

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **IMPLEMENTACIÓN DE UN PROTOTIPO PARA MONITOREO REMOTO DE CALIDAD DE AGUA Y VISUALIZACIÓN DE DATOS MEDIANTE UN DASHBOARD WEB**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR  
EN REDES Y TELECOMUNICACIONES**

**GARY ALEXANDER MORALES MORENO**

**DIRECTOR: ANDRES FERNANDO REYES**

**DMQ, AGOSTO 2024**

## **CERTIFICACIONES**

Yo, GARY ALEXANDER MORALES MORENO declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

---

**GARY ALEXANDER MORALES MORENO**

**gary.morales @epn.edu.ec**

**moralesgary100920@gmail.com**

Certifico que el presente trabajo de integración curricular fue desarrollado por Gary Alexander Morales Moreno, bajo mi supervisión.

---

**ANDRES FERNANDO REYES**

**andres.reyes@epn.edu.ec**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

GARY ALEXANDER MORALES MORENO

## DEDICATORIA

"Dedico este trabajo aquellos que me han ofrecido su respaldo inquebrantable durante este trayecto. A mis progenitores, por su cariño, paciencia y continuo estímulo. A mis profesores, por su sabiduría y guía en mi educación académica. A todos mis amigos, por la compañía y motivación que demostraron conmigo. Agradezco a todos aquellos que creyeron en mí y me impulsaron a dar lo mejor de mí mismo. Este logro no habría sido posible sin su inquebrantable confianza y respaldo. Que este trabajo sea un pequeño tributo a su confianza en mí y un reflejo de mi gratitud eterna hacia ellos. ¡Gracias!"

s padres, por su amor, paciencia y constante aliento. A mis profesores, por su sabiduría y guía en mi educación académica. A mis amigos, por su compañía y motivación en cada momento. Agradezco a todos aquellos que creyeron en mí y me impulsaron a dar lo mejor de mí mismo. Este logro no habría sido posible sin su inquebrantable confianza y respaldo. Que este trabajo sea un pequeño tributo a su confianza en mí y un reflejo de mi gratitud eterna hacia ellos. ¡Gracias!"

## AGRADECIMIENTO

Deseo aprovechar este momento para expresar mi profundo agradecimiento a todas las personas que han contribuido significativamente a la realización de esta tesis.

En particular, quiero agradecer a mi supervisor, Andrés Fernando Reyes, por su inestimable orientación, su constante apoyo y su dedicación durante todo el proceso de investigación. Sus conocimientos y consejos fueron esenciales para el desarrollo de este trabajo.

También deseo agradecer a la escuela de formación de tecnólogos ESFOT, por proporcionarme los medios y el entorno adecuado para realizar esta investigación. Su respaldo institucional a sido vital para el logro de este proyecto.

Agradezco a mis profesores y mentores, quienes me impartieron conocimientos valiosos y me inspiraron a ir más allá de mis límites. Sus enseñanzas y ejemplos han dejado una huella duradera en mi formación académica y personal.

No puedo dejar de reconocer el respaldo y la comprensión de mi familia y amigos durante este arduo proceso. Su aliento, paciencia y amor incondicional fueron mi fortaleza en los momentos desafiantes. Agradezco su constante motivación y por creer en mí cuando más lo necesitaba.

Culminando, deseo expresar mi agradecimiento a cada una de las fuentes de investigación, instituciones e individuos que amablemente compartieron sus conocimientos y datos. Sin su contribución, este proyecto no habría sido factible.

Este logro es el resultado del esfuerzo colectivo de muchas personas, y estoy sinceramente agradecido a cada una de ellas. Espero que este trabajo pueda servir como una pequeña contribución a la colectividad académica, así como a la sociedad en su totalidad.

¡Gracias a todos por su apoyo, aliento y contribuciones

# ÍNDICE DE CONTENIDOS

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDOS .....	V
RESUMEN.....	VII
ABSTRACT.....	VIII
1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1. Objetivo general.....	2
1.2. Objetivos específicos .....	2
1.3. Alcance.....	2
1.4. Marco Teórico.....	2
Internet de las Cosas (IoT).....	2
Arquitectura del IoT.....	3
IoT Aplicada al Monitoreo en Tiempo Real.....	4
Microcontroladores.....	4
Arduino .....	6
Calidad del Agua.....	6
Potencial Hidrogeno (pH) .....	7
Turbiedad.....	7
Solidos Totales .....	8
Importancia del Monitoreo de la Calidad del Agua.....	8
Normativas y estándares internacionales para la calidad del agua .....	8
2 METODOLOGÍA .....	9
3 RESULTADOS.....	10
3.1 Establecimiento de los requerimientos del prototipo.....	10
3.2 Definición de los Componentes de Hardware y Software .....	11

Microcontrolador y Comunicación Inalámbrica.....	11
Sensor de Turbidez.....	12
Sensor de pH.....	15
Plataforma IoT para visualización de datos y notificaciones .....	17
3.3 Diseño del Prototipo.....	18
Comunicación Entre ESP32 y Sensores .....	18
Circuito Electrónico .....	20
Diagrama de Flujo.....	21
3.4 Implementación del Prototipo .....	24
Diseño de la Placa PCB.....	24
Configuración de la Plataforma IoT .....	26
Código de Programación para el Funcionamiento del Prototipo .....	28
Colocación de los Elementos del Prototipo de Monitoreo de Agua .....	32
3.5 Pruebas de Funcionamiento.....	34
Verificación de los Sensores .....	34
Visualización de Datos en la Arduino Cloud .....	35
Visualización en aplicación móvil .....	35
4 CONCLUSIONES .....	39
5 RECOMENDACIONES .....	39
6 REFERENCIAS BIBLIOGRÁFICAS.....	40
ANEXOS.....	44
ANEXO I: Certificado de Originalidad .....	i
ANEXO II: Enlaces.....	ii
ANEXO III: Código Fuente .....	iii

## RESUMEN

La actual investigación de monitoreo de calidad de agua utilizando un ESP32 junto con dispositivos de detección de pH y turbidez, integrado con un *dashboard* en Arduino Cloud, ofrece una solución accesible para la supervisión continua de la calidad del agua. La destreza de procesamiento avanzada del ESP32, combinada con la precisión de los sensores seleccionados, permite la recolección y la evaluación de información en tiempo real, facilitando vigilancia eficiente y efectiva de los parámetros críticos del agua. Esta capacidad de monitoreo en tiempo real es crucial para la detección temprana de contaminantes y para la respuesta rápida ante cualquier alteración en la calidad del agua, la cual es esencial en la protección de la salud pública y el medio ambiente.

La integración con Arduino Cloud ofrece una interfaz sencilla e intuitiva que facilita a los usuarios el acceso a la información desde cualquier lugar con conexión a Internet. El *dashboard* en la nube no solo facilita la visualización clara y accesible de los datos recopilados, sino que también permite la generación de gráficos detallados y el envío de notificaciones automáticas cuando los parámetros medidos superan los límites establecidos, esto proporciona una seguridad y control adicionales.

Además, El proyecto es perfecto para una amplia gama de usos, desde la supervisión de cuerpos de agua naturales hasta el control de calidad en sistemas de provisión de agua potable o acuarios. La flexibilidad del ESP32 y la plataforma Arduino Cloud permiten la personalización del sistema según las necesidades específicas, y su potencial de escalabilidad ofrece la posibilidad de integrar sensores adicionales y nuevas funcionalidades posteriormente.

**PALABRAS CLAVE:** Monitoreo de calidad de agua, ESP32, Sensores de pH, Sensores de turbidez, Arduino Cloud, *Dashboard*.



## ABSTRACT

The current research on water quality monitoring using an ESP32 along with pH and turbidity detection devices, integrated with a dashboard in Arduino Cloud, offers an accessible solution for continuous water quality supervision. The advanced processing capability of the ESP32, combined with the precision of the selected sensors, allows for the real-time collection and evaluation of information, facilitating efficient and effective monitoring of critical water parameters. This real-time monitoring capability is crucial for the early detection of contaminants and for a rapid response to any changes in water quality, which is essential for protecting public health and the environment.

The integration with Arduino Cloud offers a simple and intuitive interface that allows users to access information from anywhere with an internet connection. The cloud dashboard not only facilitates clear and accessible visualization of the collected data but also enables the generation of detailed graphs and the automatic sending of notifications when the measured parameters exceed established limits, providing additional security and control.

Additionally, the project is perfect for a wide range of applications, from monitoring natural water bodies to quality control in drinking water supply systems or aquariums. The flexibility of the ESP32 and the Arduino Cloud platform allows for system customization according to specific needs, and its scalability potential offers the possibility of integrating additional sensors and new functionalities later on.

**Keywords:** Water Quality Monitoring, ESP32, pH Sensors, Turbidity Sensors, Arduino Cloud, Dashboard.

# 1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El objetivo del presente proyecto es implementar un sistema de monitoreo remoto de la calidad del agua utilizando una placa Arduino ESP32 y una serie de sensores especializados. El prototipo se diseñó para adquirir información en tiempo real sobre las mediciones de pH y turbidez en el agua.

El corazón del sistema es la placa Arduino ESP32, que proporciona la capacidad de conexión inalámbrica a través de Wi-Fi. Esta placa es ideal para aplicaciones de Internet de las Cosas (IoT) gracias a su mínimo costo, bajo uso de energía y amplias capacidades de programación.

Se conectaron los sensores adecuados al ESP32 para medir los parámetros específicos de interés. En este caso, se utilizaron sensores de pH y sensores de turbidez para detectar la transparencia del agua.

La programación se realizó utilizando el entorno de desarrollo de Arduino, aprovechando las bibliotecas y las funciones disponibles para interactuar con los sensores y la conectividad inalámbrica. Se establecieron intervalos de muestreo regulares para capturar los datos de los sensores y se utilizó un programa web para analizar y procesar los datos históricos.

Los datos recopilados se enviaron de forma inalámbrica a un *dashboard* o panel de control en la nube, utilizando el sitio web. El *dashboard* permitió visualizar datos en tiempo real, generar gráficos y enviar notificaciones en caso de que se superaran los límites aceptables de los parámetros.

Este prototipo de monitoreo remoto de calidad del agua tiene diversas aplicaciones prácticas, como la supervisión de la pureza del agua en fuentes naturales, sistemas de suministro de agua para consumo humano ó acuarios. También puede ser utilizado en proyectos de investigación científica o en aplicaciones industriales donde es necesario garantizar la calidad del agua en procesos de producción.

## 1.1. Objetivo general

Implementar un prototipo para monitoreo remoto de calidad de agua y visualización de datos mediante un *dashboard web*.

## 1.2. Objetivos específicos

- Establecer los requerimientos del prototipo.
- Definir los componentes de hardware y software.
- Diseñar el prototipo.
- Implementar el prototipo.
- Realizar pruebas de funcionamiento.

## 1.3. Alcance

Por medio de la presente investigación se pretende desarrollar un modelo de un sistema que facilite la supervisión remota de sensores de calidad de agua (pH y turbiedad) a través de una placa de desarrollo Arduino o similar y visualización a través de un *dashboard web*. Los datos de los sensores serán procesados por la placa de desarrollo.

Una vez procesados los datos, estos serán enviados por medio de comunicación inalámbrica desde la placa hacia el *dashboard web*, en donde las mediciones serán presentadas al usuario final. Adicionalmente, el *dashboard* permitirá realizar análisis históricos de las mediciones de calidad del agua.

## 1.4. Marco Teórico

### Internet de las Cosas (IoT)

IoT se refiere a la conexión de aparatos comunes a través de Internet, lo que facilita la recolección y el intercambio de información de manera continua. Esta tecnología permite que dispositivos como electrodomésticos, vehículos y equipos médicos se comuniquen entre sí y con los usuarios mediante redes de comunicación. Para lograrlo, IoT utiliza una combinación de sensores, que capturan datos del entorno; redes de comunicación, que transmiten esos datos; sistemas de software, que gestionan la información; y análisis de datos, que interpretan y generan *insights* valiosos a partir de la información recopilada.

El objetivo del IoT es que estos objetos recopilen datos sobre su entorno, tomen acciones basadas en esos datos y compartan información con otros dispositivos o sistemas. Por ejemplo, un termostato inteligente puede ajustar automáticamente la temperatura con base en los datos recolectados. Los sensores en los vehículos pueden recopilar información sobre el rendimiento del motor y enviarla para su análisis y mantenimiento [1].

En resumen, el IoT es la interconexión de objetos que recopilan, comparten y actúan en base a datos, lo que abre nuevas posibilidades en varios campos, pero también presenta desafíos en seguridad y privacidad [2].

### Arquitectura del IoT

La arquitectura del IoT generalmente se divide en cuatro capas que se observa en la Figura 1.1:

- Capa de Dispositivos: Incluye sensores y actuadores que recogen información del entorno.
- Capa de Red: Facilita la transmisión de datos desde los dispositivos mediante diversos protocolos de comunicación como Wi-Fi, Bluetooth, Zigbee, y redes móviles.
- Capa de Procesamiento: Involucra servidores y plataformas en la nube que procesan y almacenan los datos recibidos.
- Capa de Aplicación: Proporciona interfaces y aplicaciones a los usuarios finales para monitorear y gestionar los dispositivos.

