



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

TÍTULO

Implementación de un sistema automatizado de riego por goteo basado en IoT para un prototipo de invernadero de tomates

AUTOR

Chamba Gonzaga, Holger Andrés

TRABAJO DE TITULACIÓN

Previo a la obtención del grado académico en
MAGÍSTER EN ELECTRÓNICA Y AUTOMATIZACIÓN

TUTOR

Chávez García, Geovanny Danilo

Santa Elena, Ecuador

Año 2025



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO
TRIBUNAL DE SUSTENTACIÓN**

**Ing. Alicia Andrade Vera, Mgtr.
COORDINADORA DEL
PROGRAMA**

**Ing. Geovanny Chávez García, Ph.D.
TUTOR**

**Ing. Luis Chuquimarca Jiménez, Mgtr.
DOCENTE
ESPECIALISTA**

**Ing. Junior Figueroa Olmedo, Mgtr.
DOCENTE
ESPECIALISTA**

**Abg. María Rivera González, MSc.
SECRETARIA GENERAL
UPSE**



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por Holger Andrés Chamba Gonzaga, como requerimiento para la obtención del título de Magíster en Electrónica y Automatización.

TUTOR

Ing. Geovanny Chávez García, Ph.D.

Santa Elena, 14 de abril de 2025



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

DECLARACIÓN DE RESPONSABILIDAD

Yo, HOLGER ANDRÉS CHAMBA GONZAGA

DECLARO QUE:

El trabajo de Titulación, Implementación de un sistema automatizado de riego por goteo basado en IoT para un prototipo de invernadero de tomates, previo a la obtención del título en Magíster en Electrónica y Automatización, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Santa Elena, 14 de abril de 2025

EL AUTOR

Holger Andrés Chamba Gonzaga



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE CIENCIAS DE LA INGENIERÍA
INSTITUTO DE POSTGRADO**

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado Implementación de un sistema automatizado de riego por goteo basado en IoT para un prototipo de invernadero de tomates, presentado por el estudiante Holger Andrés Chamba Gonzaga fue enviado al Sistema Antiplagio COMPILATIO, presentando un porcentaje de similitud correspondiente al 9%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



TUTOR

Ing. Geovanny Chávez García, Ph.D.



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

AUTORIZACIÓN

Yo, HOLGER ANDRÉS CHAMBA GONZAGA

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de mi trabajo de examen complejo con fines de difusión pública, además apruebo la reproducción de este trabajo de examen complejo dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor.

Santa Elena, 14 de abril de 2025

EL AUTOR

Holger Andrés Chamba Gonzaga

AGRADECIMIENTO

En primer lugar, agradezco a Dios, fuente inagotable de sabiduría, fortaleza y esperanza, por haberme acompañado en cada momento de este proceso. Sin su guía y bendición, este logro no habría sido posible.

Quiero expresar mi gratitud a mis docentes, quienes con su paciencia, dedicación y pasión compartieron sus conocimientos, motivándome a alcanzar la excelencia. Gracias por la enseñanza impartida en los diferentes temas tecnológicos, la cual ha sido fundamental para mi desarrollo académico y profesional.

Holger Andrés, Chamba Gonzaga

DEDICATORIA

Dedico este trabajo a Dios, por ser la fuente de mi fortaleza y guía en cada paso de este camino.

A mis docentes, quienes, con su sabiduría y vocación, sembraron en mí el deseo de aprender y mejorar cada día. Su compromiso ha sido una inspiración constante en mi formación profesional.

A mis compañeros de estudio, con quienes compartí momentos de esfuerzo y superación. Gracias por su amistad y colaboración a lo largo de esta etapa.

También dedico este logro a todos aquellos que, de alguna forma, dejaron huella en mi vida. Cada palabra de aliento, consejo y gesto de apoyo ha sido fundamental para llegar hasta aquí.

Finalmente, dedico este esfuerzo a mí mismo, como un recordatorio de que el trabajo duro y la perseverancia siempre encuentran su recompensa.

Holger Andrés, Chamba Gonzaga

ÍNDICE GENERAL

Contenido

TÍTULO.....	I
TRIBUNAL DE SUSTENTACIÓN.....	II
CERTIFICACIÓN.....	III
DECLARACIÓN DE RESPONSABILIDAD	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN	VI
AGRADECIMIENTO.....	VII
DEDICATORIA.....	VIII
ÍNDICE GENERAL	IX
ÍNDICE DE TABLAS	XIII
ÍNDICE DE FIGURAS.....	XIV
RESUMEN.....	XVIII
ABSTRACT	XIX
INTRODUCCIÓN.....	1
Formulación del problema de investigación.....	4
OBJETIVO GENERAL:	4
OBJETIVOS ESPECÍFICOS:	4
HIPOTESIS:	5
CAPÍTULO 1. MARCO TEÓRICO REFERENCIAL	6
1.1. REVISIÓN DE LITERATURA.....	6
1.2 DESARROLLO TEÓRICO Y CONCEPTUAL.....	8
1.2.1 RIEGO AGRICOLA	8
FUNCIONAMIENTO	9

1.2.2 CASOS CON APLICACIONES IOT	12
1.2.3 SENSORES USADOS EN EL AREA DE LA AGRICULTURA CONSIDERANDO SU TECNOLOGIA Y FUNCION	13
1.2.4 CONECTIVIDAD EN EL INTERNET DE LAS COSAS (IOT).....	16
1.2.5 PH DEL SUELO	18
1.2.6. IMPORTANCIA DEL PH EN EL CULTIVO DE TOMATES.....	19
Un pH inferior a 5.5 puede provocar toxicidad de aluminio y deficiencias de calcio, afectando el desarrollo de la planta (Fernandes et al., 2017).....	19
1.2.7 MÉTODOS DE MEDICIÓN DEL PH	19
1.2.8. MONITOREO DE TEMPERATURA Y HUMEDAD.....	20
1.2.9. INTEGRACIÓN DE SENSORES	20
1.2.10. NODE_RED.....	20
1.2.11. AWS IOT	21
1.2.12 TUYA SMART IOT.....	22
1.2.13. API.....	23
1.2.14. MQTT	23
1.2.15. NODEMCU ESP8266	26
1.2.16 SENSOR DE HUMEDAD FC-28.....	29
1.2.17. SENSOR DE TEMPERATURA Y HUMEDAD AMBIENTE DHT11 31	
1.2.18. CONTROLADOR DE RIEGO INALÁMBRICO INTELIGENTE TUYA SMART.....	32
1.2.19 ROUTER INALÁMBRICO N TPLINK	32
CAPÍTULO 2. METODOLOGÍA.....	34
2.1 DIAGRAMA GENERAL DEL PROYECTO.....	34

2.2	CONTEXTO DE LA INVESTIGACIÓN	36
2.3	JUSTIFICACIÓN DEL MÉTODO:	36
2.4	POBLACIÓN Y MUESTRA	36
2.5.	ESQUEMA DE DISTRIBUCIÓN DE UN INVERNADERO.....	37
2.6.	TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS	38
2.6.1.	SENSORES DE HUMEDAD DEL AMBIENTE, Y TEMPERATURA AMBIENTE:	39
2.6.3	FUENTE DE PODER INS DE 850W GENERICA.....	44
2.6.4.	BLOQUE DE TERMINALES JINH	44
2.6.5.	CONSUMO DE LOS SENSORES	45
2.6.6.	SENSORES FC-28 (10 UNIDADES)	45
2.6.7.	SENSORES DHT11 (10 UNIDADES)	45
2.6.8.	PLACA ARDUINO MEGA 2560	45
2.6.9.	PLACA ESP8266.....	46
2.6.10.	SUMANDO EL CONSUMO DE TODOS LOS DISPOSITIVOS: ..	46
2.6.11.	CÁLCULO DE POTENCIA EN VATIOS (W)	46
2.7.	ESQUEMA DE LA PROPUESTA.....	46
2.7.1.	CONFIGURACIÓN DE TUYA SMART:.....	46
2.7.2.	INSTALACIÓN DE NODE-RED:	47
2.7.3.	CONFIGURACIÓN DE AWS IOT:	47
2.7.4.	INTEGRACIÓN Y PRUEBAS:.....	47
CAPITULO 3 RESULTADOS Y DISCUSIONES		48
3.1.	INTRODUCCIÓN.....	48
3.2.	DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE RIEGO DE GOTEO BASADO EN IOT	48
3.2.1	CREACIÓN DE LA NUBE TUYA SMART PARA LA VÁLVULA DE RIEGO	49

3.2.2	IMPLEMENTACIÓN DE UN SISTEMA DE RIEGO POR GOTEO BASADO EN IOT PARA UN PROTOTIPO DE INVERNADERO DE TOMATES A TRAVÉS DE AWS IOT	57
	ARCHIVO SIN EXTENSION.....	61
	61
	ARCHIVOS CON EXTENSION.....	61
3.3.	RESULTADOS DE LOS PARÁMETROS MEDIDOS	75
	DEFINIR LOS ATRIBUTOS PRINCIPALES.....	77
3.3.1.	CONFIGURACIÓN DEL PROTOCOLO MQTT	78
3.3.2.	PRESUPUESTO.....	84
3.4.	DISCUSIÓN.....	85
3.4.1	EFICIENCIA DEL SISTEMA	85
3.4.2	LIMITACIONES Y DESAFÍOS	85
3.4.3	COMPARACIÓN CON MÉTODOS TRADICIONALES	86
3.5.	CONCLUSIONES	87
	BIBLIOGRAFIA.....	88
	ANEXOS	91

ÍNDICE DE TABLAS

Tabla 1	Sensores Meteorológicos	13
Tabla 2	Sensores Climáticos	14
Tabla 3	Sensores Par	15
Tabla 4	Sensores de Agua.....	15
Tabla 5	Características Técnicas de placas lot.....	25
Tabla 6	Comparación de sensores de Humedad del Suelo	27
Tabla 7	Comparación de Sensores de Humedad y Temperatura del Ambiente	30
Tabla 8	ESPECIFICACIÓN DEL SENSOR DE TEMPERATURA Y HUMEDAD DHT11.....	40
Tabla 9	Especificaciones del NodeMCU V2 ESP8266.....	42
Tabla 10	<i>Presupuesto de materiales</i>	84

ÍNDICE DE FIGURAS

Figura 1	Logotipo de Node Red.....	21
Figura 2	Interacción entre los dispositivos AWS IOT	21
Figura 3	Enrutamiento Tuya Smart Cloud.....	22
Figura 4	Diagrama de arquitectura de comunicación de Tuya IoT Cloud.	23
Figura 5	Arquitectura MQTT.....	24
Figura 6	Placa NodeMcu-Esp8266	26
Figura 7	Sensor de Humedad del Suelo FC-28	29
Figura 8	Sensor de Humedad y Temperatura Ambiental DTH11	31
Figura 9	Controlador de riego Tuya Smart	32
Figura 10	Router TP-LINK TL-WR841N.....	33
Figura 11	Diagrama General del Proyecto, muestra la conectividad entre los diferentes elementos de la propuesta.....	34
Figura 12	Diseño del Invernadero del Proyecto.....	37
Figura13	Sensor Fc-28 conexión	39
Figura 14	Conexión Dht 11	40
Figura 15	Controlador de Riego Tuya Smart.....	43
Figura 16	Fuente de Poder Ins de 850W	44
Figura 17	Bloques de Terminales Jinh	45
Figura 18	Conexión de Comunicación de la Propuesta	47
Figura 19	Log in with Tuya Account.....	49
Figura 20	Developer Platform	50
Figura 21	Elección del Datacenter para Tuya Smart Cloud segun la region ...	50

Figura 22 Cloud - Development.....	51
Figura 23 Overview.....	51
Figura 24 Devies o Dispositivos	52
Figura 25 Devices Link App Account.....	52
Figura 26 Logo en el celular de Smart Life	53
Figura 27 Command prompt para ejecutar node red	54
Figura 28 Se Designa la url para ejecutar node red.....	54
Figura 29 Respuesta del dispositivo	56
Figura 30 Portal para la elección de un bucket AWS S3.....	59
Figura 31 Creación de un S3 Bucket para AWS Lambda	60
Figura 32 Archivo sin extensión en S3	61
Figura 33 Archivo con extensión en S3.....	61
Figura 34 Verificación de los objetos según la regla de AWS IoT Core para almacenamiento en S3.....	62
Figura 35 Menú de propiedades del bucket y se activan las opciones necesarias.....	62
Figura 36 Configuración de permisos en Amazon S3 para habilitar el alojamiento de páginas web estáticas	63
Figura 37 . Configuración de ACL en Amazon S3.....	63
Figura 38 Configuración de bloqueo de acceso público en Amazon S3	64
Figura 39 Interfaz de búsqueda de IAM en AWS.	65
Figura 40 Interfaz de administración de políticas en AWS IAM (Identity and Access Management), resaltando la opción para crear una nueva política.....	66
Figura 41 Integración entre Amazon S3 y AWS Lambda, creación de políticas	67

Figura 42 Integración entre Amazon S3 y AWS Lambda, incluyendo la creación de rol	68
Figura 43 integración entre Amazon S3 y AWS Lambda, creación de función	69
Figura 44 Integración entre Amazon S3 y AWS Lambda, ejecución de código lambda	73
Figura 45 Integración entre Amazon S3 y AWS Lambda, configuración de disparador.....	74
Figura 46 Configuración de una política de roles en AWS IAM para permitir que una función Lambda interactúe con Amazon DynamoDB y S3.	74
Figura 47 Resumen de permisos en AWS IAM para Lambda.....	75
Figura 48 Condiciones climáticas de un vivero de tomates, temperatura en tiempo real.....	75
Figura 49 Condiciones climáticas de un vivero de tomates, humedad del ambiente en tiempo real.	76
Figura 50 Condiciones climáticas de un vivero de tomates, humedad del suelo en tiempo real.	76
Figura 51 Registro de datos en AWS DynamoDB.....	78
Figura 52 Comprobación en Arduino y Placa Esp8266.....	79
Figura 53 Comprobación Placa Esp8266	79
Figura 54 Interfaz Node Red relación.....	80
Figura 55 Invernadero a Escala estado Inicial	81
Figura 56 Sembrío de Tomate semana 5	82
Figura 57 Sensor DHT11 en el cultivo.....	82
Figura 58 Sensor FC 28 de Temperatura y Humedad en el cultivo	83

RESUMEN

El proyecto tiene como objetivo la implementación de un sistema de riego por goteo automatizado para un invernadero de tomates, utilizando tecnologías basadas en el Internet de las Cosas (IoT). Este sistema permitirá el monitoreo en tiempo real de variables críticas como la humedad del suelo, humedad ambiente y temperatura ambiente, optimizando el riego según las necesidades específicas de las plantas de tomate.

A través del uso de sensores de humedad, temperatura, conectados a plataformas de análisis y control, se busca mejorar la eficiencia del proceso de riego, incrementando la productividad y asegurando un uso adecuado de los recursos. Además, se contempla la integración con plataformas en la nube para el almacenamiento y análisis de datos, facilitando la toma de decisiones y la automatización del sistema.

Este enfoque no solo mejora la gestión del riego, sino que también contribuye a la sostenibilidad y al aumento de la producción en invernaderos especializados en tomates.

Palabras clave: IoT, riego por goteo, agricultura de precisión, monitoreo de cultivos

ABSTRACT

The project aims to implement an automated drip irrigation system for a tomato greenhouse using Internet of Things (IoT) technologies. This system will enable real-time monitoring of critical variables such as soil moisture, ambient humidity, and ambient temperature, optimizing irrigation according to the specific needs of tomato plants.

By utilizing moisture and temperature sensors connected to analysis and control platforms, the goal is to enhance the efficiency of the irrigation process, increase productivity, and ensure proper resource utilization. Additionally, integration with cloud platforms for data storage and analysis is planned, facilitating decision-making and system automation.

This approach not only improves irrigation management but also contributes to sustainability and increased production in tomato-specialized greenhouses.

Keywords: IoT, drip irrigation, precision agriculture, crop monitoring

INTRODUCCIÓN

La agricultura, a nivel mundial, enfrenta el desafío de satisfacer la creciente demanda de alimentos con recursos hídricos cada vez más limitados. El sector agrícola consume alrededor del 70% del agua dulce extraída a nivel global, según el Banco Mundial y se estima que este porcentaje seguirá aumentando en las próximas décadas. Esta situación se intensifica en la producción de tomate, uno de los cultivos más importantes a nivel mundial, que requiere de grandes cantidades de agua para su desarrollo (Donaldo Zúniga, 2021).

Sin embargo, la pérdida de líquido para las plantaciones de invernaderos de tomate y plantaciones de ciclo corto aun depende de controles manuales, los cuales yacen en problemas cotidianos para el agricultor en el Ecuador, como la irrigación excesiva, esto debe ser equilibrado para que no exista desperdicio de líquido y no provoque erosión en el suelo con el tiempo, arraigando en un suelo infértil a tan temprano. (Ángel Enríquez Estrella, 2021)

El trabajo propuesto por (Martínez-Rodríguez, 2017), evalúa el volumen y frecuencia de riego en función de la capacidad de retención de agua del sustrato, así como el volumen de drenado sobre el rendimiento y calidad de fruto de tomate bola variedad Zyanya. El volumen y frecuencia de riego para tomate bola variedad Zyanya de acuerdo a la capacidad de retención de agua del tezontle con 20% de volumen de drenado fue de 6 284.75 m³ ha⁻¹ distribuido de acuerdo a la necesidad hídrica de la planta de dos a 15 riegos. Finalmente, (Cadena-Jiménez, 2022) proponen abordar el concepto de agricultura con internet de las cosas. En este proyecto se propone utilizar la plataforma de Cloud Computing Amazon Web Services (AWS) como parte integrante de su solución debido a que cumple con los requisitos de seguridad, complejidad, servicios y costos que se adaptan a las necesidades del proyecto. Porque AWS es una plataforma de cloud computing de nube pública que consta de una gama de servicios informáticos especializados para múltiples aplicaciones.

El objetivo de este trabajo es diseñar un sistema de riego por goteo automatizado basado en IoT para un invernadero de tomates. El alcance del proyecto incluye la creación, diseño e implementación del sistema, y se limita hasta punto se intervendrá en el invernadero. El invernadero cuenta un tamaño de 4 m de largo por 3 de ancho, localizado en Ecuador, provincia de Santa Elena, así como los factores tratar tales como humedad, temperatura que se controlarán a lo largo del proceso, ya que la implementación de tecnologías de riego de precisión, como el riego por goteo, permite reducir significativamente el consumo de agua en la producción de tomates, optimizando el uso del recurso hídrico y aumentando la productividad. (INTAGRI, 2019).

El Ministerio del Ambiente, Agua y Transición ecológica indica en el Plan Nacional 2021- 2026 del Ecuador, subraya la importancia de la tecnificación agrícola como un objetivo clave para mejorar la eficiencia en el uso de los recursos hídricos. (MINISTERIO DEL AMBIENTE, 2021- 2026)

En este contexto, se resalta la implementación de sistemas de riego inteligentes, basados en riegos localizados y tecnologías de Internet de las Cosas (IoT), como herramientas que permiten un uso más eficiente del agua en la agricultura. Estos sistemas no solo ofrecen un control preciso de los parámetros climáticos, como humedad y temperatura, sino que también permiten adaptar el riego a las necesidades específicas de cada planta, optimizando el uso del agua y reduciendo las pérdidas por evaporación, escorrentía y percolación profunda. La relevancia de este enfoque es crucial, particularmente en la producción de tomate, uno de los cultivos más importantes en Ecuador. (Seyar, 2023)

Por ello la propuesta técnica está basada en este proyecto consiste en un sistema de riego inteligente que emplea sensores DHT11 y FC-28 para medir las condiciones ambientales y del suelo. Estos datos son recopilados por un Arduino Mega 2560 y enviados a la nube a través de un NodeMCU ESP8266.

La información se transmite mediante MQTT, garantizando eficiencia en la comunicación con AWS Cloud, donde se almacena y gestiona con AWS Lambda, facilitando su análisis en Amazon S3 y DynamoDB.

Para conectarse con plataformas externas, Node-RED actúa como middleware entre AWS Cloud y Tuya Cloud, utilizando API REST para actualizar y consultar datos en tiempo real.

El control del riego se gestiona a través de Tuya Smart, que toma decisiones basadas en umbrales de humedad, activando la válvula de riego cuando sea necesario.

Este proyecto tiene una relevancia social notable porque los sistemas de riego automatizados basados en IoT mejoran la sostenibilidad de la agricultura en áreas con escasez de agua. La gestión responsable de los recursos naturales beneficia tanto a los agricultores como a las comunidades que dependen de ella, al reducir el desperdicio de agua y optimizar su uso. (Das1, 2021)

En el ámbito científico, este proyecto utiliza tecnologías emergentes para avanzar en la investigación sobre la optimización de los recursos hídricos, el uso de Internet de las cosas en los sistemas de riego inteligentes abre nuevas perspectivas para investigar cómo los factores climáticos y la eficiencia del riego se relacionan entre sí, con este conocimiento no solo se aplica a la producción de tomates; también puede aplicarse a otras formas de gestionar el agua en áreas afectadas por el cambio climático. Como resultado, el proyecto no solo tiene como objetivo aumentar la productividad agrícola, sino también crear conocimientos útiles que puedan aplicarse a una variedad de contextos agrícolas y ambientales. (MUHAMMAD AYAZ, 2019)

El documento se compone de un primer capítulo donde, se aborda los antecedentes y el marco teórico, seguido por la descripción de la metodología empleada. En el tercer capítulo se detalla el diseño e implementación del sistema, análisis de los resultados obtenidos, se recoge las conclusiones y recomendaciones del proyecto.

Formulación del problema de investigación

¿Cuáles son las tecnologías de tecnificación del riego que puede aportar el IOT en los invernaderos de tomate?

OBJETIVO GENERAL:

Desarrollar un sistema de riego por goteo localizado en un invernadero a escala aprovechando tecnologías emergentes para la producción de tomate.

OBJETIVOS ESPECÍFICOS:

- Realizar una revisión bibliográfica sobre sistemas de riego por goteo basado en IOT y su aplicación en cultivos de tomate para fundamentar el desarrollo del proyecto.
- Crear un invernadero a escala en función de las características del cultivo de tomate de riñón para medir los diferentes parámetros ambientales y nutritivos en un entorno controlable.
- Implementar un sistema de riego por goteo para dosificar agua y nutrientes ajustados a las necesidades del cultivo de tomate, aplicando paneles de visualización que permitan el monitoreo en tiempo real de variables ambientales y del estado del sistema de riego.

- Seleccionar sustratos con propiedades físicas y químicas óptimas para mejorar el desarrollo de las plántulas de tomate en el invernadero.
- Diseñar un sistema de monitoreo basado en sensores para la recolección y análisis de datos ambientales en tiempo real mediante AWS IoT.
- Evaluar el sistema IoT para el riego por goteo automatizado, verificando su eficiencia a través de pruebas experimentales y análisis de resultados.

HIPOTESIS:

H1: La implementación de tecnologías de riego de precisión, como el riego por goteo, permitiría reducir significativamente el consumo de agua en la producción de tomate, aprovechando el uso del recurso hídrico y aumentando la productividad.

H2: La implementación de sistemas de riego inteligentes basadas tecnologías de Internet de las Cosas (IoT) permitiría aprovechar la verificación de los parámetros climáticos en la producción de tomate, adaptando el riego a las necesidades específicas de cada planta y reduciendo las pérdidas por evaporación, escorrentía y percolación profunda

CAPÍTULO 1. MARCO TEÓRICO REFERENCIAL

1.1. REVISIÓN DE LITERATURA

La investigación realizada por Ezekiel Chimana (2024) tuvo lugar en la Granja de la Universidad Abierta de Zimbabue, situada en la provincia de Manicalandia. En este contexto, se creó una aplicación móvil para dispositivos Android con el fin de administrar el sistema de riego en los invernaderos, con la meta de optimizar la agricultura en la región. La aplicación posibilita la evaluación de la necesidad de riego, especialmente en circunstancias críticas o cuando el sistema emite una notificación. El sistema de riego opera de forma automática a través de sensores de humedad, ventilación y dispositivos de calefacción. Este modelo fue creado con el propósito de aumentar la productividad agrícola en la región, asegurando la recopilación de información exacta sobre diversos aspectos físicos, lo cual facilita la toma de decisiones de manera más efectiva. Además, posibilita una supervisión continua, empleando tecnologías emergentes de Internet de las Cosas (IoT).

