

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN



Trabajo de Titulación

Tema: Diseño de un sistema de monitoreo y control de una planta de
tratamiento de agua potable mediante el uso dispositivos IoT

AUTOR:

PAULA GISELL POVEDA SOTOMAYOR

DIRECTOR:

Henry Nelson Roa Marin, PhD

Quito, 28 de enero de 2025

DEDICATORIA

A mis queridos padres, Jorge y Shisela, quienes con su amor, sacrificio y sabiduría han sido mi guía en cada logro de mi camino universitario. A mis tres hermanos, por sus enseñanzas que siempre han favorecido mi crecimiento personal y profesional. A mi abuelita Susi, cuyos abrazos siempre sanaban mi corazón y me daban el aliento necesario para seguir adelante. Y, por último, a mi abuelito Jorge, por su amor y enseñanzas, los cuales han dejado una huella imborrable en mi vida.

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a todas las personas que han sido parte fundamental de este proceso. A mis padres, por su confianza, paciencia y por brindarme las herramientas necesarias para llegar hasta aquí. A mis hermanos, por su apoyo incondicional y por recordarme siempre la importancia de la familia. A mis abuelitos, por su amor constante y por ser mi ejemplo de vida.

RESUMEN

En Ecuador, la necesidad de abastecerse de agua potable es especialmente significativa en las zonas rurales. Según la Organización Mundial de la Salud (OMS), "Si no hay servicios de agua y saneamiento, o si estos son insuficientes o están gestionados de forma inapropiada, la población se expone a riesgos para su salud que, en realidad, se pueden prevenir" (Organización Mundial de la Salud, 2023). En la ciudad de Palora, ubicada en la región oriental del país, recientemente se construyó una planta de agua potable con tecnología civil moderna.

Esta planta se encuentra alejada de la ciudad, lo que permite evitar la contaminación del agua y reduce las molestias para los pobladores. Sin embargo, debido a su ubicación remota, el control y la gestión de la planta resultan complicados, ya que los problemas solo se detectan una vez que ya han ocurrido.

Por esta razón, el presente trabajo de titulación se enfoca en el diseño de un sistema para la monitorización y gestión automática de esta infraestructura civil. Para ello, se emplea tecnología de Internet de las Cosas (IoT, por sus siglas en inglés), que se refiere a la red de dispositivos conectados y a la tecnología que facilita la comunicación entre dichos dispositivos y la nube, así como entre los propios dispositivos (AWS, 2021). De esta forma, se integrarán sensores que funcionarán como monitores y controladores para asegurar un manejo óptimo de la planta.

Tabla de Contenidos

1. CAPÍTULO 1: INTRODUCCIÓN.....	11
1.2. Justificación.....	11
1.3. Planteamiento del problema.....	11
1.4. Objetivos.....	12
1.5. Alcance.....	12
2. CAPITULO 2: MARCO TEÓRICO Y CONCEPTUAL.....	13
2.1. Antecedentes o marco referencial.....	13
2.2. Plantas de tratamiento de agua potable.....	13
2.2.1. Agua Cruda.....	13
2.2.2. Agua Potable.....	14
2.2.3. Propiedades generales del agua.....	15
2.2.3.1. pH.....	15
2.2.3.2. Turbidez.....	15
2.2.3.3. Conductividad.....	16
2.2.4. Caudal.....	16
2.2.5. Fases de las plantas de tratamiento.....	17
2.2.5.1. Fase de coagulación.....	17
2.2.5.2. Fase de floculación.....	17
2.2.5.3. Fase de sedimentación.....	18
2.2.5.4. Fase de filtración.....	18
2.2.5.5. Fase de desinfección.....	19
2.3. Automatización IoT.....	19
2.3.1. Circuitos Electrónicos.....	20
2.3.1.1. Circuitos electrónicos digitales.....	20
2.3.1.2. Circuitos electrónicos analógicos.....	20
2.3.1.3. Circuito de lazo cerrado.....	21

2.3.2.	Sensores	21
2.3.2.1.	Sensor de turbidez	22
2.3.2.2.	Sensor de pH.....	22
2.3.2.3.	Electroválvula.....	22
2.3.2.4.	Sensor de medición de caudal	22
2.3.2.5.	Controladores.....	23
2.3.3.	Protocolos	24
2.3.3.1.	MQTT.....	24
2.3.3.2.	AWS	26
2.3.3.3.	GRAFANA.....	27
3.	Capítulo 3: Metodología.....	28
3.1.	Disign Science.....	28
3.1.1.	Identificación de problemas y motivación.....	28
3.1.2.	Determinar objetivos de la solución	29
3.1.3.	Diseño y desarrollo	29
3.1.4.	Demostración	29
3.1.5.	Evaluación.....	30
3.1.6.	Comunicación	30
4.	Capítulo 4: Diseño	31
4.1.	Diagnóstico de la planta de tratamiento de agua potable	31
4.1.2.	Identificación de problemas y motivación.....	34
4.1.3.	Objetivos del proyecto	34
4.1.4.	Diseño y desarrollo	34
4.1.4.1.	Requerimientos del usuario	34
4.1.4.2.	Diagrama de actividades.....	38
4.1.4.3.	Hardware y equipo correspondiente.....	39
4.1.4.4.	Esquema de conexión	42

4.1.4.5.	Programa de simulación con Python	43
4.1.4.6.	Conexión hacia el suscriptor AWS.....	46
4.1.4.7.	Conexión de AWS IoT CORE a TimeStream	50
4.1.4.8.	Conexión de Timestream a Grafana	53
4.1.4.9.	Demostración.....	58
4.1.5.	Evaluación.....	60
5.	Conclusiones y Recomendaciones.....	62
5.2.	Recomendaciones.....	62
6.	BIBLIOGRAFÍA.....	64
7.	Anexos	67
7.1.	Programa en Arduino ID para sensores.....	67

ÍNDICE DE TABLAS

Tabla 1. Requisito funcional 1 – Realizado por el autor de la tesis.....	35
Tabla 2. Requisito funcional 2 – Realizado por el autor de la tesis.....	35
Tabla 3. Requisito funcional 3 – Realizado por el autor de la tesis.....	35
Tabla 4. Requisito funcional 4 – Realizando por el autor de la tesis.....	35
Tabla 5. Requisito funcional 5 – Realizado por el autor de la tesis.....	36
Tabla 6. Requisito no funcional 1 – Realizado por el autor de la tesis.....	36
Tabla 7. Requisito no funcional 2 – Realizado por el autor de la tesis.....	36
Tabla 8. Requisito no funcional 3 – Realizado por el autor de la tesis.....	36
Tabla 9. Requisito no funcional 4 – Realizado por el autor de la tesis.....	37
Tabla 10. Requisito no funcional 5 – Realizado por el autor de la tesis.....	37
Tabla 11 Sensor de caudal – (DN100, 2023).....	39
Tabla 12. Sensor de turbidez – (Sarabati, 2023).....	39
Tabla 13. Sensor de PH.....	40
Tabla 14 Electroválvula industrial.....	40
Tabla 15. Microcontrolador ESP32 - (Orellana, 2023).....	41
Tabla 16. RELÉ – (LED BOX, 2025).....	41
Tabla 17. Servomotor – (Robodacta, 2025).....	41

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Esquema de conexión IoT.....	42
Ilustración 2. Consola de AWS – De autoría propia.....	46
Ilustración 3. Creación de objetos - De autoría propia	47
Ilustración 4. Certificados de seguridad - De autoría propia	47
Ilustración 5. Permisos de la política Controlador_AguaP – De autoría propia.....	48
Ilustración 6. Política creada para el objeto – De autoría propia	48
Ilustración 7. Descarga de certificados – De autoría propia	49
Ilustración 8. Claves y certificados ingresados en el programa – De autoría propia.....	49
Ilustración 9. Envío de datos por medio del protocolo MQTT – De autoría propia.....	50
Ilustración 10. Creación de base de datos - De autoría propia.....	50
Ilustración 11. Creación de tablas de la base de datos – De autoría propia.....	51
Ilustración 12. Retención de los datos históricos – De autoría propia.....	51
Ilustración 13. código SQL para captar mensajes del suscriptor – De autoría propia	52
Ilustración 14. Regla para la base de datos datos_planta – De autoría propia.....	52
Ilustración 15. Creación de dimensiones de BD – De autoría propia.....	52
Ilustración 16. usuario para la conexión de IoT core con TimeStream – De autoría propia ...	53
Ilustración 17. Datos guardados en la base de datos – De autoría propia.....	53
Ilustración 18. Creación de usuario para la conexión con programas externos – De autoría propia	53
Ilustración 19. Clave de acceso y clave secreta para conexiones – De autoría propia	54
Ilustración 20. Conexión con claves de TimeStream a Grafana – De autoría propia.....	54
Ilustración 21. Datos de la base de datos a conectar – Capturado por el autor.....	55
Ilustración 22. Diseño de la visualización del caudal – De autoría propia	55
Ilustración 23. Construcción del dashboard para turbidez – De autoría propia.....	56
Ilustración 24. Construcción de dashboard para válvulas – De autoría propia.....	56
Ilustración 25. Construcción de dashboard general – De autoría propia	56
Ilustración 26. Creación de dashboard de 3 valores – De autoría propia	57
Ilustración 27. Creación del dashboard tiempo valor – De autoría propia	57
Ilustración 28. Conexión de datos entre AWS IoT y TimeStream – De autoría propia	58
Ilustración 29. Conexión total entre servicios – De autoría propia.....	59
Ilustración 30. Dashboard con todas las pestañas de monitoreo y control – De autoría propia	59

Ilustración 31. Grafica tiempo valor – De autoría propia60

Ilustración 32. Tabla general de datos, últimos valores capturados – De autoría propia60

ÍNDICE DE IMÁGENES

Imagen 1. Escala de pH	15
Imagen 2. Turbiedad del agua.....	16
Imagen 3. Fase de coagulación	17
Imagen 4. Fase de floculación	18
Imagen 5. Fase de sedimentación	18
Imagen 6. Fase de filtración.....	19
Imagen 7. Fase de desinfección	19
Imagen 8. Grafica de la señal digital	20
Imagen 9. Grafica de señal analógica	21
Imagen 10. Sistema de lazo cerrado	21
Imagen 11. Microcontrolador	23
Imagen 12. Arquitectura de MQTT	25
Imagen 13. Conexiones con protocolo.....	25
Imagen 14. Comunicación de datos entre servicios.....	27
Imagen 15. Dising Science	28
Imagen 16. Planta de tratamiento Plano – (Jorge Poveda A, 2020)	33
Imagen 17. Diagrama de actividades	38
Imagen 18. Código de simulación de sensores en Python – De autoría propia.....	45

1. CAPÍTULO 1: INTRODUCCIÓN

1.2. Justificación

En los últimos años el acceso al agua se ha vuelto un derecho, por lo que los países buscan diversas maneras de potabilizar agua. La alternativa que más se usa en el Ecuador consiste en captar el agua en ríos alejados de las ciudades, que cumplan con ciertos parámetros de potabilización. Sin embargo, cuando ocurre un daño o alteración en la construcción y manejo, en las captaciones y plantas de tratamiento es muy complicado saber el momento y lugar del incidente. Por esta razón, se pretende diseñar un sistema que permita la automatización y control de los puntos críticos de una planta de tratamientos de agua mediante manejos centralizados de los recursos, con sensores y dispositivos IoT.

1.3. Planteamiento del problema

Al gestionar estructuras civiles, es complicado identificar los problemas a largo plazo que pueden afectar su integridad y funcionalidad. Este desafío se intensifica en el caso de obras esenciales como las plantas de tratamiento de agua potable, que no solo dependen de la infraestructura misma, sino también de factores externos como las condiciones climáticas y las políticas públicas del país.

Por ejemplo, en ciudades como Palora, el suministro de agua proviene de una captación que solo es accesible a pie, tras una caminata de cinco horas, por lo que la planta de tratamiento del agua también se encuentra alejada de la ciudad. Esto dificulta directamente la supervisión y el mantenimiento regular de la infraestructura. Si el caudal del río disminuye, la alerta sobre la falta de agua no se activará hasta que se presente un desabastecimiento evidente, lo que pone en riesgo la disponibilidad de agua potable para la comunidad. Esta situación muestra la necesidad urgente de implementar sistemas de monitoreo más eficaces que permitan detectar problemas de manera anticipada, asegurando así un suministro continuo y seguro de agua potable.

1.4. Objetivos

General:

Diseñar un sistema de gestión y control mediante dispositivos IoT para el monitoreo y automatización de plantas de tratamiento de agua potable.

Específicos:

- Proponer la integración de dispositivos IoT permitiendo la comunicación y transmisión de datos entre sensores.
- Optimizar el monitoreo, la detección de incidentes y la gestión de recursos hídricos en tiempos correctos para la acción ante incidentes.

1.5. Alcance

Este estudio se centrará en el diseño de un sistema de automatización para plantas de tratamiento de agua potable en Ecuador. Como caso de estudio, se tomará una planta de tratamiento ubicada en la ciudad de Palora, donde la infraestructura tecnológica actual es muy limitada y toda la monitorización se realiza mediante personal humano. La investigación abordará los desafíos existentes en la gestión de los recursos hídricos y propondrá soluciones basadas en (IoT) que mejoren la monitorización del recurso con un control automatizado. En una expansión de este proyecto, se podrá realizar un análisis de datos utilizando la información obtenida de la monitorización.

2. CAPITULO 2: MARCO TEÓRICO Y CONCEPTUAL

2.1. Antecedentes o marco referencial

En Ecuador, el acceso al agua potable es un desafío constante, especialmente en áreas rurales y semiurbanas. Las plantas de tratamiento de agua son fundamentales para garantizar la calidad del suministro, pero muchas de ellas enfrentan problemas relacionados con control y la gestión. Las provincias de Guayas y Pichincha, donde se concentran las principales ciudades, cuentan con un mayor desarrollo en términos de tecnología de monitoreo, como el sistema SCADA en la EPMAP (Rosero, 2015). Sin embargo, muchas plantas en regiones más aisladas carecen de sistemas automatizados que permitan un control eficaz.

El uso de tecnologías como el Internet de las Cosas (IoT) ha comenzado a expandirse en el sector del agua, ofreciendo la posibilidad de implementar soluciones que faciliten el monitoreo en tiempo real y la gestión de recursos hídricos (Campos, 2020). Sin embargo, su adopción en Ecuador es aún lejana, lo que limita la capacidad de respuesta ante problemas en las plantas de tratamiento. La implementación de sistemas automatizados podría mejorar significativamente la eficiencia operativa permitiendo una correcta y temprana detección de anomalías.

2.2. Plantas de tratamiento de agua potable

Las plantas de tratamiento de agua potable son obras civiles que logran pasar por un proceso de purificación al agua cruda que se capta de algún lado. Según Miguel Araque:

Las plantas de tratamiento reciben materia prima como el agua cruda con características físico químicas variables y entregan agua potable apta para el consumo humano que cumpla todos los estándares técnicos de calidad. Además, las plantas de tratamiento de agua potable cumplen un rol fundamental en la salud del ser humano, ya que un líquido vital de alta calidad previene el apareamiento de las enfermedades transmitidas por el agua, muchas de las cuales son mortales (Araque, 2022).

En el Ecuador la mayoría de las plantas de tratamiento son para captaciones de ríos que quedan muy lejos de las ciudades habitadas, estando muy poco contaminados y permitiendo que el agua sea apta para este proceso.

2.2.1. Agua Cruda

El agua cruda es aquella que se encuentra en la naturaleza sin haberse sometido por ningún proceso ni químico ni físico de sanitización. Es decir, no tiene cambios químicos, físicos, ni

bacteriológicos (Yépez & Ruiz, 2005). Puede contener materia orgánica e inorgánica, además de microorganismos. Esto muchas veces hace que salgan a lucir algunas características físicas del agua natural como lo es el olor, color, sabor y turbidez. Si a esta se le somete a pruebas químicas y biológicas también podemos encontrar un pH no correcto y análisis positivos de la materia antes mencionada. Las fuentes naturales de las que se extra el agua puede ser:

- Agua de lluvia: El agua de lluvia no contiene aditivos ni características físicas visibles, sin embargo, esta agua también se debe potabilizar y de echo en muchos casos es más difícil ya que, su caída desde la atmosfera puede arrastrar diversos contaminantes acumulados en capas bajas.
- Agua de manantial: El agua de manantial es el agua subterránea formada por agua de volcanes, lluvias entre otras que emerge a la superficie, por poros o fisuras. Esta agua, es de las más puras ya que al subir de la superficie tiene cierto grado de filtración llegando a una depuración natural de las distintas capas freáticas y la eliminación de sustancias (Valdivielso, 2020).
- Agua Subterránea: Se mantiene almacenadas en el interior de la tierra entran por los poros de la tierra formando, muchas veces, colchones de agua. Para utilizar esta agua se necesita hacer un estudio de resistividad eléctrica el cual informa la distancia a la que se encuentra agua que puede ser potabilizada.
- Agua Superficial: Son fuentes de agua dulce, como ríos, lagos, embalses, canales, entre otros. Estas fuentes son las más regulares en el ecuador y se utilizan ríos que no están al alcance humano se encuentran en los pies de volcanes o dentro de selvas o paramos.
- Agua del Mar: Como es conocido el agua del mar abarca más del 60% de la superficie terrestre, sin embargo, esta se compone de muchas sales minerales disueltas, por lo que es difícil y muy caro potabilizarla (ALCORA, 2021).