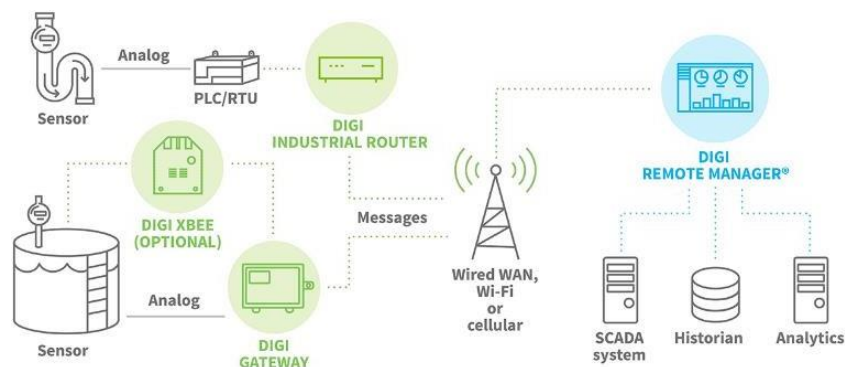


Figura 1.1 La estructura IoT [2]

## **IoT Aplicada al Monitoreo en Tiempo Real**

IoT aplicado al monitoreo en tiempo real involucra el uso de dispositivos conectados a Internet que recopilan y envían datos de manera continua sobre diversos parámetros a plataformas en la nube. Estos dispositivos pueden incluir sensores que monitorean una amplia variedad de variables, desde condiciones ambientales hasta la integridad estructural de infraestructuras, permitiendo una evaluación y decisión en tiempo real.

El IoT ha representado un gran avance en el monitoreo en tiempo real, ya que ha facilitado optimizar y acelerar la forma en que se obtiene y analiza la información. Esta tecnología facilita el acceso inmediato a los datos recopilados y permite un monitoreo constante y a distancia de diversas condiciones y entornos.

Entre las aplicaciones más importantes del IoT en el monitoreo en tiempo real se encuentra la transformación de sensores en dispositivos inteligentes, que permiten a los profesionales de diversos campos recopilar información mediante plataformas en la nube. Las aplicaciones del IoT no se limitan a instalaciones específicas, sino que también pueden estar presentes en dispositivos portátiles o remotos, permitiendo a los usuarios monitorear las condiciones desde cualquier lugar y en cualquier momento.

Entre los beneficios más importantes del IoT en el monitoreo en tiempo real se encuentran:

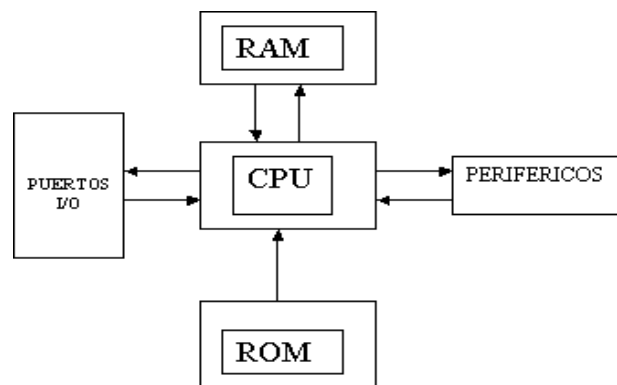
- La capacidad de ofrecer datos en tiempo real sobre diversas condiciones.
- La reducción de la necesidad de muestreos manuales, lo cual ahorra tiempo y recursos.
- La posibilidad de detectar y responder rápidamente a cambios o problemas.
- La monitorización remota de entornos en áreas de difícil acceso.
- La mejora de la gestión mediante el análisis de datos históricos y la predicción de tendencias.

## **Microcontroladores**

Los microcontroladores son dispositivos electrónicos que se encuentran en forma de chips y actúan como microcomputadoras. Tienen la capacidad de controlar elementos de ingreso y salida, y presentan similitudes a un pc.

Su estructura se compone de varios componentes esenciales como se indica en la Figura 1.2.

- **Procesador:** es responsable de ejecutar programas y traducir las instrucciones humanas a lenguaje de máquina. Está compuesto por la Unidad de Aritmética y Lógica (ALU) que ejecuta operaciones lógicas y aritméticas, mientras que la unidad de control que distribuye las señales lógicas, y los registros que almacenan los resultados de las instrucciones.
- **Unidades de memoria:** existen tres tipos de memoria en un microcontrolador. La memoria FLASH almacena los programas, la EEPROM es una memoria no temporal utilizada para configuraciones, y la memoria RAM almacena datos y variables temporales de los programas.
- **Periféricos:** son circuitos digitales que permiten la comunicación del microcontrolador con el mundo exterior. Los periféricos incluyen interfaces de entrada/salida, interfaces seriales e interfaces analógicas. Estos componentes facilitan la lectura de sensores y otras interacciones con el entorno [3].



**Figura 1.2.** Estructura de un microcontrolador [3]

Los microcontroladores son esenciales para el funcionamiento de dispositivos IoT. Algunos de los microcontroladores más populares en proyectos IoT incluyen:

- **Arduino:** Una plataforma de código abierto con una gran comunidad y muchos recursos disponibles para desarrolladores.
- **Raspberry Pi:** Aunque es más potente que un microcontrolador típico, es ampliamente utilizado en proyectos IoT debido a su capacidad de ejecutar un sistema operativo completo.
- **ESP8266 y ESP32:** Microcontroladores con capacidades de conectividad Wi-Fi integradas, populares en proyectos IoT por su bajo costo y alta funcionalidad.

## **Arduino**

Arduino es una placa de desarrollo que permite crear y programar varios proyectos. Sin embargo, en su núcleo, Arduino está compuesto por un microcontrolador ATMEL que se utiliza para cargar y ejecutar instrucciones desde la plataforma Arduino IDE. Esta combinación de hardware y software ha hecho posible desarrollar programas para una amplia gama de aplicaciones [4].

Es importante destacar que Arduino no se limita a un solo modelo de placa, sino que existen diferentes tipos de Arduino disponibles en el mercado. Algunos ejemplos incluyen Arduino UNO, Nano, Leonardo y otros modelos similares, todos los cuales constituyen una parte del ecosistema de proyectos Arduino. Cada tipo de placa Arduino tiene sus propias características y especificaciones técnicas, lo que permite adaptarse a diversas necesidades y requisitos de proyectos [5].

La versatilidad de Arduino ha llevado a su popularidad en la comunidad de desarrolladores y entusiastas de la electrónica. La plataforma Arduino ofrece una interfaz intuitiva para programar y controlar dispositivos electrónicos, lo que lo hace una alternativa atractiva tanto para novatos como para especialistas en la materia. Además, la amplia disponibilidad de documentación, tutoriales y bibliotecas de código abierto ha facilitado aún más el desarrollo de proyectos con Arduino [6] [7].

## **Calidad del Agua**

La calidad del agua implica las propiedades físicas, químicas y biológicas del agua que definen su aptitud para distintos usos [8]. El agua es un elemento necesario para el consumo del ser humano, por lo tanto, cuando esta satisface los requerimientos bacteriológicos mínimos que se exigen según la norma legal vigente en relación a su utilización, es cuando se afirma que es de calidad. Además, esto depende de factores tales como: concentración, especificaciones, elementos físicos de carácter orgánico e inorgánico. A su vez la estructura y el estado de la biota acuática que está en el agua, teniendo en cuenta que, la calidad del agua tiene variantes a nivel espacial y temporal según aspectos externos e internos al cuerpo del agua [9].

Algunos de los elementos que se generan de la contaminación del agua son [11]:

- Agroquímicos (pesticidas), que se esparcen de manera intencional en amplias áreas para la protección de diversos tipos de sembrados de plaga, así como otras pueden ser los derivados industriales, que se descargan en el agua o en el aire, ya sea de forma directa o indirecta.

- Artículos farmacéuticos y cosméticos, utilizados en la medicina humana, se introducen en las aguas superficiales mediante los residuos del sistema de alcantarillado (tratados o no), se puede dar opción al tratamiento de agua residual pero estos productos no se eliminan completamente, por lo tanto, pueden acceder a las aguas superficiales sin dificultad. De esta manera el agua para consumo humano es asegurada por las autoridades, y su inclusión es necesaria para garantizar la presencia de la población. en un sitio específico, para lo cual se deben cumplir normativas de calidad para el agua potable en todas las naciones y no puede provenir de cualquier fuente.

En general, según el detalle legal del agua potable se observa un listado de componentes que se relacionan en función a un nivel tolerable para garantizar la calidad del agua.

### **Potencial Hidrogeno (pH)**

El pH mide la concentración de iones de hidrógeno en una sustancia y es una abreviatura para el potencial de hidrógeno. Este concepto, introducido por el químico danés Sorensen, se define como el logaritmo negativo en base 10 de la actividad de los iones de hidrógeno[12].

El pH se mide en unidades logarítmicas, similar a la escala de Richter, lo que significa que cada unidad representa una variación de 10 veces en el nivel de acidez. Por ejemplo, el agua con un pH de 5 es diez veces más ácida que el agua con un pH de 6 [13].

La medición del pH es un procedimiento analítico fundamental y ampliamente utilizado en disciplinas como la química, la bioquímica y la química del suelo [12].

El pH influye en diversas propiedades clave de la estructura y función de las biomoléculas, en particular de las enzimas, y, por ende, afecta el comportamiento de células y organismos[12].

### **Turbiedad**

Por turbidez o turbiedad está entendida a la evaluación del nivel de transparencia del agua u otros líquidos incoloros, lo que se produce por la inclusión de partículas en suspensión. Ya que cuanto más sean las partículas sólidas presentes en el líquido, mayor va a ser el grado de turbidez [14].



La turbidez del agua se mide en NTU (Unidades Nefelométricas de Turbidez). El nefelómetro o turbidímetro, que mide la intensidad de la luz dispersada a 90 grados después de atravesar una muestra de agua, es el dispositivo utilizado para esta evaluación [14].

### **Sólidos Totales**

Si bien, que pueden estar suspendidos o disueltos en el agua, pueden perjudicar sanidad del agua y su suministro. La presencia de sólidos totales en el agua puede tener efectos negativos se describen como la materia que se encuentra en forma de residuo luego de la evaporación y secado a 103 o 105° C [15].

Por lo tanto, los sólidos totales son las materias sólidas que pueden estar disueltas o suspendidas en un líquido como es el agua, por lo general el agua residual puede disponer de estos [15].

Para analizar los sólidos totales se necesita separarlos del agua, y se requiere su calentamiento hasta 103-105°C hasta evaporarla, luego se perderá algunas sustancias y otras no se evaporan como los aceites pesados [15].

### **Importancia del Monitoreo de la Calidad del Agua**

El monitoreo remoto recurso esencial para asegurar salubridad y proteger los ecosistemas acuáticos, permitiendo una vigilancia continua y precisa de las condiciones ambientales. Esta tecnología no solo facilita el cumplimiento de normativas ambientales y regulaciones, sino que también asegura que las prácticas de gestión y conservación. Con el uso de sensores avanzados y sistemas de comunicación, el monitoreo remoto ofrece una solución eficiente para preservar la integridad de los ecosistemas y la salud de las comunidades dependientes del agua.

### **Normativas y estándares internacionales para la calidad del agua**

Existen diversos estándares y regulaciones internacionales que definen los requisitos de calidad del agua y los límites aceptables de los parámetros de monitoreo. Algunos de los estándares y regulaciones más reconocidos son los siguientes:

- Organización Mundial de la Salud (OMS): La OMS establece pautas para la calidad del agua potable en su "Directriz para la calidad del agua potable". Estas pautas abarcan unos parámetros, incluyendo sustancias químicas, microorganismos y características físicas del agua, y establecen límites aceptables para cada uno [18].

- Directiva Marco del Agua de la Unión Europea (DMA): La DMA establece los estándares de calidad del agua en la Unión Europea y define los requisitos para la protección y mejora de los cuerpos de agua. Establece límites para diversos parámetros, como sustancias químicas, microorganismos y características físicas del agua [11].
- Agencia de Protección Ambiental de EE.UU.(EPA): La EPA establece los Estándares Primarios de Calidad del Agua Potable en los Estados Unidos. Estos estándares establecen límites máximos permitidos para varios contaminantes en el agua potable, como metales pesados, productos químicos orgánicos y microorganismos.
- Norma ISO 5667: Esta norma internacional proporciona directrices para el muestreo de agua y define métodos para el análisis de diferentes parámetros [19]. También establece límites de calidad para algunos parámetros específicos.
- Normas de la Comisión Europea: La Comisión Europea ha establecido normas para varios parámetros de calidad del agua, como la Directiva sobre Nitratos, que establece límites para la concentración de nitratos en el agua potable, y la Directiva sobre Aguas de Baño, que establece límites para la calidad del agua en áreas destinadas al baño [20].

## 2 METODOLOGÍA

En la metodología del proyecto para el prototipo de monitoreo remoto de calidad del agua y visualización de datos mediante un *dashboard* web, se siguieron los siguientes pasos:

- Se llevó a cabo un análisis para identificar los mejores elementos para implementar el prototipo de monitoreo de la calidad de agua.
- Se evaluó cual tipo de comunicación inalámbrica es el más adecuado para el prototipo. Se evaluaron opciones como Wi-Fi o Bluetooth, teniendo en cuenta el consumo de energía y la facilidad de implementación, se realizó una investigación sobre los componentes clave del prototipo, incluyendo el microcontrolador, el sensor de turbidez, el medidor de pH y la plataforma IoT. Se compararon características, rendimiento, necesidades y tamaño de al menos tres opciones para cada componente, justificando la selección final.

- Se utilizó un programa para diseñar el circuito electrónico del prototipo y crear modelos para llevarlos a una placa que indicaran los espacios ocupados por cada componente en la baquelita. Además, se desarrolló el código de programación para el microcontrolador, siguiendo un diagrama de flujo que explica su funcionamiento. Se tuvieron en cuenta las lecturas del sensor y la comunicación con la plataforma IoT.
- Posteriormente se realizaron pruebas para verificar el correcto funcionamiento del prototipo se verificó la adecuada alimentación, la correcta lectura del sensor de turbidez y pH, la visualización correcta de la información en la plataforma y el envío de los datos históricos.

### **3 RESULTADOS**

El prototipo para monitoreo remoto de la calidad del agua permite visualizar datos obtenidos por sensores especializados en contacto directo con el medio acuático. Los datos capturados sobre la calidad del agua se envían al microcontrolador para su procesamiento y luego se transmiten a la plataforma IoT mediante el módulo inalámbrico, como Wi-Fi. La plataforma IoT, previamente configurada con herramientas de visualización, presenta la información al usuario responsable. Tanto el sitio web como la aplicación móvil de la plataforma permiten acceder a estos datos. El prototipo se adapta al entorno acuático para proporcionar datos precisos, asegurando un monitoreo efectivo de mediciones en tiempo real e históricas.

