



Monitoreo web y análisis de datos ambientales de un cultivo invernadero para control de parámetros de riego mediante una red de sensores inalámbricos.

Proyecto de investigación

Luis Miguel Avendaño Lozano
CC 1091679365
Jhon Fredy Villamizar Talero
CC 1098800707

UNIDADES TECNOLÓGICAS DE SANTANDER
Facultad ciencias naturales e ingenierías
Ingeniería en telecomunicaciones
Bucaramanga



Monitoreo web y análisis de datos ambientales de un cultivo invernadero para control de parámetros de riego mediante una red de sensores inalámbricos.

Proyecto de investigación

Luis Miguel Avendaño Lozano
CC 1091679365
Jhon Fredy Villamizar Talero
CC 1098800707

**Trabajo de Grado para optar al título de
Ingeniero de telecomunicaciones**

DIRECTOR
Johan Leandro Téllez Garzón

GNET

UNIDADES TECNOLÓGICAS DE SANTANDER
Facultad ciencias naturales e ingenierías
Ingeniería en telecomunicaciones
Bucaramanga

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 1.0

Nota de Aceptación

Firma del Evaluador

Firma del Director

DEDICATORIA

Dedicamos este trabajo a nuestros padres por habernos brindado el apoyo necesario y el valor correspondiente para poder concluir con satisfacción este presente trabajo, y de esta manera poder formarnos como unos futuros profesionales capacitados que ayuden a contribuir en el desarrollo del futuro tecnológico de nuestro país.

AGRADECIMIENTOS

Agradecemos a Dios por habernos permitido vivir todos los logros conseguidos hasta hoy, por habernos dado la fortaleza y la capacidad necesaria para enfrentar día a día las dificultades. También le doy gracias a él por habernos dado una familia maravillosa a cada uno, quienes han creído en nosotros siempre y nos han dado ejemplo de superación, humildad y sacrificio, esperamos contar con su valioso apoyo siempre.

Además, agradecemos a la Unidades tecnológicas de Santander por habernos permitido formar parte de su seno académico, permitiéndonos estudiar una carrera que nos gusta, y que nos ha aportado un acompañamiento con muchos docentes, quienes pusieron sus conocimiento y dedicación en enseñarnos.

De esta forma especial, agradecemos también a nuestro tutor de Tesis. Johan Leandro Téllez Garzón, por habernos acompañado en todo el proceso aportándonos de su conocimiento y consejos para la culminación de nuestro proyecto de grado.

Y para finalizar, a nuestros padres a quienes les debemos el alcance no solo de este logro si no de muchos que han marcado nuestra vida forjándonos en las personas que somos ahora y dedicándonos todo el amor y cariño en cada momento.

TABLA DE CONTENIDO

<u>RESUMEN EJECUTIVO</u>	12
<u>INTRODUCCIÓN</u>	14
<u>1. DESCRIPCIÓN DEL TRABAJO DE INVESTIGACIÓN.....</u>	15
1.1. PLANTEAMIENTO DEL PROBLEMA	15
1.2. JUSTIFICACIÓN	17
1.3. OBJETIVOS.....	17
1.3.1. OBJETIVO GENERAL	17
1.3.2. OBJETIVOS ESPECÍFICOS.....	18
1.4. ESTADO DEL ARTE	18
<u>2. MARCO REFERENCIAL</u>	21
2.1. REDES DE SENsoRES INALÁMBRICOS	21
2.2. INTERNET DE LAS COSAS	22
2.3. SENsoRES	24
2.3.1. SENSOR HUMEDAD Y TEMPERATURA	24
2.3.2. SENSOR DE HUMEDAD DEL SUELO	24
2.4. ACTUADORES	25
2.5. SERVIDOR	25
2.5.1. AMAZON EC2.....	25
2.5.2. AMAZON AWS	26
2.6. BRÓKER	26
2.6.1. MQTT	26
2.6.2. EMQX	27

2.7. HOSTING	28
2.8. BASE DE DATOS	28
2.9. APLICACIONES	29
2.10. NODE JS	29
2.11. SOFTWARE Y APLICACIONES	29
2.11.1. IDE ARDUINO	29
2.11.2. ATOM	29
2.11.3. VESTA.....	29
2.11.4. PHPMYADMIN	30
2.11.5. DASHBOARD.....	30
2.12. CULTIVO INVERNADERO	30
2.12.1. CULTIVO DE TOMATE EN INVERNADERO.....	31
3. DISEÑO DE LA INVESTIGACION	33
4. DESARROLLO DEL TRABAJO DE GRADO.....	34
4.1. DIAGRAMA DE FUNCIONAMIENTO DEL PROYECTO.....	34
4.2. PARTES DEL SISTEMA IoT DE LA RED SENSORIA INALÁMBRICA.....	35
4.2.1. IMPLEMENTACIÓN DEL SERVIDOR PRIVADO VIRTUAL.....	36
4.2.2. HOSTING.....	37
4.2.3. BASES DE DATOS	39
4.2.4. PANEL DE CONTROL EN PHP.....	41
4.2.5. MQTT	44
4.2.6. DISPOSITIVO IOT	47
4.2.7. NODE Js	50
4.3. MAQUETA DEL CULTIVO INVERNADERO	52
5. RESULTADOS	54
5.1. PARÁMETROS PRESENTES EN EL MONITOREO.....	55
5.1.1. PARÁMETRO DE TEMPERATURA AMBIENTE	55

5.1.2. PARÁMETRO DE HUMEDAD AMBIENTE	57
5.1.3. PARÁMETRO DE HUMEDAD DEL SUELO.....	58
5.2. ALERTAS POR E-MAIL Y FUNCIONAMIENTO DEL SISTEMA DE RIEGO.	59
6. CONCLUSIONES.....	61
7. RECOMENDACIONES	63
8. REFERENCIAS BIBLIOGRÁFICA.....	64
9. APÉNDICES.....	67
10. ANEXOS.....	70

LISTA DE FIGURAS

Figura 2.1 Topologías de redes de sensores inalámbricos	22
Figura 2.2 Diagrama sistema IoT.	23
Figura 2.3 Esquema Bróker MQTT	27
Figura 2.4 Zona de cultivos invernaderos	28
Figura 2.5 Temperaturas para el desarrollo de la planta de tomate.....	32
Figura 4.1 Diagrama completo del sistema IoT de la red sensora inalámbrica.	35
Figura 4.2 Recursos utilizados en el servidor privado virtual	36
Figura 4.3 Conexión a ssh al servidor de Ubuntu.....	37
Figura 4.4 Panel de control en Vesta	38
Figura 4.5 Configuración servicio ftp	39
Figura 4.6 Base de datos implementada.....	40
Figura 4.7 Tabla de la unidad sensora número 1	41
Figura 4.8 Dashboard sistema IoT	42
Figura 4.9 Gráfico de datos	42
Figura 4.10 Login de usuarios.	43
Figura 4.11 Imagen de los históricos de datos	43
Figura 4.12 Servicio de EMQX dentro del VPS	44
Figura 4.13 Dasboard de EMQ.....	46
Figura 4.14 Envío de mensajes desde php a MQTT	46
Figura 4.15 Recepción de mensajes con Node JS.....	47
Figura 4.16 Esquema del circuito de la red sensora inalámbrica	48
Figura 4.17 Conexiones de la Esp8266	49
Figura 4.18 Gestor de procesos pm2.....	51
Figura 4.19 Herramienta de monitoreo pm2 monit	51
Figura 4.20 Maqueta del cultivo invernadero	52
Figura 4.21 Interior de la maqueta del cultivo invernadero.....	53

Figura 5.1 Esquema interior del cultivo invernadero.....	54
Figura 5.2 Grafica de valores extraídos de la base de datos	56
Figura 5.3 Grafica de valores de humedad del ambiente extraídos de la base de datos	
.....	57
Figura 5.4 valores de humedad del suelo extraído de la base de datos.....	58
Figura 5.5 Grafica de valores de humedad del suelo extraídos de la base de datos ...	58
Figura 5.6 Envío de alertas vía correo electrónico.....	59
Figura 5.7 Activación del sistema de riego desde el panel de control en PHP	60
Figura 5.8 Sistema de riego en funcionamiento.....	60

LISTA DE TABLAS

Tabla 1 valores de temperatura extraída de la base de datos	56
Tabla 2 Registro de sistemas de alerta.	60

RESUMEN EJECUTIVO

La tecnología es un recurso que va avanzando a grandes escalas y se debe dar uso efectivo para mejorar la calidad de vida de todos los seres humanos en todos los aspectos. El desarrollo de la humanidad a través de los tiempos se ha visto marcado especialmente por este avance tecnológico, por la contaminación y deterioro ambiental que se ha venido causando a lo largo de la globalización y que ha afectado miles de cultivos perdidos y tierras infértilles. ADICIONALMENTE LA DEMANDA MUNDIAL POR ALIMENTOS SE INCREMENTA AÑO A AÑO SEGÚN DATOS DE LA FAO. Por ende, se buscó la implementación de nuevos métodos que han permitido solucionar esta problemática como son los sistemas de cultivo invernadero que han mejorado la productividad agrícola, el cuidado del medio ambiente y también la calidad de vida humana.

En Colombia se han realizado diferentes estudios con los cultivos invernaderos, y la importancia de tener un control de este cultivo conlleva varios beneficios.

Por tal motivo la finalidad de este trabajo es proponer un sistema de Monitoreo, control de riego y Registro de variables como, temperatura, humedad relativa y humedad del suelo en una maqueta de cultivo invernadero mediante un sistema IoT de redes de sensores inalámbricos. Los datos recolectados por los sensores serán enviadas a la ESP8266, que se comunicara con nuestro bróker MQTT y a su vez el bróker con los demás clientes, como Node JS y la aplicación web; todo esto estará en el servidor, el cual tendrá una base de datos para almacenamiento y consulta del histórico de cada sensor inalámbrico, y de acuerdo al estado actual de la humedad del suelo se avisara al gestor del cultivo para activar el sistema de riego. Como herramienta de desarrollo de la base de datos se usará MYSQL.

De esta manera el gestor del cultivo invernadero podrá realizar consultas del histórico de las variables de desarrollo de las plantas para hacer correcciones con

la finalidad de mejorar la productividad. Los resultados de este trabajo serán una metodología para implementar una red de sensores inalámbricos, actuador del sistema de riego, las configuraciones de la base de datos y la comunicación de la placa ESP8266 con los demás clientes. Además, se realizará un teste experimental en una maqueta de cultivo invernadero donde se presentarán las evidencias de la implementación y las variables obtenidas en función del tiempo.

PALABRAS CLAVE. MQTT, invernaderos, redes de sensores inalámbricos, ESP8266, node js, base de datos, IoT.

INTRODUCCIÓN

En la actualidad, se viene implementando una tecnología de gran impacto para la digitalización de los diferentes procesos en cualquier campo que se requiera, denominada como IoT (Internet de las Cosas). Según (Leukert, 2021), Internet de las cosas se define como aquel que forma una red de objetos físicos, de la cual se desprenden vehículos, máquinas, todo tipo de electrodomésticos que de una u otra forma utilizan sensores y/o APIs para poder conectarse y enviar datos a través de Internet. En el ámbito empresarial, las soluciones brindadas por IoT permite que las empresas puedan mejorar sus actuales sistemas, para diseñar ciertos puntos de conexión entre clientes y partners (Red Hat, 2021).

Para (Gómez, 2020), IoT es aquella tecnología que proporciona todas las herramientas necesarias para la toma de decisiones en el desarrollo y la producción de cultivos mediante un acceso a Internet y/o aplicativos móviles que desarrollen una diversidad de funciones, ligadas de la problemática de mayor auge presentada.

Este proyecto se centra directamente en los cultivos de tomate que, mediante una diversidad de sensores, tales como de humedad del suelo, temperatura ambiente y humedad ambiente; se pueda ofrecer control de dicho cultivo a través de una red sensora inalámbrica que se logra con la ayuda del IoT, los sensores capturan los datos para posteriormente ser enviados por la ESP8266 a un bróker MQTT interconectando los demás clientes, como por ejemplo la aplicación web, quien permite realizar una acción específica en respuesta a dicha situación presentada.

1. DESCRIPCIÓN DEL TRABAJO DE INVESTIGACIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

En los últimos años se han venido implementando en Colombia cultivos invernaderos con el fin de mejorar la productividad de los cultivos; puesto que este sistema les protege de plagas y daños ambientales (Perilla, Rodríguez, & Bermúdez, 2011).

Para poder mantener las plantas en condiciones es necesario conocer las variables que indiquen en qué condiciones se encuentra el cultivo. Actualmente los gestores de cultivo realizan las correcciones a partir de mediciones parciales, o muchas veces, las realizan sin saber que necesita la planta, sin saber los niveles históricos de temperatura, humedad relativa, humedad del suelo que presentan a lo largo de los días. Por otro lado, cada vez que avanza el tiempo la mano de obra en el campo es más escasa debido a las pocas garantías que ofrece el agro, causando que no se pueda, por ejemplo, regar el cultivo el día correcto por realizar otras tareas.

Existe una demanda mayor de alimentos generada por el crecimiento demográfico acelerado (FAO, 2000), pero los sistemas de producción de alimentos resultan afectados por cuestiones climáticas y sociales que reducen de forma significativa su eficiencia, además, el uso de tecnologías que permitan manejar ampliamente la producción no está masificado en el sector agrícola colombiano.

Las TIC brindan a la agricultura mayor competitividad mejorando la producción y muchos otros ámbitos, por ejemplo, aumenta la eficiencia y reduce los costos debido

que al estar el cultivo en control reduce los daños causados por factores ambientales (Negal, 2012).

En este contexto, el trabajo propone una alternativa tecnológica mediante el uso de una red de sensores inalámbricos, y una base de datos, que brindaran un histórico de las variables (temperatura, humedad relativa, humedad del suelo, luminosidad) de un cultivo invernadero, además, dependiendo del parámetro de la humedad del suelo poder brindarle la opción al administrador del cultivo si desea poner en función su sistema de riego. Así, de esta manera se podrán realizar correcciones más acertadas al considerar esta información. La problemática plantea ¿Cómo implementar un sistema de red de sensores inalámbricos que permita adquirir parámetros ambientales de un cultivo invernadero para realizar análisis de datos en tiempo real y control de los parámetros de riego para optimizar el crecimiento de las plantas? esta red de sensores inalámbricos será implementada en una maqueta de 90 cm de largo, 60 cm de ancho y 50cm de alto.

1.2. JUSTIFICACIÓN

El propósito de este proyecto es brindar una herramienta tecnológica a aquellos agricultores que cuentan con invernadero con el fin de mejorar tecnológicamente los métodos de cultivos con sistemas IoT para incrementar la productividad y de esta manera satisfacer las necesidades alimenticias.

Así se propone mejorar los cultivos invernaderos con un método que permita adquirir mediciones de temperatura, humedad relativa, humedad del suelo, además de automatizar el sistema de riego en el cultivo mediante una red de sensores permitiendo al gestor del cultivo invernadero pueda realizar correcciones para mejorar la productividad. Debido a que la FAO (Organización de las Naciones Unidas para la Alimentación) afirma que la demanda mundial de alimentos es creciente y sus proyecciones afirman que los actuales métodos de agricultura no podrán satisfacer las necesidades alimenticias de la población.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Implementar una red de sensores que permita el envío de los datos ambientales de un cultivo invernadero de tomate a un ente centralizado para ejecución de un análisis de datos con el objetivo de realimentar algunos parámetros de crecimiento de las plantas en tiempo real y mejorar la calidad y competitividad del cultivo.

1.3.2. OBJETIVOS ESPECÍFICOS

Medir variables ambientales de temperatura, humedad relativa, humedad del suelo por medio de una red de sensores inalámbricos con el fin del registro de datos.

Encontrar alternativas óptimas relación calidad precio de sensores buscando con distintos proveedores ofreciendo una solución económica al campesino.

Implementar el VPS que contendrá la base de datos y el hosting, para montar una aplicación web visualizando la información recolectada de la red de sensores.

Realizar un análisis de datos de la información adquirida en el VPS para ejecutar órdenes de control sobre variables del riego del cultivo invernadero de tomate.

Poner en funcionamiento dentro del VPS el bróker EMQX para poder manejar conexiones simultáneas.

1.4. ESTADO DEL ARTE

(En la exploración del tema de interés, buscando entre artículos y proyectos relacionados, debido a que es de vital importancia conocer que se tiene en materia y de esta manera tener una base al desarrollar el proyecto de grado.

Un artículo en la India titulado *Aplicación de redes de sensores inalámbricos para el control de parámetros de invernadero en la agricultura de precisión* (Chaudhary, Nayse, & Waghmare, 2011), proponen medir ciertos parámetros del ambiente para poder monitorearlos y controlarlos, utilizan tecnología Zigbee para medir humedad (aire y suelo), temperatura, CO₂, pH, luminosidad. Donde se concluye acerca de la

importancia que influye una red de sensores inalámbricos en el cultivo invernadero, puesto que se puede tener mejor control del cultivo.

De otro lado, en centro américa, México, María Flores Medina, Francisco Flores García, Víctor Velasco Martínez, en su artículo *Monitoreo de humedad en el suelo a través de red inalámbrica de sensores* (Flores Medina, Flores Garcia, & Velasco Martines, 2015); exponen una tecnología de sensores inalámbricos emergentes llamada RIS, enfocada al ámbito de la agricultura para medir la humedad del suelo, se implementa un servidor y se crea una aplicación para observar los resultados arrojados por los sensores. En la elaboración de esta red se obtuvieron los datos en tiempo real lo que permite hacer correcciones y hacer manejo del riego a tiempo. Puesto que en la red no cuentan con actuadores para activar el riego esta parte la tendrían que hacer manual.

En un ámbito nacional, tenemos un artículo publicado por Fernando Aparicio Urbano-Molano, titulado *Redes de sensores inalámbricos aplicadas a optimización en agricultura de precisión para cultivos de café en Colombia* (Urbano-Molano & Aparicio, 2013); en donde se implementan distintos sensores para saber con mejor detalle el estado de la planta, midiendo temperatura y humedad del suelo, temperatura de las hojas, radiación solar, flujo de fotosíntesis. En el artículo se empieza abordando los conceptos básicos para entender que es y cómo funciona una red de sensores inalámbricos, para posteriormente exponer el sistema que fue empleado. Este sistema proporciona bastante información relevante para conocer el estado de la planta y así poder optar por aplicar correcciones más acertadas, no obstante, se menciona que tuvieron algunas perdidas de datos, por lo que la calidad del servicio QoS no es buena.

Por último, en la universidad nacional de Medellín, Alejandro Cama-Pinto, Francisco Gil-Montoya, Julio Gómez-López, Francisco Manzano-Ajuglar con su artículo *Sistema inalámbrico de monitorización para cultivos en invernadero* (Cama-Pinto, Gil-Montoya, Gómez-Lopez, & Manzano-Ajugliaro, 2014); plantean una red

de Sensores inalámbricos que mide humedad, temperatura, luz y contenido volumétrico de agua en el suelo. Se abordan los temas del montaje y funcionamiento de una red de sensores inalámbricos, implementando además un servidor físico. Se plantea un sistema de medición correcto que proporciona los datos requeridos cumpliendo su objetivo, no obstante, no hay actuadores para tomar medidas correctivas a los eventos presentados, además, al ellos contar con el servidor físico puede presentar más gastos y ser poco escalable.

2. MARCO REFERENCIAL

2.1. Redes de sensores inalámbricos

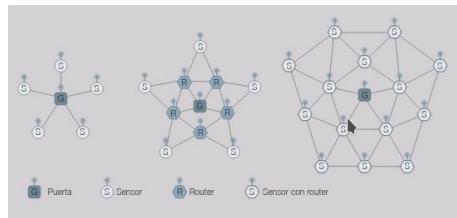
Una red de sensores inalámbricos WSN está conformada por varios dispositivos IoT que se distribuyen según la topología con la que hayan sido instalados, los sensores son usados para controlar distintas condiciones en distintos zonas (Aakvaag & Frey, 2006), como por ejemplo temperatura, humedad, luminosidad, etc.

Además de los sensores, estas redes constan de dispositivos programables, microcontroladores, estos dispositivos trabajan con baja potencia con el fin de ahorrar energía.

Las redes de sensores inalámbricos no solo se encargan de capturar información del medio, sino también de actuar en consecuencia a algún suceso presentado (Gascón, 2010).

La composición de los sensores, con el procesamiento y la comunicación son conocidos como *nodo sensor*, que cumple la etapa de recolectar procesar y enviar los datos (Quiñones-Cuenca, González-Jaramillo, Torre, & Jumbo, 2017).

Las redes de sensores inalámbricos constan de distintas topologías, las cuales cada una tiene sus ventajas y desventajas, en la figura 1.1 podremos observar algunas de ellas.

Figura 2.1 Topologías de redes de sensores inalámbricos

Nota. Existen distintas topologías de redes de sensores tal como la estrella y malla. Tomada de *Topologías de redes de sensores inalámbricos* [Figura] Niels Aakvaag; Jan-Erik Frey, 2006, (Aakvaag & Frey, 2006)

2.2. Internet de las cosas

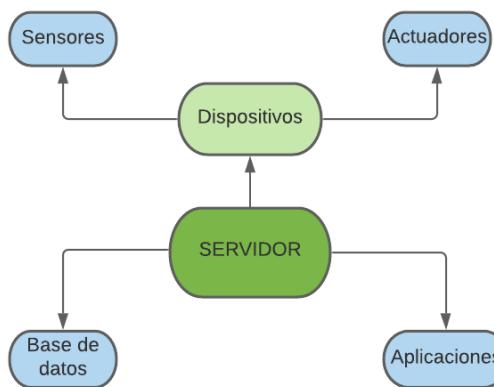
El internet de las cosas (IOT), son objetos cotidianos que por medio de una minicomputadora están conectados a la red. El internet aborda lo referente a las redes de sensores inalámbricos.

Un sistema de internet de las cosas en primer lugar funciona con un servidor, en segundo lugar, se encuentran los dispositivos como una ESP8266 que estarán conectados a internet. Para poder tomar información del entorno, a los dispositivos se les conectarán sensores. Además, un sistema IoT necesita automatizarse es decir que de acuerdo con los valores que tome del entorno se tomen diferentes acciones.

De otro lado, tenemos las aplicaciones que permitirán a los usuarios visualizar los valores recogidos por los sensores, juegan un papel de suma importancia en lo

que llamamos internet de las cosas ya que estos se encargan de la recolección de información de las distintas variables que se necesitan para la correcta optimización de las actividades básicas o funciones que realizan las personas en su vida cotidiana, teniendo en cuenta los distintos estándares y protocolos que logran adaptarse de una manera más óptima a lo que se pretende lograr (Cama-Pinto, De la Hoz, & Cama-Pinto, Las redes de sensores inalámbricos y el Internet de las cosas, 2012). y una base de datos que permitirá ver información del pasado, en la imagen 2.1 se observa un esquema básico del IoT

Figura 2.2 Diagrama sistema IoT.



Nota. Un sistema IoT está conformado por un conjunto de elementos tales como un servidor, dispositivos, sensores, actuadores bases de datos y aplicaciones. Adaptado de *Diagrama IoT [Diagrama]* Pablo Sanz, (Sanz, Esquema IoT)

No obstante, para implementar un sistema IoT en una empresa donde el proceso dependa del funcionamiento de la recolección de datos del sistema, o se tengan dispositivos que estén controlando tareas o procesos es importante implementar un motor de reglas en el servidor, como también las reglas offline que permiten al

sistema seguir recolectando información del medio o realizando alguna acción en la ausencia del internet.

2.3. Sensores

Son los encargados de recoger los datos o información del entorno para después ser mandados al microcontrolador, como por ejemplo la temperatura, humedad o movimiento.

Debemos tener en cuenta las características de los sensores, su capacidad, su calibración, el tipo de material del sensor a usar, las especificaciones técnicas para así poder optimizar el funcionamiento de ellos en la toma de la información correspondiente en el cultivo seleccionando así las mejores decisiones (Kouro, 2001).

