

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

IMPLEMENTACIÓN DE UN PROTOTIPO PARA EL CONTROL DEL CUMPLIMIENTO DE CUARENTENA DE PACIENTES CON COVID19 EN LA CIUDAD DE QUITO

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN TELECOMUNICACIONES**

DIEGO RENÉ VELÁSQUEZ GARCÍA

diego_velasquez_garcia@live.com

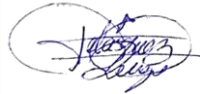
DIRECTOR: M.Sc. CARLOS ROBERTO EGAS ACOSTA

cegas@ieee.org

Quito, septiembre 2022

CERTIFICACIONES

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



DIEGO RENÉ VELÁSQUEZ GARCÍA

Certifico que el presente trabajo de integración curricular fue desarrollado por Diego René Velásquez García bajo mi supervisión.



ING. CARLOS ROBERTO EGAS ACOSTA, MSC
DIRECTOR DE TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

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

DIEGO RENÉ VELÁSQUEZ GARCÍA

MSC. CARLOS ROBERTO EGAS ACOSTA ING.

DEDICATORIA

A Dios Bendito Todopoderoso

A mi padre Fausto René Velásquez y a mi madre Marisol García

A mis hermanos Luis Enrique, Guillermo y Rafael

A mi amada Michelle y mi Joseph

A ustedes, mi familia, por su confianza, paciencia y amor

Diego René Velásquez García

AGRADECIMIENTO

Gracias a Dios por sus bendiciones, por permitirme seguir adelante, darnos el regalo de la vida y brindarnos su luz, sabiduría, fuerzas y su amor.

Agradecido con mucha razón y haciendo valer todo su esfuerzo, a mi madrecita Marisol, ella que me ha dado todo, desde la vida misma, cada gesto de amor, lucidez, paciencia, su visión de vida, su tenacidad, su fuerza, mi madre es todo para mí y ella se refleja en mí, en cada paso que doy, en cada acción, en cada logro, en cada meta, te amo Mamisol, de igual manera y con el mismo amor a mi padre Fausto, su apoyo incondicional, la manera en que hace las cosas, sus valores, justicia, humildad, rectitud, su manera de ser, su apertura y carisma, han inspirado y forjado lo que ahora soy, eternamente agradecido con ustedes mis amados padres, han sembrado en mi lo necesario para salir adelante, surgir y seguir creciendo, soy una de sus semillas que está germinando y dando frutos dulces, gracias por sus bendiciones, los amo muchísimo.

Gracias a mis hermanitos, todos formando nuestro camino con los cimientos que nuestros padres han puesto. Mis compañeritos de vida, sigamos adelante. Bendiciones para todos ustedes, los amo muchísimo.

Eternamente agradecido con la mujer que Dios me puso en el camino, la que crece conmigo, está a mi lado apoyándome, que me brinda amor, comprensión y fuerza. A ti mi amor, Michelle, te agradezco mucho mucho y te amo mucho más. Sigamos cosechando logros conjuntos e individuales, seguir creciendo y haciendo bien las cosas, de la mano de Dios, crecer juntos y ser un gran ejemplo para Joseph y los que vengan.

De manera muy especial agradezco al M.Sc. Carlos Egas por dirigir de la mejor manera este proyecto de titulación, que Dios le colme de bendiciones y sabiduría para que siga impartiendo sus conocimientos y guiando a los futuros ingenieros de la república. Gracias por haberme brindado su amistad y sus conocimientos a lo largo de la carrera, siga adelante siendo como usted es, un excelente y auténtico profesional y ser humano.

A mis amigos: Danilo Pilacuán, Jorge Zurita, Roberto Llive, Hendry Jiménez, Edison Guamba, Willy Coronel, Bryan Moreano. A los que siguen con nosotros y a los que ya no.

Agradecimientos a muchas personas que a lo largo del camino han ido aportando su granito de arena en cada aspecto de la vida.

Agradecido eternamente

Diego René Velásquez García

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN.....	VII
ABSTRACT.....	VIII
1. INTRODUCCIÓN.....	1
1.1 OBJETIVO GENERAL	2
1.2 OBJETIVOS ESPECÍFICOS.....	2
1.3 ALCANCE.....	3
1.4 MARCO TEÓRICO	4
1.4.1. PANORAMA DEL COVID 19 EN ECUADOR	4
1.4.2. VISIÓN GENERAL DE LA IMPLEMENTACIÓN DE UN SISTEMA IoT DE CONTROL DE AISLAMIENTO.....	5
1.4.3. IoT.....	5
1.4.4. TECNOLOGÍA LPWAN, AMPLIA COBERTURA Y BAJO CONSUMO DE ENERGÍA.....	6
1.4.4.1. Sigfox.....	7
1.4.5. COMPARACIONES TÉCNICAS ENTRE TECNOLOGÍAS LPWAN.....	8
1.4.6. ESPECIFICACIONES SIGFOX	9
1.4.7. CAPA FÍSICA SIGFOX.....	10
1.4.8. PROTOCOLO MAC SIGFOX	11
1.4.9. ARQUITECTURA DE RED.....	12
1.4.10. SERVICIO DE BACKEND SIGFOX.....	13
1.4.11. ARDUINO.....	13
1.4.12. APLICACIONES WEB.....	13
1.4.13. SERVIDORES.....	13
1.4.13.1. Servidor LAMP	14
1.4.13.2. Apache	14
1.4.13.3. MySQL.....	14
1.4.13.4. PHP	14
1.4.14. COMPOSER	14
1.4.15. VISUAL STUDIO CODE.....	14

