



UNIVERSIDAD TECNOLÓGICA ECOTEC

FACULTAD DE INGENIERÍA

TÍTULO DEL TRABAJO:

SISTEMA DE MONITOREO Y CONTROL EN TIEMPO REAL BASADO EN LOS
SERVICIOS EN LA NUBE PARA LA GESTIÓN DE DATOS DE VARIABLES AGRO-
CLIMATOLÓGICAS

LÍNEA DE INVESTIGACIÓN:

TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

MODALIDAD DE TITULACIÓN:

PROPUESTA TECNOLÓGICA

CARRERA:

INGENIERÍA EN SISTEMAS ÉNFASIS EN REDES

TÍTULO A OBTENER:

INGENIERO EN SISTEMAS CON ÉNFASIS EN ADMINISTRACIÓN DE REDES

AUTOR(A):

FERNANDO DAVID VIERA ESPINALES

TUTOR(A):

MGTR. WILSON RAMIRO POLO GONZÁLEZ

GUAYAQUIL – ECUADOR

2022

1. DEDICATORIA

Esta tesis está dedicada a:

A mis abuelos Josefina y David quienes con gran paciencia y esfuerzo me han permitido cumplir una de mis metas, gracias persistir con sus palabras brindando buenos ejemplos.

A mi madre y tías Edith, Patricia y Cecilia por su apoyo incondicional y cariño además de estar conmigo en todo momento gracias. A mi hermana Beatriz y sobrinos por sus ánimos y buenos deseos.

Finalmente quiero dedicar esta tesis a todos mis amigos por apoyarme y por extender su mano en momentos duros y por la comprensión brindada cada día, a todos siempre los llevo en mis pensamientos.

2. AGRADECIMIENTO

Finalizando este trabajo quiero agradecer a Dios por todas sus bendiciones, a mi familia que ha sabido apoyarme y entenderme en todo este proceso compañeros de trabajo por su comprensión.

Además, quiero agradecer a la Universidad Tecnológica Ecotec, profesores por sus oportunidades y a mi tutor por sus consejos y correcciones.

3. RESUMEN

El presente trabajo de titulación se basó en el uso de los servicios en la nube para el control y monitoreo de variables de tipo agroclimáticas, para ello se formuló como objetivo general: “Elaborar una arquitectura basada en los servicios de la nube para la gestión eficiente de variables en un proceso agrícola”, en base a ello se realizó una revisión bibliográfica en fuentes primarias y secundarias con la finalidad de determinar los aspectos teóricos relacionados con el objeto de estudio, se determinó, mediante una comparativa de servicios en la nube disponibles, la mejor alternativa para el desarrollo y además, mediante una búsqueda de equipos disponibles en el mercado Ecuatoriano, se definió los equipos y sensores mediante los cuales se realizaran las lecturas de variables. Como parte de la propuesta de desarrollo, se realizó una simulación en un invernadero en el cual se instalaron los dispositivos de lectura de variables, y se realizó el desarrollo de un sistema de control y monitoreo mediante código de programación, el cual se administra mediante una interfaz web, a partir de ello se pudo obtener en tiempo real la lectura de las variables de temperatura y humedad, mismas que se visualizaron en la web. La lectura de estas variables facilitó el control y mejoramiento del cultivo del invernadero

Palabras claves: Arduino, AWS, Placa ESP32, Variables agroclimáticas

4. ABSTRACT

The present titling work was based on the use of cloud services for the control and monitoring of agroclimatic variables, for which the general objective was formulated: "Develop an architecture based on cloud services for efficient management of variables in an agricultural process", based on this, a bibliographic review was carried out in primary and secondary sources in order to determine the theoretical aspects related to the object of study, it was determined, through a comparison of available cloud services, the best alternative for development and also, through a search for equipment available in the Ecuadorian market, the equipment and sensors through which the readings of variables were made were defined. As part of the development proposal, a simulation was carried out in a greenhouse in which the variable reading devices were installed, and the development of a control and monitoring system was carried out through programming code, which is managed through a web interface, from this it was possible to obtain in real time the reading of the temperature and humidity variables, which were displayed on the web. The reading of these variables facilitated the control and improvement of the greenhouse crop.

Keywords: Arduino, AWS, ESP32 Board, Agroclimatic variables

ÍNDICE

1. DEDICATORIA	2
2. AGRADECIMIENTO	3
3. RESUMEN	4
4. ABSTRACT	5
1 INTRODUCCIÓN	10
1.1 Antecedentes	10
1.2 Planteamiento del problema	11
1.3 Pregunta problémica	12
1.4 Objetivos	12
1.4.1 Objetivo general	12
1.4.2 Objetivos específicos	12
1.5 Justificación	13
1.6 Alcance de la investigación	13
2 CAPÍTULO I: MARCO TEÓRICO	16
2.1 Agricultura 4.0	16
2.1.1 Las tecnologías de la agricultura 4.0 y la gestión de los datos	17
2.2 Cloud Computing	18
2.2.1 Arquitectura de cloud computing	18
2.2.2 Modelos de servicios de cloud Computing	20
2.2.3 Modelos de implementación de Cloud Computing	21
2.2.4 Plataformas de Cloud computing Open source	21
2.3 Plataformas de cloud computing No open Source	24
2.4 Amazon Web Service (AWS)	25

2.4.1	Servicios de AWS	26
2.5	Gestión de datos en la nube	27
2.6	Variables Agroclimáticas	28
2.6.1	Variables con señales digitales y analógicas	29
2.7	Dispositivos IoT para agricultura	29
2.8	Sensor DHT11	30
2.9	PLACA ESP32	30
3	CAPÍTULO II: METODOLOGÍA DEL PROCESO DE DESARROLLO DE LA PROPUESTA TECNOLÓGICA	33
3.1	Tipo de investigación	33
3.2	Enfoque de la investigación	33
3.3	Periodo y lugar de desarrollo de la propuesta Tecnológica	33
3.4	Definición y comportamiento de variables	33
3.4.1	Conceptualización de variables	33
3.4.2	Operacionalización de variables	34
3.5	Métodos de recopilación de información e instrumentos a utilizar	35
3.5.1	Componentes de hardware	35
3.5.2	Componentes de Software	35
3.5.3	Para levantar los servicios en la nube de AWS se siguen los siguientes pasos:	38
3.5.4	Configuración e integración de AWS con el IDE Arduino y la placa ESP32:	49
4	CAPÍTULO III: SOLUCIÓN DE PROPUESTA TECNOLÓGICA	63
4.1	Implementación de un servidor AWS para el monitoreo y control de la temperatura y humedad.	63
4.2	Aplicación simulada en prototipo de invernadero	64

4.2.1	Funcionamiento de los módulos	65
4.3	Descripción de principales módulos de programación	66
4.4	Pruebas de monitoreo y control	68
4.4.1	Dashboard de Monitoreo y control	68
9.	Conclusiones y recomendaciones	71
4.5	Conclusiones	71
4.6	Recomendaciones	72
10.	BIBLIOGRAFÍA	74
5	ANEXOS	76
5.1	Anexo # 1: Código fuente index.js	76
		79

Índice de tablas

Tabla 1	Comparativa servicios en la nube no open source	25
Tabla 2	Tabla de operacionalización de variables	34

Índice de figuras

Figura 1	Arquitectura del cloud computing	19
Figura 2	Modelos de servicios del cloud computing	20
Figura 3	Esquema metodológico de la propuesta	37
Figura 4	Sitio Web AWS.....	38
Figura 5	Creación de cuenta AWS	39
Figura 6	Creación IoT.....	39
Figura 7	Creación del objeto	40
Figura 8	Nombrar objetos.....	40
Figura 9	Nombrar objetos.....	41
Figura 10	Crear certificados	41

Figura 11 Asignación de políticas y permisos	42
Figura 12 Crear certificados y claves	43
Figura 13 Descargar Arduino IDE.....	44
Figura 14 Crear proyecto Arduino IDE	45
Figura 15 Se procede a descargar la librería WIFI, WIFI LINK	45
Figura 16 Instalación de librerías WIFI, WIFI LINK.....	46
Figura 17 Instalación de certificados AWS	46
Figura 18 Código para lectura de temperatura y humedad	47
Figura 19 Compilación de código	48
Figura 20 Configuración de lectura de variables en AWS	48
Figura 21 Configuración de base de datos.....	49
Figura 22 Configuración de lambda.....	50
Figura 23 Función espPut	50
Figura 24 Función espPut	51
Figura 25 Función espPut	51
Figura 26 Configuración	52
Figura 27 API Gateway	53
Figura 28 Configuración de métodos.....	54
Figura 29 Publicación del servicio API REST.....	56
Figura 30 Crear bucket.....	58
Figura 31 Crear bucket.....	59
Figura 32 Generar sitio web	60
Figura 33 Diseño y arquitectura del monitoreo y control de temperatura y humedad. .	63
Figura 34 Almacenamiento de datos en un servidor a través de internet.....	64
Figura 35 Recepción de datos a través de la aplicación.	65
Figura 36 Arquitectura de solución.....	67
Figura 37 Base de Datos Dynamo.....	67
Figura 38 Página de Inicio.....	68
Figura 39 Capturas de variables.....	69

1 INTRODUCCIÓN

1.1 Antecedentes

Los avances tecnológicos se encuentran a la orden del día, cada vez existen nuevos métodos para el manejo de datos de manera efectiva. Las diferentes industrias, se han visto en la obligación de incluir nuevas tecnologías para tener disponibilidad de datos 24/7, más aún a raíz de la pandemia de covid 19. Estas nuevas tecnologías como el almacenamiento en la nube y el Internet de las cosas (IoT), hacen que las industrias obtengan un alto nivel de optimización de costos, tiempo y recursos. La industria Agrícola no es la excepción, y se ha logrado con la inclusión de dispositivos IoT para el monitoreo de cultivos, riegos, humedad de suelo, entre otros (Chanchí-Golondrino et al., 2022).

La seguridad de la información es una necesidad esencial para las empresas e industrias, es por esto por lo que se ha optado por adquirir servicios almacenados en la nube (“Cloud computing & Amazon AWS,” 2019). Esta tecnología permite acceder remotamente a software, almacenamiento de archivos y procesamiento de datos a través de internet, sin la necesidad de aplicaciones locales en la computadora (¿Qué es Cloud Computing?, 2022). Existe una relación importante entre los dispositivos IoT y los servicios en nube. El almacenamiento de datos en la nube se realiza desde dispositivos IoT y estos están disponibles en cualquier lugar y a cualquier momento. Esto resulta ideal para empresas o industrias con múltiples ubicaciones y operaciones que requieren de un constante monitoreo. Ambas soluciones combinadas apoyan a la captura, almacenamiento, gestión, procesamiento y seguridad de datos.

En el Ecuador, se presentan varios retos y obstáculos al momento de impulsar soluciones para lograr el desarrollo sostenible, ya que los costos que estos representan son altos, es aquí donde la tecnología toma un papel fundamental, demostrando que es posible buscar alternativas tecnológicas viables para mejorar la gestión de la industria agrícola. El sector agrícola en el Ecuador representa aproximadamente el 8% del PIB y el 26.8% de fuentes de empleo (Banco Central del Ecuador, 2022), por lo que es representativo e importante considerar la optimización, control y monitoreo en sus procesos, por tales motivos, se considera de gran importancia proponer una solución tecnológica viable para

mejorar el control y monitoreo de las variables agroclimáticas mediante la gestión de datos a través de la nube y el uso de dispositivos IoT.

1.2 Planteamiento del problema

Debido a los frecuentes avances tecnológicos y con la llegada de la industria 4.0, ha surgido el término de agricultura 4.0, que se refiere a las intervenciones activadas gracias al análisis preciso de datos recopilados y transmitidos mediante herramientas y tecnología avanzada (McCormick, 2021).

Actualmente en el Ecuador, la tecnología cada vez va tomando más fuerza en los sectores productivos, y debido a la llegada del Covid-19, el mundo entero se vio en la obligación de dar un paso adelante, incluyendo diferentes tecnologías, que permitan el teletrabajo y también el monitoreo y control de diferentes procesos, así nació el auge del uso de servicios en la nube como una alternativa para la disponibilidad de datos en tiempo real.

La optimización de procesos y la disponibilidad de datos se ha vuelto una necesidad en la industria, para lo cual, el uso de dispositivos IoT y los servicios en la nube son grandes aliados. La gestión de datos del sector agrícola del país suele realizarse manualmente, es proceso en el que intervienen muchos recursos, por lo que, en un panorama ideal, el uso de dispositivos IoT que almacenan información en la nube, sería un gran ahorro para controlar y monitorear información de suelo, humedad, luz ambiental, entre otras variables que afectan a la producción, y así obtener resultados óptimos y de calidad en el producto final (Saibene, 2020).

En el sector agrícola, al ser uno de los más importantes en la economía del Ecuador, se han visto en la necesidad de incluir tecnología para mejorar la calidad de sus productos, como semillas certificadas, invernaderos inteligentes, mediciones mediante drones, entre otros (Romero, 2021). Dispositivos como helicópteros para fumigación o drones para lectura de datos, ya son utilizados en algunos cultivos, sin embargo, existen factores que pueden afectar el uso de estos como las condiciones climáticas y geografía de la zona (El Productor, 2017).

Así mismo, el control de riego en este sector es fundamental para las plantaciones, ya que el exceso o falta de éste puede perjudicar la producción. También, la planificación de sembríos es un proceso que se realiza acorde a la estación climática, temperatura, humedad, terreno, debido a que existen semillas que crecen en épocas específicas y requieren un cuidado particular. Las variables más importantes y que requieren mayor atención, son las de temperatura y humedad para obtener una óptima producción.

Como solución a la problemática presentada, se realizará un sistema para la gestión de datos provenientes de sensores que transmiten variables de tipo agroclimáticas, y se almacenarán mediante la nube para su posterior control y monitoreo en tiempo real. Los dispositivos a utilizar se seleccionarán basándose en las existencias del mercado ecuatoriano, en cuanto a tecnología para el sector agrícola.

Para esto, se realizará un prototipo de invernadero para la medición de las variables agroclimáticas de temperatura y humedad mediante el sensor DHT11, el cual se conectará mediante una placa ESP32, que será configurada al sistema IDE de Arduino, y a su vez, mediante el cual se enviarán los datos a la nube, además, a través un tablero de control, se podrá realizar la lectura, análisis, control y monitoreo de los valores arrojados por los sensores, con lo cual se pretende visualizar los beneficios del uso de dispositivos tecnológicos en la agricultura.

1.3 Pregunta problémica

Basado en las situaciones mencionadas con anterioridad surge la siguiente pregunta problémica: ¿Cómo monitorear de manera eficiente las variables agroclimáticas a través del uso de dispositivos IoT y servicios en la nube?

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar un sistema de monitoreo basada en los servicios de la nube para la gestión eficiente de variables en un proceso agrícola.

1.4.2 Objetivos específicos

- 1) Determinar las bases teóricas que sustentan el uso de los servicios en la nube para la gestión de variables agroclimáticas.

- 2) Analizar y comparar las alternativas de hardware y software disponibles en el mercado y determinar la alternativa adecuada mediante un análisis de criterios de elección.
- 3) Diseñar una arquitectura de servicios en la nube para la gestión de variables agroclimáticas.
- 4) Desarrollar un sistema de gestión de variables agroclimáticas de temperatura y humedad mediante el sensor DHT11 y la placa ESP32, mediante el IDE de Arduino y AWS.

1.5 Justificación

Esta propuesta tecnológica, tiene como objetivo desarrollar un prototipo de sistema para la gestión de datos proveniente de variables agroclimáticas en AWS (Amazon Web Service), con la finalidad de conocer los beneficios que se obtendrían en los ámbitos de seguridad, procesamiento, almacenamiento y análisis de los servicios brindados mediante AWS, específicamente en relación con los dispositivos IoT, ya que Amazon web Service cuenta con servicios que son de uso específico para dispositivos de internet de las cosas (IoT), para facilitar el registro, organización, monitorización, control, actualización y administración de manera remota en una escala industrial para la solución de posibles problemas de funcionalidad.

El presente se desarrolla con el fin de brindar un aporte de conocimientos sobre las tecnologías que están revolucionando las industrias como lo son las plataformas en la nube y los dispositivos IoT, ya que permiten que las industrias se acoplen a los cambios tecnológicos, y también para demostrar los beneficios de implementar tecnologías innovadoras y emergentes, además de la facilidad de acceso y gestión de información.

1.6 Alcance de la investigación

El presente trabajo tiene como finalidad elaborar un prototipo de sistema basado en la nube para la gestión de datos de variable agroclimáticas de temperatura y humedad, se utilizará para la elaboración de estos métodos:

Descriptivo: ya que se explicarán los procesos y funciones para la gestión de datos en la nube y la recopilación de información mediante dispositivos IoT.

Explicativo: ya que se pretende detallar el funcionamiento de los sistemas basados en la nube.

Exploratorio: Se considerarán nuevas tendencias para la gestión de datos.

Se realizará la implementación del prototipo, a manera de ejemplo del funcionamiento correcto de los dispositivos utilizados en el presente trabajo de titulación para demostrar su correcto funcionamiento en una versión de pruebas.

MARCO TEÓRICO
CAPÍTULO I

2 CAPÍTULO I: MARCO TEÓRICO

En el presente capítulo se abordarán conceptos teóricos relacionados con los sistemas basados en la nube; así como, un análisis de los principales proyectos de investigación con el fin de establecer una propuesta que responda a la finalidad de este trabajo.

2.1 Agricultura 4.0

Se conoce como Agricultura 4.0 a la evolución sucesiva de la agricultura de precisión y hace referencia a todas las intervenciones que se activan en agricultura, gracias a un análisis preciso y puntual de los datos y la información recopilada y transmitida mediante herramientas y tecnologías avanzadas (McCormick, 2021).

La agricultura 4.0 comprende de todas las herramientas y estrategias que permiten hacer un uso sinérgico de una serie de tecnologías digitales 4.0, pertenecientes a la industria 4.0 que, a su vez, permiten la recopilación automática, la integración y el análisis de los datos procedentes del campo, de los sensores o de terceros (McCormick, 2021).

El objetivo de estas tecnologías es ofrecer el mayor y más preciso apoyo al agricultor en su proceso de toma de decisiones sobre su actividad y la relación con en la cadena de suministro, además, aumentar la rentabilidad y la sostenibilidad económica, medioambiental y social de los procesos agrícolas (McCormick, 2021).

Cuando se habla sobre la Agricultura 4.0, se hace referencia al uso de Internet de las Cosas (IoT), Big Data, Inteligencia artificial y robótica para ampliar, agilizar y hacer más eficientes las actividades que afectan a la cadena de producción (McCormick, 2021).

Dentro de los beneficios de implementar agricultura 4.0 se encuentra:

- Evitar los desperdicios calculando con exactitud los requerimientos de los cultivos o detectando con anticipación la aparición de enfermedades de las plantas o la presencia de parásitos
- Tener mayor control de los costes de producción y planificar con precisión todas las etapas de cultivo, siembra y cosecha, con un ahorro considerable de tiempo y dinero

- Mejorar la trazabilidad de la cadena de suministro, manteniendo el proceso de producción bajo control, lo que conduce a una cadena de suministro corta que, con poco margen de error, es capaz de producir alimentos de la más alta calidad y de forma sostenible (McCormick, 2021).

2.1.1 Las tecnologías de la agricultura 4.0 y la gestión de los datos

Como se mencionó anteriormente, al mencionar a la agricultura 4.0 se habla de:

- uso de las tecnologías más innovadoras
- capacidad de monitorear la cantidad de datos e información procedente de los campos
- capacidad de interpretarlos de manera útil para el sector

Los dispositivos tecnológicos más utilizados para incluir la digitalización en la agricultura son:

- **Drones y Sensores:** Los drones son pequeños vehículos no tripulados que pueden supervisar los cultivos en tiempo real y transmitir imágenes e información útil. Son utilizados principalmente para la cartografía del terreno, los más avanzados son capaces de utilizar sensores y cámaras infrarrojas para detectar problemas que no se pueden detectar a simple vista. Los sensores ambientales son ubicados en los campos y pueden registrar datos meteorológicos y climáticos e información sobre los requerimientos hídricos del terreno.
- **Internet de las cosas:** Tecnología que permite la conexión de varias herramientas diferentes (por ejemplo, drones, sensores o satélites) y a su vez, se comuniquen entre sí para intercambiar información y datos útiles con el fin de mejorar las condiciones de desarrollo de los cultivos.
- **Big Data:** Se refiere al conjunto de información y datos generados por las distintas tecnologías en funcionamiento y que facilitan las decisiones en el ciclo productivo eficientemente. Los datos son diferentes debido a que proceden de distintas fuentes y tienen que ser procesados posteriormente por medio de la inteligencia artificial para que sean útiles y puedan dar respuestas concretas a determinados problemas.

- **Inteligencia artificial:** Se refiere a la tecnología que da instrucciones a las máquinas para evaluar situaciones específicas y tomar decisiones en tiempo real. La acumulación y la capacidad de procesar e interpretar grandes cantidades de datos es el alimento principal para la instrucción de las propias máquinas (*machine learning*).

Existen dos ámbitos principales de aplicación:

- la robótica mediante máquinas que automatizan determinadas tareas
- los softwares de gestión que reducen las horas de trabajo de los empleados en tareas automáticas y repetitivas.
- **Nube:** Conjunto de servicios accesibles y recursos compartidos en la red; es una herramienta útil para garantizar el acceso a determinadas tecnologías y datos a un número mayor de personas (McCormick, 2021).

2.2 Cloud Computing

Otro de los componentes que pueden utilizarse dentro la agricultura es el Cloud computing, este permite el acceso a recursos compartidos de computación bajo demanda en red. Este componente se complementa con las otras tecnologías disponibles para la agricultura para mejorar sus procedimientos (Sánchez Prado, 2021).

