN INVERNADERO presentado por: Nelson Joaquín Melo Paredes y Walter Abdón Valverde Macao y dirigida por: Ingeniero Alfonso Gunsha.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. José Jinez

Presidente del Tribunal



Firma

Ing. Alfonso Gunsha

Director del Proyecto



Firma

Ing. Geovanny Cuzco

Miembro del Tribunal



Firma

Ing. Juan Carlos Cepeda

Miembro del Tribunal



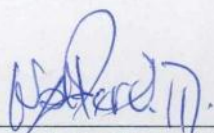
Firma

## AUTORIA DE INVESTIGACIÓN

La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: **Nelson Melo, Walter Valverde e Ingeniero Alfonso Gunsha**; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo|



Nelson Joaquín Melo Paredes  
C.C. 0603346305



Walter Abdón Valverde Macao  
C.C. 0603346305



Alfonso Javier Gunsha Morales  
C.C. 0603797853

## **AGRADECIMIENTO**

Agradezco a la facultad de Ingeniería y sobre todo a los docentes de la Carrera de Ingeniería en Electrónica y Telecomunicaciones por la ayuda incondicional para la culminación de este proyecto, a mi familia y amigos por la comprensión y entendimiento perenne a lo largo de mi carrera universitaria.

Nelson Joaquín Melo P

## **DEDICATORIA**

A Nelson y Aída por ser mis pilares fundamentales, fuentes de constancia abrumadora de trabajo, de voluntad y del arte para poder sentir lo profundo y lo más sublime del Amor.

A mis queridas hermanas Nataly y Mayra por saber escuchar y enseñar que existe un cúmulo de alegrías y desvelos para llegar a un peldaño del triunfo. A mis queridos sobrinos Martín, Steven, Daniel por su afecto y sus largas sonrisas enriquecedoras. Al Gran Espíritu que ha entrelazado el camino para forjar con las más nobles y grandes virtudes inherentes y palpables en este maravilloso soplo de vida nuestras huellas ad infinitum.

Nelson Joaquín Melo P.

## **AGRADECIMIENTO**

Un agradecimiento especial a la confianza y el apoyo brindado por parte de mi madre, que sin duda alguna en el trayecto de mi vida me ha demostrado su amor, corrigiendo mis faltas, celebrando mis triunfos y que nunca dejó de ayudarme, hasta en la cosa más mínima estuvo preocupada por mi carrera y que la pudiera culminar con éxito.

A mi hermano Miguel, que con su apoyo y consejos me ha ayudado a afrontar los retos que se me han presentado a lo largo de mi vida.

A mi novia Laura, que ha sabido apoyarme para continuar y nunca renunciar, gracias por su amor incondicional y por su ayuda en mi proyecto.

A todos con mucho cariño.

Walter Abdón Valverde M.

## **DEDICATORIA**

Dedico este trabajo principalmente a mi madre, por ser el pilar más importante y por demostrarme siempre su cariño y apoyo incondicional, su esfuerzo tuvo recompensas y espero que se sienta orgullosa de mí, esto es por ella. A mi padre, a pesar de nuestra distancia física, siento que estás conmigo siempre y aunque nos faltaron muchas cosas por vivir juntos, sé que este momento hubiera sido tan especial para ti como lo es para mí. A mis hermanos Luis, Milton, Auria y Miguel quienes me han acompañado y apoyado. A mis familiares, viejos amigos y a Laura quien recién se sumó a mi vida para hacerme compañía con sus sonrisas de ánimo.

Walter Abdón Valverde M.

## ÍNDICE GENERAL

AGRADECIMIENTO .....	iv
DEDICATORIA .....	v
ÍNDICE GENERAL.....	viii
ÍNDICE DE TABLAS .....	xix
RESUMEN.....	xx
INTRODUCCIÓN .....	1
PROBLEMA.....	2
JUSTIFICACIÓN .....	3
CAPÍTULO I.....	6
1. FUNDAMENTACIÓN TEÓRICA.....	6
1.1. INVERNADERO.....	6
1.1.1. TIPOS DE INVERNADEROS .....	6
1.2. CULTIVO DE SEMILLAS .....	7
1.2.1. SIEMBRA.....	7
1.2.2. NIVELES DE TEMPERATURA Y HUMEDAD RELATIVA.....	7
1.2.3. EFICIENCIA DEL REGADÍO.....	7
1.2.4. COSECHA .....	8
1.3. PLUVIOMETRÍA.....	8
1.3.1. LA TASA DE PRECIPITACIÓN.....	8
1.3.2. MEDICIÓN DE LA PRECIPITACIÓN .....	8
1.4. INTERNET .....	9
1.4.1. PROTOCOLO TCP/IP.....	9
1.5. COMUNICACIÓN INALÁMBRICA .....	10
1.5.1. WIFI.....	10



1.6.	DISPOSITIVOS DE COMUNICACIÓN WIFI .....	11
1.6.1.	ROUTER TL-MR3040 .....	11
1.7.	MÓDULO WIFI ESP8266.....	13
1.8.	MICRO-CONTROLADOR .....	14
1.9.	REDES LOCALES WLAN .....	15
1.10.	INTERNET OF THINGS“IOT” (INTERNET DE LAS COSAS). .....	15
1.10.1.	DEFINICIÓN .....	15
1.10.2.	HISTORIA .....	16
1.10.3.	ARQUITECTURA IoT .....	16
1.10.4.	MODELO DE REFERENCIA IoT. ....	17
1.10.5.	PROTOCOLOS IoT .....	18
1.10.6.	APLICACIONES IoT. ....	19
1.11.	SENSORES.....	20
1.11.1.	DHT11 .....	21
1.11.2.	SENSOR REED SWITCH.....	21
1.12.	ACTUADORES.....	22
1.12.1.	VENTILADOR .....	22
1.12.2.	CALEFACTOR.....	22
1.12.3.	ELECTROVÁLVULA.....	23
CAPITULO II .....		24
2.	METODOLOGÍA .....	24
2.1.	TIPO DE ESTUDIO .....	24
2.2.	POBLACIÓN Y MUESTRA.....	25
2.2.1.	POBLACIÓN.....	25
2.2.2.	MUESTRA.....	25
2.2.3.	HIPÓTESIS.....	26

2.3.	OPERACIONALIZACIÓN DE VARIABLES.....	26
2.3.1.	VARIABLE INDEPENDIENTE.....	26
2.3.2.	VARIABLE DEPENDIENTE.....	27
2.4.	PROCEDIMIENTOS.....	27
2.4.1.	DESARROLLO DEL PROTOTIPO.....	28
2.4.1.1.	DISEÑO DE LOS CIRCUITOS ELECTRÓNICOS DE ACONDICIONAMIENTO PARA LOS SENSORES TEMPERATURA/HUMEDAD Y PLUVIOMETRIA. ....	30
2.4.1.2.	DISEÑO DE LOS CIRCUITOS ELECTRÓNICOS DE ACONDICIONAMIENTO PARA LOS ACTUADORES ELECTROVÁLVULA, VENTILACIÓN Y CALEFACCIÓN. .....	40
2.4.1.3.	CONFIGURACIÓN DEL GATEWAY. ....	45
2.4.1.3.1.	CONFIGURACIÓN DE LA COMPUTADORA .....	45
2.4.1.3.2.	CONEXIÓN A LA RED INALÁMBRICA.....	47
2.4.1.3.3.	CONFIGURACIÓN DEL ENRUTADOR .....	49
2.4.1.4.	DISEÑO Y CONFIGURACIÓN EN GOOGLE SPREADSHEETS. (HOJA DE CÁLCULO DE GOOGLE) PARA EL ALMACENAMIENTO DE LA INFORMACIÓN, ADMINISTRACIÓN Y CONTROL DEL INVERNADERO.	58
2.4.1.5.	PROGRAMACIÓN EN LA NUBE APLICANDO GOOGLE APP SCRIPT Y GOOGLE SPREADSHEET. ....	68
2.4.1.5.1.	PROGRAMACIÓN DEL SENSOR DE TEMPERATURA Y HUMEDAD 1.....	69
2.4.1.5.2.	PROGRAMACIÓN DEL SENSOR DE TEMPERATURA Y HUMEDAD 2.....	75
2.4.1.5.3.	PROGRAMACIÓN DEL SENSOR DE TEMPERATURA Y HUMEDAD 3.....	76

2.4.1.5.4.	PROGRAMACIÓN DEL SENSOR DE PLUVIOMETRÍA. .....	77
2.4.1.6.	INSTALACIÓN, CONFIGURACIÓN DE ARDUINO IDE Y PROGRAMACIÓN DEL MÓDULO ESP8266.....	81
2.4.1.7.	CREACIÓN DE PAGINA WEB EN GOOGLE SITES PARA EL MONITOREO Y CONTROL DEL INVERNADERO. ....	87
2.4.1.8.	CREACIÓN DE LA APLICACIÓN ANDROID PARA CELULAR.....	90
2.4.1.9.	IMPLEMENTACIÓN DEL SISTEMA.....	104
2.5.	COMPROBACIÓN DE HIPÓTESIS .....	107
CAPITULO III.....		112
3.	PRUEBAS Y RESULTADOS DEL PROTOTIPO DE INVERNADERO.	112
3.1.	PRUEBAS DE MEDICIÓN DE TEMPERATURA Y HUMEDAD RELATIVA SENSOR 1 .....	114
3.2.	PRUEBAS DE MEDICIÓN DE TEMPERATURA Y HUMEDAD RELATIVA SENSOR 2 .....	115
3.3.	PRUEBAS DE MEDICIÓN DE TEMPERATURA Y HUMEDAD RELATIVA SENSOR 3 .....	116
3.4.	PRUEBAS DE MEDICIÓN DE TEMPERATURA PROMEDIO DE LOS SENSORES 1, 2, 3.....	118
3.5.	PRUEBAS DE FUNCIONAMIENTO AUTOMÁTICO Y MANUAL DEL PROTOTIPO DE INVERNADERO.....	119
3.5.1.	MODO AUTOMÁTICO PARA LA ACTIVACIÓN DE LOS ACTUADORES.....	119
3.5.2.	MODO MANUAL PARA LA ACTIVACIÓN DE LOS ACTUADORES.....	120
3.6.	PRUEBAS DE FUNCIONAMIENTO DE LA INTERFAZ WEB (GOOGLE SPREADSHEET Y PÁGINA WEB).....	122

3.6.1. GOOGLE SPREADSHEET.....	122
3.6.2. PÁGINA WEB.....	125
3.7. PRUEBAS DE FUNCIONAMIENTO DE LA APLICACIÓN ANDROID .....	127
3.7.1. ENCENDIDO AUTOMÁTICO DEL VENTILADOR.....	127
3.7.2. ENCENDIDO AUTOMÁTICO DEL CALEFACTOR.....	128
3.8. RESULTADOS DEL PROTOTIPO DE INVERNADERO.....	129
CAPITULO IV.....	131
4. DISCUSIÓN .....	131
CAPITULO V .....	132
5. CONCLUSIONES Y RECOMENDACIONES.....	132
5.2. RECOMENDACIONES .....	133
CAPITULO VI.....	134
6. BIBLIOGRAFÍA .....	134
CAPÍTULO VII .....	135
7. ANEXOS .....	135

## ÍNDICE DE FIGURAS

Figura 1. Invernadero tipo Túnel. ....	7
Figura 2. Pluviómetro de Cangilometro.....	8
Figura 3. Panel Frontal del Router TL-MR3040.....	11
Figura 4. Panel Posterior del Router TL-MR3040.....	12
Figura 5. Módulo ESP8266.....	13
Figura 6. Diagrama de Bloques ESP8266.....	14
Figura 7. Comunicación Wireless Lan (ESP8266 y TL-MR3040).....	15
Figura 8. Arquitectura IoT. ....	16
Figura 9. Modelo de Referencia IoT. ....	17
Figura 10. Sensor DHT11. ....	21
Figura 11. Sensor Reed Switch .....	22
Figura 12. Ventilador. ....	22
Figura 13. Electroválvula. ....	23
Figura 14. Diagrama de Bloque del Sistema.....	28
Figura 15. Sensores y Actuadores conectados al Router mediante la red WIFI. ..	29
Figura 16. Distribución de los pines para el módulo WIFI ESP8266.....	30
Figura 17. Diagrama electrónico del circuito eléctrico de acondicionamiento para el sensor de temperatura y humedad DHT11.....	31
Figura 18. Diseño y diagrama electrónico PCB del circuito de acondicionamiento para el sensor 1 de Temperatura y Humedad DHT11.....	32
Figura 19. Diseño y diagrama electrónico PCB del circuito de acondicionamiento para el sensor 2 de Temperatura y Humedad DHT11.....	32
Figura 20. Diseño y diagrama electrónico PCB del circuito de acondicionamiento para el sensor 3 de Temperatura y Humedad DHT11.....	33
Figura 21. Diseño y Diagrama electrónico PCB del circuito de acondicionamiento para el sensor 4 de Pluviometría. ....	33

Figura 22. Transferencia del diagrama electrónico a la placa de cobre. ....	34
Figura 23. Sumergida en Agua. ....	34
Figura 24. Quitado del Papel de la placa.....	34
Figura 25. Cloruro Férrico. ....	35
Figura 26. Colocación de la placa en el Acido.....	35
Figura 27. Placa Cortada y Lijada.....	35
Figura 28. Dispositivos Electrónicos. ....	36
Figura 29. Placa electrónica del circuito de acondicionamiento para el sensor 1 de Temperatura y Humedad DHT11. ....	36
Figura 30. Placa electrónica del circuito de acondicionamiento para el sensor 2 de Temperatura y Humedad DHT11. ....	36
Figura 31. Placa electrónica del circuito de acondicionamiento para el sensor 3 de Temperatura y Humedad DHT11. ....	37
Figura 32. Placa electrónica del circuito de acondicionamiento para el sensor 4 de Pluviómetro. ....	37
Figura 33. Báscula del Pluviómetro. ....	38
Figura 34. Separador en el centro de la Báscula. ....	39
Figura 35. Montaje Mecánico de la Báscula. ....	39
Figura 36. Equipo de Pluviometría. ....	40
Figura 37. Diseño del Circuito Electrónico de Acondicionamiento Actuadores. .	41
Figura 38. Diseño y Diagrama Electrónico PCB del Circuito de Acondicionamiento Actuador 1. ....	42
Figura 39. Diseño y Diagrama Electrónico PCB del Circuito de Acondicionamiento Actuador 2. ....	42
Figura 40. Diseño y Diagrama Electrónico PCB del Circuito de Acondicionamiento Actuador 3. ....	43
Figura 41. Placa Cortada y Lijada.....	43

Figura 42. Placa electrónica del circuito de acondicionamiento para el Actuador 1 de la Electroválvula.....	44
Figura 43. Placa electrónica del circuito de acondicionamiento para el actuador 2 del ventilador.....	44
Figura 44. Placa electrónica del circuito de acondicionamiento para el Actuador 3 Calefactor. ....	45
Figura 45. Parte trasera Router TP-LINK MR3040.....	45
Figura 46. Selección de Propiedades de la conexión de red inalámbrica. ....	46
Figura 47. Propiedades de la conexión de red inalámbrica.....	46
Figura 48. Parámetros de red TCP/IPv4 .....	47
Figura 49. Redes inalámbricas disponibles.....	47
Figura 50. Contraseña de red inalámbrica predeterminada.....	48
Figura 51. Intentando conectarse a red inalámbrica.....	48
Figura 52. Red inalámbrica conectada. ....	49
Figura 53. Iniciar sesión en el router.....	49
Figura 54. Iniciar sesión en Windows.....	50
Figura 55. Estado del router. ....	50
Figura 56. Menú Red. ....	51
Figura 57. WAN - IP Dinámico. ....	51
Figura 58. Parámetros LAN .....	52
Figura 59. Menú wireless.....	53
Figura 60. Configuraciones inalámbricas - WISP.....	53
Figura 61. Lista de puntos de acceso. ....	55
Figura 62. Parámetros WIFI llenados. ....	55
Figura 63. Configuración de la red inalámbrica local.....	56
Figura 64. Visualización de red inalámbrica creada. ....	56
Figura 65. Ingreso de la contraseña de red inalámbrica. ....	57

Figura 66. Conectado a la red inalámbrica.....	57
Figura 67. Creación de nuevo documento.....	58
Figura 68. Nombre hoja de Cálculo. ....	59
Figura 69. ID de Conexión.....	59
Figura 70. Hojas de cálculo creadas.....	60
Figura 71. Seleccionar Gráfico. ....	60
Figura 72. Histórico del Sensor.....	60
Figura 73. Diagrama Flujo el Sensor. ....	61
Figura 74. Datos Obtenidos del sensor. ....	62
Figura 75. Datos de Temperatura Promedio. ....	62
Figura 76. Interfaz de Google Spreadsheet. ....	63
Figura 77. Interfaz de la TempHum 2.....	63
Figura 78. Interfaz de la TempHum 3.....	64
Figura 79. Interfaz de la Precipitación. ....	65
Figura 80. Visualización de Temperatura en Celdas. ....	66
Figura 81. Visualización de Humedad en Celdas. ....	66
Figura 82. Visualización de Precipitación, Calefactor, Ventilador, Electroválvula, Control en las celdas. ....	66
Figura 83. Configuración de la Hora de Riego. ....	67
Figura 84. Estado Actual de los Actuadores, Sensores.....	67
Figura 85. Interfaz de Control.....	67
Figura 86. Creación de Cuenta de Usuario en Google.....	68
Figura 87. Página Web de google Script.....	69
Figura 88. Editor de Comandos. ....	69
Figura 89. Implementación de la Aplicación Web.....	73
Figura 90. Opciones de Acceso al Script .....	74



Figura 91. Obtención del ID .....	74
Figura 92. Visualización del parámetro Querystring. ....	75
Figura 93. Resultados del parámetro Querystring.....	75
Figura 94. Registra de Secuencia de Comandos Sensor 4. ....	79
Figura 95. Registra de Secuencia de Comandos Actuador 1. ....	79
Figura 96. Registra de Secuencia de Comandos Actuador 2. ....	80
Figura 97. Registra de Secuencia de Comandos Actuador 3. ....	80
Figura 98. Mensaje de Inicio Arduino. ....	81
Figura 99. Interfaz del IDE Arduino. ....	82
Figura 100. Dirección del Modelo de Tarjeta. ....	82
Figura 101. Selección de la Tarjeta ESP826. ....	83
Figura 102. Selección del Puerto COM. ....	83
Figura 103. Diagrama electrónico para programar al módulo WIFI ESP8266.....	84
Figura 104. Conexión al Puerto USB del Grabador de Módulos WIFI. ....	86
Figura 105. Interfaz WEB. ....	87
Figura 106. Ventana de Inicio de la Aplicación Android. ....	95
Figura 107. Ventana de Visualización de los Sensores. ....	96
Figura 108. Botón de opciones de la Aplicaciones Android.....	96
Figura 109. Configuraciones existentes en la Aplicación Android.....	97
Figura 110. Configuración de Temperatura y Humedad. ....	97
Figura 111. Ventana Control.....	99
Figura 112. Ventana Ayuda. ....	101
Figura 113. Ventana Acerca de. ....	103
Figura 114. Pluviómetro en Ejecución.....	104
Figura 115. Caja Máster.....	105
Figura 116. Conexión de Actuadores.....	105

Figura 117. Conexión de Sensores a la Caja Máster.....	106
Figura 118. Prototipo de Invernadero. ....	106
Figura 119. Encendido del Calefactor en 13°C.....	109
Figura 120. Encendido del Ventilador en 19°C. ....	110
Figura 121. Histórico de Temperatura y Humedad Sensor 1.....	115
Figura 122. Histórico de Temperatura y Humedad Sensor 2.....	116
Figura 123. Histórico de Temperatura y Humedad Sensor 3.....	117
Figura 124. Temperatura Promedio de los Sensores 1,2,3.....	119
Figura 125. Variables en Ejecución del Invernadero. ....	122
Figura 126. Temperatura y Humedad Relativa Promedio y Rangos Min/Max. .	123
Figura 127. Accionamiento de los Actuadores ON/OFF y modo de Control Automático/Manual.....	123
Figura 128. Interfaz Web. ....	125
Figura 129. Actuador Encendido en Modo Automático. ....	125
Figura 130. Actuadores Apagados en Modo Automático.....	126
Figura 131. Actuador Encendido en Modo Automático. ....	126
Figura 132. Actuador Encendido en Modo Manual.....	127
Figura 133. Modo Automático Aplicación Android.....	127
Figura 134. Encendido del Ventilador Aplicación Android. ....	128
Figura 135. Encendido del Calefactor Aplicación Android.....	128
Figura 136. Salida del Puerto COM del Calefactor.....	129
Figura 137. Salida del Puerto COM del Ventilador.....	129
Figura 138. Fase inicial del Cultivo en el Inverndaero. ....	130
Figura 139. Periodo de Crecimiento de Lechugas. ....	130

## ÍNDICE DE TABLAS

Tabla 1. Capas del Protocolo TCP/IP.....	9
Tabla 2. Características de los Leds del Router TL-MR3040.....	12
Tabla 3. Especificaciones de los pines del módulo WIFI ESP8266.....	14
Tabla 4. Operacionalización de la Variable Independiente.....	26
Tabla 5. Operacionalización de la variable dependiente.....	27
Tabla 6. Activación de Ventilador y Calefactor.....	107
Tabla 7. Frecuencias y Porcentajes para el Calefactor.....	109
Tabla 8. Frecuencias y Porcentajes para el Ventilador.....	110
Tabla 9. Datos Estadísticos.....	111
Tabla 10. Prueba de Chi -Cuadrado.....	111
Tabla 11. Funcionamiento de Actuadores.....	112
Tabla 12. Funcionamiento de Sensores.....	113
Tabla 13. Datos de Temperatura en Función del Tiempo.....	114
Tabla 14. Datos de Temperatura en función del Tiempo.....	115
Tabla 15. Datos de Temperatura en función del Tiempo.....	117
Tabla 16. Temperatura Promedio de los Sensores 1,2,3.....	118
Tabla 17. Funcionamiento Automático del Prototipo de Invernadero.....	119
Tabla 18. Funcionamiento Manual del Prototipo de Invernadero.....	120
Tabla 19. Parámetros de Configuración para el sistema de Pluviometría.....	124
Tabla 20. Estado Actual de los Actuadores y Sensores.....	124

## RESUMEN

El presente proyecto tiene como objetivo el diseño, construcción y automatización de un prototipo de invernadero, almacenado, administrado, monitoreado y controlado a través de internet utilizando los servidores de google y sus aplicaciones. Presentando así una aplicación de IoT (Internet of Things) para el control de cultivos como alternativa a la agricultura tradicional.

Con la construcción del prototipo de invernadero se logra controlar diferentes parámetros tales como temperatura, humedad relativa y riego de agua, con el fin de automatizar, mejorar la calidad y cantidad del producto. Este prototipo se puede acondicionar para cualquier tipo de cultivo.

El sistema es controlado remotamente y para la administración cuenta con un interfaz realizado y alojado en los servidores de google, para el monitoreo, control automático y manual se utiliza una página web y una aplicación para dispositivos con sistema operativo android. Se utiliza un sistema de comunicación inalámbrico que envía la información de los parámetros del invernadero hacia internet, el mismo que son receptados, procesados y almacenados por las aplicaciones de google (google app script y google spreadsheet).

Mediante la hoja de cálculo de google, página web y aplicación android se monitorea los sensores y se controla de forma manual y automática los actuadores, además se modifica las variables del invernadero de acuerdo al cultivo.

Palabras clave: Iot, invernadero, google, spreadsheet, sensores, actuadores, app script, android, web.

## Abstract

The present project aims at the design, construction and automation of a greenhouse prototype, stored, managed, monitored and controlled through the internet using Google servers and their applications. Introducing an application of IoT (Internet of Things) for crop control as an alternative to traditional agriculture. With the construction of the prototype of greenhouse it is possible to control different parameters such as temperature, relative humidity and watering, in order to automate to improve the quality and quantity of the product. This prototype can be conditioned for any type of crop. The system is controlled remotely and for the administration has an interface made and hosted in the Google servers, for monitoring, automatic control and manual is used a web page and an application for devices with android operating system. A wireless communication system is used that sends the information of the greenhouse parameters to the internet, the same that are received, processed and stored by the Google app script and Google spreadsheet. Using the Google spreadsheet, web page and android application the sensors are monitored and the actuators are manually and automatically controlled, and the greenhouse variables are modified according to the crop



Reviewed by: Barriga, Luis  
Language Center Teacher



## INTRODUCCIÓN

Internet de las Cosas es una visión de un mundo en el que la mayoría de los objetos están conectados, transmitiendo actualizaciones de los sensores y actuadores y realizar el control con mayor eficacia. Esta visión se está construyendo hoy en día, con dispositivos conectados cada vez más frecuentes en nuestra vida cotidiana.

El concepto básico detrás de IoT es que prácticamente todas las cosas físicas en este mundo también pueden convertirse en una computadora que está conectada a Internet.

En este trabajo se presenta una visión general del fenómeno de Internet de Cosas, así como su aplicación a la agricultura. En la era de Internet, donde la información desempeña un papel clave en la vida de las personas, la agricultura se está convirtiendo rápidamente en una industria muy intensiva en datos, en la que los agricultores deben acumular una gran cantidad de datos para ser más eficientes en la producción y comunicar información apropiada.

A medida que el concepto de Internet de las Cosas se vuelve cada vez más frecuente, muchos sistemas están siendo ideados para permitir recopilar y analizar todo tipo de datos, y dispositivos controlados a través de conexiones inalámbricas. Las condiciones ambientales correctas son imprescindibles para el crecimiento exitoso de las plantas, la mejora de los rendimientos de los cultivos y el uso efectivo del agua y recursos similares. También se cuenta con el desarrollo del sistema de almacenamiento, monitoreo y control del invernadero, presentando soluciones como la aplicación en android, la pagina web, el panel de control.

Entre los esfuerzos para cultivar más alimentos producidos con mayor tecnificación y optimización de recursos, una generación más joven de agricultores y entusiastas han comenzado a depender de una infusión de datos y tecnología.

## **PROBLEMA**

Actualmente los invernaderos no cuentan con algún método o sistema de monitoreo remoto que permita conocer datos referentes a la humedad, temperatura u otros factores ambientales y que cree registros de forma automática y exacta, es decir, obtener registros que sean creados las 24 horas del día y con datos reales tomados por sensores de última generación.

Al ser un limitante el monitoreo de los cultivos en invernaderos, conlleva a depender de la información entregada por la persona que está encargada del cuidado y recolección de datos, que en muchos de los casos no lo realizan adecuadamente; haciendo que la producción baje, así como la calidad del producto no sea óptima, produciendo una pérdida constante en el cultivo con invernadero en cuanto a recursos económicos y recursos humanos.

Los agricultores no utilizan un adecuado monitoreo de las precipitaciones para saber la cantidad de lluvia que ha caído durante un lapso de tiempo, haciendo de esto un factor crucial para la optimización del agua tanto en invernaderos como en sembríos a campo abierto, ya que puede ser almacenada en un reservorio para su posterior uso en el riego del invernadero.

## JUSTIFICACIÓN

Se tiene muchas ventajas al tener cultivos bajo invernadero, esto evita los cambios bruscos del clima como la variación de temperatura, la escasez o exceso de humedad. También se puede producir cultivos en las épocas del año más difíciles teniendo cosechas fuera de temporal, sustituyendo el clima de otras regiones, alargando el ciclo del cultivo y un eficaz control de plagas.

Otra de las ventajas es el de obtener productos de mejor calidad y una mayor producción en la cosecha, y así incrementar la economía. Este incremento del valor de los productos permite que el agricultor pueda invertir tecnológicamente en su explotación mejorando la estructura del invernadero.

Al optimizar los recursos al máximo con la ayuda de la ingeniería se tiene un ahorro considerable de energía eléctrica y agua. Este último es un problema grave “el 77.8% del agua a nivel mundial es para uso agrícola con una eficiencia de riego en el ecuador del 60.55%”. (Fuente: III CNA, 2000). Por esta razón es importante invertir en investigación y desarrollo de tecnologías. Es por eso que para el desarrollo económico de este país la inversión de invernaderos no es un lujo sino una necesidad.

El monitoreo remoto de cultivos en invernaderos permite visualizar, conocer y controlar de manera exacta, en tiempo real y en cualquier momento los niveles de humedad y temperatura que afecta al área monitoreada, independientemente de si se encuentra en el invernadero, o cualquier otro lugar que tenga acceso a internet, por tal motivo, se facilita llevar un registro automático, digital y continuo de los factores ya antes mencionados, y ante eventos ambientales inesperados poder tomar acciones oportunas para evitar que afecten de mayor manera a los cultivos en invernaderos.

Al contar con un sistema de almacenamiento de la información, monitoreo y control usando sensores que permite observar datos en tiempo real de las mediciones de pluviometría, humedad relativa y temperatura mediante un navegador WEB desde cualquier sitio con acceso a internet, además de llevar un



ordenado y real registro de datos obtenidos por el nodo en forma continua y con intervalos de tiempo administrables.

En actividades como la agricultura también es fundamental el uso de los pluviómetros para el control de las precipitaciones del agua a nivel local, saber cuándo realizar determinadas plantaciones, también permite calcular con anticipación, el volumen y calidad de la cosecha y sobre todo de gran importancia para el aprovechamiento racional del agua en los cultivos.

La investigación es importante porque abarca un campo del conocimiento en electrónica y telecomunicaciones muy poco explorado y que sin embargo ofrece múltiples aplicaciones. Este proyecto es importante ya que permite dar a conocer el IoT y encontrar otras aplicaciones para desarrollar en futuros proyectos.

En nuestra condición de profesionales de electrónica y telecomunicaciones se tiene la necesidad de implementar este proyecto porque conlleva múltiples beneficios principalmente a las granjas agrícolas.

Para la elaboración de este proyecto hay que tomar en cuenta que se va utilizar aplicaciones libres, por consiguiente, se puede desarrollar y realizar pruebas sin limitantes de licencias.

## OBJETIVOS

### Objetivo General

- Realizar la implementación de un sistema de almacenamiento de la información, monitoreo y control aplicando el internet de las cosas para la automatización de un invernadero.

### Objetivos Específicos

- Diseñar los circuitos electrónicos de acondicionamiento para los sensores de temperatura, humedad y pluviometría.
- Diseñar los circuitos electrónicos de acondicionamiento para los actuadores de calefacción, ventilador y electroválvula
- Configurar el Gateway que permitirá enviar información de los sensores hacia el internet y recibir instrucciones de control para la electroválvula, ventilador y calefacción desde el internet.
- Realizar la administración, almacenamiento y procesamiento de la información de los sensores y actuadores en internet.
- Crear una página web para controlar y monitorear los dispositivos IoT.
- Diseñar una aplicación para celulares con sistema operativo Android para acceder, monitorear y controlar el invernadero remotamente.
- Implementar el prototipo del sistema.

## **CAPÍTULO I**

### **1. FUNDAMENTACIÓN TEÓRICA**

#### **1.1. INVERNADERO**

Un hecho muy significativo en la historia mundial de la agricultura fue la necesidad de proteger los cultivos de condiciones climáticas adversas, así como también de aumentar la producción durante todo el año y la aplicación de una producción integrada de abordaje de gestión de la protección para un mejor control de plagas y enfermedades.

Un invernadero tiene la especial característica que es de proveer un microclima manteniendo y controlando la temperatura óptima para el crecimiento de los cultivos, en tiempo de producción permite extender las cosechas por épocas de escasas programando con antelación, la construcción de los invernaderos varía de acuerdo a los materiales utilizados, al espacio del suelo, el tipo de cultivo, el tamaño y del entorno socioeconómico.

La producción de cultivos de invernadero es ahora una realidad creciente en todo el mundo con un estimado de 405 000 ha de invernaderos repartidos por todos los continentes.

El grado de sofisticación y tecnología depende también de las condiciones climáticas locales.

##### **1.1.1. TIPOS DE INVERNADEROS**

Existen algunas tipos de invernaderos como el tipo túnel, capilla. En la Figura 1 se muestra el invernadero tipo túnel, este tipo de invernadero soporta grandes velocidades del viento así como también tiene un alto grado de luminosidad, son pequeños denotando una desventaja.

El aire caliente tiende a refrigerarse otorgando así la inversión térmica.



Figura 1. Invernadero tipo Túnel.  
Fuente: Autor.

## **1.2. CULTIVO DE SEMILLAS**

Existe en el mercado gran variedad de semillas para ser cultivadas, comúnmente en los invernaderos se cultiva col, lechuga, rábano ya que el producto es de fácil crecimiento y la altura de la planta es pequeña.

### **1.2.1. SIEMBRA**

La siembra se lo realiza de diferentes maneras ya sea utilizando semilleros directamente al suelo, se utiliza compost para un mejor crecimiento de la planta.

### **1.2.2. NIVELES DE TEMPERATURA Y HUMEDAD RELATIVA.**

Los niveles de temperatura y humedad relativa para el cultivo de lechuga en el invernadero tipo túnel es de 18°C a 20°C que son las variables a utilizar para la implementación, la humedad relativa varía entre 60% y 80% (Infoagro Systems, 2016).

### **1.2.3. EFICIENCIA DEL REGADÍO**

La eficiencia del regadío se refiere a la productividad del agua por lo que la frecuencia con la que se riega al cultivo, es muy importante como la cantidad de agua.

El regadío varía de acuerdo al cultivo, para el caso de la lechuga es de manera periódica, en época de gran producción la cantidad de agua es de 4.3L/m<sup>2</sup> diarios por metro cuadrado de invernadero, por lo que 1,000m<sup>2</sup> se demandarán 4,300L diarios. Si se toma una porción de drenaje del 30%, esta suma incrementa a 5,600L en 1000m<sup>2</sup> (Velázquez, 2015).

#### **1.2.4. COSECHA**

La cosecha suele ser de 70 a 120 días para la lechuga y según el producto que se vaya a cultivar desde la plantación hasta la recolección.

#### **1.3. PLUVIOMETRÍA**

Es un sistema de medición de la precipitación del agua obteniendo así datos espaciales y temporales, como intensidad, frecuencia, duración.

Las mediciones de precipitación son especialmente sensibles ya que al exponerlos hay parámetros como el agua, viento y la topografía que son factores que al tomar muestras sus resultados varían con cada medición.

##### **1.3.1. LA TASA DE PRECIPITACIÓN**

Es la cantidad de agua líquida que alcanza la superficie del suelo, generalmente de día mm/día o por período de tormenta.

##### **1.3.2. MEDICIÓN DE LA PRECIPITACIÓN**

Existen dos clases de instrumentos para medir la precipitación del agua; los mecánicos: pluviómetros, fluviógrafos, los de cangilómetros, y el sensado remoto: láser, satélites, radares y radiómetros que pueden estar en tierra o en satélites. En la Figura 2 se muestra el pluviómetro de cangilometro compuesto por dos cubos iguales ubicado por debajo de un embudo que forma un sistema basculante.

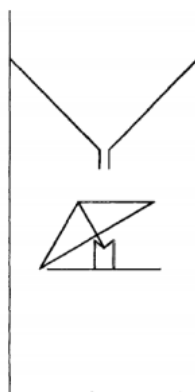


Figura 2. Pluviómetro de Cangilometro.

Fuente: Autor.

El agua que penetra por el embudo cae a uno de los cubos, cuando se llena, el punto de soporte se encuentra fuera de su centro de gravedad haciendo que se vuelque trayendo el otro cubo a la posición de cargar agua, (Renom, 2011).

El volumen del cubo es:  $V = A \Delta h$ , donde A es el área colectora del cubo.

#### 1.4. INTERNET

Es una red que opera con protocolos denominados TCP/IP (Transmission Control Protocol / Internet Protocol), dentro de las redes existentes abarcan un gran número a nivel mundial como las redes públicas, empresariales, privadas para su interconectividad y su comunicación a las mismas utilizan redes WIFI, redes de fibra óptica que están debidamente monitoreadas con equipos robustos como ruteadores, switches que sirven para el direccionamiento del tráfico de la red.

El propósito principal de Internet es facilitar el intercambio de información. Hay muchas herramientas diferentes que se utilizan en Internet para hacer esto posible.

Algunas de las herramientas más comunes incluyen correo electrónico, listas de correo, grupos de noticias, telnet, Gopher, FTP y la World Wide Web.

##### 1.4.1. PROTOCOLO TCP/IP

Las comunicaciones entre equipos de una red se realizan a través de protocolos. El conjunto de protocolos más utilizados son los protocolos TCP / IP. Un protocolo consta de una arquitectura en capas, donde cada capa representa parte de la funcionalidad que se puede llevar a cabo por un protocolo. Cada capa tiene generalmente más de una opción para llevar a cabo su funcionalidad. En la Tabla 1, se muestra las capas del protocolo TCP / IP.

Tabla 1. Capas del Protocolo TCP/IP.

<b>CAPAS TCP/IP</b>	<b>CONCEPTO</b>	<b>SERVICIOS</b>
<b>Aplicación</b>	Incorpora aplicaciones de red estándar	HTTP, FTP, TELNET, DNS
<b>Transporte</b>	Entrega datos de enrutamiento, junto con el mecanismo que permiten conocer el estado de la	TCP, UDP

	transmisión.	
<b>Internet</b>	Es responsable de proporcionar el paquete de datos	IP, ARP, ICMP
<b>Enlace de Datos</b>	Determina la forma en la que los datos deben enrutarse, sea cual sea el tipo de red utilizado.	LLC, PPP, DHCL
<b>Física</b>	Transmite a través de medios físicos, que se conectan al host mediante una red Ethernet, o enlace punto a punto.	ETHENET, FDDI, TOKEN RING

Fuente: Autor.

## 1.5. COMUNICACIÓN INALÁMBRICA

### 1.5.1. WIFI

Es una tecnología de intercambio de datos inalámbrica que sigue los estándares del IEEE y se considera como WIFI a todos los dispositivos inalámbricos que utilizan los estándares IEEE 802.11.

Al principio los protocolos con el estándar IEEE 802.15.4 han sido los más predominantes pero los desarrolladores están reduciendo el consumo de los dispositivos WIFI para competir con tecnologías como Zigbee.

La ventaja que tiene WIFI (wireless fidelity o fidelidad sin cables), es compatible para redes IP, es ampliamente extendido en la tecnología para redes LAN lo cual permite disponer de herramientas más avanzadas y una integración sencilla. (Senel, 2012)

**Frecuencia de funcionamiento.-** Funciona en las bandas de 2.4 GHz y la de 5GHz.

**Rango.-** Usando medios convencionales hasta 100 metros en espacio abierto, 20 en edificios.

**Aplicaciones.-** Routers, Tablets, laptops.

## 1.6. DISPOSITIVOS DE COMUNICACIÓN WIFI

### 1.6.1. ROUTER TL-MR3040

Es un equipo de red inalámbrica móvil que cuando se empareja con un módem USB 3G / 4G, es capaz de transmitir una señal inalámbrica de hasta 150Mbps, creando una red móvil o de entretenimiento para hasta cinco dispositivos para acceder a Internet simultáneamente.

El dispositivo es alimentado por una batería interna 2000mAh. Este dispositivo es fácil de usar, lo que permite a los usuarios configurar rápidamente una red inalámbrica conectada a Internet en el menor tiempo necesario para conectar su enrutador 3G / 4G USB o cable WAN.

Soporta los estándares IEEE 802.11b/g/n, IEEE802.3/3u, proporciona conexión automática y conexión programada en cierto tiempo a Internet, servidor NAT y DHCP incorporado que admite la dirección IP automática y dinámica Distribución de direcciones IP. (TP-LINK TECHNOLOGIES CO., 2012)

#### Panel frontal del router TL-MR3040

En la Figura 3 se muestra los nombres de cada opción del equipo al momento de que se enciende los leds.

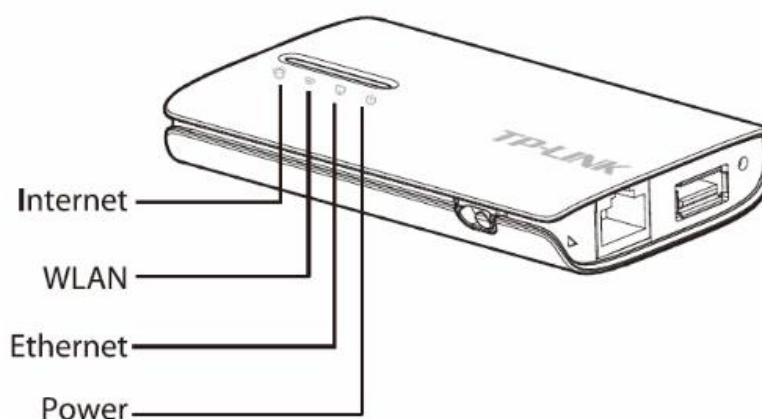






Figura 3. Panel Frontal del Router TL-MR3040  
Fuente: <http://www.tp-link.com>

En la siguiente

Tabla 2 se especifica las características de los leds del Router TL-MR3040



Tabla 2. Características de los Leds del Router TL-MR3040.

