



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Diseño de un sistema de monitorización de la calidad del
agua para zonas costeras

Trabajo Fin de Grado

Grado en Tecnologías Interactivas

AUTOR/A: Soto Perez, Maria

Tutor/a: Sendra Compte, Sandra

Cotutor/a externo: VICIANO TUDELA, SANDRA

CURSO ACADÉMICO: 2021/2022

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA POLITÈCNICA SUPERIOR DE GANDIA

Grado en Tecnologías Interactivas



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCOLA POLITÈCNICA
SUPERIOR DE GANDIA

“Diseño de un sistema de monitorización de la calidad del agua para zonas costeras”

TRABAJO FINAL DE GRADO

Autor/a:
Soto Pérez, María

Tutor/a:
Sendra Compte, Sandra
Viciano Tudela, Sandra

GANDIA, 2022

Resumen

Como propuesta para mi TFG, se pretende realizar un sistema de monitorización de la calidad del agua de las zonas costeras, en concreto, se centrará en el análisis de la bacteria *Escherichia coli*, o mejor conocida como *E. coli*, presente en aguas procedentes de los cultivos cercanos que no llega a ser tratada correctamente y desemboca en el mar. Además de ello, el dispositivo tendrá otras funcionalidades como la de medir la temperatura del agua, el nivel de turbidez y las mareas o corrientes del mar. Todo ello, será mostrado en una aplicación web disponible para todo aquel que desee ver una información sobre la costa en tiempo real. Para llevar a cabo dicha tarea, se hará uso de la tecnología de LoRaWAN, ya que el objetivo no es crear una red de comunicación que trabaje con grandes potencias, sino una red con un rango de transmisión lo suficientemente grande para que la información llegue sin problemas desde la posición del dispositivo hasta la costa más cercana. Todo esto puede suponer una mejora en la salubridad del agua del mar y para que la población conozca el estado real de las playas.

Palabras clave

Zonas Costeras, Sensores, Internet de las Cosas (IoT), Calidad del Agua, Contaminación Marina, Monitorización.

Abstract

My TFG intends to carry out a water quality monitoring system in coastal areas, specifically, it will focus on the analysis of the *Escherichia coli* bacteria, or better known as *E. coli*, which is present in waters from nearby crops that are not treated properly and flow into the sea. Moreover, the device will have other functionalities such as measuring the temperature of the water, the level of turbidity and the tides or currents of the sea. All this will be displayed in a web application available to anyone who wishes to see information about the coast in real time. To carry out this task, LoRaWAN technology will be used, since the goal is not to create a communication network that works with great powers, but rather a network with a transmission range large enough for the information to arrive without problems from the position of the device to the nearest coast. Thus, this can lead to an improvement in the healthiness of the sea water and therefore let the population know about the real state of the beaches.

Key words

Coastal Areas, Sensors, Internet of Things (IoT), Water Quality, Marine Pollution, Monitoring.

Resum

Com a proposta per al meu TFG, es pretén realitzar un sistema de monitoratge de la qualitat de l'aigua de les zones costaneres, en concret, es centrarà en l'anàlisi del bacteri *Escherichia coli*, o millor coneguda com a *E. coli*, present en aigües procedents dels cultius pròxims que no arriba a ser tractada correctament i desemboca en la mar. A més, el dispositiu tindrà altres funcionalitats com la de mesurar la temperatura de l'aigua, el nivell de torbesa i les mareas o corrents de la mar. Tot això, serà mostrat en una aplicació web disponible per a tot aquell que desitge veure una informació sobre la costa en temps real. Per a dur a terme aquesta tasca, es farà ús de la tecnologia de LoRaWAN, ja que l'objectiu no és crear una xarxa de comunicació que treballi amb grans potències, sinó una xarxa amb un rang de transmissió prou gran perquè la informació arribe sense problemes des de la posició del dispositiu fins a la costa més pròxima. Tot això pot suposar una millora en la salubritat de l'aigua de la mar i perquè la població conega l'estat real de les platges.

Paraules clau

Zones Costaneres, Sensors, Internet de les Coses (IoT), Qualitat de l'Aigua, Contaminació Marina, Monitoratge.

Índice

| | |
|--|----|
| Índice de tablas | 3 |
| Índice de figuras | 4 |
| Capítulo 1. Introducción..... | 6 |
| 1.1. Objetivos | 7 |
| 1.2. Diagrama de Gantt | 8 |
| 1.3. Presupuesto | 9 |
| 1.4. Estructura de la memoria | 11 |
| Capítulo 2. Trabajos relacionados | 12 |
| Capítulo 3. Fundamentos teóricos..... | 14 |
| 3.1. Materiales..... | 15 |
| 3.1.1. Sensor de temperatura DS18B20 | 15 |
| 3.1.2. Sensor de turbidez..... | 16 |
| 3.1.3. The Things Uno | 17 |
| 3.1.4. Acelerómetro MPU6050..... | 18 |
| 3.1.5. Laird Sentrius™ RG1xx | 19 |
| 3.1.6. Tecnología LoRaWAN | 20 |
| 3.1.7. Ubidots | 22 |
| Capítulo 4. Diseño e implementación | 23 |
| 4.1. Diseño | 23 |
| 4.2. Implementación | 25 |
| 4.2.1. Arduino | 25 |
| 4.2.2. The Things Stack..... | 28 |

| | |
|----------------------------------|----|
| 4.2.2.1. Aplicación..... | 28 |
| 4.2.2.2. Dispositivo final | 30 |
| 4.2.2.3. Puerta de enlace | 31 |
| 4.2.3. Ubidots | 34 |
| Capítulo 5. Resultados | 38 |
| Capítulo 6. Conclusiones..... | 44 |
| 6.1. Líneas futuras..... | 45 |
| Bibliografía | 46 |

Índice de tablas

| | |
|--|---|
| Tabla 1. Presupuesto aproximado de los materiales..... | 9 |
|--|---|

Índice de figuras

| | |
|---|----|
| Figura 1. Diagrama de Gantt..... | 8 |
| Figura 2. Kit de detección de bacteria E. coli [3] | 10 |
| Figura 3. Sensor de temperatura DS18B20 | 15 |
| Figura 4. Sensor de turbidez | 16 |
| Figura 5. Placa The Things Uno..... | 17 |
| Figura 6. Acelerómetro MPU6050 | 18 |
| Figura 7. Puerta de enlace Laird Sentrius™ RG1xx..... | 19 |
| Figura 8. Ejemplo de Tablero de Ubidots | 22 |
| Figura 9. Diseño del software del proyecto | 23 |
| Figura 10. Diseño en Fritzing del hardware..... | 24 |
| Figura 11. Montaje del hardware..... | 24 |
| Figura 12. Inicio del código de Arduino | 25 |
| Figura 13. Función de setup..... | 26 |
| Figura 14. Función checkSettings para el MPU6050 | 26 |
| Figura 15. Función de loop..... | 27 |
| Figura 16. Final del código de Arduino..... | 27 |
| Figura 17. Pulsar añadir aplicación | 28 |
| Figura 18. Rellenar datos para la creación de la aplicación | 28 |
| Figura 19. Vista general de la aplicación creada..... | 29 |
| Figura 20. Código decodificador..... | 29 |
| Figura 21. Se añade un nuevo dispositivo | 30 |
| Figura 22. Se rellena correctamente con los datos que se piden..... | 30 |
| Figura 23. Vista general del nodo creado..... | 31 |
| Figura 24. Login de Laird Connectivity | 31 |
| Figura 25. Se selecciona la opción The Things Network Legacy (TTN) | 32 |
| Figura 26. Pantalla principal de la consola TTN | 32 |
| Figura 27. Se pulsa en añadir gateway | 33 |
| Figura 28. Se rellena el formulario y se hace clic en registrar | 33 |
| Figura 29. Se accede a los dispositivos para añadir un nuevo plugin | 34 |
| Figura 30. Se selecciona el complemento de The Things Stack | 34 |

| | |
|---|----|
| Figura 31. Se guarda la parte final de URL y se escribe la función decodificadora en la Decoding Function | 35 |
| Figura 32. Página general de la aplicación | 36 |
| Figura 33. Se pulsa para añadir el complemento | 36 |
| Figura 34. Se rellena con los datos correspondientes..... | 36 |
| Figura 35. Diferentes tipos de widgets | 37 |
| Figura 36. Resultado por COM de Arduino | 38 |
| Figura 37. Datos en detalle | 39 |
| Figura 38. Datos en detalle 2 | 40 |
| Figura 39. Dashboard con datos | 41 |
| Figura 40. Dashboard con datos 2 | 41 |
| Figura 41. Rangos de advertencia para cada widget | 42 |
| Figura 42. Gráfica de turbidez | 42 |
| Figura 43. Gráfica de temperatura | 42 |
| Figura 44. Imagen del momento de las mediciones | 43 |

Capítulo 1. Introducción

Desde hace tiempo se ha podido observar cómo nuestras costas acogen a diversos usuarios tanto como para su disfrute como su trabajo. Por ello, en este país se hace hincapié en que la mayoría de nuestras zonas de playa estén en posesión de la bandera azul, indicativo de que la calidad del agua de baño es excelente. [1]

Este acontecimiento lo viví en primera persona, ya que trabajaba de socorrista en una de nuestras playas. Nuestro trabajo consistía en que los usuarios de dicha costa pudieran disfrutar de total protección, por lo que, aparte de nosotros, existía un equipo de científicos que recogían muestras del agua cada un par de semanas y así comprobar su calidad. Estos últimos años, hemos tenido que lidiar con estas bacterias fecales, lo que provocaba el cierre de playas y las consiguientes quejas de los usuarios por los largos períodos de clausura.

En las zonas costeras hacían presencia últimamente las bacterias fecales, entre ellas la bacteria *Escherichia Coli*, o más conocida como *E. coli*. Este tipo de microorganismo posee diferentes variantes, algunos no causan ningún tipo de problema, ya que son necesarios para el correcto funcionamiento del proceso digestivo. Pero algunas cepas de esta bacteria han adquirido capacidad de causar infecciones e incluso, provocar diarreas sangrantes, siendo más agravante en niños y ancianos. [2]

Después de haber hecho un estudio del mercado, se ha observado como actualmente no existe un dispositivo que realice un monitoreo continuo de la calidad del agua lo que provoca que haya un intervalo entre análisis bastante extenso, lo que llega a causar el cierre de diversas costas durante días.

Por lo tanto, la idea principal es que este período de tiempo sea lo más corto posible para poder disfrutar de nuestras playas sin peligro y cuidando el medio ambiente, así como tener un libre acceso a toda esta información en todo momento.

1.1. Objetivos

La finalidad que tiene este proyecto es lograr reducir el tiempo de análisis del agua de las zonas de costa y tener acceso permanente a estos datos para permitir el baño y/o en su defecto, avisar a los bañistas de posibles peligros.

También, se buscará obtener diferentes datos de utilidad ya sea para los usuarios al baño, como para los pescadores de la zona y/o diversos trabajadores que requieran de estas zonas de costa.

Entre estos datos se encontrarán los siguientes:

- Temperatura del agua, para conocer aproximadamente la sensación entre frío y calor a la hora de entrar a ella, así como indicador del efecto del cambio climático.
- Nivel de turbidez, uno de los indicadores de la calidad del agua siendo cuanto más transparente mejor.
- Niveles de variación del nivel del mar en los 3 ejes gracias a un acelerómetro, así podemos estimar las mareas y oleaje en todo momento.

Otro de los objetivos que se plantea en este proyecto es poder medir en todo momento el nivel que hay en el agua de la bacteria *E. coli* para así poder determinar si es permitido el baño en esa zona o no. Para determinar si esto es viable, se realiza un estudio de mercado, así como un presupuesto que se podrá ver más adelante.

1.2. Diagrama de Gantt

En este apartado, se mostrará el diagrama de Gantt empleado durante la creación del proyecto pudiendo ver las diferentes fases en las que se ha organizado dicho trabajo.