2.2.1 Arquitectura de cloud computing

El cloud computing puede dividirse en dos secciones comunicadas por Internet:

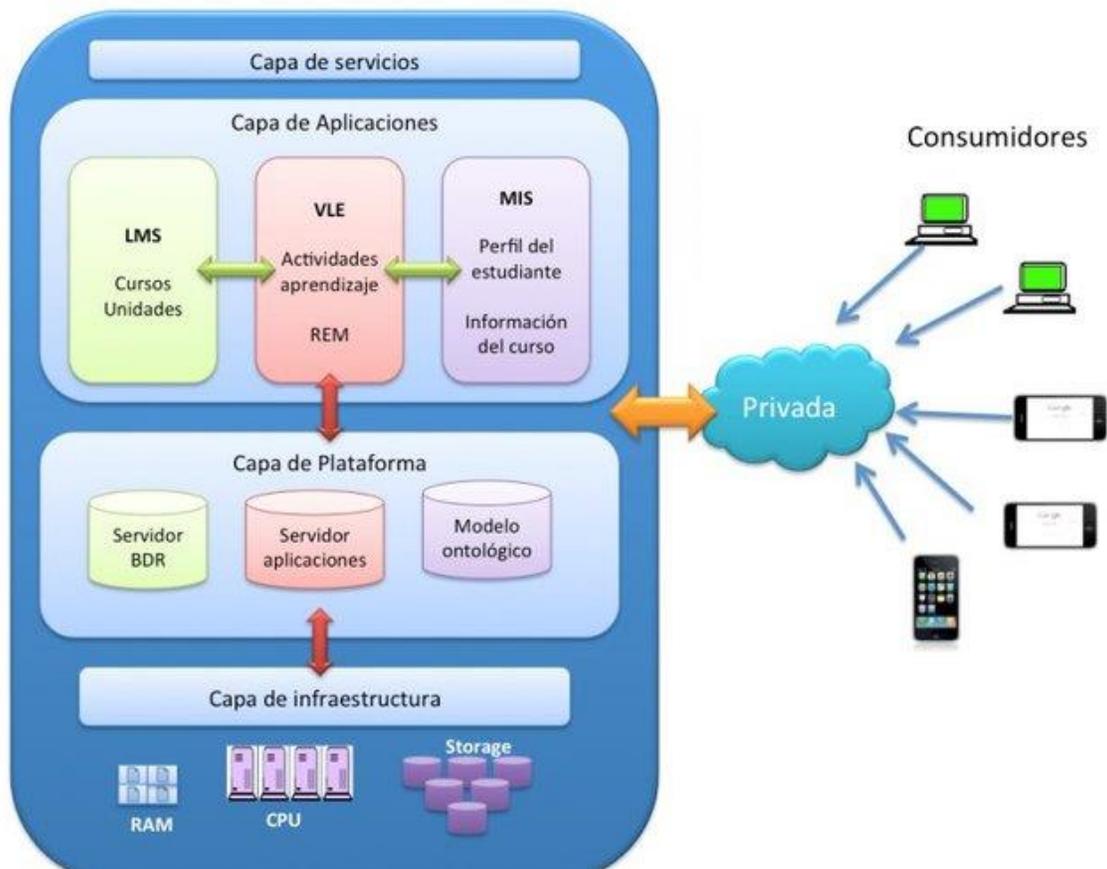
Arquitectura front-end: es la orientada al usuario, contiene el Software e Interfaz de usuario, consiste en la parte visible ejecutable por el usuario, donde realiza las interacciones con el sistema. Y el dispositivo del cliente, debido a que se realizan los procesamientos a través de la nube carecen de potencia informática.

Arquitectura BackEnd: la parte bajo responsabilidad del proveedor del servicio en la nube. Se compone por:

- La aplicación que se pone a disposición del usuario final. Donde se coordinan las necesidades del cliente con los recursos en la parte BackEnd.

- Servicio, es el desempeño de tareas de computación en la nube. Ejemplo, tenemos: almacenamiento, servicios web o entornos de desarrollo de aplicaciones.
- Almacenamiento de los datos requeridos para permitir la ejecución del software en la nube.
- Servicios de administración de los recursos de forma eficiente para conseguir la funcionalidad del sistema de manera fluida.
- Seguridad, es necesario para garantizar la integridad del servidor ante fugas de datos o ataques. Así como, la realización de respaldos de seguridad (Ortiz Monet, 2019)

Figura 1 *Arquitectura del cloud computing*



Nota: Tomado de Ortiz Monet (2019).

2.2.2 Modelos de servicios de cloud Computing

Los servicios ofrecidos en la nube presentan tres posibilidades:

- SaaS (Software as a Service): permite al usuario hacer pleno uso de aplicaciones o software del proveedor SaaS, ejecutados en la infraestructura de la nube (ej. El correo electrónico web).
- PaaS (Platform as a Service): aquí el usuario posee la capacidad de desplegar sus propios aplicativos en la plataforma, se suelen ofrecer también herramientas de programación. Es por esto que el usuario posee poder en la aplicación, pero carece de control en la infraestructura subyacente de la nube.
- IaaS (Infrastructure as a Service): este servicio presenta mayor control del usuario en la aplicación, sin embargo, es el usuario el que gestiona comunicación, procesamiento y almacenamiento. (Ortiz Monet, 2019).

Figura 2 Modelos de servicios del cloud computing



Nota: Tomado de Flores (2016)

2.2.3 Modelos de implementación de Cloud Computing

Existen cinco modelos de implementación del cloud computing:

5. **Nube Privada:** Se brinda de uso exclusivo para determinada organización. Se suele implementar dentro de la misma empresa, pero puede, en ocasiones, encontrarse en una ubicación externa.
6. **Nube Comunitaria:** La nube es ofertada a una comunidad de organizaciones con necesidades similares. Al igual que en la anterior puede ubicarse dentro o fuera de la empresa, y su gestión igualmente puede estar dada por una o varias de las empresas de la comunidad o directamente gestionado por un tercero o una combinación de ambas
7. **Nube Pública:** Presenta la característica de que posee restricciones. Puede ser propiedad y administrada por una o más organizaciones empresariales, académicas, gubernamentales o administrativas. Su ubicación se suele dar en el proveedor de los servicios de la nube
8. **Nube Híbrida:** se conforma por 2 o más de los modelos de nubes, pero vinculadas bajo una misma tecnología estandarizada o propietaria, permitiendo la portabilidad de datos y aplicaciones (Ortiz Monet, 2019).

Debido a la necesidad de una gestión rápida, ágil y eficiente para el manejo de grandes volúmenes de datos generados por los diferentes IoT, surgen plataformas de computación en la nube que ofrecen estos servicios orientados a internet de las cosas (Ortiz Monet, 2019).

2.2.4 Plataformas de Cloud computing Open source

Existen plataformas que ofrecen servicios de gestión y almacenamiento en la nube de código abierto, es decir son gratuitos, entre los cuales, se detallan los más destacados:

- **Kaa IoT Platform** :permite desarrollar IoT de tipo empresariales a cualquier escala, esta tecnología middleware posee código abierto. Ofrece soluciones de diferentes tipos enfocados en IoT.

Los puntos fundamentales que caracterizan a Kaa IoT Platform son:

- Cuenta con arquitectura de micro servicios, debido a la cual posee gran capacidad de personalización, estos se pueden integrar y reorganizar con otros sistemas
- Acorde al criterio de los desarrolladores, puede implementar aplicaciones con casi cualquier lenguaje de programación
- También posee servicios de Kaa en centros de datos, máquinas físicas o virtuales, infraestructuras de nube pública, híbrida o privada. Los servicios de la nube facilitan el control absoluto del sistema, incrementando su seguridad
- Permite el uso de protocolos IoT como son MQTT y Coapa. Siendo MQTT el protocolo predeterminado utilizado por Kaa.
- Es escalable, elástica y auto-reparable, admite la cantidad de clientes necesarios y restaura ante posibles fallos la plataforma.
- En la parte de seguridad, TLS o DTLS, estos certificados por defecto son utilizados en la comunicación Kaa con los dispositivos. En cuanto a lo que credenciales se refiere, Kaa ofrece una gestión flexible del ciclo de vida de estas credenciales
- Oferta asistencia de puerta de enlace, significando que se puede optar en la conexión directa de los dispositivos, bien de forma individual o de forma multiplexada, en la cual, múltiples dispositivos utilizan la misma conexión del servidor.
- Al dar servicio de los proveedores, oferta asistencia profesional y soporte de producción (Ortiz Monet, 2019).
- **Penito:** plataforma de middleware de código abierto que unifica IoT con los servicios de la nube. La infraestructura cuenta con una configuración flexible y permite crear algoritmos para la recolección y filtración de diferentes flujos de datos, que provienen de objetos conectados a internet

Entre sus capacidades más importantes se encuentra:

- Capacidad de monitorización de estado de los diferentes servicios IoT, de cada uno de sus datos obtenidos y el estado de sus sensores. Todo esto

- dada la oferta de registro, implementación y descubrimiento de sensores según su ubicación y tipo
- La interoperabilidad semántica de estos servicios IoT, así como sus flujos de datos en la nube, son asegurados
 - Posibilita la visualización de los datos correspondientes de los IoT en diferentes formatos, dado que presenta autenticación y autorización para su realización
 - Facilita el desarrollo de aplicaciones de baja complejidad sin necesidad de programar, lo cual vuelve más sencillo dicho proceso de creación
- **ThingSpeak:** Plataforma de código abierto que permite la recolección de los datos obtenidos desde sensores, además de su futuro almacenamiento, visualización y análisis. Así también, permite la creación de prototipos y desarrollo de sistemas IoT sin necesidad de configuración de los servidores o desarrollo de software web.

Entre las principales capacidades de esta plataforma se detalla:

- Los dispositivos cuentan con una configuración sencilla, reciben los datos y estos son enviados a la nube mediante el uso de protocolos IoT populares. ThingSpeak almacena en canales privados los datos, pero es posible también el compartir estos datos por medio de canales públicos
- Facilita análisis de los datos obtenidos. Esta característica la diferencia de otras plataformas, ya que proporciona acceso a MATLAB. Posibilitando datos gráficos, diagramas, mediadores y facilita la obtención de algoritmos, modelos y patrones predictivos
- Posteriormente a la obtención de datos y su análisis permite obtener respuestas automáticas según su resultado obtenido, esto puede crear alertas y comunicaciones a través de terceros servicios (Ortiz Monet, 2019).

2.3 Plataformas de cloud computing No open Source

Además de plataformas gratuitas u open source, también existen plataformas de pago o de código cerrado, entre las cuales se detallan las más populares:

- **Amazon Web Service (AWS):** este servicio fue creado con la finalidad de recopilar, almacenar y analizar datos obtenidos por los dispositivos previamente conectados a internet, en el cual se pueden crear también aplicaciones dirigidas a los usuarios con el fin de controlar los dispositivos. Para su realización, AWS establece una comunicación bidireccional mediante sensores, actuadores o aparatos inteligentes y de la nube de AWS.

Elementos característicos:

- Gateway de dispositivos, permite comunicar de forma segura y eficaz con AWS IoT.
 - posee notificaciones de mensajes, utilizando a MQTT o MQTT sobre WebSocket para la publicación y recepción entre los dispositivos y aplicaciones AWS IoT.
 - utilización de motor de reglas para procesar estos mensajes y el envío de los datos a los demás servicios de AWS.
 - engloba servicios de seguridad e identidad, cubriendo mediante credenciales el envío de datos desde los dispositivos.
 - registra los dispositivos, facilitando la organización de estos como recursos asociados a la nube de AWS.
 - Device Shadow, este es un servicio dado, el cual permite la publicación de la información de estado de los dispositivos para su uso en aplicaciones u otros dispositivos conectados.
 - es posible el desarrollo personalizado de autenticaciones.
 - servicio de Jobs, este permite especificar determinado grupo de operaciones remotas a determinados dispositivos para su ejecución (Ortiz Monet, 2019).
- **Microsoft Azure:** propiedad de Microsoft, permite la creación de soluciones de IoT según las necesidades del usuario, mediante el uso de diversas tecnologías

y soluciones propias de la plataforma. Esto supone facilidad para la flexibilización, según necesidades. No posee arquitectura fija en la descripción de la aplicación IoT al hacer uso de la plataforma.

- **Google cloud:** Se encuentra conformada por herramientas que posibilitan la conexión, procesado, almacenamiento y análisis de los datos en la nube. A su vez, permite múltiples sistemas operativos, destacando su funcionalidad con Debian Linux (Ortiz Monet, 2019).

Mediante la revisión y análisis de las características de cada uno de estos servicios en la nube de pago se determinaron los beneficios de cada uno de estos:

Tabla 1 Comparativa servicios en la nube no open source

	AWS	Microsoft Azure	Google Cloud
Flexibilidad y facilidad de Uso	✓	x	✓
Rentabilidad	✓	✓	✓
Velocidad	✓	x	x
Escalabilidad y elasticidad	✓	x	x
Seguridad	✓	✓	✓
Total	5	2	3

Nota: Elaboración propia

Mediante el análisis realizado, es posible determinar que la alternativa más adecuada de servicios en la nube para el presente proyecto es Amazon Web Service.

2.4 Amazon Web Service (AWS)

Amazon web services es uno de los proveedores de servicios en la nube más grande, permite disponer de almacenamiento, recursos de computación, aplicaciones móviles, bases de datos, entre otros. Una ventaja de este tipo de servicios es que no es necesaria

realizar una fuerte inversión en infraestructura para adquirir los servicios, se paga por el uso como suscripción mensual (Giménez, 2020).

AWS, cuenta con una amplia gama de más de 200 servicios integrales a nivel mundial, que ofrecen desde tecnologías de infraestructura, almacenamiento, bases de datos, hasta tecnologías emergentes como aprendizaje automático e inteligencia artificial, lagos de datos, análisis e internet de las cosas. Lo cual permite migrar aplicaciones existentes a la nube con rapidez, facilidad y rentabilidad.

AWS cuenta con el entorno informático en la nube más flexible y seguro disponible. La infraestructura cumple con requisitos de seguridad del ejército, bancos y otras organizaciones con requisitos estrictos de confidencialidad, cuenta con certificaciones y auditorías, como: PCI DSS nivel 1, FISMA Moderate, HIPAA Y SOC 1, ISO 27001 y auditoría SOC 2. También, cuentan con un respaldo amplio de herramientas de seguridad en la nube con 230 servicios y características de conformidad y gobernanza. Tiene compatibilidad con 90 estándares de seguridad y certificaciones de conformidad y los 117 servicios de AWS que almacenan datos de clientes, cumplen la función de cifrado (*¿Qué es AWS?*, 2021).

2.4.1 Servicios de AWS

Como se ha mencionado con anterioridad, Amazon Web Service cuenta con una amplia gama de servicios, entre los cuales, y los que se utilizarán en el presente trabajo se encuentran:

- IoT core AWS: Permite conectar dispositivos de IoT a los servicios de AWS y a otros dispositivos, proteger datos e interacciones, procesar y actuar sobre datos de dispositivos y habilita aplicaciones para que interactúen con otros, aunque no estén conectados (AWS, 2022).
- API Gateway AWS: Es un servicio de AWS para crear, publicar, mantener, monitorear y proteger a las API REST, HTTP y WebSocket (AWS, 2022).
- Lambda AWS: Es un servicio que permite ejecutar código sin aprovisionar ni administrar servidores. El código es ejecutado en una infraestructura de computación de alta disponibilidad (*¿Qué es AWS Lambda? - AWS Lambda*, 2022).

- DynamoDB AWS: Servicio de base de datos NoSQL que ofrece un rendimiento rápido, predecible y escalable (*¿Qué es Amazon DynamoDB? - Amazon DynamoDB, 2022*).
- S3 AWS: Amazon Simple Storage Service (Amazon S3), es el servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento (*AWS | Almacenamiento de datos seguro en la nube (S3), 2022*).

2.5 Gestión de datos en la nube

Una gestión eficiente no solo es el almacenamiento y protección de las bases de datos, también abarca el acceso a estos datos, debido a que la no accesibilidad imposibilita a los administradores el desarrollo de estrategias y productos innovadores utilizando como fundamento datos históricos. La administración de datos se refiere al proceso y estructura por medio de la cual una empresa administra y almacena las bases de datos de gran volumen, incluyendo datos de empresas privadas, datos de sus clientes, inventario de productos, información de la cadena de suministros, así como de sus empleados e incluso de la competencia.

Estos factores, a excepción del almacenamiento, no se encuentran centralizados, es decir, los datos se ubican en la nube. La administración de datos en la nube se define como un sistema de gestión de datos cuyo almacenamiento se da en plataformas de la nube en lugar de ser de manera local, ya que los datos son almacenados de manera externa, la administración en la nube facilita soluciones al momento de recuperar información, almacenamiento a largo plazo, soporte profesional y acceso sencillo a usuarios (Fernández, 2018).

La implementación de sensores y controladores en agricultura es un requisito básico para implementar agricultura 4.0. Debido al aporte de información importante para la automatización agrícola. La información captada por los sensores se convierte en información útil mediante los sistemas informáticos de alto nivel, lo cual se envía al ERP o a la Nube para su almacenamiento o gestión (Walter, 2020).

2.6 Variables Agroclimáticas

Existen diferentes tipos de variables en el sector agrícola, las variables agroclimáticas tienen una mayor incidencia en las etapas de desarrollo y producción del cultivo, estas son la temperatura, precipitación, viento, altitud.

A continuación, se detalla cada una de estas variables:

- **Temperatura.** Este factor climático influye en la vegetación y el tipo de paisaje circundante. Niveles elevados o demasiado bajos de temperatura condicionarán la siembra.
- **Aire:** El aire es una mezcla de gases en la atmósfera. El oxígeno y el dióxido de carbono en el aire son de particular importancia para la vida humana y la vegetación. El oxígeno en la respiración para la producción de energía que es utilizada en diversos procesos de crecimiento y desarrollo. El dióxido de carbono es una materia prima en la fotosíntesis de las plantas.
- **Humedad relativa:** La cantidad de vapor de agua que puede contener el aire depende de su temperatura; el aire caliente tiene la capacidad de contener más vapor de agua que el aire frío. Es la cantidad de vapor de agua en el aire, expresada como la proporción (en porcentaje) de la cantidad máxima de vapor de agua que puede contener a cierta temperatura.
- **Viento:** El movimiento del aire o el viento se debe a la existencia de presión gradiente en una escala global o local causada por las diferencias en el calentamiento. A escala mundial que consiste en el flujo de corriente en chorro y movimiento de grandes masas de aire. En la escala local solamente una menor cantidad de aire se mueve. Los vientos superficiales son más bajos y menos turbulentos por la noche debido a la ausencia de calefacción solar (Col, s. f.).

Para el presente trabajo, las variables a considerar para las lecturas serán las de Humedad y Temperatura, debido a que son las que mayor afectación representan para los cultivos debido a los distintos terrenos con los que se cuenta en el país.

2.6.1 Variables con señales digitales y analógicas

Las variables industriales emiten dos tipos de señales:

- **Digitales:** Procesa las señales analógicas y las convierte en digital, procesa la señal analógica para realizar los cálculos y procesar la variable. Estas se pueden interconectar con otros dispositivos y permiten un almacenamiento más preciso, rápido, eficiente y asequible, que se realiza conectando un dispositivo a una centralita, se complementan perfectamente con software de control.
- **Analógicas:** Son aquellas utilizadas para procesar y calcular una variable y emplea la señal exactamente como llega, como ejemplo están los instrumentos que utilizan agujas, numeración, mecánica. Estas no se conectan a otros dispositivos, lo cual hace que el control y medición se realicen manualmente por los operadores (Rubio, 2019).

2.7 Dispositivos IoT para agricultura

Entre las soluciones IoT más utilizadas en la agricultura se encuentran:

- Drones de monitoreo de Cultivos
- Sensores
- Sistemas de posicionamiento global por GPS
- PLC para control de sensores
- Placas base para control de sensores
- Tractores inteligentes

Para el presente proyecto, se realizará el desarrollo de sistema para el monitoreo y control de variables de tipo agroclimáticas, por lo tanto, las opciones más viables de hardware son: los sensores, la placa y el PLC.

Los dispositivos PLC tienen costos elevados por lo que las alternativas más viables es el uso de sensores mediante una placa, los cuales se detallan a continuación.

2.8 Sensor DHT11

El DHT11 es un sensor con una alta fiabilidad y estabilidad debido a su señal digital. En la versión sin PCB se tiene 4 pines y en la versión con PCB tenemos 3 pines (*Cómo utilizar el DHT11 para medir la temperatura y humedad con Arduino, s. f.*).

Los pines de la versión sin PCB del DHT11 son:

- VCC: alimentación
- I/O: transmisión de datos
- NC: no conectado, pin al aire
- GND: conexión a tierra

Los pines de la versión con PCB del DHT11 son:

- GND: conexión con tierra
- DATA: transmisión de datos
- VCC: alimentación

2.9 PLACA ESP32

ESP32 es una serie de SoC (por sus siglas en inglés, *System on Chip*) y módulos de bajo costo y bajo consumo de energía creado por *Espressif Systems*.

Los ESP32 poseen un alto nivel de integración. En su pequeño encapsulado se incluyen:

- interruptores de antena
- *balun* de RF
- amplificador de potencia
- amplificador de recepción de bajo ruido
- filtros y módulos de administración de energía

Además de todo eso, logra un consumo de energía muy bajo a través de funciones de ahorro de energía que incluyen sincronización de reloj y múltiples modos de operación.

Todo esto lo convierte en la herramienta ideal para tus proyectos energizados con baterías o aplicaciones IoT (Carmanate, s. f.).

**METODOLOGÍA DEL PROCESO DE DESARROLLO DE LA PROPUESTA
TECNOLÓGICA
CAPÍTULO II**

3 CAPÍTULO II: METODOLOGÍA DEL PROCESO DE DESARROLLO DE LA PROPUESTA TECNOLÓGICA

En este capítulo se muestra la metodología de investigación, métodos de recopilación de información, enfoque de investigación y variables sobre los cuales se desarrollará este trabajo.

3.1 Tipo de investigación

La presente propuesta tecnológica será de tipo descriptivo, debido a que consideran otras investigaciones relacionadas como referencia y, además, se explicarán los procesos y funciones para la gestión de los servicios en la nube AWS (Amazon web Service); también es de tipo explicativa, debido a que busca detallar el funcionamiento de los sistemas de gestión de datos de tipo agroclimática en la nube; y de tipo exploratorio, porque se considerará nuevas tendencias y la gestión de datos.