2.2.2. Agua Potable

Se puede considerar agua potable al agua cruda que pasa por un proceso químico y físico para purificar el agua, adaptándola para el consumo humano. (Kemira Raw water and drink, 2024). El tratamiento variará en función del uso que se le quiera dar y de las propiedades que contenga el agua desde su punto de partida (ALCORA, 2021). Para saber que el agua es totalmente potable debe cumplir con estándares tanto cualitativos, como cuantitativos. Entre esos están:

- Inolora

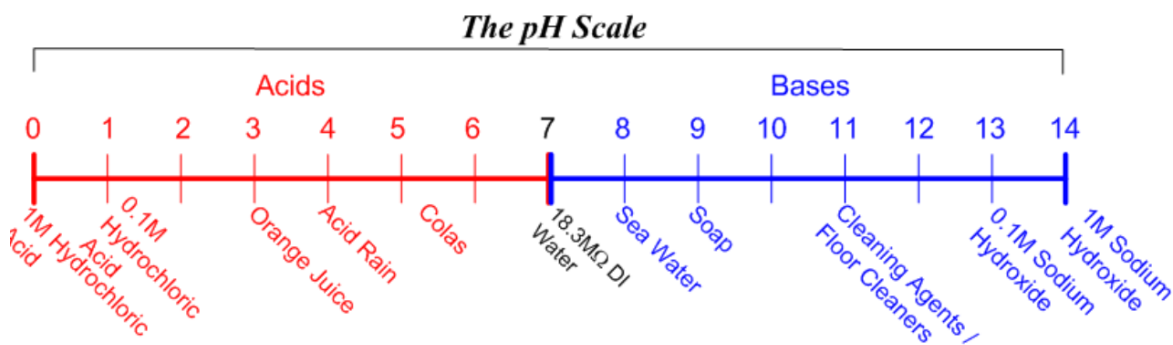
- Sin colores turbios o extraños
- Sin sabores
- Libre de riesgos microbiológicos y sustancias inorgánicas relacionadas con la salud.
- pH dentro de límites ajustados para el consumo humano (6.5 a 9.5)
- Niveles de cloro permitidos para el consumo humano

2.2.3. Propiedades generales del agua

2.2.3.1. pH

El pH es una medida que indica la acidez o la alcalinidad del agua. Se define como la concentración de iones de hidrógeno en el agua. La escala del pH es logarítmica con valores de 0 a 14. Un incremento de una unidad en la escala logarítmica equivale a una disminución diez veces mayor en la concentración de iones de hidrógeno. (California State, NF)

Imagen 1. Escala de pH



Digital Analysis Corp. (2019)

2.2.3.2. Turbidez

Se define como turbidez a la medida de la cantidad de partículas en suspensión en el agua. Las algas, sedimentos, materia organican y los contaminantes que puedan enturbiar el líquido (California State, NF)

Los niveles de turbidez del agua son:

Imagen 2. Turbiedad del agua

		Turbiedad máxima del agua			Dosis		
		Cruda	Clarificada	Tratada	Coagulante	Cloro precloración	Cloro postcloración
Turbiedad máxima del agua	Cruda	1,00	0,23**	0,55*	0,87*	0,34*	0,30*
	Clarificada	0,23**	1,00	0,38*	0,18**	⊥	⊥
	Tratada	0,55*	0,38*	1,00	0,57*	⊥	⊥
Dosis	Coagulante	0,87*	0,18**	0,57*	1,00	⊥	⊥
	Cloro precloración	0,34*	^	⊥	⊥	1,00	⊥
	Cloro postcloración	0,30*	^	⊥	⊥	⊥	1,00

* Correlación significativa al 5 %; ** Correlación no significativa al 5 %; ⊥ correlación no evaluada

(Montoya et al., 2011)

2.2.3.3. Conductividad

Según Galvín: “La conductividad es producida por los electrolitos disueltos un agua y en ella influyen: terreno drenado, composición mineralógica, tiempo de contacto, gases disueltos, pH y todo lo que afecte a la solubilidad de sales” (Galvín, 2008).

2.2.4. Caudal

Cantidad de fluido que circula a través de una sección del ducto (tubería, cañería, oleoducto, río, canal...) por unidad de tiempo (RAE, 2017).

Si denominamos al caudal Q, al volumen V y al tiempo t, entonces:

$$Q = \frac{V}{t} [1]$$

Normalmente el volumen se mide en litros y el tiempo en segundos, por tanto, el caudal vendría expresado en:

$$Q = \frac{V}{t} = \frac{\text{litros (l)}}{\text{segundo (s)}}$$

(Monge. A, 2018)

2.2.5. Fases de las plantas de tratamiento

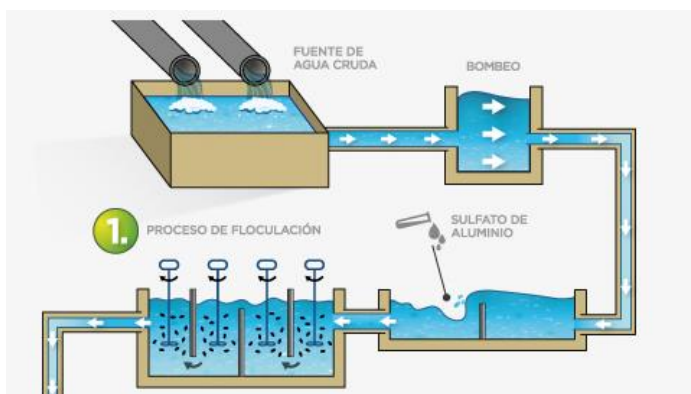
Estas construcciones tienen distintas fases por las que pasa el agua para su potabilización. Esto depende del diseño y tipo de planta de tratamiento que se piense usar (BBVA, 2024). Sin embargo, la base de la que parten las fases de cualquier diseño son las siguientes:

- Coagulación
- Floculación
- Sedimentación
- Filtración
- Desinfección
- Reserva

2.2.5.1. Fase de coagulación

Al ingresar el agua sin tratar, incluye pequeñas partículas flotantes que tienen una carga negativa. Esta carga provoca que las partículas se empujen entre sí. Como resultado de este fenómeno, no pueden agruparse y permanecen dispersas, flotando en el líquido. La coagulación es el método que utiliza químicos para aglutinar estas partículas, facilitando su eliminación. (Lorenzo, 2006)

Imagen 3. Fase de coagulación



(Instituto del Agua, 2024)

2.2.5.2. Fase de floculación

La formación de los flóculos es consecuencia de la agrupación de las partículas descargadas al ponerse en contacto unas con otras. Puede ser causada por la colisión entre las partículas, debido a que cuando se acercan lo suficiente las superficies sólidas, las fuerzas de Van der

Waals predominan sobre las fuerzas de repulsión, por la repulsión, por la reducción de la carga eléctrica que trae como consecuencia la disminución de la repulsión eléctrica. (Lorenzo, 2006)

Imagen 4. Fase de floculación



(Instituto del Agua, 2024)

2.2.5.3. Fase de sedimentación

Es el proceso mediante el cual se asientan los sólidos suspendidos en un fluido, bajo la acción de la gravedad. El proceso de sedimentación en una planta de tratamiento de agua potable convencional es en general el principal responsable en remoción de partículas causantes de turbiedad y color. El grado de remoción exigido a esta etapa, depende fuertemente de la eficiencia del proceso posterior de filtración. (Manrique & Gomez, 2016).

Imagen 5. Fase de sedimentación



(Instituto del Agua, 2024)

2.2.5.4. Fase de filtración

Una vez completada la fase de sedimentación, en la que los sólidos suspendidos se han depositado en el fondo, se inicia la última etapa de limpieza física: la filtración. En esta fase, el agua pasa a través de un filtro diseñado con capas de materiales como roca volcánica, piedras, antracita, arena, entre otros. Este sistema de filtración está diseñado para retener cualquier material sólido residual y eliminar partículas finas, incluidas aquellas no visibles al ojo humano, asegurando así una mayor pureza del agua (OSE, 2024).

Imagen 6. Fase de filtración



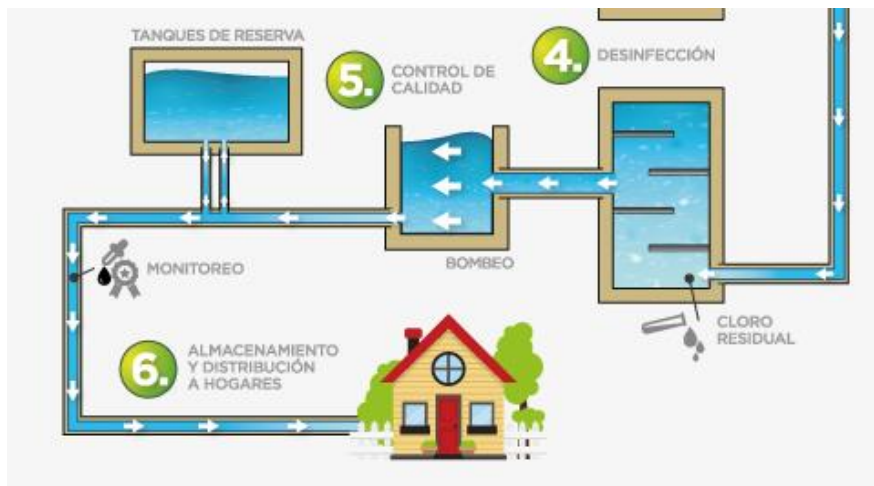
(Instituto del Agua, 2024)

2.2.5.5. Fase de desinfección

Es la última fase de potabilización del agua por lo que el tratamiento que necesita esta ya no es tan fuerte. Esta etapa se consigue mediante un agente desinfectante como el cloro.

(BBVA, 2024)

Imagen 7. Fase de desinfección



(Instituto del Agua, 2024)

2.3. Automatización IoT

Es la interconexión digital de objetos cotidianos con internet. (Garzón, 2019) Lo más importante de la tecnología IoT es la automatización que esta causa. Sin embargo, el desafío más grande que tiene IoT es el ancho de banda en las infraestructuras de red. Es así que como menciona Garzón en su libro: “En el momento en que tengamos más dispositivos, mayor será el tráfico que debemos manejar con la red actual” (Garzón, 2019)

Además, es importante tener en cuenta el problema de la seguridad que tiene esta tecnología, ya que los dispositivos IoT tienen un alto grado de vulnerabilidad.

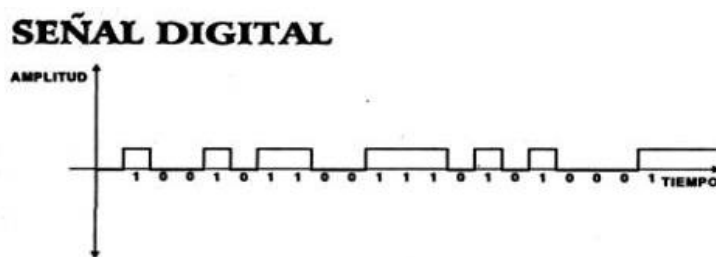
2.3.1. Circuitos Electrónicos

Los circuitos electrónicos se componen por varios objetos electrónicos conectados entre sí permitiendo circular corriente eléctrica. Para entenderlo de mejor manera, se debe saber algunos conceptos que son características de los circuitos electrónicos.

2.3.1.1. Circuitos electrónicos digitales

Los circuitos electrónicos digitales son aquellos que se manejan de manera binaria, significa que envían impulsos eléctricos que marcan 1 o 0, esto depende de lo que el dato transmite 1 nivel alto o 0 nivel bajo. Una de las ventajas de estos circuitos es que cuentan con mejor fiabilidad y precisión de los datos, esto debido a que las degradaciones de señal no influyen en el sistema digital. Además, estos circuitos tienen un soporte matemático para su construcción, están basados en el álgebra de Boole (Instituto Aprende, 2020). Además, en la electrónica digital se implementan bloques lógicos. Estos funcionan con datos lógicos de entrada para dar bloques lógicos de salida. En la construcción de estos circuitos los bloques son circuitos integrados que contienen en su interior transistores integrados dependiendo de la complejidad de los circuitos. (Hernández, 2014)

Imagen 8. Grafica de la señal digital

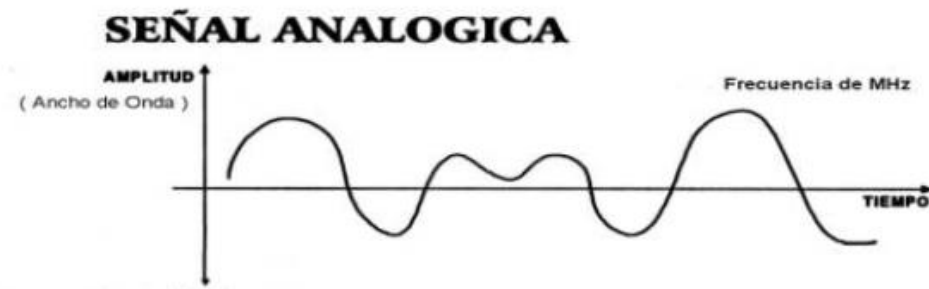


(Hernández, 2014)

2.3.1.2. Circuitos electrónicos analógicos

Las señales eléctricas son corrientes y tensiones que varían continuamente en el transcurso del tiempo. Estos valores pueden cambiar según la señal de entrada y salida que manda el sensor. Los valores no siempre son exactos y su margen de error es alto, esto se debe a interferencias causadas con frecuencias de radios o televisiones viejas (Hernández, 2014). Además, si hay alguna variación no controlada en la corriente o tensión los valores cambian teniendo algunos errores.

Imagen 9. Grafica de señal analógica

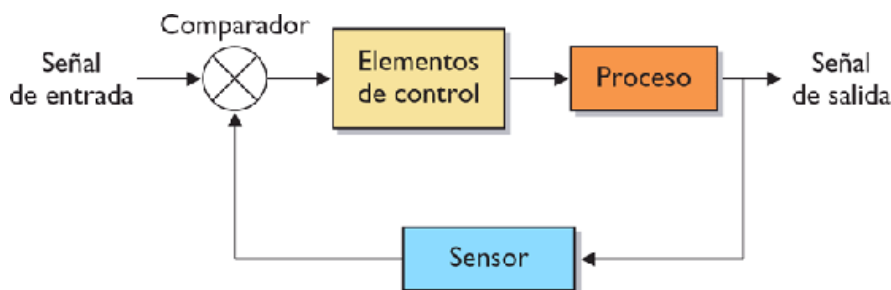


(Hernández, 2014)

2.3.1.3. Circuito de lazo cerrado

Los circuitos de lazo cerrado son aquellos en que la señal de salida de algunos sensores son la señal de control de otros, lo cual significa que automatizan la monitorización y control sin que un agente externo mande otra señal. Dentro del sistema alguna señal de salida realimenta la señal de entrada. Se tiene que tomar en cuenta que los márgenes de error de las señales de salida deben ser muy pequeñas, ya que, la medida de salida afecta directamente sobre las señales de control de entrada del siguiente sensor (Educa Madrid, SF).

Imagen 10. Sistema de lazo cerrado



(Guevara, 2018)

2.3.2. Sensores

Los sensores inteligentes son dispositivos electrónicos que llevan integrados circuitos que son compatibles con protocolos y estándares de comunicación disponibles para la red IoT como WIFI, LoraWAN, entre otros. El objetivo de estos dispositivos es extraer los datos de un sistema en común para controlar, monitorizar y/o automatizar. Existen varios tipos de sensores según su naturaleza, digital o analógica.

Los sensores de control pueden ser de cualquiera de las dos naturalezas, son dispositivos que necesitan un controlador o microcontrolador para su funcionamiento, estos pueden llegar a monitorizar, sin embargo, el trabajo principal de este tipo de sensores es controlar los procesos.

Por otro lado, los sensores de monitorización utilizan microcontroladores, pero estos no tienen un programa intermedio, por lo que no reciben señales, solo las transmiten.

2.3.2.1. Sensor de turbidez

El dispositivo emplea luz para identificar las partículas flotantes en el agua al evaluar la cantidad de luz que pasa y el grado de dispersión, lo cual varía con la concentración de sólidos en suspensión (TSS) presente en el líquido (Cotecno, 2024).

2.3.2.2. Sensor de pH

Estos sensores son dispositivos utilizados para medir el pH de una solución. (Arduino, 2022). Los sensores de pH tienen el objetivo de ver la acidez o alcalinidad en una escala del 0 al 14 donde el número medio, 7, es neutral menor a 7 ácido y mayor alcalino. Su composición es de un electrodo de vidrio que se usa para sumergir en una solución y mide la diferencia del potencial electrónico entre el electrodo y la solución.

Se usa principalmente en ambientes industriales como control en la industria alimentario, agricultura, control de agua potable e investigación científica. Este sensor es especialmente de monitorización, quiere decir que ayuda a saber el pH que se encuentra en tiempo real el líquido más no reaccionar a su solución.

2.3.2.3. Electroválvula

Los dispositivos responden a pulsos eléctricos y controlan el flujo de fluidos mediante la corriente que circula a través del solenoide, permitiendo abrir o cerrar la válvula según se requiera (DISTRITEC, 2022). Su diseño varía según las necesidades del sistema, adaptándose a diferentes diámetros de tuberías medidos en pulgadas.

Este sensor es principalmente de control. Significa que recibe impulsos electrónicos para actuar de alguna manera, en este caso entrada/salida. Además, este no permite monitorizar ya que no envía información hacia el microcontrolador.

2.3.2.4. Sensor de medición de caudal

El caudal de agua ingresa al sensor y hace girar una turbina, la turbina está unida a un imán que activa un sensor de efecto Hall, que a su vez emite un pulso eléctrico que puede ser leído por la entrada digital de un Arduino o PLC. (Electronics, 2024). La naturaleza de este sensor es el monitoreo, significa que envía pulsos al microcontrolador para saber el estado del sistema que se está monitoreando.

Además, se puede hacer una red de sensores en donde un impulso de monitorización active uno de control. Por ejemplo, si el caudal da un valor correcto según lo que se necesite en el sistema manejado este puede habilitar la válvula de compuerta.

2.3.2.5. Controladores

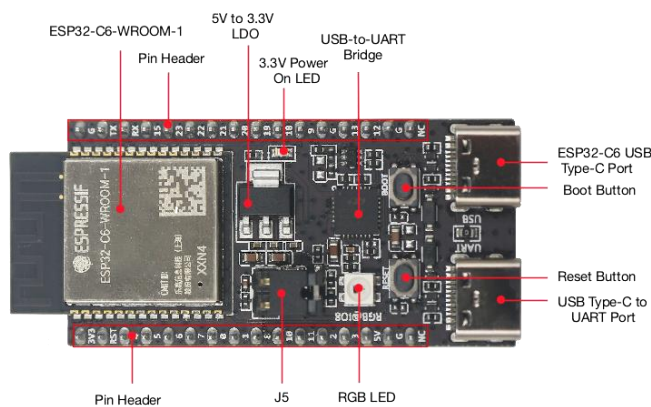
Una unidad microcontroladora (MCU) es esencialmente un pequeño ordenador en un solo chip. Está diseñado para gestionar tareas específicas dentro de un sistema integrado sin necesidad de un sistema operativo complejo. (IBM, 2024)

Los controladores o llamados también, por su tamaño, microcontroladores son los responsables de recibir y transmitir los pulsos electrónicos y convertir señales analógicas a digitales. Esto permite controlar y monitorear diferentes ámbitos como casas, agricultura entre otros automatizando los procesos y haciendo que estos sean fáciles de controlar o monitorizar desde un dispositivo.

Existen varios tipos de microcontroladores desde más avanzados hasta los más básicos, algunos cuentan como módulos integrados de Wifi y bluetooth. Entre los principales están, Arduino, Ransperry y ESP con sus diferentes modelos. Arduino nano, Arduino uno, Arduino micro, ESP32, ESP8266, Rasperry Pi.

ESP32 es un microcontrolador avanzado con modulo WIFI que permite hacer IoT con mayor facilidad, además existen muchos sensores compatibles con este controlador. Además, de ser compatibles con múltiples plataformas de programación.

