CENTRO DE ENSEÑANZA TÉCNICA INDUSTRIAL

DIVISIÓN ELECTRÓNICA

SISTEMAS ELECTRONICOS Y TELECOMUNICACIONES

ANTE PROYECTO DE INVESTIGACIÓN

Sistema para Gestión Sostenible de Ecosistemas Acuáticos

(AQUATIC WATCHER)

AUTOR:

DIEGO MIGUEL ENCISO CONTRERAS

REGISTRO:

21100444

ASESORES:

Ing. Cristina Guadalupe Velázquez Arreola

Mtro. Miguel Enciso Gonzales

Guadalajara Jal,2024, 1 de marzo de 2024

ÍNDICE

AGRADECIMIENTOS

Primero que nada, quiero agradecer a Dios por darme fuerza y salud para terminar mi proyecto. Quiero expresar mi más profundo agradecimiento a mi padre, quien siempre estuvo a mi lado apoyándome a lo largo de esta carrera. Su constante motivación me impulsó a ser una mejor persona y estudiante, enseñándome a no limitar mis retos, sino a desafiar mis propios límites.

También quiero agradecer a mi maravillosa madre, quien me acompañó incondicionalmente durante estos cuatro años. Hubo noches de desvelo, y, aun así, ella siempre estuvo ahí, ofreciéndome su amor y un chocomilk que me daba fuerzas para seguir adelante.

A mi hermana, quien fue la razón principal por la que decidí estudiar en el CETI. Agradezco profundamente sus aportaciones creativas a mis proyectos, su disposición para escucharme en los momentos difíciles de la carrera y su incondicional apoyo, siempre estando ahí para motivarme.

A mi tío y a mi abuelo, quienes son mi mayor inspiración, les agradezco por sus logros y dedicación. Han sido un ejemplo invaluable que me ha motivado a perseverar y alcanzar mis metas. Dedico este proyecto con todo mi corazón a ellos y a todos los que han creído en mí.

También quiero agradecer a toda mi familia por los sacrificios que hicieron para que pudiera culminar mi carrera. Ya sea apoyándome con su tiempo, acompañándome en noches de desvelo, o entendiendo las ocasiones en las que no pude asistir a reuniones familiares o cumpleaños debido a tareas o proyectos importantes. Gracias por comprender y respaldarme en todo momento.

A mi abuelita, gracias por estar siempre cerca de mí. Durante estos años, su apoyo incondicional fue un refugio. Saber que podía ir a su lado, descansar un poco y sentirme mejor, llenó mis días de alegría y gratitud.

A mis compañeros y amigos, gracias por hacer mi experiencia en el CETI más llevadera. Su compañía y apoyo en momentos de estrés fueron fundamentales para que hoy pueda egresar de esta institución, que ciertamente no es fácil, pero que se convirtió en un lugar más amigable gracias a ustedes.

Fuera del ámbito familiar y amistoso, quiero expresar mi gratitud hacia mis profesores. En especial, a la Ing. [illegible]tian Guadalupe Velázquez Arreola , al Ing. Luis Alejandro Mariscal Gutiérrez y, por último, pero no menos importante, al Ing. Rafel Ernesto Lora Aguilar. Este último no solo me inspiró a ser una mejor persona, sino que también fomentó en mí un pensamiento más crítico y analítico. Gracias a él, hoy soy una persona más lectora y con un interés constante por adquirir nuevos conocimientos que me permitan contribuir a transformar el mundo en el que vivimos.

Problemática

Los cuerpos de agua dulce, como los lagos, son vulnerables a cambios en la calidad del agua que pueden ser causados por actividades humanas (como la agricultura, la industria y el desarrollo urbano) o por fenómenos naturales (como lluvias intensas o sequías). Estos cambios pueden afectar negativamente a las especies que habitan en el lago y alterar los procesos ecológicos fundamentales.

Es crucial destacar las repercusiones de las plantas invasoras, una de ellas es el lirio acuático que se encuentra en el Lago de Chapala. Estas plagas afectan gravemente el ecosistema del lago al reducir la disponibilidad de oxígeno en el agua, lo que conduce a la muerte de peces y otras formas de vida acuática. Además, obstruyen los canales de navegación, dificultan el acceso al agua para actividades recreativas y afectan negativamente la economía local al disminuir el turismo. Por lo tanto, es fundamental implementar medidas preventivas para controlar el crecimiento de estas especies invasoras y preservar la salud del ecosistema.

Lamentablemente, el problema no solo se limita al crecimiento del lirio acuático. En la Huasteca Potosina, ciénegas y lagunas sufren debido al crecimiento descontrolado de vegetación acuática, amenazando la biodiversidad local y el acceso al agua potable. En el Delta del río Colorado, ubicado en Baja California y Sonora, la gestión del agua es crucial debido a la intensa actividad agrícola y los desafíos asociados con la calidad del agua en el delta. La presencia de especies invasoras como el lirio acuático agrava aún más la situación.

Asimismo, en la Península de Yucatán, los sistemas lagunares costeros y los cenotes enfrentan serios problemas con el crecimiento excesivo de vegetación acuática y la contaminación. Estos fenómenos afectan tanto la biodiversidad como las actividades recreativas y turísticas en la región.

Supuesto

Se propone el desarrollo de un dispositivo innovador diseñado para alertar a los usuarios sobre las condiciones del agua, con el objetivo de facilitar la adopción de medidas preventivas que eviten desequilibrios ecológicos. Este dispositivo proporcionará información precisa y en tiempo real mediante una página web sobre parámetros críticos de la calidad del agua, tales como el pH, la temperatura, la turbidez y los sólidos disueltos totales (TDS).

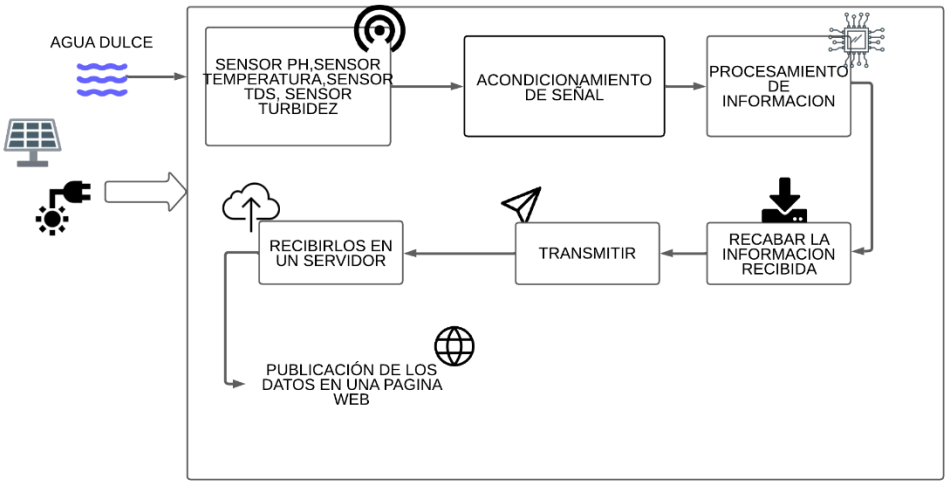
Un pH desbalanceado puede tener efectos devastadores en la vida acuática, ya que muchas especies son extremadamente sensibles a cambios en la acidez o alcalinidad del agua. Del mismo modo, variaciones en la temperatura pueden alterar el metabolismo de los organismos acuáticos, afectar la solubilidad del oxígeno y aumentar la vulnerabilidad de los ecosistemas frente a especies invasoras o patógenos.

La turbidez, que refleja la cantidad de partículas suspendidas en el agua, puede reducir la penetración de la luz solar, afectando así la fotosíntesis de plantas acuáticas y algas. Además, niveles elevados de turbidez pueden ser indicativos de erosión, contaminación o proliferación de algas nocivas.

Por último, los niveles de TDS son un indicador clave de la presencia de contaminantes como sales, metales pesados o nutrientes en exceso, los cuales pueden ser tóxicos para la vida acuática y comprometer la calidad del agua para el consumo humano o agrícola.

Aquatic Watcher es una herramienta versátil que no solo será útil para el lago de Chapala, sino que también podrá aplicarse en cualquier cuerpo de agua dulce. Se optará por el Raspberry Pi 4 debido a su capacidad de integrar pines GPIO (Entrada/Salida de Propósito General), lo que facilita la conexión con una amplia variedad de dispositivos y periféricos electrónicos. Además, una ventaja significativa de este dispositivo es su conectividad Wi-Fi integrada, lo que permite transmitir fácilmente datos a través de Internet.

Diagrama de bloques



- **Fuente de Alimentación:** Suministrar el voltaje y la corriente necesarios para mantener en funcionamiento un sistema automatizado mediante energía renovable.
- **Sensores:** Detectar y medir varios parámetros del agua, incluyendo el nivel de acidez o alcalinidad (pH), la temperatura del agua, la concentración de sólidos disueltos (TDS) y la cantidad de partículas suspendidas en el agua (turbidez).
- **Procesamiento de información:** Analizar y procesar los datos obtenidos de los sensores después de haber ajustado y amplificado la señal para su posterior utilización.
- **Memoria:** Almacenar la información recabada de los sensores en el servidor.
- **Transmisión:** Transferir los datos almacenados a un servidor remoto utilizando diferentes medios de comunicación, como redes cableadas o inalámbricas.
- **Servidor:** Recibir y gestionar la información procedente de los sensores, garantizando su integridad y disponibilidad para su posterior análisis.
- **Publicación:** Mostrar los datos recopilados en una página web accesible al público, facilitando la visualización y comprensión de la información obtenida sobre las condiciones del agua de manera remota.

Aspectos que dan origen al trabajo

El problema del crecimiento excesivo del lirio acuático en cuerpos de agua mexicanos, como el lago de Chapala, tiene sus raíces en la introducción de esta planta fuera de su hábitat. Esta especie, que puede crecer rápidamente en condiciones propicias, forma densas coberturas que obstruyen la luz solar, afectando gravemente la vida acuática al impedir la fotosíntesis de otras plantas y reducir los niveles de oxígeno, lo que causa la muerte de peces y otras especies.

El problema no solo se presenta en el lago de Chapala, sino también en la Ciudad de México, en el lago de Xochimilco. La pérdida de la actividad agrícola debido a la falta de agua ha facilitado la proliferación de asentamientos urbanos irregulares, lo que ha afectado gravemente a las especies de flora y fauna nativas, como el ajolote, las garzas de cuello de reata y la rana Tláloc. Además, esta situación impacta a animales migratorios, como el pato mexicano y otras 165 especies de aves que dependen de los lagos y canales de Xochimilco para su tránsito reproductivo.

Ante esta situación, se propone la creación de un dispositivo para alertar a los usuarios sobre las condiciones del agua. Este dispositivo proporcionaría información detallada sobre la calidad del agua, permitiendo a los usuarios tomar medidas preventivas. Esta solución busca proporcionar los datos para que las personas indicadas puedan tomar decisiones que apoyen a la conservación del agua para las especies endémicas de cada región.

Ilustración 2 Lago de Chapala

Ilustración 1 Lago de Xochimilco

Objetivos

Objetivo General : Diseñar un dispositivo capaz de medir la temperatura, el pH, los sólidos disueltos en el agua y su turbidez, que permita monitorear las condiciones del agua en tiempo real para ser enviados estos datos al servidor y así lograr observar los datos recabados en una página web que permitiría la toma de decisiones y la implementación de medidas correctivas de manera más eficiente.

Objetivos Específicos:

- Diseñar un chasis del prototipo que garantice facilidad de transporte y rápida instalación en diversas ubicaciones, mediante el uso de una impresora 3D para no permitir el ingreso del agua al circuito.
- Implementar un interfaz de 16bits con 4 canales de entrada analógica, mediante un amplificador de ganancia programable y un multiplexor analógico para obtener las lecturas de los sensores.
- Utilizar la interfaz I2C para permitir la comunicación entre los sensores, el ADC, y el microcontrolador.
- Recibir y procesar las señales digitales del ADC utilizando un Raspberry Pi 4 para lograr transmitirlas.
- Crear un circuito de alimentación mediante el uso paneles solares que permita el almacenamiento de energía para alimentar los circuitos.
- Ejecutar Raspberry Pi OS mediante el uso de una herramienta de código abierto para flashear imágenes de disco, permitiendo la configuración y el funcionamiento eficiente del Raspberry Pi.
- Emplear un sistema de recarga de batería para que el Aquatic Watcher almacene energía solar excedente, para alimentar las distintas etapas del circuito.
- Llevar a cabo la utilización de una página web, mediante un servidor, para que cualquier persona interesada o con la necesidad de acceder a los datos pueda visualizarlos de manera remota, sin necesidad de estar físicamente cerca del lago.

Limitaciones y alcances

Limitaciones:

- Debido a que en el prototipo estarán integrados los cuatro sensores, las lecturas de estos deberán ser tomadas a lo largo y ancho según la extensión de la cuenca o el lago donde se desee conocer dichos parámetros
- Factores como la vegetación acuática, la variabilidad del clima y la presencia de vida silvestre podrían interferir con las mediciones de los sensores o dañar el dispositivo.
- La necesidad de proporcionar un mantenimiento constante al dispositivo debido a su exposición a múltiples factores ambientales aumenta el riesgo de posibles daños en su funcionamiento.

Alcances:

- El proyecto permite el monitoreo a distancia de parámetros importantes del agua, como la temperatura, el pH, los sólidos disueltos y la turbidez, lo que proporciona información valiosa sobre la calidad del agua permitiendo utilizarlo en cualquier cuerpo de agua.
- Los datos recopilados por los dispositivos permiten a los responsables tomar decisiones informadas sobre la gestión y conservación del cuerpo de agua, como la implementación de medidas correctivas para controlar el crecimiento de plantas acuáticas no deseadas, como el lirio.
- El proyecto puede utilizarse como una herramienta para sensibilizar a la comunidad sobre la importancia de la conservación del agua y la gestión sostenible de los recursos hídricos, al proporcionar información accesible y comprensible sobre la calidad del agua.
- Los datos recopilados por el sistema pueden ser utilizados para estudios científicos sobre la dinámica de los ecosistemas acuáticos, el impacto del cambio climático y la calidad del agua en general.
- Aquatic Watcher puede ser utilizado en investigaciones u otros proyectos que requieran monitorear las condiciones del agua dulce de forma remota.

Interés técnico y/o científico

Aquatic Watcher tiene como interés mejorar las condiciones de vida acuática al abordar problemas como el crecimiento de plagas y la contaminación química proveniente de empresas. Aquatic Watcher proporciona datos relevantes a través de una red pública accesible para todos. Estos datos pueden ser utilizados para investigaciones científicas o para crear conciencia sobre la importancia de preservar los ecosistemas acuáticos.

Sin embargo, dado que se utilizará en exteriores, donde la conectividad Wi-Fi puede ser inconsistente, se considerará la integración de otro microcontrolador adicional. Este microcontrolador se encargará de conectar un chip que proporcionará acceso a la red celular, similar al funcionamiento de un teléfono móvil. Esto garantizará que el dispositivo tenga acceso a Internet en cualquier lugar donde haya señal celular disponible, simplemente mediante la inserción de una tarjeta SIM con saldo.

Ilustración 3 Placa para conectar al Raspberry para tener conexión a internet mediante sim.

Ilustración 4 Raspberry pi 4

Las unidades de medición que utilizarán los sensores incluyen grados centígrados, partes por millón (ppm), pH, y Unidades de Turbidez Nefelométricas (NTU). A continuación, se explican cada una de estas unidades y su relevancia en el proyecto. La temperatura y el pH son parámetros fundamentales en la evaluación de la calidad del agua. La medición de la temperatura permite comprender las condiciones térmicas del agua, las cuales pueden influir en la vida acuática y en los procesos químicos que ocurren en el ecosistema. Por otro lado, el pH es un indicador crucial del nivel de acidez o alcalinidad del agua, afectando tanto a los organismos que viven en el agua como a la estabilidad de los procesos biológicos y químicos.

Las partes por millón (ppm) son una medida de la concentración de sólidos disueltos en el agua, proporcionando información valiosa sobre la presencia de minerales, sales, y otros elementos disueltos que pueden afectar la calidad del agua. Los sensores de Sólidos Disueltos Totales (TDS, por sus siglas en inglés) miden la concentración de sólidos disueltos en un líquido, generalmente agua. Estos sólidos pueden incluir sales, minerales, metales, cationes, aniones y cualquier otra sustancia que esté disuelta en el agua. El sensor de TDS determina la cantidad total de estas sustancias en partes por millón (ppm) o miligramos por litro (mg/L).

Los sensores de turbidez determinan la concentración de partículas suspendidas en el agua, que pueden incluir sedimentos, microorganismos, materia orgánica e inorgánica. La unidad de medida estándar es la Unidad Nefelométrica de Turbidez (NTU), para tener más en claro que es turbidez o como se mide la turbidez tenemos la siguiente grafica.

