



## **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

### **Facultad de Ingeniería en Electricidad y Computación**

“Sistema de monitoreo en la nube de la calidad del agua en  
piscinas camarones a través de una LPWAN usando  
protocolo LoRaWAN”

#### **INFORME DE PROYECTO INTEGRADOR**

Previo a la obtención del Título de:

#### **INGENIERIA EN ELECTRÓNICA Y TELECOMUNICACIONES**

RICARDO ANDRES MACÍAS YÉPEZ  
JOHNNY PAUL CHANG MACÍAS

GUAYAQUIL – ECUADOR

AÑO: 2018

## **DECLARACIÓN EXPRESA**

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

---

Johnny Paul Chang  
Macías

---

Ricardo Andres  
Macías Yépez

## **RESUMEN**

Para el desarrollo tecnológico de la industria camaronera, quienes están en búsqueda de nuevas tecnologías que les puedan facilitar sus procesos de cría y estudio de camarón, se diseñó un sistema de monitoreo de calidad de agua con el fin de facilitar la obtención de datos vitales para la producción y crecimiento del camarón. El sistema consiste en dos nodos que recolectan los datos en las piscinas camaroneras para ser enviados mediante protocolo LoRaWAN en una banda no licenciada a un equipo Gateway, este sube la información a un servidor en la nube. Una vez que los datos están en la nube, son procesados para ser mostrados de manera amigable a través de una interfaz gráfica, a la cual se puede acceder desde cualquier punto con acceso a internet.

Se implementaron dos prototipos de nodos que sensaron la calidad de agua, y comparándolo con un registro manual se verificó que los datos recibidos eran correctos. También fue probado en piscinas de larvas en laboratorios del sector donde hubo mucho interés por el prototipo. Este prototipo facilita la obtención de datos, ahorra tiempos y costos de operación en comparación con la medición manual, y mantiene informados a los administradores sin necesidad de estar en el sitio.

## **ABSTRACT**

For the technological development of the shrimp industry, who are in search of new technologies that can facilitate their processes of breeding and study of shrimp, a water quality monitoring system was designed in order to facilitate the collection of vital data for the production and growth of shrimp. The system consists of two nodes that collect the data in the shrimp ponds to be sent through the LoRaWAN protocol in an unlicensed band to a Gateway which uploads the information to a server in the cloud, once the data is in the cloud they are processed to be displayed in a friendly way through a graphical interface which can be accessed from any point with internet access. Two prototypes of nodes were implemented, which measured the quality of water and, comparing it with a manual census, it was verified that the received data was correct. It was also tested in larval pools in laboratories from the sector where there was a lot of interest in the prototype. This prototype facilitates data collection and saves operating times compared to manual measurement, saves costs of manual measurements and keeps administrators informed without having to be on site.

## ÍNDICE GENERAL

RESUMEN.....	I
<i>ABSTRACT</i> .....	II
ÍNDICE GENERAL .....	III
ABREVIATURAS.....	V
SIMBOLOGÍA.....	VI
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS.....	IX
CAPÍTULO 1.....	1
1.    Introducción .....	1
1.1    Descripción del problema .....	1
1.2    Objetivos.....	2
1.2.1    Objetivo General .....	2
1.2.2    Objetivos Específicos .....	3
1.3    Marco Teórico.....	3
1.3.1    Red de área amplia de baja potencia .....	3
1.3.2    Transmisión inalámbrica de largo alcance.....	3
1.3.3    Modos de conexión en una red LoRaWAN .....	6
1.3.4    Puerta de enlace .....	7
1.3.5    Sensores de calidad de agua .....	8
CAPÍTULO 2.....	12
2.    ANALISIS, DISEÑO E IMPLIMENTACION DE RED IoRaWAN .....	12
2.1    Criterios de diseño de red de sensores piscinas camaronesas. ....	12
2.2    Análisis de componentes de la red de sensores. ....	13
2.3    Análisis de la camaronera y diseño físico de la red de sensores. ....	15

2.4	Implementación de la red de sensores.....	19
2.5	Implementación de la nube.....	22
2.5.1	Configuración de la nube de TTN.....	22
2.5.2	Configuración de la nube de Ubidots for Education.....	26
2.6	Procesamiento de datos.....	29
CAPÍTULO 3.....		29
3.	ANÁLISIS DE Resultados.....	29
3.1	Pruebas en campo .....	29
3.1.1	Primera visita .....	29
3.1.2	Segunda visita.....	32
3.2	Análisis de costos.....	36
CAPITULO 4.....		41
4.	Conclusiones Y RECOMENDACIONES.....	41
4.1	Conclusiones .....	41
4.2	Recomendaciones.....	42
BIBLIOGRAFÍA.....		44

## **ABREVIATURAS**

ESPOL	Escuela Superior Politécnica del Litoral
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Area Network
PIB	Producto Interno Bruto
M2M	Machine to Machine
IoT	Internet of Things
OTAA	Over the Air Activation
TTN	The Things Network
API	Application programming interface
AES	Advanced Encryption Standard
LTE	Long Term Evolution
BLE	Bluetooth low energy
LFSL	Free space path loss
EIRP	Potencia de transmisión efectiva del equipo.
IDE	Integrated Development Environment

## **SIMBOLOGÍA**

MHz	Megahercio
Km	Kilómetro
mg/L	Miligramos por Litro
C°	Grados Centígrados
Kbps	Kilobit por segundo

# ÍNDICE DE FIGURAS

Figura 1.1 Ejemplo de estructura de red [5].....	5
Figura 1.2 Diagrama de conexión sensor de oxígeno disuelto [8]. .....	9
Figura 1.3 Diagrama de conexión sensor de temperatura [9].....	10
Figura 1.4 Temperatura [°C] vs error, del sensor de temperatura. ....	11
Figura 2.1 Nodo prototipo. ....	13
Figura 2.2 Modulo MKRWAN1300. ....	14
Figura 2.3 Gateway “The Things Network”.....	14
Figura 2.4 Kit Voltaic Systems panel solar y batería. ....	15
Figura 2.5 Distancias nodos a Gateway en piscinas. ....	16
Figura 2.6 Pruebas de Nodo-Gateway con línea de vista.....	16
Figura 2.7 Alimentadoras automáticas. ....	19
Figura 2.8 Mediciones de oxígeno disuelto y.....	20
Figura 2.9 Nodo durante mediciones en.....	21
Figura 2.10 Consola de TTN. ....	22
Figura 2.11 Plantilla de nueva aplicación. ....	22
Figura 2.12 Tabla de aplicaciones.....	23
Figura 2.13 Device EUI de un nodo MKR1300 vía serial.....	23
Figura 2.14 Plantilla de registro de equipo. ....	24
Figura 2.15 Lista de dispositivos finales (Arduinos) registrados en TTN. ....	24
Figura 2.16 Selección de tipo de conexión. ....	25
Figura 2.17 Ingreso de dirección y llave del controlador.....	25
Figura 2.18 Mensaje de conexión exitosa. ....	26
Figura 2.19 Página principal Ubidots Education. ....	26
Figura 2.20 Ventana “Devices” en Ubidots. ....	27
Figura 2.21 Variables del “device” en Ubidots. ....	27
Figura 2.22 Opciones de “widgets” en Ubidots. ....	28
Figura 2.23 Ventana de “Events” en Ubidots. ....	28
Figura 2.24 Diagrama de bloques general de la red.....	29
Figura 2.25 Paquetes recibidos por los servidores de TTN enviados por los nodos. ....	30
Figura 2.26 Captura de datos usando Python. ....	31
Figura 2.27 Información enviada por los nodos en la interfaz gráfica.....	31

Figura 3.1 Oxígeno [mg/L] vs tiempo de la medición, nodo 1.....	29
Figura 3.2 Oxígeno [mg/L] vs tiempo de la medición, nodo 2.....	30
Figura 3.3 Temperatura [°C] vs tiempo de la medición, nodo 1.....	31
Figura 3.4 Temperatura [°C] vs tiempo de la medición, nodo 2.....	31
Figura 3.5 Temperatura [°C] vs tiempo de la medición, prototipo y paramétristas. .....	32
Figura 3.6 Oxígeno [mg/L] vs tiempo de la medición, prototipo y paramétristas. .....	33

## **ÍNDICE DE TABLAS**

Tabla 3.1. Tabla estadística de resultados temperatura y oxígeno (prototipo) .....	34
Tabla 3.2 Tabla estadística de resultados temperatura y oxígeno (mediciones de la camaronera) .....	34
Tabla 3.3. Tabla de costos de adquisición de los nodos. ....	36
Tabla 3.4. Tabla de adquisición del Gateway. ....	37
Tabla 3.5. Tabla de costos de operación y mantenimiento.....	37
Tabla 3.6. Tabla de costos. ....	38

# CAPÍTULO 1

## 1. INTRODUCCIÓN

La venta de camarón en el Ecuador representa el segundo mayor ingreso no petrolero, y como país en vía a la modernización, se plantea el reto de facilitar procesos con miras a la optimización. Debido al crecimiento de este sector (según cifras de la Corporación Financiera Nacional) [1], actualmente las camaroneras están implementando sistemas de telecomunicaciones, recolección de datos, aplicaciones móviles, entre otros, con el propósito de automatizar procesos debido a que esto les brinda competitividad en el mercado.

Dicho lo anterior, este proyecto plantea una propuesta para facilitar la recolección y monitoreo de datos de parámetros de calidad del agua en la que se desarrolla el camarón, a través de un sistema de nodos sensores interconectados a distancia usando el protocolo LoRaWAN (red de largo alcance). Esto permitirá a los encargados del control del proceso de cría, conocer, a través de una aplicación web y en tiempo real, el estado de las piscinas donde se desarrollan las distintas etapas desde que el camarón es una larva hasta llegar a su tamaño comercial, para finalmente su pesca. Lo cual es importante debido a que permitirá reaccionar más rápidamente ante eventualidades y mitigar pérdidas económicas.

### 1.1 Descripción del problema

El sector camaronero aporta con el 0.72% del PIB en el Ecuador siendo así el segundo mayor aporte no petrolero [1], debido a esto, es importante para el país ser más competitivos en los mercados internacionales, por lo que es necesaria una sinergia entre estado y empresa. El estado debe crear leyes que fomenten el desarrollo e investigación y la empresa debe mejorar su productividad [2]. Esto plantea un desafío para las empresas camaroneras, por lo que se pueden optar medidas para la optimización de los procesos, una de ellas es el uso de la tecnología, reduciendo así tiempos de operación, riesgo al error humano y la inversión en mano de obra.

Existen etapas del proceso, como la medición de la calidad del agua, que son de vital importancia y podrían ser mejoradas. Es importante para el proceso de cultivo y cría del camarón medir pH, temperatura y oxígeno disuelto, puesto que brindan las condiciones propicias para que el camarón se alimente y se mantenga sano [3]. Considerando que la industria camaronera es pilar fundamental en la economía del país, se puede optimizar el proceso de medición de oxígeno disuelto en el agua, parámetro que está altamente correlacionado con la calidad del producto, con el fin de reducir los tiempos de operación del proceso de medición y reducir costos, debido a que actualmente se movilizan una o varias personas a cada piscina a sensar dicho parámetro. Por otra parte, es fundamental para la industria camaronera aumentar la productividad y la eficiencia para poder ser competitivos, en ese sentido, es imperativo implementar alternativas de mejora continua en los diferentes procesos y una forma de hacerlo es a través de la tecnología.

Hoy en día ya existen soluciones similares para comunicaciones inalámbricas en el mercado como lo son Zigbee, Bluetooth y Wi-Fi, sin embargo, estas representan costos elevados que superan los miles de dólares. Siendo empáticos con la problemática, lo que se plantea en el presente proyecto no solo es conveniente en términos de costo, sino que también en términos energéticos ya que el consumo es mucho más eficiente frente a las soluciones planteadas con otras tecnologías ya existentes.

## 1.2 Objetivos

### 1.2.1 Objetivo General

Automatizar el proceso de medición de parámetros de calidad del agua, a través de la implementación de una red de baja potencia con protocolo LoRaWAN y monitoreo en la nube en tiempo real.

### **1.2.2 Objetivos Específicos**

- Realizar una inspección de la camaronera para realizar el diseño de la red LoRaWAN.
- Probar los equipos “The Things Network” en campo y analizar resultados de limitaciones y condiciones.
- Brindar al sistema conectividad a la nube con el propósito de monitorear la calidad del agua en tiempo real.
- Brindar auto sustentabilidad a nuestro sistema a través de alimentación energética por paneles solares.
- Mostrar los datos en tiempo real a través de una interfaz gráfica.