Imagen 11. Microcontrolador



(Gabriela, 2023)

2.3.3. Protocolos

2.3.3.1. MQTT

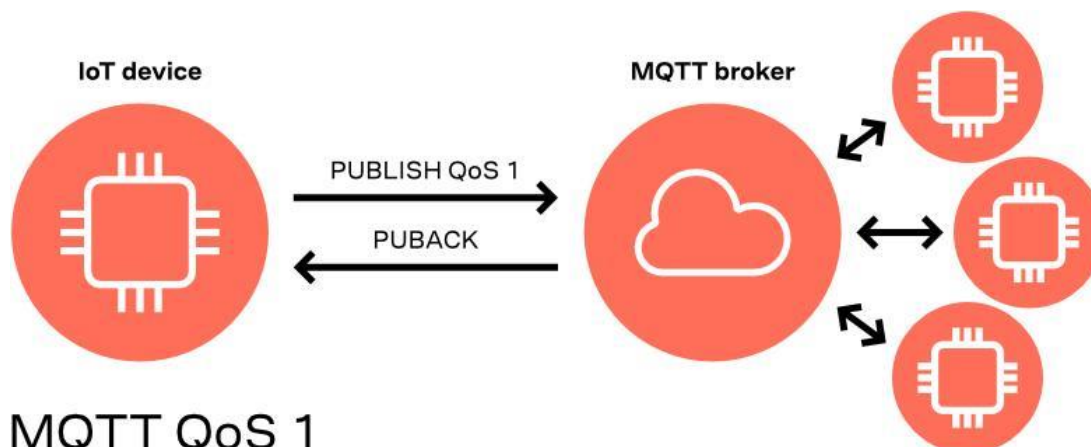
MQTT es un protocolo de mensajería basado en estándares, diseñado para conexiones M2M (de máquina a máquina). Este protocolo permite la comunicación entre dispositivos de diversas naturalezas, como sensores inteligentes, laptops, computadoras y otros equipos que necesitan transmitir y recibir pequeños volúmenes de datos, incluso en condiciones de recursos limitados y anchos de banda bajos. MQTT admite tanto conexiones del dispositivo a la nube como de la nube al dispositivo. (AWS, 2022)

Este protocolo funciona según el modelo de publicación y suscripción, esto significa, que se maneja por medio de un patrón de envío para desacoplar el remitente o editor del mensaje y el receptor o suscriptor. Esto funciona ya que, hay un componente medio llamado agente que controla la comunicación entre el editor y el suscriptor. De esta manera el agente filtra los mensajes y los distribuye entre los suscriptores correctos.

Al implementar Internet de las Cosas (IoT), es fundamental considerar ciertas características que aseguren conexiones seguras, ligeras, escalables y bien integradas. En el ámbito de IoT, existen diversos protocolos como MQTT; entre los más competitivos están CoAP, HTTPS, AMQP, LwM2M, entre otros. Sin embargo, MQTT presenta varias ventajas clave: su comunicación ligera de datos permite operar con anchos de banda muy bajos y recursos limitados. Además, su velocidad de transmisión es una gran ventaja para aplicaciones que requieren datos en tiempo real. (MQTT, 2024)

Sin embargo, un gran problema de usar este protocolo, muchas veces, es la seguridad. Por esta razón, MQTT ha implementado la ventaja de poner certificados de autenticidad y seguridad como TLS o SSL además de autenticación de dispositivos de entrada y salida. También, se puede tener la opción de implementar una red de área virtual (VPN) que permitirá las conexiones totalmente aisladas a los dispositivos en caso de requerir seguridad adicional.

Imagen 12. Arquitectura de MQTT

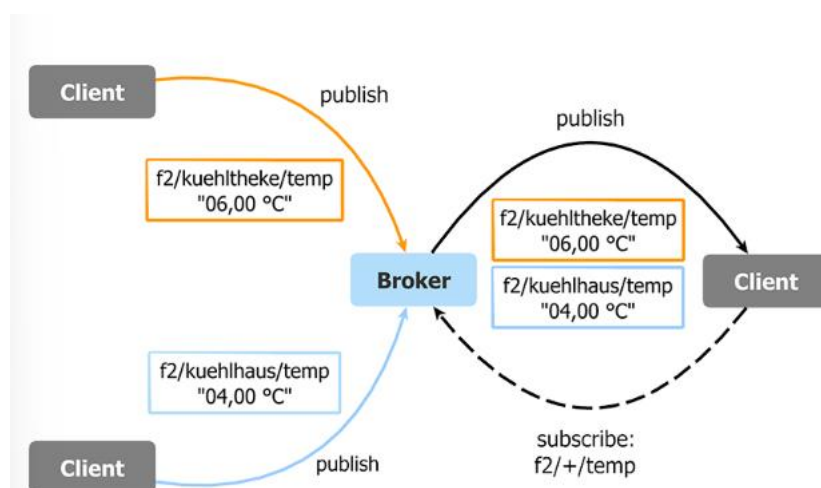


MQTT QoS 1

(B Priya, 2024)

El envío y recepción de mensajes se hace mediante el nombre de dominio o IP a donde se esté enviando los mensajes por medio del agente, nube o bróker y el tópicos que se quiere suscribir. Los tópicos son cadenas de caracteres de estructuras jerárquicas como árboles. Su estructura es el nombre de la primera jerarquía seguido de las barras inclinadas separando los niveles superiores o particiones. Se debe tener en consideración que un protocolo es simplemente un intermediario y mediador no es un almacén o memoria de datos, si es necesario guardar estos datos se necesitaría la creación de un almacén de datos (W&T, 2025).

Imagen 13. Conexiones con protocolo



(W&T, 2025)

2.3.3.2. AWS

Es una nube que ofrece servicios en casi todo el mundo, tiene varias funciones de utilidad que se suelen dividir en servicios pueden ser PAAS, SAAS, IAAS, entre otros. Dependiendo de la necesidad que se tenga en el momento. Ente sus servicios cuenta con módulos para IoT con diferentes protocolos como Lora WAN, HTTPS o MQTT. AWS puede integrar sus servicios entre sí para una correcta comunicación.

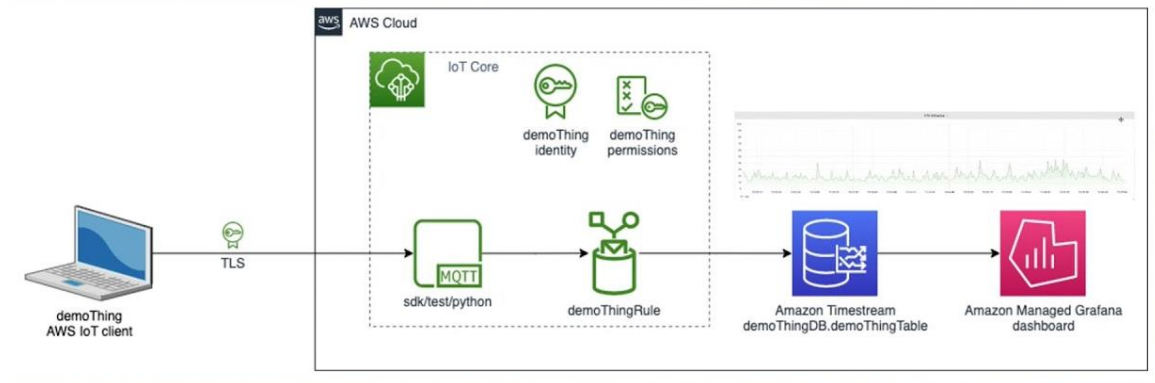
AWS IoT core

Es un servicio en la nube de Amazon, entra en la categoría PAAS, que permite conectar dispositivos IoT a la nube y facilitar su comunicación con otros servicios de AWS. Este servicio resuelve problemas de seguridad en las comunicaciones, ya que, al establecer la conexión entre un dispositivo y una suscripción MQTT, se generan certificados únicos para cada producto, lo que permite autenticar la conexión entre el Publicador y el Suscriptor. Además, proporciona encriptación de datos para mayor protección. Amazon también ofrece la posibilidad de crear redes privadas virtuales (VPN) para garantizar comunicaciones más seguras. (AWS, 2020)

AWS TimeStream

AWS Timestream es un servicio gestionado de base de datos en la nube, clasificado como DBaaS (Database as a Service), diseñado específicamente para manejar datos de series temporales. Es rápido, escalable y totalmente administrado, lo que lo convierte en una excelente opción para aplicaciones en tiempo real, como el Internet de las Cosas (IoT). El motor de Timestream almacena temporalmente los datos en memoria, para luego enviarlos a una base de datos histórica, donde pueden permanecer el tiempo que el usuario configure. Esto permite acceder y analizar tanto datos recientes como históricos de manera eficiente, sin necesidad de especificar su ubicación, garantizando que los datos más recientes siempre estén disponibles. Los datos se pueden enviar a Timestream a través de servicios como AWS IoT Core, Amazon MSK, entre otros. Además, es compatible con herramientas de análisis como Grafana y Amazon QuickSight, entre otros. Que facilitan la visualización y análisis de los datos para obtener información valiosa en tiempo real y en el largo plazo. (AWS, 2020)

Imagen 14. Comunicación de datos entre servicios



(AWS, 2023)

2.3.3.3. GRAFANA

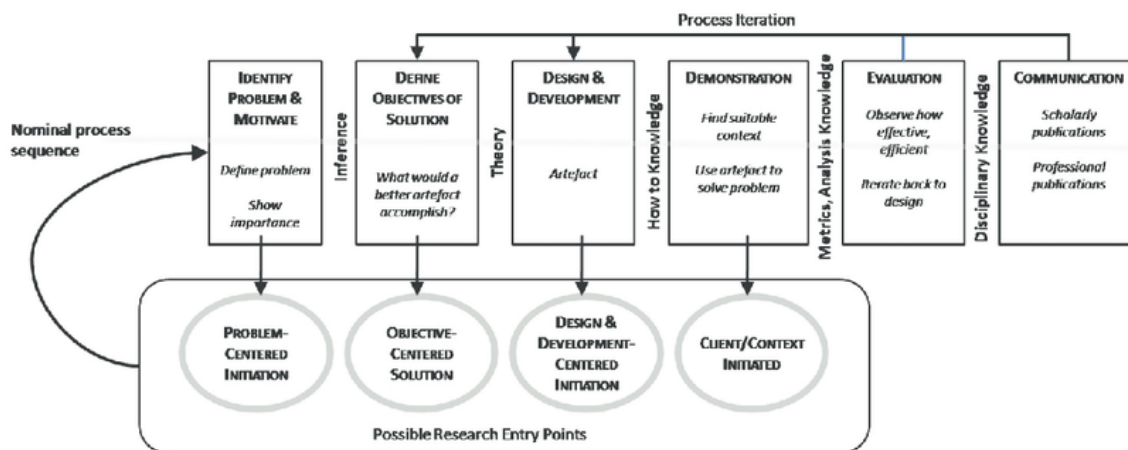
Es un software de código abierto que funciona para ejecutar análisis de datos en gráficas, estos datos pueden ser en tiempo real. Además, deja analizar cantidades enormes de datos de diferentes aplicaciones. Los controles y dashboards del software son totalmente personalizables. Permite que los datos estén unificados, que todos los datos se puedan visualizar y tiene paneles que cualquiera puede usar. El software se conecta con muchos motores de base de datos tanto *in situ* o en la nube y la aplicación directamente puede tener 3 modelos de uso en nube, local o híbrido. (Martinez, 2021)

3. Capítulo 3: Metodología

3.1. Design Science

Es una metodología con enfoques interactivos para abordar problemas, utilizada en investigaciones o proyectos enfocados en soluciones innovadoras para dificultades prácticas. Se emplea principalmente en el desarrollo de sistemas o artefactos, como modelos, marcos o prototipos, dentro de un dominio específico. Esta metodología se hace en un ambiente controlado de interacción. Por esta razón esta se divide en dos acciones, la primera es diseñar y la segunda investigar el funcionamiento del artefacto. Por medio del contexto en el que se encuentre la evaluación puede ser mediante interacciones reales si el dispositivo es construido o por simulaciones o prototipos. Es importante antes de aplicar la metodología entender el ambiente actual en el que se encuentra el sistema en caso de ser un integración o mejora. Estas acciones se visualizan en seis actividades organizadas secuencialmente presentadas en la siguiente figura (Lawrence, 2010).

Imagen 15. Dising Science



(Lawrence, 2010)

3.1.1. Identificación de problemas y motivación

Esta actividad pretende determinar los problemas y la motivación de la investigación o trabajo específico que justifica la búsqueda de la solución. Además, ayuda a bordar el verdadero valor que tiene el proyecto y la complejidad de este. Esto es muy importante para la comprensión sólida del estado actual del problema y la importancia de encontrar la solución. Esta etapa está definida en dos pasos clave el primero es la definición del problema; este consiste en la descripción detallada y precisa de lo que se quiere abordar, esto permite diseñar las soluciones más efectivas. Como segundo paso se tiene la justificación del valor de la solución; consiste en

el argumento de porque es importante resolver el problema, suele incluir la relevancia del problema en un contexto y ambiente específico.

3.1.2. Determinar objetivos de la solución

Esta actividad consiste en determinar lo que se espera lograr, significa que se puede lograr una o más cosas, por lo tanto, los objetivos pueden ser más de uno. Además, en este punto se debe considerar partes del diseño como los requisitos funcionales y no funcionales sabiendo lo que debe lograr el dispositivo, proyecto, diseño o prototipo.

Es importante que todos estos objetivos estén estrechamente relacionados con el proyecto contribuyendo a la solución de este. Los objetivos deben estar alineados con relevancia y pertenencia del contexto específico, es así como se tomará en cuenta las restricciones, recursos disponibles, tiempo y viabilidad técnica del proyecto estableciendo metas realistas y alcanzables. (Reyes, 2022)

3.1.3. Diseño y desarrollo

Esta actividad, en realidad, se divide en dos aspectos importantes que dependen de la naturaleza del proyecto. Los proyectos pueden ser sistemas, artefactos, arquitecturas o prototipos, y dependiendo de esto, se puede determinar el alcance de cada uno, ya sea que llegue al diseño o al desarrollo. Sin embargo, siempre que solo se llegue al diseño, debe existir un prototipo, simulación o prueba que demuestre que este funcionará dentro del desarrollo del proyecto. En este caso, el prototipo, la simulación o la prueba estarán dentro del desarrollo mismo.

3.1.4. Demostración

Esta actividad ayuda a saber, visualizar y probar el funcionamiento del proyecto realizado. El artefacto por probar puede ser definitivo o simplemente un prototipo, simulación o diseño. En esta demostración el artefacto debe estar alineado y cumplir con los requerimientos funcionales y no funcionales dependiendo de los objetivos planteados en el inicio de la metodología. Además, esta fase está enlazada directamente con la evaluación, ya que, sabiendo que el dispositivo funciona se determinará la calidad de este.

3.1.5. Evaluación

La penúltima actividad mide la funcionalidad del artefacto y que tan bien respalda la solución del problema. Compara los objetivos de la solución con los resultados reales observados en la demostración del producto. Esta fase es decisiva, ya que, si el artefacto no está en las condiciones esperadas se puede volver pasos atrás e iterar de nuevo en procesos necesarios o documentar el resultado y dejar mejoras o recomendaciones para proyectos futuros.

3.1.6. Comunicación

La fase de comunicación tiene como objetivo informar de manera clara y completa todos los aspectos del problema y el artefacto diseñado a los interesados en el proyecto. Esto incluye a todos los stakeholders, como usuarios, ingenieros, desarrolladores, diseñadores, entre otros. Es fundamental adaptar el enfoque comunicativo según las necesidades y el nivel de conocimiento de cada grupo, asegurando que la información sea relevante y adecuada para alcanzar los objetivos planteados.

4. Capítulo 4: Diseño

4.1. Diagnóstico de la planta de tratamiento de agua potable

El diagnóstico de agua potable es muy importante para evaluar su operatividad, eficiencia y sostenibilidad. A continuación, se describen los aspectos clave que deben abordarse:

4.1.1. Recolección de datos de la planta de tratamiento

La planta de tratamiento considerada tiene características compactas y cuenta con un modelo de potabilización que asegura la potabilización del agua y cumple la norma 1108 del INEC. El cumplir esta norma en el Ecuador es fundamental para considerar la potabilización. La planta considera el análisis químico, físico y bacteriológico del agua. En la imagen 16 se puede visualizar el plano civil de la planta de tratamiento, con sus etapas definidas.

La planta de tratamiento antes mencionada cuenta con la siguiente información para el levantamiento de datos:

Diagnóstico de una Planta de Agua Potable

Descripción General

Nombre de la planta: Planta de agua potable de Palora

Ubicación geográfica: Cantón Palora, Provincia de Morona Santiago

Capacidad de producción: 27 l/s 23322.8 m³/día

Tipo de fuente de agua: Superficial

Infraestructura tecnología: WiFi

Infraestructura y Equipamiento

Sistemas de captación:

Descripción de la estructura utilizada para captar el agua. El sistema de captaciones para dotar de agua potable a Palora son del tipo caucasiana de rejilla lateral en un número de tres. Consta de muros laterales, rejilla lateral, azud de control y desarenador.

Tratamiento:

Estructura de Aforo. - el aforo o medida del caudal se realizará con la instalación de un macro medidor. El mismo que será automatizado.

Luego del macro medidor se encuentra la caja de válvulas la misma que se abrirán o se cerrarán de acuerdo con el valor de la turbiedad.

Si la turbiedad es mayor que 1 pasará a la estructura de mezcla rápida donde se dosificará el químico coagulante en este caso (sulfato de aluminio).

Floculación Está será de paredes horizontales con velocidades mínimas. Siendo el flujo de tipo laminar

Sedimentación

Sedimentador, de agitación hidráulica, con sus respectivas paredes inclinadas y dotado con módulos de sedimentación de alta tasa tipo panal construidos en lámina de ABS.

Filtración

Un (1) Tanque Filtro de retro lavado hidráulico manual de flujo descendente y de filtración rápida con lecho filtrante mixto (antracita, arena sílice tipo cuarzo redondeada y grava redondeada). 95%

Cloración

La Cloración es del tipo cloro gas descargado a inyección al tanque correspondiente.

Estabilizador de PH este químico a base de cal será dosificado al final del tanque de cloración si fuese necesario el cual será automático.

Equipos de bombeo:

Consta de 2 bombas dosificadoras de 1 hp.

Una bomba de 3 hp para distribución interna de la planta es decir agua potable para el laboratorio casa de guardián etc.

Sistema de distribución:

Longitud de la red: La red de distribución de Palora tiene una longitud de 70 km aproximadamente

Materiales: PVC de 1 mpa de presión de trabajo y la unión es de tipo elastomérica.

Evaluación del Proceso

Calidad del agua cruda: Parámetros físicos, químicos y microbiológicos. Cumple con la norma 1108 del INEN en lo referente a parámetros de agua para consumo humano.

Calidad del agua tratada:

Comparación con los estándares nacionales o internacionales. En el Ecuador se considera agua tratada la que cumple los estándares dados en la norma 1108 del INEN correspondiente a agua potable.

Identificación de Problemas

Ineficiencias en los procesos de tratamiento: procesos 100% humanos.

