

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Diseño e implementación de una plataforma de interpretación
de datos de trazabilidad en el transporte de los camarones

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Telecomunicaciones

Presentado por:

María de los Ángeles Segovia Sabando

Diego Antonio Silva Astudillo

GUAYAQUIL – ECUADOR

Año: 2022

DEDICATORIA

El presente proyecto va dedicado a mi padres, abuelos, familiares y seres queridos. A mis amigos y compañeros de cada semestre que trabajaron conmigo. De igual manera, va dirigido a mis profesores por sus enseñanzas, experiencias y amplio conocimiento en cada materia vista en mi etapa universitaria. Finalmente, a la rama estudiantil IEEE que son un grupo el cual conocí excelentes personas donde siempre me dieron su ayuda y apoyo en momentos que más necesitaba.

Diego Antonio Silva Astudillo

Este trabajo, culminación de una larga etapa de estudios que parecían interminables y muchas noches sin dormir, va dedicado a mis padres.

A mi madre, Rocío Sabando, por el apoyo y el ánimo incondicional y por ser el cimiento de mi vida.

A mi padre, Fernando Segovia, alumni ESPOL, que me inspiró a seguir esta carrera y poder ser su fuente de orgullo.

**María de los Ángeles Segovia
Sabando**

AGRADECIMIENTOS

Agradezco de todo corazón a mis profesores de la carrera, por sus enseñanzas que me dieron el conocimiento necesario para la elaboración de este proyecto. Al Ph.D Francisco Novillo y Ph.D Patricia Chávez por guiarme en el desarrollo del proyecto integrador, al Ing. Lenin Morejón por su cordial aporte. Además, agradezco a mis padres y amigos que siempre me acompañaron en todas las materias y gracias a ellos pude culminar mi carrera universitaria.

Diego Antonio Silva Astudillo

Quiero agradecerle a Dios por sobre todas las cosas, ya que gracias a Él y la sabiduría que me ha prestado he sido capaz de lograr esta gran meta.

También agradecer a mis padres, que han estado a mi lado en todo momento brindando su apoyo. A mis amigos, que siempre estuvieron ahí con sus palabras de apoyo y dándome fuerzas para no rendirme y seguir luchando.

Finalmente, gracias a todos los docentes de ESPOL que fueron guías en el camino recorrido durante la vida universitaria en la distinguida y destacada Escuela Superior Politécnica del Litoral (ESPOL).

María de los Ángeles Segovia

Sabando

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; María de los Ángeles Segovia Sabando y Diego Antonio Silva Astudillo, 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”




**María de los Ángeles
Segovia Sabando**



**Diego Antonio Silva
Astudillo**

EVALUADORES

Ph.D Francisco Vicente Novillo Parales
PROFESOR DE LA MATERIA



Ph.D Patricia Ximena Chávez Burbano
PROFESOR TUTOR

RESUMEN

El presente trabajo de titulación estuvo encaminado al diseño y la implementación de una plataforma de la lectura de la información de trazabilidad en la movilización y exportación de camarones.

Este proyecto se elaboró utilizando LoRaWAN, los dispositivos finales que miden la temperatura y localización del camarón, y la energía del dispositivo. La comunicación de datos es recibida al servidor de redes The Things Network en la que se usó la integración WebHooks para visualizar los datos en la plataforma Datacake.

Las medidas fueron decodificadas para su fácil interpretación de los datos y el monitoreo del producto. Además, por medio del dashboard se puede añadir widgets o plantillas según la conveniencia del parámetro medido. Los widgets pueden ser desde valores numéricos, gráficos, mapas, controles, etc. La personalización de visualizar los datos en función del tiempo ya sea en horas, días o meses. Establecer reportes diarios o programados y reglas con disparadores de sus valores límites medidos para recibir notificación por email, SMS, mensaje downlink entre otros. Todo el monitoreo y control de la trazabilidad en el transporte de camarones se tiene acceso por computadora de escritorio y por aplicación móvil, en la que el administrador de la plataforma puede compartir a sus usuarios dicha información.

Palabras Clave: codificación, dashboard, monitoreo y temperatura.

ABSTRACT

This degree work was aimed at the design and implementation of a platform for reading traceability information in the mobilization and export of shrimp.

This project was developed using LoRaWAN, the final devices that measure the temperature and location of the shrimp, and the energy of the device. The data communication is received to The Things Network network server where WebHooks integration was used to visualize the data in the Datacake platform.

The measurements were decoded for easy data interpretation and product monitoring. In addition, through the dashboard, widgets or templates can be added according to the convenience of the measured parameter. The widgets can range from numerical values, graphs, maps, controls, etc. The customization of displaying data according to time either in hours, days or months. Establish daily or scheduled reports and rules with triggers of your measured limit values to receive notification by email, SMS, downlink message among others. All monitoring and control of shrimp transport traceability can be accessed by desktop computer and mobile application, where the platform administrator can share this information with users.

Keywords: coding, dashboard, monitoring and temperature.

ÍNDICE GENERAL

EVALUADORES	7
RESUMEN	I
<i>ABSTRACT</i>	II
ÍNDICE GENERAL.....	III
ABREVIATURAS	V
ÍNDICE DE FIGURAS	VI
CAPÍTULO 1	7
1 Introducción.....	7
1.1 Descripción del problema.....	8
1.2 Justificación del problema.....	8
1.3 Objetivos.....	9
1.3.1 Objetivo General	9
1.3.2 Objetivos Específicos	9
1.4 Metodología	10
CAPÍTULO 2.....	11
2. Marco teórico	11
2.1 LoRaWan.....	11
2.2 TTN.....	12
2.3 Webhook.....	12
2.4 Datacake.....	13
2.5 Servidor Ubuntu	14
2.6 Servidor LAMP (Linux, Apache, MySQL, PHP).....	14
2.7 Manejo de bases de datos MySQL con phpMyAdmin	15
CAPÍTULO 3.....	16

3.	Diseño e implementación del escenario.....	16
3.1	Dispositivo final.....	17
3.2	LoRaWAN Gateway.....	17
3.3	Enrutador A.....	18
3.4	VPN	18
3.5	Enrutador B.....	19
3.6	Servidor Base de Datos.....	19
3.7	Network Server	20
3.8	App Server.....	21
3.8.1	Dashboard.....	21
3.8.2	Aplicación móvil.....	21
CAPÍTULO 4.....		23
4.	Resultados	23
4.1	Pruebas con el dispositivo final.....	23
4.2	Decodificación de la trama.....	24
4.3	Integración de datos a Datacake mediante Webhooks.....	25
4.4	Visualización en Datacake.....	26
CAPÍTULO 5.....		27
5.	Conclusiones y Recomendaciones	27
5.1	Conclusiones	27
5.2	Recomendaciones.....	27
BIBLIOGRAFÍA.....		29
6.	Bibliografía	29
APÉNDICES		31
7.	Código de la decodificación del Payload en TTN.....	31
8.	Código de la decodificación de la trama JSON en Datacake.....	32