1.4.15.1.	Laravel.....	15
1.4.16.	AWS.....	15
1.4.16.1.	Instancias EC2.....	15
2.	METODOLOGÍA.....	16
2.1.	DESARROLLO	18
2.2.	DISEÑO DE HARDWARE	18
2.2.1.	ELEMENTOS COMPONENTES DEL NODO	18
2.2.1.1.	Arduino MEGA.....	18
2.2.1.2.	Thinextra Xkit Sigfox	20
2.2.1.3.	Módulo GPS	21
2.2.1.4.	Fingerprinter	22
2.2.1.5.	Conversor Serial - TTL.....	23
2.2.1.6.	Powerbank.....	24
2.2.2	IMPLEMENTACIÓN FÍSICA DEL NODO	25
2.3	DISEÑO DE SOFTWARE	27
2.3.1	CONFIGURACIÓN DE TRANSMISIÓN DEL NODO	28
2.3.2	CONFIGURACIÓN DE AMAZON WEB SERVICES	34
2.3.3	INSTALACIÓN DE HERRAMIENTAS LAMP	37
2.3.4	CODIFICACIÓN EN VISUAL STUDIO	37
2.3.4.1	Codificación de la Base de Datos	38
2.3.4.2	Comunicación con el Backend de Sigfox.....	39
2.3.4.3	Codificación del Dashboard.....	43
3.	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	47
3.1.	RESULTADOS	47
3.1.1.	PRUEBAS DE FUNCIONAMIENTO.....	48
3.2.	CONCLUSIONES	58
3.3.	RECOMENDACIONES	59
4.	REFERENCIAS BIBLIOGRÁFICAS.....	60
5.	ANEXOS	65

RESUMEN

El presente trabajo detalla la implementación de un nodo-sensor de comunicación IoT (*Internet of Things*) mediante las tecnologías Sigfox y Arduino en un sistema desarrollado para el control de cumplimiento de cuarentena debido a la pandemia del COVID-19.

Hasta la fecha, se registran rebrotes a nivel mundial y el descubrimiento de nuevas cepas del virus, obligando a la ciudadanía mundial a cambiar su estilo de vida, optando por cumplir encierros prolongados.

Este documento presenta una breve descripción de las comunicaciones inalámbricas LPWAN (*Low Power and Wide Area Networks*), características de la tecnología Sigfox, funcionamiento y especificaciones operativas. Así mismo, detalles de la tecnología Arduino, características de la placa de desarrollo MEGA utilizada para diseño de hardware y software.

El procesamiento de datos en la nube a través de servidores web, comunicación con el Backend Sigfox, levantamiento de instancia virtual de AWS (*Amazon Web Services*), instalación y configuración de servicios del paquete de características de servidores LAMP (Linux, Apache, MySQL, PHP), creación y manejo de base de datos.

La presentación de los datos se realiza en una página web desarrollada en PHP, mediante el framework Laravel en la plataforma Visual Studio Code.

Esta solución IoT, que permite realizar el seguimiento y monitoreo de pacientes aislados con COVID 19, se presenta como una alternativa funcional, de bajo costo, bajo consumo energético, que se logre implementar de una manera rápida y eficiente, con resultados similares o superiores a otras alternativas existentes como grilletes remotos, sistemas WiFi (*Wireless Fidelity*) o la red celular.

Palabras clave: Nodo-Sensor, IoT, Sigfox, Servidor Web, Página Web

ABSTRACT

This paper details the implementation of an IoT (Internet of Things) communication sensor-node using Sigfox and Arduino technologies in a system developed to control quarantine compliance due to the COVID-19 pandemic.

To date, outbreaks have been recorded worldwide and the discovery of new strains of the virus, forcing world citizens to change their lifestyle, opting for prolonged confinement.

This document presents a brief description of LPWAN (Low Power and Wide Area Networks) wireless communications, characteristics of Sigfox technology, operation and operational specifications. Likewise, details of Arduino technology, characteristics of the MEGA development board used for hardware and software design.

Data processing in the cloud through web servers, communication with the Sigfox Backend, creation of a virtual instance of AWS (Amazon Web Services), installation and configuration of services from the LAMP server features package (Linux, Apache, MySQL, PHP), database creation and management.

The presentation of the data is done through a web page developed in PHP, using the Laravel framework on the Visual Studio Code platform.

This IoT solution, which allows the tracking and monitoring of isolated patients with COVID 19, is presented as a functional, low-cost, low-energy alternative that can be implemented quickly and efficiently, with similar or superior results to other existing alternatives such as remote shackles, WiFi systems (Wireless Fidelity) or the cellular network.