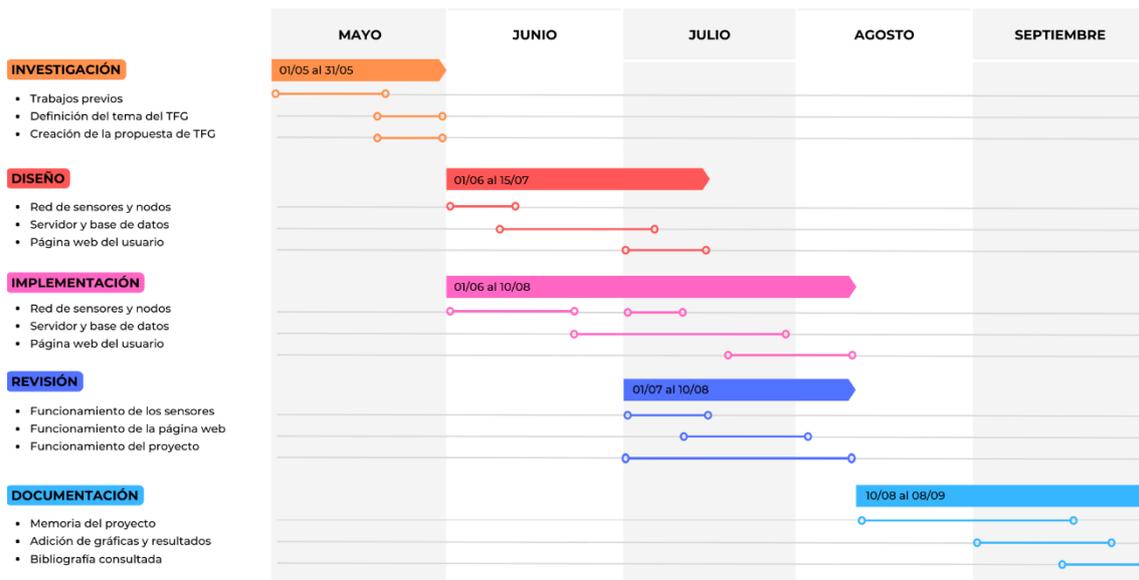


Figura 1. Diagrama de Gantt

Como podemos ver en el esquema, se ha dividido en 5 fases y dentro de cada una de ellas se han contemplado 3 aspectos principales para cerrar la tarea. En la primera parte, principalmente el primer mes, se ha llevado a cabo un trabajo de investigación, en el cual se han trabajado aspectos como la búsqueda de trabajos previos o similares para buscar ideas de cara a este proyecto. Después de ello, se procedió a la decisión del tema final del trabajo de fin de grado, así como la creación de la que sería la propuesta elegida.

Una vez completada esta fase, se ha trabajado paralelamente la parte del diseño como la implementación de todo el proyecto, la cual duro más de 2 meses. Cabe destacar que una vez se completaba tanto el diseño como la implementación de una de las partes, se procedía a la revisión de cada una de ellas para finalmente realizar una prueba final con todas las partes completadas.

Por último, el trabajo se centró en realizar la memoria final del proyecto así cómo mostrar los resultados obtenidos durante el proceso.

1.3. Presupuesto

A continuación, se muestra una tabla donde se ha realizado un presupuesto inicial aproximado del material que se utilizará para llevar a cabo el proyecto.

| | CANTIDAD | PRECIO / UNIDAD (€) | COSTE TOTAL (€) |
|-------------------------------|----------|---------------------|-----------------|
| PLACA THE THINGS UNO | 1 | 40 | 40 |
| SENTRIUS GATEWAY | 1 | 260 | 260 |
| SENSOR DE TEMPERATURA DS18B20 | 1 | 12 | 12 |
| SENSOR ANALÓGICO DE TURBIDEZ | 1 | 13,50 | 13,50 |
| ACELERÓMETRO MPU6050 | 1 | 5,90 | 5,90 |
| RESISTENCIA 4.7 KΩ | 1 | 0,10 | 0,10 |
| CABLEADO | varios | 1,75 | 1,75 |
| TOTAL | | | 333,25 |

Tabla 1. Presupuesto aproximado de los materiales

Como hemos podido observar, la mayoría de los materiales que van a ser empleados en este prototipo son bastante asequibles, pero cabe destacar que el dispositivo de puerta de enlace se usará visto de los precios del resto de componentes. Más adelante se explicarán las razones por las cuales se han elegido estos materiales en vez de otros que pueden resultar más baratos.

Por otro lado, destacamos que no se va a poder realizar un análisis del nivel de la bacteria *E. coli* ya que debido a las investigaciones previas que se han realizado, hemos podido ver cómo todavía no es viable hacer dicho análisis de bajo coste. Se han encontrado una especie de analizadores del agua que se venden por Amazon, pero no sería compatible con este proyecto, ya que no sería posible adaptar estos frascos al prototipo.



Figura 2. Kit de detección de bacteria E. coli [1]

Finalmente, debemos tener en cuenta la mano de obra durante el proceso de creación de este prototipo para poder cerrar el presupuesto aproximado de todo este trabajo. Para ello, cogeremos las horas equivalentes a los créditos de los que consta este proyecto (12 créditos ECTS), las cuales serían unas 25 horas por cada crédito ECTS, lo que daría un total de 300 horas de trabajo. Esto se dividiría de manera que el alumno realizara una media de 4 horas diarias de lunes a viernes hasta completar dicho trabajo. Teniendo en cuenta que el sueldo promedio de un ingeniero junior en España es de 12,31€ la hora [4], esto nos daría un total de 3.693€ por la mano de obra.

1.4. Estructura de la memoria

El siguiente documento estará dividido en las siguientes fases, así como diferentes anexos que podemos encontrar a lo largo de su contenido.

El primer capítulo que está a continuación es el capítulo 2, donde tenemos una recopilación de trabajos relacionados y que han sido útiles en la investigación previa a realizar este proyecto para guiar al alumno a la idea final del trabajo.

En el capítulo 3, tendremos el apartado más teórico donde se explicará con todo detalle la base teórica sobre la que se fundamenta el trabajo y los materiales elegidos para la realización del proyecto, así como la justificación de su uso.

Más adelante, en el capítulo 4, se encuentra la parte del diseño del hardware y software del proyecto dónde se emplearán los programas de Fritzing y Daw.io respectivamente para llevar a cabo dicho trabajo, así como la implementación de dichos diseños en el prototipo.

Para comprobar el funcionamiento del proyecto, en el capítulo 5 encontraremos una recopilación de los resultados obtenidos de los sensores implementados y diversas tablas para poder corroborar el correcto funcionamiento del dispositivo.

Finalmente, en el capítulo 6, se recogerán las conclusiones que se han obtenido al finalizar el proyecto y cuáles serían los trabajos futuros para mejorar el dispositivo. Por otro lado, se encontrará la bibliografía con los enlaces de las webs de consulta que se han usado a lo largo del desarrollo del trabajo, así como diversos artículos y/o noticias relevantes.

Capítulo 2. Trabajos relacionados

Para poder entender el iniciador por el cual se ha elegido tratar sobre la calidad del agua en este proyecto, aquí se muestran los trabajos o artículos que se han hallado sobre este tema y que han motivado al alumno para seguir por esta misma línea de trabajo.

Como base de este proyecto, se contó con el trabajo de fin de grado de Pablo Novo Cañizares de la Universidad de Cantabria, graduado en Ingeniería de Tecnologías de Telecomunicación, el cual realizó el proyecto de “Diseño e implementación de una boya oceanográfica para la predicción del oleaje” [5]. El motivo por el cual se tomó como referencia dicho trabajo es que el alumno buscaba una manera de anticiparse al posible oleaje para poder asegurar la seguridad de los usuarios, por lo tanto, aquí también se buscará como finalidad una forma de prevenir posibles infecciones procedentes de la calidad de las aguas.

En toda la investigación previa que se realizó, no se pudo hallar ningún proyecto en el cual se midiera la calidad del agua en todo momento. En su defecto, sí que se encontraron diferentes referencias por separado, pero no un dispositivo en el cual se agruparan todos los factores de los que depende la calidad de las aguas.

Otro de los motivos por los cuales sea tratado de buscar un dispositivo que pueda monitorizar la calidad del agua, es el potencial turismo que recogen nuestras costas ya que como se observó en el estudio que se realizó en la Universidad de Colima (México) [6], aparte de un posible riesgo sanitario, esto también conlleva un claro descenso de la actividad turística de la zona costera. Esta situación también se ve agravada por la pandemia que sufrimos durante estos últimos dos años ya que la gente toma más precaución a la hora de ir según a qué sitios y/o tomar decisiones.

También se ha encontrado el “Estudio sobre la cobertura de la modulación LoRa en la banda de 915 MHz en un ambiente urbano”, realizado por Daniel Antonio Jiménez Ballart de la Universidad de Chile. [7]

En este estudio podemos ver cómo la tecnología LoRa puede alcanzar rangos kilométricos siempre y cuando no haya edificaciones u otro tipo de obstáculos entre puertas de enlace y nodos. Por lo tanto, éste ha sido uno de los motivos por los cuales se ha decantado por esta tecnología, así como el bajo costo energético que ofrece.

Capítulo 3. Fundamentos teóricos

En este capítulo, analizaremos de forma más teórica los factores que definen la calidad del agua y porque se han elegido tres de ellos en este proyecto.

En primer lugar, vamos a tener en cuenta la temperatura del agua ya que en estos últimos años esta se ha visto afectada por el cambio climático, lo que está provocando que el nivel del mar suba en ciertas áreas y provoque desastres naturales serios. Por lo tanto, deduciremos qué si la temperatura del agua es elevada, será perjudicial para los usuarios e incluso puede llegar a proliferar organismos que alteren la calidad de esta. A su vez, este fenómeno puede causar el efecto contrario, es decir, puede hacer que especies u organismos que se encuentren en nuestras costas actualmente, tengan que realizar un desplazamiento hacia zonas más frías para sobrevivir.

Otro de los factores, y uno de los más importantes, que se ha tenido en cuenta para determinar la calidad del agua es la turbidez. Esta puede ser un indicativo de una mayor probabilidad de contaminación microbiológica y por compuestos químicos, que se adhieren a la materia dispersa en el agua [8]. Por lo general, se define como la propiedad óptica de una suspensión, que hace que la luz se disperse y no se transmita a través de la suspensión. En consecuencia, para este factor daremos un resultado binario ya que, en presencia de un cierto número de partículas, daremos cómo agua de mala calidad a esta.

Por último, pero no menos importante, analizaremos la marea gracias a un acelerómetro de 3 ejes que nos indicará si la boya se encuentra estática o en movimiento. Este dato lo tendremos en cuenta ya que para los usuarios de las costas puede ser una información relevante a la hora de hacer uso de ellas.

3.1. Materiales

3.1.1. Sensor de temperatura DS18B20

El DS18B20 es un sensor digital de temperatura que utiliza el protocolo 1-Wire para comunicarse, este protocolo necesita solo un pin de datos para comunicarse y permite conectar más de un sensor en el mismo bus.

Con este sensor se pueden medir temperatura desde los -55°C hasta los 125°C , con una precisión de 0.5°C de -10°C a 85°C y con una resolución programable desde 9 bits hasta 12 bits.

Cada sensor tiene una dirección única de 64 bits establecida de fábrica, esta dirección sirve para identificar al dispositivo con el que se está comunicando, puesto que en un bus 1-wire pueden existir más de un dispositivo.
[9]

En este proyecto, haremos uso de una resistencia de $4,7\text{k}\Omega$ para que el sensor funcione correctamente.



Figura 3. Sensor de temperatura DS18B20

3.1.2. Sensor de turbidez

El sensor de turbidez tiene dos partes, transmisor y receptor. El transmisor consta de una fuente de luz, típicamente un LED y un circuito de controlador. En el extremo del receptor, hay un detector de luz como un fotodiodo o un LDR. [10]

El transmisor simplemente transmite la luz, las ondas de luz pasan a través de la solución y el receptor recibe la luz. Normalmente (sin la presencia de una solución) la luz transmitida recibe completamente en el lado del receptor. Pero en presencia de una solución turbia, la cantidad de luz transmitida es muy baja. Es decir, en el lado del receptor, obtenemos solo una luz de baja intensidad y esta intensidad es inversamente proporcional a la turbidez.

Las características clave del módulo de turbidez son:

- Voltaje de funcionamiento: 5 V CC.
- Corriente: 30 mA (MAX).
- Temperatura de funcionamiento: -30 ° C a 80 ° C.
- Compatible con Arduino, Raspberry Pi, AVR, PIC, etc.