## **1.3 Marco Teórico**

### **1.3.1 Red de área amplia de baja potencia**

La red de área amplia de baja potencia (LPWAN, por sus siglas en inglés) es una tecnología de red de área amplia inalámbrica que interconecta dispositivos de bajo ancho de banda, con bajas tasas de bits en largos rangos y alimentados por batería. Las redes LPWAN, creadas para redes de máquina a máquina (M2M) e internet de las cosas (IoT), operan a un costo menor con mayor eficiencia de energía que las redes móviles tradicionales. También pueden admitir un mayor número de dispositivos conectados en un área más grande. Las LPWAN pueden admitir paquetes de tamaños de 10 a 1.000 bytes, a velocidades de enlace ascendente de hasta 200 Kbps. El largo alcance de LPWAN varía de 2 km a 1.000 km, dependiendo de la tecnología (protocolo de comunicación, antenas, amplificadores y tipos de encriptación) [4].

### **1.3.2 Transmisión inalámbrica de largo alcance**

LoRa (Long Range) es el tipo de modulación en radiofrecuencia patentado por Semtech y entre sus principales ventajas y características se encuentran:

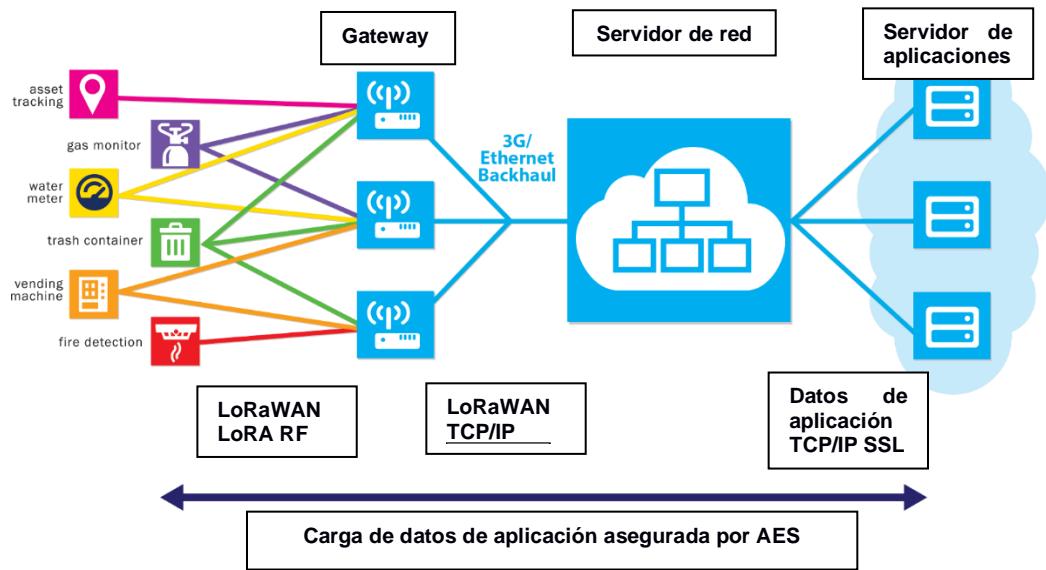
- Alta tolerancia a las interferencias.

- Alta sensibilidad para recibir datos (-168dB).
- Bajo Consumo (hasta 10 años con una batería).
- Largo alcance 10 a 20km.
- Baja transferencia de datos (hasta 255 bytes).
- Conexión punto a punto.
- Frecuencias de trabajo: 915MHz América, 868 Europa, 433 Asia.

Todo esto hace a la tecnología ideal para conexiones a grandes distancias y para redes de IoT que se pueden utilizar en ciudades inteligentes, lugares con poco cobertura celular o redes privadas de sensores o actuadores [5]. LoRaWAN es un protocolo de red que usa la tecnología LoRa para comunicar y administrar dispositivos LoRa, se compone de dos dispositivos principalmente: equipos Gateway, encargados de recibir y enviar información a los nodos; y nodos, dispositivos finales que envían y reciben información hacia el Gateway [5].

Las principales características de LoRaWAN son:

- Topología estrella.
- Alcance de entre 10km a 15km en línea de vista.
- Encriptación AES 128.
- Soporte para nodos de clase A, B y C.
- Administración de dispositivos.
- Redes públicas y privadas.
- Bajo consumo y largo alcance.
- Baja transferencia de datos (hasta 242 bytes).



**Figura 1.1 Ejemplo de estructura de red [5].**

En la figura 1.1 se muestra la estructura de una red LoRaWAN clásica que se compone de una serie de nodos que se conectan a gateways y estos envían todos los datos a un servidor. Luego por medio de una interfaz de programación se descripta y entrega los datos a una aplicación final para el usuario [5].

En protocolo LoRaWAN existen tres tipos de clases de nodo:

- Clase A: La más soportada en casi todos los dispositivos. Este tipo de clase ofrece el mayor ahorro de energía debido a que solo entra en modo escucha (llamado ventana RX) después de enviar datos hacia el Gateway, por eso es ideal para dispositivos que usan una batería.
- Clase B: Este tipo de dispositivos tiene las ventanas de recepción con base a tiempos predeterminados con el Gateway. Este tipo de nodos puede usar una batería o una fuente externa dependiendo de los tiempos asignados de escucha.
- Clase C: Ofrece el menor ahorro de energía debido a que siempre está en modo escucha y solo cuando es necesario en modo transmitir. La recomendación es usarlo en dispositivos que cuentan con una fuente externa de alimentación [6].

### 1.3.3 Modos de conexión en una red LoRaWAN

Existen dos maneras de conectarse a una red LoRaWAN, para lograr entrar a la red se requieren una serie de claves y número de identificación por parte del nodo para lograr el correcto funcionamiento de la red y mantener su seguridad [6].

El modo OTAA (Over-The-Air-Activation) es la manera más segura de conectarse a la red. Los parámetros de configuración son:

- DevEUI: Este es un identificador de fábrica, hace a cada dispositivo único. Esta configuración en algunos dispositivos se puede ajustar.
- AppEUI: Identificador de aplicación único utilizado para agrupar objetos. Esta dirección de 64 bits se utiliza para clasificar los dispositivos por aplicación.
- AppKey: Es una clave secreta AES de 128bits compartida entre el dispositivo periférico y la red. Se utiliza para determinar las claves de sesión. Esta configuración se puede ajustar.

Con estos datos (DevEUI, AppEUI o AppKey), la conexión se realiza de la siguiente manera:

1. El nodo solicita un inicio de sesión a la red con los datos de configuración y abre la ventana de recepción.
2. El Gateway recibe la solicitud y la envía al servidor.
3. El servidor verifica que el nodo esté dado de alta y la llave de encriptación sea correcta.
4. Si es correcta asigna una sesión temporal y la envía por medio del Gateway al nodo, si los datos son incorrectos rechaza el join.
5. El nodo recibe la sesión temporal y puede enviar datos a la red.

La principal ventaja de la conexión tipo OTAA es la seguridad, ya que la sesión se crea en el aire y se renueva cada vez que el dispositivo pierde la

conexión, es apagado o reiniciado, esto dificulta que alguien pueda robar la sesión y clonar el dispositivo [6].

El modo ABP (Activation By Personalization) es el modo más sencillo de conexión, los parámetros de conexión son:

- DevAddress: Dirección lógica (equivalente a una dirección IP) que se utiliza para toda comunicación posterior con la red.
- NetworkSessionKey: Clave de cifrado entre el dispositivo y el operador utilizado para las transmisiones y para validar la integridad de los mensajes.
- ApplicationSessionKey: Clave de cifrado entre el dispositivo y el operador (a través de la aplicación) utilizada para transmisiones y para validar la integridad de los mensajes.

Con estos parámetros la conexión se realiza de la siguiente manera:

1. El dispositivo envía datos hacia al Gateway.
2. El Gateway valida que los datos correspondan a la sesión.
3. Si es correcta la sesión se procesan los datos, sino se rechazan.

La principal ventaja de este tipo de conexión es que no se requiere hacer un join a la red para poder enviar datos. La confirmación en el servidor no es necesaria, ya que la sesión está manualmente asignada. Para dispositivos que están en movimiento o no tienen excelente recepción, este tipo de conexión es idónea. La desventaja es que al encontrarse la llave de encriptación en el dispositivo pudiera ser extraída y clonada por un atacante [6].

#### **1.3.4 Puerta de enlace**

La puerta de enlace o Gateway es el concentrador al que se conectan todos los dispositivos en una red LoRaWAN. Al igual que en una red Wi-Fi, el router se encarga de recibir y enviar a internet la información de las

computadoras y viceversa cuando recibe información desde internet. El Gateway se encarga de comunicarse de forma bidireccional vía LoRa con los nodos finales.

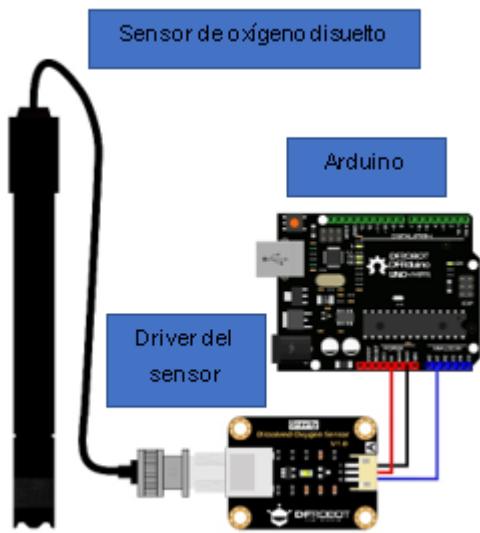
Hay una gran variedad de gateways en el mercado, sin embargo, la mayoría comparten características como [7]:

- Para América y Europa chip SX301 (para 915 y 868 MHz).
- 8 canales de recepción simultánea.
- 1 canal de transmisión.
- Sistema operativo basado en Linux.

### 1.3.5 Sensores de calidad de agua

Para la medición de calidad de agua en camarones se utilizan 2 tipos de sensores los cuales miden el oxígeno disuelto en el agua y temperatura.

El sensor de oxígeno DFRobot modelo SEN0237, es un kit de sensor de oxígeno disuelto, que es compatible con los microcontroladores Arduino. Este producto se utiliza para medir el oxígeno disuelto en el agua, para reflejar la calidad del agua. Se utiliza ampliamente en muchas aplicaciones de calidad del agua, como acuicultura, monitoreo ambiental, ciencias naturales, etc. La sonda es una sonda galvánica, no necesita tiempo de polarización y está disponible en cualquier momento. La solución de llenado y la tapa de la membrana son reemplazables, lo que conlleva un bajo costo de mantenimiento. La placa del convertidor de señal es plug-and-play, y tiene una buena compatibilidad. Se puede integrar fácilmente a cualquier sistema de control o detección [8].



**Figura 1.2 Diagrama de conexión sensor de oxígeno disuelto [8].**

En la figura 1.2 se muestra la conexión del sensor de oxígeno a la placa controladora Arduino, conectando alimentación (cable rojo) al pin de 3-5V, tierra (cable negro) al pin Ground, y análogo (cable azul) a uno de los puertos análogos (A0-A5).

### Especificaciones técnicas:

Sensor de oxígeno disuelto:

- Tipo: sonda galvánica.
- Rango de detección: 0 ~ 20mg / L.
- Tiempo de respuesta: hasta un 98% de respuesta completa, dentro de los 90 segundos.
- Rango de presión: 0 ~ 50PSI.
- Vida útil del electrodo: 1 año (uso normal).
- Período de mantenimiento: Período de reemplazo de la tapa de la membrana: 1 ~ 2 meses (en agua fangosa); 4 ~ 5 meses (en agua limpia). Período de reemplazo de la solución de llenado: Una vez al mes.
- Longitud del cable: 2 metros.
- Conector de sonda: BNC.

Controladora del sensor (driver):

- Voltaje de funcionamiento: 3.3 ~ 5.5V.
- Señal de salida: 0 ~ 3.0V.
- Conector del cable: BNC.
- Conector de señal: interfaz analógica de gravedad.
- Dimensión: 42mm x 32mm.

El sensor de temperatura DS18B20 para Arduino se usa para medir la temperatura en ambientes húmedos y dentro del agua, debido a que viene con una sonda impermeable. Utiliza protocolo 1-Wire para la comunicación con el controlador Arduino [9].