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
LAMP	Linux, Apache, MySQL, PHP
LNS	LoRa Network Server
LPWA	Low Power Wide Area
TTN	The Things Network
VPN	Virtual Private Network

ÍNDICE DE FIGURAS

<i>Figura 2.1 Estructura de red LoRaWan.....</i>	<i>12</i>
<i>Figura 2.2 Tablero de visualización widgets con datos de la plataforma Datacake [13]</i>	<i>13</i>
<i>Figura 3.1 Diagrama esquemático del diseño.....</i>	<i>16</i>
<i>Figura 3.2 Dispositivo Final.....</i>	<i>17</i>
<i>Figura 3.3 LoRaWan Gateway.....</i>	<i>18</i>
<i>Figura 3.4 Aplicación Cisco AnyConnect.....</i>	<i>19</i>
<i>Figura 3.5 Iniciar sesión en phpMyAdmin</i>	<i>20</i>
<i>Figura 3.6 The Things Network.....</i>	<i>21</i>
<i>Figura 3.7 Aplicación móvil Datacake</i>	<i>22</i>
<i>Figura 4.1 Medición de temperatura</i>	<i>23</i>
<i>Figura 4.2 Envío de datos codificados desde el dispositivo final.....</i>	<i>24</i>
<i>Figura 4.3 Código de decodificación del payload en TTN.....</i>	<i>25</i>
<i>Figura 4.4 Payload decodificado.....</i>	<i>25</i>
<i>Figura 4.5 Integración a Datacek con Webhooks</i>	<i>25</i>
<i>Figura 4.6 Dashboard de la temperatura en Datacake</i>	<i>26</i>

CAPÍTULO 1

1 INTRODUCCIÓN

Según la revista Aquaculture, Ecuador lideró la lista de los mayores productores de camarón en el 2021 [1]. En el año 2022 desde el mes de enero hasta septiembre, las exportaciones de camarón obtuvieron un incremento de hasta USD 5.676 millones, esto significa que obtuvieron un crecimiento del 54% en comparación al mismo período en 2021, según datos del Banco Central [2].

El ministerio de Producción, Comercio Exterior, Inversiones y Pesca (MPCEIP) en la “AquaExpo Guayaquil” de noviembre del 2020, manifestó que Ecuador impulsará estrategias para ampliar el liderazgo en calidad y seguridad alimentaria del mercado mundial del camarón. Además, expresó que somos los primeros en el mundo en desarrollar tecnología blockchain en el camarón ecuatoriano. Con esta nueva herramienta se pretende asegurar la trazabilidad total de su producción, con el objetivo de que todos los consumidores tengan acceso a la información relacionada al cultivo del camarón [3].

Esta trazabilidad permite rastrear todos los procesos que ha seguido el camarón, desde el nacimiento de las larvas en el laboratorio, hasta la camaronera y la empacadora, afirmó el ministro Iván Ontaneda.

Es importante el sistema de control sobre el producto al ser un factor clave a la hora de mantener la seguridad alimentaria en todas las etapas que pasa el camarón. Este sistema permite detectar y alertar de posibles problemas como contaminación, ruptura de la cadena de frío o desvíos en la distribución y transporte del producto, al ser capaz de identificar el problema y facilitar la información pertinente en tiempo real.

Al exportar el camarón, una vez termina su proceso de producción, es transportado en gavetas especiales con temperaturas de 5°C para camarones frescos y -10°C para camarones congelados a granel, estas temperaturas son exigidas ya que de no ser así el camarón puede sufrir descomposición [4].

En nuestro proyecto de materia integradora se plantea el diseño e implementación de una plataforma de monitoreo y control de trazabilidad en el transporte de los camarones. De esta manera, tener registrado el estado del producto de camarón y conservar su alta calidad hasta que llegue a su lugar de destino.

1.1 Descripción del problema

La trazabilidad de los datos permite a las empresas planificar la gestión de datos en sus proyectos para prevenir o asegurar una fácil localización de errores que simplifique su corrección. Generar una trazabilidad de los datos brinda la capacidad de conocer todo el ciclo de vida de un dato [5].

Nuestro cliente de materia integradora, el Ing. Lenin Morejón, desarrolla dispositivos para aplicaciones en telemetría y ha diseñado nodos sensores que realizan la lectura de los datos durante el transporte del camarón. La tecnología utilizada para la comunicación de los datos es LoRaWAN, en el que la información llega a una plataforma de red global de Internet de las cosas (IoT) llamada “The Things Network” (TTN).

Los datos recibidos en la plataforma TTN son tramas codificadas en formato hexadecimal en el que indica las lecturas de temperatura y ubicación del camarón como también la energía del sensor. Sin embargo, la forma de presentación de los datos no es la mejor opción para hacer una correcta trazabilidad o interpretación por parte del usuario. Adicionalmente, el almacenamiento disponible para la información es limitado por lo que pasado un periodo de tiempo estos datos se pierden y no queda registrado en una base de datos. Al realizar la trazabilidad de los datos, al usuario se le dificulta la tarea debido a que la información no es comprensible en su formato actual. De igual manera, el monitoreo y control de los procesos del camarón se ven afectados por los problemas de interpretación de la data que presentan los usuarios.

1.2 Justificación del problema

Las empresas dedicadas a la producción de camarón no poseen, en su mayoría, tecnologías capaces de realizar una correcta medición de los datos del camarón, por ejemplo, peso de recepción, peso del descabezado, pelado, desvenado, temperatura de frío en bodega y peso del producto terminado de manera inmediata. En estos casos, se deben tomar diferentes muestras para un análisis generando retrasos y lentitud en el proceso. En consecuencia, no pueden implementar un eficaz método para el monitoreo del área camaronero. Esta poca atención al momento de exportar el producto resulta en una información desorganizada, dispersa y con alto margen de error que afecta a los procesos [6].

La implementación de software de control en los procesos de producción de la industria camaronera permite maximizar su potencial competitivo en el mercado internacional. El uso de este tipo de software trae importantes beneficios: mejora y optimiza múltiples recursos en cada una de las áreas, por ejemplo, el control de producción y trazabilidad del proceso de sus mercaderías [6]. Además, la recolección de metadatos ayuda a recopilar nueva información y variables que serán de ayuda para un mejor entendimiento y análisis de la problemática [7].

1.3 Objetivos

El objetivo general de nuestro proyecto integradora es el resultado final de nuestra propuesta de solución, acompañado de objetivos específicos que serán las actividades y desafíos que nos facilitarán a conseguir el propósito general del proyecto.