En su estudio, Atendido et al. (2024) analizaron el fenómeno del estrés hídrico originado por la escasez de agua, lo cual conduce a sequías y repercute de manera adversa en la productividad agrícola de cultivos como el tomate. El estudio se centró en el desarrollo de una interfaz gráfica (GUI) que exhibía las mediciones de los sensores empleados para supervisar el sistema de riego y la fertilización del cultivo de tomates. La propuesta tecnológica utilizó un sistema controlado por un Arduino Mega 2560, conectado a internet a través de un módulo ESP8266 Wi-Fi, para regular el flujo de agua y fertilizantes mediante válvulas solenoides. Se realizó una comparación entre la altura de las plantas de tomate en un diseño propuesto de cuatro pies cuadrados y la configuración convencional, empleando un suelo de sustrato específico para tomates cherry. En cuanto a los resultados obtenidos, el análisis estadístico reveló un valor estimado de t de -0.00072 . Dicho valor, al encontrarse dentro del rango de -2.23 a 2.23 , condujo a los investigadores a no descartar la hipótesis nula. Esto sugiere que no se observaron diferencias significativas en la altura de las plantas entre las dos configuraciones

analizadas. Además, el valor de p de 0.9994, que es mayor a 0.05, ha confirmado la igualdad de las alturas. No obstante, el modelo propuesto se distinguió por disminuir el uso de agua en un 61.3% y de fertilizantes en un 53.3%, ahorrando 34,553 mililitros de agua y 119 mililitros de fertilizante. El gasto de electricidad del prototipo fue de 53 pesos por cada día de 24 horas de funcionamiento a lo largo de un lapso de seis semanas.

El trabajo de investigación realizado por Younsuk Dong (2024), titulado "Implementation of an in-field IoT system for precision irrigation management", se enfoca en la creación de una estación de sensores utilizando tecnología IoT. Esta estación se complementa con un sitio web accesible y una aplicación móvil, ambos diseñados para mejorar la gestión del riego de manera eficiente. Además, se investiga la eficacia de la estación de sensores basada en el Internet de las cosas a través de una demostración práctica. Antes de ser utilizado en el terreno, se realizó un proceso de ajuste de los sensores de humedad del suelo en un ambiente de laboratorio. Los resultados muestran que los sensores calibrados han cumplido con los estándares estadísticos tanto en suelos arenosos como en suelos franco-arenosos. En suelos arenosos, se obtuvo un error cuadrático medio (RMSE) de $0.01 \text{ cm}^3/\text{cm}^3$, un índice de concordancia (IA) de 0.97 y un error medio sesgado (MBE) de 0.01. En suelos franco-arenosos, los valores fueron de $0.023 \text{ cm}^3/\text{cm}^3$ para el RMSE, 0.98 para la IA y -0.02 para el MBE.

Asimismo, el artículo de Younsuk Dong (2024) expone investigaciones de casos en cultivos de maíz, arándanos y tomates en Michigan, Estados Unidos, centrándose especialmente en el análisis del cultivo de tomates. En esta situación, se realizó una exhibición de riego automatizado utilizando una estación de sensores IoT. En ciertos terrenos con suelos de textura arenosa y sistemas de riego por goteo, se ha detectado un exceso de irrigación. Sin embargo, no se observaron disparidades estadísticamente relevantes en la cantidad de tomates aptos para la comercialización ($p = 0.382$) ni en su peso ($p = 0.756$) entre el sistema de riego empleado por los agricultores y la técnica de riego sugerida. Sin embargo, en la parcela donde se implementó la estrategia sugerida, se redujo el consumo de agua en un 30%. En resumen, los resultados señalan que la

estrategia de riego utilizando sensores IoT puede disminuir el uso de agua en un 30% sin impactar el rendimiento ni la calidad del producto.

La investigación realizada por Vinayak Paradkar (2023) puso en marcha un sistema automatizado para programar la fertirrigación en invernaderos. Este sistema empleó sensores de humedad del suelo ubicados a tres niveles distintos (15, 30 y 45 cm) dentro del área de raíces de los tomates. Para determinar la exactitud de los sensores, se emplearon los parámetros de R^2 , RMSE, NSE y MAE, los cuales demostraron una significativa relación entre los datos obtenidos por los sensores y la humedad real del suelo, con coeficientes de determinación (R^2) que variaron entre 0.8642 y 0.9528. Además, se examinaron diversos parámetros biométricos del cultivo de tomate con el fin de evaluar el desempeño del sistema. Estos parámetros incluyeron la estatura de la planta, el grosor del tallo, la cantidad de hojas y flores, las dimensiones y peso de los frutos, la productividad por planta y por hectárea, así como el contenido de azúcar. Los resultados indicaron un rendimiento sobresaliente del sistema de riego por goteo al aplicar el 100% de la evapotranspiración del cultivo (ETc) y el 100% de la dosis recomendada de fertilizante (RDF), obteniendo valores óptimos en todos estos aspectos. En contraste con el tratamiento estándar, este sistema aumentó la productividad del cultivo de tomate en un 62.92% y disminuyó el uso de agua en un 14.84%. El sistema automatizado de fertirrigación empleó 2.27 litros de agua por planta diariamente y mostró ser sumamente eficaz para el cultivo de vegetales en invernadero, mejorando tanto la eficiencia en el uso del agua como la producción de cultivos.

1.2 DESARROLLO TEÓRICO Y CONCEPTUAL

1.2.1 RIEGO AGRICOLA

Para el Riego Agrícola existen varios métodos que se pueden intervenir en un vivero de tomates por consiguiente se mencionan.

RIEGO POR GOTEO

Este método agrícola entrega el agua y fertilizantes directamente a la zona radicular del cultivo, en la cantidad correcta cada planta recibe exactamente lo que necesita, cuando lo necesita para desarrollarse. Por el riego por goteo, los productores pueden tener mejores rendimientos, ahorran agua, así como fertilizantes, energía e incluso agroquímicos.

FUNCIONAMIENTO

El agua y los nutrientes se entregan en el campo a través de un sistema de tuberías llamados “líneas de goteo”, las cuales contienen pequeños dispositivos llamados “goteros”. Cada gotero emite un flujo controlado de gotas que contienen agua y fertilizantes, resultando en una uniforme aplicación de agua y nutrientes directamente a la zona radicular de cada planta a lo largo de todo el campo.

RIEGO POR ASPERSION

Consiste en aplicar el agua imitando la lluvia, mediante un chorro de agua pulverizada en gotas. El mecanismo funciona a través de una red de tuberías que transporta el agua hasta los aspersores, los cuales utilizan presión para dispararla. El riego es potenciado a través de un sistema de bombeo.

Con este mecanismo es posible alcanzar eficiencias de aplicación entre 80 y 85 %, al implementar este sistema de riego por aspersión incluso llega a los rincones más profundos de su cosecha, de forma tal que todo el terreno esté cubierto lo más homogéneamente posible. Por lo general, las tuberías que componen el sistema van enterradas. Se compone de tomas de agua o hidrantes, ramales de aspersión y los aspersores como tal.

FUNCIONAMIENTO

Debe saber que tanto el riego como el goteo se utilizan como dispositivos de agua a través de un sistema presurizado, pero en el caso de la aspersión funciona principalmente por:

- Amplia adaptabilidad a las dosis de riego.
- Automatización hidráulica para la cosecha.
- Posibilita el tratamiento con fertilizantes, fitosanitarios y anti-heladas.

- Mayor volumen de agua para que los cultivos se conserven en el mejor estado.

Asimismo, es importante que sepa que el agua que se recoge para el riego de plantas u otras cosechas, puede provenir de un pozo, río, lago, etc. Entonces, se usa una estructura que permite el almacenamiento del agua para que este funcione de forma precisa. El sistema aplica para terrenos grandes o pequeños.

RIEGO POR GRAVEDAD O SURCOS

El riego por surcos consiste en hacer correr el agua a través de una zanja o surco a toda la parcela; el agua llega hasta las raíces de los cultivos introduciéndose hacia los lados y hacia el fondo del surco de riego. Esta forma de riego es la que más se usa en la agricultura. Los surcos se hacen en forma de V con una profundidad de 15-20 centímetros y 25 a 30 cm. de ancho en la parte de arriba y con un desnivel del 1% para que el agua corra sin dificultad y sin causar erosión.

FUNCIONAMIENTO

Precipitación anual Se recomienda la tecnología para zonas secas o semi húmedas con precipitaciones que llegan hasta los 1500 milímetros anuales, principalmente cuando se tienen periodos secos claramente definidos **Disponibilidad de agua** el riego por surco demanda una buena disponibilidad de agua en la finca con fuentes permanentes y con buen caudal. **Porcentaje de pendiente** Esta tecnología se recomienda para terrenos con pendientes moderadas ya que en terrenos con pendientes fuertes se hace difícil el trazado de los surcos. **Textura del suelo** En terrenos arcillosos la distancia entre surcos puede ser mayor ya que la infiltración del agua es más lenta, a diferencia de un suelo que tiene una infiltración más rápida, en este caso la distancia entre surcos debe ser menor. **Capacidad de infiltración** En suelos con alta capacidad de infiltración el agua se va rápidamente razón por la cual los intervalos entre riego deben ser más cortos.

RIEGO SUBTERRANEO

El riego por goteo subterráneo mejora la eficiencia del riego por las razones indicadas, pero, además, para optimizar su manejo, también disponemos de herramientas que nos permiten monitorizar la humedad en el suelo a distintas profundidades y nos indican a qué profundidad se encuentran las raíces que están realizando la absorción y, por lo tanto, hasta qué profundidad deberíamos aplicar el agua. Con ello, podemos ir ajustando la frecuencia y el tiempo de los riegos, ahorrando agua y fertilizantes.

Un sistema de riego por goteo subterráneo debe incluir además de las tuberías de goteo, los elementos de cabezal como bomba de presión (si es necesaria), filtros, fertirrigación y las válvulas de sector y de corte, los siguientes elementos:

Colectores de drenaje al final de los laterales de goteo para poder realizar el lavado de las tuberías de goteo. Mediante estos colectores, solo con la apertura de válvulas, se realizan periódicamente el lavado de las tuberías para mantener el sistema en buenas condiciones de limpieza y asegurar el funcionamiento.

Ventosas al inicio de la red, junto a válvulas de sector, en puntos altos y en colectores de drenaje. Con las ventosas conseguimos extraer el aire que no debe estar en las tuberías durante su funcionamiento porque las bolsas de aire provocan, entre otros problemas, la disminución de la sección de paso de agua de la tubería y, por tanto, la pérdida de presión. También son necesarias para introducir aire en las tuberías, por ejemplo, durante el lavado mediante los colectores y permitir que salga el agua.

Contador o caudalímetro: muy recomendable para comprobar que el caudal de agua aplicado no varía con el tiempo. Un aumento de caudal puede indicar que hay fugas y una disminución de caudal puede indicar que hay obstrucciones en los goteros.

1.2.2 CASOS CON APLICACIONES IOT

Sistema Inteligente de automatización, monitoreo y control de invernaderos:

La elección del controlador para que acceda a internet debe poder conectarse por medio de Wifi, para ello este artículo Moreno Perdomo, É. J. (2023): desarrolla un sistema IoT basado en **ESP32** para la **automatización, monitoreo y control de invernaderos**. Se emplea un **enfoque cuantitativo y estudio descriptivo**, utilizando la metodología ágil **SCRUM** para el desarrollo por **Sprints**, donde se usa como hardware ESP32 que es una placa para realizar proyecto que necesiten comunicarse por medio de internet, su uso en este artículo se usa para medir las variables críticas del invernadero como humedad del suelo, humedad y temperatura ambiente, luz donde si se produce una anomalía se activa un actuador como una bomba, una válvula de riego, ventilación entre otros.

El controlador ESP32 y los sensores se relacionan para tener una comunicación mediante un proceso que se basa en recopilar la información de datos en el momento actual y son almacenados en la PC y la Nube, ya que estos datos son importantes para el análisis de datos para predecir y toma de decisiones, para ser el sistema más eficiente se desarrolla una plataforma de visualización un aplicativo web de nombre aplicativo web INVERNAP que muestra los valores que reciben por los instrumentos electrónicos

Moreno Perdomo, É. J. (2023). *Sistema inteligente de automatización, monitoreo y control de invernaderos mediante uso de IoT y el microcontrolador ESP-32 con aplicación de aprendizaje automático*. Revista Educación en Ingeniería, 18(35), 1-11. <https://doi.org/10.26507/paper.2818>

En un segundo caso donde trata Serrano Montiel, C. A. (s.f.). *Sistema automatizado de control de un invernadero usando la plataforma Arduino se basa en el uso de la electrónica, donde se miden las variables ambientales que actúan en el desarrollo de hortalizas, aquí en esta propuesta se utiliza un computadora Raspberry Pi 3B (RP)*, donde se activa el proceso de acuerdo a la toma de decisiones pertinentes donde el modelo de control permite actuar en el vivero de acuerdo a los límites establecidos la comunicación es Local, en el mismo se aplica la técnica de hidroponía en el cual se controla las variables como el uso del agua, humedad, temperatura e iluminación.

El tercer caso de uso publicado por Pacheco Zavala, C., Hernández Varela, R. S., Gutiérrez Avelar, A. E., Mora Martínez, M. A., Saldaña Robles, N., & Tapia Tinoco, G. (2021). Se presenta el diseño y desarrollo de un sistema de monitoreo y control para un invernadero a escala utilizando la tecnología de Internet de las cosas . El monitoreo y control se implementa mediante una estructura distribuida de tres sistemas embebidos basados en el microcontrolador ESP32. Cada sistema embebido contiene un microcontrolador, el cual se conecta a una red WiFi e intercambiar información con un gestor denominado Broker mediante el protocolo MQTT. Se describe el prototipo del invernadero, el montaje de sensores y el control de bombas y ventiladores. Además, se detalla el diseño de los algoritmos implementados en cada uno de los ESP32 y del software de monitoreo y control mediante la plataforma Node-RED.

1.2.3 SENSORES USADOS EN EL AREA DE LA AGRICULTURA CONSIDERANDO SU TECNOLOGIA Y FUNCION

1. Sensores Meteorológicos
2. Sensores Climáticos
3. Sensores PAR (Radiación Fotosintéticamente Activa)
4. Sensores de Agua

SENSORES METE REOLÓGICOS

Miden variables atmosféricas para predecir y monitorear el clima.

Tabla 1

Sensores Meteorológicos

SENSORES	TECNOLOGÍA	CLASIFICACIÓN
SENSOR DE TEMPERATURA	Termistores, termopares, RTDs (detectores de temperatura resistivos)	Sensores resistivos, sensores termoeléctricos.

SENSORES DE HUMEDAD	Sensores capacitivos, sensores resistivos.	Sensores electroquímicos, sensores de capacitancia.
SENSORES DE PRESIÓN ATMOSFÉRICA	Sensores piezoresistivos, sensores capacitivos.	Sensores de presión diferencial, sensores de presión absoluta.
ANEMÓMETROS	Sensores de copa, sensores ultrasónicos.	Sensores mecánicos, sensores ultrasónicos.
VELETAS (DIRECCIÓN DEL VIENTO)	Potenciómetros, codificadores ópticos.	Sensores mecánicos, sensores ópticos.
PLUVIÓMETROS (PRECIPITACIÓN)	Sensores de cubeta basculante, sensores de radar.	Sensores mecánicos, sensores de radar.

SENSORES CLIMÁTICOS

Miden variables ambientales específicas para el control de microclimas, especialmente en agricultura y jardinería.

Tabla 2

Sensores Climáticos

SENSORES	TECNOLOGÍA	CLASIFICACIÓN
SENSORES DE CO2	Sensores infrarrojos no dispersivos (NDIR).	Sensores ópticos.
SENSORES DE RADIACIÓN SOLAR	Piranómetros, sensores fotovoltaicos.	Sensores ópticos.

SENSORES PAR (RADIACIÓN FOTOSINTÉTICAMENTE ACTIVA)

Miden la cantidad de luz disponible para la fotosíntesis de las plantas

Tabla 3

Sensores Par

SENSORES	TECNOLOGIA	CLASIFICACION
SENSORES PAR (RADIACIÓN FOTOSINTÉTICAMENTE ACTIVA):	Sensores fotodiodo, sensores espectro radiométricos.	Sensores ópticos, sensores espectrales.

SENSORES DE AGUA

Miden la humedad del suelo, la conductividad eléctrica del agua y otros parámetros relacionados con el agua.

Tabla 4

Sensores de Agua

SENSORES	TECNOLOGIA	CLASIFICACION
SENSORES DE HUMEDAD DEL SUELO	Sensores capacitivos, sensores resistivos, tensiómetros.	Sensores electroquímicos, sensores de capacitancia, sensores de presión.
SENSORES DE CONDUCTIVIDAD ELÉCTRICA (EC)	Electroquímica	Sensores electroquímicos.
SENSORES DE PH	Electroquímica.	Sensores electroquímicos

SENSORES DE NIVEL DE AGUA:	Sensores ultrasónicos, sensores de flotador, sensores de presión	Sensores ultrasónicos, sensores mecánicos, sensores de presión.
-----------------------------------	--	---

1.2.4 CONECTIVIDAD EN EL INTERNET DE LAS COSAS (IOT)

La conectividad en el Internet de las Cosas (IoT) es un aspecto fundamental que permite a los dispositivos comunicarse y compartir datos. Dada la diversidad de aplicaciones IoT, existen múltiples tecnologías de conectividad, cada una con sus propias características y ventajas. A continuación

REDES DE CORTO ALCANCE Y BAJO CONSUMO

BLUETOOTH

La tecnología Bluetooth envía señales de voz y datos hasta una distancia de 10 metros y es ideal para la transferencia de datos de alta velocidad.

NFC

Conjunto de protocolos para la comunicación entre dos dispositivos electrónicos que se encuentren a una distancia de 4 cm o menos. NFC ofrece una conexión de baja velocidad con una configuración sencilla que se puede usar para iniciar conexiones inalámbricas de más capacidad.

WI-FI/802.11

El bajo coste del uso del Wi-Fi lo convierte en un estándar en hogares y oficinas. Sin embargo, es posible que no sea la opción adecuada para todos los escenarios, por su alcance limitado y el consumo energético ininterrumpido.

Z-WAVE

Red en malla que usa ondas de radio de baja potencia para la comunicación de dispositivo a dispositivo.

ZIGBEE

Especificación basada en IEEE 802.15.4 para un conjunto de protocolos de comunicación de alto nivel que se usan para crear redes de área personal con un radio digital pequeño de baja potencia.

REDES DE ÁREA EXTENSA DE BAJO CONSUMO (LPWAN)

Las redes LPWAN permiten la comunicación en un radio mínimo de 500 metros, tienen un consumo de energía mínimo y se usan para la mayoría de los dispositivos IoT. Los siguientes son algunos ejemplos comunes de redes LPWAN:

4G LTE PARA IOT

Estas redes ofrecen alta capacidad y baja latencia, por lo que son una excelente opción para escenarios de IoT que requieren actualizaciones o información en tiempo real.

5G PARA IOT

Aunque aún no está disponible, se espera que las redes 5G para IoT permitan más innovaciones en este campo, ya que proporcionan velocidades de descarga mucho más rápidas y conectividad con muchos más dispositivos en un área determinada.

CAT-0

Estas redes basadas en LTE son la opción menos costosa. Sientan las bases para Cat-M, una tecnología que reemplazará al 2G.

CAT-1

Este estándar para IoT de telefonía móvil reemplazará finalmente al 3G. Las redes Cat-1 son fáciles de configurar y ofrecen una excelente solución para las aplicaciones que requieren una interfaz de explorador o de voz.

LORAWAN

Las redes de área extensa de largo alcance (LoRaWAN) conectan dispositivos móviles, seguros y con batería bidireccional.

LTE CAT-M1

Estas redes son totalmente compatibles con las redes LTE. Optimizan el coste y el consumo de energía en una segunda generación de chips LTE diseñada específicamente para aplicaciones de IoT.

Banda estrecha o NB-IoT/CAT-M2

NB-IoT/Cat-M2 utiliza la modulación del espectro ensanchado por secuencia directa (DSSS) para enviar datos directamente al servidor, lo que elimina la necesidad de una puerta de enlace. Aunque la configuración de las redes NB-IoT es más costosa, el hecho de no necesitar una puerta de enlace hace que su uso resulte más económico.

SIGFOX

Este proveedor internacional de redes de IoT ofrece redes inalámbricas para conectar objetos de baja potencia que emiten datos continuamente.

1.2.5 PH DEL SUELO

Es un factor crucial para el cultivo de tomates, ya que influye en la disponibilidad de nutrientes y en la salud de las plantas. Un pH óptimo para el cultivo de tomates se sitúa

entre 6.0 y 6.8, lo que favorece la absorción de nutrientes esenciales. Con un pH bajo, el sabor del fruto es más agrio o ácido. El sabor se mide normalmente mediante catadores que califican los tomates por una cantidad de características como el olor, aroma, firmeza, jugo, harinosidad, calidad de la piel, ácidos y azúcares. (Yarecuador, 2024)

1.2.6. IMPORTANCIA DEL PH EN EL CULTIVO DE TOMATES

Un pH adecuado permite una mejor disponibilidad de nutrientes como nitrógeno, fósforo y potasio, fundamentales para el crecimiento del tomate (Souza et al., 2010).

Un pH inferior a 5.5 puede provocar toxicidad de aluminio y deficiencias de calcio, afectando el desarrollo de la planta (Fernandes et al., 2017).

1.2.7 MÉTODOS DE MEDICIÓN DEL PH

Se han utilizado sensores para medir el pH del suelo, lo que facilita la identificación de variaciones y la toma de decisiones en el manejo agronómico (López-Betancourt et al., 2016).

La medición precisa del pH permite ajustar las enmiendas del suelo, optimizando así el rendimiento de los cultivos (López-Betancourt et al., 2016).

La implementación de sensores de temperatura en el suelo para el cultivo de tomates es crucial para optimizar el riego y mejorar la salud de las plantas. Los sistemas modernos utilizan tecnología avanzada para monitorear parámetros clave, como la temperatura del suelo, la humedad y el pH, lo que permite a los agricultores tomar decisiones informadas.

1.2.8. MONITOREO DE TEMPERATURA Y HUMEDAD

Un sistema basado en IoT permite a los agricultores medir la temperatura del suelo, que debe mantenerse entre 24-28 °C para un crecimiento óptimo (Gunawan et al., 2019).

La investigación muestra que el control de la humedad del suelo es esencial, con niveles ideales entre 60%-80% para evitar el estrés hídrico (Gunawan et al., 2019).

1.2.9. INTEGRACIÓN DE SENSORES

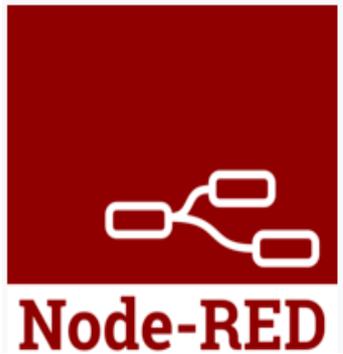
Un prototipo de sistema multi-sensor en invernaderos mide temperatura, humedad y luz, ajustando automáticamente el riego y la ventilación para mantener condiciones óptimas (Bouketir & Boukazoul, 2022). La precisión de los sensores es alta, con errores promedio de 0.92% en la medición de temperatura (Gunawan et al., 2019).

1.2.10. NODE_RED

Node-RED, una plataforma de código abierto, facilita la implementación de técnicas de control avanzadas, permitiendo la integración fluida entre hardware y software especializados. El enfoque de tiempo real de Node-RED asegura una respuesta rápida y precisa, lo que garantiza, además, su capacidad para interactuar con otros dispositivos y sistemas mejora la conectividad y la comunicación eficiente en entornos críticos por ende es una herramienta de programación para interconectar dispositivos de hardware, APIs y servicios en línea, comenzó con la prueba de un concepto para visualizar y manipular topics de MQTT que poco a poco fue mejorando y se convirtió en un proyecto que se extendió para varias direcciones (Pérez, 2021).

Figura 1

Logotipo de Node Red



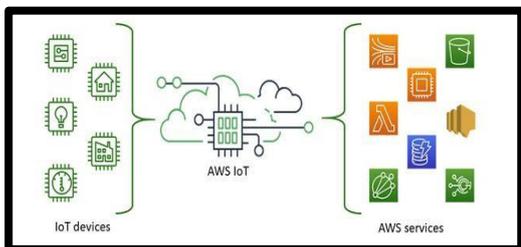
Nota. Esta figura muestra el logotipo de identificación de NODE_RED. "Node-RED: Low-code programming for event-driven applications", por Node-RED, s.f., Node-RED (<https://nodered.org>). Copyright por Node-RED.

1.2.11. AWS IOT

AWS IoT proporciona los servicios en la nube que conectan sus dispositivos de IoT a otros dispositivos y servicios AWS en la nube. AWS IoT proporciona software para dispositivos que puede ayudarlo a integrar sus dispositivos de IoT en soluciones AWS IoT basadas en datos. Si sus dispositivos se pueden conectar AWS IoT, AWS IoT puede conectarlos a los servicios en la nube que AWS proporcionan. (Amazon, 2024).