<b>Power</b> 	Verde solido	La batería está llena o la fuente de alimentación es normal.
	Sólido (naranja)	La batería se está cargando.
	Sólido (rojo)	La carga de la batería es baja, es necesario cargarla.
	Rojo intermitente	La batería está anormal.
<b>Ethernet</b> 	ON	Un dispositivo está conectado al puerto correspondiente pero no hay actividad.
	Flashing	El puerto Ethernet está transfiriendo datos.
	Off	No hay ningún dispositivo conectado al puerto correspondiente.
<b>WLAN</b> 	ON	La función inalámbrica está activada.
	flashing	Hay datos que se están transfiriendo vía radio.
	Off	La función inalámbrica está deshabilitada.
<b>Internet</b> 	Sólido	El enrutador está conectado a Internet.
	Flashing	Hay datos que se están transfiriendo.

Fuente: Autor.

### Panel Posterior ROUTER TL-MR3040

En la Figura 4 se especifica cada característica del dispositivo.

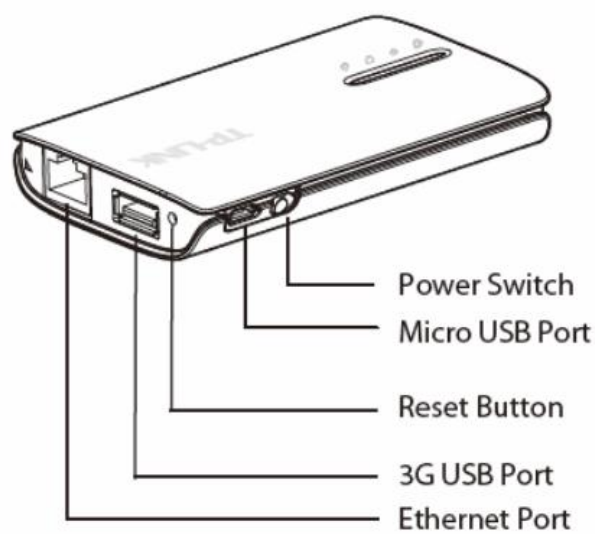


Figura 4. Panel Posterior del Router TL-MR3040

Fuente: <http://www.tp-link.com>

- **Ethernet Port:** Este puerto puede ser puerto LAN o WAN dependiendo del modo de trabajo.
- **USB 3G / 4G Port:** Este puerto se utiliza para conectar un módem 3G / 4G.
- **Reset Button:** Con el Router encendido, se mantiene presionado el botón Reset durante al menos 10 segundos y, a continuación, el enrutador restablecerá la configuración predeterminada.
- **Micro USB Port:** Este puerto se utiliza para conectar el adaptador de alimentación suministrado.
- **Power Switch:** Este interruptor se utiliza para cambiar el estado de alimentación del enrutador.

### 1.7. MÓDULO WIFI ESP8266

El ESP8266 es el nombre de un microprocesador diseñado por Espressif Systems. Un dispositivo ESP8266 puede desempeñar el papel de un punto de acceso, una estación o ambos al mismo tiempo, tiene una conexión de red a Internet y actúa como un puente entre la red inalámbrica y la red TCP / IP más amplia que es Internet. En la Figura 5 se muestra el módulo ESP8266.



Figura 5. Módulo ESP8266.

Fuente: Autor.

En la Tabla 3 se muestra las características y especificaciones de los 8 pines del módulo WIFI ESP8266.

Tabla 3. Especificaciones de los pines del módulo WIFI ESP8266.

Función	Descripción
<b>TXD</b>	Transmisor
<b>RXD</b>	Receptor: utilizar siempre un convertidor para datos. No tolera 5V
<b>CH_PD</b>	Chip Enable: Alta en operación normal 0 - disable 1 - enabled
<b>RST</b>	Reset externo 0 - reset 1 - normal
<b>GPIO 0</b>	Alto en el arranque, Bajo para la actualización del flash.
<b>GPIO 2</b>	Alto en el arranque
<b>VCC</b>	3.3V
<b>GND</b>	Tierra

Fuente: Autor.

### 1.8. MICRO-CONTROLADOR

El ESP8266 está integrado con el micro-controlador de 32 bits Tensilica L106, que incorpora consumo de energía y RSIC de 16 bits. La velocidad del reloj de la CPU es 80MHz. Alcanza un máximo valor de 160MHz. Actualmente, sólo el 20% de MIPS ha sido ocupado por la pila WIFI, el resto puede ser utilizado para la programación de aplicaciones de usuario y desarrollo. (Systems, 2015)

En la Figura 6 se muestra el diagrama de Bloques del micro-controlador ESP8266

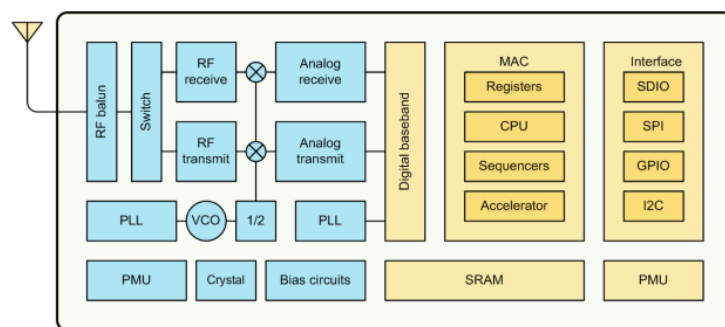


Figura 6. Diagrama de Bloques ESP8266.

Fuente: ESP8266 Data Sheet

Entra las características del micro-controlador se tiene:

- 802.11 b / g / n

- ADC de 10 bits integrado
- Pila de protocolos TCP / IP integrada
- WIFI 2.4 GHz, soporte WPA / WPA2
- Soporta la función Smart Link para dispositivos Android e iOS
- Transmisión de paquetes en <2ms
- Potencia de salida de +20 dBm en modo 802.11b
- Rango de temperatura de funcionamiento -40C ~ 125C

## 1.9. REDES LOCALES WLAN

Wireless Local Area Network por sus siglas en ingles hace referencia para la transmisión de la información entre usuarios de una red local, estos cuentan con antenas WIFI que oscilan de entre los 100 a 500 metros, la velocidad de transmisión que ofrecen estas redes es de 11Mbit/s a 54Mbit/s, en la Figura 7 se muestra la comunicación entre el dispositivo ESP8266 y el Router TL-MR3040 dentro de la conexión.

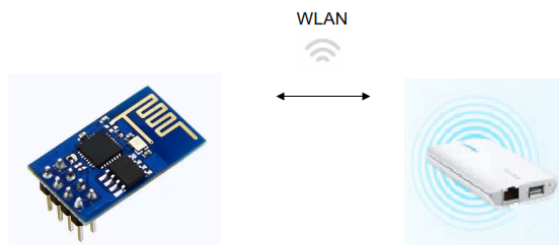


Figura 7. Comunicación Wireless Lan (ESP8266 y TL-MR3040).

Fuente: Autor.

## 1.10. INTERNET OF THINGS “IOT” (INTERNET DE LAS COSAS).

### 1.10.1. DEFINICIÓN

El internet de las cosas, IoT por sus siglas en ingles (Internet Of Things) se puede definir cómo un sistema en el que los objetos cotidianos, que son legibles, reconocibles, localizables, direccionables a través de dispositivo de detección de información sean controlables a través de Internet, independientemente de los medios de comunicación. Estos sensores pueden utilizar varios tipos de

conexiones de área local como RFID, NFC, WIFI, Bluetooth y Zigbee. Los sensores también pueden tener conectividad de área amplia como GSM, GPRS, 3G y LTE.

### 1.10.2. HISTORIA

A principios de los años 2000, Kevin Ashton estaba sentando las bases para lo que se convertiría en Internet de las Cosas (IoT), fue uno de los pioneros que concibió esta noción mientras buscaba información para mejorar su negocio mediante la vinculación de la información RFID a Internet. El concepto era simple pero poderoso. Si todos los objetos en la vida diaria estaban equipados con identificadores y conectividad inalámbrica, estos objetos podrían comunicarse entre sí y ser manejados por ordenadores. La tecnología RFID y de sensores permite a las computadoras observar, identificar y comprender el mundo, sin limitaciones de los datos introducidos por el ser humano, (Ashton, 2009).

### 1.10.3. ARQUITECTURA IoT

Si bien la IOT se estructura en capas, las tecnologías se han clasificado en tres grupos como se observa en la Figura 8.

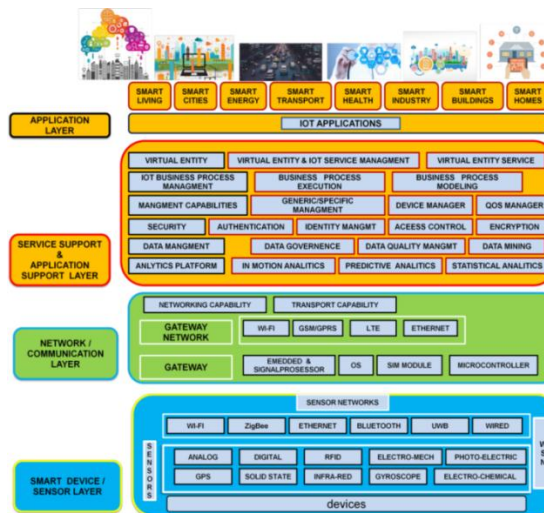


Figura 8. Arquitectura IoT.

Fuente: <http://ijesc.org/Architecture,EnablingTechnologies,Application&FutureChallenges.pdf>

- El primer grupo de tecnologías impacta a los dispositivos, chips de microprocesador, sensores de baja potencia para la energía y la sostenibilidad energética.
- El segundo grupo comprende tecnologías que apoyan el intercambio de redes y abordan los problemas de capacidad y latencia.
- El tercer grupo afecta los servicios de gestión que soportan las aplicaciones de la IOT, tecnologías inteligentes de toma de decisiones, como el análisis predictivo, velocidad de las tecnologías de procesamiento de datos.

#### 1.10.4. MODELO DE REFERENCIA IoT.

Es fundamental entender la expansión del Internet de las Cosas en el terreno de las tecnologías de la información y la comunicación, este modelo de referencia está compuesto por 4 capas como se muestra en la Figura 9.

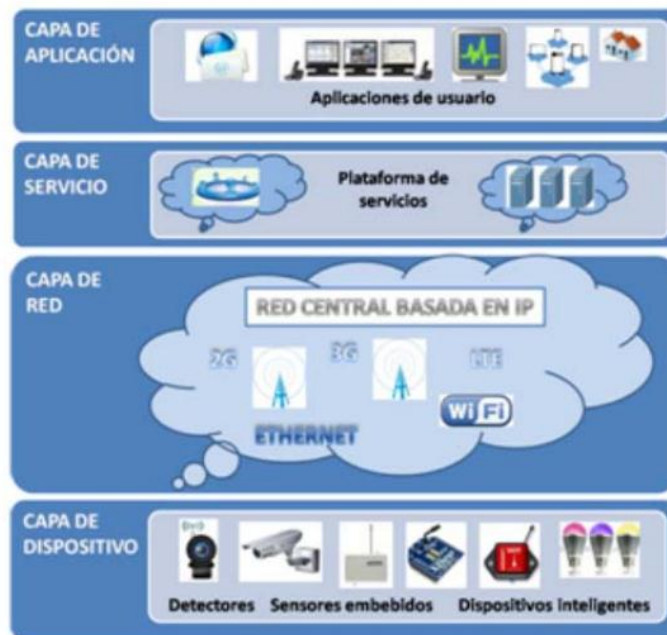


Figura 9. Modelo de Referencia IoT.

Fuente: Autor.

#### Capa de aplicación

La capa de servicio de aplicaciones proporciona servicios de contenido para varios usuarios.

### **Capa de Servicio**

No es otra cosa que la computación en la nube, dentro de esta capa existe dos funciones de soporte general que se utilizan para aplicaciones de procesamiento o almacenamiento de datos y específico que se utilizan para proporcionar diferentes funciones de apoyo a las diferentes aplicaciones del IoT.

### **Capa de Red**

Son las redes cableadas e inalámbricas con sus diferentes tipo de infraestructura, dentro de esta capa existe dos funciones, la de red que entrega funciones de control como la gestión de la movilidad, acceso, autenticación y la de transporte que es la encargada de proporcionar conectividad para el transporte del servicio del IoT

### **Capa de dispositivo**

Utiliza la tecnología inalámbrica para conectar todo con el internet. La tecnología inalámbrica se basa en IEEE 802.15.4, dentro de esta capa existe dos funciones, de gateway y de dispositivo que son infraestructura claves para el funcionamiento de IoT.

## **1.10.5. PROTOCOLOS IoT**

IoT explora protocolos estándar y tecnologías de redes, las principales tecnologías y protocolos de IoT son:

### **NFC y RFID**

RFID (identificación por radiofrecuencia) y NFC (near-field communication) proporcionan opciones sencillas, de baja energía y versátiles para las fichas de identidad y de acceso, las conexiones de arranque y los pagos.

La tecnología RFID emplea transmisores-receptores bidireccionales para identificar y rastrear etiquetas asociadas con objetos.

NFC consiste en protocolos de comunicación para dispositivos electrónicos, típicamente un dispositivo móvil y un dispositivo estándar.

### **Bluetooth de baja energía**

Esta tecnología soporta la baja potencia y la necesidad de uso prolongado de la función IoT mientras explora una tecnología estándar con soporte nativo a través de sistemas.

### **Wireless de baja energía**

Esta tecnología reemplaza el aspecto de mayor consumo de energía de un sistema IoT. Aunque los sensores y otros elementos pueden apagarse durante largos períodos, los enlaces de comunicación (es decir, inalámbricos) deben permanecer en modo de escucha. La energía inalámbrica de baja energía no sólo reduce el consumo, sino que también prolonga la vida del dispositivo mediante un menor uso.

### **Protocolos de Radio**

ZigBee, Z-Wave y Thread son protocolos de radio para crear redes de área privada de baja tasa. Estas tecnologías son de baja potencia, pero ofrecen un alto rendimiento a diferencia de muchas opciones similares. Esto aumenta la potencia de las pequeñas redes de dispositivos locales sin los costes típicos.

### **LTE-A**

LTE-A o LTE Advanced, ofrece una actualización importante a la tecnología LTE, aumentando no sólo su cobertura, sino también reduciendo su latencia y aumentando su rendimiento. Le da a IoT un tremendo poder a través de la expansión de su gama, con sus aplicaciones más significativas siendo vehículo, UAV, y la comunicación similar.

### **WIFI direct**

WIFI-Direct elimina la necesidad de un punto de acceso. Permite conexiones P2P (peer-to-peer) con la velocidad de WIFI, pero con menor latencia. WIFI-Direct elimina un elemento de una red que a menudo lo atasca, y no compromete la velocidad o el rendimiento.

### **1.10.6. APLICACIONES IoT.**

IoT tiene aplicaciones en todas las industrias y mercados, a grupos de usuarios de aquellos que quieren reducir el uso de energía en su hogar a las grandes organizaciones que quieren racionalizar sus operaciones.



**Ingeniería, Industria e Infraestructura.-** Las aplicaciones de IoT en estas áreas incluyen el mejoramiento de la producción, comercialización, prestación de servicios y seguridad. IOT proporciona un medio fuerte de monitorear varios procesos; y transparencia real creando una mayor visibilidad para las oportunidades de mejora.

**Gobierno y seguridad.-** IoT aplicado al gobierno y la seguridad permite mejorar la aplicación de la ley, la defensa, la planificación urbana y la gestión económica. La tecnología completa los vacíos actuales, corrige muchos defectos actuales y amplía el alcance de estos esfuerzos.

**Hogar y Oficina.-** En nuestra vida cotidiana, IoT ofrece una experiencia personalizada desde el hogar a la oficina a las organizaciones con las que frecuentemente se hace negocios. Esto mejora nuestra satisfacción general, mejora la productividad y mejora nuestra salud y seguridad. Por ejemplo, IoT ayudar a personalizar el espacio de oficina para optimizar el trabajo.

**Salud y Medicina.-** IoT empuja hacia el futuro imaginario de la medicina que explota una red altamente integrada de dispositivos médicos sofisticados. Hoy en día, IoT puede mejorar drásticamente la investigación médica, dispositivos, atención y atención de emergencia. La integración de todos los elementos proporciona más precisión, más atención al detalle, reacciones más rápidas a los acontecimientos y mejora constante al tiempo que reduce la típica sobrecarga de la investigación médica y las organizaciones.

**Agricultura.-** Controlar las condiciones micro-climáticas en invernaderos para maximizar la producción de frutas, hortalizas y su calidad, control de los niveles de humedad y temperatura en alfalfa, paja, para prevenir hongos y otros contaminantes microbianos, localización e identificación de animales pastando en lugares abiertos o ubicación en grandes establos.

## **1.11. SENSORES**

Un sensor es un componente, o un subsistema electrónico cuyo propósito es detectar eventos o cambios en su entorno y enviar la información a otro sistema electrónico. Un sensor se utiliza siempre con otros aparatos electrónicos, ya sea tan simple como una luz o tan compleja como una computadora.

### 1.11.1. DHT11

Es un sensor de temperatura y humedad que a su salida tiene una señal digital calibrada, dentro de su estructura contiene un componente de medición de humedad de tipo resistivo y un componente de medición de temperatura NTC, y se conecta a un micro-controlador de alto rendimiento de 8 bits, ofreciendo excelente calidad, respuesta rápida y anti interferencia. En la Figura 10 se muestra el dispositivo DHT11, el voltaje de alimentación es de 3.5v Dc a 5vDc. (UK, 2010).



Figura 10. Sensor DHT11.

Fuente: <https://i.ebayimg.com/images/g/MLsAAOSwOVpXZtgT/s-l300.jpg>

### 1.11.2. SENSOR REED SWITCH

Un reed switch consiste en dos alambres ferro magnético de níquel-hierro y las láminas de contacto colocadas en una cápsula de cristal herméticamente sellada con una abertura entre ellas y en una atmósfera protectora. La cápsula de vidrio se llena con gas inerte para evitar la activación de los contactos.

Las superficies de contacto plateadas con rutenio o rodio están aisladas del entorno exterior, lo que protege a los contactos de la contaminación, en la Figura 11 se muestra el componente electrónico. Operan usando un campo magnético generado por un imán permanente o una bobina que transporta corriente. La fuerza de atracción magnética conduce al cierre de los contactos de lámina. Al retirar el campo magnético, el contacto se abre de nuevo debido a la elasticidad de las láminas.

Hay interruptores de lámina que funcionan con el imán permanente, mientras que hay interruptores de lámina que funcionan con las bobinas que se pueden magnetizar enviando la corriente a través de ella ( EngineersGarage, 2012).

El Voltaje de Alimentación es de 3,3vDC a 5vDC. Su salida es una señal digital.

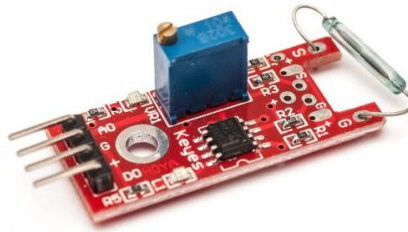


Figura 11. Sensor Reed Switch

Fuente: <http://teslabem.com/media/catalog/1/image/.jpg>

## 1.12. ACTUADORES

### 1.12.1. VENTILADOR

Para el enfriamiento del invernadero, se utilizan 2 ventiladores de 110Vac, 0.21A como se muestra en la Figura 12.



Figura 12. Ventilador.

Fuente: Autor.

### 1.12.2. CALEFACTOR

El calefactor es un sistema que consta de una niquelina, un ventilador de 110Vac que es el encargado de inyectar aire caliente al sistema, tiene un control ON/OFF.

### 1.12.3. ELECTROVÁLVULA

Es un dispositivo electromecánico utilizado para controlar el flujo de líquido o gas. La válvula de solenoide es controlada por la corriente eléctrica, que se pasa a través de una bobina. Cuando la bobina se energiza, se crea un campo magnético, haciendo que un émbolo dentro de la bobina se mueva. Dependiendo del diseño de la válvula, el émbolo abrirá la válvula de solenoide o cerrará la válvula. Cuando la corriente eléctrica se elimina de la bobina, la válvula volverá a su estado des energizado.

En válvulas de solenoide de acción directa, el émbolo abre y cierra directamente un orificio dentro de la válvula. En válvulas accionadas por piloto (también llamadas servo-tipo), el émbolo abre y cierra un orificio piloto. La presión de la línea de entrada, que es conducida a través del orificio piloto, abre y cierra el sello de la válvula. Las electroválvulas permiten la automatización del control de fluidos y gases. Las válvulas de solenoide modernas ofrecen una operación rápida, alta confiabilidad, larga vida útil y diseño compacto, en la Figura 13 se observa el dispositivo.



Figura 13. Electroválvula.

Fuente: Autor.

## CAPITULO II

### 2. METODOLOGÍA

Métodos y técnicas para el desarrollo de la implementación.

Los métodos que se utilizan son:

- **Método Descriptivo.**

Consiste en realizar una exposición narrativa numérica y/o gráfica, lo más detallada y exhaustiva posible de la realidad que se va a implementar.

- **Método Investigativo Experimental.**

Es un tipo de investigación que utiliza experimentos y los principios encontrados en el método científico. Los experimentos pueden ser llevados a cabo en el laboratorio o en la vida real (Al, 2014).

- Obtener los datos con fuentes bibliográficas especializadas en desarrollo de internet de las cosas y aplicaciones con características similares ya implementados en invernaderos.
- Realizar las correspondientes pruebas para determinar el mejor diseño y correcto funcionamiento del mismo.

#### 2.1. TIPO DE ESTUDIO

- **Investigación Analítica.-** La investigación se especifica en realizar un estudio donde el fenómeno se da de manera natural y para obtener información con datos reales de la implementación.
- **Investigación Aplicada.-** Busca la aplicación o utilización de los conocimientos que se adquieren, depende de los avances y resultados de la investigación básica, lo que le interesa al investigador son las consecuencias prácticas.

## 2.2. POBLACIÓN Y MUESTRA

### 2.2.1. POBLACIÓN.

Para determinar una población se toma en cuenta el número de veces que se acciona el ventilador y el calefactor.

### 2.2.2. MUESTRA.

La población es desconocida, ya que no cuenta con un registro identificable por razón que el número de pruebas tiende a ser infinito, por lo tanto se establece una muestra de acuerdo al cálculo con la siguiente fórmula matemática.

$$n = \frac{Z_{\infty}^2 * p * q}{i^2}$$

Componentes de la ecuación:

- $i^2$ : Error considerado que se prevé cometer de un 10% con exponente al cuadrado.  $i = 0.1$
- $p$ : Prevalencia esperada de parámetro a evaluar, el valor tomado es un porcentaje alto de prevalencia  $p = 0.98$  es decir el 98%.
- $Z_{\infty}^2$ : Distribución de Gauss, donde  $Z_{\infty} = 1.96$
- $q$ :  $p - 1$

$$n = \frac{1,96^2 * 0,9 * 1 - 0,9}{0,1^2}$$

$$n = \frac{3,8416 * 0,9 * 0,1}{0,01}$$

$$n = \frac{-0,3457}{0,01}$$

$$n = 34,57$$

$$n = 35$$

### 2.2.3. HIPÓTESIS.

*“La implementación de un sistema de almacenamiento de la información, monitoreo y control aplicando el internet de las cosas permitiría la automatización de un invernadero”.*

### 2.3. OPERACIONALIZACIÓN DE VARIABLES.

#### 2.3.1. VARIABLE INDEPENDIENTE

En la Tabla 4 se indica la operacionalización de la variable independiente.

Tabla 4. Operacionalización de la Variable Independiente.

VARIABLES	CONCEPTO	INDICADORES	INSTRUMENTOS
La implementación de un sistema de almacenamiento de la información, monitoreo y control aplicando el internet de las cosas.	Es la integración de sistemas de comunicación y Servidores a través de Internet.	Programación en HTML, JavaScript, C+	IDE Arduino Google Script Google spreadsheet Google Sites Android Studio
		Documentación	Datasheet de los sensores y actuadores
		Conexión WIFI	Módulo ESP8266 Router TL-MR3040
		Aplicación WEB	HTTP
		Aplicación Android	Celular con Android 5.1 o superior
		Electroválvula	Válvula 110Vca
		Ventilador	110 Vca
		Calefactor	110Vca
Sensor de Temperatura	DHT11		

		Sensor de Humedad	DHT11
		Sensor Magnético	Reed Switch

Fuente: Autor.

### 2.3.2. VARIABLE DEPENDIENTE

En la Tabla 5 se indica la operacionalización de la variable dependiente.

Tabla 5. Operacionalización de la variable dependiente

VARIABLES	CONCEPTO	INDICADORES	INSTRUMENTOS
Permitiría la automatización de un invernadero.	El invernadero debe tener un almacenamiento de la información, monitoreo y control de los actuadores (calefactor, electroválvula y ventilador) y sistema de pluviometría de manera automática	Temperatura Optima	Aplicaciones de Google
		Cantidad de Agua Precipitada	Pluviómetro
		Activación del Ventilador	Smartphone, Página Web de forma automática
		Activación el Calefactor	Smartphone, Página Web de forma automática
		Activación de la electroválvula	Smartphone, Página Web de forma automática

Fuente: Autor.

### 2.4. PROCEDIMIENTOS

Se emplea el método hipotético deductivo, ya que a partir de un problema reconocido se enunciará una hipótesis que se espera ratificar con la práctica.

El método descriptivo que consiste en hacer una exposición narrativa numérica y/o gráfica, lo más exhaustivo posible de la realidad y evaluar ciertas características de una situación particular en uno o más puntos de la investigación.



Analizar los datos reunidos para descubrir así, cual es el mejor diseño a implementar para registrar detalles de la investigación.

Se aplica el método analítico ya que se debe tener un conocimiento claro de cada uno de los elementos y dispositivos electrónicos que forman parte de las etapas del sistema a implementar para observar las causas, la naturaleza y los efectos del mismo.

Método investigativo experimental; que es un tipo de investigación que bien utiliza experimentos y los principios encontrados en el método científico. Los experimentos pueden ser llevados a cabo en el laboratorio o fuera de él.

#### 2.4.1. DESARROLLO DEL PROTOTIPO

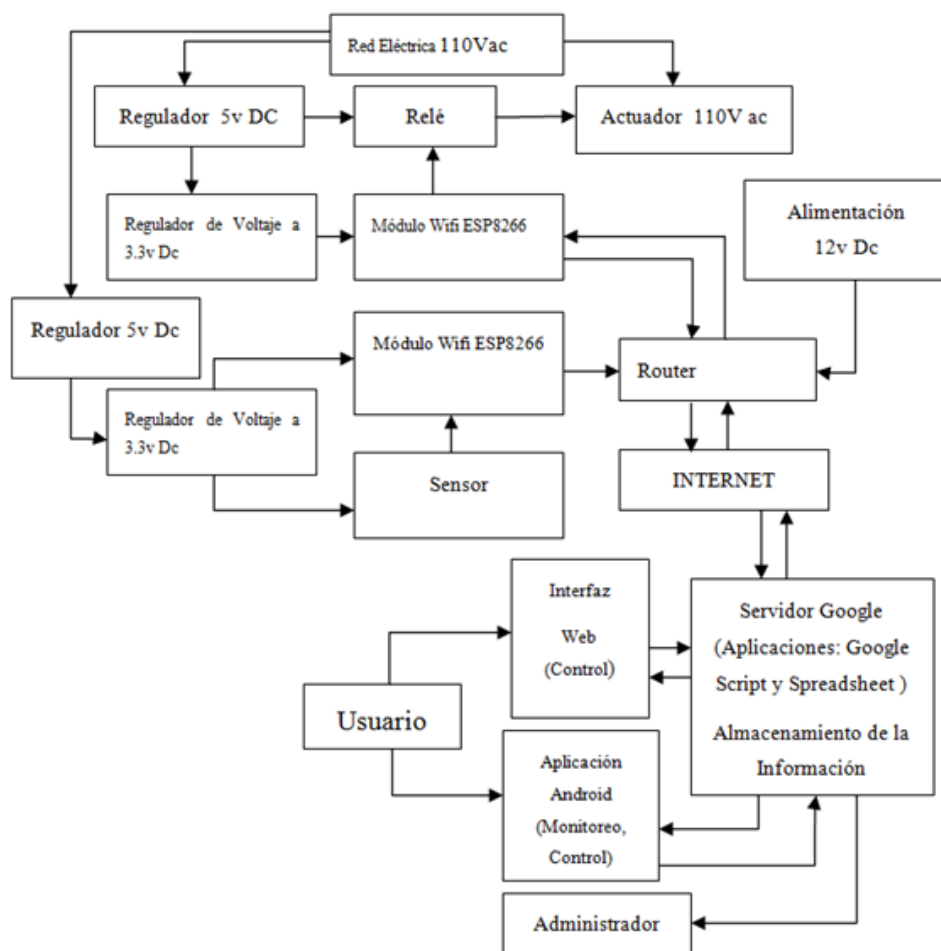


Figura 14. Diagrama de Bloque del Sistema.

Fuente: Autor.

- Selección de sensores y actuadores
- Configuración del gateway.
- Programación en Google App Script y Google spreadsheet
- Diseño y Configuración en Google Spreadsheets. (Hoja de Cálculo de Google) para el almacenamiento de la información y administración del invernadero.
- Programación del módulo ESP8266 para los sensores y actuadores
- Creación de la Pagina Web para el monitoreo y control.
- Creación de la Aplicación Android para celular.

En la Figura 14 se muestra el diagrama de bloques del sistema.

### SELECCIÓN E INSTALACIÓN DE SENSORES Y ACTUADORES EN EL PROTOTIPO DE INVERNADERO.

En la Figura 15 se visualiza el diagrama de los sensores y actuadores que se utilizaron en esta implementación, estos son conectados independientemente a cada una de las diferentes placas electrónicas con el respectivo módulo WIFI Esp8266, la función de este módulo es de recibir y transmitir la información desde los sensores así como también la activación de los actuadores hacia el Router que luego irán a la plataforma de google.

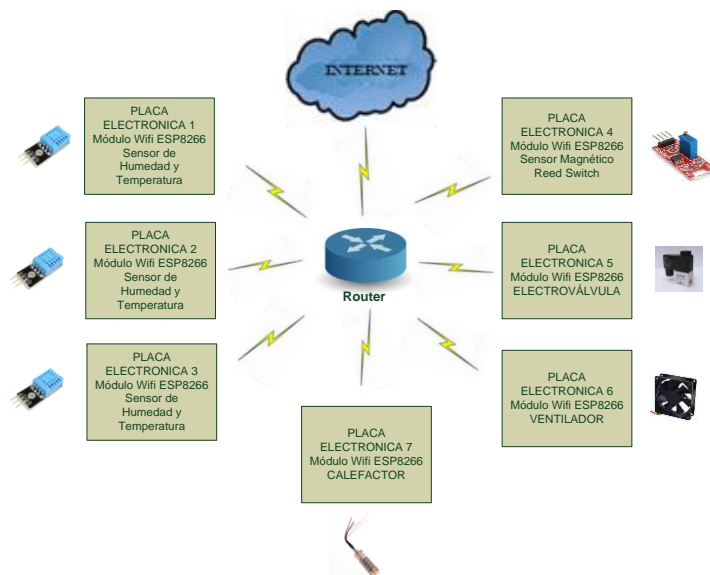


Figura 15. Sensores y Actuadores conectados al Router mediante la red WIFI.

Fuente: Autor.

Los sensores utilizados son:

- Módulos Sensores de Humedad y Temperatura DHT11
- MÓDULO Reed Switch Magnético

Los actuadores utilizados son:

- Ventiladores de 110 Vca
- Calefactorde 110 Vca
- Electroválvula 110 Vca

#### 2.4.1.1. DISEÑO DE LOS CIRCUITOS ELECTRÓNICOS DE ACONDICIONAMIENTO PARA LOS SENSORES TEMPERATURA/HUMEDAD Y PLUVIOMETRIA.

Cada una de placa está conformada por un diagrama electrónico que permite la conexión adecuada del módulo ESP8266, también el módulo DTH11 que es para sensar el ambiente dentro del prototipo de invernadero, módulo Relé para el accionado de los actuadores, y sensor Magnético Reed Switch para el sistema de pluviometría con sus voltajes de alimentación de 3.3 Vcc y 5 Vcc respectivamente para cada uno de los módulos.

Para proceder a desarrollar el diagrama electrónico se identifica las características y especificaciones del módulo WIFI ESP8266. En la Figura 16 se muestra la distribución de los pines del módulo WIFIESP8266.

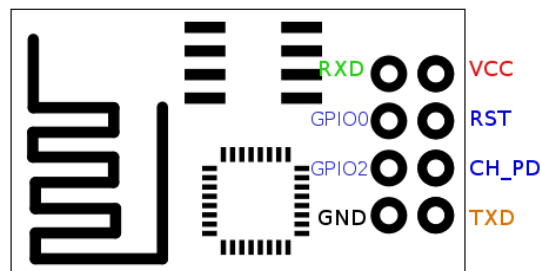


Figura 16. Distribución de los pines para el módulo WIFI ESP8266.

Fuente: Autor.

Se utiliza el regulador de Voltaje LM1117 que ofrece una salida estable de +3.3 Vcc que es indispensable para alimentar al módulo ESP8266. Este regulador de voltaje trabaja perfectamente con un voltaje de entrada de entre 2 Vcc a 15 Vcc, su corriente de salida es de 800 mA.

También 2 condensadores de 10uF para moderar la tensión eléctrica de salida y las fluctuaciones de corriente existentes.

Para alimentar al regulador de Voltaje se utiliza una fuente de alimentación de 5V.

El sensor de temperatura y humedad DHT11 trabaja con un rango de voltaje de 3V a 5V y está conectado a la salida del regular de voltaje. Se realiza las conexiones respectivas para cada uno de los elementos electrónicos y se identifica muy cuidadosamente la polarización de voltaje en cada uno de los dispositivos electrónicos. En la Figura 17 se muestra el diagrama electrónico de conexiones final que se lo realizó en el software Eagle 7.6.0.

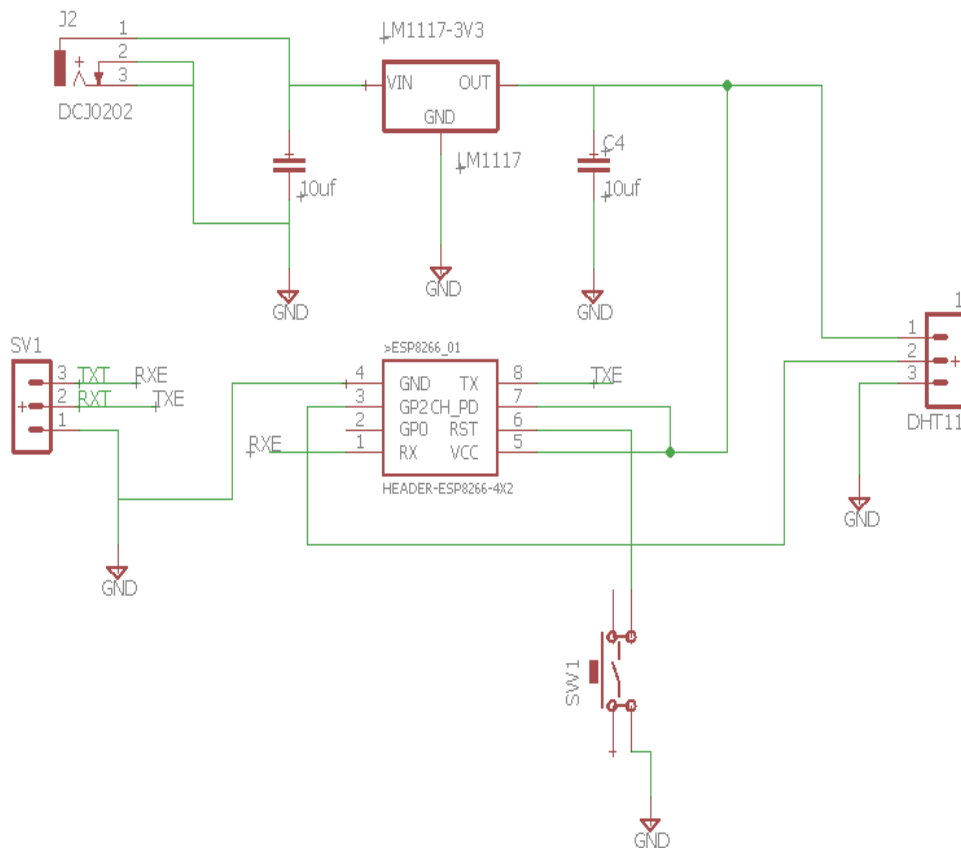


Figura 17. Diagrama electrónico del circuito eléctrico de acondicionamiento para el sensor de temperatura y humedad DHT11.

Fuente: Autor.

Para la realización del diagrama electrónico para su posterior impresión se utilizó el software Eagle 7.6.0 ya que permite realizar diseños de circuitos electrónicos PCB eficaces al detalle, en la Figura 18 se muestra el diseño y diagrama

electrónico PCB del circuito de acondicionamiento para el sensor 1 de Temperatura y Humedad DHT11 implementado en el proyecto.

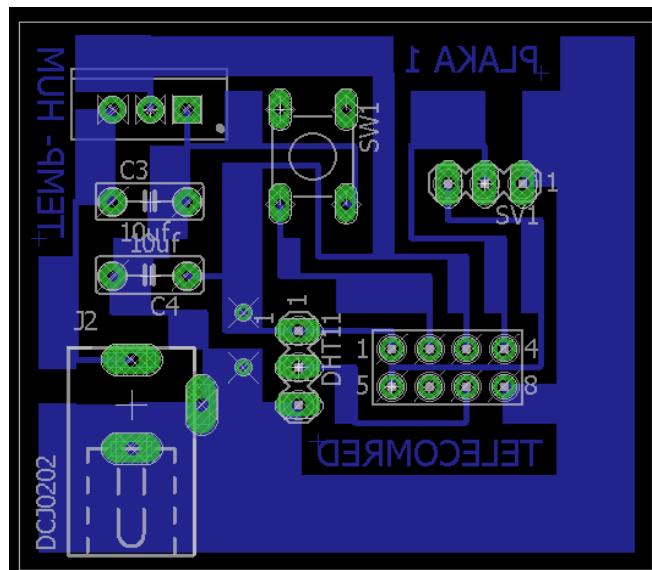


Figura 18. Diseño y diagrama electrónico PCB del circuito de acondicionamiento para el sensor 1 de Temperatura y Humedad DHT11.

Fuente: Autor.

En la Figura 19 se muestra el diseño y diagrama electrónico PCB del circuito de acondicionamiento para el sensor 2 de Temperatura y Humedad DHT11 implementado en el proyecto.

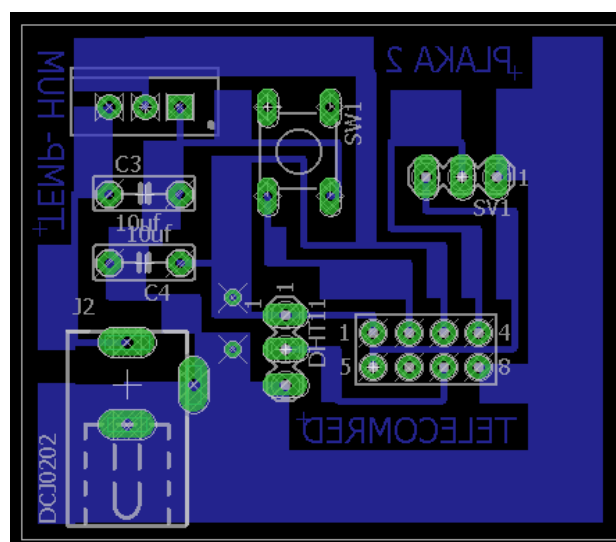


Figura 19. Diseño y diagrama electrónico PCB del circuito de acondicionamiento para el sensor 2 de Temperatura y Humedad DHT11.