1.3.1 Objetivo General

- Desarrollar una interfaz visual para el monitoreo del proceso de trazabilidad del camarón.

1.3.2 Objetivos Específicos

- Identificar las tramas de datos para decodificar la información de las mediciones.
- Diseñar una base de datos y categorizar los datos decodificados para relacionar los campos de cada parámetro.
- Implementar una plataforma de visualización que muestre los parámetros registrados en la base de datos.

1.4 Metodología

En primer lugar, se propuso crear un servidor local en la nube con la base de datos almacenando los datos medidos por los sensores, así mismo llevar la integración de los datos hacia una plataforma gráfica que se utilizan en proyectos IoT como son Datacake, Ubidots, entre otros. El acceso a este servidor se pretendía que sea público mediante una IP fija en la que el usuario y el cliente tengan a disposición la información del monitoreo en la trazabilidad del transporte del camarón.

Debido a las dificultades del servicio, seguridad e integridad de tener una IP pública fija, se escogió instalar el servidor local en una computadora fija en uno de los laboratorios de la universidad con acceso hacia el servidor por VPN, de esta manera el estudiante puede conectarse a su servidor de manera local si se encuentra dentro de la universidad o remotamente desde cualquier lugar externo por VPN.

La información que llega a la plataforma TTN, es extraída todos los datos y registrados en una base de datos con un formato personalizado de la trama de cada parámetro medido. El servidor web tiene sistema operativo Linux, específicamente la distribución Ubuntu Server 22.04, se instalan los softwares LAMP: Linux, Apache, MySQL y PHP. Para la gestión de las bases de datos utilizamos MySQL y el entorno de trabajo para crear, configurar y monitorear dichos datos es phpMyAdmin.

Se crearon bases de datos de prueba para simular los datos que van a extraerse de la plataforma TTN, para luego realizar la integración mediante la herramienta WebHooks. Entre las plantillas que WebHooks tiene disponible usamos la plataforma visual para IoT Datacake.

Finalmente se diseña la aplicación usando graficadores de temperatura, seguimiento de rutas en el mapa a tiempo real del transporte e indicadores de energía del sensor que facilitan la interpretación de los datos de una manera sencilla, con la posibilidad de generar reportes y notificaciones de alarmas para tener en absoluto control la trazabilidad del camarón.

CAPÍTULO 2

2. MARCO TEÓRICO

En este capítulo se describirá la teoría que abarca nuestro proyecto, los conceptos de cada tema y una breve explicación de las herramientas, tecnologías y servicios que se utilizan a lo largo del proyecto.

2.1 LoRaWan

LoRaWAN se define como un protocolo de red con gran cobertura de área y baja potencia (LPWA), diseñado con el propósito de conectar inalámbricamente dispositivos operados con baterías a redes de Internet globales. LoRaWan se enfoca en factores especializados de IoT, como servicios de comunicación bidireccionalidad, seguridad, movilidad y ubicación. Como puede visualizarse en la Figura 2.1, este protocolo está compuesto por puertas de enlaces y nodos. Sus puertas de enlaces, o antenas, se encargan de recibir información de los nodos, los cuales son dispositivos finales que envían datos hacia el concentrador [8].

Estos nodos pueden ser activados de dos maneras:

- Inalámbrica: un nodo establece una contraseña única a su sesión de aplicación y de red, para comunicarse por medio de internet.
- Personalización: un nodo requiere un codificado con las contraseñas pertinentes para conectarse a la red [9].

Entre las características más destacadas se tiene:

- Comunicación a largas distancias.
- Mayor duración de la batería.
- Implementación de bajo costo.
- Espectro de radio sin licencia, con regulaciones específicas de la región [8].

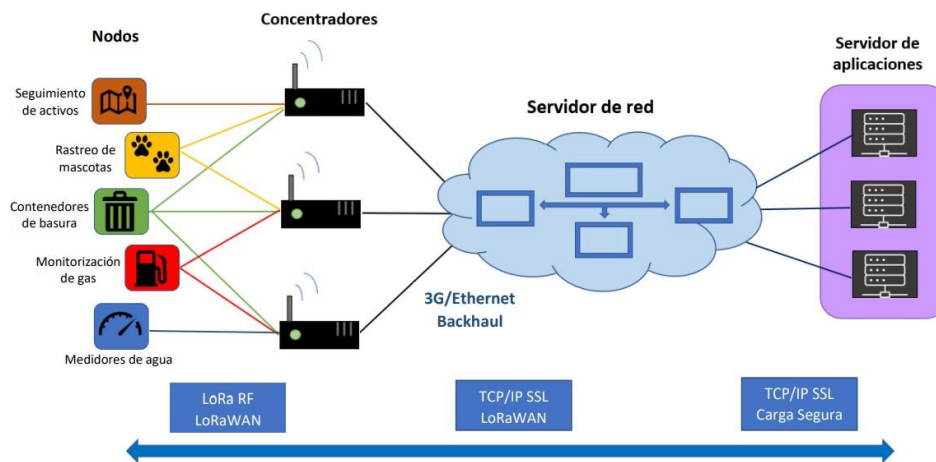


Figura 2.1 Estructura de red LoRaWan

2.2 TTN

The Things Network (TTN) es una iniciativa basada en la comunidad de desarrolladores de aplicaciones para IoT, la cual tiene como objetivo la implementación de redes LoRaWan en todas las ciudades alrededor del mundo, para interconectar estas redes a través de una infraestructura global. TTN proporciona herramientas abiertas, recursos completos y una red universal que permite a usuarios la búsqueda de hardware que permitan una óptima creación de aplicaciones IoT de manera económica y segura [10].

2.3 Webhook

Un webhook es una respuesta de llamada HTTP provocada por una notificación de evento. Luego, esta solicitud HTTP se registra como una URL de webhook y se usa para almacenar data. Los webhooks facilitan esa transferencia de datos sin que alguien tenga que ingresar y extraer los datos manualmente.

Los webhooks son eventos que desencadenan una acción. En la mayoría de los casos, se utilizan para la comunicación entre sistemas. Esta es la forma más sencilla de recibir una alerta cuando algo sucede en otro sistema. Se denominan webhooks porque funcionan como enlaces de software en la web.

Los webhooks se utilizan para notificaciones en tiempo real, por lo que su sistema puede actualizarse justo cuando se lleva a cabo el evento. Además de esto, permiten sincronizar sistemas en tiempo real, enviar cualquier tipo de notificación, procesar datos y crear informes [11].

2.4 Datacake

Datacake es una plataforma IoT que permite al usuario crear un diseño, prototipo, y visualizar soluciones IoT rápidamente, se puede utilizar Datacake como herramienta para visualizar datos en tiempo real y registrar un historial de la información, enviado a través de TTN [12].