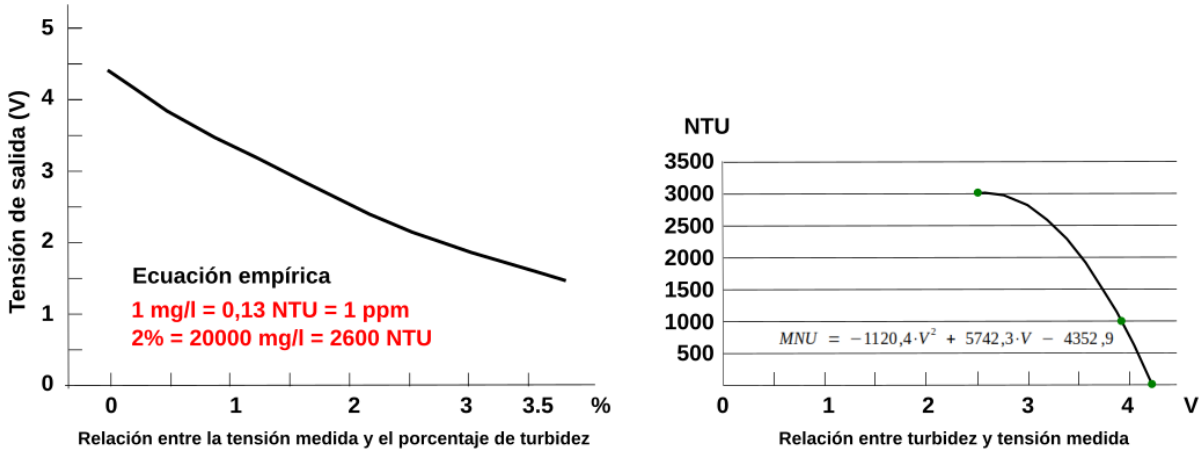


Ilustración 5 Grafica de conversión (NTU)

Impacto

Aquatic Watcher tiene el potencial de generar múltiples impactos positivos en diversas áreas. En primer lugar, al obtener datos de manera constante y lograr monitorear las condiciones del agua, será posible tomar decisiones para el futuro del lago, abordar la contaminación del agua u otras circunstancias relacionadas con la calidad del agua.

Además, el monitoreo y control del crecimiento de especies invasoras, como el lirio u otras plantas acuáticas, podría ayudar a preservar la diversidad biológica y el equilibrio de los ecosistemas acuáticos, promoviendo así la conservación de la vida silvestre y la estabilidad ambiental.

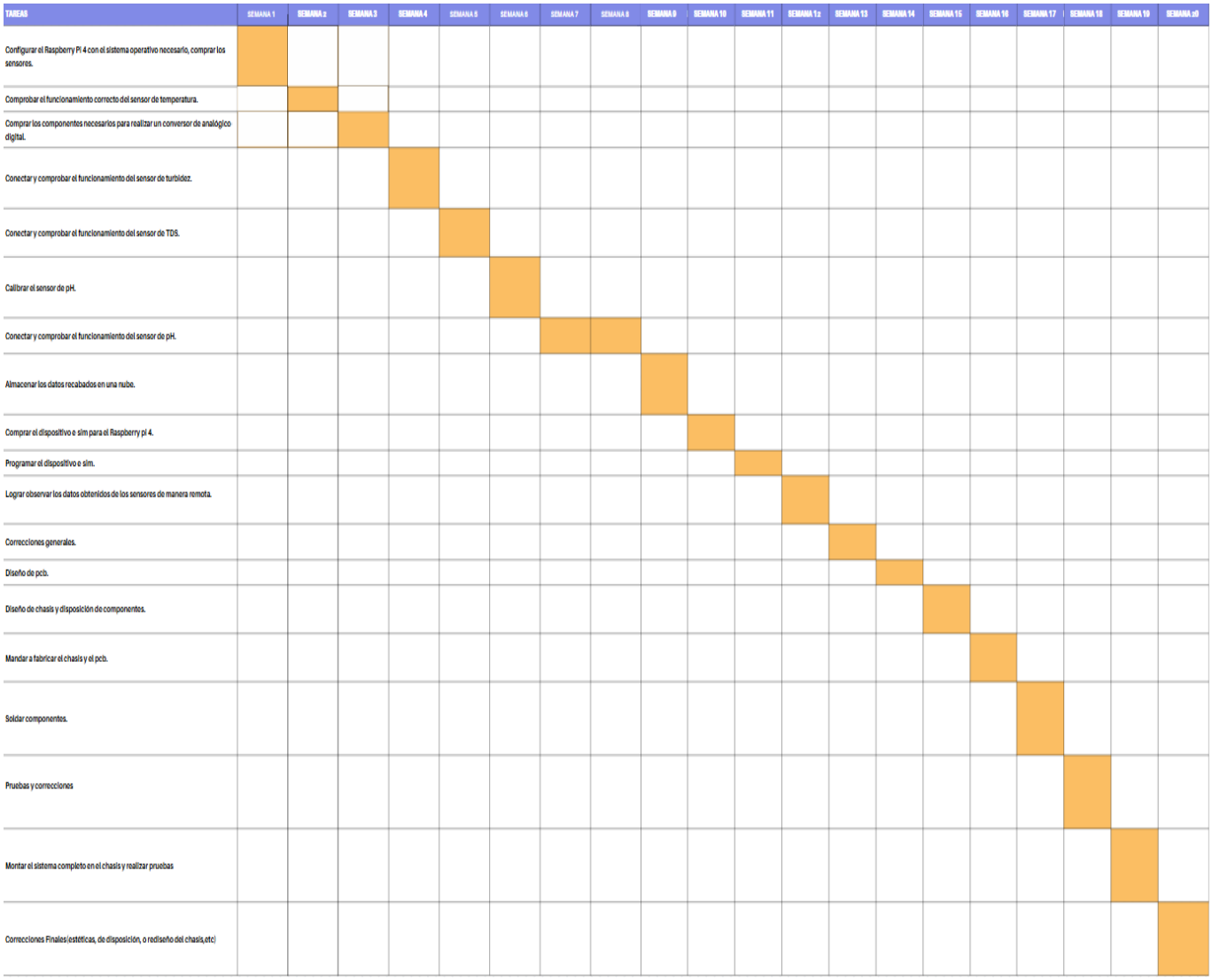
Por otro lado, al proporcionar datos precisos sobre la calidad del agua, el proyecto podría facilitar la toma de decisiones informadas por parte de las autoridades locales y los administradores de recursos hídricos, promoviendo un uso más sostenible de los recursos hídricos y contribuyendo así a la gestión responsable de estos recursos naturales, así como su conservación. Además, los datos recopilados por el dispositivo podrían ser utilizados por científicos e investigadores para realizar estudios a mayor profundidad sobre la dinámica de los ecosistemas acuáticos, así como para analizar los efectos del cambio climático y la contaminación en los cuerpos de agua, lo que ayudaría a avanzar en la investigación científica en este campo.

Finalmente, Aquatic Watcher podría desempeñar un papel crucial en la educación ambiental y la sensibilización pública al aumentar la conciencia sobre la importancia de proteger y conservar los recursos hídricos. Se podría fomentar la conciencia sobre la calidad del agua en nuestro entorno y promover formas de mejorar.

Planeación

Descripción	Semana	Fecha de Inicio	Fecha de finalización
Configurar el Raspberry Pi 4 con el sistema operativo necesario, comprar los sensores, entregar el primer avance del documento de proyecto.	1	20/05/24	24/05/24
Comprobar el funcionamiento correcto del sensor de temperatura.	2	27/05/24	31/05/24
Comprar los componentes necesarios para realizar un conversor de analógico-digital, correcciones del documento de proyecto.	3	03/06/24	07/06/24
Conectar y comprobar el funcionamiento del sensor de turbidez.	4	10/06/24	14/06/24
Conectar y comprobar el funcionamiento del sensor de TDS, realizar la descripción de los capítulos del proyecto.	5	17/06/24	21/06/24
Calibrar el sensor de pH.	6	24/06/24	28/06/24
Conectar y comprobar el funcionamiento del sensor de pH.	7 y 8	05/08/24	16/08/24
Almacenar los datos recabados en una nube, realizar el capítulo número uno del proyecto.	9	19/08/24	23/08/24
Comprar el dispositivo e-sim para el Raspberry pi 4, correcciones del capítulo uno.	10	26/08/24	30/08/24
Programar el dispositivo e-sim, realizar el capítulo 2 del proyecto.	11	02/09/24	06/09/24
Lograr observar los datos obtenidos de los sensores de manera remota, correcciones del capítulo 2 y comenzar el capítulo 3 del proyecto.	12	09/09/24	13/09/24
Correcciones generales.	13	16/09/24	20/09/24
Diseño de pcb, comenzar el capítulo 4 y 5 del proyecto.	14	23/09/24	27/10/24
Diseño de chasis y disposición de componentes.	15	30/09/24	04/10/24
Mandar a fabricar el chasis y el pcb, correcciones del capítulo 4 y 5.	16	07/10/24	11/10/24
Soldar componentes, realizar el capítulo 6 del proyecto.	17	14/10/24	18/10/24
Pruebas y correcciones	18	21/10/24	25/11/24
Montar el sistema completo en el chasis y realizar pruebas	19	28/11/24	1/11/24
Correcciones Finales (estéticas, de disposición, correcciones del chasis.)	20	04/11/24	08/11/24

Diagrama de Gantt



SÍNTESIS DE CAPITULOS

Capítulo I. DESCRIPCION DEL PROYECTO

En este primer capítulo, vamos a explicar detalladamente por qué y cómo estamos desarrollando este proyecto. También hablaremos sobre los desafíos que enfrentamos, cómo la gente trata de controlar el crecimiento del lirio, o algún otro problema relacionado con el agua para saber su calidad así en los componentes y dispositivos que utilizaremos.

Capítulo II. ANALISIS Y DISEÑO

Dentro de este capítulo, se explorarán los análisis realizados en cada etapa, desde la recopilación de datos hasta los métodos de comunicación que se utilizarán para transmitir esa información. Además, se presentarán los diseños y algunos bocetos preliminares que han sido desarrollados anteriormente.

Capítulo III. PROGRAMACION

En esta sección, se detalla la implementación del sistema operativo para el Raspberry Pi 4, junto con las librerías necesarias y las instrucciones para su instalación. Esto permitirá iniciar la codificación básica para recopilar, transmitir y enviar los datos recibidos por los sensores. Luego, se elaborará un seudocódigo y se presentará el diagrama de flujo para obtener la codificación correspondiente.

Capítulo IV. FUNCIONAMIENTO

En esta sección, nos centraremos en los aspectos fundamentales del funcionamiento general del proyecto. Abordaremos desde la correcta obtención de los datos captados por los sensores, hasta su procesamiento y transmisión eficiente al servidor.

Capítulo V. CONSTRUCCION

En esta parte, nos adentraremos en el diseño en 3D para la fabricación de la estructura y el cableado necesario a su vez abordaremos el cómo es lograr una estructura que flote en el agua de forma segura, sin permitir la entrada de agua. Además, se diseñará un espacio para la colocación del panel solar y las baterías necesarias para garantizar su funcionamiento adecuado.

Capítulo VI. GUIA DE USARIO Y MANTENIMIENTO

En este capítulo introductorio de la guía de usuario y mantenimiento, se proporcionará una explicación clara y concisa del funcionamiento completo de Aquatic Watcher. Se abordará cómo visualizar los datos obtenidos y se ofrecerán pautas sobre el mantenimiento de los componentes. Además, se indicará la frecuencia recomendada para realizar chequeos o pruebas a fin de garantizar su correcto funcionamiento.

Capítulo I.
DESCRIPCIÓN DEL PROYECTO

1.Funcionamiento por etapas

Basándonos en el diagrama de bloques mencionado, desglosaremos cada uno de estos bloques para explicar y describir su implementación, desde la alimentación del dispositivo hasta la visualización de los resultados por parte del usuario en una página web.

Inicialmente, se plantea la alimentación del dispositivo mediante paneles solares, que proporcionarán los 5V necesarios al Raspberry Pi 4 para su funcionamiento continuo. Además, se implementará un método de almacenamiento de energía para asegurar la operatividad del dispositivo incluso en condiciones de cielo nublado o adversidades climáticas.

Una vez que el Raspberry Pi 4 esté correctamente alimentado y funcionando, se conectarán los cuatro sensores: pH, turbidez, sólidos disueltos en el agua (TDS) y temperatura. Estos sensores capturarán los datos, que serán convertidos de analógicos a digitales. Finalmente, el Raspberry Pi 4 procesará la información obtenida.

Posteriormente, se implementarán tablas de conversión (proporcionadas por el fabricante) mediante programación. Estas tablas permitirán interpretar los valores de voltaje obtenidos por los sensores, transformándolos en datos precisos y correctos de las mediciones realizadas. Una vez que los datos hayan sido transformados, se enviarán al servidor mediante el método HTTP POST, lo que permitirá visualizarlos en la página web.

1.1 Sensor de Temperatura

El sensor DS18B20 es un componente digital utilizado para la medición de temperatura. Este sensor fue elegido en mi proyecto debido a su uso del protocolo 1-Wire, el cual es altamente eficiente al necesitar solo un pin de datos para la comunicación. Esto reduce el cableado y simplifica el diseño del sistema, permitiendo la conexión de múltiples sensores en un solo bus, lo que resulta ideal para proyectos que requieren la medición de temperatura en varios puntos simultáneamente, como en el monitoreo de lagos y cuerpos de agua.

Además, el DS18B20 se presenta en una cápsula de acero inoxidable resistente al agua, lo cual es una característica clave para mi proyecto de monitoreo de calidad del agua. Esta resistencia permite que el sensor funcione de manera confiable en entornos húmedos y sumergidos, como los lagos, asegurando lecturas precisas y durabilidad en condiciones adversas.

Ilustración 6 Sensor de Temperatura

1.2 Sensor de Solidos Disueltos en el Agua (TDS)

Este sensor de TDS es ideal para mi proyecto porque soporta un rango de entrada de voltaje entre 3.3V y 5.5V, lo que lo hace compatible tanto con sistemas de control de 3.3V como de 5V. Además, su salida de voltaje analógico (0 ~ 2.3V) facilita la integración con diferentes microcontroladores, permitiendo una lectura precisa y sencilla de los niveles de sólidos disueltos.

Dado que el monitoreo de la calidad del agua requiere mediciones constantes y fiables, este sensor es una opción excelente por su compatibilidad y precisión. La capacidad de detectar fluctuaciones en la concentración de sólidos en el agua ayuda a identificar cambios en la calidad, lo cual es fundamental para la gestión y preservación de los ecosistemas acuáticos.

Ilustración 7 Sensor de Solidos Disueltos en el Agua (TDS)

1.3 Sensor de pH

Elegí este sensor de pH para el proyecto porque las variaciones en los niveles de pH pueden afectar directamente a los ecosistemas acuáticos y a la vida que depende de ellos. Por ejemplo, niveles de pH demasiado bajos o altos pueden resultar perjudiciales para peces y otras formas de vida acuática. Además, las mediciones del pH proporcionan datos clave sobre posibles contaminantes en el agua, como ácidos o productos químicos alcalinos.

Este sensor de pH es ideal para monitorear continuamente la calidad del agua en tiempo real, ya que ofrece una lectura precisa del balance ácido-alcalino del agua. Su capacidad para detectar cambios en la acidez o alcalinidad es fundamental para identificar problemas en los ecosistemas acuáticos y tomar medidas correctivas si es necesario.

Ilustración 8 Sensor de pH

1.4 Sensor de Turbidez

Este sensor fue seleccionado debido a su simplicidad y precisión en la detección de TSS (Total Suspended Solids) o sólidos suspendidos totales. El sensor utiliza un diodo infrarrojo y un fototransistor colocados uno frente al otro, lo que permite medir la dispersión y transmitancia de la luz a través del agua. La luz infrarroja se dispersa al chocar con partículas en el agua, y al medir este fenómeno, el sensor puede determinar cuántas partículas están presentes.

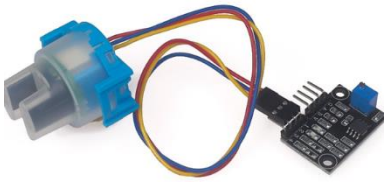


Ilustración 9 Sensor de Turbidez

1.5 Etapa de control

Al adquirir el Raspberry Pi 4, el primer paso es instalar un programa en cualquier computadora para poder cargar el sistema operativo. En este proyecto, se ha decidido utilizar el sistema operativo Raspbian debido a su facilidad de uso, así como su funcionalidad y capacidad para facilitar la comunicación y conectividad necesarias para la operación del sistema.

Ilustración 10 Programa para la instalación del Sistema Operativo

Después de completar la instalación de la aplicación para cargar el sistema operativo, será necesario conectar una tarjeta microSD, preferiblemente de 32 GB. Si se utiliza una tarjeta de mayor capacidad, no habrá ningún problema, pero si se usa una de menor capacidad, podrías experimentar problemas de almacenamiento. Al abrir la aplicación, su interfaz es muy intuitiva; simplemente se debe descomprimir el sistema operativo en la microSD. De esta manera, cuando la tarjeta se conecte al Raspberry Pi 4, este solo tendrá que ejecutar el sistema operativo ya descomprimido.

1.6 Envió de información

El SIM7600A-H 4G HAT es un módulo de comunicación 4G y de posicionamiento GNSS, que soporta LTE CAT4 de hasta 150 Mbps para transferencia de datos. Es muy bajo consumo de energía. Soporta redes 4G LTE, 3G, y 2G, lo que te permite conectarte a internet desde cualquier lugar con cobertura celular. Se conecta directamente a los pines GPIO de la Raspberry Pi 4, Incluye interfaces para UART, USB, e I2C, lo que permite la comunicación con la Raspberry Pi y otros dispositivos.