3.2 Enfoque de la investigación

La investigación emplea enfoque cualitativo, ya que la información nombrada está comprendida por información escrita previamente analizada, basada en los datos obtenidos durante el proceso de investigación.

3.3 Periodo y lugar de desarrollo de la propuesta Tecnológica

La propuesta se desarrollará en la ciudad de Guayaquil, Ecuador, durante el periodo Julio a septiembre del año 2022, por medio de un prototipo para la gestión de datos de variables agroclimáticas.

3.4 Definición y comportamiento de variables

3.4.1 Conceptualización de variables

- Variable dependiente: Sistema en la nube de Amazon web Service
- Variable Independiente: Gestión de datos de variables agroclimáticas

3.4.2 Operacionalización de variables

Tabla 2 Tabla de operacionalización de variables

Variables	Conceptualización Práctica	Indicadores	Técnicas Y Herramientas
<p>Variable Independiente Gestión de datos de variables Agroclimáticas</p>	<p>La gestión de datos de variables agroclimáticas corresponde al monitoreo, gestión, almacenamiento de datos (Col, s. f.).</p>	<p>-Optimización de información -Datos recibidos por dispositivos industriales</p>	<p>Configuración de placa ESP32 con sistema en la nube de Amazon web Service para la gestión correcta de datos de tipo agroclimáticas emitidos por sensores DHT11 IoT.</p>
<p>Variable Dependiente Sistema en la nube de Amazon web Service</p>	<p>Amazon web Service (AWS) es una plataforma en la nube que permite almacenar sistemas a nivel mundial, además, cuenta con múltiples servicios que se adaptan a las necesidades de cada negocio (¿Qué es AWS?, 2021).</p>	<p>- Capacidad de almacenamiento -Funcionalidades</p>	<p>Configuración de placa ESP32 con sistema en la nube de Amazon web Service para la gestión correcta de datos de tipo agroclimáticas emitidos por sensores DHT11 IoT.</p>

Nota: Elaboración propia

3.5 Métodos de recopilación de información e instrumentos a utilizar

Para la recopilación de información, se realizó un análisis de los principales conceptos que tienen relación con los temas a tratar en esta propuesta tecnológica.

3.5.1 Componentes de hardware

Los dispositivos utilizados para la conexión y traspaso de datos a la nube son los siguientes:

- Placa ESP32
- Sensor DHT11 AM2302 de temperatura y humedad
- Cable USB tipo B

3.5.2 Componentes de Software

En cuanto a los componentes de software a utilizar, se trabajará con:

- IDE Arduino
- AWS (Amazon Web Service)
- IoT Core AWS
- API Gateway AWS
- Lambda AWS
- Dynamodb AWS
- S3 AWS
- HTML/CSS

Acorde a lo estudiado en la sección teórica, la tecnología que mejor se adecua a los objetivos planteados en la presente propuesta tecnológica, son los componentes de hardware y software descritos anteriormente, ya que garantizan el uso adecuado de los componentes de internet de las cosas con el protocolo MQTT, el cual a su vez, facilita

la conexión de varios dispositivos y tiene la facilidad de comprobar la recepción del mensaje al objetivo, proporciona seguridad a través de certificados electrónicos.

Adicionalmente a todo lo anterior, la selección de los componentes de hardware y software se debe también, a la disponibilidad de los equipos en el país por parte de los proveedores existentes y, además, por los costos de estos.

Posterior a la selección de componentes, se procede a la instalación del software IDE Arduino, el cual servirá para configurar el dispositivo PLC con la nube de AWS para el envío de datos de variable industrial.

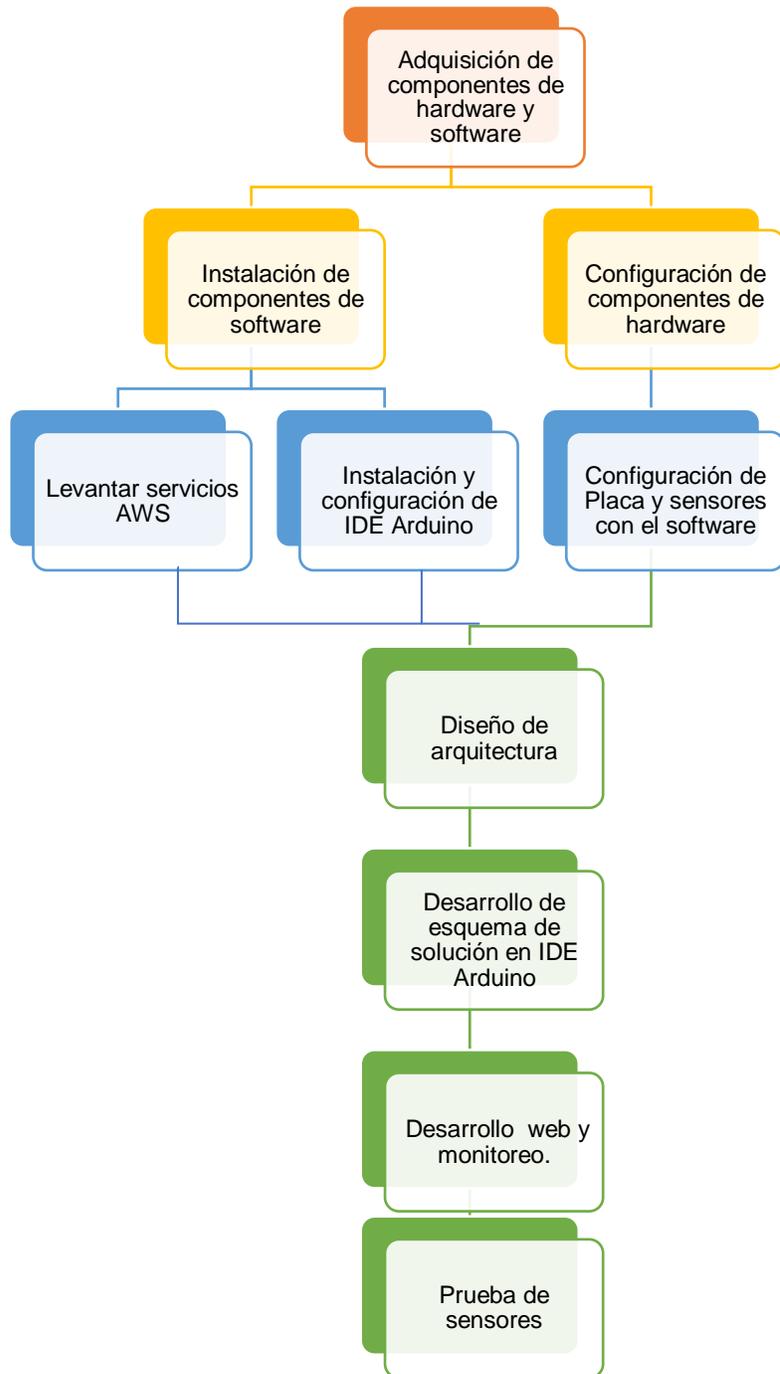
Para acceder al software IDE de Arduino, se realizó la compra de la licencia del programa.

Para acceder a los servicios de Amazon web service, se realiza mediante el siguiente enlace:

- <https://aws.amazon.com/>

A continuación, se detalla el método utilizado para la conexión entre el servidor de Amazon Web Service y la placa ESP32.

Figura 3 Esquema metodológico de la propuesta



Nota: El esquema incluye todas las etapas del presente trabajo, desde la instalación, configuración hasta la simulación de la implementación. Elaboración Propia

En base al esquema anterior, se describen los pasos de la propuesta de instalación de los servicios en la nube AWS con IDE Arduino:

3.5.3 Para levantar los servicios en la nube de AWS se siguen los siguientes pasos:

1) Ingresar a la página de Amazon web services

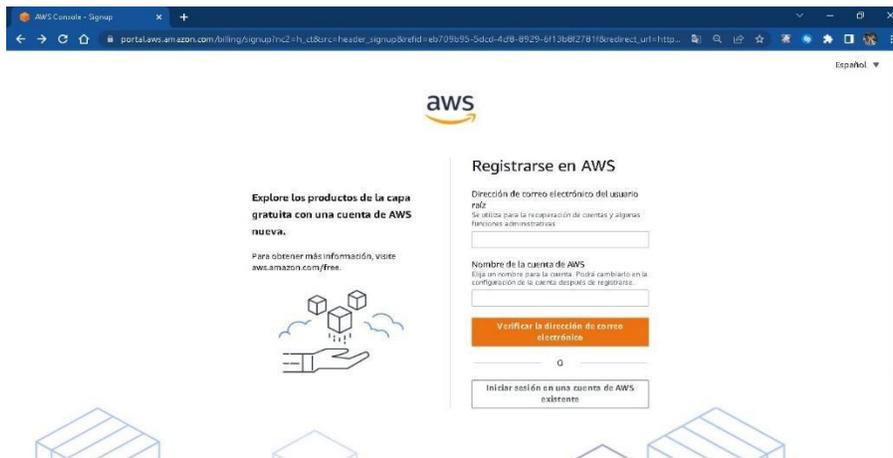
Figura 4 Sitio Web AWS



Nota: Se muestra el sitio de AWS para la creación de la cuenta. Elaboración Propia

2) Se procede a la creación del usuario para acceder a la consola de administración de AWS.

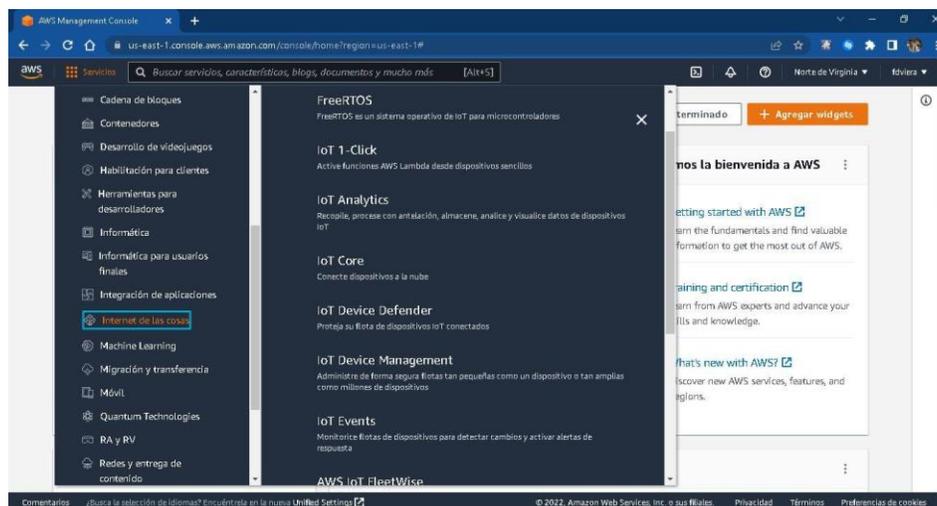
Figura 5 Creación de cuenta AWS



Nota: Se crea el usuario para acceder a los servicios de AWS. Elaboración Propia.

3) Se procede a crear el IoT

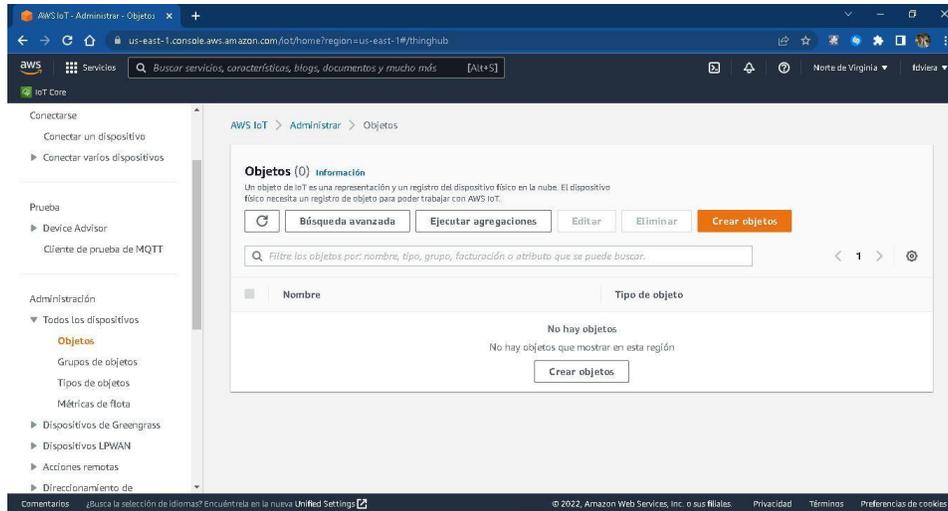
Figura 6 Creación IoT



Nota: Cuando se ingresa, se procede a crear el servicio IoT el cual permite la conexión de la placa ESP32 a la nube de AWS. Elaboración Propia.

4) Creación del Objeto

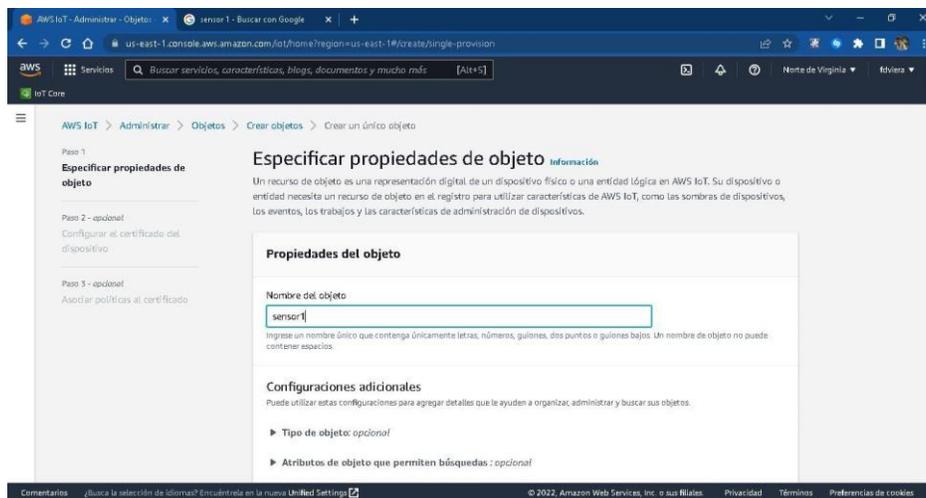
Figura 7 Creación del objeto



Nota: Una vez creado el servicio IoT, se procede a crear los objetos. Elaboración Propia.

5) Registrar las propiedades del objeto

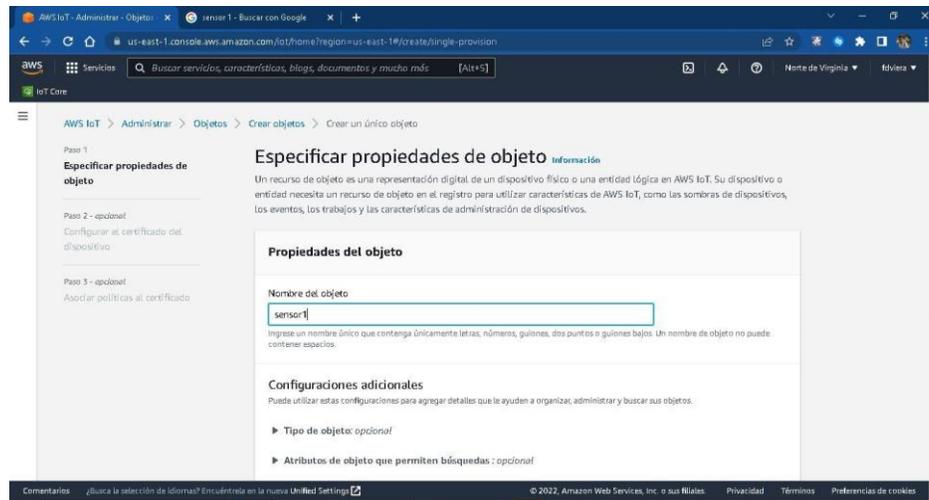
Figura 8 Nombrar objetos



Nota: Una vez creado el servicio IoT, se procede a crear los objetos. Elaboración Propia.

6) Registrar las propiedades del objeto

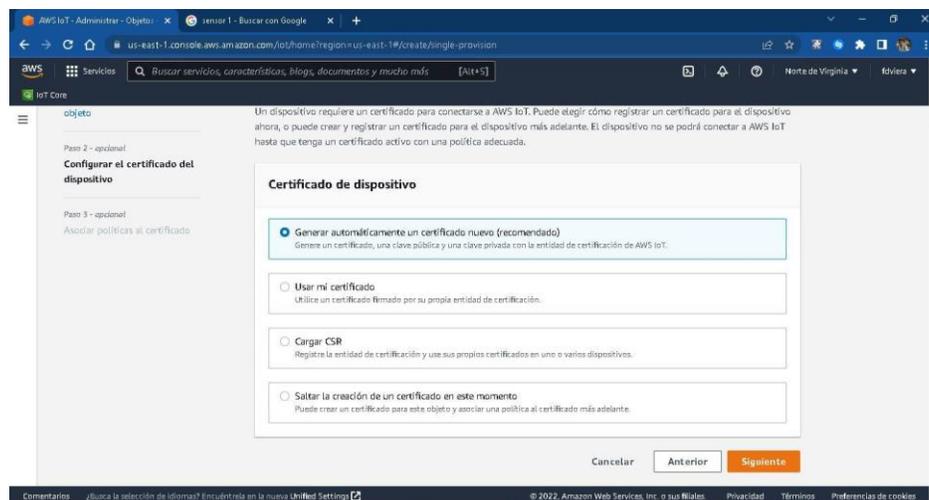
Figura 9 Nombrar objetos



Nota: Se procede a la asignación de las especificaciones de los objetos creados. Elaboración Propia.

7) Crear certificados de dispositivos

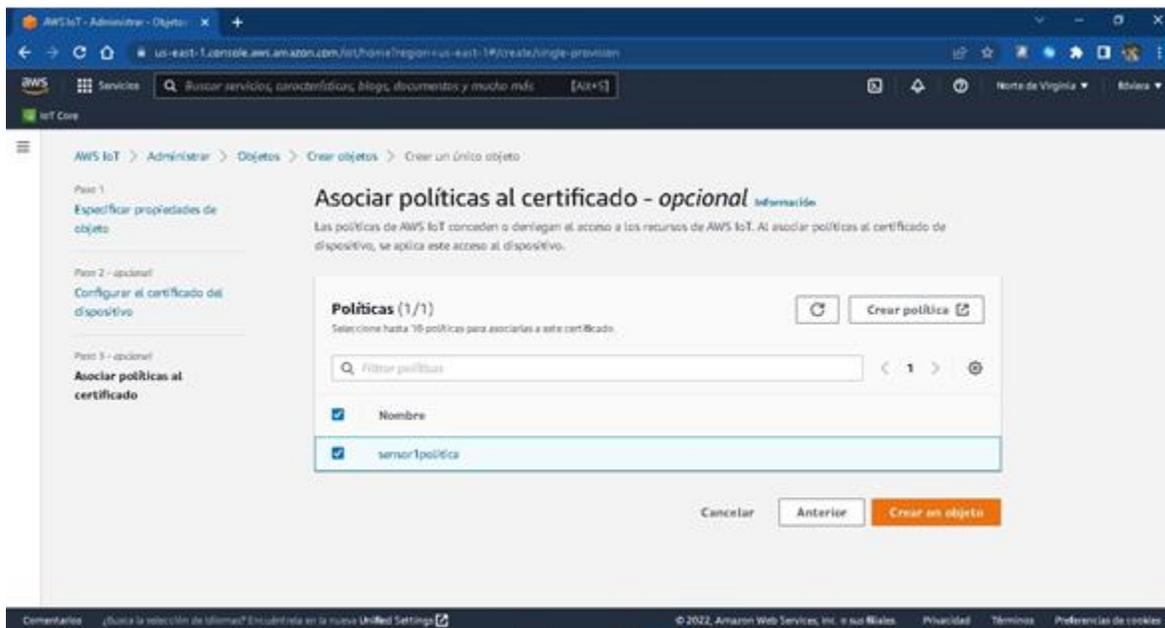
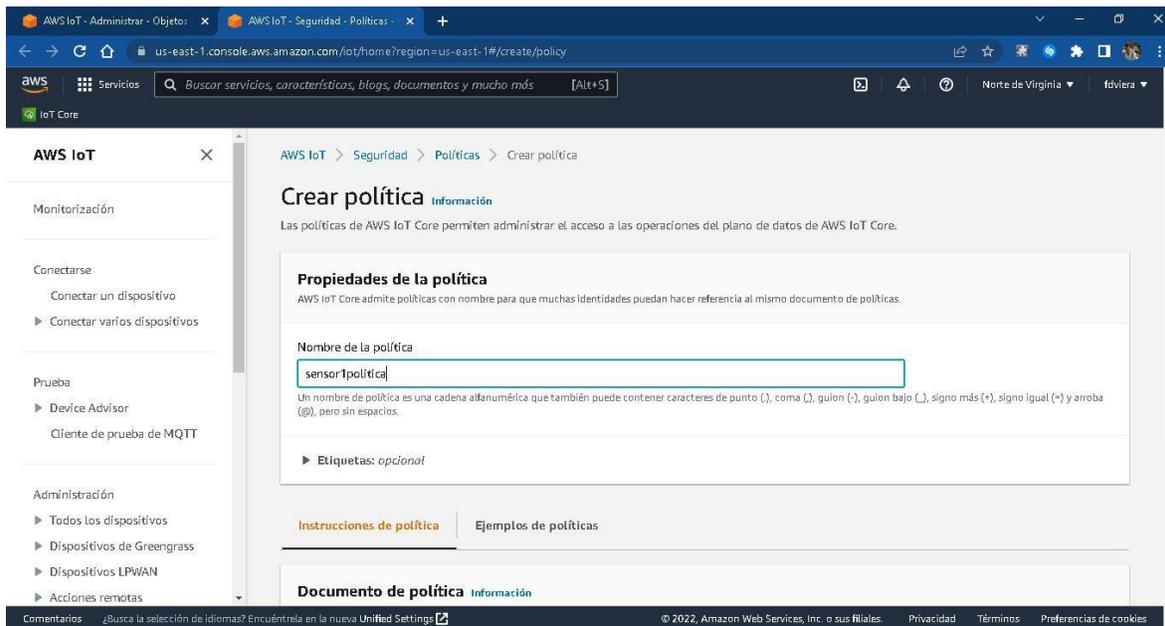
Figura 10 Crear certificados

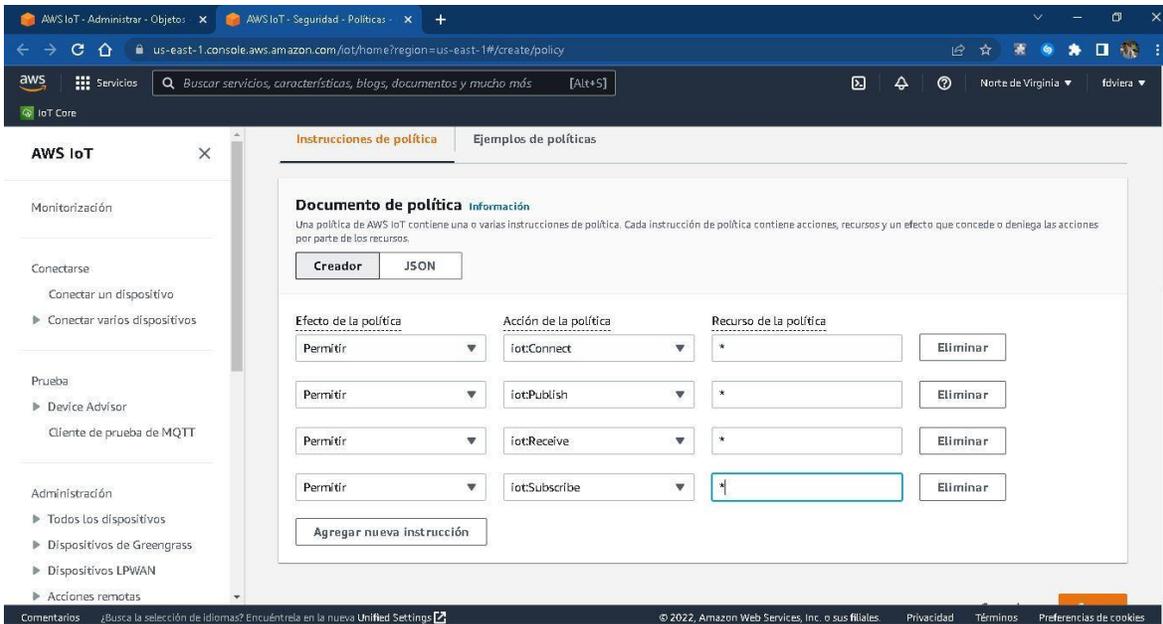


Nota: Se procede a crear los certificados a los objetos creados. Elaboración Propia.