Entre los dispositivos que trabaja con Datacake son la Raspberry, Arduino y LoRa, este último es el que utilizamos en nuestro proyecto mediante el protocolo LoRaWAN. La característica principal de Datacake es el crear dashboard, como se observa en la figura 2.2, es la pantalla de monitoreo en la que podemos organizar, manejar, personalizar, monitorear y controlar nuestros dispositivos conectados. Una de las funciones es tener acceso a nuestros proyectos de manera remota desde una aplicación móvil, usar widgets para ajustar los controles y visualizar datos de temperatura, humedad, energía, trazas y localización en el mapa. El recibir notificaciones por mensaje de texto o correo electrónico basado a eventos específicos como anomalías en el entorno para que se activen los disparadores y tomar solución mediante la aplicación [13].

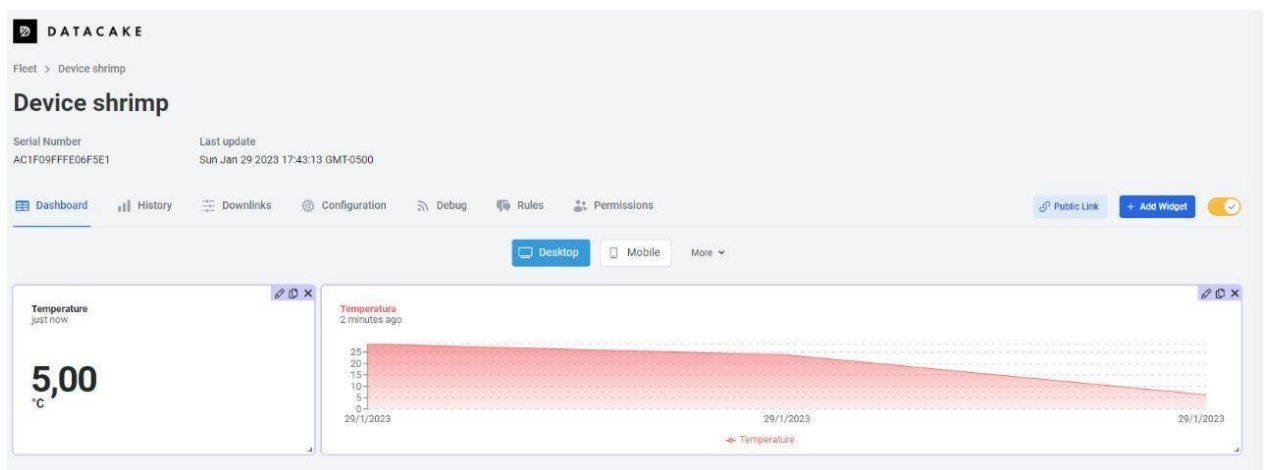


Figura 2.2 Tablero de visualización widgets con datos de la plataforma Datacake [13]

2.5 Servidor Ubuntu

Ubuntu Server es una plataforma de código abierto que funciona como un sistema operativo de servidor interno o para escalar hacia arriba y hacia afuera de acuerdo con las necesidades de la empresa. Opera en la mayoría de las plataformas de virtualización o computadoras. El sistema Ubuntu Server instalado tiene estos requisitos mínimos:

- RAM: 4 GB
- CPU: 2 GHz
- Almacenamiento: 25 GB de espacio disponible en disco duro [14].

2.6 Servidor LAMP (Linux, Apache, MySQL, PHP)

LAMP es una plataforma de desarrollo web de código abierto, donde cada letra de su acrónimo representa uno de sus cuatro bloques de construcción de código. Utiliza Linux como sistema operativo, Apache como servidor web, MySQL como sistema de gestión de bases de datos relacionales y PHP como lenguaje de programación orientado a objetos. Debido a que la plataforma tiene cuatro capas, LAMP a veces se denomina pila LAMP, sin embargo, las pilas se pueden construir en diferentes sistemas operativos.

LAMP denota una de las pilas de software más comunes para muchas de las aplicaciones más populares de la web. Actualmente, LAMP se refiere a un modelo de pila de software genérico y sus componentes son en gran parte intercambiables [15].

2.7 Manejo de bases de datos MySQL con phpMyAdmin

PhpMyAdmin es una herramienta escrita en PHP con el propósito de manejar la administración MySQL a través de una página web. Puede crear y eliminar base de datos, diseñar tablas, editarlas, borrarlas, añadir campos, ejecutar sentencias en el lenguaje SQL, administrar claves, dar privilegios, exportar e importar datos en varios formatos [16]. Es una herramienta gratuita que garantiza el control de nuestras bases de datos, ofrece una gran cantidad de características y opciones para manejar una base de datos con una interfaz sencilla e intuitiva de alto nivel [17].

CAPÍTULO 3

3. DISEÑO E IMPLEMENTACIÓN DEL ESCENARIO

El diseño implementado del proyecto posee una arquitectura LoRaWAN que es una topología de red tipo estrella, los dispositivos finales se comunican a través de un concentrador/ LoRa Gateway hasta el enrutador A. La conexión entre el servidor de base de datos MySQL y nuestra red externa es mediante el enrutador B por medio de VPN, tal como se puede apreciar en la Figura 3.1.

La segunda estrella del diseño inicia desde el servidor de red central que conecta The Things Network hacia nuestra plataforma de aplicación Datacake, el cual permite la visualización y monitoreo de los datos mediante un dashboard y aplicación móvil [18].

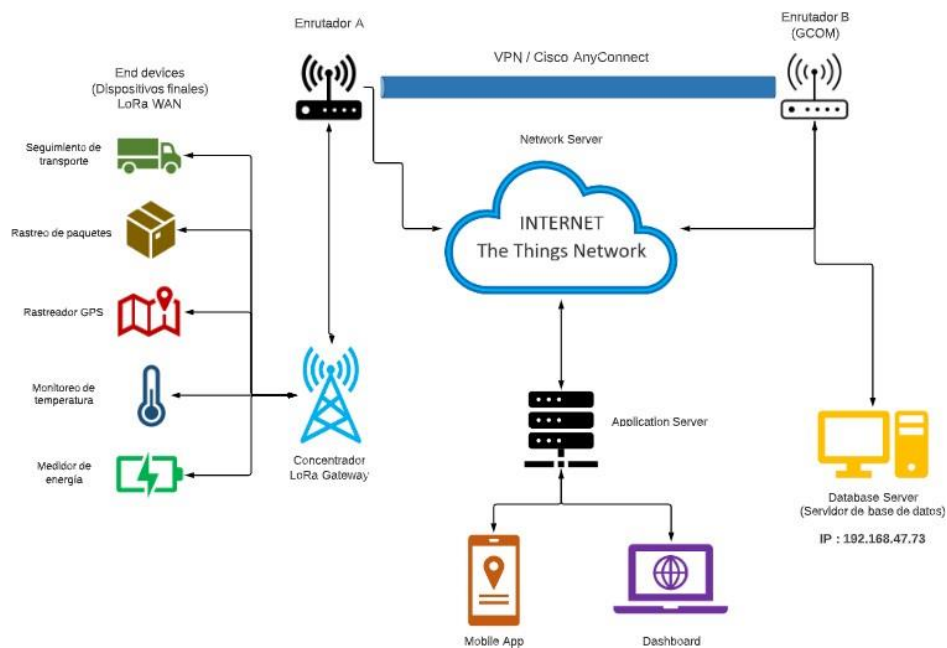


Figura 3.1 Diagrama esquemático del diseño

3.1 Dispositivo final

El dispositivo final de la Figura 3.2, está ubicado en el transporte que lleva el camarón y se utilizan sensores para tener un control del producto durante su exportación. Este dispositivo final mide los siguientes parámetros:

- Temperatura: El camarón debe permanecer bajo una temperatura de frío ideal para su conservación.
- Ubicación: Se obtienen las coordenadas en valores de longitud y latitud del transporte. Me permite conocer la localización del camarón en tiempo real y observar la ruta por el cual es transportado hasta llegar a su destino.
- Energía: La energía del dispositivo en caso de que requiera un cambio de batería y mantener los dispositivos encendidos en todo momento.



Figura 3.2 Dispositivo Final

3.2 LoRaWAN Gateway

Permite la comunicación bidireccional de los datos provenientes de los dispositivos finales hacia nuestro servidor de red central. En el concentrador de la Figura 3.3 se configura la ruta de envío del paquete y la trama codificada de dicha información para almacenarla en el servidor de bases de datos MySQL.



Figura 3.3 LoRaWan Gateway

3.3 Enrutador A

El enrutador A actúa como cualquier red externa con acceso a internet. Si el usuario se encuentra bajo una dirección de red externa a una red de ESPOL, se puede conectar al servidor de base de datos por medio de la VPN de ESPOL. El acceso remoto es con la aplicación Cisco AnyConnect.

3.4 VPN

El servidor de bases de datos está instalado en una computadora dentro de los laboratorios de ESPOL. Para acceder al servidor fuera de ESPOL, se debe conectar a la VPN de la universidad y luego acceder por conexión SSH a la dirección IP 192.168.47.73. La conexión por VPN es utilizando el software Cisco AnyConnect. La aplicación nos solicita ingresar el nombre de usuario de ESPOL y la contraseña como se aprecia en la Figura 3.4.

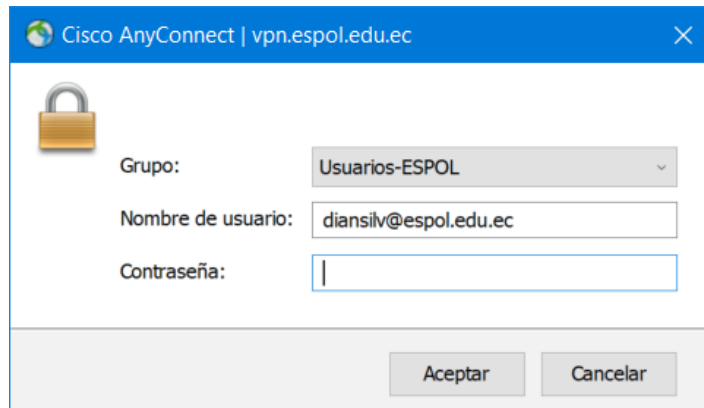


Figura 3.4 Aplicación Cisco AnyConnect

3.5 Enrutador B

El enrutador B es el punto de conexión hacia internet desde ESPOL. También es el que permite la comunicación hacia la red externa o enrutador A por VPN. Si el usuario se encuentra utilizando la red de la universidad, puede ingresar al servidor de bases de datos con solo escribir la dirección IP en cualquier navegador web.

3.6 Servidor Base de Datos

El servidor de base de datos fue implementado con el sistema de gestión MySQL, con la integración de phpMyAdmin como herramienta para manejar, administrar, gestionar, insertar e importar las tablas de datos con la información procesada. La dirección IP del servidor es 192.168.47.73 y para ingresar a la base de datos se escribe la extensión “/phpmyadmin”. Una vez dentro se puede observar lo mostrado en la Figura 3.5 que corresponde a la pantalla de inicio de sesión.

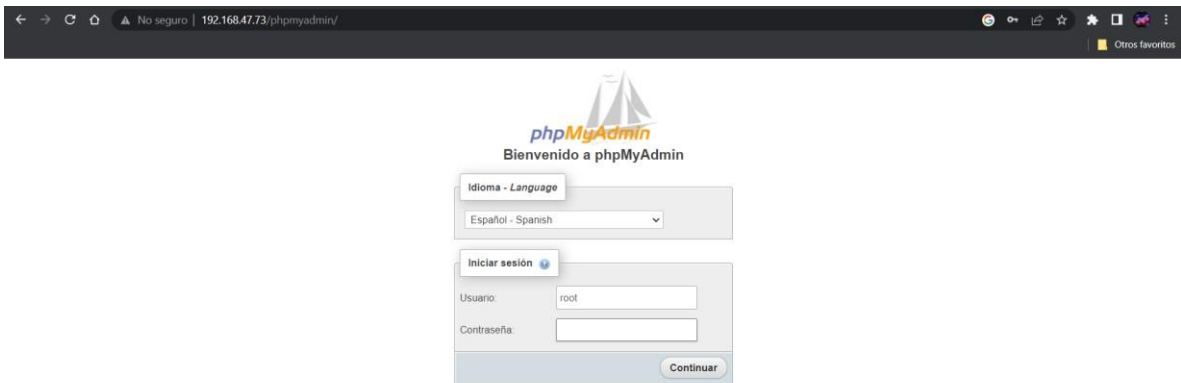


Figura 3.5 Iniciar sesión en phpMyAdmin

3.7 Network Server

Consiste en una red de redes que sirve como canal de paso de la información, en otras palabras, es el encargado de que la topología se encuentre intercomunicada. Para nuestro caso, el Network Server a utilizar es The Things Network. En la Figura 3.6 se puede visualizar la llegada de los datos y su constante actualización de información receptada desde el LoRa gateway.