2.3.1. Sensor humedad y temperatura

Es el encargado de recopilar la información que se solicita en un ambiente seguro, suele estar conformado por un display iluminado, también es un transmisor de humedad y temperatura que proporciona alta precisión y estabilidad de las mediciones, en amplio rango de operación.

2.3.2. Sensor de humedad del suelo

El sensor de humedad del suelo es una herramienta valiosa debido a su amplia gama de aplicaciones. Este sensor se usa ampliamente en el control del riego en explotaciones agrícolas y de jardinería que permiten recopilar datos de humedad y temperatura del suelo. Estas medidas ayudarán a las personas a cargo a tomar las decisiones adecuadas.

La protección provista por el sensor permite su instalación enterrada en el suelo. El sensor debe instalarse a una profundidad suficiente, de modo que los factores ambientales externos no afecten las mediciones registradas por el sensor. La distancia de enterramiento dependerá del tipo de terreno y las condiciones ambientales.

2.4. Actuadores

Actuadores: Son los encargados de generar una acción sobre un proceso, donde la orden es recibida por el microcontrolador como por ejemplo un relé, o un servomotor

2.5. Servidor

Es un equipo capaz de procesar y entregar datos a un cliente. Un servidor puede ser una computadora u ordenador en un rack de algún proveedor de servicios, o un ordenador en la empresa que esté conectado a internet.

El motor de reglas trabaja en función directa con el servidor, donde este tiene reglas para controlar los valores, y en función de estos actué o no.

Un servidor tiene que realizar diferentes tareas, por eso debe contener un Hosting, un Bróker, una base de datos y un motor de reglas

2.5.1. Amazon EC2

Es un servicio que brinda computación en la nube, en el cual nos evitamos invertir en una computadora física, lo cual trae muchas ventajas, una de ellas es la escalabilidad. (AMAZON, s.f.)

2.5.2. Amazon AWS

Es una plataforma en la nube, así como google cloud, ofrece más de 175 servicios integrales de centros de datos.

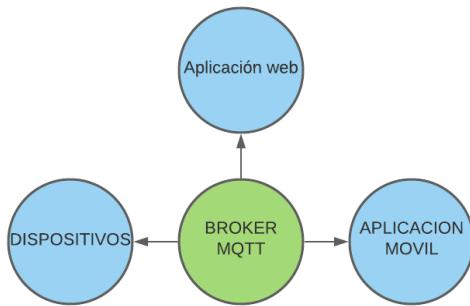
2.6. Bróker

Una de las tareas más relevantes son las del bróker, es un sistema que recibe los datos de los dispositivos y los retransmite a las aplicaciones de los usuarios. Además, envía la información cuando se quiera comunicar desde una aplicación a los dispositivos.

2.6.1. MQTT

Uno de las mayores preocupaciones del llamado Internet de las cosas (IoT) es lo que se conoce como interoperabilidad, el mayor desafío es el poder conectar mecanismos industriales con tecnologías de información y plataformas IoT, allí es donde los protocolos juegan un papel importante, en el caso del presente proyecto el protocolo MQTT que nos facilita este proceso (Semle & Kepware, 2016).

Es un protocolo de red publicación-suscripción (MQTT, s.f.), el cual es muy eficiente por transmitir datos de forma liviana. Su función es comunicar aplicaciones web con aplicaciones móviles y con los dispositivos IoT, como se observa en la figura 2.4.1

Figura 2.3 Esquema Bróker MQTT

Nota. Un bróker MQTT permite comunicar mensajes entre dispositivos, aplicaciones móviles, y aplicaciones web, es decir entre clientes. Adaptado de *Esquema Bróker MQTT [Diagrama]* Pablo Sanz, (Sanz, Introducción a MQTT)

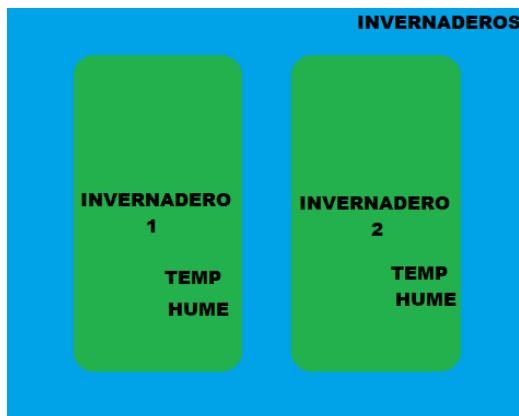
2.6.2. EMQX

Es un agente de mensajes MQTT distribuido en tiempo real y altamente escalable para aplicaciones IoT a gran escala en la era 5G. Es un intermediario de un solo nodo y puede soportar un millón de conexiones simultáneas (EMQX, s.f.). EMQX en comparación con HiveMQ y VerneMQ, proporciona un mejor rendimiento (H., S., & J., 2020)

EMQX maneja los conceptos de tópicos, que son rutas a las cuales los clientes se suscriben. En la imagen 2.4.2 podemos observar una zona con dos cultivos invernaderos donde se manejan dos sensores, uno de temperatura y otro de humedad, en los cuales se podrían tener 4 rutas:

- Publish: INVERADEROS/INVERNADERO 1/TEMP
- Publish: INVERADEROS/INVERNADERO 1/HUME
- Publish: INVERADEROS/INVERNADERO 2/TEMP
- Publish: INVERADEROS/INVERNADERO 2/HUME

De esta forma tendríamos que el tópico general será “INVERNADEROS”, los subtopicos serían “INVERADERO 1”, “INVERNADERO 2” y el tópico en concreto sería “TEMP” y “HUME”.

Figura 2.4 Zona de cultivos invernaderos

Nota. Zona de cultivos invernaderos con los sensores de temperatura y humedad, Adaptado de *Tópicos* [Diagrama] Pablo Sanz, (Sanz, Tópicos)

2.7. Hosting

Permite almacenar la aplicación web, información, imágenes, y video. Se encarga de gestionar servicios de correo. Funciona cuando un usuario ingrese el nombre del dominio de internet, se haga una llamada al hosting el cual devolverá la página web.

2.8. Base de datos

Es un conjunto de datos almacenados, con el objetivo de hacer uso de ellos en un futuro, teniendo de esta manera la información agrupada y organizada. Un ejemplo de ello se puede notar en un sistema IoT que lee ciertos datos del ambiente; para estos sistemas se ve la importancia de las bases de datos, puesto que, si se quiere ver un histórico de las variables ambientales que han recolectado los sensores, toda la información necesita ser guardada.

2.9. Aplicaciones

Es considerado un cliente en el sistema IoT, su función es mostrar los datos que ya han sido tomado y procesados. Es la vista donde el usuario podrá observar y manejar el sistema IoT.

2.10. Node JS

Fue creado con el fin de servir a la creación de programas de red altamente escalables; es un servicio, el cual se va a suscribir a los tópicos para poder guardar los datos tomados por los sensores. Node JS ofrece la integración de cliente-servidor (Chaniotis, Kyriakou, & Tselikas, 2015).

2.11. SOFTWARE Y APLICACIONES

2.11.1. IDE ARDUINO

El IDE de Arduino es una plataforma de creación de electrónica de código abierto, se conforma por hardware y software libre, ofrece multiplataforma pudiéndose instalar en diferentes sistemas operativos (Arduino, s.f.).

2.11.2. Atom

Es un editor de texto y puede ser utilizado para la programación básica de un IDE, es una aplicación útil para programar sobre todo para aplicaciones de escritorio

2.11.3. Vesta

Vesta es un panel de control, su función es de almacenamiento para el hosting web. En él se puede gestionar diferentes servidores como por ejemplo FTP y DNS, además de sus muchas opciones cuenta con certificados SSL gratuitos para una mayor seguridad (VESTA, s.f.).

Vesta está diseñado para proporcionar acceso a archivos en paralelo a programas de aplicación que se ejecutan en equipos múltiples con subsistemas. Vesta utiliza una nueva abstracción de archivos: un archivo no es una secuencia de bytes, sino que se puede dividir en múltiples secuencias disjuntas a las que se puede acceder en paralelo (Corbett & Feitelson, 1996).

2.11.4. *PhpMyAdmin*

Es una herramienta de gestión de base de datos, ofreciendo funciones como crear, eliminar, modificar tablas y los datos que se encuentran dentro de ella. Cuenta con una interface gráfica muy amigable ofreciendo también la opción de trabajar mediante código. Su interface es web y está disponible para bases de datos como Drizzle, MYSQL y MariaDB (phpMyAdmin, s.f.).

2.11.5. *Dashboard*

Es una herramienta cuya función es la de monitorizar datos recolectados para su posterior análisis, mostrándolos de forma visual para un fácil seguimiento de estos. Dentro de sus componentes encontramos botones, graficas, tablas y muchos más elementos.

2.12. Cultivo invernadero

Un invernadero es una estructura cuya función es aislar cultivos de las condiciones climáticas del exterior con el fin de poder controlar la temperatura y la humedad del ambiente y de esta manera sacar cultivos de forma más rápida y efectiva. Además, ayuda al cultivo a estar más libre de plagas que existan en el exterior (ALPI & TOGNONI, 1999).

Está construido por una estructura ya sea hecha en madera, metal o incluso en PVC. También consta de una membrana translúcida cuya función es permitir atrapar

la energía del sol para poder entregarla a las plantas y aislar las cosechas del medio exterior, está elaborado en plástico, vidrio o policarbonato. Además, el invernadero debe contar con sistemas de riego, ventilación y termómetros

2.12.1. *Cultivo de tomate en invernadero*

El tomate es un fruto que proviene de la planta *Solanum lycopersicum* y tiene sus orígenes en América del sur. Su importancia en el mercado es vital para la economía, puesto que el tomate es la hortaliza más cultivada según la FAO (Instituto de investigaciones agropecuarias, 2017) .

Es necesario comprender las condiciones ambientales de la planta, en los cultivos de tomate la cantidad de agua no es muy demandante, según la FAO cada planta va a requerir 75 litros de agua repartidos durante 60 días. Además, se recomienda administrar cada día más de dos litros de agua por planta a partir de la floración del segundo racimo.

En cuanto a la humedad relativa, para un óptimo desarrollo del tomate requiere que este entre un 60% y 80%. Además, la temperatura también es importante controlarla, debido a que se pueden prevenir distintos daños producidos por este factor, en la tabla 4.1 podremos observar las temperaturas para el cultivo de tomate.

El cultivo de tomate bajo invernadero debido a su tecnología es mucho más amigable con el medio ambiente. El tomate es un cultivo transitorio y altamente productivo. El tomate es un producto que usamos casi todos los días debido a sus funciones en el ámbito culinario

Figura 2.5 Temperaturas para el desarrollo de la planta de tomate.

Se hiela la planta	-2°C	
Detiene su desarrollo	10-12°C	
Desarrollo normal de la planta	18-25°C	
Mayor desarrollo de la planta	21-24°C	
Germinación óptima	25-30°C	
Temperaturas óptimas		
Desarrollo	Diurna	23-26°C
	Nocturna	13-16°C
Floración	Diurna	23-26°C
	Nocturna	15-18°C
Maduración		15-22°C

Nota. En el manual de cultivo de tomate bajo invernadero se recomiendan estas temperaturas para el crecimiento y desarrollo óptimo de la planta, Tomado de *Temperaturas para el desarrollo de la planta de tomate*. [Tabla], Instituto de Investigaciones agropecuarias, 2017, (Instituto de investigaciones agropecuarias, 2017).

3. DISEÑO DE LA INVESTIGACION

La metodología en la que se basa este proyecto recae sobre el método experimental, debido que a partir de unos resultados que van a ser obtenidos por medio de sensores, se puede predecir que será bueno o no para los cultivos de tomate, se hará un estudio respectivo en base a dichos datos. Basándose en el hecho anterior, se puede decir que también es un tipo de investigación cuantitativa, ya que los datos que se van a recibir y relacionar, son datos numéricos, que luego al ser empleados en los distintos algoritmos usados se pueden obtener respuestas a cada problemática.

Etapas para el diseño de la investigación

Etapa 1: El proyecto se da inicio con la adquisición de los materiales, y a su vez estableciendo un estudio de caso en un cultivo de tomate.

Etapa 2: Se procede a la implementación y configuración del VPS, además de su hosting interno.

Etapa 3: Se implementa la base de datos a utilizar y el panel de control en lenguaje php.

Etapa 4: Seguidamente se configura MQTT y demás dispositivos que se van a programar.

Etapa 5: Se procede al montaje del servicio Node JS, programando a su vez el código de análisis de datos.

Etapa 6: Por último, se realiza las respectivas pruebas en la maqueta de cultivo invernadero.

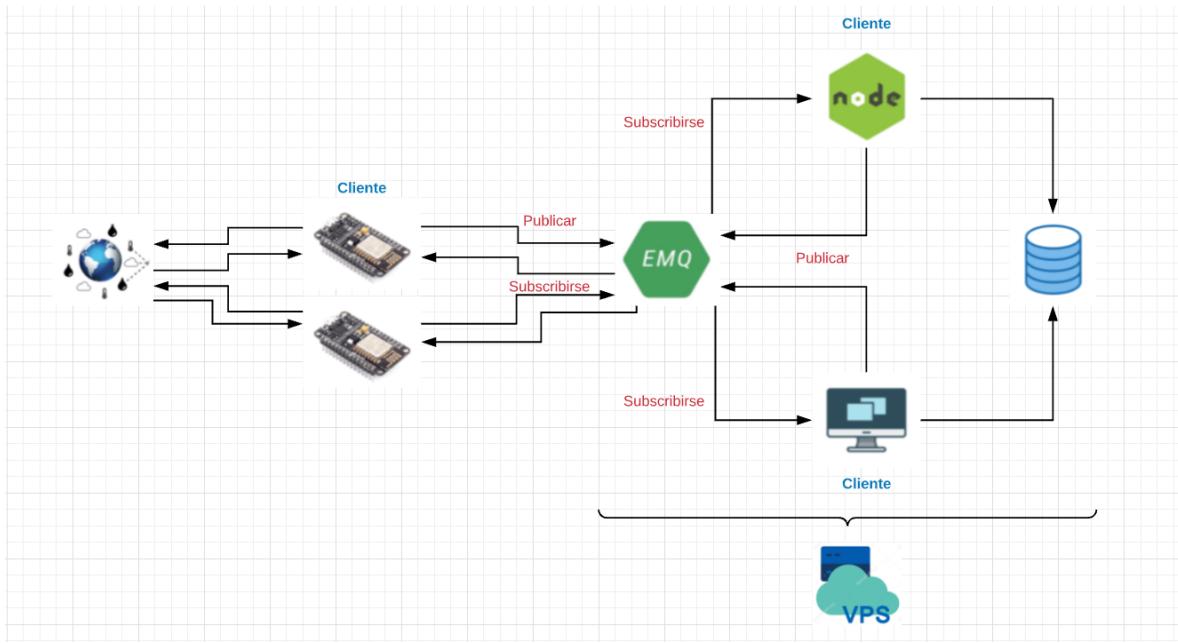
4. DESARROLLO DEL TRABAJO DE GRADO

En esta sección se detalla acerca de la realización del proyecto en general. También, se explica cada parte del proyecto hasta conseguir el producto final. En primer lugar, se explicará el funcionamiento de forma general, para después ir adentrándonos más a fondo en cada una de las partes que conforman todo el proyecto. Es aquí donde se presenta el diagrama general que muestra el funcionamiento de cada una de las partes. En segundo lugar, nos adentraremos a explicar más a fondo cada una de los 7 apartados que conforman el producto final mostrando a su paso diagramas que faciliten la comprensión del proyecto. La Explicación de cada una de las partes se muestra en el orden conforme a como se realizó el sistema IoT de redes de sensores inalámbricos.

4.1. Diagrama de funcionamiento del proyecto

La figura 4.1 muestra el diagrama de funcionamiento del proyecto en general. Como elemento principal de nuestro sistema tenemos el uso de EMQX, un bróker MQTT que tiene como función interconectar todos nuestros clientes ya sea la ESP8266 con la página web o el servicio de Node JS. La placa nodemcu ESP8266 será la encargada de procesar los datos que luego serán publicados en el bróker MQTT. Estos datos son recolectados por los sensores que están conectados a la placa; para el almacenamiento de la información es necesario la implementación de una base de datos, la cual recibe la información del servicio Node Js quien se suscribe a un tópico en el bróker MQTT para la lectura de datos. Otro subscriptor es la aplicación web quien cumple la función de visualización de datos y accionamiento del sistema de riego. Cabe resaltar que todo el aplicativo está ubicado dentro de un servidor privado virtual.

Figura 4.1 Diagrama completo del sistema IoT de la red sensora inalámbrica.



Fuente: Autores

Nota. El diagrama muestran las partes a nivel general que conforman el sistema IoT de la red sensora inalámbrica desde sus aplicativos hasta la esp8266 y los sensores.

4.2. Partes del sistema IoT de la red sensora inalámbrica.

En este apartado se describirán más a fondo cada una de las partes que conforman el sistema IoT de la red sensora inalámbrica que se pueden apreciar en la figura 4.1. Además, se explica cómo se realizó cada una de las secciones que conforman el producto final.

4.2.1. Implementación del servidor privado virtual

La elección del servidor privado virtual estaba entre 3 proveedores, Microsoft azure, google cloud, y Amazon AWS. Para el proyecto se empleó Amazon AWS que ofrece una prueba gratuita de un año.

Dentro del servidor privado virtual (VPS) se instaló el sistema operativo Ubuntu server Its. El servidor es utilizado para almacenar todo el aplicativo utilizado en el sistema IoT, tal como el hosting, la base de datos, el bróker MQTT, el servicio de Node JS, etc. La ventaja de usar un servidor privado virtual son muchas, como por ejemplo ahorrar costos energéticos, internet, hardware entre otros. En la figura 4.2 Podemos ver los recursos utilizados en el servidor privado virtual.

Figura 4.2 Recursos utilizados en el servidor privado virtual

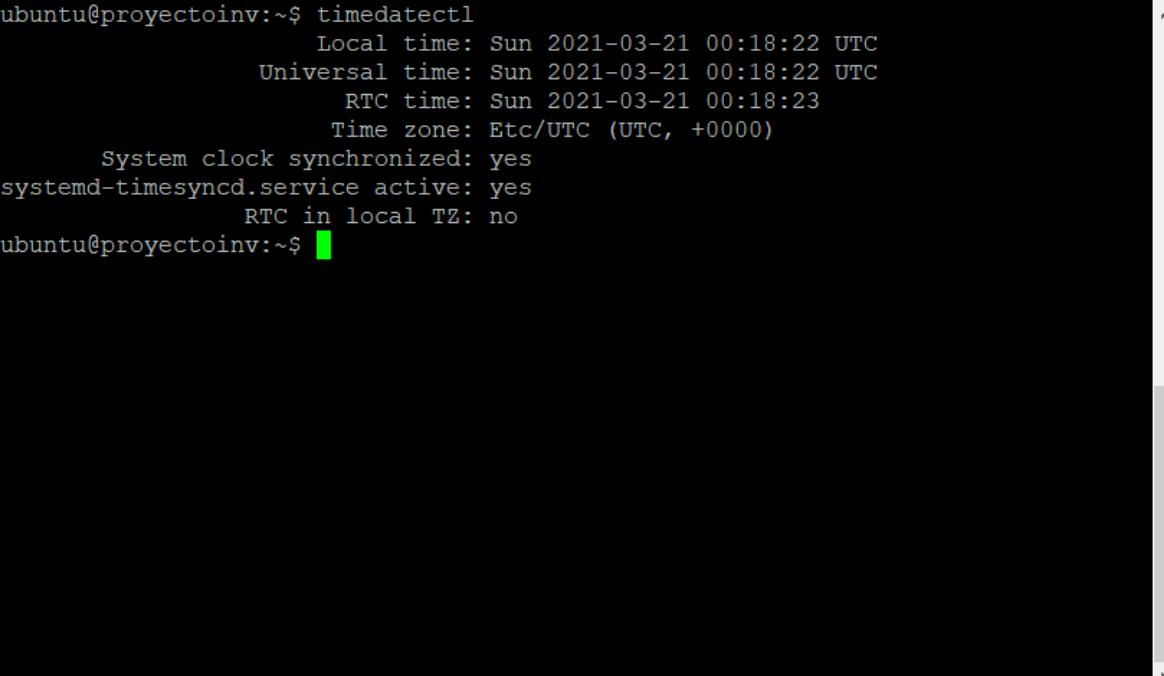
Instancias en ejecución	1	Balanceadores de carga	0
Direcciones IP elásticas	1	Grupos de seguridad	2
Grupos de ubicación	0	Hosts dedicados	0
Instancias (todos los estados)	1	Instantáneas	0
Pares de claves	1	Volúmenes	1

Fuente: Autores

Nota. La imagen muestra los recursos que han sido utilizados del servidor privado virtual que se encuentra en la región de américa del sur (Sao Paulo)

Para poder acceder al servidor es necesario conectarnos remotamente por ssh, generando unas llaves directamente desde Amazon EC2 e insertándolas, ya sea desde la terminal si se usa Linux o Mac, o desde Putty si se usa Windows. De esta manera podremos tener una conexión exitosa tal como observamos en la Imagen 4.3. Es necesario tener en cuenta no perder las llaves ya que no se pueden recuperar y tendrían que generarse nuevamente.

Los requerimientos del hardware se escogerán de acuerdo a la complejidad del sistema IoT a implementar, en nuestro caso, al escoger la prueba gratis no habían muchas opciones así que se escogieron los requerimientos que obsequiaban los cuales fueron más que suficientes para desarrollar todo el proyecto.

Figura 4.3 Conexión a ssh al servidor de Ubuntu.

```
ubuntu@proyectoinv:~$ timedatectl
    Local time: Sun 2021-03-21 00:18:22 UTC
    Universal time: Sun 2021-03-21 00:18:22 UTC
        RTC time: Sun 2021-03-21 00:18:23
      Time zone: Etc/UTC (UTC, +0000)
  System clock synchronized: yes
systemd-timesyncd.service active: yes
        RTC in local TZ: no
ubuntu@proyectoinv:~$
```

Fuente: Autores

Nota. En esta imagen encontramos una conexión ssh al servidor mediante putty y observamos la zona horaria que se maneja.

4.2.2. **Hosting**

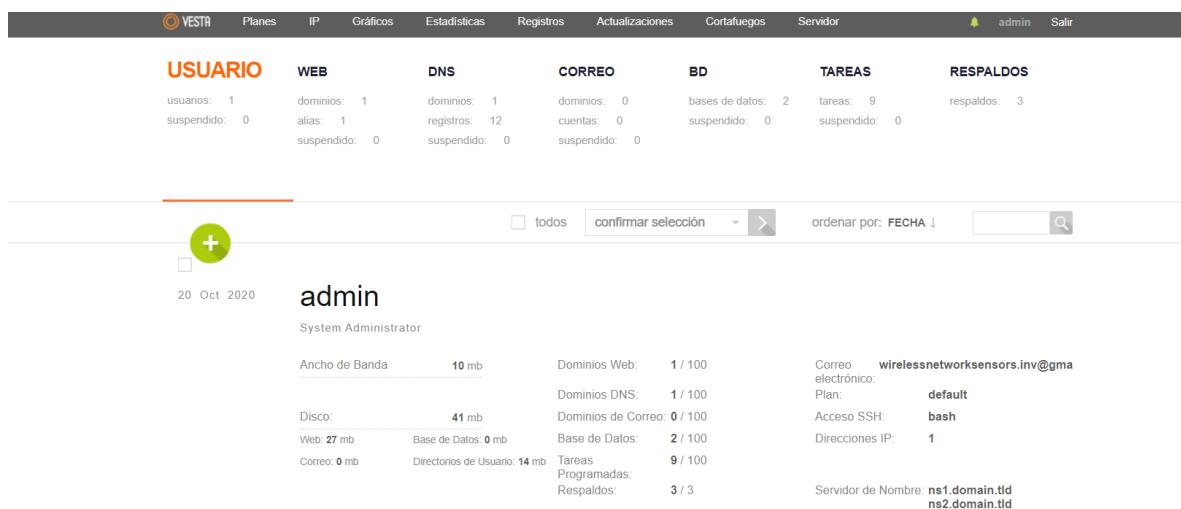
Para tener un hosting funcionando dentro del servidor es necesario de un panel para facilitar la tarea de la configuración del mismo. Es por eso que para este trabajo se utilizó un servicio llamado Vesta control panel.