#### **3.1 Establecimiento de los requerimientos del prototipo**

El desarrollo y la implementación de un prototipo para el monitoreo remoto de la calidad de agua implica la consideración de los siguientes requerimientos:

- El microcontrolador utilizado debe ser capaz de establecer conexiones, lo que implica que debe tener acceso a Internet para la transmisión de datos, esto permitirá el monitoreo en tiempo real y la comunicación con plataformas IoT.
- Para el monitoreo de la calidad de agua se utilizará una plataforma IoT que mostrará los datos recogidos por los sensores en tiempo real.
- La plataforma web debe brindar datos históricos recopilados de las mediciones anteriormente realizadas permitiendo establecer un análisis de manera intuitiva.

### 3.2 Definición de los Componentes de Hardware y Software

Se determinaron los elementos de hardware y software destinados al prototipo, según los requisitos establecidos. En la selección de cada componente se consideraron aspectos como dimensiones, eficiencia y otras características generales que aborden de manera efectiva las demandas esenciales para el desarrollo del prototipo.

#### Microcontrolador y Comunicación Inalámbrica

En lo que respecta a la conectividad inalámbrica, se identificaron diversos microcontroladores que incorporan esta funcionalidad. Por consiguiente, se evaluaron tres modelos específicos: NodeMCU-ESP8266, Arduino LilyPad y NodeMCU-ESP32. Se detallan las especificaciones de cada versión en la Tabla 3.1.

**Tabla 3.1** Comparativa de microcontroladores con capacidad de comunicación inalámbrica [20], [22]

Especificación	NodeMCUESP32	NodeMCUESP8266	Arduino lilypad
Procesador CPU	Xtensa Dual-Core LX6 de 32 bits	Tensilica RISC CPU Xtensa LX106 de 32 bits	ATmega168 V ATmega328 P
Voltaje de alimentación	3.3 - 5 (V)	5 (v)	2.5 - 5 (v)
Puertos	38	16	14
Pines de entrada y salida digital	18	1	6
Dimensión	25.50 x 18.00 (mm)	49x26 (mm)	50 (nm)
Estándar de comunicación inalámbrica	802.11 b/g/n	802.11 b/g/n	Ninguno
Consumo de corriente	70 (mA)	70 (mA)	40 (mA)
Módulo wifi/Bluetooth	Si-no	Si-si	No-no

El microcontrolador NodeMCU-ESP8266 incluye un módulo Wi-Fi integrado que es uno de los requisitos del prototipo. Sin embargo, el microcontrolador NodeMCU-ESP32 ofrece aún más ventajas. Además de contar con un módulo Wi-Fi, también tiene un

módulo Bluetooth incorporado, lo cual amplía las posibilidades de conectividad del dispositivo.

El NodeMCU-ESP32 cuenta con un total de 18 pines, lo cual es más que el Arduino Lilypad y el NodeMCU-ESP8266. Esto proporciona más opciones de conexión con otros dispositivos o sensores externos, ampliando las capacidades del prototipo [21].

Además, el NodeMCU-ESP32 tiene un tamaño más compacto en comparación con otros microcontroladores, lo que facilita su transporte y montaje en el prototipo. Esto es especialmente útil si se necesita desplegar el dispositivo en diferentes ubicaciones o en un espacio reducido.

Una ventaja adicional del NodeMCU-ESP32 es su módulo Wi-Fi integrado, lo cual elimina la necesidad de utilizar componentes adicionales para el envío de notificaciones. Esto simplifica el diseño y reduce la complejidad del prototipo [22].

El microcontrolador NodeMCU-ESP32 es el dispositivo más adecuado para el proyecto de monitoreo de la calidad de agua debido a su tamaño compacto, mayor número de pines, módulos Wi-Fi y Bluetooth integrados, y su capacidad para trabajar con una amplia variedad de sensores.

### **Sensor de Turbidez**

El sensor de turbidez de Arduino por gravedad es una herramienta esencial para detectar la calidad del agua. Este dispositivo evalúa la turbidez, o la opacidad del agua, utilizando luz para identificar las partículas suspendidas. La medición se basa en la transmisión y dispersión de la luz, las cuales varían según la concentración de sólidos suspendidos totales (TSS) en el agua. A medida que aumenta la TSS, también lo hace el nivel de turbidez del agua..

Este sensor de turbidez de Arduino tiene una amplia gama de aplicaciones, que incluyen la monitorización de la calidad del agua en ríos y arroyos, el seguimiento de aguas residuales y efluentes, el control en estanques de sedimentación, y la investigación sobre el transporte de sedimentos. También se utiliza en pruebas de laboratorio, proporcionando datos críticos para el análisis ambiental y la gestión del agua [23].

El sensor de turbidez de Arduino por gravedad ofrece salidas de señal tanto analógicas como digitales. En el modo digital, los usuarios pueden ajustar el umbral de detección, lo que permite una mayor flexibilidad dependiendo de la placa de desarrollo utilizada en el proyecto. Esta característica hace que el sensor sea adaptable a diversas necesidades y entornos de uso, facilitando su integración en diferentes sistemas de

monitoreo de calidad del agua [24], [25]. Las especificaciones de estos sensores están detalladas en la Tabla 3.2.

**Tabla 3.2** Comparativa de sensores de turbidez [26] [30]

<b>Sensor de Turbidez</b>	<b>Gravity: Analog Turbidity Sensor</b>	<b>SEN0189 Turbidity Sensor</b>	<b>DFRobot Turbidity Sensor</b>	<b>YF-B1 Turbidity Sensor</b>	<b>Gravity: Analog Turbidity Sensor V2</b>
Método de Medición	Óptico	Óptico	Óptico	Óptico	Óptico
Salida Voltaje de Funcionamiento	Analógica	Analógica	Analógica	Analógica	Analógica
Corriente de Funcionamiento	5(V DC	5(V DC	5(V DC	5(V DC	5(V DC
Rango de Medición	40 (mA) máx.	40(mA) máx.	40 (mA) máx.	40 (mA) máx.	40(mA) máx.

Entre los sensores de turbidez disponibles, el DFRobot Turbidity Sensor se destaca por varias razones. Todos los sensores listados, incluidos el Gravity: Analog Turbidity Sensor, el SEN0189 Turbidity Sensor, el YF-B1 Turbidity Sensor, y el Gravity: Analog Turbidity Sensor V2, comparten especificaciones similares en términos de método de medición (óptico), salida (analógica), voltaje de funcionamiento (5V DC) y corriente de funcionamiento (40 mA máx.). Sin embargo, el DFRobot Turbidity Sensor sobresale debido a la reputación de DFRobot, una reconocida marca en el campo de los sensores y módulos para Arduino y otros microcontroladores.

El sensor de DFRobot está diseñado para proporcionar mediciones precisas de la turbidez del agua y es fácil de usar e integrar en proyectos de Arduino y otros microcontroladores. Viene con una interfaz analógica estándar que permite una conexión sencilla y rápida. Además, DFRobot proporciona documentación detallada y ejemplos de código, lo que facilita su implementación incluso para personas con poca experiencia en electrónica.

El DFRobot Turbidity Sensor tiene un rango de medición amplio, adecuado para una variedad de aplicaciones, detectando tanto niveles bajos como altos de turbidez en el agua. Esta versatilidad lo hace ideal para proyectos que requieren monitoreo de la calidad del agua. Además, está diseñado para ser duradero y resistente, con una carcasa robusta que protege el componente interno, garantizando un funcionamiento confiable incluso en entornos adversos o en presencia de agua.

En comparación con los otros sensores listados, el DFRobot Turbidity Sensor destaca como una opción confiable y precisa a un precio competitivo, lo que lo convierte en la mejor opción para proyectos que requieren monitoreo de la calidad del agua. Las específicas detalladas de este sensor se presentan en la Tabla 3.3, y un ejemplo visual del sensor se presenta en la Figura 3.1.

**Tabla 3.3** Características del sensor de turbidez DFRobot Turbidity Sensor [26]

<b>Característica</b>	<b>Valor</b>
Funcionamiento	5 (V) CC
Corriente de funcionamiento	40 mA (máx.)
Tiempo de respuesta	<500 (ms)
Resistencia de aislamiento	100M (Min)
Método de salida	Analógico
Salida analógica	0-4,5 (V)
Salida digital Señal de nivel alto/bajo	(ajustable mediante potenciómetro)
Temperatura de funcionamiento	5 - 90 (°C)
Peso	30 (g)
Dimensiones del adaptador	38 (mm) * 28 (mm) * 10 (mm) / 1,5 pulgadas * 1,1 pulgadas * 0,4 pulgadas



**Figura 3.1** Sensor de turbidez DFRobot Turbidity Sensor [26]

### Sensor de pH

Las características de cada sensor de pH considerado para el prototipo se las detalla en la Tabla 3.4.

**Tabla 3.4** Comparativa de sensores de pH [31] [36]

Sensor de PH	Gravity: Analog pH Sensor	SEN0161 pH Sensor	PH-4502C Sensor de PH	Atlas Scientific pH Sensor	Atlas Scientific EZO pH Circuit
Método de medición	Electroquímico	Electroquímico	Electroquímico	Electroquímico	Electroquímico
Salida	Analógica	Analógica	Analógica	I2C	Serial
Voltaje de funcionamiento	5 (V) DC	5 (V) DC	5 (V) DC	3.3-5 (V) DC	3.3-5 (V) DC
Corriente de funcionamiento	5-10 (mA)	5-10 (mA)	5-10 (mA)	1-2 (mA)	1-2 (mA)
Rango de medición	0-14 pH	0-14 pH	0-14 pH	0-14 pH	0-14 pH

Entre los sensores de pH disponibles, el módulo PH-4502C se destaca por varias razones. Todos los sensores listados, incluidos el Gravity: Analog pH Sensor, el SEN0161 pH Sensor, el Atlas Scientific pH Sensor y el Atlas Scientific EZO pH Circuit, comparten características similares en términos de método de medición



(electroquímico) y rango de medición (0-14 pH). Sin embargo, el módulo PH-4502C sobresale debido a su diseño y compatibilidad con diversas plataformas.

El módulo PH-4502C utiliza un electrodo de pH de alta calidad que proporciona mediciones precisas y confiables del pH. La sonda de electrodo está diseñada para ofrecer una respuesta rápida y estable, garantizando resultados precisos en diferentes condiciones de pH. Además, tiene una amplia gama de medición de pH, adecuado para diversas aplicaciones, ya que puede medir el pH en un rango típico de 0 a 14 pH, cubriendo tanto soluciones ácidas como alcalinas.

Este sensor incluye un conector BCN para conectar la sonda de electrodo, lo que facilita una conexión segura y estable entre el módulo y la sonda. Es compatible con Arduino y ESP32, lo que permite su fácil integración en proyectos basados en estas plataformas y facilita su uso en combinación con otros componentes electrónicos para desarrollar aplicaciones personalizadas.

A continuación, se presentan las características del sensor PH-4502C en la Tabla 3.5, mientras que los pines de salida se observan en la Figura 3.2.

**Tabla 3.5** Características del sensor de PH-4502C Sensor de pH [33]

Especificación	Valor
Voltaje de alimentación	5v
electrodo	E201-BNC
Tipo de sonda	Grado de laboratorio
Tiempo de respuesta	5 seg
Rango de detección	0 ---14
Rango de temperatura	0 --- 80
Temperatura de trabajo	10 --- 50
Humedad de trabajo	95% RH sin condensación

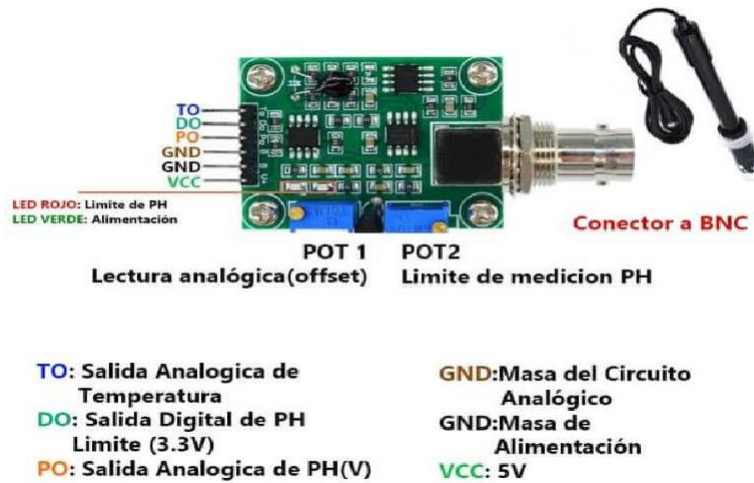


Figura 3.2 Sensor de PH-4502C [33]

### Plataforma IoT para visualización de datos y notificaciones

Se consideraron diversas plataformas IoT, incluyendo Arduino Cloud, Blynk, ThingSpeak, Adafruit IO, Google Cloud IoT y AWS IoT Core, para asegurar un monitoreo continuo, una efectiva presentación de datos y un sistema de alertas eficiente. La evaluación se centró en comparar las capacidades de cada plataforma para seleccionar la que mejor se ajustara a las necesidades del proyecto. en la Tabla 3.6.