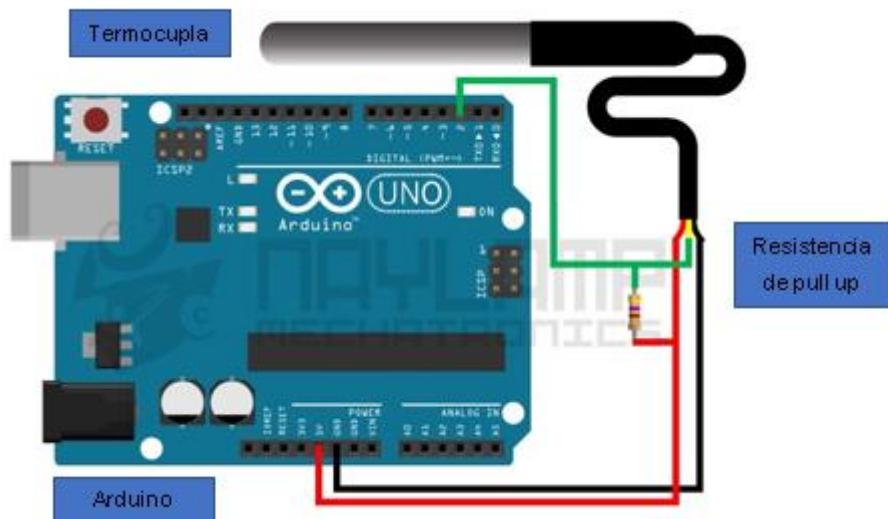


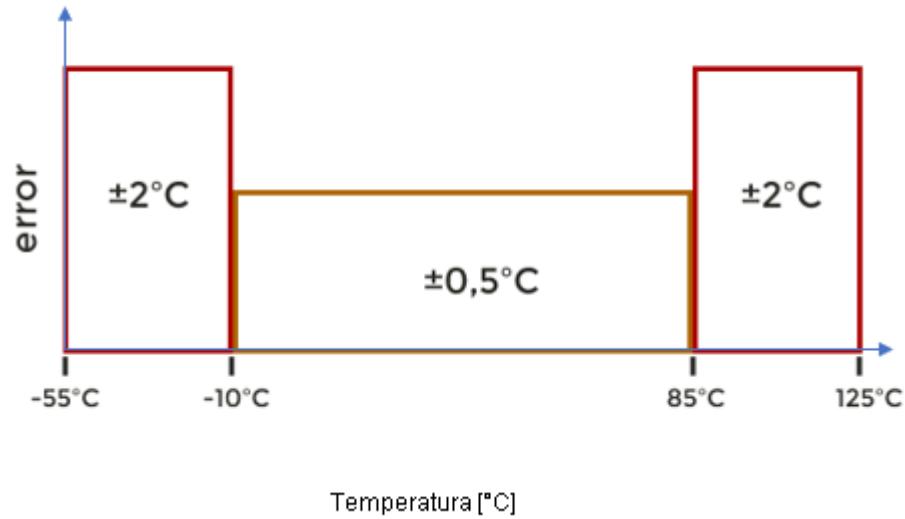
Figura 1.3 Diagrama de conexión sensor de temperatura [9].

Es necesario poner una resistencia pull-up de 4.7K, como se muestra en la figura 1.3, debido a que esta es la que permite la comunicación entre el sensor y el microcontrolador.

#### Especificaciones técnicas:

- Voltaje de alimentación: 3V a 5,5V.
- Rango de temperaturas: -55°C a 125°C.

- Resolución programable: 9-bit, 10-bit, 11-bit o 12-bit (default).



**Figura 1.4 Temperatura [°C] vs error, del sensor de temperatura.**

En la figura 1.4 se muestra la gráfica de error en medición en el sensor de temperatura, en el cual notamos que para rangos entre  $-10^{\circ}\text{C}$  a  $85^{\circ}\text{C}$  el error de medición es de  $\pm 0,5^{\circ}\text{C}$  y para el resto del espectro de medición el error es de  $\pm 2^{\circ}\text{C}$ , es decir, que para diferentes rangos de temperatura al que se someta el sensor, existe un error diferente.

## CAPÍTULO 2

### 2. ANALISIS, DISEÑO E IMPLAEMETACION DE RED LORAWAN

#### 2.1 Criterios de diseño de red de sensores piscinas camaroneras.

Debido a que en la camaronera las piscinas abarcan áreas entre 10 y 15 hectáreas y el único punto eléctrico de la camaronera se encuentra en la oficina de administración, se pueden definir dos criterios importantes para la selección de la tecnología de comunicación: i) Distancia de cobertura, ii) Eficiencia energética. Debido a lo mencionado anteriormente, se debe proponer una solución autosustentable en términos energéticos, por otra parte, se debe de considerar también, que dicha tecnología de comunicación debe ser abierta y no de propietario, y que el espectro de frecuencia de operación no sea licenciado; esto es para disminuir los costos por infraestructura de red; motivo por el cual se descartan Sigfox y LTE (evolución a largo plazo). Por otra parte, también se descartan tecnologías como Zigbee, Wi-Fi o Bluetooth debido a la distancia de cobertura.

Una vez que hemos definido los criterios a cumplir de la solución, la propuesta consiste en implementar una red de bajo consumo energético con tecnología LoRa en la capa física y LoRaWAN como protocolo de comunicación. La arquitectura de red está formada por dos tipos de elementos: los nodos que recolectan la información a través de sensores, cuya alimentación viene dada por un panel solar con su batería y está ubicado en cada una de las piscinas a monitorear; y el Gateway, que trabaja como una central para los nodos y que a su vez envía los datos a la nube a través de un punto de red ubicado en la oficina de la administración de la camaronera. Finalmente, la interfaz de usuario, que interpreta los datos en la nube y los presenta gráficamente.

Los nodos ubicados en las piscinas monitorean dos parámetros de la calidad del agua: el nivel de oxigenación del agua (mg/L), y la temperatura del agua (°C). Son parámetros vitales para aumentar la tasa de supervivencia de los camarones y asegurar un crecimiento óptimo.

## 2.2 Análisis de componentes de la red de sensores.

Para la implementación de la red utilizamos 2 nodos, cada uno compuesto por:

- Un arduino modelo: MKR-1300.
- Un sensor de oxígeno disuelto (DFRobot, gravity, modelo: SEN0237).
- Una termocupla modelo: DS18B20.
- Una antena potencia: 2dB.
- Un panel solar (marca Voltaic Systems), potencia de 6 Watts.
- Una batería (marca Voltaic Systems) 4000mAh.
- Encapsulado Schneider Electric norma IP55.

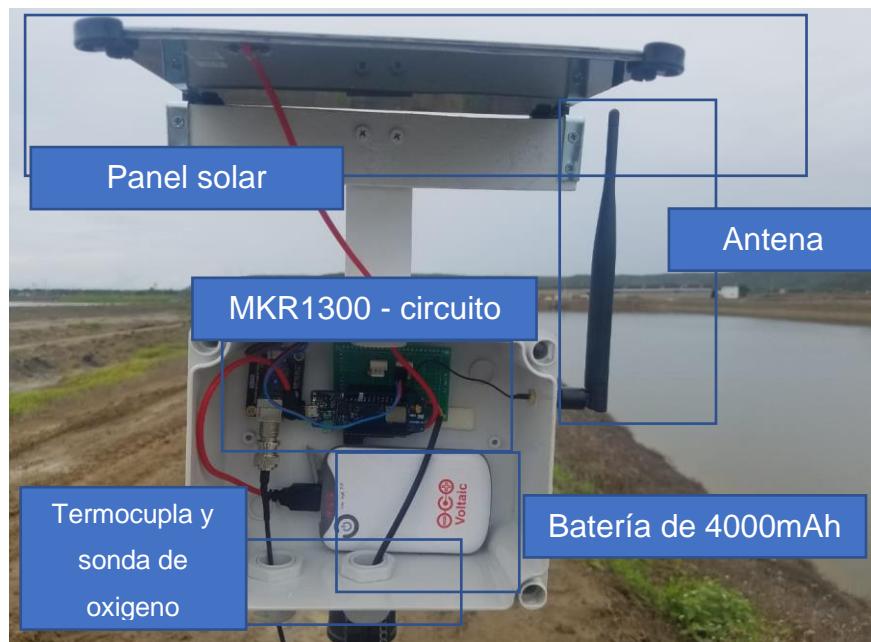


Figura 2.1 Nodo prototipo.

En la figura 2.1 se observa el prototipo y sus componentes previamente mencionados. Los componentes electrónicos fueron colocados dentro de

una caja impermeable y sellada para evitar daños por condiciones ambientales y la salinidad del lugar.



**Figura 2.2 Modulo MKRWAN1300.**

El controlador usado para el nodo prototipo es un Arduinio MKRWAN1300 que se muestra en la figura 2.2. Esta es la solución más práctica y rentable en el mercado, el cual se basa en el Atmel SAMD21 y un módulo Murata CMWX1ZZABZ Lo-Ra.



**Figura 2.3 Gateway “The Things Network”.**

El Gateway utilizado para esta solución es el modelo “The Things Gateway”, que se muestra en la figura 2.3. Se escogió este modelo debido a que nos abstraemos de la infraestructura de servidores y usamos los del fabricante, posteriormente se usa una interfaz de programación para descargar la información a una red local.

Para los sensores, se utilizó el sensor de oxígeno disuelto marca DFRobot modelo SEN0237 y la termocupla modelo DS18B20 debido a sus costos accesibles.



**Figura 2.4 Kit Voltaic Systems panel solar y batería.**

El kit de panel solar y batería marca Voltaic Systems, que se muestra en la figura 2.4, es a prueba de agua y resistentes a condiciones ambientales necesarias para este proyecto.

### **2.3 Análisis de la camaronera y diseño físico de la red de sensores.**

Se realizó el diseño de la red para la camaronera ubicada en la Provincia de Santa Elena en la comuna “El Palmar”.



**Figura 2.5 Distancias nodos a Gateway en piscinas.**

Para este proyecto se colocaron 2 nodos en las piscinas más cercanas a la oficina central donde se instaló el Gateway. Las distancias entre los nodos 1 y 2 al Gateway son de 581 y 668m respectivamente, como se muestra en la figura 2.5.



**Figura 2.6 Pruebas de Nodo-Gateway con línea de vista.**

En la figura 2.6 se muestra cómo se realizaron las pruebas de enlace a los lados de cada piscina para confirmar un enlace exitoso.

Para el diseño de la red calculamos el presupuesto de enlace en la modulación LoRa. Se considera la potencia de transmisión y la sensibilidad del receptor, ecuación (2.1), Estos valores permiten obtener la perdida máxima de potencia en el enlace ( $Q_{max}$ ) que se admite en el receptor para poder transmitir información de los nodos al Gateway correctamente.

$$Q_{max} [\text{dBm}] = \text{EIRP}[\text{dBm}] - \text{RSSI}[\text{dBm}] \quad (2.1)$$

Para este cálculo no se consideran las perdidas en conectores ni cables, pero si la ganancia de las antenas de 2dBi. Según las especificaciones técnicas de los equipos, se tiene una sensibilidad mínima del receptor (RSSI) de -107dBm y una potencia de transmisión de 10dBm. Para obtener la potencia de transmisión efectiva (EIRP) sumamos la potencia de transmisión con la ganancia de la antena para un total de 12dBm. Se usan estos datos en la ecuación (2.1), lo cual indica que entre el emisor y el receptor se admite una pérdida de hasta 119dBm sin que haya problemas en la transmisión.

Dado que el enlace es con línea de vista sin obstrucciones, solo se consideró la perdida en el espacio libre. Para su cálculo se utilizó la ecuación (2.2):

$$\text{FSPL} = 32,45 + 20 \log(f) + 20 \log(d) \quad (2.2)$$

Donde  $f$  es la frecuencia de operación en MHz y  $d$  la distancia de separación entre equipos en kilómetros. Utilizando la ecuación (2.2) calculamos la distancia de cobertura máxima de los nodos al Gateway con frecuencia 915MHz y una perdida teórica máxima de 119dBm, obteniendo como resultado una distancia máxima de enlace de 23.36Km.

Para calcular la potencia recibida teórica en función de la distancia usamos la ecuación (2.3).

$$\text{RSSI[dBm]} = \text{EIRP[dBm]} - \text{FSPL[dBm]} \quad (2.3)$$

Donde FSPL son las perdidas en vacío calculadas en la ecuación (2.2).

De acuerdo con los cálculos teóricos de presupuesto de enlace, se puede realizar un enlace de hasta 23.36Km con línea de vista. En las especificaciones técnicas se indica una distancia de 15Km en áreas rurales con línea de vista, esta diferencia en las distancias puede deberse a que el fabricante consideró las pérdidas de potencia omitidas en nuestro cálculo, o alguna otra atenuación de la señal debido al tipo de modulación o condiciones ambientales.