Figura 2

Interacción entre los dispositivos AWS IOT



Nota. Esta figura muestra cómo los dispositivos IoT se conectan a través de AWS IoT con varios servicios de AWS, permitiendo el procesamiento y análisis de los datos en la nube para mejorar la automatización y la toma de decisiones. "AWS IoT

Core", por Ausum Cloud, s.f., Ausum Cloud (<https://ausum.cloud/aws-iot-core/>).
Copyright por Ausum Cloud.

1.2.12 TUYA SMART IOT

Es un proveedor mundial de servicios de plataforma en la nube con la misión de construir un ambiente de desarrolladores para dar soluciones inteligentes y ayudar a que todas las cosas sean inteligentes. Tuya es una plataforma de desarrollo en la nube diseñada específicamente con capacidades de nube e inteligencia artificial generativa que ofrece una suite completa de servicios, incluyendo Plataforma como Servicio (PaaS), Software como Servicio (SaaS), y soluciones inteligentes para desarrolladores de dispositivos inteligentes, aplicaciones comerciales e industrias. A través de su plataforma de desarrollo en la nube, Tuya ha activado a implosionado a desarrolladores de marcas, OEMs, Agentes de IA, integradores de sistemas y proveedores de software independientes para esforzarse colectivamente por un ecosistema de soluciones inteligentes que encarna los principios de verde y bajo en carbono, seguridad, alta eficiencia, agilidad y apertura. (tuya, 2024)

Figura 3

Enrutamiento Tuya Smart Cloud



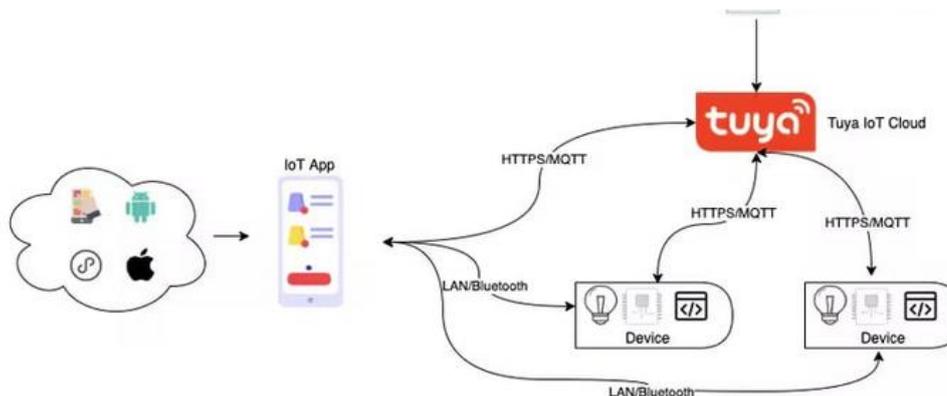
Nota. La figura muestra la arquitectura de Tuya Smart IoT, donde varios dispositivos conectados, como dispositivos NB-IoT, GPRS, 5G, móviles y altavoces inteligentes, se integran a través de TuyaCloud. Esta nube conecta los dispositivos a una red Wi-Fi, permitiendo la comunicación con otros dispositivos y gateways, como Zigbee, Wi-Fi y BLE. "Fundamental elements of IoT: Deploying your IoT projects with expertise", por Dusun IoT, 2023, Dusun IoT (<https://www.dusuniot.com/es/blog/fundamental-elements-of-iot-deploying-your-iot-projects-with-expertise/>). Copyright 2023 por Dusun IoT.

1.2.13. API

El propósito de una API es permitir la comunicación entre componentes de software mediante el soporte de funciones de otros servicios, lo que permite la abstracción del código. Los aspectos fundamentales que se cumplen en la arquitectura son las llamadas HTTP realizadas desde un cliente hacia la base de datos, pasando a través de la API, encargada de resolver estas solicitudes en el formato solicitado en ambos sentidos. La API considera una serie de características para controlar la entrega de la información, siendo el tipo de usuario uno de los factores más importantes para poder realizar o no alguna acción específica. (CHANTAL ALEJANDRA MORALES ROJAS, 2021)

Figura 4

Diagrama de arquitectura de comunicación de Tuya IoT Cloud.



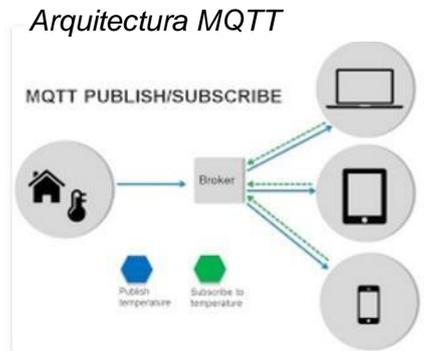
Hackaday. (2022). *Tuya Cloud Country and City APIs*. Hackaday.io. Recuperado el 12 de febrero de 2025, de <https://hackaday.io/project/183324-tuya-cloud-country-and-city-apis>

1.2.14. MQTT

El protocolo de transporte MQTT (Message Queue Telemetry Transport) es utilizado para la comunicación de entornos bajo la tecnología de internet de las cosas y funciona dentro del protocolo de control de transporte TCP. Fue creado por IBM como un protocolo máquina a máquina y estandarizado por ISO/IEC 20922, es un protocolo de mensajes que

no requiere actualizaciones minimizando el uso de recurso y permitiendo que se trabaje en un ambiente con bajo ancho de banda.

Figura 5



El protocolo MQTT trabaja bajo el modelo de publicación-suscripción y tiene 3 componentes básicos: El Broker MQTT que es el servidor de las comunicaciones dentro del protocolo, los dispositivos, las aplicaciones móviles o aplicaciones web, estos últimos 3 son considerados los clientes. (SALAZAR, 2022)

Nota. La imagen ilustra el funcionamiento del protocolo MQTT en un modelo de Publish/Subscribe. Un dispositivo, como un sensor de temperatura, publica los datos a través de un broker central. Luego, los dispositivos suscritos (como un ordenador portátil, una tableta o un teléfono móvil) reciben la información del broker. "Arquitectura MQTT", por Aprendiendo Arduino, s.f., Aprendiendo Arduino (<https://aprendiendoarduino.wordpress.com/tag/arquitectura-mqtt/>). Copyright por Aprendiendo Arduino

COMPARACION DE LAS DIFERENTES PLACAS DE DESARROLLO CON CONECTIVIDAD A INTERNET

Para definir la placa o dispositivo que se va a conectar a internet, se buscó las diferentes características propuesta en la siguiente tabla.

Tabla 5*Características Técnicas de placas lot*

	NODEM	ESP32	RASPBERRY
CARACTERÍSTICA	CU	DEVKI	RY PI
	ESP8266	TC	ZERO W
MICROCONTROLADOR	ESP8266	ESP32	BCM2835
		Wi-Fi	Wi-Fi
CONECTIVIDAD INALÁMBRICA	Wi-Fi	802.11	802.11
	802.11	b/g/n,	b/g/n,
	b/g/n	Bluetooth	Bluetooth
		h LE	4.1
VELOCIDAD DE RELOJ	80 MHz	160-240 MHz	1 GHz
MEMORIA FLASH	4 MB	4-16 MB	- (requiere tarjeta microSD)
MEMORIA RAM	128 KB	520 KB	512 MB
PINES GPIO	17	30+	40
VOLTAJE DE OPERACIÓN	3.3V	3.3V	5V
SISTEMA OPERATIVO	N/A	N/A	Linux (Raspbian, etc.)

La placa de desarrollo para internet se escoge la **NODEMCU** ESP8266 por el bajo costo, porque se mantiene en la mayoría de tiendas electrónicas del mercado, por también ser accesible a amplia variedad de proyectos, su conectividad Wifi integrada ideal para

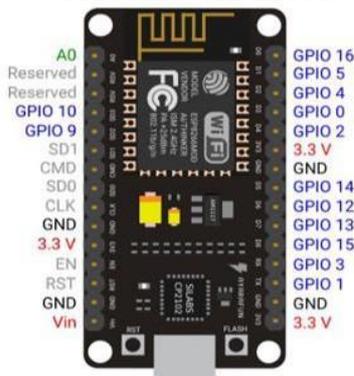
enlazar dispositivos inalámbricos, por último, por su compatibilidad con el software Arduino.

1.2.15. NODEMCU ESP8266

El NodeMCU ESP-12E es una completa plataforma de desarrollo, módulo SoC (System on Chip), basada en un microcontrolador Tensílica L106 de 32 bits, frecuencia de 80 MHz, con memoria flash de 4 MB y SRAM de 32 KB. Es una conexión Wi-Fi estándar 802.11 b/g/n para controlar dispositivos que actúan como servidores de Internet (servidores web). El NodeMCU tiene 13 pines de E/S llamados GPIO (Puerto General de Entrada y Salida) que pueden usarse de forma independiente como servidor web y enviar comandos remotos para controlar esos pines o activar dispositivos, para dar acceso a música digital. También incluye pines con capacidades de conversión A/D de 10 bits, PWM, I2C, SPI, UART. Se puede alimentar directamente a través de una conexión micro USB o mediante una fuente externa. La tensión de alimentación es de 5V, que se reduce a 3,3V mediante un regulador incorporado con una carga media de 100 mm. La capacidad actual aumenta durante la transmisión y el alojamiento Wi-Fi, por lo que se recomienda una capacidad de 300 mm si se alimenta desde una fuente externa (S.A, 2022).

Figura 6

Placa NodeMcu-Esp8266



Nota. La imagen ilustra el microcontrolador NodeMCU-ESP8266 es el dispositivo que procesa la información de entrada que es leída por los sensores de temperatura, sensores de humedad y envía una señal analógica o digital por medio de los pines de salida hacia los actuadores. " Explorando internet de las cosas basado en ESP8266: herramientas y

caso de estudio", por Elva Lilia Reynoso Jardón, Universidad Autónoma de Ciudad Juárez, Manuel Díaz H. No. 518-B Zona Pronaf Condominio, 32315 Cd Juárez, Chihuahua, México. E-mail: elva.reynoso@uacj.mx. ORCID: 0000-0002-0729-2822.

COMPARACION DE SENSORES DE HUMEDAD DEL SUELO

Para el proyecto del vivero se escoge dos tipos de humedad del ambiente y del suelo, en este apartado se compara diferentes tipos de sensores de humedad del suelo, la humedad del suelo se refiere a la cantidad de agua que contiene los espacios porosos del suelo.

Tabla 6

Comparación de sensores de Humedad del Suelo

Parámetro Técnico	FC-28	YL-69	Decagon 5TE
Tipo de medición	Resistiva	Resistiva	Capacitiva
Rango de humedad	0% - 100% (valor relativo)	0% - 100% (valor relativo)	0% - 100% VWC
Precisión	±5% RH	±5% RH	±1-2% VWC
Resolución	10 bits - ADC externo	10 bits- ADC externo	12 bits- salida digital
Voltaje de operación	3.3V - 5V DC	3.3V - 5V DC	3.6V - 15V DC

Corriente de operación	~5 mA	~5 mA	~25 mA
Salida	Analógica (0-1023) y digital (LM393)	Analógica (0-1023) y digital (LM393)	SDI-12 / RS-232 / Analógica
Frecuencia de muestreo	10-50 Hz	10-50 Hz	Hasta 1 Hz
Resistencia al agua/corrosión	Baja (metales expuestos)	Baja (metales expuestos)	Alta (sellado)
Calibración	Manual	Manual	Calibrado de fábrica
Interferencia ambiental	Alta (afectado por sales/minerales)	Alta (afectado por sales/minerales)	Baja
Temperatura de operación	0°C a 60°C	0°C a 60°C	-40°C a 60°C
Durabilidad	3-6 meses	3-6 meses	5-7 años
Costo aproximado	\$2 - \$5 USD	\$2 - \$5 USD	\$150 - \$300 USD

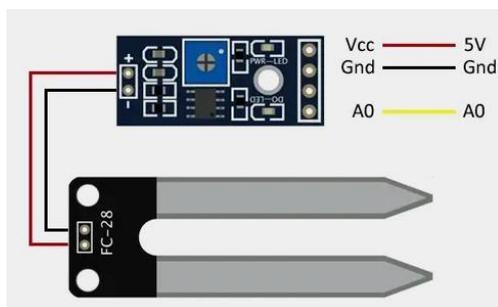
Se puede observar que el FC-28 tiene un tipo de medición resistiva, un rango de humedad de 0% a 100%, una precisión de $\pm 5\%$ RH y una salida analógica y digital, es uno de los más accesibles y compatibles con las diferentes placas de desarrollo, por cuestiones técnicas de análisis se escoge este sensor y se detalla además las características que posee este dispositivo

1.2.16 SENSOR DE HUMEDAD FC-28

El sensor de humedad es un captador de mediciones de voltaje, el proceso en sí que al integrarse con el sustrato y el agua de la plantación de tomate produce conductividad en este caso señales en milivoltios que son recibidas por el microcontrolador nodemcu-esp8266

Figura 7

Sensor de Humedad del Suelo FC-28



Nota. La imagen ilustra el sensor de humedad del suelo FC-28, la conexión de entrada de la señal analógica y la alimentación del dispositivo Llamas, L. (s.f). Sensor de humedad de suelo FC-28 en Arduino. Luis Llamas. <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>

COMPARACIÓN DE SENSORES DE HUMEDAD Y TEMPERATURA DEL AMBIENTE

Tabla 7

Comparación de Sensores de Humedad y Temperatura del Ambiente

Sensor	DHT11	DS18B20	BME280
Medidas	Temperatura Humedad	Temperatura	Temperatura Humedad Presión
Protocolo de comunicación	Un cable	Un cable	Interfaz de comunicación I2C
Tension de alimentación	3 a 5,5 V CC	3 a 5,5 V CC	1,7 a 3,6 V (para el chip) 3,3 a 5 V para la placa
Rango de temperatura	0 a 50°C	-55 a 125°C	-40 a 85°C
Exactitud	+/- 2°C (de 0 a 50°C)	+/-0,5°C (entre -10 y 85°C)	+/-0,5°C (a 25°C)
Soporte (IDE de Arduino)	Biblioteca DHT de Adafruit	Temperatura de Dallas OneWire	Biblioteca BME280 de Adafruit
Soporte (MicroPython)	Módulo dht	Módulo ds18b20	Biblioteca Adafruit BME280

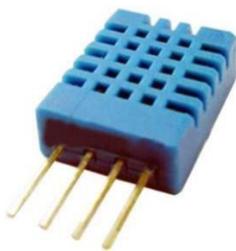
La humedad del ambiente se precisa al vapor de agua incidente en el aire, por consiguiente, observando el rango de temperatura y el soporte se escogió el Sensor de Humedad del Ambiente DHT11

1.2.17. SENSOR DE TEMPERATURA Y HUMEDAD AMBIENTE DHT11

El sensor de temperatura y humedad DHT11 integra un sistema avanzado de medición que proporciona una señal digital calibrada de manera precisa. Empleando una técnica exclusiva para la adquisición de señales digitales junto con tecnología de detección de temperatura y humedad, garantiza una alta confiabilidad y una excelente estabilidad a largo plazo. El dispositivo incluye un sensor de humedad de tipo resistivo y un sensor de temperatura NTC, los cuales están conectados a un microcontrolador de 8 bits de alto rendimiento. Esto asegura una respuesta rápida, una gran resistencia a las interferencias y un excelente equilibrio entre calidad y costo.

Figura 8

Sensor de Humedad y Temperatura Ambiental



Nota: Esta imagen visualiza la forma del sensor de temperatura y humedad ambiente. OSEPP Electronic. (s.f.). DHT11 technical data sheet (Versión traducida). Mouser Electronics.

1.2.18. CONTROLADOR DE RIEGO INALÁMBRICO INTELIGENTE TUYA SMART

Es un controlador que permite abrir y cerrar una válvula de riego que es utilizada remota o local dependiendo de la condición establecida en nuestra propuesta tecnologica,funciona con alimentación DC con cuatro pilas AA de 3.3 voltios está diseñado con el protocolo de protección IP67,ademas tiene control de voz inteligente que permite controlar por medio de un temporizador de riego el Agua, es util para el hogar, jardinería y muchas otras utilidades.

Figura 9

Controlador de riego Tuya Smart



Nota: Esta imagen muestra las dimensiones del controlador de riego, el indicador de prendido y el pulsador on/off que permite activar o desactivar de acuerdo a la funcion.

FrankEver. (s.f.). Controlador de riego inteligente inalámbrico Tuya Smart

[Imagen]. Made in China. https://es.made-in-china.com/co_hzfrankever/product_Smart-Life-Tuya-Intelligent-Wireless-WiFi-Water-Timer-Sprinklers_yuuyiguorg.html?pv_id=1i83n1kdr3c5&faw_id=1i83n2mbp7db

1.2.19 ROUTER INALÁMBRICO N TPLINK

El router inalámbrico N TP-Link TL-WR841N ofrece una velocidad Wi-Fi de hasta 300 Mbps, ideal para aplicaciones que requieren un alto ancho de banda, como la transmisión de video en alta definición y videoconferencias. Su rendimiento se debe a la tecnología

2x2 MIMO (Multiple Input, Multiple Output), que mejora tanto la fiabilidad como el alcance de la señal inalámbrica. Las dos antenas externas de 5 dBi aumentan la cobertura al mejorar la transmisión y recepción de señales, lo que garantiza una conectividad más estable en áreas amplias. Está diseñado para entornos con múltiples dispositivos conectados, proporcionando un rendimiento estable.

Figura 10

Router TP-LINK TL-WR841N



Nota: Esta imagen ilustra el tipo de comunicación entre dispositivos con respecto a la propuesta de invernaderos de tomate, siendo el dispositivo móvil o la placas iot esp8266 como el cliente. TP-Link. (s.f.). TL-WR841N 300Mbps Wireless N Router. TP-Link. <https://www.tp-link.com/es/home-networking/wifi-router/tl-wr841n/>

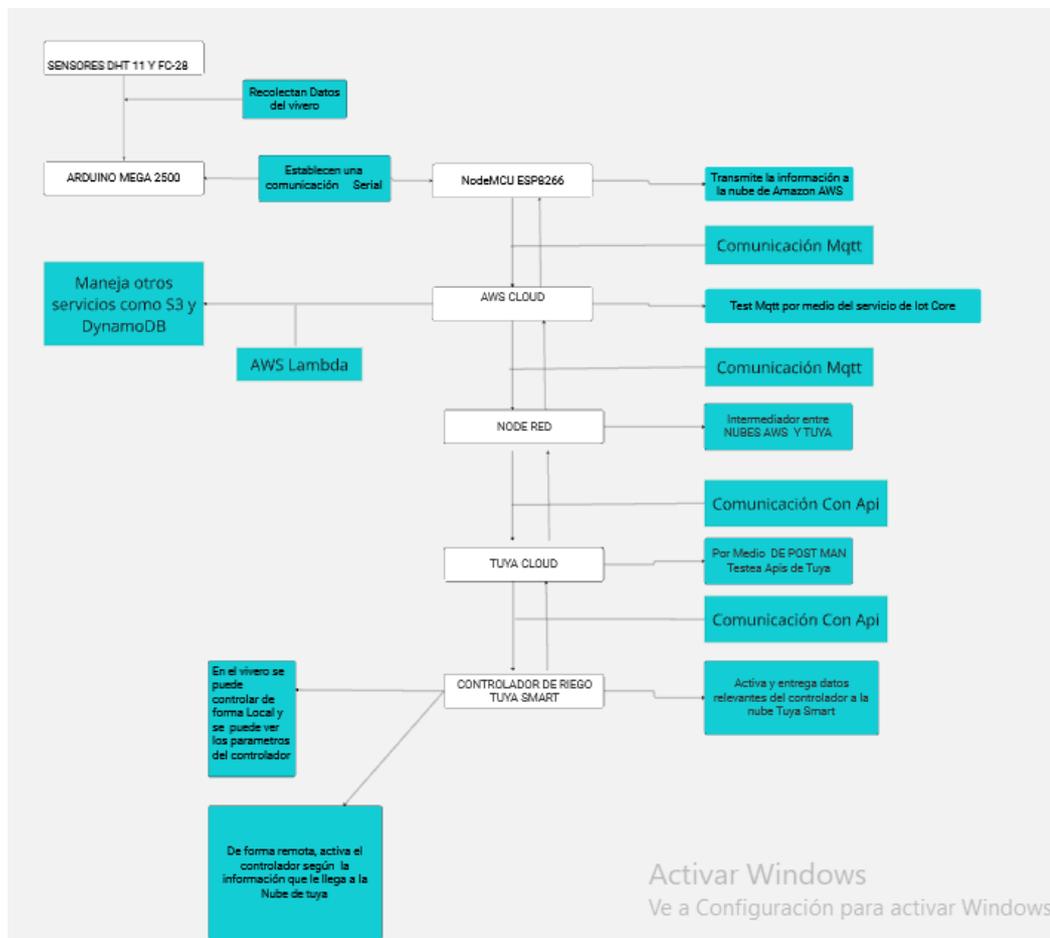
CAPÍTULO 2. METODOLOGÍA

2.1 DIAGRAMA GENERAL DEL PROYECTO

A continuación, se muestra el diagrama de bloques que describe la estructura general del sistema de riego de goteo localizado por medio de sensores IoT, que describe la interacción entre el sensor, el sistema de control y el mecanismo de riego

Figura 11

Diagrama General del Proyecto, muestra la conectividad entre los diferentes elementos de la propuesta



DESCRIPCION DEL PROCESO:

El sistema de riego automático basado en IoT se estructura en varios niveles de adquisición, procesamiento y control de datos. En la etapa de sensado, se emplean módulos DHT11 para medir la temperatura y humedad ambiental, junto con sensores FC-28 para evaluar la humedad del suelo. Estos dispositivos están conectados a un Arduino Mega 2560, que centraliza la captura de datos y los envía mediante comunicación serial a un NodeMCU ESP8266, el cual actúa como pasarela de datos hacia la nube.

La transmisión de información se realiza a través del protocolo MQTT, optimizado para entornos de IoT debido a su bajo consumo de ancho de banda y eficiencia en la entrega de mensajes. Los datos son enviados a AWS Cloud, donde son almacenados en bases de datos distribuidas y gestionados mediante AWS Lambda, permitiendo su integración con otros servicios como Amazon S3 y DynamoDB para análisis y almacenamiento histórico.

Para la interoperabilidad con plataformas de terceros, se implementa Node-RED como middleware, facilitando la conexión entre AWS Cloud y Tuya Cloud. La comunicación con Tuya Smart se establece mediante API REST, utilizando métodos GET y POST para la actualización y consulta de parámetros en tiempo real.

El controlador de riego, basado en la plataforma Tuya Smart, recibe los datos procesados y toma decisiones en función de umbrales predefinidos. Cuando la humedad del suelo cae por debajo de un nivel crítico, el sistema activa automáticamente la válvula de riego y registra la operación en la nube para auditoría y análisis posterior.

Desde la aplicación móvil de Tuya Smart, los usuarios pueden supervisar el estado del sistema, modificar parámetros de operación y activar el riego de manera manual si es necesario. La arquitectura distribuida del sistema garantiza una operación

robusta y escalable, optimizando el consumo de agua y mejorando la eficiencia del proceso de irrigación agrícola.

2.2 CONTEXTO DE LA INVESTIGACIÓN

La investigación se realizará en un entorno controlado en la Península de Santa Elena en la manzana M con coordenadas 2°14'06.1"S una latitud de 2 grados hacia el hemisferio sur y coordenadas de 80°52'11.5"W una longitud de 80 grados hacia el hemisferio oeste, de modo cuantitativo-experimental, en el que se busca medir variables específicas como humedad del ambiente, temperatura ambiente y humedad del suelo que se ve comprometido las plantas de tomate de riñón, para así aprovechar el recurso de líquido vital usando tecnificación como es el riego por goteo localizado.

Este enfoque permite un análisis preciso de los datos obtenidos de sensores y dispositivos IoT integrados en el sistema de riego, garantizando objetividad y confiabilidad en la interpretación de los resultados.

2.3 JUSTIFICACIÓN DEL MÉTODO:

Un enfoque cuantitativo es adecuado porque puede medir variables clave como la humedad del suelo, el uso del agua y el rendimiento.

Se eligió el enfoque experimental porque era necesario probar el sistema en condiciones controladas, ajustar las variables de riego y analizar sus efectos sobre el crecimiento de las plantas de tomate.