Fuente: Autor.

En la Figura 20 se muestra el diseño y diagrama electrónico PCB del circuito de acondicionamiento para el sensor 3 de Temperatura y Humedad DHT11 implementado en el proyecto.

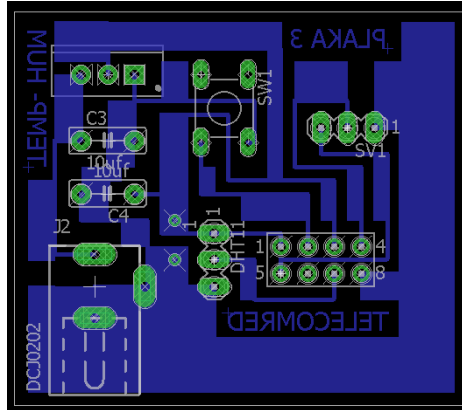


Figura 20. Diseño y diagrama electrónico PCB del circuito de acondicionamiento para el sensor 3 de Temperatura y Humedad DHT11.

Fuente: Autor.

En la Figura 21 se muestra el diseño y diagrama electrónico PCB del circuito de acondicionamiento para el sensor 4 de Pluviometría implementado en el proyecto.

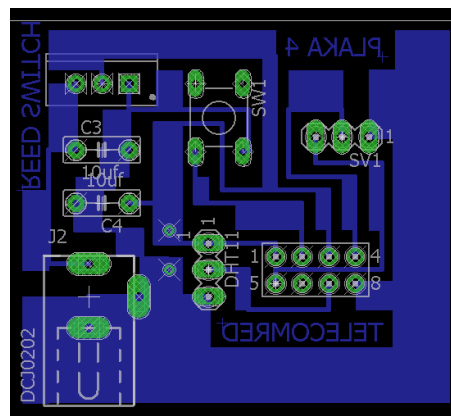


Figura 21. Diseño y Diagrama electrónico PCB del circuito de acondicionamiento para el sensor 4 de Pluviometría.

Fuente: Autor.

Obteniendo el diseño de las diferentes placas electrónicas se procede con la impresión a laser del diagrama electrónico en papel termo transferible para que mediante el calor se transfiera la placa de cobre el diagrama electrónico. Se dimensiona la posición del circuito impreso en la placa, se coloca cinta adhesiva para evitar que se mueva.

Con la ayuda de una plancha como se muestra en la Figura 22 se procede a la transferencia del circuito electrónico, el tiempo expuesto no debe superar los 3 min, la temperatura ideal es entre los 100°C a 160°C, ya que si existe un sobrecalentamiento al exceder la temperatura se corre el riesgo de levantar la fina lámina de cobre existente en la placa.



Figura 22. Transferencia del diagrama electrónico a la placa de cobre.

Fuente: Autor.

El siguiente paso es colocar en un recipiente con agua la placa para que el papel termotransferible se enfríe y se desprenda fácilmente y quede visible el diseño de las pistas electrónicas en la placa de cobre como se muestra en la Figura 23, luego de estar aproximadamente unos 20 min sumergido en agua, se retira y se quita el papel como se muestra en la Figura 24.

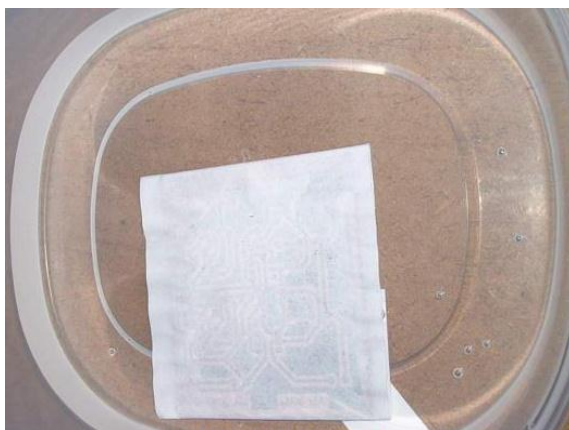


Figura 23. Sumergida en Agua.

Fuente: Autor.

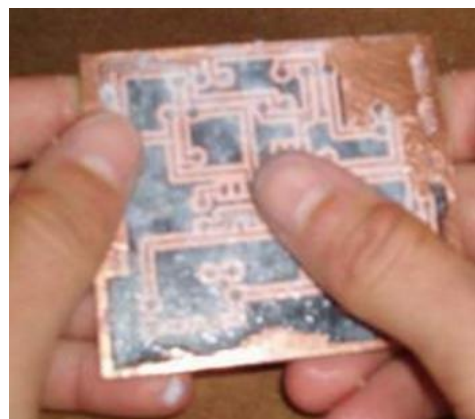


Figura 24. Quitado del Papel de la placa.

Fuente: Autor.



Figura 25. Cloruro Férrico.

Fuente: Autor.



Figura 26. Colocación de la placa en el Acido.

Fuente: Autor.

Con la ayuda de otro recipiente se introduce cloruro férrico se disuelve en agua como se muestra en la Figura 25, a continuación se sumerge las plaquetas de los circuitos impresos en el recipiente con la solución ácida formada como se muestra en la Figura 26. Con este proceso se consigue disolver todo el cobre de la plaqueta permitiendo así que las pistas y todo el circuito sea de acuerdo al diseño realizado anteriormente, en la Figura 27 se muestra la placa cortada y lijada previamente por residuos existentes dejando un acabado perfecto.

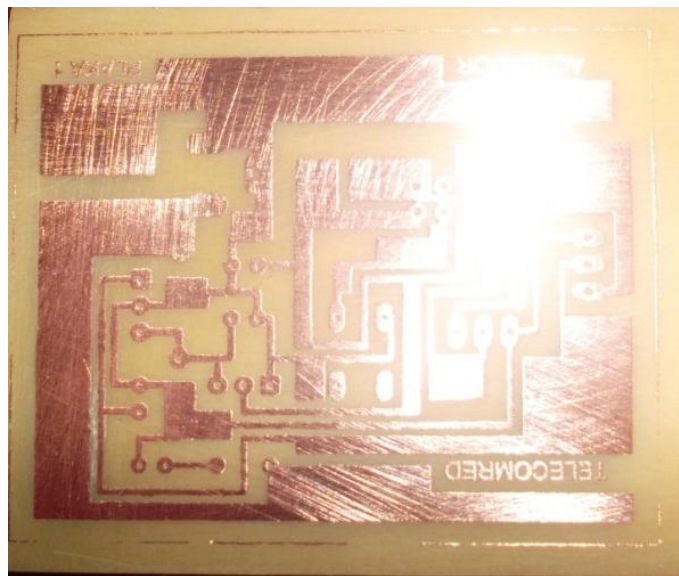


Figura 27. Placa Cortada y Lijada.

Fuente: Autor.

Luego se procede a marcar con la punta de un clavo y con un leve golpe de martillo en cada uno de los puntos de conexión de la pista, esto sirve para realizar el perforado con un taladro que acepte brocas de 1mm.



A continuación se hace el montaje de cada uno de los componentes electrónicos a la placa para su posterior soldado con estaño y cautín. En la Figura 28 se muestra los dispositivos electrónicos que van a ir sobre la placa de los sensores de temperatura y humedad relativa DTH11 y en la Figura 29 se muestra la placa electrónica terminada del circuito de acondicionamiento para el sensor de Temperatura y Humedad DHT11.

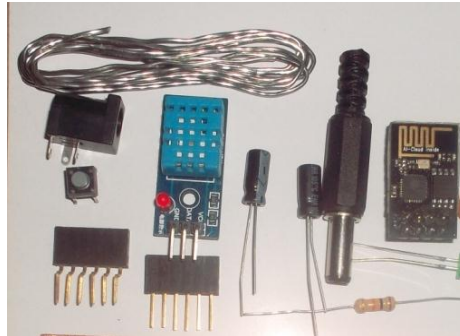


Figura 28. Dispositivos Electrónicos.

Fuente: Autor.

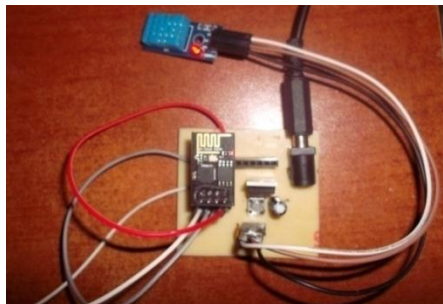


Figura 29. Placa electrónica del circuito de acondicionamiento para el sensor 1 de Temperatura y Humedad DHT11.

Fuente: Autor.

En la Figura 30 se muestra la placa electrónica del circuito 2 de acondicionamiento para el sensor de Temperatura y Humedad DHT11.

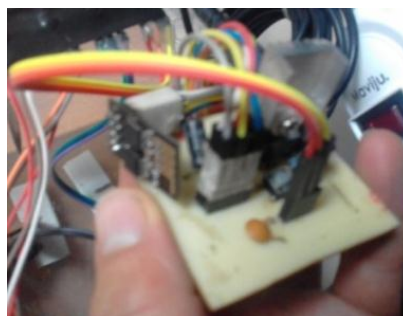


Figura 30. Placa electrónica del circuito de acondicionamiento para el sensor 2 de Temperatura y Humedad DHT11.

Fuente: Autor.

En la Figura 31 se muestra la placa electrónica terminada del circuito 3 de acondicionamiento para el sensor de Temperatura y Humedad DHT11.

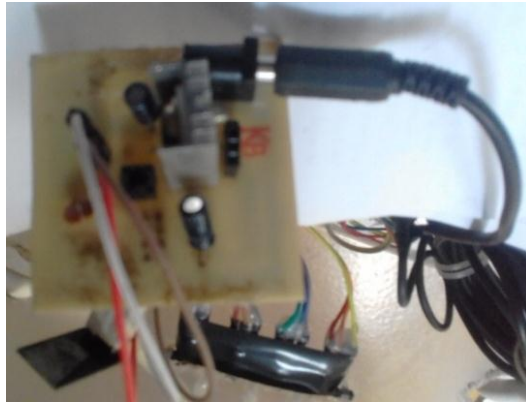


Figura 31. Placa electrónica del circuito de acondicionamiento para el sensor 3 de Temperatura y Humedad DHT11.

Fuente: Autor.

En la Figura 32 se muestra la placa electrónica terminada del circuito 4 de acondicionamiento para el sensor Pluviómetro.

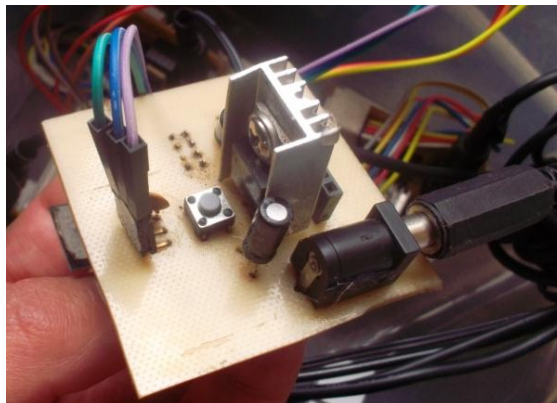


Figura 32. Placa electrónica del circuito de acondicionamiento para el sensor 4 de Pluviómetro.

Fuente: Autor.

El sistema de pluviometría es un sensor de precipitación, en otras palabras un equipo que mide el volumen y/o la tasa de precipitación de lluvia. El funcionamiento se basa en captar una zona de muestreo proporcional, es decir, el área de captura censada es un porcentaje de área dada. Por cada 1mm de lluvia en un área de 1 metro cuadrado tendrá 1 litro de agua,  $100\text{ cm} \times 100\text{ cm} \times 0,1\text{ cm} = 1000\text{ cm}^3$ .

Hay muchos pluviómetros que varían en especial la forma de registrar las precipitaciones. Los pluviómetros más comunes usan un cilindro de medición para

recoger y registrar la cantidad de lluvia en un período determinado. Este pluviómetro hace referencia un sistema basculante que tiene un embudo que recolecta el agua lluvia y la envía por gravedad a una bascula compuesta de dos recipientes, uno de los recipientes se ubica debajo del embudo donde se deposita el agua lluvia que cae, cuando esta se llena, la báscula es obligada a eliminar el agua por efecto del peso de la misma, lo cual implica que el segundo recipiente de la báscula se posicione bajo el embudo para repetir el procedimiento continuamente.

El movimiento de la báscula es usado para activar un interruptor magnético con la ayuda de un imán, transformando el movimiento de la báscula en una señal de impulso / electrónico que se envía a un circuito electrónico que procesa y registra este pulso / señal, (cada pulso corresponde a un porcentaje de la precipitación de un metro cuadrado).

Para la construcción del equipo de pluviometría se construyó la báscula que tiene 110 mm de longitud y 36 mm de altura (punto más alto). En el centro se colocó un eje de acero como se muestra en la Figura 33.



Figura 33. Báscula del Pluviómetro.

Fuente: Autor.

En el centro de la báscula hay un separador, con 30 mm de base, 33 mm de altura y 25 mm de ancho (distancia de los laterales de la báscula) como se muestra en la Figura 34.



Figura 34. Separador en el centro de la Báscula.

Fuente. Autor.

La báscula se fija a través de 2 placas de aluminio de 85 mm de altura y de 15 mm de ancho, que a su vez se fijan en la base del sistema basculante. El montaje mecánico montado se muestra en la Figura 35. Los tornillos fijados debajo de los extremos de la báscula son para ajustar el balance del conjunto, pues la báscula debe moverse a ambos lados con el mismo volumen de agua.



Figura 35. Montaje Mecánico de la Báscula.

Fuente: Autor..

Para el montaje del colector de precipitación se usa un embudo de plástico con una boca de 110 mm, y una botella plástica de 255 mm de altura con un radio de 50 mm.



Figura 36. Equipo de Pluviometría.

Fuente: Autor.

El sistema basculante se coloca dentro de la botella plástica y se fija en los laterales de la base con tornillos, se coloca el embudo sujetando con silicona, el equipo de pluviometría se muestra en la Figura 36.

El sensor Reed Switch se coloca a un lado de la placa de fijación del sistema basculante que mediante el imán colocado en el punto más alto de la báscula censa la precipitación del agua que se requiere para el prototipo de invernadero.

#### **2.4.1.2. DISEÑO DE LOS CIRCUITOS ELECTRÓNICOS DE ACONDICIONAMIENTO PARA LOS ACTUADORES ELECTROVÁLVULA, VENTILACIÓN Y CALEFACCIÓN.**

Se utiliza el mismo diseño electrónico para cada uno de los circuitos electrónicos de acondicionamiento como se muestra en la Figura 37. Esta placa consta de dos reguladores de voltaje fijos que son alimentados por una fuente externa de los cuales el regulador L7805 tiene un voltaje de salida de 5 Vcc para que polarice al módulo relé.

El regulador LM1117 tiene un voltaje de salida de 3.3 Vcc necesario para polarizar al módulo WIFI ESP8266, el diodo 1n4004 sirve para limitar el exceso de corriente cuando haya un sobre voltaje que pueda existir en la red eléctrica, seguido por los cálculos de la resistencia.

Cálculos:

Ventrada= 5V

V diodo led=3.3V

I=620mA

R1=?

$$R1 = \frac{Ve - Vdl}{I}$$

$$R1 = \frac{5v - 3.3v}{620mA}$$

R1 = 2.74K = 2.7K existente en el mercado.

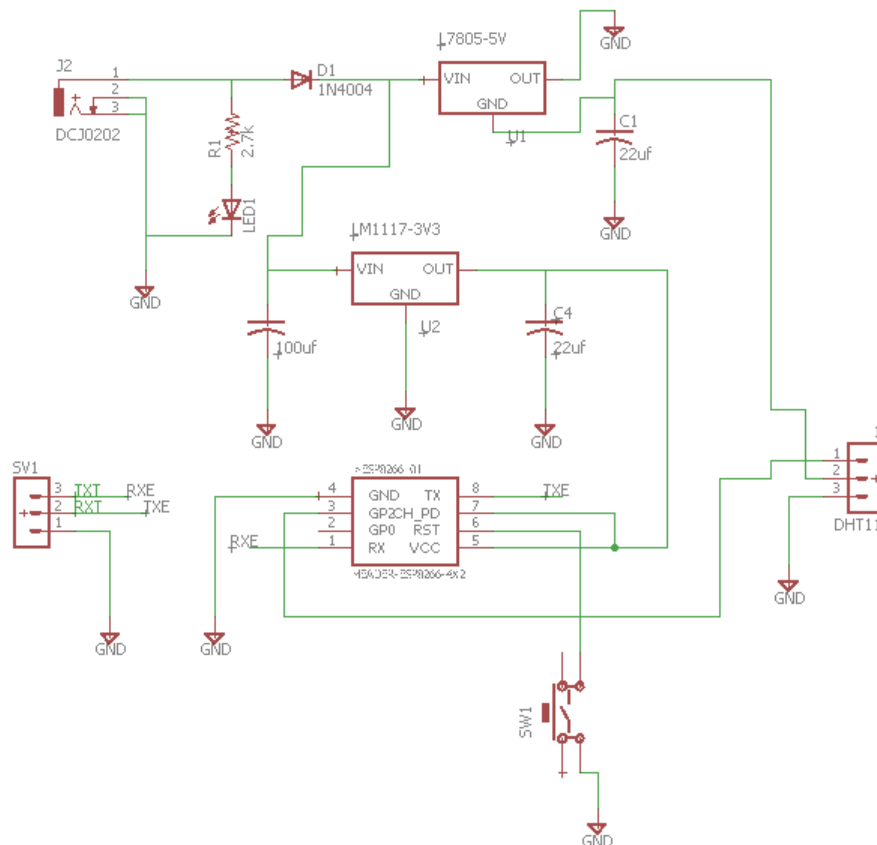


Figura 37. Diseño del Circuito Electrónico de Acondicionamiento Actuadores.

Fuente: Autor.

En la Figura 38 se muestra el diseño y Diagrama Electrónico PCB el Circuito de Acondicionamiento del Actuador 1.

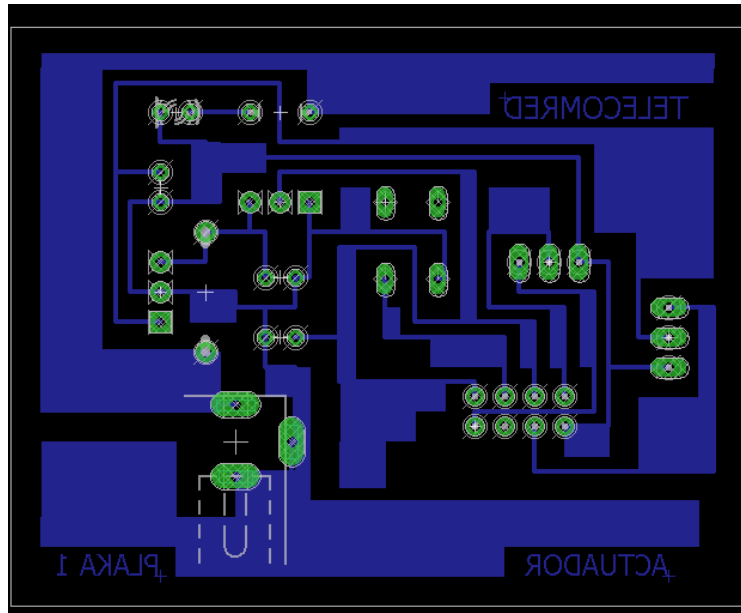


Figura 38. Diseño y Diagrama Electrónico PCB del Circuito de Acondicionamiento Actuador 1.

Fuente: Autor.

En la Figura 39 se muestra el diseño y Diagrama Electrónico PCB el circuito de acondicionamiento del actuador 2.

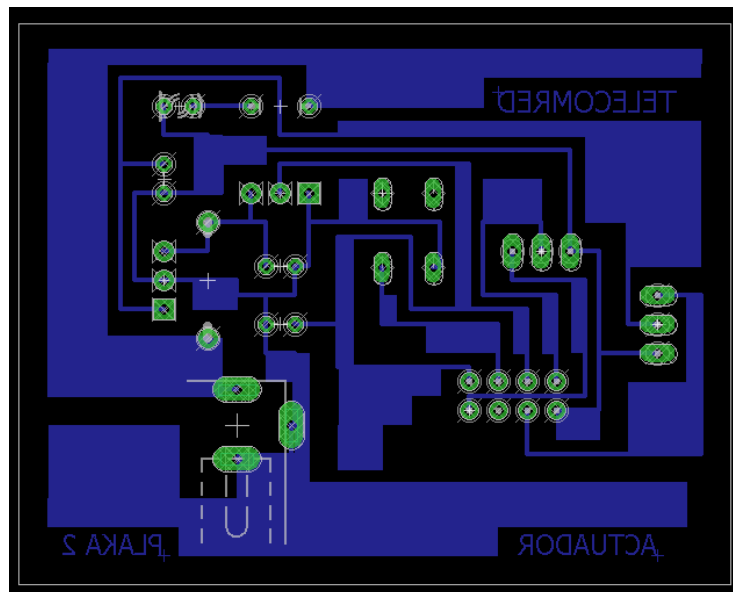


Figura 39. Diseño y Diagrama Electrónico PCB del Circuito de Acondicionamiento Actuador 2.

Fuente: Autor.

En la Figura 40 se muestra el diseño y Diagrama Electrónico PCB el Circuito de Acondicionamiento del Actuador 3.

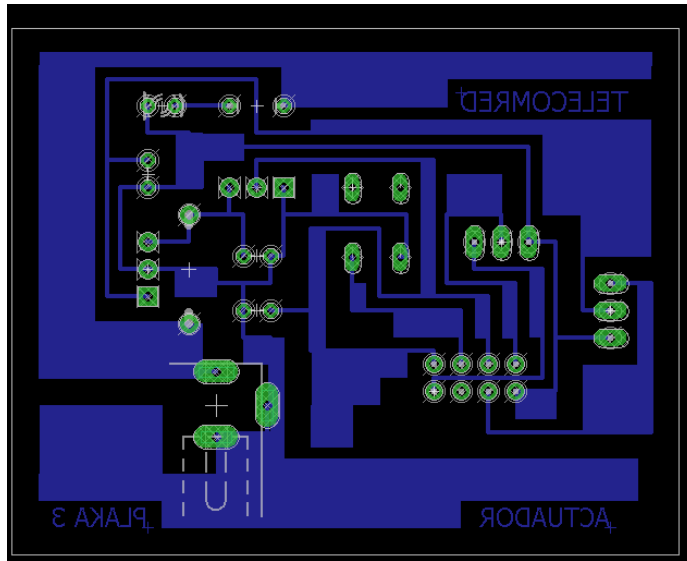


Figura 40. Diseño y Diagrama Electrónico PCB del Circuito de Acondicionamiento Actuador 3.

Fuente: Autor.

Para el proceso de quemado y perforado de la placa se explica en el apartado anterior sección sensores tomando en cuenta todas las recomendaciones ya citadas, en la Figura 41 se muestra la placa cortada y lijada para el circuito de acondicionamiento, y en la Figura 42 se muestra la placa terminada con el montaje de los elementos electrónicos para el actuador 1.

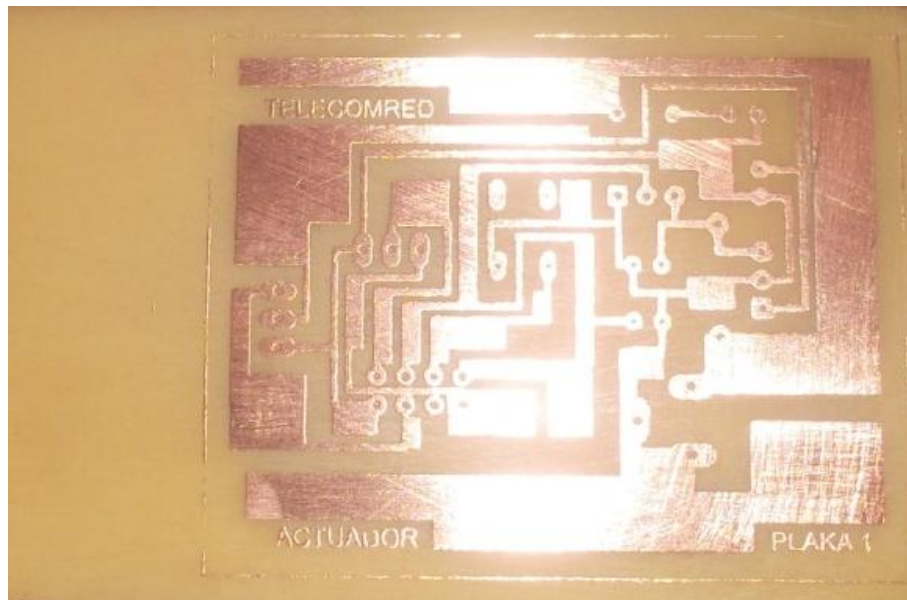


Figura 41. Placa Cortada y Lijada.

Fuente: Autor.



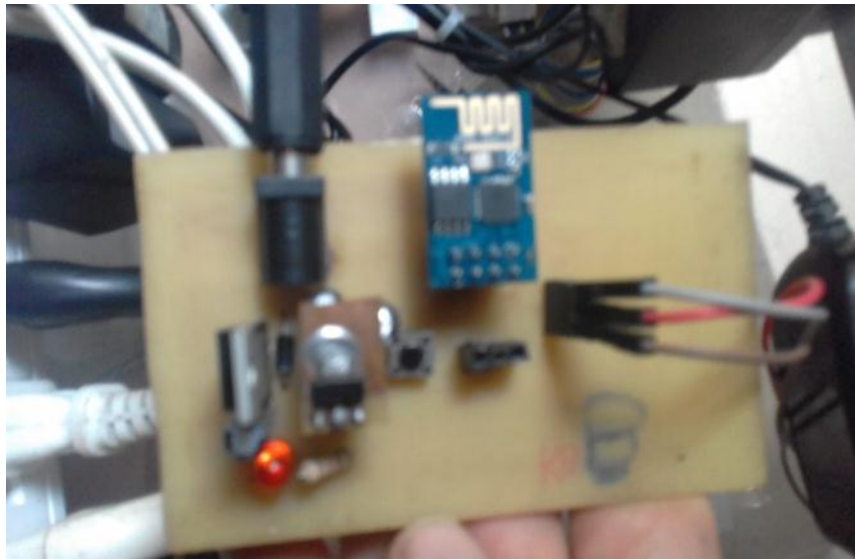


Figura 42. Placa electrónica del circuito de acondicionamiento para el Actuador 1 de la Electroválvula.

Fuente: Autor.

En la Figura 43 se muestra la placa terminada con el montaje de los elementos electrónicos para el actuador 2.

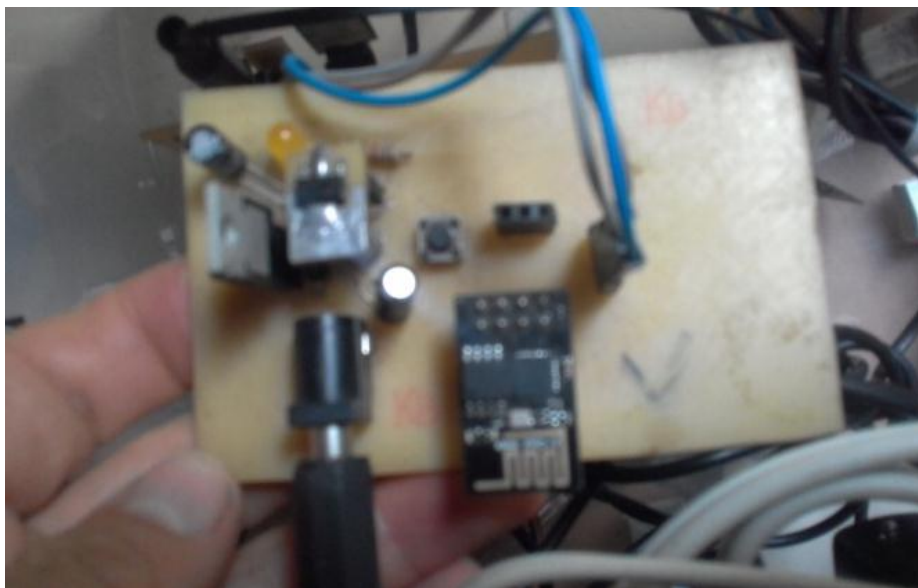


Figura 43. Placa electrónica del circuito de acondicionamiento para el actuador 2 del ventilador.

Fuente: Autor.

En la Figura 44 se muestra la placa terminada con el montaje de los elementos electrónicos para el actuador 3.

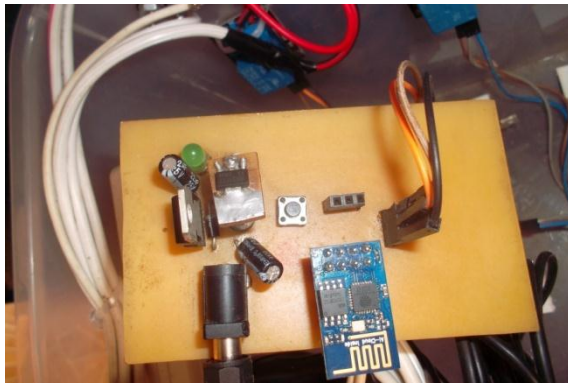


Figura 44. Placa electrónica del circuito de acondicionamiento para el Actuador 3 Calefactor.

Fuente: Autor.

### 2.4.1.3. CONFIGURACIÓN DEL GATEWAY.

En la Figura 45 se ve la parte trasera del Router TP-LINK MR3040 que va a ser el puente entre la red inalámbrica local e internet, mediante el cual se envía datos de los sensores y se recibe datos para los actuadores.



Figura 45. Parte trasera Router TP-LINK MR3040

Fuente: Autor.

#### 2.4.1.3.1. CONFIGURACIÓN DE LA COMPUTADORA

Aquí se toma la conexión de red inalámbrica como un ejemplo. (También se puede ir a Conexión de Área Local para configurar el PC para la conexión de red por cable, y luego configurar el enrutador)

- Se va a Inicio → Configuración → Panel de control → Ver estado de red y tareas → Administrar de conexión de red. Dar clic con el botón secundario en Conexión de red inalámbrica y seleccionar Propiedades como en la Figura 46.

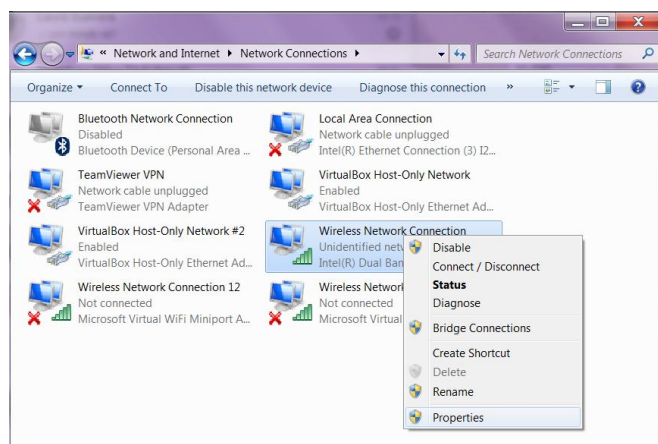


Figura 46. Selección de Propiedades de la conexión de red inalámbrica.

Fuente: Autor.

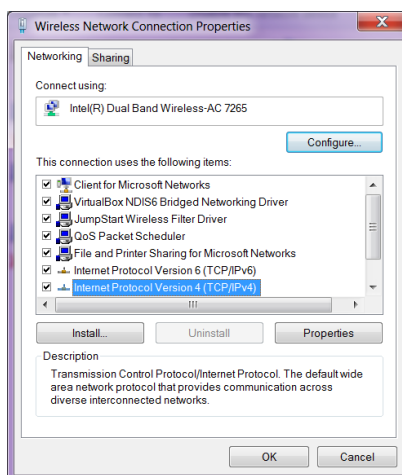


Figura 47. Propiedades de la conexión de red inalámbrica.

Fuente: Autor.

- Hacer doble clic en Protocolo de Internet Versión 4 (TCP / IPv4) ver la Figura 47.
- Seleccionar "Obtener una dirección IP automáticamente" y "Obtener dirección de servidor DNS automáticamente". Haga clic en Aceptar para finalizar la configuración como se muestra en la Figura 48.

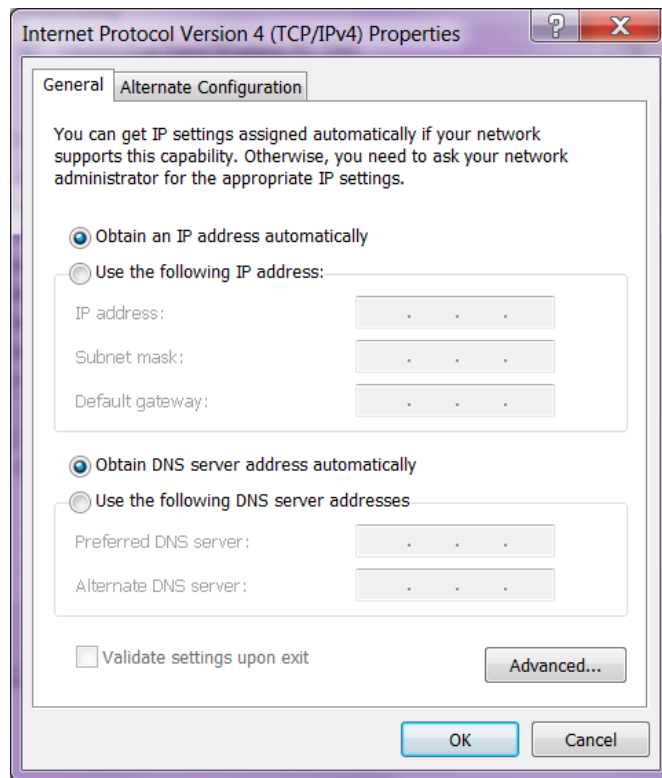


Figura 48. Parámetros de red TCP/IPv4

Fuente: Autor.

#### 2.4.1.3.2. CONEXIÓN A LA RED INALÁMBRICA.



Figura 49. Redes inalámbricas disponibles.

Fuente: Autor.

- Hacer clic en el icono de redes en la esquina inferior derecha.
- Hacer clic en "Actualizar lista de red" como se muestra en la Figura 49, a continuación, seleccione la red y hacer clic en Conectar.

El SSID predeterminado de la red es TP-LINK\_MR3040\_8F0474. (8F0474 es de los últimos seis caracteres de la dirección MAC del enrutador) ver la Figura 45.

- En la Figura 50 se ingresa la contraseña de la red predeterminada que viene en el interior del router TL-MR3040 que se muestra en la Figura 45.

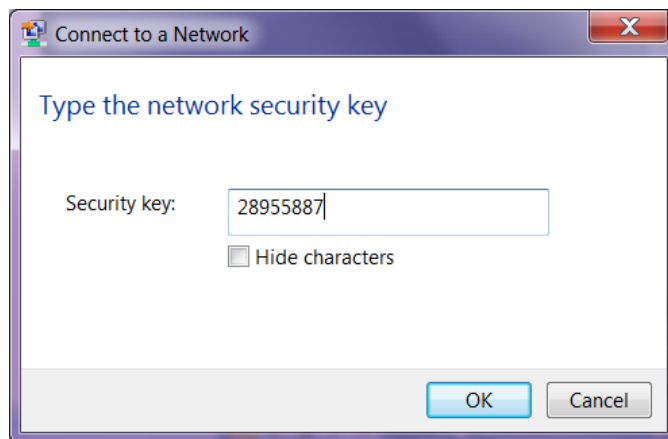


Figura 50. Contraseña de red inalámbrica predeterminada.

Fuente: Autor.

- En la Figura 51 se visualiza que está intentando completar la conexión.

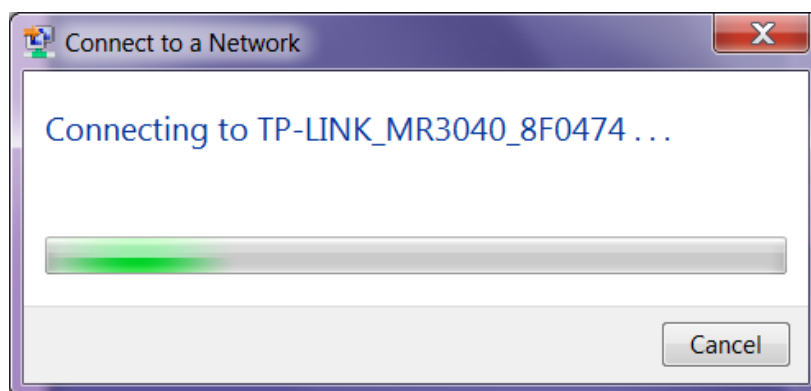


Figura 51. Intentando conectarse a red inalámbrica.

Fuente: Autor.

- En la Figura 52 se puede ver cuando aparece Conected, se ha conectado correctamente a la red inalámbrica.



Figura 52. Red inalámbrica conectada.

Fuente: Autor.

### 2.4.1.3.3. CONFIGURACIÓN DEL ENRUTADOR

Para acceder a la utilidad de configuración, abra un navegador web y escriba la dirección predeterminada 192.168.0.1 en el campo de dirección del navegador como en la Figura 53.



Figura 53. Iniciar sesión en el router.

Fuente: Autor.

Después de un momento, aparecerá una ventana de inicio de sesión como en la Figura 54. Ingresar admin para el Nombre de usuario y Contraseña, ambos en minúsculas. A continuación, haga clic en el botón Iniciar sesión.

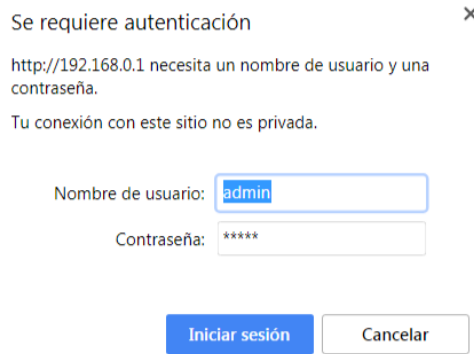


Figura 54. Iniciar sesión en Windows.

Fuente: Autor.

En la Figura 55 se muestra página Status que proporciona la información de estado actual del enrutador. Toda la información es de solo lectura.

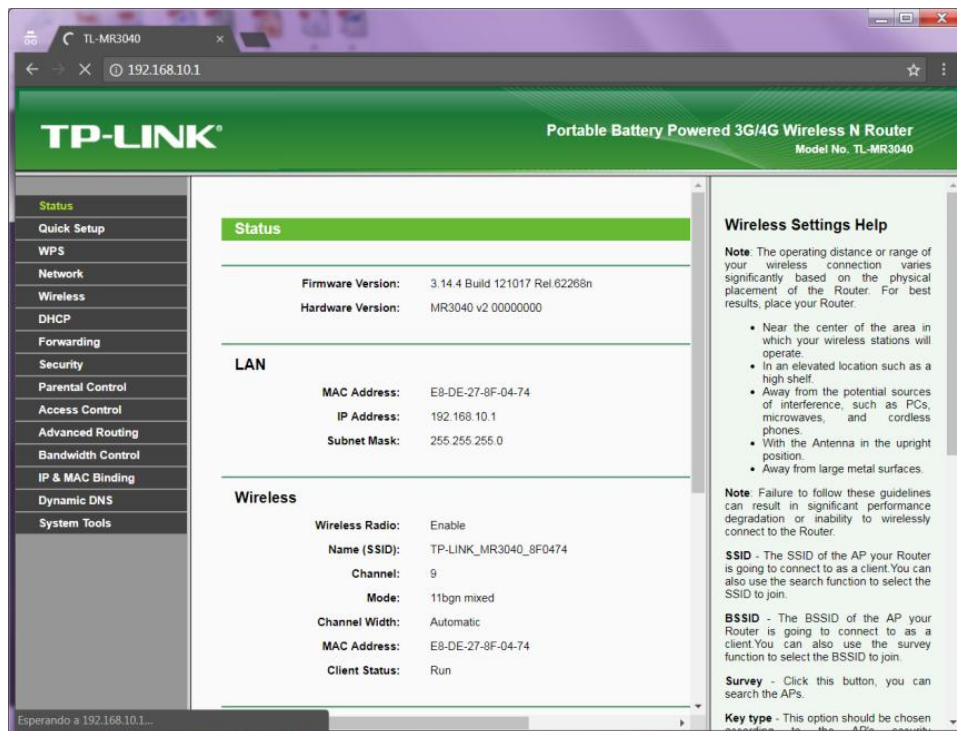


Figura 55. Estado del router.