Keywords: Node-Sensor, IoT, Sigfox, Web Server, Web Page

1. INTRODUCCIÓN

El mundo entero vive una nueva realidad debido al COVID 19, virus que provoca afecciones respiratorias severas como traumas, infecciones, desgaste, malformaciones, deterioros significativos del sistema traqueo-bronquio-respiratorio y en muchos casos, hasta la muerte. A más de dos años de los primeros contagios, la situación despliega alarmantes cifras en cuanto a índices de contagio y mortandad en todo el mundo. En Ecuador a fecha junio 2022, se tiene un registro oficial de 898 667 contagios positivos y un total aproximado de 35 725 defunciones con causal probable de COVID 19. La provincia de Pichincha acumula la mayor cantidad de contagios a nivel nacional con 181 530 casos. [1]

La capital Quito a la fecha de julio del 2021, se han reportado 161 310 casos de covid-19, según reportes emitidos por la Secretaría de Salud del Municipio. [2]

El COVID 19 tiene la particularidad de ser muy contagiosa, mostrar una rápida mutación, la aparición de nuevas cepas, su expansión y el número de contagios, muestran su alcance y letalidad. Por lo general el contagio se lleva a cabo cuando una persona infectada exhala millones de partículas que contienen el virus. Estas gotitas y partículas en el ambiente, pueden ser inhaladas por otras personas o depositarse sobre ojos, nariz o boca, y pueden contaminar las superficies cercanas expuestas. Quienes se encuentren a menos de 2 metros de distancia de una persona infectada tienen mayor probabilidad de infectarse, por lo que el uso de mascarillas es una medida primordial de prevención. [3]

Además de las mascarillas, el distanciamiento social es una de las medidas que el COE Nacional ha decretado en todo el territorio. La propagación del virus evidencia la falta de conciencia social y el bajo control de las personas enfermas al cumplir el aislamiento obligatorio, por lo que es necesario contar con un sistema que ayude a verificar que las personas estén en un lugar específico definido para realizar el aislamiento y evitar mayor número de contagios.

Este sistema prototipo brinda una solución referente al monitoreo del control de cumplimiento de aislamiento por medio del uso de la tecnología Sigfox, procedente de Francia, esta compañía es creada para ofrecer servicios de conectividad IoT, caracterizada por brindar una amplia cobertura, bajo consumo de potencia, bajo costo de desarrollo y operación. [4].

Mediante la implementación de un nodo sensor es posible verificar que la persona a controlar se encuentre en el lugar donde guarda aislamiento al realizar un reconocimiento dactilar periódico, asegurar y comprobar la identidad de la misma y su estado de

confinamiento. Su módulo GPS (*Global Positioning System*), determina la posición exacta del nodo al momento del reconocimiento dactilar. Los datos del nodo se transmiten por internet hacia Sigfox, los datos de fecha y hora se recuperan del registro de la trama en la nube, se almacenan toda la información en una base de datos, se enlazan con servidores web que permitan desplegar mediante un aplicativo IoT la información de cada usuario registrado, por medio de un dashboard en el dispositivo de usuario.

Este sistema propone brindar una alternativa innovadora para el control de cumplimiento de aislamiento, mediante el uso de tecnologías de red y comunicación inalámbricas de bajo consumo energético y largo alcance (LPWAN), el uso práctico y la aplicación del Internet de las Cosas, y así, evitar el contagio y disminuir la propagación del virus, e incorporar a la Escuela Politécnica Nacional en el grupo de instituciones que proponen soluciones tecnológicas para combatir y/o controlar la pandemia.

1.1 OBJETIVO GENERAL

Implementar un sistema prototipo para el control del cumplimiento de cuarentena de pacientes enfermos con COVID 19 en la ciudad de Quito.

1.2 OBJETIVOS ESPECÍFICOS

- Implementar el nodo sensor prototipo con tecnología Sigfox para la adquisición y transmisión de los datos de identificación, fecha, hora, geolocalización.
- Desarrollar el software para la configuración y funcionamiento del nodo sensor Sigfox compatible con Arduino.
- Desarrollar el software por medio de un servidor web que permita obtener, almacenar, procesar y presentar los datos en la PC de usuario.

1.3 ALCANCE

En el presente trabajo se realiza la implementación de un sistema-prototipo como una propuesta nueva basada en IoT, que permite monitorear la ubicación del paciente de COVID-19 en aislamiento, los datos de identificación por huella digital y geolocalización, registro de hora y fecha por intermedio de un nodo de comunicación con sensores integrados, a través de la red 0G de Sigfox en la ciudad de Quito.

Se inicia con la recopilación de datos relevantes como el estudio de las redes LPWAN, redes inalámbricas de telecomunicaciones de amplia cobertura y bajo consumo de potencia, diseñadas para comunicaciones de largo alcance con tasas de bits bajas entre varios objetos conectados.[5]

Se continua con el estudio de la tecnología Sigfox, la comunicación a través de la red 0G, composición del nodo de comunicación, configuración y transmisión de los sensores anexados, conexión del nodo a Internet, comunicación con el Backend Sigfox, configuración de los servicios y desarrollo del software aplicativo IoT para procesar la información mediante un servidor web.

Seguidamente, la elección de los componentes de hardware y software necesario para el desarrollo del proyecto, primero la configuración del nodo de comunicación que permitirá enviar datos sobre la red Sigfox a través de internet y segundo, el desarrollo de la aplicación IoT que realice la recuperación de los datos en la nube Sigfox, almacenamiento, disposición y presentación en la PC (*Personal Computer*) de usuario.

La construcción del módulo físico, nodo de comunicación, utilizando el Thinxtra Dev Xkit para la transmisión a Sigfox compatible con Arduino, que brinda la disponibilidad de hardware, software y un esquema estable de programación. Como placa central de procesamiento del nodo se utiliza la placa Arduino MEGA 2560, que permite anexar de manera apropiada el sensor digital biométrico, un módulo GPS y un conversor Serial-TTL.