Fases Graficas:

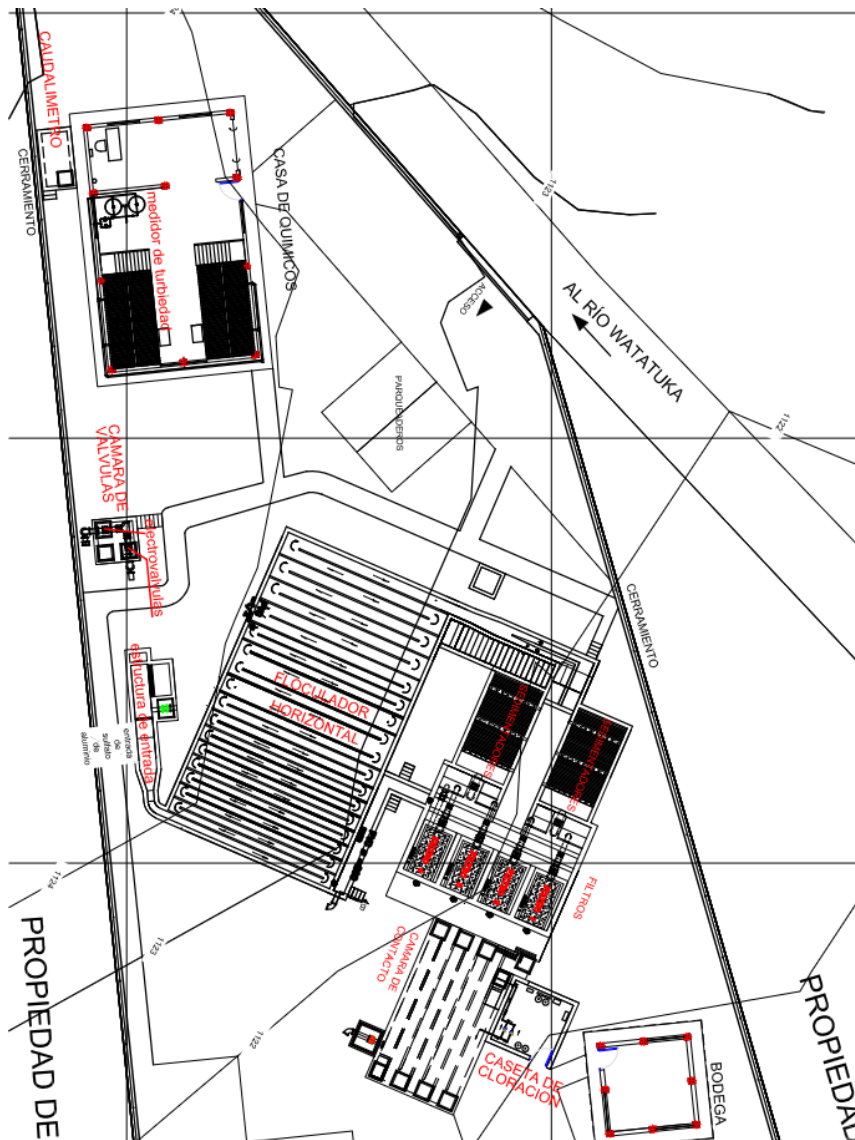


Imagen 16. Planta de tratamiento Plano – (Jorge Poveda A, 2020)

4.1.2. Identificación de problemas y motivación

La planta de tratamiento de agua potable cuenta con numerosos procesos organizados para la potabilización del agua. Sin embargo, todos estos procesos son manuales y dependen del personal humano, lo que genera ciertas ineficiencias. En primer lugar, es necesaria la supervisión física del macro-medidor para controlar el caudal de entrada, lo que conlleva riesgos de sobrecarga cuando se superan los 27 l/s.

A lo largo del proceso, la medición de turbidez y la gestión de válvulas también dependen de la intervención humana, lo que puede retrasar decisiones críticas. Además, la medición del pH y el reparto de productos químicos para su corrección se realizan manualmente, aumentando el riesgo de errores y prolongando los tiempos de respuesta. Cabe mencionar que la planta de tratamiento se encuentra a una considerable distancia de la ciudad, lo que requiere que siempre haya más de tres personas presentes para su manejo.

Por esta razón, una forma de prevenir estos problemas es automatizar todos los procesos mencionados. Esto reduciría la necesidad de intervención manual y proporcionaría un monitoreo continuo, permitiendo un tiempo adecuado para intervenir en caso de problemas. Para lograrlo, se pueden integrar sensores para el control del caudal, la turbidez y el pH, evitando así problemas de sobrecarga y minimizando los errores humanos.

4.1.3. Objetivos del proyecto

- Diseñar un sistema de gestión y control mediante dispositivos IoT para el monitoreo y automatización de plantas de tratamiento de agua potable.
- Proponer la integración de dispositivos IoT permitiendo la comunicación y transmisión de datos entre sensores.
- Optimizar el monitoreo, la detección de incidentes y la gestión de recursos hídricos en tiempos correctos para la acción ante incidentes

4.1.4. Diseño y desarrollo

4.1.4.1. Requerimientos del usuario Requerimientos funcionales

Especificación de requerimiento funcional:	
Nombre	Medición de caudal por medio de sensores

Requerimiento funcional N°	RQF1
Prioridad	Alta
Descripción:	
El sistema debe ser capaz de capturar y registrar datos extraídos de los sensores que marcan el caudal dando señales analógicas y digitales tanto para monitorización como para control del RQF2.	

Tabla 1. Requisito funcional 1 – Realizado por el autor de la tesis

Especificación de requerimiento funcional:	
Nombre	Control de válvulas de compuerta
Requerimiento funcional N°	RQF2
Prioridad	Alta
Descripción:	
El sistema debe ser capaz de controlar válvulas electrónicas después de detectar el caudal o turbidez y dependiendo si este es apto para la planta de tratamiento o no.	

Tabla 2. Requisito funcional 2 – Realizado por el autor de la tesis

Especificación de requerimiento funcional:	
Nombre	Monitorización de pH por medio de sensores
Requerimiento funcional N°	RQF3
Prioridad	Alta
Descripción:	
El sistema debe ser capaz de capturar y registrar datos extraídos de los sensores que marcan el pH en modo de monitorización.	

Tabla 3. Requisito funcional 3 – Realizado por el autor de la tesis

Especificación de requerimiento funcional:	
Nombre	Monitorización de turbidez por medio de sensores.
Requerimiento funcional N°	RQF4
Prioridad	Alta
Descripción:	
El sistema debe ser capaz de capturar y registrar datos extraídos de los sensores que marcan la turbidez en modo de monitorización.	

Tabla 4. Requisito funcional 4 – Realizando por el autor de la tesis

Especificación de requerimiento funcional:	
Nombre	Control de bombas de goteo mediante sensores
Requerimiento funcional N°	RQF5
Prioridad	Alta

Descripción:
El sistema debe ser capaz de controlar abrir la válvula de sulfato según los factores mencionados en RQF3 y RQF4

Tabla 5. Requisito funcional 5 – Realizado por el autor de la tesis

Requerimientos no funcionales

Especificación de requerimiento no funcional:	
Nombre	Tiempo de respuesta del sistema
Requerimiento no funcional N°	RNF1
Prioridad	Alta
Descripción:	
El sistema debe ser capaz de tener un tiempo de respuesta de lectura y procesamiento de datos apto para la monitorización y actuación. El tiempo apto debe incluir la transmisión de datos al servidor en la nube y la activación de la válvula de compuerta si es necesario.	

Tabla 6. Requisito no funcional 1 – Realizado por el autor de la tesis

Especificación de requerimiento no funcional:	
Nombre	Seguridad de la comunicación
Requerimiento no funcional N°	RNF2
Prioridad	Alta
Descripción:	
El sistema debe ser capaz de tener un tiempo de respuesta de lectura y procesamiento de datos apto para la monitorización y actuación. El tiempo apto debe incluir la transmisión de datos al servidor en la nube y la activación de la válvula de compuerta si es necesario.	

Tabla 7. Requisito no funcional 2 – Realizado por el autor de la tesis

Especificación de requerimiento no funcional:	
Nombre	Compatibilidad con estándares abiertos
Requerimiento no funcional N°	RNF3
Prioridad	Alta
Descripción:	
El sistema debe ser capaz de conectarse con protocolos y dispositivos del diseño por medio de estándares compatibles según buenas prácticas.	

Tabla 8. Requisito no funcional 3 – Realizado por el autor de la tesis

Especificación de requerimiento no funcional:	
Nombre	Usabilidad
Requerimiento no funcional N°	RNF4
Prioridad	Media

Descripción:
El sistema debe ser capaz de dar la facilidad de ser usable por una persona no obligatoria mente del área tecnológica.

Tabla 9. Requisito no funcional 4 – Realizado por el autor de la tesis

Especificación de requerimiento no funcional:	
Nombre	Confiability del sistema
Requerimiento no funcional N°	RNF5
Prioridad	Alta
Descripción:	
El sistema debe tener una disponibilidad mínima del 90% durante su operación con recuperación automática de fallos como reconexiones a protocolos como MQTT.	

Tabla 10. Requisito no funcional 5 – Realizado por el autor de la tesis

4.1.4.2. Diagrama de actividades

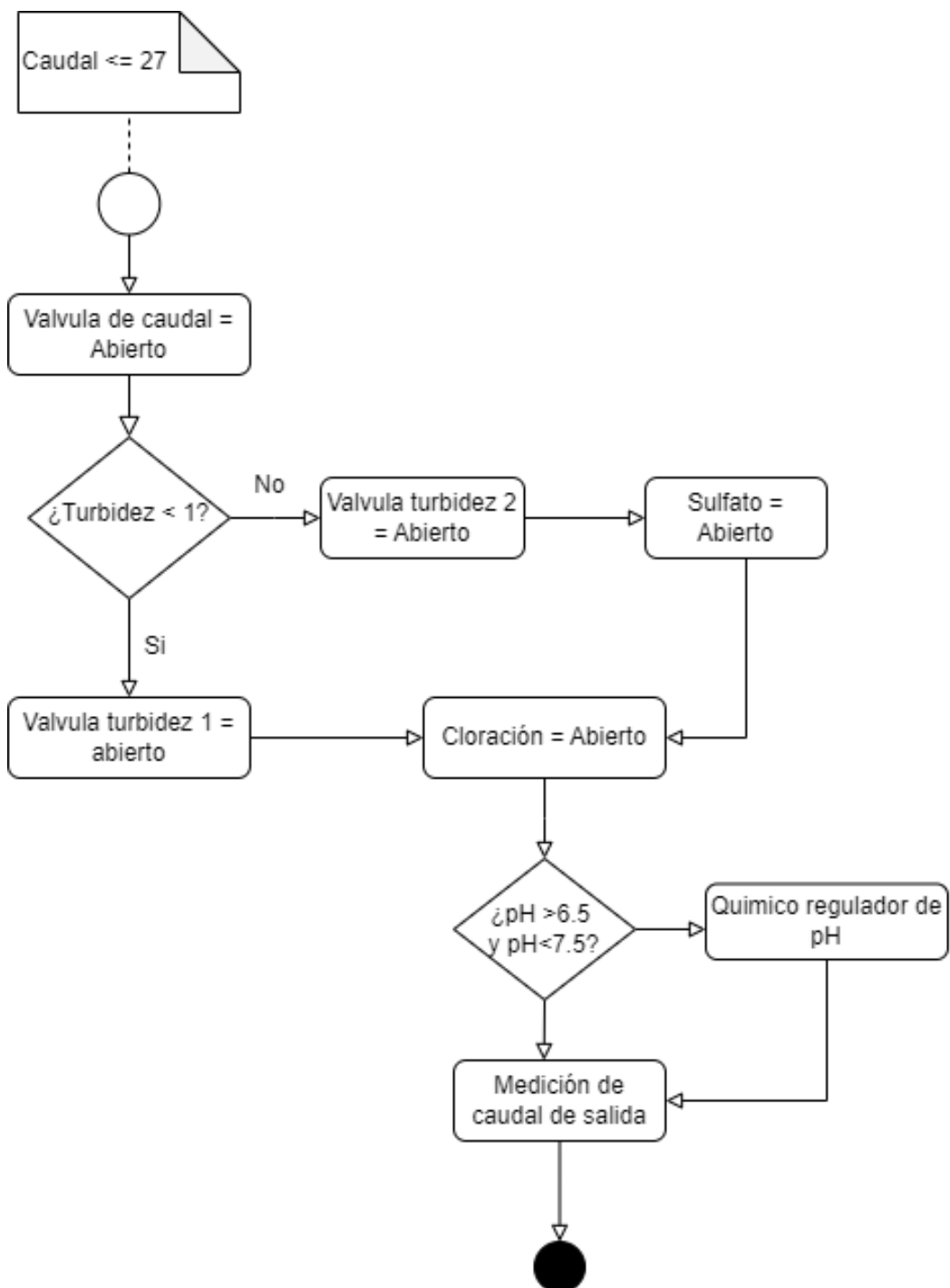


Imagen 17. Diagrama de actividades

4.1.4.3. Hardware y equipo correspondiente

En el presente trabajo de titulación se tienen presentes dispositivos o sensores a la vanguardia de la tecnología IoT especializados en infraestructuras civiles.


SENSOR DE MEDICIÓN DE CAUDAL	
Grafica	
Descripción	<ul style="list-style-type: none"> • Macromedidor DN100 LD300 • Voltaje 24V DC • Señales de salida: 0~10ma/4~20ma /pulse/modbus/hart • Temperatura de -10 a 90 °C • Con relé

Tabla 11 Sensor de caudal – (DN100, 2023)


SENSOR DE MEDICIÓN DE TURBIDEZ	
Grafica	
Descripción	<ul style="list-style-type: none"> • Voltaje: 3,3 V ~ 5 V (con circuito antirretroceso) • Salida: analógica estándar (puede configurar salida alta y baja) • Pines disponibles: VCC, SALIDA, GND

Tabla 12. Sensor de turbidez – (Sarabati, 2023)


SENSOR DE MEDICIÓN DE PH	
Grafica	
Descripción	<ul style="list-style-type: none"> • Voltaje de 5V • Consumo de 5 – 10 mA • Rango de medición 0 – 14pH • Precisión +/- 0.1pH • Tiempo de respuesta <=5s

Tabla 13. Sensor de PH


ELECTRO VALVULA PARA MICROCONTROLADORES	
Válvula de compuerta	
Descripción	<ul style="list-style-type: none"> • Voltaje: 12 – 20V • Señal: Digital • Pines disponibles: VCC, SALIDA, GND • Con relé

Tabla 14 Electroválvula industrial

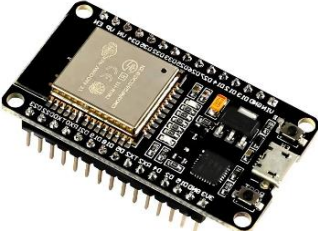
MODULO ESP32	
Grafica	
DESCRIPCIÓN	<ul style="list-style-type: none"> • CPU Xtensa de doble núcleo de 32 bits • 160 o 240 MHz • Memoria 520KiB SRAM • WIFI 802.11 b/g/n • Seguridad con WPA2

Tabla 15. Microcontrolador ESP32 - (Orellana, 2023)


Relé	
DESCRIPCIÓN	<ul style="list-style-type: none"> • Dispositivo electromagnético • Tres contactos de control (NA), (NC) (COM)

Tabla 16. RELÉ – (LEDBOX, 2025)

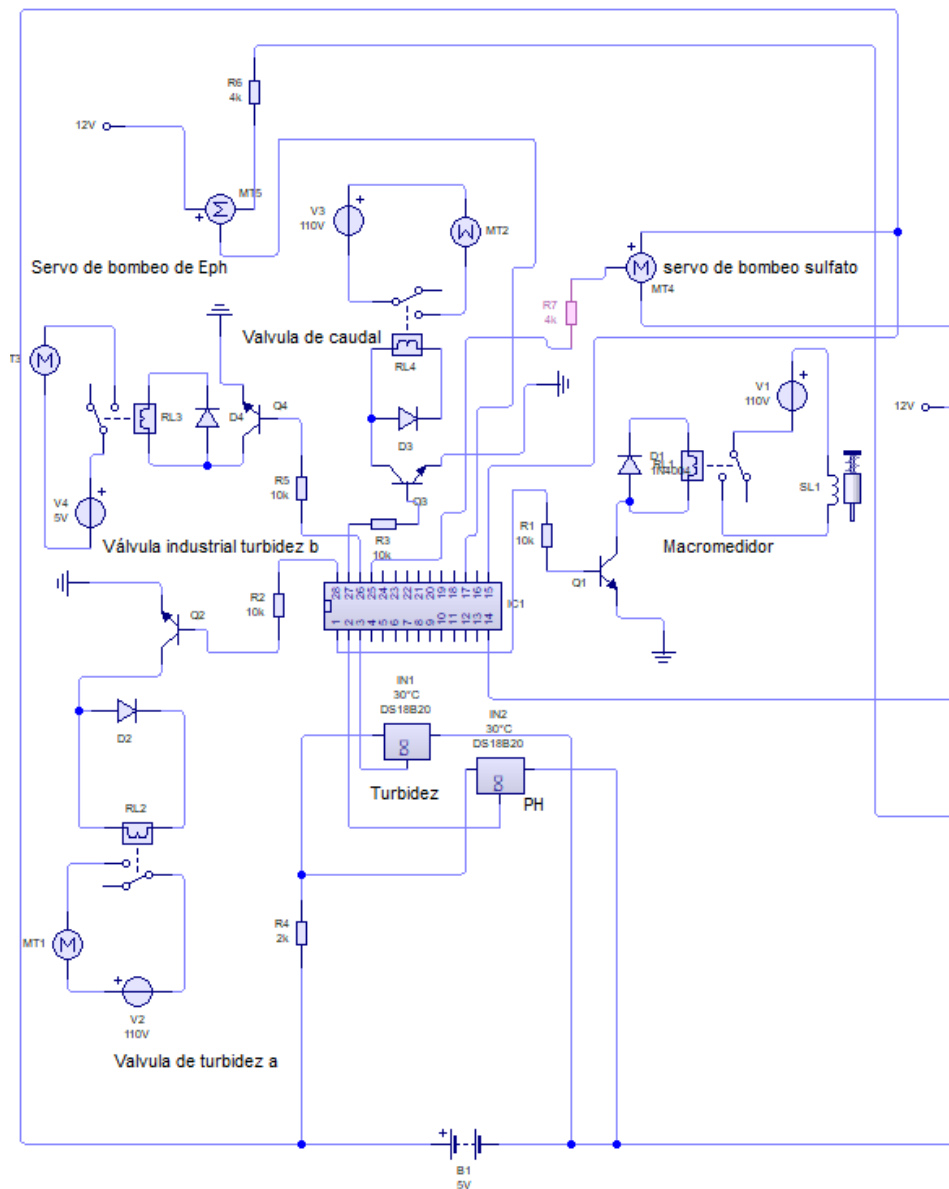
Servomotor	
DESCRIPCIÓN	<ul style="list-style-type: none"> • Alimentación: 12V • Potencia: 12W • Consumo de corriente: 1A

Tabla 17. Servomotor – (Robodacta, 2025)

4.1.4.4. Esquema de conexión

En el siguiente esquema de conexión se presenta el diseño electrónico del sistema de IoT, donde se pueden observar los niveles de conexión de cada uno de los sensores. Por un lado, se encuentran los sensores que operan a voltajes y corrientes altos, por lo que se necesita un módulo relé para transmitir la información hacia un microcontrolador. Existen sensores como el de turbidez y pH que no necesitan el módulo relé, ya que funcionan al mismo voltaje que el módulo. Los servomotores que se conectan a las válvulas de las bombas también operan a voltajes no tan alto, por lo que tampoco necesitan un relé para su funcionamiento, sin embargo, se debe detener en consideración colocar resistencias para evitar daños en el microcontrolador.

Ilustración 1. Esquema de conexión IoT



4.1.4.5. Programa de simulación con Python

Al ser un diseño se elabora un código para la simulación y prueba de validez de este. El código esta echo en Python y la simulación es de cada uno de los sensores. Además, en el código se visualiza las conexiones hacia el servidor suscriptor MQTT en este caso AWS IoT CORE, en este servidor se cuenta con certificados de seguridad y autenticidad los cuales permiten conexiones seguras entre las dos partes de la conexión. Cada que el programa que publica se conecta a la nube MQTT para publicar hacia el servidor este autentica su conexión mediante el cliente_id y otras certificaciones.