8) Asignación de políticas y permisos

Figura 11 Asignación de políticas y permisos

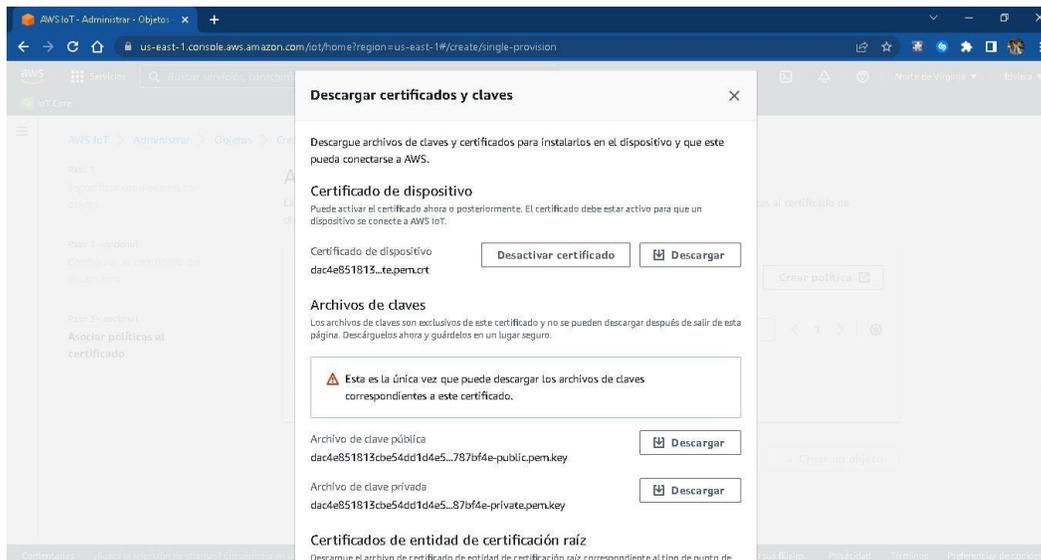


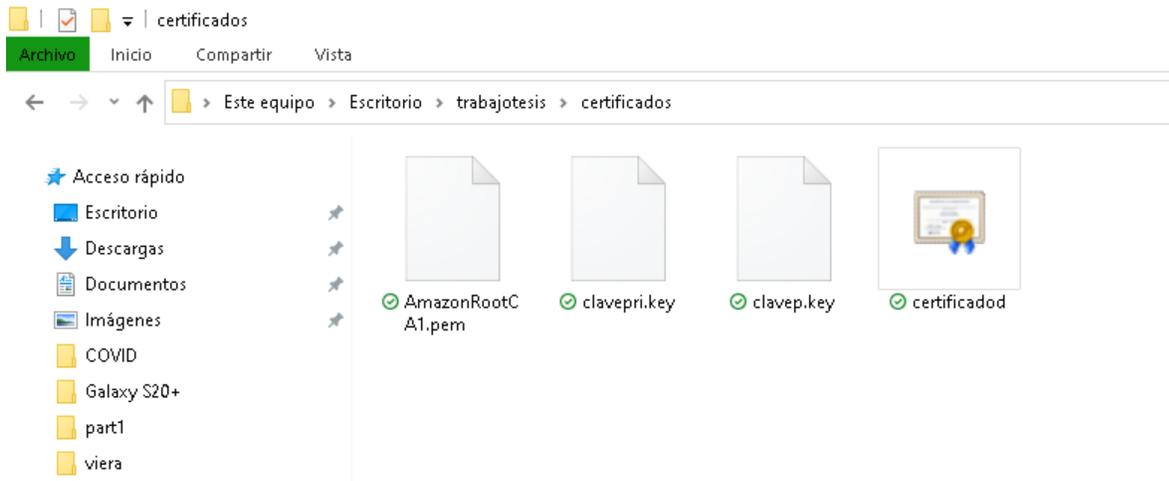


Nota: Se procede a crear y asignar políticas a los objetos. Elaboración Propia.

9) Descargar certificados y clave

Figura 12 Crear certificados y claves

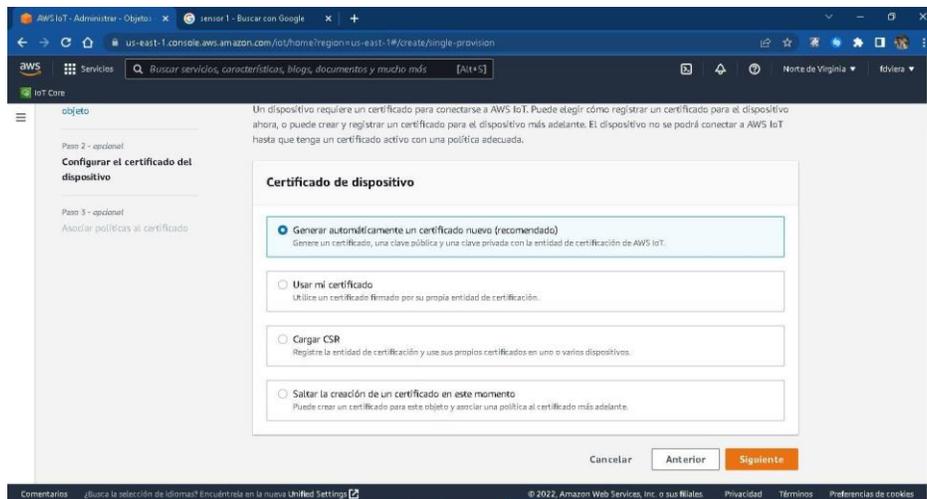




Nota: Se procede a crear y descargar los certificados y claves y además a descargarlos creados. Elaboración Propia.

10) Descargar sistema Arduino

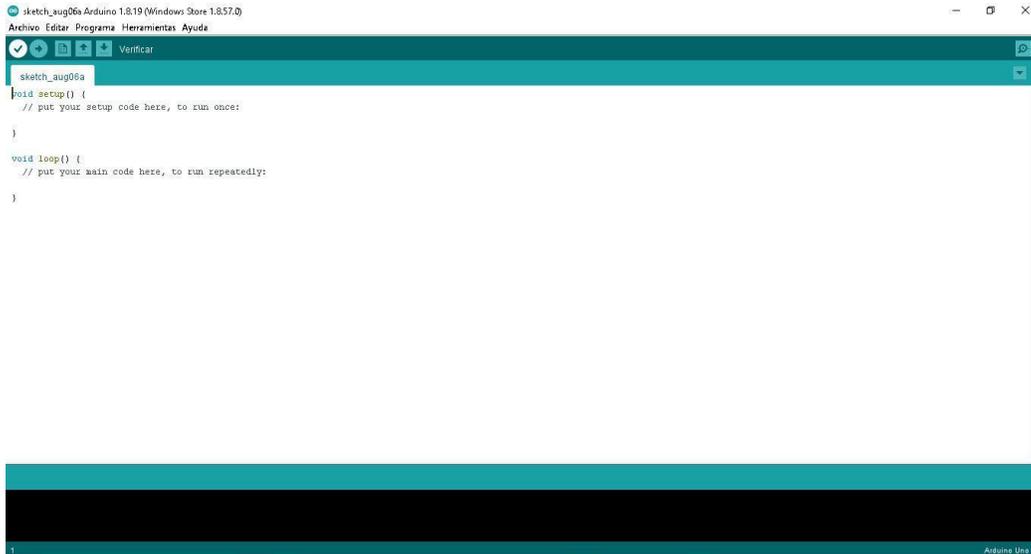
Figura 13 Descarga Arduino IDE



Nota: Se realiza la descarga del sistema Arduino IDE. Elaboración Propia.

11) Crear proyecto en Arduino

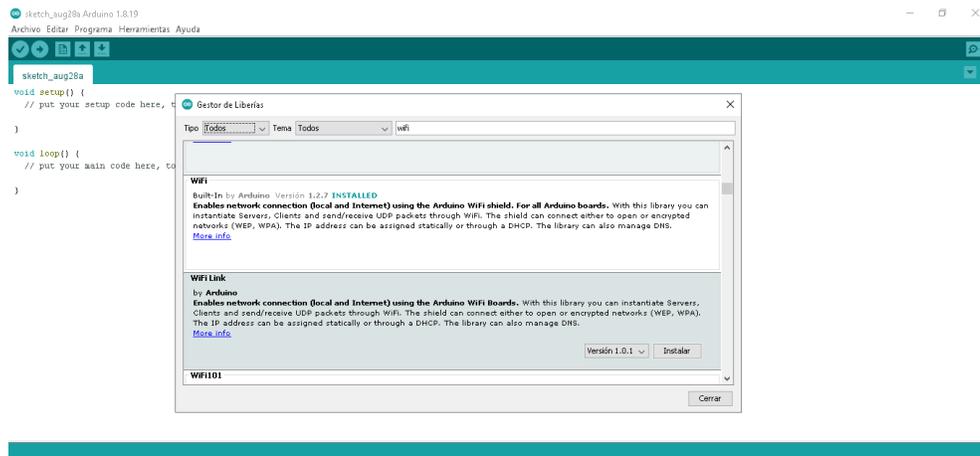
Figura 14 Crear proyecto Arduino IDE



Nota: Se realiza la descarga del sistema Arduino IDE. Elaboración Propia.

12) Descarga de librerías

Figura 15 Se procede a descargar la librería WIFI, WIFI LINK



Nota: Se descargan las librerías del sistema Arduino IDE, WIFI, WIFI LINK. Elaboración Propia.

13) Instalación de librerías

Figura 16 Instalación de librerías WIFI, WIFI LINK

```

prueba2222 Arduino 1.8.19
Archivo Editor Programa Herramientas Ayuda

prueba2222 $
#include <ArduinoJson.h>
#include <WiFi.h>
#include <WiFiLink.h>

#define AWS_IOT_PUBLISH_TOPIC "viera2020" // nombre de objeto
#define AWS_IOT_SUBSCRIBE_TOPIC "viera2020"

//-----
#define SECRET
#define THINGNAME "ESP32_DHT11_2" // Politica

int pinDHT11 = 14;
SimpleDHT11 dht11(pinDHT11);
float h;
float t;

const char WIFI_SSID[] = "Gengas"; //Red WIFI
const char WIFI_PASSWORD[] = "CasaViera0811"; //CLAVE
const char AWS_IOT_ENDPOINT[] = "ajqgm7k5c15-ats.iot.us-east-1.amazonaws.com"; //Punto de enlace

// Amazon Root CA 1 Certificado
static const char AWS_CERT_CA[] PROGMEM = R"EOF(
-----BEGIN CERTIFICATE-----
MIIDQTCAAgAIBAgI7BwYfS5a/JAo54w841kPa1j2byjAMBgkqk109w0BAQF
ADASMQwCCTDPOQ0eWVUzEPMAGAIUEChR0ZWIha9w8RwPwYDQ0c8B809F6
b24gDa9w4cCGAkd4C7E1MDyYlAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMD
MAGAIUEBhMCYVb0cANBg9VBAcTRFct7qpbjEZMBcGAIUEAAMQV1hea9u1F3v
b3Qg0eQMTCCASi00QY9o2IhwHAAQEBBQABggFADCCAgCggEBAl74g8R6dC
ca9H9F0EFTTl4a2911o91ghFPI0A2v41I4a09j3p0e4T08o8w3e2MgHfzIK
90E1Re6c41I4aASw4r3e4g9H7Z8o4I2p4YV0wF4e0R0w8GcaKs1L1gr
IFAGHtqgl2a+4/s8kaPFDgH3E2B0Vj4u1Vp+ThaM7u1Ih4b4uJ0cAhaabF9a6
Vouj5H5S2z/0epuX0tHhA114gk957EWW67c4c0Xj3GKlhd+cdqg08p8kd1L
93F0am/8p0zylR1A4b597LWb0cc0P0F40CID5yH19Y/OCB/11DEg8w0y0a
jg0ub4r1eg0Aw8AAACAEANw0VROTA08:8ADNA4eS:3AG8w8F08Q8AE8EAMC
ATYw0YVRO08EFTI0Yt1U071wM1J0u0Pac7I0T9p1MA0GCSGQSIb3DQEBCqUA
A4EBAQY3da02Ch6v42SggM1M0cUy06c4LKS3p08/Gvk3U0y6K0S:bxendI
USFBCjjaCXPIET53HTEIUDcU6adTcC2q4eHZEReh1b11Bj)tc/asv0ta0d1W5
N4q563p4a0v08My7h3j3f08eE5F74g9V0v1T094L3V0E113EY0909v0
o4uIQ3VCMY801P8B8j1x8F8Hc42V4c4FYq2R1b142wq3ctAapzY2F6G1030T
S8eI-yR0K0J1oa1dCg3URK642M4ht8V8ob2c3MD42ZhaInoQdeXeGAbkpy
rQ8Ebo0no2e6qSFT7468S0v05
-----END CERTIFICATE-----
)EOF";

// Device Certificado
static const char AWS_CERT_CRTI[] PROGMEM = R"KEY(
Complando programa
    
```

Nota: Se procede a instalar las librerías del sistema Arduino IDE, WIFI, WIFI LINK.
Elaboración Propia.

14) Incluir certificados AWS en Arduino

Figura 17 Instalación de certificados AWS

```

prueba2222 Arduino 1.8.19
Archivo Editor Programa Herramientas Ayuda

prueba2222 $
const char WIFI_PASSWORD[] = "CasaViera0811"; //CLAVE
const char AWS_IOT_ENDPOINT[] = "ajqgm7k5c15-ats.iot.us-east-1.amazonaws.com"; //Punto de enlace

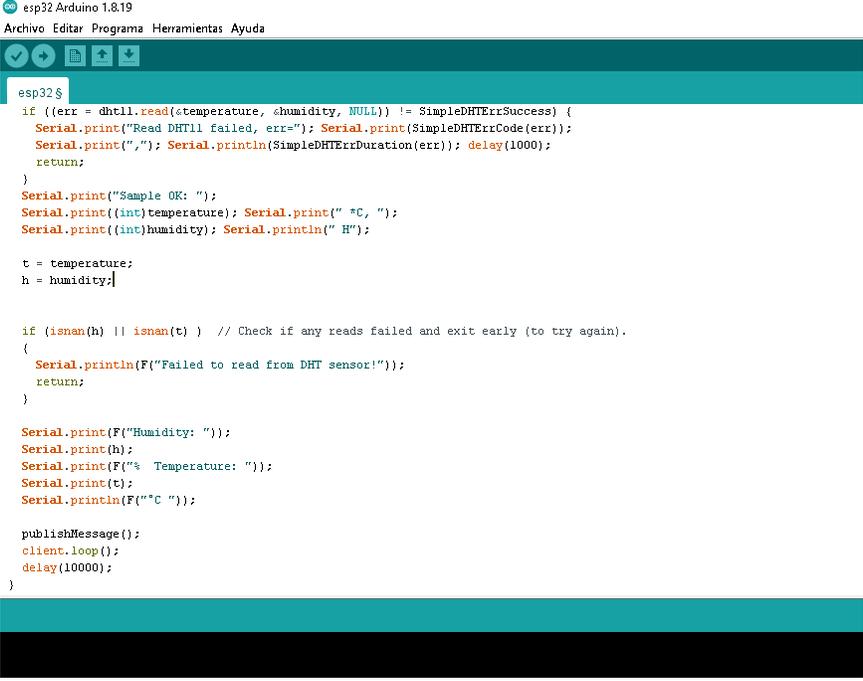
// Amazon Root CA 1 Certificado
static const char AWS_CERT_CA[] PROGMEM = R"EOF(
-----BEGIN CERTIFICATE-----
MIIDQTCAAgAIBAgI7BwYfS5a/JAo54w841kPa1j2byjAMBgkqk109w0BAQF
ADASMQwCCTDPOQ0eWVUzEPMAGAIUEChR0ZWIha9w8RwPwYDQ0c8B809F6
b24gDa9w4cCGAkd4C7E1MDyYlAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMD
MAGAIUEBhMCYVb0cANBg9VBAcTRFct7qpbjEZMBcGAIUEAAMQV1hea9u1F3v
b3Qg0eQMTCCASi00QY9o2IhwHAAQEBBQABggFADCCAgCggEBAl74g8R6dC
ca9H9F0EFTTl4a2911o91ghFPI0A2v41I4a09j3p0e4T08o8w3e2MgHfzIK
90E1Re6c41I4aASw4r3e4g9H7Z8o4I2p4YV0wF4e0R0w8GcaKs1L1gr
IFAGHtqgl2a+4/s8kaPFDgH3E2B0Vj4u1Vp+ThaM7u1Ih4b4uJ0cAhaabF9a6
Vouj5H5S2z/0epuX0tHhA114gk957EWW67c4c0Xj3GKlhd+cdqg08p8kd1L
93F0am/8p0zylR1A4b597LWb0cc0P0F40CID5yH19Y/OCB/11DEg8w0y0a
jg0ub4r1eg0Aw8AAACAEANw0VROTA08:8ADNA4eS:3AG8w8F08Q8AE8EAMC
ATYw0YVRO08EFTI0Yt1U071wM1J0u0Pac7I0T9p1MA0GCSGQSIb3DQEBCqUA
A4EBAQY3da02Ch6v42SggM1M0cUy06c4LKS3p08/Gvk3U0y6K0S:bxendI
USFBCjjaCXPIET53HTEIUDcU6adTcC2q4eHZEReh1b11Bj)tc/asv0ta0d1W5
N4q563p4a0v08My7h3j3f08eE5F74g9V0v1T094L3V0E113EY0909v0
o4uIQ3VCMY801P8B8j1x8F8Hc42V4c4FYq2R1b142wq3ctAapzY2F6G1030T
S8eI-yR0K0J1oa1dCg3URK642M4ht8V8ob2c3MD42ZhaInoQdeXeGAbkpy
rQ8Ebo0no2e6qSFT7468S0v05
-----END CERTIFICATE-----
)EOF";

// Device Certificado
static const char AWS_CERT_CRTI[] PROGMEM = R"KEY(
Complando programa
    
```

Nota: Se procede a instalar los certificados de AWS en el sistema Arduino IDE.
Elaboración Propia.

15)Elaboración del código para la lectura de la temperatura y humedad mediante la placa ESP32

Figura 18 Código para lectura de temperatura y humedad



```
esp32 Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda
esp32 $
if ((err = dht11.read(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {
  Serial.print("Read DHT11 failed, err="); Serial.print(SimpleDHTErrCode(err));
  Serial.print(","); Serial.println(SimpleDHTErrDuration(err)); delay(1000);
  return;
}
Serial.print("Sample OK: ");
Serial.print((int)temperature); Serial.print(" °C, ");
Serial.print((int)humidity); Serial.println(" H");

t = temperature;
h = humidity;

if (isnan(h) || isnan(t) ) // Check if any reads failed and exit early (to try again).
{
  Serial.println(F("Failed to read from DHT sensor!"));
  return;
}

Serial.print(F("Humidity: "));
Serial.print(h);
Serial.print(F("% Temperature: "));
Serial.print(t);
Serial.println(F("°C "));

publishMessage();
client.loop();
delay(10000);
}
```

Nota: Desarrollo de código para la lectura de variables de temperatura y humedad en el sistema Arduino IDE. Elaboración Propia.