Ilustración 11 Modulo de conectividad 4GLTE para Raspberry pi 4

1.7 Fuente de alimentación

La fuente de alimentación del sistema se basará en paneles solares capaces de proporcionar una corriente de 3A y un voltaje de 5V, que es lo necesario para el correcto funcionamiento del sistema. Además, se integrará un sistema de carga de baterías para asegurar su operación durante la noche o en condiciones de baja luminosidad, garantizando un suministro de energía continuo y confiable.

Capítulo II.
ANÁLISIS Y DISEÑO

2.Configuración del Raspberry pi 4

Después de haber instalado correctamente el sistema operativo Raspbian en la microSD, insértala en la ranura correspondiente ubicada en la parte trasera del Raspberry Pi 4. Una vez que el sistema operativo esté listo, solo necesitas conectar un teclado, un ratón y un monitor al Raspberry Pi 4 para proceder con la configuración de la tarjeta y del sistema. optimizar este proceso. A continuación, utilizaremos el compilador Genny debido a su sencillez. Este compilador permite ejecutar los programas directamente desde la terminal, lo que facilita y optimiza el proceso de desarrollo.

Ilustración 12.- Arranque inicial de raspberry pi 4

2.1 Entradas

Como entradas, tenemos cuatro entradas físicas censando : pH, turbidez, sólidos disueltos en el agua y temperatura. Estos se conectarán a un convertidor analógico a digital (ADC) de 4 canales, permitiendo asignar cada sensor a una entrada del ADC sin inconvenientes. Luego, la Raspberry Pi 4 interpretará los datos a través de sus pines de entrada. La Raspberry Pi 4 cuenta con pines SCL y SDA, que facilitan la comunicación por medio de I2C y la interpretación de la información proporcionada por el convertidor analógico-digital.

Ilustración 13.- Convertidor Analógico-Digital

2.2 Comunicación

Para transmitir los datos sin depender de una red Wi-Fi y desde cualquier ubicación, se requiere una tarjeta SIM7600 en formato "HAT", que se coloca sobre la Raspberry Pi 4, como se observa en la imagen de referencia. Sin embargo, la comunicación con la Raspberry Pi no se lleva a cabo a través de los pines GPIO. En su lugar, la conexión a internet se obtiene mediante un cable micro USB, que establece la comunicación directa entre la SIM7600 y la Raspberry Pi 4.

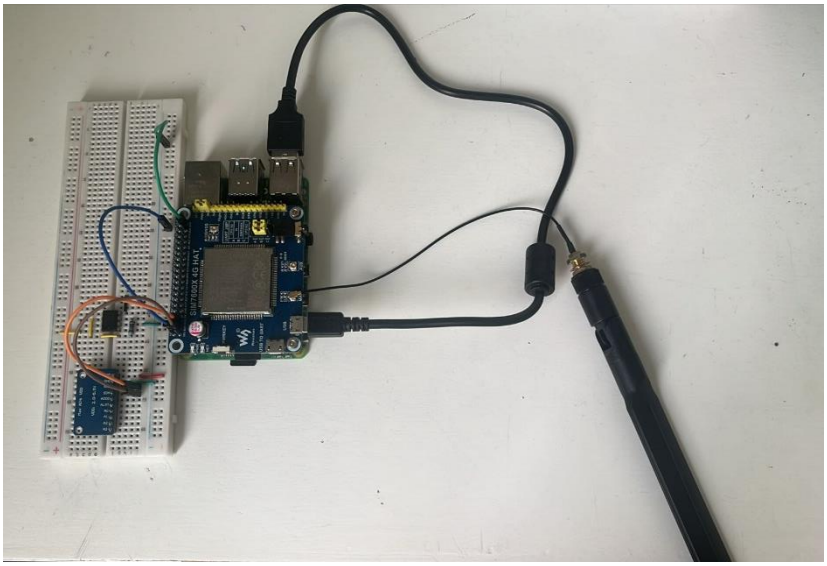


Ilustración 14.- Etapas Acopladas con el convertidor

2.3 Alimentación mediante paneles solares

Para conocer el voltaje y amperaje necesarios para alimentar la Raspberry Pi 4, consultamos su datasheet, la cual indica que se alimenta con 5V y 3A. Esto es crucial, ya que la Raspberry Pi suministra energía tanto a los sensores como a la tarjeta SIM7600. Por lo tanto, es necesario un panel solar que pueda proporcionar al menos ese voltaje. Además, será necesario diseñar un circuito que capture la energía generada y, además de alimentar los sensores, tenga la capacidad de almacenarla para garantizar el funcionamiento durante la noche. Se propone también implementar un sistema de iluminación con alimentación independiente de la Raspberry Pi 4. Este sistema permitirá visualizar el dispositivo durante la noche y ayudará a prevenir accidentes, ya sea con embarcaciones o personas que se encuentren en las cercanías. La iluminación servirá como señal para alertar de la presencia del dispositivo y asegurar su integridad.

2.4 Análisis para utilizar el panel solar

En vista de que la Raspberry Pi requiere 3.4 W de potencia para operar de manera continua las 24 horas del día, y suponiendo que contamos solo con 6 horas de luz solar intensa, el panel solar deberá generar 4 veces más energía durante esas 6 horas para alimentar la batería que mantendrá el sistema funcionando el resto del tiempo. De esta forma, la potencia que el panel solar debe suministrar es de 3.4 W x 4, lo que equivale a 13.6 W. Sin embargo, se decidió adquirir un panel solar de 10 W. Aunque esto deja un déficit de 3.6 W según los cálculos, este problema se resuelve con un banco de baterías que proporcionará la energía adicional necesaria en caso de días nublados.

Ilustración 15.- Panel Solar

2.5 Banco de baterías

El banco de baterías seleccionado es el Pi Sugar 2 Plus, ya que proporciona una salida de 3A con una capacidad de 5000 mAh, lo que ofrece una duración de entre 8 y 10 horas. Se eligió este modelo específicamente porque cuenta con función de UPS (sistema de alimentación ininterrumpida). Esto significa que, cuando la fuente de alimentación externa se apaga, el banco de baterías puede mantener el equipo en funcionamiento sin interrupciones y apagarse de manera segura cuando el nivel de energía es demasiado bajo. Gracias a su capacidad, durante el día se puede cargar utilizando el panel solar, y por la noche, el sistema puede operar de manera autónoma durante 8 a 10 horas con el banco de baterías, lo que lo convierte en una opción ideal para el proyecto Aquatic Watcher.

2.6 Configuración del envió de datos mediante Telcel

Se utiliza una tarjeta SIM para el envío de datos, se configuró la aplicación "Mi Telcel" para gestionar el saldo según las necesidades del usuario. El usuario solo tiene que iniciar sesión en la aplicación como lo haría en cualquier otro dispositivo y seleccionar el plan que mejor se ajuste a sus requerimientos. Por ejemplo, existen planes que por tan solo 15 pesos permiten el envío de datos durante 2 horas. No obstante, también se han estado realizando pruebas con la modalidad "Amigo por Segundo", la cual cobra únicamente por los megabytes utilizados a una tarifa de 0.85 pesos por MB, aplicable tanto en México como en Estados Unidos y Canadá.

Ilustración 16.- Pantalla de inicio de la aplicación mi Telcel

2.7 Tiempo del envió de datos

Considerando que es un prototipo, se optó por enviar los datos cada 5 minutos para demostrar su funcionamiento. Sin embargo, como las condiciones del agua no cambian con tanta rapidez, este intervalo se puede ajustar según las preferencias del usuario, permitiendo envíos cada 1 o 2 horas si es necesario. Durante las pruebas, no es posible reducir el intervalo de envío a menos de 5 minutos, ya que los sensores necesitan primero captar la información para luego enviarla. Si el envío se realizara demasiado rápido, se corre el riesgo de enviar datos incompletos, ya que no todos los sensores habrán terminado de recopilar la información.

2.8 Diseño de los sensores en la entrada

Para calibrar el sensor de pH, es necesario utilizar sustancias especiales conocidas como soluciones buffer, las cuales tienen valores de pH específicos. En este caso, utilizaremos tres buffers con los siguientes valores: 4.00, 6.86 y 9.18. El siguiente paso es conectar el sensor de pH al convertidor ADS (convertidor analógico a digital) para observar las variaciones de voltaje que corresponden a cada valor de pH. De esta manera, podremos obtener la relación entre el voltaje medido y los valores de pH, lo que nos permitirá llegar a una ecuación que describe dicha relación, tomando en cuenta que voltage4 es la salida del sensor.

```
voltage4 = chan2.voltage

pH_value = (voltage4*7)/4.2
```

A continuación, vamos a calibrar el sensor de temperatura. A diferencia de los otros tres sensores, este sensor genera una respuesta digital, por lo que no es necesario conectarlo al convertidor ADS. Para leer los datos del sensor de temperatura, se configuró un pin específico en la Raspberry Pi, utilizando el puerto GPIO4. Los datos del sensor se almacenan en la memoria RAM interna de la Raspberry Pi y luego se asignan a una variable en el código. Este proceso se repite continuamente. Por lo tanto, no fue necesario realizar los mismos pasos de calibración que con los otros sensores.

Después, calibrar el sensor de TDS es un poco más complicado. A diferencia del sensor de pH, no existen soluciones estándar con partes por millón para la calibración. En su lugar, nos basaremos en el datasheet, que explica cómo varía el voltaje en función de las partes por millón. Para la medición de TDS, se obtiene el voltaje del sensor en el canal 3. Si el voltaje es menor o igual a 0.76V, se considera que el nivel de TDS es de 200 ppm. Si el voltaje es mayor o igual a 0.73V, se interpreta que el nivel de TDS es de 400 ppm. En caso de que el voltaje no cumpla con estas condiciones, se asigna un valor de 0 ppm.

Finalmente, para calibrar el sensor de turbidez, es necesario realizar una gráfica que relacione el voltaje con los valores de turbidez obtenidos. A partir de esta gráfica y utilizando la hoja de datos del sensor, se puede derivar una ecuación que permita convertir el voltaje de entrada en un valor de turbidez.

2.9 Diseño del esquemático

Para el diseño del esquemático, se decidió crear una placa similar que pueda acoplarse directamente a los pines GPIO de la Raspberry Pi 4, como si fuera una tarjeta adicional. Esta placa contendrá las pistas necesarias en su parte inferior, permitiendo que se conecte fácilmente a la tarjeta de envío de información. Sin embargo, es crucial tener en cuenta la ubicación del banco de baterías, asegurando que no haya problemas al momento de ensamblar todo el sistema. Además, el chasis debe poder alojar tanto la placa como el banco de baterías sin ocupar demasiado espacio. Por esta razón, se optó por que la tarjeta tenga las mismas dimensiones que la Raspberry Pi 4, maximizando el aprovechamiento del espacio y permitiendo un diseño compacto y eficiente, ideal para ser sumergido en el agua.

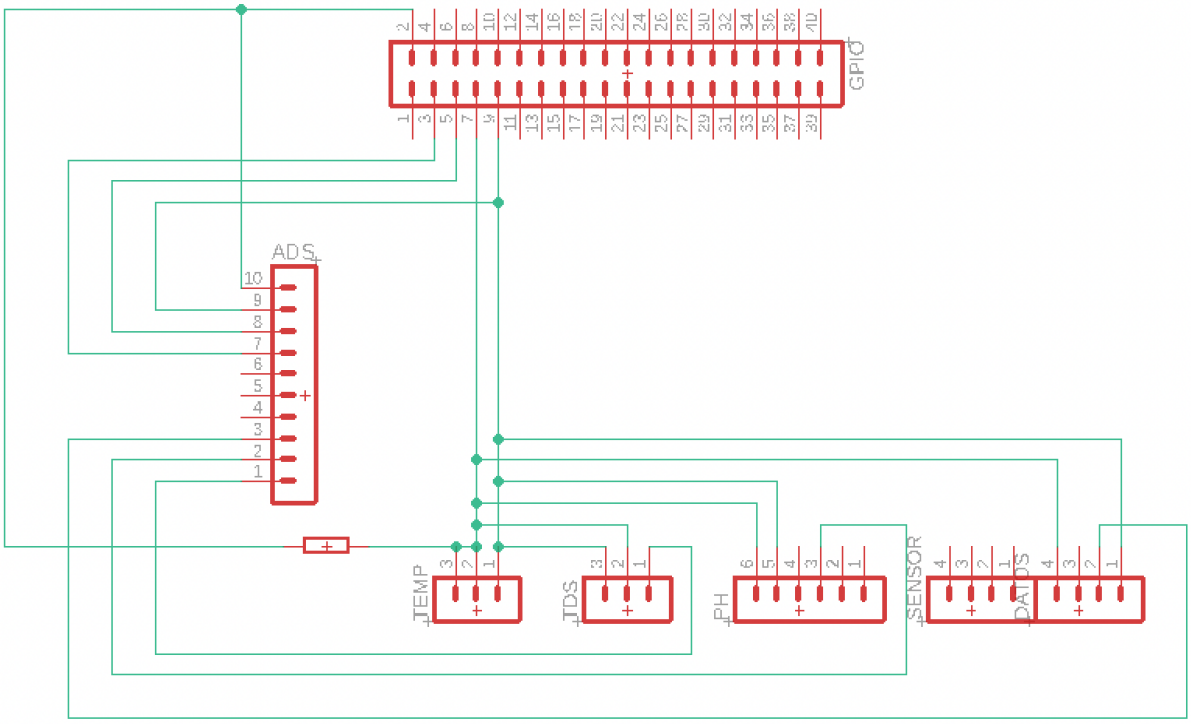


Ilustración 17.- Esquemático de los sensores de entrada

Capítulo III.
PROGRAMACIÓN

3. Algoritmo Principal

1. **Inicialización de hardware:**

- Se crea un bus I2C y un objeto ADC para leer entradas analógicas.
- Se cargan los módulos necesarios para interactuar con un sensor de temperatura.
- Se determina la ruta del sensor de temperatura conectado.

2. **Lectura de entradas analógicas:**

- Se configuran cuatro canales de entrada analógica (chan0 a chan3) para diferentes sensores:
- chan1 mide la turbidez.
- chan2 mide el pH.
- chan3 mide TDS (Total de Sólidos Disueltos).

3. **Subrutinas para los sensores:**

- Temperatura: Lee los datos de un sensor de temperatura a través del archivo del sistema y espera hasta que la lectura sea válida.
- Turbidez: Convierte el voltaje leído del sensor de turbidez a un valor en NTU (Unidades Nefelométricas de Turbidez).
- TDS: Convierte el voltaje leído del sensor en ppm.
- pH: Convierte el voltaje leído en el valor de pH utilizando una fórmula basada en el rango de voltaje.

4. **Bucle principal:**

- Se realiza la lectura de cada parámetro de manera continua:
- Temperatura.
- Turbidez.
- TDS.
- pH.
- Los datos se formatean para tener solo dos decimales.

5. **Envío de datos:**

- Los datos leídos se envían a un servidor web utilizando solicitudes HTTP (GET) con los parámetros necesarios para cada tipo de medición (temperatura, turbidez, TDS, y pH).
- Si la solicitud es exitosa, imprime un mensaje de confirmación. Si no, imprime un error.

6. **Repetición del ciclo:**

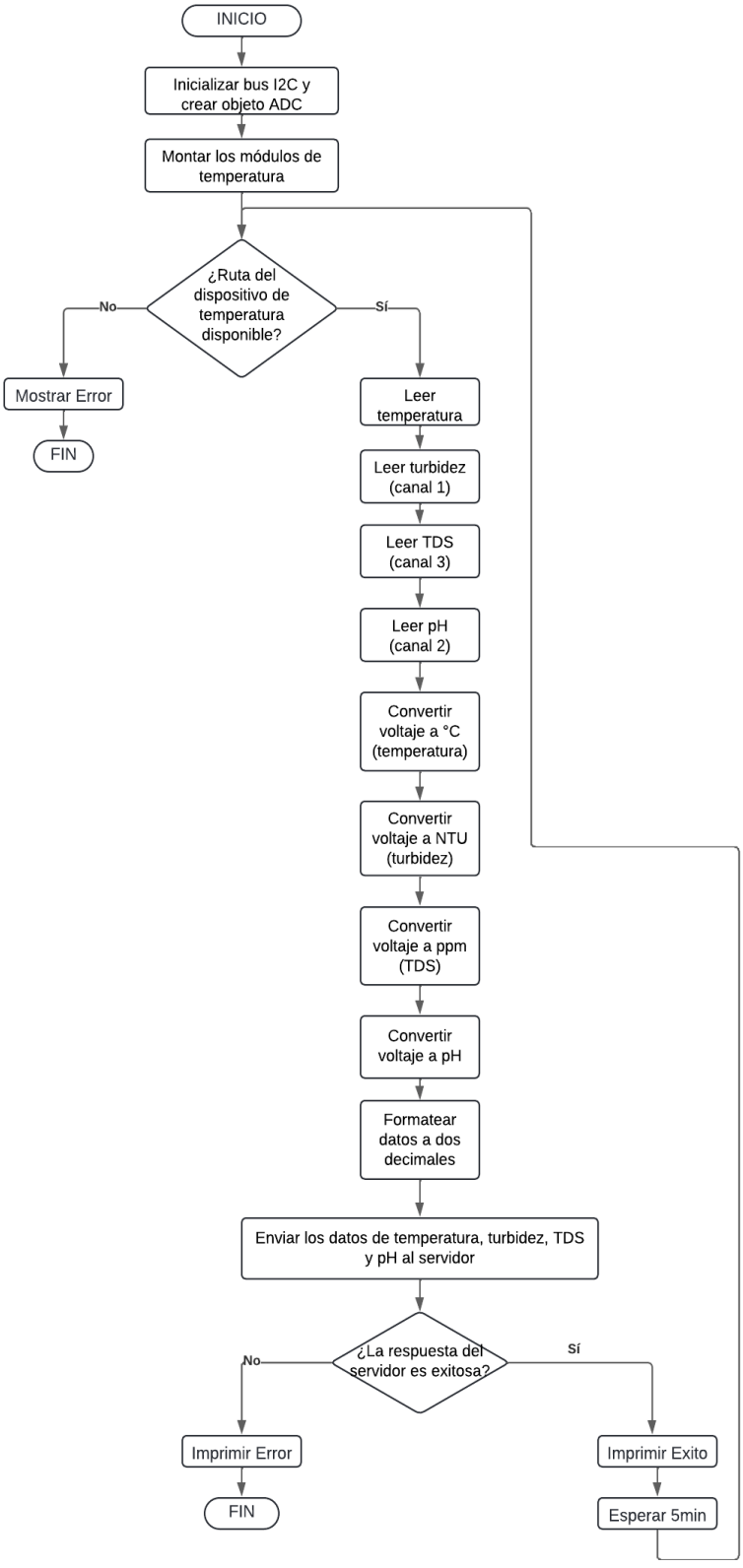
- Después de enviar los datos, el sistema espera 300 segundos (5 minutos) antes de volver a realizar las mediciones y enviar una nueva ronda de datos.