Si la construcción del proyecto se lleva a cabo y es necesario ingresar el código en el microcontrolador ESP32 se puede hacer por medio de Arduino ID con el mismo código exceptuando las líneas de simulación del dispositivo y adjuntando sus librerías, como se visualiza en el punto 7 Anexo 1.

```
from AWSIoTPythonSDK.MQTTLib import AWSIoTClient
import random
import time
import json

# Configuración del cliente AWS IoT Core
client_id = "ESP32_P"
endpoint = "d04482212v9h1iqvjp2tj-ats.iot.us-east-1.amazonaws.com"
cert_path =
"C:/Users/pgpov/Documents/tesis/certificados/2c4868d4a1b2d28cdb45cebb3ab7f4552e65e63cec9fb4ab
dd2249357922cfc5-certificate.pem.crt"
key_path =
"C:/Users/pgpov/Documents/tesis/certificados/2c4868d4a1b2d28cdb45cebb3ab7f4552e65e63cec9fb4ab
dd2249357922cfc5-private.pem.key"
ca_path = "C:/Users/pgpov/Documents/tesis/certificados/AmazonRootCA1 (1).pem"

#Autenticación con el cliente
mqtt_client = AWSIoTClient(client_id)
mqtt_client.configureEndpoint(endpoint, 8883)
mqtt_client.configureCredentials(ca_path, key_path, cert_path)

mqtt_client.configureAutoReconnectBackoffTime(1, 32, 20)
mqtt_client.configureOfflinePublishQueueing(-1)
mqtt_client.configureDrainingFrequency(2)
mqtt_client.configureConnectDisconnectTimeout(10)
mqtt_client.configureMQTTOperationTimeout(5)

def connect_to_aws(): #conexión a aws
    try:
```

```

    mqtt_client.connect()
    print("Conectado a AWS IoT Core")
except Exception as e:
    print(f"Error conectando: {e}")
import time

def generar_clave_con_timestamp():
    return str(int(time.time() * 1000)) # Tiempo actual en milisegundos

def caudal(): #sensor de caudal
    return round(random.uniform(15, 33))

def turbidity(): #sensor de turbidez

    return round(random.uniform(0, 2), 2)

def pH(): #sensor de pH
    return round(random.uniform(5, 8), 2)

def valvula(estado): #sensor de valvula
    return "ABIERTA" if estado else "CERRADA"

def publicar_datos():
    try:
        while True:
            valor_caudal = caudal()
            valor_turbidez = turbidity()
            valor_ph = pH()
            secuencial = generar_clave_con_timestamp()
            sulfato = valvula()

            if valor_caudal > 27:
                valor_turbidez = 0
                valor_ph = 0
                estado_valvula1 = valvula(False)
                estado_valvula2 = valvula(False)
                estado_valvula3 = valvula(False)
                estabilizador_ph = valvula(False)
                sulfato = valvula(False)
            else:
                estado_valvula3 = valvula(True)
                if valor_ph < 6.5 and valor_ph > 7.5:
                    estabilizador_ph = valvula(True)
                else:
                    estabilizador_ph = valvula(False)
            if valor_turbidez > 1:
                estado_valvula1 = valvula(False)
                estado_valvula2 = valvula(True)

```

```

        sulfato = valvula(True)
    else:
        estado_valvula1 = valvula(True)
        estado_valvula2 = valvula(False)

# Consolidar datos en un mensaje JSON
mensaje = {
    "id": secuencial,
    "caudal": valor_caudal,
    "turbidez": valor_turbidez,
    "pH": valor_ph,
    "valvula1": estado_valvula1,
    "valvula2": estado_valvula2,
    "valvula3": estado_valvula3,
    "E pH": estabilizador_ph,
    "sulfato": sulfato
}

# Publicar mensaje JSON consolidado
mqtt_client.publish("PlantaAgua/sensores", json.dumps(mensaje), 1)
print(f'Publicado JSON: {mensaje}')

time.sleep(7200)
except KeyboardInterrupt:
    print("\nDesconectando...")
    mqtt_client.disconnect()

# Ejecutar conexión y publicación
connect_to_aws()
publicar_datos()

```

Imagen 18. Código de simulación de sensores en Python – De autoría propia

4.1.4.6. Conexión hacia el suscriptor AWS

El primer paso para establecer la conexión es la creación de una cuenta en AWS. Una vez realizada, se accede al panel de control, donde se encuentran algunos de los servicios principales de la plataforma, como se muestra en la *Ilustración 2*.

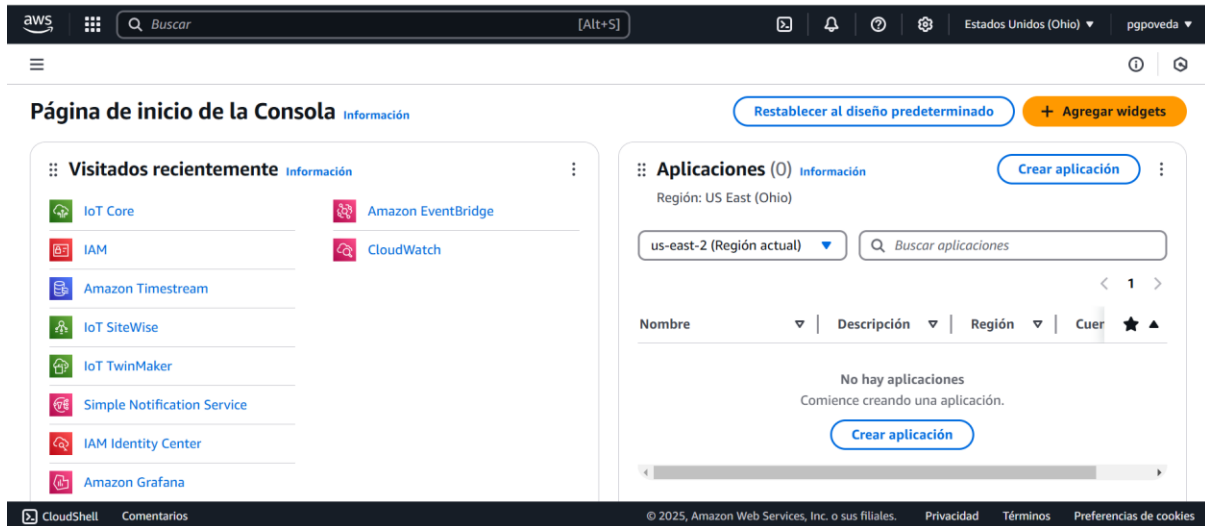


Ilustración 2. Consola de AWS – De autoría propia

Al acceder al servicio IoT core, que se encuentra como principal en la consola, se debe crear un objeto que será el controlador del circuito en este caso el que recibe el control sería el dispositivo ESP32. El primer paso es dirigirse hacia la pestaña objetos y crear el objeto como se visualiza en las *Ilustración 3*. Se puede escoger la regla y el tipo de certificado que se desea para seguridad y encriptación de datos, como se visualiza en la *Ilustración 4*.

Propiedades del objeto

Nombre del objeto

Ingrese un nombre único que contenga únicamente letras, números, guiones, dos puntos o guiones bajos. Un nombre de objeto no puede contener espacios.

Configuraciones adicionales

Puede utilizar estas configuraciones para agregar detalles que le ayuden a organizar, administrar y buscar sus objetos.

- ▶ Tipo de objeto: *opcional*
- ▶ Atributos de objeto que permiten búsquedas : *opcional*
- ▶ Grupos de objetos: *opcional*
- ▶ Grupo de facturación : *opcional*
- ▶ Paquetes y versiones : *opcional*

Sombra de dispositivo

Las sombras de dispositivos permiten que los dispositivos conectados sincronicen sus estados con AWS. También es posible obtener, actualizar o eliminar la información de estado de la sombra de este objeto mediante temas MQTT o HTTP.

- Sin sombra
- Sombra con nombre
Cree varias sombras con nombres diferentes para administrar el acceso a las propiedades y agrupe lógicamente las propiedades del dispositivo.
- Sombra sin nombre (clásica)
Un objeto solo puede tener una sombra sin nombre.

Ilustración 3. Creación de objetos - De autoría propia

Certificado de dispositivo

- Generar automáticamente un certificado nuevo (recomendado)**
Genere un certificado, una clave pública y una clave privada con la entidad de certificación de AWS IoT.
- Usar mi certificado
Utilice un certificado firmado por su propia entidad de certificación.
- Cargar CSR
Registre la entidad de certificación y use sus propios certificados en uno o varios dispositivos.
- Saltar la creación de un certificado en este momento
Puede crear un certificado para este objeto y asociar una política al certificado más adelante.

Cancelar
Anterior
Siguiente

Ilustración 4. Certificados de seguridad - De autoría propia

Es importante saber que la política es para adaptar los permisos que va a tener el objeto, por lo tanto, la política también se puede crear y adaptar a los permisos necesarios. En este caso se crea una política llamada **Controlador_AguaP** que el administrador tiene todos los permisos. De esta manera los administradores pueden crear roles que solo puedan entrar a las vistas. El detalle se visualiza en la Ilustración 5 y 6.

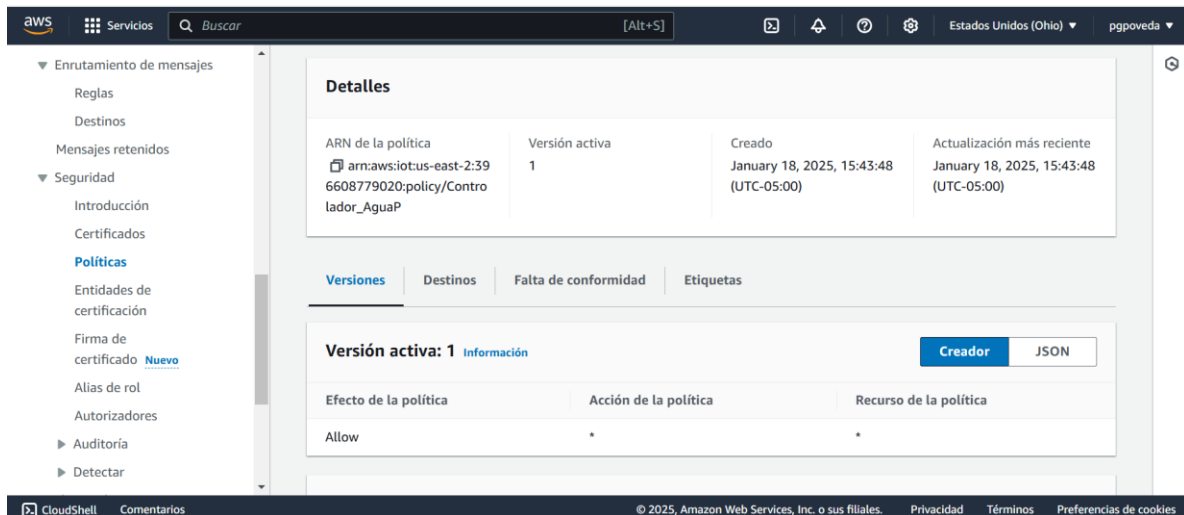


Ilustración 5. Permisos de la política Controlador_AguaP – De autoría propia



Ilustración 6. Política creada para el objeto – De autoría propia

Una vez creado el objeto y descargado todos los certificados y llaves se procede a insertar esto dentro del código, es importante que el código tenga la biblioteca para poder usar AWS como servicio para eso se importa con el siguiente código **from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient**. De esta manera el publicador, podrá conectarse con el suscriptor sin problema alguno. Además, la conexión será totalmente segura, ya que en cada conexión los dos puntos se conectarán con autenticación de claves encriptadas para empezar el envío y recepción de datos, como se visualiza en la Ilustración 7 y 8.

Certificado de dispositivo

Puede activar el certificado ahora o posteriormente. El certificado debe estar activo para que un dispositivo se conecte a AWS IoT.

Certificado de dispositivo
a97b29c438f...te.pem.crt

[Desactivar certificado](#) [↓ Descargar](#)

Archivos de claves

Los archivos de claves son exclusivos de este certificado y no se pueden descargar después de salir de esta página. Descárguelos ahora y guárdelos en un lugar seguro.

⚠ Esta es la única vez que puede descargar los archivos de claves correspondientes a este certificado.

Archivo de clave pública
a97b29c438f114933d1679f...ce29034-public.pem.key [↓ Descargar](#)

Archivo de clave privada
a97b29c438f114933d1679f...e29034-private.pem.key [↓ Descargar](#)

Certificados de entidad de certificación raíz

Descargue el archivo de certificado de entidad de certificación raíz correspondiente al tipo de punto de...

[Listo](#)

Ilustración 7. Descarga de certificados – De autoría propia

```

from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient
import random
import time
import json

# Configuración del cliente AWS IoT Core
client_id = "ESP32_P"
endpoint = "d02579861kkzf5jrdsvvo-ats.iot.us-east-2.amazonaws.com"
cert_path = "C:/Users/PALORA/Documents/Tesis/e122064d89c63121b17f181c17fccb3e0eb943fa7285f5ffcbfc15b0524d1e-certificate.pem.crt"
key_path = "C:/Users/PALORA/Documents/Tesis/e122064d89c63121b17f181c17fccb3e0eb943fa7285f5ffcbfc15b0524d1e-private.pem.key"
ca_path = "C:/Users/PALORA/Documents/Tesis/AmazonRootCA1.pem"

mqtt_client = AWSIoTMQTTClient(client_id)
mqtt_client.configureEndpoint(endpoint, 8883)
mqtt_client.configureCredentials(ca_path, key_path, cert_path)

mqtt_client.configureAutoReconnectBackoffTime(1, 32, 20)
mqtt_client.configureOfflinePublishQueueing(0)
mqtt_client.configureDrainingFrequency(2)
mqtt_client.configureConnectDisconnectTimeout(10)
mqtt_client.configureMQTTOperationTimeout(5)

def connect_to_aws():
    try:
        mqtt_client.connect()
        print("Conectado a AWS IoT Core")
    except Exception as e:
        print(f"Error conectando: {e}")

```

Ilustración 8. Claves y certificados ingresados en el programa – De autoría propia

Para saber si está bien la conexión y el objeto, enviamos el programa a correr junto con la suscripción de MQTT en AWS y que los datos sean igual tanto en la consola de impresión del programa como en el suscriptor, como se visualiza en la *Ilustración 9*.

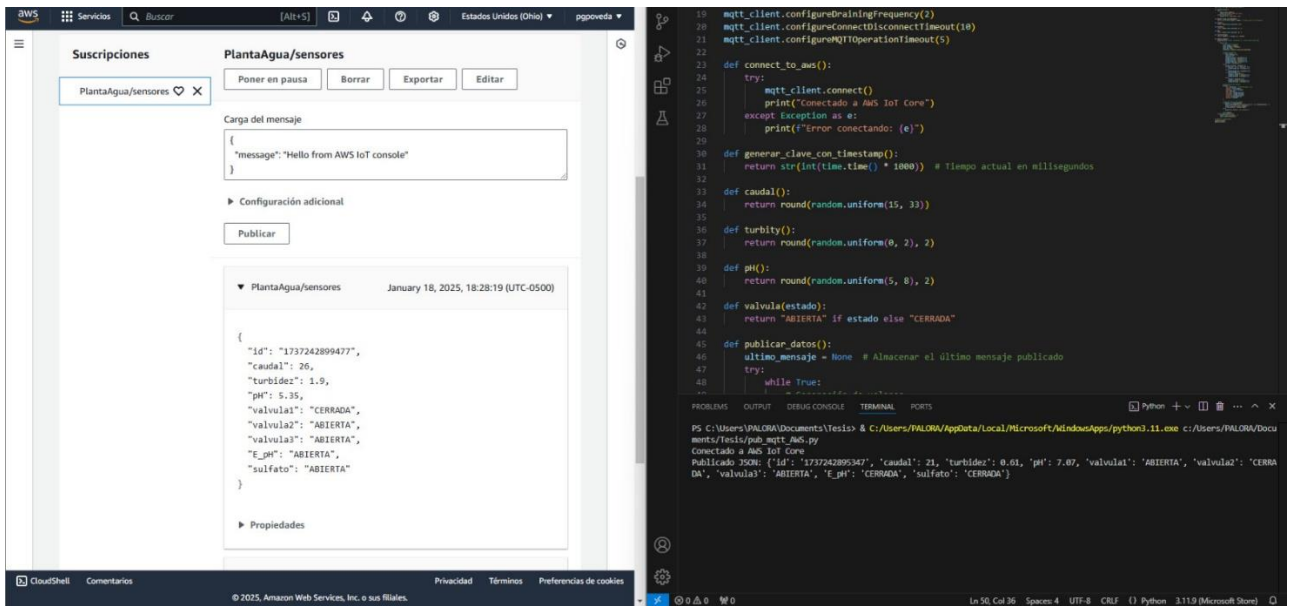


Ilustración 9. Envío de datos por medio del protocolo MQTT – De autoría propia

4.1.4.7. Conexión de AWS IoT CORE a TimeStream

Como primer paso se debe buscar el servicio de TimeStream, dentro de los servicios de AWS, se escoge el apartado base de datos y se crea una nueva base de datos y una nueva tabla. En la tabla se debe tomar en cuenta el tiempo de duración que se quiera guardar los datos históricos dentro de un servicio. También en este apartado se puede escoger el almacenamiento en nube para datos históricos como S3, en el caso de la tesis no ha sido utilizado este servicio como se visualiza en la Ilustración 10, 11 y 12.