16) Compilación de código en Arduino IDE y la placa ESP32

Figura 19 Compilación de código

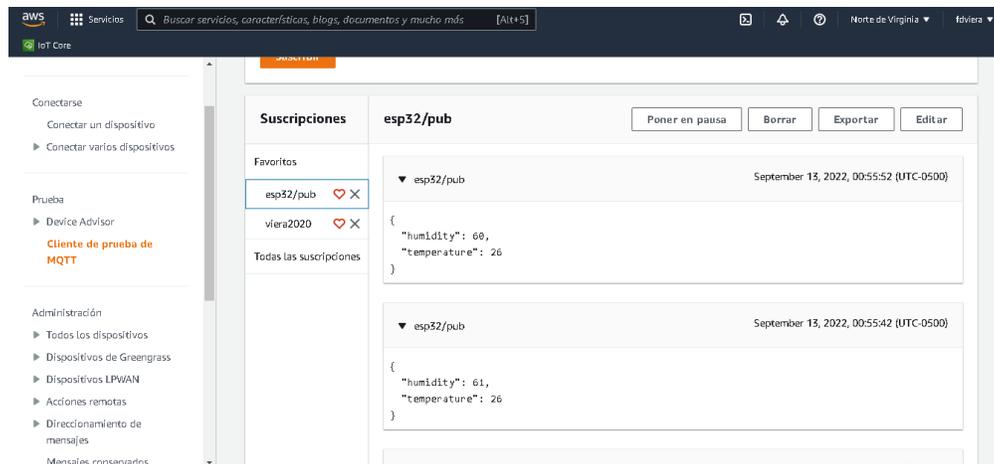


```
prueba22222 Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda
Nuevo
prueba22222
// Device Private Key Certificado
static const char AWS_CERT_PRIVATE[] PROGMEM = R"KI
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAxGJ1QAXeMtv19QthClAVN70Y93T+8+Mc2SI
3PkAr5TrH1iJ1752RtKw/1Guu2cY2XqQ6rNAZbxCDFwMH5MK3dI
JY6sZ3C8Yw+seMdVaqe1FX04yk4N9rPFP3D1jEQ8ewLmHZ+/nu
vYvUDqV3c/4vJe5DQON09i3NBL/1JjCYZK5Zowdiu/itpShkJGf
1Ej3hlSyMRRK6ITZLaWSJqUD5aqb9117wclLbq3V0kZ2k1WBFf
E/DjY6GGebRKvmWZ3I1Mjy2/02N4wx5g5yVxQQIDAQABAoIBAB'
PCInelptyuuLPPRNYff4yWagnb5NBL3rFJmWPDBiETs4kvmD2;
PbdH9mL6J+7/zPZ1dzszsuvEuGEJOAbQxGFOBTyi8GEk54TU11f
tRDHhfFAD6nGpn+Ns9wUd3LpjtwFb/NcrdQNvftioguiUEWb/L
```

Nota: Compilación de código para la lectura de variables de temperatura y humedad en el sistema Arduino IDE. Elaboración Propia.

17) Configuración de placa ESP32 y variables en AWS

Figura 20 Configuración de lectura de variables en AWS



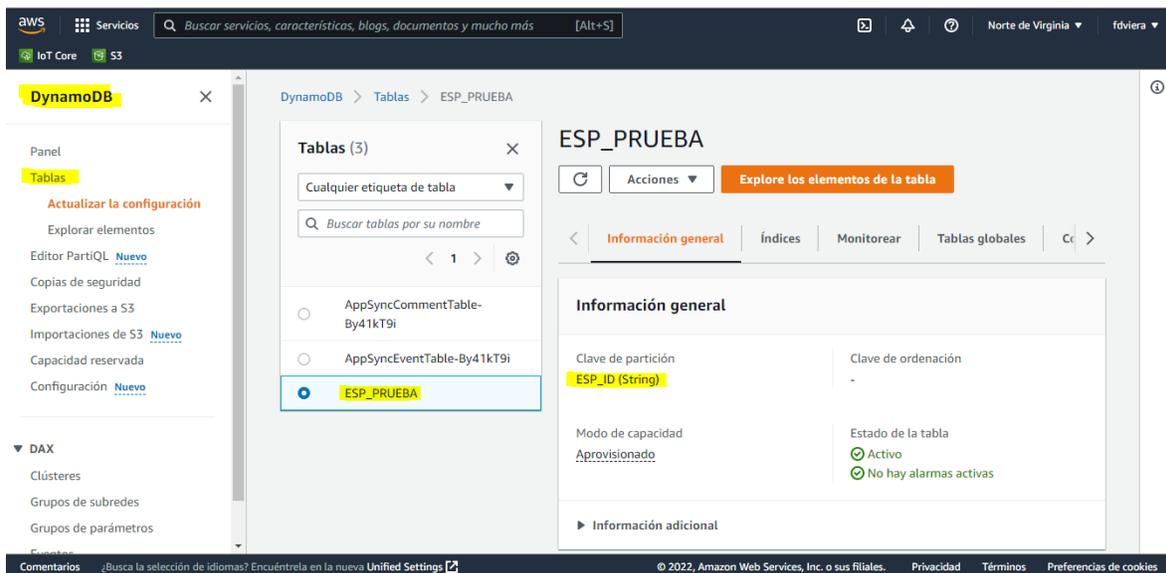
Nota: Se realiza la configuración de las variables a almacenar en AWS mediante las lecturas de la placa ESP32 en el sistema Arduino IDE. Elaboración Propia.

3.5.4 Configuración e integración de AWS con el IDE Arduino y la placa ESP32:

Para la integración de Arduino y AWS, se utilizan diferentes aplicaciones para soportar la conexión y almacenamiento de datos.

18) Base de datos DynamoDB

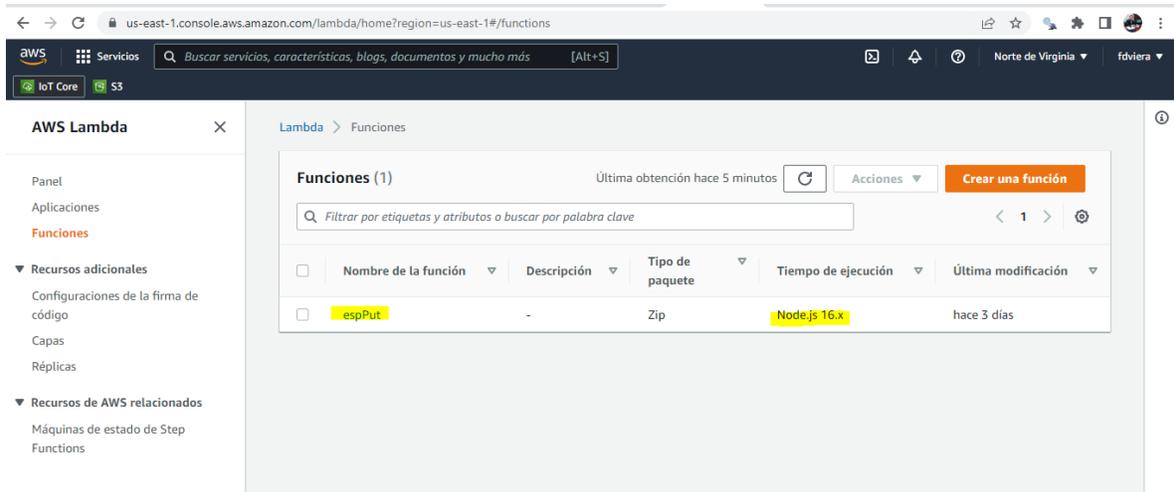
Figura 21 Configuración de base de datos



Nota: En la base de datos, se crea la tabla ESP_PRUEBA, en la cual se almacenará la información de las capturas que se realizan desde Arduino. Elaboración Propia.

19) Lambda

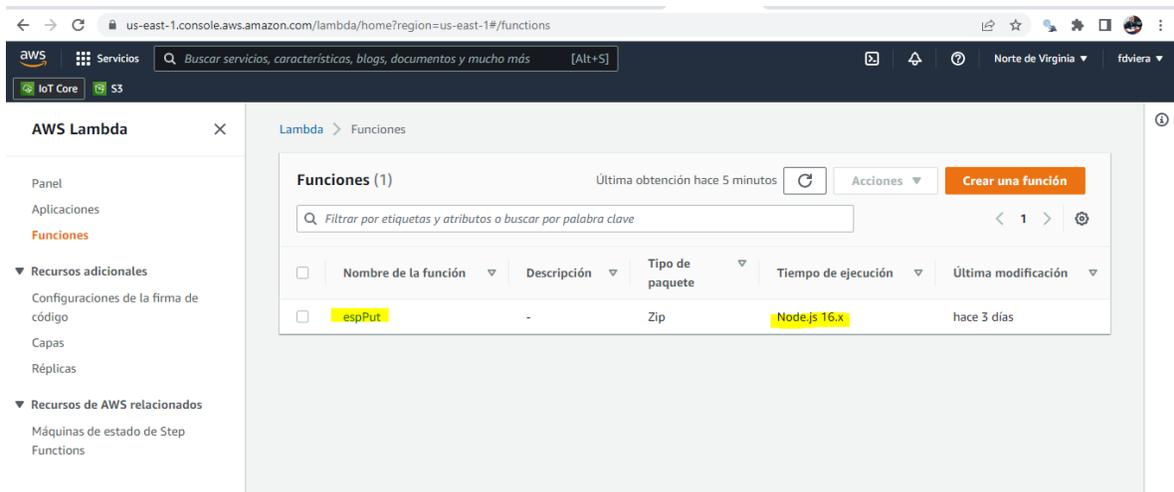
Figura 22 Configuración de lambda



Nota: El servicio Lambda permite la interacción de la base de datos con el SDK de AWS. Elaboración Propia.

20) Crear función en Lambda

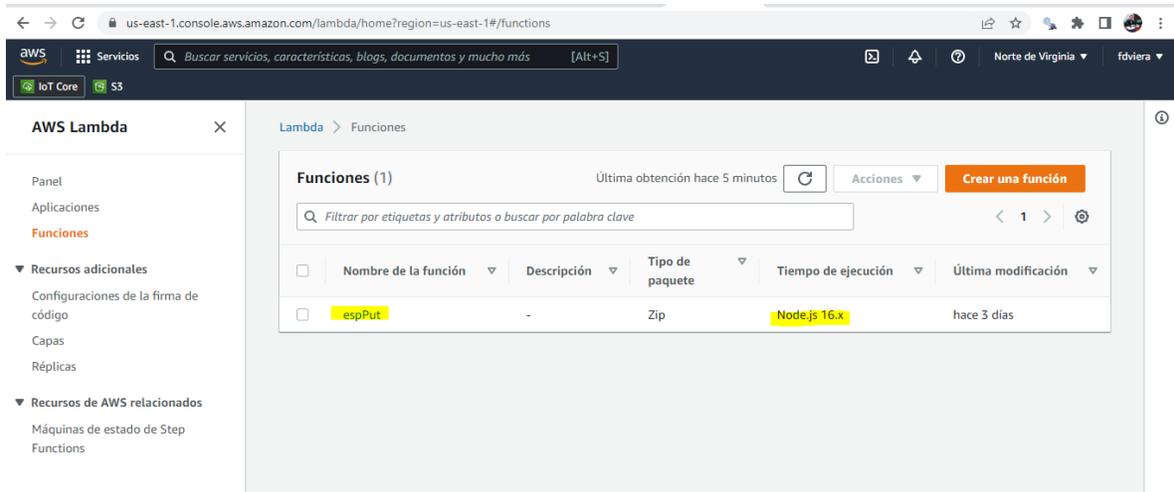
Figura 23 Función espPut



Nota: Se crea la función espPut, en la cual se creará un node js para interactuar en conjunto con la base de datos. Elaboración Propia.

21) Crear función en Lambda

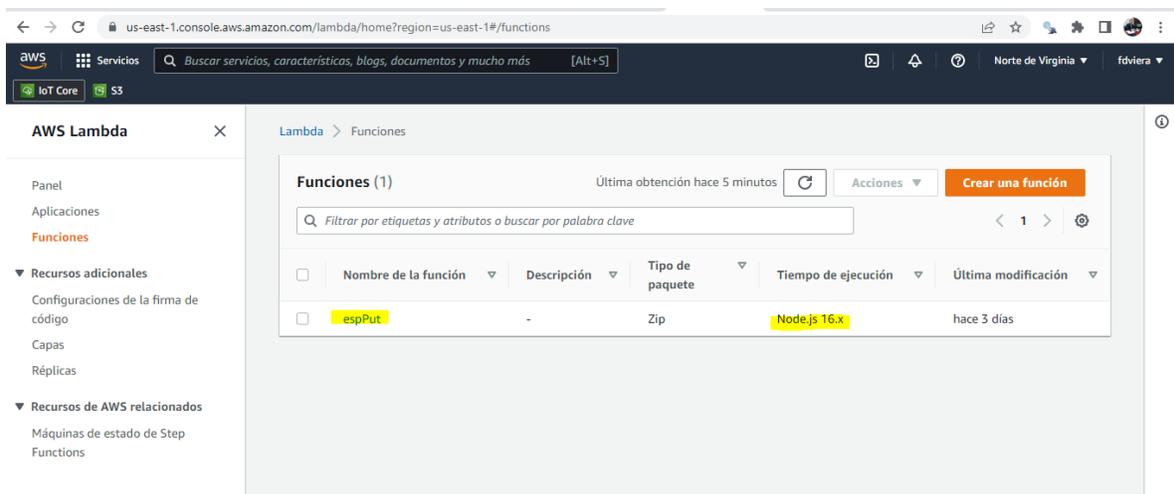
Figura 24 Función espPut



Nota: Se crea la función espPut, en la cual se creará un node js para interactuar en conjunto con la base de datos. Elaboración Propia.

22) Creación de protocolo para comunicación HTTP

Figura 25 Función espPut



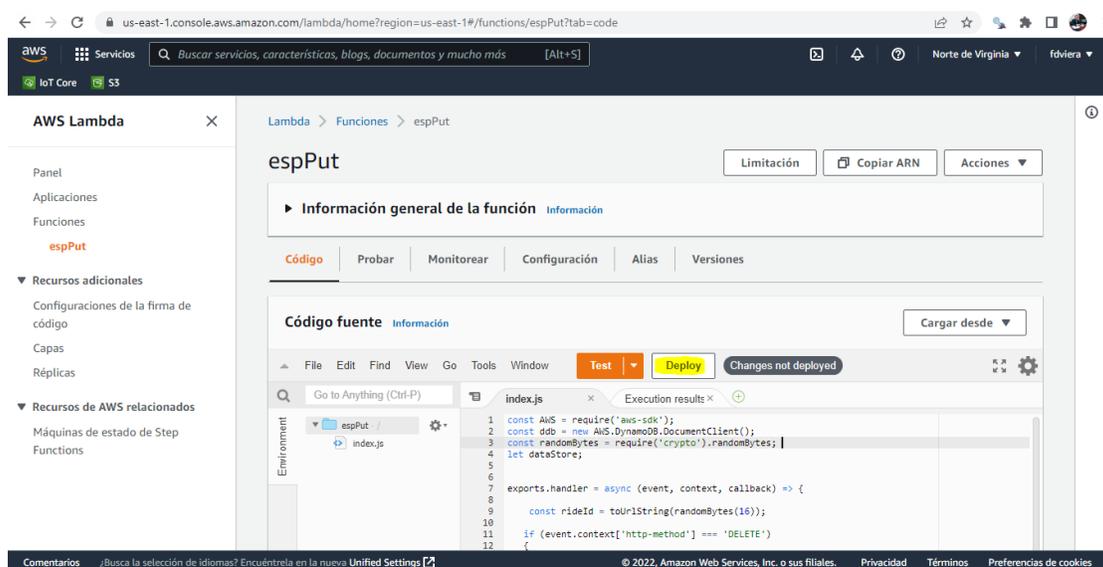
Nota: Elaboración Propia.

23) Creación de archivo index.js

En el archivo index.js, cuyo código fuente se muestra en los anexos, permite la conexión final del sistema.

24) Finalizar configuración de Lambda

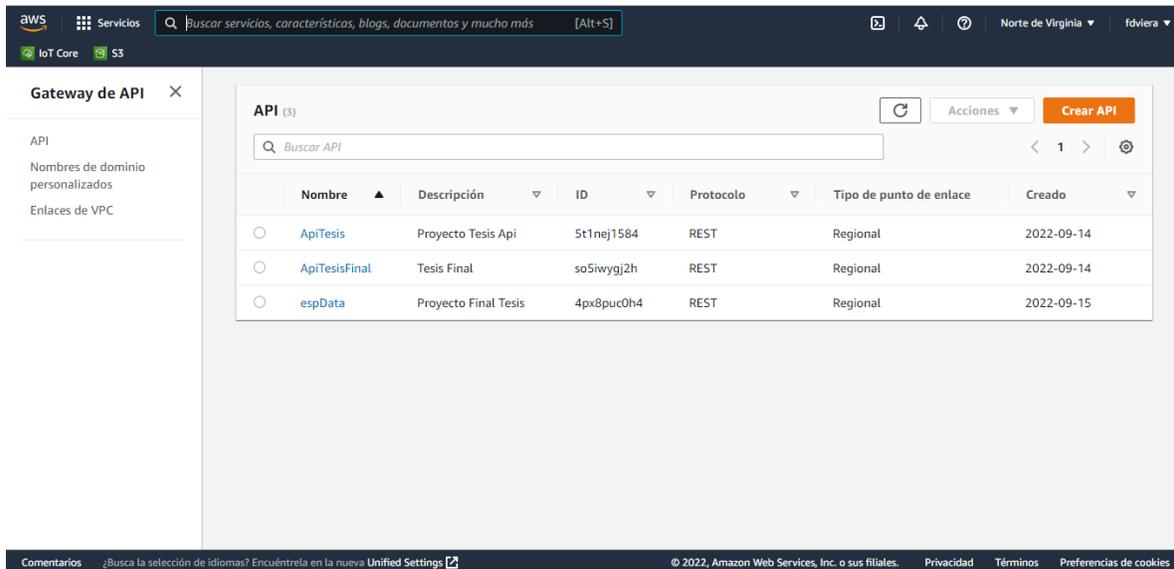
Figura 26 Configuración



Nota: Finalizada la configuración de Lambda, se guarda el archivo y se despliega presionando el botón Deploy. Elaboración Propia.

25) Creación del API Gateway

Figura 27 API Gateway



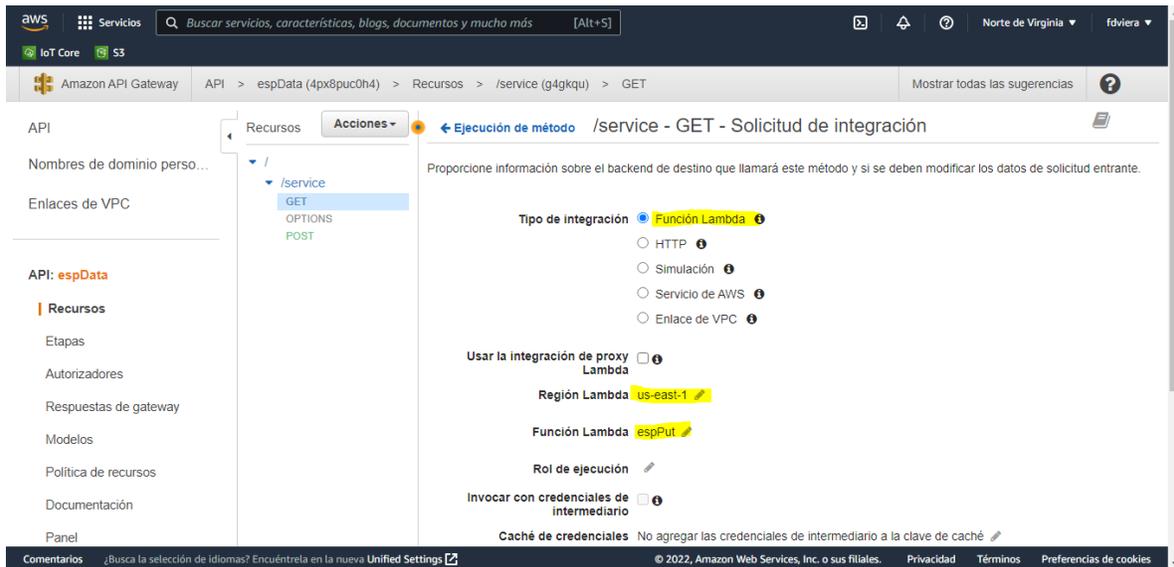
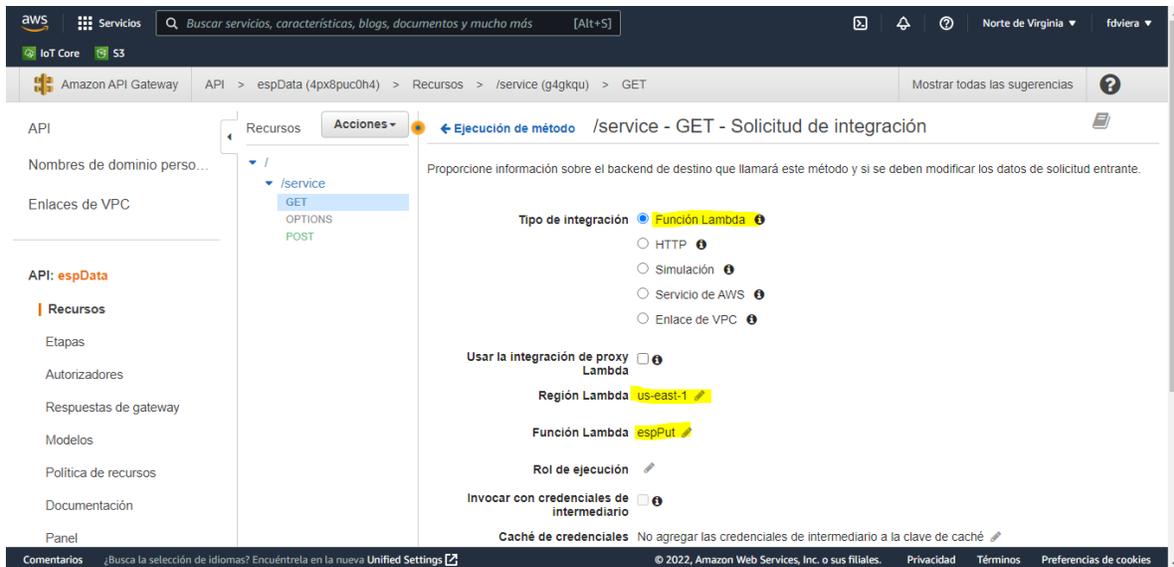
The screenshot displays the AWS API Gateway console interface. On the left, there is a navigation menu with options like 'API', 'Nombres de dominio personalizados', and 'Enlaces de VPC'. The main area shows a list of APIs under the heading 'API (3)'. A search bar is present above the table. The table lists three APIs with columns for Name, Description, ID, Protocol, Endpoint type, and Created date.