Al realizar las pruebas en campo el prototipo, se ubica en la piscina más lejana de la oficina de administración, a 8Km de distancia, pero no se obtuvo respuesta de enlace, el enlace se lo obtuvo a una distancia de 7.5Km aproximadamente, con línea de vista. Esta reducción en la distancia efectiva del equipo en comparación a los cálculos teóricos se puede deber a que los equipos se ubicaron a 2 metros de altura con respecto al suelo lo cual puede causar perdidas por obstáculos en la zona de Fresnel, también se consideró que al pasar la señal por múltiples piscinas se podría dar un efecto multirayo, lo cual contribuiría negativamente a la propagación de la señal para establecer un enlace exitoso.

El cálculo de perdidas en el vacío a la distancia de 7.5 kilómetros para determinar la perdida teórica usando la ecuación (2.2), se obtiene una perdida en vacío de 109.18dB. Usando la ecuación (2.3), verificamos que la potencia teórica recibida por el equipo es de -97.18dBm. Con este cálculo y tomando en consideración la sensibilidad mínima del receptor de -107dBm para un enlace exitoso se determinó que existe otra perdida a esa distancia de 9.82dB en la señal de enlace la cual puede haberse dado por las causas ya mencionadas.



**Figura 2.7 Alimentadoras automáticas.**

En las piscinas existen alimentadoras automáticas, como se muestra en la figura 2.7, estas son colocadas en balsas para luego ser ubicadas dentro de las piscinas. Inicialmente se consideró colocar cada nodo en la misma balsa en la cual estaba instalado el sistema de alimentación, sin embargo, con el fin de no interferir con ningún sistema ya implementado en las piscinas, los especialistas de la camaronera indicaron que se instalen los nodos de forma aislada en una balsa de madera en un punto a 3m de distancia de la compuerta de salida del agua.

## **2.4 Implementación de la red de sensores.**

Para la implementación de la red de sensores se crea una cuenta en “The Things Network”, que permite configurar el Gateway para operar en el espectro de frecuencia de 902 – 928 MHz, con una frecuencia central de 915 MHz, y generar la aplicación con un estándar de encriptación avanzado. Dicha aplicación posee un código de aplicación y una llave única, las mismas que deben ser configuradas en cada nodo para que estos puedan realizar la activación por aire (OTAA). La configuración del nodo y el Gateway en la plataforma “The Things Network” es detallada en la sección 2.5.1.

Para la implementación de la interfaz gráfica se usó la plataforma Ubidots. Creamos una cuenta en Ubidots Education que es una plataforma gratuita. La implementación de la interfaz y su integración de datos es detallada en la sección 2.5.2.

Una vez que el paquete llega al Gateway, éste es enviado a los servidores de “The Things Network”, que es desempaquetado y convertido en un objeto java script.

Se implemento una base de datos de respaldo y una interfaz desarrollada en Python que se ejecuta en un servidor local con el propósito de integrar “The Things Network”, “Ubidots” y dicha base de datos, de manera que al recibir la información en los servidores de TTN se cree una copia en la base de datos de respaldo y se envíe a la nube de Ubidots.



**Figura 2.8 Mediciones de oxígeno disuelto y temperatura en campo con equipo manual.**

Se realizaron pruebas de campo en las piscinas 1 y 2 (las más cercanas a la oficina administrativa), donde se realizaron mediciones con un medidor de oxígeno disuelto YSI Pro20, como se muestra en la figura 2.8. Este equipo es utilizado por los paramétristas de la camaronesa.



**Figura 2.9 Nodo durante mediciones en la compuerta de salida del agua.**

Posteriormente se realizaron las mediciones en la compuerta de salida de agua como se muestra en la figura 2.9. se realizaron las mediciones en este lugar ya que los paramétristas realizan sus mediciones en este punto. Debido a que la piscina es muy grande, un cambio en la zona de medición podría alterar los valores medidos. Para evitar errores en la comparación con los registros de la camaronera.

Los nodos fueron configurados usando la librería MKRWAN, OneWire, DallasTemperature y avr/pgmspace de Arduino, para sensar y enviar la información referente a las piscinas, por medio del protocolo LoRa.

El sistema necesita un mantenimiento semestral, en el cual se calibra y da mantenimiento a los sensores en el caso de ser necesario, se limpia los paneles solares que alimentan los nodos y se verifica el estado físico en busca de daños ocasionados por las condiciones ambientales del sitio.

## 2.5 Implementación de la nube.

Para la implementación de la nube se utilizaron las plataformas “The Things Network” y Ubidots.

### 2.5.1 Configuración de la nube de TTN

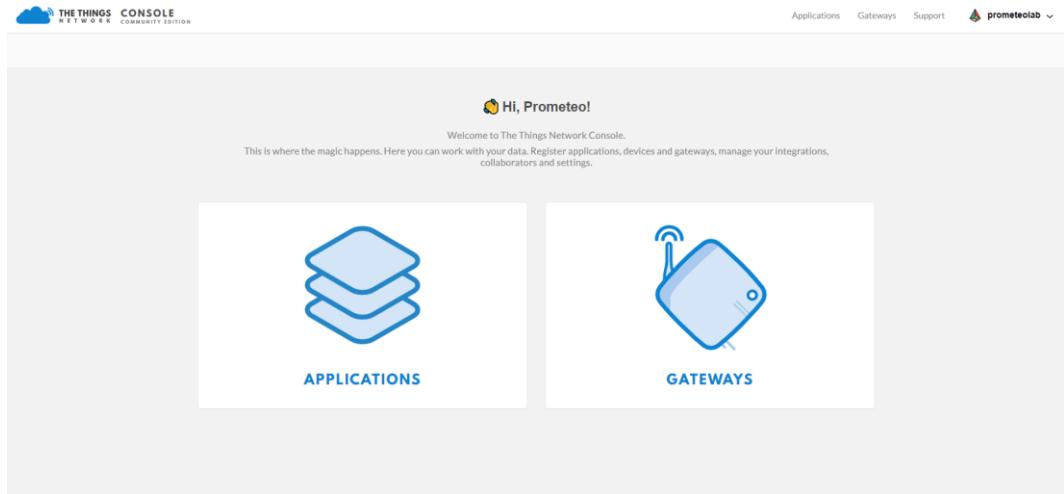


Figura 2.10 Consola de TTN.

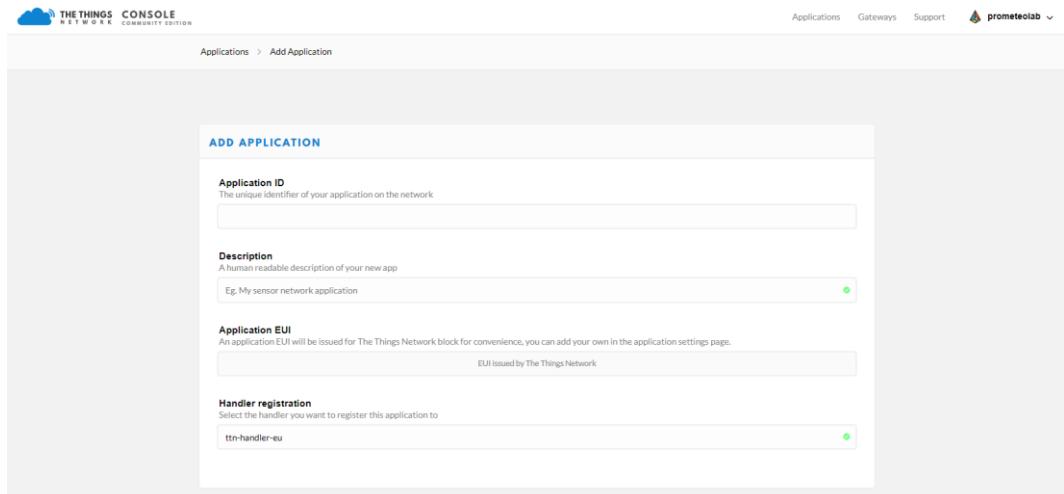
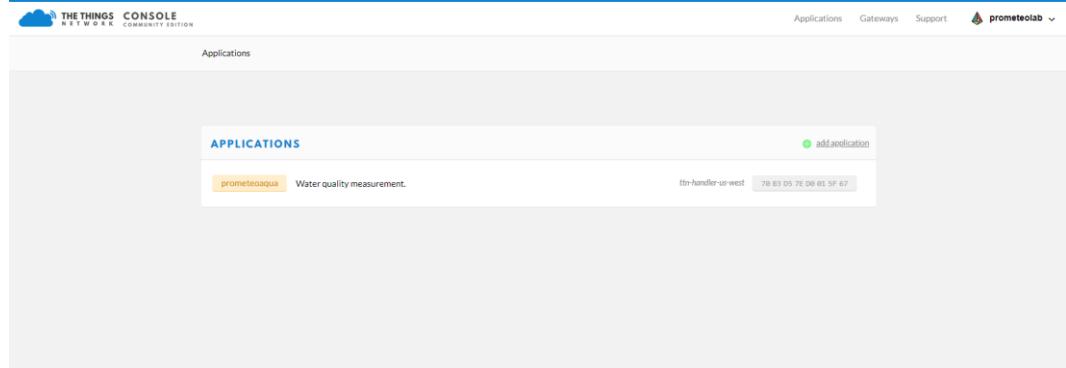


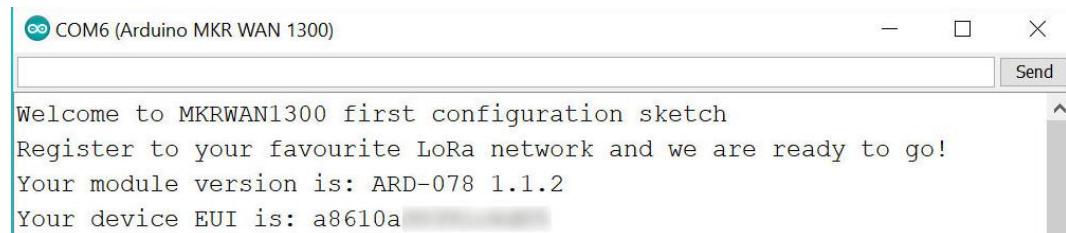
Figura 2.11 Plantilla de nueva aplicación.

Como se muestra en la figura 2.10, se registra el Gateway en la plataforma de “The Things Network” usando la “Gateway Key” que es una llave especificada por el fabricante para su sincronización con la consola virtual. Una vez a la consola procedemos a añadir una aplicación en la pestaña de

“Applications”. Así mismo, se debe crear una aplicación llenando un formulario.



**Figura 2.12 Tabla de aplicaciones.**



**Figura 2.13 Device EUI de un nodo MKR1300 vía serial.**

Después de crear una aplicación, se registran los dispositivos finales que van a hacer uso de esta, es decir los Arduinos MKR1300, para esto, en la aplicación se indica el “Device EUI” de cada una de las placas, es decir, su dirección física con el propósito de que puedan ser reconocidas. Para leer ese número, se necesita subir el código llamado “First Configuration” a cada uno de los Arduinos y posteriormente se abre la ventana de comunicación serial del IDE de Arduino donde podremos ver impreso el “Device EUI” como se muestra en la Figura 2.13; el código utilizado forma parte de la librería MKRWAN de Arduino.

The screenshot shows the 'REGISTER DEVICE' page in the The Things Network Console. At the top, there are navigation links for Applications, Gateways, Support, and a dropdown for 'prometeolab'. Below the header, a breadcrumb trail shows 'Applications > prometeoqua > Devices'. The main area is titled 'REGISTER DEVICE' with a 'bulk import devices' button. It contains four input fields: 'Device ID' (a text input), 'Device EUI' (a text input with a character limit of 8 bytes), 'App Key' (a text input with a note that it will be generated), and 'App EUI' (a text input containing the value '70 B3 D5 7E D0 01 F6 67').

**Figura 2.14 Plantilla de registro de equipo.**

The screenshot shows the 'DEVICES' list in the The Things Network Console. The header includes 'register device' and a note '1 - 2 / 2'. Below the header, there are two entries: 'mkr\_1300\_node\_1' and 'mkr\_1300\_node\_2'. Each entry shows a MAC address: 'AB 61 0A 3B 33 64 0B' for node 1 and 'AB 61 0A 3B 37 64 0B' for node 2. There are also small green circular icons next to each entry.

**Figura 2.15 Lista de dispositivos finales (Arduinos) registrados en TTN.**

Cada dispositivo se agrega llenando la plantilla de datos que se puede observar en la Figura 2.14, uno de los campos es el “Device EUI”.

Una vez que se han configurado el Gateway y los dispositivos finales en la nube de TTN, se pueden enviar y almacenar los paquetes de datos.

The screenshot shows the Arduino Serial Monitor window titled "COM6 (Arduino MKR WAN 1300)". The window displays the following text:  
Welcome to MKRWAN1300 first configuration sketch  
Register to your favourite LoRa network and we are ready to go!  
Your module version is: ARD-078 1.1.2  
Your device EUI is: a8610a  
Are you connecting via OTAA (1) or ABP (2)?  
At the bottom, there are three buttons: "Autoscroll" (checked), "Carriage return", "115200 baud" (selected), and "Clear output".