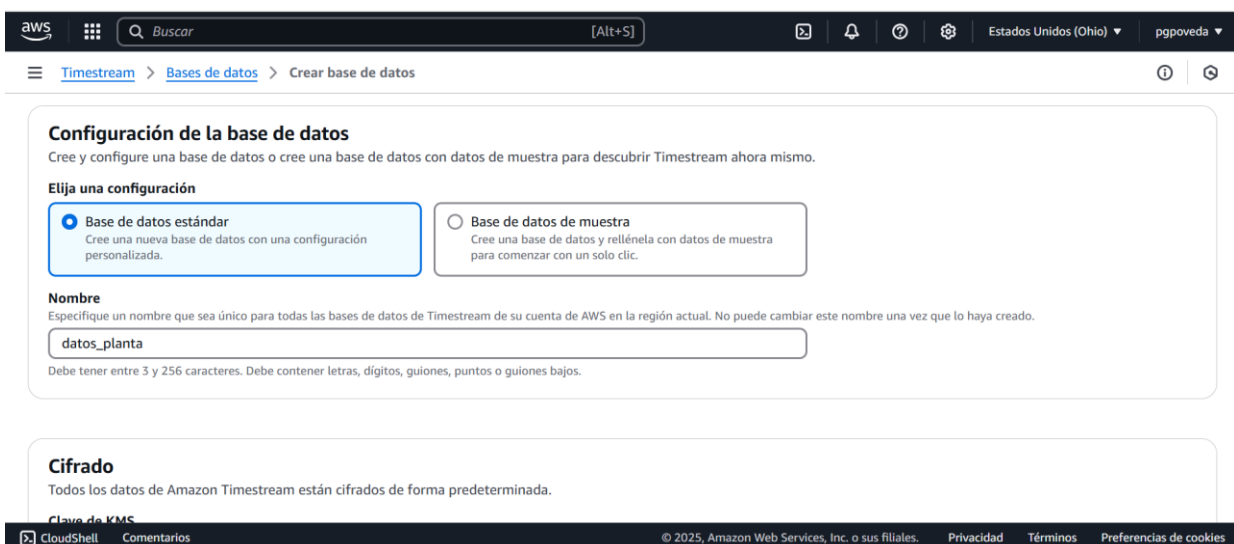


Ilustración 10. Creación de base de datos - De autoría propia

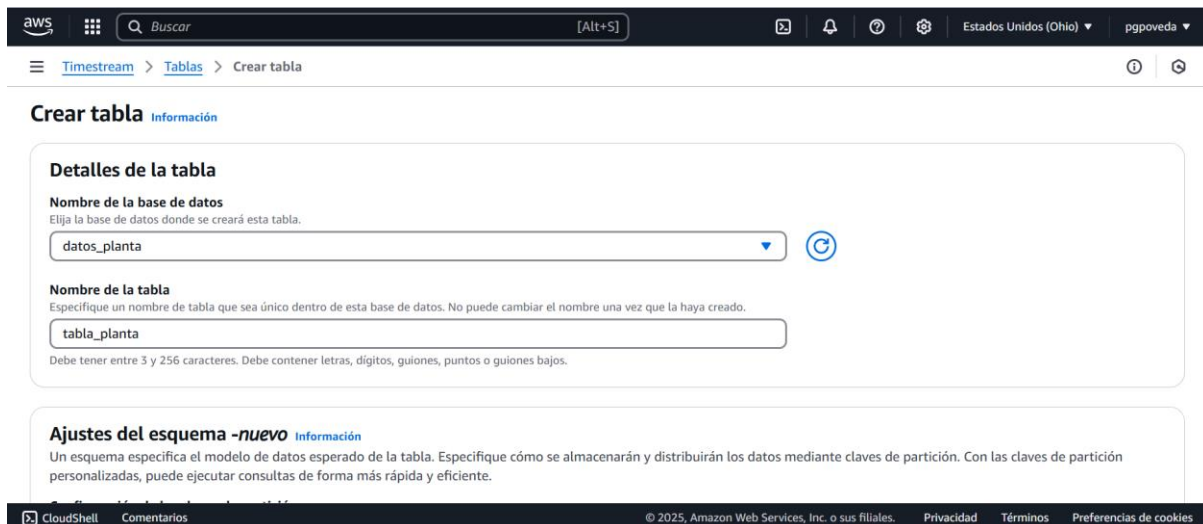


Ilustración 11. Creación de tablas de la base de datos – De autoría propia

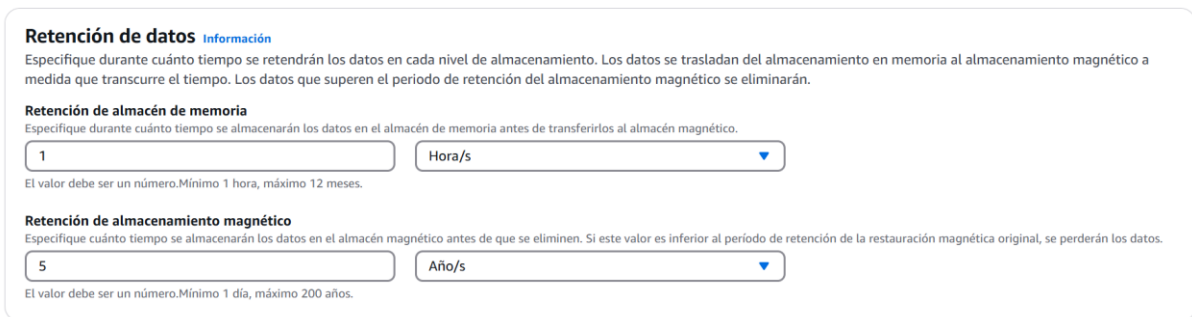


Ilustración 12. Retención de los datos históricos – De autoría propia

En la consola de IoT core se debe crear una regla de comunicación hacia el motor de base de datos TimeStream. Primero, se realiza una regla SQL para tomar los datos del suscriptor, para esto es necesario conocer el tópic en el que se encuentran los mensajes, como se visualiza en la *Ilustración 13*.

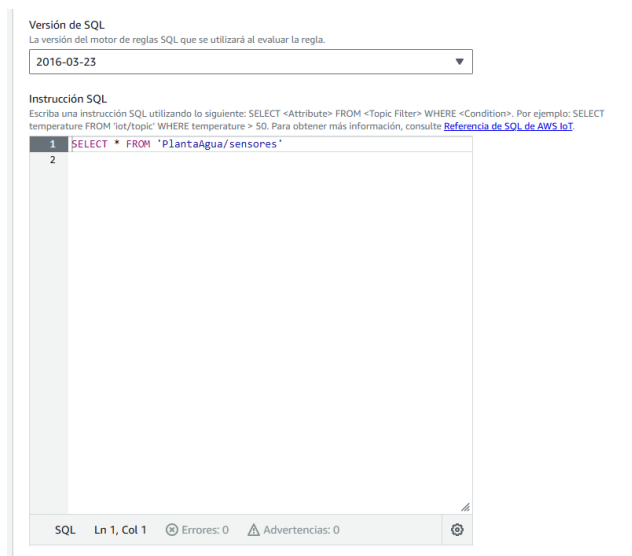


Ilustración 13. código SQL para captar mensajes del suscriptor – De autoría propia

Al continuar se encontrará los espacios para la conexión con la base de datos. Es importante tener en cuenta los campos que se requieren para los datos creando de esta manera dimensiones para cada campo. Puede ser importante crear un timestamp para saber la hora de creación de los datos y que no a llegado un dato no requerido, como se visualiza en la Ilustración 14, 15.

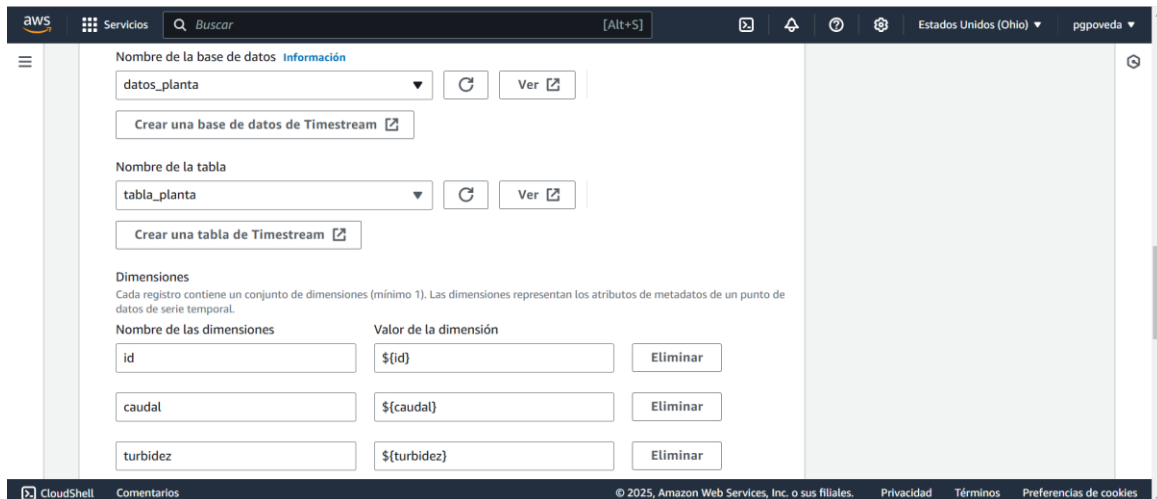


Ilustración 14. Regla para la base de datos datos_planta – De autoría propia

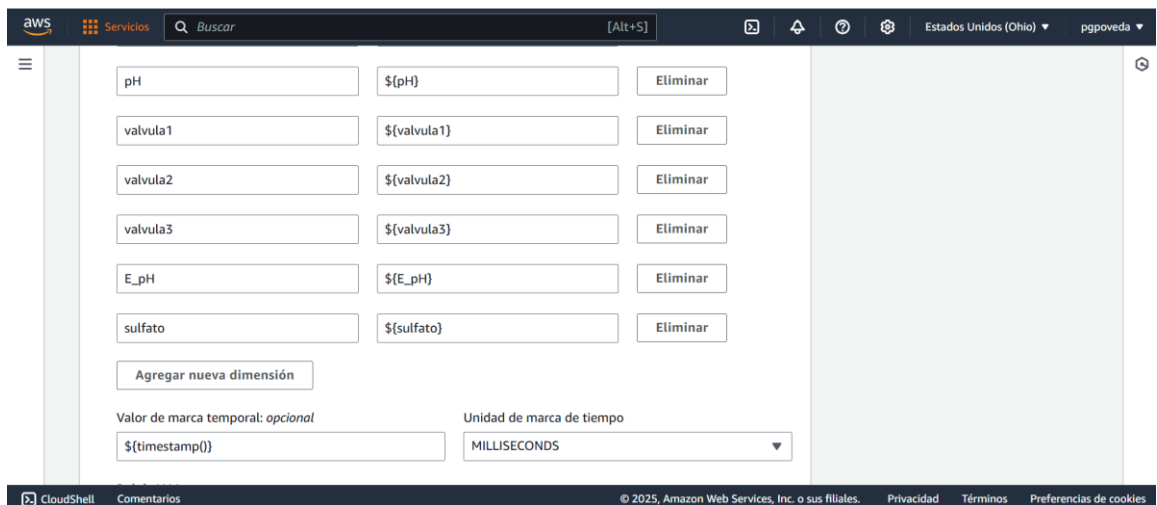


Ilustración 15. Creación de dimensiones de BD – De autoría propia

Se debe crear un usuario que tenga permisos para hacer esta acción. El usuario se puede crear en la misma pantalla de la regla y se pondrán las políticas de manera automática como se ve en la Ilustración 16.

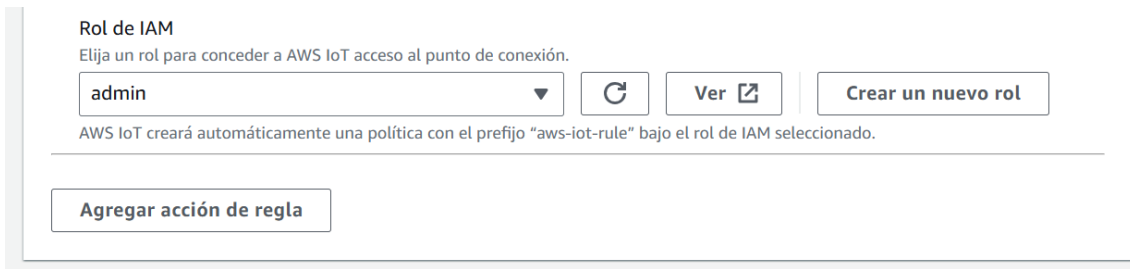


Ilustración 16. usuario para la conexión de IoT core con TimeStream – De autoría propia

Por último, se debe verificar que los datos enviados por el programa python sean iguales a los que llegan a la base de datos especificada, como se muestra en la *Ilustración 17*.

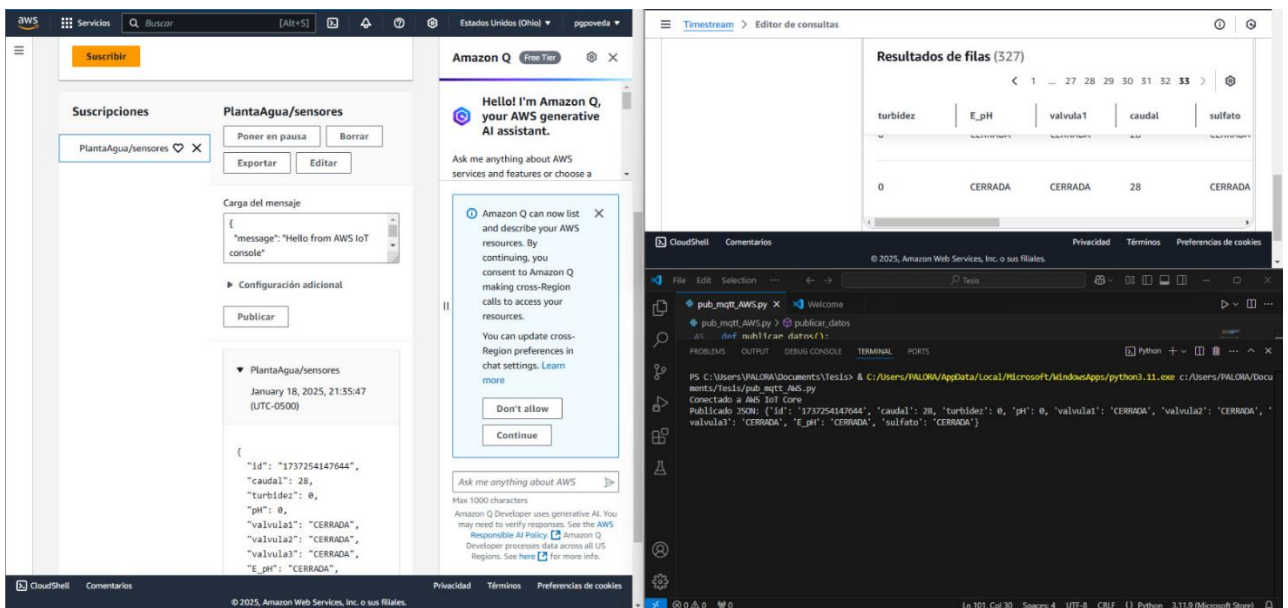


Ilustración 17. Datos guardados en la base de datos – De autoría propia

4.1.4.8. Conexión de Timestream a Grafana

Para realizar esta conexión es importante crear un usuario el cual tenga permisos y claves de creación de objetos externos, estas claves son únicas. Existe la clave de acceso que podemos verla en cualquier momento sin ningún problema, sin embargo, la clave secreta solo se podrá ver al crear el usuario. Esta clave debe ser muy bien protegida y no se debe compartir, como se muestra en la *Ilustración 18 y 19*.




Ilustración 18. Creación de usuario para la conexión con programas externos – De autoría propia

Recuperar claves de acceso Información

Clave de acceso

Si pierde u olvida la clave de acceso secreta, no podrá recuperarla. En su lugar, cree una nueva clave de acceso y deje inactiva la antigua.

Clave de acceso

 AKIAVYV5ZWMGKHOSOD7C

Clave de acceso secreta

 ***** [Mostrar](#)

Prácticas recomendadas para la clave de acceso

- Nunca almacene la clave de acceso en texto plano, en un repositorio de código o en el código.
- Desactive o elimine la clave de acceso cuando ya no sea necesaria.
- Habilite los permisos con privilegios mínimos.
- Rote con regularidad las claves de acceso.

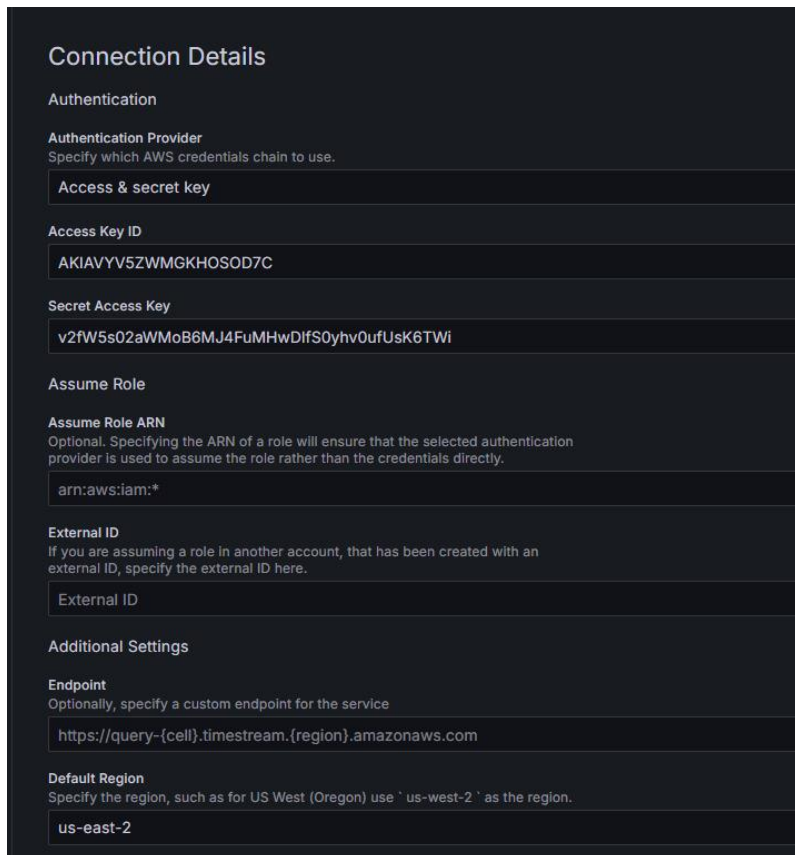
Para obtener más información sobre cómo administrar las claves de acceso, consulte las [prácticas recomendadas para administrar las claves de acceso de AWS](#).

[Descargar archivo.csv](#)

Listo

Ilustración 19. Clave de acceso y clave secreta para conexiones – De autoría propia

Después de completar los pasos de conexión en AWS, es necesario dirigirse a Grafana. En el menú principal, se debe acceder a la sección de conexiones y buscar TimeStream. Dentro de la pantalla, se completan las credenciales de conexión y los datos de la base de datos, como se muestra en la *Ilustración 20 y 21*.



Connection Details

Authentication

Authentication Provider
Specify which AWS credentials chain to use.

Access & secret key

Access Key ID
AKIAVYV5ZWMGKHOSOD7C

Secret Access Key
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

Assume Role

Assume Role ARN
Optional. Specifying the ARN of a role will ensure that the selected authentication provider is used to assume the role rather than the credentials directly.

arn:aws:iam:*

External ID
If you are assuming a role in another account, that has been created with an external ID, specify the external ID here.