2.4 POBLACIÓN Y MUESTRA

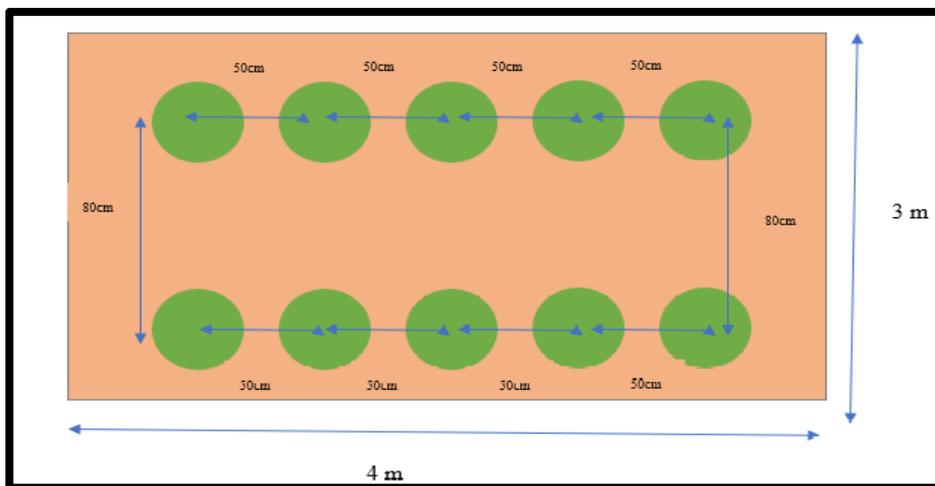
El grupo objetivo son los cultivos de tomate en invernadero. Sin embargo, en este estudio utilizaremos un prototipo de invernadero a una escala que replicará el entorno típico y las condiciones de crecimiento de un invernadero comercial.

Se seleccionará y propagará una muestra de 10 plantas de tomate en un sistema de riego por goteo automatizado, el vivero está diseñado con dimensiones de 4 m de longitud por 3mts de ancho, con 2 filas de 5 plantas cada una, separadas 50 cm entre planta a planta de longitud y 80 cm de planta a planta de ancho.

DISEÑO DEL INVERNADERO

Figura 12

Diseño del Invernadero del Proyecto



Nota. Imagen de elaboración propia

2.5. ESQUEMA DE DISTRIBUCIÓN DE UN INVERNADERO

Mediante el uso de prototipos a escala, es posible simular condiciones reales en un invernadero más grande y al mismo tiempo mantener de manera precisa y eficiente el control sobre las variables experimentales.

Permite probar las capacidades técnicas de los sistemas de riego por goteo, ajustar parámetros como el volumen de agua, la frecuencia de riego y las condiciones

ambientales temperatura, suelo y humedad sin afectar recursos financieros o logísticos significativos.

Además, los prototipos a escala permiten una mayor flexibilidad en la experimentación, ya que se pueden realizar múltiples iteraciones de diseño, ajustes y pruebas en un período de tiempo relativamente corto. Esto reduce el riesgo de errores a gran escala y permite identificar y solucionar problemas potenciales antes de implementar todo el invernadero.

Otro aspecto importante es que la naturaleza controlada del prototipo promueve la repetibilidad del sistema. Los resultados obtenidos durante estas pruebas se pueden extrapolar a escenarios más amplios, lo que garantiza que el sistema no solo funcione de manera óptima a escala de prototipo, sino que también pueda adaptarse y ampliarse a invernaderos comerciales más grandes. Esta repetibilidad es esencial para verificar la robustez y confiabilidad del sistema y para garantizar que las condiciones y los resultados puedan repetirse en la misma configuración.

Finalmente, trabajar a esta escala proporciona una manera de evaluar y optimizar el uso de nuevas tecnologías como Internet de las cosas (IoT) sin requerir grandes inversiones iniciales. Esto es esencial para garantizar la viabilidad económica del sistema y demostrar su potencial como solución innovadora para la agricultura moderna.

2.6. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

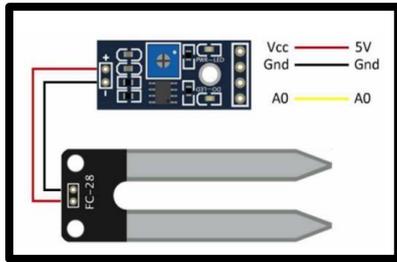
Sensores de humedad del suelo: Utilizados para medir la humedad en las zonas de raíces de las plantas, permitiendo activar o desactivar el riego automáticamente.

Para ello se necesita un sensor de humedad del suelo FC-28. El FC-28 es un sensor simple que mide la humedad del suelo cambiando su conductividad eléctrica. El FC-28 está equipado con una tarjeta de medición estándar que permite obtener un valor de medición

en forma de valor analógico o salida digital que se activa cuando la humedad supera un determinado umbral. Sensores de temperatura y humedad ambiental: Monitorizan las condiciones del invernadero, cruciales para el crecimiento del tomate.

Figura13

Sensor Fc-28 conexión



Nota. La imagen ilustra el sensor de humedad del suelo FC-28, la conexión de entrada de la señal analógica y la alimentación del dispositivo Llamas, L. (s.f.). Sensor de humedad de suelo FC-28 en Arduino. Luis Llamas. <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>

2.6.1. SENSORES DE HUMEDAD DEL AMBIENTE, Y TEMPERATURA AMBIENTE:

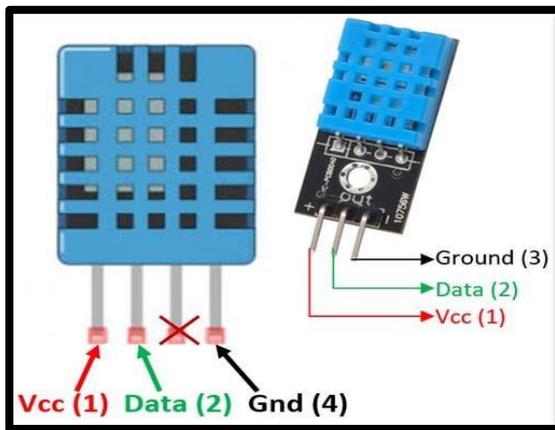
Los sensores utilizados para este proyecto son DHT11, Supervisan las condiciones del invernadero, que son esenciales para el crecimiento óptimo del tomate, ya que pueden ajustar los parámetros ambientales y mantenerlos en niveles ideales.

Estos sensores recopilan información importante como la temperatura, la humedad y la humedad del suelo, que son esenciales para proporcionar un ambiente controlado que promueva el crecimiento de las plantas.

Al detectar cambios en estas variables, el sistema puede iniciar automáticamente acciones correctivas como ajustar el riego, encender ventiladores o sistemas de calefacción para garantizar el crecimiento de las plantas en condiciones óptimas. Además, el seguimiento continuo permite prevenir situaciones desfavorables, como falta de agua o calor excesivo, que pueden perjudicar la salud y productividad de los cultivos de tomate.

Figura 14

Conexión Dht 11



Nota: En esta imagen se ilustra los pines de conexión de alimentación y el pin que hace referencia a la señal. Aprendiendo Arduino. (s.f.). DHT11. Aprendiendo Arduino.

<https://aprendiendoarduino.wordpress.com/tag/dht11/>

Tabla 8

ESPECIFICACIÓN DEL SENSOR DE TEMPERATURA Y HUMEDAD DHT11

CATEGORÍA	ESPECIFICACIÓN
Alimentación	3.3 ~ 5 V
Rango de medición de temperatura	0 ~ 50 °C
Precisión de medición de temperatura	± 2.0 °C
Resolución Temperatura	0.1 °C
Rango de medición de humedad	20% ~ 90% RH
Precisión de medición de humedad	4 % RH
Resolución Humedad	1 % RH
Tiempo de sensado	1 seg

Nota. Datos obtenidos de Components101 (s.f.).

Plataforma IoT (Tuya Smart, AWS IoT, Node-RED): Proporcionan la infraestructura para la monitorización remota y la automatización del sistema de riego, utilizando las plataformas IoT, se pueden usar dispositivos conectados como móviles inteligentes que generacionalmente evolucionan, tabletas u PC personales, en la industria ver desde un HMI, Sistema Scada para sustraer los datos recopilados por los sensores de humedad del suelo, temperatura ambiente, humedad ambiental alrededor del mundo haciendo más fácil monitorear el estado del invernadero, también recibir alertas automáticas si ciertas variables clave exceden ciertos límites. Si la humedad del suelo cae por debajo de un cierto umbral, el sistema enviará un mensaje o comenzará a regar automáticamente.

Estableciendo la seguridad de los datos la nube AWS IOT tiene una ventaja es que permite ver la trayectoria de los datos seguros en tiempo real, asegurando que la información crucial para el ámbito de riego en las condiciones ambientales puedan ser evaluadas, en cambio la nube de origen chino tuya Smart que cuenta con millones de usuarios alrededor del mundo, permite la integridad de los dispositivos IOT, asegurando una intercomunicación, asegurando la evolución de la cosas o things que van en un auge en el crecimiento para el servicio de la comunidad mundial.

Microcontrolador ESP8266: Conectará los sensores y actuadores para la toma de decisiones en tiempo real sobre el riego.

NodeMCU ESP8266 es una plataforma de desarrollo orientada a Internet de las cosas (IoT) similar a Arduino basada en el SoC Wi-Fi ESP8266. La placa NodeMCU v2 ESP8266 integra un conversor CP2102 de USB a TTL serie y un conector micro-USB para programación y comunicación.

El GPIO funciona a 3,3 V, por lo que se requiere un cambiador de nivel para usarlo con sistemas de 5 V. NodeMCU incluye firmware que permite el uso del idioma traducido

LUA y también es compatible con otros lenguajes como Arduino, MicroPython y C/C a través del entorno Arduino.

El ESP8266 de Espressif Systems es un microcontrolador de 32 bits con conectividad Wi-Fi ideal para proyectos de IoT.

Las placas de desarrollo basadas en ESP8266, como NodeMCU, Wemos D1 mini y ESP-01, se utilizan ampliamente en proyectos de IoT debido a su flexibilidad y apoyo de la comunidad.

Tabla 9

Especificaciones del NodeMCU V2 ESP8266

CATEGORÍA	ESPECIFICACIÓN
Voltaje de Alimentación	5V DC
Voltaje de Entradas/Salidas	3.3V DC (No usar 5V)
Frecuencia de Reloj	80MHz/160MHz
Instrucción RAM	32KB
Data RAM	96KB
Memoria Flash Externa	4MB
Pines Digitales GPIO	17 (4 pueden configurarse como PWM a 3.3V)
Pin Analógico ADC	1 (0-1V)
Pulsador RESET y FLASH	Sí
Puerto Serial UART	2

Nota. Datos obtenidos de Naylamp Mechatronics (s.f.).

2.6.2. Controlador de riego TUYA SMART

El Temporizador de Riego por Goteo es un dispositivo inteligente controlado por Bluetooth 5.0 que permite programar el riego de forma remota y ajustar fácilmente el cronograma mediante una aplicación móvil en un rango de 15 metros. Tiene múltiples funciones, como riego automático, riego manual, retardo de lluvia, etc., y se puede ajustar según la temperatura del suelo y las condiciones de humedad. El temporizador requiere un concentrador Bluetooth para un control remoto y un historial de riego más estables en la aplicación Tuya Smart.

Construido con materiales de alta resistencia para evitar fugas, el dispositivo puede soportar una presión de agua de hasta 116 PSI y tiene una clasificación de resistencia al agua IP67.

Ideal para sistemas de riego por goteo, aspersores y barriles de lluvia, es fácil de instalar sin herramientas.

Funciona con 4 pilas AA y puede funcionar durante 24 meses, ahorrando agua y energía.

Figura 15

Controlador de Riego Tuya Smart



Nota: La imagen muestra el control de monitoreo local de tuya Smart Hangzhou Frankever Electronic Co., Ltd. (s.f.). *Tuya smart home automatic intelligent garden WiFi drip watering system smart water timer sprinkler irrigation pump controller:* Recuperado de <https://es.made-in->

china.com/co_hzfrankever/product_Tuya-Smart-Home-Automatic-Intelligent-Garden-WiFi-Drip-Watering-System-Smart-Water-Timer-Sprinkler-Irrigation-Pump-Controller_yuserooogg.html

2.6.3 FUENTE DE PODER INS DE 850W GENERICA

Para la alimentación de los dispositivos iot ESP83266 se realizó el uso de una fuente de poder de Pc, DE 850W GENERICA, basados en los principios en que las fuentes tienen diferentes tipos de voltaje y amperaje y es idóneo para microprocesadores de baja

Figura 16

Fuente de Poder Ins de 850W

potencia, y por la siguiente característica, que se enmarcan a continuación:



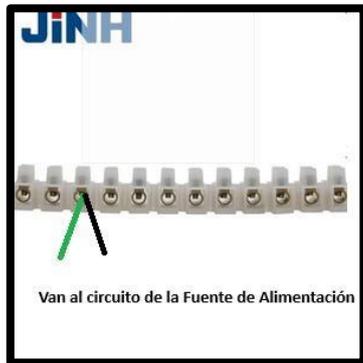
Nota: Esta imagen hace referencia a la fuente de alimentación para los dispositivos iot. TECNIT. (2024). Fuente de poder ATX INS P4-800W de 800W conectores SATA/IDE 38A TECNIT. <https://www.tecnit.com.ec/fuente-de-poder-atx-p4-800w>

2.6.4. BLOQUE DE TERMINALES JINH

Para ordenar los diferentes tipos de voltaje para los dispositivos iot se utilizó bloque de terminales, y cerrar el circuito para que funcione la fuente de alimentación se unió el cable verde con un cable negro.

Figura 17

Bloques de Terminales Jinh



Nota: En esta imagen se observa el bloque terminales. CNJINH. (s.f.). 80A 35mm PE plastic terminal block. CNJINH. <http://es.cnjinh.com/plastic-terminal/80a-35mm-pe-plastic-terminal-block.html>

2.6.5. CONSUMO DE LOS SENSORES

- DHT11: consume aproximadamente 0.3mA cuando está inactivo y hasta 2.5mA durante la medición.
- FC-28: consume alrededor de 3-5mA.

2.6.6. SENSORES FC-28 (10 UNIDADES)

Consumo por sensor: 20 mA

Consumo total (10 sensores): $10 \times 20 \text{ mA} = 200 \text{ mA}$

2.6.7. SENSORES DHT11 (10 UNIDADES)

Consumo por sensor: 2.5 mA (en funcionamiento) Consumo total (10 sensores): $10 \times 2.5 \text{ mA} = 25 \text{ mA}$

2.6.8. PLACA ARDUINO MEGA 2560

Consumo: 70-80 mA

Consumo máximo con periféricos: ~200 mA (Depende de los pines activos y cargas adicionales)

2.6.9. PLACA ESP8266

Consumo en funcionamiento: ~70-250 mA Consumo en picos (transmisión Wi-Fi): ~400 mA Total del consumo del sistema

2.6.10. SUMANDO EL CONSUMO DE TODOS LOS DISPOSITIVOS:

FC-28: 200 mA

DHT11: 25 mA

Arduino Mega: ~200 mA ESP8266 (pico): ~400 mA Consumo total máximo:

$$200 + 25 + 200 + 400 = 825 \text{ mA}$$

2.6.11. CÁLCULO DE POTENCIA EN VATIOS (W)

Si alimentas todo el sistema a 5V (común en Arduino y ESP8266):

$$P=V \times I=5V \times 0.825A=4.125 \text{ W}$$

$$P=V \times I=5V \times 0.825A=4.125W$$

Por lo tanto, el sistema completo consume aproximadamente 4.1 W.

2.7. ESQUEMA DE LA PROPUESTA

Para integrar los dispositivos a la propuesta de implementación de riego por goteo basado en iot, se realizaron los siguientes pasos:

2.7.1. CONFIGURACIÓN DE TUYA SMART:

- Se Registra dispositivos en la plataforma de Tuya.
- Se Configura la API de Tuya para acceso desde Node-RED.

2.7.2. INSTALACIÓN DE NODE-RED:

- Se Configura los flujos en Node-RED para procesar datos de los dispositivos Tuya.
- Se Integra nodos MQTT para conectarse con AWS IoT.

2.7.3. CONFIGURACIÓN DE AWS IOT:

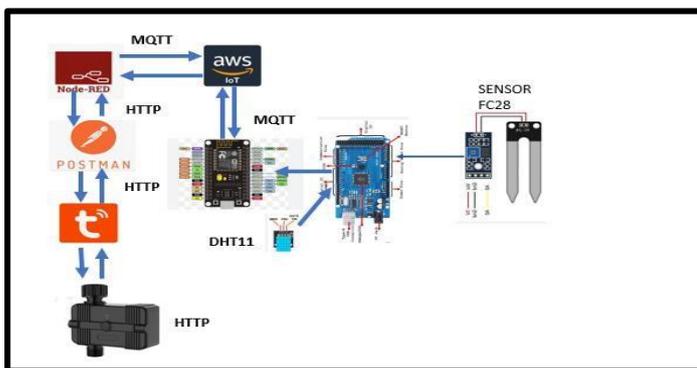
- Se Crea dispositivos y políticas en AWS IoT.
- Se usa MQTT para enviar y recibir datos de Node-RED a AWS IoT.

2.7.4. INTEGRACIÓN Y PRUEBAS:

- Se confirma que los flujos de datos entre Tuya, Node-RED, y AWS IoT sean seguros y funcionales

Figura 18

Conexión de Comunicación de la Propuesta



Nota: Diseño de la arquitectura entre las nubes AWS IOT y Tuya Smart. Elaboración propia

CAPITULO 3 RESULTADOS Y DISCUSIONES

3.1. INTRODUCCIÓN

Este capítulo presenta los resultados de implementar un sistema de riego por goteo basado en Internet de las cosas en un prototipo de invernadero de tomates. Los datos recopilados durante el experimento se analizarán para evaluar el funcionamiento del sistema en términos de riego preciso y uso eficiente del agua. Además, se discutirá cómo los resultados alcanzan los objetivos propuestos y qué aspectos del sistema podrían mejorarse en futuros desarrollos.

3.2. DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE RIEGO DE GOTEO BASADO EN IOT

Los sensores de humedad del suelo, temperatura y humedad ambiental, controlados por plataformas de IoT, en este caso Tuya Smart y AWS IoT, se utilizaron para implementar el sistema de riego automatizado. Se utilizó una válvula inteligente de riego conectada al suministro durante el proceso, que se activaron automáticamente según los valores que recibieron los sensores.

Se creó un modelo prototipo de un invernadero de tomates en el que se monitoreaban continuamente diez plantas. Los datos de humedad del suelo fueron esenciales para iniciar el sistema de riego. Cuando el nivel de humedad del suelo caía por debajo de, el sistema se programó para iniciar el riego.

3.2.1 CREACIÓN DE LA NUBE TUYA SMART PARA LA VÁLVULA DE RIEGO

Un paso importante en la implementación del sistema de riego automatizado basado en Internet de las cosas es la creación de la nube para el control de la válvula de riego en la plataforma Tuya Smart. Este proceso implica la creación de un entorno en la nube que permita la integración de dispositivos IoT, como la válvula de riego, para ser monitoreados y controlados remotamente a través de Internet. Los pasos principales para la creación de la nube en Tuya Smart se describen a continuación.

Paso 1: Registrarse y configurar su plataforma

Para empezar, debe iniciar sesión en Tuya IoT Platform. Para administrar dispositivos inteligentes y desarrollar aplicaciones personalizadas para el control remoto de dispositivos IoT a través de esta plataforma.

Figura 19

Log in with Tuya Account

Log in with Tuya Account

Tuya Account

jchamba13@yahoo.com

Password [Forgot Password?](#)

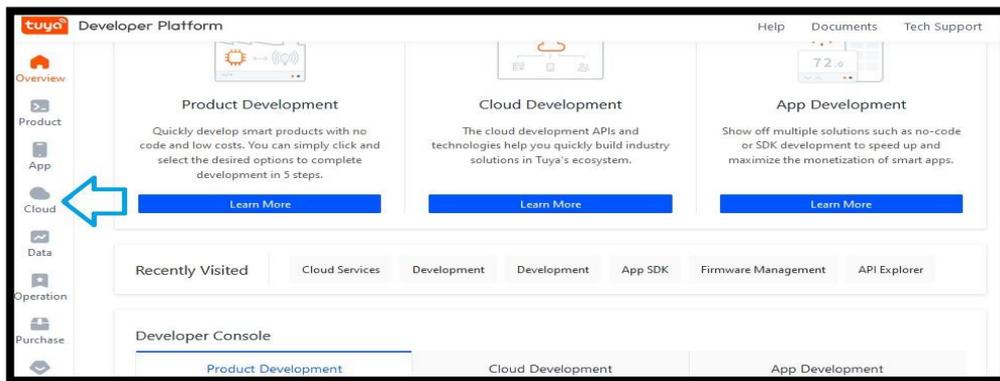
Log in

[Sign Up](#)

Después del proceso de registro, se crea el proyecto IoT: se puede crear un nuevo proyecto en la plataforma asignándole un nombre y una categoría específica, por siguiente se elije el menú Cloud que hace referencia a la Nube de Tuya Smart

Figura 20

Developer Platform



Seleccionar el entorno de desarrollo: en este paso, se elige el entorno de desarrollo en la nube y se configuran las regiones donde estará disponible el servicio. Idealmente, esto sería la región cercana a la ubicación del invernadero, en este caso la plataforma de Tuya Smart, señala en su apartado que el data center asignado para Ecuador es el DataCenter, Wester

Sabiendo las modificaciones necesarias como el correo validado, la región con el datacenter más cercano de la nube Tuya Smart, se procede a configurar el dispositivo.

Figura 21

Elección del Datacenter para Tuya Smart Cloud según la región

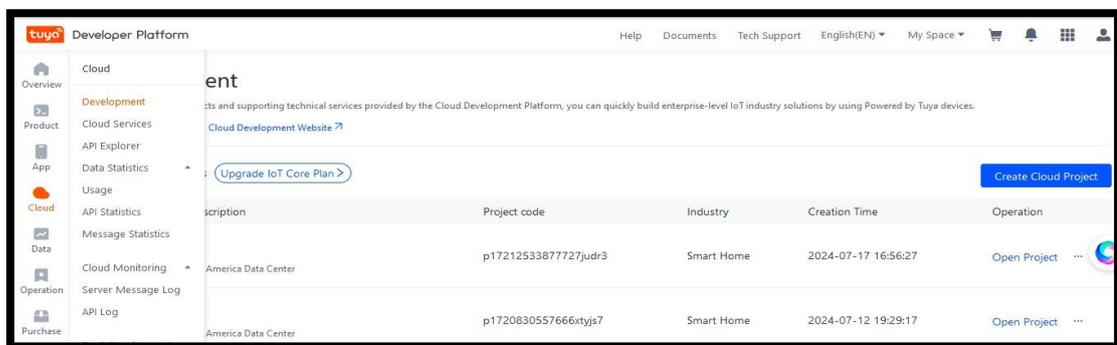
Western America Data Center					
Country code	Country/region	Country code	Country/region	Country code	Country/region
1	United States	84	Vietnam	682	Cook Islands
1	Canada	95	Myanmar	683	Niue
51	Peru	239	São Tomé and Príncipe	686	Kiribati
52	Mexico	245	Guinea-Bissau	690	Tokelau
54	Argentina	502	Guatemala	852	Hong Kong SAR, China
55	Brazil	591	Bolivia	853	Macao SAR, China
56	Chile	593	Ecuador	886	Taiwan, China

Paso 2: Instalación del dispositivo válvula de irrigación

La integración de la válvula de riego como dispositivo IoT es el siguiente paso después de crear el proyecto en Tuya Smart. Para lograr esto, la válvula física debe conectarse a un módulo de comunicación compatible con la plataforma, como un módulo Wi-Fi o ZigBee. Para la integración se va a la plataforma Tuya en el apartado Cloud dentro de la plataforma se da clic, se prosigue a buscar Development que es la parte para el desarrollador.

Figura 22

Cloud - Development



Por consiguiente, se crea un proyecto, con el nombre riego, y se ponen todas las configuraciones necesarias.

En todos, Overview, nos muestra un resumen del Access ID/ Client ID, Access Secret y el nombre del código del Proyecto.