Tabla 3.6 Comparativa de las plataformas IoT [37] [42]

Plataforma IoT	Arduino Cloud	Thing Speak	Blynk	Adafruit IO	Google Cloud IoT	AWS IoT Core
Compatibilidad de Dispositivos	Alta: Arduino ESP8266, ESP32	Moderada: Matlab	Alta: ESP8266, ESP32	Alta: Arduino, ESP8266, ESP32	Alta: Diversos dispositivos	Alta: Diversos dispositivos
Facilidad de Uso	Muy alta	Alta	Muy alta	Alta	Moderada	Moderada
Visualización de Datos	Excelente	Buena	Buena	Muy buena	Excelente	Excelente
Costo	Bajo	Gratuito	Moderado	Gratuito Bajo	Alto	Alto
Documentación y Soporte	Excelente	Buena	Buena	Muy Buena	Buena	Excelente

En el ámbito de las plataformas IoT para la presentación de datos y las notificaciones, Arduino Cloud se destaca como una opción superior por diversas razones. En comparación con otras plataformas como ThingSpeak, Blynk, Adafruit IO, Google Cloud IoT y AWS IoT Core, Arduino Cloud ofrece una alta compatibilidad con dispositivos populares como Arduino, ESP8266 y ESP32. Esta compatibilidad facilita la integración con una amplia gama de hardware y asegura mayor flexibilidad en el diseño y desarrollo de proyectos IoT. Mientras que ThingSpeak es compatible principalmente con Matlab y Raspberry Pi, lo que puede limitar las opciones de hardware, y Blynk, aunque altamente compatible, puede ser más complejo de configurar para usuarios novatos, Arduino Cloud se distingue por su interfaz intuitiva y el soporte para bocetos de Arduino.

Arduino Cloud también proporciona herramientas excepcionales para la exhibición de datos, incluyendo gráficos en tiempo real y *dashboards* personalizables, facilitando a los usuarios monitorear y analizar datos de manera eficiente. Además, plataformas como Google Cloud IoT y AWS IoT Core, aunque ofrecen capacidades avanzadas y excelente compatibilidad con diversos dispositivos, son más costosas y pueden ser excesivas para pequeños proyectos o usuarios con presupuesto limitado.

La plataforma también cuenta con funcionalidades de notificación robustas, permitiendo recibir alertas basadas en condiciones específicas, esenciales para el monitoreo en tiempo real. Con un costo competitivo y accesible, junto con una documentación extensa y una comunidad activa de soporte, Arduino Cloud se convierte en la elección ideal tanto para aficionados como para desarrolladores experimentados en proyectos IoT.

### **3.3 Diseño del Prototipo**

#### **Comunicación Entre ESP32 y Sensores**

El ESP32 cuenta con soporte para diferentes protocolos de comunicación, incluyendo I2C. Para establecer la comunicación I2C entre el ESP32 y los sensores de turbidez y PH, es necesario conectar los pines correspondientes en ambos dispositivos.

Por lo general, en el ESP32 se utilizan los pines GPIO 21 como SDA (Serial Data Line) y GPIO 22 como SCL (Serial Clock Line) para la comunicación I2C como nos indica la Figura 3.3.

Una vez que se han determinado los pines específicos, es posible establecer la comunicación I2C entre el ESP32 y los sensores de turbidez y PH. Esto permitirá al ESP32 enviar y recibir datos del sensor de turbidez como del sensor de PH, lo que simplifica la interpretación de la información de turbidez del agua y el PH de la misma.

Es importante tener en cuenta que, además de los pines de comunicación I2C, es posible que se requieran conexiones adicionales, como pines de alimentación y conexión a tierra, para garantizar el adecuado funcionamiento de los sensores, como se muestra en la Figura 3.4 y Figura 3.5.

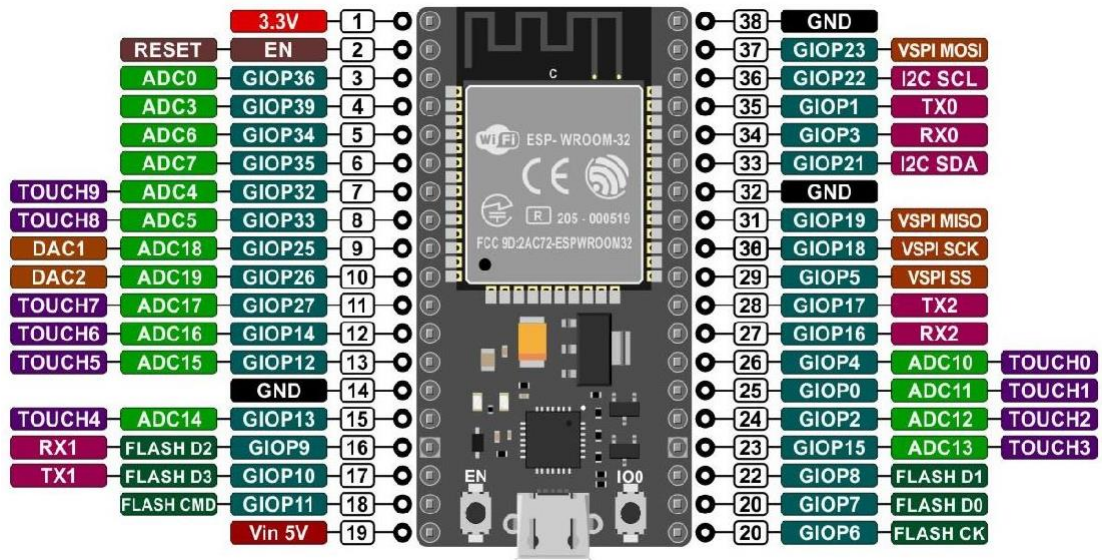


Figura 3.3 Esquema del microcontrolador NodeMCU-ESP32

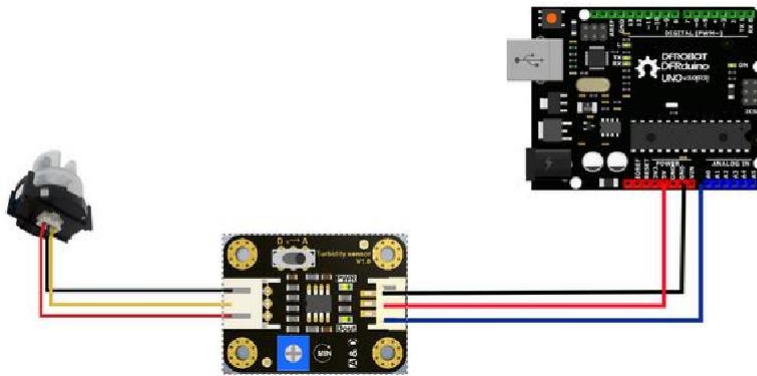
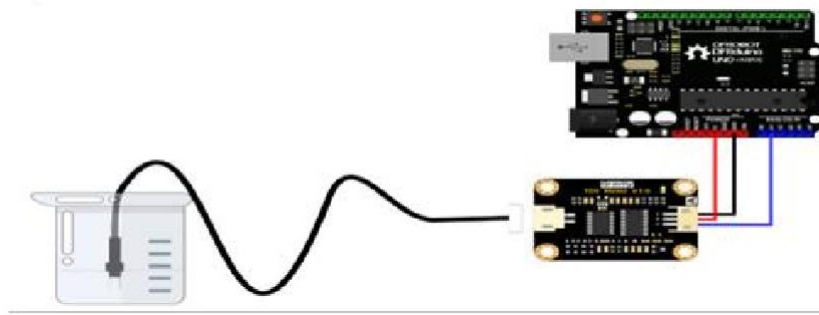


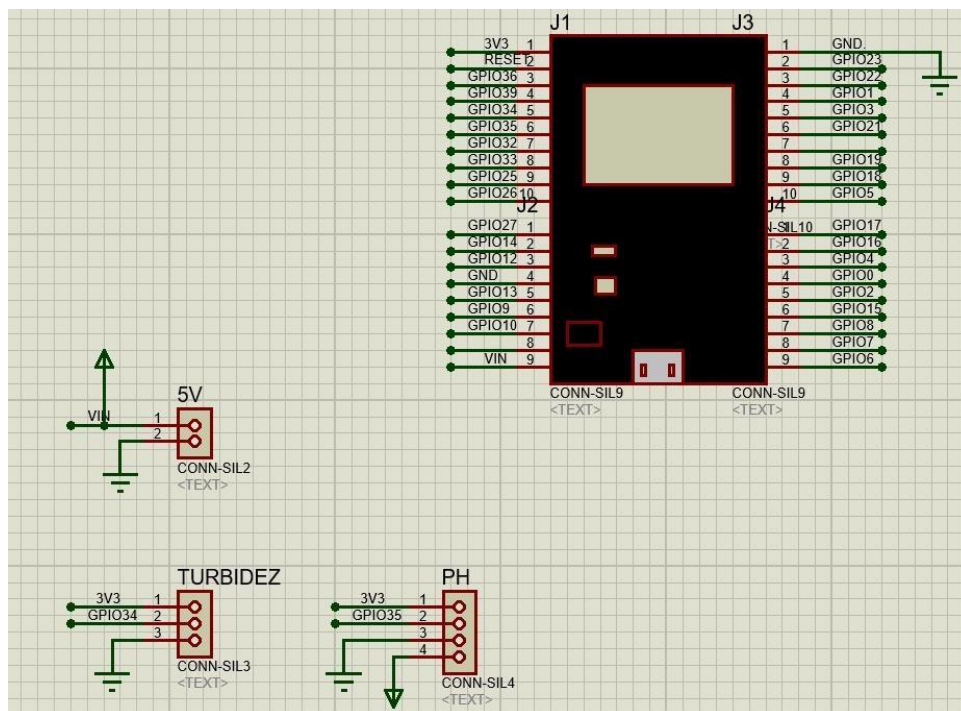
Figura 3.4 Esquema de conexión entre el ESP32 y el sensor de turbidez



**Figura 3.5** Esquema de conexión entre el ESP32 y sensor de pH

### Circuito Electrónico

Crear un diagrama esquemático del sistema es fundamental para establecer cómo se interconectan los componentes electrónicos. Se utilizó el software Proteus para esta tarea, ya que permite no solo diseñar el diagrama electrónico, sino también simular los circuitos y diseñar las placas electrónicas PCB. Este diagrama del circuito del prototipo es crucial para el desarrollo del proyecto. Está ilustrado en la Figura 3.6.

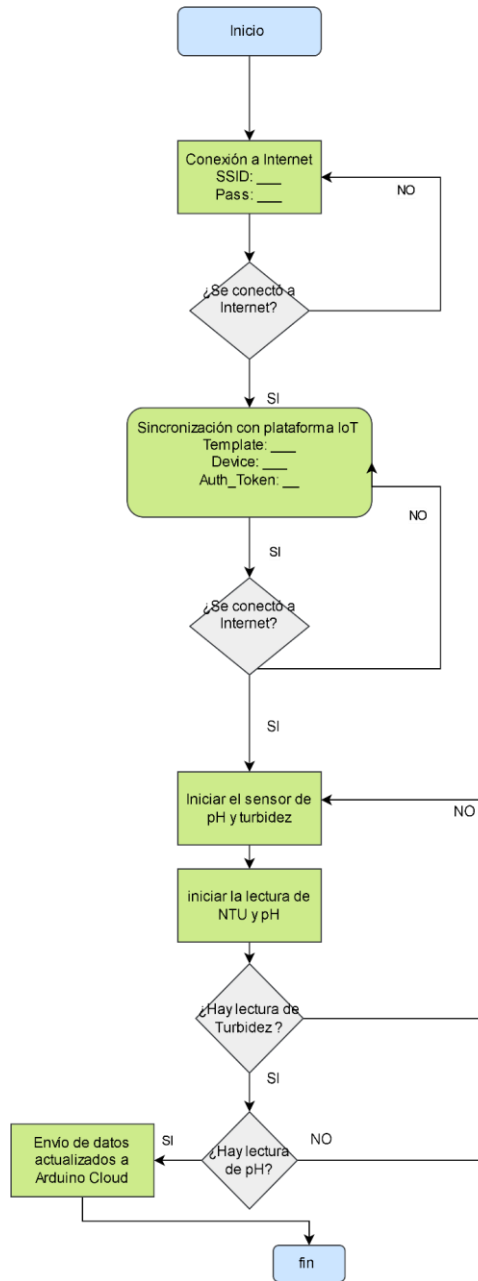


**Figura 3.6** Circuito del Prototipo

## Diagrama de Flujo

En el diagrama de flujo detalla el funcionamiento de un sistema de monitoreo de pH y turbidez utilizando un ESP32 y un sensor de pH y turbidez:

- **Conexión Wi-Fi:** El programa comienza estableciendo la conexión Wi-Fi proporcionando el nombre de la red y la contraseña. El ESP32 se conecta a la red Wi-Fi especificada.
- **Sincronización con la plataforma IoT:** El sistema se sincroniza con la plataforma IoT, lo que implica proporcionar la información necesaria para la comunicación. Se configuran las variables de Turbidez y PH.
- **Comunicación con los sensores:** El programa verifica la comunicación con el sensor de pH y turbidez. Si no hay errores de inicialización, se realiza la lectura de datos. En caso de que haya un error de inicialización del sensor, el programa intenta nuevamente establecer la comunicación.
- **Lectura de datos:** Se realiza la lectura del sensor de pH y se almacena el valor obtenido en una variable. Se realiza la lectura del sensor de turbidez y se almacena el valor obtenido en otra variable.
- **Envío de datos a la plataforma IoT:** Los valores de pH y turbidez se envían a la plataforma IoT para su registro y análisis, los datos se envían utilizando los identificadores y la comunicación previamente establecida con la plataforma.
- **Repetición del proceso:** El programa espera un intervalo de tiempo antes de repetir el proceso de lectura y envío de datos, se vuelve al Paso 3 para realizar una nueva lectura y enviar los datos actualizados a la plataforma IoT. Este diagrama de flujo describe el flujo general del sistema de monitoreo de agua utilizando un ESP32 y un sensor de pH y turbidez. Como podemos consultar la Figura 3.7.