Luego se realiza el establecimiento de la conexión y la operación del nodo capaz de transmitir por intermedio del Thinxtra Xkit los datos recopilados en la trama de información hacia la nube Sigfox, se dispone y asocia a través de la comunicación con el Backend por medio del servidor en la nube AWS, utilizando las herramientas AWS IoT de tal forma que recibe, almacena y visualiza la información en un dispositivo final. La figura 1.1 muestra un esquema de los elementos del sistema.



Figura 1.1. Esquema elementos en la transmisión de los datos

Por último, se realiza las pruebas pertinentes, donde se presentan varios procesos: desde el nodo-sensor el registro y escaneo de huellas digitales de los pacientes, transmisión de los datos correspondientes en el caso de coincidencia de huella, nivel de confianza de huella, datos de nombre del paciente, hora y geolocalización, la visualización de los datos en la nube Sigfox, almacenamiento y visualización de los mismos en una base de datos creada mediante el servidor de AWS en la nube, recuperación de datos almacenados y despliegue en un dashboard que permita visualizar en una página web desde el equipo del usuario monitor.

1.4 MARCO TEÓRICO

1.4.1. PANORAMA DEL COVID 19 EN ECUADOR

Ecuador se encuentra entre los países con mayor número de contagios de COVID 19, sus dos ciudades más importantes, Quito y Guayaquil, han mostrado un panorama alarmante en cuanto a los contagios y decesos, por otro lado, se ha evidenciado el manejo de la pandemia en el transcurso de los 2 años de expansión y mutación del virus en gran parte prometedor por parte de los registros de pacientes recuperados y las campañas masivas de vacunación.

Se puede evidenciar que están funcionando las diferentes medidas que han tomado las entidades gubernamentales para poner freno y contener los contagios en Quito, pero cabe recalcar que todavía se presentan problemas en cuanto al control de aglomeraciones y la falta de conciencia de cierta parte de la población en varios puntos de la ciudad. [6]

Desde cierres de fronteras hasta estados de excepción, se han venido viviendo en la ciudad capital y en todo el territorio nacional, siendo medidas que promuevan el distanciamiento

social y las aglomeraciones en general, tales como: suspensión de actividades laborales, educativas y mercantiles, dando paso a un ámbito tecnológico telemático, restricción para los servicios de transporte masivo, prohibición de actos masivos y eventos sociales. [7]

1.4.2. VISIÓN GENERAL DE LA IMPLEMENTACIÓN DE UN SISTEMA IoT DE CONTROL DE AISLAMIENTO

Una de las principales razones de la propagación del virus, se debe a que no se puede controlar que tanto las personas enfermas o las que han estado en contacto con una, cumplan con la cuarentena obligatoria, por lo que es necesario contar con un sistema que permita identificar y verificar que las personas estén en el lugar definido para realizar el correspondiente de aislamiento.

Los sistemas de seguimiento electrónico utilizados en la actualidad para comprobar el lugar donde se encuentra una persona, en su gran mayoría tienen un costo elevado por el valor del dispositivo y el costo de servicio a la operadora de telecomunicaciones lo cual impide utilizar esta tecnología si se desea implementar para un gran número de personas por el alto costo que representa.

Por lo que la implementación del sistema de control de aislamiento como aplicación IoT permite solucionar estas cuestiones, brindando además varias ventajas: Conexión directa de los nodos sensores a Internet, intercambio de información de forma rápida y práctica, ahorro energético, bajo nivel de inversión económica, sostenibilidad, entre otros grandes beneficios y posibilidades que se podrán adaptar a las necesidades de conectividad y transmisión requeridas.[9]

1.4.3. IoT

Internet of Things (IoT) se puede interpretar como un paradigma tecnológico en el que impera la dotación de conectividad a objetos alimentados, logrando el establecimiento, cambio y medición de valores, parámetros o aplicaciones para dar un tratamiento de datos inteligentes. [8]

En el siglo XX, con el auge y la expansión de las redes inalámbricas WLAN, tanto Wi-Fi como redes celulares, se evidencia un crecimiento e innovación sin precedentes del IoT en todos los campos de la tecnología, las redes y las telecomunicaciones.[9] [10].

Varias de las tecnologías aplicadas como las Redes de Sensores WSN (*Wireless Sensors Networks*), Redes de Sensores Vehiculares VSN (*Vehicular Sensors Networks*) o las

conexiones Máquina a Máquina M2M (*Machine to Machine*), son basadas en IoT, todas ellas abarcan un conjunto de fundamentos en protocolos, estándares, bandas, licencias, topologías, consorcios, redes, interfaces, tecnologías de comunicación, etc, que permiten que IoT se establezca como una destacada tecnología en desarrollo en la actualidad. [11]

1.4.4. TECNOLOGÍA LPWAN, AMPLIA COBERTURA Y BAJO CONSUMO DE ENERGÍA

El desarrollo de IoT a través de tecnologías inalámbricas, permiten la capacidad de interconectar nodos y dispositivos formando una red que se caracteriza por tener requisitos específicos de amplia cobertura, baja tasas de datos, bajo consumo de energía y por lo general bajo costo de implementación, siendo estas las características ideales de los ambientes LPWAN. [12]

En la figura 1.2 se observa una figura que compara varias tecnologías inalámbricas con respecto al ancho de banda y alcance.