Fuente: Autor.

## MENÚ NETWORK

Hay tres submenús en el menú Red como se puede ver en la Figura 56: WAN, MAC Clone y LAN. Al hacer clic en cualquiera de ellos se podrá configurar la función correspondiente.

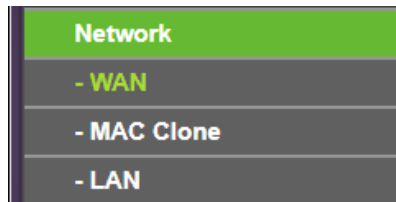


Figura 56. Menú Red.

Fuente: Autor.

## WAN

Elija el menú "Network → WAN" y, a continuación, puede configurar los parámetros IP de la WAN.

En la Figura 57 se muestra en el caso que el ISP proporciona el servicio DHCP, entonces se elija Tipo de IP dinámico y el enrutador automáticamente obtiene los parámetros IP de su ISP.

A screenshot of the WAN configuration page. The page has a green header with the word "WAN". Below the header, there are several configuration fields. The "WAN Connection Type" is set to "Dynamic IP" with a dropdown arrow and a "Detect" button. Below this, the "IP Address" is 192.168.0.107, "Subnet Mask" is 255.255.255.0, and "Default Gateway" is 192.168.0.1. There are "Renew" and "Release" buttons below the gateway. The "MTU Size (in bytes)" is set to 1500, with a note "(The default is 1500, do not change)". There is a checkbox for "Use These DNS Servers" which is checked. Below it, "Primary DNS" is 172.20.0.2 and "Secondary DNS" is 8.8.8.8 (Optional). The "Host Name" is TL-MR3040. At the bottom, there is a checkbox for "Get IP with Unicast DHCP (It is usually not required)" which is unchecked. A "Save" button is at the bottom of the form.

Figura 57. WAN - IP Dinámico.

Fuente: Autor.

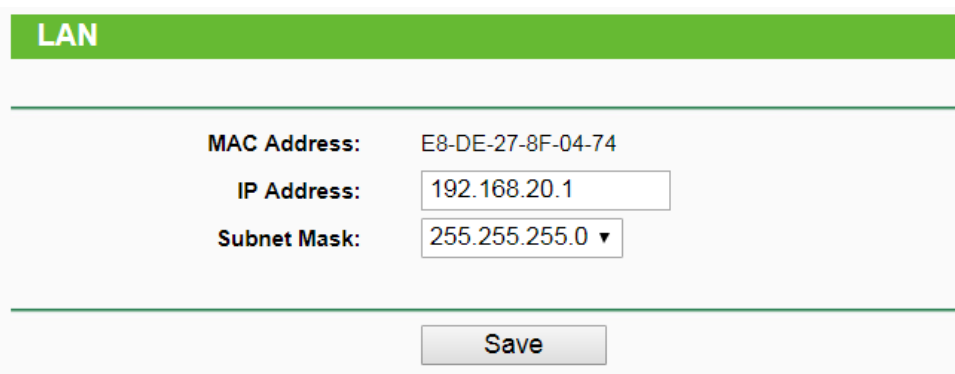
En esta página se muestra los parámetros IP de la WAN asignados dinámicamente por su ISP, incluyendo Dirección IP, máscara de subred, puerta de enlace



predeterminada, etc. Al hacer clic en el botón Renew sirve para renovar los Parámetros IP de su ISP. Al hacer clic en el botón Release sirve para liberar los parámetros IP. MTU Size (Tamaño MTU): el valor normal de MTU (Maximum Transmission Unit) para la mayoría de Ethernet es de 1500 bytes. No se recomienda cambiar el tamaño predeterminado de MTU a menos que lo requiera su ISP. Los servidores DNS serán asignados dinámicamente desde su ISP.

## LAN

Seleccione el menú "Red → LAN" y, a continuación, puede configurar los parámetros IP de la LAN como se ve en la Figura 58.



The screenshot shows a configuration window titled "LAN" with a green header. Below the header, there are three fields for configuration: "MAC Address" with the value "E8-DE-27-8F-04-74", "IP Address" with the value "192.168.20.1" in a text box, and "Subnet Mask" with the value "255.255.255.0" in a dropdown menu. At the bottom of the window is a "Save" button.

Figura 58. Parámetros LAN

Fuente: Autor.

**MAC Address:** La dirección física del enrutador, como se ve desde la LAN. El valor no se puede cambiar.

**IP Address:** Introduzca la dirección IP de su enrutador o restablecerla en notación decimal punteada (Valor predeterminado de fábrica: 192.168.0.1).

**Subnet Maskt:** Un código de dirección que determina el tamaño de la red. Normalmente, utilice 255.255.255.0 como la máscara de subred.

## MENÚ WIRELESS

En la Figura 59 se puede ver que hay cinco submenús en el menú Inalámbrico: Wireless Settings, Wireless Security, Wireless MAC Filtering, Wireless Advanced y Wireless Statistics.

Para este caso específico se utiliza únicamente Wireless Settings y Wireless Security.

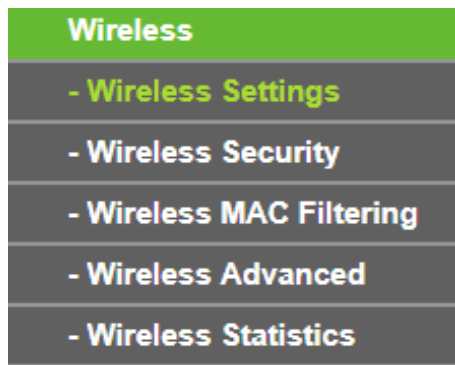


Figura 59. Menú wireless.

Fuente: Autor.

## CONFIGURAR RED INALÁMBRICA.

Elegir el menú "**Wireless**→**Wireless Setting**" y, a continuación, puede configurar los ajustes básicos de la red inalámbrica como se puede ver en la Figura 60.

A screenshot of a web interface titled "Wireless Settings". It is divided into two main sections: "Client Setting" and "AP Setting".  
**Client Setting** includes:

- SSID: [text input]
- BSSID: [text input] Example:00-1D-0F-11-...
- Survey: [button]
- Key type: [dropdown menu] None
- WEP Index: [dropdown menu] 1
- Auth type: [dropdown menu] open
- Password: [password input]

**AP Setting** includes:

- Local SSID: [text input] TP-LINK\_MR3040\_8F0474
- Enable Wireless Router Radio
- Enable SSID Broadcast
- Disable Local Wireless Access

A "Save" button is located at the bottom of the form.

Figura 60. Configuraciones inalámbricas - WISP.

Fuente: Autor.

**SSID:** El SSID del AP al que va a conectarse el enrutador como cliente. También puede utilizar la función de búsqueda para seleccionar el SSID a unirse.

**BSSID:** El BSSID de la AP su router se va a conectar como un cliente. También puede utilizar la función de búsqueda para seleccionar el BSSID a unirse.

**Survey:** Al hacer clic en este botón para examinar el AP que se ejecuta en el canal actual.

**Key type:** Esta opción debe elegirse de acuerdo con la configuración de seguridad del AP. Se recomienda que el tipo de seguridad sea el mismo que el tipo de seguridad de su AP.

**WEP Index:** Esta opción debe elegirse si el tipo de clave es WEP (ASCII) o WEP (HEX). Indica el índice de la clave WEP.

**Auth type:** Esta opción debe elegirse si el tipo de clave es WEP (ASCII) o WEP (HEX). Indica el tipo de autorización del AP raíz.

**Password:** Si el AP de su router va a conectar necesita llenar el espacio en blanco con una contraseña.

**Local SSID:** Se ingresa un valor de hasta 32 caracteres. El mismo nombre (SSID) debe asignarse a todos los dispositivos inalámbricos de su red.

**Enable Wireless Router Radio:** La radio inalámbrica del enrutador puede activarse o desactivarse para permitir el acceso a las estaciones inalámbricas. Si está activada, las estaciones inalámbricas podrán acceder al enrutador.

De lo contrario, las estaciones inalámbricas no podrán acceder al enrutador.

**Enable SSID Broadcast:** Si selecciona la casilla de verificación Enable SSID Broadcast, el enrutador inalámbrico emitirá su nombre (SSID) en el aire.

**Disable Local Wireless Access:** Si selecciona esta casilla, el enrutador inalámbrico desactivará el acceso inalámbrico local, las otras estaciones no podrán acceder al enrutador inalámbricamente.

Al hacer clic en el botón Survey de la Figura 60 y aparecerá la página AP List como se puede ver en la Figura 61. Se busca el SSID del punto de acceso al que desea acceder en este caso Familia Valverde y hacer clic en Conect.

**AP List**

AP Count: 5

ID	BSSID	SSID	Signal	Channel	Security	Choose
1	48-F8-B3-F5-83-DA	Familia Valverde_2.4Ghz	32dB	1	ON	<a href="#">Connect</a>
2	00-15-6D-A9-04-64	Valverde Macao	16dB	1	ON	<a href="#">Connect</a>
3	E8-DE-27-82-3B-3C	DAYANARA__	8dB	9	ON	<a href="#">Connect</a>
4	18-A6-F7-B9-FA-A8	FAMILIA VALVERDE	29dB	9	ON	<a href="#">Connect</a>
5	AC-CF-85-71-77-8C	TATIANA	15dB	10	ON	<a href="#">Connect</a>

Back Refresh

Figura 61. Lista de puntos de acceso.

Fuente: Autor.

El SSID de la red seleccionada se rellenará automáticamente en la casilla correspondiente como se puede ver en la Figura 62, así como también se nombra al SSID local retlawy2k.

**Wireless Settings**

**Client Setting**

SSID: FAMILIA VALVERDE

BSSID: 18-A6-F7-B9-FA-A8 Example:00-1D-0F-11-22-3

Survey

Key type: WPA-PSK/WPA2-PSK

WEP Index: 1

Auth type: open

Password: passwisp

**AP Setting**

Local SSID: retlawy2k

Enable Wireless Router Radio

Enable SSID Broadcast

Disable Local Wireless Access

Save

Figura 62. Parámetros WIFI llenados.

Fuente: Autor.

## SEGURIDAD INALÁMBRICA

Seleccione el menú " **Wireless**→**Wireless Security** ", y luego se configura la red inalámbrica local.

De la Figura 63 se escoge el modo de seguridad inalámbrica WPA/WPA2 Personal→ WPA2-Personal, en **Encryption** se selecciona **AES** y en **Password** se ingresa como contraseña mi WIFI.

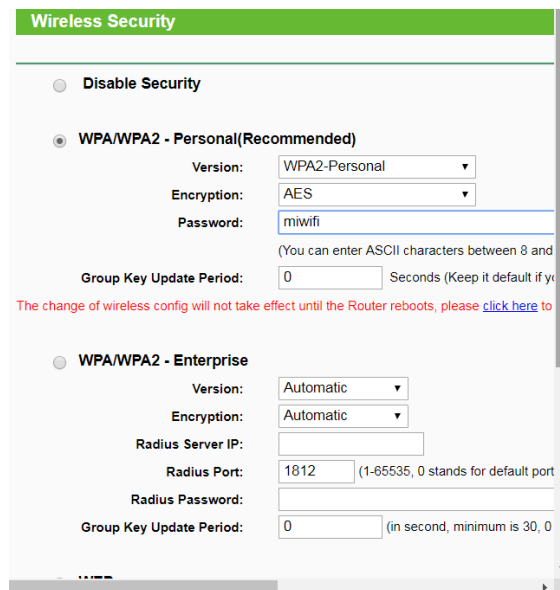


Figura 63. Configuración de la red inalámbrica local.

Fuente: Autor.

Se guarda todo y reinicia el router, como se puede ver en la Figura 64 ya aparece la red creada, dar click y presionar el botón Conectar.



Figura 64. Visualización de red inalámbrica creada.

Fuente: Autor.

En la Figura 65 hay que ingresar la contraseña configurada y dar click en el botón OK.



Figura 65. Ingreso de la contraseña de red inalámbrica.

Fuente: Autor.

En la Figura 66 se puede ver que ya aparece como conectado a la red retlawy2k.



Figura 66. Conectado a la red inalámbrica.

Fuente: Autor.

#### 2.4.1.4. DISEÑO Y CONFIGURACIÓN EN GOOGLE SPREADSHEETS. (HOJA DE CÁLCULO DE GOOGLE) PARA EL ALMACENAMIENTO DE LA INFORMACIÓN, ADMINISTRACIÓN Y CONTROL DEL INVERNADERO.

La hoja de cálculo o Google Spreadsheets es una cuadrícula que funciona con matrices bidimensionales, para recuperar los datos de la hoja de cálculo, se debe tener acceso a la hoja de cálculo donde se almacena la información, obtiene el rango en la hoja de cálculo que contiene los datos y los valores de las celdas. Apps Script facilita el acceso a los datos mediante la lectura de datos estructurados en la hoja de cálculo.

Todas las programaciones almacenadas en google script para cada uno de los sensores y actuadores que se necesita para que el invernadero permita el monitoreo, control y el almacenamiento de la información mediante los sensores que están ubicados estratégicamente dentro del invernadero se utiliza la aplicación de google drive y mediante google spreadsheet u hoja de cálculo se crea la interface que permite hacer lo antes mencionado.

Dentro de la aplicación de google drive en la pestaña **Nuevo** se crea el documento google spreadsheet u hoja de cálculo de google como se muestra en la Figura 67.

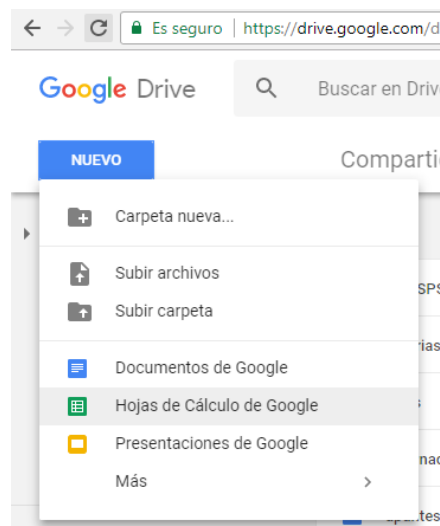


Figura 67. Creación de nuevo documento.

Fuente: Autor.

Una vez abierto la nueva hoja de cálculo se edita con el nombre Invernadero IoT que se encuentra ubicado en la esquina superior izquierda como se muestra en la Figura 68.

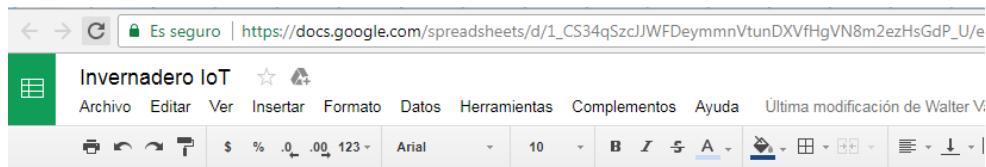


Figura 68. Nombre hoja de Cálculo.

Fuente: Autor.

El objetivo al crear la hoja de cálculo es para enlazar con google script y poder almacenar la información que llega desde los sensores a cada una de las celdas asignadas previamente programadas en google script.

Para la conexión con google app script se necesita el ID de la hoja de cálculo que es único y no se repite en los servidores de google, se encuentra en la url de la hoja de cálculo. En la Figura 69 se muestra el ID para la conexión con las aplicaciones de google app script que es:

*1\_CS34qSzcJJWFDeymmnVtunDXVfHgVN8m2ezHsGdP\_U*

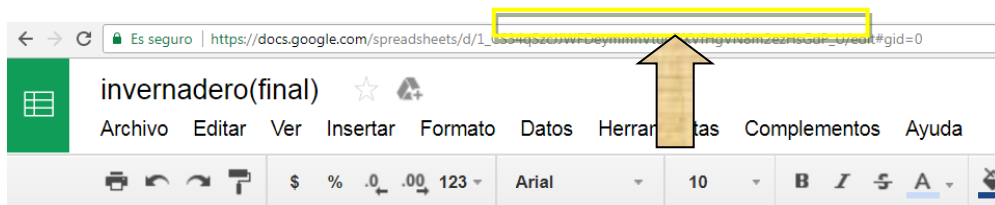


Figura 69. ID de Conexión.

Fuente: Autor.

Cuando se realiza la solicitud POST al URL, la solicitud fallará si no hay ninguna función doPost () escrita para servir la solicitud POST. Por lo tanto hay que verificar ambas solicitudes en la secuencia de comandos de google app script.

Se añade desde la parte inferior izquierda las hojas de cálculo TempHum1, TempHum2, TempHum3, Preci y Control, los nombres de las pestañas se utilizan luego en la programación de los scripts, en la Figura 70 se muestra las hojas de cálculo creadas.



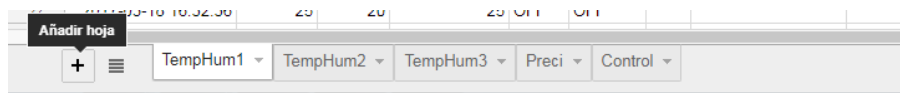


Figura 70. Hojas de cálculo creadas.

Fuente: Autor.

En la pestaña Insertar seleccionar la opción gráfico como se muestra en la Figura 71.

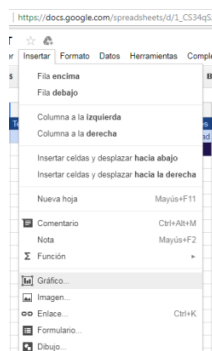


Figura 71. Seleccionar Gráfico.

Fuente: Autor.

Al insertar el gráfico muestra los valores obtenidos para cada uno de los sensores visualizando resultados de temperatura y humedad, google spreadsheet no necesita de “pluggins” del lado del cliente como “flash” o “java” ya que contiene gráficos que el usuario puede modificar constantemente.

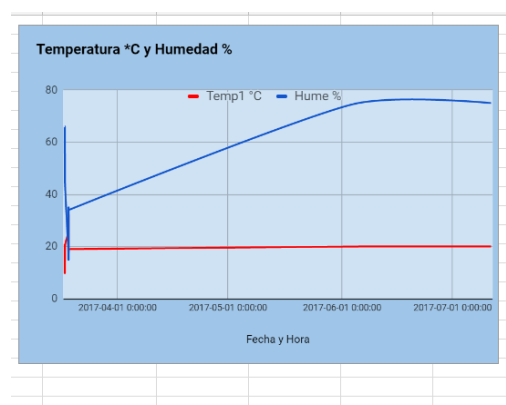


Figura 72. Histórico del Sensor

Fuente: Autor.

En Google AppScript se declara cada una de las variables para que la hoja de cálculo lea los datos obtenidos de los sensores. En la Figura 72 se muestra el

histograma de temperatura y humedad, donde se observa el valor actual de temperatura y humedad censada dentro del invernadero.

Los valores que obtiene el gráfico se actualizan cada 10 segundos de manera que cada sensor irá ejecutándose de acuerdo al siguiente diagrama de flujo como se muestra en la Figura 73.

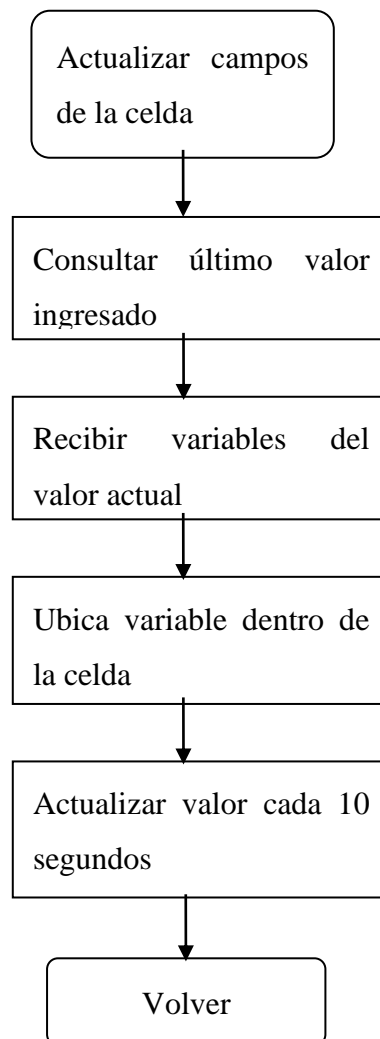


Figura 73. Diagrama Flujo el Sensor.

Fuente: Autor.

En la Figura 74 se muestra los datos obtenidos del sensor por fecha y hora exacta que recibe los valores del invernadero a partir de la celda A4, además muestra el valor de temperatura y humedad en las celdas B4 y C4 respectivamente.

	A	B	C
1	Fecha y Hora	Temp1 °C	Hume %
2			
3			
4	2017-03-17 16:43:37	20	66
5	2017-03-17 16:44:08	20	66
6	2017-03-17 16:45:27	20	66
7	2017-03-17 16:46:39	20	66
8	2017-03-17 16:47:12	10	66
9	2017-03-17 17:52:35	20	45
10	2017-03-18 16:30:34	25	18
11	2017-03-18 16:30:44	25	18
12	2017-03-18 16:30:54	25	19
13	2017-03-18 16:31:04	25	19
14	2017-03-18 16:31:14	25	20
15	2017-03-18 16:31:25	25	21
16	2017-03-18 16:31:36	25	21
17	2017-03-18 16:31:46	25	21
18	2017-03-18 16:31:56	25	21
19	2017-03-18 16:32:06	25	20
20	2017-03-18 16:32:16	25	20
21	2017-03-18 16:32:26	25	20
22	2017-03-18 16:32:36	25	20

Figura 74. Datos Obtenidos del sensor.

Fuente: Autor.

En la hoja de cálculo denominada TempHum1 se visualiza la temperatura promedio de los tres sensores así como también el prendido y apagado de los actuadores como se muestra en la Figura 75.

PROM Temp	CAL	VENT
31	OFF	OFF
31	OFF	ON
24	OFF	OFF
20	OFF	OFF
17	ON	OFF
20	OFF	OFF
22	OFF	OFF
22	OFF	OFF
22	OFF	OFF
22	OFF	OFF
24	OFF	OFF
25	OFF	OFF
25	OFF	OFF
25	OFF	OFF

Figura 75. Datos de Temperatura Promedio.

Fuente: Autor.

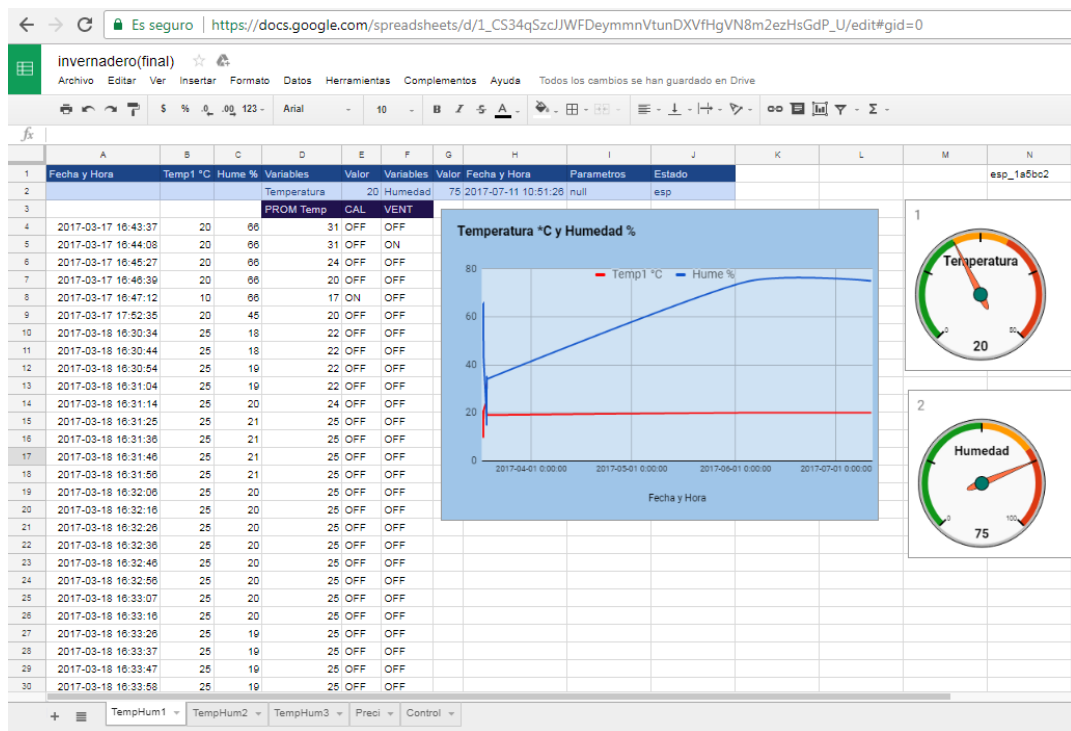


Figura 76. Interfaz de Google Spreadsheet.

Fuente: Autor.

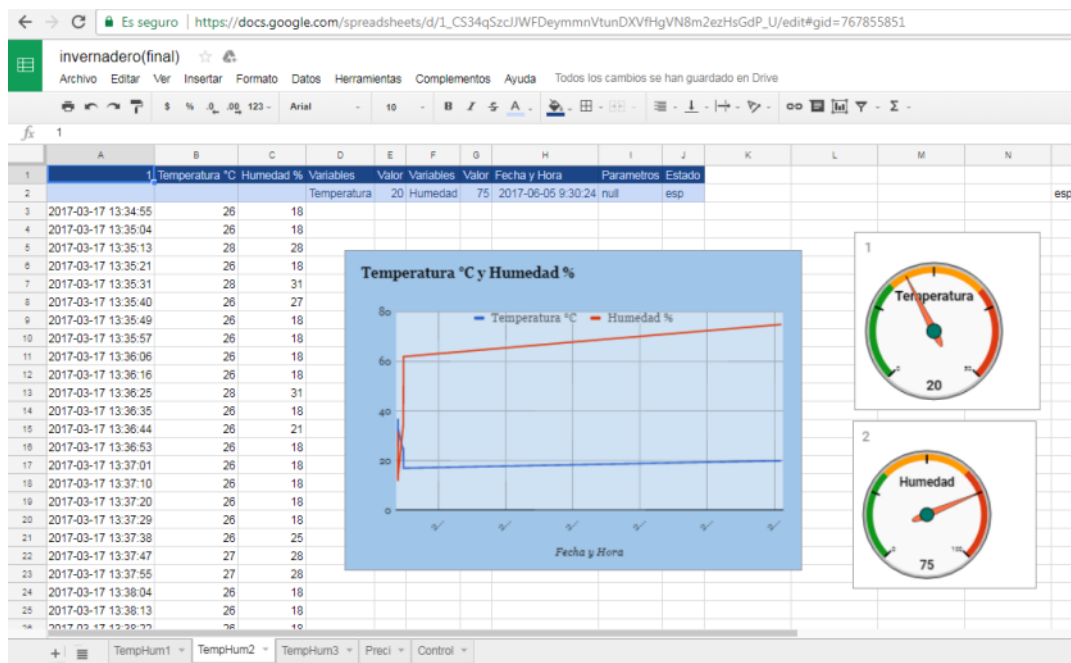


Figura 77. Interfaz de la TempHum 2.

Fuente: Autor.

El resultado del diseño y configuración es la interfaz que se muestra en la Figura 76 donde se observa el valor actual de la variable de temperatura y humedad, la fecha y hora, el promedio de las temperaturas, el estado del módulo ESP8266, y sus gráficas.

Tanto el interfaz de TempHum2, TemHum3, Precipitación se diseña y se configura de idéntica manera ya que todos poseen el valor actual de la variable de temperatura y humedad, la fecha y hora, estado del módulo ESP8266 y sus histogramas.

En la Figura 77 se muestra el interfaz de la hoja de cálculo TempHum2.

En la Figura 78 se muestra el interfaz de la hoja de cálculo TempHum3

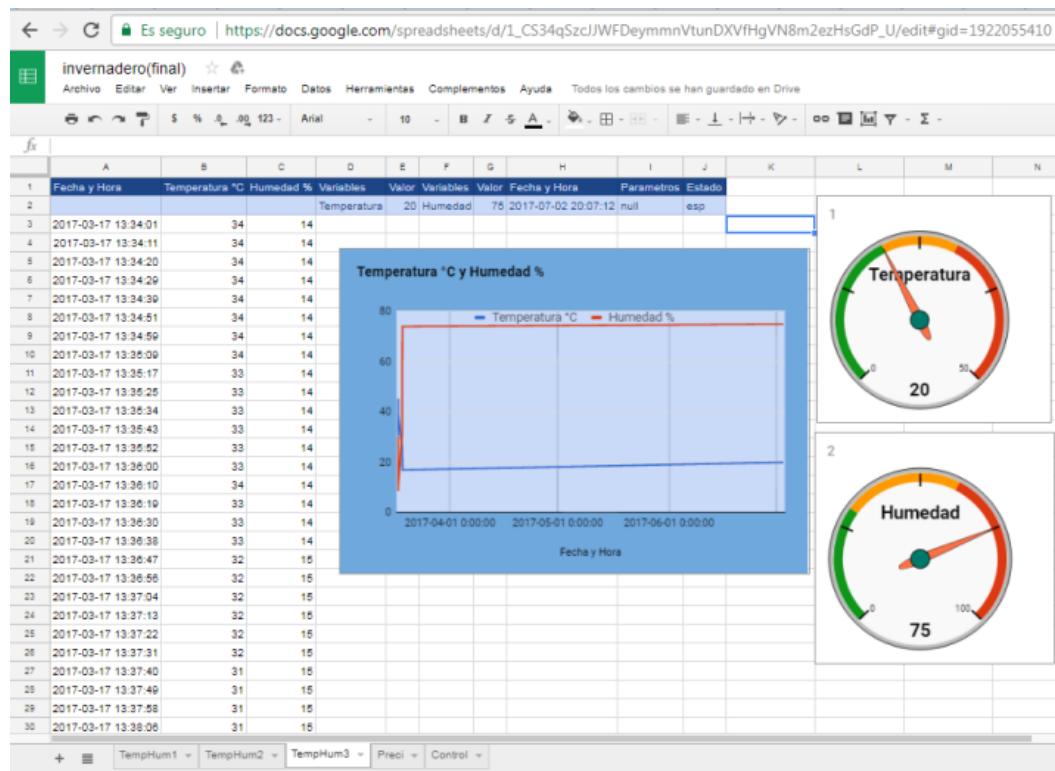


Figura 78. Interfaz de la TempHum 3.

Fuente: Autor.

En la Figura 79 se muestra el interfaz de Precipitación en la hoja de cálculo denominado Preci.

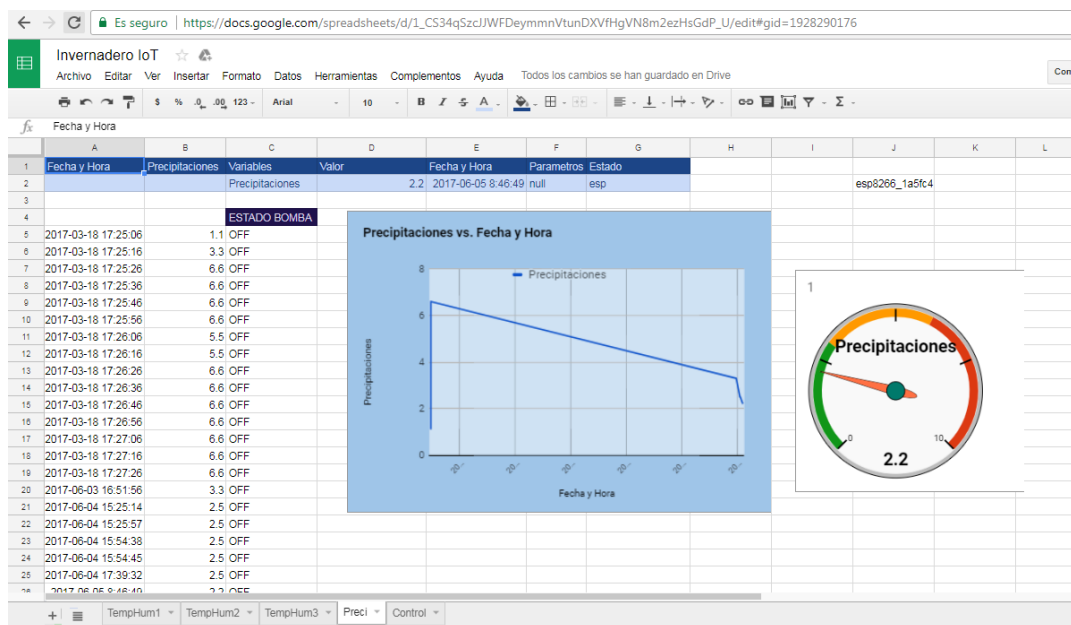


Figura 79. Interfaz de la Precipitación.

Fuente: Autor.

En la hoja de cálculo denominada Control se tiene el acceso para configurar los datos de acuerdo a las especificaciones y características del cultivo a ser sembrado. Estos valores son previamente comparados en la programación de google App script y extraídos a la hoja de cálculo para el procesamiento y ejecución del encendido y apagado de los actuadores de acuerdo al promedio de las temperaturas totales previamente censados, si al rango configurado le excede a la temperatura máxima al momento de sensar el invernadero se prende el ventilador, mientras que si al rango configurado es inferior a la temperatura mínima al momento de sensar el invernadero se prende el calefactor.

Este encendido y apagado de los actuadores se lo realiza de forma manual y automática, en la celda K se digita el número 1 cuando se requiere de forma automática y 0 de forma manual y también al momento de digitar ON/OFF en cada una de las celdas H2, I2, J2 que es la sección de los actuadores.

En la Figura 80 se muestra en las celdas A1, B1, C1 la Temperatura, Temperatura Mínima, Temperatura Máxima respectivamente, y en las celdas A2, B2, C2 se visualiza los datos censados que son también digitados de acuerdo a las características del cultivo a sembrar.

	A	B	C
1	temperatura	temp_min	temp_max
2	20	17	23

Figura 80. Visualización de Temperatura en Celdas.

Fuente: Autor.

En la Figura 81 se muestra en las celdas D1, E1, F1 la Humedad, Humedad Minina, Humedad Máxima respectivamente, y en las celdas D2, E2, F2 se visualiza los datos censados que son también digitados de acuerdo a las características del cultivo a ser sembrado.

D	E	F
humedad	hum_min	hum_max
75	72	78

Figura 81. Visualización de Humedad en Celdas.

Fuente: Autor.

En la Figura 82 se muestra en las celdas G1, H1, I1, J1, K1, la Precipitación, Switch calefactor, Switch ventilador, Switch electroválvula y control respectivamente, y en las celdas G2, H2, I2, J2, K2 se visualiza los datos censados en la precipitación y los datos de encendido y apagado de los actuadores, y el control automático y manual del invernadero.

G	H	I	J	K
precipitaciones	switch calefactor	switch ventilador	switch bomba	control
2.2	ON	ON	ON	1

Figura 82. Visualización de Precipitación, Calefactor, Ventilador, Electroválvula, Control en las celdas.

Fuente: Autor.

Mediante los parámetros y especificaciones del tipo de siembra se modifica la hora de riego actual, las precipitaciones requeridas, la totalidad de precipitaciones, la hora de riego de finalizado y el número de basculadas que otorga el sistema de pluviometría como se observa en la Figura 83.

6	LECHUGA	
7	hora_riego =	17:20
8	mmPreciReq =	2.00
9	TotPreci =	0.00
10	hora_riegoM =	16:22
11	LLuvia_Actual =	2.2

Figura 83. Configuración de la Hora de Riego.

Fuente: Autor.

Se monitorea el estado actual de cada uno de los sensores y actuadores conectados al módulo ESP8266 vía WIFI y la hora exacta al momento de la comunicación con la hoja de cálculo, en la Figura 84 se muestra los datos de monitoreo.

HORA	ESTADO	ESP 8266
2017-07-11 10:51:45	esp	TEMP1
2017-06-05 9:32:14	esp	TEMP2
2017-07-02 20:07:35	esp	TEMP3
2017-06-05 8:46:49	Recibiendo datos	PLUVIO
		BOMBA
		CALEF
		VENTI

Figura 84. Estado Actual de los Actuadores, Sensores.

Fuente: Autor.

En la Figura 85 se muestra el interfaz de la hoja de cálculo Control.

The screenshot shows a Google Sheets spreadsheet with the following data:

temperatura	temp_min	temp_max	humedad	hum_min	hum_max	precipitaciones	switch calefactor	switch ventilador	switch bomba	control
20	17	23	75	72	78	2.2	OFF	OFF	OFF	0

LECHUGA	
hora_riego =	17:20
mmPreciReq =	2.00
TotPreci =	0.00
hora_riegoM =	16:22
LLuvia_Actual =	2.2

HORA	ESTADO	ESP 8266
2017-07-11 10:51:45	esp	TEMP1
2017-06-05 9:32:14	esp	TEMP2
2017-07-02 20:07:35	esp	TEMP3
2017-06-05 8:46:49	Recibiendo datos	PLUVIO
		BOMBA
		CALEF
		VENTI

Figura 85. Interfaz de Control

Fuente: Autor.



#### 2.4.1.5. PROGRAMACIÓN EN LA NUBE APLICANDO GOOGLE APP SCRIPT Y GOOGLE SPREADSHEET.

Google App Script es un lenguaje de scripting basado en JavaScript que permite interactuar con las aplicaciones que se encuentran alojadas en los servidores de google, como Documentos, Hojas de cálculo y Formularios.

Entre las características de Google App Script añade menús personalizados, diálogos y barras laterales a Google Docs, escribe funciones personalizadas para hojas de cálculo de Google, publica aplicaciones web, independientes o integradas en Google Sites, interactúa con otros servicios de google como: google drive, gmail, convierte una aplicación de Android en un complemento para que pueda intercambiar datos con google docs en un dispositivo móvil.

Como se analiza en el capítulo uno, al hacer aplicaciones utilizando el IoT (internet de las cosas) a sistemas automatizados para invernaderos, la plataforma de google ofrece diversos servicios para que interactúe con el módulo WIFI ESP8266 que permite enviar y recibir los datos censados del DHT11, red switch hacia la web, así como también de los actuadores. En primera instancia se crea una cuenta de usuario en google como se muestra en la Figura 86.

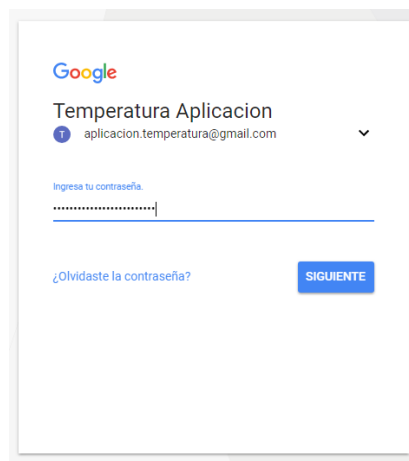


Figura 86. Creación de Cuenta de Usuario en Google

Fuente: Autor.

Obteniendo la cuenta se dirige al url: <https://script.google.com> como se muestra en la Figura 87.

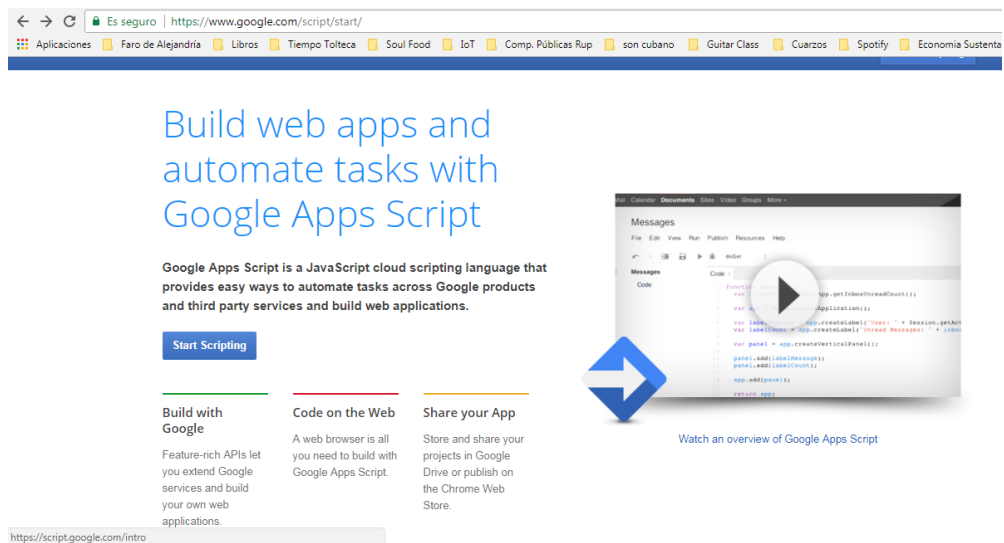


Figura 87. Página Web de google Script

Fuente: Autor.