Vesta control panel permite tener una serie de servicios como firewall, servir web a través de nginx + apache, configurar un servidor de mails, instalar un motor de base de datos, y por último configurar un servicio ftp para poder transferir archivos. Uno de los objetivos del proyecto es tratar de economizar gasto y es por eso que Vesta es ideal debido a que es gratuito.

Anteriormente se mencionaba la necesidad de tener un panel de control para poder solventar la administración de un servicio web hosting, debido que al

configurar todo esto directamente desde Linux por consola es complicado y tedioso, puesto que solo para el montaje se necesitan ciertos conocimientos de lenguaje de programación en bash y conocimientos en Linux para poder implementar este servicio; además, la administración sería otro tema complejo de realizar, Vesta ofrece una interfaz visual y amigable como se puede observar en la figura 4.4. donde aparecen distintas opciones que se pueden realizar.

Figura 4.4 Panel de control en Vesta



The screenshot shows the Vesta Control Panel interface. At the top, there's a navigation bar with links for VESTA, Planes, IP, Gráficos, Estadísticas, Registros, Actualizaciones, Cortafuegos, Servidor, and user icons for admin and Salir. Below the navigation bar, there are several sections with status counts:

- USUARIO:** usuarios: 1, suspendido: 0
- WEB:** dominios: 1, alias: 1, suspendido: 0
- DNS:** dominios: 1, registros: 12, suspendido: 0
- CORREO:** dominios: 0, cuentas: 0, suspendido: 0
- BD:** bases de datos: 2, suspendido: 0
- TAREAS:** tareas: 9, suspendido: 0
- RESPALDOS:** respaldos: 3

Below these stats, there's a search bar with filters for 'todos' and 'confirmar selección'. To the right, there's an 'ordenar por: FECHA ↓' button and a search icon. A large green '+' button is visible on the left. The main area shows a user profile for 'admin' with the title 'System Administrator'. It lists resource usage: Ancho de Banda (10 mb), Disco (41 mb), Web (27 mb), Correo (0 mb), Dominios Web (1 / 100), Dominios DNS (1 / 100), Dominios de Correo (0 / 100), Base de Datos (2 / 100), Tareas Programadas (9 / 100), and Respaldos (3 / 3). It also shows connection details: Correo electrónico (wirelessnetworksensors.inv@gmail.com), Plan (default), Acceso SSH (bash), Direcciones IP (1), and Servidor de Nombre (ns1.domain.tld, ns2.domain.tld).

Fuente: Autores

Nota. Panel de control en Vesta donde se pueden realizar todas las configuraciones pertinentes referentes al hosting.

Una vez obtenido el panel de control en Vesta se procede a adquirir un dominio gratuito por medio de la página freenom que ofrece ciertos dominios gratuitos; y aquí mismo se establece el dominio adquirido a la Ip del VPS

En las redes de comunicación uno de los factores más importantes y que cada día empieza a coger aún más importancia es la seguridad. Al entrar a la web por primera vez se encontrará que la página no tendrá certificados SSL, lo cual no será nada bueno si se quiere establecer un nivel de seguridad apropiado para nuestros proyectos debido a que los datos entre el cliente y el servidor no viajaran encriptados. No obstante, no hay porque preocuparse debido a que Vesta provee unos certificados gratuitos mediante Lets Encrypt, los cuales fueron instalados en el servidor obteniendo encriptación cliente-servidor.

Otro de los factores que se configuraron en Vesta fue el servicio de FTP debido a que, si se quiere tener una aplicación web, va a ser necesario la transferencia de archivos; es por eso que en Vesta se configuro este servicio como se observa en la figura 4.5.

Figura 4.5 Configuración servicio ftp

The screenshot shows the 'Cuenta Adicional FTP' (Additional FTP Account) configuration page. It includes fields for the user name ('Nombre de Usuario') which is 'proyectoinv' and the generated password ('admin_proyectoinv'), a password field ('Contraseña / generar'), a path field ('Ruta') containing 'public_html' and its full path '/home/admin/web/proyectoinv.ga/public_html', and a checkbox for sending credentials via email ('Enviar credenciales FTP al correo'). A link at the bottom left says 'AÑADIR UNA CUENTA FTP ADICIONAL' (Add a new FTP account).

Fuente: Autores

Nota. Configuración del servicio ftp en Vesta cp. para la transferencia de archivos.

4.2.3. *Bases de datos*

En primer lugar, antes de decidir qué base de datos usar, se debe saber qué tipos de bases de datos existen, y básicamente son dos, relacionales y no relacionales. Las relacionales son aquellas en las que se pueden organizar la información en tablas y estas se pueden vincular entre ellas. Por otro lado, se tienen las bases de datos no relacionales en las cuales el concepto de tabla no existe, debido a que ellas manejan algo llamado documentos, y los datos van a ser guardados en esos archivos de texto en una estructura parecida a un json.

Para el desarrollo del proyecto se emplearon bases de datos relacionales debido al mejor manejo que se le tienen a estas. Se utilizó MYSQL que fue creada desde el panel de control Vesta que ya se visualizó anteriormente. Para poder gestionar las bases de datos se empleó un programa llamado Heidi SQL.

En Heidi fueron creadas diferentes tablas como se puede ver en la figura 4.6, para control de usuarios a la página web, control de usuarios al bróker MQTT, manejar una lista de control de accesos en MQTT y por ultimo dos tablas para cada unidad sensora que estará conformada por la ESP8266, sensor de temperatura, humedad ambiente y humedad del suelo que recolectaran y enviaran los datos para posteriormente ser almacenados en la base de datos como se puede observar en la figura 4.7. Si se llegasen a tener más unidades censadoras, sería necesario crear una tabla por unidad sensora implementada

En cuanto a las tablas creadas que hacen referencia al servicio MQTT, el código fue sacado de la documentación oficial de la página de EMQ quienes ofrecen una documentación muy amplia para poder desarrollar cualquiera aplicación usando este servicio.

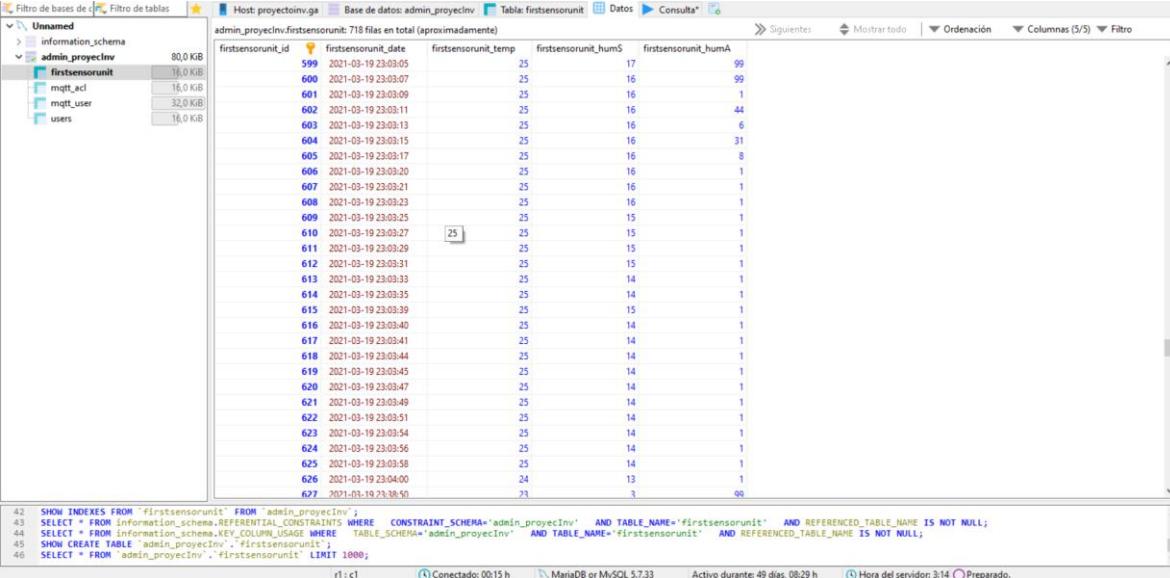
Figura 4.6 Base de datos implementada

<table border="1"> <thead> <tr> <th colspan="2">admin_proyecInv mqtt_acl</th></tr> </thead> <tbody> <tr> <td>#</td><td>id : int(11) unsigned</td></tr> <tr> <td>#</td><td>allow : int(1)</td></tr> <tr> <td> @</td><td>ipaddr : varchar(60)</td></tr> <tr> <td> @</td><td>username : varchar(100)</td></tr> <tr> <td> @</td><td>clientid : varchar(100)</td></tr> <tr> <td>#</td><td>access : int(2)</td></tr> <tr> <td> @</td><td>topic : varchar(100)</td></tr> </tbody> </table>	admin_proyecInv mqtt_acl		#	id : int(11) unsigned	#	allow : int(1)	@	ipaddr : varchar(60)	@	username : varchar(100)	@	clientid : varchar(100)	#	access : int(2)	@	topic : varchar(100)	<table border="1"> <thead> <tr> <th colspan="2">admin_proyecInv secondsensorunit</th></tr> </thead> <tbody> <tr> <td> @</td><td>secondsensorunit_id : int(11)</td></tr> <tr> <td> @</td><td>secondsensorunit_date : timestamp</td></tr> <tr> <td>#</td><td>secondsensorunit_temp : int(4)</td></tr> <tr> <td>#</td><td>secondsensorunit_humA : int(4)</td></tr> </tbody> </table>	admin_proyecInv secondsensorunit		@	secondsensorunit_id : int(11)	@	secondsensorunit_date : timestamp	#	secondsensorunit_temp : int(4)	#	secondsensorunit_humA : int(4)
admin_proyecInv mqtt_acl																											
#	id : int(11) unsigned																										
#	allow : int(1)																										
@	ipaddr : varchar(60)																										
@	username : varchar(100)																										
@	clientid : varchar(100)																										
#	access : int(2)																										
@	topic : varchar(100)																										
admin_proyecInv secondsensorunit																											
@	secondsensorunit_id : int(11)																										
@	secondsensorunit_date : timestamp																										
#	secondsensorunit_temp : int(4)																										
#	secondsensorunit_humA : int(4)																										
<table border="1"> <thead> <tr> <th colspan="2">admin_proyecInv mqtt_user</th> </tr> </thead> <tbody> <tr> <td> @</td><td>id : int(11) unsigned</td></tr> <tr> <td> @</td><td>username : varchar(100)</td></tr> <tr> <td> @</td><td>password : varchar(100)</td></tr> <tr> <td> @</td><td>salt : varchar(35)</td></tr> <tr> <td>#</td><td>is_superuser : tinyint(1)</td></tr> <tr> <td> @</td><td>created : datetime</td></tr> </tbody> </table>	admin_proyecInv mqtt_user		@	id : int(11) unsigned	@	username : varchar(100)	@	password : varchar(100)	@	salt : varchar(35)	#	is_superuser : tinyint(1)	@	created : datetime	<table border="1"> <thead> <tr> <th colspan="2">admin_proyecInv users</th> </tr> </thead> <tbody> <tr> <td> @</td><td>users_id : int(11)</td></tr> <tr> <td> @</td><td>users_date : timestamp</td></tr> <tr> <td> @</td><td>users_email : varchar(60)</td></tr> <tr> <td> @</td><td>users_password : varchar(60)</td></tr> </tbody> </table>	admin_proyecInv users		@	users_id : int(11)	@	users_date : timestamp	@	users_email : varchar(60)	@	users_password : varchar(60)		
admin_proyecInv mqtt_user																											
@	id : int(11) unsigned																										
@	username : varchar(100)																										
@	password : varchar(100)																										
@	salt : varchar(35)																										
#	is_superuser : tinyint(1)																										
@	created : datetime																										
admin_proyecInv users																											
@	users_id : int(11)																										
@	users_date : timestamp																										
@	users_email : varchar(60)																										
@	users_password : varchar(60)																										
	<table border="1"> <thead> <tr> <th colspan="2">admin_proyecInv firstsensorunit</th> </tr> </thead> <tbody> <tr> <td> @</td><td>firstsensorunit_id : int(11)</td></tr> <tr> <td> @</td><td>firstsensorunit_date : timestamp</td></tr> <tr> <td>#</td><td>firstsensorunit_temp : int(4)</td></tr> <tr> <td>#</td><td>firstsensorunit_humS : int(4)</td></tr> <tr> <td>#</td><td>firstsensorunit_humA : int(4)</td></tr> </tbody> </table>	admin_proyecInv firstsensorunit		@	firstsensorunit_id : int(11)	@	firstsensorunit_date : timestamp	#	firstsensorunit_temp : int(4)	#	firstsensorunit_humS : int(4)	#	firstsensorunit_humA : int(4)														
admin_proyecInv firstsensorunit																											
@	firstsensorunit_id : int(11)																										
@	firstsensorunit_date : timestamp																										
#	firstsensorunit_temp : int(4)																										
#	firstsensorunit_humS : int(4)																										
#	firstsensorunit_humA : int(4)																										

Fuente: Autores

Nota. En esta imagen se puede ver las tablas que fueron implementadas en el sistema IoT de la red sensora inalámbrica realizada.

Figura 4.7 Tabla de la unidad sensora número 1



The screenshot shows the MySQL Workbench interface with the following details:

- Host:** proyectoInv.ga
- Base de datos:** admin_proyecInv
- Tabla:** firstsensorunit
- Datos:** Consulta*
- Columnas (5/5):** firstsensorunit_id, firstsensorunit_date, firstsensorunit_temp, firstsensorunit_humS, firstsensorunit_humA
- Data Preview:** A large table with 718 rows showing sensor data from March 19, 2021.
- SQL History:** Shows several SHOW and SELECT queries related to the table.
- Status Bar:** Conectado: 00:15 h, MariaDB or MySQL 5.7.33, Activo durante: 49 días, 08:29 h, Hora del servidor: 3:14, Preparado.

Fuente: Autores.

Nota. En esta imagen se puede apreciar ciertos datos almacenados en la base de datos provenientes de los sensores que capturaron dicha información.

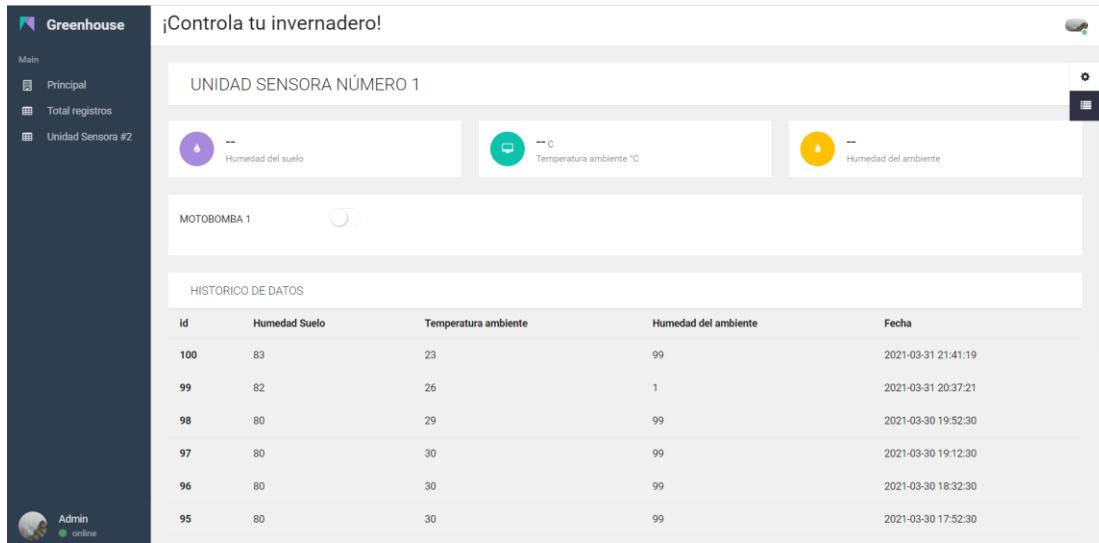
4.2.4. Panel de control en PHP

Esta sección corresponde a la parte visual del sistema IoT, puesto que aquí es donde se van a ver los datos que son recolectados por los sensores, además de poder tomar acciones sobre el cultivo invernadero como por ejemplo activar el sistema de riego.

Para la creación de la página web se utilizó el lenguaje de maquetado HTML, y el lenguaje de programación PHP, además se utiliza JavaScript para realizar ciertas tareas del proyecto.

Existen distintos paneles gratuitos y de pago que se pueden emplear para empezar con la construcción de la página web a nuestro gusto; en este caso se usó flatkit, un panel con muchas herramientas en las que solo fue necesario empezar a modificar para obtener el resultado esperado, como se puede observar en la figura 4.8 y en la figura 4.9 en las cuales se muestra los datos en una tabla y en una gráfica.

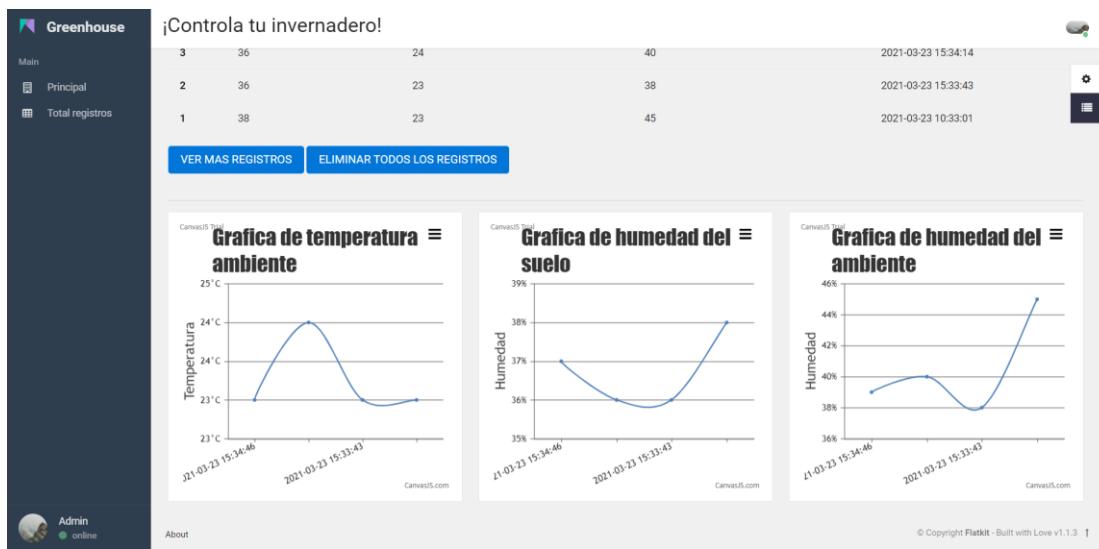
Figura 4.8 Dashboard sistema IoT



Fuente: Autores

Nota. Datos obtenidos por los sensores.

Figura 4.9 Gráfico de datos

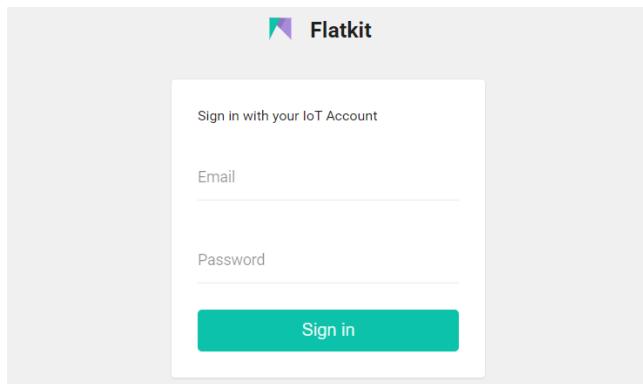


Fuente: Autores

Nota. Gráfico de los datos obtenidos por los sensores.

Además de lo visto anteriormente, la página web cuenta con un login de usuarios para controlar el acceso a la información que se está mostrando, también existe otra pestaña donde se puede observar todo el histórico de datos, y poder filtrar datos por fecha teniendo un acceso más fácil a la información. En la figura 4.10 se muestra el login de usuarios y en la figura 4.11 el histórico de datos.

Figura 4.10 Login de usuarios.



Fuente: Autores

Nota. Login de usuarios de la aplicación web

Figura 4.11 Imagen de los históricos de datos



ID	Humedad Suelo	Temperatura ambiente	Humedad del ambiente	Fecha
5	37	25	39	2021-03-23 15:35:08
4	37	23	39	2021-03-23 15:34:46
3	36	24	40	2021-03-23 15:34:14
2	36	23	38	2021-03-23 15:33:43

Fuente: Autores

Nota. En la pestaña de total de registros se encuentran el histórico de datos y un filtro de fecha para encontrar de forma rápida los datos que se quisiesen visualizar.

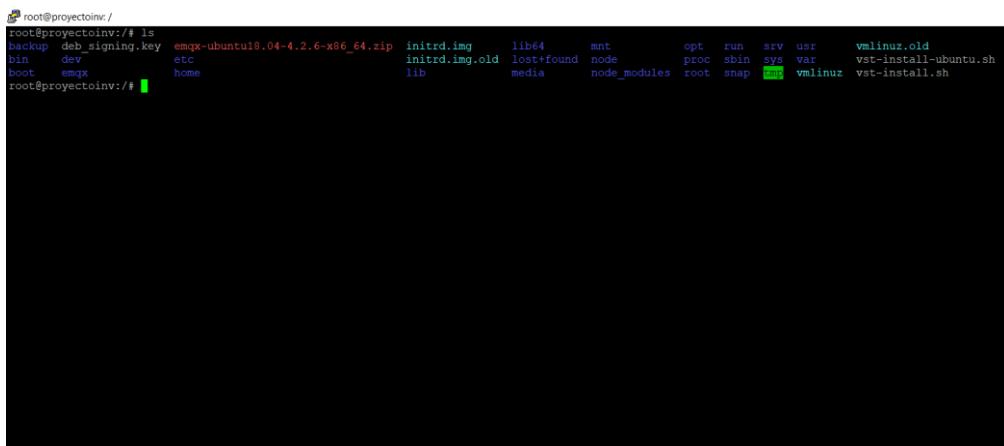
4.2.5. MQTT

Una de las partes más importantes del proyecto es el bróker MQTT. Su función es conectar los clientes para que haya una comunicación entre ellos, en otras palabras, MQTT es un protocolo que ayuda de intermediario haciendo que fluya la información entre aplicaciones web, dispositivos y demás clientes que se tengan conectados al bróker.

En el sistema IoT de la red sensora inalámbrica el bróker MQTT empleado fue EMQ quien es líder en mensajería MQTT de código abierto. Una de las razones por las cuales se escogió este bróker fue su alta escalabilidad y su motor de mensajes en tiempo real.