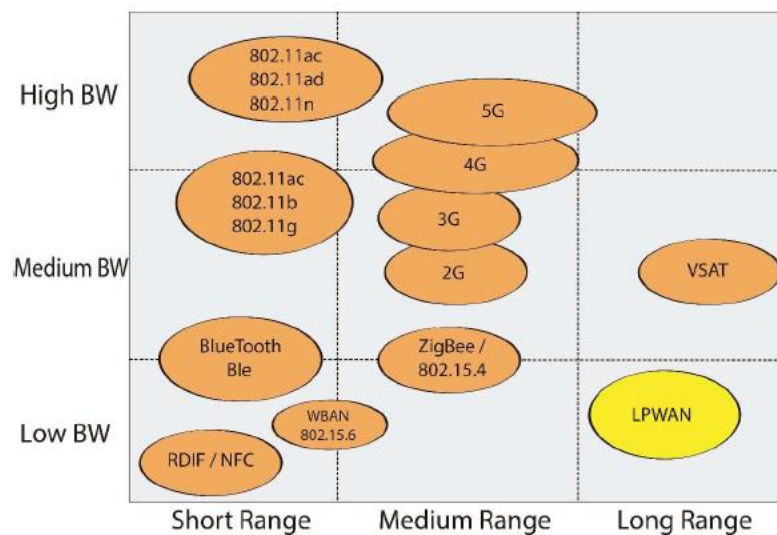


Figura 1.2. Comparación Alcance vs Ancho de Banda de tecnologías inalámbricas [13]

LPWAN destaca y cumple con las necesidades previstas en sistemas interconectados de sensores inteligentes al ser comparadas con otras tecnologías como Bluetooth o ZigBee, que presentan corto alcance o los sistemas celulares 4G/5G, que poseen un consumo de energía mayor y un alto costo debido al despliegue de su infraestructura, [14]

LPWAN abarca redes de amplio alcance con la posibilidad de conectar un gran número de dispositivos como nodos, transceivers compactos, sensores independientes, etc, con una carga baja de datos, que se transfieren con una tasa de consumo de energía baja. [15]

Los protocolos LPWAN permiten que los dispositivos conectados, intercambien pequeños mensajes a largas distancias, en el orden de decenas de kilómetros, con un nivel de consumo muy bajo de potencia, por debajo de 50 uW, consiguiendo así, una vida funcional de los dispositivos de varios años con el uso de baterías pequeñas.[16][17].

La figura 1.3 muestra un gráfico comparativo de características entre varias tecnologías inalámbricas.

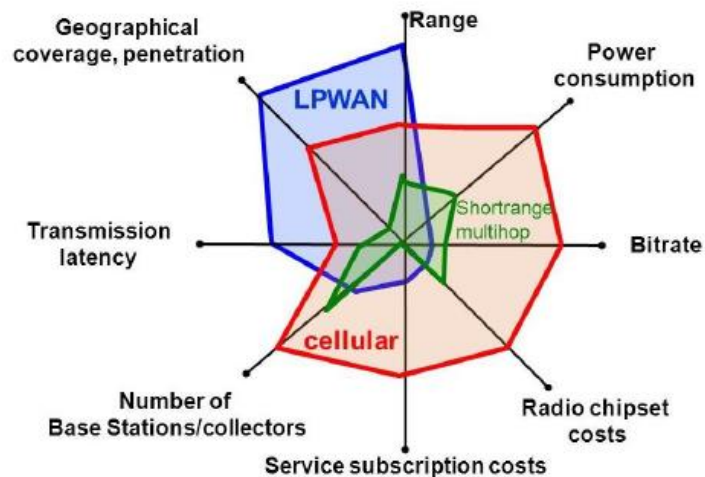


Figura 1.3. Comparativa de Tecnologías inalámbricas y características principales [13].

Las tecnologías dentro del ambiente LPWAN presentan las características idóneas para las aplicaciones IoT, envío de pequeñas tramas de información utilizando bajo ancho de banda, autonomía energética de bajo consumo, amplio alcance con un número reducido de estaciones base que ofrecen buena cobertura, adicional el bajo costo de implementación y suscripción al enlace. Generalmente las redes LPWAN poseen una topología tipo estrella, similar a Wi-Fi, donde cada punto final se conecta directamente a los puntos de acceso central adyacentes.[17] [18]

1.4.4.1. Sigfox

Sigfox es una empresa desarrolladora de tecnología francesa que implementa su visión de conectar todos los objetos del mundo físico al mundo digital mediante el desarrollo de tecnología LPWAN usando la banda ultra estrecha, en inglés UNB (*Ultra Narrow Band*).

Concretamente diseñada para IoT, Sigfox ofrece conectividad a través de su propia tecnología patentada, sin la necesidad de implementar infraestructuras de red específicas para cada aplicación y con versatilidad de fabricación de dispositivos.

La red de estaciones bases, repetidores y antenas de Sigfox son independientes de cualquier operador y además aprovechan infraestructuras existentes que son utilizadas para poder integrar la red y ampliar la cobertura.

Regido a los parámetros de los estándares ETSI, establece valores de potencia máxima de 14 dBm, que permite extender la vida de la batería y crear una celda más densa para la red Sigfox que abarcará una media de alcance de 40 Km en sectores rurales y de 5 Km en sectores urbano. [14]

1.4.5. COMPARACIONES TÉCNICAS ENTRE TECNOLOGÍAS LPWAN

En la Tabla 1.1 se observa una comparación entre las 3 tecnologías más utilizadas como son Sigfox, LoRa y NB IoT que permite comparar las diferentes características técnicas y operativas que manejan cada una de estas tecnologías LPWAN para satisfacer aplicaciones IoT.