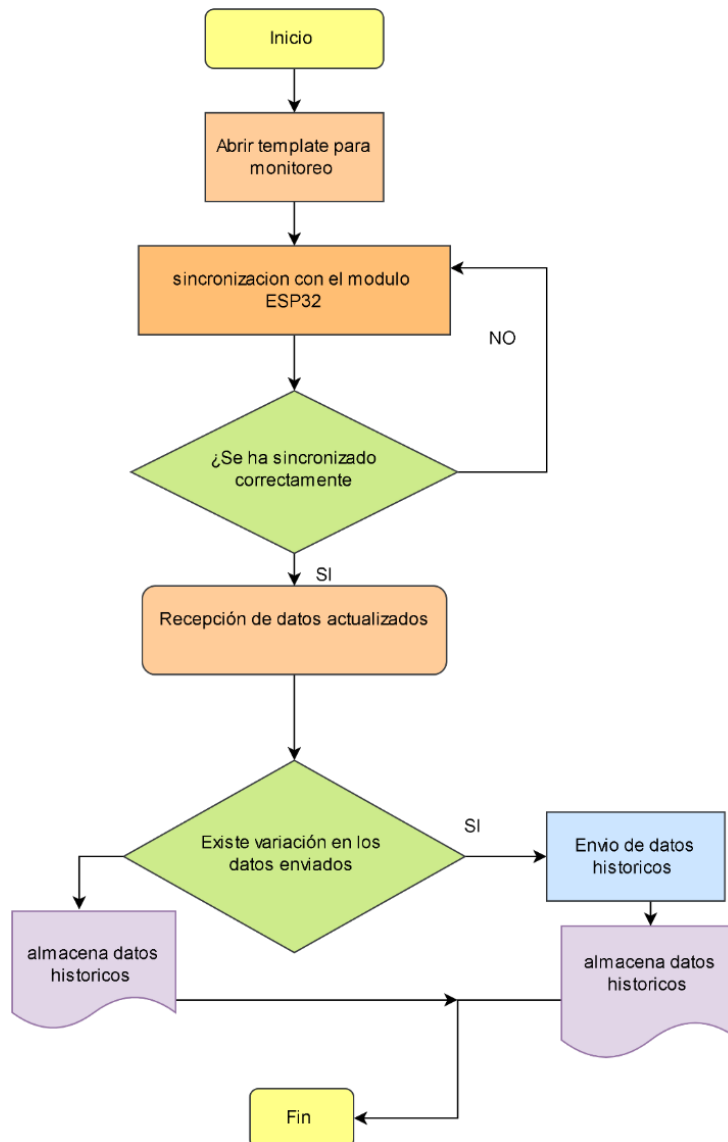
**Figura 3.7** Diagrama del sistema de monitoreo

Para el proceso de manejo de la plataforma Arduino Cloud se siguen los siguientes pasos:

El primer paso del proceso consiste en iniciar el *dashboard* previamente creada en la plataforma Arduino Cloud. A continuación, se realiza la sincronización entre el microcontrolador y la plataforma IoT. Este proceso de sincronización es automático y se encarga de verificar las credenciales proporcionadas para completar la conexión correctamente. Una vez que se establece la sincronización de manera exitosa, la plataforma está lista para recibir los datos actualizados de PH y turbidez enviados por

el microcontrolador a través de Internet. Estos datos son impresos en el *dashboard* de la plataforma Arduino Cloud y pueden ser visualizados por el usuario responsable de la supervisión.

El programa continúa ejecutándose en un ciclo infinito, a la espera de recibir nuevos datos provenientes del microcontrolador a través de Internet. En cada iteración, se actualiza la información en el *dashboard* de la plataforma Arduino Cloud, permitiendo al usuario visualizar los datos actualizados en tiempo real y almacenarlos como datos históricos mostrado en la Figura 3.8.



**Figura 3.8** Diagrama de Flujo de la plataforma Arduino Cloud



### 3.4 Implementación del Prototipo

#### Diseño de la Placa PCB

Se diseñó la placa electrónica PCB en el programa Proteus para el prototipo de monitoreo y control de calidad de agua con un ESP32, conectando las pistas que van desde la fuente de alimentación como observamos en la Figura 3.9.

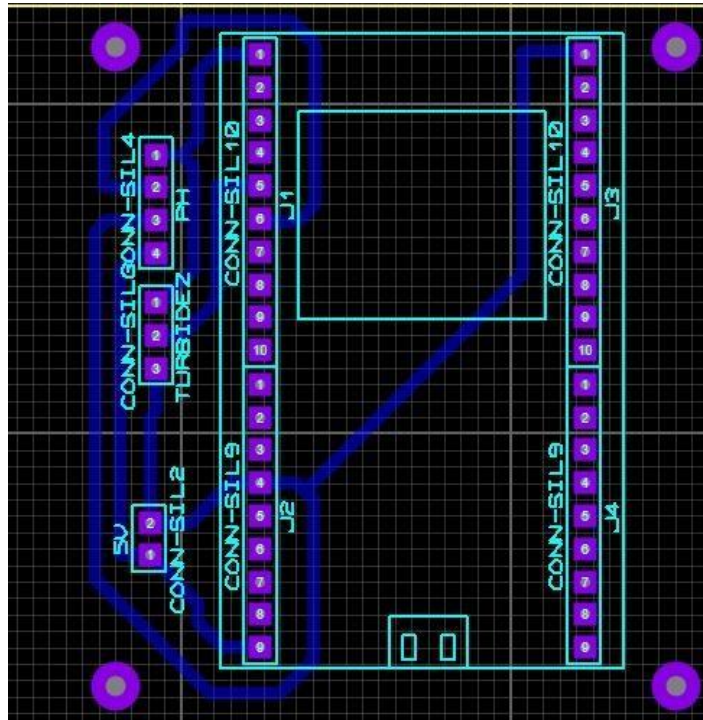
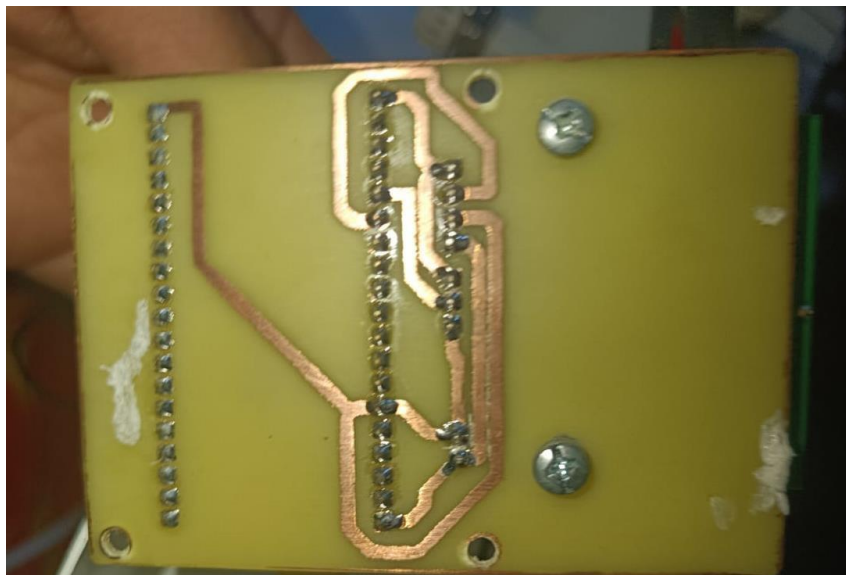


Figura 3.9 PCB Proteus ESP32

Para realizar el proceso de impresión de pistas en la baquelita para el prototipo, se consideró utilizar el método de impresión de pistas. Este método implica los siguientes materiales y pasos:

- Se seleccionó un papel especial para impresión de circuitos, que permite transferir las pistas desde un diseño impreso en papel a la baquelita.
- Además de la baquelita de cobre cortada según las dimensiones necesarias, se utilizó un marcador permanente negro o azul para delinear las pistas sobre la baquelita.
- Se emplearon materiales de limpieza como una esponja de lana de acero, alcohol y un trapo para preparar la superficie de la baquelita y asegurar una transferencia adecuada de las pistas.

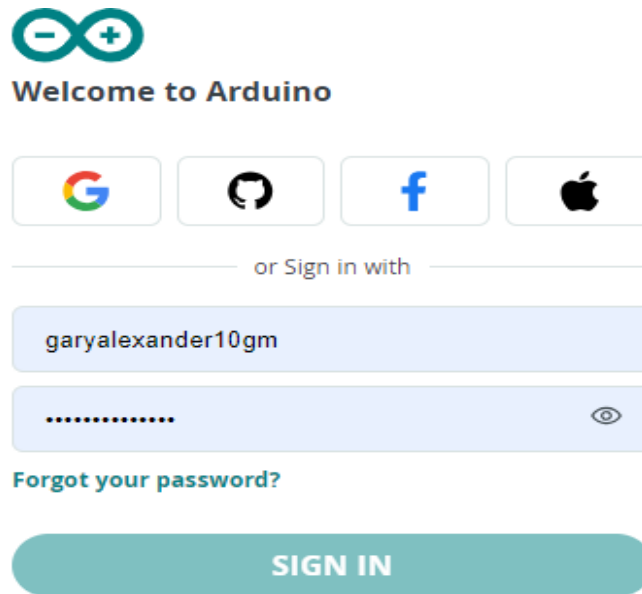
- Se utilizó la esponja de lana de acero para limpiar la superficie de cobre de la baquelita, asegurándose de eliminar cualquier impureza presente. Posteriormente, se aplicó alcohol para darle un acabado brillante final a la baquelita.
- Se utilizó papel Bond para imprimir las pistas, las cuales fueron luego transferidas a la baquelita mediante calor por medio de una plancha. Con un marcador permanente negro volvemos a marcar las pistas que no quedaron bien trazadas, facilitando así el proceso de grabado.
- El proceso de grabado de las pistas utilizó cloruro férrico, el cual se mezcló con agua en un recipiente pequeño, produciendo un líquido de color café. Después de preparar la baquelita con las pistas trazadas utilizando un marcador permanente, se sumergió la placa en el líquido de cloruro férrico. El tiempo total de inmersión fue de 40 minutos, durante los cuales se movió el recipiente de lado a lado cada 5 minutos para facilitar el proceso de eliminación del cobre de la baquelita y asegurar que solo quedaran las pistas grabadas.
- Finalmente, se perforaron los agujeros necesarios en la baquelita utilizando un taladro. Posteriormente, se aplicó estaño a las pistas impresas en la baquelita y se soldaron los módulos y la fuente en sus ubicaciones asignadas. Como se ilustra en la Figura 3.10.



**Figura 3.10** Baquelita terminada

## Configuración de la Plataforma IoT

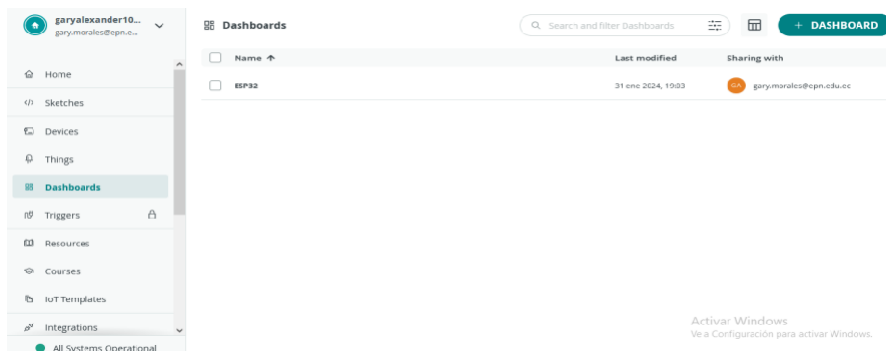
En la plataforma Arduino Cloud, es necesario tener una cuenta de usuario para poder acceder a sus funcionalidades. Para registrarse, es necesario proporcionar una dirección de correo electrónico y completar los datos necesarios para el registro, como se indica en la Figura 3.11. Cuando se inicia la cuenta registrada, hay varias opciones de desarrollo, incluyendo crear un nuevo proyecto.



**Figura 3.11** Registro en Arduino Cloud

Al crear un nuevo proyecto en Arduino Cloud, se pueden establecer diferentes configuraciones, como el nombre del proyecto, el tipo de hardware empleado y el tipo de conexión requerida. Además, es posible agregar una descripción para el proyecto, lo que facilita la identificación y comprensión de su propósito.

El objetivo de crear un nuevo proyecto en Arduino Cloud es poder utilizar las herramientas y servicios disponibles en la plataforma para desarrollar y monitorear dispositivos basados en Arduino. Esto incluye la posibilidad de agregar y configurar diferentes componentes electrónicos, escribir y cargar el código en los dispositivos, así como acceder a funciones de monitoreo y control a través de una interfaz como se puede visualizar en la Figura 3.12.



**Figura 3.12** Creación de una plantilla en Arduino Cloud

Una vez creada la nueva plantilla en Arduino Cloud, se presentan la opción de configuración del *dashboard* web en donde se procede a crear las variables de los sensores a utilizar, uno tipo *int* para turbidez y otra en tipo *float* para el pH como se muestra en la Figura 3.13.

Name ↓	Last Value	Last Update
<input type="checkbox"/> Ph float ph;	9.556	17 Jun 2024 23:40:07
<input type="checkbox"/> turbidez int turbidez;	948	17 Jun 2024 23:40:07

**Figura 3.13** Variables utilizadas en el dashboard

En la interfaz de Arduino Cloud, configura los *Datastreams* correspondientes a los sensores de pH y turbidez. Asigna nombres descriptivos a los *Datastreams* y establece las unidades de medida adecuadas como se indica en la Figura 3.14.

```

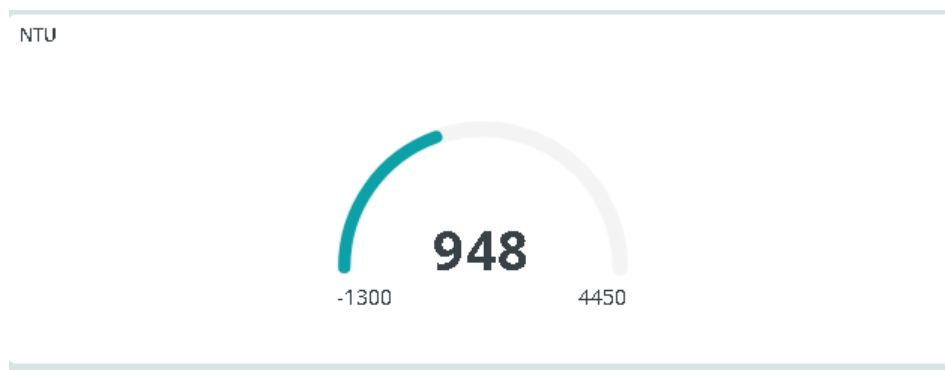
Untitled_dec13a.ino  ReadMe.adoc  thingProperties.h  Secret  ▼
1  :Author: garyalexander10gm
2  :Email:
3  :Date: 13/12/2023
4  :Revision: version#
5  :License: Public Domain
6
7  = Project: {Project}
8
9  Describe your project
10
11 == Step 1: Installation
12 Please describe the steps to install this project.
13
14 For example:
15
16 1. Open this file
17 2. Edit as you like
18 3. Release to the World!