En primer lugar, para empezar a hacer uso de EMQ es necesario instalarlo dentro del servidor, EMQ se puede instalar en diferentes sistemas operativos ya sea Ubuntu, Windows, macOS, openSUSE etc. El servidor adquirido se le instaló Ubuntu entonces se escogió la versión para Ubuntu 4.2.6, en la página oficial de EMQ muestran los pasos de instalación, una vez instalado quedará en la raíz el directorio de EMQX, como se puede observar en la figura 4.12

Figura 4.12 Servicio de EMQX dentro del VPS



```
root@proyectoinv:/# ls
backup  deb_signing.key  emqx-ubuntu18.04-4.2.6-x86_64.zip  initrd.img   lib64      mnt      opt      run      srv      usr      vmlinuz.old
bin     dev      etc       initrd.img.old    lost+found  node     proc     sbin     sys      var      vmlinuz.old-ubuntu.sh
boot   emqx    home      lib        media     node_modules  root     snap    tmp      vmlinuz  vmlinuz.vst-install.sh
root@proyectoinv:/#
```

Fuente: Autores

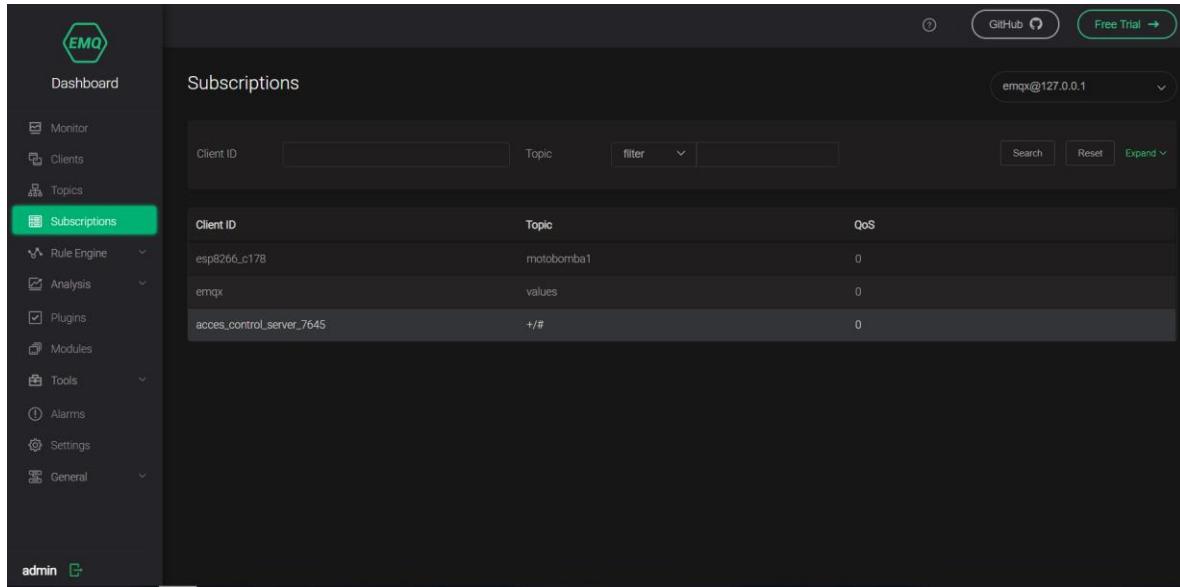
Nota. En esta imagen se encuentra la raíz del sistema donde se puede observar la carpeta del servicio de EMQX

Hay que tener en cuenta que EMQX al igual que muchos otros servicios de telecomunicaciones utilizan ciertos puertos lógicos para la comunicación, es necesario saber que los puertos que necesite EMQX no estén ocupados por otros servicios, de ser así hay que modificar los puertos como fue en este caso, estos puertos fueron modificados dentro del directorio de EMQX en un archivo que está programado en el lenguaje bash; adicionalmente estos puertos hay que abrirlos en el VPS y en VESTA, puesto que aquí es donde están nuestros firewall.

Una vez abierto los puertos, se puede empezar a configurar la conexión entre php y EMQX para enviar y recibir datos por medio de WebSockets. En un principio se hizo de esta manera para poder almacenar valores en la base de datos, pero esto implicó cierto inconveniente, el cual fue que solo se iba a almacenar datos si la aplicación Web estaba abierta, lo cual no era óptimo, es por eso que después se empleó un servicio de node Js para esta tarea. No obstante, la conexión entre php y EMQX se dejó establecida puesto que si se necesitaba la comunicación para poder encender o apagar la motobomba que está conectada a la placa ESP8266.

Para recibir y enviar mensajes fue necesario crear los tópicos dentro de EMQX, ver figura 4.13. Para que los clientes se subscribieran y pudiesen leer y publicar mensajes. En sistema se crearon 4 tópicos el primero llamado motobomba1 que envía el mensaje de “on”, “off” para encender y apagar la motobomba desde la aplicación web, ver figura 4.14. El segundo tópico creado fue el de valúes donde van los datos de los sensores para su registro en la base de datos, ver figura 4.15, el tercer tópico empleado fue vlesp2; existe un tópico adicional que es un comodín, significa que se está subscribiendo a todos los tópicos que hallan dentro del bróker, este comodín es usado por el servicio node Js. Cabe mencionar que los códigos para subscribirse y poder publicar o leer en diferentes tópicos se encuentra en la documentación oficial de EMQ, lo cual es una gran ventaja debido a que no hay que buscar demasiados códigos sin saber que funcionen o no.

Figura 4.13 Dasboard de EMQ



Fuente: Autores

Nota. En esta imagen se puede apreciar los tópicos existentes en EMQX y los clientes que están suscritos a dichos tópicos.

Figura 4.14 Envío de mensajes desde php a MQTT

Encendido	dashboard.php:671
Mensaje enviado!!!	dashboard.php:674
Apagado	dashboard.php:678
Mensaje enviado!!!	dashboard.php:681
Encendido	dashboard.php:671
Mensaje enviado!!!	dashboard.php:674
Apagado	dashboard.php:678
Mensaje enviado!!!	dashboard.php:681

Fuente: Autores

Nota. En la imagen se puede observar los mensajes de encendido y apagado desde el PHP al bróker MQTT para el control de la motobomba.

Figura 4.15 Recepción de mensajes con Node JS.

```
Subscripción exitosa!
Mensaje Recibido desde -> $SYS/brokers/emqx@127.0.0.1/sysdescr Mensaje -> EMQ X
Broker
Mensaje Recibido desde -> $SYS/brokers/emqx@127.0.0.1/version Mensaje -> 4.2.6
Mensaje Recibido desde -> $SYS/brokers Mensaje -> emqx@127.0.0.1
Mensaje Recibido desde -> values Mensaje -> 2,24,99
Fila insertada correctamente
Mensaje Recibido desde -> values Mensaje -> 2,24,99
Fila insertada correctamente
Mensaje Recibido desde -> values Mensaje -> 2,24,99
Fila insertada correctamente
Mensaje Recibido desde -> values Mensaje -> 2,24,99
Fila insertada correctamente
Mensaje Recibido desde -> values Mensaje -> 2,24,99
Fila insertada correctamente
Mensaje Recibido desde -> values Mensaje -> 2,24,99
Fila insertada correctamente
Mensaje Recibido desde -> values Mensaje -> 2,24,99
Fila insertada correctamente
Mensaje Recibido desde -> values Mensaje -> 2,24,99
Fila insertada correctamente
Mensaje Recibido desde -> values Mensaje -> 2,24,99
Fila insertada correctamente
```

Fuente: Autores

Nota. En esta imagen se observa los mensajes que recibe node Js desde la ESP8266 actuando EMQ como intermediario, así como la inserción a la base de datos de estos valores.

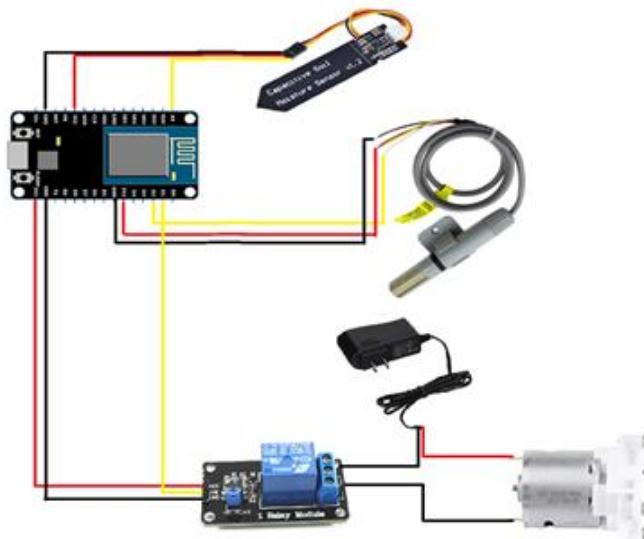
4.2.6. Dispositivo IoT

Una vez configurado el bróker MQTT se puede conectar cualquier cliente que se desee, tal como los dispositivos IoT que en este caso se empleó una ESP8266. La programación de la placa es una de las más sencillas de todo el sistema implementado, no obstante, no siempre será así, puesto que en ciertos casos se requerirán sistemas más robustos.

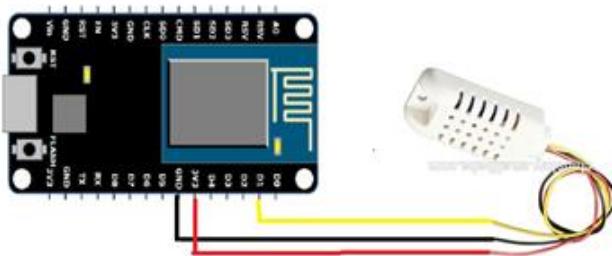
Para la red sensora inalámbrica se empleó, aparte de la ESP8266 un sensor de temperatura y humedad del ambiente, un sensor de humedad del suelo, un relé y una mini bomba de agua empleada en el sistema de riego; también se empleó otra esp8266 solamente con un sensor de temperatura humedad. El grafico se podrá observar en la figura 4.16 a y b respectivamente.

Figura 4.16 Esquema del circuito de la red sensora inalámbrica

A)



B)



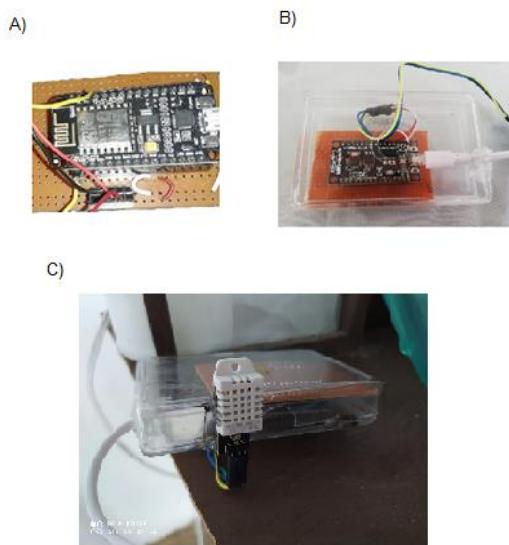
Fuente: Autores

Nota. En esta imagen podemos observar el diseño del circuito empleado el cual fue ubicado dentro de la maqueta para la toma de variables ambientales.

Para poder realizar las conexiones, el circuito fue ubicado en una baquela, ver figura 4.17 a, posteriormente se ubica en una cajita dentro del cultivo para protegerlo del agua del sistema de riego. Los materiales utilizados para proteger a la ESP8266 del ambiente deberán variar al implantarlo en un invernadero real utilizando acrílico

en vez de madera para mayor seguridad de aislamiento como se observa en la figura 4.17 b y c que fue la segunda unidad sensora implementada. En cuanto al sistema eléctrico deberá estar diseñado para la intemperie.

Figura 4.17 Conexiones de la Esp8266



Fuente: Autores

Nota. En la imagen A se encuentra la baquela utilizada con la ESP8266 con las respectivas conexiones que parten a los sensores en cuestión de la unidad sensora uno, en la imagen B mostramos la unidad sensora número 2 con sus respectivas conexiones.

En cuanto a la programación realizada en la ESP8266 es muy sencilla primero que todo se utilizó un editor de código llamado visual studio code junto con una herramienta llamada platformio que ayudara a programar cómodamente. Aquí se programa la conexión a internet, la lectura de datos de los sensores, la conexión a MQTT para subscribirse a un tópico y poder publicar información que posteriormente será leída por node Js y almacenada en la base de datos, además de leer información que vendrá desde PHP para poder activar la motobomba y poner en marcha el sistema de riego.

4.2.7. Node Js

Con las partes explicadas anteriormente ya se puede poner a funcionar un sistema IoT, que tome datos, los envíe y muestre en php, habiendo una comunicación en ambas direcciones entre php y la ESP8266, pero surge un pequeño problema y es que para que se estén recibiendo y almacenando información en la base de datos la aplicación web tendría que estar abierta todo el día. Es aquí donde entra node Js un servicio que estará dentro de nuestro servidor y que se suscribirá a todos los tópicos para obtener y guardar la información para futuro análisis. Este servicio será un cliente de MQTT y una base de datos en MYSQL.

En primer lugar, para implementar el servicio de node Js, es necesario realizar la instalación que se hará dentro de nuestro servidor privado virtual, una vez instalado y creado el proyecto de node Js se procede a conectarse con MYSQL y posteriormente se conectará a MQTT suscribiéndose a todos los tópicos y poder recibir la información que se estén publicando en ellos. Con una consulta se insertan los valores que llegan de MQTT y una vez realizado todo esto se ejecuta el archivo de node Js desde el servidor, como se observa en la figura 4.15.

Una vez se quiera cerrar la conexión presentara el inconveniente que al finalizar la sesión ssh la ejecución del programa va a pararse, es aquí donde entra un gestor de procesos de node en Linux que se llama “pm2”, el cual tendrá que instalarse también en el servidor y se ejecuta como se observa en la figura 4.18. Una vez hecho esto se podrá terminar la sesión ssh puesto que el script estará ejecutándose en segundo plano y no habrá problema de que se cortase la ejecución.

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

Figura 4.18 Gestor de procesos pm2

```
root@proyectoinv:/home/admin/web/proyectoinv.ga/public_html/node#
root@proyectoinv:/home/admin/web/proyectoinv.ga/public_html/node# pm2 start index.js
[PM2] Applying action restartProcessId on app [index] (ids: 0)
[PM2] [index] (0) ✓
[PM2] Process successfully started

```

id	name	namespace	version	mode	pid	uptime	status	cpu	mem	user	watching	
0	index	default	1.0.0	fork	20258	0s	0	online	0%	1.2mb	root	disabled

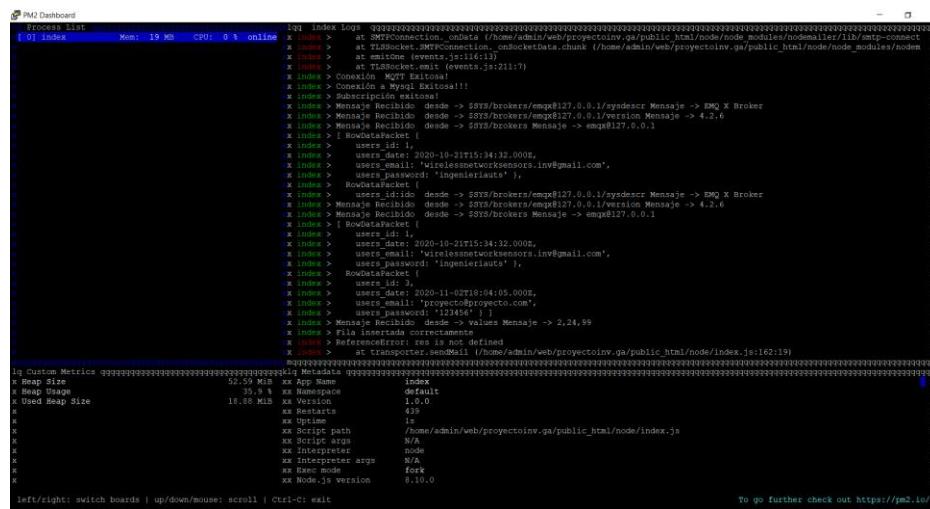
```
root@proyectoinv:/home/admin/web/proyectoinv.ga/public_html/node#
```

Fuente: Autores

Nota. En esta imagen muestra el servicio de pm2 corriendo.

Para poder monitorear las salidas, pm2 cuenta con una herramienta llamada pm2 monit, que permitirá revisar si hay algún problema con la ejecución o si están llegando datos del bróker MQTT, ver figura 4.19.

Figura 4.19 Herramienta de monitoreo pm2 monit



The screenshot shows the pm2 Dashboard interface. The top section displays a process list for 'index' with details like Mem: 19 MB, CPU: 0 %, and status online. Below this, the 'Custom Metrics' section provides detailed metrics for the process, including:

- Heap Size: 52.4 MB
- Used Heap Size: 10.88 MB
- Heap Usage: 35.9 %
- NameSpace: default
- Version: 1.0.0
- Uptime: 1s
- Script path: /home/admin/web/proyectoinv.ga/public_html/node/index.js
- ES6: N/A
- Interpreter mode: node
- Interpreter args: N/A
- Exec mode: fork
- Node.js version: 8.10.0

At the bottom right, there is a link: 'To go further check out https://pm2.io/'.

Fuente: Autores

Nota. En esta imagen podemos ver la herramienta pm2 en ejecución que me muestra ciertos errores, además de algunos logs que se programaron para corroborar algunos eventos realizados.

Dentro de Node JS se desarrolló envío de alertas a correos electrónicos, con la finalidad de avisar al gestor sobre eventos de temperatura altas o bajas, o si necesita regar el cultivo debido a falta de humedad en el suelo. Para poder realizar estas alarmas se empleó nodemailer un módulo para aplicaciones Node JS para permitir fácil envío de correos electrónicos. (Nodemailer, s.f.). La implementación e instalación de nodemailer se puede encontrar dentro de la página oficial. En el proyecto solo se evalúa las condiciones para enviar un mensaje que se estará ejecutando cada cierto tiempo enviando así las alertas vía correo electrónico.

4.3. Maqueta del cultivo invernadero

La elaboración de la maqueta del cultivo invernadero fue realizada en madera, y para su recubrimiento se emplearon los materiales utilizados en invernaderos reales para simular las condiciones lo más parecidos a la vida real como se observa en la figura 4.20.

Figura 4.20 Maqueta del cultivo invernadero



Fuente: Autores

Nota. En esta imagen se observa la maqueta del cultivo invernadero con su sistema de riego ya instalado.

Para el sistema de riego debido a que no se consiguen surtidores, uniones, y reducciones del diámetro de la manguera implementada; se pasó por la parte superior la manguera, implementando un sistema de riego por goteo, en el cual el agua es absorbida y depositada en el mismo lugar mientras fluye por la manguera y va cayendo el agua al suelo.

La ESP8266 fue ingresada dentro de una caja para aislarla del ambiente y de allí sale el sensor de temperatura y humedad del ambiente que fue colocado sobre una tabla que va en dirección al techo del invernadero, y el sensor de humedad del suelo. Como se puede observar en la figura 4.21 a y b respectivamente

Figura 4.21 Interior de la maqueta del cultivo invernadero

A)



B)

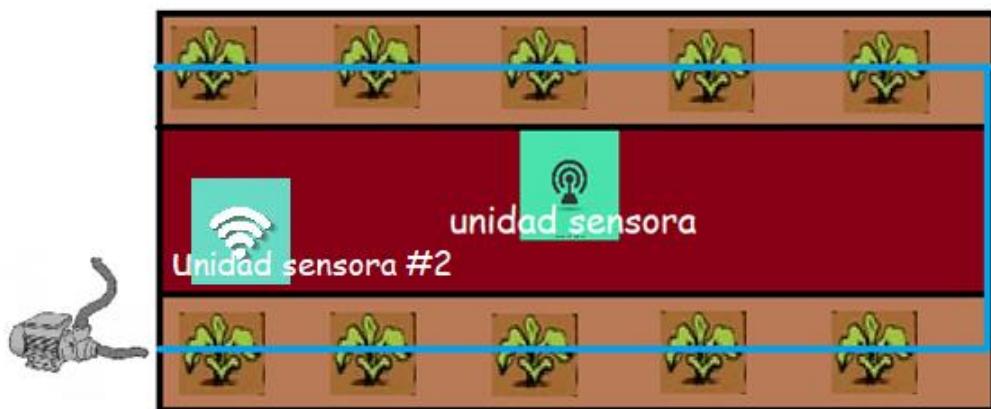


Fuente: Autores

5. RESULTADOS

Esta sección muestra los resultados obtenidos del funcionamiento del proyecto de grado mediante la obtención de valores de humedad del suelo, humedad y temperatura del ambiente. Dichos resultados serán mostrados en unas tablas y graficas dentro de la aplicación web gracias a los valores almacenados en una base de datos MYSQL. Por otro lado, comprobaremos el funcionamiento del envío de correos electrónicos ante eventos presentados y a su vez la puesta en marcha del sistema de riego desde la página web.

Figura 5.1 Esquema interior del cultivo invernadero



Fuente: Autores

Nota. Disposición de la red sensora inalámbrica dentro del cultivo invernadero.

El objetivo del proyecto es comprobar el funcionamiento de la red sensora inalámbrica para mejorar la condición de la planta monitoreando en este caso temperatura y humedad del ambiente y humedad del suelo. Para el cumplimiento de este fin se dispone de una unidad sensora que está conformada por tres sensores y una motobomba de agua y otra unidad sensora que mide solamente temperatura y humedad del ambiente como se muestra en la figura 5.1.

5.1. Parámetros presentes en el monitoreo.

La medición tomada por la unidad sensora tiene como base estudios agronómicos que muestran las condiciones ambientales necesarias para el buen crecimiento de la planta. Estos parámetros de referencia son establecidos dentro de la programación del sistema IoT y servirán como análisis para el uso y envío de alertas los parámetros son los siguientes:

- **Temperatura:** Se considera una temperatura alta para cultivos de tomate después de los 29 °C y si está por debajo de los 20 °C se le considera temperaturas bajas. Estos valores se tienen en cuenta en el dia, en la noche varia el rango de aceptación de temperatura.
- **Humedad del suelo:** Este factor depende de la ubicación y tipo de suelo, por ejemplo, para suelos profundos y no tan secos se utiliza un rango de humedad dentro del 30% al 45%.
- **Humedad del ambiente:** Este factor no es tan importante como los otros debido a que hoy día se puede cultivar tomate en climas más calidos, sin embargo, se establece que la humedad relativa para un cultivo de tomate debe estar entre el 50% y 80%.

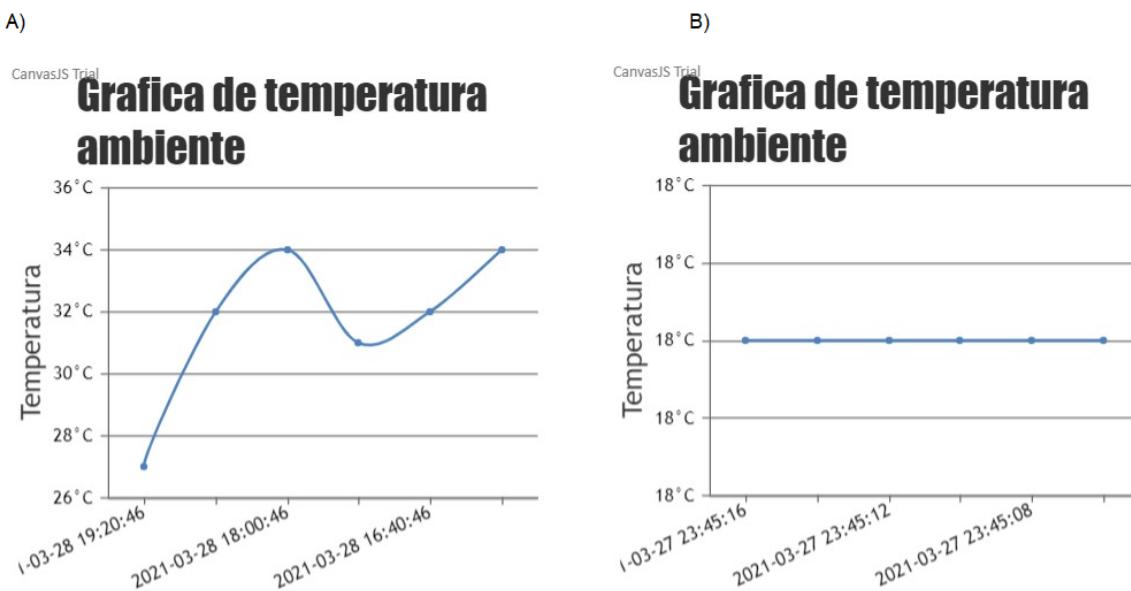
Hay que tener en cuenta que estos factores dependen de la ubicación del sensor, y del material con el que está fabricado el invernadero debido que hay de diferentes tipos.