Tabla 1.1. Comparativa entre características técnicas de Sigfox, LoRa y NB-IoT [14]

	Sigfox	LoRa	NB-IoT
Espectro	No licenciado (ISM)	No licenciado (ISM)	Licenciado
Frecuencia	EU 868 MHz US 915 MHz	EU 868 MHz US 915 MHz	Bandas de frecuencia LTE
Modulación	UL: UN DBPSK DL: GFSK	CSS	OFDMA (UL) SC-FDMA (DL)
Ancho de banda	100 Hz	250 kHz y 125 kHz	200 kHz
Máx. velocidad de datos	EU: 100 bps US: 600 bps	50 kbps	UL: 200 kbps DL: 250 kbps
Bidireccionalidad	Limitada/Half duplex	Si/Half duplex	Si/Half duplex
Máx. mensajes/día	UL: 140 DL: 4	Ilimitado	Ilimitado
Máx. payload	UL: 12 bytes DL: 8 bytes	243 bytes	1600 bytes
Rango	Urbano: 10 km Rural: 40 km	Urbano: 5 km Rural: 20 km	Urbano: 1 km Rural: 10 km
Inmunidad a interferencias	Muy alta	Muy alta	Baja
Handover	Los dispositivos finales no se suscriben a una única estación base	Los dispositivos finales no se suscriben a una única estación base	Los dispositivos finales se suscriben a una única estación base
Localización	Sí, RSSI	Sí, RDOA	No (bajo especificación)
Permite red privada	No	Sí	No
Estandarización	Sigfox en colaboración ETSI	LoRa-Alliance	3GPP

Sigfox destaca en términos de cobertura, duración de batería y eficiencia de coste, además de la característica de payload limitado que nos permiten cumplir con las necesidades de enlace, transmisión e implementación para el sistema prototipo. En la figura 1.4 se observa una comparativa entre varios parámetros de varias tecnologías IoT

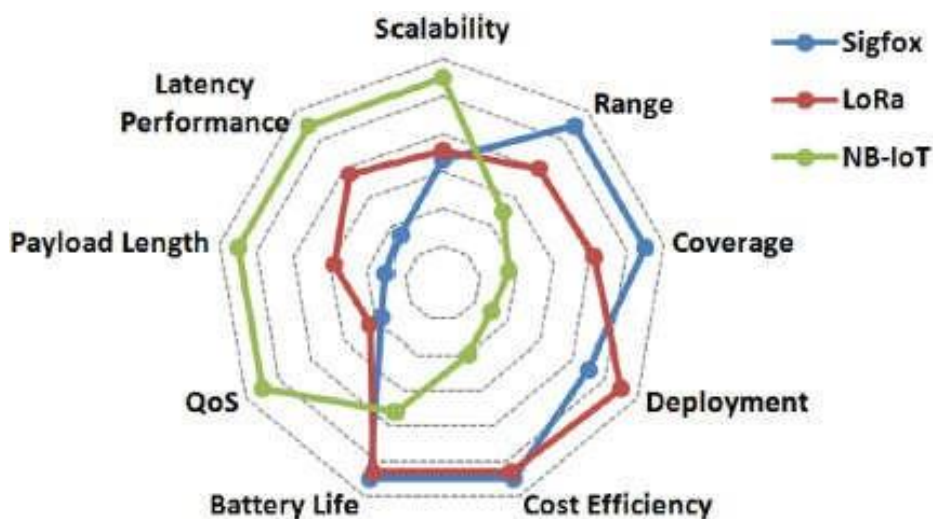


Figura 1.4. Ventajas de Sigfox, LoRa y NB-IoT en términos de factores IoT [20]

1.4.6. ESPECIFICACIONES SIGFOX

Sigfox se enfoca en brindar autonomía con bajo consumo de potencia, brindando un amplio tiempo de vida útil. Los dispositivos se conectan y funcionan sin necesidad de complejas configuraciones. Es posible precisar cifras acerca de su capacidad de expansión y alcance global, 20 millones de dispositivos activos, 79 millones de mensajes enviados por día, cobertura en más de 75 países y regiones, 6 millones de kilómetros cuadrados de cobertura con más de 1.4 billones de personas con disponibilidad de conexión. [19]

Propone y realiza el cálculo de tarifas de uso, conexión y despliegue en función a la cobertura y el número de dispositivos conectados y registrados en la red, la duración del contrato, la frecuencia de renovación y la cantidad de tráfico anual. [21]

1.4.7. CAPA FÍSICA SIGFOX

Sigfox funciona dentro de un espectro sin licencia, específicamente en la banda de radio industrial, científica y médica de espectro libre ISM (*Industrial, Scientific and Medical frequency band*) por sus siglas en inglés. Las frecuencias de uso para cada zona geográfica son 868 MHz en Europa, 915 MHz en América y 433 MHz en Asia.

Los dispositivos finales se conectan a las estaciones base mediante modulación BPSK (*Binary Phase Shift Keying*) en una portadora de ancho de banda ultra-estrecha. Mediante la modulación UNB, Sigfox utiliza un ancho de banda muy pequeño de 192 kHz, para el intercambio de mensajes. Estos mensajes enviados poseen una amplitud de 100 Hz y su tasa de transmisión de datos va de 100 a 600 bps.[14].