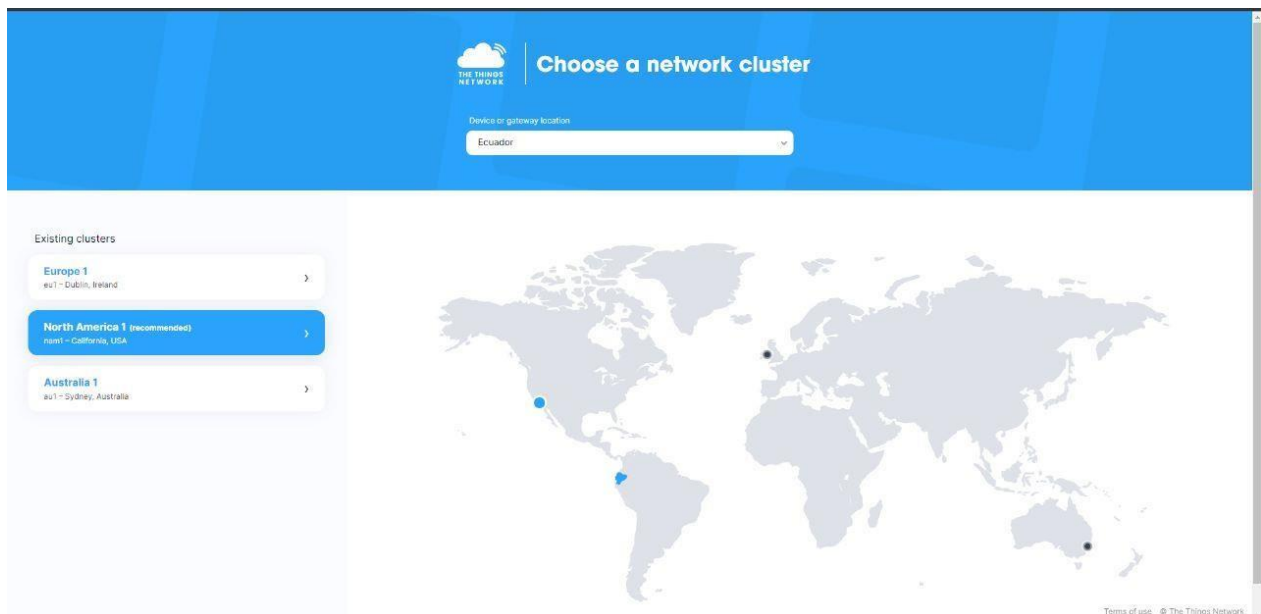


Figura 3.6 The Things Network

3.8 App Server

El servidor de aplicación, que para nosotros es Datacake, tiene como finalidad interpretar los datos provenientes de los dispositivos finales y agruparlos en una sola aplicación. Para la comunicación de The Things Network con Datacake, se debe realizar la integración de datos mediante la herramienta Webhooks.

3.8.1 Dashboard

También conocido como tablero, este es mostrado en Datacake con una pestaña de descripción, donde se muestran widgets como mapas de ubicación y datos del sensor. Los datos son actualizados en vivo, de manera periódica y automática. Además, que se puede acceder al historial de datos registrados previamente.

3.8.2 Aplicación móvil

Con el servidor de aplicación Datacake, es posible visualizar los mismos datos llegados al dashboard en formato para aplicación móvil como se observa en la Figura 3.7, lo cual permite el monitoreo de los datos de una manera más compacta y fácil de transportar.

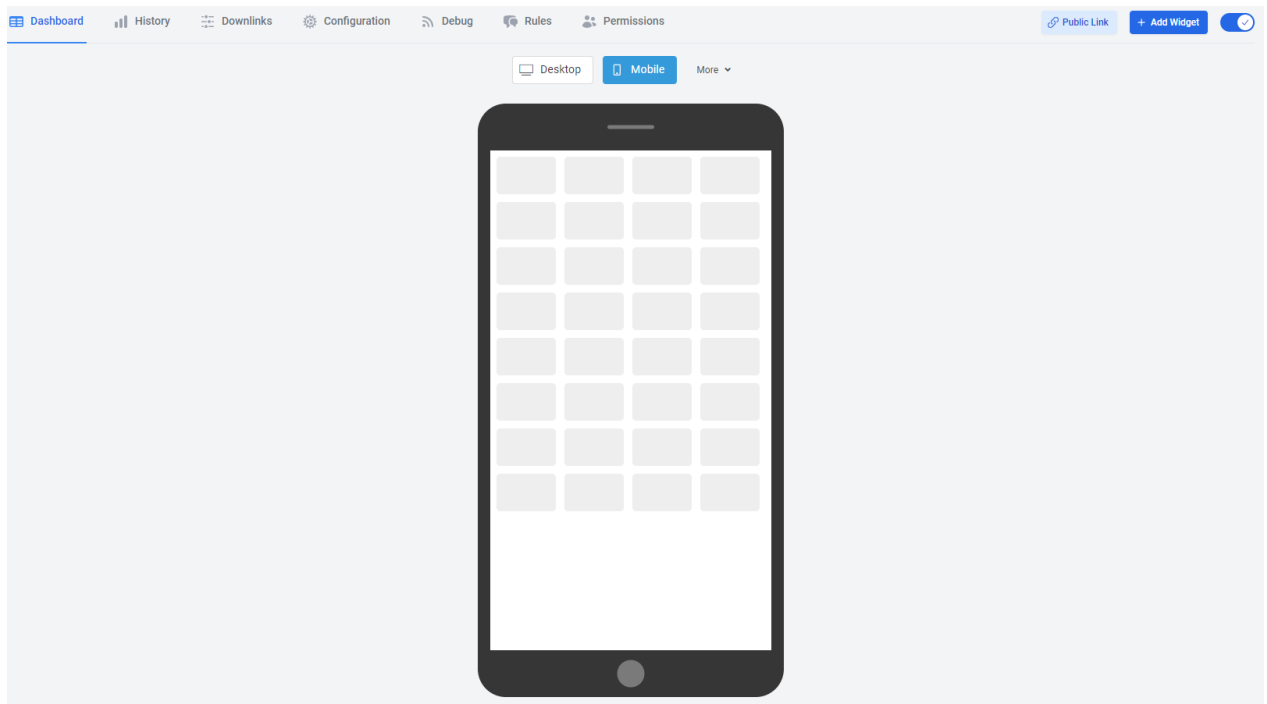


Figura 3.7 *Aplicación móvil Datacake*

CAPÍTULO 4

4. RESULTADOS

En este capítulo se procede a explicar las pruebas realizadas y el resultado final de la implementación del prototipo. Además, la creación del diseño de una plataforma visual en la que se ilustra de manera gráfica la interpretación de los datos.

4.1 Pruebas con el dispositivo final

Inicialmente se hicieron pruebas con el dispositivo final midiendo la temperatura con el sensor digital DS18B20. El sensor se lo colocó en un recipiente con agua helada, ver Figura 4.1, para comprobar que los datos están siendo enviados correctamente al TTN.



Figura 4.1 Medición de temperatura

Como se puede observar en la Figura 4.2, el dispositivo final fue configurado correctamente y envía de forma periódica, cada 20 segundos, el dato de temperatura (considerado el payload) de forma codificada.