5.1.1. Parámetro de temperatura ambiente

En la figura 5.2, a y b contiene las mediciones con fecha de la temperatura del ambiente en el día y la noche respectivamente del año 2021. Se puede observar claramente que la temperatura puede variar dependiendo de las horas del día; incluso del tiempo como tal puesto que habrá días nublados y otros que no. Además, podremos obtener un registro más detallado por medio de una tabla y se podrá filtrar

por fecha si así se quisiese, sacando distintos dantos que se muestran en la tabla 1.

Figura 5.2 Grafica de valores extraídos de la base de datos



Fuente: Autores

Nota. La figura a) corresponde a los valores obtenidos de día; la figura b) corresponde a los valores obtenidos en la noche.

Tabla 1 valores de temperatura extraída de la base de datos

Fecha	Temperatura
2021-03-28 08:22:37	17
2021-03-28 12:00:46	18
2021-03-28 12:40:46	20
2021-03-28 13:20:46	22
2021-03-28 16:40:46	32
2021-03-28 19:20:46	27
2021-03-28 20:00:46	25

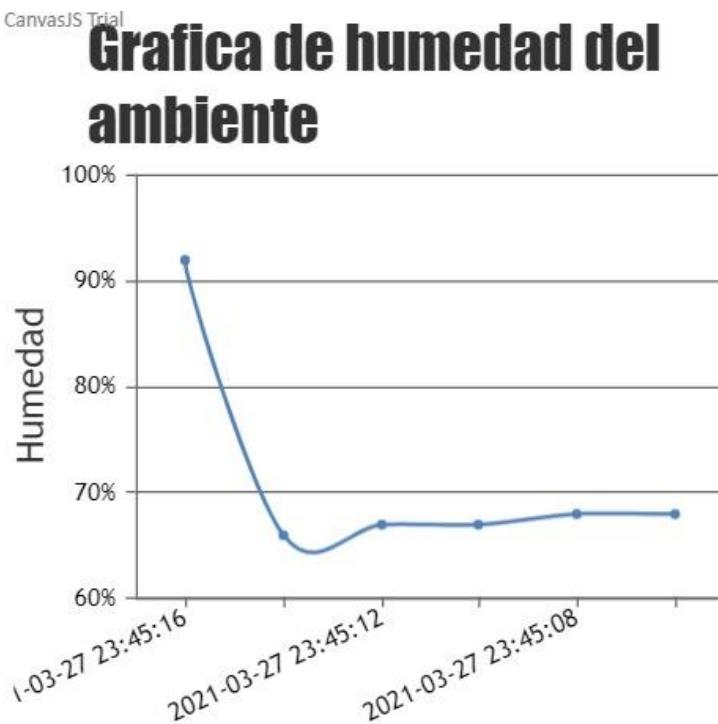
Fuente: Autores

Nota. La tabla mostrada en la figura muestra los datos que fueron filtrados de acuerdo a las fechas establecidas, tener en cuenta que la fecha que aparece corresponde a la fecha del servidor en Brasil.

5.1.2. Parámetro de humedad ambiente

La figura 5.3 muestra los datos de humedad del ambiente obtenido por el sensor en forma gráfica, en el día y en la noche. Además, que al igual que los otros datos, serán mostrados en una tabla para el total de registros como se muestra en la tabla 1.

Figura 5.3 Grafica de valores de humedad del ambiente extraídos de la base de datos



Fuente: Autores

Nota. La figura a) corresponde a los valores obtenidos de día; la figura b) corresponde a los valores obtenidos en la noche.

Como se observa en la figura 5.4 la humedad ambiente aumenta en la noche con respecto al día, pero los valores de cuidado del cultivo tienden a ser más de cuidado en el día que en la noche.

5.1.3. Parámetro de humedad del suelo

La humedad del suelo es importante debido a que un suelo con abundante agua puede dañar el cultivo, de otro lado, un cultivo con falta de agua también va a ser perjudicial para la planta. La figura 5.4 nos muestra los valores tanto de humedad y temperatura del ambiente, como la humedad del suelo en tiempo real de esta manera al entrar a la página nos encontraremos con todos estos datos que irán cambiando de acuerdo a las condiciones del ambiente. Al igual que en las otras dos variables ambientales la humedad del suelo también es graficada como se puede observar en la figura 5.5.

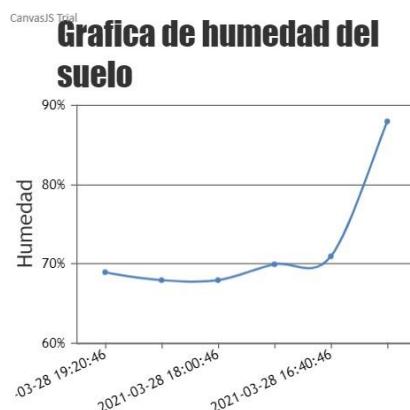
Figura 5.4 valores de humedad del suelo extraído de la base de datos.



Fuente: Autores

Nota. Además de la temperatura del suelo, en esta imagen se puede observar los valores obtenidos en tiempo real.

Figura 5.5 Grafica de valores de humedad del suelo extraídos de la base de datos



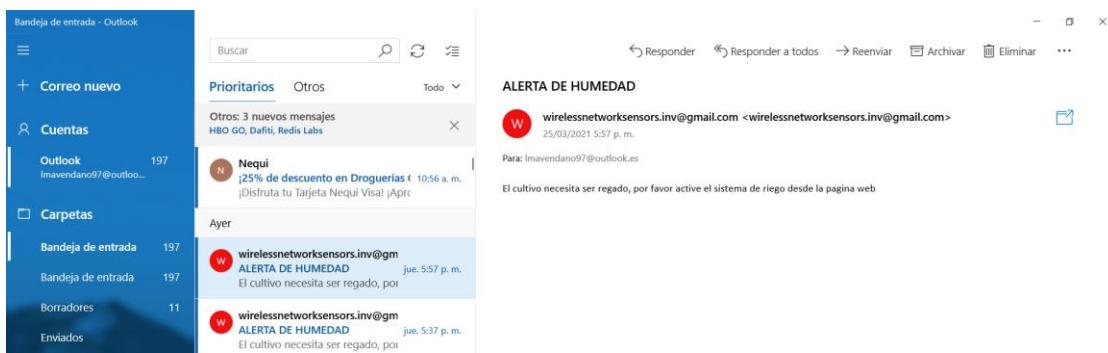
Fuente: Autores

5.2. Alertas por E-mail y funcionamiento del sistema de riego.

Las alertas en un sistema IoT son importantes debido a que, si hay algo que se quiere medir, es necesario conocer cuando los resultados de esa variable están fuera de los rangos de aceptación que se necesita para obtener una buena calidad del producto. Una de las formas más económicas de enviar alarmas es por medio de correos electrónicos; en el sistema IoT de la red sensora inalámbrica cada 40 minutos se están haciendo análisis de las condiciones ambientales para decidir si se envía una alerta o no.

Al implementar estas alertas se tuvo unos muy buenos resultados, pues el sistema de alertas funcionó perfectamente, detectando, por ejemplo, cuando la humedad del suelo estaba por debajo de los umbrales aceptados enviando un mensaje para que se activara el sistema de riego como se puede apreciar en la figura 5.6

Figura 5.6 Envío de alertas vía correo electrónico.



Fuente: Autores

Nota. La Imagen muestra el mensaje que llega desde el correo electrónico del proyecto a un correo electrónico usuario, acerca de una alerta de humedad.

Una vez leído el E-mail se procede a activar el sistema de riego desde la aplicación web como se muestra en la figura 5.7, y fue estimulante el ver como en cuestión de milisegundos el sistema de riego ya se había activado, ver figura 5.8, teniendo en cuenta que la orden de activación viaja desde Brasil que es donde se

encuentra la aplicación web hasta Colombia que es donde se encuentra la maqueta del invernadero

En la tabla 2 se muestra el tiempo en que se han enviado mensajes de alertas durante un día y de qué tipo; comparando de esta manera que tan eficiente puede llegar a ser el sistema de alarma implementado.

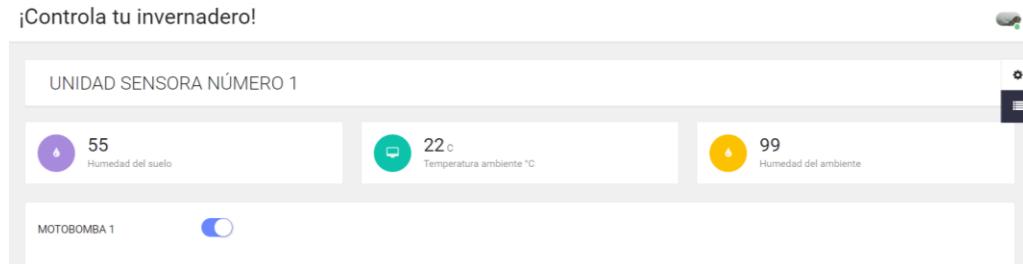
Tabla 2 Registro de sistemas de alerta.

hora	Asunto de alarma
4:02 am	Alerta de temperatura (baja)
5:42 am	Alerta de temperatura (baja)
6:12 am	Alerta de humedad (baja)
9:00 am	Alerta de temperatura (alta)
2:20 pm	Alerta de temperatura (alta)
3:00 pm	Alerta de humedad (alta)

Fuente: Autores.

Figura 5.7 Activación del sistema de riego desde el panel de control en PHP

¡Controla tu invernadero!



Fuente: Autores.

Figura 5.8 Sistema de riego en funcionamiento.



Fuente: Autores.

6. CONCLUSIONES

El sistema IoT de la red sensora inalámbrica montada con la Esp8266, el bróker EMQ, y los demás clientes que hicieron parte de este sistema, es completamente confiable en cuanto a transmisión de datos, ya que no presento perdida de información teniendo en cuenta la distancia que existe desde el invernadero al servidor.

De acuerdo a los datos obtenidos por los sensores se concluye que la variación en el clima varía dependiendo de la cantidad de radiación solar halla, por lo que el cultivo está en constante cambio dependiendo de la hora y de cómo se encuentre el día en general. Estos análisis son posibles gracias a todas las partes montadas en el sistema IoT de la red sensora inalámbrica, como la aplicación web que sirvo como visualización, la base de datos que mostro su utilidad almacenando información, el bróker MQTT, interconectando los diferentes clientes para la comunicación, y por supuesto los sensores quienes tomaban la información.

En cuanto a la construcción del sistema IoT fue bastante económico debido a que la mayoría de las herramientas implementadas son gratuitas, y los equipos utilizados para la medición no fueron muy costosos; brindando incluso un sistema de alertas vía E-mail gracias al servicio de Node JS implementado dentro del VPS. No obstante, este sistema de alertas provocaba spam de correos electrónicos, debido a esto se implementó que el análisis del ambiente para él envío de correos electrónicos se hiciese cada media hora.

Lo anterior permite señalar que se pudo implementar un sistema IoT de una red sensora inalámbrica, tomando datos del ambiente y analizándolos, para generar alertas de acuerdo a eventos ocurridos en el clima. Permitiendo así mejor control para el cuidado de las plantas, además del control sobre el sistema de riego implementado.

No obstante, esto no indica que sea la mejor red inalámbrica implementada para un cultivo invernadero, debido a que se puede implementar mayor variedad de sensores que ayuden a la salud y el cuidado de la planta, a fin de evitar que en un futuro otros factores puedan atentar con el cultivo.

En cuanto al factor de humedad del suelo, se pudo controlar el nivel de agua sobre el cultivo, permitiendo ahorros energéticos en cuanto a bombeo de agua, mejor manejo de este recurso, y por ultimo evitando daños drásticos por exceso de humedad en el suelo.

La humedad del ambiente medida por el sensor am2305 no funcionó como se esperaba debido a que a veces tomaba valores acordes y otras veces no, de otro lado, el sensor am2302 tomó valores más precisos y reales.

Es necesario tener en cuenta que la velocidad en que el sistema IoT de la red sensora inalámbrica depende de la velocidad del internet que se esté implementando en el sistema

Gracias a este sistema implementado las plantas evolucionan mejor debido a que conociendo las condiciones en las que están se puede tomar acción correctiva ante dicha situación, esto hace posible que se puedan evitar enfermedades de las plantas mejorando la productividad lo que favorece económicaamente al campesino. Además, se aprovecha de mejor manera el agua, un recurso natural importante que se debe cuidar y no malgastar.

7. RECOMENDACIONES

Como primera medida se recomienda montar la red inalámbrica en un invernadero real, haciendo las respectivas pruebas y análisis con más unidades censoras y por más tiempo, empleando, además, diferentes referencias de sensores para comprar la durabilidad y fiabilidad de los sensores empleados. También se recomienda comprar la producción entre un sistema invernadero con una red sensora inalámbrica y otro que carezca de esta para saber en qué porcentaje aumenta la producción.

Como segunda medida, se recomienda utilizar mayor variedad de sensores con el fin de mejorar la producción y salud de la planta obteniendo como resultado final mayores ganancias por cultivo. Además de implementar control sobre el sistema de alimentación de las plantas como se hizo con el sistema de riego puesto que ambos funcionaran de la misma forma.

Otra recomendación necesaria es en cuanto a la presentación de los datos, para que se implementen distintas graficas cubriendo un mayor alcance el tiempo para facilitar el análisis y visualización de datos; además, la posibilidad de generar informes de cómo va el cultivo hasta el momento de acuerdo con los datos obtenidos y de esta manera aprovechar los datos realizando un mejor tratamiento de estos.

En cuando al sistema de envío de notificaciones se recomienda hacerlas vía WhatsApp o mediante mensajes de texto, si bien es cierto que generara un costo extra, estos medios son más utilizados por las personas en general lo que brindara en cierta manera mejor comunicación entre el sistema IoT y los usuarios.

8. REFERENCIAS BIBLIOGRÁFICA

- Aakvaag, N., & Frey, J.-E. (Febrero de 2006). Redes de sensores inalámbricos. *ABB*, 39-42. Obtenido de http://www.materialelectrico.com.co/pdf/ABB/02-2006/39-42%25202M631_SPA72dpi.pdf
- ALPI, A., & TOGNONI, F. (1999). *Cultivo en invernadero*. Madrid: Ediciones Mundi-Prensa.
- AMAZON. (s.f.). AMAZON AWS. Obtenido de AMAZON AWS: <https://aws.amazon.com/es/ec2/>
- Arduino. (s.f.). *Arduino*. Obtenido de Arduino: <https://www.arduino.cc/en/Guide/Introduction>
- Cama-Pinto, A., De la Hoz, E., & Cama-Pinto, D. (2012). Las redes de sensores inalámbricos y el Internet de las cosas. *REDICUC*, 163-172.
- Cama-Pinto, A., Gil-Montoya, F., Gómez-Lopez, J., & Manzano-Agugliaro, F. (2014). Sistema inalámbrico de monitorización para cultivos en invernadero. *revistas UNAL*, 164-170.
- Chaniotis, I. K., Kyriakou, K.-I. D., & Tsilikas, N. D. (2015). ¿Es Node.js una opción viable para crear aplicaciones web modernas? Un estudio de evaluación del desempeño. *Springer*, 1023-1044.
- Chaudhary, D., Nayse, S., & Waghmare, L. (2011). Application of wireless Sensor Networks for greenhouse parameter control in precision agriculture. *International Journal of Wireless & Mobile Networks*, 140-149.
- Corbett, P. F., & Feitelson, D. G. (1996). El sistema de archivos Paralelos Vesta. *ACM DIGITAL LIBRARY*, 225-264.
- EMQX. (s.f.). *EMQX*. Obtenido de EMQX: <https://www.emqx.io/>
- FAO. (24 de Julio de 2000). [fao.org](http://www.fao.org/noticias/2000/000704-s.htm). Obtenido de fao.prg: <http://www.fao.org/noticias/2000/000704-s.htm>
- Flores Medina, M., Flores Garcia, F., & Velasco Martínes, V. (2015). Monitoreo de humedad en suelo a través de red inalámbrica de sensores. *Scielo*, 75-88.

- Gascón, D. (2010). Redes de Sensores Inalámbricos, la tecnología invisible. *Tecnología y sociedad*, 53-55.
- Gómez, J. L. (2020). *El IoT como aliado del agro*. Obtenido de El IoT como aliado del agro: <https://www.portafolio.co/opinion/jose-luis-gomez/el-iot-como-aliado-del-agro-532902>
- H., K., S., G., & J., R. (2020). *Una comparación de los agentes de MQTT para la computación periférica de IoT distribuida*. Springer. Obtenido de https://link.springer.com/chapter/10.1007/978-3-030-58923-3_23#citeas
- Instituto de investigaciones agropecuarias. (2017). Manual del cultivo del tomate bajo invernadero. (A. T. P., Ed.) Santiago de chile, Santiago de chile, chile. Obtenido de http://bibliotecadigital.ciren.cl/bitstream/handle/123456789/29478/INIA_Libro_0048.pdf?sequence=1&isAllowed=y
- Kouro, S. (1 de 06 de 2001). Automatización Industrial: Sensores de Humedad. Chile. Obtenido de https://d1wqtxts1xzle7.cloudfront.net/42590997/Sensores_de_Humedad.pdf?1455226747=&response-content-disposition=inline%3B+filename%3DAutomatizacion_Industrial.pdf&Expires=1602199267&Signature=awZSvbCiT554PAXMw77ZU5fCE98bp7TwgUvzGnxuuO42VABsjofkDBCjWBIfWNi
- Leukert, B. (2021). ¿Qué es *internet de las cosas*? Obtenido de Definición de internet de las cosas: <https://www.sap.com/latinamerica/insights/internet-of-things.html>
- MQTT. (s.f.). *mqtt*. Obtenido de mqtt.org: <https://mqtt.org/>
- Negal, J. (2012). *Repositorio Cepal*. Obtenido de https://repositorio.cepal.org/bitstream/handle/11362/4011/S2012079_es.pdf
- Nodemailer. (s.f.). *Nodemailer*. Obtenido de <https://nodemailer.com/about/>
- Perilla, A., Rodríguez, L. F., & Bermúdez, L. T. (2011). Estudio técnico-económico del sistema de producción de tomate bajo invernadero en Guateque, Sutatenza y Tenza(Boyacá). *Revista Colombiana de ciencias hortícolas*, 220-232.
- phpMyAdmin. (s.f.). *phpMyAdmin*. Obtenido de <https://www.phpmyadmin.net/>

Quiñones-Cuenca, M., González-Jaramillo, V., Torre, R., & Jumbo, M. (febrero de 2017).

Sistema De Monitoreo de Variables Medioambientales Usando Una Red de Sensores Inalámbricos y Plataformas De Internet De Las Cosas. *Scielo*, 8, 329-343. Obtenido de http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S1390-65422017000100329

Red Hat. (2021). ¿Qué es el Internet de las cosas? Obtenido de ¿Qué es el Internet de las cosas?: <https://www.redhat.com/es/topics/internet-of-things/what-is-iot>

Semle, A., & K. -e. (2016). Protocolos IIoT para considerar. AADECA, 32-35.

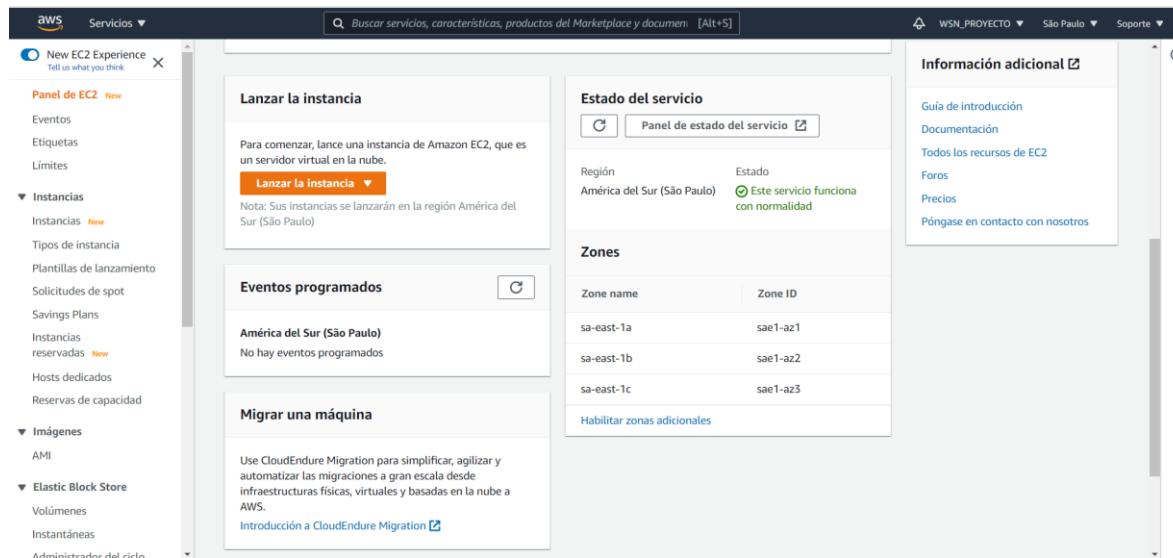
Urbano-Molano, & Aparicio, F. (2013). Redes de Sensores Inalambricos Aplicadas a Optimizacion en agricultura de precision para cultivos de cafe en Colombia. *Journal de Ciencia e Ingenieria*, 46-52.

VESTA. (s.f.). VESTA. Obtenido de VESTA CONTROL PANEL: <https://vestacp.com/>

9. APÉNDICES

APÉNDICE A

Plataforma implementada para el servidor privado virtual

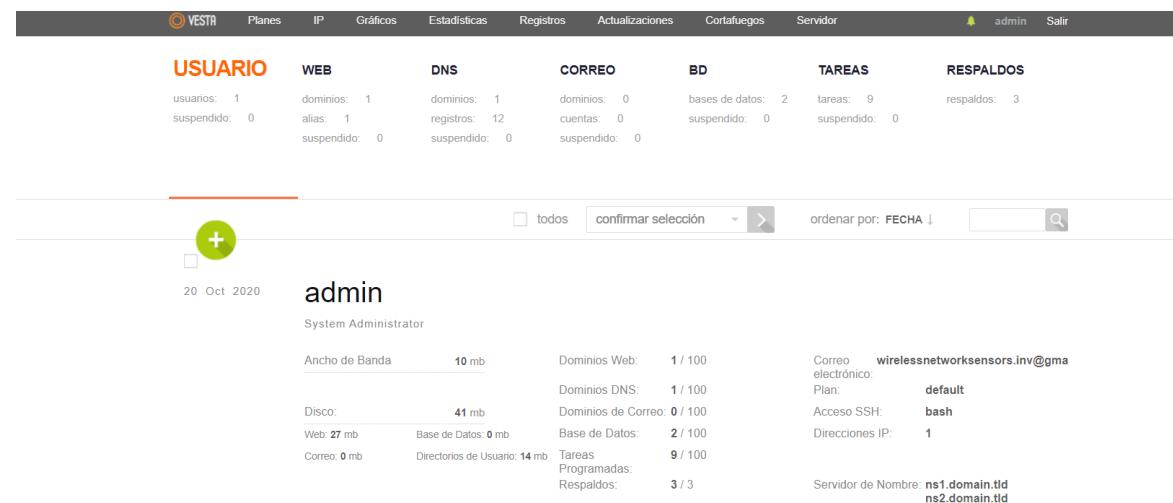


The screenshot shows the AWS EC2 console interface. On the left, there's a sidebar with navigation links for EC2, Images, and Elastic Block Store. The main area has several sections:

- Lanzar la instancia:** A wizard for launching a new instance. It says "Para comenzar, lance una instancia de Amazon EC2, que es un servidor virtual en la nube." and has a "Lanzar la instancia" button.
- Estado del servicio:** Shows the service status as "Este servicio funciona con normalidad" in the "América del Sur (São Paulo)" region.
- Eventos programados:** Shows no scheduled events.
- Zones:** Lists three availability zones: sa-east-1a, sa-east-1b, and sa-east-1c, each associated with Zone ID sae1-az1, sae1-az2, and sae1-az3 respectively.
- Migrar una máquina:** Information about CloudEndure Migration.