External ID

Additional Settings

Endpoint
Optionally, specify a custom endpoint for the service

https://query-`{cell}`.timestream.`{region}`.amazonaws.com

Default Region
Specify the region, such as for US West (Oregon) use `us-west-2` as the region.

us-east-2

Ilustración 20. Conexión con claves de TimeStream a Grafana – De autoría propia

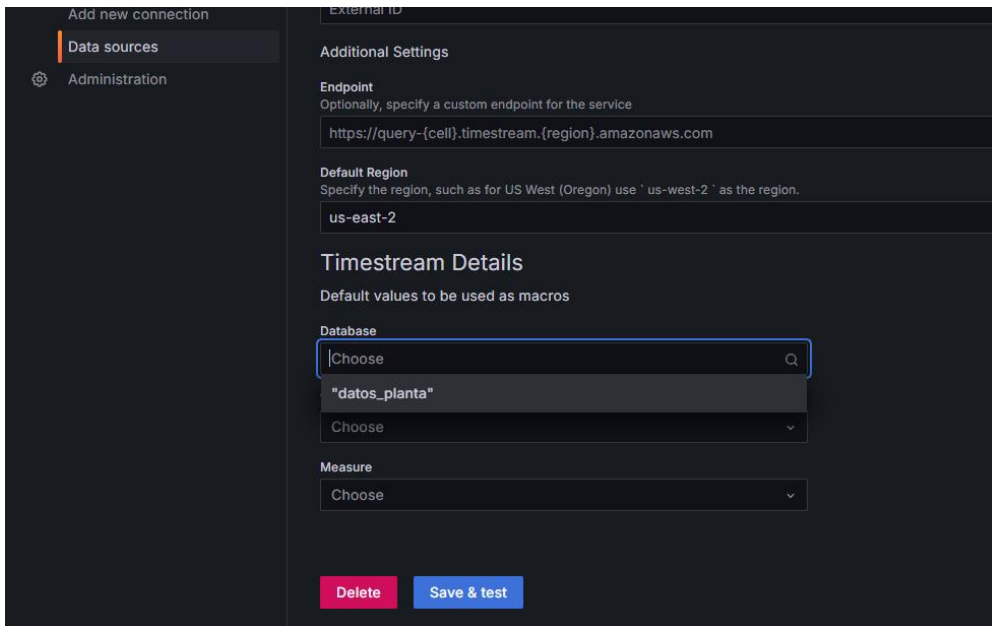


Ilustración 21. Datos de la base de datos a conectar – Capturado por el autor

Después de guardar y realizar la prueba, se debe ir a *Build Dashboard*. En el panel, es posible comenzar a construir el dashboard según los datos que se deseen mostrar, como se observa en las siguientes ilustraciones.

En la *ilustración 22 y 23* se puede observar la búsqueda a los datos del caudal y turbidez. La base de datos recibe todo el archivo JSON como carácter (varchar) es por eso que, para manejar los datos como número es importante transformarlos como se ve en las Ilustraciones mencionadas, el código SQL para transformar estos datos en este caso es: **CAST**.

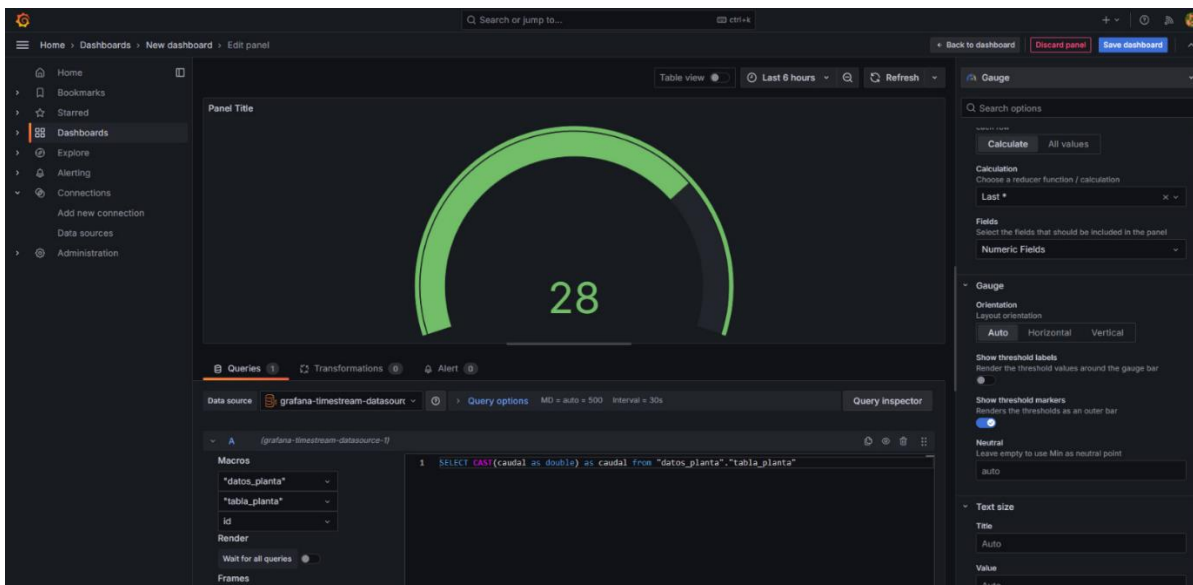


Ilustración 22. Diseño de la visualización del caudal – De autoría propia

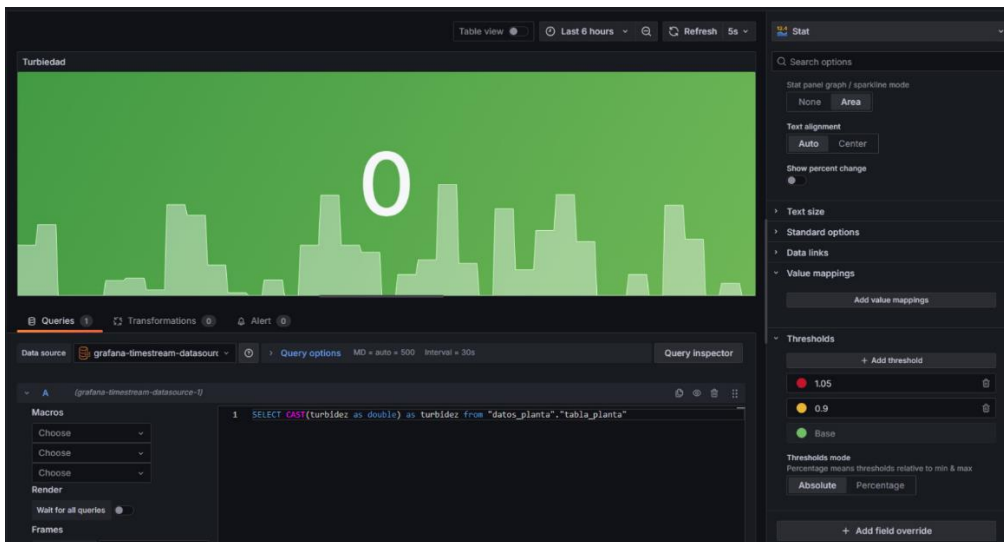


Ilustración 23. Construcción del dashboard para turbidez – De autoría propia

Por otro lado, los datos de las válvulas son dos valores, abierto o cerrado, esto si se maneja por medio de caracteres como se observa en la Ilustración 24.

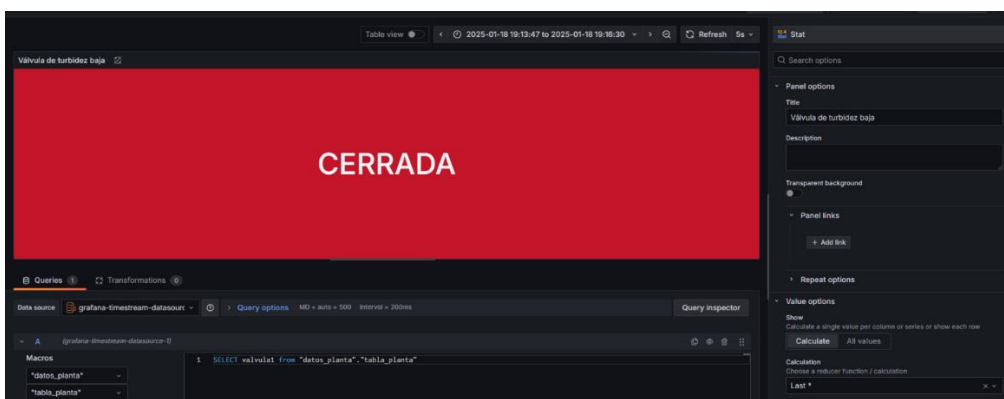


Ilustración 24. Construcción de dashboard para válvulas – De autoría propia

También, se puede hacer que todos los datos se vean según el orden de llegada y por medio de la fecha que tengan, para esto se buscan los datos por medio de la id única que se basa en el tiempo, como se observa en la Ilustración 25.

Panel Title	E_pH	valvula1	ciudad	sulfato	valvula2	valvula3	pH	id	mess
turbidez	0.44	ABIERTA	ABIERTA	25	CERRADA	CERRADA	6.36	1737242919584	id
	1.11	CERRADA	CERRADA	26	ABIERTA	ABIERTA	6.74	1737242920110	id
	1.72	ABIERTA	CERRADA	15	ABIERTA	ABIERTA	5.38	1737253841716	id
	0.92	CERRADA	ABIERTA	18	CERRADA	CERRADA	7.08	1737253845839	id
	0	CERRADA	CERRADA	28	CERRADA	CERRADA	0	1737253849902	id
	0	CERRADA	CERRADA	32	CERRADA	CERRADA	0	1737253854088	id
	0.36	ABIERTA	ABIERTA	17	CERRADA	CERRADA	6.31	1737253858213	id
	0	CERRADA	CERRADA	32	CERRADA	CERRADA	0	1737253862354	id
	1.54	ABIERTA	CERRADA	26	ABIERTA	ABIERTA	5.37	1737253878729	id
	0.98	ABIERTA	ABIERTA	23	CERRADA	CERRADA	5.41	1737253882852	id
	0	CERRADA	CERRADA	28	CERRADA	CERRADA	0	1737253717291	id
	0.21	CERRADA	ABIERTA	19	CERRADA	CERRADA	6.75	1737253737416	id
	0	CERRADA	CERRADA	28	CERRADA	CERRADA	0	1737254147644	id

Ilustración 25. Construcción de dashboard general – De autoría propia

Por último, es posible unir varias métricas en una sola pestaña del dashboard. Se han realizado dos uniones. La primera es de las métricas numéricas y la segunda de todas las métricas en función al tiempo de creación y rangos de valor como turbidez y caudal, como se muestran en las *Ilustraciones 26 y 27*.

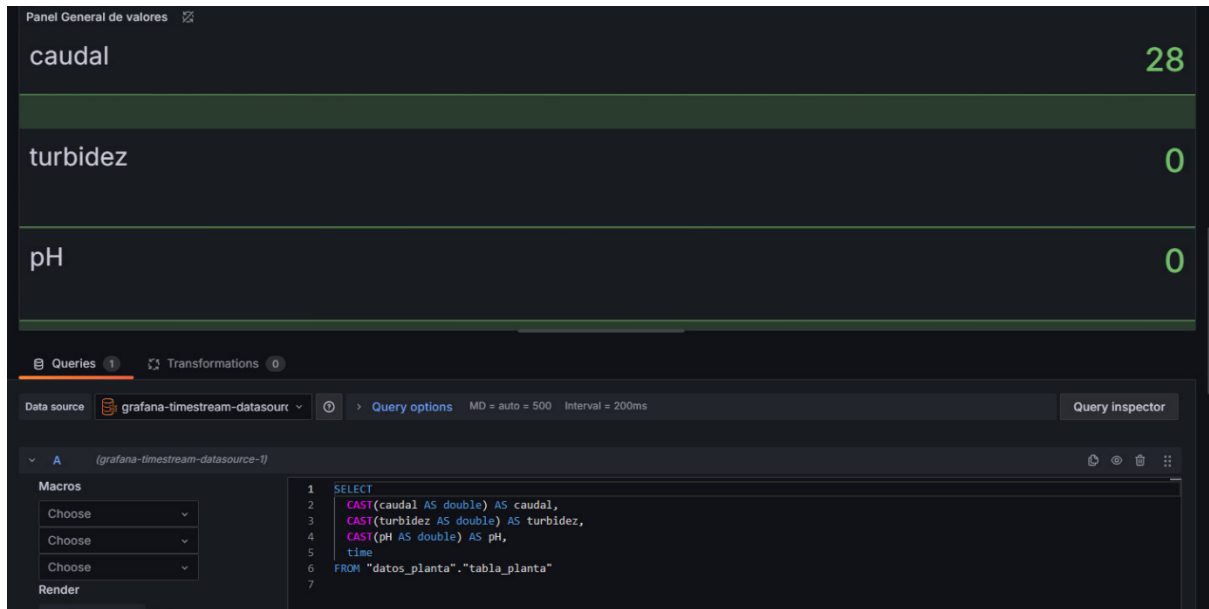


Ilustración 26. Creación de dashboard de 3 valores – De autoría propia



Ilustración 27. Creación del dashboard tiempo valor – De autoría propia

4.1.4.9. Demostración

En la documentación presentada en el diseño, se describen las conexiones y los pasos necesarios para llevar los datos desde los sensores hasta el dashboard del cliente final. Este proceso se ha realizado utilizando diversos equipos de hardware y software interconectados. En este apartado, se explicarán y documentarán las pruebas realizadas, así como la demostración de estas conexiones, incluyendo la validación de la simulación de los sensores.

Se accede a la plataforma AWS y se navega al apartado de IoT Core. En una pestaña separada, se abre la base de datos para realizar la primera validación. Este proceso consiste en enviar datos desde el programa publicador hacia el software suscriptor, verificando que la base de datos reciba y registre los mismos datos correctamente, como se observa en la *Ilustración 28*.

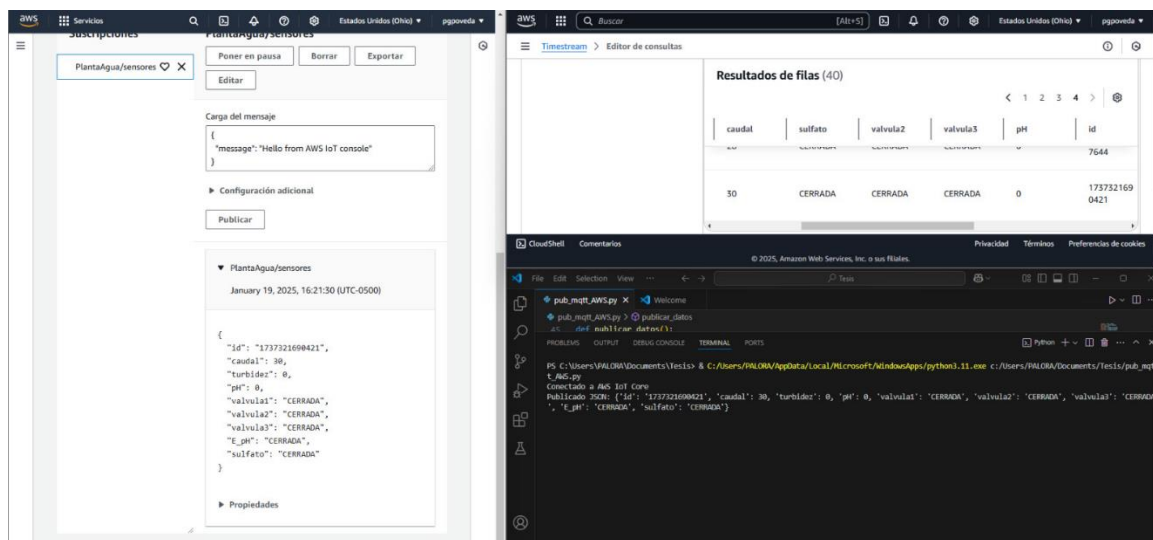


Ilustración 28. Conexión de datos entre AWS IoT y TimeStream – De autoría propia

Para concluir las pruebas de conexión y validación, se enviarán nuevamente datos al servicio de publicación, permitiendo la visualización de las gráficas en Grafana. Además, será posible observar el valor enviado previamente en las tablas de datos generales y en las tablas de tiempo creadas en Grafana, como se muestra en las *ilustraciones 29, 30, 31 y 32*.

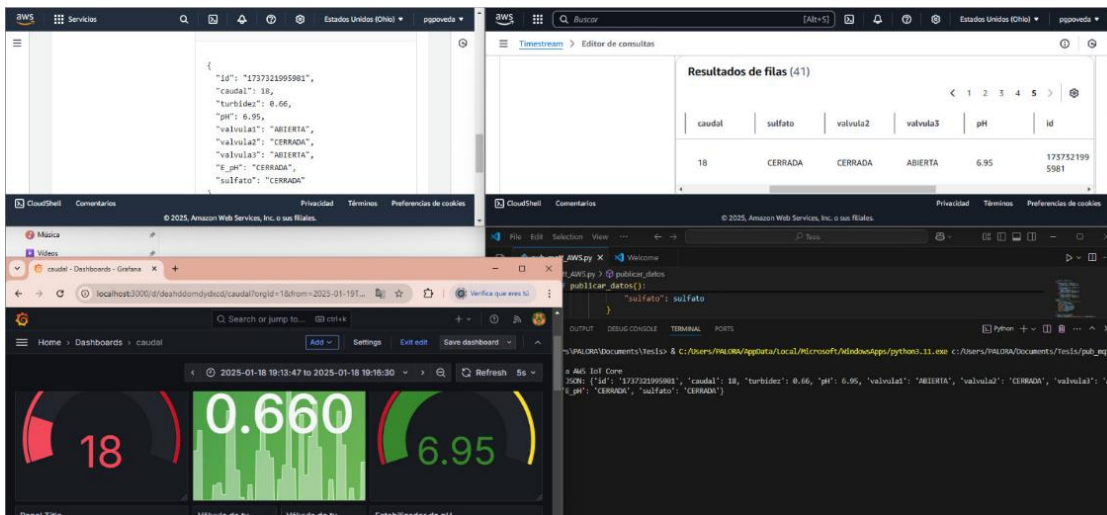


Ilustración 29. Conexión total entre servicios – De autoría propia



Ilustración 30. Dashboard con todas las pestañas de monitoreo y control – De autoría propia

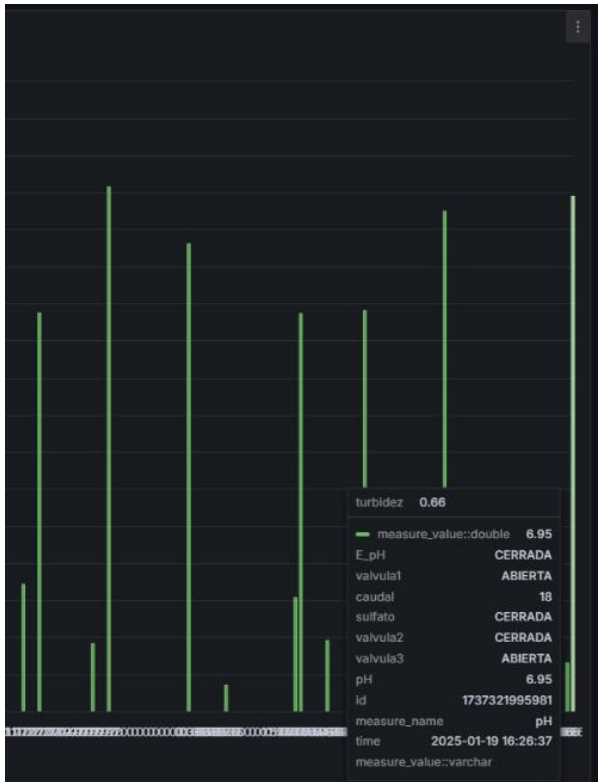


Ilustración 31. Grafica tiempo valor – De autoría propia

Valore Totales										
turbidez	E_pH	valvula1	caudal	sulfato	valvula2	valvula3	pH	id	measure_name	time
0	CERRADA	CERRADA	30	CERRADA	CERRADA	CERRADA	0	1737321690421	id	2025-01-19 16:21:31
0.66	CERRADA	ABIERTA	18	CERRADA	CERRADA	ABIERTA	6.95	1737321995981	id	2025-01-19 16:26:37

Ilustración 32. Tabla general de datos, últimos valores capturados – De autoría propia

4.1.5. Evaluación

El diseño del sistema de automatización para el monitoreo y control de una planta de tratamiento de agua potable ha pasado por las etapas de identificación del problema, definición de objetivos, diseño y pruebas, obteniendo los resultados esperados. Esto confirma que se han cumplido los objetivos planteados.