Se inicia la secuencia de comandos para continuar con el editor de secuencias de comandos, en la Figura 88 se muestra el editor.

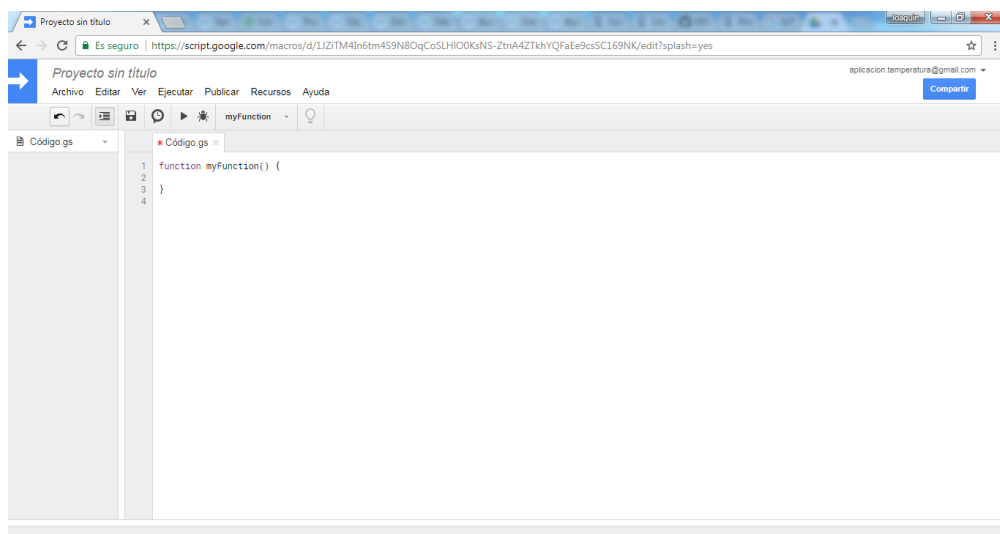


Figura 88. Editor de Comandos.

Fuente: Autor.

#### 2.4.1.5.1. PROGRAMACIÓN DEL SENSOR DE TEMPERATURA Y HUMEDAD 1

Básicamente se utiliza las funciones doGet() y doPost() para programar los parámetros del Sensor 1, Sensor 2, Sensor 3, Sensor 4 y para controlar cada uno

de los actuadores: Ventilador, Calefactor, electroválvula, que posteriormente son enlazados a la hoja de cálculo o google spreadsheet.

Al inicio del código se define la variable timezone que muestra el formato de zona horaria que sirve como encabezado para todas las programaciones de los sensores y actuadores.

```
var timezone = "GMT-5"; //zona horaria ecuador
```

La variable timestamp\_format que muestra el formato de fecha y hora.

```
var timestamp_format = "yyyy-MM-dd HH:mm:ss"; // Formato de fecha y hora.
```

Al trabajar con la función doPost permite que se llame después de una consulta de tipo Http Post para solicitar una página al servidor.

```
function doPost(e) {
  var val = e.parameter.value;

  if (e.parameter.value !== undefined){
    var range = sheet.getRange('A2');
    range.setValue(val);
  }
}
```

Mientras que con doGet envía al servidor pares de nombre y valor pero dentro del mensaje HTTP solicitando acceso a la página del servidor, se utiliza los protocolos HTTP y específicamente la librería HttpsRedirect ya que sirve para registrar los datos, establecer comunicación y controlar sin interrupciones cada una de las peticiones que se realiza para la aplicación del internet de las cosas en el proyecto.

La función doGet(e) devuelve valores "Ok", "Sin parámetros", "unsupported parameter" al haber recibido parámetros erróneos luego de hacer la comparación con el condicional if.

```
function doGet(e) {
  Logger.log( JSON.stringify(e) ); // muestra los parametros

  var result = 'Ok'; // asume correcto

  if (e.parameter == undefined) {
    result = 'Sin parametros';
  }
}
```

En caso de recibir parámetros verdaderos o correctos se procede a ejecutar las líneas de código dentro del else.

La línea de código siguiente permite definir la identificación de la hoja de cálculo de google creada anteriormente que permite interactuar con el script.

```
var id = '1_CS34qSzcJJWFDeymnVtunDXVfHgVN8m2ezHsGdP_U'; // ID de hoja de cálculo
```

En la variable sheet se define la hoja activa que siempre es la primera pestaña de la hoja de cálculo de google con el ID de la variable creada anteriormente en este caso la hoja TempHum1.

```
var sheet = SpreadsheetApp.openById(id).getActiveSheet(); //hoja activa TempHum1
```

Se define también como variables las hojas de cálculo adicionales que se van a utilizar: Control, TempHum2, TempHum3 cada una con su respectiva variable, ya que aquí se va a realizar la suma y promedio de las tres temperaturas y ese valor se envía a la hoja Control.

```
var control = SpreadsheetApp.openById(id).getSheetByName("Control");//hoja control
var temp2 = SpreadsheetApp.openById(id).getSheetByName("TempHum2");//Hoja TempHum2
var temp3 = SpreadsheetApp.openById(id).getSheetByName("TempHum3");//Hoja TempHum3
var newRow = sheet.getLastRow() + 1; //Nos posiciona en la ultima fila
var rowData = []; //creacion de variable de columna
```

A partir de aquí se realiza un ciclo repetitivo con for por cada parámetro

```
for (var param in e.parameter) {
```

Al haber varios parámetros se utiliza el switch y por cada caso un conjunto de subrutinas.

```
switch (param) {
```

**Parámetro now.-** Se verifica en qué estado se encuentra el esp8266 del sensor de temperatura y humedad 1, estado que puede ser "esp conectado" y "transmitiendo datos", también la fecha y hora de la última comunicación.

```
case 'now': //Parametro de estado
var range = control.getRange('J8'); // rango celda J8
range.setValue(value); // ESTADO ACTUAL ESP en celda J8
control.getRange('I8').setValue(date);//fecha en la celda I8 de control
sheet.getRange('J2').setValue(value); // ESTADO ACTUAL ESP en celda J2
rowData[0] = ''; //Limpiamos la ultima celda de la columna A
break; //terminamos y salimos del parametro now
```

**Parámetro value.-** Corresponde a temperatura actual recibidos del sensor de Temperatura 1 mediante el ESP8266 mismo que será pasado a la hoja TempHum1 y la Hoja Control, también se ubica la fecha del último dato recibido y el estado del Esp8266 correspondiente a este sensor.

```
case 'value': //Parametro de temperatura
  rowData[1] = value; //Temperatura en la columna B
  rowData[0] = date; //fecha y hora en la columna A
  control.getRange('I8').setValue(date);//fecha de ultimo dato
  control.getRange('J8').setValue("Recibiendo datos");//estado del sensor
  var range = sheet.getRange('E2'); //inserta valor actual
  range.setValue(value);
```

**Promedio y sumatoria de temperaturas.-** Se promedia las 3 temperaturas otorgadas por los sensores DHT11, básicamente con getValue se lee los datos de las celdas de las hojas de cálculo requeridas y con setValue se escribe los datos en las celdas de las hojas de cálculo requeridas, la temperatura promedio servirá para realizar el control del invernadero.

```
var rangeTemp1 = sheet.getRange('E2'); //rango celda E2 temp1
var temperatura1=rangeTemp1.getValue(); //Leo temp1
var rangeTemp2 = temp2.getRange('E2'); //rango celda E2 temp2
var temperatura2=rangeTemp2.getValue(); //Leo temp2
var rangeTemp3 = temp3.getRange('E2'); //rango celda E2 temp3
var temperatura3=rangeTemp3.getValue(); //Leo temp3
var temperatura=parseFloat((temperatura1+temperatura2+temperatura3)/3).toFixed(0);//promedio
//temperaturas redondeado sin decimales
var rangec = control.getRange('A2'); //obtengo rango celda A2 de control
rangec.setValue(temperatura); //inserto valor de temperatura promedio en la celda de control
rowData[3] = temperatura; //promedio temperaturas en la columna C
```

**Control automático de Temperatura.-** Se compara la temperatura mínima y máxima requeridas de acuerdo a lo que se esté cultivando con la temperatura promedio para encender/apagar el calefactor o encendido/apagado del ventilador según corresponda.

```
var rangecam = control.getRange('K2'); //control automatico/manual
var controlam=rangecam.getValue(); //Leo control
var rangeTMin = control.getRange('B2'); //Temp min
var TempMin=rangeTMin.getValue(); //Leo temp minima
var rangeTMax = control.getRange('C2'); //Temp min
var TempMax=rangeTMax.getValue(); //Leo temp minima
var rangeCal = control.getRange('H2'); //calefactor
var calefactor=rangeCal.getValue(); //Leo calefactor
var rangeVent = control.getRange('I2'); //ventilador
var ventilador=rangeVent.getValue(); //Leo ventilador
```

```

if (controlam=="1"){
  if(temperatura<TempMin){           //temperatura actual < temperatura minima
    rangeCal.setValue("ON");        //enciendo el calefactor
  }else {                             //caso contrario
    rangeCal.setValue("OFF");       //apago el calefactor
  }
  if(temperatura>TempMax){           //temperatura actual > temperatura maxima
    rangeVent.setValue("ON");       //enciendo el ventilador
  }else {                             //caso contrario
    rangeVent.setValue("OFF");      //apago el ventilador
  }
}
rowData[4] = rangeCal.getValue();    //ubico el valor actual de calefactor en la columna E
rowData[5] = rangeVent.getValue();   //ubico el valor actual de ventilador en la columna F

```

**Parámetro humedad.-** Posiciona el valor de humedad recibido del sensor de temperatura 1 en la columna de la hoja de cálculo TempHum1.

```

case 'humed': //Parametro de humedad
  rowData[2] = value; //Humedad en la columna C
  var range = sheet.getRange('G2'); //inserta valor actual
  range.setValue(value);

```

**Promedio y sumatoria de humedad.-** Suma las humedades sacando un promedio para luego ser insertado en la celda de control D2 con el valor promedio de humedad.

```

var rangeHum1 = sheet.getRange('G2'); //hum1
var humedad1=rangeHum1.getValue();   //Leo hum1
var rangeHum2 = temp2.getRange('G2'); //hum2
var humedad2=rangeHum2.getValue();   //Leo hum2
var rangeHum3 = temp3.getRange('G2'); //hum3
var humedad3=rangeHum3.getValue();   //Leo hum3
var humedad=parseFloat((humedad1+humedad2+humedad3)/3).toFixed(0); //promedio humedades
//redondeado sin decimales

var rangec = control.getRange('D2'); //obtengo celda de control
rangec.setValue(humedad); //inserto valor de humedad promedio en la celda de control

```

**Cómo obtener la URL.-** Una vez escrito el script, publicar usando la opción Publicar e implementar como aplicación, como se muestra en la Figura 89.

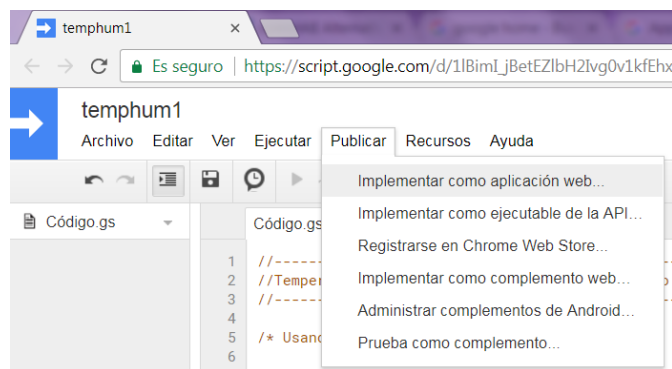


Figura 89. Implementación de la Aplicación Web.

Fuente: Autor.

Guardar como una revisión con descripción, luego se elige cómo ejecutar el script y quién tiene acceso al script. Para que los dispositivos accedan al script, se selecciona las opciones que se indican en la Figura 90, escoger la opción anónima ya que no tiene hay forma de iniciar sesión en la cuenta de Google para autorizar el script en su dispositivo ESP8266.

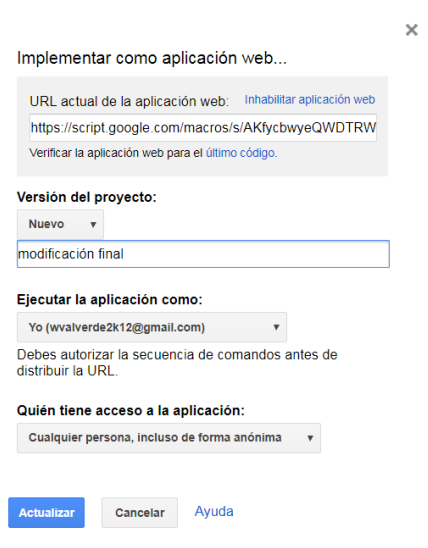


Figura 90. Opciones de Acceso al Script

Fuente: Autor.

De la Figura 91 se obtiene el ID que es la parte en negrilla que se hará uso en la programación del ESP8266 correspondiente a Temperatura y Humedad 1.

`https://script.google.com/macros/s/AKfycbwyeQWDTRWNpMPWP15l2DLj6hp101ZmibQr9lA-OzLSeB6qYLg/exec`

Entonces el ID queda de la siguiente manera:

*ID: **AKfycbwyeQWDTRWNpMPWP15l2DLj6hp101ZmibQr9lA-OzLSeB6qYLg***

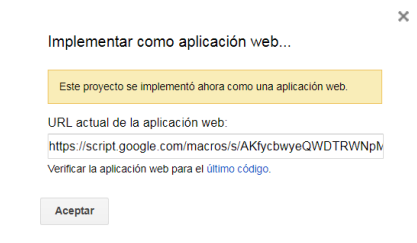


Figura 91. Obtención del ID

Fuente: Autor.

### Pasar el parámetro QueryString al Script:

El parámetro QueryString se puede pasar a la URL de la secuencia de comandos de la aplicación tan normal como lo hace con otras URL. Por ejemplo, value=20 y humed=66 como se puede ver en la Figura 92.

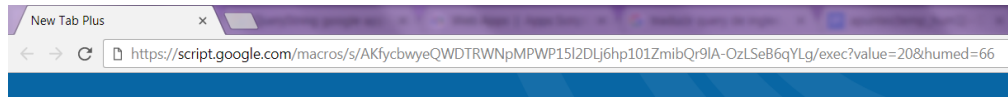


Figura 92. Visualización del parámetro Querystring.

Fuente: Autor.

Esto da como resultado Ok ya que los parámetros son correctos y funciona el script como se muestra en la Figura 93.

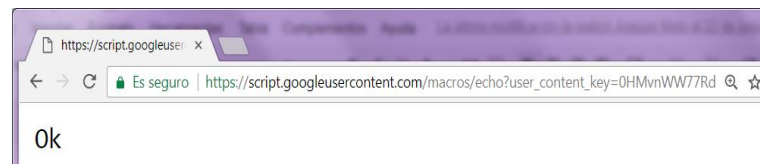


Figura 93. Resultados del parámetro Querystring.

Fuente: Autor

### 2.4.1.5.2. PROGRAMACIÓN DEL SENSOR DE TEMPERATURA Y HUMEDAD 2.

La programación de las demás aplicaciones para los otros sensores/actuadores es similar con pequeñas modificaciones de celdas y hojas de cálculo por lo tanto el ID de la hoja de cálculo es la misma ver la Figura 69.

Aquí se define como variables las hojas de cálculo que se va a utilizar: TempHum2 y Control.

```
var sheet = SpreadsheetApp.openById(id).getSheetByName("TempHum2");//Hoja TempHum2
var control = SpreadsheetApp.openById(id).getSheetByName("Control");//Hoja de control
var newRow = sheet.getLastRow() + 1; //rango de la ultima fila mas uno
var rowData = []; //declaro varibale de vector vacio
```

**Parámetro now.-** Se verifica en qué estado se encuentra el ESP8266 del sensor de temperatura y humedad 2, estado que puede ser "esp conectado" y "transmitiendo datos", también la fecha y hora de la última comunicación.



```

var range = control.getRange('J9'); //rango de la celda J9
range.setValue(value); //estado del ESP en la celda J9
control.getRange('I9').setValue(date); //fecha y hora en la celda I9
sheet.getRange('J2').setValue(value); //estado del ESP en la celda J2
rowData[0] = '' ; //limpiamos la ultima celda de la columna A
break; //terminamos y salimos del parametro now

```

**Parámetro value.-** Corresponde a temperatura actual recibidos del sensor de Temperatura 2 mediante el ESP8266 mismo que será pasado a la hoja TempHum2 y la Hoja Control, también se ubica la fecha del último dato recibido y el estado del Esp8266 correspondiente a este sensor.

```

case 'value': //Parametro de temperatura
  rowData[1] = value; //Temperatura en la columna B
  rowData[0] = date; //fecha y hora en la columna A |
  control.getRange('I9').setValue(date); //inserta fecha de ultimo dato en la celda I9
  control.getRange('J9').setValue("Recibiendo datos"); //estado del sensor en la celda J9
  var range = sheet.getRange('E2'); //obtiene el rango de la celda E2
  range.setValue(value); //inserta el valor en la celda E2

  var range = sheet.getRange('H2'); // obtiene el rango de la celda H2
  range.setValue(date); // inserta la fecha en la celda H2
  break;

```

**Parámetro humedad.-** Posiciona el valor de humedad recibido del sensor de temperatura 2 en la columna de la hoja de cálculo TempHum2.

```

case 'humed': //Parametro de humedad
  rowData[2] = value; //Humedad en la columna C
  var range = sheet.getRange('G2'); //rango de celda G2
  range.setValue(value); //valor de humedad en celda G2

```

### 2.4.1.5.3. PROGRAMACIÓN DEL SENSOR DE TEMPERATURA Y HUMEDAD 3.

Aquí se define como variables las hojas de cálculo que se van utilizar TempHum3 y Control.

```

var id = '1_CS34qSzcJJWFdeymnVtunDXVfHgVN8m2ezHsGdP_U'; //ID de hoja de cálculo
var sheet = SpreadsheetApp.openById(id).getSheetByName("TempHum3"); //hoja tempHum3
var control = SpreadsheetApp.openById(id).getSheetByName("Control"); //hoja control
var newRow = sheet.getLastRow() + 1; //Nos posiciona en la ultima fila
var rowData = []; //creacion de variable de columna

```

**Parámetro now.-** Se verifica en qué estado se encuentra el ESP8266 del sensor de temperatura y humedad 2, estado que puede ser "esp conectado" y "transmitiendo datos", también la fecha y hora de la última comunicación.

```

case 'now': //Parametro de estado
    var range = control.getRange('J10'); //rango de la celda J10
    range.setValue(value); //estado del esp en la celda J10
    control.getRange('I10').setValue(date); //fecha y hora en la celda I10
    sheet.getRange('J2').setValue(value); //estado del esp en la celda J2

```

**Parámetro value.-** Corresponde a temperatura actual recibidos del sensor de Temperatura 3 mediante el ESP8266 mismo que será pasado a la hoja TempHum3 y la Hoja Control, también se ubica la fecha del último dato recibido y el estado del Esp8266 correspondiente a este sensor.

```

case 'value': //Parametro de temperatura
    rowData[1] = value; //Temperatura en la columna B
    rowData[0] = date; //fecha y hora en la columna A
    control.getRange('I10').setValue(date); //fecha de ultimo dato
    control.getRange('J10').setValue("Recibiendo datos"); //estado del sensor
    var range = sheet.getRange('E2'); //rango de la celda E2
    range.setValue(value); //inserta value en la celda E2

```

**Parámetro humedad.-** Posiciona el valor de humedad recibido del sensor de temperatura 3 en la columna de la hoja de cálculo TempHum3.

```

case 'humed': //Parametro de humedad
    rowData[2] = value; //Humedad en la columna C
    var range = sheet.getRange('G2'); //rango de la celda G2
    range.setValue(value); //inserto valor en la celda G2

```

#### 2.4.1.5.4. PROGRAMACIÓN DEL SENSOR DE PLUVIOMETRÍA.

Aquí se define como variables las hojas de cálculo que se van utilizar en Pluviometría y Control.

```

var sheet = SpreadsheetApp.openById(id).getSheetByName("Preci"); //Hoja de precipitaciones
var control = SpreadsheetApp.openById(id).getSheetByName("Control"); //Hoja de control
var newRow = sheet.getLastRow() + 1; //Nos posiciona en la ultima fila
var rowData = []; //creacion de variable de columna

```

**Parámetro now.-** Se verifica en qué estado se encuentra el ESP8266 del sensor de Pluviometría, estado que puede ser "esp conectado" y "transmitiendo datos", también la fecha y hora de la última comunicación.

```

case 'now': //Parametro now
    var range = control.getRange('J11'); //rango de la celda J11
    range.setValue(value); //estado del esp en la celda J11
    control.getRange('I11').setValue(date); //fecha y hora en la celda I11
    sheet.getRange('G2').setValue(value); //estado del esp en la celda G2

```

**Parámetro value.-** Corresponde a cantidad de lluvia actual recibidos del sensor de Pluviometría mediante el ESP8266 mismo que será pasado a la hoja Pluvio y la Hoja Control, también se ubica la fecha del último dato recibido y el estado del Esp8266 correspondiente a este sensor.

```

case 'value': //Parametro de lluvia
    rowData[1] = value; //Valor en la columna B en la hoja Preci
    rowData[0] = date; //Fecha en la columna A en la hoja Preci
    control.getRange('I11').setValue(date); //fecha y hora en la celda I11 de la hoja control
    control.getRange('J11').setValue("Recibiendo datos");//estado del sensor en la celda J11
    var range = sheet.getRange('D2'); //rango de la celda D2 de la hoja Preci
    range.setValue(value); //lluvia actual en la celda D2
    var range = sheet.getRange('E2'); //rango de la celda E2 de la hoja Preci
    range.setValue(date); //fecha y hora en la celda E2
    var range = control.getRange('G2'); //rango de la celda G2 de la hoja control
    range.setValue(value); //inserto valor en la celda G2

```

#### - Control Automático de la Electroválvula en la celda K2

Para el control de riego de un invernadero se toma en cuenta la hora de riego y la cantidad de agua a regar ya sea activando una electroválvula que funciona como llave de paso o en su defecto una bomba de agua.

```

var rangecam = control.getRange('K2'); //rango de la celda K2
var controlam=rangecam.getValue(); //Leo control control automatico/manual de la celda K2
var rangeBomba = control.getRange('J2'); //rango de la celda J2
var Bomba=rangeBomba.getValue(); //Leo hora de riego de la celda J2
var rangeHoraRiego = control.getRange('B7'); //rango de la celda B7
var HoraRiego=rangeHoraRiego.getValue(); //Leo hora de riego de la celda B7
var rangeHoraRiegoMaximo = control.getRange('B10'); //rango de la celda B10
var HoraRiegoMaximo=rangeHoraRiegoMaximo.getValue(); //Leo hora de riego maximo de la celda B10
var rangeCantRiego = control.getRange('B8'); //rango de la celda B8
var CantRiego=rangeCantRiego.getValue(); //Leo Cantidad de riego en mm de la celda B8
var rangeLluviaActual=control.getRange('B11'); //rango de la celda B11
rangeLluviaActual.setValue(CantLluvia); //inserto valor de lluvia actual en la celda B11
var rangeTotalRiego = control.getRange('B9'); //rango de la celda B9
var TotalRiego=rangeTotalRiego.getValue(); //Leo Cantidad Total de riego de la celda B9
control.getRange('C9').setValue(TotalRiego+CantLluvia); //sumatoria de lluvia en la celda C9
var rangeHoraActRiego = control.getRange('D7'); //rango de la celda D7
rangeHoraActRiego.setValue(dateRiego); //Grabo la hora actual de riego en la celda D7
var HoraActual=rangeHoraActRiego.getValue(); //Leo hora actual de la celda D7
if (controlam=="1"){ //verifico si el control es automatico o manual
    if(HoraRiego <= HoraActual && HoraRiegoMaximo >= HoraActual){//compruebo hora de riego
        if (TotalRiego<CantRiego){ //compruebo cantidad de riego
            rangeBomba.setValue("ON"); //se enciende la bomba
            rangeTotalRiego.setValue(TotalRiego+CantLluvia); //sumo cantidad de lluvia
        }else rangeBomba.setValue("OFF"); //caso contrario apago la bomba
    }else { //caso contrario no es hora de regar
        rangeTotalRiego.setValue(0); //reseteo variable total de lluvia
        rangeBomba.setValue("OFF"); //apago la bomba
    }
}
}

```

#### - Envío de parámetro a ESP8266

```

return ContentService.createTextOutput(sheet.getRange('F2').getValue());

```

En la Figura 94 se observa cómo se registra la secuencia de comandos que se utiliza para la programación del sensor 4.

```

Ver Ejecutar Publicar Recursos Ayuda
Código.gs
31 }
32 }
33 }
34 else {
35   var id = '1_CS34qSzcJWFDeymmnVtunDXVfHgVN8m2ezHsGdP_U'; // Spreadsheet ID
36   var sheet = SpreadsheetApp.openById(id).getSheetByName("Preci");//Nombre de la pestaña que vamos a trabajar
37   var control = SpreadsheetApp.openById(id).getSheetByName("Control");//Pestaña control
38   var newRow = sheet.getLastRow() + 1;
39   var rowData = [];
40   //var waktu = new Date();
41   //rowData[0] = new Date(); // Fecha y hora en la columna A
42
43   var date = Utilities.formatDate(new Date(), timezone, timestamp_format);
44   var dateRiego = Utilities.formatDate(new Date(), timezone, timestamp_format_riego);
45
46   for (var param in e.parameter) {
47     Logger.log('In for loop, param='+param);
48     var value = stripQuotes(e.parameter[param]);
49     //Logger.log(param + ':' + e.parameter[param]);
50     switch (param) {
51       case 'now': //Parameter
52         var range = control.getRange('J11'); // ESTADO ACTUAL ESP
53         range.setValue(value);
54         control.getRange('I11').setValue(date);
55         sheet.getRange('G2').setValue(value);
56         rowData[0] = '' ;

```

Figura 94. Registra de Secuencia de Comandos Sensor 4.

Fuente: Autor.

En la Figura 95 se observa cómo se registra la secuencia de comandos que se utiliza para la programación del actuador 1.

```

Ventilador
Archivo Editar Ver Ejecutar Publicar Recursos Ayuda
Código.gs
11 if (e.parameter.value !== undefined){
12   var range = sheet.getRange('I2');
13   range.setValue(val);
14 }
15 }
16
17 function doGet(e){
18   var val = e.parameter.value;
19   var cal = e.parameter.cal;
20   var read = e.parameter.read;
21
22   if (cal !== undefined){
23     return ContentService.createTextOutput(GetEventsOneWeek());
24   }
25
26   if (read !== undefined){
27     return ContentService.createTextOutput(sheet.getRange('I2').getValue());
28   }
29
30   if (e.parameter.value === undefined)
31     return ContentService.createTextOutput("No value passed as argument to script Url.");
32
33
34
35   var range = sheet.getRange('I2');
36   var retVal = range.setValue(val).getValue();
37
38   //var newRange = sheet.getRange(newRow, 1, 1, rowData.length);
39   //newRange.setValues([rowData]);
40

```

Figura 95. Registra de Secuencia de Comandos Actuador 1.

Fuente: Autor.

En la Figura 96 se observa cómo se registra la secuencia de comandos que se utiliza para la programación del actuador 2.

```
3
4 var ss = SpreadsheetApp.openById('1_CS34qSzcJJWFDeymmVtunDXVfHgVN8m2ezHsGdP_U');
5 //Var sheet = ss.getSheetByName('Esp8266 trends');
6 var sheet = ss.getSheetByName('Control');
7
8 function doPost(e) {
9   var val = e.parameter.value;
10
11   if (e.parameter.value !== undefined){
12     var range = sheet.getRange('H2');
13     range.setValue(val);
14   }
15 }
16
17 function doGet(e){
18   var val = e.parameter.value;
19   var cal = e.parameter.cal;
20   var read = e.parameter.read;
21
22   if (cal !== undefined){
23     return ContentService.createTextOutput(GetEventsOneWeek());
24   }
25
26   if (read !== undefined){
27     return ContentService.createTextOutput(sheet.getRange('H2').getValue());
28   }
29
30   if (e.parameter.value === undefined)
31     return ContentService.createTextOutput("No value passed as argument to script Url.");
32 }
```

Figura 96. Registra de Secuencia de Comandos Actuador 2.

Fuente: Autor.

En la Figura 97 se observa cómo se registra la secuencia de comandos que se utiliza para la programación del actuador 3.

```
4 var timezone = "GMT-5";//Zona horaria
5 var timestamp_format = "yyyy-MM-dd HH:mm:ss"; // Formato de fecha y hora.
6
7
8 var ss = SpreadsheetApp.openById('1_CS34qSzcJJWFDeymmVtunDXVfHgVN8m2ezHsGdP_U');
9 var sheet = ss.getSheetByName('Control');
10
11 function doPost(e) {
12   var val = e.parameter.value;
13
14   if (e.parameter.value !== undefined){
15     var range = sheet.getRange('J2');
16     range.setValue(val);
17   }
18 }
19
20 function doGet(e){
21   var val = e.parameter.value;
22   var cal = e.parameter.cal;
23   var read = e.parameter.read;
24
25   var date = Utilities.formatDate(new Date(), timezone, timestamp_format);
26
27
28   if (cal !== undefined){
29     return ContentService.createTextOutput(GetEventsOneWeek());
30   }
31
32   if (read !== undefined){
33     return ContentService.createTextOutput(sheet.getRange('J2').getValue());
34   }
```

Figura 97. Registra de Secuencia de Comandos Actuador 3.

Fuente: Autor.

## 2.4.1.6. INSTALACIÓN, CONFIGURACIÓN DE ARDUINO IDE Y PROGRAMACIÓN DEL MÓDULO ESP8266.

### Arduino IDE

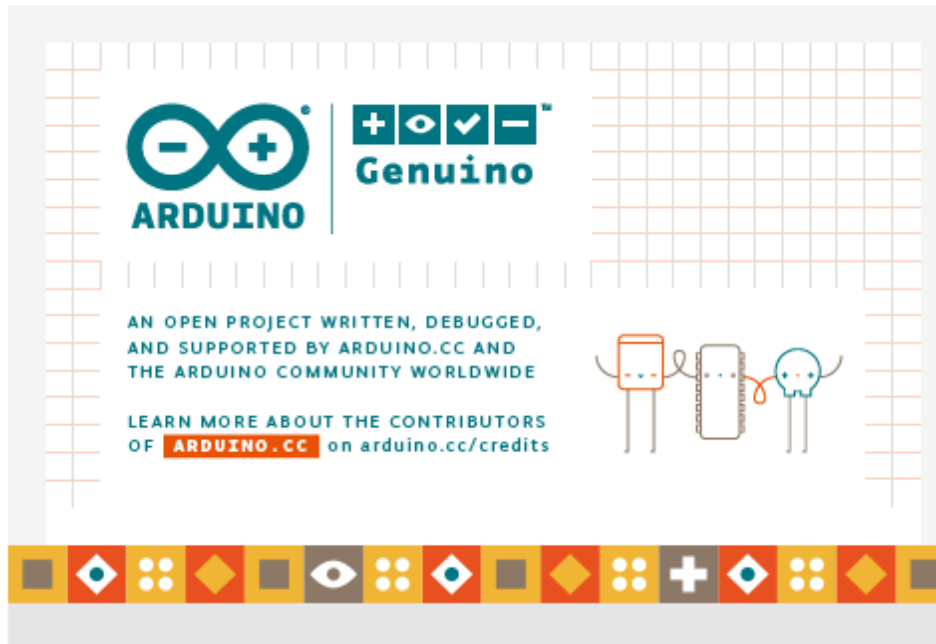


Figura 98. Mensaje de Inicio Arduino.

Fuente: Autor.

En la Figura 98 se muestra el mensaje de inicio del software de Arduino, el entorno de desarrollo integrado de Arduino contiene un editor de texto para codificar, área de mensaje, consola de texto, barra de herramientas con botones para funciones comunes.

Cada programa escrito se denominada sketch, estos se escriben en el editor de texto como se muestra en la Figura 99 y se guardan con la extensión de archivo .ino. El editor tiene múltiples funciones de copiar/pegar, buscar/ir a línea, muestra también los errores que se dan en el proceso de editar el código.

En la parte inferior de la ventana muestra el modelo de placa a ser configurado y el puerto serie al cual está conectado. En la pestaña barra de herramientas existe las opciones de seleccionar el modelo de placa, abre el monitor serie, la velocidad específica que se requiera.

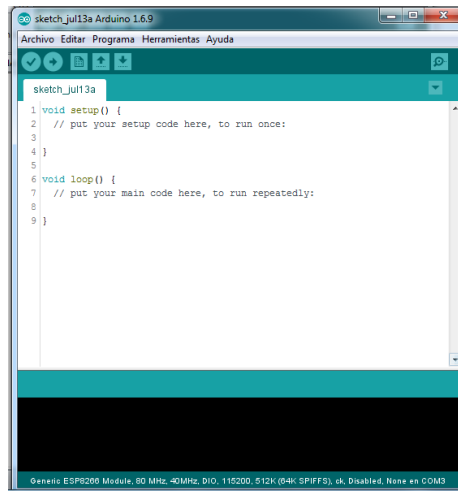


Figura 99. Interfaz del IDE Arduino.

Fuente: Autor.

### Programación del módulo ESP8266.

Una vez instalado el software se configura la interfaz IDE Arduino para el módulo ESP8266 que permite escribir bocetos utilizando funciones y bibliotecas familiares de Arduino, y ejecutarlos directamente en el MÓDULO, no requiere micro-controlador externo. El ESP8266 o modelo de tarjeta viene con las bibliotecas para comunicarse a través de WIFI usando TCP y UDP, configurar servidores HTTP, mDNS, SSDP y DNS, hacer actualizaciones OTA, utilizar un sistema de archivos en memoria flash, trabajar con tarjetas SD, servos, periféricos SPI e I2C. En la pestaña preferencias y en el gestor de URLs, se adiciona la dirección del modelo de tarjeta como se muestra en la Figura 100.

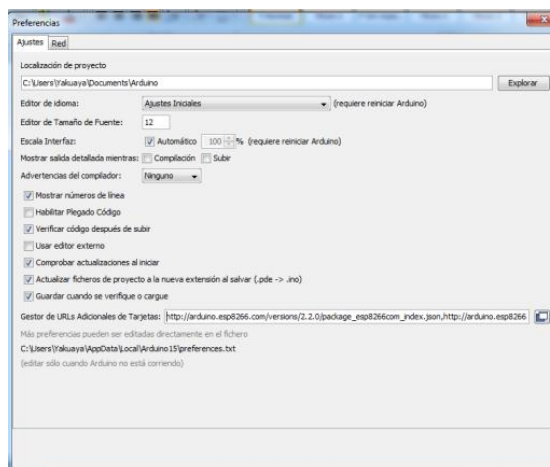


Figura 100. Dirección del Modelo de Tarjeta.

Fuente: Autor.

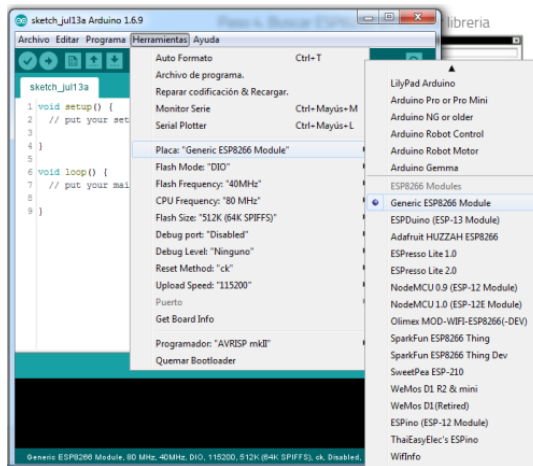


Figura 101. Selección de la Tarjeta ESP826.

Fuente: Autor.

En la pestaña Herramientas se selecciona la tarjeta ESP8266 como se muestra en la Figura 101 así como el puerto COM3 para compilar el programa a través del ConvertidorFTDI Usb – TTL como se muestra en la Figura 102.

En la Figura 103 se muestra el circuito electrónico de conexión entre el módulo ESP8266 y el Convertidor FTDI Usb – TTL serial que sirve para programar al MÓDULO.

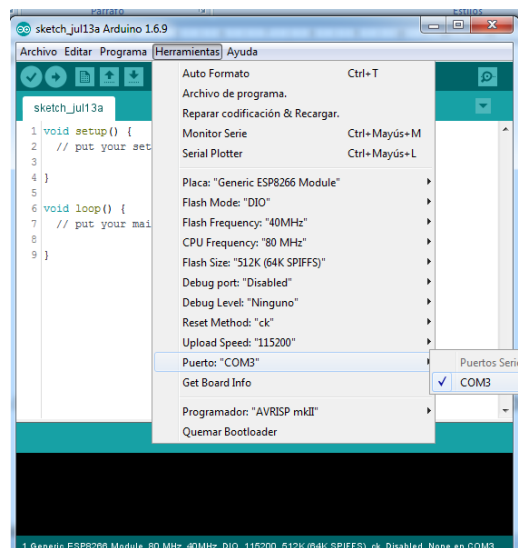


Figura 102. Selección del Puerto COM.

Fuente: Autor.



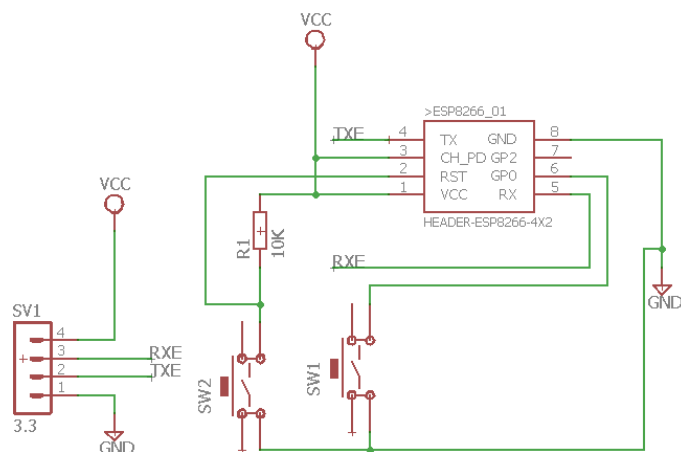


Figura 103. Diagrama electrónico para programar al módulo WIFI ESP8266.

Fuente: Autor.

### CONEXIONES.

SW1, GPIO 0 (este pin a LOW = Modo Bootloader – Modo Programación)

SW2, RST Reset (un pulso resetea el MÓDULO).

TX del convertidor FTDI Usb – TTL al RX del módulo ESP8266

RX del convertidor FTDI Usb – TTL al TX del módulo ESP8266

La librería que se utiliza para empezar a realizar la programación en la placa ESP8266 conjuntamente en el IDE es HTTPSRedirect ya que permite trabajar directamente con servicios web sin necesidad de servicios de terceros como IFTTT o Thingspeak. Esta librería Implementa las solicitudes GET y POST de acuerdo con la especificación realizada en google app script, también maneja la identidad, la codificación de transferencia en el cuerpo de respuesta, suficiente para seguir el número de redirecciones que se requieran. Estas peticiones hacen referencia para poder conectarse con los script ejecutados anteriormente y ser visualizados en la hoja de cálculo de Google.

Esta programación interactúa con los sensores y actuadores, obteniendo toda la información desde el invernadero como el accionamiento del ventilador y calefactor para controlar la temperatura, además de medir constantemente las variables de temperatura y humedad en el sensor y el control de prendido y apagado de la electroválvula mediante el sistema de pluviometría.