Para el enlace ascendente se utiliza la modulación D-BPSK, proporcionando una eficiencia espectral muy alta (1 Hz=1 bps). Permite una fácil implementación, al funcionar con UNB, experimenta niveles muy bajos de ruido, resultando a un consumo de energía muy bajo, costes bajos de los componentes y estaciones receptoras con un alto nivel de sensibilidad (a 100 bps, -142 dBm y a 600 bps, -134 dBm). [22].

Para el enlace descendente, Sigfox utiliza GFSK (*Gaussian Frequency Shift Keying*) en una banda de 600 Hz, donde el mensaje se transmite a 600 bps, con una potencia de 500 mW (27 dBm). [14]

Sigfox maneja un protocolo de transmisión que limita el número de mensajes en el enlace ascendente a 140 mensajes por día, o 6 mensajes por hora, con un payload de 12 bytes. Por otro lado, el número de mensajes del enlace de bajada se limita a 4 por día y su payload es de 8 bytes, garantizando la ocupación del ancho de banda del 1% máximo.

Es posible asegurar la confiabilidad de la comunicación de enlace ascendente a través de duplicación de transmisión y Acceso Múltiple por División de Tiempo y Frecuencia Aleatoria, por sus siglas en inglés RFTDMA (*Random Frequency and Time Division Multiple Access*). Desde el dispositivo final, cada uno de los mensajes se transmiten tres veces, de forma predeterminada, a través de diferentes canales de frecuencia aleatorio.[24]

Las estaciones base son configuradas con la capacidad de recibir mensajes en todos los canales simultáneamente, por lo que el dispositivo final podrá elegir cualquier canal aleatorio para transmisión, simplificando la parte de diseño del dispositivo. [14]

A continuación, la tabla 1.2 resume los diferentes parámetros operativos para los enlaces de transmisión Sigfox.

Tabla 1.2. Parámetros físicos operativos principales de la comunicación por Sigfox [21]

	Enlace de Subida (UL)	Enlace de Bajada (DL)
Límite de Payload (bytes)	12	8
<i>Throughput</i> (bps)	100	600
Máximo de mensajes por día	140	4
Esquema de Modulación	DBPSK	GFSK
Sensibilidad (dBm)	<14	<27

1.4.8. PROTOCOLO MAC SIGFOX



Figura 1.5. Trama MAC de uplink de Sigfox [21]

- El preámbulo indica el tipo de trama: uplink.
- La trama sync. maneja la sincronización de los dispositivos.
- El identificador de dispositivo final, reconoce el nodo, objeto o sensor que se ha conectado a la red Sigfox.
- El payload es la información que se transmite
- La parte de autenticación, identifica y reconoce los dispositivos.
- El verificador (FCS) comprobará la integridad de la información.[21]

1.4.9. ARQUITECTURA DE RED

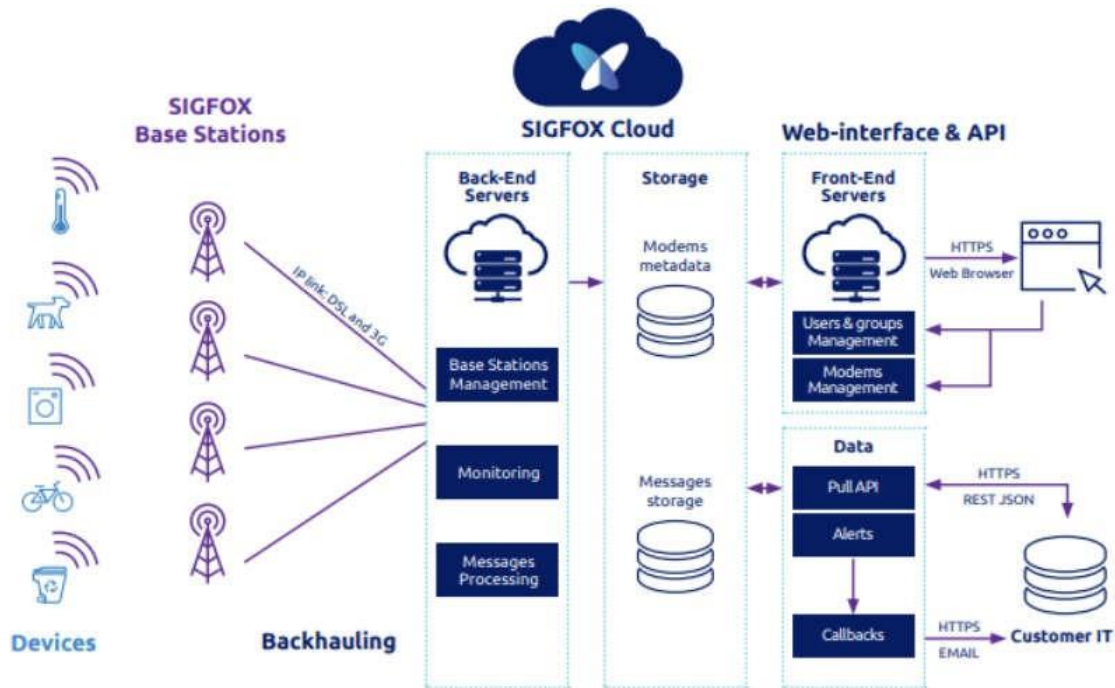


Figura 1.6. Arquitectura de red Sigfox [24]

La arquitectura de red en IoT se define como un modelo que abarca algunas tecnologías, tales como WSN, redes móviles, redes de actuadores, entre otras donde se encuentran objetos inteligentes conectados con dirección única. [25]

La red Sigfox utiliza una infraestructura de estaciones base, antenas y equipos propietarios, dispuestos en topología estrella con un funcionamiento detallado y simplificado: muchos objetos se comunican, emiten datos a las estaciones bases que se encuentran en permanente escucha del espectro, reciben los datos y los transmiten hacia sus sistemas de soporte, la nube Sigfox.