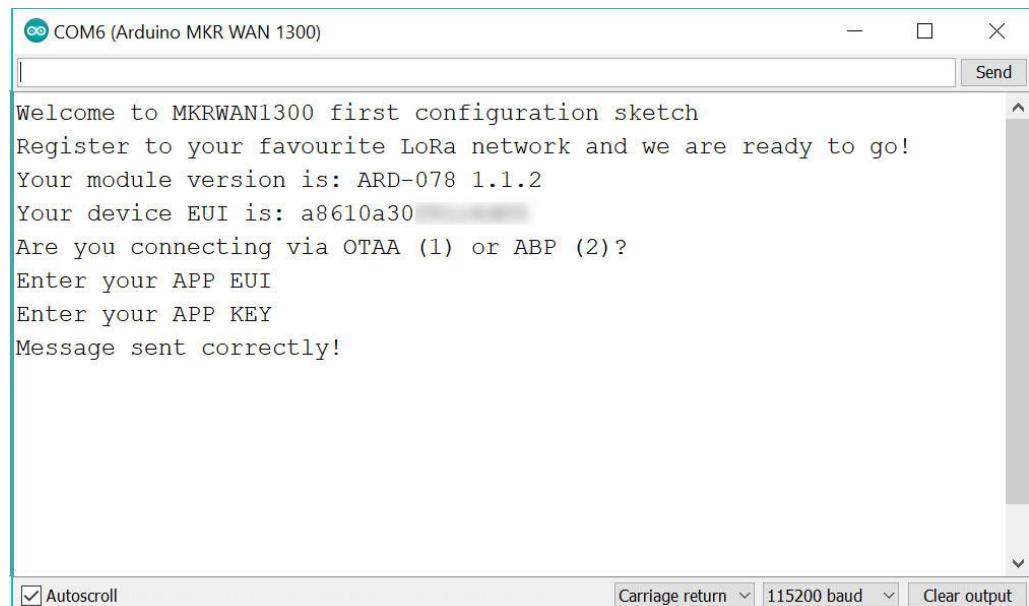
**Figura 2.16 Selección de tipo de conexión.**

En la Figura 3.16 se puede ver que a través del monitor serial del IDE de Arduino se puede comunicar con el MKR1300 con el propósito de enviar datos de prueba, para ello se envía un mensaje en texto plano indicando el tipo de conexión LoRa, es decir se envía “1” para una comunicación OTAA.

The screenshot shows the Arduino Serial Monitor window titled "COM6 (Arduino MKR WAN 1300)". The window displays the following text:  
Welcome to MKRWAN1300 first configuration sketch  
Register to your favourite LoRa network and we are ready to go!  
Your module version is: ARD-078 1.1.2  
Your device EUI is: a8610a  
Are you connecting via OTAA (1) or ABP (2)?  
Enter your APP EUI  
Enter your APP KEY  
At the bottom, there are three buttons: "Autoscroll" (checked), "Carriage return", "115200 baud" (selected), and "Clear output".

**Figura 2.17 Ingreso de dirección y llave del controlador.**

Posteriormente se envían a la placa por serial el “APP EUI” y el “APP KEY” como se observa en la Figura anterior; esta información se encuentra en la pestaña de “Applications” seleccionando la aplicación en uso.



The screenshot shows a terminal window titled "COM6 (Arduino MKR WAN 1300)". The window displays the following text:

```
Welcome to MKRWAN1300 first configuration sketch
Register to your favourite LoRa network and we are ready to go!
Your module version is: ARD-078 1.1.2
Your device EUI is: a8610a30
Are you connecting via OTAA (1) or ABP (2)?
Enter your APP EUI
Enter your APP KEY
Message sent correctly!
```

At the bottom of the window, there are several buttons: "Autoscroll" (checked), "Carriage return", "115200 baud" (selected), and "Clear output".

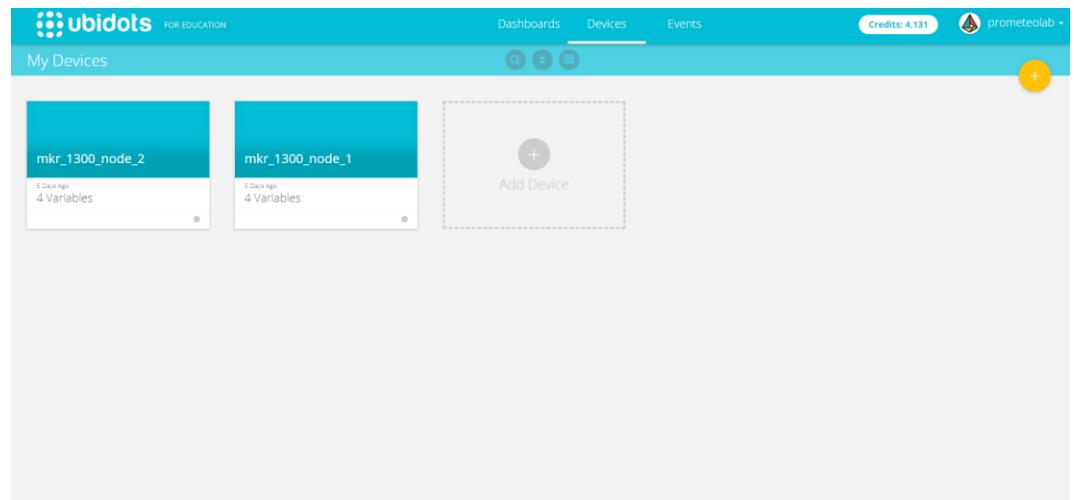
**Figura 2.18 Mensaje de conexión exitosa.**

### 2.5.2 Configuración de la nube de Ubidots for Education



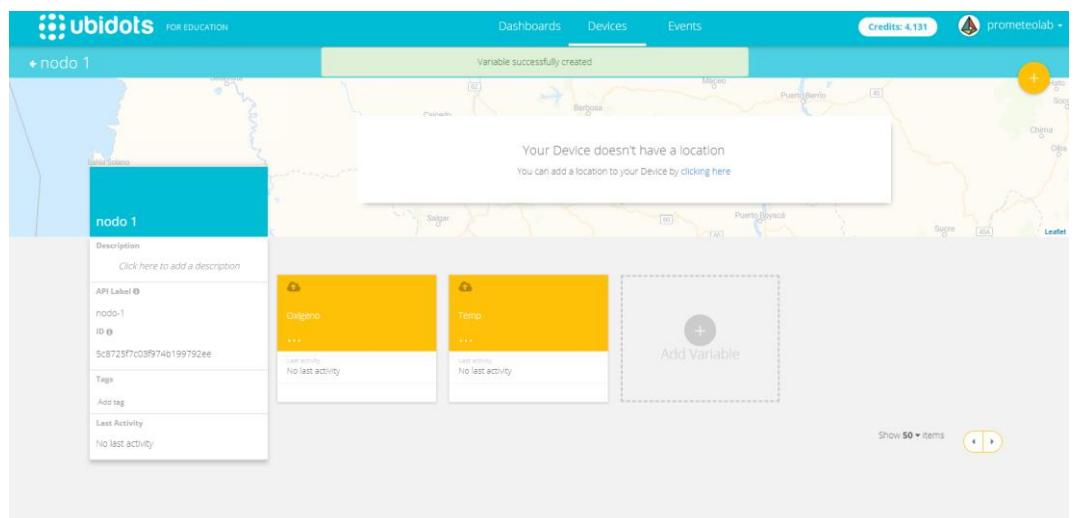
**Figura 2.19 Página principal Ubidots Education.**

Una vez creada una cuenta en Ubidots for Education, se proceden a agregar los dispositivos que van a ser monitoreados, en este caso, los dos nodos MKRWAN.



**Figura 2.20 Ventana “Devices” en Ubidots.**

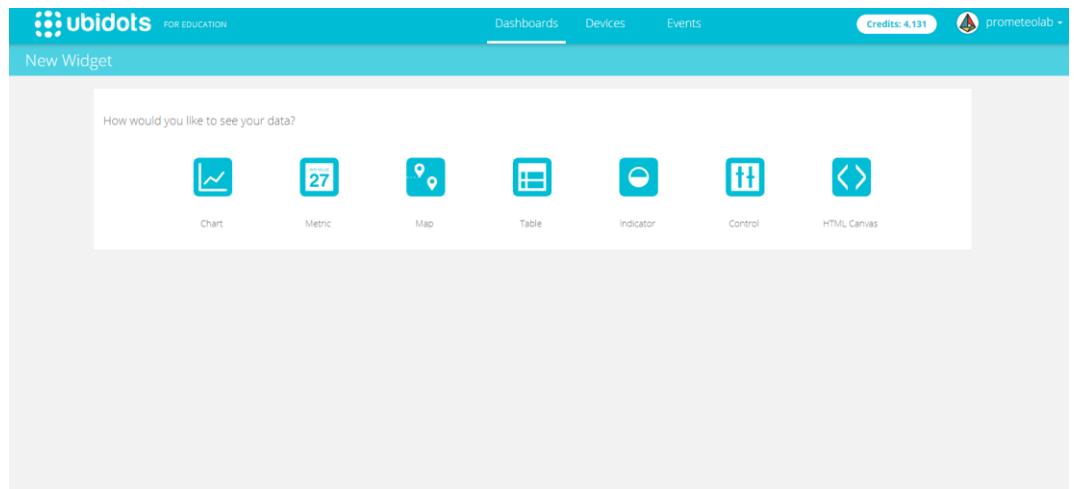
Luego se añaden los dispositivos en la pestaña “Devices” como se puede ver en la Figura 2.20.



**Figura 2.21 Variables del “device” en Ubidots.**

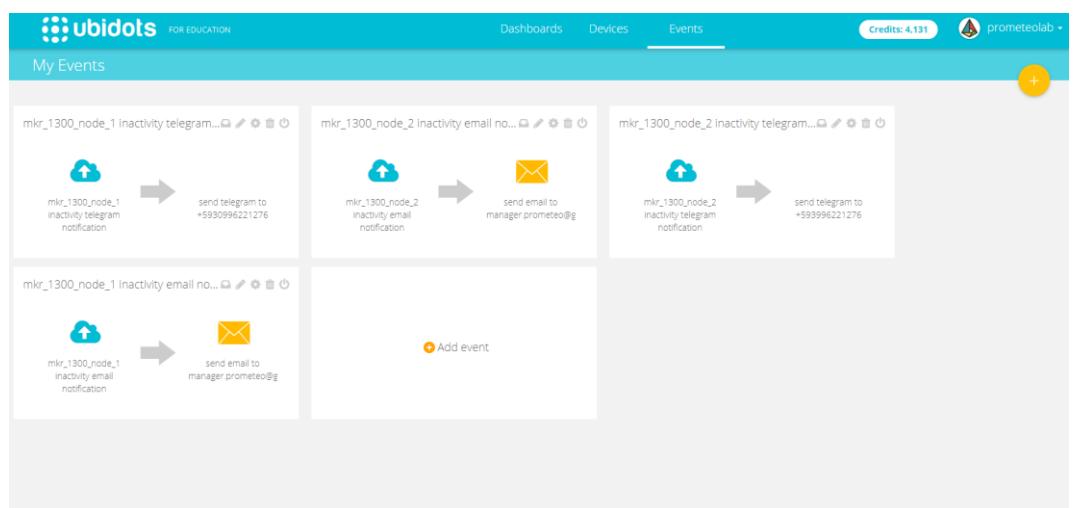
A cada “Device” o nodo se deben agregar “Variables”, es decir, contenedores donde se almacena el dato que se va a enviar desde el MKR1300, es importante considerar que estas variables poseen un

identificador único que debe ser configurado en el arduino con el propósito de que este sepa a donde enviar la información; En la figura anterior se muestra un “Device” con sus “Variables”



**Figura 2.22 Opciones de “widgets” en Ubidots.**

Para convertir los datos en información útil para el monitoreo, en la pestaña “Dashboard” se crean los indicadores y los gráficos con respecto al tiempo, estos elementos se llaman “Widgets” y pueden ser observados en la Figura 2.22.



**Figura 2.23 Ventana de “Events” en Ubidots.**

Finalmente, como medida de seguridad en la ventana “Events”, se generaron eventos condicionados, los cuales envían un correo electrónico en caso de inactividad por 30 minutos, esto nos indicaría que hay un fallo en el nodo.