Por lo tanto se ha hecho uso de esta librería al ser disponible para los programadores que trabajan en el entorno IDE de Arduino por ser un software libre que también facilita el uso de la placa ESP8266 que a continuación se describen los parámetros utilizados y la función de cada una de ellas

Esta librería activa la comunicación WIFI que permite comunicar cada una de las placas ESP8266 con internet.

```
#include <ESP8266WIFI.h>
```

Activa la comunicación al servidor de google

```
#include "HTTPSRedirect.h"
```

Activa la comunicación del sensor de temperatura y humedad.

```
#include "DHT.h"
```

Se define el pin de salida del ESP8266 y el tipo de sensor a utilizar

```
#define DHTPIN 2
```

```
#define DHTTYPE DHT11
```

Los datos específicos para tener el acceso a la de red inalámbrica:

```
const char* ssid = "retlawy2k";  
const char* password = "t3$i$un@ch";
```

Activa la conexión a la url de google script.

```
const char* host = "script.google.com";  
const char* googleRedirHost = "script.googleusercontent.com";  
const char *GScriptId =  
"AKfycbweyQWDTRWNpMPWP15I2DLj6hp10IZmibQr9IA-  
OzLSeB6qYLg"; //exec ID
```

El puerto de comunicación que se utiliza es el 443 que es el puerto estándar para el protocolo HTTPS.

```
const int httpsPort = 443;
```

Para autenticar la conexión con el servidor se requiere obtener la verificación de huellas dactilares o certificado para servidores remotos.

Google envía las solicitudes de clientes HTTP de diferentes servidores en función de la ubicación geográfica y la hora del día Por lo tanto, cualquier huella digital obtenida, será de la ubicación del usuario obtenida localmente.

El comando a ejecutar es `openssl s_client` en la máquina virtual Linux creada en Windows.

El certificado resultante es el siguiente:

```
certificado: const char* fingerprint = "76 63 20 C3 BC 52 21 DD D9 DB EF 33 21  
81 F1 6E 7A 69 E2 92"
```

Escribe en la hoja de cálculo de google

```
String url = String("/macros/s/") + GScriptId +  
"/exec?value=Conected_ESP8266"
```

Imprime la temperatura y humedad en el puerto serie

```
Serial.print("Humedad: ");  
Serial.print(humidityData);  
Serial.print(" %\t");
```

Cada uno de los sensores y actuadores se programan de igual manera obteniendo así los resultados para nuestra conexión al servidor de google con las herramientas existentes de google script y spreadsheet.

En la Figura 104 se muestra la conexión al puerto USB del computador con el convertidor FTDI Usb/TTL y el módulo ESP8266.



Figura 104. Conexión al Puerto USB del Grabador de Módulos WIFI.

Fuente: Autor.

### 2.4.1.7. CREACIÓN DE PAGINA WEB EN GOOGLE SITES PARA EL MONITOREO Y CONTROL DEL INVERNADERO.

Google Sites es un servicio gratuito proporcionado por Google para los titulares de cuenta. Para crear el sitio, la dirección es <https://sites.google.com> y clic en el botón "Crear", se escribe un nombre al sitio web y se elige una plantilla, luego clic en "Crear" para obtener el sitio web.

En la parte superior derecha ir a "Configuración" y a la opción "Administrar el sitio", a continuación se selecciona Scripts de la Aplicación, esto sirve para crear una nueva secuencia de comandos y luego se selecciona "Proyecto en Blanco". El script de Google crea una página HTML y devuelve al sitio web.

Después de escribir el script, ir al sitio web creado, hacer clic en "Editar página", luego click en "Insertar - AppScript" y se elije la appscript.

Esto enlaza la appscript al sitio web. Cuando se carga el sitio web, el appscript insertado hará una llamada a la función doGet () para obtener la página HTML. El código script genera el interfaz siguiente para el invernadero cómo se muestra en la Figura 105.



Figura 105. Interfaz WEB.

Fuente: Autor.

Se copia y pega el código ID de la hoja de cálculo para tener acceso a las variables de temperatura, humedad, Precipitación, y controlar las variables Switch Calefactor, Ventilador, electroválvula y el automático/manual del invernadero.

El procedimiento en las siguientes líneas de código

```
var SPID="1_CS34qSzcJJWFDeymmnVtunDXVfHgVN8m2ezHsGdP_U";  
function doGet() {  
  return HtmlService  
    .createTemplateFromFile('web')  
    .evaluate();
```

La función `getMyData` devuelve el rango completo "A2: I2" como una matriz de dos dimensiones que se recibe en la función `onSuccess ()` en la variable "data", direccionando también a la hoja de cálculo o spreadsheet.

```
function getMyData(){  
  var dsheet =  
    SpreadsheetApp.openById(SPID).getSheetByName("Control");  
  return dsheet.getRange("A2:I2").getValues();
```

Para cada función `MyData` se declara e identifica la variable y la celda en la que se encuentra para visualizar los valores de temperatura, humedad, precipitación y hacer el control de los actuadores.

Variable Temperatura en la celda A2

```
function getDataTemperatura() {  
  var dsheet =  
    SpreadsheetApp.openById(SPID).getSheetByName("Control");  
  var hval=dsheet.getRange("A2").getValue();  
  return hval;
```

Variable Humedad en la celda D2

```
function getDataHumedad() {  
  var hval=dsheet.getRange("D2").getValue();
```

Variable Precipitación en la celda G2

```
function getDataPrecipitaciones() {  
  var hval=dsheet.getRange("G2").getValue();
```

Variable Calefactor en la celda H2

```
function getDataCalefactor() {  
    var hval=dsheet.getRange("H2").getValue();
```

Variable Ventilador en la celda I2

```
function getDataVentilador() {  
    var hval=dsheet.getRange("I2").getValue();
```

Variable Bomba en la celda J2

```
function getDataBomba() {  
    var hval=dsheet.getRange("J2").getValue();
```

Variable Control en la celda K2

```
function getDataControl() {  
    var hval=dsheet.getRange("K2").getValue();
```

El servicio HTML permite obtener páginas web que pueden interactuar con las funciones del Script de aplicaciones del lado del servidor. Es particularmente útil para crear aplicaciones web o agregar interfaces de usuario personalizadas en Google Docs, Hojas de cálculo y Formularios.

La pagina HTML es dinámica utilizando una API JavaScript asincrónica del lado del cliente llamada "google.script.run".

```
<script  
src="http://ajax.googleapis.com/ajax/libs/jquery/1.5.2/jquery.min.js">  
</script>  
<script>  
function onHeaterStatus(data){  
    celda = document.getElementById("heater");  
    if(data=='OFF')  
        celda.style.backgroundColor="green";  
    else  
        celda.style.backgroundColor="red";  
    function update() {  
        $.ajaxSetup({ cache: false });  
        var data=google.script.run.withSuccessHandler(onSuccess).getMyData();
```

*window.setTimeout(update, 10000);*

La función jquery /javascript utiliza además \$ ("# dstatus") que es el objeto <div> por la "id" y ".html ()" actualizando el contenido. La función update () llama a setTimeout () para activarse cada 10 segundos. Google.script.run llama a la función del lado del servidor getMyData () y al devolver correctamente withSuccessHandler () llama a la función onSuccess () con los datos devueltos. Así que getMyData () es clave para escribir el script para leer los datos de la hoja de cálculo.

El Url del sitio web que se obtiene es el siguiente:

<https://sites.google.com/site/invernaderounach/>

#### **2.4.1.8. CREACIÓN DE LA APLICACIÓN ANDROID PARA CELULAR.**

Para el desarrollo de la aplicación Android para el dispositivo celular se utiliza el software denominado Android Studio IDE. El código de programación permite controlar los actuadores de forma automática y manual, modificar los valores de temperatura max/min, humedad max/min, permite además visualizar en tiempo real el estado actual de temperatura, humedad y precipitaciones.

Se puede descargar Android Studio desde <http://developer.android.com/sdk/>. Que se encuentra las instrucciones para instalar y configurar Android Studio en su computadora/laptop. Descargar la aplicación y actualizar los SDK a través de "SDK Manager".

Esta aplicación puede ser instalada en cualquier teléfono móvil que posea Android 5.1 o superior. Al iniciar la aplicación, se pedirá que apruebe la lista de permisos requeridos que sirven para indicar al sistema operativo Android cuáles son todos los servicios que va a utilizar.

Si se quiere escribir un programa de Android, se debe saber qué es Activity y View, la Activity class en Android es similar a su función main ( ) en C, C++, Java. Después de la función main (), tendrá que llamar a las funciones de interfaz de usuario para crear una ventana para su aplicación. A continuación se muestra cómo es el ciclo de una actividad.

```

public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
    }
    @Override
    protected void onStart() {
    }
    @Override
    protected void onRestart() {
    }
    @Override
    protected void onPause() {
    }
    @Override
    protected void onStop() {
    }
    @Override
    protected void onDestroy() {
    }
}

```

Cualquier objeto que pueda verse en la pantalla del teléfono Android es desde View. Así que View no es más que una interfaz de usuario. Android Studio ofrece plantillas al iniciar un proyecto que creará una actividad y una vista predeterminada. Una vez que se crea un proyecto, se dirige a editar la vista (your project)-->app-->src-->layout folder (in xml format). Hay que recordar que en un programa de Android el archivo AndroidManifest.xml es muy importante. Cuando se instala una aplicación de Google AppStore, este pide que pruebe la lista de permisos requeridos de la aplicación, esa lista proviene de este archivo, esto es para decirle al sistema operativo Android cuáles son todos los servicios que va a utilizar en la aplicación (como INTERNET, STORAGE, etc.). Este archivo no es sólo para eso, la mayor parte de la configuración de la



aplicación debe declararse aquí. El objetivo es tener una aplicación con algunos botones y un servicio ejecutándose en segundo plano para comprobar los cambios de datos en la hoja de cálculo o spreadsheet. Para ello, se requiere dos entradas importantes en el archivo AndroidManifest.xml.

```
<uses-permission android:name="android.permission.INTERNET"/>
```

Para decirle a Android que nuestra aplicación necesita acceso a Internet

```
<service android:enabled="true"
android:name=".IoTDemoService"/>
```

Para decirle a Android que nuestra aplicación se ejecuta como servicio

El servicio y la aplicación de interfaz de usuario se comunican entre sí para que la aplicación sea eficiente y que el servicio del manejo de datos en la hoja de cálculo o spreadsheets se ejecute sólo como interfaz de usuario.

El Android Services similar al "Servicio" que se conoce en Windows. Se ejecuta en segundo plano sin una interfaz de usuario. El siguiente código se muestra el ciclo del servicio.

```
publicclass IoTDeviceService extends Service {
    @Override
    public IBinder onBind(Intent intent) {
        }
    @Override
    publicvoid onCreate() {
        }
    @Override
    publicint onStartCommand(Intent intent, int flags, int startId) {
        return START_STICKY;
        }
    @Override
    publicvoid onDestroy() {
        }
}
```

En `OnStartCommand ()` comenzará a ejecutarse cuando se active la función `startService ()`. En la aplicación de interfaz de usuario, se ejecuta `startService` como se indica a continuación para iniciar el servicio y trabajar con la hoja de cálculo de Google.

```
startService(new Intent(this, IoTDeviceService.class));
```

### **Acceso a la Hoja de Cálculo de Google o Spreadsheet**

Para acceder al servicio de hoja de cálculo de Google, se instala la siguiente biblioteca: Google Play Service a través de SDK Manager. Abrir `build.gradle` en la carpeta "app" de la ventana del explorador de proyectos. Se tiene en cuenta que puede haber más de un archivo `build.gradle` en el proyecto (uno en la carpeta raíz), se elige el archivo de la aplicación y se agrega la siguiente línea bajo el bloque de dependencias, compile '`com.google.android.gms:play-services:5.0.+`'.

Se vuelve a `AndroidManifest.xml` y se declara la siguiente entrada en:

```
<application>  
  
<meta-data android:name="com.google.android.gms.version"  
            android:value="@integer/google_play_services_version"/>
```

Se descarga `gdata java client libraries`, el link para la descarga es el siguiente:

<https://code.google.com/p/gdata-java-client/downloads/list>

Se Copia los archivos `gdata-client`, `gdata-core`, `gdata-maps`, `gdata-spreadsheet`, en la carpeta "lib" del proyecto (ventana Project Explorer en el IDE de Android Studio).

Después de copiar, hacer clic derecho en cada archivo jar y se hace clic en "Agregar como biblioteca" para agregarlos a la aplicación como bibliotecas. Se Guarda el proyecto y se hace clic en "Proyecto de sincronización con archivos Gradle".

Tener en cuenta que, para utilizar la hoja de cálculo de Google, se necesita una cuenta de usuario de Google, así que se pide al usuario de la aplicación que seleccione una cuenta de usuario durante el inicio, en las siguientes líneas de código que se indica se programa lo mencionado.

```
String[] accountTypes = newString[]{"com.google"};

Intent intent = AccountPicker.newChooseAccountIntent(null, null,
accountTypes, false, null, null, null, null);

startActivityForResult(intent, REQUEST_CODE_PICK_ACCOUNT);
```

Una vez que el usuario selecciona una cuenta, se obtiene StringToken para la cuenta de usuario seleccionada, pero esto requiere permisos del usuario.

```
String token=
GoogleAuthUtil.getTokenWithNotification(IoTDeviceService.this,
email,"oauth2:https://spreadsheets.google.com/feeds
https://docs.google.com/feeds", null);
```

Esto le notifica al usuario para autorizar que la aplicación acceda a Google Service con la cuenta de Google seleccionada. El usuario tiene que autorizar antes de continuar. Una vez que se recibe el token, la aplicación está lista. Debajo de código snap se enumeran las hojas de cálculo bajo la cuenta de usuario seleccionada.

```
SpreadsheetService s =new SpreadsheetService("IoTDeviceService",
"https", "google.com");

s.setAuthSubToken(token);

URL SPREADSHEET_FEED_URL =
newURL(https://docs.google.com/spreadsheets/d/1\_CS34qSzcJJWFDeym\_mnVtunDXVfHgVN8m2ezHsGdP\_U/edit#gid=0);

SpreadsheetFeed feed;

feed = s.getFeed(SPREADSHEET_FEED_URL, SpreadsheetFeed.class);

List<SpreadsheetEntry> spreadsheets = feed.getEntries();

String sheets = "";

// Run through all of the spreadsheets returned
```

```
for (SpreadsheetEntry spreadsheet : spreadsheets) {  
    Log.d("IotDeviceServie", spreadsheet.getTitle().getPlainText());  
    String title =spreadsheet.getTitle().getPlainText();  
}
```

Tener en cuenta que esta aplicación busca una hoja de cálculo denominada "Invernadero (final)" y una hoja con el nombre "Control" en Google Drive de la cuenta de usuario seleccionada.

La ventana de inicio de la aplicación se muestra en la Figura 106.



Figura 106. Ventana de Inicio de la Aplicación Android.

Fuente: Autor..



Figura 107. Ventana de Visualización de los Sensores.

Fuente: Autor.

El administrador tendrá que pulsar INGRESAR para visualizar en la segunda pantalla los datos de temperatura, humedad y precipitaciones como se muestra en la Figura 107. Además en la parte superior derecha hay un botón que son las OPCIONES que contiene la aplicación como se muestra en la Figura 108.



Figura 108. Botón de opciones de la Aplicaciones Android.

Fuente: Autor.

En la Figura 109 se muestra las opciones que existe en la aplicación.

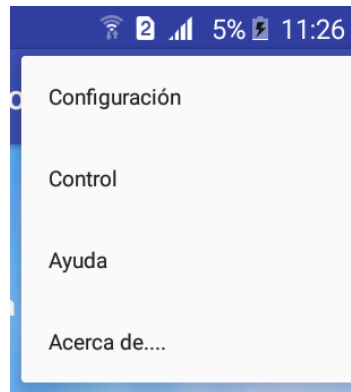


Figura 109. Configuraciones existentes en la Aplicación Android.

Fuente: Autor.

En esta ventana se digita los datos de temperatura máxima, temperatura mínima, humedad máxima, humedad mínima y actualización como se muestra en la Figura 110.



Figura 110. Configuración de Temperatura y Humedad.

Fuente: Autor.

Las siguientes líneas de código ejecutan la actualización de temperatura.

```
public void actualizarTemperatura(){  
    EditText temperaturaActualTexto = (EditText)  
    findViewById(R.id.txtTemperaturaActual);
```

```

EditText temperaturaMaximaTexto = (EditText)
findViewById(R.id.txtTemperaturaMaxima);
EditText temperaturaMinimaTexto = (EditText)
findViewById(R.id.txtTemperaturaMinima);
myTaskM = new
GetDataAsyncTaskM(this,temperaturaActualTexto.getText().toString(),temperatu
raMinimaTexto.getText().toString(),temperaturaMaximaTexto.getText().toString(
));
myTaskM.execute();
AlertDialog alertDialog;
alertDialog = new AlertDialog.Builder(this).create();
alertDialog.setTitle("Temperatura");
alertDialog.setMessage("Temperatura Actualizada");
alertDialog.show();
}

```

Las siguientes líneas de código ejecutan la actualización de humedad.

```

public void actualizarHumedad(){
EditText humedadActualTexto = (EditText)
findViewById(R.id.txtHumedadActual);
EditText humedadMaximaTexto = (EditText)
findViewById(R.id.txtHumedadMaxima);
EditText humedadMinimaTexto = (EditText)
findViewById(R.id.txtHumedadMinima);
myTaskH = new
GetDataAsyncTaskH(this,humedadActualTexto.getText().toString(),humedadMin
imaTexto.getText().toString(),humedadMaximaTexto.getText().toString());
myTaskH.execute();
AlertDialog alertDialog;
alertDialog = new AlertDialog.Builder(this).create();
alertDialog.setTitle("Humedad");
alertDialog.setMessage("Humedad Actualizada");
alertDialog.show();
}

```

```
}
```

```
public boolean onCreateOptionsMenu(Menu menu) {
```

```
public class GetDataAsyncTaskC extends AsyncTask<Void, Void, Void> {
```

```
public float
```

En la Figura 111 se muestra la ventana control que sirve para seleccionar el modo automático o manual según elija el usuario, así como también el encendido y apagado de los actuadores y el botón para actualización.



Figura 111. Ventana Control.

Fuente: Autor.

Las siguientes líneas de código ejecutan la actualización, estado de actuadores y el modo de trabajo a ejecutar.

```
public void actualizarBomba(){  
    String valor;  
    String controlautma = "0";  
    Switch switchBomba = (Switch) findViewById(R.id.swBomba);  
    if(switchBomba.isChecked()){  
        valor = "ON";
```



```

    }
else
    {
valor = "OFF";
    }
myTasB = new GetDataAsyncTaskB(this,valor,controlautma);
myTasB.execute();
    }
public void actualizarVentilador(){
    String valor;
    String controlautma = "0";
    Switch switchVentilador = (Switch) findViewById(R.id.swVentilador);
if(switchVentilador.isChecked()){
valor = "ON";
    }
else
    {
valor = "OFF";
    }
myTasV = new GetDataAsyncTaskV(this,valor,controlautma);
myTasV.execute();
    }
public void actualizarCalefactor(){
    String valor;
    String controlautma = "0";
    Switch switchCalefactor = (Switch) findViewById(R.id.swCalefactor);
if(switchCalefactor.isChecked()){
valor = "ON";
    }
else
    {
valor = "OFF";
    }
myTasCal = new GetDataAsyncTasCal(this,valor,controlautma);
myTasCal.execute();
    }
public void actualizarControl(){
    String valor;
    RadioButton rbManual = (RadioButton) findViewById(R.id.rbManual);
if(rbManual.isChecked()){
valor = "0";
    }
else
    {
valor = "1";
    }

```

```

myTasCon = new GetDataAsyncTasCon(this,valor);
myTasCon.execute();
}

```

En la ventana ayuda como se muestra en la Figura 112 se explica cada opción para ejecutar la ventana control.



Figura 112. Ventana Ayuda.

Fuente: Autor.

Las siguientes líneas de código ejecutan la ventana:

```

package com.g13enterprise.invernadero;
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
public class ayudainvernadero extends AppCompatActivity {

```

```

    String mEmail;
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_ayudainvernadero);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);
mEmail= extras.getString("email");
    }
public boolean onCreateOptionsMenu(Menu menu) {
getMenuInflater().inflate(R.menu.menu_ayudainvernadero, menu);
return true;
    }
public boolean onOptionsItemSelected(MenuItem item) {
int id = item.getItemId();
if (id == R.id.action_inicio) {
    Intent form Configuracion = new Intent (this, principal.class );
formConfiguracion.putExtra("email", mEmail);
startActivity(formConfiguracion);
return true;
    }
if (id == R.id.action_salir) {
finish();
return true;
    }
return super.onOptionsItemSelected(item);
}

```

En la ventana Acerca de... se registra los derechos de autor como se visualiza en la Figura 113.



Figura 113. Ventana Acerca de.

Fuente: Autor.

Las siguientes líneas de código ejecutan la ventana:

```
package com.g13enterprise.invernadero;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
public class acercade extends AppCompatActivity {
    String mEmail;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_acercade);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        Bundle extras= getIntent().getExtras();
        mEmail= extras.getString("email");
    }
}
```

#### 2.4.1.9. IMPLEMENTACIÓN DEL SISTEMA

El prototipo de invernadero está diseñado de acuerdo a los invernaderos convencionales de túnel, el armazón consta de tubería de ½ pulgada y se adecuó el sistema de riego con dos tubos de cañería de ½ pulgada, dichos tubos tiene 8 agujeros cada uno para que el agua riegue los 2 recipientes que contiene el cultivo.

El equipo de pluviometría se encuentra ubicado dentro del invernadero debajo de uno de los tubos donde existe un agujero para que el agua caiga sobre el sistema de pluviometría y pueda sensar la cantidad necesaria para el cultivo, en la Figura 114 se muestra el pluviómetro.

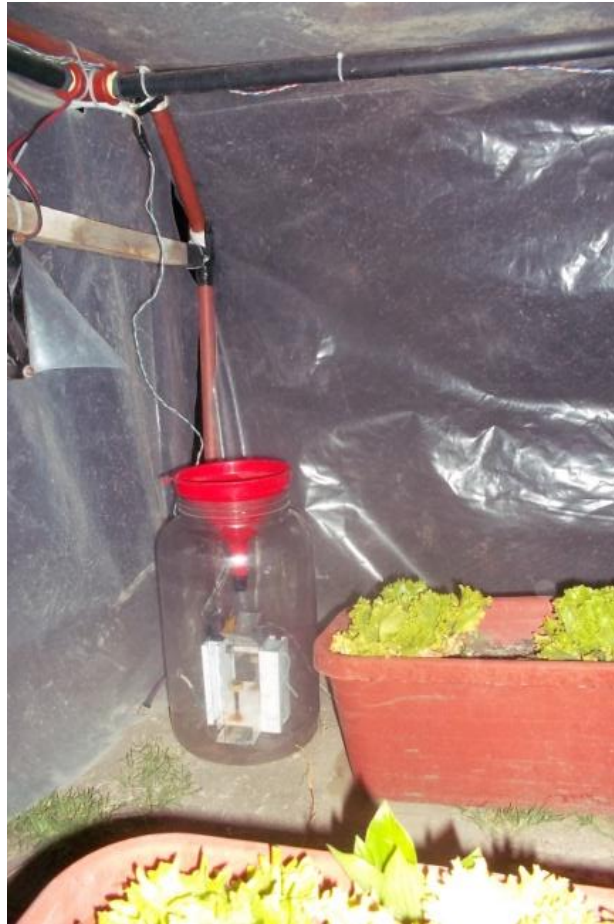


Figura 114. Pluviómetro en Ejecución.

Fuente: Autor.

Las plaquetas de los sensores y actuadores están colocadas en una caja máster cómo se muestra en la Figura 115.

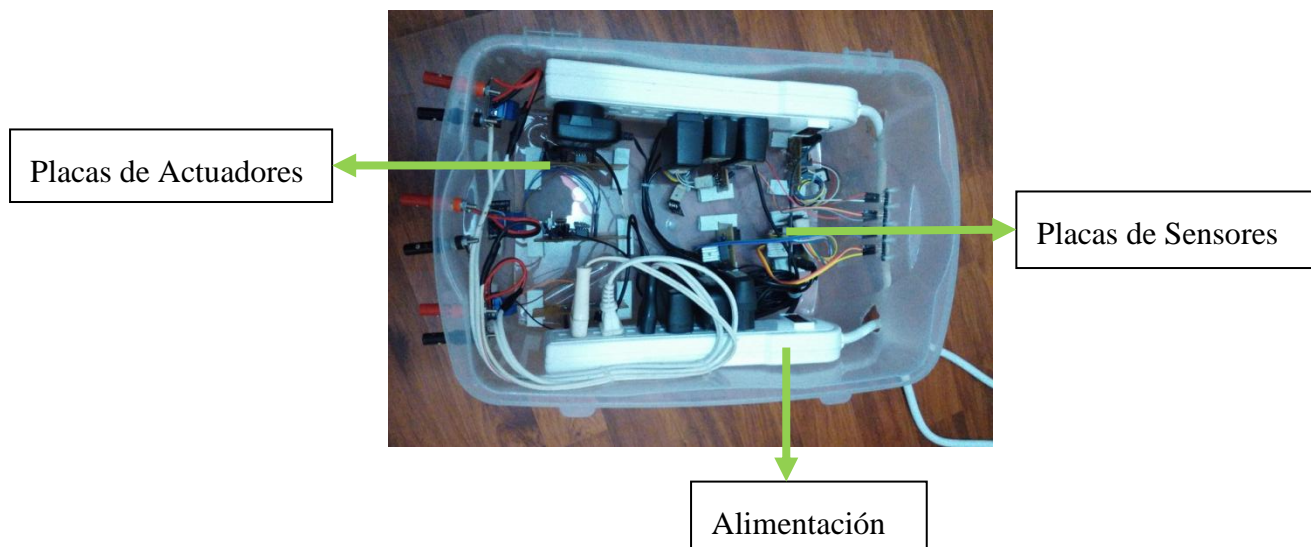


Figura 115. Caja Máster.

Fuente: Autor.

Las siete plaquetas tienen su voltaje de alimentación que van conectadas a la red eléctrica. La conexión para los actuadores en la caja máster se muestra en la Figura 116, los 3 sensores DTH11 y el sensor Reed Switch van conectados mediante cable utp hacia la caja para comunicarse con cada una las placas como se muestra en la Figura 117.

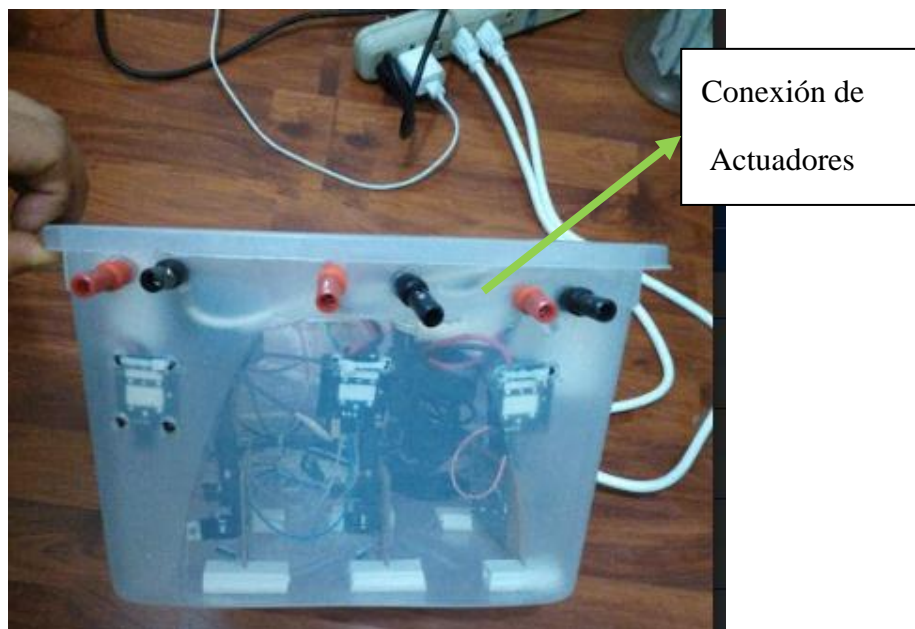


Figura 116. Conexión de Actuadores.

Fuente: Autor.

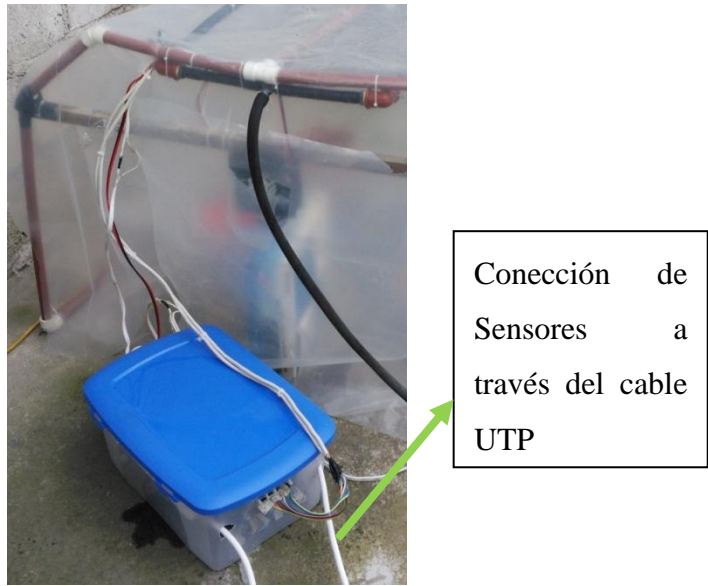


Figura 117. Conexión de Sensores a la Caja Máster.

Fuente: Autor.

Los sensores DHT11 están colocados en la parte superior del invernadero como se muestra en la Figura 118, los ventiladores se encuentran uno al extremo del otro y el calefactor dentro del invernadero, el ingreso del agua es a través de una manguera que es colocada en la tubería instalada.

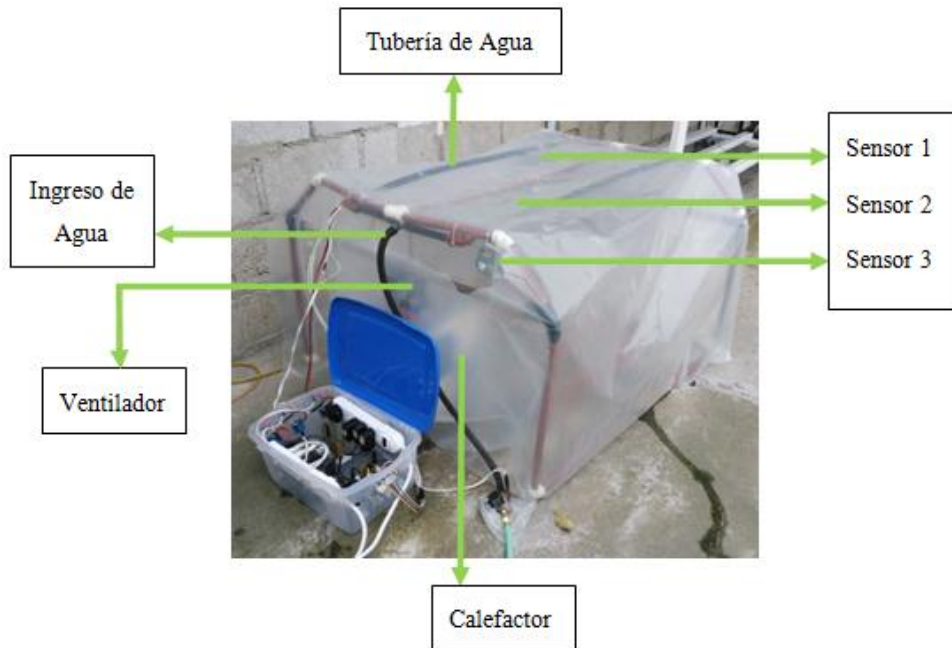


Figura 118. Prototipo de Invernadero.

Fuente: Autor.

## 2.5. COMPROBACIÓN DE HIPÓTESIS

Para la comprobación de la hipótesis se debe establecer una serie de procesos para afirmar o negar dicha hipótesis, se desarrolla pruebas con un esquema de aseveraciones aproximadas a una media de población.

- **Planteamiento de la hipótesis**

H0: “La implementación de un sistema de almacenamiento de la información, monitoreo y control aplicando el internet de las cosas permitiría la automatización de un invernadero”.

H1: “La implementación de un sistema de almacenamiento de la información, monitoreo y control aplicando el internet de las cosas no permitiría la automatización de un invernadero”.

$$H0: U_t \neq U_{real}$$

$$H1: U_t \neq U_{real}$$

En una prueba real para la activación automática del sistema, se establece el rango de temperatura min y temperatura max (14°C 18°C) del cultivo, si la temperatura promedio disminuye el rango de temp\_min se acciona el calefactor, y si la temperatura promedio aumenta el rango de temp\_max se acciona el ventilador, cuando la temperatura promedio se encuentra en el rango establecido los actuadores se mantienen apagados.

En la siguiente

Tabla 6 se muestra los datos obtenidos.

Tabla 6. Activación de Ventilador y Calefactor.

Pruebas	Temperatura Promedio °C	Calefactor	Ventilador
1	19	OFF	ON
2	19	OFF	ON



3	19	OFF	ON
4	19	OFF	ON
5	19	OFF	ON
6	19	OFF	ON
7	19	OFF	ON
8	18	OFF	OFF
9	18	OFF	OFF
10	18	OFF	OFF
11	18	OFF	OFF
12	18	OFF	OFF
13	18	OFF	OFF
14	17	OFF	OFF
15	17	OFF	OFF
16	17	OFF	OFF
17	17	OFF	OFF
18	17	OFF	OFF
19	16	OFF	OFF
20	16	OFF	OFF
21	16	OFF	OFF
22	16	OFF	OFF
23	16	OFF	OFF
24	15	OFF	OFF
25	15	OFF	OFF
26	14	OFF	OFF
27	14	OFF	OFF
28	13	ON	OFF
29	13	ON	OFF
30	13	ON	OFF
31	13	ON	OFF
32	13	ON	OFF
33	13	ON	OFF
34	13	ON	OFF
35	13	ON	OFF

Fuente: Autor.

### **ENCENDIDO AUTOMÁTICO DEL CALEFACTOR.**

En la Tabla 7 se observa que existen ocho datos necesarios para que el calefactor se mantenga prendido.

Tabla 7. Frecuencias y Porcentajes para el Calefactor.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
ON	8	23,3	39,1	39,1
OFF	11	30,3	21,7	60,9
Válidos OFF	7	21,2	13,0	73,9
OFF	9	25,2	26,1	100,0
Total	35	69,7	100,0	
Total	35	100,0		

Fuente: Autor.

En la Figura 119 se observan el porcentaje obtenido al momento que se enciende el calefactor

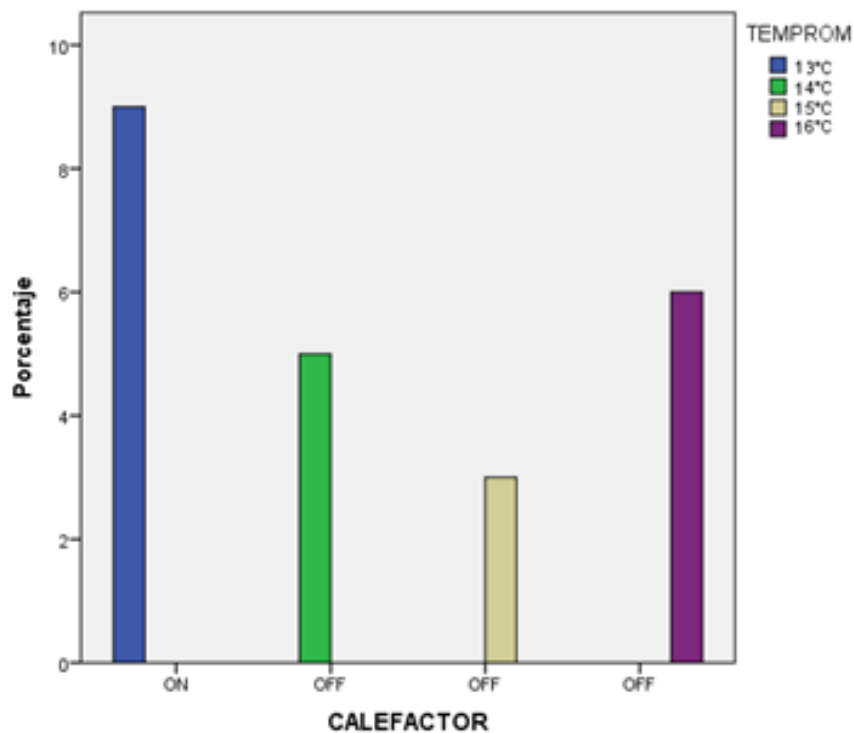


Figura 119. Encendido del Calefactor en 13°C.

Fuente: Autor.

### ENCENDIDO AUTOMÁTICO DEL VENTILADOR.

Se observa en la Tabla 8 que existen siete datos necesarios para que el ventilador se mantenga prendido.

Tabla 8. Frecuencias y Porcentajes para el Ventilador.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
OFF	11	32,3	21,7	21,7
OFF	8	22,1	17,4	39,1
Válidos OFF	9	25,2	21,7	60,9
ON	7	20,4	39,1	100,0
Total	35	69,7	100,0	
Total	35	100,0		

Fuente: Autor.

En la Figura 120 se observan el porcentaje obtenido al momento que se enciende el Ventilador.

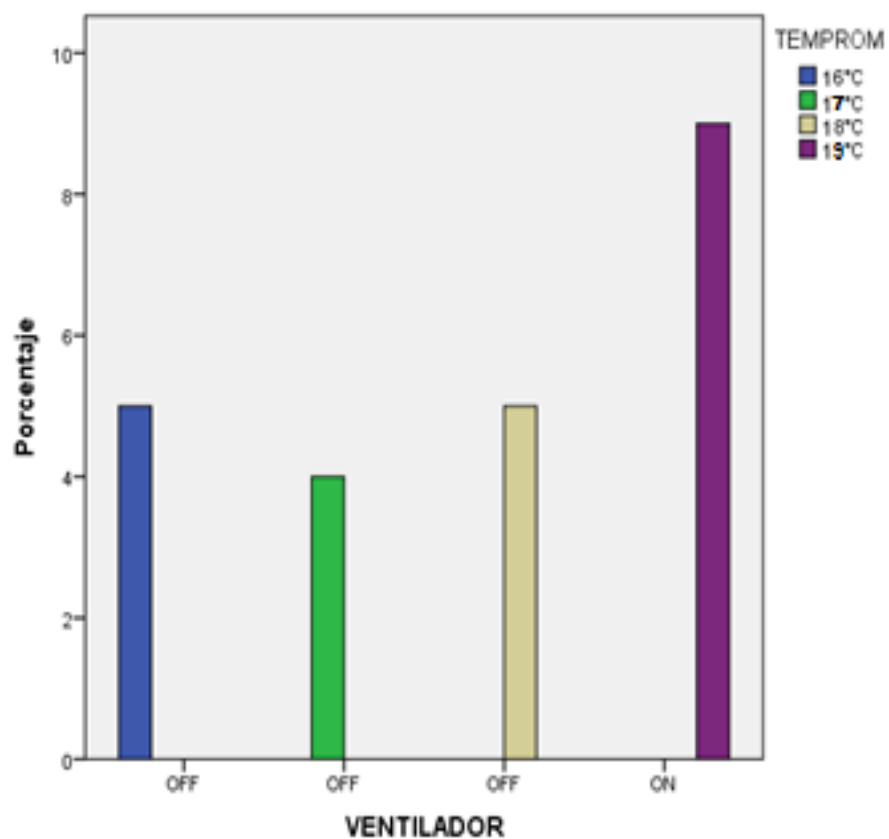


Figura 120. Encendido del Ventilador en 19°C.

Fuente: Autor.

En la Tabla 9 se muestra los valores obtenidos de la Media, Varianza y el Rango.

Tabla 9. Datos Estadísticos.

N	Válidos	35
Media		24,7826
Varianza		1,451
Rango		3,00

Fuente: Autor.

En la Tabla 10 se muestra la prueba de chi-cuadrado

Tabla 10. Prueba de Chi -Cuadrado.

	Valor	gl	Sig. asintótica (bilateral)
Chi-cuadrado de Pearson	69,000 <sup>a</sup>	9	,000
Razón de verosimilitudes	61,404	9	,000
Asociación lineal por lineal	22,000	1	,000
N de casos válidos	35		

Fuente: Autor.