	Nombre ▲	Descripción ▼	ID ▼	Protocolo ▼	Tipo de punto de enlace	Creado ▼
<input type="radio"/>	ApiTesis	Proyecto Tesis Api	5t1nej1584	REST	Regional	2022-09-14
<input type="radio"/>	ApiTesisFinal	Tesis Final	so5iwygj2h	REST	Regional	2022-09-14
<input type="radio"/>	espData	Proyecto Final Tesis	4px8puc0h4	REST	Regional	2022-09-15

Nota: Se creó el API rest, con la finalidad que desde la nube AWS, pueda comunicar la interfaz de Arduino y el componente que se encuentra publicado en AWS, además se crearon los métodos POST y GET, para mejorar la comunicación desde Arduino para ser enviado desde los sensores configurados. Elaboración Propia.

26) Configuración de métodos

Figura 28 Configuración de métodos



aws Servicios [Alt+S] Norte de Virginia fdviera

IoT Core S3

Planes de uso
Claves de API
Certificados de cliente
Configuración

▶ Parámetros de ruta URL
▶ Parámetros de cadenas de consulta de URL
▶ Encabezados HTTP
▼ Plantillas de mapeo

Acceso directo de cuerpo de solicitud

Cuando no haya definida ninguna plantilla (recomendado)

Cuando ninguna plantilla coincide con el encabezado Content-Type de la solicitud

Nunca

Content-Type

application/json

➕ Agregar plantilla de mapeo

application/json

Generar plantilla:

aws Servicios [Alt+S] Norte de Virginia fdviera

IoT Core S3

Planes de uso
Claves de API
Certificados de cliente
Configuración

▶ Parámetros de ruta URL
▶ Parámetros de cadenas de consulta de URL
▶ Encabezados HTTP
▼ Plantillas de mapeo

Acceso directo de cuerpo de solicitud

Cuando no haya definida ninguna plantilla (recomendado)

Cuando ninguna plantilla coincide con el encabezado Content-Type de la solicitud

Nunca

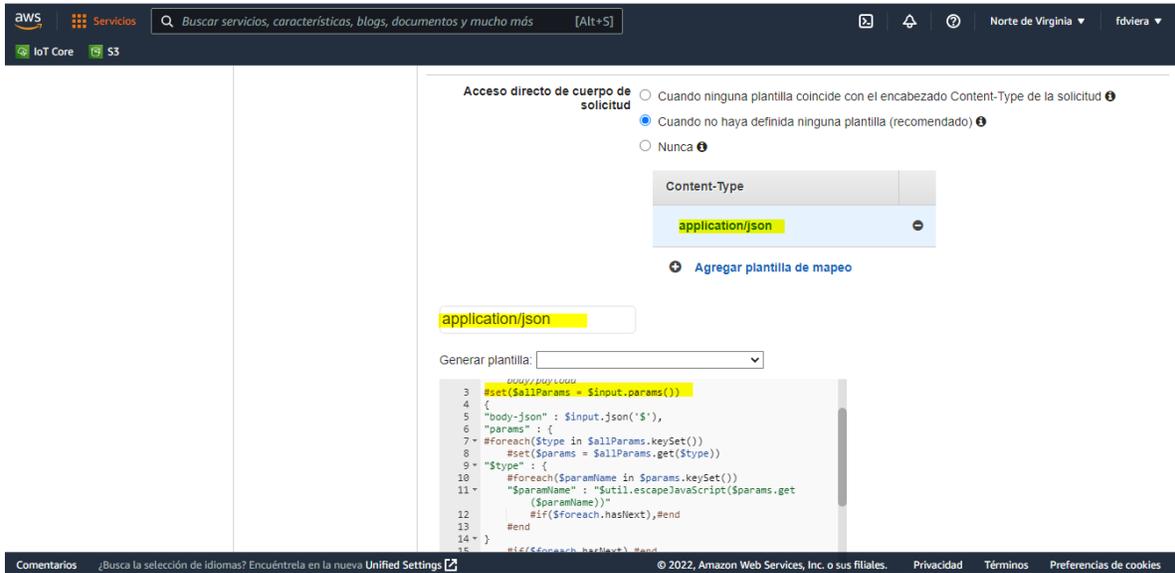
Content-Type

application/json

➕ Agregar plantilla de mapeo

application/json

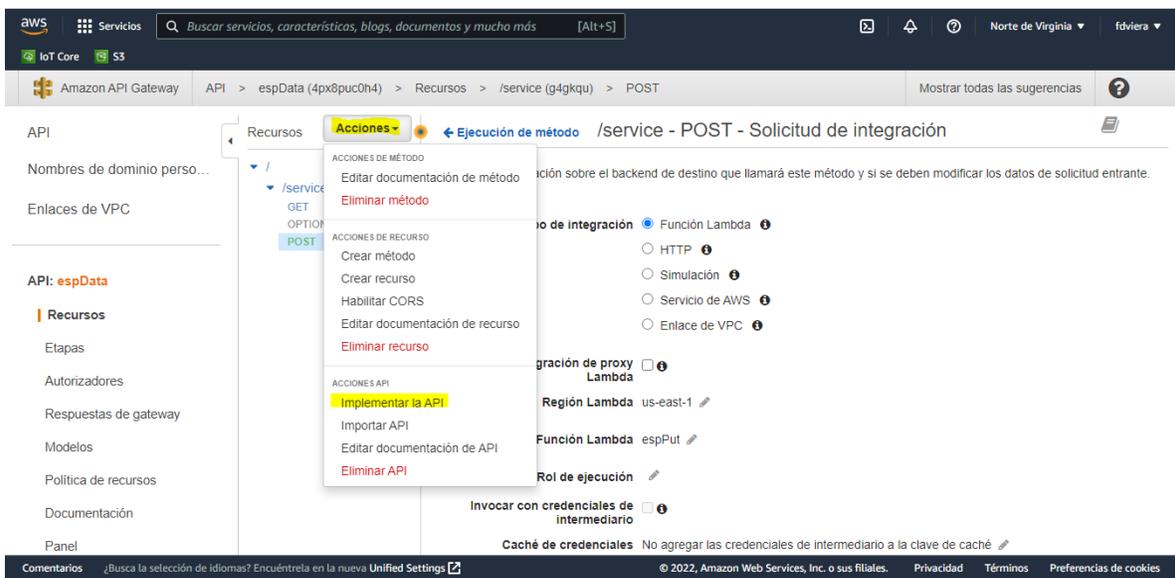
Generar plantilla:

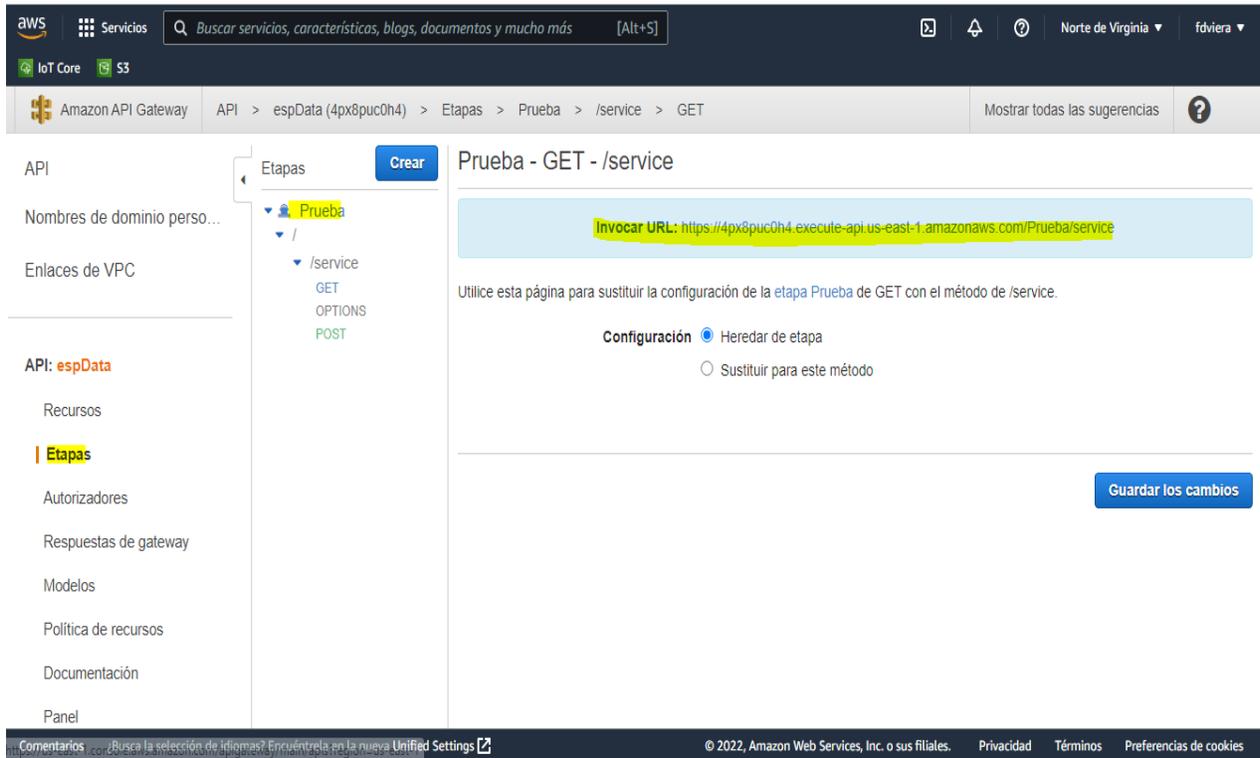
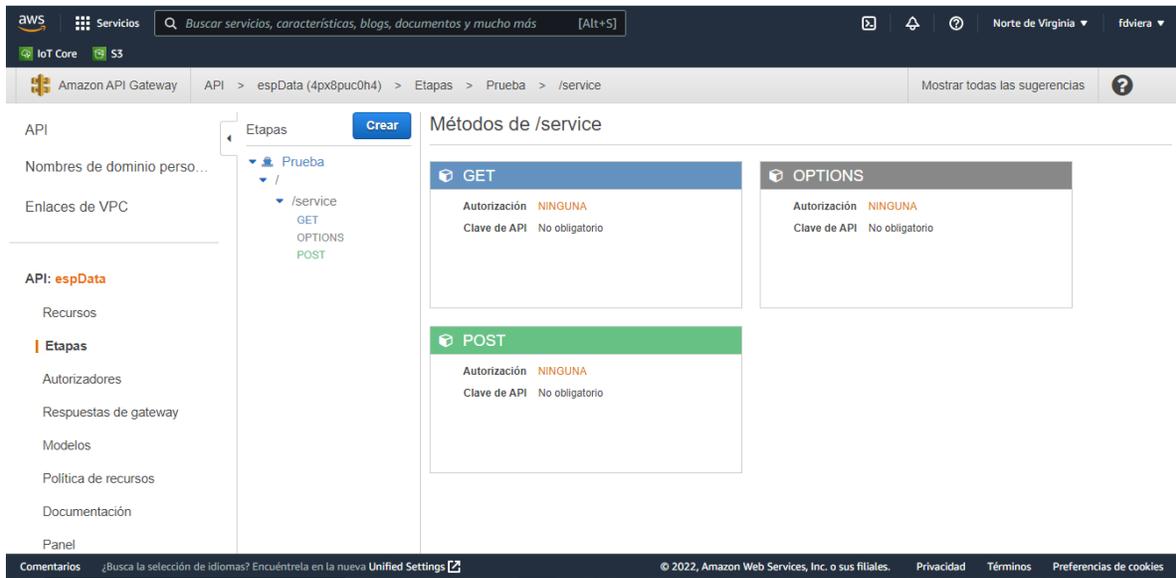


Nota: Se creó el API rest, con la finalidad que desde la nube AWS, pueda comunicas la interfaz de Arduino y el componente que se encuentra publicado en AWS, además se crearon los métodos POST y GET, para mejorar la comunicación desde Arduino para ser enviado desde los sensores configurados. Elaboración Propia.

27)Publicación del servicio API REST

Figura 29 Publicación del servicio API REST

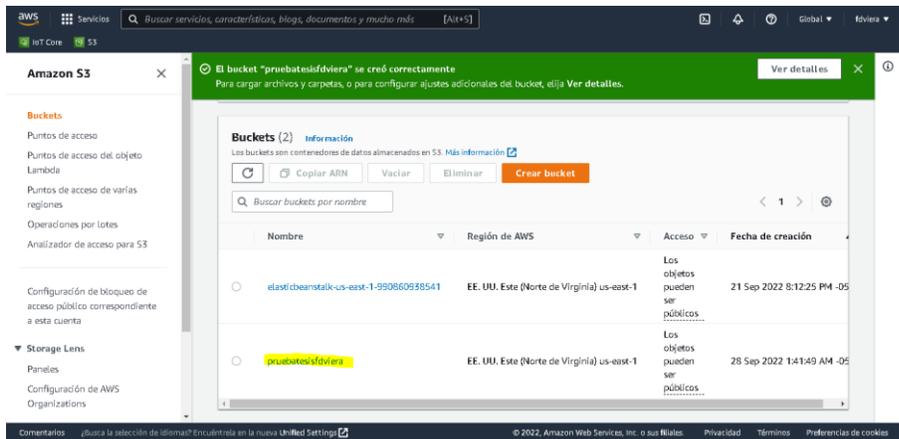
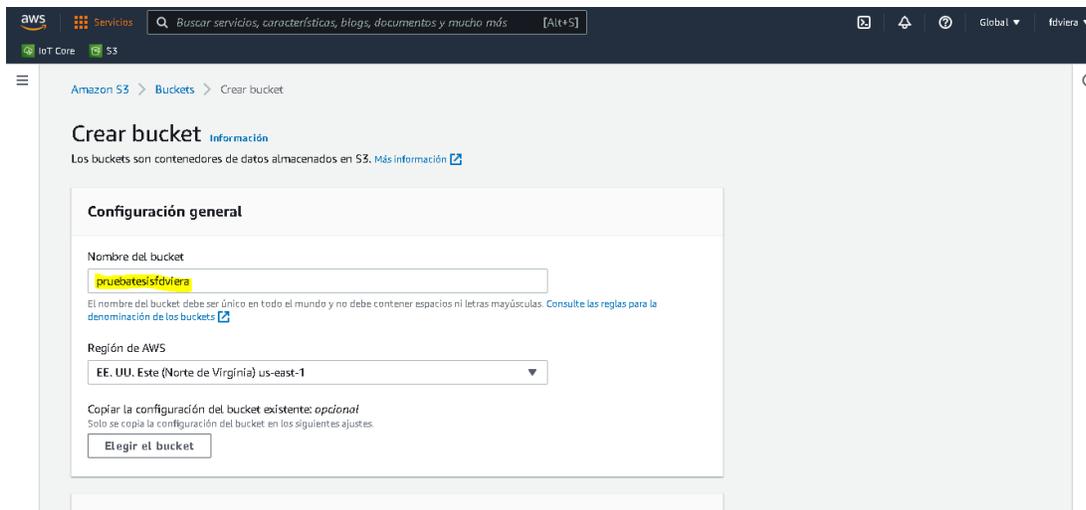
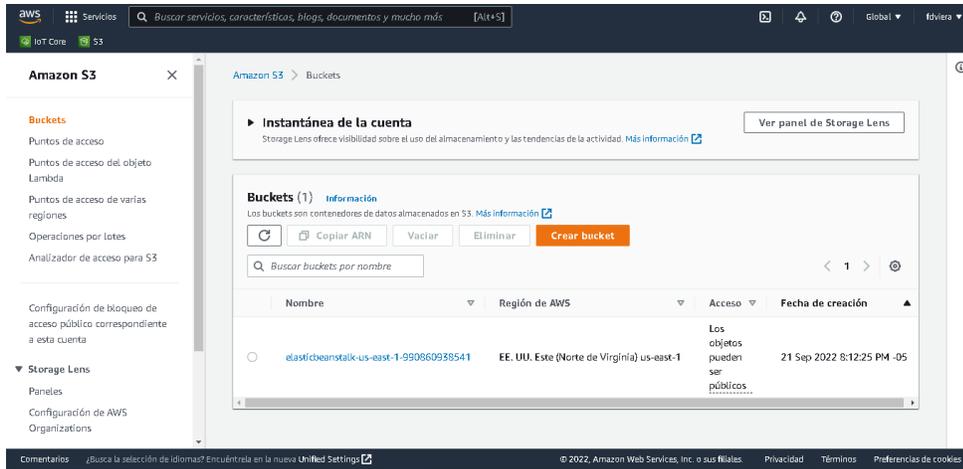




Nota: Una vez que realizan todos los pasos de implementación y configuración, se genera el siguiente link: <https://4px8puc0h4.execute-api.us-east-1.amazonaws.com/Prueba/service>, en el que se encuentra el servicio de AWS. Elaboración Propia.

28) Creación de página en la nube

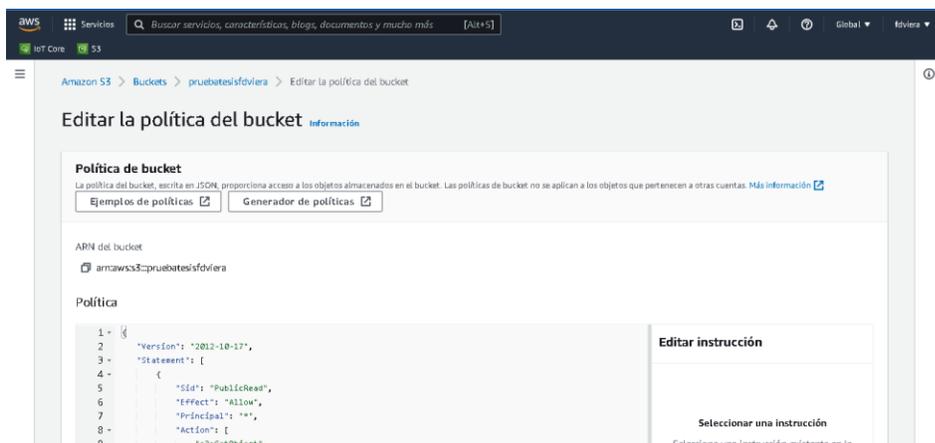
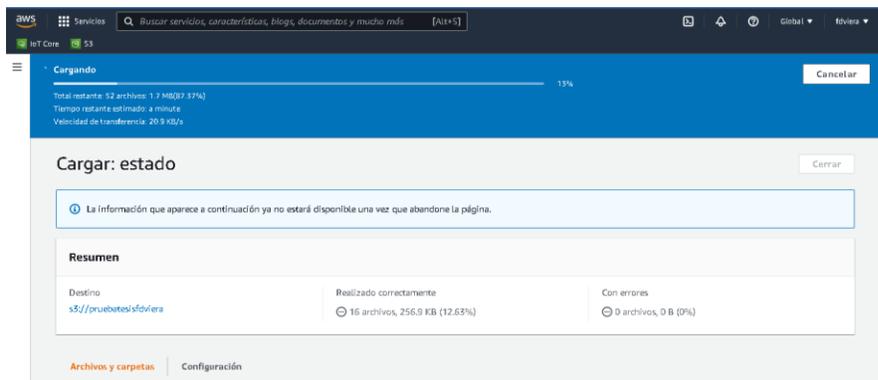
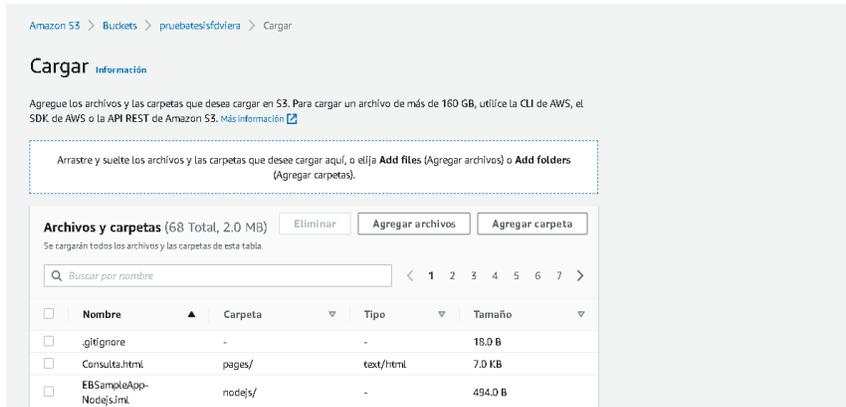
Figura 30 Crear bucket

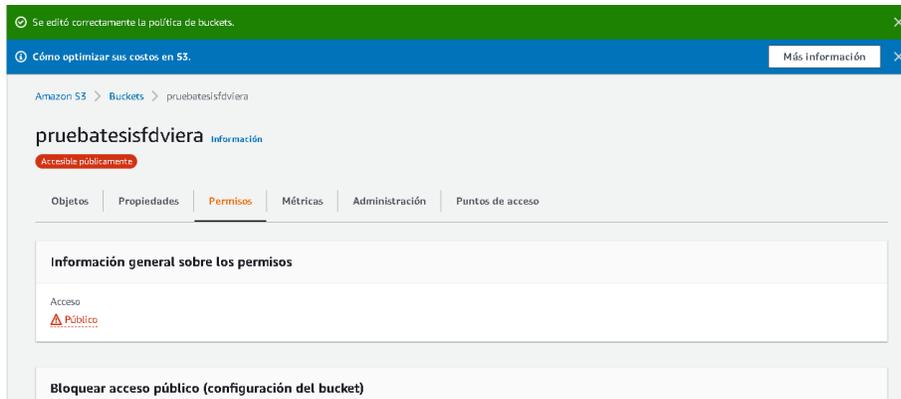


Nota: Se crea un bucket que es el espacio de almacenamiento del sitio web, luego se nombra y crea un repositorio, luego se ingresa al bucket para cargar la página.
Elaboración Propia.