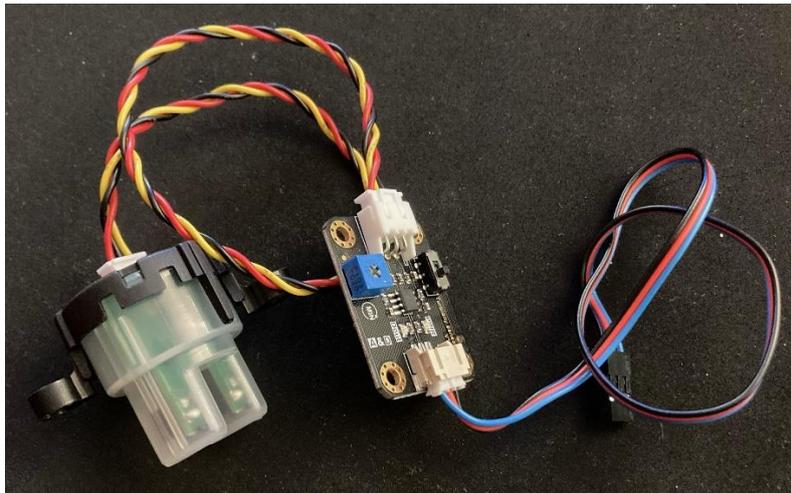


Figura 4. Sensor de turbidez

3.1.3. The Things Uno

Things Uno es una placa de desarrollo compatible con Arduino que facilita la creación de prototipos de dispositivos LoRaWAN. Usando la API Arduino de código abierto y la biblioteca Things Network, puede estar listo y funcionando en minutos. [11]

La placa es compatible con Arduino Leonardo y puede usar escudos Arduino para ampliar sus capacidades. Things Uno tiene un microcontrolador y utiliza el popular módulo Microchip RN2483 para conectividad LoRa, a través de la antena integrada.

Estas son las características clave que posee esta placa:

- Arduino Leonardo compatible
- Banda EU868 LoRaWAN
- Biblioteca TTN
- Microchip RN2483 integrado con pila LoRaWAN integrada
- Admite conexiones LoRaWAN y modos LoRa MAC P2P
- Incluye huella para conector u.FL para agregar una antena externa
- Diseño de hardware abierto



Figura 5. Placa The Things Uno

3.1.4. Acelerómetro MPU6050

El MPU-6050 es una unidad de medición inercial (IMU) de seis grados de libertad (DoF), que combina un acelerómetro de 3 ejes y un giroscopio de 3 ejes.

Las características principales [12] por las cuales se usó este dispositivo son:

- Voltaje de operación: 3V/3.3V~5V DC
- Regulador de voltaje en placa
- Grados de libertad (DoF): 6
- Rango Acelerómetro: 2g/4g/8g/16g
- Rango Giroscopio: 250Grad/Seg, 500Grad/Seg, 1000Grad/Seg, 2000Grad/Seg
- Sensibilidad Giroscopio: 131 LSBs/dps
- Interfaz: I2C
- Conversor AD: 16 Bits (salida digital)
- Tamaño: 2.0cm x 1.6cm x 0.3cm

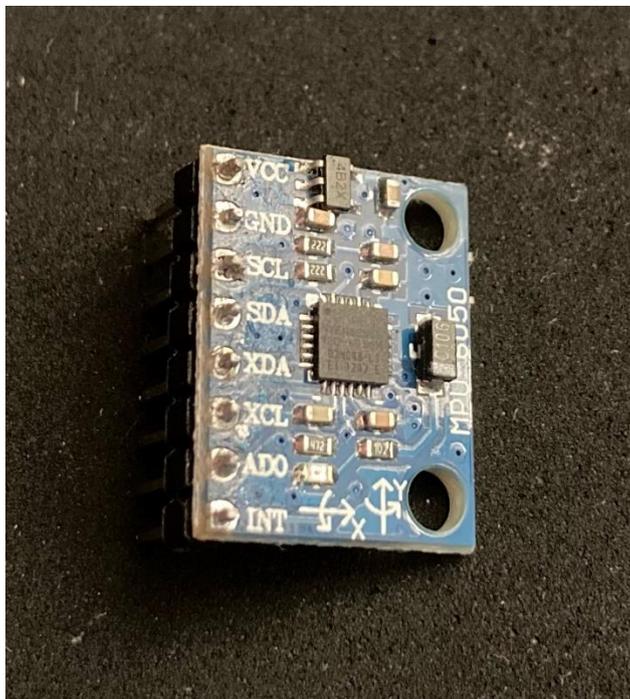


Figura 6. Acelerómetro MPU6050

3.1.5. Laird Sentrius™ RG1xx

Laird Sentrius™ RG1xx es una puerta de enlace segura, escalable y robusta habilitada para LoRaWAN que admite backends Wi-Fi, Ethernet y LTE.

La puerta de enlace Laird Sentrius™ RG1xx LoRaWAN admite dos formas de conectarse con The Things Stack, utilizando el enrutador de paquetes Semtech o la estación básica LoRa. [13]



Figura 7. Puerta de enlace Laird Sentrius™ RG1xx

En el siguiente capítulo se podrá ver como se debe configurar la puerta de enlace correctamente, ya que sin ella no podremos hacer uso de nuestro dispositivo.

3.1.6. Tecnología LoRaWAN

LoRaWan es un protocolo de red que usa la tecnología Lora y que se implementa en la parte superior de la capa física de Lora. LoRaWan es la encargada de especificar como se realizan las comunicaciones y la arquitectura de la red y está gestionada por la Lora Alliance. [14]

Las ventajas que presenta esta tecnología frente a otras serían:

- Conexiones bidireccionales seguras mediante encriptación de extremo a extremo.
- Largo alcance de comunicación (10 - 20 km) en frecuencias libres entre 433 MHz, 868 MHz y 915 MHz [15].
- Conexión de infinidad de sensores y equipos a redes públicas o privadas (hasta 1 millón de nodos en red).
- Baja frecuencia de transmisión, movilidad, velocidades de datos y servicios de localización.
- Interoperabilidad de las diversas redes LoRaWAN en todo el mundo.

Permite la interconexión entre objetos inteligentes sin la necesidad de instalaciones locales complejas, y además otorga amplia libertad de uso al usuario final, al desarrollador y a las empresas que quieran instalar su propia red para Internet de las Cosas (IoT). [16]

El dispositivo final o los nodos en LoRa y LoRaWAN es un sensor o un actuador. Es el que está en contacto directo con el objeto del que se desea obtener información, en este caso se usará The Things Uno. En él debe ocurrir la captura de información, transmisión de ésta y recepción de paquetes provenientes del servidor. [17]

La capa de aplicación del dispositivo final está conectada a un servidor de aplicaciones específico en la nube. Todos los payload de la capa de aplicación de este dispositivo final se enrutan a su servidor de aplicaciones correspondiente.

Existen tres clases de nodos: Clase A (dispositivo final bidireccional), Clase B (dispositivo final bidireccional con ventanas de recepción programadas) y Clase C (dispositivo final bidireccional con ranuras de recepción máximas).

- **Clase A:** esta permite una comunicación bidireccional gracias a que después de cada transmisión enlace ascendente se habilitan dos ventanas cortas para recepción enlace descendente. Esto hace que sea la clase que se use en este prototipo.
- **Clase B:** esta clase no sólo utiliza una ventana de recepción aleatoria como la Clase A, sino que añade al dispositivo una ventana extra programada en ciertos tiempos.
- **Clase C:** esta clase tiene casi continuamente una ventana abierta de recepción, que solamente es cerrada durante la transmisión, por lo tanto, esto provoca un mayor consumo de potencia en comparación con las otras dos.

3.1.7. Ubidots

Ubidots es un creador de aplicaciones de Internet de las cosas (IoT) con análisis y visualización de datos. Convierte los datos de los sensores, en información importante para las decisiones comerciales, las interacciones máquina a máquina, la investigación educativa y una mayor economía de los recursos globales. Ubidots existe como un medio fácil y asequible para integrar el poder y el valor del IoT en los negocios y la investigación.

La plataforma de habilitación de aplicaciones de Ubidots admite visualización de datos interactiva en tiempo real (widgets) y un IoT App Builder que permite a los desarrolladores ampliar la plataforma con su propio código HTML/JS para una personalización privada cuando corresponda. [18]

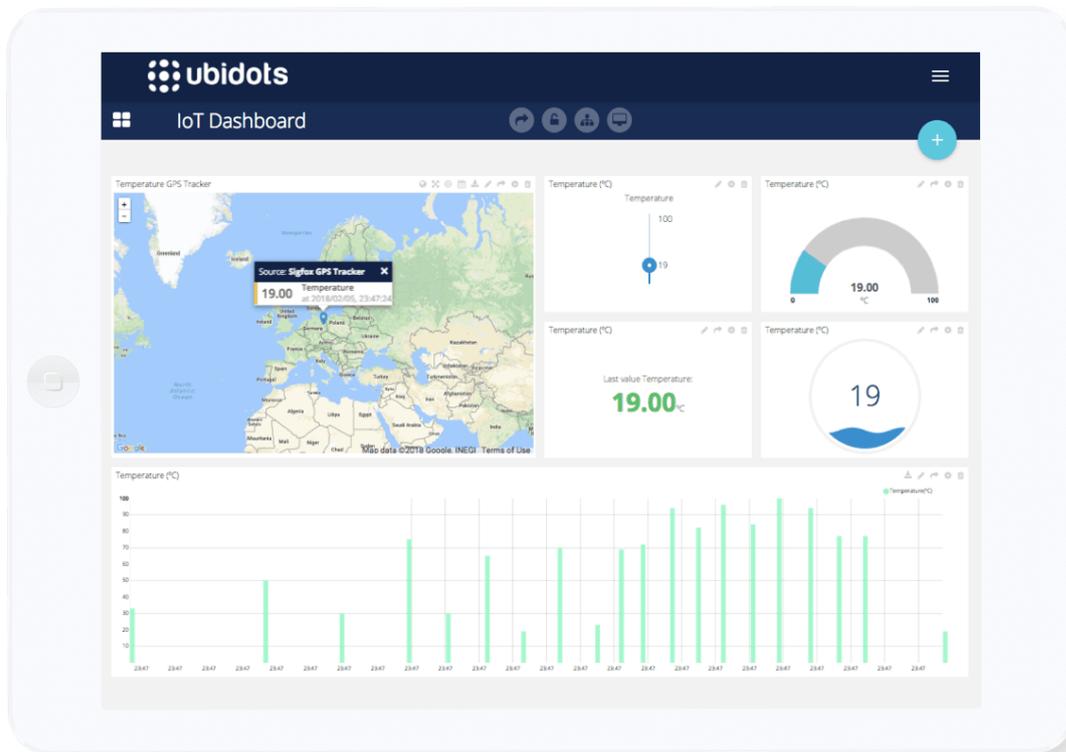


Figura 8. Ejemplo de Tablero de Ubidots

Capítulo 4. Diseño e implementación

En este capítulo, se encuentran los diseños del prototipo tanto a nivel de hardware como de software, así como la implementación de todos ellos en el dispositivo final.

4.1. Diseño

Por lo que respecta al diseño, debemos diferenciar entre el diseño de la parte hardware del de la parte software (creado en Draw.io).

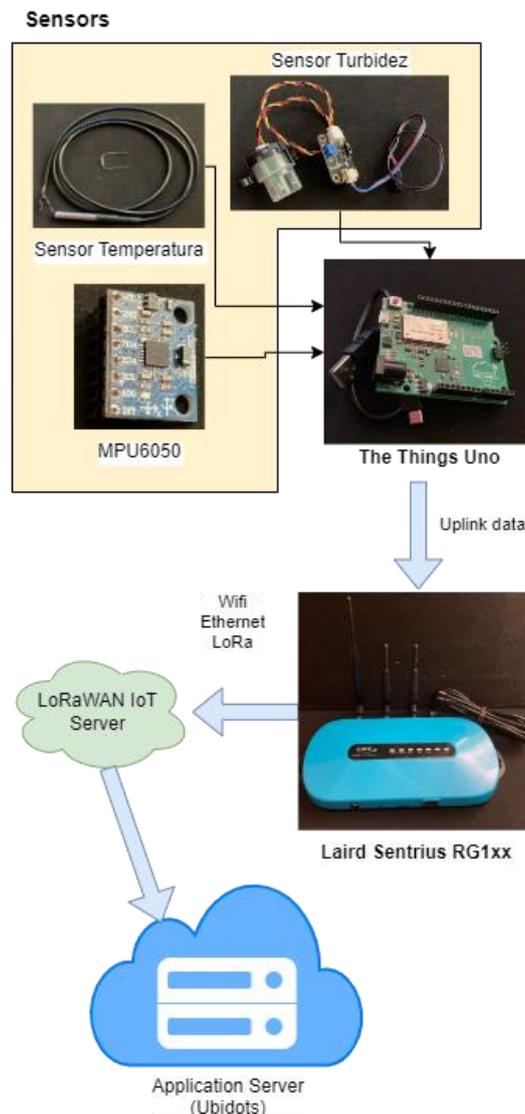


Figura 9. Diseño del software del proyecto

Este proceso va desde que los sensores mandan los datos al nodo, hasta que son mostrados en la web, pasando por la puerta de enlace y el servidor de LoRaWAN. A continuación, se observa el diseño de la parte hardware (creado en Fritzing), en este caso, lo que está conectado al nodo.