El algoritmo se puede desglosar en los siguientes pasos:

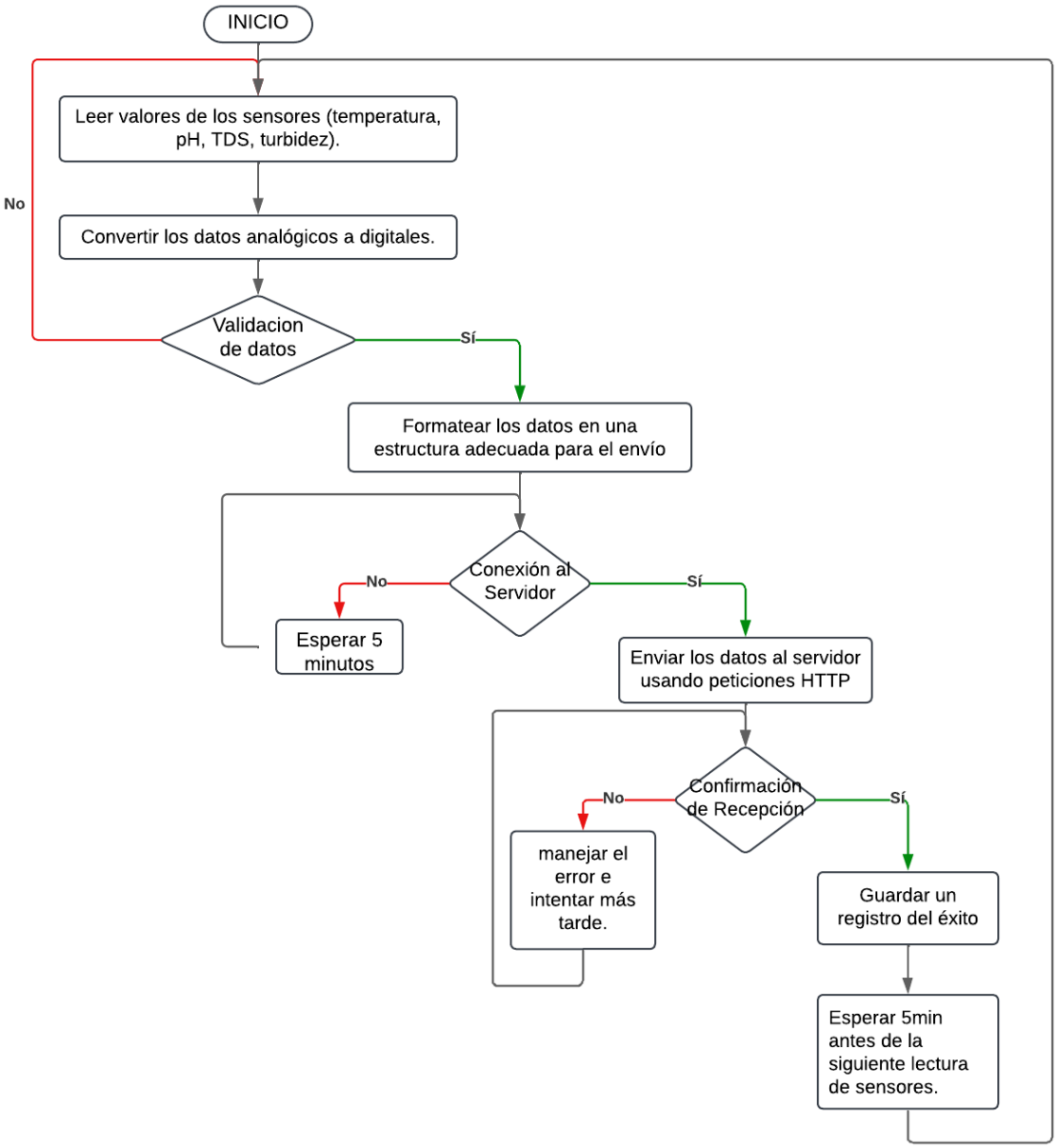
1. Inicializar bus I2C y el ADC.
2. Configurar los módulos para la lectura de temperatura.
3. Leer los datos de temperatura.
4. Leer los valores de turbidez, TDS, y pH.
5. Convertir las lecturas a sus respectivas unidades.
6. Formatear los datos para enviarlos al servidor.
7. Enviar las lecturas al servidor web mediante solicitudes GET.
8. Esperar 5 minutos y repetir el proceso.

Es un ciclo continuo que monitorea y reporta las condiciones ambientales en intervalos definidos.

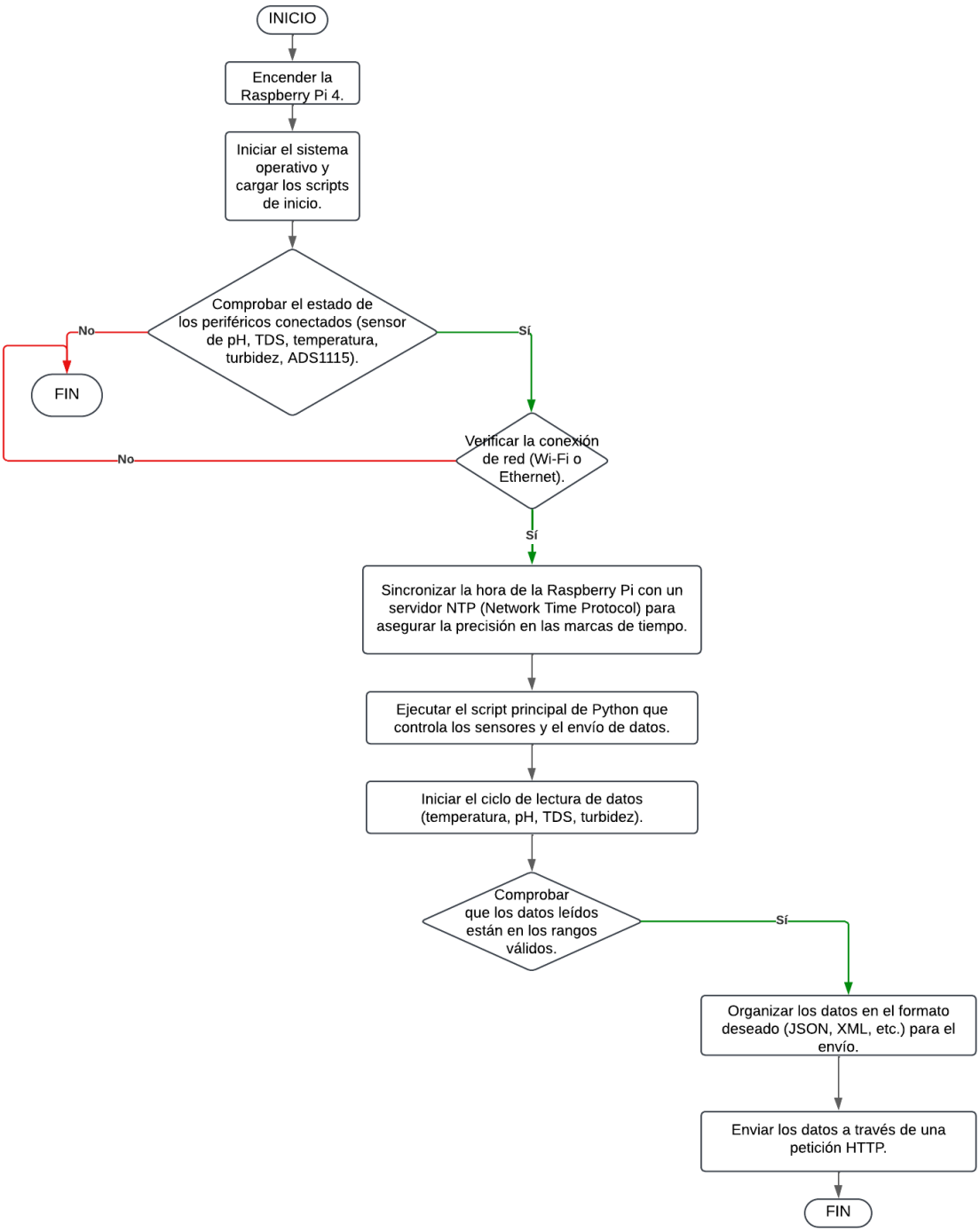
3.1 Diagrama de flujo principal



3.2 Diagrama de flujo de envió de datos



3.3 Diagrama de flujo de Inicio del Rasberry pi 4



3.4 Pseudocódigo Envió de datos

1. Inicialización y Configuración: Se enciende la Raspberry Pi, se carga el sistema operativo y las bibliotecas necesarias.
2. Conexión a Internet: Se verifica la conexión. Si no está disponible, se intentan varios intentos de reconexión.
3. Sincronización del Reloj: Se sincroniza el reloj para asegurar precisión en los datos.
4. Lectura de Sensores: Los sensores de temperatura, pH, TDS y turbidez son leídos uno por uno.
5. Validación de Datos: Se verifica que los datos estén dentro de rangos válidos.
6. Preparación y Envío de Datos: Se preparan los datos en formato JSON o XML y se envían al servidor.
7. Registro y Espera: Se registra el resultado del envío (éxito o fallo) y se espera antes de la siguiente iteración.

3.5 Pseudocódigo Inicio de la Rasberry pi 4

1. Encendido de la Raspberry Pi: La Raspberry Pi se enciende y carga el sistema operativo.
2. Verificación de Componentes: Se cargan todas las bibliotecas y módulos necesarios. Si falta algún módulo crítico, se registra un error y el programa se detiene.
3. Conexión de Red: La Raspberry Pi intenta conectarse a la red (Wi-Fi o Ethernet). Si la conexión falla, se intenta reconectar un número definido de veces antes de detener el programa.
4. Sincronización del Reloj: Se sincroniza la hora con un servidor NTP. Si falla, se registra una advertencia, pero el programa continúa.
5. Verificación de los Sensores: Se inicializan todos los sensores conectados. Si algún sensor no responde, se registra un error y el programa se detiene.
6. Configuración del ADS1115: Se configura el convertidor de analógico a digital para recibir datos de los sensores. Si la configuración falla, se registra un error y el programa se detiene.
7. Inicio del Programa Principal: Si todos los pasos anteriores se completan sin errores, se inicia el ciclo de monitoreo y envío de datos.

3.6 Instalación de librerías y/o comandos de ejecución en la terminal

1. Actualizar la Raspberry pi 4:
- sudo apt-get update
- sudo apt-get upgrade

2. Instalar el gestor de paquetes para Python:
- sudo apt-get install python3-pip

3. Instalar la biblioteca board para el Ads1115:
- pip3 install adafruit-circuitpython-bundle

4. Instalar la librería para la comunicación I2C:
- pip3 install adafruit-circuitpython-busdevice

5. Para verificar que la instalación fue exitosa, vamos a ejecutar un pequeño script de prueba. Creando un archivo llamado test_board.py:
   import board
   print(¡"Biblioteca 'board' instalada correctamente!")
   print("I2C:", board.I2C())
   print("SPI:", board.SPI())
6. Para habilitar la interfaz I2C en la Raspberry Pi 4, sigue estos pasos:
- Abre la terminal y ejecuta el siguiente comando: sudo raspi-config
- Se abrirá una ventana con el menú de configuración. Navega hasta la opción Interfacing Options.
- Selecciona I2C y, cuando se te pregunte, elige Enable.
- Una vez habilitado, reinicia la Raspberry Pi para que los cambios surtan efecto: sudo reboot.
7. Instalar el paquete usb-modeswitch, minicom y ppp para habilitar la comunicación mediante la tarjeta SIM7600 y el raspberry pi 4.
   - sudo apt install usb-modeswitch minicom ppp
8. Verificar que el módulo esté correctamente conectado y reconocido por el sistema.
   - Lsusb
   - Se debe ver una línea como esta:
     Bus 001 Device 005: ID 2c7c:0125 Quectel Wireless Solutions Co., Ltd. EC25 LTE modem
   - Verificar los puertos seriales creados por el módulo:
     dmesg | grep ttyUSB
   - Después de verificarlos se debe ver algo como esto:
     [5.086120] usb 1-1.1: GSM modem (1-port) converter now attached to ttyUSB0
     [5.086323] usb 1-1.1: GSM modem (1-port) converter now attached to ttyUSB1
     [5.086476] usb 1-1.1: GSM modem (1-port) converter now attached to ttyUSB2
     [5.086645] usb 1-1.1: GSM modem (1-port) converter now attached to ttyUSB3

9. Conectar la SIM7600 y la interfaz USB de Raspberry Pi a través de un cable USB, a través de la conexión UART también se puede acceder a Internet por marcación como PPP, la velocidad de UART es más lenta, si desea realizar el acceso rápido a Internet 4G, se requieren NDIS y RNDIS para conectar el USB, es decir, es una tarjeta de red USB, como se muestra en la imagen:

Ilustración 18.- Conexión de la Tarjeta

10. SIM7600 módulo se conecta a la Raspberry Pi o Jetson Nano a través del puerto USB, y luego ejecuta el comando para ver si el ttyUSB2 se puede reconocer normalmente. Si es posible, abra el puerto a través de minicom:
    - ls /dev/ttyUSB*
    - sudo apt-get install minicom
    - sudo minicom -D /dev/ttyUSB2

11. Enviar el siguiente comando a través de minicom y, a continuación, esperar a que el módulo se reinicie:
    - AT+CUSBPIDSWITCH=9011,1,1

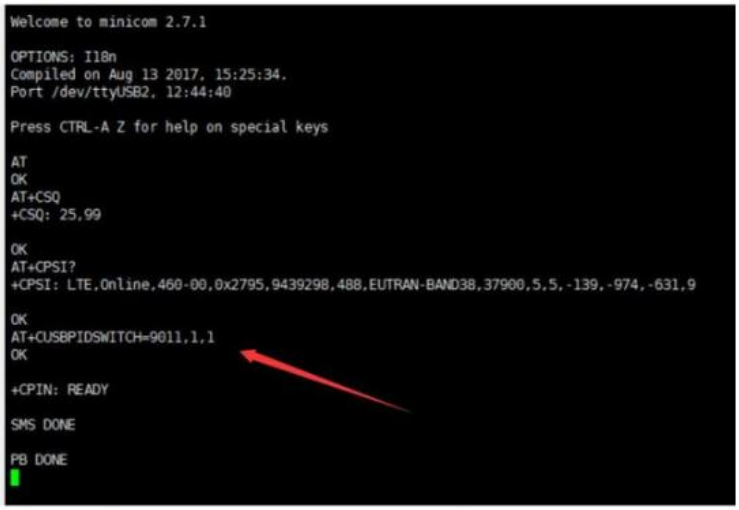


Ilustración 19.- Imagen demostrativa del comando

12. ifconfig para ver si se reconoce una tarjeta de red USB0
    - ifconfig

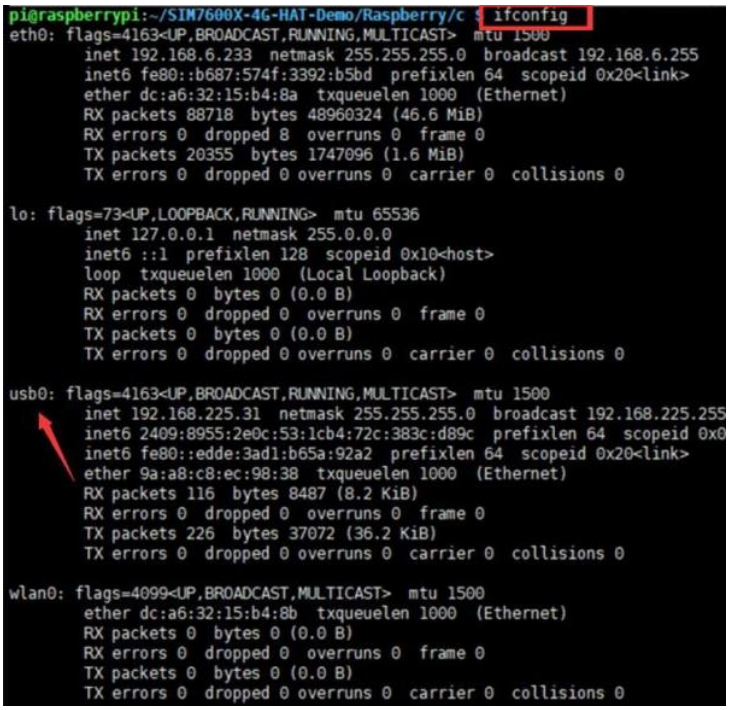


Ilustración 20.- Detección de la tarjeta Sim por medio de USB

13. Mediante el siguiente comando se habilita el uso del internet siempre y cuando si haya detectado la conexión al puerto USB0:
    - sudo dhclient -v usb0