Se comprueba mediante el programa estadístico SPSS que es posible la implementación de un sistema de almacenamiento de la información, monitoreo y control aplicando el internet de las cosas para la automatización de un invernadero.

### CAPITULO III

#### 3. PRUEBAS Y RESULTADOS DEL PROTOTIPO DE INVERNADERO

Con la implementación se logró crear un sistema completo que permite el almacenamiento de la información, monitorear parámetros ambientales, así como también controlar la activación de actuadores mediante la página web, aplicación Android y aplicando el Internet de las Cosas para que el prototipo de invernadero funcione correctamente.

Se verifica que todos los elementos del sistema funcionen de forma adecuada en modo automática y manual. Para comprobar el funcionamiento del prototipo se realizaron pruebas que permiten comprobar el accionamiento de los actuadores mediante el Interfaz web y la aplicación Android, y su correspondiente cambio en el interfaz web, spreadsheet y en la aplicación Android.

En las pruebas se enumera cada actuador, sensor y la función específica que cumplen, además de un método de comprobación. En la Tabla 11 y Tabla 12 se detalla su funcionamiento.

Tabla 11. Funcionamiento de Actuadores.

FUNCIONAMIENTO DE ACTUADORES					
ITEM	DESCRIPCIÓN DE TAREAS	ACCIONAMIENTO	MÉTODO DE COMPROBACIÓN	SI	NO
Alimentación de los circuitos electrónicos de acondicionamiento para electroválvula ventilador y calefacción,	Energizar la caja master que contiene todas las placas de los circuitos electrónicos de acondicionamiento para la electroválvula, ventilador y calefacción,	Manual	Observar led de encendido en las placas de los circuito electrónicos de acondicionamiento	X	
Selector automático-manual	Seleccionar si los cambios se hacen de modo automático o	Interfaz WEB (Google Spreadsheet y Página Web)	Observar el estado del actuador encendido o apagado	X	

	manual	Aplicación Android			
Encendido de Electroválvula	Dejar pasar la cantidad de agua para regadío del cultivo	Interfaz WEB (Google Spreadsheet y Página Web)	Observar el estado del actuador encendido o apagado	x	
		Aplicación Android			
Encendido Ventilador	Aumentar la Temperatura del prototipo de invernadero	Interfaz WEB (Google Spreadsheet y Página Web)	Observar el estado del actuador encendido o apagado	x	
		Aplicación Android			
Encendido Calefactor	Disminuir la temperatura del prototipo de invernadero	Interfaz WEB (Google Spreadsheet y Página Web)	Observar el estado del actuador encendido o apagado	x	
		Aplicación Android			
Módulo ESP8266 comunicación WIFI	Enviar información a las aplicaciones de google	Interfaz WEB (Google Spreadsheet y Página Web)	Observar información de sensores y actuadores en spreadsheet y página web y aplicación Android	x	
		Aplicación Android			

Fuente: Autor.

Tabla 12. Funcionamiento de Sensores.

Funcionamiento de Sensores					
ITEM	DESCRIPCIÓN DE TAREAS	ACCIONAMIENTO	MÉTODO DE COMPROBACIÓN	SI	NO
Temperatura	Medir la temperatura del invernadero	Interfaz WEB (Google Spreadsheet y Página Web)	Observar el valor de Temperatura en la página web, spreadsheet y en la aplicación Android	x	
		Aplicación Android			
Humedad Relativa	Medir la humedad relativa del Invernadero	Interfaz WEB (Google Spreadsheet y Página Web)	Observar el valor de Humedad Relativa en la página web, spreadsheet y en la aplicación Android	x	
		Aplicación Android			

Reed Switch	Medir la cantidad de riego de agua con el sistema pluviométrico	Interfaz WEB (Google Spreadsheet y Página Web)	Observar el valor de precipitaciones en la página web, spreadsheet y en la aplicación Android	x	
		Aplicación Android			

Fuente: Autor.

Con estos resultados se verifica que todos los componentes funcionan adecuadamente.

### 3.1. PRUEBAS DE MEDICIÓN DE TEMPERATURA Y HUMEDAD RELATIVA SENSOR 1

Tabla 13. Datos de Temperatura y Humedad 1 en Función del Tiempo.

FECHA Y HORA	TEMPERATURA °C	HUMEDAD RELATIVA %
2017-03-18 16:00:37	20	47
2017-03-18 16:05:08	20	47
2017-03-18 16:10:27	20	48
2017-03-18 16:15:39	20	50
2017-03-18 16:20:12	20	52
2017-03-18 16:25:35	21	55
2017-03-18 16:30:34	21	55
2017-03-18 16:35:44	21	55
2017-03-18 16:40:54	22	66
2017-03-18 16:45:02	22	66
2017-03-18 16:50:14	22	66
2017-03-18 16:55:25	22	66

Fuente: Autor.

Se han tomado medidas de temperatura en el lapso de una hora cada cinco minutos, los datos fueron tomados el 18/3/2017 de 16:00:00 a 17:00:00. Los datos de temperatura en función del tiempo se muestran en la Tabla 13. En la interfaz web se obtiene el histórico de la temperatura para los valores tabulados, como se muestra en la Figura 121, se verifica también que el envío y recepción de datos a la aplicación de google spreadsheet son exitosos. El valor de Temperatura y Humedad Relativa del Sensor 1 oscilan de acuerdo a la temperatura dentro del

invernadero, el sensor se encuentra colocado en el extremo superior izquierdo otorgando valores ambientales reales.

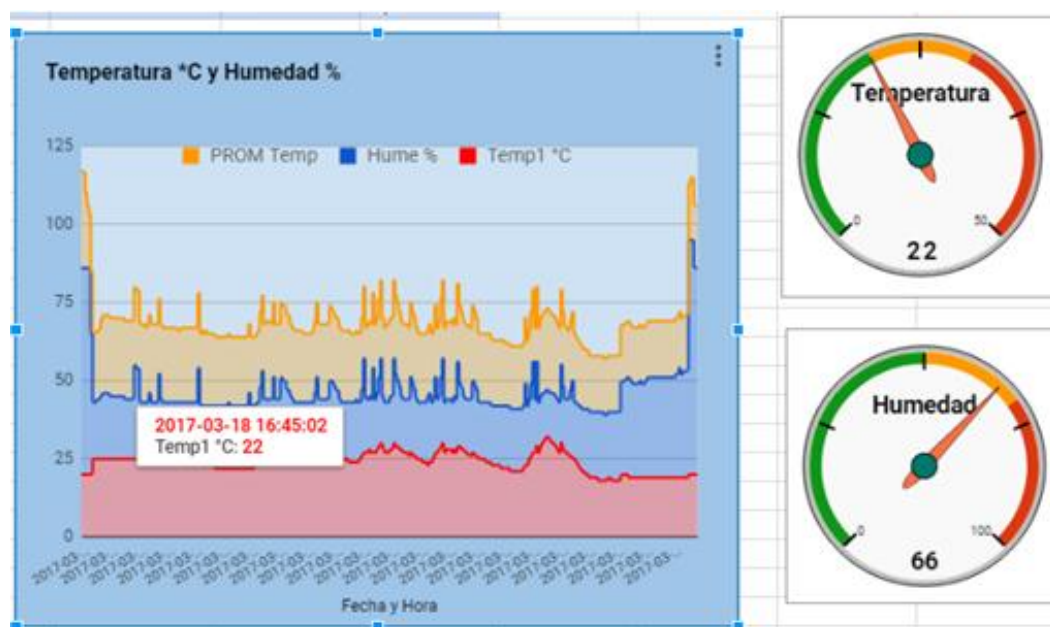


Figura 121. Histórico de Temperatura y Humedad Sensor 1.

Fuente: Autor.

### 3.2. PRUEBAS DE MEDICIÓN DE TEMPERATURA Y HUMEDAD RELATIVA SENSOR 2

Se toman medidas de temperatura en el lapso de una hora cada cinco minutos, los datos fueron tomados el 18/3/2017 de 16:00:00 a 17:00:00 como se muestra en la Tabla 14.

Tabla 14. Datos de Temperatura y Humedad 2 en función del Tiempo

FECHA Y HORA	TEMPERATURA °C	HUMEDAD RELATIVA %
2017-03-18 16:00:24	22	55
2017-03-18 16:05:33	22	55
2017-03-18 16:10:41	22	55
2017-03-18 16:15:49	23	60
2017-03-18 16:20:58	23	65
2017-03-18 16:25:08	23	70
2017-03-18 16:30:16	24	75



2017-03-18 16:35:08	24	75
2017-03-18 16:40:33	24	75
2017-03-18 16:45:42	24	75
2017-03-18 16:55:50	25	75

Fuente: Autor.

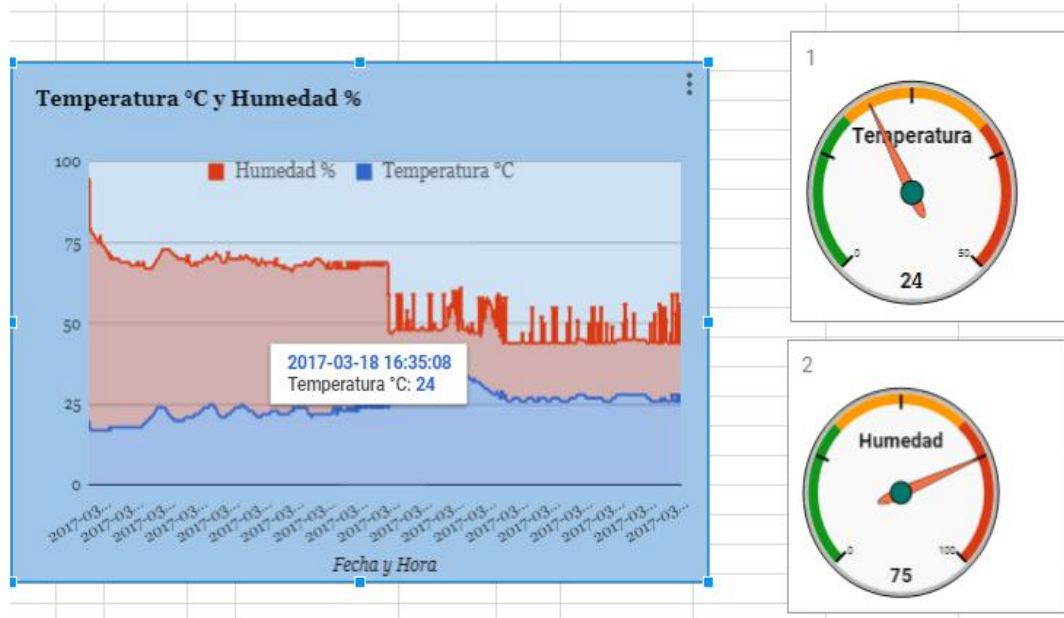


Figura 122. Histórico de Temperatura y Humedad Sensor 2.

Fuente: Autor.

En la interfaz web se obtiene el histórico de la temperatura para los valores tabulados, como se muestra en la Figura 122, se verifica también que el envío y recepción de datos a la aplicación de google spreadsheet son exitosos.

El valor de Temperatura y Humedad Relativa del Sensor 2 oscilan de acuerdo a la temperatura dentro del invernadero, el sensor se encuentra colocado en la parte superior central del invernadero otorgando valores ambientales reales.

### 3.3. PRUEBAS DE MEDICIÓN DE TEMPERATURA Y HUMEDAD RELATIVA SENSOR 3

Se toman medidas de temperatura en el lapso de una hora cada cinco minutos, los datos fueron tomados el 18/3/2017 de 16:00:00 a 17:00:00 como se muestra en la Tabla 15.

Tabla 15. Datos de Temperatura y Humedad 3 en función del Tiempo.

FECHA Y HORA	TEMPERATURA °C	HUMEDAD RELATIVA %
2017-03-18 16:00:08	17	55
2017-03-18 16:05:16	17	55
2017-03-18 16: 10:25	17	55
2017-03-18 16:15:33	18	60
2017-03-18 16:20:42	18	65
2017-03-18 16:25:50	18	70
2017-03-18 16:30:59	18	73
2017-03-18 16:35:08	19	73
2017-03-18 16:40:16	19	73
2017-03-18 16:45:25	20	75
2017-03-18 16:50:33	20	75
2017-03-18 16:55:53	20	75

Fuente: Autor.

En la interfaz web se obtiene el histórico de la temperatura para los valores tabulados, como se muestran en la Figura 123, se verifica también que la transmisión y recepción de datos a la aplicación de google spreadsheet son exitosos. El valor de Temperatura y Humedad Relativa del Sensor 3 oscilan de acuerdo a la temperatura dentro del invernadero, el sensor se encuentra colocado en el extremo superior derecho otorgando valores ambientales reales.

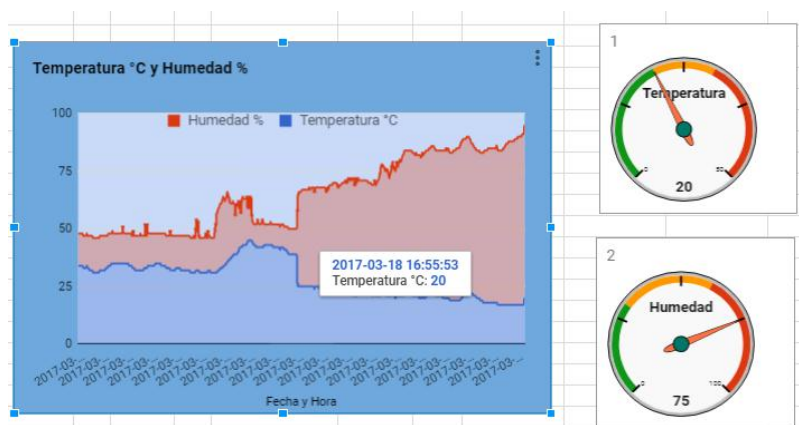


Figura 123. Histórico de Temperatura y Humedad Sensor 3.

Fuente: Autor.

### 3.4. PRUEBAS DE MEDICIÓN DE TEMPERATURA PROMEDIO DE LOS SENSORES 1, 2, 3.

Se toman medidas de temperatura en el lapso de una hora cada cinco minutos, los datos fueron tomados el 18/3/2017 de 16:00:00 a 17:00:00 pm.

El promedio de las los 3 sensores se muestra en la Tabla 16.

Tabla 16. Temperatura Promedio de los Sensores 1,2,3.

FECHA Y HORA	TEMPERATURA PROMEDIO
2017-03-18 16:00:01	24
2017-03-18 16:05:11	24
2017-03-18 16:10:21	24
2017-03-18 16:15:32	24
2017-03-18 16:20:42	23
2017-03-18 16:25:52	23
2017-03-18 16:30:02	23
2017-03-18 16:35:12	23
2017-03-18 16:40:23	23
2017-03-18 16:45:12	22
2017-03-18 16:50:46	22
2017-03-18 16:55:35	22

Fuente: Autor.

Finalmente del interfaz web se obtiene el histórico del Promedio de las 3 Temperaturas cómo se muestra en la Figura 124, que indica que los datos son exitosamente adquiridos por los sensores, enviados a través del módulo ESP8266 y procesados en google script y tomados por el interfaz web de google spreadsheet. Este promedio de los 3 sensores se utiliza como referencia para activar y desactivar de forma automática o manual cada uno de los actuadores del prototipo de invernadero.

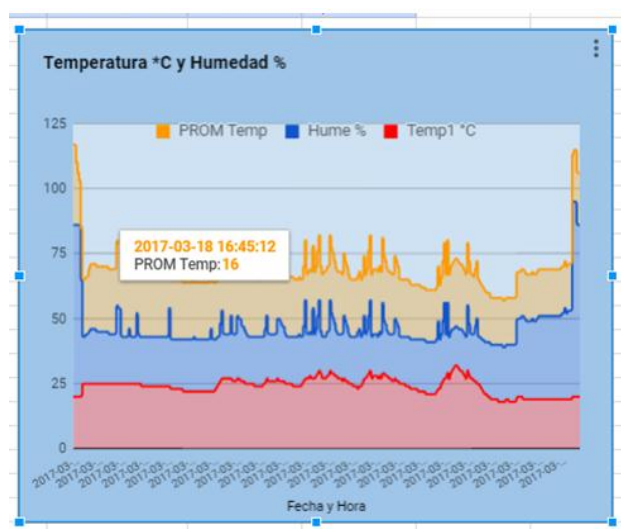


Figura 124. Temperatura Promedio de los Sensores 1,2,3.

Fuente: Autor.

### 3.5. PRUEBAS DE FUNCIONAMIENTO AUTOMÁTICO Y MANUAL DEL PROTOTIPO DE INVERNADERO.

Para realizar las pruebas de funcionamiento se verifica que el invernadero funcione en modo automático y manual. Dependiendo del modo escogido se comprueba que el sistema de control actúe para mantener al invernadero dentro de los rangos de trabajo específicos de la variable medida para el cultivo de lechuga.

#### 3.5.1. MODO AUTOMÁTICO PARA LA ACTIVACIÓN DE LOS ACTUADORES

En la Tabla 17 se detalla el resultado obtenido para activación automática de los actuadores.

Tabla 17. Funcionamiento Automático del Prototipo de Invernadero.

ITEM	DESCRIPCIÓN DE TAREAS	ACCIONAMIENTO	MÉTODO DE COMPROBACIÓN	SI	NO
Temperatura	Controlarla temperatura del invernadero	Interfaz WEB (Google Spreadsheet y Página Web) Aplicación Android	Observar el valor de Temperatura Promedio en la interfaz web y aplicación Android.	x	
Calefactor	Controlar el	Interfaz WEB	La temperatura		

	encendido y Apagado	(Google Spreadsheet y Página Web)	promedio disminuye al valor de temperatura min se enciende el calefactor.	x	
		Aplicación Android			
Ventilador	Controlar el Encendido y Apagado	Interfaz WEB (Google Spreadsheet y Página Web)	La temperatura promedio excede al valor de temperatura max se enciende el ventilador.	x	
		Aplicación Android			
electroválvula	Controlar el encendido y apagado de la electroválvula y la cantidad de riego de agua, haciendo uso del pluviómetro	Interfaz WEB (Google Spreadsheet y Página Web)	Observar el valor de precipitaciones en la interfaz web y en la aplicación Android	x	
		Aplicación Android	Observar el encendido de la electroválvula.		

Fuente. Autor.

### 3.5.2. MODO MANUAL PARA LA ACTIVACIÓN DE LOS ACTUADORES

En la Tabla 18 se muestra el funcionamiento Manual del Prototipo de Invernadero.

Tabla 18. Funcionamiento Manual del Prototipo de Invernadero.

ITEM	DESCRIPCIÓN DE TAREAS	ACCIONAMIENTO	MÉTODO DE COMPROBACIÓN	SI	NO
Ingreso de los valores de referencia	Cambiar el valor de temperatura del	Interfaz WEB (Google Spreadsheet y página web)	Cambiar el valor de temperatura máxima y mínima		

de temperatura máxima y mínima	invernadero	Aplicación Android	en la hoja de cálculo de Google, Cambiar a través de la pagina web el accionamiento ON/OFF y en la aplicación Android ON/OFF de cada uno de los actuadores	x	
Ingreso del valor de referencia de humedad relativa máxima y mínima	Cambiar el valor de humedad relativa del invernadero	Interfaz WEB (Google Spreadsheet y pagina web) Aplicación Android	Cambiar el valor de humedad relativa máxima y mínima	x	
Ingreso el valor de hora de riego y precipitación	Cambiar el valor de riego de agua y precipitación	Interfaz WEB (Google Spreadsheet)	Cambiar el valor de riego de agua de inicio y finalización , cambiar el valor de precipitación requerida	x	

Fuente: Autor.

Con estas pruebas se ha comprobado que el sistema reacciona adecuadamente a los cambios de temperatura que los 3 sensores detectan, haciendo que se mantenga la temperatura ideal en el prototipo de invernadero con el accionamiento del ventilador o del calefactor de modo automático y manual de acuerdo a los rangos

específicos de temperatura máxima y mínima del cultivo. Se comprueba también el accionamiento de la electroválvula, que mediante el sistema de pluviometría censa y controla la cantidad de agua que se requiere para regar cualquier tipo de cultivo.

### 3.6. PRUEBAS DE FUNCIONAMIENTO DE LA INTERFAZ WEB (GOOGLE SPREADSHEET Y PÁGINA WEB)

#### 3.6.1. GOOGLE SPREADSHEET

En la hoja de cálculo o google spreadsheet alojado en las aplicaciones de google drive como se muestra en la Figura 125 se visualiza las variables al ser sensadas en el invernadero, es decir la temperatura, humedad, precipitación, el encendido y apagado de los actuadores y el modo de control en que el prototipo de invernadero va a funcionar.

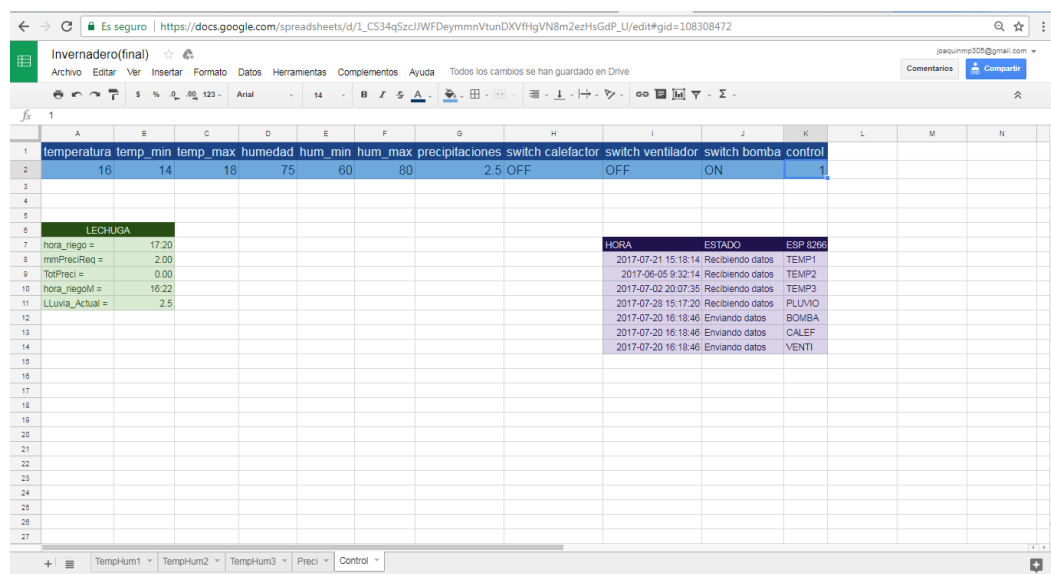


Figura 125. Variables en Ejecución del Invernadero.

Fuente: Autor.

En la Figura 126 se muestra la temperatura y humedad relativa promedio de los 3 sensores, también se muestran los rangos de temp\_min y temp\_max como también los rangos de hum\_min y hum\_max que están establecidos según los valores específicos del cultivo. El valor de temperatura promedio es la referencia para que los actuadores cumplan la función específica. En este caso se tiene que el valor es de 16, si el valor de temperatura promedio es menor a la temp\_min el

calefactor se acciona mientras que el ventilador permanece apagado y si el valor de temperatura promedio es mayor a la temp\_max el ventilador se acciona mientras que el calefactor permanece apagado, como se observa que la temperatura promedio está entre el rango específico de temp\_min y temp\_max no se acciona ninguno de los actuadores como se muestra en la Figura 127. Si el modo de control es automático se visualiza 1 y si es en modo de control manual se visualiza 0, también se observa las precipitaciones, es decir la cantidad de agua censada que se requiere para el cultivo.

temperatura	temp_min	temp_max	humedad	hum_min	hum_max
16	14	18	75	60	80

Figura 126. Temperatura y Humedad Relativa Promedio y Rangos Min/Max.

Fuente: Autor.

precipitaciones	switch calefactor	switch ventilador	switch bomba	control
2.5	OFF	OFF	OFF	1

Figura 127. Accionamiento de los Actuadores ON/OFF y modo de Control Automático/Manual.

Fuente: Autor.

Para el prototipo de invernadero se cultiva lechuga, entre los 18°C a 20°C es la temperatura de germinación de la lechuga, los rangos de crecimiento son: en el día entre 14°C a 18°C, y en la noche 5°C a 8°C. En la fase de acogollado los rangos de temperatura son: en el día 12°C y de 3°C a 5°C por la noche. Los cultivos de lechuga soporta temperaturas elevadas hasta los 30°C y temperaturas mínimas hasta los -6°C, cuando existe exceso de temperatura baja la lechuga pierde su coloración normal.

En cuanto a la humedad relativa conveniente para la lechuga es del 60% al 80%, aunque en determinados momentos agradece menos del 60%. Los riegos se dan de manera frecuente y con poca cantidad de agua, procurando que el suelo quede aparentemente seco en la parte superficial, para evitar podredumbres del cuello. En la primera semana de cultivo, si se resecaen los cepellones se darán riegos de 1 a 2 litros/m<sup>2</sup>. A continuación, y hasta el estado de 16 a 18 hojas se regará con



dosis bajas de 4 a 8 l/m<sup>2</sup>. Posteriormente, hasta el final del cultivo, deben utilizarse dosis de riego de 8 a 20 l/m<sup>2</sup>. En la Tabla 19 se muestra la hora de riego actual, las precipitaciones requeridas en mm, el total de precipitaciones, la hora de riego máxima y el riego actual.

Tabla 19. Parámetros de Configuración para el sistema de Pluviometría.

LECHUGA	
hora_riego =	16:30
mmPreciReq =	2.00
TotPreci =	2.00
hora_riegoM =	16:33
LLuvia_Actual =	2.5

Fuente: Autor.

En la siguiente Tabla 20 se muestra los resultados del estado actual y la hora de cada uno de los módulos ESP8266 con sus respectivos actuadores y sensores que se comunican a través de la Red WIFI.

Tabla 20. Estado Actual de los Actuadores y Sensores.

HORA	ESTADO	ESP 8266
2017-03-18 16:18:14	Recibiendo datos	TEMP1
2017-03-18 16:19:10	Recibiendo datos	TEMP2
2017-03-18 16:20:35	Recibiendo datos	TEMP3
2017-03-18 16:17:20	Recibiendo datos	PLUVIO
2017-03-18 16:39:46	Enviando datos	ELECTRO
2017-03-18 16:45:05	Enviando datos	CALEF
2017-03-18 16:53:16	Enviando datos	VENTI

Fuente: Autor.

### 3.6.2. PÁGINA WEB

En el monitoreo a través de la interfaz web como se observa en la Figura 128 se obtiene los valores de temperatura, humedad y precipitación, a su vez también indica el modo de control que se desea utilizar, es decir automático y manual. Para establecer en modo automático al sistema, se pulsa el botón control, este cambia de color a verde indicando que está en funcionamiento y a color rojo que el sistema se establece en modo manual. Cada uno de los botones de los actuadores hace la misma función es decir:

Verde - Prendido y en el google spreadsheet u hoja de cálculo ON

Rojo - Apagado y en el google spreadsheet u hoja de cálculo OFF



Figura 128. Interfaz Web.

Fuente: Autor.

En modo automático se observa que se enciende el ventilador ya que el valor de la temperatura promedio es menor que la temp\_min. En la Figura 129 se muestra los resultados obtenidos.

### Invernadero IoT

Link para administración : [Hoja de cálculo: Administración, Monitoreo y Control del Invernadero](#)

Temperatura	Humedad	Precipitaciones	Switch Calefactor	Switch Ventilador	Switch Bomba	Control
13	75	2.5	Heater Switch	Cooler Switch	Pump Switch	Automatico/Manual

Figura 129. Actuador Encendido en Modo Automático.

Fuente: Autor.

Al momento de enviar aire caliente al prototipo de invernadero se observa que la temperatura promedio aumenta haciendo que se apague el calefactor al momento que entra en el rango establecido de  $temp\_min$  y  $temp\_max$ , La temperatura promedio entra en el rango establecido haciendo que el ambiente en el invernadero sea estable acorde a los valores mínimos y máximos de temperatura que necesita el cultivo. En la Figura 130 se muestra los resultados obtenidos

## Invernadero IoT

Link para administración : [Hoja de cálculo: Administración, Monitoreo y Control del Invernadero](#)

Temperatura	Humedad	Precipitaciones	Switch Calefactor	Switch Ventilador	Switch Bomba	Control
16	75	2.5	Heater Switch	Cooler Switch	Pump Switch	Automatico/Manual

Figura 130. Actuadores Apagados en Modo Automático.

Fuente: Autor.

Cuando aumenta la temperatura promedio en el prototipo de invernadero se observa que es mayor a la  $temp\_max$  en la hoja de cálculo o google spreadsheet, el ventilador se accionan haciendo que entre aire frío, expulsado el aire caliente con el otro ventilador y así mantener la temperatura ideal en el invernadero. En la Figura 131 se muestra el resultado obtenido.

## Invernadero IoT

Link para administración : [Hoja de cálculo: Administración, Monitoreo y Control del Invernadero](#)

Temperatura	Humedad	Precipitaciones	Switch Calefactor	Switch Ventilador	Switch Bomba	Control
19	75	2.5	Heater Switch	Cooler Switch	Pump Switch	Automatico/Manual

Figura 131. Actuator Encendido en Modo Automático.

Fuente: Autor.

Para que el prototipo de invernadero funcione de modo manual se acciona el botón control y así el usuario activa y desactiva cada uno de los actuadores, independientemente que la temperatura promedio se encuentre en el rango establecido de  $temp\_min$  y  $temp\_max$ . En la Figura 132 se muestra los resultados obtenidos encendiendo el ventilador con temperatura promedio 16.

# Invernadero IoT

Link para administración : [Hoja de cálculo: Administración, Monitoreo y Control del Invernadero](#)

Temperatura	Humedad	Precipitaciones	Switch Calefactor	Switch Ventilador	Switch Bomba	Control
16	75	2.5	Heater Switch	Cooler Switch	Pump Switch	Automatico/Manual

Figura 132. Actuador Encendido en Modo Manual.

Fuente: Autor.

## 3.7. PRUEBAS DE FUNCIONAMIENTO DE LA APLICACIÓN ANDROID

En estas pruebas de funcionamiento se realiza de forma automática el encendido del calefactor, el encendido del ventilador y el apagado de los actuadores cuando la temperatura promedio se encuentra entre los rangos min y max.

En la Figura 133 se muestra que los actuadores están apagados y la temperatura promedio es de 16°C.



Figura 133. Modo Automático Aplicación Android.

Fuente: Autor.

### 3.7.1. ENCENDIDO AUTOMÁTICO DEL VENTILADOR.

En la Figura 134 se muestra que el VENTILADOR se enciende a una temperatura de 19°C.

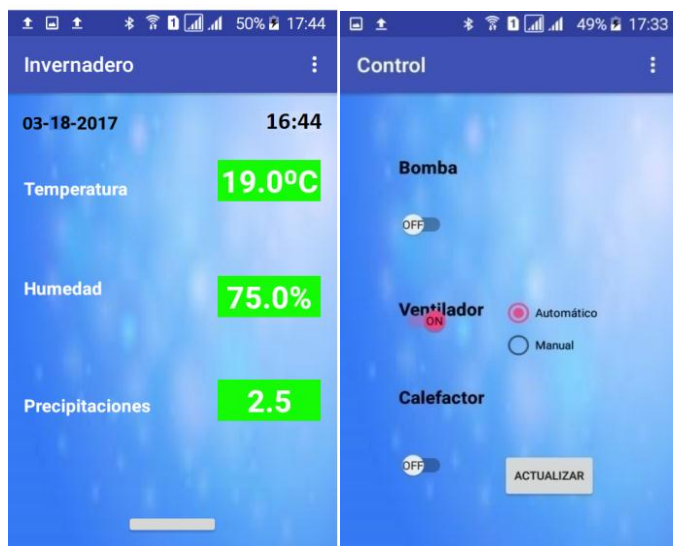


Figura 134. Encendido del Ventilador Aplicación Android.

Fuente: Autor.

### 3.7.2. ENCENDIDO AUTOMÁTICO DEL CALEFACTOR.

En la Figura 135 se muestra que el CALEFACTOR se enciende ON a una temperatura de 13°C.

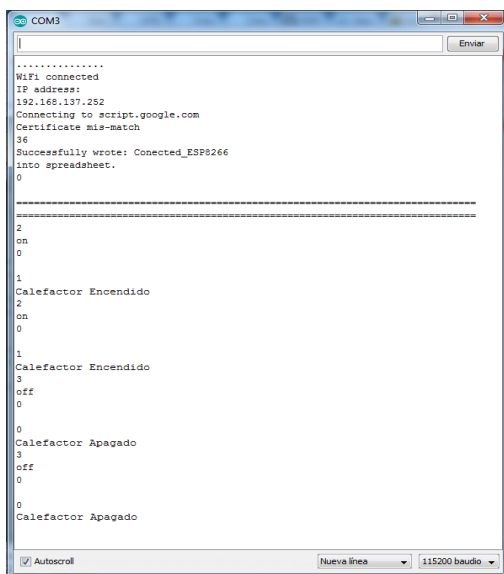


Figura 135. Encendido del Calefactor Aplicación Android.

Fuente: Autor.

Las salidas del puerto COM a través de la consola de Arduino y conectados a la placas de los actuadores se muestran a continuación en la Figura 136 y Figura 137.

## Calefactor

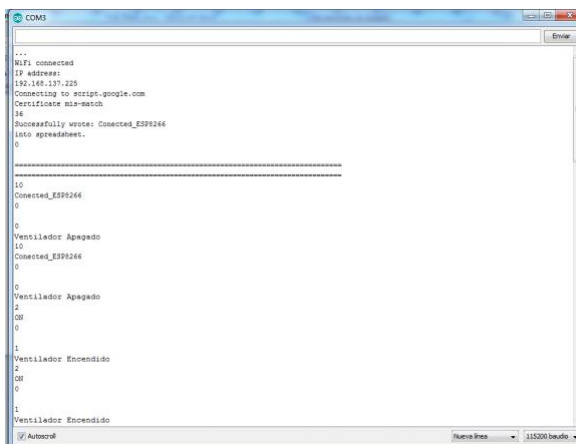


```
.....  
WiFi connected  
IP address:  
192.168.137.252  
Connecting to script.google.com  
Certificate mis-match  
36  
Successfully wrote: Conected_ESP8266  
into spreadsheet.  
0  
-----  
2  
on  
0  
1  
Calefactor Encendido  
2  
on  
0  
1  
Calefactor Encendido  
3  
off  
0  
0  
Calefactor Apagado  
3  
off  
0  
0  
Calefactor Apagado
```

Figura 136. Salida del Puerto COM del Calefactor.

Fuente: Autor.

## Ventilador



```
.....  
WiFi connected  
IP address:  
192.168.137.252  
Connecting to script.google.com  
Certificate mis-match  
36  
Successfully wrote: Conected_ESP8266  
into spreadsheet.  
0  
-----  
10  
Conected_ESP8266  
0  
0  
Ventilador Apagado  
10  
Conected_ESP8266  
0  
0  
Ventilador Apagado  
2  
on  
0  
1  
Ventilador Encendido  
2  
on  
0  
1  
Ventilador Encendido
```

Figura 137. Salida del Puerto COM del Ventilador.

Fuente: Autor.

### 3.8. RESULTADOS DEL PROTOTIPO DE INVERNADERO

El crecimiento de la planta es exitoso ya que se puede observar su crecimiento a lo largo del tiempo. Primero se procede a la preparación y siembra de las semillas, para ello se prepara la tierra con abono orgánico que luego es colocado en los recipientes y puestos dentro del invernadero para que el proceso de crecimiento de

la planta sea efectuado adecuadamente. En la Figura 138 se muestra la fase inicial del prototipo de invernadero y el cultivo.



Figura 138. Fase inicial del Cultivo en el Inverndaero.

Fuente: Autor.

En la segunda fase se observa en la Figura 139 que el proceso de la lechuga se encuentra en crecimiento y no presenta anomalías en cuento al producto que será cultivado a la quinta semana desde el período de siembra a cosecha que comprende de 60 a 120 días.

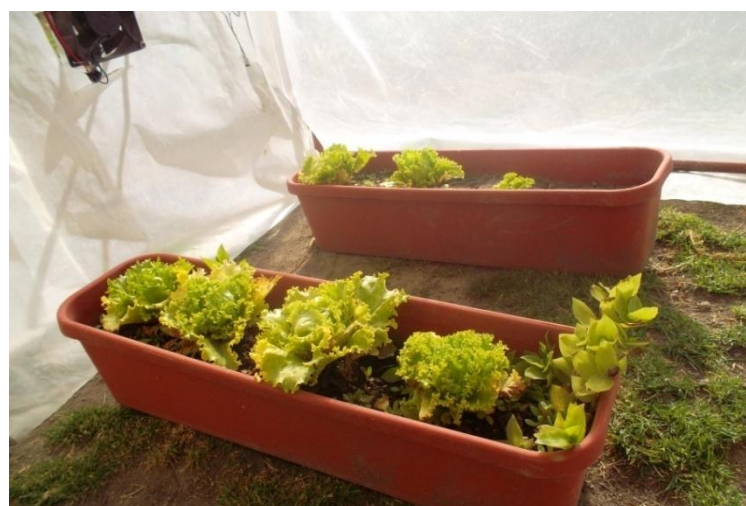


Figura 139. Periodo de Crecimiento de Lechugas.

Fuente: Autor.

## CAPITULO IV

### 4. DISCUSIÓN

La implementación de los sensores y actuadores están basados para la tecnificación del almacenamiento de la información, el monitoreo y el control en la agricultura y específicamente en los invernaderos, se utilizó sensores electrónicos precisos y servicios web que interactúan eficazmente, mediante el análisis e investigación en la tecnología permite al usuario un manejo acorde a las necesidades básicas del cultivo en invernaderos. La base de esta implementación es el Internet de las Cosas (IoT) y la red WIFI ya que la comunicación a través del internet entre el usuario y los dispositivos permite hacerlo sin afectar el lugar donde se localicen, el acceso es por medio de las herramientas de google: drive, appscript y la hoja de cálculo o spreadsheet, que permite almacenar la información de los sensores, monitorear y controlar el invernadero en tiempo real, y el interfaz mediante la página Web así como también una aplicación para el sistema operativo Android. La red WIFI tiene gran cobertura en zonas sin obstáculos, lo que permite una transmisión y recepción de paquetes de información a largas distancias, esto proporciona una conexión rápida y segura para el equipo que establece la comunicación entre los sensores y actuadores.

El IoT permite que los objetos puedan tener una comunicación con el mundo exterior para percibir el entorno e interactuar con él, ya sea mediante sensores (velocidad, temperatura, humedad); cambiar algo en el ambiente por medio de actuadores, y almacenar datos para comunicar lo que se ha captado mediante los sensores. Las aplicaciones IoT están estrechamente relacionadas con las plataformas de Computación en la Nube para proveer características de almacenamiento con un mejor intercambio de la información, los servicios web y las herramientas que ofrece google como spreadsheet (hoja de cálculo), google script, drive resultan ser adecuados y de fácil manejo para esta implementación.



## CAPITULO V

### 5. CONCLUSIONES Y RECOMENDACIONES

#### 5.1. CONCLUSIONES

- El estudio previo a las diferentes técnicas de cultivo en semillero, ha permitido seleccionar la forma más adecuada para implementar un cultivo de este tipo. Es de gran importancia conocer los rangos adecuados de las variables como la de temperatura, humedad relativa, sensor magnético (reed switch) para tener un adecuado rendimiento del sistema.
- El prototipo cubre los requerimientos necesarios para el crecimiento del semillero, almacenando controlando y monitoreando las variables ambientales dentro del invernadero.
- Mediante el uso de la plataforma Arduino se puede implementar distintos tipos de algoritmos para controlar procesos tales como ON – OFF, así como también implementar sistemas de comunicaciones inalámbricas para ello es importante manipular los registros y librerías brindadas por la plataforma Arduino.
- Mediante el MÓDULO ESP8266 que utiliza el protocolo HTTPS para enviar información desde el prototipo al servidor WEB, hoja de cálculo de google spreadsheet, se puede monitorear dispositivos de forma remota a través de internet, almacenando la información que recoge el sistema microprocesado para la adquisición de valores de sensores y estados de actuadores.
- Mediante la implementación de la interfaz gráfica alojada en la web desarrollada utilizando HTML, Java Scrip, se muestra los valores almacenados en una hoja de cálculo de google spreadsheet, el mismo que se puede configurar los parámetros de temperatura, humedad relativa, pluviometría implementados en el Módulo WIFI ESP8266, que su vez monitorea y controla los actuadores manualmente y automáticamente.
- El Sistema Operativo Android se perfecciona a futuro. Android ofrece un entorno de desarrollo que facilita la implementación de aplicaciones y versiones de manera ágil y práctico, aprovechando al máximo las características de cada dispositivo móvil.