APÉNDICE B

Panel de control empleado para el hosting



The screenshot shows the VESTA control panel. At the top, there's a navigation bar with links for Planes, IP, Gráficos, Estadísticas, Registros, Actualizaciones, Cortafuegos, Servidor, and a user dropdown for "admin".

USUARIO

USUARIO	WEB	DNS	CORREO	BD	TAREAS	RESPALDOS
usuarios: 1 suspenido: 0	dominios: 1 alias: 1 suspendido: 0	dominios: 1 registros: 12 suspendido: 0	dominios: 0 cuentas: 0 suspendido: 0	bases de datos: 2 suspendido: 0	tareas: 9 suspendido: 0	respaldos: 3

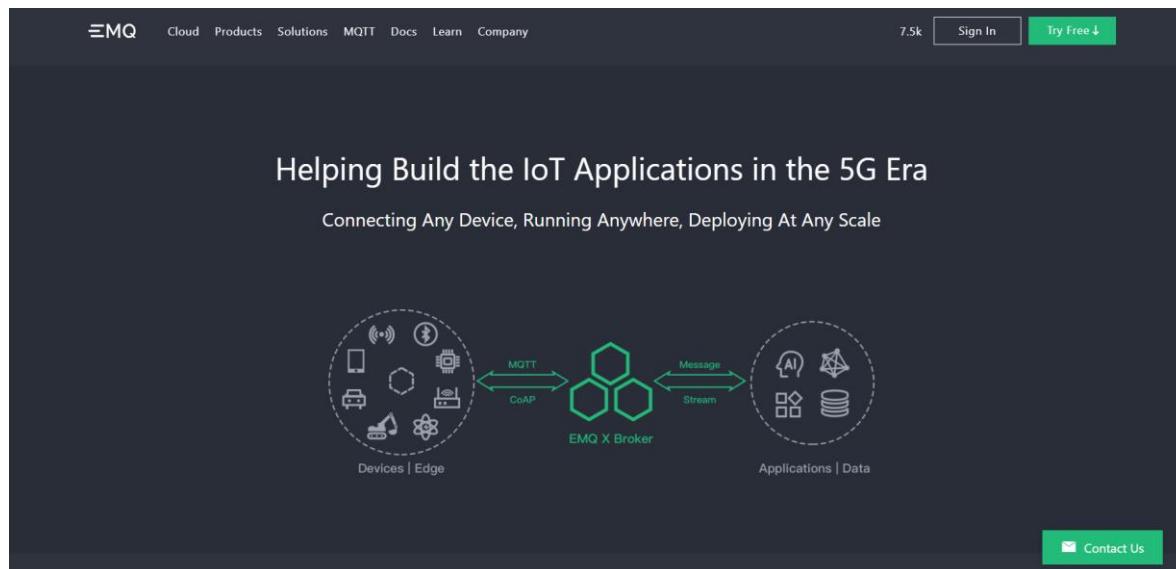
admin

System Administrator

Ancho de Banda	10 mb	Dominios Web:	1 / 100	Correo electrónico:	wirelessnetworksensors.inv@gma
Disco:	41 mb	Dominios DNS:	1 / 100	Plan:	default
Web: 27 mb	Base de Datos: 0 mb	Dominios de Correo:	0 / 100	Acceso SSH:	bash
Correo: 0 mb	Directorios de Usuario: 14 mb	Base de Datos:	2 / 100	Direcciones IP:	1
		Tareas Programadas:	9 / 100		
		Respaldos:	3 / 3	Servidor de Nombre:	ns1.domain.tld ns2.domain.tld

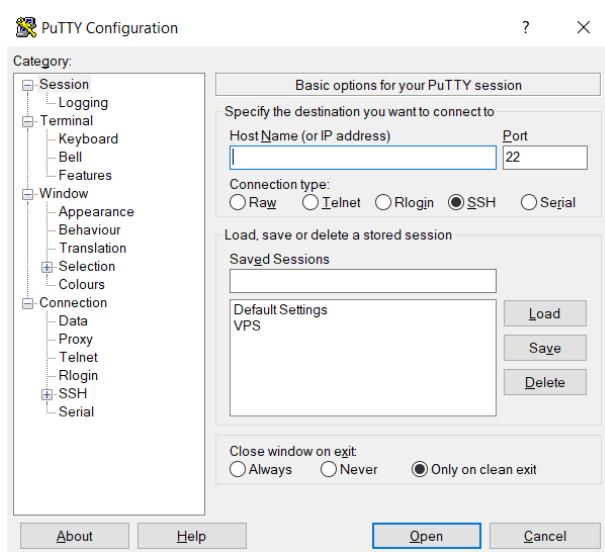
APÉNDICE C

Servicio implementado para comunicación entre clientes



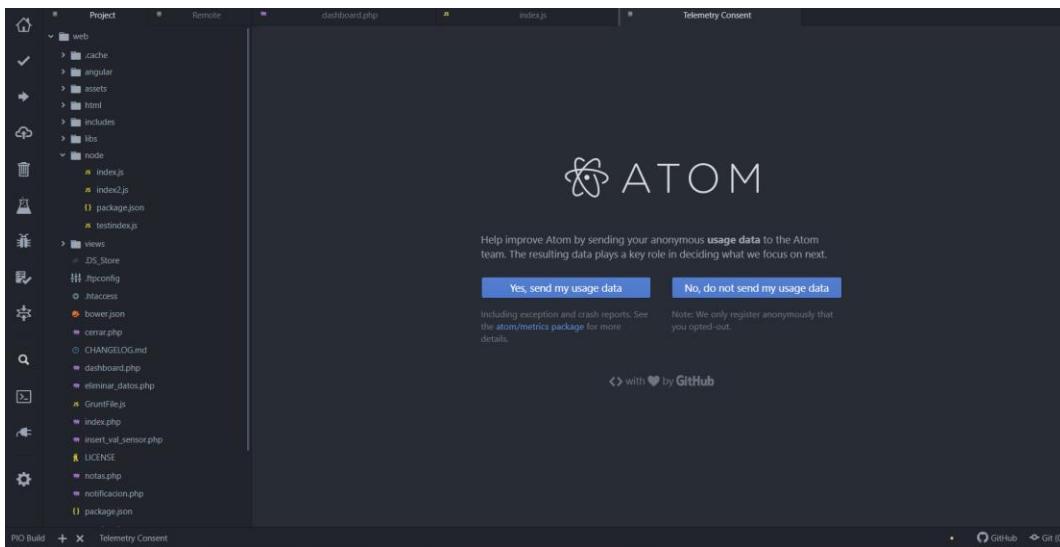
APÉNDICE D

Herramienta utilizada para conexión ssh con el servidor



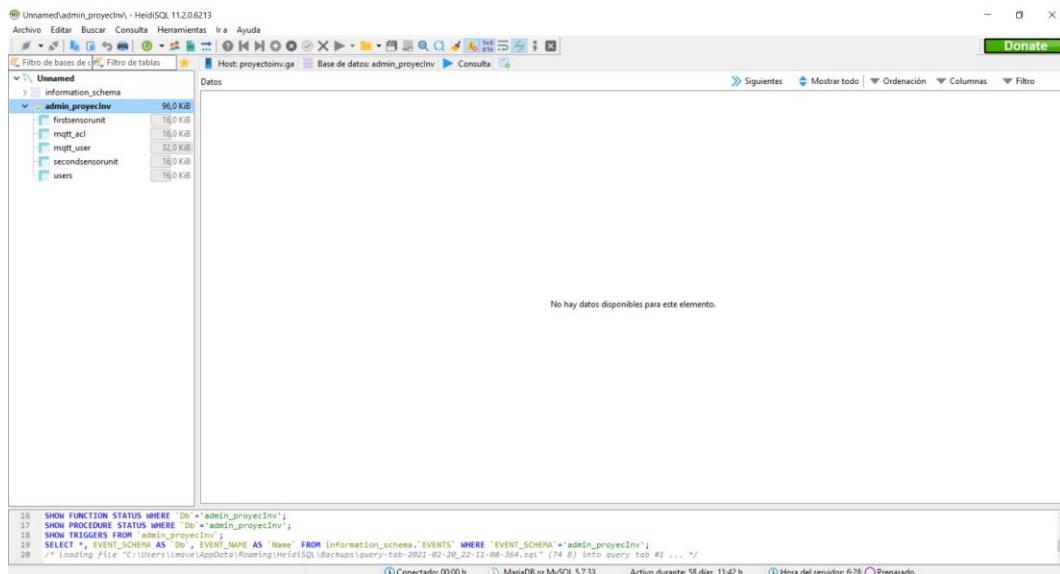
APÉNDICE E

Herramienta utilizada como editor de texto



APÉNDICE F

Herramienta implementada para administrar la base de datos MYSQL



10. ANEXOS

ANEXO A Código login de php

```
<?php require_once 'includes/conexion.php'; ?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <title>Proyecto Invernadero</title>
    <meta name="description" content="Admin, Dashboard, Bootstrap, Bootstrap 4, Angular, AngularJS" />
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, minimal-ui" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <!-- for ios 7 style, multi-resolution icon of 152x152 -->
    <meta name="apple-mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-status-barstyle" content="black-translucent">
    <link rel="apple-touch-icon" href="../assets/images/logo.png">
    <meta name="apple-mobile-web-app-title" content="Flatkit">
    <!-- for Chrome on Android, multi-resolution icon of 196x196 -->
    <meta name="mobile-web-app-capable" content="yes">
    <link rel="shortcut icon" sizes="196x196" href="../assets/images/logo.png">

    <!-- style -->
    <link rel="stylesheet" href="assets/animate.css/animate.min.css" type="text/css" />
    <link rel="stylesheet" href="assets/glyphicons/glyphicons.css" type="text/css" />
    <link rel="stylesheet" href="assets/font-awesome/css/font-awesome.min.css" type="text/css" />
    <link rel="stylesheet" href="assets/material-design-icons/material-design-icons.css" type="text/css" />

    <link rel="stylesheet" href="assets/bootstrap/dist/css/bootstrap.min.css" type="text/css" />
    <!-- build:css ../assets/styles/app.min.css -->
    <link rel="stylesheet" href="assets/styles/app.css" type="text/css" />
    <!-- endbuild -->
    <link rel="stylesheet" href="assets/styles/font.css" type="text/css" />
</head>
```

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

```

33  <body>
34    <div class="app" id="app">
35
36    <!-- ##### LAYOUT START-->
37    <div class="center-block w-xxl w-auto-xs p-y-md">
38      <div class="navbar">
39        <div class="pull-center">
40          <div ui-include="../../views/blocks/navbar.brand.html"></div>
41        </div>
42      </div>
43      <div class="p-a-md box-color r box-shadow-z1 text-color m-a">
44        <div class="m-b text-sm">
45          Sign in with your IoT Account
46        </div>
47
48
49
50      <form target="" method="post" name="form">
51        <div class="md-form-group float-label">
52          <input name="email" type="email" class="md-input" value=<?php echo $email ?> ng-model="user.email" required >
53          <label>Email</label>
54        </div>
55        <div class="md-form-group float-label">
56          <input name="password" type="password" class="md-input" ng-model="user.password" required >
57          <label>Password</label>
58        </div>
59        <button type="submit" class="btn primary btn-block p-x-md">Sign in</button>
60      </form>
61
62      <div style="color:red" class="">
63        <?php echo $msg ?>
64      </div>
65    </div>
66
67    </div>
68
69    <!-- build:js scripts/app.html.js -->
70    <!-- jquery -->
71    <script src="libs/jquery/jquery/dist/jquery.js"></script>
72    <!-- Bootstrap -->
73    <script src="libs/jquery/tether/dist/js/tether.min.js"></script>
74    <script src="libs/jquery/bootstrap/dist/js/bootstrap.js"></script>
75    <!-- core -->
76    <script src="libs/jquery/underscore/underscore-min.js"></script>
77    <script src="libs/jquery/jquery-storage-API/jquery.storageapi.min.js"></script>
78    <script src="libs/jquery/PACE/pace.min.js"></script>
79
80    <script src="html/scripts/config.lazyload.js"></script>
81
82    <script src="html/scripts/palette.js"></script>
83    <script src="html/scripts/ui-load.js"></script>
84    <script src="html/scripts/ui-jp.js"></script>
85    <script src="html/scripts/ui-include.js"></script>
86    <script src="html/scripts/ui-device.js"></script>
87    <script src="html/scripts/ui-form.js"></script>
88    <script src="html/scripts/ui-nav.js"></script>
89    <script src="html/scripts/ui-screenfull.js"></script>
90    <script src="html/scripts/ui-scroll-to.js"></script>
91    <script src="html/scripts/ui-toggle-class.js"></script>
92
93    <script src="html/scripts/app.js"></script>
94
95    <!-- ajax -->
96    <script src="libs/jquery/jquery-pjax/jquery.pjax.js"></script>
97    <script src="html/scripts/ajax.js"></script>
98
99
100
101
102
103  <!-- endbuild -->
104  </body>
105  </html>
106

```

ANEXO B Código Dashboard php

```
2 <?php require_once 'includes/encabezado.php';
3 | | //header("Refresh: 2; URL='dashboard.php'");
4 ?>
5 <body>
6 | | <!-- BARRA IZQUIERDA -->
7 <?php require_once 'includes/asside.php'; ?>
8 | | <!-- / -->
9 |
10 | | <!-- content -->
11 | | <div id="content" class="app-content box-shadow-z0" role="main">
12 |
13 <?php require_once 'includes/navbar.php'; ?>
14 |
15 | | <?php $_SESSION['vp_prom_temp'] = 0; ?>
16 <div ui-view class="app-body" id="view">
17 |
18 |
19 | | <!-- SECCION CENTRAL -->
20 <div class="padding">
21 |
22 | | <!-- VALORES EN TIEMPO REAL -->
23 <div class="p-a white lt box-shadow">
24 | | <div class="col-sm-6">
25 | | | <h4 class="mb-0 _300">UNIDAD SENSORA NÚMERO 1</h4>
26 |
27 | | </div>
28 </div>
29 |
30 | | <br>
31 |
32 | | <!--Pantallas donde se muestran los valores en tiempo real-->
33 <?php $conn = mysqli_connect("proyectoinv.ga","admin_proyecInv","ingenieriauts","admin_proyecInv");
34 | | $sql_1 = " SELECT * FROM firstsensorunit ORDER BY firstsensorunit_date DESC limit 1";
35 | | $result_1 = mysqli_query($conn, $sql_1);
36 | | $mostrar_1 = mysqli_fetch_array($result_1);
37 ?>
```

```

40  <div class="row">
41   <div class="col-xs-12 col-sm-4">
42    <div class="box p-a">
43     <div class="pull-left m-r">
44      <span class="w-48 rounded accent">
45        | <i class="fa fa-tint"></i>
46      </span>
47     </div>
48    <div class="clear">
49      <h4 class="m-0 text-lg _300"><b id="display_hums">--</b><span class="text-sm"> </span></h4>
50      <small class="text-muted">Humedad del suelo</small>
51    </div>
52   </div>
53   <div class="col-xs-6 col-sm-4">
54    <div class="box p-a">
55     <div class="pull-left m-r">
56      <span class="w-48 rounded primary">
57        | <i class="fa fa-desktop"></i>
58      </span>
59    </div>
60    <div class="clear">
61      <h4 class="m-0 text-lg _300"><b id="display_temp">--</b><span class="text-sm"> C</span></h4>
62      <small class="text-muted">Temperatura ambiente °C</small>
63    </div>
64   </div>
65   </div>
66 </div>

67   <div class="col-xs-6 col-sm-4">
68    <div class="box p-a">
69     <div class="pull-left m-r">
70      <span class="w-48 rounded warn">
71        | <i class="fa fa-tint"></i>
72      </span>
73    </div>
74    <div class="clear">
75      <h4 class="m-0 text-lg _300"><b id="display_humA">--</b><span class="text-sm"> </span></h4>
76      <small class="text-muted">Humedad del ambiente</small>
77    </div>
78   </div>
79   </div>
80   <!-- fin de Pantallas donde se muestran los valores en tiempo real-->
81
82   <!-- SWITCH1 MOTOBOMBA-->
83   <div class="row">
84    <div class="col-xs-12 col-sm-12">
85     <div class="box p-a">
86      <div class="form-group row">
87        <label class="col-sm-2 form-control-label">MOTOBOMBA 1</label>
88        <div class="col-sm-10">
89          <label class="ui-switch ui-switch-md info m-t-xs">
90            <input id="input_motobomba1" onchange="process_motobomba1()" type="checkbox" >
91            | <i></i>
92          </label>
93        </div>
94      </div>
95    </div>
96   </div>
97   </div>
98   <!-- FIN SWICHE MOTOBOMBA -->
99

```

```
101      <!--Tabla de historicos-->
102      <div class="">
103      <div class="p-a white lt box-shadow">
104      <div class="col-sm-6">
105          <h6 class="mb-0 _300">HISTORICO DE DATOS</h6>
106      </div>
107      </div>
108
109
110
111      <div class="">
112      <table class="table">
113      <thead>
114      <tr>
115          <th scope="col">id</th>
116          <th scope="col">Humedad Suelo</th>
117          <th scope="col">Temperatura ambiente</th>
118          <th scope="col">Humedad del ambiente</th>
119          <th scope="col">Fecha</th>
120      </tr>
121      </thead>
122      <tbody>
123          <?php
124
125              $sql = " SELECT * FROM firstsensorunit ORDER BY firstsensorunit_date DESC limit 10";
126              $result = mysqli_query($conn, $sql);
127
128      while($mostrar =mysqli_fetch_array($result)){
129          ?>
130          <tr>
131              <th><?php echo $mostrar['firstsensorunit_id'] ?></th>
132              <td><?php echo $mostrar['firstsensorunit_hums'] ?></td>
133              <td><?php echo $mostrar['firstsensorunit_temp'] ?></td>
134              <td><?php echo $mostrar['firstsensorunit_huma'] ?></td>
135              <td><?php echo $mostrar['firstsensorunit_date'] ?></td>
136          </tr>
137      }
138  
```

```
137 |           | <?php
138 |           |     }
139 |           |     | ?>
140 |           |     | </tbody>
141 |           |     | </table>
142 |           |     | </div>
143 |           |     <div class="">
144 |
145 |           |     <a href="registros.php" class="btn btn-primary">VER MAS REGISTROS</a>
146 |           |     <a href="eliminar_datos.php" class="btn btn-primary" onclick="return ConfirmDelete()">ELIMINAR TODOS LOS REGISTROS</a>
147 |
148 |
149 |
150 |           |     </div>
151 |           |     </div>
152 |           |     <br>
153 |           |     <hr>
154 |           |     <!--Fin tabla de historicos-->
155 |           |     <div id="notifi">
156 |
157 |           |     </div>
158 |           |     <!--GRAFICA DE HISTORICOS-->
159 |           |     <div class="row">
160 |           |       <div class="col-sm-6 col-md-4">
161 |           |         <div class="box">
162 |           |           <div class="box-body">
163 |           |             | <div id="temper" style="height: 370px; width: 100%;"></div>
164 |           |           | </div>
165 |           |         </div>
166 |           |       </div>
167 |           |       <div class="col-sm-6 col-md-4">
168 |           |         <div class="box">
169 |           |           <div class="box-body">
170 |           |             | <div id="hum_suelo" style="height: 370px; width: 100%;"></div>
171 |           |           | </div>
172 |           |         </div>
173 |           |       </div>
```

```
174     </div>
175     <div class="col-sm-6 col-md-4">
176         <div class="box">
177             <div class="box-body">
178                 <div id="hum_ambiente" style="height: 370px; width: 100%;"></div>
179             </div>
180         </div>
181     </div>
182     <!-- FIN GRAFICA DE HISTORICOS-->
183     <!-- ##### PAGE END-->
184
185
186     </div>
187
188 </div>
189 <!-- / -->
190 <br>
191 <!-- SELECTOR DE TEMAS -->
192 <?php require_once 'includes/selectortemas.php'; ?>
193 <?php require_once 'includes/footer.php'; ?>
194
195 <!-- ##### LAYOUT END-->
196
197 </div>
198 <!-- build:js scripts/app.html.js -->
199 <!-- jQuery -->
200 <script src="libs/jquery/jquery/dist/jquery.js"></script>
201 <!-- Bootstrap -->
202 <script src="libs/jquery/tether/dist/js/tether.min.js"></script>
203 <script src="libs/jquery/bootstrap/dist/js/bootstrap.js"></script>
204 <!-- core -->
205 <script src="libs/jquery/underscore/underscore-min.js"></script>
206 <script src="libs/jquery/jQuery-Storage-API/jquery.storageapi.min.js"></script>
207 <script src="libs/jquery/PACE/pace.min.js"></script>
208
209 <script src="html/scripts/config.lazyload.js"></script>
```

```
209 <script src="html/scripts/config.lazyload.js"></script>
210
211 <script src="html/scripts/palette.js"></script>
212 <script src="html/scripts/ui-load.js"></script>
213 <script src="html/scripts/ui-jp.js"></script>
214 <script src="html/scripts/ui-include.js"></script>
215 <script src="html/scripts/ui-device.js"></script>
216 <script src="html/scripts/ui-form.js"></script>
217 <script src="html/scripts/ui-nav.js"></script>
218 <script src="html/scripts/ui-screenfull.js"></script>
219 <script src="html/scripts/ui-scroll-to.js"></script>
220 <script src="html/scripts/ui-toggle-class.js"></script>
221
222 <script src="html/scripts/app.js"></script>
223 <script src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>
224 <!-- ajax -->
225 <script src="libs/jquery/jquery-pjax/jquery.pjax.js"></script>
226 <script src="html/scripts/ajax.js"></script>
227
228 <script src="https://unpkg.com/mqtt/dist/mqtt.min.js"></script>
```

```
<script type="text/javascript">
$(document).ready(function(){
    setInterval('notificacion()',10000);
});
function ConfirmDelete(){
    var respuesta = confirm("¿Estas seguro que deseas eliminar todos los datos?");

    if(respuesta == true)
    {
        return true;
    }
    else
    {
        return false;
    }
}
</script>
```

```
250 <! --GRAFICAS -->
251
252
253 <?php
254
255
256
257 $conn = mysqli_connect("proyectoinv.ga","admin_proyecInv","ingenieriauts","admin_proyecInv");
258     $sql_2 = " SELECT firstsensorunit_date,firstsensorunit_temp
259             | FROM firstsensorunit
260             | ORDER BY firstsensorunit_date DESC limit 7";
261
262     $result_2 = mysqli_query($conn, $sql_2);
263     $mostrar_2 = mysqli_fetch_array($result_2);
264     while ($row = mysqli_fetch_array($result_2)) {
265         $dataPoints [] = array("label" => $row['firstsensorunit_date'], "y" => $row['firstsensorunit_temp']);
266     }
267
268 ?>
269 <?php
270
271
272
273 $conn = mysqli_connect("proyectoinv.ga","admin_proyecInv","ingenieriauts","admin_proyecInv");
274     $sql_3 = " SELECT firstsensorunit_date,firstsensorunit_hums
275             | FROM firstsensorunit
276             | ORDER BY firstsensorunit_date DESC limit 7";
277
278     $result_3 = mysqli_query($conn, $sql_3);
279     $mostrar_3 = mysqli_fetch_array($result_3);
280     while ($row = mysqli_fetch_array($result_3)) {
281         $dataPoints2 [] = array("label" => $row['firstsensorunit_date'], "y" => $row['firstsensorunit_hums']);
282     }
283
284 ?>
285 <?php
```

```
289 $conn = mysqli_connect("proyectoInv.ga", "admin_proyecInv", "ingenieriauts", "admin_proyecInv");
290     $sql_4 = " SELECT firstsensorunit_date,firstsensorunit_humA
291             | FROM firstsensorunit
292             | ORDER BY firstsensorunit_date DESC limit 7";
293
294     $result_4 = mysqli_query($conn, $sql_4);
295     $mostrar_4 = mysqli_fetch_array($result_4);
296     while ($row = mysqli_fetch_array($result_4)) {
297         $dataPoints3 [] = array("label" => $row['firstsensorunit_date'], "y" => $row['firstsensorunit_humA']);
298     }
299
300 ?>
301 <!--GRAFICA TEMPERATURA -->
302 <script>
303 window.onload = function () {
304
305 var chart = new CanvasJS.Chart("temper", {
306     animationEnabled: true,
307     exportEnabled: true,
308     title:{ 
309         text: "Grafica de temperatura ambiente"
310     },
311     axisY: { 
312         title: "Temperatura",
313         valueFormatString: "#",
314         suffix: "°C"
315     },
316     data: [{
317         type: "spline",
318         markerSize: 5,
319         xValueFormatString: "YYYY",
320         yValueFormatString: "#",
321         xValueType: "dateTime",
322         dataPoints: <?php echo json_encode($dataPoints, JSON_NUMERIC_CHECK); ?>
323     }]
324 });
325 chart.render();
326
327 var chart = new CanvasJS.Chart("hum_suelo", {
328     animationEnabled: true,
329     exportEnabled: true,
330     title:{ 
331         text: "Grafica de humedad del suelo"
332     },
333     axisY: { 
334         title: "Humedad",
335         valueFormatString: "#",
336         suffix: "%"
337     },
338     data: [{
339         type: "spline",
340         markerSize: 5,
341         xValueFormatString: "YYYY",
342         yValueFormatString: "#",
343         xValueType: "dateTime",
344         dataPoints: <?php echo json_encode($dataPoints2, JSON_NUMERIC_CHECK); ?>
345     }]
346 });
347
```