Time	Type	Data preview
↑ 17:45:41	Forward uplink data message	DevAddr: 27 FE 09 CC <> Payload: 04 08 7F <> FPort: 2 Data rate: SF78u125 SNR: 9.5 RSSI: -16
↑ 17:45:41	Successfully processed data message	DevAddr: 27 FE 09 CC <>
↑ 17:45:22	Forward uplink data message	DevAddr: 27 FE 09 CC <> Payload: 04 08 7F <> FPort: 2 Data rate: SF78u125 SNR: 10.2 RSSI: -18
↑ 17:45:22	Successfully processed data message	DevAddr: 27 FE 09 CC <>
↑ 17:44:01	Forward uplink data message	DevAddr: 27 FE 09 CC <> Payload: 04 08 6C <> FPort: 2 Data rate: SF78u125 SNR: 9.5 RSSI: -18
↑ 17:44:01	Successfully processed data message	DevAddr: 27 FE 09 CC <>
↑ 17:44:41	Forward uplink data message	DevAddr: 27 FE 09 CC <> Payload: 04 08 66 <> FPort: 2 Data rate: SF78u125 SNR: 10 RSSI: -18
↑ 17:44:41	Successfully processed data message	DevAddr: 27 FE 09 CC <>
↑ 17:44:21	Forward uplink data message	DevAddr: 27 FE 09 CC <> Payload: 04 08 5F <> FPort: 2 Data rate: SF78u125 SNR: 7 RSSI: -13
↑ 17:44:21	Successfully processed data message	DevAddr: 27 FE 09 CC <>
↑ 17:44:01	Forward uplink data message	DevAddr: 27 FE 09 CC <> Payload: 04 08 59 <> FPort: 2 Data rate: SF78u125 SNR: 10 RSSI: -18
↑ 17:44:01	Successfully processed data message	DevAddr: 27 FE 09 CC <>
↑ 17:43:41	Forward uplink data message	DevAddr: 27 FE 09 CC <> Payload: 04 08 53 <> FPort: 2 Data rate: SF78u125 SNR: 10.2 RSSI: -18
↑ 17:43:41	Successfully processed data message	DevAddr: 27 FE 09 CC <>
↑ 17:43:21	Forward uplink data message	DevAddr: 27 FE 09 CC <> Payload: 04 08 4D <> FPort: 2 Data rate: SF78u125 SNR: 7.8 RSSI: -15
↑ 17:43:21	Successfully processed data message	DevAddr: 27 FE 09 CC <>
↑ 17:43:01	Forward uplink data message	DevAddr: 27 FE 09 CC <> Payload: 04 08 46 <> FPort: 2 Data rate: SF78u125 SNR: 10 RSSI: -19

Figura 4.2 *Envío de datos codificados desde el dispositivo final*

4.2 Decodificación de la trama

El payload de temperatura tiene una longitud de 3 bytes, el primer byte es “04” que es la cabecera de un dato de temperatura y los bytes siguientes corresponden al valor medido. Para la decodificación se utilizó el lenguaje de programación Javascript en el que creamos una función de decodificación para los datos Uplink, es decir, desde el dispositivo final al TTN.

La plataforma TTN automáticamente convierte la trama en un formato decimal, por lo que en el código tiene como entrada el payload en decimal. Se elige el puerto por el cual se recibe la trama y se crea la variable temperatura en la que añadiremos los últimos 2 bytes de la trama recibida. La temperatura se divide para 100 para que el resultado retorne con 2 cifras significativas, como se observa en la Figura 4.3. En la Figura 4.4, tenemos la pantalla “Live data” que observamos la temperatura ya convertida y recibida en el TTN.

Setup

Formatter type*

Formatter code*

```

1 function decodeUplink(input) {
2   var data = {};
3   var warnings = [];
4   var errors = [];
5
6   if(input.FPort == 2) {
7     var temperature = (input.bytes[1]<<8) | input.bytes[2];
8     data.temperature = temperature/100;
9   } else {
10    errors.push("payload unknown");
11  }
12  return {
13    data: data,
14    warnings: warnings,
15    errors: errors
16  };
17 }

```

Test

Byte payload FPort

Decoded test payload

```

{
  "temperature": 17.18
}

```

Complete uplink data

```

{
  "f_port": 2,
  "firm_payload": "8Aa2",
  "decoded_payload": {
    "temperature": 17.18
  },
  "ix_metadata": [
    {
      "gateway_ids": {
        "gateway_id": "test"
      }
    }
  ]
}

```

Figura 4.3 Código de decodificación del payload en TTN

[Overview](#) | [Live data](#) | [Messaging](#) | [Location](#) | [Payload formatters](#) | [Claiming](#) | [General settings](#)

Time	Type	Data preview
↑ 17:35:32	Forward uplink data message	DevAddr: <input type="text" value="27 FE 09 CD"/> <input type="button" value="<>"/> <input type="button" value="🗑️"/> Payload: { temperature: 28.56 } <input type="text" value="04 06 B6"/> <input type="button" value="<>"/> <input type="button" value="🗑️"/> FPort: 2 Data rate: SF7BW125 SNR: 9 RSSI: -10
↑ 17:35:32	Successfully processed data message	DevAddr: <input type="text" value="27 FE 09 CD"/> <input type="button" value="<>"/> <input type="button" value="🗑️"/>

Figura 4.4 Payload decodificado

4.3 Integración de datos a Datacake mediante Webhooks

En TTN se realizó la integración de los datos con Webhooks y se eligió como plantilla la plataforma Datacake. En la Figura 4.5 tenemos que el formato Webhook de envío que utilizamos es JSON, por lo que toda la data enviada hacia la URL de la aplicación tendrá este tipo de formato de entrada.

Edit webhook for Datacake

Send data to Datacake via TTI adapter

[About Datacake](#) | [Documentation](#)

General settings

Webhook ID*

Webhook format*

Base URL*

Figura 4.5 Integración a Datacake con Webhooks

4.4 Visualización en Datacake

Debido a la integración con Webhooks los datos que recibe Datacake son en formato JSON. En las configuraciones de la aplicación está la opción de decodificar esa trama, similar como en TTN, con la diferencia que se creó un código de codificación para toda la trama del dato y no únicamente del payload. Es por esto por lo que podemos obtener datos como SNR y RSSI que son provenientes del Gateway. Estos datos pueden ir modificando según la potencia de las antenas y la distancia en que se encuentra el dispositivo, pero al ser una arquitectura LoRaWAN es posible tener una comunicación de gran alcance sin pérdidas de paquetes.

En la creación de la función de decodificación, hecha con Javascript, se determina la variable temperatura y los campos de información del Gateway. Una vez se comprobó que la decodificación funciona, automáticamente se visualizará los datos en la pestaña Dashboard en Datacake. Presentó muy poco retardo por lo que era prácticamente imperceptible la latencia que hay al envío de la carga útil.