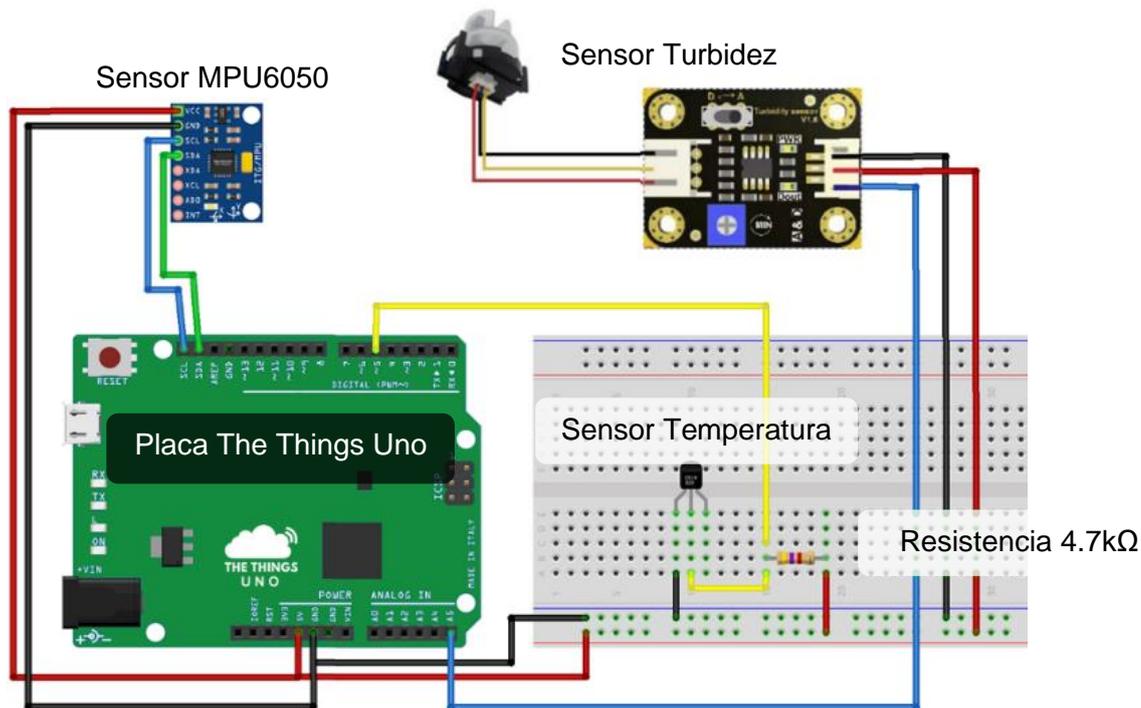


Figura 10. Diseño en Fritzing del hardware

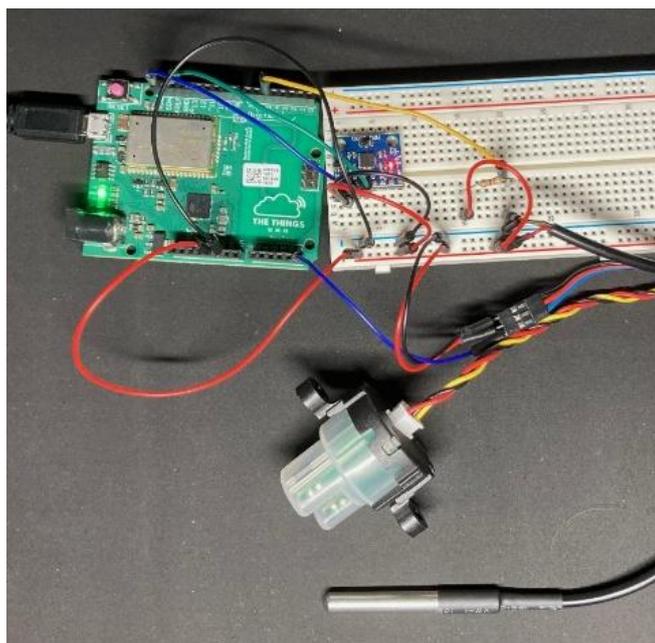


Figura 11. Montaje del hardware

4.2. Implementación

En esta sección del capítulo 4, se comentará el código creado para poder obtener los datos, subirlos a la web de TTN y, posteriormente, poder decodificarlos en la web de Ubidots. A parte, también encuentra la sucesión de pasos que hay que seguir para poder configurar tanto la puerta de enlace, cómo el dispositivo final usado (en este caso The Things Uno), cómo la integración del webhook implementado (en cuyo caso es Ubidots).

4.2.1. Arduino

En primer lugar, tenemos el código .ino, que proviene de Arduino, en el cual hacemos una llamada al nodo para que pueda subir al servicio de TTN los datos de los sensores.

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include <MPU6050.h>
#include <TheThingsNetwork.h>

#define loraSerial Serial1
#define debugSerial Serial
#define freqPlan TTN_FP_EU868
#define ONE_WIRE_BUS 5
MPU6050 mpu;

// Set your AppEUI and AppKey
const char *appEui = "0000000000000000";
const char *appKey = "563C2795E5A15E1321E2AB33B331F0CE";

OneWire oneWire(ONE_WIRE_BUS);
TheThingsNetwork ttn(loraSerial, debugSerial, freqPlan);
DallasTemperature sensors(&oneWire);

uint16_t temp=0;
uint16_t turb=0;
```

Figura 12. Inicio del código de Arduino

Se puede ver como se añaden las librerías necesarias para usar los diferentes sensores y servicios.

Después se definen diferentes variables globales que se usarán más adelante.

Aquí se crea la variable que llamará a la appKey obtenida en la página de TTN del dispositivo final creado.

Además, se definen otras variables que serán utilizadas más adelante para almacenar las medidas obtenidas.

```

void setup(void) {
  Serial.begin(9600);
  sensors.begin();
  loraSerial.begin(57600);
  debugSerial.begin(9600);

  // Wait a maximum of 10s for Serial Monitor
  while (!debugSerial && millis() < 10000)
    ;

  debugSerial.println("-- STATUS");
  ttn.showStatus();

  debugSerial.println("-- JOIN");
  ttn.join(appEui, appKey);

  // Initialize MPU6050
  Serial.println("Initialize MPU6050");
  while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))
  {
    Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
    delay(500);
  }
  mpu.calibrateGyro();
  mpu.setThreshold(3);
  checkSettings();
}

```

Figura 13. Función de setup

```

void checkSettings()
{
  Serial.println();

  Serial.print(" * Sleep Mode:      ");
  Serial.println(mpu.getSleepEnabled() ? "Enabled" : "Disabled");

  Serial.print(" * Clock Source:      ");
  switch(mpu.getClockSource())
  {
    case MPU6050_CLOCK_KEEP_RESET:      Serial.println("Stops the clock and keeps the timing generator in reset"); break;
    case MPU6050_CLOCK_EXTERNAL_19MHZ:  Serial.println("PLL with external 19.2MHz reference"); break;
    case MPU6050_CLOCK_EXTERNAL_32KHZ:  Serial.println("PLL with external 32.768kHz reference"); break;
    case MPU6050_CLOCK_PLL_ZGYRO:       Serial.println("PLL with Z axis gyroscope reference"); break;
    case MPU6050_CLOCK_PLL_YGYRO:       Serial.println("PLL with Y axis gyroscope reference"); break;
    case MPU6050_CLOCK_PLL_XGYRO:       Serial.println("PLL with X axis gyroscope reference"); break;
    case MPU6050_CLOCK_INTERNAL_8MHZ:   Serial.println("Internal 8MHz oscillator"); break;
  }

  Serial.print(" * Gyroscope:      ");
  switch(mpu.getScale())
  {
    case MPU6050_SCALE_2000DPS:         Serial.println("2000 dps"); break;
    case MPU6050_SCALE_1000DPS:         Serial.println("1000 dps"); break;
    case MPU6050_SCALE_500DPS:          Serial.println("500 dps"); break;
    case MPU6050_SCALE_250DPS:          Serial.println("250 dps"); break;
  }

  Serial.print(" * Gyroscope offsets: ");
  Serial.print(mpu.getGyroOffsetX());
  Serial.print(" / ");
  Serial.print(mpu.getGyroOffsetY());
  Serial.print(" / ");
  Serial.println(mpu.getGyroOffsetZ());

  Serial.println();
}

```

Figura 14. Función checkSettings para el MPU6050

En esta parte del código se inicializan las variables antes creadas.

También, se muestra el estado en el que se encuentra el nodo, es decir, si se ha podido establecer conexión con el servidor de TTN o no.

Por otro lado, se comprueba si el MPU6050 está bien conectado y calibrado para su uso. Para comprobar todos los aspectos técnicos de este dispositivo, se crea una función en específico para corroborar que todo funciona correctamente.

```

void loop(void) {
  sensors.requestTemperatures();
  temp=sensors.getTempCByIndex(0)*100;
  int sensorValue = analogRead(A5);

  Serial.println("-----");
  Serial.println("Mediciones");
  Serial.println("-----");

  Serial.print("Nivel de turbidez: ");
  if(sensorValue < 800){
    Serial.println("Agua turbia");
    turb=0;
  }
  if(sensorValue > 800){
    Serial.println("Agua limpia");
    turb=1;
  }
  Serial.print("Temperatura del agua: ");
  Serial.print(temp);
  Serial.println(" C°");

  Vector rawGyro = mpu.readRawGyro();
  Vector normGyro = mpu.readNormalizeGyro();
  uint16_t gyroX=normGyro.XAxis;
  uint16_t gyroY=normGyro.YAxis;
  uint16_t gyroZ=normGyro.ZAxis;

  Serial.println("Movimiento de la boya:");
  Serial.print("Xraw = ");
  Serial.print(rawGyro.XAxis);
  Serial.print(" Yraw = ");
  Serial.print(rawGyro.YAxis);
  Serial.print(" Zraw = ");
  Serial.println(rawGyro.ZAxis);

  Serial.print("Xnorm = ");
  Serial.print(normGyro.XAxis);
  Serial.print(" Ynorm = ");
  Serial.print(normGyro.YAxis);
  Serial.print(" Znorm = ");
  Serial.println(normGyro.ZAxis);
}

```

Figura 15. Función de loop

```

byte payload[10];
payload[0] = highByte(turb);
payload[1] = lowByte(turb);
payload[2] = highByte(temp);
payload[3] = lowByte(temp);
payload[4] = highByte(gyroX);
payload[5] = lowByte(gyroX);
payload[6] = highByte(gyroY);
payload[7] = lowByte(gyroY);
payload[8] = highByte(gyroZ);
payload[9] = lowByte(gyroZ);