Figura 23

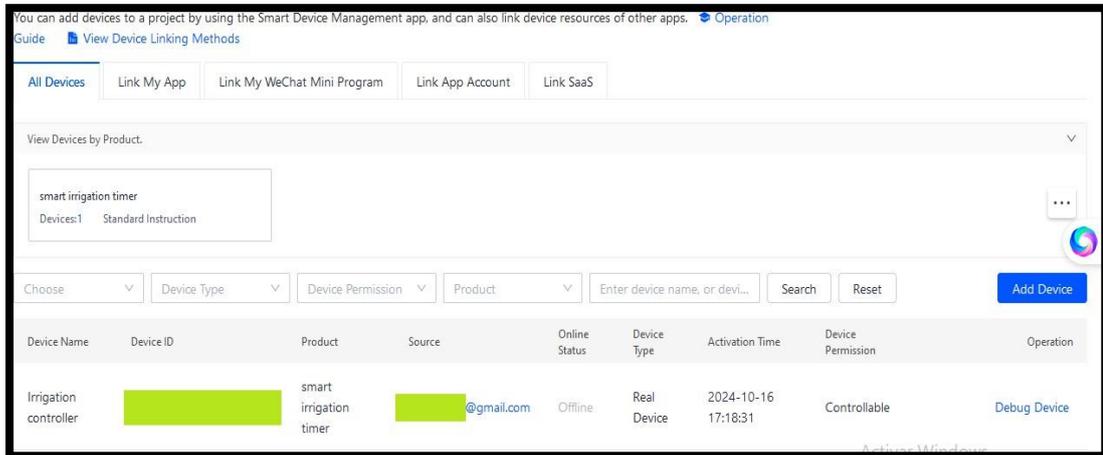
Overview



Luego nos dirigimos a Devies o Dispositivos, y creamos un nuevo dispositivo en la nube

Figura 24

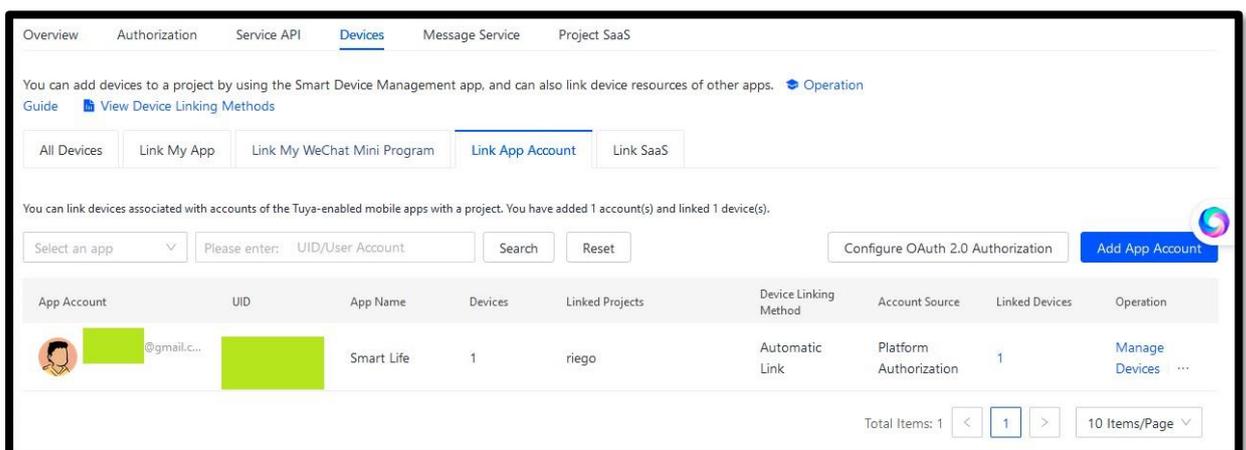
Devies o Dispositivos



En la plataforma móvil de tuya Smart life se crea el usuario que se enlaza con la estructura en la nube de tuya Smart tuya o Smart Life, para ello en la nube se va a la opción de link de cuentas.

Figura 25

Devices Link App Account



Como se puede observar la aplicación en Smart Life ha sido creado en el móvil, esta se direcciona por medio del correo electrónico.

Figura 26

Logo en el celular de Smart Life



Paso 3: Configurar el control de la válvula.

Es posible configurar su control automatizado una vez que el dispositivo está registrado en la plataforma y conectado a la red. La funcionalidad de escenas de Tuya en este caso permite la creación de reglas automatizadas basadas en las lecturas de los sensores de humedad del suelo.

Paso 4: Seguimiento y Control Remoto

La válvula de riego se puede monitorear y controlar remotamente desde cualquier dispositivo conectado a Internet una vez que se ha configurado el sistema en la nube. Los usuarios pueden ver el estado de la válvula en tiempo real, verificar las condiciones del suelo y realizar ajustes manuales en caso de ser necesario con la aplicación de Tuya Smart. Notificaciones en tiempo real: Si el sistema detecta alguna anomalía, como una falla en el dispositivo o un problema con el sistema de riego, el sistema también puede enviar alertas o notificaciones al usuario.

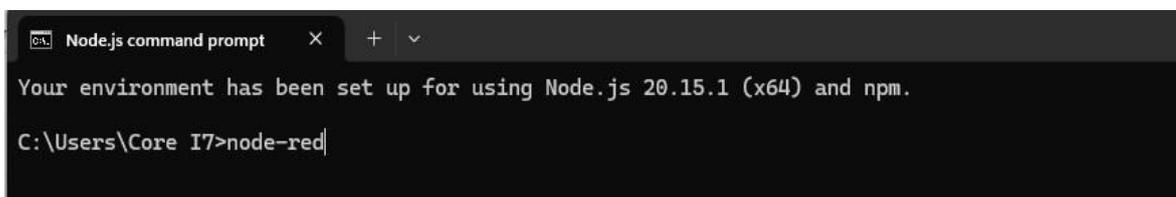
Paso 5: Conectarse a Node-RED y AWS IoT

La plataforma Tuya permite la integración con otras herramientas, como Node-RED y AWS IoT, para crear un sistema con más funcionalidad.

Para conectarse a node Red, primero hay que descargar la versión ultima de node js, luego de instalarla, se escribe en el comando de ejecución, node-red

Figura 27

Command prompt para ejecutar node red

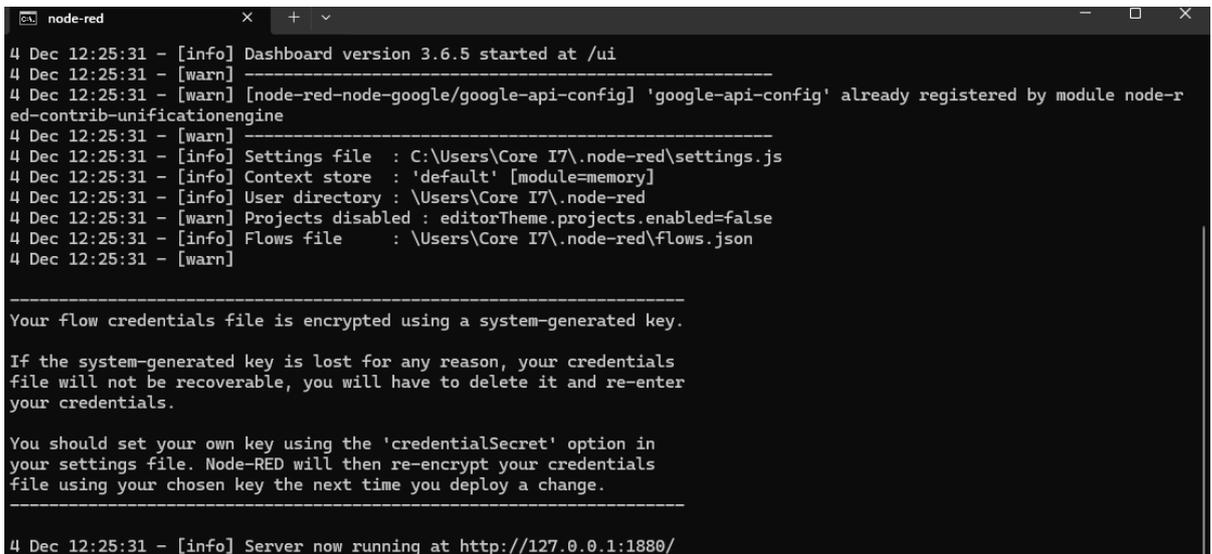


```
Node.js command prompt x + v
Your environment has been set up for using Node.js 20.15.1 (x64) and npm.
C:\Users\Core I7>node-red
```

Luego de dar clic sobre la tecla Enter, se desplega información sobre la dirección local a la cual se va a acceder para desarrollar el programa en Node red.

Figura 28

Se Designa la url para ejecutar node red



```
node-red x + v
4 Dec 12:25:31 - [info] Dashboard version 3.6.5 started at /ui
4 Dec 12:25:31 - [warn] -----
4 Dec 12:25:31 - [warn] [node-red-node-google/google-api-config] 'google-api-config' already registered by module node-r
ed-contrib-unificationengine
4 Dec 12:25:31 - [warn] -----
4 Dec 12:25:31 - [info] Settings file : C:\Users\Core I7\.node-red\settings.js
4 Dec 12:25:31 - [info] Context store : 'default' [module=memory]
4 Dec 12:25:31 - [info] User directory : \Users\Core I7\.node-red
4 Dec 12:25:31 - [warn] Projects disabled : editorTheme.projects.enabled=false
4 Dec 12:25:31 - [info] Flows file : \Users\Core I7\.node-red\flows.json
4 Dec 12:25:31 - [warn] -----

Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.

-----
4 Dec 12:25:31 - [info] Server now running at http://127.0.0.1:1880/
```

Luego dar clic en la dirección `http: x.x.x.x`, en este caso para la aplicación es la `127.0.0.1` ubicada en el puerto `1880`, entonces se abre un cuadro de dialogo en el cual va estar algunos nodos y los que no están, hay que instalarlo en el administrador de aplicaciones en el lado de derecho.

Para crear el acceso por medio del Access token, se necesita los siguientes pasos:

1. Variables iniciales donde conste la identidad del cliente
2. Configuración de los parámetros de comunicación para obtener la información en este caso el método GET
3. Un archivo vacío, en general un archivo hash
4. Construcción de la cadena para firma en este caso la nube de Tuya, necesita 4 cosas método GET para solicitar la información, hash del cuerpo, un campo vacion entre saltos de línea y la url de la solicitud.
5. Luego se genera la cadena de firma final que esta está constituida por medio del cliente, el tiempo de referencia y la cadena con los 4 pasos anteriormente descrito.
6. Por último se guarda la cadena de firma final, y el tiempo de referencia por ello el `msg.payload` se le asigna la firma de la cadena final, y en `msg.time` la cadena de tiempo actual para ser utilizado en el próximo bloque.

Se encripta la clave secreta que en este caso es la cadena de firma o solicitud, con un algoritmo SHA256.

Luego prosigue a configurarse la solicitud `http`, configurada para obtener un token de acceso, y estos paso están dado por:

1. Configuración de la url
2. Asignación de la marca de tiempo y la firma de la cadena,
3. Obtención del cliente

4. Configuración de los encabezados de la solicitud
5. Configuración del cuerpo de la solicitud en este caso vacía por se declaro directamente en la url
6. Definición de la Url y el método

Luego se envia al bloque de peticiones http requests, después se extrae el token con las siguientes características.

- 1.- Obtención de la respuesta
- 2.- Extracción de los tokens
- 3.- Creación de un objeto para almacenar las credenciales
- 4.- Almacenamiento en el contexto del flujo
- 5.- Luego retorna el mensaje

Luego de haber realizado los bloques de access token, se realiza un token para que calcula de diferencia de tiempo esto es el refresh token, sas la primera programación para obtener:

- access_token
- refresh_token

Cada una de programación que está en el bloque de la función se envía hacia un bloque http request que es el que pide la petición al servidor de la nube tuya , en la siguiente figura se puede apreciar de una forma más clara.

Figura 29

Respuesta del dispositivo



3.2.2 IMPLEMENTACIÓN DE UN SISTEMA DE RIEGO POR GOTEO BASADO EN IOT PARA UN PROTOTIPO DE INVERNADERO DE TOMATES A TRAVÉS DE AWS IOT

Un paso importante para integrar el control y la monitorización de los dispositivos en la nube de Amazon Web Services (AWS) es la implementación de un sistema de riego por goteo basado en IoT a través de AWS IoT. La plataforma sólida de AWS IoT permite la conexión de dispositivos, la gestión de datos en tiempo real y la automatización de procesos, como la gestión de la válvula de riego en un invernadero de tomates. La configuración y el uso de AWS IoT en este proyecto se describen a continuación.

Paso 1: Configuración inicial de AWS IoT y registro en AWS

El primer paso es crear una cuenta de AWS. Se puede acceder al servicio de IoT Core desde el panel de control una vez que se ha registrado. La recopilación de datos, el control de dispositivos y la conexión segura de dispositivos son posibles con AWS IoT Core.

Configuración del núcleo IoT: Se crea un entorno específico para la implementación del sistema de riego dentro de IoT Core. Aquí se crea un "objeto" dispositivo de Internet de las cosas que representará la válvula de riego en el sistema.

Paso 2: Establecer y registrar dispositivos IoT en AWS

Los dispositivos del sistema de riego se registran en AWS IoT Core. La válvula de riego, que se controlará de forma automática, es el dispositivo clave en este caso.

Crear un objeto en AWS IoT: se accede a la sección de gestión de AWS IoT Core y se selecciona la opción "crear objeto". Aquí se le da a la válvula de riego un nombre descriptivo.

Certificados de seguridad: se deben generar certificados de seguridad para que el dispositivo se comunique de manera segura con AWS IoT. Un certificado X.509 es proporcionado por AWS para autenticar y cifrar la comunicación entre dispositivos y la nube.

Políticas de permisos: Las políticas de permisos deben establecer qué puede hacer un dispositivo IoT. Por ejemplo, es necesario que la válvula tenga permisos para publicar y suscribirse a temas MQTT, lo que permite recibir comandos y enviar actualizaciones de estado.

Paso 3: Configurar la conexión de los dispositivos con AWS IoT

Se va a las consolas de AWS, luego de iniciar sesión en la cuenta se busca en el menú la palabra AWS IOT CORE.

Estando en el apartado de AWS IOT CORE en el lado izquierdo se escoge la opción crear certificados y políticas.

Para crear los certificados en AWS IOT

Después de crear la programación del dispositivo en la placa esp8266 y en el Arduino Mega, selecciona la opción para crear certificados.

Por consiguiente, los siguientes archivos para poder establecer la comunicación entre la placa esp8266 y la plataforma AWS IOT.

- Certificado del dispositivo (.pem.crt).
- Clave privada (.pem.key).
- Certificado raíz de Amazon (Amazon Root CA 1).

Para ello se va a panel de búsqueda de la aplicación de Aws - Iot Core y se busca en la parte izquierda

El menú de administración, se escoge objetos , luego crear objetos, crear un único objeto y se pone

Nombre de InvernaderoIot se presigue que no tenga sombra posteriormente siguiente, se debe recalcar que en lo posterior se crea el certificado, al finalizar en este

paso se crea el objeto listo para crear la política correspondiente y el certificado correspondiente.

CREAR POLITICA EN AWS IOT CORE

Para crear la Política se va a seguridad, se elige política se le da un nombre a la política, y en el documento de la política debe permitir toda acción que ejecute cualquier servicio en este caso, y al finalizar se pone crear

CREAR CERTIFICADO EN AWS IOT CORE

Para crear certificado se va uno a crear certificado, estado de certificado inactivo, luego se pone crear para luego descargar los archivos, y proseguir a activar los certificados, y al finalizar darle en atachar Politicaesp8266

Realizado el grupo administrador, grupo de usuarios en IAM, haber modificado los parámetros en AWS iot Core, se procede a seguir, verificar:

Crear una Cuenta AWS

Crear usuarios con acceso administrativo para cada recurso

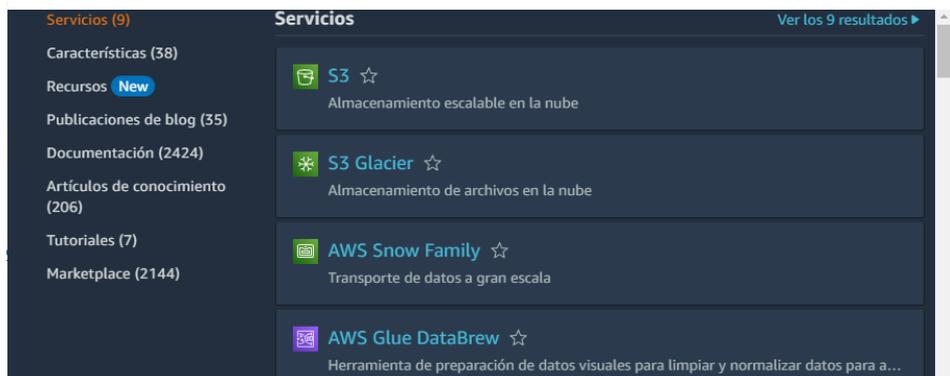
Crear políticas certificados, punto de acceso para acceder Iot Core

CREAR UN BUCKET DE AMAZON S3

Al abrir en un navegador de preferencia, en este caso Google Chrome, ya en el panel de usuario, se escribe el nombre de S3 en el panel en la barra de búsqueda, se da clic en balde este balde se denomina bucket

Figura 30

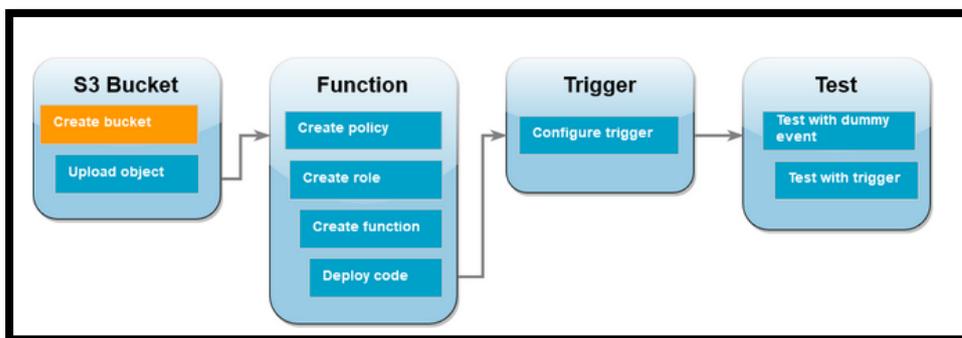
Portal para la elección de un bucket AWS S3



Al ingresar al bucket se despliega un cuadro de dialogo, en el se eligen los diferentes parámetros que son, el área de la región donde vamos a desplegar el bucket en este caso va a ser en la zona de Norte de Virginia (us-east-1), también hay que tener en cuenta que el bucket debe ser único en todo el mundo, y además contener únicamente de letras minúsculas, números, puntos (.) y guiones (-), los otros parámetros deben estar por defecto, y se usaran en secuencia para un mejor explicación del proceso.

Figura 31

Creación de un S3 Bucket para AWS Lambda



Nota: Amazon Web Services. (s.f.). *Creación de un S3 Bucket para AWS Lambda*. AWS Documentation. Recuperado el 12 de febrero de 2025, de https://docs.aws.amazon.com/es_es/lambda/latest/dg/with-s3-example.html

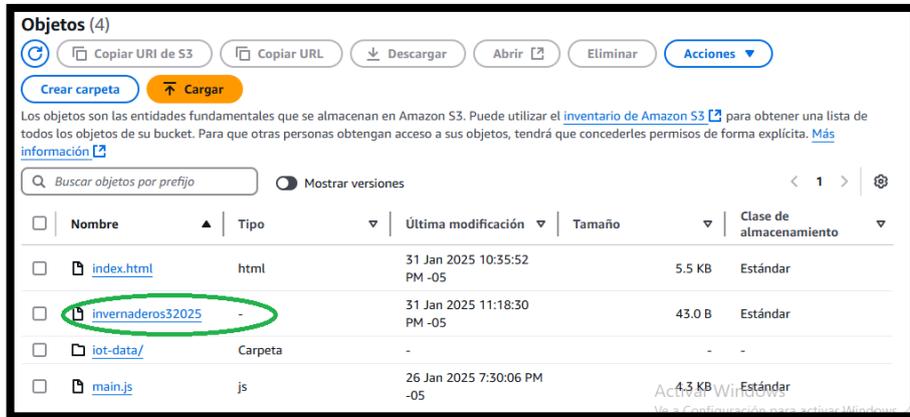
CARGAR UN OBJETO DE PRUEBA EN UN BUCKET

Se crea el bucket con el nombre de, y dentro del se cargan 4 objetos, 3 de ellos de forma local, y 1 enviado por medio de la información enviada por los sensores este archivo es sin extensión.

ARCHIVO SIN EXTENSION

Figura 32

Archivo sin extensión en S3

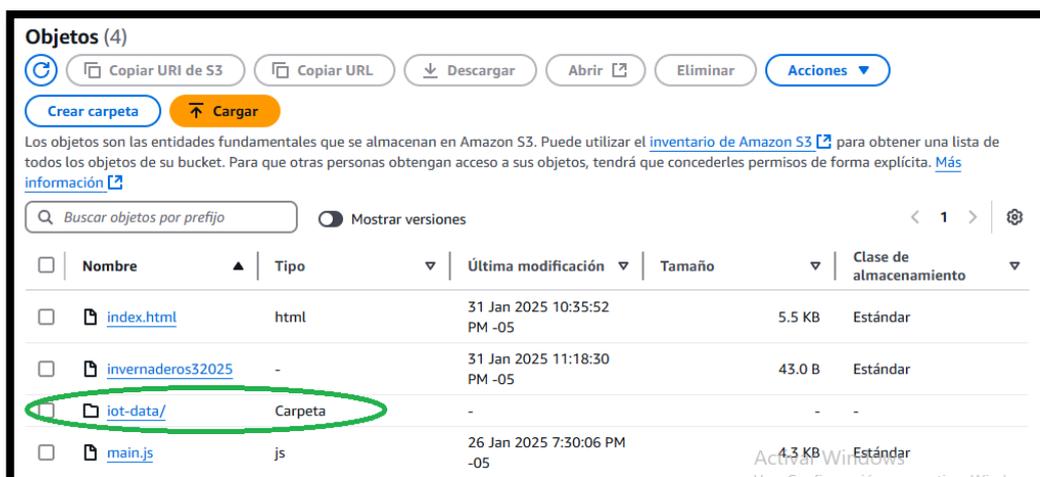


ARCHIVOS CON EXTENSION

En los archivos con extensión se tiene un archivo con extensión index.html, una carpeta llamada iot-data/donde se actualizará los archivos mediante el tiempo unix proporcionado por la programación, cada lectura que los sensores recoge va junto al tiempo.

Figura 33

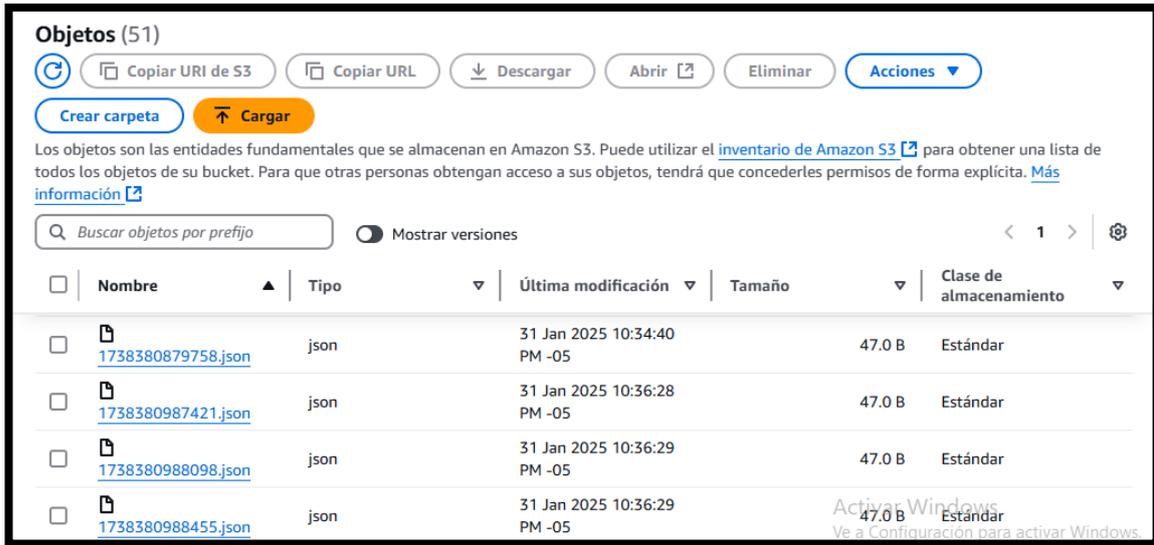
Archivo con extensión en S3



En la carpeta se va apilando los archivos uno en uno, esto es debido a la declaración de una de las reglas en AWS IOT CORE, `iot-data/${timestamp()}.jso`

Figura 34

Verificación de los objetos según la regla de AWS IoT Core para almacenamiento en S3



CREAR UNA POLÍTICA DE PERMISOS

Antes de configurar la política en IAM, se configura los parámetros del bucket, y de los archivos en estos llamados objetos.