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 1.0

```
347 chart.render();
348 var chart = new CanvasJS.Chart("hum_ambiente", {
349     animationEnabled: true,
350     exportEnabled: true,
351     title:{  
352         text: "Grafica de humedad del ambiente"  
353     },
354     axisY: {  
355         title: "Humedad",
356         valueFormatString: "#",
357         suffix: "%"  
},
358     data: [<{
359         type: "spline",
360         markerSize: 5,
361         xValueFormatString: "YYYY",
362         yValueFormatString: "#",
363         xValueType: "dateTime",
364         dataPoints: <?php echo json_encode($dataPoints3, JSON_NUMERIC_CHECK); ?>
365     }]
366 });
367 });
368
369 chart.render();
370 }
371 </script>
372
373 <script type="text/javascript">
```

```

377 function update_values(humS, temp, humA){
378     $("#display_humS").html(humS);
379     $("#display_temp").html(temp);
380     $("#display_humA").html(humA);
381     var result = document.getElementById("notifi");
382     var xmlhttp;
383     if (window.XMLHttpRequest) {
384         xmlhttp = new XMLHttpRequest();
385     } else {
386         xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
387     }
388     xmlhttp.onreadystatechange = function () {
389         if (xmlhttp.readyState === 4 && xmlhttp.status === 200) {
390             result.innerHTML = xmlhttp.responseText;
391         }
392     }
393     //!!!!!!CON ESTO SE ENVIAN VALORES A LA BASE DE DATOS!!!!!!
394     //xmlhttp.open("GET", "https://proyectoinv.ga/insert_val_sensor.php?tem="+temp+"&hum_amb="+humA+"&hum_sue="+humS, true);
395     xmlhttp.send();
396 }
397
398 function process_msg(topic, message){
399     if(topic == "values"){
400         var msg = message.toString();
401         var sp = msg.split(",");
402         var humS = sp[0];
403         var temp = sp[1];
404         var humA = sp[2];
405         update_values(humS,temp,humA);
406     }
407 }
408
409 }

411 function process_motobomba1(){
412     if($('#input_motobomba1').is(":checked")){
413         console.log("Encendido");
414
415         client.publish('motobomba1', 'on', (error) => {
416             console.log(error || 'Mensaje enviado!!!');
417         })
418
419     }else{
420         console.log("Apagado");
421
422         client.publish('motobomba1', 'off', (error) => {
423             console.log(error || 'Mensaje enviado!!!');
424         })
425     }
426 }
427

```

```

431  /* -----CONEXION----- */
432
433  // connect options
434  const options = {
435    connectTimeout: 4000,
436    // Authentication
437    clientId: 'emqx',
438    username: 'web_client',
439    password: '54321',
440    keepalive: 60,
441    clean: true,
442  }
443
444  var connected = false;
445
446  // WebSocket connect url
447  const WebSocket_URL = 'wss://proyectoinv.ga:8094/mqtt'
448
449  const client = mqtt.connect(WebSocket_URL, options)
450
451
452  client.on('connect', () => {
453    console.log('Mqtt conectado por ws! Exito!')
454
455    client.subscribe('values', { qos: 0}, (error) => {
456      if(!error){
457        console.log('Subscripción exitosa!')
458      }else{
459        console.log('Subscripción fallida!')
460      }
461    })
462    // publish(topic, payload, options/callback)
463    client.publish('invernadero', 'esto es un verdadero éxito', (error) => {
464      console.log(error || 'Mensaje enviado!!!');
465    })
466  })
467
468  client.on('message', (topic, message) => {
469  console.log('Mensaje recibido bajo tópico: ', topic, ' -> ', message.toString());
470  process_msg(topic, message);
471  })
472
473  client.on('reconnect', (error) => {
474    console.log('Error al reconnectar', error)
475  })
476
477  client.on('error', (error) => {
478    console.log('Error de conexión:', error)
479  })

```

```
</script>

<script type="text/javascript">
function notificacion() {
    var result = document.getElementById("alerta");
    var xmlhttp;
        if (window.XMLHttpRequest) {
            xmlhttp = new XMLHttpRequest();
        } else {
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        xmlhttp.onreadystatechange = function () {
            if (xmlhttp.readyState === 4 && xmlhttp.status === 200) {
                result.innerHTML = xmlhttp.responseText;
            }
        }
        xmlhttp.open("GET", "https://proyectoinv.ga/notificacion.php?var=1", true);
        xmlhttp.send();
}

function cerrar_notifi(){
    var btn = document.getElementById("alerta");
    btn.style.display = "none"
}
</script>

<!-- endbuild -->
</body>
</html>
```

ANEXO C Código dashboard2.php

```
<?php require_once 'includes/encabezado.php';
|   //header("Refresh: 2; URL='dashboard.php'");
?>
<body>
|   <!-- BARRA IZQUIERDA -->
<?php require_once 'includes/asside.php'; ?>
|   <!-- / -->

|   <!-- content -->
<div id="content" class="app-content box-shadow-z0" role="main">

<?php require_once 'includes/navbar.php'; ?>

<?php $_SESSION['vp_prom_temp'] = 0; ?>
<div ui-view class="app-body" id="view">

|   <!-- SECCION CENTRAL -->
<div class="padding">

|       <!-- VALORES EN TIEMPO REAL -->
<div class="p-a white lt box-shadow">
|           <div class="col-sm-6">
|               <h4 class="mb-0 _300">UNIDAD SENSORA NÚMERO 2</h4>

|           </div>
|       </div>

|       <br>

|       <!--Pantallas donde se muestran los valores en tiempo real-->
<?php $conn = mysqli_connect("proyectoinv.ga","admin_proyecInv","ingenieriauts","admin_proyecInv");
?>
```

```

36 <div class="row">
37   <div class="col-xs-6 col-sm-4">
38     <div class="box p-a">
39       <div class="pull-left m-r">
40         <span class="w-48 rounded primary">
41           | <i class="fa fa-desktop"></i>
42         </span>
43       </div>
44       <div class="clear">
45         <h4 class="m-0 text-lg _300"><b id="display_temp2">--</b><span class="text-sm"> C</span></h4>
46         <small class="text-muted">Temperatura ambiente °C</small>
47       </div>
48     </div>
49   </div>
50   <div class="col-xs-6 col-sm-4">
51     <div class="box p-a">
52       <div class="pull-left m-r">
53         <span class="w-48 rounded warn">
54           | <i class="fa fa-tint"></i>
55         </span>
56       </div>
57       <div class="clear">
58         <h4 class="m-0 text-lg _300"><b id="display_humA2">--</b><span class="text-sm"> </span></h4>
59         <small class="text-muted">Humedad del ambiente</small>
60       </div>
61     </div>
62   </div>
63   <!--fin de Pantallas donde se muestran los valores en tiempo real-->
64   <hr>
65   <!-- Buscar Registros por Fecha-->
66
67
68   <div class="">
69     <div class="p-a white lt box-shadow">
70       <div class="col-sm-6">
71         <h6 class="mb-0 _300">BUSCAR DATOS POR FECHA</h6>
72       </div>
73     </div>
74   </div>
75   <div class="box-body">
76     <form role="form" method="POST" action="search_data.php">
77       <div class="form-group">
78         <label for="fechainicial">Fecha Inicial</label>
79         <input type="text" class="form-control" id="fechainicial" name="fechainicial" placeholder="Ingresar Fecha en su respectiva casilla" required>
80       </div>
81       <div class="form-group">
82         <label for="fechafinal">Fecha Final </label>
83         <input type="text" class="form-control" id="fechafinal" name="fechafinal" placeholder="Ingresar Fecha en su respectiva casilla" required>
84       </div>
85       <button type="submit" class="btn white m-b" action="search_data.php">Buscar</button>
86     </form>
87   </div>
88
89   <!--Tabla de historicos-->
90   <div class="">
91     <div class="p-a white lt box-shadow">
92       <div class="col-sm-6">
93         <h6 class="mb-0 _300">HISTORICO DE DATOS</h6>
94       </div>
95     </div>
96   </div>
97
98

```

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 1.0

```
99 <div class="">
100 <table class="table">
101   <thead>
102     <tr>
103       <th scope="col">id</th>
104       <th scope="col">Temperatura ambiente</th>
105       <th scope="col">Humedad del ambiente</th>
106       <th scope="col">Fecha</th>
107     </tr>
108   </thead>
109   <tbody>
110     <?php
111       $conn = mysqli_connect("proyectoinv.ga","admin_proyecInv","ingenieriauts","admin_proyecInv");
112       $sql = " SELECT * FROM secondsensorunit ORDER BY secondsensorunit_date DESC";
113       $result = mysqli_query($conn, $sql);
114
115       while($mostrar =mysqli_fetch_array($result)){
116         <?>
117         <tr>
118           <th><?php echo $mostrar['secondsensorunit_id'] ?></th>
119           <td><?php echo $mostrar['secondsensorunit_temp'] ?></td>
120           <td><?php echo $mostrar['secondsensorunit_humA'] ?></td>
121           <td><?php echo $mostrar['secondsensorunit_date'] ?></td>
122         </tr>
123         <?php
124       }
125       <?>
126     </tbody>
127   </table>
128 </div>
129 </div>
130 <br>
131 <hr>
132 <!--Fin tabla de historicos-->
133 <">
```

```
135 | | | | | <!-- SELECTOR DE TEMAS -->
136 <?php require_once 'includes/selectortemas.php'; ?>
137 <?php require_once 'includes/footer.php'; ?>
138
139
140 </div>
141
142
143 <!-- build:js scripts/app.html.js -->
144 <!-- jQuery -->
145 <script src="libs/jquery/jquery/dist/jquery.js"></script>
146 <!-- Bootstrap -->
147 <script src="libs/jquery/tether/dist/js/tether.min.js"></script>
148 <script src="libs/jquery/bootstrap/dist/js/bootstrap.js"></script>
149 <!-- core -->
150 <script src="libs/jquery/underscore/underscore-min.js"></script>
151 <script src="libs/jquery/jQuery-Storage-API/jquery.storageapi.min.js"></script>
152 <script src="libs/jquery/PACE/pace.min.js"></script>
153
154 <script src="html/scripts/config.lazyload.js"></script>
155
156 <script src="html/scripts/palette.js"></script>
157 <script src="html/scripts/ui-load.js"></script>
158 <script src="html/scripts/ui-jp.js"></script>
159 <script src="html/scripts/ui-include.js"></script>
160 <script src="html/scripts/ui-device.js"></script>
161 <script src="html/scripts/ui-form.js"></script>
162 <script src="html/scripts/ui-nav.js"></script>
163 <script src="html/scripts/ui-screenfull.js"></script>
164 <script src="html/scripts/ui-scroll-to.js"></script>
165 <script src="html/scripts/ui-toggle-class.js"></script>
166
167 <script src="html/scripts/app.js"></script>
168 <script src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>
169 <!-- ajax -->
170 <script src="libs/jquery/jquery-pjax/jquery.pjax.js"></script>
171 <script src="html/scripts/ajax.js"></script>
```

```

173   <script src="https://unpkg.com/mqtt/dist/mqtt.min.js"></script>
174
175   <script type="text/javascript">
176
177   /* -----PROCESOS----- */
178
179   function update_values1(temp, humA){
180     $("#display_temp2").html(temp);
181     $("#display_humA2").html(humA);
182
183     var result = document.getElementById("notifi");
184     var xmlhttp;
185     if (window.XMLHttpRequest) {
186       xmlhttp = new XMLHttpRequest();
187     } else {
188       xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
189     }
190     xmlhttp.onreadystatechange = function () {
191       if (xmlhttp.readyState === 4 && xmlhttp.status === 200) {
192         result.innerHTML = xmlhttp.responseText;
193       }
194     }
195     // !!!!!!!CON ESTO SE ENVIAN VALORES A LA BASE DE DATOS!!!!!!
196     //xmlhttp.open("GET", "https://proyectoinv.ga/insert_val_sensor.php?tem="+temp+"&hum_amb="+humA+"&hum_sue="+humS, true);
197     xmlhttp.send();
198   }
199
200   function process_msg(topic, message){
201     if(topic == "vlesp2"){
202       var msg = message.toString();
203       var sp = msg.split(",");
204       var temp = sp[0];
205       var humA = sp[1];
206       update_values1(temp,humA);
207     }
208   }
209
210
211
212   /* -----CONEXION----- */
213
214   // connect options
215   const options = {
216     connectTimeout: 4000,
217     // Authentication
218     clientId: 'emqx',
219     username: 'web_client',
220     password: '54321',
221     keepalive: 60,
222     clean: true,
223   }
224
225   var connected = false;
226
227   // WebSocket connect url
228   const WebSocket_URL = 'wss://proyectoinv.ga:8094/mqtt'
229
230   const client = mqtt.connect(WebSocket_URL, options)
231

```

```
232
233   client.on('connect', () => {
234     console.log('Mqtt conectado por WS! Exito!')
235
236     client.subscribe('vlesp2', { qos: 0}, (error) => {
237       if(!error){
238         console.log('Subscripción exitosa!')
239       }else{
240         console.log('Subscripción fallida!')
241       }
242     })
243     // publish(topic, payload, options/callback)
244     client.publish('invernadero', 'esto es un verdadero éxito', (error) => {
245       console.log(error || 'Mensaje enviado!!!');
246     })
247   })
248
249   client.on('message', (topic, message) => {
250     console.log('Mensaje recibido bajo tópico: ', topic, ' -> ', message.toString());
251     process_msg(topic, message);
252   })
253
254   client.on('reconnect', (error) => {
255     console.log('Error al reconectar', error)
256   })
257
258   client.on('error', (error) => {
259     console.log('Error de conexión:', error)
260   })
261
262   </script>
263
264
265   </body>
266   </html>
267
```

ANEXO D Código del total de registros php

```

2   <?php require_once 'includes/encabezado.php';
3   | | | //header("Refresh: 2; URL='registros.php'");
4   ?>
5   <body>
6   | | | <!-- BARRA IZQUIERDA -->
7   <?php require_once 'includes/asside.php'; ?>
8   | | | <!-- / -->
9
10  | | | <!-- content -->
11  | | | <div id="content" class="app-content box-shadow-z0" role="main">
12
13  | | | | <?php require_once 'includes/navbar.php'; ?>
14
15
16  | | | | <div ui-view class="app-body" id="view">
17
18
19  | | | | | <!-- SECCION CENTRAL -->
20  | | | | | <div class="padding">
21
22  | | | | | | <!-- VALORES EN TIEMPO REAL -->
23  | | | | | | <div class="p-a white lt box-shadow">
24  | | | | | | | <div class="col-sm-6">
25  | | | | | | | | <h4 class="mb-0 _300">UNIDAD SENSORA NÚMERO 1</h4>
26
27  | | | | | | | </div>
28  | | | | | </div>
29
30  | | | | | | <br>
31  | | | | | | | <!-- Buscar Registros por Fecha-->
32
33  | | | | | <div class="">
34  | | | | | | <div class="p-a white lt box-shadow">
35  | | | | | | | <div class="col-sm-6">
36  | | | | | | | | <h6 class="mb-0 _300">BUSCAR DATOS POR FECHA</h6>
37
38  | | | | | </div>
39  | | | </div>
40  | | | <div class="box-body">
41  | | | | <form role="form" method="POST" action="search_data.php">
42  | | | | | <div class="form-group">
43  | | | | | | <label for="fechainicial">Fecha Inicial</label>
44  | | | | | | <input type="text" class="form-control" id="fechainicial" name="fechainicial" placeholder="Ingresar Fecha en su respectivo formato AAAA-MM-DD" required="required"/>
45
46  | | | | | <div class="form-group">
47  | | | | | | <label for="fechafinal">Fecha Final </label>
48  | | | | | | <input type="text" class="form-control" id="fechafinal" name="fechafinal" placeholder="Ingresar Fecha en su respectivo formato AAAA-MM-DD" required="required"/>
49
50  | | | | | <button type="submit" class="btn white m-b" action="search_data.php">Buscar</button>
51
52  | | | </div>
53

```

```

56      | <!--Tabla de historicos-->
57      | <div class="">
58      | <div class="p-a white lt box-shadow">
59      |   <div class="col-sm-6">
60      |     <h6 class="mb-0 _300">HISTORICO DE DATOS</h6>
61      |   </div>
62      | </div>
63
64
65
66      <div class="">
67      <table class="table">
68        <thead>
69          <tr>
70            <th scope="col">id</th>
71            <th scope="col">humedad suelo</th>
72            <th scope="col">Temperatura ambiente</th>
73            <th scope="col">humedad del ambiente</th>
74            <th scope="col">Fecha</th>
75          </tr>
76        </thead>
77        <tbody>
78          <?php
79            $conn = mysqli_connect("proyectoinv.ga", "admin_proyecInv", "ingenieriauts", "admin_proyecInv");
80            $sql = " SELECT * FROM firstsensorunit ORDER BY firstsensorunit_date DESC";
81            $result = mysqli_query($conn, $sql);
82
83            while($mostrar =mysqli_fetch_array($result)){
84              ?>
85              <tr>
86                <th><?php echo $mostrar['firstsensorunit_id'] ?></th>
87                <td><?php echo $mostrar['firstsensorunit_humS'] ?></td>
88                <td><?php echo $mostrar['firstsensorunit_temp'] ?></td>
89                <td><?php echo $mostrar['firstsensorunit_humA'] ?></td>
90                <td><?php echo $mostrar['firstsensorunit_date'] ?></td>
91              </tr>
92
93      <?php
94      }
95      ?>
96      </tbody>
97    </table>
98  </div>
99  <br>
100 <hr>
101 <!--Fin tabla de historicos-->
102 <!-- ##### PAGE END-->
103 </div>
104 </div>
105 <!-- / -->
106 <!-- SELECTOR DE TEMAS -->
107 <?php require_once 'includes/selectortemas.php'; ?>
108 <?php require_once 'includes/footer.php'; ?>
109 <!-- ##### LAYOUT END-->

```

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 1.0

```
110
111  </div>
112  <!-- build:js scripts/app.html.js -->
113  <!-- jQuery -->
114  <script src="libs/jquery/jquery/dist/jquery.js"></script>
115  <!-- Bootstrap -->
116  <script src="libs/jquery/tether/dist/js/tether.min.js"></script>
117  <script src="libs/jquery/bootstrap/dist/js/bootstrap.js"></script>
118  <!-- core -->
119  <script src="libs/jquery/underscore/underscore-min.js"></script>
120  <script src="libs/jquery/jQuery-Storage-API/jquery.storageapi.min.js"></script>
121  <script src="libs/jquery/PACE/pace.min.js"></script>
122
123  <script src="html/scripts/config.lazyload.js"></script>
124
125  <script src="html/scripts/palette.js"></script>
126  <script src="html/scripts/ui-load.js"></script>
127  <script src="html/scripts/ui-jp.js"></script>
128  <script src="html/scripts/ui-include.js"></script>
129  <script src="html/scripts/ui-device.js"></script>
130  <script src="html/scripts/ui-form.js"></script>
131  <script src="html/scripts/ui-nav.js"></script>
132  <script src="html/scripts/ui-screenfull.js"></script>
133  <script src="html/scripts/ui-scroll-to.js"></script>
134  <script src="html/scripts/ui-toggle-class.js"></script>
135  <script src="html/scripts/app.js"></script>
136  <!-- ajax -->
137  <script src="libs/jquery/jquery-pjax/jquery.pjax.js"></script>
138  <script src="html/scripts/ajax.js"></script>
139
140  <script src="https://unpkg.com/mqtt/dist/mqtt.min.js"></script>
141  <script type="text/javascript">
142  [/script]
143  <!-- endbuild -->
144  </body>
145  </html>
```