## 2.6 Procesamiento de datos.

Los datos de los sensores de oxígeno disuelto y temperatura son adquiridos por los nodos y enviados al Gateway a través del protocolo LoRaWAN, usando la frecuencia en banda libre de 915MHz. El Gateway recibe los datos y los almacena en los servidores “The Things Network”. Una vez que la información llega a los servidores se usa un programa desarrollado en Python, que a su vez utiliza una interfaz de programación de la consola con el propósito de recibir dicha información de manera local para desempaquetar la carga útil. Esta interfaz de programación usa internamente un servicio llamado “Message Queue Telemetry Transport”, posteriormente se construye un objeto java script con dicha carga útil con el formato de Ubidots para enviar la información a la interfaz gráfica.

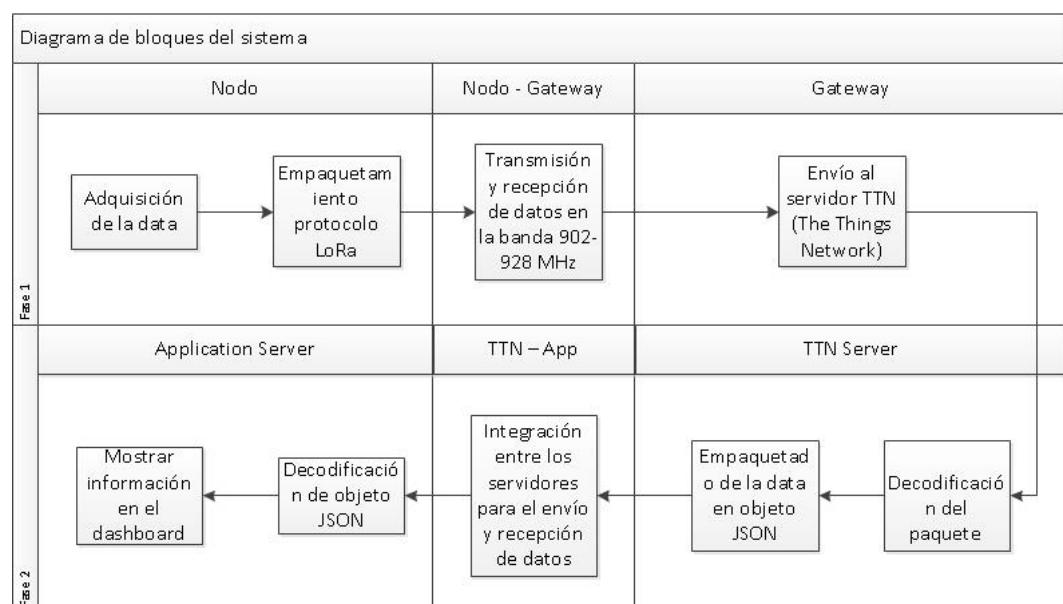
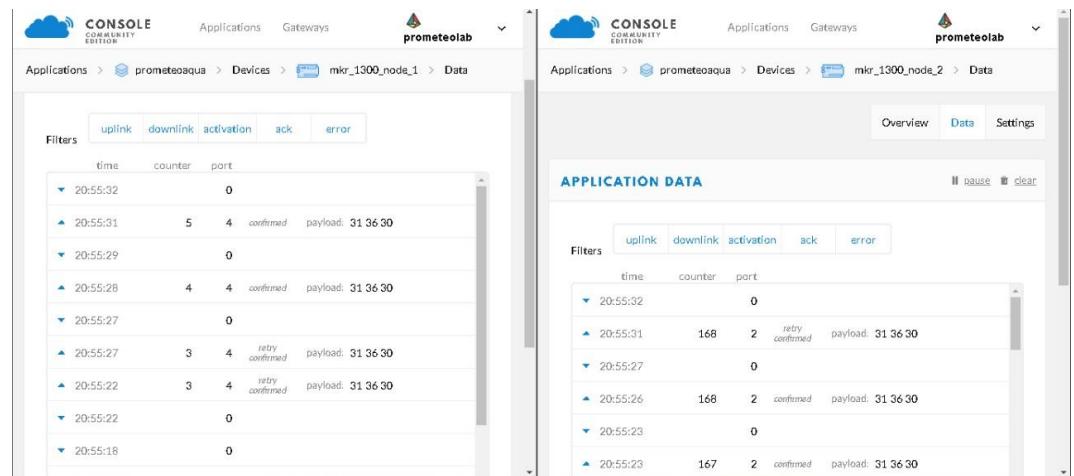


Figura 2.24 Diagrama de bloques general de la red.

La figura 2.24 muestra un diagrama de bloques de todo el procesamiento de datos. En el nodo se adquieren los datos de los sensores y se empaquetan para ser enviados mediante protocolo LoRa, estos datos empaquetados son enviados en la banda de 902-928Mhz hacia el Gateway. El Gateway sube estos datos al servidor de “The Things Network” mediante internet donde son desempaquetados y decodificados. Finalmente, se realiza una integración del servidor con la interfaz gráfica en la cual se envían los datos del servidor de una manera compatible con la aplicación para ser mostrados en la interfaz gráfica.



**Figura 2.25 Paquetes recibidos por los servidores de TTN enviados por los nodos.**

```

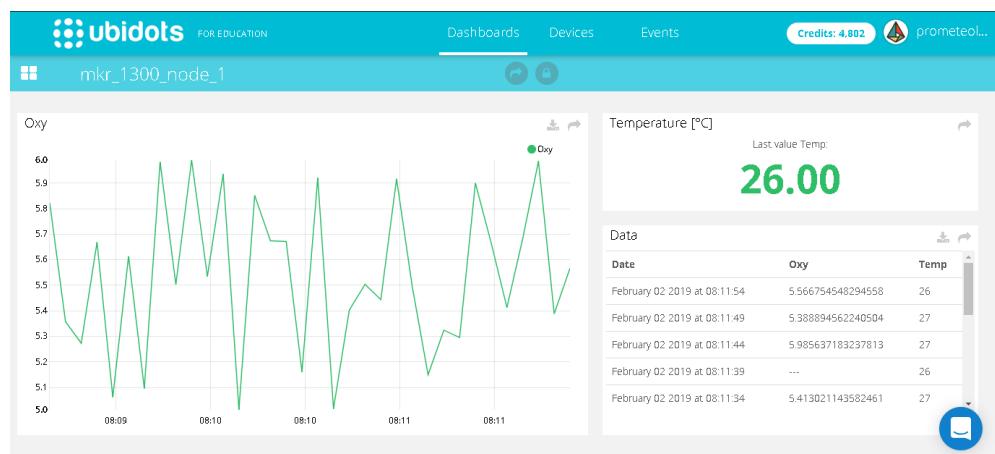
Run: main x
('oxyen node 1': 5.766701652616423, 'temp node 1': 27}
{'temp node 2': 28, 'oxyen node 2': 5.238559231694646}
{'oxyen node 1': 5.366732653138001, 'temp node 1': 26}
{'temp node 2': 28, 'oxyen node 2': 5.633497544196436}
{'oxyen node 1': 5.593927491668412, 'temp node 1': 26}
{'temp node 2': 28, 'oxyen node 2': 5.0641806072092255}
{'oxyen node 1': 5.626217077574118, 'temp node 1': 27}
{'temp node 2': 27, 'oxyen node 2': 5.544898268479596}
{'oxyen node 1': 5.33401508196263, 'temp node 1': 26}
{'temp node 2': 27, 'oxyen node 2': 5.706320215987542}
{'oxyen node 1': 5.150110293612014, 'temp node 1': 27}
{'temp node 2': 27, 'oxyen node 2': 5.9232092729415236}
{'oxyen node 1': 5.644431969672895, 'temp node 1': 26}
{'temp node 2': 28, 'oxyen node 2': 5.981982013825013}
{'oxyen node 1': 5.061039803636466, 'temp node 1': 27}
{'temp node 2': 28, 'oxyen node 2': 5.14386901125254}
{'oxyen node 1': 5.201471400584126, 'temp node 1': 27}
{'temp node 2': 28, 'oxyen node 2': 5.392014632837411}

```

Run 4: Run TODO Terminal Python Console

**Figura 2.26 Captura de datos usando Python.**

La figura 2.25 muestra los paquetes encriptados recibidos en el servidor TTN. Estos datos son descargados localmente y desencriptados usando un IDE de Python, como se muestra en la figura 2.26, con el propósito de almacenar la información en una base de datos local.



**Figura 2.27 Información enviada por los nodos en la interfaz gráfica.**

La interfaz gráfica fue creada en la plataforma Ubidots donde se reciben las 2 variables (temperatura y oxígeno). La variable de temperatura muestra el valor actual de temperatura en tiempo real, y la variable oxigeno muestra

una gráfica del nivel de oxígeno en [mg/L] vs tiempo como se muestra en la figura 2.27.

# CAPÍTULO 3

## 3. ANÁLISIS DE RESULTADOS

Se implementaron dos nodos prototipos como los mostrados en la figura 2.1 con los cuales logramos obtener datos de calidad de agua en tiempo real que son importantes para el crecimiento del camarón y su supervivencia. Los datos de oxígeno fueron validados con un sensor de oxígeno disuelto “YSI Pro20”, el mismo que es utilizado para hacer las mediciones en el sitio.

### 3.1 Pruebas en campo.

#### 3.1.1 Primera visita

En nuestra primera visita a la camaronera se realizó una inspección del sitio, se instalaron 2 nodos en las compuertas de entrada de las piscinas más cercanas al Gateway, y se recopilo 1 hora de datos.

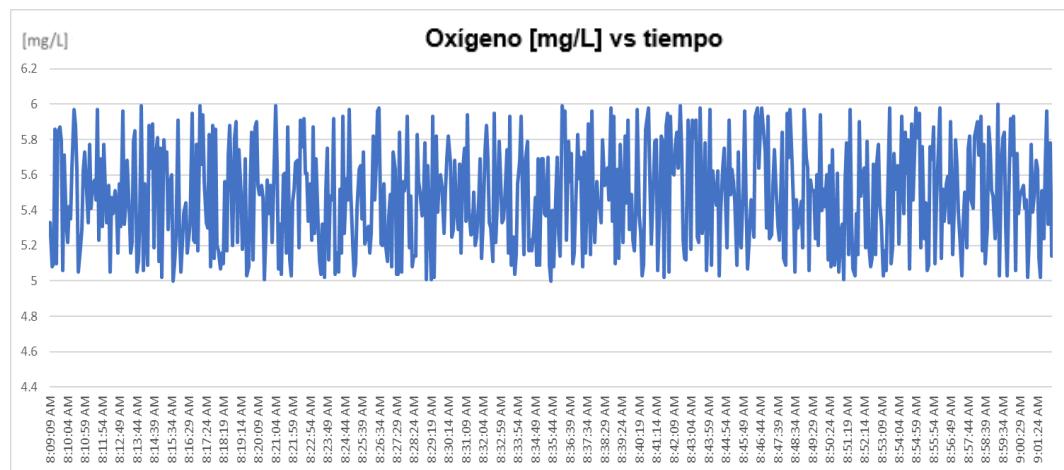
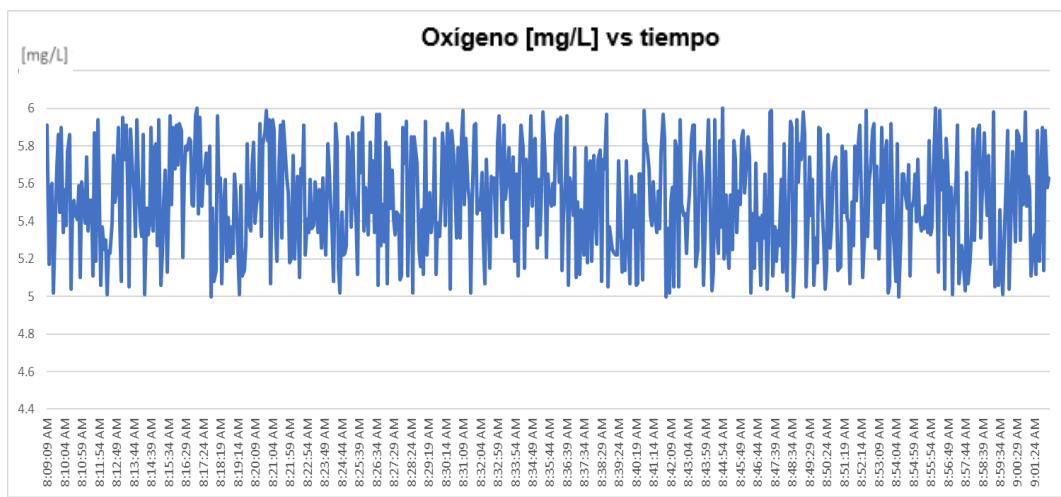
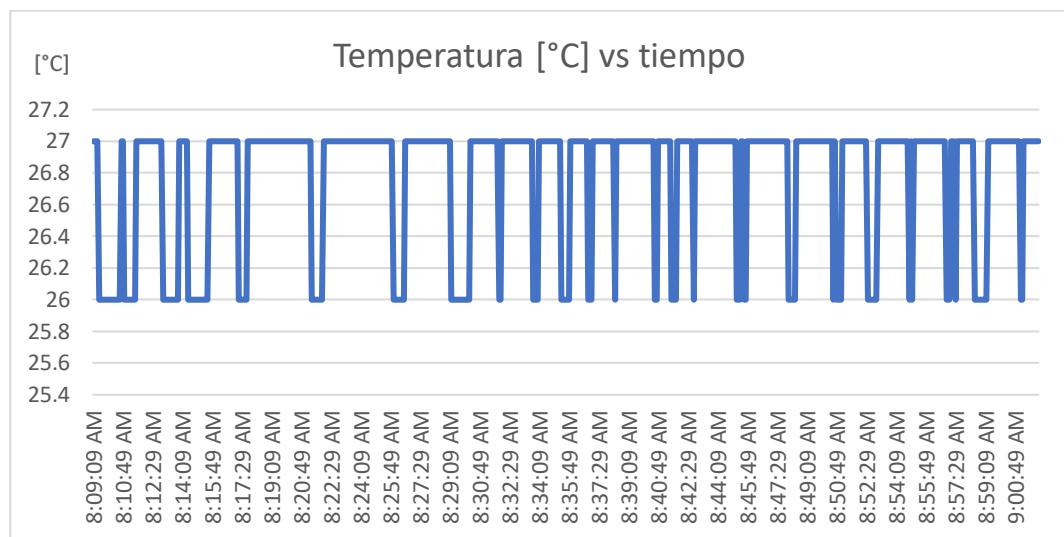


Figura 3.1 Oxígeno [mg/L] vs tiempo de la medición, nodo 1.