## **5.2.RECOMENDACIONES**

- Se recomienda este tipo de invernadero en las ciudades debido al reducido espacio que ocupa y a la facilidad de operación por parte del usuario, además promueve el consumo de alimentos nutritivos
- Al momento de programar rutinas en el IDE de Arduino es importante conocer que valores devuelve y los tipos de datos que utiliza para evitar conflictos al momento de intercambiar información entre ellas.
- Se recomienda utilizar el prototipo para probar diferentes tipos de cultivo, para poder observar de mejor manera el desenvolvimiento de cada tipo de planta.

## CAPITULO VI

### 6. BIBLIOGRAFÍA

- [1] *EngineersGarage*. (2012). Obtenido de EngineersGarage: <https://www.engineersgarage.com/articles/reed-switch-specifications>
- [2] Al, A. (19 de 03 de 2014). Obtenido de [http://www.academia.edu/6180549/Investigacion\\_cientifica](http://www.academia.edu/6180549/Investigacion_cientifica)
- [3] Infoagro Systems, S. (2016). *Agricultura Ecológica*. Obtenido de Requerimientos Edafoclimáticos: <http://www.infoagro.com>
- [4] Jashton. (2009). *the-internet-of-things*. Obtenido de <http://kevinjashton.com/2009/06/22/the-internet-of-things>
- [5] Renom, M. (2011). Obtenido de [http://meteo.fisica.edu.uy/Materias/PBMA/PBMA\\_teotico/Bolilla6-PLUVIOMETRIA.pdf](http://meteo.fisica.edu.uy/Materias/PBMA/PBMA_teotico/Bolilla6-PLUVIOMETRIA.pdf)
- [6] Senel, M. (2012). *Wi-Fi Enabled Sensors for Internet of Things*. Obtenido de <http://cpham.perso.univ-pau.fr/ENSEIGNEMENT/PAU-UPPA/RHD/PAPER/OLD/WifiSensor.pdf>
- [7] Systems, E. (2015). *Espressif Systems IOT Team*. Obtenido de <http://bbs.espressif.com/>
- [8] TP-LINK TECHNOLOGIES CO., L. (2012). *TL-MR3040 USER GUIDE*. Obtenido de <http://www.tp-link.com>
- [9] UK, D.-R. (2010). *DHT11 Humidity & Temperature Sensor*. Obtenido de <http://www.micropik.com/PDF/dht11.pdf>
- [10] Velázquez. (29 de 1 de 2015). *Hortalizas*. Obtenido de Eficiencia en el uso del agua: <http://www.hortalizas.com>
- [11] Vildósola, E. (s.f.). <http://www.aie.cl>. Recuperado el 2016, de <http://www.aie.cl/files/file/comites/ca/abc/actuadores.pdf>

## CAPÍTULO VII

### 7. ANEXOS

#### CÓDIGO DE PROGRAMACIÓN MÓDULO ESP8266 SENSOR 1

```
#include <ESP8266WIFI.h>
#include "HTTPSRedirect.h"
#include "DHT.h"
#define DHTPIN 2 // PIN GPIO2 del ESP8266
#define DHTTYPE DHT11 //Definir tipo de sensor DHT11
DHT dht(DHTPIN,DHTTYPE);
const char* ssid = "retlawy2k";
const char* password = "t3$i$un@ch";
const char* host = "script.google.com";
const char* googleRedirHost = "script.googleusercontent.com";
const char GScriptId =
"AKfycbwyeQWDTRWNpMPWP15l2DLj6hp101ZmibQr9lA-OzLSeB6qYLg";
//exec ID
const int httpsPort = 443; // Puerto https
const char* fingerprint = "76 63 20 C3 BC 52 21 DD D9 DB EF 33 21 81 F1 6E
7A 69 E2 92"; //mio
String url = String("/macros/s/") + GScriptId +
"/exec?value=Conected_ESP8266";
String url3 = String("/macros/s/") + GScriptId + "/exec?read";
void setup() {
pinMode(2, OUTPUT);
pinMode(0, OUTPUT);
Serial.begin(115200);
dht.begin();
Serial.println();
Serial.print("Conectando a WIFI: ");
Serial.println(ssid);
Serial.flush();
WIFI.begin(ssid, password);
while (WIFI.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WIFI conectado");
Serial.println("Dirección IP: ");
Serial.println(WIFI.localIP());
HTTPSRedirect client(httpsPort);
Serial.print("Connecting to ");
Serial.println(host);
bool flag = false;
for (int i=0; i<5; i++){
int retval = client.connect(host, httpsPort);
```

```

if (retval == 1) {
flag = true;
break;
}
Else
Serial.println("Connección fallida. Reintentando...");
}
Serial.flush();
if (!flag){
Serial.print("No se puede conectar al servidor: ");
Serial.println(host);
Serial.println("Saliendo...");
return;
}
Serial.flush();
if (client.verify(fingerprint, host)) {
Serial.println("Certificado emparejado.");
} else {
Serial.println("Certificado no emparejado");
}
Serial.println("Fallo al leer desde el sensor DHT!");
return;
}
Serial.print("Humedad: ");
Serial.print(humidityData);
Serial.print(" %\t");
Serial.print("Temperatura en Celsius: ");
Serial.print(ceIData);
Serial.print(" *C\n");
char outstr1[15];
dtostrf(ceIData,4, 2, outstr1);
String celsius= outstr1;
char outstr2[15];
dtostrf(humidityData,4, 2, outstr2);
String humed= outstr2;
HTTPSRedirect client(httpsPort);
if (!client.connected())
client.connect(host, httpsPort);
url = String("/macros/s/") + GScriptId + "/exec?value="+celsius + "&humed=" +
humed;
client.printRedir(url, host, googleRedirHost);
Serial.println("=====");
delay(5000);
}

```

## **CÓDIGO DE PROGRAMACIÓN MÓDULO ESP8266 SENSOR 2**

```

#include <ESP8266WIFI.h>
#include "HTTPSRedirect.h"
#include "DHT.h"
#define DHTPIN 2 // PIN GPIO2 del ESP8266
#define DHTTYPE DHT11 //Definir tipo de sensor DHT11
DHT dht(DHTPIN,DHTTYPE);
const char* ssid = "retlawy2k";
const char* password = "t3$i$un@ch";
const char* host = "script.google.com";
const char* googleRedirHost = "script.googleusercontent.com";
const          char          *GScriptId          =
"AKfycbzm4IcM1upeNTWszhj3anS4FHMMrRuxsiKWJDWQ2fFLKN6aIgXG";
const int httpsPort = 443; // Puerto https
const char* fingerprint = "76 63 20 C3 BC 52 21 DD D9 DB EF 33 21 81 F1 6E
7A 69 E2 92";
String url = String("/macros/s/") + GScriptId +
"/exec?value=Conected_ESP8266";
String url3 = String("/macros/s/") + GScriptId + "/exec?read";
void setup() {
pinMode(2, OUTPUT);
pinMode(0, OUTPUT);
Serial.begin(115200);
dht.begin();
Serial.println();
Serial.print("Conectando a WIFI: ");
Serial.println(ssid);
Serial.flush();
WIFI.begin(ssid, password);
while (WIFI.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WIFI conectado");
Serial.println("Dirección IP: ");
Serial.println(WIFI.localIP());
HTTPSRedirect client(httpsPort);
Serial.print("Connecting to ");
Serial.println(host);
bool flag = false;
for (int i=0; i<5; i++){
int retval = client.connect(host, httpsPort);
if (retval == 1) {
flag = true;
break;
}
else
Serial.println("Connección fallida. Reintentando...");
}
}

```

```

}
Serial.flush();
if (!flag){
Serial.print("No se puede conectar al servidor: ");
Serial.println("Saliendo...");
return;
}
Serial.flush();
if (client.verify(fingerprint, host)) {
Serial.println("Certificado emparejado.");
} else {
Serial.println("Certificado no emparejado");
}
Serial.println("Fallo al leer desde el sensor DHT!");
return;
}
Serial.print("Humedad: ");
Serial.print(humidityData);
Serial.print(" %\t");
Serial.print("Temperatura en Celsius: ");
Serial.print(CELData);
Serial.print(" *C\n");
char outstr1[15];
dtostrf(CELData,4, 2, outstr1);
String celsius= outstr1;
char outstr2[15];
HTTPSRedirect client(httpsPort);
if (!client.connected())
client.connect(host, httpsPort);
url = String("/macros/s/") + GScriptId + "/exec?value="+celsius + "&humed=" +
humed;
client.printRedir(url, host, googleRedirHost);
Serial.println("=====
");
delay(5000);
}

```

### CÓDIGO DE PROGRAMACIÓN MÓDULO ESP8266 SENSOR 3

```
#include <ESP8266WIFI.h>
#include "HTTPSRedirect.h"
#include "DHT.h"
#define DHTPIN 2 // PIN GPIO2 del ESP8266
#define DHTTYPE DHT11 //Definir tipo de sensor DHT11
DHT dht(DHTPIN,DHTTYPE);
const char* ssid = "retlawy2k";
const char* password = "t3$i$un@ch";
const char* host = "script.google.com";
const char* googleRedirHost = "script.googleusercontent.com";
const          char          *GScriptId          =
"AKfycbzbvOGN7sT7ANTa6ENLckiLscQyjS3NMzTIEgJUzFPQvbBoP3HX";
const int httpsPort = 443; // Puerto https
const char* fingerprint ="76 63 20 C3 BC 52 21 DD D9 DB EF 33 21 81 F1 6E
7A 69 E2 92";
String url          =          String("/macros/s/")          +          GScriptId          +
"/exec?value=Conected_ESP8266";
String url3 = String("/macros/s/") + GScriptId + "/exec?read";
void setup() {
pinMode(2, OUTPUT);
pinMode(0, OUTPUT);
Serial.begin(115200);
dht.begin();
Serial.println();
Serial.print("Conectando a WIFI: ");
Serial.println(ssid);
Serial.flush();
WIFI.begin(ssid, password);
while (WIFI.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WIFI conectado");
Serial.println("Dirección IP: ");
Serial.println(WIFI.localIP());
HTTPSRedirect client(httpsPort);
Serial.print("Connecting to ");
Serial.println(host);
bool flag = false;
for (int i=0; i<5; i++){
int retval = client.connect(host, httpsPort);
if (retval == 1) {
flag = true;
break;
}
}
```



```

else
Serial.println("Connección fallida. Reintentando...");
}
Serial.flush();
if (!flag){
Serial.print("No se puede conectar al servidor: ");
Serial.println(host);
Serial.println("Saliendo...");
return;
}
Serial.flush();
if (client.verify(fingerprint, host)) {
Serial.println("Certificado emparejado.");
} else {
Serial.println("Certificado no emparejado");
}
Serial.println("Fallo al leer desde el sensor DHT!");
return;
}
Serial.print("Humedad: ");
Serial.print(humidityData);
Serial.print(" %\t");
Serial.print("Temperatura en Celsius: ");
Serial.print(CELData);
Serial.print(" *C\n");
char outstr1[15];
dtostrf(CELData,4, 2, outstr1);
String celsius= outstr1;
char outstr2[15];
dtostrf(humidityData,4, 2, outstr2);
String humed= outstr2;
HTTPSRedirect client(httpsPort);
if (!client.connected())
client.connect(host, httpsPort);
url = String("/macros/s/") + GScriptId + "/exec?value="+celsius + "&humed=" +
humed;
client.printRedir(url, host, googleRedirHost);
Serial.println("=====
");
delay(5000);
}

```

## CÓDIGO DE PROGRAMACIÓN MÓDULO ESP8266 SENSOR 4

```
#include <ESP8266WIFI.h>
#include "HTTPSRedirect.h"
#define PIN_RAIN_GAUGE 0
volatile int numClicksRainGauge = 0;
float x;
long previousMillis = 0;
long interval = 9990;
volatile unsigned long raintime, rainlast, raininterval;
const char* ssid = "retlawy2k";
const char* password = "t3$i$un@ch";
const char* host = "script.google.com";
const char* googleRedirHost = "script.googleusercontent.com";
const char *GScriptId = "AKfycbxDDAZbg-
PNchJqKjZrSA81oUcdICv93ItWiPQVLivk7Jjlu1O"; //exec ID
const int httpsPort = 443;
const char* fingerprint = "76 63 20 C3 BC 52 21 DD D9 DB EF 33 21 81 F1 6E
7A 69 E2 92"; //mio
String url = String("/macros/s/") + GScriptId +
"/exec?value=Conected_ESP8266";
String url3 = String("/macros/s/") + GScriptId + "/exec?read";
long T0 = 0 ; // Variable global para tiempo
void setup() {
Serial.begin(115200);
pinMode(PIN_RAIN_GAUGE, INPUT);
attachInterrupt(0, countRainGauge, FALLING); //pin 2 del esp8266 pluviometro
Serial.print("Conectando a WIFI: ");
Serial.println(ssid);
Serial.flush();
WIFI.begin(ssid, password);
while (WIFI.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WIFI conectado");
Serial.println("Dirección IP: ");
Serial.println(WIFI.localIP());
HTTPSRedirect client(httpsPort);
Serial.print("Connecting to ");
Serial.println(host);
bool flag = false;
for (int i=0; i<5; i++){
int retval = client.connect(host, httpsPort);
if (retval == 1) {
flag = true;
```

```

break;
}
else
Serial.println("Connección fallida. Reintentando...");
}
Serial.flush();
if (!flag){
Serial.print("No se puede conectar al servidor: ");
Serial.println(host);
Serial.println("Saliendo...");
return;
}
Serial.flush();
if (client.verify(fingerprint, host)) {
Serial.println("Certificado emparejado.");
} else {
Serial.println("Certificado no emparejado");
}
if (!client.connected())
client.connect(host, httpsPort);
url = String("/macros/s/") + GScriptId + "/exec?value="+x;
client.printRedir(url, host, googleRedirHost);
Serial.println("=====
");
numClicksRainGauge=0;//resetear el contador de lluvia
}
}
void countRainGauge() {
raintime = millis(); // grab current time
raininterval = raintime - rainlast; // calculate interval between this and last event
if (raininterval > 1000) // ignore switch-bounce glitches less than 10mS after
initial edge
{
numClicksRainGauge++;
rainlast = raintime; // set up for next event
}
}
}

```

## CÓDIGO DE PROGRAMACIÓN MÓDULO ESP8266 ACTUADOR 1

```
#include <ESP8266WIFI.h>
#include "HTTPSRedirect.h"
const char* ssid = "retlawy2k";
const char* password = "t3$i$un@ch";
const char* host = "script.google.com";
const char* googleRedirHost = "script.googleusercontent.com";
const char *GScriptId = "AKfycbxhwYWTBOJCd-
qpTQKTqWBeXygYI85DjiSq8OVXXpoGpUxmnO8";
const int httpsPort = 443;
const char* fingerprint = "76 63 20 C3 BC 52 21 DD D9 DB EF 33 21 81 F1 6E
7A 69 E2 92";
String url = String("/macros/s/") + GScriptId +
"/exec?value=Conected_ESP8266";
String url3 = String("/macros/s/") + GScriptId + "/exec?read";
int estado;
void setup() {
pinMode(2, OUTPUT);
pinMode(0, OUTPUT);
Serial.begin(115200);
Serial.println();
Serial.print("Connecting to WIFI: ");
Serial.println(ssid);
Serial.flush();
WIFI.begin(ssid, password);
while (WIFI.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WIFI connected");
Serial.println("IP address: ");
Serial.println(WIFI.localIP());
HTTPSRedirect client(httpsPort);
Serial.print("Connecting to ");
Serial.println(host);
bool flag = false;
for (int i=0; i<5; i++){
int retval = client.connect(host, httpsPort);
if (retval == 1) {
flag = true;
break;
}
else
Serial.println("Connection failed. Retrying...");
}
Serial.flush();
```

```

if (client.verify(fingerprint, host)) {
Serial.println("Certificate match.");
} else {
Serial.println("Certificate mis-match");
}
client.printRedir(url, host, googleRedirHost);
Serial.println("=====
");
Serial.println("=====
");
}
void loop() {
estado = digitalRead(2);
Serial.println(estado);
if (estado == 1){
url = String("/macros/s/") + GScriptId + "/exec?value=Led Encendido";
client.printRedir(url, host, googleRedirHost);
Serial.println("Led Encendido");
}
else{
url = String("/macros/s/") + GScriptId + "/exec?value=Led Apagado";
client.printRedir(url, host, googleRedirHost);
Serial.println("Led Apagado");
}
}
}

```

## CÓDIGO DE PROGRAMACIÓN MÓDULO ESP8266 ACTUADOR 2

```
#include <ESP8266WIFI.h>
#include "HTTPSRedirect.h"
const char* ssid = "retlawy2k";
const char* password = "t3$i$un@ch";
const char* host = "script.google.com";
const char* googleRedirHost = "script.googleusercontent.com";
const          char          *GScriptId          =
"AKfycbz29tD4l16Sf7hjEXJpll47AxalWn7uldKywD96YuuVwIlu0Ue1";
const int httpsPort = 443;
const char* fingerprint = "76 63 20 C3 BC 52 21 DD D9 DB EF 33 21 81 F1 6E
7A 69 E2 92";
String url = String("/macros/s/") + GScriptId +
"/exec?value=Conected_ESP8266";
String url3 = String("/macros/s/") + GScriptId + "/exec?read";
int estado;
void setup() {
pinMode(2, OUTPUT);
pinMode(0, OUTPUT);
Serial.begin(115200);
Serial.println();
Serial.print("Connecting to WIFI: ");
Serial.println(ssid);
Serial.flush();
WIFI.begin(ssid, password);
while (WIFI.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WIFI connected");
Serial.println("IP address: ");
Serial.println(WIFI.localIP());
HTTPSRedirect client(httpsPort);
Serial.print("Connecting to ");
Serial.println(host);
bool flag = false;
for (int i=0; i<5; i++){
int retval = client.connect(host, httpsPort);
if (retval == 1) {
flag = true;
break;
}
else
Serial.println("Connection failed. Retrying...");
}
Serial.flush();
```

```

if (client.verify(fingerprint, host)) {
Serial.println("Certificate match.");
} else {
Serial.println("Certificate mis-match");
}
client.printRedir(url, host, googleRedirHost);
Serial.println("=====
");
Serial.println("=====
");
}
void loop() {
estado = digitalRead(2);
Serial.println(estado);
if (estado == 1){
url = String("/macros/s/") + GScriptId + "/exec?value=Led Encendido";
client.printRedir(url, host, googleRedirHost);
Serial.println("Led Encendido");
}
else{
url = String("/macros/s/") + GScriptId + "/exec?value=Led Apagado";
client.printRedir(url, host, googleRedirHost);
Serial.println("Led Apagado");
}
}
}

```

### CÓDIGO DE PROGRAMACIÓN MÓDULO ESP8266 ACTUADOR 3

```
#include <ESP8266WIFI.h>
#include "HTTPSRedirect.h"
const char* ssid = "retlawy2k";
const char* password = "t3$i$un@ch";
const char* host = "script.google.com";
const char* googleRedirHost = "script.googleusercontent.com";
const          char          *GScriptId          =
"AKfycbxCLoStOjC_ZRPVz1VE4IQKdlPMR0Pp83T81NpFyEeAFP1R-JzL";
const int httpsPort = 443;
const char* fingerprint = "76 63 20 C3 BC 52 21 DD D9 DB EF 33 21 81 F1 6E
7A 69 E2 92";
String url = String("/macros/s/") + GScriptId +
"/exec?value=Conected_ESP8266";
String url3 = String("/macros/s/") + GScriptId + "/exec?read";
int estado;
void setup() {
pinMode(2, OUTPUT);
pinMode(0, OUTPUT);
Serial.begin(115200);
Serial.println();
Serial.print("Connecting to WIFI: ");
Serial.println(ssid);
Serial.flush();
WIFI.begin(ssid, password);
while (WIFI.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WIFI connected");
Serial.println("IP address: ");
Serial.println(WIFI.localIP());
HTTPSRedirect client(httpsPort);
Serial.print("Connecting to ");
Serial.println(host);
bool flag = false;
for (int i=0; i<5; i++){
int retval = client.connect(host, httpsPort);
if (retval == 1) {
flag = true;
break;
}
else
Serial.println("Connection failed. Retrying...");
}
Serial.flush();
```



```

if (client.verify(fingerprint, host)) {
Serial.println("Certificate match.");
} else {
Serial.println("Certificate mis-match");
}
client.printRedir(url, host, googleRedirHost);
Serial.println("=====");
Serial.println("=====");
");
}
void loop() {
estado = digitalRead(2);
Serial.println(estado);
if (estado == 1){
url = String("/macros/s/") + GScriptId + "/exec?value=Led Encendido";
client.printRedir(url, host, googleRedirHost);
Serial.println("Led Encendido");
}
else{
url = String("/macros/s/") + GScriptId + "/exec?value=Led Apagado";
client.printRedir(url, host, googleRedirHost);
Serial.println("Led Apagado");
}
}
}

```

## CÓDIGO DE PROGRAMACIÓN PÁGINA WEB

### Código Gs.

```
var SPID="1_CS34qSzcJJWFDeymmnVtunDXVfHgVN8m2ezHsGdP_U";
function doGet() {
return HtmlService.createTemplateFromFile('web')
.evaluate();
}
function getMyData(){
var dsheet = SpreadsheetApp.openById(SPID).getSheetByName("Control");
return dsheet.getRange("A2:I2").getValues();
}
function getDataTemperatura() {
var dsheet = SpreadsheetApp.openById(SPID).getSheetByName("Control");
var hval=dsheet.getRange("A2").getValue();
return hval;
}
function getDataHumedad() {
var dsheet = SpreadsheetApp.openById(SPID).getSheetByName("Control");
var hval=dsheet.getRange("D2").getValue();
return hval;
}
function getDataPrecipitaciones() {
var dsheet = SpreadsheetApp.openById(SPID).getSheetByName("Control");
var hval=dsheet.getRange("G2").getValue();
return hval;
}
function getDataCalefactor() {
var dsheet = SpreadsheetApp.openById(SPID).getSheetByName("Control");
var hval=dsheet.getRange("H2").getValue();
return hval;
}
function getDataVentilador() {
var dsheet = SpreadsheetApp.openById(SPID).getSheetByName("Control");
var hval=dsheet.getRange("I2").getValue();
return hval;
}
function getDataBomba() {
var dsheet = SpreadsheetApp.openById(SPID).getSheetByName("Control");
var hval=dsheet.getRange("J2").getValue();
return hval;
}
function getDataControl() {
var dsheet = SpreadsheetApp.openById(SPID).getSheetByName("Control");
var hval=dsheet.getRange("K2").getValue();
return hval;
}
```

```

function toggleHeater(){
var dsheet = SpreadsheetApp.openById(SPID).getSheetByName("Control");
var hval=dsheet.getRange("H2").getValue();
if(hval=='ON')
dsheet.getRange("H2").setValue("OFF");
else
dsheet.getRange("H2").setValue("ON");
return hval;
}
function togglePump(){
var dsheet = SpreadsheetApp.openById(SPID).getSheetByName("Control");
var cval=dsheet.getRange("J2").getValue();
if(cval=='ON')
dsheet.getRange("J2").setValue("OFF");
else
dsheet.getRange("J2").setValue("ON");
return cval;
}
function toggleControl(){
var dsheet = SpreadsheetApp.openById(SPID).getSheetByName("Control");
var cval=dsheet.getRange("K2").getValue();
if(cval=='1' )
dsheet.getRange("K2").setValue("0");
else
dsheet.getRange("K2").setValue("1");
return cval;
}

```

## CÓDIGO HTML

```
<!-- HTML Starts here -->
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<center>
<h1>Invernadero IoT</h1>
Link para administración :<a
href="https://docs.google.com/spreadsheets/d/1_CS34qSzcJJWFDeymmnVtunD
XVfHgVN8m2ezHsGdP_U/edit#gid=108308472">
Hoja de cálculo: Administración, Monitoreo y Control del Invernadero
</a>
<br><br>
<div id="IotContent">
<table border=1 width="600">
<td align='center' valign='middle'> Temperatura </td>
<td align='center' valign='middle'> Humedad </td>
<td align='center' valign='middle'> Precipitaciones </td>
<td align='center' valign='middle'> Switch Calefactor</td>
<td align='center' valign='middle'> Switch Ventilador</td>
<td align='center' valign='middle'> Switch Bomba</td>
<td align='center' valign='middle'> Control</td><tr>
<td align='center' valign='middle' id="temp"><?!= getDataTemperatura();
?></td>
<td align='center' valign='middle' id="hume"><?!= getDataHumedad(); ?></td>
<td align='center' valign='middle' id="dstatus"><?!= getDataPrecipitaciones();
?></td>
<? if (getDataCalefactor()=='ON') { ?>
<td align='center' valign='middle' id="heater" bgcolor='green'>
<? } else { ?>
<td align='center' valign='middle' id="heater" bgcolor='red'>
<? } ?>
<input type="button" value="Heater Switch"
onclick="google.script.run.withSuccessHandler(onHeaterStatus).toggleHeater()"
/>
</td>
<? if (getDataVentilador()=='ON') { ?>
<td align='center' valign='middle' id="cooler" bgcolor='green'>
<? } else { ?>
<td align='center' valign='middle' id="cooler" bgcolor='red'>
<? } ?>
<input type="button" value="Cooler Switch"
onclick="google.script.run.withSuccessHandler(onCoolerStatus).toggleCooler()"
/>
</td>
```

```

<? if (getDataBomba()=='ON') { ?>
<td align='center' valign='middle' id="pump" bgcolor='green'>
<? } else { ?>
<td align='center' valign='middle' id="pump" bgcolor='red'>
<? } ?>
<input type="button" value="Pump Switch"
onclick="google.script.run.withSuccessHandler(onPumpStatus).togglePump()" />
</body>
</html>
<!-- Scripting Starts here -->
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.5.2/jquery.min.js">
</script>
<script>
}
function onControlStatus(data){
celda = document.getElementById("control");
if(data==0)
celda.style.backgroundColor="red";
}
function onSuccess(data) {
if(data == null)
$("#IotContent").html("Oooops!! Something went wrong! Cannot able to connect
to Google Drive!");
else
{
f(data[0][3] == 0)
$("#dstatus").html("<img
src='https://drive.google.com/uc?id=0B6RqYkndeKb_YmN1VXhsbzhNNnc'/>");
else if(data[0][4] == 1)
$("#dstatus").html("<img
src='https://drive.google.com/uc?id=0B6RqYkndeKb_SEwtall2M0pEeDQ'/>");
else
$("#dstatus").html("<img
src='https://drive.google.com/uc?id=0B6RqYkndeKb_eWt2djRFQTA0STQ'/>");
$("#temp").html(data[0][0]);
}
}
function update() {
$.ajaxSetup({ cache: false });
var data=google.script.run.withSuccessHandler(onSuccess).getMyData();
window.setTimeout(update, 10000);
}
update();
</script>

```

## CODIGO DE PROGRAMACIÓN ANDROID

```
package com.g13enterprise.invernadero;
import android.accounts.Account;
import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.StrictMode;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import com.google.android.gms.auth.GoogleAuthException;
import com.google.android.gms.auth.GoogleAuthUtil;
import com.google.android.gms.auth.UserRecoverableAuthException;
import com.google.gdata.client.spreadsheet.SpreadsheetService;
import com.google.gdata.data.spreadsheet.CellEntry;
import com.google.gdata.data.spreadsheet.CellFeed;
import com.google.gdata.data.spreadsheet.ListFeed;
import com.google.gdata.data.spreadsheet.SpreadsheetEntry;
import com.google.gdata.data.spreadsheet.SpreadsheetFeed;
import com.google.gdata.data.spreadsheet.WorksheetEntry;
import com.google.gdata.util.AuthenticationException;
import com.google.gdata.util.ServiceException;
import java.io.IOException;
import java.net.MalformedURLException;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.List;
public class principal extends AppCompatActivity {
    String mEmail;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_principal);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        Calendar cal = new GregorianCalendar();
```

```

Date dt = cal.getTime();
SimpleDateFormat df = new SimpleDateFormat("dd-MM-yyyy");
String formateDate = df.format(dt);
TextView fechaActual = (TextView) findViewById(R.id.txtFecha);
fechaActual.setText(String.valueOf(formateDate));
new GetDataAsyncTaskC(this).execute();
Bundle extras= getIntent().getExtras();
mEmail= extras.getString("email");
Button boton1 = (Button)findViewById(R.id.btnActualizar);
boton1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
actualizarDatos();
}
});
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
}
public void actualizarDatos(){
new GetDataAsyncTaskC(this).execute();
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.menu_principal, menu);
return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
// Handle action bar item clicks here. The action bar will
// automatically handle clicks on the Home/Up button, so long
// as you specify a parent activity in AndroidManifest.xml.
int id = item.getItemId();
//noinspection SimplifiableIfStatement
if (id == R.id.action_configuracion) {
Intent formConfiguracion = new Intent(this,configuracion.class);
formConfiguracion.putExtra("email", mEmail);
startActivity(formConfiguracion);
return true;
}
if (id == R.id.action_control) {
Intent formControl = new Intent(this, control.class );
formControl.putExtra("email", mEmail);
startActivity(formControl);
return true;
}
if (id == R.id.action_ayuda) {

```

```

Intent formAyuda = new Intent(this, ayudainvernadero.class );
formAyuda.putExtra("email", mEmail);
startActivity(formAyuda);
return true;
}
if (id == R.id.action_acercade) {
Intent formAcercade = new Intent(this, acercade.class );
formAcercade.putExtra("email", mEmail);
startActivity(formAcercade);
return true;
}
return super.onOptionsItemSelected(item);
}
public class GetDataAsyncTaskC extends AsyncTask<Void, Void, Void> {
private float temperaturaActual;
private float humedadActual;
private float precipitacionActual;
private Context context;
static final int REQUEST_CODE_USER_RECOVERABLE_ERROR=3;
String status;
// constructor
GetDataAsyncTaskC(Context context) {
this.context = context;
}
@Override
protected Void doInBackground(Void... params) {
// TODO Auto-generated method stub
try {
getTemperaturaActual1();
getHumedadActual1();
getPrecipitacionActual1();
//setDataHeader();
} catch (AuthenticationException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} catch (MalformedURLException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} catch (ServiceException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} catch (URISyntaxException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
}
}

```



```

return null;
    }
public float getTemperaturaActual1() throws AuthenticationException,
    MalformedURLException, IOException, ServiceException,
    URISyntaxException {
    //String mEmail = "aplicacion.temperatura@gmail.com"; // your google
account name
    String mType = "com.google";
    Account account;
account = new Account(mEmail, mType);
    String scopes = "oauth2:https://spreadsheets.google.com/feeds "
"https://www.googleapis.com/auth/plus.login "
    + "https://www.googleapis.com/auth/drive";
    String token = null;
try {
token = GoogleAuthUtil.getToken(context, account, scopes);
    } catch (UserRecoverableAuthException userRecoverableError){
        Intent intent =
((UserRecoverableAuthException)userRecoverableError).getIntent();
startActivityForResult(intent,
REQUEST_CODE_USER_RECOVERABLE_ERROR);
status = "User Recoverable Error";
    } catch (GoogleAuthException e) {
e.printStackTrace();
    }
    // TODO Auto-generated method stub
    SpreadsheetService service =
new SpreadsheetService("MySpreadsheetIntegration-v1");
//service.setProtocolVersion(SpreadsheetService.Versions.V3);
service.setAuthSubToken(token);
    // TODO: Authorize the service object for a specific user (see other
sections)
    // Define the URL to request. This should never change.
    URL SPREADSHEET_FEED_URL = new URL(
"https://spreadsheets.google.com/feeds/spreadsheets/private/full");
try {
int SDK_INT = android.os.Build.VERSION.SDK_INT;
    StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder()
        .permitAll().build();
StrictMode.setThreadPolicy(policy);
    // Make a request to the API and get all spreadsheets.
    SpreadsheetFeed feed =
service.getFeed(SPREADSHEET_FEED_URL, SpreadsheetFeed.class);
    List<SpreadsheetEntry> spreadsheets = feed.getEntries();

for (SpreadsheetEntry spreadsheet : spreadsheets) { // outer for loop

```

```

if (spreadsheet.getTitle().getPlainText().equalsIgnoreCase("invernadero(final)"))
{ // your spreadsheet name
    List<WorksheetEntry> worksheets = spreadsheet.getWorksheets();

for (WorksheetEntry worksheet : worksheets) { // inner for loop
    String title = worksheet.getTitle().getPlainText();
if (title.equalsIgnoreCase("Control")) { // your worksheet name
    URL listFeedUrl = worksheet.getListFeedUrl();
    ListFeed listFeed = service.getFeed(listFeedUrl,
ListFeed.class);
    // scan through each row in worksheet

    URL cellFeedUrl = new
URI(worksheet.getCellFeedUrl().toString()
    + "?min-row=2&min-col=1&max-col=1").toURL();
    CellFeed cellFeed = service.getFeed(cellFeedUrl,
CellFeed.class);
    // Iterate through each cell, printing its value.
for (CellEntry cell : cellFeed.getEntries()) {
    // Print the cell's address in A1 notation
System.out.print(cell.getTitle().getPlainText() + "\t");
    //Log.d(TAG, cell.getTitle().getPlainText());
if (cell.getTitle().getPlainText().equals("A2")) {
temperaturaActual = Float.valueOf(cell.getCell().getValue());
    }
    }
    }
    }
    }
    } catch (Exception e) {
e.printStackTrace();
    }
return temperaturaActual;
}

public float getHumedadActual1() throws AuthenticationException,
    MalformedURLException, IOException, ServiceException,
    URISyntaxException {
    //String mEmail = "aplicacion.temperatura@gmail.com"; // your google
account name
    String mType = "com.google";
    Account account;
account = new Account(mEmail, mType);
    String scopes = "oauth2:https://spreadsheets.google.com/feeds "
    + "https://www.googleapis.com/auth/plus.login "
    + "https://www.googleapis.com/auth/drive";

```

```

String token = null;

try {
token = GoogleAuthUtil.getToken(context, account, scopes);
    } catch (UserRecoverableAuthException userRecoverableError) {
        Intent intent =
((UserRecoverableAuthException)userRecoverableError).getIntent();
startActivityForResult(intent,
REQUEST_CODE_USER_RECOVERABLE_ERROR);
status = "User Recoverable Error";
    } catch (GoogleAuthException e) {
e.printStackTrace();
    }
// TODO Auto-generated method stub
SpreadsheetService service =
//service.setProtocolVersion(SpreadsheetService.Versions.V3);
service.setAuthSubToken(token);
// TODO: Authorize the service object for a specific user (see other
sections)
// Define the URL to request. This should never change.
URL SPREADSHEET_FEED_URL = new URL(
"https://spreadsheets.google.com/feeds/spreadsheets/private/full");
try {
int SDK_INT = android.os.Build.VERSION.SDK_INT;
        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder()
            .permitAll().build();
StrictMode.setThreadPolicy(policy);
// Make a request to the API and get all spreadsheets.
SpreadsheetFeed feed =
service.getFeed(SPREADSHEET_FEED_URL, SpreadsheetFeed.class);
        List<SpreadsheetEntry> spreadsheets = feed.getEntries();

for (SpreadsheetEntry spreadsheet : spreadsheets) { // outer for loop
if (spreadsheet.getTitle().getPlainText().equalsIgnoreCase("invernadero(final)"))
{ // your spreadsheet name
            List<WorksheetEntry> worksheets = spreadsheet.getWorksheets();

for (WorksheetEntry worksheet : worksheets) { // inner for loop
                String title = worksheet.getTitle().getPlainText();
if (title.equalsIgnoreCase("Control")) { // your worksheet name
                    URL listFeedUrl = worksheet.getListFeedUrl();
                    ListFeed listFeed = service.getFeed(listFeedUrl,
ListFeed.class);
// scan through each row in worksheet

                    URL cellFeedUrl = new
URI(worksheet.getCellFeedUrl().toString()

```

```

        + "?min-row=2&min-col=4&max-col=4").toURL());
        CellFeed cellFeed = service.getFeed(cellFeedUrl,
CellFeed.class);

        // Iterate through each cell, printing its value.
        for (CellEntry cell : cellFeed.getEntries()) {
            // Print the cell's address in A1 notation
            System.out.print(cell.getTitle().getPlainText() + "\t");
            //Log.d(TAG, cell.getTitle().getPlainText());
            if (cell.getTitle().getPlainText().equals("D2")) {
                humedadActual = Float.valueOf(cell.getCell().getValue());
            }
        }
    }
}

    } catch (Exception e) {
        e.printStackTrace();
    }
    return humedadActual;
}

public float getPrecipitacionActual1() throws AuthenticationException,
        MalformedURLException, IOException, ServiceException,
        URISyntaxException {
    //String mEmail = "aplicacion.temperatura@gmail.com"; // your google
account name
    String mType = "com.google";
    Account account;
    account = new Account(mEmail, mType);
    "https://www.googleapis.com/auth/plus.login "
        + "https://www.googleapis.com/auth/drive";
    String token = null;
    try {
        token = GoogleAuthUtil.getToken(context, account, scopes);
    } catch (UserRecoverableAuthException userRecoverableError){
        Intent intent =
((UserRecoverableAuthException)userRecoverableError).getIntent();
REQUEST_CODE_USER_RECOVERABLE_ERROR);
        status = "User Recoverable Error";
    } catch (GoogleAuthException e) {
        e.printStackTrace();
    }
    // TODO Auto-generated method stub
    SpreadsheetService service =

```

```

//service.setProtocolVersion(SpreadsheetService.Versions.V3);
service.setAuthSubToken(token);
    // TODO: Authorize the service object for a specific user (see other
sections)
    // Define the URL to request. This should never change.
    URL SPREADSHEET_FEED_URL = new URL(
"https://spreadsheets.google.com/feeds/spreadsheets/private/full");
try {
int SDK_INT = android.os.Build.VERSION.SDK_INT;
    StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder()
        .permitAll().build();
StrictMode.setThreadPolicy(policy);
    // Make a request to the API and get all spreadsheets.
    SpreadsheetFeed feed =
service.getFeed(SPREADSHEET_FEED_URL, SpreadsheetFeed.class);
    List<SpreadsheetEntry> spreadsheets = feed.getEntries();

for (SpreadsheetEntry spreadsheet : spreadsheets) { // outer for loop
if (spreadsheet.getTitle().getPlainText().equalsIgnoreCase("invernadero(final)"))
    { // your spreadsheet name
        List<WorksheetEntry> worksheets = spreadsheet.getWorksheets();

for (WorksheetEntry worksheet : worksheets) { // inner for loop
            String title = worksheet.getTitle().getPlainText();
if (title.equalsIgnoreCase("Control")) { // your worksheet name
                URL listFeedUrl = worksheet.getListFeedUrl();
                // scan through each row in worksheet

                //URL cellFeedUrl = worksheet.getCellFeedUrl(); ---Normal
                URL cellFeedUrl = new
URI(worksheet.getCellFeedUrl().toString()
                    + "?min-row=2&min-col=7&max-col=7").toURL();
                CellFeed cellFeed = service.getFeed(cellFeedUrl,
CellFeed.class);
                // Iterate through each cell, printing its value.
for (CellEntry cell : cellFeed.getEntries()) {
                    // Print the cell's address in A1 notation
System.out.print(cell.getTitle().getPlainText() + "\t");
                    //Log.d(TAG, cell.getTitle().getPlainText());
if (cell.getTitle().getPlainText().equals("G2")) {
                        precipitationActual = Float.valueOf(cell.getCell().getValue());
                    }
                }
            }
        }
    }
}

```

```

        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return precipitacionActual;
}
protected void onPostExecute(Void result) {
    TextView temperaturaActualTexto = (TextView)
    findViewById(R.id.txtTemperatura);
    temperaturaActualTexto.setText(String.valueOf(temperaturaActual) + "\u2070" +
    "C");
    TextView humedadActualTexto = (TextView)
    findViewById(R.id.txtHumedad);
    TextView precipitacionActualTexto = (TextView)
    findViewById(R.id.txtPrecipitaciones);
    precipitacionActualTexto.setText(String.valueOf(precipitacionActual));
}
}
}

```