29) Carga de archivos a la página web

Figura 31 Crear bucket

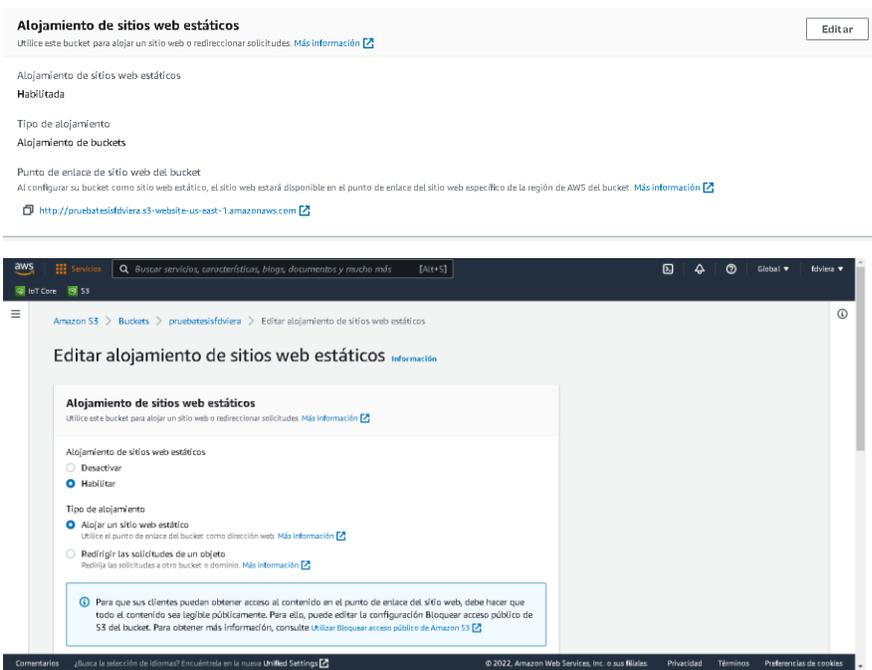




Nota: Dentro del bucket se cargan los archivos de la página, posteriormente se crean políticas y se genera el acceso. Elaboración Propia.

30) Generación de sitio web estático y dirección web publica

Figura 32 Generar sitio web



Nota: Se crea un sitio web estático, se genera la dirección web publica y se genera el link de la página web. Elaboración Propia.

En este capítulo se describió el método utilizado y los pasos a seguir para la conexión de la placa ESP32 y sensores con la nube de AWS a partir de lo cual se pudo evidenciar el funcionamiento de estos para el envío y recepción de datos a través de internet en tiempo real, lo cual permite un mejor control de información que puede afectar de una u otra manera al curso regular de las actividades.

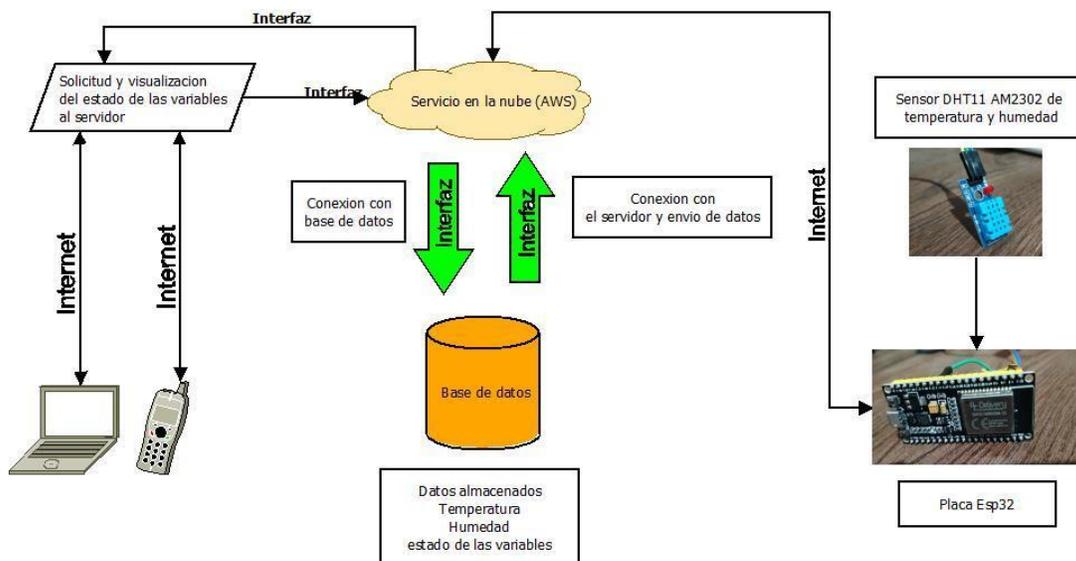
**SOLUCIÓN DE PROPUESTA TECNOLÓGICA: DISEÑO DE ARQUITECTURA
DE SERVICIOS EN LA NUBE PARA LA GESTIÓN DE VARIABLES
AGROCLIMÁTICAS.
CAPÍTULO III**

4 CAPÍTULO III: SOLUCIÓN DE PROPUESTA TECNOLÓGICA

Cuando ya se ha conectado el servidor AWS y se encuentra publicado el servicio API REST, se procede a realizar las pruebas de conexión y lecturas de las variables en tiempo real. Dicho esto, en el presente capítulo, se determina el sistema de gestión y tratamiento de los datos de variables agroclimáticas de temperatura y humedad, para la cual se crea un ambiente controlado de invernadero para simular la obtención de datos.

A continuación, se muestra el diseño propuesto.

Figura 33 *Diseño y arquitectura del monitoreo y control de temperatura y humedad.*



Nota: Elaboración propia.

4.1 Implementación de un servidor AWS para el monitoreo y control de la temperatura y humedad.

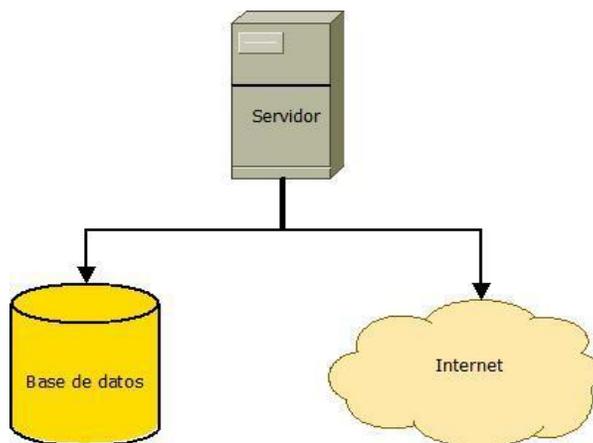
Como se ha mencionado anteriormente, la transformación digital se encuentra presente en las compañías sin importar su tamaño o su giro de negocio, especialmente con la inclusión del internet de las cosas (IoT) para el monitoreo y control de dispositivos como: alarmas, sensores, cámaras de seguridad, entre otros. Los cuales están conectados a internet y también, hace posible que puedan ser controlados y monitoreados desde cualquier parte del mundo mediante el acceso de un servidor web (Álvarez, n.d.).

Para ello, en el presente trabajo, ha sido necesario implementar un servidor para el monitoreo y control de las variables de temperatura y humedad a través del protocolo *http//*: o páginas web. En él, los usuarios podrán visualizar los valores de las variables y realizar el análisis y operaciones correspondientes según el requerimiento.

Para crear los servicios requeridos del sistema, se utilizaron diferentes servicios e instancias proporcionados por Amazon Web service (AWS) como: IoT Core AWS, API Gateway AWS, Lambda AWS, Dynamodb AWS, S3 AWS; los cuales permiten utilizar el servidor virtual de la nube de Amazon Web Service (AWS). Para esto, se creó un usuario, para brindar un nivel de seguridad suficiente al acceso y administración de recursos controlados por AWS.

Para generar la conectividad de los equipos con la nube de AWS, y para habilitar el acceso desde cualquier sitio, el servidor se encuentra conectado a internet, por tal motivo, es necesario considerar que el ancho de banda sea suficiente para garantizar el servicio.

Figura 34 Almacenamiento de datos en un servidor a través de internet

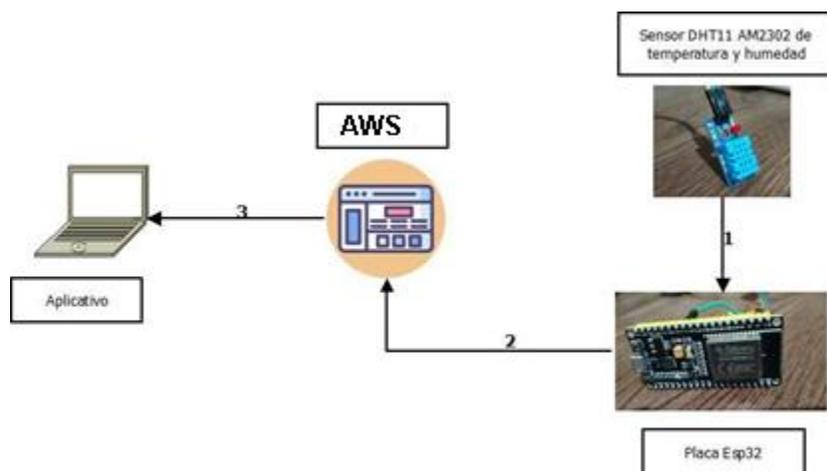


Nota: Elaboración Propia

4.2 Aplicación simulada en prototipo de invernadero

Siguiendo los requerimientos del sistema, se propone un desarrollo de tres módulos, los cuales interactúan entre ellos y se muestran en la siguiente figura:

Figura 35 Recepción de datos a través de la aplicación.



Nota: 1. Envió de señal de sensor DHT11 AM2302 a la placa Esp32; 2. Sistema interfaz entre la computadora, Placa y Amazon web service que recibe las señales para procesarlas; 3. Aplicativo de monitoreo de variables agroclimáticas. Elaboración Propia.

4.2.1 Funcionamiento de los módulos

A continuación, se detalla el funcionamiento de los módulos:

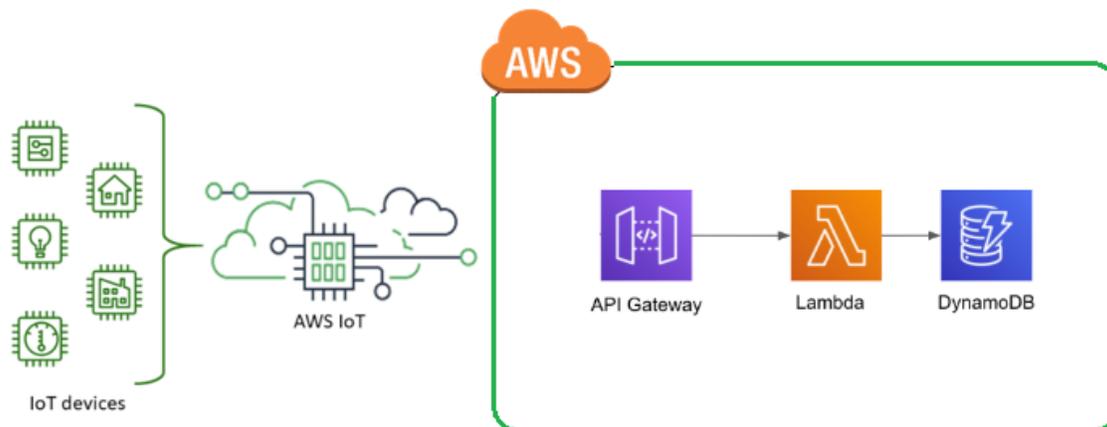
1. **Sistema de envío de señal de sensor a placa:** Los sensores perciben las variaciones de temperatura y humedad de tal manera que, a través de la placa, estas señales son enviadas a la nube de AWS para su próxima gestión.
2. **Sistema de recepción de datos desde Amazon web service:** dentro de los requerimientos, está el envío y recepción de señales, en este caso será la temperatura y la humedad. El sistema contará con la capacidad de enviar o recibir señales desde la placa hacia el servidor AWS. Adicionalmente, se requiere un control de seguridad, el cual debe permitir el acceso al sistema mediante una contraseña, así, el aplicativo debe contar con los campos correspondientes en una pantalla de Login, para que el usuario pueda ingresar los datos.
3. **Aplicativo de control y monitoreo de variables agroclimáticas:** El aplicativo presenta los datos captados en Arduino mediante el envío de señal de la placa.

4.3 Descripción de principales módulos de programación

Descripción del diagrama de soluciones, describimos la arquitectura con el siguiente gráfico.

1. Interfaces electrónicas para medir la temperaturas y humedad
2. Arduino.
Artefacto electrónico el cual se programa para poder realizar las capturas de temperatura y humedad
3. Interfaces IoT
Interface el cual procede mostrar las capturas realizadas en el Arduino el cual mostrará por medio de la consola del AWS las lecturas
4. Servicios AWS
Aquí se procedió con la configuración de las interfaces para poder presentar todas las capturas que son enviadas desde el Arduino
5. API Gateway.
Aquí se procedió con la configuración de las interfaces del API Rest para poder enviar los métodos POST y GET con el objetivo de enviar los datos desde el Arduino hacia el AWS por medio de Lambda y la base de datos
6. Lambda Aplicación el cual se configura desde el AWS de Amazon el cual integra desde el API Rest hacia la base de datos
7. DynamoDB
Base de datos No SQL el cual almacena la información que es enviada desde la interfaz de Arduino, esta información se graba en un formato json.

Figura 36 Arquitectura de solución



Nota: Elaboración Propia

Es importante mencionar, en referencia a lo detallado en puntos anteriores, que las tablas creadas en la base de dato Dynamo, no son relacionales, en la base de datos se registra únicamente las variables de temperatura y humedad.

Figura 37 Base de Datos Dynamo

La imagen muestra la interfaz de usuario de la consola de AWS DynamoDB. En la parte superior izquierda, se indica 'DynamoDB > Elementos > ESP_PRUEBA'. A la izquierda, un panel 'Tablas (3)' muestra tres tablas: 'AppSyncCommentTable-By41kT9i', 'AppSyncEventTable-By41kT9i' y 'ESP_PRUEBA', esta última seleccionada. El panel principal muestra 'ESP_PRUEBA' con un botón 'Vista previa automática' y un enlace 'Ver los detalles de la tabla'. Debajo, se indica 'Escanear o consultar elementos' y 'Completado' con 'Unidades de capacidad de lectura consumidas: 2'. Se muestran 'Elementos devueltos (104)'. Una tabla muestra los detalles de los elementos:

ESP_ID	RequestTime	Unicorn
JlyHai5v2NWBL_UZD...	2022-10-31T00:14:10.922Z	{ "messages": { "M": { "temperature": { "N": "27", "humidity": { "N": "57" } } } } }
7b67bR4P3Zi6XYKn...	2022-10-31T00:00:25.214Z	{ "messages": { "M": { "temperature": { "N": "32", "humidity": { "N": "47" } } } } }
-fst4uc4z2LajJ7Yca...	2022-10-31T00:03:21.603Z	{ "messages": { "M": { "temperature": { "N": "28", "humidity": { "N": "55" } } } } }
iHcvV44Q565dQn2M...	2022-10-31T00:15:08.797Z	{ "messages": { "M": { "temperature": { "N": "27", "humidity": { "N": "57" } } } } }
Oxkd0_F5F9M3OXh...	2022-10-31T00:02:12.069Z	{ "messages": { "M": { "temperature": { "N": "30", "humidity": { "N": "52" } } } } }
6Qn9_JUbcepOINDr...	2022-10-30T23:33:46.865Z	{ "messages": { "M": { "temperature": { "N": "33", "humidity": { "N": "43" } } } } }
QNIYK4nb BU4ce6Q...	2022-10-30T23:30:53.237Z	{ "messages": { "M": { "temperature": { "N": "33", "humidity": { "N": "41" } } } } }

Nota: Elaboración Propia

4.4 Pruebas de monitoreo y control

Para realizar las pruebas de funcionamiento del sistema, se realizó un registro de datos sin almacenamiento en base de datos para verificar el paso de información y lectura correcta, se realizaron estas pruebas en un prototipo de invernadero en el cual se instalaron los sensores para medir las variables agroclimáticas.

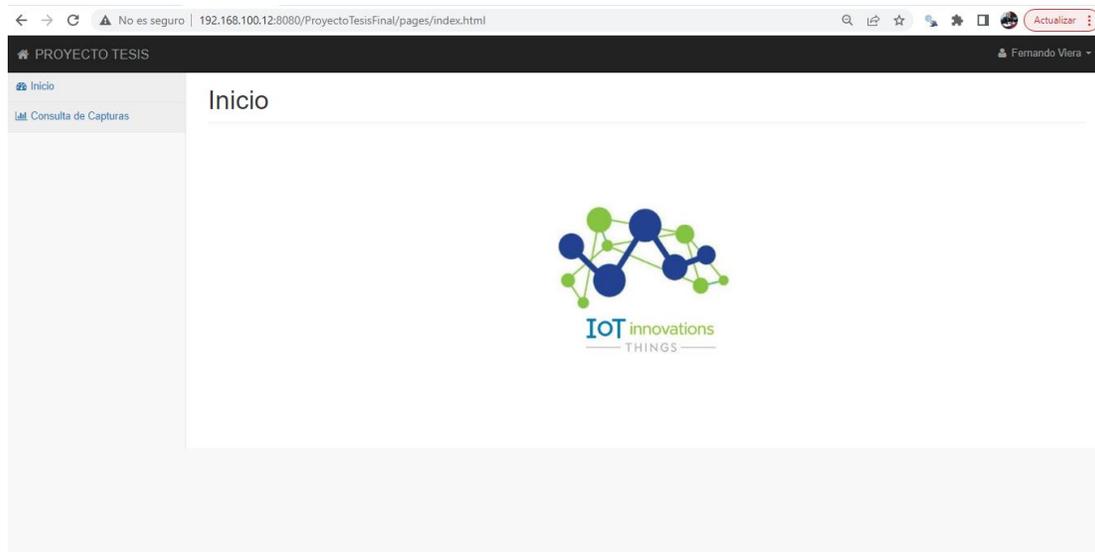
Mediante la interfaz desarrollada se pudo captar las lecturas y posterior se visualizaron en la nube de AWS en la que se comprobó la conexión adecuada a los servicios en la nube.

Posterior a las pruebas iniciales, se iniciaron las pruebas con base de datos, para lo cual, de igual manera, mediante las lecturas generadas por los sensores, se pudo captar la humedad y temperatura del prototipo de invernadero, a continuación, detalla el dashboard utilizado.

4.4.1 Dashboard de Monitoreo y control

Para el monitoreo y control, se desarrolló un dashboard web, mediante el cual se podrá visualizar las variables de temperatura y humedad.

Figura 38 *Página de Inicio*



Nota: Elaboración Propia

Dentro del sitio se procede a seleccionar la opción del menú de Consulta de Capturas el cual mostrará por medio de un gráfico la lectura de la temperatura y la humedad ingresada por fecha y hora de registro, que fueron capturadas en el proceso.

Figura 39 Capturas de variables



Nota: Elaboración Propia

Estas variables son leídas mediante los sensores conectadas a la placa ESP32 que, a su vez, envían dichas señales mediante el código de Arduino, hacia la nube de AWS y pueden reflejarse en el dashboard desarrollado.

En este capítulo se realizó el desarrollo de la propuesta tecnológica, en la que se diseñó un sistema basado en los servicios en la nube que permitió la lectura de datos de temperatura y humedad mediante sensores y placa ESP32. Además, se desarrolló panel web mediante el cual se pudo monitorear la temperatura y humedad, esto permitió tener un mayor control de variables de tipo agroclimáticas en un prototipo invernadero en tiempo real y con una disponibilidad total.

CONCLUSIONES Y RECOMENDACIONES
CAPÍTULO IV

9. Conclusiones y recomendaciones

4.5 Conclusiones

Del presente proyecto se puede concluir que la industria 4.0 tiene un gran aporte para el sector agricultor mediante el uso de dispositivos de internet de las cosas y servicios en la nube, es posible administrar, monitorear y controlar las condiciones que puedes mejorar o afectar a los cultivos en tiempo real, las 24 horas del día.

Mediante un análisis realizado entre los beneficios de los diferentes servicios de la nube disponibles, se pudo confirmar que la nube de Amazon Web Server (AWS) es la más recomendable para el diseño de servicio de infraestructuras en la nube, ya que brinda mayor seguridad y adaptabilidad frente a los demás servicios. De esta manera, se logró realizar el desarrollo utilizando también los servicios de IOT Core de AwS que fueron de soporte para la conexión con el hardware y que la solución propuesta sea adecuada y efectiva.

Debido a la accesibilidad, disponibilidad e investigaciones realizadas en este documento se optó por el uso de sistemas Arduino y como hardware la placa ESP32 junto con los sensores DH11, estos dispositivos permitieron la lectura y envió de datos hacia la nube de AWS, que posteriormente se muestran y monitorean, por lo que se puede visualizar las variaciones de humedad y temperatura de un invernadero prototipo.

El uso de los servicios en la nube de AwS con sus diferentes instancias y equipos especializados para lectura de datos, brindan análisis y datos de forma eficientes, en tiempo real y altamente controlables, lo cual permitirá a grandes industrias del sector agrícola una eficiente mejora de sus procesos para la gestión de datos críticos que, de alguna manera, pueden afectar la producción.