En primer lugar, se diseñó un sistema de gestión y control basado en dispositivos IoT para el monitoreo y automatización de plantas de tratamiento de agua potable. Este objetivo se alcanzó mediante el uso de software especializado en conexiones IoT y la simulación de dispositivos, realizando pruebas que validaron el diseño propuesto en el proyecto de titulación.

Además, se logró la integración de dispositivos IoT para permitir la comunicación y transmisión de datos entre los sensores. Esto se consiguió utilizando protocolos de mensajería ligeros, que requieren un bajo consumo de ancho de banda. El microcontrolador propuesto en el diseño cuenta con tecnología WiFi, al igual que la planta de tratamiento, lo que facilita una comunicación eficiente tanto entre los sensores como entre el controlador y el bróker.

Como resultado final del proyecto, se optimizó el monitoreo del sistema, eliminando la necesidad de supervisión humana constante en la obra civil. Gracias a la integración con el analizador Grafana, es posible visualizar los datos emitidos por la planta, permitiendo una intervención temprana ante posibles incidentes y mejorando la gestión de los tiempos operativos.

5. Conclusiones y Recomendaciones

5.1. Conclusiones

El trabajo de titulación se enfocó en el diseño e implementación de un sistema de monitoreo y gestión automática para una planta de tratamiento de agua potable, utilizando dispositivos IoT. A través de simulaciones, se demostró la funcionalidad y eficiencia del diseño propuesto, validando los aspectos clave de la automatización.

Se logró la integración exitosa de los dispositivos IoT, lo que permitió la comunicación y transmisión de datos entre los sensores de la planta. Este aspecto es fundamental para el control del sistema, ya que, al tratarse de un circuito de lazo cerrado, el funcionamiento del sistema depende de dicha integración.

Se optimizó la gestión y monitoreo de los recursos hídricos priorizando la simplicidad en la accesibilidad del sistema para facilitar su uso por los operadores, integrando herramientas y plataformas que optimizan la visualización y el análisis de datos.

Así mismo, se simplificó las comunicaciones entre dispositivos, empleando protocolos ligeros que minimizan el consumo de ancho de banda sin sacrificar la eficacia en el monitoreo y control. A través de estos avances, se logró cumplir con todos los objetivos establecidos en el proyecto, demostrando que el diseño no solo es viable, sino también eficiente y seguro.

5.2. Recomendaciones

El proyecto desarrollado se enfoca en el diseño de un sistema de monitoreo y gestión automatizada para una planta de tratamiento de agua potable, utilizando dispositivos IoT. Si se desea implementar este sistema, es necesario considerar varios aspectos clave en la construcción de la infraestructura tecnológica. En primer lugar, se deben establecer equipos de borde (edge devices) capaces de procesar y soportar la recepción de los datos generados por los sensores IoT. Además, es fundamental contar con una red dedicada para garantizar que los datos se transmitan de manera eficiente y segura, evitando problemas relacionados con la congestión de red o posibles brechas de seguridad.

Si bien AWS IoT ofrece certificados TLS y SSL que garantizan una conexión segura, una red privada virtual (VPN) sería una opción recomendable para asegurar una capa adicional de privacidad en las conexiones y transmisiones de datos. La implementación de una VPN

garantizaría que toda la infraestructura de comunicación sea completamente privada, añadiendo un nivel de protección adicional.

En cuanto a la gestión y análisis de datos, el sistema podría incluir funcionalidades para realizar análisis avanzado de los datos extraídos por los sensores. Esto permitiría la visualización de los resultados tanto de las actuaciones en la construcción civil de la planta como del desempeño del propio sistema IoT. Estos análisis ayudarían a optimizar las operaciones y a realizar intervenciones preventivas basadas en patrones de comportamiento del sistema.

6. BIBLIOGRAFÍA

- ALCORA. (27 de Octubre de 2021). *Especialistas en*. Obtenido de Agua potable y agua tratada: <https://alcora.es/blog/agua-potable-y-agua-tratada-que-son-caracteristicas-y-diferencias/#:~:text=Qu%C3%A9%20caracter%C3%ADsticas%20debe%20tener%20el%20agua%20potable,-El%20agua%20debe&text=Incolora%2C%20el%20agua%20debe%20ser,en%20suspensi%C3%B3n%20que%2>
- ARCOTEL. (2023). *Cuadro Estadístico 2020*. Quito: Arcotel.
- Arduino. (22 de Diciembre de 2022). *Proyectos con Arduino*. Obtenido de Proyectos con Arduino: <https://proyectosconarduino.com/sensores/sensores-de-ph-para-arduino/>
- AWS. (2020). *AWS*. Obtenido de AWS IoT Core: https://docs.aws.amazon.com/es_es/iot/latest/developerguide/what-is-aws-iot.html
- AWS. (2020). *Qué es TimeStream*. Obtenido de aws: <https://docs.aws.amazon.com/timestream/latest/developerguide/what-is-timestream.html>
- AWS. (2021). *IOT*. Obtenido de IOT: <https://aws.amazon.com/es/what-is/iot/>
- AWS. (2022). *AWS*. Obtenido de ¿QUE ES MQTT?: <https://aws.amazon.com/es/what-is/mqtt/>
- AWS (Dirección). (2023). *AWS IoT - Device data to dashboard in 10 minutes - A demonstration* [Película]. Obtenido de <https://www.youtube.com/watch?v=z8T4hAERuOg>
- B Priya, B. P. (2024). *Applications of Artificial Intelligent and Internet of Things in Electric Vehicle*. Obtenido de Applications of Artificial Intelligent and Internet of Things in Electric Vehicle: <https://www.ijert.org/research/applications-of-artificial-intelligent-and-internet-of-things-in-electric-vehicle-battery-management-system-IJERTCONV12IS01005.pdf>
- BBVA. (04 de septiembre de 2024). *BBVA*. Obtenido de ¿Qué es el proceso de potabilización del agua y cuáles son sus fases?: <https://www.bbva.com/es/sostenibilidad/que-es-el-proceso-de-potabilizacion-del-agua-y-cuales-son-sus-fases/>
- BBVA. (04 de Septiembre de 2024). *Tratamiento de agua*. Obtenido de Tratamiento de agua: <https://www.bbva.com/es/sostenibilidad/que-es-el-proceso-de-potabilizacion-del-agua-y-cuales-son-sus-fases/>
- Campos, E. S. (2020). *Sistema IoT para el análisis de calidad de aguas*. Madrid.
- Cotecno. (12 de Noviembre de 2024). *Cotecno*. Obtenido de Cotecno: <https://www.cotecno.cl/sensor-de-turbidez/>
- DISTRITEC. (2022). *¿Qué es una electroválvula y para qué sirve?* Obtenido de ¿Qué es una electroválvula y para qué sirve?: <https://www.distritec.com.ar/que-es-una-electrovalvula-y-para-que-sirve/>
- Educa Madrid. (SF). *Sistema de control de lazo abierto*. Obtenido de Sistema de control de lazo abierto: <https://www.educa2.madrid.org/web/4656105/robotica/-/book/sistemas-de>

control-lazo-abierto-lazo-cerrado?_book_viewer_WAR_cms_tools_chapterIndex=42e420b2-3c94-4431-b6b0-bdc01391e2c8

- Electronics. (13 de Noviembre de 2024). *Sensor de flujo de agua*. Obtenido de Sensor de flujo de agua: <https://avelectronics.cc/producto/sensor-de-flujo-de-agua-yf-s201/?srsltid=AfmBOorXRokwlvStB2FsmYCYZmguhdKH6aaN9ROnwE-BxwkZftxDdo3tJ>
- Gabriela, A. (13 de Enero de 2023). *Comunicaciones Inalambricas*. Obtenido de Modulos: <https://www.comunicacionesinalambricashoy.com/modulo-inalambrico-esp32-c6/>
- Galvín, R. M. (2008). CARACTERÍSTICAS FÍSICAS, QUÍMICAS Y BIOLÓGICAS DE LAS AGUAS. En R. M. Galvín, *CARACTERÍSTICAS FÍSICAS, QUÍMICAS Y BIOLÓGICAS DE LAS AGUAS* (págs. 3-5). Cordoba.
- Garzón, W. F. (2019). ¿Qué es IOT? En W. F. Garzón, & J. C. López, *Tecnología internet of things (IoT) y el big data* (pág. 3). Colombia.
- Guevara, L. V. (2018). *Makinando*. Obtenido de El comparador dentro de un sistema de control de lazo cerrado: <https://makinandovelez.wordpress.com/2018/02/08/el-comparador-dentro-de-un-sistema-de-control/>
- Hernández, A. (2014). *TallerElectronica.com / Blog*. Obtenido de Introducción a la Electrónica Digital: <https://tallerelectronica.com/2014/10/13/introduccion-a-la-electronica-digital/>
- IBM. (4 de Junio de 2024). *IBM*. (IBM, Editor) Obtenido de ¿Qué es un microcontrolador?: <https://www.ibm.com/es-es/think/topics/microcontroller>
- Instituto Aprende. (2020). *Instituto Aprende*. Obtenido de Circuitos electronicos digitales: <https://aprende.com/blog/oficios/reparacion-electronica/circuito-electronico-digital-definicion-tipos-y-dibujos/>
- Instituto del Agua. (10 de Febrero de 2024). *Instituto del Agua*. Obtenido de Instituto del Agua: <https://institutodelagua.es/calidad-del-agua/proceso-de-potabilizacion-del-agua-filtracionfiltracion-de-agua/>
- Kemira Raw water and drink. (28 de mayo de 2024). *Agua cruda y agua potable*. Obtenido de Agua cruda y agua potable: <https://www.kemira.com/water/raw-water-drinking-water/>
- Lawrence, C. (2010). *Design science research*.
- Lorenzo, Y. (2006). *Estado del arte del tratamiento de aguas por coagulación-floculación*. La Habana, Cuba: Instituto Cubano de Investigaciones de los. Recuperado el 2 de Diciembre de 2024, de <https://www.redalyc.org/pdf/2231/223120664002.pdf>
- Manrique, A., & Gomez, J. (2016). *DISEÑO Y CONSTRUCCIÓN DE UNA TORRE DE SEDIMENTACIÓN*. Bogotá, Colombia: Universidad Católica de Colombia. Recuperado el 2 de Diciembre de 2024, de <https://repository.ucatolica.edu.co/server/api/core/bitstreams/2b8190b5-503c-49b0-9c38-991acbb69325/content>
- Martinez, J. (2021). *Qué es Grafana y primeros pasos*. Obtenido de Qué es Grafana y primeros pasos: <https://openwebinars.net/blog/que-es-grafana-y-primeros-pasos/>
- MQTT. (2024). *MQTT*. Obtenido de Especifications: <https://mqtt.org/>

- Orellana. (2023). *Orellana*. Obtenido de Módulo Nodemcu:
<https://orellanaelec.com/producto/modulo-nodemcu-bluetooth-wifi-esp32-wroom32/>
- Organización Mundial de la Salud. (2023). *Agua y salud*. Obtenido de Agua y salud:
<https://www.who.int/es/news-room/fact-sheets/detail/drinking-water>
- OSE. (2024). *OSE*. Obtenido de Etapas del Proceso de Potabilización:
<http://www.ose.com.uy/agua/etapas-del-proceso-de-potabilizacion#:~:text=La%20filtraci%C3%B3n%20consiste%20en%20pasar,las%20garant%C3%ADas%20de%20que%20el>
- Reyes, K. (2022). *Investigación en las ciencias del diseño*. Chiclayo: Manfred Acero Gómez.
- Robodacta. (2025). *Bomba De Aire Vacío*. Obtenido de Bomba De Aire Vacío :
<https://robodacta.com.mx/producto/mini-bomba-de-aire-vacio-succion-de-diafragma-12v/>
- Rosero, R. (2015). *SCADA DEL SISTEMA DE DISTRIBUCIÓN DE AGUA DE LA EMPRESA PÚBLICA*. Quito.
- Valdivielso, A. (11 de Septiembre de 2020). *iagua*. Obtenido de ¿Qué son los manantiales de agua?:
<https://www.iagua.es/respuestas/que-son-manantiales-agua>
- W&T. (2025). *Wiesemann & Theis GmbH*. Obtenido de MQTT: comunicación en el Internet de las Cosas: <http://u-blox.com/en/blogs/insights/mqtt-beginners-guide>
- Yépez, L., & Ruiz, N. (2005). *ESTUDIO DE IMPACTO AMBIENTAL DE LA EJECUCIÓN DEL PROYECTO DE AGUA PARA RIEGO Y CONSUMO HUMANO DEL CANTÓN PIMAMPIRO, PRIMERA FASE*. Ecuador: Universidad Técnica del Norte.
- Digital Analysis Corp. (2019). *Phadjustment.com*. pH Adjustment - A Primer -
https://doi.org/1071901437/ZjuJCKzB46IBEP3Vj_8D

7. Anexos

7.1. Programa en Arduino ID para sensores

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DFRobot_PH.h>

// Configuración de Wi-Fi
const char* ssid = "Tu_SSID";
const char* password = "Tu_PASSWORD";

// Configuración de MQTT (AWS IoT Core)
const char* mqtt_server = "d04482212v9h1iqvjp2tj-ats.iot.us-east-1.amazonaws.com";
const int mqtt_port = 8883;
const char* mqtt_user = "";
const char* mqtt_pass = "";
const char* mqtt_topic = "PlantaAgua/sensores";

// Certificados para la conexión segura (AWS IoT)
const char* certificate = "-----BEGIN CERTIFICATE-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA...";
const char* private_key = "-----BEGIN PRIVATE KEY-----
\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCL9zqPbXXkKLYg...
";
const char* ca_cert = "-----BEGIN CERTIFICATE-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA...";

// Definición de pines y sensores
#define DHTPIN 4 // Pin donde está conectado el sensor DHT
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// Sensor de pH
DFRobot_PH ph_sensor;
int pin_ph = 34; // Pin de entrada analógica para el sensor de pH

// Sensor de caudal (YF-S201)
#define FLOW_SENSOR_PIN 12
volatile int flow_count = 0; // Contador de pulsos del sensor de flujo
float caudal_value = 0.0; // Caudal en litros por minuto

// Instanciamos el cliente WiFi y MQTT
WiFiClientSecure espClient;
PubSubClient client(espClient);
```

```

// Variables de estado
unsigned long lastFlowTime = 0;
unsigned long lastPublishTime = 0;

// Función para conectarse a Wi-Fi
void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Conectando a Wi-Fi: ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }

  Serial.println("Conectado a Wi-Fi");
}

// Función de reconexión MQTT
void reconnect() {
  while (!client.connected()) {
    Serial.print("Conectando a MQTT...");
    if (client.connect("ESP32_P")) {
      Serial.println("Conectado a MQTT");
    } else {
      Serial.print("Error: ");
      Serial.print(client.state());
      delay(5000);
    }
  }
}

// Generación de un ID único basado en timestamp
String generar_clave_con_timestamp() {
  return String(millis()); // ID único basado en el tiempo en milisegundos
}

// Función para leer el sensor de turbidez
float turbidez() {
  int valor_analogico = analogRead(A0); // Lectura analógica del sensor de turbidez
  // Conversión del valor analógico a una escala representativa de turbidez
  return map(valor_analogico, 0, 1023, 0, 2); // Ejemplo de mapeo de valores
}

// Función para leer el sensor de pH

```

```

float pH() {
    int valor_analogico = analogRead(pin_ph); // Lectura analógica del sensor de pH
    float voltage = valor_analogico * (5.0 / 1023.0); // Convertir valor a voltaje
    return ph_sensor.readPH(voltage); // Conversión a valor de pH
}

void flowSensor_ISR() {
    flow_count++; // Incrementar el contador por cada pulso
}

float caudal() {
    unsigned long currentTime = millis();

    if (currentTime - lastFlowTime >= 1000) {
        float flow_rate = flow_count; // Número de pulsos por segundo
        flow_count = 0; // Reiniciar contador
        lastFlowTime = currentTime;

        // Conversión de pulsos por segundo a litros por minuto (depende del sensor)
        caudal_value = flow_rate * 0.0023 * 60.0; // 0.0023 es un factor de calibración (ajustar según tu
sensor)
    }

    return caudal_value;
}

// Función para publicar datos
void publicar_datos() {
    while (true) {
        // Obtener los valores de los sensores
        float valor_caudal = caudal();
        float valor_turbidez = turbidez();
        float valor_ph = pH();
        String secuencial = generar_clave_con_timestamp();

        String estado_valvula1 = (valor_caudal > 27) ? "CERRADA" : "ABIERTA";
        String estado_valvula2 = (valor_turbidez > 1) ? "ABIERTA" : "CERRADA";
        String estado_valvula3 = (valor_ph < 6.5 || valor_ph > 7.5) ? "ABIERTA" : "CERRADA";
        String estabilizador_ph = (valor_ph < 6.5 || valor_ph > 7.5) ? "ACTIVADO" :
"DESACTIVADO";
        String sulfato = (valor_turbidez > 1) ? "ACTIVADO" : "DESACTIVADO";

        // Construir el mensaje JSON
        StaticJsonDocument<200> doc;
        doc["id"] = secuencial;
        doc["caudal"] = valor_caudal;
        doc["turbidez"] = valor_turbidez;
        doc["pH"] = valor_ph;
    }
}

```

```

doc["valvula1"] = estado_valvula1;
doc["valvula2"] = estado_valvula2;
doc["valvula3"] = estado_valvula3;
doc["E_pH"] = estabilizador_ph;
doc["sulfato"] = sulfato;

// Publicar el mensaje JSON en el t3pico MQTT
char output[256];
serializeJson(doc, output);
client.publish(mqtt_topic, output);

Serial.println("Publicado JSON: ");
Serial.println(output);

delay(7200000); // Publicar cada 2 horas
}
}

void setup() {
// Inicializaci3n serial
Serial.begin(115200);

// Conectar a Wi-Fi
setup_wifi();

// Inicializar los sensores
dht.begin();
ph_sensor.begin(pin_ph);

// Configurar interrupci3n para el sensor de caudal
pinMode(FLOW_SENSOR_PIN, INPUT);
attachInterrupt(digitalPinToInterrupt(FLOW_SENSOR_PIN), flowSensor_ISR, FALLING);

// Configuraci3n de certificados para conexi3n segura
espClient.setCACert(ca_cert);
espClient.setCertificate(certificate);
espClient.setPrivateKey(private_key);

// Configurar el servidor MQTT
client.setServer(mqtt_server, mqtt_port);
client.setCallback(reconnect);

// Conectar a AWS IoT Core
reconnect();

// Ejecutar la publicaci3n de datos
publicar_datos();
}

```

```
void loop() {  
  if (!client.connected()) {  
    reconnect();  
  }  
  client.loop();  
}
```