ANEXO E Includes de asside, navbar, verifyConexion, footer, conexión y encabezado

```
asside.php ×  
web > includes > asside.php > ...  
1  <!-- aside -->  
2  <div id="aside" class="app-aside modal nav-dropdown">  
3  | <!-- fluid app aside -->  
4  | <div class="left navside dark dk" data-layout="column">  
5  | | <div class="navbar no-radius">  
6  | | | <!-- brand -->  
7  | | | <a class="navbar-brand">  
8  | | | | <div ui-include="'assets/images/logo.svg'"></div>  
9  | | | |   
10 | | | | <span class="hidden-folded inline">Greenhouse</span>  
11 | | | </a>  
12 | | | <!-- / brand -->  
13 | | </div>  
14 | | <div class="hide-scroll" data-flex>  
15 | | | <nav class="scroll nav-light">  
16 | | | |  
17 | | | | <ul class="nav" ui-nav>  
18 | | | | | <li class="nav-header hidden-folded">  
19 | | | | | | <small class="text-muted">Main</small>  
20 | | | | | </li>  
21 | | | |  
22 | | | <li>  
23 | | | | <a href="dashboard.php" >  
24 | | | | | <span class="nav-icon">  
25 | | | | | | <i class="fa fa-building-o"></i>  
26 | | | | | </span>  
27 | | | | <span class="nav-text">Principal</span>  
28 | | | | </a>  
29 | | | </li>  
30 | | |  
31 | | <li>  
32 | | | <a href="registros.php" >  
33 | | | | <span class="nav-icon">  
34 | | | | | <i class="fa fa-table"></i>  
35 | | | | </span>  
36 | | | | <span class="nav-text">Total registros</span>  
37 | | | | </a>  
38 | | </li>
```

```
38 |         </li>
39 |
40 |     <li>
41 |         <a href="dashboard2.php" >
42 |             <span class="nav-icon">
43 |                 <i class="fa fa-table"></i>
44 |             </span>
45 |             <span class="nav-text">Unidad Sensora #2</span>
46 |         </a>
47 |     </li>
48 | </ul>
49 | </nav>
50 | </div>
51 | <div class="b-t">
52 |     <div class="nav-fold">
53 |         <a href="profile.html">
54 |             <span class="pull-left">
55 |                 
56 |             </span>
57 |             <span class="clear hidden-folded p-x">
58 |                 <span class="block _500">Admin</span>
59 |                 <small class="block text-muted"><i class="fa fa-circle text-success m-r-sm"></i>online</small>
60 |             </span>
61 |         </a>
62 |     </div>
63 | </div>
64 | </div>
65 | </div>
66 |
```

```
● conexion.php ●
web > includes > ● conexion.php > ...
1  <?php
2  session_start();
3  $_SESSION['logged'] = false;
4  $msg="";
5  $email="";
6  if(isset($_POST['email']) && isset($_POST['password'])) {
7
8      if ($_POST['email']==""){
9          $msg.= "Debe ingresar un email <br>";
10     }else if ($_POST['password']=="") {
11         $msg.= "Debe ingresar la clave <br>";
12     }else [
13         $email = strip_tags($_POST['email']);
14         $password= strip_tags($_POST['password']);
15         //momento de conectarnos a db
16         $conn = mysqli_connect("proyectoinv.ga","admin_proyecInv","ingenieriauts","admin_proyecInv");
17         if ($conn==false){
18             echo "Hubo un problema al conectarse a María DB";
19             die();
20         }
21         $result = $conn->query("SELECT * FROM `users` WHERE `users_email` = '".$email."' AND `users_password` = '".$password."' ");
22         $users = $result->fetch_all(MYSQLI_ASSOC);
23         //cuento cuantos elementos tiene $tabla,
24         $count = count($users);
25         if ($count == 1){
26             //cargo datos del usuario en variables de sesión
27             $_SESSION['user_id'] = $users[0]['users_id'];
28             $_SESSION['users_email'] = $users[0]['users_email'];
29             $msg .= "|Conexión exitosa!";
30             $_SESSION['logged'] = true;
31             echo "<meta http-equiv='refresh' content='2; url=dashboard.php'>";
32         }else{
33             $msg .= "Acceso denegado!!!";
34             $_SESSION['logged'] = false;
35         }
36     }
37 }
38 }
```

```
encabezado.php X
web > includes > encabezado.php > ...
1 <?php require_once 'verifyConexion.php'; ?>
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5   <meta charset="utf-8" />
6   <title>Invernaderos</title>
7   <meta name="description" content="Admin, Dashboard, Bootstrap, Bootstrap 4, Angular, AngularJS" />
8   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, minimal-ui" />
9   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
10
11
12  <!-- for iOS 7 style, multi-resolution icon of 152x152 -->
13  <meta name="apple-mobile-web-app-capable" content="yes" />
14  <meta name="apple-mobile-web-app-status-bar-style" content="black-translucent" />
15  <link rel="apple-touch-icon" href="assets/images/logo.png" />
16  <meta name="apple-mobile-web-app-title" content="Flatkit" />
17  <!-- for Chrome on Android, multi-resolution icon of 196x196 -->
18  <meta name="mobile-web-app-capable" content="yes" />
19  <link rel="shortcut icon" sizes="196x196" href="assets/images/logo.png" />
20
21  <!-- style -->
22  <link rel="stylesheet" href="assets/animate.css/animate.min.css" type="text/css" />
23  <link rel="stylesheet" href="assets/glyphicons/glyphicons.css" type="text/css" />
24  <link rel="stylesheet" href="assets/font-awesome/css/font-awesome.min.css" type="text/css" />
25  <link rel="stylesheet" href="assets/material-design-icons/material-design-icons.css" type="text/css" />
26
27  <link rel="stylesheet" href="assets/bootstrap/dist/css/bootstrap.min.css" type="text/css" />
28  <!-- build:css assets/styles/app.min.css -->
29  <link rel="stylesheet" href="assets/styles/app.css" type="text/css" />
30  <!-- endbuild -->
31  <link rel="stylesheet" href="assets/styles/font.css" type="text/css" />
32 </head>
33
```

```
footer.php X
web > includes > footer.php > div.app-footer
1 <div class="app-footer" />
2   <div class="p-2 text-xs">
3     <div class="pull-right text-muted py-1">
4       &copy; Copyright <strong>Flatkit</strong> <span class="hidden-xs-down">- Built with Love v1.1.3</span>
5       <a ui-scroll-to="content"><i class="fa fa-long-arrow-up p-x-sm" /></i></a>
6     </div>
7     <div class="nav">
8       <a class="nav-link" href="#">About</a>
9     </div>
10    </div>
11  </div>
12
```

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 1.0

```
(navbar.php) ●  
web > includes > navbar.php > div.app-header.white.box-shadow > div.navbar.navbar-toggleable-sm.flex-row.align-items-center  
1   <div class="app-header white box-shadow">  
2     <div class="navbar navbar-toggleable-sm flex-row align-items-center">  
3       <!-- Open side - Navigation on mobile -->  
4       <a data-toggle="modal" data-target="#aside" class="hidden-lg-up mr-3">  
5         <i class="material-icons">#xe5d2;</i>  
6       </a>  
7       <!-- / -->  
8       <h3>  
9         ¡Controla tu invernadero!  
10      </h3>  
11      <!-- Page title - Bind to $state's title -->  
12      <div class="mb-0 h5 no-wrap" ng-bind="$state.current.data.title" id="pageTitle"></div>  
13  
14      <ul class="nav navbar-nav ml-auto flex-row">  
15        <li class="nav-item dropdown pos-stc-xs">  
16          <div ui-include="'views/blocks/dropdown.notification.html'"></div>  
17        </li>  
18        <li class="nav-item dropdown">  
19          <a class="nav-link p-0 clear" href="#" data-toggle="dropdown">  
20            <span class="avatar w-32">  
21                
22              <i class="on b-white bottom"></i>  
23            </span>  
24          </a>  
25          <div ui-include="'includes/dropdown.user.php'"></div>  
26        </li>  
27        <li class="nav-item hidden-md-up">  
28          <a class="nav-link pl-2" data-toggle="collapse" data-target="#collapse">  
29            <i class="material-icons">#xe5d4;</i>  
30          </a>  
31        </li>  
32      </ul>  
33      <!-- / navbar right -->  
34    </div>  
35  </div>  
36
```

```
verify Conexion.php X

web > includes > verify Conexion.php > ...
1  <?php
2  session_start();
3  $logged = $_SESSION['logged'];
4
5  if(!$logged){
6      session_destroy();
7      header('Location: https://proyectoinv.ga/');
8      die();
9  }
10
11 ?>
12
```

ANEXO F Código php para eliminar registros de la base de datos.

```
eliminar_datos.php X

web > eliminar_datos.php > ...
1  <?php
2  //require_once 'includes/conexion.php';
3  $conn = mysqli_connect("proyectoinv.ga","admin_proyecInv","ingenieriauts","admin_proyecInv");
4
5  $sentencia="DELETE FROM firstsensorunit";
6  mysqli_query($conn,$sentencia);
7
8  header("Location: dashboard.php");
9
10 ?>
11
12 <script type="text/javascript">
13 | alert("Datos Eliminados Satisfactoriamente");
14
15 </script>
```

ANEXO G Código del servicio node.js

```
JS index.js  X
web > node > JS index.js > ...
1  var nodemailer = require('nodemailer');
2
3
4  var mysql = require('mysql');
5  var mqtt = require('mqtt');
6
7  var humS = 0;
8  var temp = 0;
9  var humA = 0;
10 var temp1 = 0;
11 var humA1 = 0;
12
13 var con = mysql.createConnection({
14   host: "proyectoinv.ga",
15   user: "admin_proyecInv",
16   password: "ingenieriauts",
17   database: "admin_proyecInv"
18 });
19
20 con.connect(function(err){
21   if (err) throw err;
22   console.log("Conexión a Mysql Exitosa!!!")
23   var query = "SELECT * FROM users WHERE 1";
24   con.query(query, function (err, result, fields){
25     if (err) throw err;
26     if (result.length > 0){
27       console.log(result);
28     }
29   });
30 });
31
32 // Credenciales Mqtt
33
```

```

34  //CREDENCIALES MQTT
35  var options = {
36    port: 1883,
37    host: 'proyectoinv.ga',
38    clientId: 'acces_control_server_' + Math.round(Math.random() * (0- 10000) * -1) ,
39    username: 'web_client',
40    password: '54321',
41    keepalive: 60,
42    reconnectPeriod: 1000,
43    protocolId: 'MQIsdp',
44    protocolVersion: 3,
45    clean: true,
46    encoding: 'utf8'
47  };
48
49  var client = mqtt.connect("mqtt://proyectoinv.ga", options);
50
51  //SE REALIZA LA CONEXION
52  client.on('connect', function () {
53    console.log("Conexión MQTT Exitosa!");
54  client.subscribe('+/#', function (err) {
55    console.log("Subscripción exitosa!")
56  });
57 });
58
59  // CUANDO SE RECIBEN MENSAJES
60  client.on('message', function (topic, message){
61    console.log("Mensaje Recibido desde -> "+ topic + " Mensaje -> "+ message.toString());
62    if (topic == "values"){
63      var msg = message.toString();
64      var sp = msg.split(",");
65      humS = sp[0];
66      temp = sp[1];
67      humA = sp[2];
68    }
69  });
70
71  client.on('message', function (topic, message){
72    console.log("Mensaje Recibido desde -> "+ topic + " Mensaje -> "+ message.toString());
73    if (topic == "vlesp2"){
74      var msg = message.toString();
75      var sp = msg.split(",");
76      temp1 = sp[0];
77      humA1 = sp[1];
78    }
79  });
80
81  setInterval(function(){
82    var query = "INSERT INTO `admin_proyecInv`.`firstsensorunit` (`firstsensorunit_temp`, `firstsensorunit_humS`, `firstsensorunit_humA`)"
83  con.query(query, function (err, result, fields){
84    if (err) throw err;
85    console.log("Fila insertada correctamente");
86  });
87  }, 2400000);
88
89  setInterval(function(){
90    var query = "INSERT INTO `admin_proyecInv`.`secondsensorunit` (`secondsensorunit_temp`, `secondsensorunit_humA`) VALUES (" + temp1 +
91  con.query(query, function (err, result, fields){
92    if (err) throw err;
93    console.log("Fila insertada correctamente");
94  });
95  }, 2400000);
96

```

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

```

101 setinterval(function(){
102   con.query("SELECT `firstsensorunit_temp` AS ult_tempa FROM `admin_proyecInv`.`firstsensorunit` ORDER BY `firstsensorunit_date` DESC LIMIT 1", (err, row) => {
103     if(err) return console.log("err getting the value.");
104     tempea = row[0].ult_tempa;
105
106     if(tempea > 29){
107       var transporter = nodemailer.createTransport({
108         host: 'smtp.gmail.com',
109         port: 465,
110         secure: true,
111         auth: {
112           user: 'wirelessnetworksensors.inv@gmail.com',
113           pass: 'irivnxcenjhufsst'
114         }
115       });
116       //enviamos las opciones del correo a enviar
117       var mailOptions = [
118         from: "remitente",
119         to: "lmaevendano97@outlook.es",
120
121         subject: "ALERTA DE TEMPERATURA",
122         text: "La temperatura esta por encima de los 29°C, por favor tomar una corrección para solventar este evento"
123       ];
124       transporter.sendMail(mailOptions,(error,info)=>{
125         if(error){
126           res.status(500).send(error.message);
127         }else{
128           console.log("Mail enviado exitosamente");
129           res.status(200).jsonp(res.body);
130           //console.log(res.status(200).jsonp(res.body));
131         }
132       });
133     }
134   });
135   //end
136 }, 2400000);
137
138 setinterval(function(){
139   con.query("SELECT `firstsensorunit_temp` AS ult_tempb FROM `admin_proyecInv`.`firstsensorunit` ORDER BY `firstsensorunit_date` DESC LIMIT 1", (err, row) => {
140     if(err) return console.log("err getting the value.");
141     tempeb = row[0].ult_tempb;
142
143     if(tempeb < 20){
144       var transporter = nodemailer.createTransport({
145         host: 'smtp.gmail.com',
146         port: 465,
147         secure: true,
148         auth: {
149           user: 'wirelessnetworksensors.inv@gmail.com',
150           pass: 'irivnxcenjhufsst'
151         }
152       });
153       //enviamos las opciones del correo a enviar
154       var mailOptions = [
155         from: "remitente",
156         to: "lmaevendano97@outlook.es",
157
158         subject: "ALERTA DE TEMPERATURA",
159         text: "La temperatura esta por debajo de los 20°C, por favor tomar una corrección para solventar este evento"
160       ];
161       transporter.sendMail(mailOptions,(error,info)=>{
162         if(error){
163           res.status(500).send(error.message);
164         }else{
165           console.log("Mail enviado exitosamente");
166           res.status(200).jsonp(res.body);
167           //console.log(res.status(200).jsonp(res.body));
168         }
169       });
170     }
171   });
172 });
173 }, 2400000);

```

```
213 setinterval(function(){
214   con.query("SELECT `firstsensorunit_hums` AS ult_humSA FROM `admin_proyecInv`.`firstsensorunit` ORDER BY `firstsensorunit_date` DESC LIMIT 1", (err, row) => {
215     if(err) return console.log("err getting the value.");
216     humsUA = row[0].ult_humSA;
217     console.log(humsUA);
218     if(humsUA > 70){
219       var transporter = nodemailer.createTransport({
220         host: 'smtp.gmail.com',
221         port: 465,
222         secure: true,
223         auth: {
224           user: 'wirelessnetworksensors.inv@gmail.com',
225           pass: 'irivnxcenjhufsst'
226         }
227       });
228       //enviamos las opciones del correo a enviar
229       var mailOptions = {
230         from: "remitente",
231         to: "Imavendano9@outlook.es",
232
233         subject: "ALERTA DE HUMEDAD",
234         text: "La Humedad del suelo supera el 45%, es necesario si esta regando dejar de hacerlo."
235       }
236
237       transporter.sendMail(mailOptions,(error,info)=>{
238         if(error){
239           res.status(500).send(error.message);
240         }else{
241           console.log("Mail enviado exitosamente");
242           res.status(200).jsonp(res.body);
243           //console.log(res.status(200).jsonp(res.body));
244         }
245       });
246     }
247   });
248 });
249 
```

```

setInterval(function(){
  con.query("SELECT `firstsensorunit_hums` As ult_hums FROM `admin_proyecInv`.`firstsensorunit` ORDER BY `firstsensorunit_date` DESC LIMIT 1", (err, row) => {
    if(err) return console.log("err getting the value.");
    humsu = row[0].ult_hums;
    console.log(humsu);
    if(humsu < 30){
      var transporter = nodemailer.createTransport({
        host: "smtp.gmail.com",
        port: 465,
        secure: true,
        auth: {
          user: 'wirelessnetworksensors.inv@gmail.com',
          pass: 'irivnxcenjhufsst'
        }
      });
      //enviamos las opciones del correo a enviar
      var mailOptions = {
        from: "remitente",
        to: "lmavendano97@outlook.es",

        subject: "ALERTA DE HUMEDAD",
        text: "El cultivo necesita ser regado, por favor active el sistema de riego desde la pagina web"
      }

      transporter.sendMail(mailOptions,(error,info)=>{
        if(error){
          res.status(500).send(error.message);
        }else{
          console.log("Mail enviado exitosamente");
          res.status(200).jsonp(res.body);
          //console.log(res.status(200).jsonp(res.body));
        }
      });
    }
  });
}, 2400000);

```

```

252 setInterval(function(){
253   var query = 'SELECT 1 + 1 as result';
254
255   con.query(query, function (err, result, fields){
256     if (err) throw err;
257   });
258
259 }, 5000);
260

```

ANEXO H Código de la ESP8266 Unidad sensora #1.

```
⌚ main.cpp  X
access_control > src > ⌚ main.cpp > ⚡ loop()
1  #include <Arduino.h>
2  #include <ESP8266WiFi.h>
3  #include <PubSubClient.h>
4  #include "DHT.h"
5
6  // TEST DHT
7
8  #define DHTPIN 4
9  #define DHTTYPE DHT22
10
11 DHT dht(DHTPIN, DHTTYPE);
12
13 // CIERRE TEST DHT
14
15 const int sensorPin = A0;
16 const int dry = 870;
17 const int wet = 380;
18
19 const char* ssid = "Randomsux";
20 const char* password = "KenkerSux2020";
21
22 const char *mqtt_server = "proyectoinv.ga";
23 const int mqtt_port = 1883;
24 const char *mqtt_user = "web_client";
25 const char *mqtt_pass = "54321";
26
27 WiFiClient espClient;
28 PubSubClient client(espClient);
29
30 long lastMsg = 0;
31 char msg[25];
32
33 int humS = 0;
34 int temp = 1;
35 int humA = 2;
36
37
```

```
43 void setup_wifi();
44 void callback(char* topic, byte* payload, unsigned int length);
45 void reconnect();
46
47 void setup() {
48     pinMode(5, OUTPUT);
49     Serial.begin(115200);
50     randomSeed(micros());
51     setup_wifi();
52     client.setServer(mqtt_server, mqtt_port);
53     client.setCallback(callback);
54
55     // TEST DHT
56     dht.begin();
57     // CIERRE TEST DHT
58 }
59
60 void loop() {
61     int sensorVal = analogRead(sensorPin);
62
63     // TEST DHT
64     delay(2000);
65
66     // CIERRE TEST DHT
67
68
69     if (!client.connected())
70     {
71         reconnect();
72     }
73
74     client.loop();
75
76 }
```

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 1.0

```
75     client.loop();
76
77     long now = millis();
78     if (now - lastMsg > 500){
79         lastMsg = now;
80
81         humS = map(sensorVal, wet, dry, 100, 0);
82         temp = dht.readTemperature();
83         humA = dht.readHumidity();
84
85         if(isnan(temp) || isnan(humA)){
86             Serial.println("fallo al leer el sensor ");
87             return;
88         }
89
90         String to_send = String(humS) + "," + String(temp) + "," + String(humA);
91         to_send.toCharArray(msg, 25);
92
93         Serial.print("Publicamos mensaje: ");
94         Serial.println(msg);
95         client.publish("values", msg);
96     }
97
98 }
99
100 //*****
101 //*** CONEXIÓN WIFI ***
102 //*****
103
104 void setup_wifi(){
105     delay(10);
106
107     Serial.println();
108     Serial.print("Conectado a ");
109     Serial.println(ssid);
110
111     WiFi.begin(ssid, password);
112 }
```

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 1.0

```
113     while (WiFi.status() != WL_CONNECTED){  
114         delay(500);  
115         Serial.print(".");  
116     }  
117  
118     Serial.println("");  
119     Serial.println("Conectado a red WiFi!");  
120     Serial.println("Dirección Ip: ");  
121     Serial.println(WiFi.localIP());  
122 }  
123  
124 //*****  
125 //*** FUN CALLBACK ***  
126 //*****  
127  
128 void callback(char* topic, byte* payload, unsigned int length){  
129     String incoming = "";  
130     Serial.print("Mensaje recibido desde -> ");  
131     Serial.print(topic);  
132     Serial.println("");  
133     for (int i = 0; i < length; i++){  
134         incoming += (char)payload[i];  
135     }  
136     incoming.trim();  
137     Serial.println("Mensaje -> " + incoming);  
138  
139     if( incoming == "on"){  
140         digitalWrite(5, HIGH);  
141     }else if ( incoming == "off")  
142     {  
143         digitalWrite(5, LOW);  
144     }  
145  
146 }  
147 }
```

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 1.0

```
148
149     //*****
150     //*** RECONEXIÓN ***
151     //*****
152
153 void reconnect(){
154
155     while (!client.connected()){
156         Serial.print("Intentando conexión MQTT");
157
158         String clientId = "esp8266_";
159         clientId += String(random(0xffff), HEX);
160
161         if (client.connect(clientId.c_str(), mqtt_user, mqtt_pass))
162         {
163             Serial.println("Conectado!");
164
165             client.subscribe("motobomba1");
166         }else{
167             Serial.print("Falló :( con error -> ");
168             Serial.print(client.state());
169             Serial.println(" Intentemos de nuevo en 5 segundos");
170
171             delay(5000);
172         }
173     }
174 }
175 }
```

ANEXO I Código de la ESP8266 unidad sensora #2

```
main.cpp  X

sensor_control > src > main.cpp > ...
1  #include <Arduino.h>
2  #include <ESP8266WiFi.h>
3  #include <PubSubClient.h>
4  #include "DHT.h"
5
6  // TEST DHT
7
8  #define DHTPIN 5
9  #define DHTTYPE DHT22
10
11 DHT dht(DHTPIN, DHTTYPE);
12
13 // CIERRE TEST DHT
14
15
16 const char* ssid = "Randomsux";
17 const char* password = "KenkerSux2020";
18
19 const char *mqtt_server = "proyectoinv.ga";
20 const int mqtt_port = 1883;
21 const char *mqtt_user = "web_client";
22 const char *mqtt_pass = "54321";
23
24 WiFiClient espClient1;
25 PubSubClient client(espClient1);
26
27 long lastMsg1 = 0;
28 char msg1[25];
29
30
31 int temp1 = 1;
32 int humA1 = 2;
```

```
40 void setup_wifi();
41 void callback(char* topic, byte* payload, unsigned int length);
42 void reconnect();
43
44 void setup() {
45
46     Serial.begin(115200);
47     randomSeed(micros());
48     setup_wifi();
49     client.setServer(mqtt_server, mqtt_port);
50     client.setCallback(callback);
51
52     // TEST DHT
53     dht.begin();
54     // CIERRE TEST DHT
55 }
56
57 void loop() {
58
59     // TEST DHT
60     delay(2000);
61
62
63     // CIERRE TEST DHT
64
65
66
67 if (!client.connected())
68 {
69     reconnect();
70 }
71
72 client.loop();
73 }
```

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 1.0

```
74     long now = millis();
75     if (now - lastMsg1 > 500){
76         lastMsg1 = now;
77
78         temp1 = dht.readTemperature();
79         humA1 = dht.readHumidity();
80
81         if(isnan(temp1) || isnan(humA1)){
82             Serial.println("falló al leer el sensor ");
83             return;
84         }
85
86         String to_send = String(temp1) + "," + String(humA1);
87         to_send.toCharArray(msg1, 25);
88
89         Serial.print("Publicamos mensaje: ");
90         Serial.println(msg1);
91         client.publish("vlesp2", msg1);
92
93
94
95
96     }
97
98 }
```

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 1.0

```
104 void setup_wifi(){
105     delay(10);
106
107     Serial.println();
108     Serial.print("Conectado a ");
109     Serial.println(ssid);
110
111     WiFi.begin(ssid, password);
112
113     while (WiFi.status() != WL_CONNECTED){
114         delay(500);
115         Serial.print(".");
116     }
117
118     Serial.println("");
119     Serial.println("Conectado a red WiFi!");
120     Serial.println("Dirección Ip: ");
121     Serial.println(WiFi.localIP());
122 }
123
124 //*****
125 //*** FUN CALLBACK ***
126 //*****
127
128 void callback(char* topic, byte* payload, unsigned int length){
129     String incoming = "";
130     Serial.print("Mensaje recibido desde -> ");
131     Serial.print(topic);
132     Serial.println("");
133     for (int i = 0; i < length; i++){
134         incoming += (char)payload[i];
135     }
136     incoming.trim();
137     Serial.println("Mensaje -> " + incoming);
138
139 }
```

INFORME FINAL DE TRABAJO DE GRADO EN MODALIDAD DE
PROYECTO DE INVESTIGACIÓN, DESARROLLO TECNOLÓGICO,
MONOGRAFÍA, EMPRENDIMIENTO Y SEMINARIO

VERSIÓN: 1.0

```
146 void reconnect(){
147
148     while (!client.connected()){
149         Serial.print("Intentando conexión Mqtt");
150
151         String ClientId = "esp_";
152         clientId += String(random(0xffff), HEX);
153
154         if (client.connect(ClientId.c_str(), mqtt_user,mqtt_pass))
155         {
156             Serial.println("Conectado!");
157
158         }else{
159             Serial.print("Falló :( con error -> ");
160             Serial.print(client.state());
161             Serial.println(" Intentemos de nuevo en 5 segundos");
162
163             delay(5000);
164         }
165     }
166 }
167 }
168 }
```