La arquitectura diseñada propuesta, muestra el paso de la información desde los sensores DH11 hacia la placa ESP32, los cuales envían los datos a la nube y por medio de una interfaz de AWS, se envían a la base de datos, desde la nube, por medio del sitio web, es posible visualizar, monitorear las variables de temperatura y humedad que se

emiten desde el prototipo de invernadero, lo cual demuestra los beneficios de contar con dispositivos IoT y un manejo de datos altamente disponible en la agricultura. Es posible indicar que la arquitectura diseñada trabaja de manera eficiente, cumpliendo el objetivo de gestionar variables agroclimáticas por medio de los servicios en la nube.

El sistema desarrollado en este trabajo de titulación permite el monitoreo de las variables temperatura y humedad agroclimáticas mediante cualquier sistema operativo ya sea Linux Windows o Mac, o a través de cualquier sistema operativo de dispositivo móvil como Android o IOS. Es posible indicar que, el IDE Arduino que se utilizó para desarrollo el sistema que permite la lectura hacia AWS, es una herramienta muy eficaz y útil para visualizar e interpretar los resultados a través de la página web, esto permitirá que otras variables de este tipo puedan ser monitoreadas y gestionadas con gran eficiencia.

4.6 Recomendaciones

Debido a los resultados obtenidos, del presente trabajo se recomienda:

- Para el envío o recepción de información utilizando servicios en la nube se utilizó protocolo MQTT para el manejo de variables agroclimáticas ya que su respuesta es inmediata para comunicar hardware a la nube.
- Para desarrollos relacionados al área agrónoma, se usó la placa ESP32 y los sensores de lectura de humedad y temperatura DHT11 AM2302.
- Se demostró que al utilizar los equipos seleccionados se pudo monitorear en forma eficiente el valor de las variables en tiempo real.
- Adquirir servicios en la nube, para mejorar las condiciones técnicas de procesos empresariales, industriales y agrónomos, por su flexibilidad para el manejo de sistemas y aplicativos que sean alojados en la nube.
- Fomentar el uso de dispositivos IOT para analizar condiciones críticas en procesos agrícolas en tiempo real de forma remota, lo cual servirá para ofertar productos de mejor calidad en condiciones más controladas.
- Continuar con el estudio y formular nuevas maneras de medir otras variables agroclimáticas que afectan a las tierras fértiles del Ecuador.

10. BIBLIOGRAFÍA

Banco Central del Ecuador. (2022). *Banco Central del Ecuador*. <https://www.bce.fin.ec/>

Carmanate, J. (s. f.). *Cómo programar ESP32 con IDE Arduino*. Recuperado 16 de agosto de 2022, de <https://programarfacil.com/esp8266/programar-esp32-ide-arduino/>

Chanchí-Golondrino, G.-E., Ospina-Alarcón, M.-A., & Saba, M. (2022). Sistema IoT para el monitoreo de variables climatológicas en cultivos de agricultura urbana. *Revista Científica*, 44(2), 257-271. <https://doi.org/10.14483/23448350.18470>

Col, I., Laboratorio, monitoreo, consultoría y equipo Bucaramanga-. (s. f.). *Variables Climáticas | Publicaciones*. Induanalisis, Laboratorio, monitoreo, consultoría y equipo. Bucaramanga - Col. Recuperado 18 de agosto de 2022, de https://www.induanalisis.com/publicacion/detalle/variables_climaticas_23

Cómo utilizar el DHT11 para medir la temperatura y humedad con Arduino. (s. f.). Recuperado 16 de agosto de 2022, de <https://programarfacil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/>

El Productor. (2017). *Ecuador: La tecnología vuela alto en la agricultura | Noticias Agropecuarias*. <https://elproductor.com/2017/03/ecuador-la-tecnologia-vuela-alto-en-la-agricultura-2/>

ESP32 Wifi + Bluetooth en un solo lugar. (s. f.). Recuperado 16 de agosto de 2022, de <https://programarfacil.com/esp8266/esp32/>

Flores. (2016, julio 8). Cloud Computing: MODELOS DE SERVICIO. *Cloud Computing*. <http://cloud-computing-bcs.blogspot.com/2016/07/modelos-de-servicio.html>

McCormick. (2021, noviembre 24). Agricultura 4.0: Qué es y cuáles son sus herramientas y beneficios. *McCormick*. <https://www.mccormick.it/es/agricultura-4-0-que-es-y-cuales-son-sus-herramientas-y-beneficios/>

- Méndez López, A. L. (2021). *PROTOTIPO DE SISTEMA DE MONITOREO Y CONTROL AGRÍCOLA BASADO EN IoT. CASO DE ESTUDIO: CULTIVOS SEMI HIDROPÓNICOS DE FRESA EN EL MUNICIPIO DE GUASCA CUNDINAMARCA*. 149.
- Ortiz Monet, M. (2019). *Implementación y Evaluación de Plataformas en la Nube para Servicios de IoT* [Proyecto/Trabajo fin de carrera/grado, Universitat Politècnica de València]. <https://riunet.upv.es/handle/10251/127825>
- Oyarce Valenzuela, F. I., Pérez Guzmán, R. (Profesor guía), & Fuentes Peñailillo, F. (Profesor co-guía). (2020). *Sistema integrado de gestión hídrica orientado a pequeños y medianos productores de cultivos y frutales de la región del Maule* [Thesis, Universidad de Talca (Chile). Escuela de Ingeniería Civil en Computación.]. <http://dspace.otalca.cl/handle/1950/12735>
- Pinilla, M. (2019, junio 25). *Test «ponte en mi lugar»: 10 preguntas para saber si eres una persona empática o no*. Hass. https://www.yasss.es/saber/test-ponte-lugar-preguntas-saber-persona-empatia_18_2769645212.html
- Romero, H. (2021, marzo 23). *Ecuador presenta actividades tecnológicas que implementa para desarrollar el agro – Ministerio de Agricultura y Ganadería*. <https://www.agricultura.gob.ec/ecuador-presenta-actividades-tecnologicas-que-implementa-para-desarrollar-el-agro/>
- Rubio, A. (2019, mayo 2). *Diferencias entre instrumentación digital e instrumentación analógica. Paneles digitales y analizadores de red*. <https://www.instrumentaciondigital.es/diferencias-entre-instrumentacion-digital-y-instrumentacion-analogica/>
- Saibene, B. (2020). *Cosechando Datos. Desarrollos para la agricultura en la era digital*. 32.
- Sánchez Prado, S. (2021). *Cloud Computing: Fundamentos y Despliegue de un Servicio en la Nube*. 93.

Tejada, J. E. R., & Téllez, J. D. D. (2020). *SISTEMA DE MONITOREO DE VARIABLES AGROCLIMÁTICAS PARA LOS CULTIVOS DE AGUACATE HASS EN EL VALLE DEL CAUCA*. 113.

5 ANEXOS

5.1 Anexo # 1: Código fuente index.js

```
const AWS = require('aws-sdk');
const ddb = new AWS.DynamoDB.DocumentClient();
const randomBytes = require('crypto').randomBytes;
let dataStore;
exports.handler = async (event, context, callback) => {
  const rideld = toUrlString(randomBytes(16));
  if (event.context['http-method'] === 'DELETE')
  {
    dataStore = undefined;
  }
  else if (event.context['http-method'] === 'GET')
  {
    let params = {
      TableName: 'ESP_PRUEBA',
      Limit: 30,
      ScanIndexForward : false
    };
    await ddb.scan(params, function(err, data){
      if(err){
        callback(err, null);
      }else{
```

```

        callback(null, data);
    }
}).promise();
}
else if (event.context['http-method'] === 'POST')
{

dataStore = event['body-json'];
    await recordEsp(rideld).then(() => {
        callback(null, {
            statusCode: 201,
            body: JSON.stringify({
                ESP_ID: rideld,
                Unicorn: dataStore,
                UnicornName: 'captura',
                //Eta: '30 seconds',
                // Rider: 'prueba',
            }),
            headers: {
                'Access-Control-Allow-Origin': '*',
            },
        });
    }).catch((err) => {
        console.error(err);

        errorResponse(err.message, context.awsRequestId, callback)
    });
}
else if (event.context['http-method'] === 'PUT')

```

```

    {
      this.dataStore = event['body-json'];
    }
  };

function recordEsp(rideId, username, unicorn) {
  return ddb.put({
    TableName: 'ESP_PRUEBA',
    Item: {
      ESP_ID: rideId,
      Unicorn: dataStore,
      UnicornName: 'captura',
      // Eta: '30 seconds',
      //Rider: 'prueba',
      RequestTime: new Date().toISOString(),
    },
  }).promise();
}

function toUrlString(buffer) {
  return buffer.toString('base64')
    .replace(/\+/g, '-')
    .replace(/\//g, '_')
    .replace(//=g, "");
}

function errorResponse(errorMessage, awsRequestId, callback) {
  callback(null, {
    statusCode: 500,
    body: JSON.stringify({
      Error: errorMessage,

```

```

    Reference: awsRequestId,
  }},
  headers: {
    'Access-Control-Allow-Origin': '*',
  },
});
}

```

Anexo # 2: Código Arduino en Esp32

```

//#include "secrets.h"
#include <WiFiClientSecure.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>           //Librerias
#include "WiFi.h"
#include <SimpleDHT.h>
#include <HTTPClient.h>
#include <Time.h>

#define AWS_IOT_PUBLISH_TOPIC "esp32/pub"
#define AWS_IOT_SUBSCRIBE_TOPIC "esp32/sub"

////////////////////////////////////

#define SECRET

#define THINGNAME "ESP32_DHT11_2"           //Nombre de Objeto

int pinDHT11 = 14;
SimpleDHT11 dht11(pinDHT11);

```

```

float h;

float t;

const char WIFI_SSID[] = "Gengar";          //WIFI

const char WIFI_PASSWORD[] = "CasaViera0811";    //CLAVE wifi

const char AWS_IOT_ENDPOINT[] = "ajjqwn71x5ci5-ats.iot.us-east-1.amazonaws.com";    // URL IOT
CORE

const char* serverName = "https://4px8puc0h4.execute-api.us-east-
1.amazonaws.com/Prueba/service"; //API URL

// Amazon Root CA 1 ok Certificado

static const char AWS_CERT_CA[] PROGMEM = R"EOF(
-----BEGIN CERTIFICATE-----
MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQsF
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBbWF6
b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFowOTEL
MAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXNjZm9udjEzMBcGA1UEAxMQW1hem9uFjV
b3QgQ0EgMTCCASlwdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeNXj
ca9HgFB0fW7Y14h29Jlo91ghYPI0hAEvrAlthtOgQ3pOsqTQNroBvo3bSMgHFzZM
9O6II8c+6zf1tRn4SWiw3te5djgdYZ6k/ol2peVKVuRF4fn9tBb6dNqcmzU5L/qw
IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMJbulHheb4mjUcAwhmahRWa6
VOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kDi1L
93FcXmn/6pUCyziKrlA4b9v7LWIbxcceVOF34GfID5yHI9Y/QCB/IIDeGw+OyQm
jgSubJrlqg0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AYYwHQYDVR0OBBYEFIQYzIU07LwMIJQuCFmcx7IQTgoIMA0GCSqGSIb3DQEBCwUA
A4IBAQC8Y8jdaQZChGsV2USggNiMOruYou6r4IK5IpDB/G/wkjUu0yKGX9rbxenDI
U5PMCCjmcXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhbl1Bjtt/mstv0tadQ1wUs
N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKWIIbYK8U90vv
o/ufQJVtMVT8QtPHRh8jrdkPSHca2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6XU

```

5Msl+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy
rqXRfboQnoZsG4q5WTP468SQvvG5

-----END CERTIFICATE-----

)EOF";

// Dispositivo Certificado ok

//Certificado

static const char AWS_CERT_CERT[] PROGMEM = R"KEY{

-----BEGIN CERTIFICATE-----

MIIDWjCCAKKgAwIBAgIVAKfP6se1NFoYv+rbTbLS3KSIABmtMA0GCSqGSIb3DQEB
CwUAME0xSzBJBgNVBAsMQkFtYXpvbiBxZWlglU2VydmljZXMgTz1BbWF6b24uY29t
IEluYy4gTD1TZWF0dGxllFNUPVdhc2hpbmd0b24gQz1VUzAeFw0yMjA4MjkwMzQ1
MzZaFw00OTEyMzEyMzU5NTlaMB4xHDAaBgNVBAMME0FXUyBJb1QgQ2VydGlmaWNh
dGUwggEiMA0GCSqGSIb3DQEBQUAA4IBDwAwggEKAoIBAQQDDRIsyGPO8UO4QPPy6
FyChtSXe5LsHdckw0/bb8yqomkfl8aQGuzfl3WA0W2gbpM7cPIrAGqzOoe5zE2O3
o+WqQnpGyzDmOJBn6CluyGGv2oORoyCxCIV4aox5FifsOgh1DL92Z8y6o4LBIIYZ
BN9vCWHXVCw8zAlmR/x2fbFp7W1sx5JJos5fk21wAzzhTk0aikpiw+mpJBL2NTO7
gLuA56UWpzcqOEPHuaaABJbAJsghgBEUQM0o2u7yXOc0OvWGHhWaeKZr0w4KCw7H
vDalvc4CvH9tMGYAQIs4jJ2JC1ITov9RwmayGzdWLSXW4QnjZyY6R/XN82V+fhZ
4sLXAgMBAAGjYDBeMB8GA1UdIwQYMBaAFcMDTRsmy3gkfjJrNR09hwdCv4RxMB0G
A1UdDgQWBBRytBQe6uNSX31iNn375YOISsVK4zAMBgNVHRMBAf8EAJAAMA4GA1Ud
DwEB/wQEAwIHgDANBgkqhkiG9w0BAQsFAAOCAQEAhbnL5NTInluRuJF2IfGrAyBH
vNnJ2Y0/+y7uMnqhBLMsIJd6YJ5sTFvscz1i5o1biKuxKv2f4hFLQKkx8pbr3KW
Hz+DiVO1FoxcKDUxx30a049JCzNdfzsjQnWA3s9tHMJBFXwtdVPxuh53vljcZEIz
CPwRI14xh/hIKKIU84YmWadgDmRkhMhORWaQv+6AeD7skp0r6W8chc8b0XsxPtou
weL7gkaaTYLWBSg2TvcHcHF2TGCRO5O6Ydwx9hFd4etPy75pNw5v9kXrnwmZ2XPQ
32AXNG68b0XMqV7fT2Mt8itngzz3n2NPmBp3nsf4CAAaQWpT9y4IEgA+vZhhHA==

-----END CERTIFICATE-----

)KEY";

// Device Private Key

//Certificado 3

static const char AWS_CERT_PRIVATE[] PROGMEM = R"KEY(

-----BEGIN RSA PRIVATE KEY-----

MIIEpAIBAAKCAQEAwOSLMhjzVFDuEDz8uhcgobUI3uS7B3XJMNP22/MqqJpHy/Gk
Brs3yN1gNFtoG6TO3DyKwBqsZqHucxNjt6PlqkJ6Rssw5jiQZ+giLshhr9qDkaMg
sQpVeGqMeRSH7DoldQy/dmfMuqOCwZSGGQTfbwIh11QsPMwCJkf8dn2xae1tbMeS
SaLOX5NtcAM84U5NGopKYsPpqSQS9jUzu4C7gOelFqc3KjhdX7mmgASWwCbIIYAR
FEDNKNru8lznNdr1hh4Vmnima9MOCgsOx7w2pb3OArx/bTBmAECLOlyWdiQtZUzr
/UcJmshmXVi0l1uEJ42cmOkf1zfNlfn4WeLC1wIDAQABAoIBACr8PNQzb/xlqHKE
qnHgw3s1oHeXzKBQypBoKPC0cBCbnfCEakJdFmOucA42jQ2A01TiMv0IliN+vDoy
64Gyn81CQx6wfwjw/+XIWROgdjnui09HMola22qfedLv2zBExmUkDHGbZZNU1R4SL
yJVbYEBWRWacjfx4QITi1e3Tr6mMi1kZwcfYr/33ZyTZQeE7AvAQYvcAZIEC+ZN
bQXrzHKQn6GJsIStOUIYCHLrUMqzxFDD0J2gwBicj38IDcmqq+YPyUzu1kQET7LP
Tiw5yJ5AuPuynNeSty52i8KoyFkdAiJaKXBlsfUZZV+voJ6x0aJwL644ebH3ulcA
+kX9LjkCgYEA37KLZjwq0BO07uFK4/qdv3ZEWcjbWjwSbqP6lea1zXRvY4+KMux
n4hIS1tHF+8JPdLnqnfBaH4pHqhWww5DCNnYTa0J58+sKOVlc5bBqPp/AD7CfNqp
5gM9NFSQellQGSEw0bVSKklN+nqpS06NSTKK/BHr4nnsuLcXtoY0WGsCgYEA33cl
5H6fWzxeA4t0e0YVwgd+wKLAze9kD+rusyLfJu5fS7FpgIf66LfbnhdichbgPbG
gxVZVg5+66QDrYyKkqRdNwxuCAD71YkzDo6Cw1PiERk17HWd7VWCfhovRGBySOhZ
AWk+Kyl1RGJ/tWcR5uoeSeQyxsCVi+NPNjKoSkUCgYEA3wgnEHGgxE//oRFpPQ0
37ZNA6/NdDQ8tG9t4pM6ZHRZGH5n30jODdNyGyYQxf4gGrHzJPVtsn9HBDgW+Td5
ApQgo6GJJMTcA5jFtYSICXdBiOMyetCRgCov0opGrN9tvBuukeseljJrDvD+/
gmGQTS6ZHx8tNCXo0KUR/60CgYBPuXh1Cn3HBUba00a9+8L9fWuhNiT69V+vxtW1
AUv1pehuFiw9VAfdVF2GlgOeRWE2BmOxZIKx6bNkMnSbxqfqi5wvBjS/oZm5q0Z
Hb2NWNmG+EwMBDRra09HNERTIWIwOApGvdubtrvJ6q0q18ujDLhzwm3D39k2Lxqz
mdvjgQKBgQDECVSQ/2Nx1jVhTBn0ldOIRcVOyLN1/kSVLIFJ99wM9BGpt3r9e//k
boRjSZno9cZ5ga9MkXJo6Y9UFLls+YjuOM1aCmjUg2a0VkkIObONp+jvlkcXHkfk

I9xwORTdHDol9X8m1VgQ3tlwOIEcV9E0WvYS/Q/89tXO5jb7cy5rbw==

-----END RSA PRIVATE KEY-----

)KEY";

////////////////////////////////////

WiFiClientSecure net = WiFiClientSecure();

PubSubClient client(net);

void connectAWS()

{

WiFi.mode(WIFI_STA);

WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

Serial.println("Connecting to Wi-Fi");

while (WiFi.status() != WL_CONNECTED)

{

delay(500);

Serial.print(".");

}

// Configuracion de credenciales

net.setCACert(AWS_CERT_CA);

net.setCertificate(AWS_CERT_CERT);

net.setPrivateKey(AWS_CERT_PRIVATE);

```
// Conecta a MQTT y AWS endpoint
client.setServer(AWS_IOT_ENDPOINT, 8883);

// Create a message handler
client.setCallback(messageHandler);

Serial.println("Connecting to AWS IOT");

while (!client.connect(THINGNAME))
{
  Serial.print(".");
  delay(100);
}

if (!client.connected())
{
  Serial.println("AWS IoT Timeout!");
  return;
}

// Subscribe to a topic
client.subscribe(AWS_IOT_SUBSCRIBE_TOPIC);

Serial.println("AWS IoT Connected!");
}

void publishMessage()
{
```

```

StaticJsonDocument<200> doc;
doc["humidity"] = h;
doc["temperature"] = t;
char jsonBuffer[512];
serializeJson(doc, jsonBuffer); // impresion

client.publish(AWS_IOT_PUBLISH_TOPIC, jsonBuffer);
}

void messageHandler(char* topic, byte* payload, unsigned int length)
{
  Serial.print("incoming: ");
  Serial.println(topic);

  StaticJsonDocument<200> doc;
  deserializeJson(doc, payload);
  const char* message = doc["message"];
  Serial.println(message);
}

void setup()
{
  Serial.begin(115200);
  connectAWS();
}

void envioDatosAPI() {

```

```

StaticJsonDocument<200> doc;

char jsonBuffer[512];

//doc["fecha"] = timerDelay;

JsonObject obj = doc.createNestedObject("messages");
obj["humidity"] = h;
obj["temperature"] = t;
serializeJson(doc, jsonBuffer);

if(WiFi.status()== WL_CONNECTED){
//String mensajeJson = "{\"Humidity\":\"+h+\", \"Temperature\":\"+ t + \"}";

  HTTPClient http;

  http.begin(serverName);
  http.addHeader("Content-Type", "application/json");
  int httpResponseCode = http.POST(jsonBuffer);

  if (httpResponseCode > 0) { //chequeo

    String payload = http.getString();
    Serial.println(httpResponseCode);
    Serial.println(payload);
  }

  else {
    Serial.println("Error on HTTP request");
  }

  http.end();

```

```

}
else {
    Serial.println("WiFi Disconnected");
}
//lastTime = millis();
//}
}

void loop()
{
    byte temperature = 0;
    byte humidity = 0;
    int err = SimpleDHTErrSuccess;
    if ((err = dht11.read(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {
        Serial.print("Read DHT11 failed, err="); Serial.print(SimpleDHTErrCode(err));
        Serial.print(","); Serial.println(SimpleDHTErrDuration(err)); delay(1000);
        return;
    }
    Serial.print("Sample OK: ");
    Serial.print((int)temperature); Serial.print(" *C, ");
    Serial.print((int)humidity); Serial.println(" H");

    t = temperature;
    h = humidity;

    if (isnan(h) || isnan(t) ) // chequeo de falla
    {
        Serial.println(F("Error sensor "));
    }
}

```

```
    return;
}

Serial.print(F("Humidity: "));
Serial.print(h);
Serial.print(F("% Temperature: "));
Serial.print(t);
Serial.println(F("°C "));

publishMessage(); //publico los mensajes en la consola
envioDatosAPI(); // publico los mensajes en el API
client.loop();
delay(10000); //tiempo de lectura de temperatura y humedad
}
```