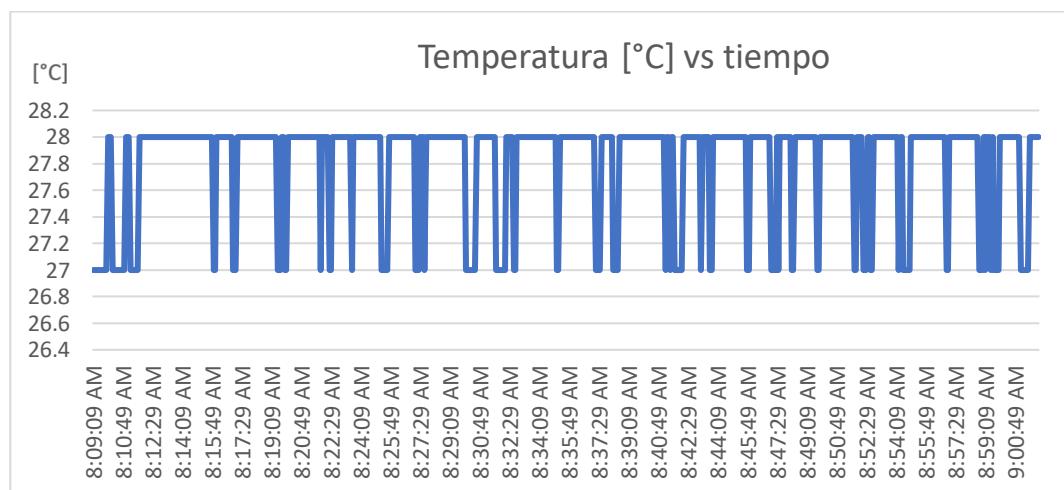


**Figura 3.2 Oxígeno [mg/L] vs tiempo de la medición, nodo 2.**

La figura 3.1 y 3.2 se muestran los datos tomados por los sensores de oxígeno durante una hora de operación en las piscinas, sacando valores promedio tenemos 5.53 y 5.46 [mg/L] respectivamente, lo cual coincide con los valores tomados con el equipo de medición de la camaronera. En el sitio se realiza estas mediciones 3 veces al día, verificando que los niveles de oxígeno de las piscinas no sean menores a 3 [mg/L], ya que a este nivel el camarón deja de comer, no crece o muere. Esto podría representar una gran pérdida para la empresa. En el caso de tener una medición cercana al valor crítico se realiza un cambio de agua encendiendo todas las bombas de agua de las piscinas.



**Figura 3.3 Temperatura [°C] vs tiempo de la medición, nodo 1.**



**Figura 3.4 Temperatura [°C] vs tiempo de la medición, nodo 2.**

En la figura 3.3 y 3.4 se observan los datos de temperatura recolectados por cada nodo durante una hora, sacando valores promedio tenemos 26.77 y 27.80°C respectivamente, el valor tomado por el equipo de medición del sitio fue de 27.57°C. Al observar los datos se puede notar un desfase de 1°C en el nodo 1, esto podría deberse a una falla en la calibración del sensor de temperatura en el puerto analógico del nodo. Las mediciones de temperatura en el sitio se realizan en la noche donde hace más frio, lo que es malo para el camarón. Al realizar las mediciones se busca mantener una

temperatura mayor a 25C° y menor a 32C°. El sitio cuenta con calefactores los cuales calientan agua y es bombeada a la piscina de ser necesario.

### 3.1.2 Segunda visita

En la segunda visita a la camaronería se dejó el equipo tomando datos por más tiempo durante la jornada de la madrugada. La jornada de la madrugada comienza desde las seis de la tarde hasta las nueve de la mañana. También se realizaron mediciones manuales cada hora de la jornada para poder compararlas con las mediciones del prototipo.

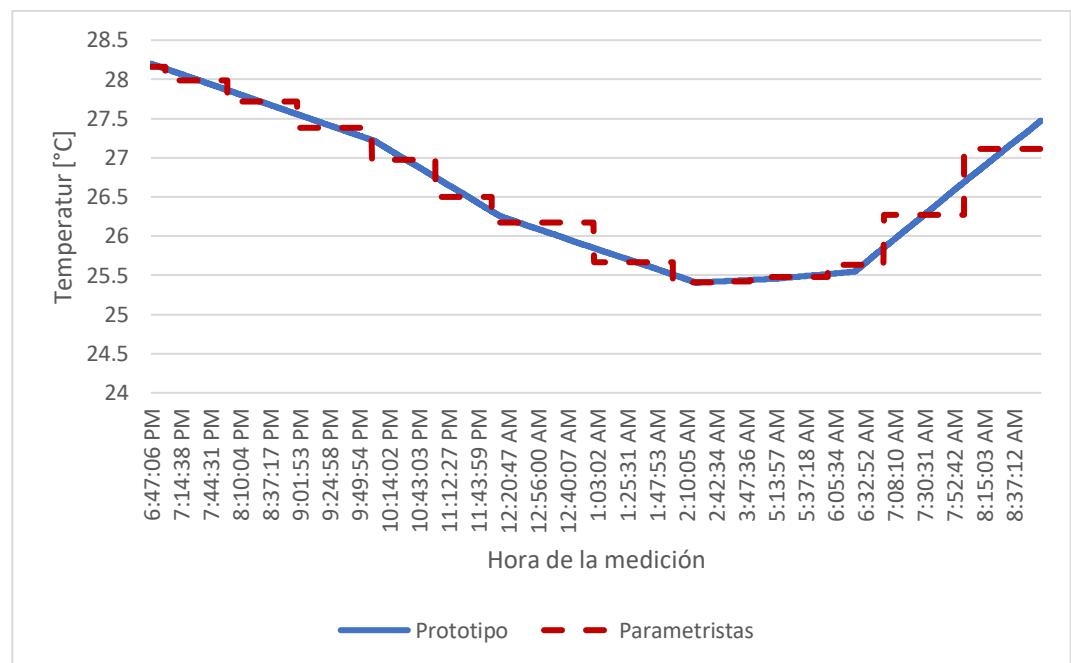
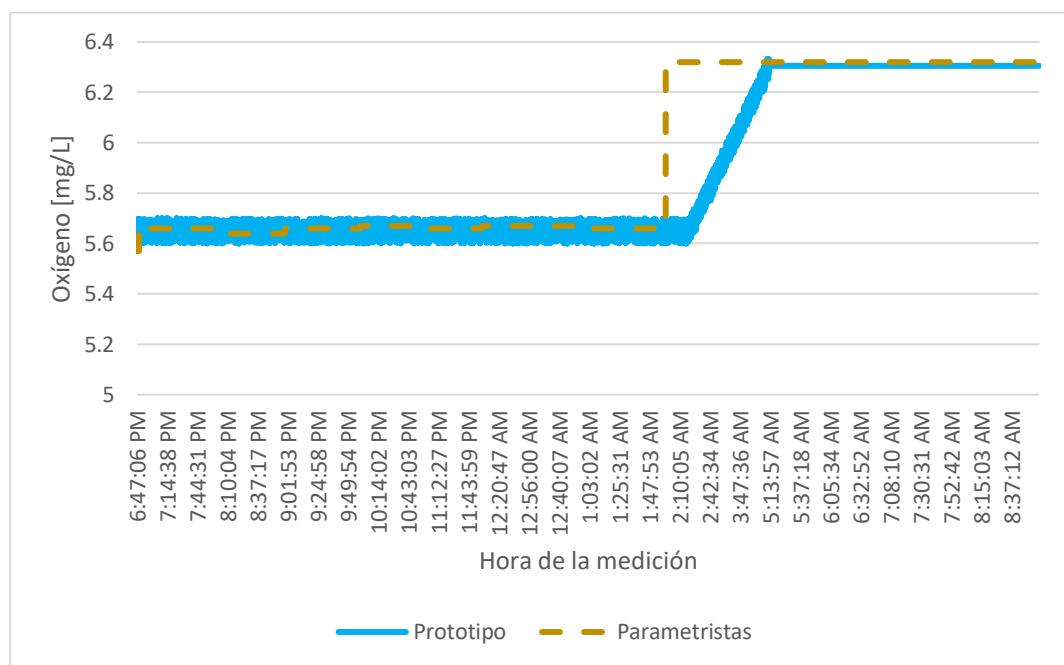


Figura 3.5 Temperatura [°C] vs tiempo de la medición, prototipo y paramétristas.



**Figura 3.6 Oxígeno [mg/L] vs tiempo de la medición, prototipo y parametristas.**

En las figuras 3.5 y 3.6 se detallan las mediciones tomadas de oxígeno y temperatura, respectivamente. se observa las mediciones tomadas por el prototipo representadas con una línea continua y las tomadas por los parametristas cada hora de la jornada representada con una línea punteada.

Se puede notar un aumento en el nivel de oxígeno a partir de las 2am, esto se debe a que a esa hora empezó a llover causando oxigenación del agua de la piscina. También se nota que la temperatura en ese intervalo de tiempo fue el más bajo registrado durante la jornada, casi llegando a niveles de alerta en lo que debería encender los calefactores de agua para regular la temperatura.

Al comparar estos datos tomados se puede apreciar lo cerca que están las gráficas una de la otra lo cual confirma la fiabilidad de las mediciones de nuestro prototipo en comparación con la medición manual realizada por los parametristas.

**Tabla 3.1. Tabla estadística de resultados temperatura y oxígeno (prototipo).**

Mediciones con el prototipo	
temperatura [°C]	oxígeno [mg/L]
Media	26.50819856
Error Estándar	0.012621281
Mediana	26.3535
Moda	25.453
Desviación Estándar	0.87897049
Varianza de la muestra	0.772589121
Curtosis	-1.2666838
Oblicuidad	0.321446387
Rango	2.792
Mínimo	25.407
Máximo	28.199
Suma	128564.763
Conteo	4850
Nivel de confianza (95,0%)	0.024743432
Nivel de confianza (95,0%)	
0.008497	

**Tabla 3.2 Tabla estadística de resultados temperatura y oxígeno (mediciones manuales)**

Mediciones de la camarónera	
temperatura [°C]	oxígeno [mg/L]
Media	26.51604742
Error Estándar	0.012419478
Mediana	26.27
Moda	26.17
Desviación Estándar	0.864916587
Varianza de la muestra	0.748080703
Curtosis	-1.236053394
Oblicuidad	0.297377927
Rango	2.75
Mínimo	25.41
Máximo	28.16
Suma	128602.83
Conteo	4850
Nivel de confianza (95,0%)	0.024347808
Nivel de confianza (95,0%)	
6.82642E-15	

La tabla 3.1 muestra la temperatura del agua de la piscina, este parámetro es importante monitorearlo puesto que el agua tiene que estar en un umbral óptimo para garantizar el crecimiento sano del camarón, actualmente tanto el encendido de los calefactores de agua como la medición de la temperatura es indirecta, es decir, durante las noches, el sistema de calefacción detecta si la temperatura de un tanque de agua ubicado cerca de la administración cae demasiado baja (menor a 25°C) y se activan las bombas de agua caliente hasta que se regule la temperatura (mayor a 27°C), sin embargo, esta medición se realiza en un solo tanque y no en cada piscina. Al mismo tiempo se oxigena el agua ya que ingresa un flujo de agua nuevo desde los calefactores.