```

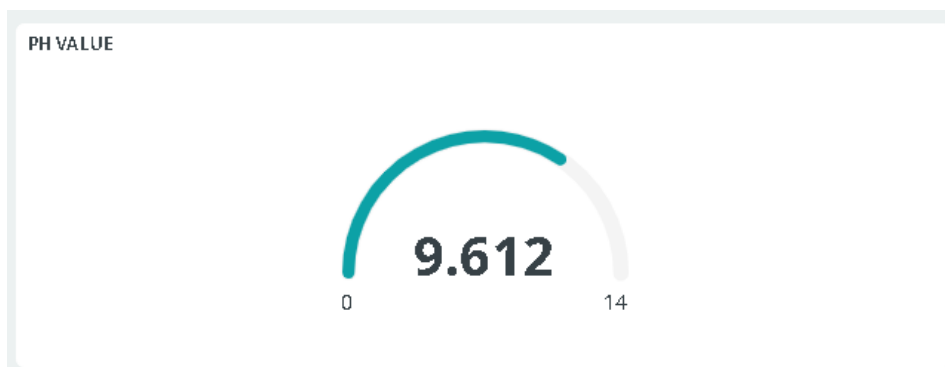
**Figura 3.14** Asignación de nombre descriptivos

En el código del Arduino, envía los datos de los sensores a los *Datastreams* correspondientes utilizando las funciones y métodos proporcionados por la biblioteca

En la interfaz de Arduino Cloud, configura *widgets* o paneles de visualización para mostrar los datos de los sensores de pH y turbidez. Se utiliza *widgets*, indicadores o cualquier otro widget adecuado para representar los datos de manera comprensible como se indica en la Figura 3.15 y Figura 3.16.



**Figura 3.15** Configuración de widgets Turbidez



**Figura 3.16** Configuración de widgets pH

### **Código de Programación para el Funcionamiento del Prototipo**

El microcontrolador ESP32 se programó utilizando el entorno de desarrollo Arduino IDE.

La primera parte del código incluye la librería *thingProperties.h* para las funciones relacionadas con Arduino IoT Cloud y define varias variables cruciales para la medición del pH. Entre estas variables se encuentran *calibration\_value*, utilizada para ajustar el cálculo del pH; *phval*, que almacena temporalmente una lectura de pH; *avgval*, que guarda la suma de los valores del arreglo *buffer\_arr* para calcular el promedio; y *buffer\_arr* (10) junto con *temp*, que se usan para almacenar y ordenar las lecturas del sensor de pH. En la función *setup* (), se inicializa la comunicación serial a 9600 (bps) con *Serial.begin(9600)*, se configura la resolución de lectura analógica a 12 bits con *analogReadResolution* (12), y se introduce un retraso de 1.5 segundos con *delay* (1500) para permitir la conexión de la consola serial como se muestra en la Figura 3.17.

```

#include "thingProperties.h" // Incluye las definiciones y funciones relacionadas con las propiedades de IoT Cloud

// Variable de calibración utilizada para ajustar el cálculo del pH
float calibration_value = 11.34;
// Variable temporal para almacenar una lectura de pH (aunque no se usa explícitamente en el código)
int phval = 0;
// Variable para almacenar la suma de valores del arreglo buffer_arr (para calcular el promedio)
unsigned long int avgval;
// Arreglo para almacenar las lecturas del sensor de pH y variable temporal para el ordenamiento de esas lecturas
int buffer_arr[10], temp;

void setup() {
  // Inicializa la comunicación serial a 9600 bps para la depuración y monitorización de datos
  Serial.begin(9600);
  // Espera 1.5 segundos para permitir que la consola serial se conecte, si está disponible
  analogReadResolution(12);

  delay(1500);
}

```

**Figura 3.17** Primera parte del código

En la Figura 3.18. Se muestra la siguiente parte del código en el cual utilizamos la función `"ArduinoCloud.begin(ArduinoIoTPreferredConnection)"` para establecer la conexión con Arduino IoT Cloud. También se configura el nivel de mensajes de depuración con `"setDebugMessageLevel (2)"` y se imprimen la información de depuración con `"ArduinoCloud.printDebugInfo()"`.

```

// Defined in thingProperties.h
initProperties();
// Connect to Arduino IoT Cloud
ArduinoCloud.begin(ArduinoIoTPreferredConnection);
/*
The following function allows you to obtain more information
related to the state of network and IoT Cloud connection and errors
the higher number the more granular information you'll get.
The default is 0 (only errors).
Maximum is 4
*/
setDebugMessageLevel(2);
ArduinoCloud.printDebugInfo();
}

```

**Figura 3.18** Iniciación del código

La Figura 3.19, muestra un fragmento de código en Arduino que pertenece a la función `"loop ()"`, se realiza la lectura del sensor de turbidez y la cual se ejecuta de manera continua. Dentro de esta función, se realiza una lectura analógica del valor en el pin 34 utilizando `analogRead (34)` y este valor se almacena en la variable turbidez. Posteriormente, se realiza un mapeo de los valores obtenidos como 0 a 4095 para un sensor que utiliza una resolución de 12 bits. En este caso, el mapeo se realiza para adaptar los valores de entrada al rango de salida del dispositivo. Mapear el valor de

turbidez de esta manera que se asegura una correcta calibración y ajuste de los valores del sensor. A continuación, se llama a la función `"readph ()"` para leer el sensor de pH y calcular el valor de pH. Posteriormente, se actualiza la conexión con Arduino IoT Cloud con `"ArduinoCloud.update()"` y se imprimen los valores de turbidez y pH en la consola. La función `map ()` en Arduino se emplea para transformar un número de un rango determinado a otro diferente. Esto significa que el valor de turbidez, originalmente en un rango de 0 a 4095 (el rango de del conversor analógico digital de 12 bits del ESP32), se convierte a un nuevo rango de 4550 a 0, que es el valor estándar de salida del sensor de turbidez en unidades NTU.

```
void loop () {  
  turbidez = analogRead(34);  
  turbidez=map(turbidez,0,4095,4550,0);  
}
```

**Figura 3.19** Función void loop código

En la función `"readph ()"`, se realiza una lectura del sensor de pH y se almacena en un arreglo. Luego, se ordenan los valores del arreglo y se calcula el promedio de los valores centrales. Basado en la lectura analógica del sensor de pH, se calcula el valor de pH utilizando una fórmula específica como se presenta en la Figura 3.20.

```
readph ();  
//ph = analogRead(35);  
ArduinoCloud.update();  
// Your code here  
Serial.print("Turbidez:");  
Serial.println(turbidez);  
Serial.print("PH:");  
Serial.println(ph);
```

**Figura 3.20** Función readph

El microcontrolador lee diez valores del pin analógico 35, guardándolos en el *array buffer\_arr* con una pausa breve entre cada lectura. Luego, ordena estos valores de menor a mayor utilizando el algoritmo de ordenamiento el cual toma 10 valores y elimina los 2 de extremo para tener así mayor precisión en la lectura. A continuación, se calcula el promedio de los valores centrales del *array* para minimizar el impacto de posibles valores atípicos. Convierte el valor analógico del pin 35 a voltaje, considerando un voltaje de referencia de 5 (V) y una resolución de 12 bits. Usa una fórmula de calibración como se muestra en la Ecuación 3.1 , ajustada con *calibration\_value*, para calcular el valor de pH y asigna este valor calculado a la variable pH, como se ilustra en la Figura 3.21.

```

void readph() {
    // Lee el valor del sensor de pH diez veces y almacena las lecturas en buffer_arr
    for (int i = 0; i < 10; i++) {
        buffer_arr[i] = analogRead(35); // Lee el valor analógico del pin 35
        delay(30); // Espera 30 ms entre lecturas para estabilizar las medidas
    }

    // Ordena el arreglo buffer_arr en orden ascendente usando un algoritmo de burbuja simple
    for (int i = 0; i < 9; i++) {
        for (int j = i + 1; j < 10; j++) {
            if (buffer_arr[i] > buffer_arr[j]) {
                // Intercambia los valores si el valor actual es mayor que el siguiente
                temp = buffer_arr[i];
                buffer_arr[i] = buffer_arr[j];
                buffer_arr[j] = temp;
            }
        }
    }

    // Inicializa avgval a 0 para comenzar a calcular el promedio
    avgval = 0;
    // Suma los valores del índice 2 al 7 del arreglo ordenado, excluyendo los valores extremos
    for (int i = 2; i < 8; i++)
        avgval += buffer_arr[i];

    // Convierte la lectura analógica a un voltaje, considerando una referencia de 3.3V y una resolución de 12 bits (4095)
    float volt = analogRead(35) * 3.3 / 4095 / 6;
    // Calcula el valor del pH utilizando una fórmula lineal basada en el voltaje y el valor de calibración
    float ph_act = -4.70 * volt + calibration_value;
}

```

**Figura 3.21** Arreglo buffer del sensor pH

$$y = mx + b$$

**Ecuación 3.1** Calibración del sensor de pH

Donde:

- $y$  : (pH) El valor del pH
- $x$  : (V) Voltaje o lectura del sensor
- $m$  : (pH/V) Factor de conversión que relaciona el voltaje con el pH.
- $b$  : Corrección o ajuste necesario para que la lectura del sensor sea precisa.

Usando la Ecuación 3.1 Calibración del sensor de pH

se obtiene:

$$y = -4.7x + \text{valor de calibración}$$

Para configurar una conexión a internet para un dispositivo a una red Wi-Fi, primero se incluyen las bibliotecas necesarias para la conexión. Se definen constantes para el nombre de inicio de sesión del dispositivo, la SSID y la contraseña de la red Wi-Fi, así como la clave única del dispositivo. En la función *initProperties()*, se configuran las propiedades de pH y turbidez para que se puedan leer y escribir, especificando que las funciones *onPhChange* y *onTurbidezChange* se llamen cuando estas propiedades cambien. La conexión Wi-Fi se maneja mediante *WiFiConnectionHandler*. por medio de la SSID y la contraseña proporcionada, como se muestra en la Figura 3.22.



Finalmente, en Arduino Cloud se verifica que el dispositivo esté asociado y que las variables estén configuradas para proceder a configurar los widgets en el *dashboard*.

```
// Code generated by Arduino IoT Cloud, DO NOT EDIT.

#include <ArduinoIoTCloud.h>
#include <Arduino_ConnectionHandler.h>

const char DEVICE_LOGIN_NAME[] = "e8090583-e9cb-4813-be5a-7caf4ca97634";

const char SSID[] = SECRET_SSID;           // "Gary"
const char PASS[] = SECRET_OPTIONAL_PASS;  // "dreams100920"
const char DEVICE_KEY[] = SECRET_DEVICE_KEY; // "AP8gIhneQHs6?E?F!06FyVrII"

void onPhChange();
void onTurbidezChange();

float ph;
int turbidez;

void initProperties() {
    ArduinoCloud.setBoardId(DEVICE_LOGIN_NAME);
    ArduinoCloud.setSecretDeviceKey(DEVICE_KEY);
    ArduinoCloud.addProperty(ph, READWRITE, 1 * SECONDS, onPhChange);
    ArduinoCloud.addProperty(turbidez, READWRITE, 1 * SECONDS, onTurbidezChange);
}

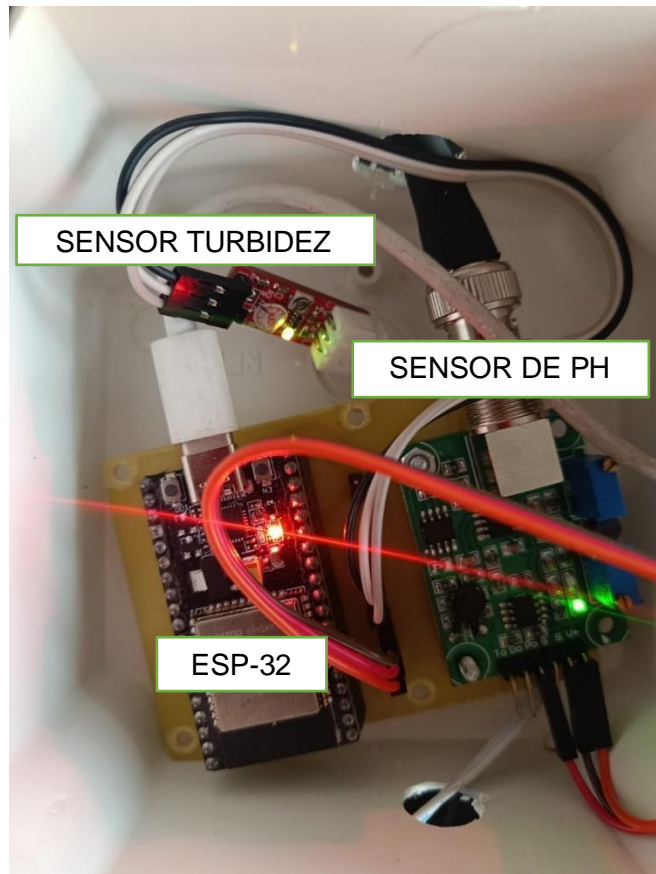
WiFiConnectionHandler ArduinoIoTPreferredConnection(SSID, PASS);
```

**Figura 3.22** Conexión microcontrolador con la página IoT

### **Colocación de los Elementos del Prototipo de Monitoreo de Agua**

La colocación de los componentes en la caja para el prototipo de monitoreo de agua implica organizarlos de manera adecuada para asegurar un funcionamiento óptimo.