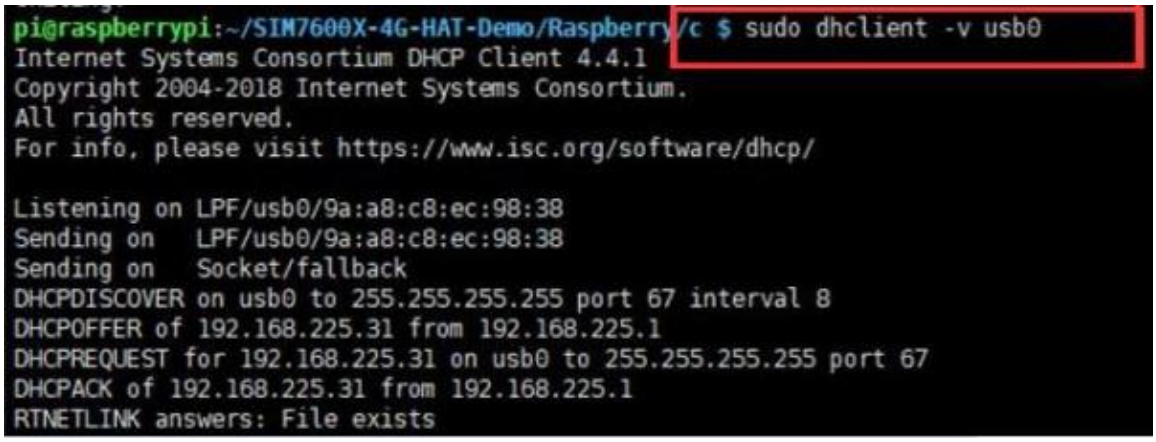


Ilustración 21.- Conexión Exitosa

3.7 Código

1. Primero importamos todas las librerías ya instaladas.

```
import time
import board
import busio
import adafruit_ads1x15.ads1015 as ADS
from adafruit_ads1x15.analog_in import AnalogIn
import requests
import json
import glob
import os
```

2. Configuramos la comunicación IC2 para el ADS1115.

```
# Create the I2C bus%
i2c = busio.I2C(board.SCL, board.SDA)
# Create the ADC object using the I2C bus
ads = ADS.ADS1015(i2c)
```

3. Las siguientes líneas de código configuran el sensor de temperatura, activando la lectura de los pines GPIO y configurándolo como un sensor de temperatura, almacenando los datos en la variable llamada rom.

```
#these tow lines mount the device:
os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')
base_dir = '/sys/bus/w1/devices/'
device_path = glob.glob(base_dir + '28*')[0] #get file path of sensor
rom = device_path.split('/')[-1] #get rom name
```

4. Asignamos las variables para leer las entradas del ADS1115 para lo demás sensores.

```
# Create single-ended input on channel 0
chan = AnalogIn(ads, ADS.P0)
chan1 = AnalogIn(ads, ADS.P1)
chan2 = AnalogIn(ads, ADS.P2)
chan3 = AnalogIn(ads, ADS.P3)
```

5. Esta subrutina interactúa con el archivo del sensor de temperatura, que está disponible en el sistema de archivos de la Raspberry Pi cuando está activado el protocolo 1-Wire. El archivo contiene tanto el estado como la temperatura en formato crudo. El programa verifica que la lectura sea válida y luego procesa la temperatura, convirtiéndola en un formato entendible (grados Celsius).

```
#Subrutine for temperature
def read_temp_raw():
    with open(device_path +'/w1_slave','r') as f:
        valid, temp = f.readlines()
    return valid, temp

def read_temp():
    valid, temp = read_temp_raw()

    while 'YES' not in valid:
        time.sleep(0.2)
        valid, temp = read_temp_raw()

    pos = temp.index('t=')
    if pos != -1:
        #read the temperature .
        temp_string = temp[pos+2:]
        temp_c = float(temp_string)/1000.0
        return temp_c,
```

6. Posteriormente, implementamos la subrutina para leer el sensor de turbidez. Primero, se obtiene la lectura de la entrada del ADS1115, donde está conectado el sensor. A continuación, se aplica la ecuación para calcular los valores de NTU (Unidades Nefelométricas de Turbidez). La ecuación consiste en restar el voltaje obtenido al voltaje máximo, multiplicar el resultado por 3000 (los NTU máximos) y dividir entre 10.

```
#Subrutine for Turbidity
def voltage_to_ntu(voltage2):
    NTU= (5-voltage2)*3000/(10)
    voltage2=chan1.value
    return NTU
```

7. La subrutina para el sensor de TDS sigue consideraciones similares a las de los otros sensores, ya que también es necesario leer los valores desde el ADS1115. La principal diferencia radica en cómo se procesa la información. En esta parte del código, después de leer el voltaje del canal 3, se establece una condición: si el voltaje es menor o igual a 0.70, se divide entre 0.00171, valor obtenido de la ecuación de la recta. De lo contrario, el voltaje se divide entre 0.0171.

```
#Subrutine for TDS
def read_tds():
    voltage3 = chan3.voltage
    if voltage3 >= 0.70:
        ppm=voltage3/0.00171
        return ppm
    else:
     ppm=(voltage3)/0.0171
     return ppm
```

8. La subrutina para el sensor de pH es similar a la del sensor de TDS. Primero, se lee el voltaje obtenido y luego se aplican condiciones. Si el voltaje es menor o igual a 2.5 V, se multiplica por 2.8; de lo contrario, se multiplica por 3.8. Estos valores se determinaron midiendo el comportamiento de las sustancias de pH y observando la variación del voltaje en función del pH.

```
#Subrutine for pH
def read_ph():
    voltage4 = chan2.voltage
    if voltage4 >= 2.549:
        pH_value =voltage4*2.8
        return pH_value
    else:
     pH_value=voltage4*3.8
     return pH_value
```

9. Antes de entrar en el bucle while, declaramos una variable llamada url, que contendrá la dirección de la página a la que enviaremos los datos obtenidos de los sensores.

```
#Subrutine for Send Data
url = "https://diego.enciso.mx/aqua.asp"
```

10. Todo el código a continuación se ejecuta dentro de un bucle while, ya que es necesario que se repita continuamente. Lo primero que hacemos es leer las respuestas de todos los sensores utilizando las subrutinas correspondientes.

```
while True:
    c = read_temp()
    NTU=chan1.voltage
    ntu_value= voltage_to_ntu(NTU)
    tds_value= read_tds()
    ph=read_ph()
```

11. Después, imprimimos los valores obtenidos de los sensores en la consola. Aunque no es estrictamente necesario, ya que el usuario verificará los datos desde el servidor, considero prudente hacerlo para confirmar que los datos enviados son correctos.

```
while True:
    c = read_temp()
    NTU=chan1.voltage
    ntu_value= voltage_to_ntu(NTU)
    tds_value= read_tds()
    ph=read_ph()
    print("Temperatura:                {:,.3f} ".format(c[0]))
    print("Turbidez:                   {:.2f} NTU".format(ntu_value))
    print("Total de Solidos Disueltos: {:.2f} ppm".format(tds_value))
    print(f"pH:                          {ph:.2f}")
```

12. Al imprimir los datos en la consola, procedemos a formatear los valores obtenidos, ya que los sensores devuelven demasiados decimales. Utilizando el comando formatter, reducimos los decimales a solo dos, lo que hace los datos más precisos y exactos.

```
while True:
    c = read_temp()
    NTU=chan1.voltage
    ntu_value= voltage_to_ntu(NTU)
    tds_value= read_tds()
    ph=read_ph()
    print("Temperatura:                {:,.3f} ".format(c[0]))
    print("Turbidez:                   {:.2f} NTU".format(ntu_value))
    print("Total de Solidos Disueltos:  {:.2f} ppm".format(tds_value))
    print(f"pH:                          {ph:.2f}")
    formatted_temperature_c= "{:.2f}".format(c[0])
    formatted_ntu_value="{:.2f}".format(ntu_value)
    formatted_tds_value="{:.2f}".format(tds_value)
    formatted_ph="{:.2f}".format(ph)
    time.sleep(6.5)
```

13. Una vez que los datos son correctos y están formateados, podemos proceder con el envío. Para esto, utilizamos el método HTTP POST, que consiste en incluir los datos necesarios en la URL. Para ello, se crea una variable llamada params, donde construimos la URL con los datos para enviarla y permitir que el servidor los guarde.

```
while True:
    c = read_temp()
    NTU=chan1.voltage
    ntu_value= voltage_to_ntu(NTU)
    tds_value= read_tds()
    ph=read_ph()
    print("Temperatura:                {:,.3f} ".format(c[0]))
    print("Turbidez:                   {:.2f} NTU".format(ntu_value))
    print("Total de Solidos Disueltos:  {:.2f} ppm".format(tds_value))
    print(f"pH:                          {ph:.2f}")
    formatted_temperature_c= "{:.2f}".format(c[0])
    formatted_ntu_value="{:.2f}".format(ntu_value)
    formatted_tds_value="{:.2f}".format(tds_value)
    formatted_ph="{:.2f}".format(ph)
    time.sleep(6.5)
    params = {
    "Maquina": "watcher1",
    "Parametro": "Temperatura",
    "Dato": formatted_temperature_c,
    "Run": "1"
        }
```

14. Es necesario hacerlo con todos los sensores, por lo que creamos cuatro variables llamadas parámetros, cada una correspondiente a los datos de un sensor.

```
while True:
    c = read_temp()
    NTU=chan1.voltage
    ntu_value= voltage_to_ntu(NTU)
    tds_value= read_tds()
    ph=read_ph()
    print("Temperatura:                {:,.3f} ".format(c[0]))
    print("Turbidez:                   {:.2f} NTU".format(ntu_value))
    print("Total de Solidos Disueltos:  {:.2f} ppm".format(tds_value))
    print(f"pH:                          {ph:.2f}")
    formatted_temperature_c= "{:.2f}".format(c[0])
    formatted_ntu_value="{:.2f}".format(ntu_value)
    formatted_tds_value="{:.2f}".format(tds_value)
    formatted_ph="{:.2f}".format(ph)
    time.sleep(6.5)
    params = {
    "Maquina": "watcher1",
    "Parametro": "Temperatura",
    "Dato": formatted_temperature_c,
    "Run": "1"
        }
    params1 = {
    "Maquina": "watcher1",
    "Parametro": "Turbidez",
    "Dato": formatted_ntu_value,
    "Run": "1"
        }
    params2 = {
    "Maquina": "watcher1",
    "Parametro": "PPM",
    "Dato": formatted_tds_value,
    "Run": "1"
        }
    params3 = {
    "Maquina": "watcher1",
    "Parametro": "pH",
    "Dato": formatted_ph,
    "Run": "1"
        }
```

15. Finalmente, construimos la URL junto con los parámetros y la enviamos al servidor. Establecemos un tiempo de espera de 5 minutos. Si la respuesta del servidor es correcta, se imprimirá "Petición exitosa" en la consola; de lo contrario, se mostrará "Error en la petición".

```
response = requests.get (url,params)
response = requests.get (url,params1)
response = requests.get (url,params2)
response = requests.get (url,params3)
time.sleep(300)
if response.status_code == 200:
    print("Petición exitosa!")
    print("Contenido de la respuesta:")
    print(response.text)
else:
  print(f"Error en la petición: {response.status_code}")
```

Capítulo IV.
FUNCIONAMIENTO

4.1 Etapa de alimentación

Estamos utilizando una batería de 3.7V con una capacidad de 5000mAh, lo que nos da un total de 18.5Wh (vatios-hora), esta batería está conectada a un módulo que se conecta a un panel solar, proporcionando energía a las baterías. Cuando las baterías están cargadas, suministran voltaje a la Raspberry Pi 4. Además, los Leds indican el porcentaje de carga de la batería: si los cuatro Leds están encendidos, significa que la batería está completamente cargada, y a medida que se apagan, indica un menor porcentaje de carga. Por último, el panel solar está diseñado para brindar 6W en óptimas condiciones.

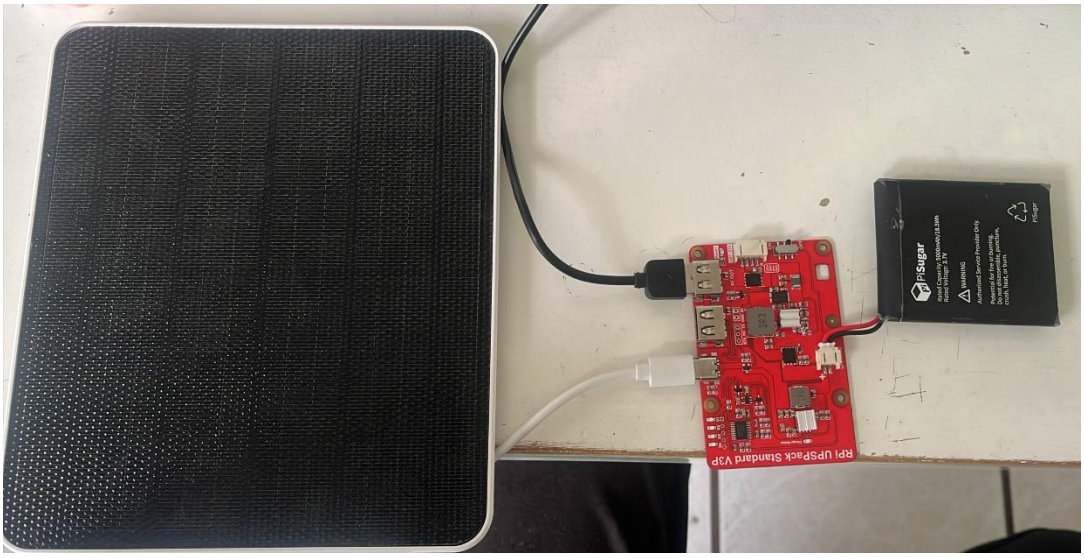


Ilustración 22.- Fuente de Alimentación

4.2 Etapa de Control

Al alimentar la Raspberry Pi, esta se encarga de alimentar el convertidor analógico-digital. Una vez verificado que está recibiendo el voltaje correcto, procedemos a conectar los cuatro sensores: tres al ADS y el sensor de temperatura al pin digital D4 de la Raspberry Pi. Esto nos permite leer las variaciones de voltaje en función de las condiciones ambientales, realizar las ecuaciones necesarias y así obtener los valores correctos, como se muestra a continuación:

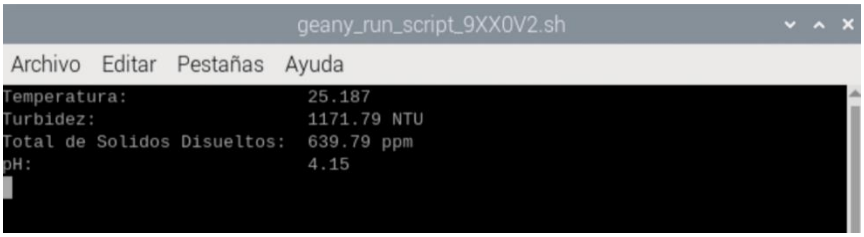


Ilustración 23.- Valores de los Sensores en la Consola

4.3 Envió de Datos

Una vez que los sensores están leyendo los datos, enviamos la información obtenida al servidor. Tras un envío exitoso, se mostrará un mensaje indicando que la petición fue exitosa. Además, verificaremos que los datos enviados coincidan con los que fueron impresos en la consola.