En las mediciones no se tuvieron casos en los cuales los niveles de oxígeno lleguen a estados críticos (menor a 3mg/L), lo cual es bueno, ya que en este punto se tendrían perdidas de producto debido a la muerte de los camarones. Se preguntó sobre casos en los cuales se llegaron a estos niveles y se nos indicó que antes de tener las alimentadoras automáticas en la camaronera se lanzaba el balanceado a las piscinas con palas, pero este método causaba que se acumule el balanceado en lugares específicos de las piscinas causando que no pueda ser comido en su totalidad por los camarones, este balanceado se asienta en el suelo de la piscina y se descompone; esto ocasiona un aumento de materia orgánica en la piscina y esta consumía el oxígeno disuelto en el agua. En el peor de los casos el camarón moría y se perdía la totalidad del producto en dicha piscina, pero si se lograba notarse a tiempo se realiza una limpieza de la piscina aumentando el flujo de agua de esta y filtrando la salida para retirar la materia orgánica responsable. En este proceso se pierde aproximadamente una cantidad del producto del 40% de la piscina.

Al comparar la tabla 3.1 y 3.2 se observa que los datos de temperatura son muy similares, pero en los datos de oxígeno hay una ligera diferencia, esto podría deberse debido a que el lugar donde se realiza la medición manual y la medición del prototipo no es exactamente el mismo sitio y por qué el

estado de la membrana de los sensores tampoco es el mismo, puesto que estos sensores funcionan con una membrana que se desgastan con el tiempo, lo cual pudo haber afectado en las mediciones realizadas por los mismos. A pesar de que ambos sensores de oxígeno se encuentran en diferentes estados, la diferencia entre las mediciones es mínima. Con estos resultados podemos concluir que las mediciones del prototipo son fiables y necesarias para la optimización de los procesos de la camaronera.

### 3.2 Análisis de costos.

Se han considerado 2 costos principales para el diseño de esta solución:

- CapEx: gastos de capital utilizados por una empresa para adquirir, actualizar o mantener activos físicos como propiedades, tecnología o equipamiento que se usa para mejorar el rendimiento a futuro.
- OpEx: costos en los que incurre una empresa para llevar a cabo sus operaciones día a día.

**Tabla 3.3. Tabla de costos de adquisición de un nodo.**

Cantidad	Detalle	Costo Unitario	Costo Total
1	Arduino MKR1300	\$40	\$40
1	Antenas 915MHz, 2dB	\$10	\$10
1	Panel Solar/ batería Voltaic Systems	\$90	\$90
1	Sensor oxígeno disuelto DFRobot	\$169	\$169
1	Termocupla DS18B20	\$10	\$10
1	Caja dexon Scheneider Electric IP65	\$20	\$20
3	Conectores a prueba de agua	\$0,25	\$0,75
1	Soporte panel solar	\$10	\$10
1	Amarra metálica	\$0,50	\$0,50
15	Tornillos M5	\$0,01	\$0,15
1	Cinta termo retráctil	\$3	\$3
1	Electrónica	\$15	\$15
Total			\$368,40

De la tabla 3.3 se observa que el costo total de un nodo es \$368,40, en caso de querer replicar el dispositivo, se debe multiplicar dicho costo por n cantidad de nodos.

**Tabla 3.4. Tabla de adquisición del Gateway.**

Cantidad	Detalle	Costo Unitario	Costo Total
1	TTN Gateway	\$350	\$350
Total			\$350

En la tabla 3.4 se observa que se utiliza un Gateway, esto es debido a que ubicándolo en el edificio de la administración tenemos un área de cobertura que cubre toda la camaronera; la piscina más lejana se encuentra a 8Km de distancia del punto mencionado y la máxima distancia de cobertura del Gateway es de 15Km. En caso de no tener cobertura en toda la camaronera se debe incurrir en la adquisición de otro Gateway y el costo de este se multiplicaría por la “n” cantidad de Gateways. No se ha considerado costo de implementación de una torre de transmisión debido a que ya existe una en el sitio.

Los costos de operación y mantenimiento dependen de los costos por servicios en la nube y se detallan a continuación:

**Tabla 3.5. Tabla de costos de operación y mantenimiento.**

Cantidad	Detalle	Costo Unitario	Costo Total
1	Licencia de ubidots	\$48	\$48
Total, mensual			\$48

En la tabla 3.5 se detalla el costo del mantenimiento mensual del servicio de alojamiento en la nube de ubidots para 25 nodos (plan mínimo). Para este estudio no se han considerado costos de mano de obra, por lo que no se incluye el costo de movilizar un técnico a realizar el mantenimiento del

equipo, el cual consiste en una limpieza minuciosa de conectores, calibración de los sensores y debe realizarse una vez cada 6 meses, pero por el hecho de ser una solución en etapa de desarrollo se recomienda se realice una vez al mes por un periodo de mínimo 3 meses.

**Tabla 3.6. Tabla de costos.**

Tipo de costo	Detalle	Cantidad	Valor Unitario	Subtotal	Total
Implementación	Gateway	1	\$350	\$350	\$1086,80
	Nodo	2	\$368,40	\$736,80	
mantenimiento	Soporte y mantenimiento	1	\$48	\$48	\$48
Total					\$1134,80

La tabla 3.6 detalla el costo inicial total de este estudio, en caso de querer escalar el proyecto hay que tener en consideración el tamaño de la camaronera debido a que eso determina la cantidad de gateways y la cantidad de piscinas que se desean medir debido a que de esto depende la cantidad de nodos y el precio de la licencia de ubidots.

Estos costos resultan razonables debido a que reduce enormemente el riesgo de enfermedades y muerte temprana de los camarones, los cuales resultarían en grandes pérdidas de dinero; teniendo en consideración que, en una piscina de 30 hectáreas, donde se producen 333 libras por hectárea mensual de camarón tipo 16-20 con precio referencial de \$5.15 por libra de acuerdo al ministerio de agricultura, ganadería, acuacultura y pesca [11], en el caso de contaminación de una piscina podría resultar en la pérdida de aproximadamente \$51448,50 más el costo de alimentos, vitaminas y larvas usados.

## CAPITULO 4

### 4. CONCLUSIONES Y RECOMENDACIONES

#### 4.1 Conclusiones

Con este sistema los dueños de la camaronera pueden monitorear constantemente sin necesidad de estar presentes físicamente en el sitio mediante una conexión a internet, lo cual es de gran ayuda ya que estos sitios se pueden encontrar muy alejados de las ciudades. Adicionalmente este proyecto puede ser aplicado a los laboratorios de larvas de camarón donde se estudia el mejoramiento de la calidad de las larvas para que estas puedan crecer y reproducirse más, y evitar la mortalidad temprana del camarón.

La tecnología LoRa puede ser adaptada a un sinnúmero de aplicaciones en el ámbito de desarrollo para facilitar procesos y mejorar la productividad. Los criterios utilizados para la selección fueron eficiencia energética, larga distancia de cobertura, que opere en bandas no licenciadas y que la tecnología sea libre, es decir, que no sea de una empresa propietaria, las cuales fueron cubiertas en su totalidad por la solución propuesta.

La solución representa una oportunidad para aportar con el sector productivo del país debido a que hay interés en el sector camaronero por ver este proyecto convertido en un producto final. Sin embargo, el prototipo debe ser sometido a más pruebas con el fin de definir casos de uso y transformarlo en un producto final.

Un factor importante para mencionar es que la reducción en el consumo energético convierte a esta implementación en una mucho más eficiente con relación a otras soluciones similares que usan otras tecnologías.

El estándar LoRa tiene un consumo de energía menor (35mA/h) y una mayor distancia de cobertura que otras tecnologías como ZigBee, Bluetooth y Wi-Fi. Esto significa que LoRa no solo es más eficiente en términos energéticos, sino que también en términos de costo debido a que ZigBee, en función de los criterios de selección, su principal competidor por su función de multihopping, le permite llegar a largas distancias usando varios nodos, a su vez representa una elevación en los costos.

Al realizar pruebas en un ambiente de laboratorio se perdía el enlace al pasar por varias paredes con esto se pudo constatar que la línea de vista es de vital importancia si lo que se necesita es cubrir largas distancias.

El protocolo LoRa utiliza comunicación half-dúplex el cual puede ser una limitante en el desarrollo de aplicaciones, ya que no permite la comunicación de varios dispositivos a la vez. En el proyecto se utilizó el Gateway TTN que tiene un sistema de identificación de nodos mediante OTAA, el cual le permite manejar una gran cantidad de nodos a la vez sin tener interferencias entre sí.

## 4.2 Recomendaciones

Para llevar esta solución a producción se recomienda usar sensores mas robustos debido a que aquellos que se usaron en este estudio son para desarrollo.

Es de vital importancia, encapsular correctamente el dispositivo debido a que la salinidad del medio es dañina para los circuitos.

Verificar que los sensores a usarse puedan resistir las condiciones del agua de las piscinas.

En caso de no existir un punto de red se puede considerar la implementación de un modem satelital cuyos costos rondan entre los \$60 - \$90 mensuales.

La implementación de una red Sigfox permitiría la reducción de costos en infraestructura, sin embargo, habría dependencia respecto a la red.

Se recomienda usar sensores de gama alta debido a su robustez y mantenimiento mínimo.

## BIBLIOGRAFÍA

- [1] CFN, Corporacion Financiera Nacional, «Ficha Sectorial del Camarón,» Septiembre 2017. [En línea]. Available: <https://www.cfn.fin.ec/wp-content/uploads/2017/10/Ficha-Sectorial-Camaron.pdf>.
- [2] M. Ing. Manuel A. Muñoz Suárez, I. F. V. D. Ganchoza y M. Econ. Mayiya González Illescas, «Análisis del sector camaronero ecuatoriano y sus ventajas competitivas y comparativas para encarar un mercado internacional competitivo.,» de // *Congreso CTEC-UTMACH, Universidad Técnica de Machala.*, Machala, El Oro, Ecuador., 2016.
- [3] <https://www.cooking-hacks.com>, «LoraWAN for arduino and raspberry pi,» 1 Diciembre 2015. [En línea]. Available: <https://www.cooking-hacks.com/documentation/tutorials/lorawan-for-arduino-raspberry-pi-wasp mote-868-900-915-433-mhz>.
- [4] M. Rouse, «Low Power Wide Area Network,» 26 Septiembre 2017. [En línea]. Available: <https://internetofthingsagenda.techtarget.com/definition/LPWAN-low-power-wide-area-network>.
- [5] Sabas, «Haciendo IoT con Lora,» 25 Septiembre 2017. [En línea]. Available: <https://medium.com/beelan/haciendo-iot-con-lora-cap%C3%ADtulo-1-qu%C3%A9%C3%A9s-lora-y-lorawan-8c08d44208e8>.
- [6] Sabas, «Haciendo IoT con Lora,» 2 Octubre 2017. [En línea]. Available: <https://medium.com/beelan/haciendo-iot-con-lora-capítulo-2-tipos-y-clases-de-nodos-3856aba0e5be>.
- [7] Sabas, «Haciendo IoT con Lora,» 9 Octubre 2017. [En línea]. Available: <https://medium.com/beelan/haciendo-iot-con-lora-capítulo-3-tipos-de-gateways-756afdf0487d>.
- [8] Dfrobot, «Gravity Analog Dissolved Oxygen Sensor,» Dfrobot, 8 septiembre 2017. [En línea]. Available: <https://www.dfrobot.com/product-1628.html>.
- [9] naylampmechatronics, «Tutorial Sensor de Temperatura DS18B20,» naylampmechatronics, 2 Abril 2016. [En línea]. Available:

[https://naylampmechatronics.com/blog/46\\_Tutorial-sensor-de-temperatura-DS18B20.html](https://naylampmechatronics.com/blog/46_Tutorial-sensor-de-temperatura-DS18B20.html).

- [10] A. Sancho, «Smart Agriculture,» 29 Junio 2017. [En línea]. Available: [http://libelium.com/downloads/training/webinar\\_smart\\_agriculture\\_libelium\\_29\\_06\\_2017.pdf](http://libelium.com/downloads/training/webinar_smart_agriculture_libelium_29_06_2017.pdf).
- [11] Ministerio de Acuacultura y Pesca, «Precio Referencial del Camarón,» 11 Enero 2017. [En línea]. Available: <http://www.acuaculturypesca.gob.ec/wp-content/uploads/downloads/2017/01/Del-1-de-enero-al-31-de-enero-de-2017.pdf>.