// Send it off
ttn.sendBytes(payload, sizeof(payload));

```

Figura 16. Final del código de Arduino

En esta función de loop, lo que hace será mostrar los datos por la COM de Arduino y subir estas medidas obtenidas al servidor TTN.

En primer lugar, nos devolverá el valor de turbidez, así como la temperatura en grados Celsius.

Por otro lado, se analiza el giróscopo que posee el sensor MPU6050, para determinar si el dispositivo se encuentra en movimiento o no, en caso afirmativo se tomará la referencia de qué se trate de un mar movido, por el contrario, se tomará cómo mar en calma.

En esta parte final de la función loop, se crea una secuencia payload que luego será descodificada para poder obtener los datos según a la variable que se refieran.

4.2.2. The Things Stack

A continuación, se explicará cómo configurar desde la consola de The Things Network la aplicación, el dispositivo final y la puerta de enlace que se han empleado en este prototipo. Para ello, se debe estar en posesión previa de una cuenta de The Things Stack.

4.2.2.1. Aplicación

En primer lugar, debemos configurar la aplicación en la cual vamos a tener nuestro dispositivo final y, por tanto, vamos a recibir los datos de, en nuestro caso, la placa The Things Uno. Para ello, nos dirigiremos a la pantalla principal de nuestro perfil y le daremos a ir a nuestras aplicaciones. A continuación, se procederá a crear una nueva aplicación y se rellenará con los datos que correspondan.

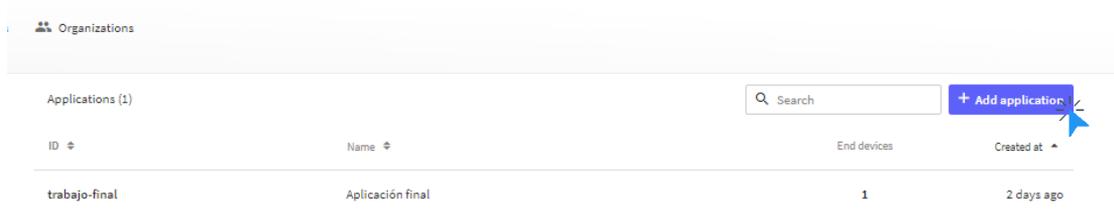


Figura 17. Pulsar añadir aplicación

Add application

Application ID *

Application name

Description

Optional application description; can also be used to save notes about the application

Create application

Figura 18. Rellenar datos para la creación de la aplicación

Por último, debemos crear en JavaScript un código que deberá decodificar la secuencia de bytes proveniente del nodo (por lo tanto, se añadirá en el apartado de Uplink) y así poder mostrar los datos en el apartado de live data.

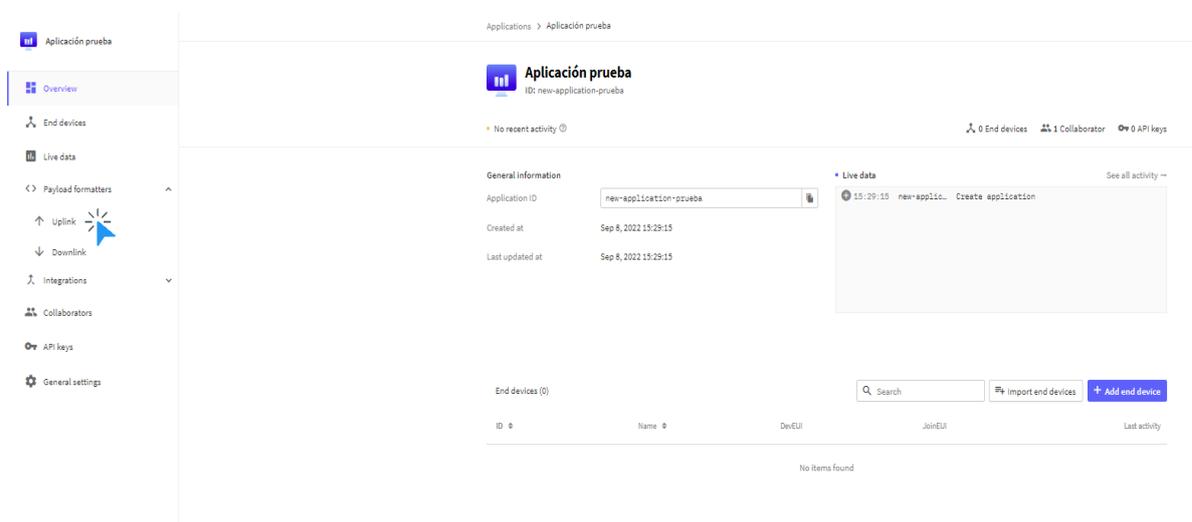


Figura 19. Vista general de la aplicación creada

Finalmente, se selecciona la opción de JavaScript y se añade el siguiente código:

```
Formatter type*
Custom Javascript formatter

Formatter code*
1 function decodeUplink(input) {
2   var data = {};
3   data.turb = (input.bytes[0]<<8|input.bytes[1]);
4   data.temperature = (input.bytes[2]<<8|input.bytes[3])/100;
5   if (data.turb === 0){
6     data.turb = "Agua turbia";
7   } else {
8     data.turb = "Agua limpia";
9   }
10  data.coordX = (input.bytes[4]<<8|input.bytes[5])/100;
11  data.coordY = (input.bytes[6]<<8|input.bytes[7])/100;
12  data.coordZ = (input.bytes[8]<<8|input.bytes[9])/100;
13  if (data.coordX !== 0 || data.coordY !== 0 || data.coordZ !== 0 ){
14    data.marea = "Mar movido";
15  } else {
16    data.marea = "Mar en calma";
17  }
18  return{
19    data: data
20  };
21 }
```

Figura 20. Código decodificador

4.2.2.2. Dispositivo final

Para añadir un dispositivo final a la aplicación, primero se debe pulsar en añadir un dispositivo final y rellenar los siguientes datos con las especificaciones de la placa, en este caso, The Things Uno. Se puede añadir haciendo uso del repositorio de LoRaWAN o de forma manual, lo importante es asegurar que se introducen correctamente las características de este dispositivo.

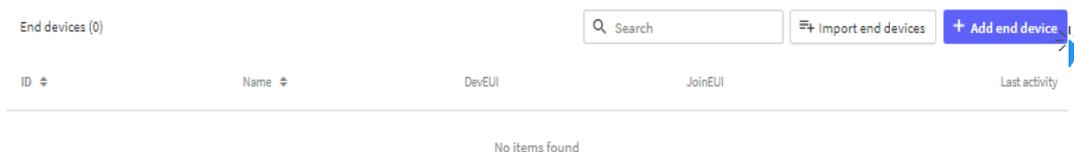


Figura 21. Se añade un nuevo dispositivo

Register end device

From The LoRaWAN Device Repository Manually

Frequency plan [ⓘ] *

Europe 863-870 MHz (SF9 for RX2 - recommended) | v

LoRaWAN version [ⓘ] *

LoRaWAN Specification 1.0.2 | v

Regional Parameters version [ⓘ] *

RP001 Regional Parameters 1.0.2 | v

Show advanced activation, LoRaWAN class and cluster settings v

DevEUI [ⓘ] *

70 B3 D5 7E D0 05 52 A0 1/50 used

AppEUI [ⓘ] *

00 00 00 00 00 00 00 00

AppKey [ⓘ] *

77 ED 71 3F 17 61 27 A2 28 D1 5F F4 6C E1 B8 CB

End device ID [ⓘ] *

eui-70b3d57ed00552a0

This value is automatically prefilled using the DevEUI

After registration

View registered end device

Register another end device of this type

Figura 22. Se rellena correctamente con los datos que se piden

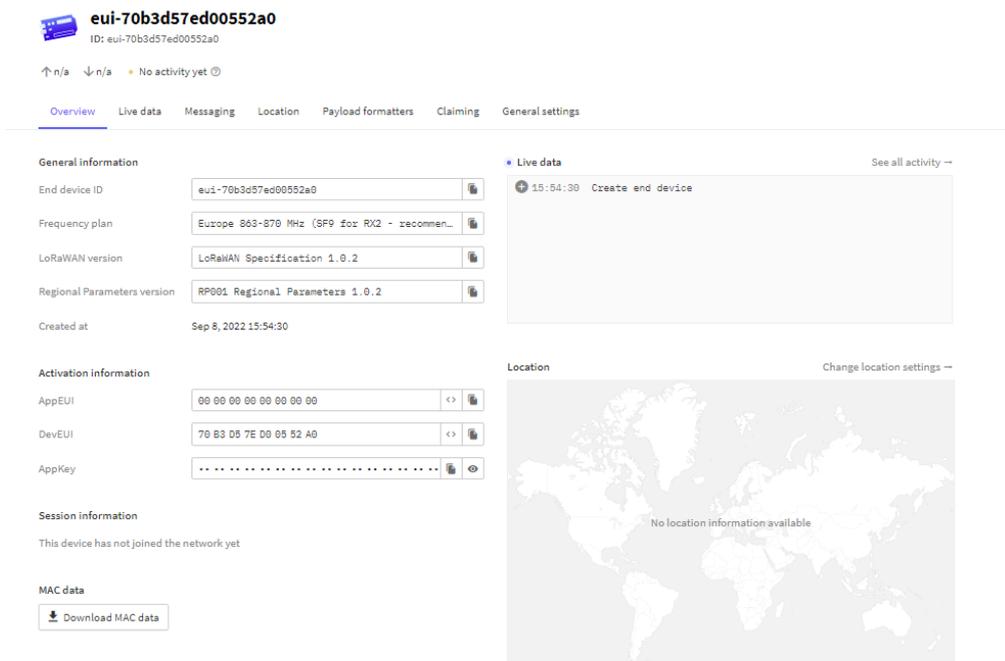


Figura 23. Vista general del nodo creado

4.2.2.3. Puerta de enlace

En este apartado veremos cómo se configurará la puerta de enlace en TTN, en nuestro caso, la Sentries RG1xx. Por ello, primero se debe conectar, ya sea por cable Ethernet o por Wifi al router para, a continuación, acceder al panel de control a través del siguiente enlace, <https://rg1xx296ccf.local/>, donde “296ccf” serán los últimos 6 dígitos de la dirección MAC del Ethernet. Se deberá aceptar el certificado de seguridad y se podrá acceder introduciendo el nombre de usuario sentrius con contraseña RG1xx.

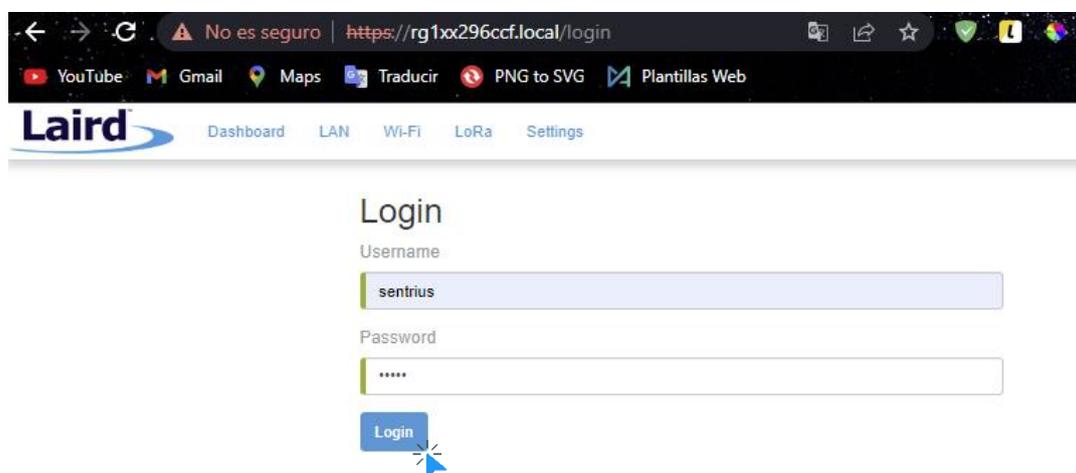


Figura 24. Login de Laird Connectivity

Una vez dentro, se debe entrar en la sección de LoRa y se elegirá la opción de The Things Network Legacy (TTN) y se hará clic en aplicar (se debe comprobar que la dirección del servidor es la misma que se generará en el siguiente paso).

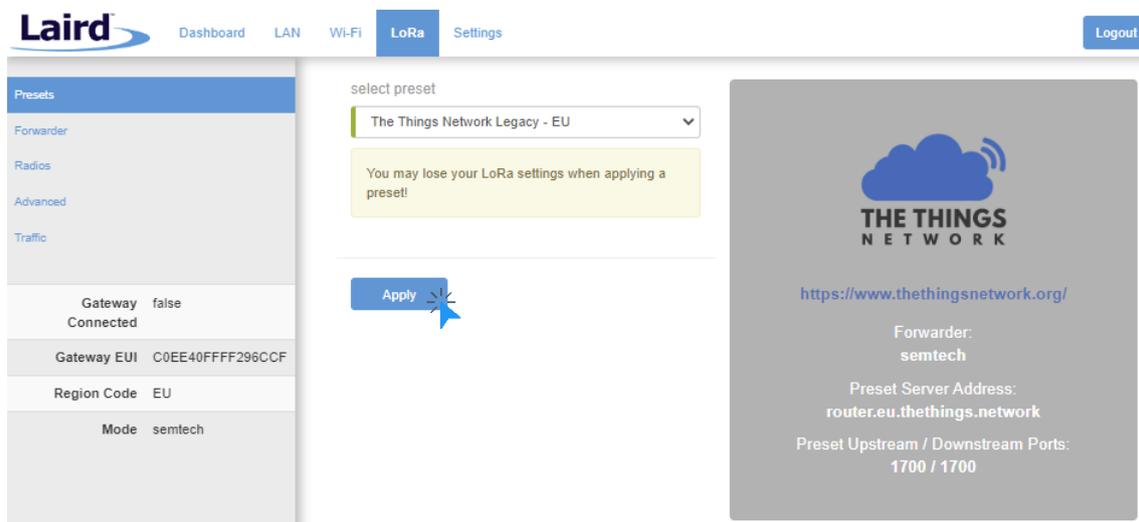


Figura 25. Se selecciona la opción The Things Network Legacy (TTN)

Ahora, se debe acceder de nuevo a la consola de TTN y se pulsará en crear una puerta de enlace y se rellenará con los datos correspondientes (se tiene en cuenta el Gateway EUI obtenido en el anterior paso).

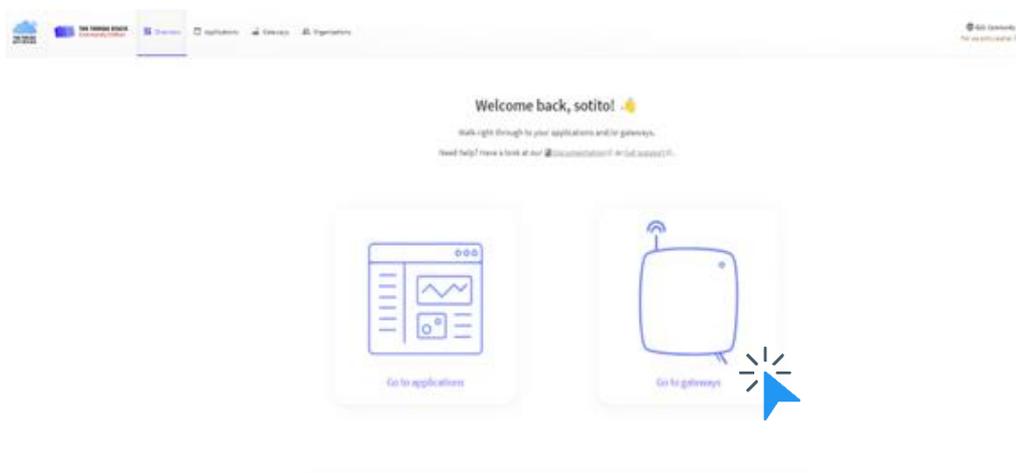


Figura 26. Pantalla principal de la consola TTN

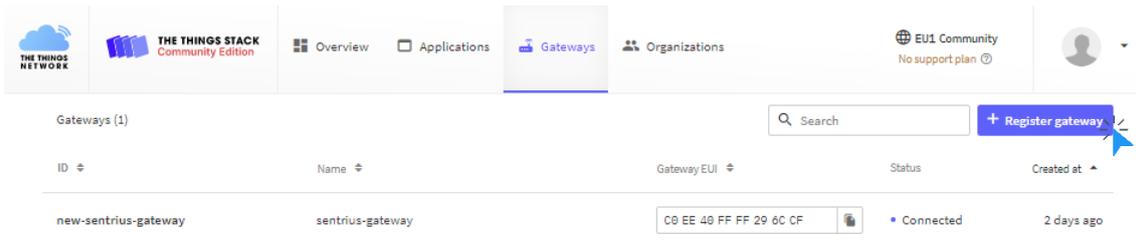


Figura 27. Se pulsa en añadir gateway

Register gateway

Register your gateway to enable data traffic between nearby end devices and the network. Learn more in our [Gateway Guide](#).

Gateway EUI *

C0 EE 40 FF FF 29 6C CF

Gateway ID *

eui-c0ee40fff296ccf

Gateway name

new-gateway

Frequency plan *

Europe 863-870 MHz (SF9 for RX2 - recommended)

Require authenticated connection

Choose this option eg. if your gateway is powered by [LoRa Basic Station](#)

Share gateway information

Select which information can be seen by other network participants, including [Packet Broker](#)

Share status within network

Share location within network

Figura 28. Se rellena el formulario y se hace clic en registrar

4.2.3. Ubidots

Por último, para añadir el servicio webhook de Ubidots, previamente se debe estar en posesión de una cuenta. Seguidamente, se entrará en el apartado de Dispositivos y la opción de Plugins.

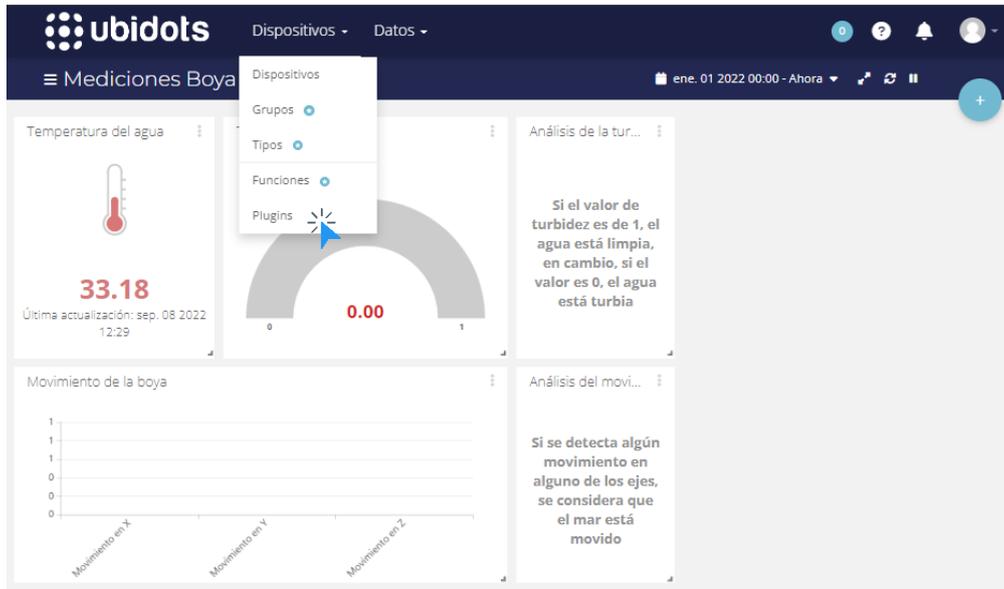


Figura 29. Se accede a los dispositivos para añadir un nuevo plugin

Se hará clic en el botón de + y se seleccionará el complemento de The Things Stack, se le proporcionará un nombre, así como un token nuevo que se puede encontrar en las Credenciales de API del usuario.

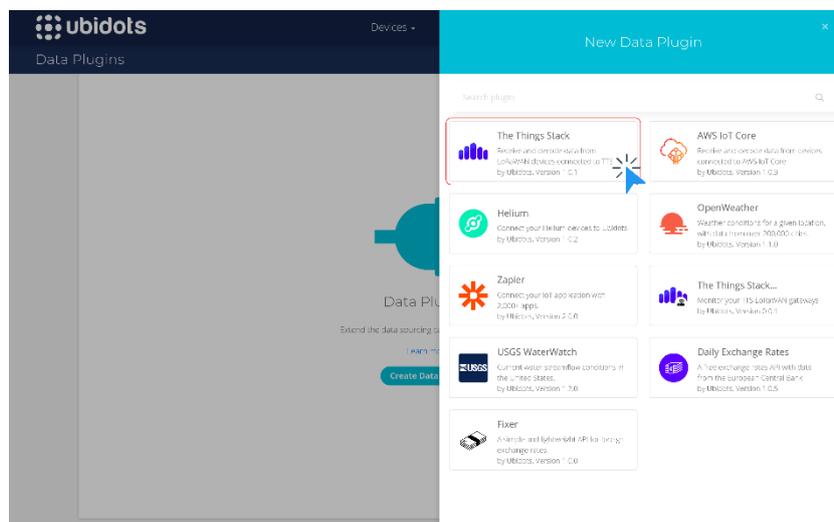


Figura 30. Se selecciona el complemento de The Things Stack

Lo siguiente que se debe realizar es dirigirse al apartado de Decoder Settings y coger la parte final del HTTPs Endpoint URL para posteriormente configurar el webhook en la consola de TTN. Otro paso que se debe seguir llegados a este punto, es añadir en la sección de Decoding Function el código decodificador que se mencionó con anterioridad en el apartado de creación de la aplicación en TTN.

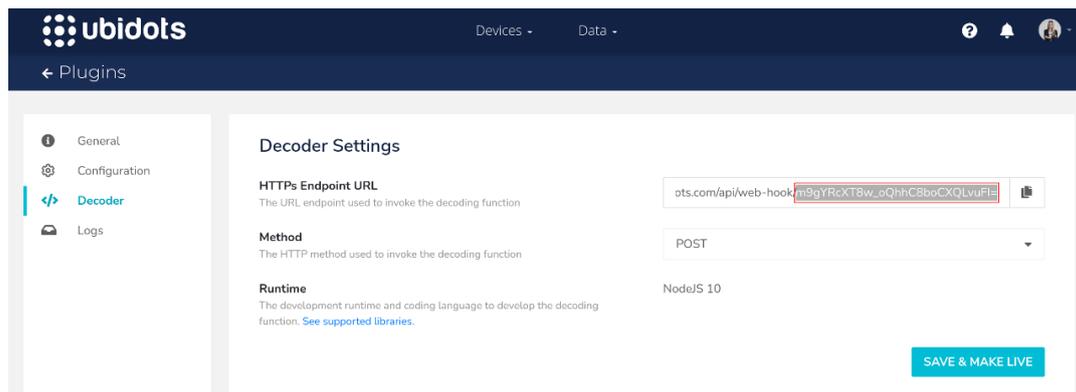


Figura 31. Se guarda la parte final de URL y se escribe la función decodificadora en la Decoding Function

Para poder añadir este servicio a la aplicación de TTN, se debe pulsar la opción de añadir webhook y rellenar el formulario con los datos necesarios para su vinculación.

Applications > Aplicación prueba

Aplicación prueba
ID: new-application-prueba

No recent activity ⓘ 1 End device 1 Collaborator 0 API keys

General information

Application ID: new-application-prueba

Created at: Sep 8, 2022 15:29:15

Last updated at: Sep 8, 2022 15:29:15

Live data See all activity →

- 15:54:30 eui-70b3d5... Create end device
- 15:51:45 new-applic... Issue DevEUI for application
- 15:29:15 new-applic... Create application

End devices (1)

Search Import end devices Add end device

| ID | Name | DevEUI | JoinEUI | Last activity |
|----------------------|------|---------------------|---------------------|---------------|
| eui-70b3d57ed00552a0 | | 70 B3 D5 7E 00 0... | 00 00 00 00 00 0... | Never |

Figura 32. Página general de la aplicación

Applications > Aplicación prueba > Webhooks

Webhooks (0) + Add webhook

| ID | Base URL | Template ID | Status | Created at |
|----------------|----------|-------------|--------|------------|
| No items found | | | | |

Figura 33. Se pulsa para añadir el complemento

Applications > Aplicación prueba > Webhooks > Add > Ubidots

Setup webhook for Ubidots
Integrate with Ubidots using Plugins
[About Ubidots](#) | [Documentation](#)

Webhook ID *
new-plugin

Plugin ID *
32543875t98743t9fw120
Unique identifier found in your Plugin's HTTPS URL

Ubidots token *
.....
Token used to authenticate Ubidots API requests

Create Ubidots webhook

Figura 34. Se rellena con los datos correspondientes

Finalmente, para observar las mediciones en el dashboard de Ubidots, se deberán añadir los widgets que se deseen teniendo en cuenta los datos que se reciben a través del dispositivo.

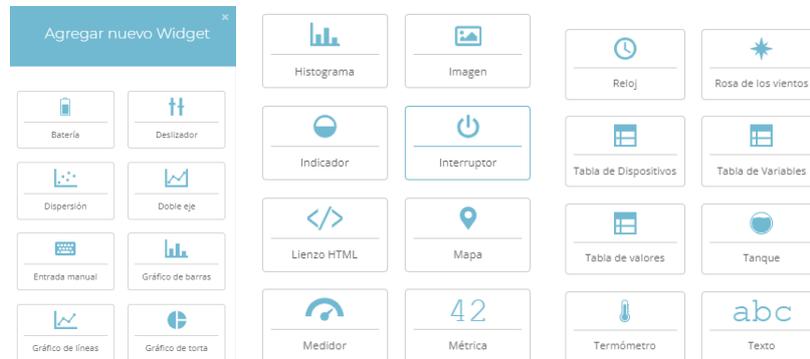


Figura 35. Diferentes tipos de widgets

Capítulo 5. Resultados

En este capítulo, se podrá ver cuál ha sido el resultado que se ha obtenido después de cargar el programa en el nodo y tener todo configurado y conectado. En primer lugar, cuando subimos el archivo .ino a la placa, en el COM recibiremos los datos obtenidos por los diferentes sensores de esta forma:

```
-----  
Mediciones  
-----  
Nivel de turbidez: Agua turbia  
Temperatura del agua: 2556 C°  
Movimiento de la boya:  
Xraw = 32767.00 Yraw = 15501.00 Zraw = 26519.00  
Xnorm = 0.00 Ynorm = 0.00 Znorm = 1680.08  
  
Sending: mac tx uncnf 1 000009FC000000000690  
Successful transmission  
-----  
Mediciones  
-----  
Nivel de turbidez: Agua turbia  
Temperatura del agua: 2543 C°  
Movimiento de la boya:  
Xraw = -404.00 Yraw = -2239.00 Zraw = 2107.00  
Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00  
  
Sending: mac tx uncnf 1 000009EF000000000000  
Successful transmission  
-----  
Mediciones  
-----  
Nivel de turbidez: Agua limpia  
Temperatura del agua: 2531 C°  
Movimiento de la boya:  
Xraw = -19.00 Yraw = -248.00 Zraw = -325.00  
Xnorm = 0.00 Ynorm = 0.00 Znorm = 0.00  
  
Sending: mac tx uncnf 1 000109E3000000000000  
Successful transmission  
-----  
Mediciones  
-----  
Nivel de turbidez: Agua turbia  
Temperatura del agua: 2500 C°  
Movimiento de la boya:  
Xraw = -20453.00 Yraw = -25768.00 Zraw = 354.00  
Xnorm = 0.00 Ynorm = -1647.04 Znorm = 0.00  
  
Sending: mac tx uncnf 1 000009C40000F9910000  
Successful transmission
```

Figura 36. Resultado por COM de Arduino

Como poder ver, nos muestra los datos que se recogen gracias a los sensores conectados, así como una verificación como que estas medidas han sido subidas correctamente al servidor de LoRoWAN. Una vez se han enviado los datos a la consola de TTN, en la pestaña de Live data, se observa en tiempo real la sucesión de las cadenas de bytes con las medidas que se han obtenido y están siendo decodificadas gracias al código implementado especialmente para poder leer estos datos.

Y estos datos se muestran en la consola de TTN de la siguiente forma:

```
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 20.5, turb: "Agua limpia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 23.18, turb: "Agua limpia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 27.62, turb: "Agua limpia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 28.31, turb: "Agua turbia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 28.62, turb: "Agua turbia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 29, turb: "Agua turbia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 28.93, turb: "Agua turbia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 28.93, turb: "Agua turbia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 28.87, turb: "Agua turbia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 28.87, turb: "Agua turbia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 28.75, turb: "Agua turbia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 28.62, turb: "Agua turbia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 28.62, turb: "Agua turbia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 28.62, turb: "Agua turbia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 28.62, turb: "Agua turbia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 28.62, turb: "Agua turbia" }
{ coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 28.62, turb: "Agua turbia" }
```

Figura 37. Datos en detalle

Como se puede apreciar, el nodo nos devuelve una secuencia de bytes que transformamos en estos datos. En este primer caso podemos ver cómo el estado del mar es tranquilo, pero se observa como desciende la temperatura a su vez qué el nivel de turbidez, por lo que se deduce que la calidad del agua pasa a ser óptima.

Ahora se presenta un segundo caso en el cual el estado del mar es bueno, pero en cierto punto comienza a ver movimiento en el agua, así como un ascenso de la temperatura, lo que puede llegar a indicar el paso de algún tipo de vehículo de navegación marítima.

| | |
|--|----------------|
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 27, turb: "Agua limpia" } | 00 01 0A 0C 0E |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 27.06, turb: "Agua limpia" } | 00 01 0A 9 |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 27.12, turb: "Agua limpia" } | 00 01 0A 9 |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 27.18, turb: "Agua limpia" } | 00 01 0A 9 |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 27.31, turb: "Agua limpia" } | 00 01 0A A |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 27.5, turb: "Agua limpia" } | 00 01 0A BE |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 27.68, turb: "Agua limpia" } | 00 01 0A D |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 27.56, turb: "Agua limpia" } | 00 01 0A C |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 27.43, turb: "Agua limpia" } | 00 01 0A B |
| { coordX: 0, coordY: 20, coordZ: 0, marea: "Mar movido", temperature: 27.31, turb: "Agua limpia" } | 00 01 0A AB |
| { coordX: 635.32, coordY: 635.52, coordZ: 20.2, marea: "Mar movido", temperature: 27.06, turb: "Agua limpia" } | |
| { coordX: 0, coordY: 7.65, coordZ: 9.57, marea: "Mar movido", temperature: 26.75, turb: "Agua limpia" } | 00 01 |
| { coordX: 0, coordY: 642.37, coordZ: 0, marea: "Mar movido", temperature: 26.37, turb: "Agua limpia" } | 00 01 E |
| { coordX: 0, coordY: 7.35, coordZ: 12.91, marea: "Mar movido", temperature: 25.75, turb: "Agua limpia" } | 00 0: |
| { coordX: 19.91, coordY: 20.11, coordZ: 0, marea: "Mar movido", temperature: 24.81, turb: "Agua limpia" } | 00 |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 23.18, turb: "Agua limpia" } | 00 01 09 0 |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 20.31, turb: "Agua limpia" } | 00 01 07 E |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 16.93, turb: "Agua turbia" } | 00 00 06 9 |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 16.5, turb: "Agua turbia" } | 00 00 06 72 |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 16.62, turb: "Agua limpia" } | 00 01 06 7 |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 16.75, turb: "Agua limpia" } | 00 01 06 8 |
| { coordX: 0, coordY: 0, coordZ: 0, marea: "Mar en calma", temperature: 17.18, turb: "Agua limpia" } | 00 01 06 B |

Figura 38. Datos en detalle 2

En tiempo real también podemos observar estos datos en el dashboard de Ubidots, donde se pueden ver los diferentes widgets creados para poder ver de forma sencilla las mediciones obtenidas, así como haber establecido unos avisos de que, en caso de sobrepasarlos, avisar de forma visual al usuario de que la calidad del agua no es la óptima para su uso en ese momento.

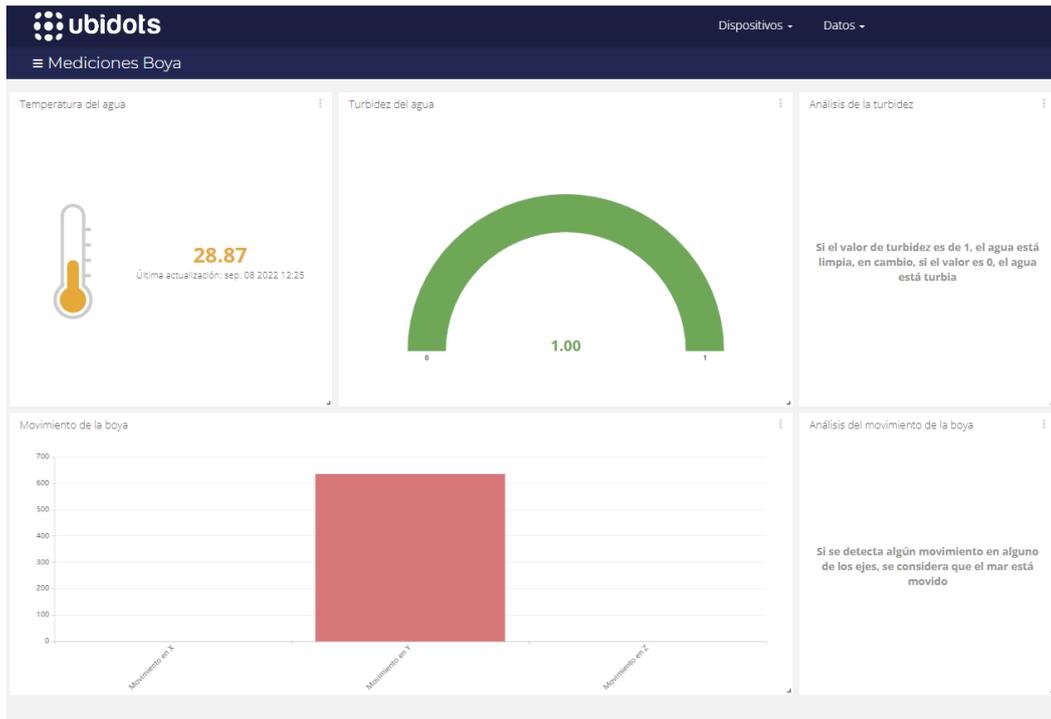


Figura 39. Dashboard con datos

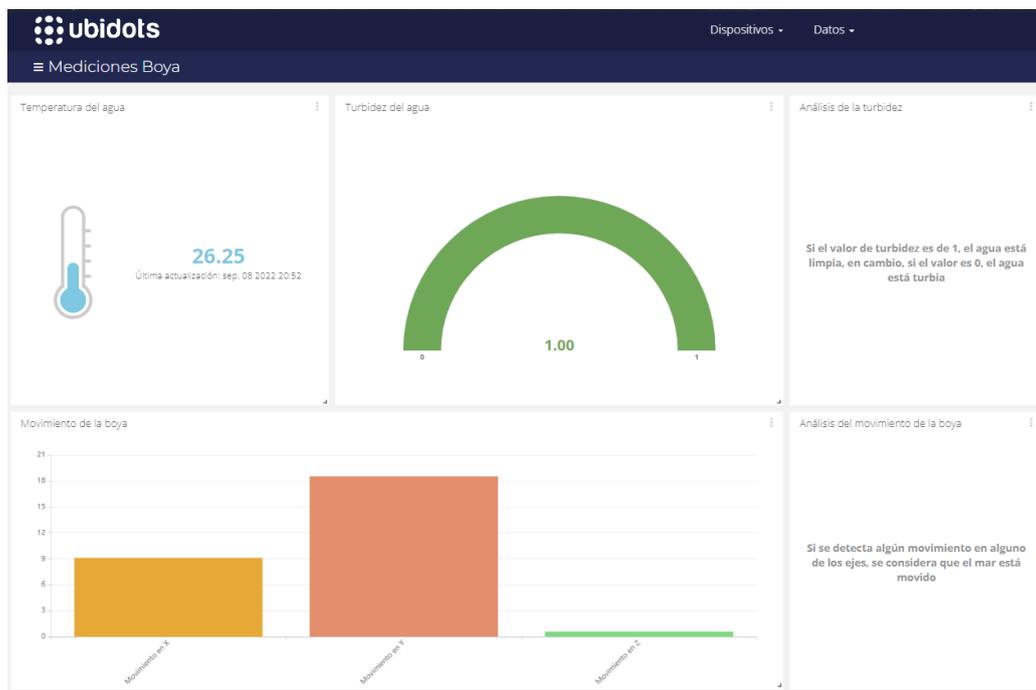


Figura 40. Dashboard con datos 2

Para advertir de forma visual y clara al usuario, sean hecho uso de unas reglas para en caso de obtener datos nocivos para el usuario del baño, cambiar el color de los widgets a tonos más cálidos.

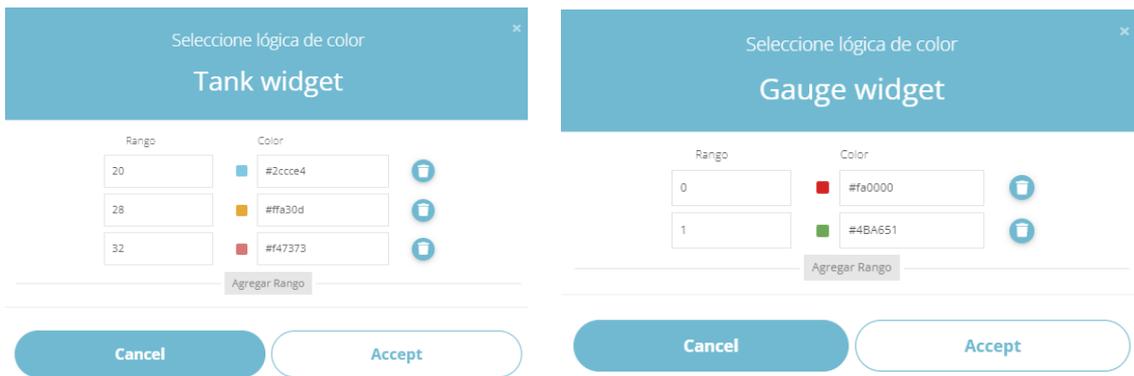


Figura 41. Rangos de advertencia para cada widget

Finalmente, en la propia web de Ubidots, podemos crear gráficas para observar los datos obtenidos por el nodo en un intervalo de tiempo determinado, por lo que se podría llegar a prescindir de una base de datos. Se muestra a continuación dos gráficas obtenidas durante un intervalo de aproximadamente 36 horas de las medidas de turbidez y temperatura de una de las pruebas realizadas.



Figura 42. Gráfica de turbidez



Figura 43. Gráfica de temperatura

Por último, se adjunta una imagen en el momento en el que se realizó una de las diversas pruebas y que determinaron el correcto funcionamiento del prototipo final.

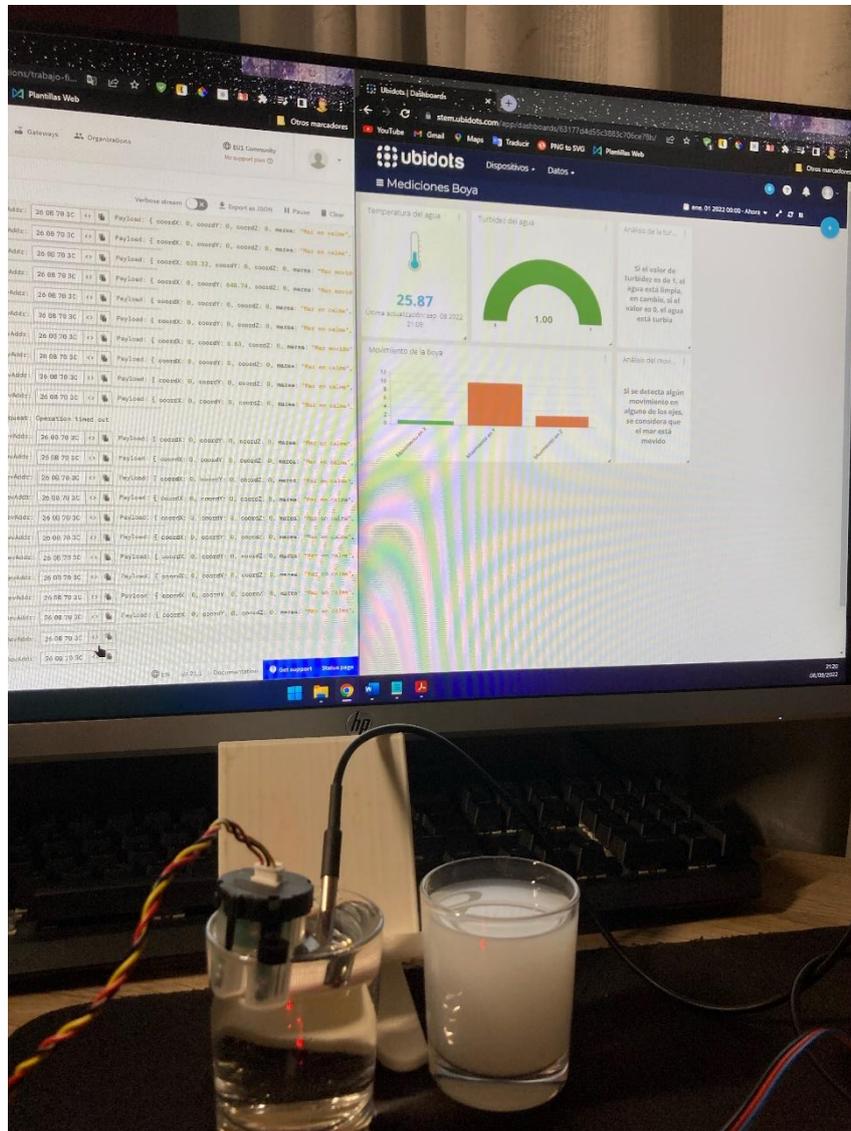


Figura 44. Imagen del momento de las mediciones

Capítulo 6. Conclusiones

Se ha desarrollado un dispositivo con el cual podemos obtener la calidad del agua en tiempo real cumpliendo así con los objetivos mínimos que se presentaban en un principio. Gracias al uso de la herramienta de LoRaWAN y a sus diversas opciones de integración, se ha conseguido realizar este prototipo de forma minimalista, es decir, gastando los mínimos recursos tanto de software como de hardware.

Para poder visualizar de forma intuitiva los datos obtenidos por el nodo, se ha logrado crear un tablero en Ubidots en el que podemos observar en tiempo real cuáles son las medidas en un momento determinado. Aparte, en la propia web, en el apartado de dispositivos, se puede obtener una gráfica con la recopilación de todas las medidas recogidas en un período de tiempo determinado.

Centrándonos en la parte de los datos, cómo se puede apreciar no se ha creado una base de datos para poder consultarlos en cualquier momento. Primeramente, sí que se tuvo la idea de crear una forma de poder almacenar esta información para que en caso de que el usuario desee consultarla pueda hacerlo fácilmente. Pero finalmente, se decidió prescindir de esta opción para quitarle complejidad al proyecto, ya que se analizó que, por parte de un usuario habitual a la zona de costa, a este solo le interesarían las medidas en tiempo real.

Este proyecto tenía un objetivo codicioso, en el cual se buscaba obtener una medida o un rango de valores sobre la concentración de bacterias *E. coli*, pero se llegó a la conclusión de que esta idea debía de ser aplazada, ya que no se contaba con el presupuesto necesario para llevar a cabo esta idea.

Teniendo en cuenta los resultados obtenidos, podemos afirmar que se ha obtenido un prototipo fiable y minimalista en el que se puede obtener la información necesaria para conocer la calidad del agua en tiempo real, así como una web visualmente sencilla para hacer más fácil al usuario su comprensión.

6.1. Líneas futuras

Este prototipo puede tener un desarrollo mayor y poder analizar diversas características del agua que determinan su calidad. Como se lleva comentando a lo largo de este documento, una de las ideas de las que primero se buscaría realizar sería el análisis periódico de concentración de bacteria *E. coli*, ya que últimamente está teniendo una gran importancia por su recurrente aparición en nuestras costas.

Otro factor para tener en cuenta, aunque dependería de la costa (ya sea mar u océano), sería el nivel de salinidad y pH del agua. Ambos datos determinan dos de las características físicas del agua más importantes para tener presente a la hora de medir la calidad de esta y al tratarse de zonas en las que los usuarios sí bañan e incluso pueden llegar a ingerir, estos datos pueden ser claves a la hora de saber si existe algún riesgo para ellos. Dentro de estos datos, podríamos llegar a incluir un análisis del color del agua, ya que nuestras costas suelen estar amenazadas por el vertido de diferentes sustancias provenientes de fábricas e industrias de las zonas colindantes a ellas. Por ello, podemos lograr obtener información cómo sería la presencia petróleo en las zonas de baño y/o sustancias vertidas por las industrias en las zonas de pesca.

Por último, otra de las mejoras de las que podemos añadir a este prototipo sería la conexión con la información que prestan los ayuntamientos sobre el color de las banderas de cada una de las playas para así tener un informe completo en tiempo real de las zonas de baño. Aparte de ello, en la propia aplicación web se podrían añadir diversos factores (los cuales han sido propuestos en este apartado) y tener toda la información agrupada en un único tablero y que es accesible fácilmente.

Bibliografía

- [1]. Criterios Playas [Internet]. Banderaazul.org. [consultado el 8 de mayo de 2022]. Disponible en: <https://www.banderaazul.org/criterios-playas>
- [2]. E. coli - Síntomas y causas - Mayo Clinic [Internet]. Mayo Clinic.org. [consultado el 8 de mayo de 2022]. Disponible en: <https://www.mayoclinic.org/es-es/diseases-conditions/e-coli/symptoms-causes/syc-20372058>
- [3]. Amazon.es. [consultado el 15 de mayo de 2022]. Disponible en: <https://www.amazon.es/AquaVial-Coli-Coliform-Prueba-Paquete/dp/B07QMD9DRF>
- [4]. Salario para Ingeniero Junior en España - Salario Medio [Internet]. Talent.com. [consultado el 18 de mayo de 2022]. Disponible en: <https://es.talent.com/salary?job=Ingeniero+junior>
- [5]. Unican.es. [consultado el 22 de mayo de 2022]. Disponible en: <https://repositorio.unican.es/xmlui/bitstream/handle/10902/14653/410397.pdf?sequence=1&isAllowed=y>
- [6]. Redalyc.org. [consultado el 27 de mayo de 2022]. Disponible en: <https://www.redalyc.org/pdf/539/53908206.pdf>
- [7]. Ballart J, Antonio D. Estudio de la cobertura de la modulación LoRa en la banda de 915[Mhz] en un ambiente urbano. 2018 [consultado el 3 de junio de 2022]; Disponible en: <https://repositorio.uchile.cl/handle/2250/168163>
- [8]. ¿Que nos dice la turbidez sobre la calidad del agua potable? [Internet]. Higieneambiental.com. [consultado el 18 de junio de 2022]. Disponible en: <https://higieneambiental.com/aire-agua-y-legionella/que-nos-dice-la-turbidez-sobre-la-calidad-del-agua-potable>

- [9]. Tutorial sensor digital de temperatura DS18B20 [Internet]. Naylamp Mechatronics - Perú. [consultado el 18 de junio de 2022]. Disponible en: https://naylampmechatronics.com/blog/46_tutorial-sensor-digital-de-temperatura-ds18b20.html
- [10]. Ecuarobot.com, View all posts by Ecuarobot.com →. Medición de la turbidez del agua para determinar la calidad del agua mediante Arduino y el sensor de turbidez [Internet]. EcuRobot. 2021 [consultado el 20 de junio de 2022]. Disponible en: <https://ecuarobot.com/2021/03/24/medicion-de-la-turbidez-del-agua-para-determinar-la-calidad-del-agua-mediante-arduino-y-el-sensor-de-turbidez/>
- [11]. The Things Uno Development Board EU868 [Internet]. Connected Things. [consultado el 20 de junio de 2022]. Disponible en: <https://connectedthings.store/gb/lorawan-development/the-things-uno-development-board.html>
- [12]. Módulo MPU6050: Acelerómetro, Giroscopio I2C [Internet]. Naylamp Mechatronics - Perú. [consultado el 20 de junio de 2022]. Disponible en: <https://naylampmechatronics.com/sensores-posicion-inerciales-gps/33-modulo-mpu6050-acelerometro-giroscopio-i2c.html>
- [13]. Home [Internet]. Thethingsindustries.com. [consultado durante el proceso de la creación del proyecto]. Disponible en: <https://www.thethingsindustries.com/docs/>
- [14]. Storefront y WooCommerce CC. LoraWan [Internet]. Productos IoT. 2018 [consultado el 8 de julio de 2022]. Disponible en: <http://productos-iot.com/lorawan-3/>

- [15]. Silíceo. Lora y LoraWAN redes inalámbricas de hasta 20km para Internet de las Cosas (IoT) [Internet]. SILICEO Tienda Online Blog & Shopping. 2021 [consultado el 8 de julio de 2022]. Disponible en: <https://www.siliceo.es/lora-y-lorawan-redes-inalambricas-de-hasta-20km-para-internet-de-las-cosas-iot/>
- [16]. Tecnología LoRA y LoRaWAN [Internet]. Catsensors.com. [consultado el 10 de julio de 2022]. Disponible en: <https://www.catsensors.com/es/lorawan/tecnologia-lora-y-lorawan>
- [17]. Rivas J. Nodos en LoRa y LoRaWAN [Internet]. LoRa Panamá. 2021 [consultado el 10 de julio de 2022]. Disponible en: <http://lora-panama.com/nodos-en-lora-y-lorawan/>
- [18]. Ubidots IoT Application Development Platfrom [Internet]. Sigfox.com. [consultado el 15 de julio de 2022]. Disponible en: <https://partners.sigfox.com/products/ubidots-for-business>