Se deben colocar los componentes dentro de la caja de manera apropiada para garantizar su correcto funcionamiento en el prototipo de monitoreo de agua. Es importante organizarlos de forma que haya espacio suficiente y evitando posibles interferencias entre ellos, como se indica en la Figura 3.23.



**Figura 3.23** Posición de los elementos en la caja

Los sensores, como el sensor de turbidez y el sensor de pH, deben colocarse en áreas donde puedan tener acceso directo al agua que se va a monitorear, como se muestra en la Figura 3.24.



**Figura 3.24** Ubicación de los sensores

En relación con el presupuesto del prototipo, la Tabla 3.7 presenta una lista detallada de todos los componentes y materiales empleados en su implementación, junto con sus correspondientes precios.

**Tabla 3.7** Costos de los componentes y materiales.

<b>Componente</b>	<b>Precio unitario (USD)</b>	<b>Cantidad</b>	<b>Precio total (USD)</b>
ESP32	10.00	1	10.00
Sensor de pH	50.00	1	50.00
Sensor de turbidez	23.00	1	23.00
Caja para el prototipo	8.00	1	8.00
Cables y conectores	5.00	1	5.00
Baquelita	10.00	1	10.00
Total		-	106.00
<b>Componente y Material</b>	<b>Precio unitario (USD)</b>	<b>Cantidad</b>	<b>Precio total (USD)</b>
Estaño 1 (m)	1.00	1	1.00
Esponja lana de acero	0.50	2	1.00
Cloruro Férrico	0.60	1	0.60
Brocas 1 (mm)	2.00	2	4.00
Hilo conductor	11.75	1	11.75
Conectores	3.20	3	9.60
Impresión 3D	5.00	1	5.00
Mano de obra	100.00	1	100.00
Total	-	-	133.95

### 3.5 Pruebas de Funcionamiento

#### Verificación de los Sensores

La primera prueba es verificar que los sensores estén en correcto funcionamiento y verificar la comunicación entre estos y el microcontrolador ESP32. Esto implica verificar que los sensores de pH y turbidez estén calibrados y proporcionando lecturas precisas, como se visualiza en la Figura 3.25.

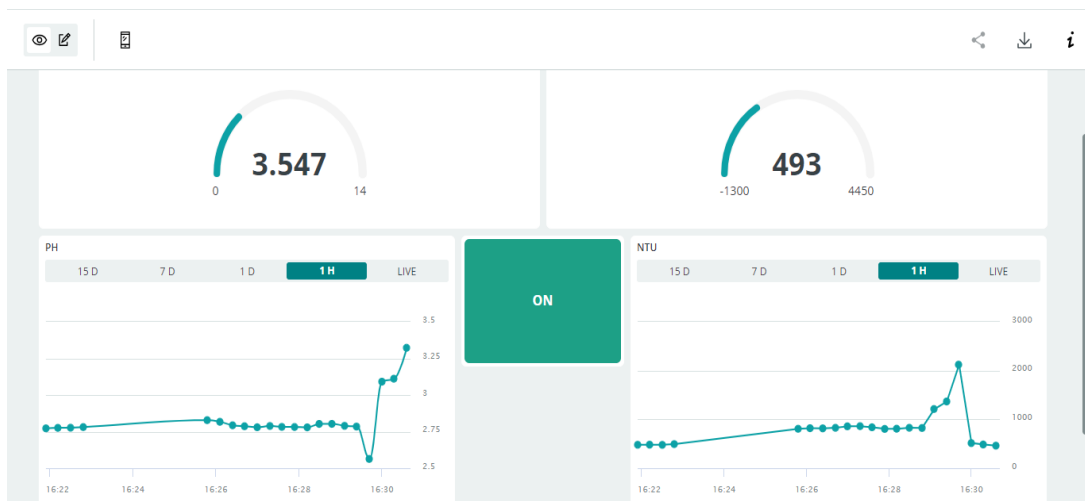
```
PH:9.59
Turbidez:4550
PH:9.59
ArduinoIoTCloudTCP::handle_SyncTime could not get valid time. Retrying now.
Turbidez:4550
PH:9.59
Turbidez:4550
PH:9.59
ArduinoIoTCloudTCP::handle_SyncTime could not get valid time. Retrying now.
Turbidez:4550
PH:9.59
Turbidez:4550
PH:9.59
```

**Figura 3.25** Verificación de lectura de sensores

### Visualización de Datos en la Arduino Cloud

El siguiente paso es la presentación de datos en la plataforma Arduino Cloud, que permite acceder a esta información por Internet y en tiempo real, ya sea mediante aplicación móvil o página web.

Como siguiente paso se verifica que existe transmisión y recepción de datos en tiempo real e histórico tanto en la aplicación móvil como en la web (ver Figura 3.26).

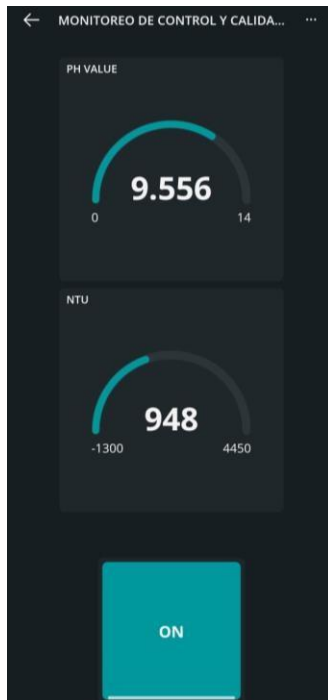


**Figura 3.26** Presentación de datos en la plataforma Arduino Cloud

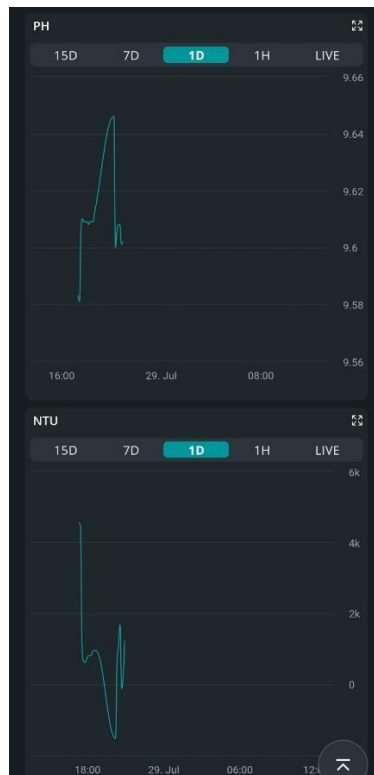
### Visualización en aplicación móvil

Para dispositivos móviles, se puede encontrar la versión de Arduino Cloud en la Play Store o AppStore. Una vez descargada la aplicación, se inicia sesión con la cuenta previamente creada. A continuación, se accede al modo desarrollador de la aplicación,

donde aparecerá la plantilla creada en pasos anteriores. Tal como se muestra la Figura 3.27 y Figura 3.28.



**Figura 3.27** Visualización en la App móvil



**Figura 3.28** Datos históricos en App móvil

En la Tabla 3.8 se presentan todos los criterios evaluados y sus resultados en cuanto al desempeño, los cuales fueron satisfactorios.

**Tabla 3.8** Aspectos considerados en la evaluación del prototipo.

<b>Criterio Evaluado</b>	<b>Funciona</b>	<b>No Funciona</b>
Visualización de datos en aplicación móvil y página web	x	
Lectura del sensor de pH	x	
Lectura del sensor de turbidez	x	
Visualización de datos históricos recopilando lecturas de los sensores	x	

En la Tabla 3.9 se muestran los resultados de las mediciones realizadas a una muestra de agua tanto con el prototipo como con un medidor de calidad de agua comercial. Además, se incluye el porcentaje de error en cada una de las siete pruebas efectuadas. Para el cálculo del error, se utilizó el dato del medidor de calidad de agua como referencia y el dato del prototipo de monitoreo y calidad de agua como valor medido como se indica en la Figura 3.29 y Figura 3.30.

**Tabla 3.9** Resultados obtenidos de las mediciones efectuadas.

<b>Fecha y Hora</b>	<b>Turbidez</b>	<b>pH analógico</b>	<b>pH digital</b>	<b>Error (%)</b>
2024-05-01 08:00:00	1805	7.4	7.5	2.66
2024-06-01 09:00:00	1800	7.4	7.3	1.36
2024-06-01 10:00:00	1808	7.4	7.6	2.63
2024-06-01 11:00:00	1090	4.4	4.5	2.22
2024-07-01 12:00:00	1093	4.4	4.3	2.32
2024-07-01 13:00:00	2054	6.8	6.9	1.44
2024-07-01 14:00:00	2056	6.8	6.5	4.6



**Figura 3.29** Pruebas de funcionamiento



**Figura 3.30** pH digital y analógico

## 4 CONCLUSIONES

- El establecimiento de los requerimientos y funcionalidades del prototipo de forma clara, permitió desarrollar un sistema robusto y eficiente capaz de proporcionar información precisa y en tiempo real sobre la calidad del agua a formar.
- Para el éxito del proyecto fue necesario seleccionar los componentes de hardware apropiados, como sensores de pH y turbidez, y el microcontrolador ESP32 con software de integración en la nube, como Arduino Cloud, ya que estos se integraron de forma óptima, logrando una comunicación fluida entre los componentes, lo que garantizó la recopilación de datos.
- La implementación exitosa del modelo demostró la viabilidad del sistema para monitorear parámetros críticos de calidad del agua. Además, se optimizó el proceso de programación y ensamblaje de componentes, lo que permitió que el sistema fuera automatizado y sostenible, así como proporcionó acceso en tiempo real a los datos a través del tablero.
- Se ha verificado el correcto funcionamiento del prototipo, ya que el porcentaje de error en las mediciones del sensor de pH se mantiene por debajo del 5%. Esto indica que el prototipo es fiable y proporciona resultados consistentes en las pruebas realizadas.
- Las pruebas funcionales confirmaron la precisión y confiabilidad del sistema de monitoreo. Esto fue posible gracias a la capacidad del sistema para detectar e informar cambios en tiempo real en la calidad del agua.
- La plataforma permite realizar un análisis histórico de las mediciones de hasta 15 días, proporcionando al usuario del prototipo una visión detallada y continua de los datos.

## 5 RECOMENDACIONES

- Considerar la integración de sensores adicionales para monitorear más parámetros de la calidad del agua, como temperatura, conductividad y niveles de oxígeno disuelto, los cuales ampliarían las capacidades del sistema y permitiría un análisis más completo de las condiciones del agua.
- Aunque el prototipo ha mostrado un buen rendimiento, es crucial realizar mantenimiento y calibración periódica de los sensores y el equipo para asegurar la precisión continua de las mediciones y evitar cualquier desviación en los resultados.



- Para mantener los sensores en óptimas condiciones y asegurar lecturas precisas, es recomendable limpiarlos cada 20 días con agua destilada. Esta práctica previene la acumulación de polvo y otros residuos que podrían afectar la precisión de las mediciones.

## 6 REFERENCIAS BIBLIOGRÁFICAS

- [1] Redhat, «¿Qué es el Internet de las cosas (IoT)?», 20 enero 2023. [En línea]. Available: <https://www.redhat.com/es/topics/internet-of-things/what-is-iot>. [Último acceso: 24 julio 2024].
- [2] Kaspersky, «¿Qué es el Internet de las cosas? Definición y explicación», 2024. [En línea]. Available: <https://latam.kaspersky.com/resource-center/definitions/what-is-iot>. [Último acceso: 24 julio 2024].
- [3] GSL, «¿Qué es un microcontrolador. Industrias GSL?», 14 julio 2021. [En línea]. Available: [https://industriagsl.com/blogs/automatizacion/que\\_es\\_un\\_microcontrolador](https://industriagsl.com/blogs/automatizacion/que_es_un_microcontrolador). [Último acceso: 24 julio 2024].
- [4] Aula 21, «Arduino: todo lo que necesitas saber», 03 05 2023. [En línea]. Available: <https://www.cursosaula21.com/arduino-todo-lo-que-necesitas-saber/>. [Último acceso: 24 julio 2024].
- [5] Y. Fernández, «Qué es arduino cómo funciona y qué puedes hacer con uno», Xataka, 23 09 2022. [En línea]. Available: <https://xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>. [Último acceso: 24 julio 2024].
- [6] Fundación Aquae, «¿Sabes qué es un Arduino y para qué sirve?», 30 octubre 2020. [En línea]. Available: <https://www.fundacionaquae.org/wiki/sabes-arduino-sirve/>. [Último acceso: 24 julio 2024].
- [7] Sigma Electrónica, «ESP32-WROOM-32D», sigmaelectronica, 26 06 2024. [En línea]. Available: <https://www.sigmaelectronica.net/producto/esp32-wroom-32d/>. [Último acceso: 24 julio 2024].