Una de las características de Datacake es enviar downlinks desde la aplicación, se escribe el mensaje en formato codificado para que el payload sea enviado hacia el puerto de destino. De esta manera es posible interactuar con el dispositivo final a través del Gateway y cada movimiento de información uplink y downlink se puede visualizar en el TTN.

Se añadió widgets como valores numéricos para los parámetros del LoRa Gateway y un gráfico en función del tiempo de la temperatura, como se observa en la Figura 4.6.



Figura 4.6 Dashboard de la temperatura en Datacake

CAPÍTULO 5

5. CONCLUSIONES Y RECOMENDACIONES

En esta sección se explica la importancia del trabajo desarrollado, el significado de los resultados y las fortalezas que determina el proyecto para trabajos futuros.

5.1 Conclusiones

- El diseño implementado para el proyecto funciona correctamente, presentando las gráficas requeridas de temperatura sin inconvenientes en un tiempo de 20 segundos con un retardo de aproximadamente 0.10 segundos, como se observa en la sección de resultados, específicamente en la Figura 4.2.
- Gracias a la plataforma TTN se facilitó la decodificación de tramas de datos de temperatura, en la que configuramos un código para la transformación de una trama hexadecimal a decimal. Se realizó con la ayuda de la herramienta Payload Formatters de TTN y el lenguaje de programación Javascript, como se ilustra en la Figura 4.3.
- En nuestra plataforma de visualización Datacake, se mostró los datos decodificados en un formato decimal para que el usuario final pueda interpretar fácilmente las mediciones. El dashboard diseñado ha sido totalmente personalizado de acuerdo con las especificaciones del cliente. De esta manera, los valores de la temperatura y otras mediciones que se deseen añadir en el futuro se pueden visualizar amigablemente con widgets y gráficos, tal como se observa en la Figura 4.6.

5.2 Recomendaciones

- Es recomendado revisar el plan de usuario utilizado en la aplicación visual debido a las limitaciones de datos permitidos en un día. Esto te obliga en aumentar el periodo de envíos de los datos desde la plataforma TTN hacia nuestra página en Datacake.

- Al ser los datos recibidos por los sensores receptados de manera codificada, se recomienda la utilización de una función decodificadora en un lenguaje de programación, para que de esta manera la trama pueda ser convertida a información entendible para el usuario una vez que esta sea presentada en la plataforma de TTN.
- Para futuros proyectos similares al mostrado en el presente documento se recomienda la integración de datos a través de la herramienta Webhooks, para de esta manera el envío de datos es por protocolo HTTP que es un método más sencillo utilizado en proyectos de IoT.

BIBLIOGRAFÍA

6. BIBLIOGRAFÍA

- [1] Acuicultura, «Ecuador primer producto de camarón a nivel mundial,» p. 9, 2 Marzo 2022.
- [2] G. Coba, «Las exportaciones de camarón batirán récord histórico,» 2022.
- [3] Ministerio de Producción, Comercio Exterior, Inversiones y Pesca, «Camarón ecuatoriano, primero en el mundo en garantizar completa trazabilidad de su producción,» Guayaquil, 2020.
- [4] A. Zuñiga, «Cadena Logística del Camarón,» 29 Julio 2015. [En línea].
- [5] Logicalis , «Trazabilidad de los datos,» 23 Abril 2014. [En línea].
- [6] Acuicultura, «Ventajas competitivas al implementar software en el control de producción y trazabilidad del camarón,» pp. 58-62, 25 Febrero 2021.
- [7] Informatica, «Gestión de metadatos: el enfoque de Informatica,» [En línea].
- [8] AWS IoT Core, «What is LoRaWAN?,» [En línea]. Available: <https://docs.aws.amazon.com/iot/latest/developerguide/connect-iot-lorawan-what-is-lorawan.html>.
- [9] Trend Micro, «LoRaWAN,» [En línea]. Available: <https://www.trendmicro.com/vinfo/us/security/definition/lorawan>.
- [10] The Things Network, [En línea]. Available: <https://www.thethingsnetwork.org/>.
- [11] Mailjet, «What is a webhook and why do you need it?,» 27 Septiembre 2022. [En línea]. Available: <https://www.mailjet.com/blog/email-best-practices/what-is-webhook/>.
- [12] The Things Stack, «Datacake,» [En línea]. Available: <https://www.thethingsindustries.com/docs/integrations/cloud-integrations/datacake/>.
- [13] Datacake, «Datacake Low-Code IoT Platform,» 2021. [En línea]. Available: <https://datacake.co/low-code-iot-platform>.

- [14] J. Wallen, «Ubuntu Server: A cheat sheet,» TechRepublic, 10 Diciembre 2020. [En línea]. Available: <https://www.techrepublic.com/article/ubuntu-server-the-smart-persons-guide/>.
- [15] B. Lutkevich, «LAMP (Linux, Apache, MySQL, PHP),» WhatIs.com, [En línea]. Available: <https://www.techtarget.com/whatis/definition/LAMP-Linux-Apache-MySQL-PHP>.
- [16] M. M. Carbonell, «phpMyadmin Introducción a la creación de base de datos,» Valencia, España.
- [17] J. M. L. Banqueri, «Creación y gestión de una base de datos con MySQL y phpMyAdmin,» Jaén, 2018.
- [18] Productos IoT, «Tecnología LoraWan: Internet de las cosas,» 2022. [En línea]. Available: <http://productos-iot.com/lorawan-3/>.

APÉNDICES

7. CÓDIGO DE LA DECODIFICACIÓN DEL PAYLOAD EN TTN

```
function decodeUplink(input) {
  var data = {};
  var warnings = [];
  var errors = [];

  if(input.fPort == 2) {
    var temperature = (input.bytes[1]<<8) | input.bytes[2];
    data.temperature = temperature/100;
  } else {
    errors.push("payload unknown");
  }
  return {
    data: data,
    warnings: warnings,
    errors: errors
  };
}
```

8. CÓDIGO DE LA DECODIFICACIÓN DE LA TRAMA JSON EN DATACAKE

```
function Decoder(bytes, port) {
  if (bytes.length == 3){
    var temperature = (bytes[1]<<8) | bytes[2];
  }
  datacakeFields = [
    {
      field : 'Temperature'
      value : temperature/100
    },
  ];
  // Extract Gateway Information

  if (normalizedPayload.data_rate && normalizedPayload.gateways &&
normalizedPayload.gateways[0]){

datacakeFields.push({field:"LORA_SNR",value:normalizedPayload.gateways[0].snr});
datacakeFields.push({field:"LORA_RSSI",value:normalizedPayload.gateways[0].rssi});
datacakeFields.push({field:"LORA_SNR",value:normalizedPayload.data_rate});
  } else {
    console.log("No Gateway Metada found - Skipping")
  }
  return datacakeFields
}
```