Dentro del bucket en el menú modificamos las propiedades para poder alojar páginas web y permisos

En propiedades no desplazamos hasta encontrar la opción para permitir alojar páginas web, se da clic en Editar, luego se marca la casilla para alojamiento de web estáticos de S3

Figura 35

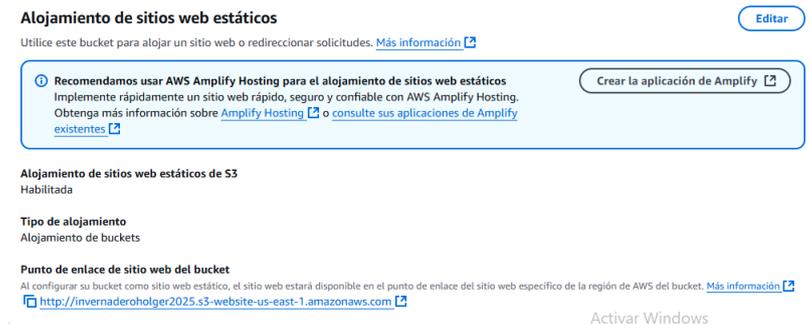
Menú de propiedades del bucket y se activan las opciones necesarias.



En propiedades no desplazamos hasta encontrar la opción para permitir alojar páginas web, se da clic en Editar, luego se marca la casilla para alojamiento de web estáticos de S3

Figura 36

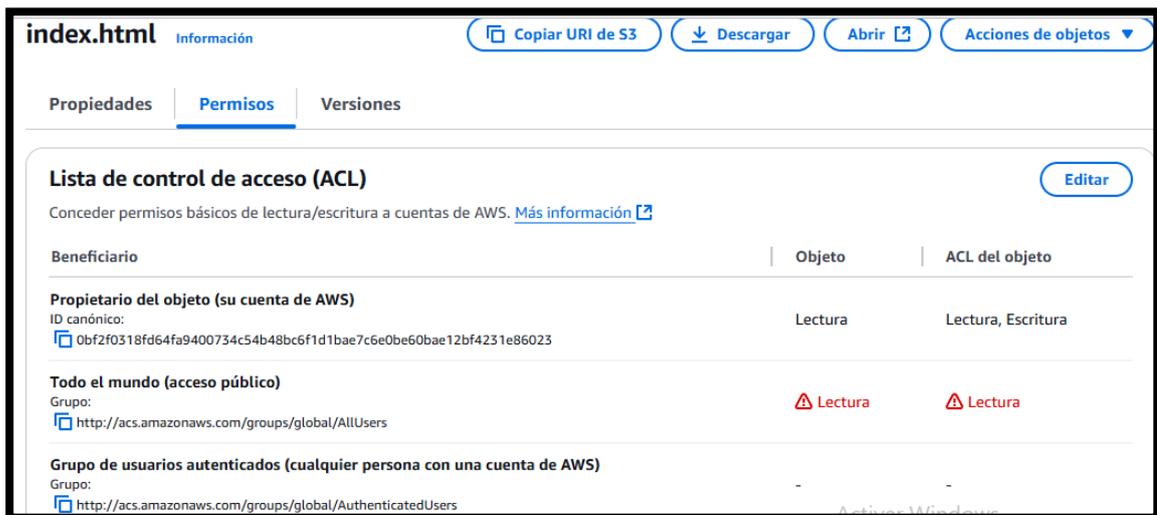
Configuración de permisos en Amazon S3 para habilitar el alojamiento de páginas web



Todos los archivos se da clic y se verifica el ACL, si son vistas por la comunidad, o son de lectura o escritura todo dependerá, al requerimiento en este caso se quiere que el propietario puede leer y escribir y el resto del mundial leer la información.

Figura 37 .

Configuración de ACL en Amazon S3



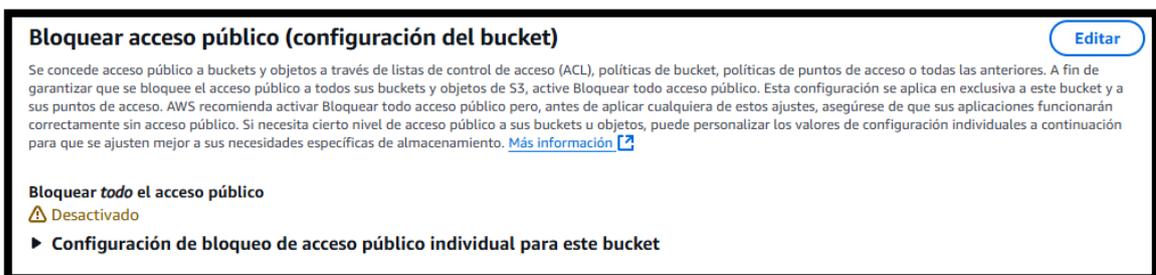
Nota. La imagen muestra la configuración de la Lista de Control de Acceso (ACL) en Amazon S3 para un archivo *index.html*. Se observan los permisos de lectura y escritura otorgados al propietario del objeto y las restricciones de acceso público. *Configuración de ACL en Amazon S3*, por Amazon Web Services, s.f., en AWS Documentation (<https://docs.aws.amazon.com/>). Copyright por Amazon Web Services.

En cuanto al menú que estaba alado de propiedades, llamado permisos se modifica los siguientes elementos:

Bloquear acceso público (configuración del bucket)

Figura 38

Configuración de bloqueo de acceso público en Amazon S3



POLÍTICA PARA LOS BUCKETS

La política de los buckets, son aplicables a los objetos que se encuentran en los buckets, para interactuar con otras cuentas se debe utilizar políticas de IAM.

POLITICA GENERAL PARA EL BUCKET

```
{
  "Version": "2012-10-17",
  "Id": "Policy1738037929560",
  "Statement": [
    {
      "Sid": "Stmt1738037926544",
      "Effect": "Allow",
      "Principal": "*",
```

```

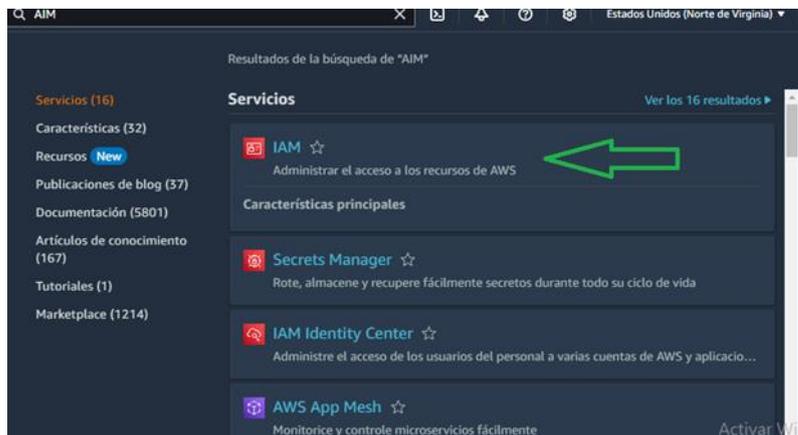
    "Action": "s3:*",
    "Resource": "arn:aws:s3:::invernaderoholger2025"
  }
]
}

```

Luego de crear la política general para los objetos que pertenecen al bucket se procede a aplicar la política AIM, para ello se escribe en la barra de búsqueda AWS IAM

Figura 39

Interfaz de búsqueda de IAM en AWS.

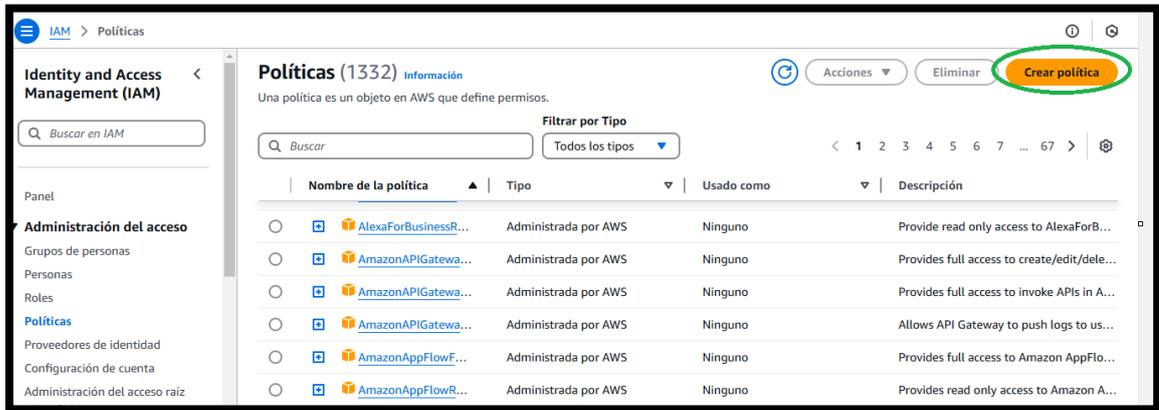


Estando en el servicio AIM, se escoge políticas, y se escoge una política de servicios que le permita a Lambda obtener objetos de un bucket de Amazon S3 y escribir en los Registros de Amazon CloudWatch.

La política siguiente cumple ese objetivo, para ello hay que seleccionar la palabra crear política y luego se escribe este archivo json

Figura 40

Interfaz de administración de políticas en AWS IAM (Identity and Access Management), resaltando la opción para crear una nueva política



Al elegir Crear Política se va al editor de políticas, para mayor facilidad se escribe json y se coloca el siguiente código

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogGroup",
        "logs:CreateLogStream"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ]
    }
  ]
}
```

```

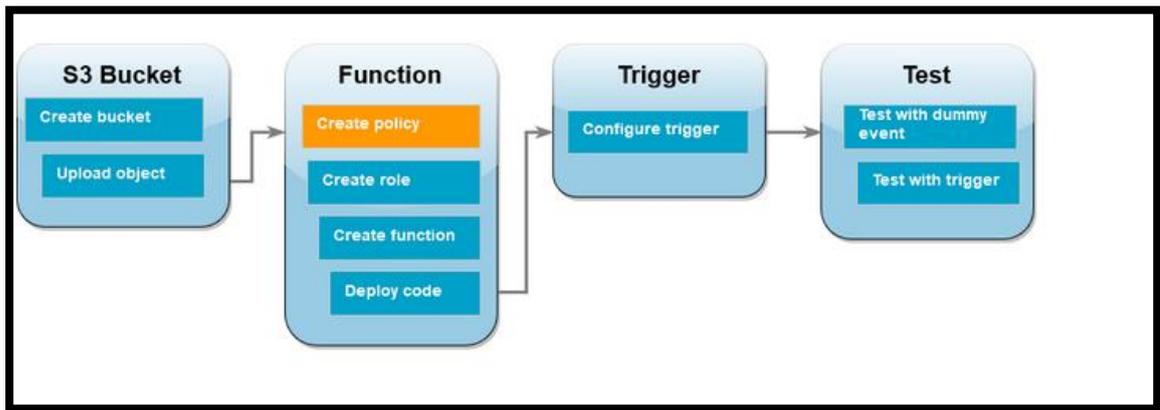
    ],
    "Resource": "arn:aws:s3:::*/*"
  }
]
}

```

Para el nombre de la política, ingrese s3-trigger-tutorial.
Y presionar por último crear política

Figura 41

Integración entre Amazon S3 y AWS Lambda, creación de políticas



CREACIÓN DE UN ROL DE EJECUCIÓN

La creación de roles permite a la función de Lambda (escogida) tener acceso a recursos y servicios de AWS.

Para hacer esto, abre la página de Roles en la consola de IAM y selecciona Crear rol.

En Tipo de entidad de confianza, elige Servicio de AWS. Para el caso de uso, selecciona Lambda y haz clic en Siguiente.

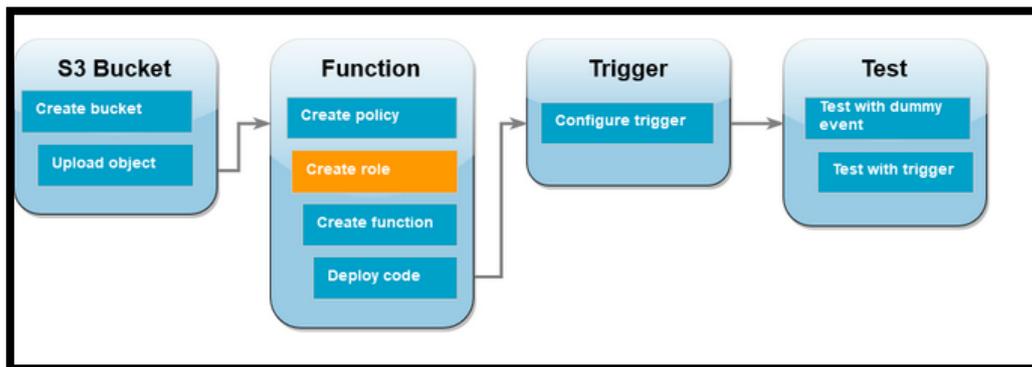
En el cuadro de búsqueda de políticas, escribe s3-trigger-tutorial. Luego, escoge la política que creaste (s3-trigger-tutorial) y haz clic en Siguiente.

En la sección de Detalles del rol, escribe lambda-s3-trigger-role como nombre.

Para terminar, selecciona Crear rol.

Figura 42

Integración entre Amazon S3 y AWS Lambda, incluyendo la creación de rol



CREAR LA FUNCIÓN DE LAMBDA

Para crear la función de Lambda Abra la página de Funciones en la consola de Lambda.

Asegúrese de trabajar en la misma Región de AWS en la que creó el bucket de Amazon S3. Puede cambiar la región usando la lista desplegable en la parte superior de la pantalla.

Seleccione Creación de función.

Elija Crear desde cero.

Bajo Información básica, haga lo siguiente: En Nombre de la función, escriba s3-trigger-tutorial.

En Tiempo de ejecución, seleccione Python 3.12 o la versión más actual.

En Arquitectura, elija x86_64.

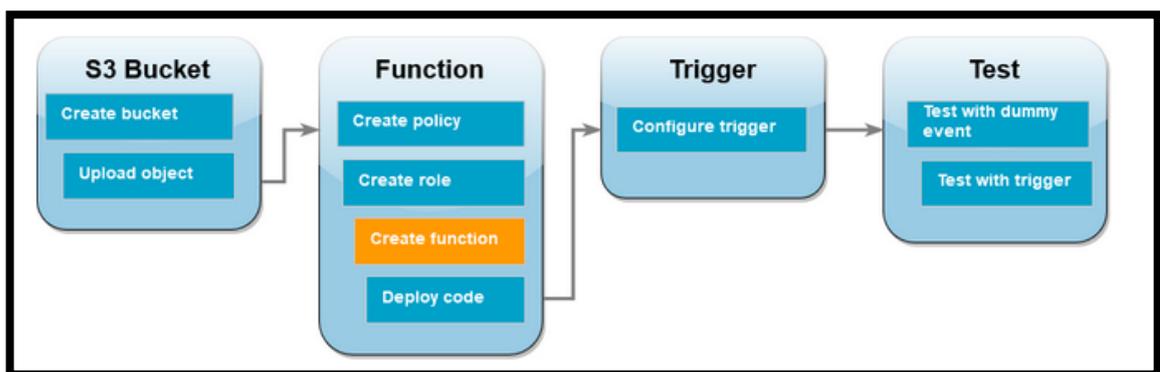
En la pestaña Cambiar rol de ejecución predeterminado, haga lo siguiente: Amplíe la pestaña y luego elija Utilizar un rol existente.

Seleccione el lambda-s3-trigger-role que creó antes.

Seleccione Creación de función.

Figura 43

integración entre Amazon S3 y AWS Lambda, creación de función



IMPLEMENTAR EL CÓDIGO DE LA FUNCIÓN

La función de Lambda obtendrá el nombre de la clave del objeto cargado y el nombre del bucket desde el parámetro event que recibe de Amazon S3.

CODIGO DE LA FUNCION

```
import json
import boto3
from datetime import datetime, timedelta
```

```
s3_client = boto3.client("s3")
```

```
# Ajuste de zona horaria (por ejemplo, UTC-4 para América/Bogotá, Lima, Caracas)
```

```
TIMEZONE_OFFSET = -5 # Ajusta según tu zona horaria (horas de diferencia con
UTC)
```

```
def extract_date_time(timestamp_str):
    """ Extrae el día y la hora de un timestamp en formato UNIX o ISO 8601 y ajusta la
zona horaria. """
    try:
        # Convertir el timestamp a string (por si viene como número)
        timestamp_str = str(timestamp_str)

        # Si el timestamp es numérico (puede ser formato UNIX en milisegundos o
segundos)
        if timestamp_str.isdigit():
            timestamp_int = int(timestamp_str)

            # Si el número es mayor a 10^10, probablemente está en milisegundos,
convertirlo a segundos
            if timestamp_int > 10**10:
                timestamp_int = timestamp_int / 1000 # Convertir de ms a segundos

            # Convertir UNIX timestamp a datetime en UTC
            timestamp_obj = datetime.utcfromtimestamp(timestamp_int)
        else:
            # Normalizar formato de ISO 8601 si tiene 'Z'
            if "Z" in timestamp_str:
                timestamp_str = timestamp_str.replace("Z", "+00:00")

            # Intentar parsear el timestamp en formato ISO 8601
            timestamp_obj = datetime.fromisoformat(timestamp_str)

        # Ajustar la zona horaria sumando o restando horas
        timestamp_obj = timestamp_obj + timedelta(hours=TIMEZONE_OFFSET)
```

```

# Extraer el día y la hora en formato separado
dia = timestamp_obj.strftime("%Y-%m-%d") # Formato: "AAAA-MM-DD"
hora = timestamp_obj.strftime("%H:%M:%S") # Formato: "HH:MM:SS"

return dia, hora

except Exception as e:
    print(f"Error al extraer día y hora: {e}")
    return None, None # En caso de error, retornar valores nulos

def extract_variables_from_s3(bucket, key):
    try:
        # Descargar el archivo desde S3
        response = s3_client.get_object(Bucket=bucket, Key=key)
        content = response["Body"].read().decode("utf-8")
        data = json.loads(content) # Cargar el JSON desde S3

        # Obtener timestamp o generar uno nuevo si no está presente
        timestamp_str = str(data.get("timestamps", int(datetime.now().timestamp() *
1000))) # Guardar en ms

        # Extraer día y hora
        dia, hora = extract_date_time(timestamp_str)

        # Construir la estructura con 2 campos principales
        extracted_data = {
            "timestamps": timestamp_str,
            "dia": dia,
            "hora": hora,
            "datos": {
                "temperatura": data.get("temperatura"),
                "humedad": data.get("humedad"),
                "humedad_suelo": data.get("humedad_suelo"),
            }
        }

```

```

    }

    # Eliminar claves con valores None en "datos"
    extracted_data["datos"] = {k: v for k, v in extracted_data["datos"].items() if v is
not None}

    # Validar que al menos un subcampo en "datos" esté presente
    if not extracted_data["datos"]:
        raise ValueError("Faltan valores válidos para temperatura, humedad o
humedad_suelo")

    return extracted_data
except Exception as e:
    print(f"Error al extraer variables desde S3: {e}")
    raise

def insert_into_dynamodb(data):
    dynamodb = boto3.resource("dynamodb")
    table = dynamodb.Table("TablainvernaderoHolger")

    # Insertar el item en DynamoDB
    response = table.put_item(Item=data)
    return response

def lambda_handler(event, context):
    try:
        for record in event["Records"]:
            bucket = record["s3"]["bucket"]["name"]
            key = record["s3"]["object"]["key"]

            # Obtener datos desde el archivo en S3
            data = extract_variables_from_s3(bucket, key)

```

```

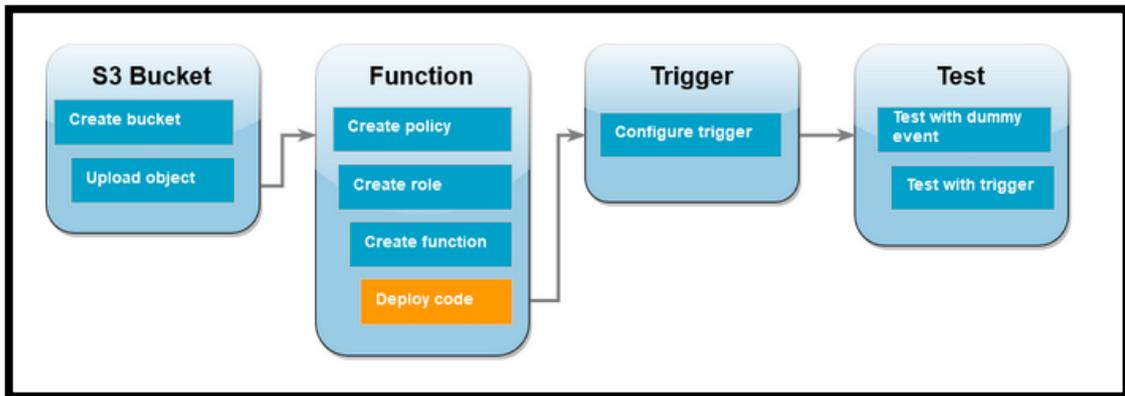
# Insertar los datos en DynamoDB
response = insert_into_dynamodb(data)
print(response)

return {
    "statusCode": 200,
    "body": json.dumps("Datos registrados con éxito en DynamoDB")
}
except Exception as e:
    print(f"Error: {e}")
    return {
        "statusCode": 500,
        "body": json.dumps(f"Error al procesar el evento: {e}")
    }

```

Figura 44

Integración entre Amazon S3 y AWS Lambda, ejecución de código lambda



CREE UN DESENCADENADOR DE AMAZON S3

Para configurar un activador de S3 en AWS Lambda, primero entre a la consola de Lambda y en la parte de información general de la función, elija la opción para agregar un activador. Luego, elija S3 como la fuente del evento y seleccione el bucket que creó

en-el-tutorial.

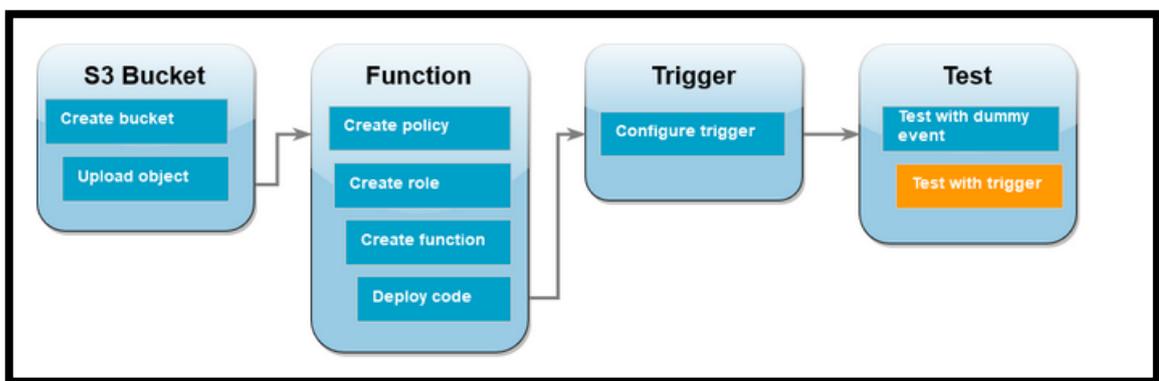
Asegúrese de elegir todos los eventos de creación de objetos en los tipos de eventos. Para evitar invocaciones recursivas, marque la casilla que confirma que no se recomienda usar el mismo bucket de Amazon S3 para la entrada y la salida.

Finalmente, guarde la configuración eligiendo la opción de añadir.

PROBAR LA FUNCIÓN DE LAMBDA CON EL DESENCADENADOR DE AMAZON S3

Figura 45

Integración entre Amazon S3 y AWS Lambda, configuración de disparador



Para podernos comunicar con AWS dynamoDB, se debe modificar la política de role, que está establecida con la función lambda y debe quedar de la siguiente forma

Figura 46

Configuración de una política de role en AWS IAM para permitir que una función Lambda interactúe con Amazon DynamoDB y S3.