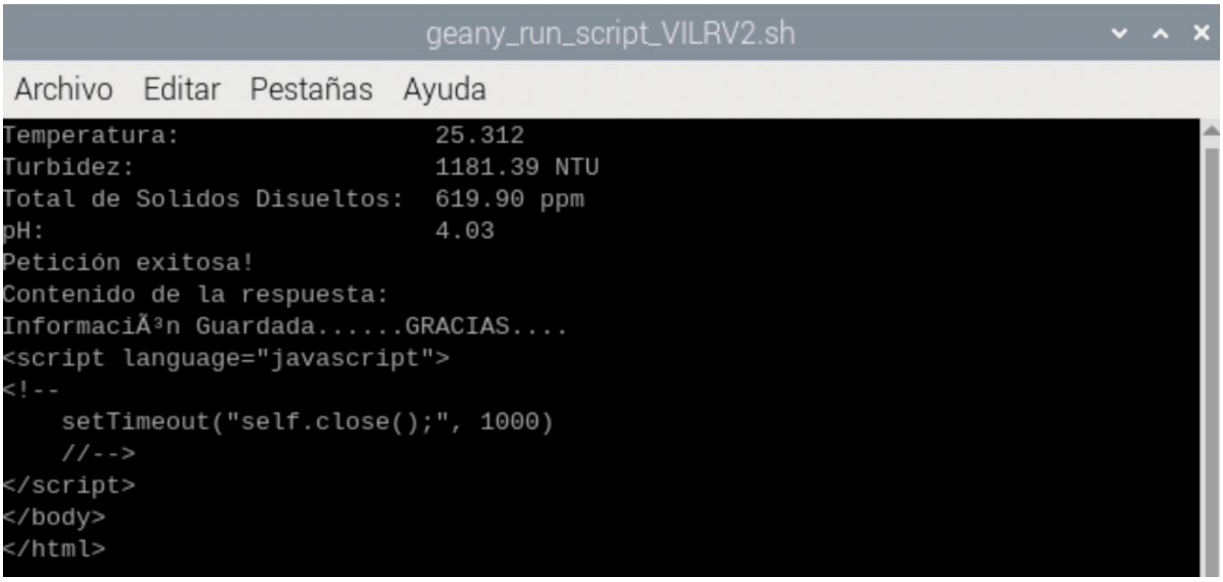


Ilustración 24.-Envió de datos

watcher1	pH	4.03	1	23/10/2024 03:56:10 p. m.	96586
watcher1	PPM	619.9	1	23/10/2024 03:56:10 p. m.	96585
watcher1	Turbidez	1181.39	1	23/10/2024 03:56:09 p. m.	96584
watcher1	Temperatura	25.31	1	23/10/2024 03:56:09 p. m.	96583

Ilustración 25.- Datos obtenidos en el servidor

4.4 Conexión de los sensores

La conexión de los sensores es bastante similar, con la excepción del sensor de temperatura, que tiene una salida digital y se conecta directamente a un pin de la Raspberry Pi 4. Por otro lado, los sensores de turbidez, pH y TDS son analógicos, por lo que se conectan primero a un convertidor analógico-digital (ADS1115). Luego, mediante comunicación I2C, el convertidor envía los datos capturados a la Raspberry Pi 4. Para una representación más clara de estas conexiones, se elaboró un diagrama demostrativo que se muestra a continuación.

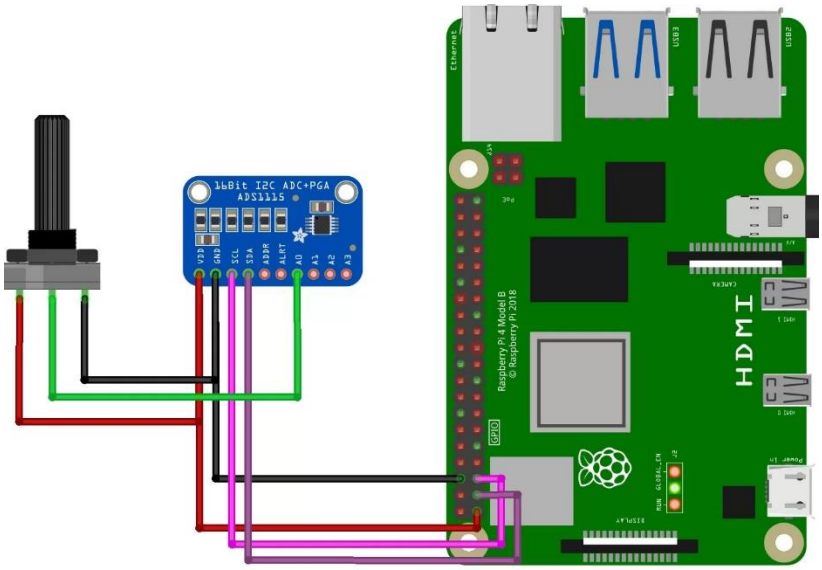


Ilustración 26.- Imagen demostrativa

4.5 Conexión de tarjeta de envío de datos

La principal razón para elegir una Raspberry Pi 4 fue su compatibilidad con una tarjeta HAT equipada con una SIM, lo cual permite enviar datos y mantener una conexión a internet sin inconvenientes, similar a un teléfono móvil. Este HAT con SIM facilita la transferencia de datos a través del puerto USB de la Raspberry Pi 4, garantizando una comunicación eficiente y continua con el servidor remoto.

Ilustración 27.- Conexión de la tarjeta

4.6 Lectura de los Sensores

Para demostrar el correcto funcionamiento de los sensores, los introducimos directamente en el agua para medir la turbidez, las partes por millón, el pH y la temperatura. Luego, enviamos los datos obtenidos, como se muestra a continuación:

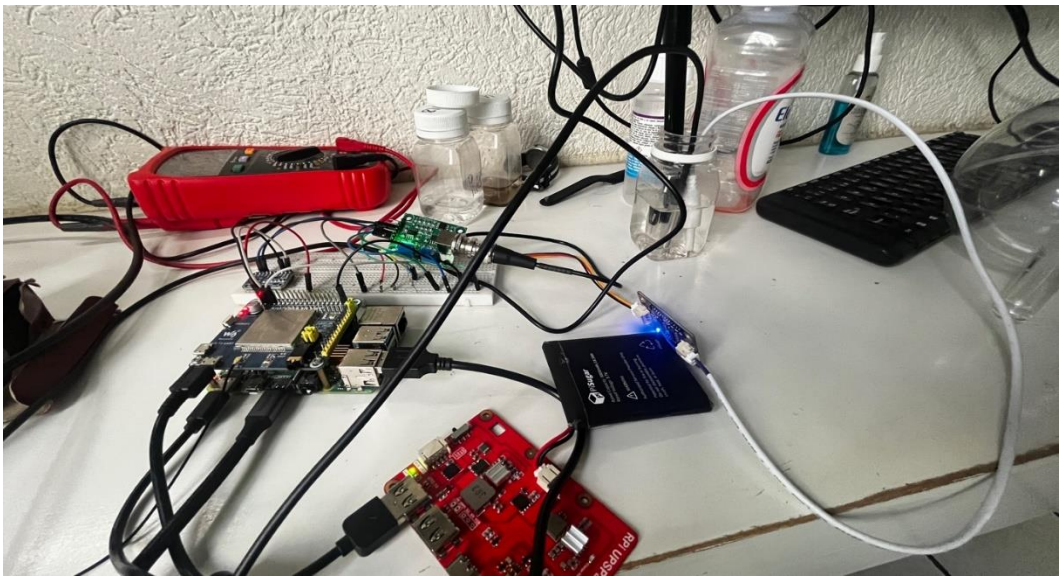


Ilustración 28.- Prueba de los sensores con agua

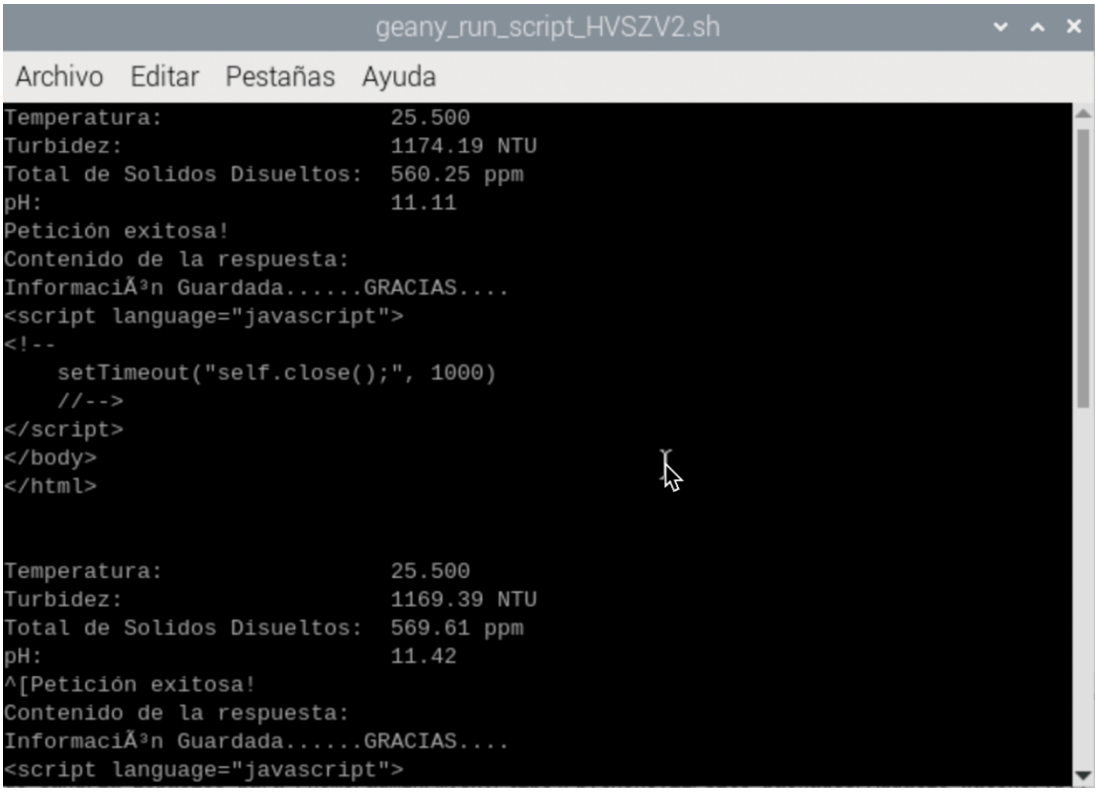


Ilustración 29.- Envió de datos de los sensores

Capítulo V.
CONSTRUCCION

5. Diseño del PCB

Las dimensiones de la tarjeta PCB serán las mismas que las de la Raspberry Pi 4. También agregaremos los 40 pines GPIO de la Raspberry, aunque no se utilicen todos, para mantener la estética y facilitar la conexión de la tarjeta PCB. Comenzamos diseñando la parte inferior (bottom), tomando en cuenta el esquema, y realizamos las siguientes conexiones.

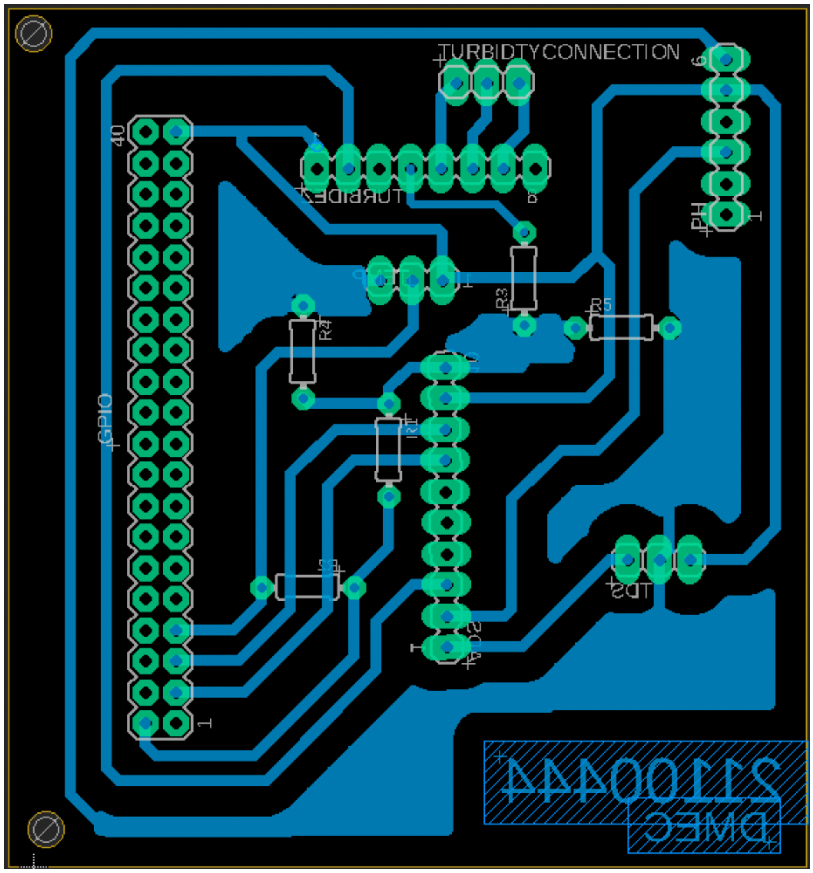


Ilustración 30.- Bottom

Después de haber realizado las conexiones del bottom, realizamos el top:

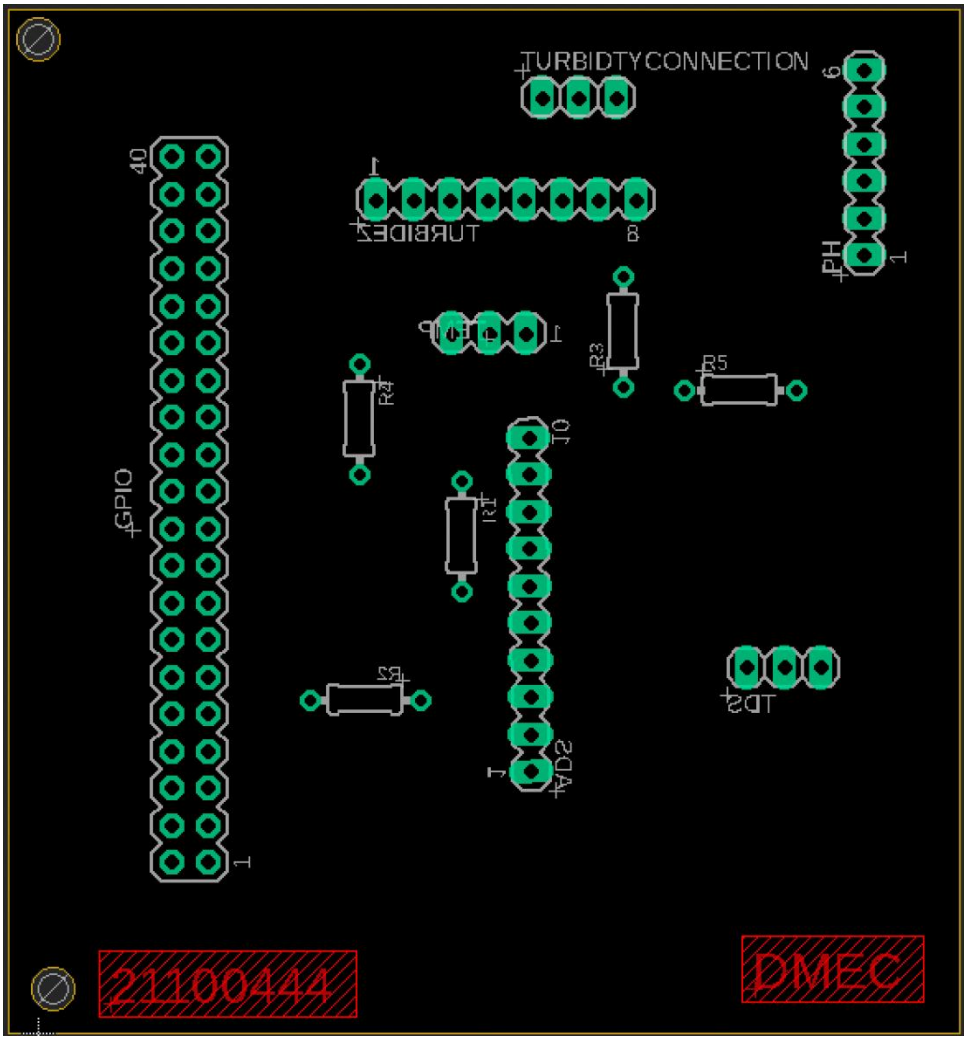


Ilustración 31.- Top

5.1 Diseño del chasis en 3D

Para mantener el dispositivo flotando, se propone utilizar hielo seco, representado en la parte roja de la imagen. En el centro, la esfera azul actuará como un núcleo de soporte, mientras que el panel solar estará conectado al flotador de hielo seco, proporcionando energía de manera eficiente.

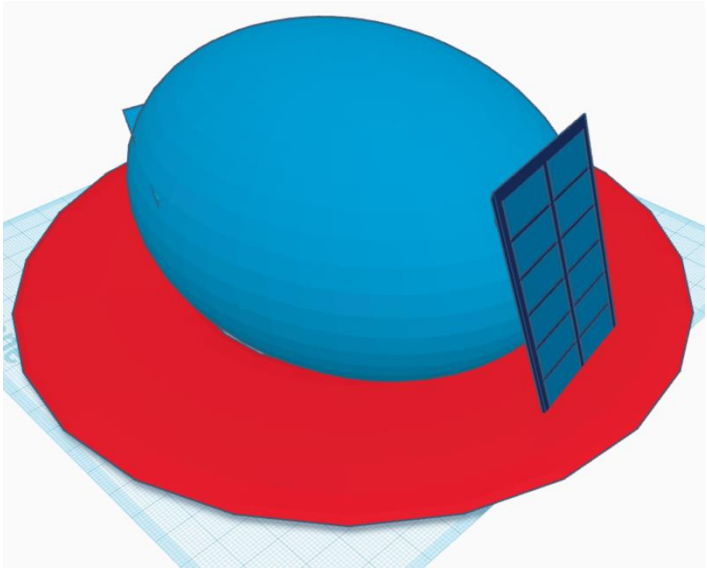


Ilustración 32.- Propuesta de diseño

5.2 Conexión del chasis en 3D

Dentro de la esfera azul se encontrará toda la circuitería, incluyendo la Raspberry Pi, para protegerla del agua y asegurar que el dispositivo se mantenga flotando mientras envía datos al servidor sin problemas. Los sensores saldrán de la esfera a través de un orificio en la parte superior, permitiendo su exposición al entorno para realizar las mediciones necesarias.