- [8] Fundación Aquae, «¿Cuál es la calidad perfecta del agua?,» Fundación Aquae, 11 agosto 2021. [En línea]. Available: <https://www.fundacionaquae.org/wiki/calidad-agua/>. [Último acceso: 24 julio 2024].
- [9] ONU, «Progresos en la calidad de las aguas ambientales,» UNWATER, 2021.
- [10] CEPAL, «Estadísticas ambientales y de cambio climático para América Latina y el Caribe,» Naciones Unidas, 2022.
- [11] L. Zarza, «¿Qué es la directiva marco del agua?,» iagua, 23 05 2024. [En línea]. Available: [https://www.iagua.es/respuestas/que-es-directiva-marco-agua#:~:text=La%20Directiva%20Marco%20del%20Agua%20\(DMA\)%20de%20la%20UE%2C,en%20concreto%2C%20las%20cuencas%20hidrogr%C3%A1ficas.](https://www.iagua.es/respuestas/que-es-directiva-marco-agua#:~:text=La%20Directiva%20Marco%20del%20Agua%20(DMA)%20de%20la%20UE%2C,en%20concreto%2C%20las%20cuencas%20hidrogr%C3%A1ficas.) [Último acceso: 24 julio 2024].
- [12] I. Monte, «Agua, pH y equilibrio químico: entendiendo el efecto del dióxido de carbono en la acidificación de los océanos,» SEMS, 2016.
- [13] L. Álava, M. Louison y N. Gallo, «Evaluación de la calidad del agua para consumo humano en la cuenca baja del río Lelía (Santo Domingo de los Tsáchilas – Ecuador),» *Dom Cien*, vol. 7, nº 4, pp. 625-648, 2021.
- [14] D. Carrera y S. Pérez, «Salinidad, fosfatos, nitratos y problemas de infiltración. En las aguas de irrigación del Cantón Milagro, Ecuador,» *Revista Ciencia UNEMI*, nº 9, pp. 85-95, 2013.
- [15] C. Sierra, *Calidad del agua, evaluación y diagnóstico*, Medellín: Universidad De Medellín Ediciones De La U, 2011.
- [16] Celec, «Capítulo 4 Línea Base Ambiental,» 2015, pp. 1-45.
- [17] Instituto de Toxicología de la Defensa, «Protocolo de toma de muestras de agua residual,» 2015. [En línea]. Available: [https://www.defensa.gob.es/itoxdef/Galerias/documentacion/protocolos/ficheros/PROTOCOLO\\_DE\\_TOMA\\_DE\\_MUESTRAS\\_DE\\_AGUA\\_RESIDUAL\\_ver\\_2.pdf](https://www.defensa.gob.es/itoxdef/Galerias/documentacion/protocolos/ficheros/PROTOCOLO_DE_TOMA_DE_MUESTRAS_DE_AGUA_RESIDUAL_ver_2.pdf). [Último acceso: 24 julio 2024].
- [26] DFRobot, "Gravity: Analog turbidity sensor for Arduino," DFRobot.com, <https://www.dfrobot.com/product-1394.html>. [Accedido: 24-jul-2024].

- [18] Wasserlab, «Estándares para la calidad del agua,» Wasserlab, 22 07 2022. [En línea]. Available: <https://www.wasserlab.com/es/agua-pura/estandares-para-la-calidad-del-agua>. [Último acceso: 24 julio 2024].
- [19] EPA, «EPA en español,» US EPA, 01 06 2024. [En línea]. Available: <https://espanol.epa.gov/>. [Último acceso: 24 julio 2024].
- [20] Comisión Europea, «Normas y principios de servicio,» Comisión Europea, 22 07 2024. [En línea]. Available: [https://commission.europa.eu/about-european-commission/service-standards-and-principles\\_es](https://commission.europa.eu/about-european-commission/service-standards-and-principles_es). [Último acceso: 24 julio 2024].
- [21] L. Llamas, «Comparativa ESP8266 frente a ESP32, los SoC de Espressif,» 10 abril 2018. [En línea]. Available: <https://www.luisllamas.es/comparativa-esp8266-esp32/>. [Último acceso: 24 julio 2024].
- [22] Thido, «Programador Nodemcu ESP 32 38 pines WIFI+ Bluetooth 4.2 Ble,» Devsia, s.f. [En línea]. Available: <https://electronicathido.com/detallesProducto.php?id=VEhMVmxLTFIxSnFDQU5JNzAyNTZBZz09>. [Último acceso: 24 julio 2024].
- [23] Endress y Hauser S.A, «Sensores de turbidez del agua y turbidímetros,» s.f. [En línea]. Available: <https://www.es.endress.com/es/instrumentacion-campo/analisis-agua-liquidos-industria/turbidimetros-sensores-turbidez-agua>. [Último acceso: 24 julio 2024].
- [24] Tesla electronic EIRL, «Sensor de turbidez de agua,» Tesla electronic, 07 mayo 2024. [En línea]. Available: <https://www.teslaelectronic.com.pe/producto/sensor-de-turbidez-de-agua/>. [Último acceso: 24 julio 2024].
- [25] Vistronica, «Sensor de Turbidez Analógica para Arduino,» s.f. [En línea]. Available: <https://www.vistronica.com/sensores/sensor-de-turbidez-analogica-para-arduino-detail.html>. [Último acceso: 24 julio 2024].
- [27] DFRobot, "Turbidity Sensor SKU: SEN0189," DFRobot Wiki, [https://wiki.dfrobot.com/Turbidity\\_sensor\\_SKU\\_\\_S](https://wiki.dfrobot.com/Turbidity_sensor_SKU__S). [Accedido: 24-jul-2024].
- [28] EVELta, "SEN0189 Gravity Analog Turbidity Sensor for Arduino," EVELta.com. <https://evelta.com/sen0189-gravity-analog-turbidity-sensor-for-arduino/>. [Accedido: 24-jul-2024].

- [29] DFRobot, "Gravity: Analog turbidity sensor for Arduino," <https://www.dfrobot.com/product-1394.html>. [Accedido: 24-jul-2024].
- [30] DFRobot, "Product page for [DFRobot](#)," sin fecha. [Accedido: 24-jul-2024].
- [31] DFRobot, "Product page for [DFRobot](#)," sin fecha. [Accedido: 24-jul-2024].
- [32] DFRobot, "pH Meter SKU: SEN0161," DFRobot Wiki. [https://wiki.dfrobot.com/PH\\_meter\\_SKU\\_SEN016](https://wiki.dfrobot.com/PH_meter_SKU_SEN016). [Accedido: 24-jul-2024].
- [33] UElectronics, "Sensor de pH líquido," [https://uelectronics.com/producto/sensor-de-ph-liquido/?srsltid=AfmBOooA3529hPAXEzRjfdNv-aQuYxxAoZWdO0e\\_9wFoqczyXISCZeq](https://uelectronics.com/producto/sensor-de-ph-liquido/?srsltid=AfmBOooA3529hPAXEzRjfdNv-aQuYxxAoZWdO0e_9wFoqczyXISCZeq). [Accedido: 24-jul-2024].
- [34] Atlas Scientific, "Atlas Scientific pH Kit: 0-14," <https://www.amazon.com/-/es/Atlas-Scientific-pH-Kit-0-14/dp/B00798612C>. [Accedido: 24-jul-2024].
- [35] Atlas Scientific, "pH Kit," <https://atlas-scientific.com/kits/ph-kit/>. [Accedido: 24-jul-2024].
- [36] Arduino, "Gravity Analog pH Sensor Meter Kit," [https://store.arduino.cc/products/gravity-analog-ph-sensor-meter-kit?srsltid=AfmBOooeNygnQPRGBN4mYtWt\\_9xd\\_j5uD7-1K9llbWKgPFDDMHTK0iyS](https://store.arduino.cc/products/gravity-analog-ph-sensor-meter-kit?srsltid=AfmBOooeNygnQPRGBN4mYtWt_9xd_j5uD7-1K9llbWKgPFDDMHTK0iyS). [Accedido: 24-jul-2024].
- [37] Arduino, "Arduino Cloud," <https://cloud.arduino.cc/>. [Accedido: 24-jul-2024].
- [38] "ThingSpeak," IES La Fuen Santa, <https://sites.google.com/ieslafuensanta.es/iot/proceso/estudio-plataformas-iot/software/thingspeak>. [Accedido: 24-jul-2024].
- [39] Blynk, "Blynk," <https://blynk.io/>. [Accedido: 24-jul-2024].
- [41] Adafruit, "Adafruit IO," <https://io.adafruit.com/>. [Accedido: 24-jul-2024].
- [42] Google Cloud, "Cloud IoT Core," <https://cloud.google.com/iot-core?hl=es-419>. [Accedido: 24-jul-2024].
- [43] Amazon Web Services, "AWS IoT Core," <https://aws.amazon.com/es/iot-core/>. [Accedido: 24-jul-2024].

## ANEXOS

La lista de los **Anexos** se muestra a continuación:

ANEXO I. Certificado de originalidad

ANEXO II. Enlaces

ANEXO III. Código fuente

# **ANEXO I: Certificado de Originalidad**

## **CERTIFICADO DE ORIGINALIDAD**

Quito, D.M. 1 de Agosto de 2024

De mi consideración:

Yo, Andrés Fernando Reyes Castro, en calidad de Director del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE UN PROTOTIPO PARA MONITOREO REMOTO DE CALIDAD DE AGUA Y VISUALIZACIÓN DE DATOS MEDIANTE UN DASHBOARD WEB elaborado por el estudiante Gary Alexander Morales Moreno de la carrera en TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 11%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

LINK

Atentamente,

Andres Reyes

Docente

Escuela de Formación de Tecnólogos

## ANEXO II: Enlaces

Anexo II.I Código QR de la implementación y pruebas de funcionamiento



## ANEXO III: Código Fuente

```
#include "arduino_secrets.h"
```

```
/*
```

```
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
```

```
https://create.arduino.cc/cloud/things/929fd5e7-17f6-41a8-9e63-c9325dee70d9
```

```
Arduino IoT Cloud Variables description
```

The following variables are automatically generated and updated when changes are made to the Thing

```
float ph;
```

```
int turbidez;
```

Variables which are marked as READ/WRITE in the Cloud Thing will also have functions which are called when their values are changed from the Dashboard.

These functions are generated with the Thing and added at the end of this sketch.

```
*/
```

```
#include "thingProperties.h" // Incluye las definiciones y funciones relacionadas con las propiedades de IoT Cloud
```

```
#include <ArduinoloTCloud.h>
```

```
#include <Arduino_ConnectionHandler.h>
```

```
const char DEVICE_LOGIN_NAME[] = "e8090583-e9cb-4813-be5a-7caf4ca97634";
```

```
const char SSID[] = SECRET_SSID; // "Gary"
```

```
const char PASS[] = SECRET_OPTIONAL_PASS; // "dreams100920"
```

```
const char DEVICE_KEY[] = SECRET_DEVICE_KEY; //  
"AP8gIhneQHs6?E?F!06FyVrII"
```

```
void onPhChange();
```

```
void onTurbidezChange();
```

```
float ph;
```

```
int turbidez;
```

```
void initProperties() {
```

```
    ArduinoCloud.setBoardId(DEVICE_LOGIN_NAME);
```

```
    ArduinoCloud.setSecretDeviceKey(DEVICE_KEY);
```

```
    ArduinoCloud.addProperty(ph, READWRITE, 1 * SECONDS, onPhChange);
```

```
        ArduinoCloud.addProperty(turbidez, READWRITE, 1 * SECONDS,  
onTurbidezChange);
```

```
}
```



```

WiFiConnectionHandler ArduinoIoTPreferredConnection(SSID, PASS);

// Variable de calibración utilizada para ajustar el cálculo del pH
float calibration_value = 11.34;
// Variable temporal para almacenar una lectura de pH
int phval = 0;
// Variable para almacenar la suma de valores del arreglo buffer_arr (para calcular el
promedio)
unsigned long int avgval;
// Arreglo para almacenar las lecturas del sensor de pH y variable temporal para el
ordenamiento de esas lecturas
int buffer_arr[10], temp;

void setup() {
  // Inicializa la comunicación serial a 9600 bps para la depuración y monitorización de
datos
  Serial.begin(9600);
  // Espera 1.5 segundos para permitir que la consola serial se conecte, si está disponible
analogReadResolution(12);

  delay(1500);

  // Inicializa las propiedades definidas en thingProperties.h (conexión y variables de la
nube)
  initProperties();

  // Conecta a la plataforma Arduino IoT Cloud utilizando la conexión preferida
ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
  La función setDebugMessageLevel configura el nivel de detalle de los mensajes de
depuración:
  - Nivel 0: Solo muestra errores.
  - Nivel máximo es 4: Muestra la información más detallada.
  */
  setDebugMessageLevel(2);
  // Imprime información detallada sobre el estado de la conexión a la nube y posibles
errores
  ArduinoCloud.printDebugInfo();
}

void loop() {
  turbidez = analogRead(34);
  turbidez=map(turbidez,0,4095,4550,0);
  Serial.print("Turbidez:");
  Serial.println(turbidez);
  Serial.print("PH:");
  Serial.println(ph);
}

```

```

}

void readph() {
  // Lee el valor del sensor de pH diez veces y almacena las lecturas en buffer_arr
  for (int i = 0; i < 10; i++) {
    buffer_arr[i] = analogRead(35); // Lee el valor analógico del pin 35
    delay(30); // Espera 30 ms entre lecturas para estabilizar las medidas
  }

  // Ordena el arreglo buffer_arr en orden ascendente usando un algoritmo de burbuja
  simple
  for (int i = 0; i < 9; i++) {
    for (int j = i + 1; j < 10; j++) {
      if (buffer_arr[i] > buffer_arr[j]) {
        // Intercambia los valores si el valor actual es mayor que el siguiente
        temp = buffer_arr[i];
        buffer_arr[i] = buffer_arr[j];
        buffer_arr[j] = temp;
      }
    }
  }
}

// Inicializa avgval a 0 para comenzar a calcular el promedio
avgval = 0;
// Suma los valores del índice 2 al 7 del arreglo ordenado, excluyendo los valores
extremos
for (int i = 2; i < 8; i++)
  avgval += buffer_arr[i];

// Convierte la lectura analógica a un voltaje, considerando una referencia de 3.3V y
una resolución de 12 bits (4095)
float volt = analogRead(35) * 3.3 / 4095 / 6;
// Calcula el valor del pH utilizando una fórmula lineal basada en el voltaje y el valor de
calibración
float ph_act = -4.70 * volt + calibration_value;
// Asigna el valor calculado a la variable global ph
ph = ph_act;
}

```