```
1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Action": [
7-         "logs:PutLogEvents",
8-         "logs:CreateLogGroup",
9-         "logs:CreateLogStream"
10-      ],
11-      "Resource": "arn:aws:logs:*:*:*"
12-     },
13-     {
14-       "Effect": "Allow",
15-       "Action": [
16-         "s3:GetObject"
17-      ],
18-      "Resource": "arn:aws:s3::*/*"
19-     },
20-     {
21-       "Effect": "Allow",
22-       "Action": "dynamodb:PutItem",
23-       "Resource": "arn:aws:dynamodb:us-east-1:084375545722:table/TablainvernaderoHolger"
24-     }
25-   ]
26- }
```

En el siguiente cuadro de dialogo se puede verificar los servicios nivel de acceso tienen de forma de Resumen, denotando tres servicios CloudWatch Logs para ver el registro de los servicios, DynamoDB con la opción de escribir, y S3 con la opción de Leer, estas operaciones de nivel de acceso las hace por medio de la función lambda

Figura 47

Resumen de permisos en AWS IAM para Lambda

Servicio	Nivel de acceso	Recurso	Condición de solicitud
CloudWatch Logs	Limitado: Escribir	region cadena como [Todos]	None
DynamoDB	Limitado: Escribir	TableName cadena como [TablaInvernaderoHolger, region] cadena como [us-east-1]	None
S3	Limitado: Leer	BucketName cadena como [Todos, ObjectPath] cadena como [Todos]	None

3.3. RESULTADOS DE LOS PARÁMETROS MEDIDOS

Test por medio del MQTT de prueba en IOT Core, ingresando invernadero/# , para todos los topic y invernadero/temperatura en publicador para ver el mensaje de la temperatura.

Figura 48

Condiciones climáticas de un vivero de tomates, temperatura en tiempo real.



PARA ACCEDER A AWS DYNAMODB

- Inicia sesión en AWS Management Console.
- Ve a DynamoDB en la consola de AWS.
- En el panel izquierdo, haz clic en Tablas y luego en Crear tabla.

DEFINIR LOS ATRIBUTOS PRINCIPALES

- Ingresar un nombre relevante para la tabla, por ejemplo, TablainvernaderoHolger.
Define el esquema de clave primaria, que puede ser:

- Clave de partición (Partition Key): timestamp (en este caso se designó como cadena 'S')
- Clave de ordenación (Sort Key, opcional), estos dos primeros son los campos guías.
- Los otros campos que se insertó son Datos que a su vez tiene los subcampos temperatura, humedad y humedad del suelo y también son tipo cadena
- Además de ello se procedió a crear 2 campos más como son Fecha y Hora que sale de la variable timestamps
- Haz clic en Crear tabla
- Agregar Variables y Datos a la Tabla, como se mencionó se incorporan los demás subcampos
- Insertar Datos Manualmente en la Consola, para poder leer la información que se almacena por medio de la operación `aws iot core` `aws lambda` `aws dynamo`

Por último, en la consola de Aws DynamoDB, selecciona la tabla recién creada.

HAZ CLIC EN CREAR ELEMENTO.

Ingresa los siguientes valores:

timestamps: Cadena(Ejemplo: 17383835)

Datos: Cadena ("humedad" o "temperatura")

dia: Convertida de la variable Timestamps

Hora: Convertida de la variable Timestamps

Guarda los cambios haciendo clic en Guardar elemento, debe quedar de la siguiente manera

Para actualizar los valores se dan en la flecha que alada de acciones.

Figura 51

Registro de datos en AWS DynamoDB



The screenshot shows the AWS DynamoDB console interface for a table named 'TablainvernaderoHolger'. At the top, there are buttons for 'Vista previa automática' and 'Ver los detalles de la tabla'. Below that, there is a section 'Escanear o consultar elementos' with a subtext 'Expanda para consultar o examinar elementos.' A green notification bar indicates 'Completado. Unidades de capacidad de lectura consumidas: 2'. The main area shows 'Elementos devueltos (2)' with a 'Crear elemento' button. A table displays the following data:

timestamp...	datos	dia	hora	ttl_timestamp (TTL)
1739320726...	"temperatura": {"S": "25"}, "humedad": {"S": "67"}, "humedad_suelo": {"S": "72" }	2025-02-11	19:38:46	1739320730
1739320615...	"temperatura": {"S": "28"}, "humedad": {"S": "78"}, "humedad_suelo": {"S": "67" }	2025-02-11	19:36:55	1739320621

3.3.1. CONFIGURACIÓN DEL PROTOCOLO MQTT

La conexión del hardware con la nube ocurre después de la creación del dispositivo en AWS IoT Core. Los sensores de humedad del suelo y el controlador de la válvula de riego se conectan a un módulo compatible con AWS, como un microcontrolador que puede soportar MQTT.

Configuración del protocolo MQTT: el protocolo MQTT se utiliza para conectar dispositivos IoT a AWS. A través de este protocolo, los sensores y la válvula de riego publicarán los datos de humedad y recibirán órdenes para abrir o cerrar el flujo de agua.

Los temas se publican y suscriben en MQTT para que los dispositivos se comuniquen.

Durante el periodo de prueba, los sensores de humedad del suelo capturaron información continua, la cual se almacenó en AWS IoT.

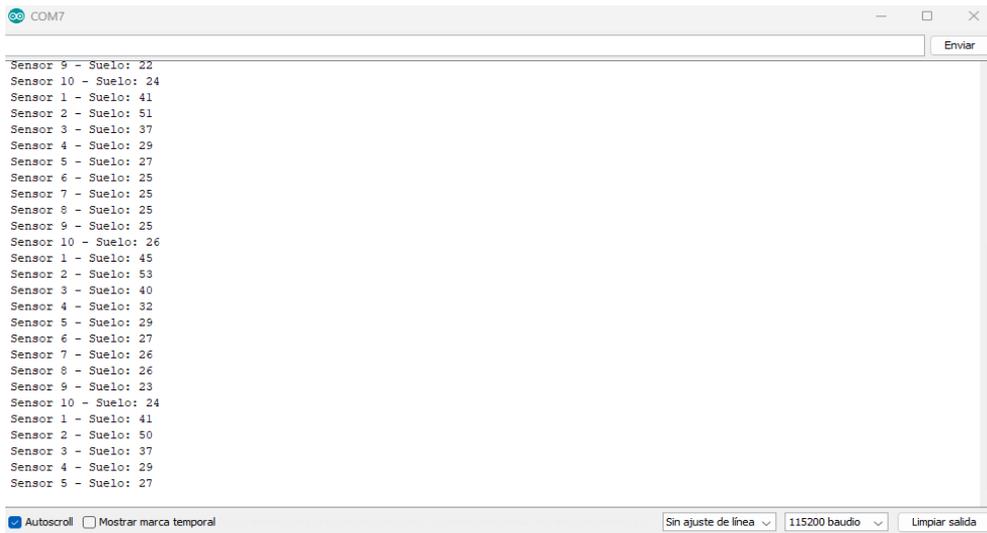
Se comparo los resultados con una Arduino Mega junto con un NODECUM ESP8266, se pudo comprobar que el ajuste necesario hay que realizar debido a la latencia y la

interferencia de ruido. Valores recogidos por una Arduino Mega que son los valores analógicos se tienen que tratar debido a error de formatos.

VALORES MEDIDOS DESDE UNA PLACA ARDUINO MEGA

Figura 52

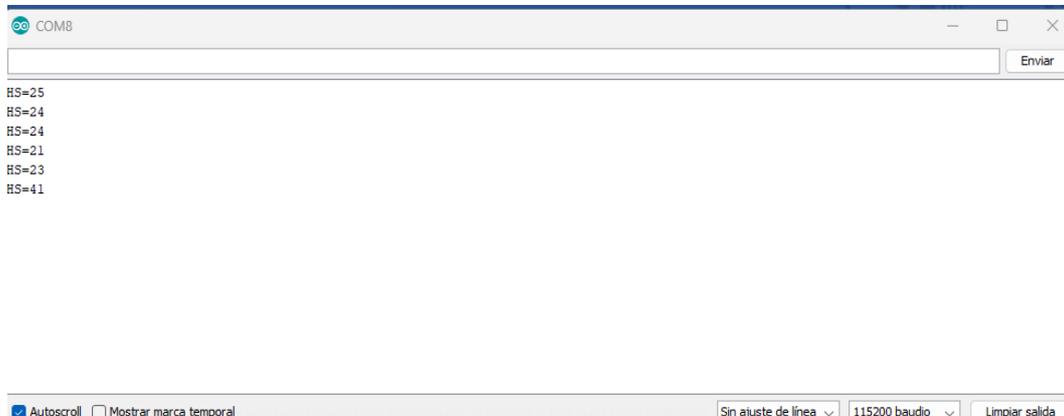
Comprobación en Arduino y Placa Esp8266



Valores Medidos desde la NodeMcu esp8266, se puede observar que la variable HS que corresponde a la Humedad de Suelo son los valores recolectados por Arduino Mega y enviados a la placa esp8266.

Figura 53

Comprobación Placa Esp8266



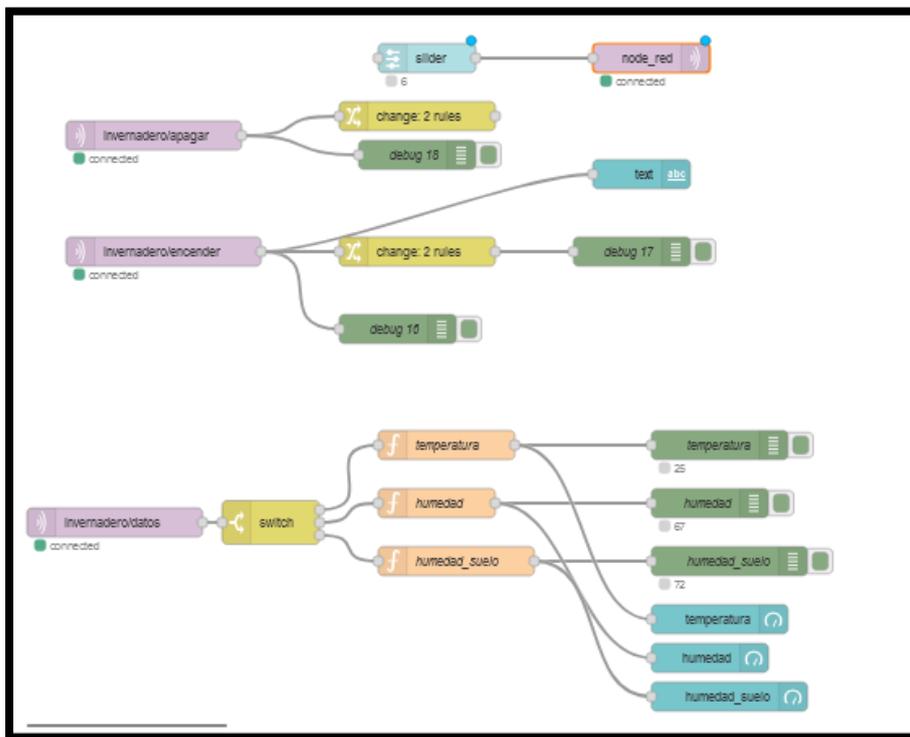
También se hizo las respectivas comprobaciones con la tarjeta esp8266 y conectándola directamente a AWS IOT, como para la nube Tuya Smart y Node Red

NODE RED

Nodo red nos permite interactuar con AWS IOT y tuya Smart, pedir peticiones desde Node red hacia la Nube, y viceversa. Node_red nos permite visualizar el comportamiento de la temperatura ambiente, humedad ambiente y humedad del suelo.

Figura 54

Interfaz Node Red relación



Nota: Describe la interacción de Node Red. Elaboración Propia

Para suscribirse se usa el topic invernadero/#, este hará que cualquier topic extraído del medio sea verificado desde AWS IOT

Se comienza a distribuir las plantaciones de acuerdo al esquema mostrado anterior el terreno está distribuido en un área de 4 m x 3 m y cada plantación está separado .50 cm de cada planta en fila, y 80 cm de ancho de las plantas.

Figura 55

Invernadero a Escala estado Inicial



Se puede ajustar el sistema de acuerdo de riego, en esta topología se ha realizado también un canal intermedio.

Figura 56

Sembrío de Tomate semana 5



El sensor de humedad y temperatura de cada planta es colocado de acuerdo en un lugar no muy alejado de planta y de acuerdo al desarrollo ir colocando su nueva localización

Figura 57

Sensor DHT11 en el cultivo

(AWS)



El monitoreo continuo de la humedad del suelo se lo realiza mediante sensores FC 28 colocados en el suelo de cada planta, como se puede corroborar en la siguiente figura.

Figura 58

Sensor FC 28 de Temperatura y Humedad en el cultivo



3.3.2. PRESUPUESTO

Tabla 10

Presupuesto de materiales

PRESUPUESTO			
ELEMENTO	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
SENSORES FC-28	10	2.13	21.30
DHT11 SENSOR DE TEMPERATURA Y HUMEDAD	10	2.01	20.10
ARDUINO MEGA 2560 R3 PARA ARDUINO	1	21.25	21.25
ARDUINO ESP8266	1	7.14	7.14
CABLE DE PARLANTE 2X18 AWG	30 mtrs	0.4	12
CABLE DUO PARA PARLANTE	30 mtrs	0.45	13.5
CABLE MACHO-MACHO	1 bloque	2.38	2.38
CABLE HEMBRA-MACHO	1 bloque	2.38	2.38
ROUTER TPLINK	1	25	25
CABLE 18 FLEXIBLE	30	0.5	15
CAJETIN DE PLASTICO PLASTIGAMA 20X20	1	10	10
MALLA POLISOMBRA	15 X2	2.40	36
SEMILLA DE TOMATE	SOBRE	1	0.8
CAÑA PARA SOSTENER	3 CAÑAS	15	15
MALLA LATERAL	15 MTRS	0.6	9
TRAVESAÑOS	8	5	40

3.4. DISCUSIÓN

3.4.1 EFICIENCIA DEL SISTEMA

El sistema de riego por goteo automatizado mostró un desempeño sobresaliente en términos de eficiencia y precisión en la entrega de agua a las plantas de tomate. Uno de los aspectos clave fue la notable reducción en el consumo de agua, alcanzada gracias a la automatización y la capacidad de ajustar el riego en función de las condiciones del suelo y las necesidades de la planta. La plataforma permitió una monitorización en tiempo real, brindando la oportunidad de hacer ajustes remotos, lo que aseguró que el riego se activara exclusivamente cuando fuera necesario, optimizando así el uso del recurso hídrico.

Este enfoque no solo redujo el desperdicio de agua, sino que también mejoró las condiciones de crecimiento de las plantas, ya que el riego se adaptaba específicamente a sus requerimientos. Este tipo de optimización es esencial en áreas donde el agua es un recurso escaso, y la capacidad de control remoto también mejora la eficiencia operativa, reduciendo la necesidad de intervención humana constante.

3.4.2 LIMITACIONES Y DESAFÍOS

A pesar del éxito global del sistema, se identificaron ciertos desafíos y limitaciones. Uno de los principales problemas fue la latencia en la transmisión de datos, ya que el tiempo de procesamiento y la dependencia de la nube causaron retrasos leves en la activación del riego. Esto puede resolverse mediante la implementación de soluciones híbridas que combinen procesamiento local edge computing con el almacenamiento y análisis en la nube, reduciendo así los tiempos de respuesta.

Otro reto fue la variabilidad en las lecturas de algunos sensores de humedad del suelo. Durante las pruebas, se observó que ciertas unidades presentaban fluctuaciones en sus

datos, lo que sugiere la necesidad de una calibración más precisa de los sensores para asegurar la consistencia en las lecturas. La calidad de los sensores es crucial para mantener la exactitud del sistema, y su mejora futura puede tener un impacto directo en la eficiencia del riego.

Además, la conectividad fue un desafío relevante. En ciertas ocasiones, la falta de una conexión estable a Internet limitó la funcionalidad del sistema. Esto es particularmente crítico en áreas rurales o de difícil acceso donde las redes de comunicación no son confiables. En estos casos, sería beneficioso contar con sistemas de respaldo que permitan al sistema funcionar de manera autónoma ante fallos de conectividad, garantizando así la continuidad en el riego.

3.4.3 COMPARACIÓN CON MÉTODOS TRADICIONALES

Al comparar el sistema automatizado con los métodos tradicionales de riego manual, los beneficios son evidentes. El riego manual requiere un uso intensivo de mano de obra y suele resultar en una aplicación menos precisa del agua, lo que conlleva un mayor desperdicio del recurso. En contraste, el sistema basado en IoT optimizó el consumo de agua y minimizó la necesidad de intervención humana, permitiendo que cada planta recibiera la cantidad exacta de agua según sus necesidades en tiempo real.

El riego tradicional no tiene la capacidad de ajuste automático que ofrece el sistema basado en sensores, lo que conduce a una distribución desigual del agua. Además, el monitoreo y control remoto permitieron al agricultor gestionar el riego sin estar físicamente presente, liberando tiempo para otras tareas importantes. Esta comparación destaca la efectividad del sistema automatizado no solo en la reducción de costos y tiempo, sino también en la sostenibilidad agrícola.

3.5. CONCLUSIONES

- **Creación del invernadero a escala:**
Se logró construir un invernadero funcional que reproduce condiciones controladas adecuadas para el cultivo de tomate de riñón. Este entorno permitió medir y controlar parámetros ambientales críticos como temperatura, humedad y niveles de nutrientes, garantizando condiciones óptimas para el crecimiento de las plantas.
- **Implementación del sistema de riego por goteo:**
El sistema de riego por goteo automatizado permitió dosificar agua de manera precisa según las necesidades específicas del cultivo. Además, la integración de paneles de visualización facilitó el monitoreo en tiempo real de variables ambientales y del estado del riego, mejorando la toma de decisiones agronómicas.
- **Selección de sustratos óptimos:**
Se seleccionaron sustratos con propiedades físicas y químicas adecuadas que favorecieron el desarrollo saludable de las plántulas de tomate. Esta elección impactó positivamente en la retención de humedad, el pH y la disponibilidad de nutrientes esenciales.
- **Diseño del sistema de monitoreo basado en sensores:**
Se diseñó e implementó un sistema de monitoreo eficiente utilizando sensores DHT11 y FC-28 para registrar datos ambientales críticos. Estos datos fueron transmitidos en tiempo real a la nube mediante AWS IoT, permitiendo un análisis continuo y una supervisión remota del invernadero.
- **Evaluación del sistema IoT de riego por goteo automatizado:**
Las pruebas experimentales demostraron la eficacia del sistema IoT implementado, evidenciando una optimización en el uso de agua. El sistema respondió adecuadamente a las condiciones ambientales detectadas, activando el riego según los parámetros establecidos, lo que mejoró la eficiencia hídrica y la productividad del cultivo.

BIBLIOGRAFIA

- AWS, A. (s.f.). *Uso de un desencadenador de Amazon S3 para invocar una función de Lambda*. AMAZON .
- Microsoft. (s.f.). *Protocolos y tecnologías de IoT*. Mexico: Microsoft.
- Amazon. (2024). AWS IoT Core. Amazon Web Services.
- ÁngelEnríquezEstrella, M. (2021). Evaluación de la producción y el manejo postcosecha del tomate riñón (*Solanum*). Chimborazo: Universidad Estatal Amazónica.
- CHANTAL ALEJANDRA MORALES ROJAS, J. I. (2021). DESARROLLO DE API REST PARA AUTOMATIZAR PROCESO. QUITO: ESCUELA POLITÉCNICA NACIONAL.
- Das1, A. S. (2021). A Comprehensive Review on the Application of Internet. *Wireless Personal Communications*, 31.
- Donaldo Zúniga, R. M. (2021). Gestión y manejo del agua en la agricultura. San José, Costa Rica: Instituto Interamericano de Cooperación para la Agricultura y Red COMAL.
- INTAGRI, E. E. (2019). Diseño Agronómico del sistema de riego por goteo. *intagri*, 5p.
- MINISTERIO DEL AMBIENTE, A. Y.-M. (2021-2026). PLAN NACIONAL DE RIEGO Y DRENAJE 2021 – 2026. Ecuador: Subsecretaría de Agua Potable, Saneamiento, Riego y Drenaje – MAATE.
- MUHAMMAD AYZAZ, M. A.-U.-H. (2019). Internet-of-Things (IoT)-Based Smart Agriculture. *IEEEAccess*, 33.
- Pérez, M. Á. (2021). NODE-RED COMO HERRAMIENTA. JAEN: Escuela Politécnica Superior de Linares.
- SALAZAR, P. F. (2022). ANÁLISIS DE LA SEGURIDAD DEL PROTOCOLO DE TRANSPORTE MQTT EN. POPAYÁN: UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA – UNAD.

Seyar, M. H. (2023). Development of an IoT-Based Precision Irrigation System for. Appl. Sci, 19.tuya. (2024). about. California: Tuya Inc.

Yarecuador. (2024). Gestionar el sabor del tomate. Ecuador: Yara.

Netafim Ecuador. (s.f.). *Riego por goteo*. Recuperado de <https://www.netafim.ec/drip-irrigation/>

<https://jhuate.es/equipamiento/tecnologia-para-invernaderos/control-del-clima-en-invernaderos/sensores-para-invernaderos>

<https://www.scielo.org.mx/scielo.php>

<https://repositorioinstitucionaluacm.mx/jspui/bitstream/123456789/118/3/Carlos%20Alberto%20Serrano%20Montiel.pdf>

Ministerio de Agricultura y Ganadería (MAG). (2021, Junio). *Riego por surco* [Archivo PDF]. <https://www.mag.gob.sv/wp-content/uploads/2021/06/39riego-por-surco.pdf>

Espressif Systems. (s.f.). ESP8266. Retrieved from <https://www.espressif.com/en/products/socs/esp8266>

Espressif Systems. (s.f.). ESP32. Retrieved from <https://www.espressif.com/en/products/socs/esp32>

Raspberry Pi Foundation. (s.f.). Raspberry Pi Foundation. Retrieved from <https://www.raspberrypi.org/>

Regaber. (s.f.). *Riego por goteo subterráneo: Instalación cuidadosa y profesional*. <https://regaber.com/riego-por-goteo-subterraneo/#instalacin-cuidadosa-y-profesional>

Dispositivos Decagon, Inc. (2017). Guía del integrador 5TE . Recuperado de <https://geomor.com.pl/wp-content/uploads/2017/05/5TE-Integrators-Guide.pdf>

Electrónica Caldas. (sf). *YL-69 HL-69 Sensor de humedad del suelo* . Recuperado de <https://www.electronicoscaldas.com/datasheet/YL-69-HL-69.pdf>

Mecatrónica Naylamp. (sf). *Sensor de Humedad de Suelo FC-28* . Recuperado de <https://naylampmechatronics.com/sensores-temperatura-y-humedad/47>

Random Nerd Tutorials. (n.d.). *DHT11 vs. DHT22 vs. LM35 vs. DS18B20 vs. BME280 vs. BMP180*. Recuperado de <https://randomnerdtutorials.com/dht11-vs-dht22-vs-lm35-vs-ds18b20-vs-bme280-vs-bmp180/>

Agropinos. (s.f.). *Sistema de riego por aspersión*. Recuperado de <https://www.agropinos.com/blog/sistema-de-riego-por-aspersion>

Ministerio de Agricultura y Ganadería. (2021). *Riego por surco* [PDF]. Recuperado de <https://www.mag.gob.sv/wp-content/uploads/2021/06/39riego-por-surco.pdf>

ANEXOS

ACCESS TOKEN EN TUYA SMART CLOUD

Luego, usas la segunda programación para actualizar el token antes de que expire el access_token usando el refresh_token.

Función comparativa para llamar Access token:

```
var url = "https://openapi.tuyaus.com/v1.0/token?grant_type=1"; var t = msg.time;
var sign = msg.payload;
var client_id = flow.get("tuya_client_id"); msg.headers = {
  "sign_method": "HMAC-SHA256", "client_id" : client_id, "t": t.toString(),
  "sign": sign.toUpperCase(),
},
msg.payload = ""; msg.url = url; msg.method = "GET"; return msg;
```

Funciones comparativas para llamar a Refresh token:

```
var creds = flow.get("tuya");
var refresh_token = creds.refresh_token; var t = msg.time;

var url = "https://openapi.tuyaus.com/v1.0/token/" + refresh_token; var sign =
msg.payload;
var client_id = flow.get("tuya_client_id"); msg.headers = {
  "sign_method": "HMAC-SHA256", "client_id" : client_id,
  "t": t.toString(),
  "sign": sign.toUpperCase(),
},
msg.payload = ""; msg.url = url; msg.method = "GET"; return msg;
```

Luego se crea una función para activar el controlador de riego y una función para desactivar.

Función para activar el controlador de riego

```
var device_id = flow.get("device_id");
//var url = "https://openapi.tuyaus.com/v1.0/iot-03/devices/"+device_id+"/commands";

var url = "https://openapi.tuyaus.com/v1.0/iot-03/devices/"+device_id+"/commands";

var t = msg.time;
var client_id = flow.get("tuya_client_id"); var creds = flow.get("tuya");
var access_token = creds.access_token; var sign = msg.payload;

msg.headers = {
  "sign_method": "HMAC-SHA256",

  "client_id" : client_id, "t": t.toString(),
  "mode" : "cors",
  "Content-Type": "application/json", "sign": sign.toUpperCase(), "access_token" :
  access_token,
},
msg.payload = '{"commands":[{"code":"switch","value":false}]}'; msg.url = url;
msg.method = "POST";
//msg.payload = ""; return msg;
```

Función para desactivar el controlador de riego