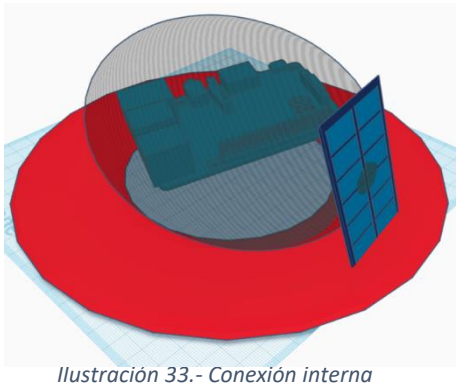


Ilustración 33.- Conexión interna

5.3 Planchado del diseño

Para transferir el diseño previamente elaborado en Eagle, es necesario imprimirlo en blanco y negro sobre papel couché, incluyendo las capas de "top" y "bottom".

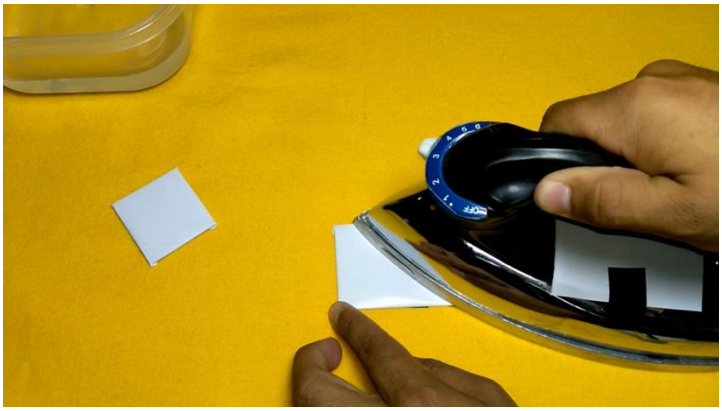


Ilustración 34.- Planchado de Baquelita

5.4 Corroer el circuito impreso

Después de planchar el circuito en la baquelita, preparamos una solución de cloruro férrico con agua caliente a 25 grados Celsius para acelerar el proceso y corroer el cobre más rápidamente.

Ilustración 35.- Baquelita en Cloruro férrico

5.5 Planchar el Top

Después de haber terminado de corroer la baquelita y limpiarla, se procede a verificar la continuidad en cada una de las líneas para asegurarse de que no interfieran entre sí. Luego, se realiza el planchado de la capa superior (top), que corresponde al lado opuesto al que se ha corroído la baquelita.

Ilustración 36.- Baquelita en agua para el diseño del top

5.6 Soldar los componentes a la baquelita

Después de haber finalizado el planchado y la corrosión en ambos lados de la baquelita, podemos soldar cada uno de los componentes. En este caso, para evitar problemas y errores, se han colocado únicamente bases en la tarjeta, de modo que, si algún sensor llega a fallar, sea fácil y rápido reemplazarlo sin inconvenientes.

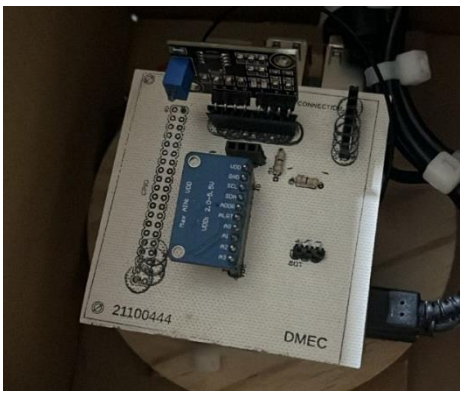


Ilustración 37.- Componentes soldados

5.7 Armado de la estructura interna del chasis

Después de finalizar las pruebas de los sensores, para evitar el movimiento de las tarjetas, las colocaremos en un soporte circular de madera mediante tornillos. Utilizaremos tornillos con bases para fijar las demás tarjetas, tal como se muestra en las siguientes dos imágenes.

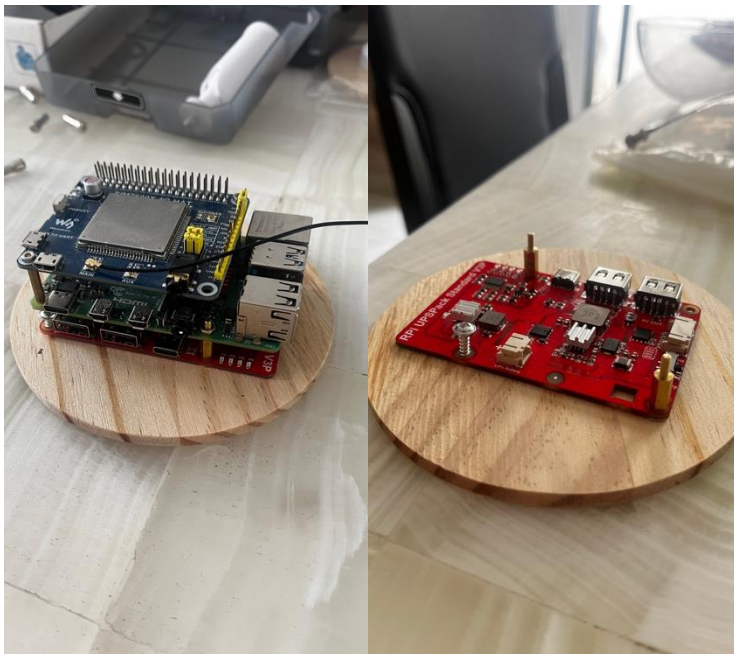


Ilustración 38.- Unión de tarjetas a la madera

5.8 Armado externo de la estructura del chasis

Después de colocar todas las tarjetas, conectamos la batería de alimentación en el lado opuesto de la rueda de madera para mantener la practicidad del diseño. Los cables de alimentación se sujetaron con cinchos para mayor orden y estabilidad. Además, para evitar interferencias, el panel solar se colocó en un soporte de madera separado, alejado del circuito principal.

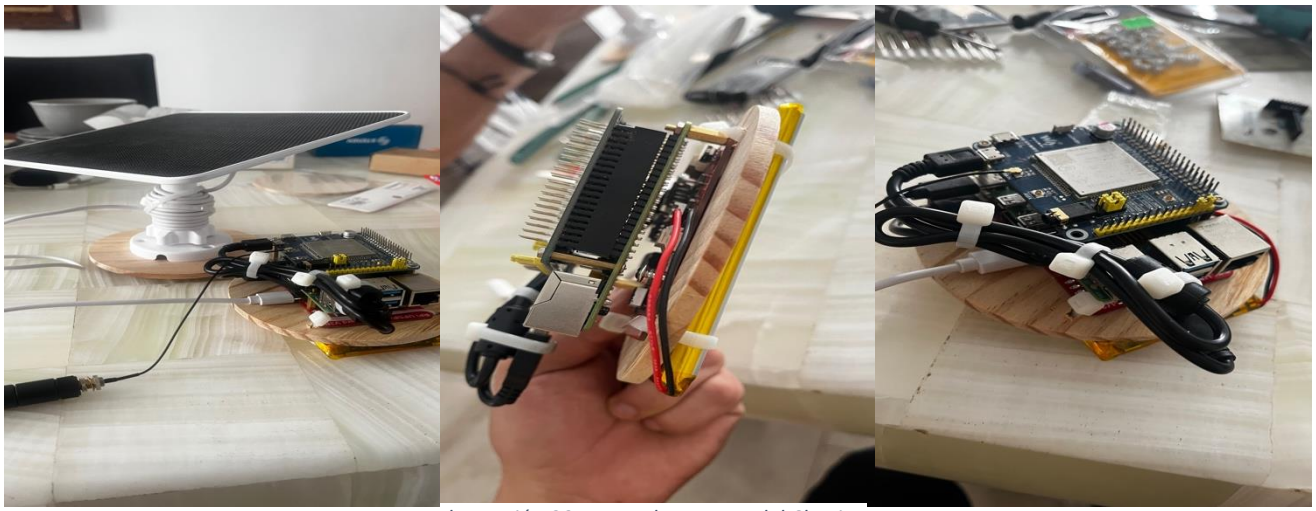


Ilustración 39.- Armado Externo del Chasis

5.9 Flotador del Chasis

Por último, se generará el conjunto final del chasis, el cual incluye un flotador para sostener los circuitos y una esfera de acrílico que sella los componentes, protegiéndolos de la lluvia y de cualquier contacto con agua. Se hicieron agujeros con el diámetro adecuado para permitir el paso de los cables de los sensores, los cuales se sellaron con adhesivo o herramientas especializadas para evitar filtraciones. Además, se instaló la base del panel solar cerca del chasis, proporcionando energía al circuito en caso de que no haya suficiente suministro.

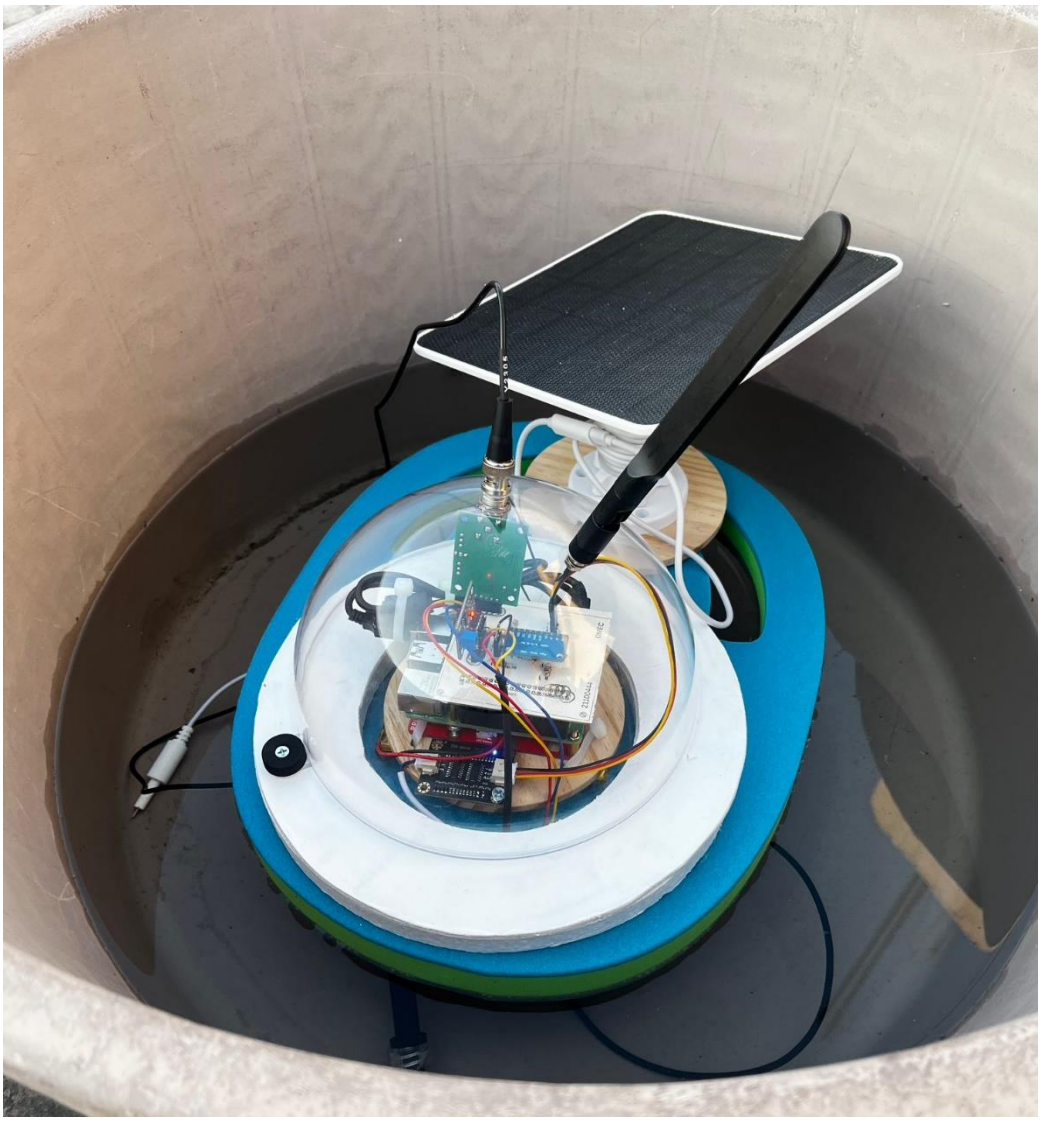


Ilustración 40.- Chasis terminado

Capítulo VI.
GUIA DE USUARIO
Y MANTENIMIENTO

6.1 Guía de usuario

Antes de encender el dispositivo, es importante contar con un plan de datos móviles activo para garantizar el envío de información. Puedes elegir la opción que mejor se adapte a tus necesidades, ya sea el envío de datos por $0.85 MXN por MB, un paquete de $15 MXN por dos horas ilimitadas, o cualquier otro plan o renta de internet disponible en la aplicación de Telcel.

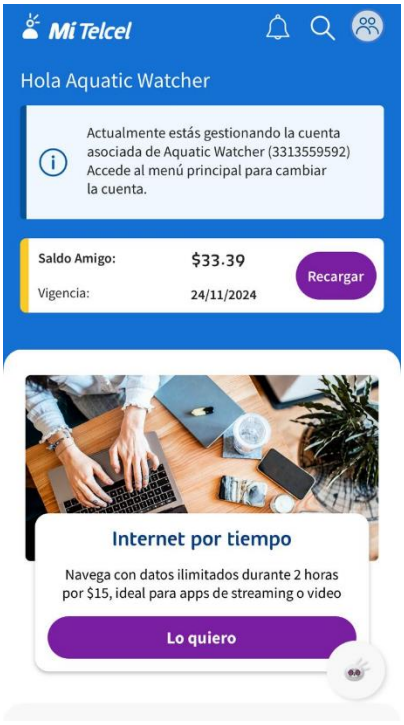


Ilustración 41.- Megas del dispositivo

6.2 Encendido y apagado

Para el correcto funcionamiento de Aquatic Watcher, asegúrate de mover el interruptor a la posición "ON". Una vez encendido, varios Leds se iluminarán, indicando que el dispositivo está operando correctamente. Si todo funciona bien, coloca nuevamente la tapa del dispositivo y asegúrate de sellarla adecuadamente. Después de encender el dispositivo correctamente, Aquatic Watcher enviará los primeros cuatro datos como confirmación de que está funcionando adecuadamente. Posteriormente, continuará enviando datos cada 5 minutos de manera automática.

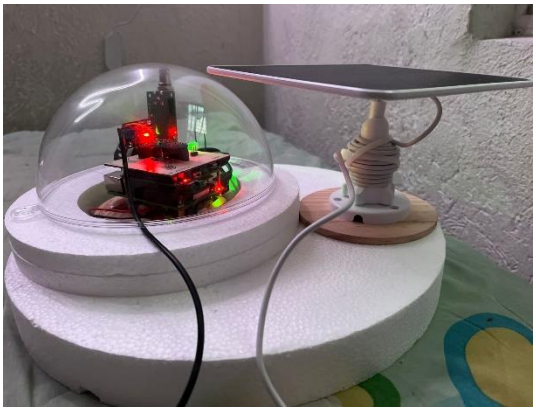


Ilustración 42.- Sistema encendido

6.3 Carga del banco de baterías

Las baterías de Aquatic Watcher se recargan automáticamente mediante el panel solar integrado. Sin embargo, si es necesario, ya sea por decisión del usuario o debido a variaciones climáticas, se puede conectar un cable micro USB en el mismo puerto donde está conectado el panel solar, sin necesidad de abrir la carcasa. Solo es necesario desconectar el panel solar y conectar el cable micro USB.

El sistema está diseñado para que el panel solar suministre energía a las baterías, las cuales, a su vez, alimentan la Raspberry Pi 4. La tarjeta de alimentación incluye 4 Leds verdes que indican el nivel de carga de la batería: cada LED representa un 25% de capacidad. Por ejemplo, si están encendidos 3 Leds, la batería tiene un 75% de carga. Además, cuenta con un LED rojo que parpadea cuando el sistema está recibiendo energía externa, ya sea del panel solar o del cable micro USB.

6.4 Manual de mantenimiento

Este manual tiene como objetivo proporcionar las pautas y procedimientos necesarios para realizar un mantenimiento efectivo del dispositivo. Se describen tres tipos de mantenimiento: predictivo, preventivo y correctivo, cada uno enfocado en diferentes aspectos. Estos enfoques complementarios ayudan a garantizar el rendimiento óptimo y prolongar la vida útil del dispositivo.

6.5 Mantenimiento predictivo

Analizar los datos de los sensores para verificar que estén funcionando correctamente y asegurarse de que estén conectados adecuadamente, ya que durante el transporte podría ocurrir que algún sensor se haya desconectado o dañado. Además, es importante supervisar los leds indicadores de la batería para evitar una sobrecarga. También se recomienda inspeccionar visualmente la batería o tocarla ligeramente para verificar que no esté inflada debido a su uso prolongado.

Cabe destacar que el sistema está diseñado con protección contra sobrecargas: si se detecta una sobrecarga de energía, el dispositivo se apagará automáticamente para evitar daños en los sensores o en otros componentes, asegurando así su correcto funcionamiento a futuro.

6.6 Mantenimiento preventivo

Inspección del hardware:

- Verificar la integridad del panel solar, asegurándose de que no haya obstrucciones, suciedad o daños que puedan afectar la generación de energía.
- Inspeccionar las conexiones del panel solar y el cable micro USB, comprobando que no haya corrosión, desgaste o desconexiones.
- Revisar la carcasa del dispositivo para asegurar que esté completamente sellada y sin fisuras, protegiendo los componentes internos de la humedad y el polvo.

Limpieza regular:

- Limpiar la superficie del panel solar con un paño suave y seco para evitar la acumulación de suciedad que pueda reducir su eficiencia.
- Retirar cualquier residuo que pueda adherirse a los sensores, especialmente si el dispositivo está en contacto directo con agua.

6.7 Mantenimiento correctivo

El mantenimiento correctivo se realiza para reparar fallas o averías que impiden el funcionamiento normal del dispositivo. Este tipo de mantenimiento incluye la identificación del problema, su diagnóstico, y la reparación o reemplazo de los componentes afectados para restaurar el sistema a su estado óptimo.

Procedimientos de mantenimiento correctivo

1. Diagnóstico del problema:
    - Identificación: Revisar las alertas generadas por el sistema, como la ausencia de datos enviados o un LED de estado que no encienda.
2. Reparación del hardware:
    - Reemplazar cualquier sensor que reporte datos inconsistentes o que no responda a pruebas manuales.
    - Asegurarse de que los nuevos sensores estén correctamente calibrados antes de volver a instalarlos.

- Sustituir el panel si presenta daños físicos o no genera energía suficiente, incluso en condiciones de luz solar adecuadas.
- Cambiar las baterías si no retienen la carga o muestran signos de hinchazón o corrosión en los contactos.
- Reparar cables dañados o reemplazar conectores que presenten fallas de continuidad.

3. Corrección del software:

- Volver a cargar el programa base en la Raspberry Pi 4 en caso de fallos críticos o corrupción del código.
- Identificar errores en el código de Python y corregirlos para garantizar el envío adecuado de datos al servidor.

4. Pruebas de funcionamiento:

- Después de realizar las reparaciones, encender el dispositivo y confirmar que los leds de estado funcionan correctamente.
- Verificar que los datos sean enviados de manera continua y que las lecturas sensores sean precisas.

CONCLUSIÓN

Aquatic Watcher demuestra que es posible crear una herramienta eficiente y económica para el monitoreo de las condiciones del agua, accesible para diversas aplicaciones. Si bien su precisión no alcanza la de dispositivos especializados como medidores de pH o turbidez, el Aquatic Watcher destaca por ofrecer mediciones confiables a un costo significativamente menor, haciendo que el monitoreo ambiental sea más accesible.

Además, su diseño permite obtener una visión general de las condiciones del agua, y con una red de dispositivos interconectados, puede mejorar la precisión al recopilar más muestras y procesarlas en un servidor central. Este enfoque amplía su potencial para estudios detallados y gestión ambiental.

La elección de la Raspberry Pi 4 como unidad central fue estratégica, permitiendo independencia de redes Wi-Fi al utilizar una conexión a Internet autónoma, similar a un teléfono celular. Esto convierte al dispositivo en una solución versátil para entornos remotos, abriendo nuevas posibilidades para el monitoreo de ecosistemas acuáticos y contribuyendo a la preservación de recursos hídricos de manera innovadora y económica.

BIBLIOGRAFIA

- Raspberry Pi 4 - Search videos. (s. f.). https://www.bing.com/videos/riverview/relatedvideo?&q=Raspberry+Pi+4+&&mid=5337F5E4C120988190265337F5E4C12098819026&&FORM=VRDGAR
- Raspberry Pi documentation - Waveshare Wiki. (s. f.). https://www.waveshare.com/wiki/Raspberry_Pi_Documentation
- SIM7600E-H 4G HAT - Waveshare Wiki. (s. f.). https://www.waveshare.com/wiki/SIM7600E-H_4G_HAT
- alldatasheet.com. (s. f.). ADS1113_1 Datasheet(PDF). Texas Instruments. https://www.alldatasheet.com/datasheet-pdf/pdf/345942/TI/ADS1113_1.html
- IndustrySurfer, & IndustrySurfer. (2023, 3 marzo). Alimentación de la Raspberry Pi 4. Industry Surfer. https://industrysurfer.com/blog-industrial/ingenieria/ingenieria-electrica-ingenieria/alimentacion-de-la-raspberry-pi-4/#:~:text=Resumiendo%20lo%20anterior%2C%20una%20fuente%20de%20alimentaci%C3%B3n%20adecuada,fluctuaciones%20en%20el%20voltaje%20de%20suministro%20son%20inaceptables.
- Rus, C. (2019, 25 julio). Raspberry Pi 4 es oficial: una completa actualización con procesador Cortex-A72, hasta 4 GB de RAM y. . . Xataka. https://www.xataka.com/ordenadores/raspberry-pi-4-caracteristicas-precio-ficha-tecnica
- Panel solar Raspberry Pi 4 - Search Videos. (s. f.). https://www.bing.com/videos/riverview/relatedvideo?q=Panel+solar+Raspberry+Pi+4&qpvt=Panel+solar+Raspberry+Pi+4&view=riverview&mmscn=mtsc&mid=133EA831A73571E6B937133EA831A73571E6B937&&aps=126&FORM=VMSOVR
- POST - HTTP | MDN. (2023, 24 julio). MDN Web Docs. https://developer.mozilla.org/es/docs/Web/HTTP/Methods/POST

APÉNDICE

Codigo de funcionamiento general:

```
import time

import board

import busio

import adafruit_ads1x15.ads1015 as ADS

from adafruit_ads1x15.analog_in import AnalogIn

import requests

import json

import glob

import os

# Create the I2C bus

i2c = busio.I2C(board.SCL, board.SDA)

# Create the ADC object using the I2C bus

ads = ADS.ADS1015(i2c)

#these tow lines mount the device:

os.system('modprobe w1-gpio')

os.system('modprobe w1-therm')

base_dir = '/sys/bus/w1/devices/'

device_path = glob.glob(base_dir + '28*')[0] #get file path of sensor

rom = device_path.split('/')[-1] #get rom name


# Create single-ended input on channel 0

chan = AnalogIn(ads, ADS.P0)

chan1 = AnalogIn(ads, ADS.P1)
```

```
chan2 = AnalogIn(ads, ADS.P2)
chan3 = AnalogIn(ads, ADS.P3)

#Subrutine for temperature
def read_temp_raw():
    with open(device_path +'/w1_slave','r') as f:
        valid, temp = f.readlines()
    return valid, temp

def read_temp():
    valid, temp = read_temp_raw()

    while 'YES' not in valid:
        time.sleep(0.2)
        valid, temp = read_temp_raw()

    pos = temp.index('t=')
    if pos != -1:
        #read the temperature .
        temp_string = temp[pos+2:]
        temp_c = float(temp_string)/1000.0
        return temp_c,

#Subrutine for Turbidity
```

```
def voltage_to_ntu(voltage2):
   NTU= (5-voltage2)*3000/(10)
   voltage2=chan1.value
   return NTU


#Subrutine for TDS
def read_tds():
    voltage3 = chan3.voltage
    if voltage3 >= 0.80:
        ppm=171/voltage3
        return ppm
    else:
     ppm=171/voltage3
     return ppm


#Subrutine for pH
def read_ph():
    voltage4 = chan2.voltage
    if voltage4 >= 2.549:
        pH_value =voltage4*1.8
        return pH_value
    else:
     pH_value=voltage4*1.8
     return pH_value
```

```
#Subrutine for Send Data
url = "https://diego.enciso.mx/aqua.asp"


while True:
    c = read_temp()
    NTU=chan1.voltage
    ntu_value= voltage_to_ntu(NTU)
    tds_value= read_tds()
    ph=read_ph()
    print("Temperatura:                {:,.3f} ".format(c[0]))
    print("Turbidez:                   {:.2f} NTU".format(ntu_value))
    print("Total de Solidos Disueltos:  {:.2f} ppm".format(tds_value))
    print(f"pH:                         {ph:.2f}")
    formatted_temperature_c= "{:.2f}".format(c[0])
    formatted_ntu_value="{:.2f}".format(ntu_value)
    formatted_tds_value="{:.2f}".format(tds_value)
    formatted_ph="{:.2f}".format(ph)
    time.sleep(6.5)
    params = {
    "Maquina": "watcher1",
    "Parametro": "Temperatura",
    "Dato": formatted_temperature_c,
    "Run": "1"
```

```
    }
params1 = {
"Maquina": "watcher1",
"Parametro": "Turbidez",
"Dato": formatted_ntu_value,
"Run": "1"
    }
params2 = {
"Maquina": "watcher1",
"Parametro": "PPM",
"Dato": formatted_tds_value,
"Run": "1"
    }
params3 = {
"Maquina": "watcher1",
"Parametro": "pH",
"Dato": formatted_ph,
"Run": "1"
    }
response = requests.get (url,params)
response = requests.get (url,params1)
response = requests.get (url,params2)
response = requests.get (url,params3)
time.sleep(300)
```

```
if response.status_code == 200:
    print("PeticiÃ³n exitosa!")
    print("Contenido de la respuesta:")
    print(response.text)
else:
  print(f"Error en la peticiÃ³n: {response.status_code}")
```

Ilustración 43.- Codigo QR del servidor

Usuario: diego

Contraseña: admin2005

ANEXOS

Raspberry Pi 4 Model B features a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro HDMI ports, hardware video decode at up to 4Kp60, up to 8GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability (via a separate PoE HAT add-on). For the end user, Raspberry Pi 4 Model B provides desktop performance comparable to entry-level x86 PC systems. This product retains backwards compatibility with the prior-generation Raspberry Pi 3 Model B+ and has similar power consumption, while offering substantial increases in processor speed, multimedia performance, memory, and connectivity. The dual-band wireless LAN and Bluetooth have modular compliance certification, allowing the board to be designed into end products with significantly reduced compliance testing, improving both cost and time to market.

 *Raspberry Pi 4 Modelo B*	BCM2711	1 GB 2 GB 4 GB 8 GB	Cabezal GPIO de 40 pines	• 2 × micro HDMI • 2 × USB 2.0 • 2 × USB 3.0 • Puerto de cámara CSI • Puerto de pantalla DSI • Conector AV de 3,5 mm • Gigabit Ethernet compatible con PoE (1 Gb/s) • Wi-Fi 802.11ac de doble banda de 2,4/5 GHz (120 Mb/s) • Bluetooth 5, Bluetooth de baja energía (BLE) • Ranura para tarjeta microSD • Alimentación USB-C (5 V, 3 A (15 W))

In some circumstances it may be necessary to update the VideoCore firmware in a Raspberry Pi operating system (OS) image without going through the normal upgrade process. This whitepaper documents how to use the normal upgrade process, and also gives information on how to bypass the standard update process if it is not suitable.

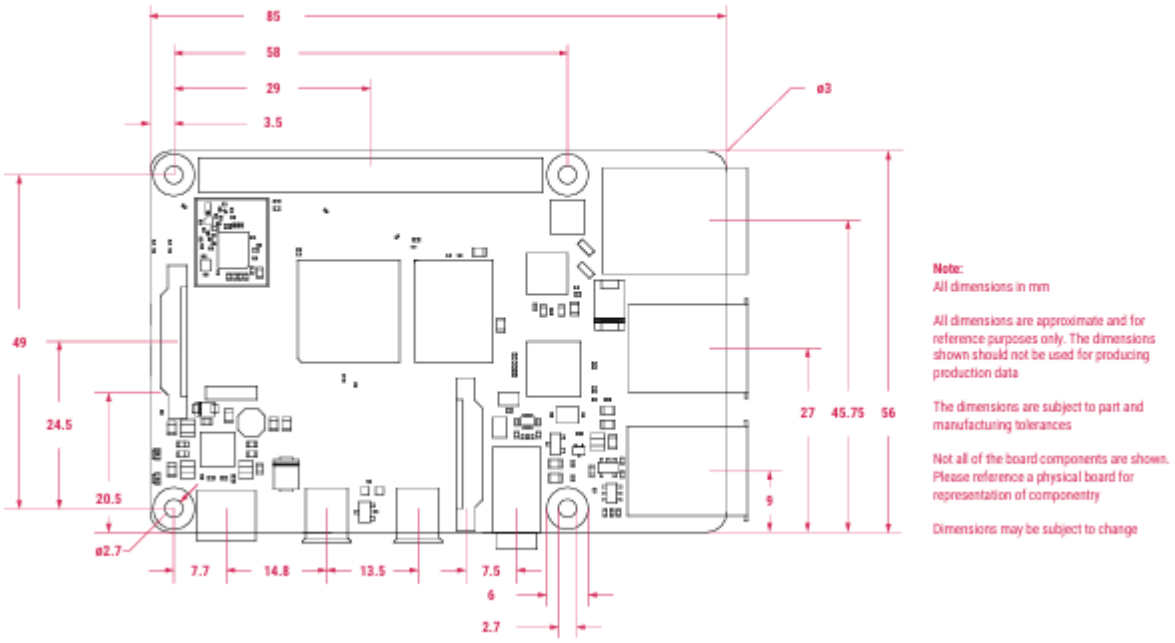
There are standard commands you can use in the Raspberry Pi OS (and many third-party OS distributions) that will upgrade the system and any firmware. Raspberry Pi Ltd recommends using these processes wherever possible. To upgrade the Linux kernel and all Raspberry Pi–specific firmware to the latest release version, the following commands should be used:

```
sudo apt update
sudo apt full-upgrade
```

Note that this process will not upgrade between major OS versions. While it is possible to implement a full upgrade between major versions in place, Raspberry Pi Ltd does not recommend this — it is not a tested procedure due to the huge number of changes involved. In this case we recommend starting afresh, installing the OS from scratch on a new Secure Digital (SD) card using Raspberry Pi Imager. You will need to reinstall all the required software in a new installation

Specification

Processor:	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory:	1GB, 2GB, 4GB or 8GB LPDDR4 (depending on model) with on-die ECC
Connectivity:	• 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE • Gigabit Ethernet • 2 × USB 3.0 ports • 2 × USB 2.0 ports.
GPIO:	Standard 40-pin GPIO header (fully backwards-compatible with previous boards)
Video & sound:	• 2 × micro HDMI ports (up to 4Kp60 supported) • 2-lane MIPI DSI display port • 2-lane MIPI CSI camera port • 4-pole stereo audio and composite video port
Multimedia:	H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphics
SD card support:	Micro SD card slot for loading operating system and data storage
Input power:	• 5V DC via USB-C connector (minimum 3A[1]) • 5V DC via GPIO header (minimum 3A[1]) • Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment:	Operating temperature 0–50°C
Production lifetime:	Raspberry Pi 4 Model B will remain in production until at least January 2034.
Compliance:	For a full list of local and regional product approvals, please visit pip.raspberrypi.com



WARNINGS:

This product should only be connected to an external power supply rated at 5V/3A DC or 5.1V/ 3A DC minimum1. Any external power supply used with Raspberry Pi 4 Model B shall comply with relevant regulations and standards applicable in the country of intended use. • This product should be operated in a well-ventilated environment and, if used inside a case, the case should not be covered. • This product should be placed on a stable, flat, non-conductive surface in use and should not be contacted by conductive items. • The connection of incompatible devices to the GPIO connection may affect compliance and result in damage to the unit and invalidate the warranty. • All peripherals used with this product should comply with relevant standards for the country of useand be marked accordingly to ensure that safety and performance requirements are met. These articles include but are not limited to keyboards, monitors and mice when used in conjunction with Raspberry Pi. • Where peripherals are connected that do not include the cable or connector, the cable or connectormust offer adequate insulation and operation in order that the relevant performance and safety requirements are met.

Medios de arranque

Los modelos de Raspberry Pi carecen de almacenamiento interno, por lo que hay que suministrarlo. Puede arrancar su Raspberry Pi desde una imagen del sistema operativo instalada en cualquier medio compatible: las tarjetas microSD se usan comúnmente, pero también están disponibles el almacenamiento USB, el almacenamiento en red y el almacenamiento conectado a través de un PCIe HAT. Sin embargo, solo los modelos recientes de Raspberry Pi admiten todos estos tipos de medios.

Todos los modelos de consumo de Raspberry Pi desde la Raspberry Pi 1 Modelo A+ cuentan con una ranura microSD. Su Raspberry Pi arranca automáticamente desde la ranura microSD cuando la ranura contiene una tarjeta.