```
var device_id = flow.get("device_id");
var url = "https://openapi.tuyaus.com/v1.0/iot-03/devices/"+device_id+"/commands";
```

```

var t = msg.time;
var client_id = flow.get("tuya_client_id"); var creds = flow.get("tuya");
var access_token = creds.access_token; var sign = msg.payload;

msg.headers = {
  "sign_method": "HMAC-SHA256", "client_id" : client_id,
  "t": t.toString(),
  "mode" : "cors",
  "Content-Type": "application/json", "sign": sign.toUpperCase(), "access_token" :
  access_token,
},
msg.payload = '{"commands":[{"code":"switch","value":true}]}'; msg.url = url;
msg.method = "POST";
//msg.payload = "; return msg;

```

CODIGO ELABORADO EN ARDUINO MEGA

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

```
#define NUM_SENSORS 10
```

```
const uint8_t sensorPins[NUM_SENSORS] = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11}; // Pines
DHT11 const uint8_t sensorPins2[NUM_SENSORS] = {A0, A1, A2, A3, A4, A5, A6,
A7, A8, A9}; // Pines analógicos para humedad del suelo
```

```
DHT dht[NUM_SENSORS] = {DHT(sensorPins[0], DHT11), DHT(sensorPins[1],
DHT11), DHT(sensorPins[2], DHT11), DHT(sensorPins[3], DHT11),
DHT(sensorPins[4], DHT11), DHT(sensorPins[5], DHT11), DHT(sensorPins[6],
DHT11), DHT(sensorPins[7], DHT11), DHT(sensorPins[8], DHT11),
DHT(sensorPins[9], DHT11)};
```

```

// Definir pines para dispositivos de control #define ventiladorPin 12 // Ventilador
#define focoPin 13 // Foco incandescente

const int chunkSize = 64; // Tamaño del bloque para envío en partes void setup() {

Serial.begin(115200); // Comunicación con consola

Serial1.begin(115200); // Comunicación con ESP8266

// Inicializar sensores DHT

for (int i = 0; i < NUM_SENSORS; i++) { dht[i].begin();

}

// Configuración de los pines de salida pinMode(ventiladorPin, OUTPUT);
pinMode(focoPin, OUTPUT);

// Apagar dispositivos al inicio digitalWrite(ventiladorPin, LOW); digitalWrite(focoPin,
LOW);

delay(2000); // Esperar a que los sensores se estabilicen

}

void enviarEnPartes(String data) { int dataLength = data.length();

for (int i = 0; i < dataLength; i += chunkSize) {

int sizeToSend = min(chunkSize, dataLength - i); Serial1.write(data.substring(i, i +
sizeToSend).c_str());

while (!Serial1.available()) { delay(1000); // Esperar confirmación

}

```

```
char ack = Serial1.read();

if (ack != 'A') {
  Serial.println("Error en la recepción");
}
delay(100); // Pausa para evitar saturación
}
}

void controlarAmbiente(float temperatura) { if (temperatura < 15.0) {
  digitalWrite(focoPin, HIGH); // Encender foco digitalWrite(ventiladorPin, LOW);
  Serial.println("Foco encendido para elevar temperatura.");
} else if (temperatura > 27.0) { digitalWrite(focoPin, LOW);
  digitalWrite(ventiladorPin, HIGH); // Encender ventilador Serial.println("Ventilador
  encendido para enfriar.");
} else {
  digitalWrite(focoPin, LOW); digitalWrite(ventiladorPin, LOW); // Apagar ambos
  Serial.println("Temperatura en rango ideal.");
}
}

void loop() {
  float promedioTemp = 0.0, promedioHum = 0.0; int promedioSuelo = 0;
  int c=0; int d=0;

  // Leer datos de los 10 sensores
```

```

for (int i = 0; i < NUM_SENSORS; i++) { float temperatura = dht[i].readTemperature();
float humedad = dht[i].readHumidity();

int valorSuelo = analogRead(sensorPins2[i]);

int sueloPorcentaje = map(valorSuelo, 0, 1023, 0, 100);

// Sumar valores para promediar

if (!isnan(temperatura) && !isnan(humedad)) { c=c+1;

promedioTemp += temperatura; promedioHum += humedad;

}

d=d+1;

promedioSuelo += sueloPorcentaje;

}

// Calcular promedios promedioTemp /= c; promedioHum /= c; promedioSuelo /= d;

// Mostrar promedios Serial.print("Temp Promedio: ");

Serial.print(promedioTemp, 2); // Imprime con 2 decimales Serial.println("°C");

Serial.print("Humedad Promedio: "); Serial.print(promedioHum, 2); // Imprime con 2
decimales Serial.println("%");

Serial.print("Humedad Suelo Promedio: "); Serial.println(promedioSuelo); // Si es un
entero no requiere decimales

// Enviar datos promediados

enviarEnPartes("TE=" + String(promedioTemp, 2));

```

```
enviarEnPartes("HU=" + String(promedioHum, 2)); enviarEnPartes("HS=" +  
String(promedioSuelo));
```

```
// Controlar dispositivos en función de la temperatura  
controlarAmbiente(promedioTemp);  
  
c=0; d=0;
```

```
delay(5000); // Esperar antes de la siguiente iteración  
  
}
```

CODIGO ELABORADO EN LA PLC IOT NODECUM ESP8266

```
#include <SoftwareSerial.h>  
  
// Configurar pines Rx (GPIO14 - D5) y Tx (GPIO12 - D6)  
SoftwareSerial swSer(14, 12);  
  
#include <ESP8266WiFi.h>  
  
#include <PubSubClient.h>  
  
#include <DHT.h> // Librería para el sensor de temperatura  
  
extern "C" {  
  
#include "libb64/cdecode.h"  
  
}
```

```
String inputString = ""; // Variable para almacenar el dato recibido por Serial  
  
boolean stringComplete = false;
```

```
String hsuelo="";

String humedad="";

String temperatura="";

int hs;

float t, h;

// Umbrales de riego

const float TEMP_MIN = 18.0; // Temperatura mínima para reducir riego
const float TEMP_MAX = 25.0; // Temperatura máxima para aumentar riego
const float HUMIDITY_MIN = 65.0; // Humedad relativa mínima
const float HUMIDITY_MAX = 85.0; // Humedad relativa máxima
const int SOIL_MIN = 50; // Umbral mínimo de humedad del suelo
const int SOIL_OPTIMAL = 75; // Humedad del suelo óptima para riego

// Credenciales de WiFi

const char* ssid = "CONSORCIOCHAMBA";

const char* password = "2450120148ch@mb@";

// AWS IoT Endpoint

const char* awsEndpoint = "a3uh6fvotpx035-ats.iot.us-east-1.amazonaws.com";

// Certificados y claves (agregar tus certificados)
```

```
// xxxxxxxxxxx-certificate.pem.crt
const String certificatePemCrt = \
//-----BEGIN CERTIFICATE-----
"MIIDWjCCAkKgAwIBAgIVANGZ4x4b+qHzDkdzI4QiRjebGp3cMA0GCSqGSIb3D
QEB"\
"CwUAME0xSzBJBgNVBAsMQkFtYXpvbiBXZWlGU2VydmljZXMGtZ1BbWF6b24
uY29t"\
"IEluYy4gTD1TZWF0dGxIIFNUPVdhc2hpbmd0b24gQz1VUzAeFw0yNTAyMDUy
MDU3"\
"MjBaFw00OTEyMzEyMzU5NTlaMBA4xHDAaBgNVBAMME0FXUyBJb1QgQ2Vyd
GlmaWNh"\
"dGUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDVMBJsWitXg
qxm41Cy"\
"YaWVDyIcLz2fJJd33PnaGUxSZr3TOeTb8ox/p63pWULqaFm8hPzw0Fv3ryKxEW8
v"\
"sZ3K0LEN0b5ylKiCJnkUDLzmKNIYP3sIGjs+oqxs072c/zKbjHFHbj99qfNmt0P0"\
"XzqYzo6VDLXJXpPqAcABwKjvLa+wa4uCbHXHxQSN8yZEPcV2tYQG/KcRjt3eN
ZkL"\
"a9eoG46aE6RkU4JyijoYl2RZcnS3v7D34Rv0q3NbE7KkAPNQSbe1gb/7p27DYr4W"
\
"tdEcCJhPpaSWJs8Jf/7q+wZ4X32T92nvjPeRSUWQWrj2yGZ1g5ZugAjLQ3cGHcxj"\
"0kQzAgMBAAGjYDBeMB8GA1UdIwQYMBaAFLbmN3fabsBNcSGA/i6bJqIZlCZx
MB0G"\
"A1UdDgQWBBSQLf7d5Lsio6pgVejkKMdrumRg1DAMBgNVHRMBAf8EAjAAM
A4GA1Ud"\
"DwEB/wQEAwIHgDANBgkqhkiG9w0BAQsFAAOCAQEAJCGt5tJrtX+WPHH7OJS
0TqOo"
```

```
"YaTqe2xJOXvCZlytTrFT9w5RQjNUQQGGvw0UE/j3Z4EI/61UOijJ1aDgysZRjRwq"  
\n  
"DdJZ3gX9GZT9YelThEGBKzLPHfb3McTBp8zbRFckkr8izzUO9LlPhyPvmtKGMJr  
6"\n  
"VD4gm82ZmHBtD8RorS6tWcSlByCfv5qK5y6YTU5o0C3hbYKcgl35TQZfxUDqrE  
wF"\n  
"2DedHsk6XnU0AqDpmk3c5XLQPM+67Q3ILFSSbTy/Fp93S1ovAn2ZytQEQ4a+xQ  
mR"\n  
"6BuOzZoZLbzgykS+e2FH36qQnw3nEbvwxigKgVZChwJ33/ZiUVzJkS44H7vbw";  
  
//-----END CERTIFICATE-----
```

```
// xxxxxxxxxxx-private.pem.key
```

```
const String privatePemKey = \
```

```
//-----BEGIN RSA PRIVATE KEY-----
```

```
"MIIEpAIBAAKCAQEA1TASbForV4KsZuNQsmGllQ8iHC89nySXd9z52hlMUma90  
znk" \n  
"2/KMf6et6VIC6mhZvIT88NBb968isRFvL7GdytCxDdG+cpSogiZ5FAy85ijZWD97" \n  
"CBo7PqKsbNO9nP8ym4xxR24/fanzZrdD9F86mM6OIQy1yV6T6gHAACo7y2vsGu  
L" \n  
"gmx1x8UEjfmMrd3FdrWEBvynEY7d3jWZC2vXqBuOmhOkZFOCcoo6GJdkWXJ0  
t7+w" \n  
"9+Eb9KtzWxOypADzUEm3tYG/+6duw2K+FrXRHAiYT6WklibPCX/+6vsGeF99k/d  
p" \n  
"74z3kUIFkFq49shmdYOWboAIy0N3Bh3MY9JEMwIDAQABAoIBACmGFICQS5L  
e+++K" \n
```

"ZSRpipPu/6jXtrvxp6MMIopNISUTp/OleiMxKhaPNXuXDEaWDqx+hrntC4zByxMM
" \

"H0j94osnfOUBLoQo2m2HStwR4xU9baAXtUINwe5gRw9naq5SsqvDfij4m1tBqAwg
" \

"tBa8GbC7wt/bOuk/B7Zzos90IDMJqv+xypRxjcSlftv+t9Rz9GY8st4KJ/XVQOPI" \

"x4Fib0PPok5yzUJARDoSuIFlp3KncXca3HCMybjg4Y7BTCLbviofMATDPY4OMW
V7" \

"1n0J4LqrcReLEDk5CcqeBqdqZ5vaes4ndK+wTZrN9ARicTt6Vs/i6J8mzzKLU0qt" \

"VqhrwdECgYEA7yteF5Af5apOfEsQU1k2L2LmOFbzHYyJUvi/Mtdm0etDa8PuDLx
A" \

"OeaLzfBsOBxz7E4G5+ob1k/Ma08dWivrNfbiamMmzw1VmO6nmDpuMDAM9CB
W4Csd" \

"HTjgpklbY7y7rW9wS7y3/b+xEsYYIrTH4n80Qh1v/SBFkFi7d4SR9qkCgYEA5DCI"
 \

"XODy7wK0XwnrMLYMkBO/Y9vljw4SdqNO5zZyjIV0cO2rQ4aDVxmP3l3pk5g+93
c2" \

"UVaVy/pfeVLhDK3BDDAA1sxxowgyCg9tRohDQuTMxykl0ejGxXSdULNpOfYZF
J4D" \

"yDNVYY9sJDk23M/1cHwAQ1Q5ArdlL9E1n2FwWXsCgYB71Kxm7sxVERSJ7iOJx
Ar2" \

"BfLFAN/nULhZKUI1fWITDshdp5g9J8rpL6D7A5xEgRbUJqIUuXWoRLDGbrP31/1
0" \

"d1tLUgLp+TF0eTx06OvcL0TFsWjJsAjm9Gf8MMcMoM9bQPOXgyKQUwvPeJlOfu
Nv" \

"NgVMpgkxud/KmOaXPYCD4QKBgQCzBdc2PuVYzm5kAYwa8zh38IY++c3npcqm
eQj2" \

"LKIBxZHM5yluGKICO3O7qUIPIWqFC2n59kwfc5ZjwLn/EHJb9W0zF8IvfcaETgY
V" \

```
"R+I9IPPnbEjXniAT2aFyJs9vQbv7iNy+vEdH+xWEdb6tDhoo3Z/3MqRnlSo1kZTY" \  
"LcHO8QKBgQDNEr8dPHIQMZbHYazaGH+C/+m6cntII9bk2j7P2YZekZ/o1A1mCp  
+y" \  
"2biXyNKM+mc3jd3UXjSHvV0UrgUFu1NkbG+Tnw87H8O9JRzQIzyJSBC4iphZ4kv  
b" \  
"kvCY4TBOdHD4Gcx0tHbYt3nelA8cpsnWoa+zFrFUa7H8DeIM72v6TA";  
//-----END RSA PRIVATE KEY-----
```

```
//-----END RSA PRIVATE KEY-----
```

```
// This is the AWS IoT CA Certificate from:
```

```
// https://docs.aws.amazon.com/iot/latest/developerguide/managing-device-certs.html#server-authentication
```

```
// This one in here is the 'RSA 2048 bit key: Amazon Root CA 1' which is valid
```

```
// until January 16, 2038 so unless it gets revoked you can leave this as is:
```

```
const String caPemCrt = \  

```

```
//-----BEGIN CERTIFICATE-----
```

```
"MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQs  
F" \  

```

```
"ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYD  
VQQDExBBbWF6" \  

```

```
"b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMDEwOTEL" \  

```

```
"MAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvcjEzMBcGA1UEAxMQW1hem9uIFJv" \  

```

"b3QgQ0EgMTCCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4g
HHKeNXj" \

"ca9HgFB0fW7Y14h29Jlo91ghYPI0hAEvrAlthtOgQ3pOsqTQNroBvo3bSMgHFzZM"
\

"9O6II8c+6zf1tRn4SWiw3te5djgdYZ6k/oI2peVKVuRF4fn9tBb6dNqcmzU5L/qw" \

"IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMJbulHheb4mjUcAwhmah
RWa6" \

"VOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kDi1L
" \

"93FcXmn/6pUCyziKrlA4b9v7LWlBxcceVOF34GfID5yHI9Y/QCB/IIDegEw+OyQm
" \

"jgSubJrIqg0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAObgNVHQ8B
Af8EBAMC" \

"AYYwHQYDVR0OBBYEFIQYzIU07LwMIJQuCFmcx7IQTgoIMA0GCSqGSib3D
QEBCwUA" \

"A4IBAQCY8jdaQZChGsV2USggNiMOruYou6r4IK5IpDB/G/wkjUu0yKGX9rbxenD
I" \

"U5PMCCjImCXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhIbI1Bjtt/msv0tadQ1wUs" \

"N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKWIJbYK8
U90vv" \

"o/ufQJVtMVT8QtPHRh8jrdkPSHCa2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6X
U" \

"5MsI+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADb
kpy" \

"rqXRfboQnoZsG4q5WTP468SQvvG5";

//-----END CERTIFICATE-----

```

WiFiClientSecure wiFiClient;

void msgReceived(char* topic, byte* payload, unsigned int len);

PubSubClient pubSubClient(awsEndpoint, 8883, msgReceived, wiFiClient);

// Pines para controlar la bomba

#define bombaPin 5 // GPIO para controlar la bomba

bool bombaEncendida = false; // Estado de la bomba

bool estadoPublicado = false; // Nueva variable para controlar la publicación del estado

unsigned long lastPublish = 0;

int msgCount = 0;

// Variables para almacenar datos recibidos por Serial

char receivedData[65];

int i = 0;

// No se necesita la función msgReceived ya que no controlaremos manualmente

void msgReceived(char* topic, byte* payload, unsigned int length) {

    // Esta función queda vacía ya que no recibiremos comandos manuales

}

void setup() {

```

```
Serial.begin(115200);

swSer.begin(115200); // Inicializar el puerto serie software

pinMode(bombaPin, OUTPUT); // Configurar el pin de la bomba

digitalWrite(bombaPin, LOW); // Asegurarse de que la bomba esté apagada al inicio

Serial.print("Conectando a WiFi: ");

Serial.println(ssid);

WiFi.begin(ssid, password);

WiFi.waitForConnectResult();

Serial.print("Conectado, IP: ");

Serial.println(WiFi.localIP());

// Configurar los certificados y conectar a AWS IoT

setCurrentTime();

configAWS();

lastPublish = millis();

}

void loop() {

char payload[500];

pubSubCheckConnect();
```

```

if (swSer.available()) {
    i = 0;

    // Leer los datos recibidos del puerto serial en fragmentos
    while (swSer.available() && i < 64) {
        receivedData[i++] = swSer.read();
    }

    receivedData[i] = '\0'; // Termina la cadena
    stringComplete = true; // Marca que los datos están completos
}

// Procesar los datos solo si stringComplete es true
if (stringComplete) {
    // Verificar y procesar las diferentes entradas

    if (strncmp(receivedData, "HS=", 3) == 0) { // Compara los primeros 3 caracteres
        hs = atoi(receivedData + 3); // Convertir los caracteres después de "HS=" a entero
        Serial.print("Humedad_Suelo: ");
        Serial.println(hs);
    }

    else if (strncmp(receivedData, "HU=", 3) == 0) { // Comparar si empieza con "HU="
        h = atof(receivedData + 3); // Convertir a número lo que sigue de "HU="
        Serial.print("Humedad: ");
        Serial.println(h);
    }

    else if (strncmp(receivedData, "TE=", 3) == 0) { // Comparar si empieza con "TE="

```

```

t = atof(receivedData + 3); // Convertir a número lo que sigue de "TE="

Serial.print("Temperatura: ");

Serial.println(t);

}

// Limpiar el buffer y resetear la bandera después de procesar
memset(receivedData, 0, sizeof(receivedData)); // Vacía el contenido de
receivedData

stringComplete = false; // Resetea la bandera para esperar los próximos datos
}

sprintf(payload, "{\"temperatura\": \"%f\", \"humedad\": \"%f\", \"humedad_suelo\":
\"%d\"}", t, h, hs);

pubSubClient.publish("invernadero/datos", payload);

delay(6000);

if (hs < SOIL_MIN) {

    if (t > TEMP_MIN && t < TEMP_MAX && h >= HUMIDITY_MIN && h <=
HUMIDITY_MAX) {

        activarRiego(); // Se activa si el suelo está seco y las condiciones son normales

    } else {

        desactivarRiego(); // No se riega si la humedad y temperatura están fuera del
rango

    }

} else if (hs >= SOIL_OPTIMAL) {

    desactivarRiego(); // Se apaga el riego si la humedad del suelo es suficiente

```

```

    }

// Enviar confirmación al Arduino Mega

    delay(1000);

    swSer.write('A'); // 'A' indica que está listo para más datos

    pubSubClient.loop(); // Mantener la conexión MQTT
}

// Funciones auxiliares para la conexión a AWS IoT y el manejo de certificados

void configAWS() {

    uint8_t binaryCert[certificatePemCert.length() * 3 / 4];

    int len = b64decode(certificatePemCert, binaryCert);

    WiFiClient.setCertificate(binaryCert, len);

    uint8_t binaryPrivate[privatePemKey.length() * 3 / 4];

    len = b64decode(privatePemKey, binaryPrivate);

    WiFiClient.setPrivateKey(binaryPrivate, len);

    uint8_t binaryCA[caPemCert.length() * 3 / 4];

    len = b64decode(caPemCert, binaryCA);

    WiFiClient.setCACert(binaryCA, len);

}

void pubSubCheckConnect() {

```

```
if (!pubSubClient.connected()) {  
    Serial.print("Conectando a AWS IoT...");  
    while (!pubSubClient.connected()) {  
        pubSubClient.connect("ESP8266");  
    }  
    Serial.println(" Conectado");  
  
    // Suscripciones eliminadas ya que no necesitamos comandos manuales  
}  
  
}  
  
void activarRiego() {  
    // Publicar datos en AWS IoT  
    char payload[128];  
    digitalWrite(bombaPin, HIGH); // Encender bomba automáticamente  
    Serial.println("Riego activado.");  
    sprintf(payload, "{\"estado\": \"encender\"}");  
    pubSubClient.publish("invernadero/encender", payload);  
    delay(5000);  
}  
  
void desactivarRiego() {  
    //Publicar datos en AWS IoT  
    char payload[128];
```

```
digitalWrite(bombaPin, LOW); // Encender bomba automáticamente

Serial.println("Riego desactivado.");

sprintf(payload, "{\"estado\": \"apagar\"}");

pubSubClient.publish("invernadero/apagar", payload);

delay(5000);

}

//AQUI TERMINA ESTE CODIGO

void setCurrentTime() {

    configTime(3 * 3600, 0, "pool.ntp.org", "time.nist.gov");

    time_t now = time(nullptr);

    while (now < 8 * 3600 * 2) {

        delay(500);

        now = time(nullptr);

    }

}

int b64decode(String b64Text, uint8_t* output) {

    base64_decodestate s;

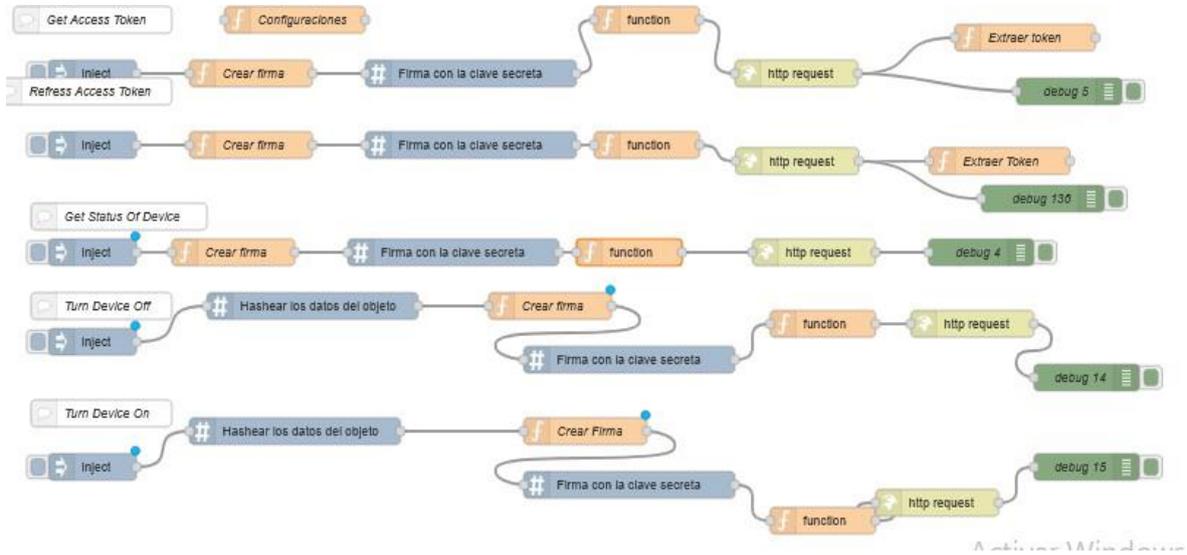
    base64_init_decodestate(&s);

    int cnt = base64_decode_block(b64Text.c_str(), b64Text.length(), (char*)output, &s);

    return cnt;

}
```

BLOQUES EN NODE RED



FUNCIONES QUE ENVIA COMANDOS A LA APPI DE TUYA SMART