Por medio de UNB, es capaz de reducir las posibles colisiones que producen interferencias en la transmisión, brindando una conexión más resistente y robusta, por lo que se mantienen niveles de QoS altos, teniendo una gran escalabilidad que permite poseer múltiples objetos interconectados.

El tramo de la conexión entre las estaciones base y la nube Sigfox, se establece mediante tecnologías ya dispuestas como Ethernet, radioenlace satelital o red celular 3G/4G/5G. *Sigfox Cloud* (Servicio en la nube de Sigfox), se maneja como proveedor hacia los servidores para el uso y almacenamiento de los datos de sus clientes, permitiendo acceder directamente a sus datos, características y mensajes. [14]

1.4.10. SERVICIO DE BACKEND SIGFOX

En la nube de Sigfox se presenta el servicio de Backend de Sigfox que proporciona una interfaz para las aplicaciones web que administran los dispositivos, visualiza los mensajes transmitidos, configura los datos almacenados y obtenidos por los sensores.

En la comunicación se presentan dos extremos: el Backend de Sigfox y la base de datos que recibe, procesa y almacena la información del dispositivo.[26] [27]

1.4.11. ARDUINO

Es una plataforma de hardware libre. Posee características plenas para el desarrollo e implementación de proyectos de ingeniería de alto y bajo nivel de complejidad e impacto. Versátil, de sencilla funcionalidad y bajo costo de implementación, se encuentra al alcance y disponibilidad de todo tipo de usuario.[28]

Arduino proporciona su entorno de desarrollo Integrado *Arduino IDE*, una interfaz de desarrollo para las tarjetas y componentes Arduino. Caracterizado por su facilidad de estudio y comprensión, utiliza comandos inspirados en lenguaje de programación de mediana y baja complejidad. [29]

1.4.12. APLICACIONES WEB

Son herramientas que permiten realizar operaciones desde un PC a través de Internet logrando una optimización al reducir el tiempo en cada actividad. Son aplicaciones del tipo Cliente/Servidor. Utilizan el formato estándar HTML (*HyperText Markup Language* o Lenguaje de Hipertextos) para ejecutar las peticiones del usuario y brindar acceso simultaneo a sus operaciones mediante una combinación de procesos y comunicaciones internas con las bases de datos. [30]

1.4.13. SERVIDORES

Los servidores son ordenadores super potenciados que procesan, almacenan, transfieren información requerida. Son los destinados a manejar las respuestas de las solicitudes que hacen los clientes, orientan su operación sobre el protocolo HTTP (*Hypertext Transfer Protocol* o Protocolo de transferencia de hipertextos). Este protocolo pertenece a la capa de aplicación de TCP/IP, define la forma como los clientes web solicitan objetos a un servidor y cómo los servidores los transfieren a los clientes. [31]

1.4.13.1. Servidor LAMPP

Este servidor se define como la constitución de 4 servicios y/o tecnologías: Linux, Apache, MySQL y PHP/Perl, cuya utilidad principal es la creación de aplicaciones y páginas web. LAMPP brinda versatilidad, simplicidad de uso y configuración y estabilidad en el montaje de las aplicaciones web desarrolladas.[32]

1.4.13.2. Apache

Apache es un servidor web, actualmente usado en más de la mitad de todos los servidores web activos del mundo. Brinda altos rangos de flexibilidad al igual que de potencia, permite almacenar y descartar las páginas web ejecutadas con PHP.[32]

1.4.13.3. MySQL

MySQL es el servidor de bases de datos relacionales de código abierto, brinda rapidez, seguridad y simplicidad de uso. Accesible para la gran mayoría de lenguajes de programación. Es el encargado de crear bases de datos para aplicaciones web. Con la capacidad plena de enlazar, recuperar, almacenar y replicar gran cantidad de datos. [32]

1.4.13.4. PHP

Es el lenguaje de programación más utilizado en el desarrollo de páginas web, embebido en páginas HTML, posee similar sintaxis a lenguajes como Java o C++, de interpretación en alto nivel, se ejecuta directamente en el servidor web, presenta compatibilidad con varios sistemas operativos (Windows, Linux, Unix, MAC OS, etc), servidores web (Apache, IIS, Netscape, etc) y bases de datos (PostgreSQL, dBase, Informix, MySQL), etc. [32]

1.4.14. COMPOSER

Es un sistema de gestión de dependencias para el desarrollo en PHP, brinda el acceso a un repositorio de dependencias, formatos estandarizados de administración de librerías y paquetes de software disponibles y actualizables para cada proyecto. Proporciona versatilidad, la mayoría de dependencias base en el desarrollo se encuentran preinstaladas y disponibles para ajuste y configuración según los requerimientos y necesidades. [33]

1.4.15. VISUAL STUDIO CODE

Visual Studio es un software con un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Como editor de código su IDE es compatible con varios

sistemas operativos (Windows, MAC, Linux, etc), habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Brinda facilidad en la depuración del proyecto, un entorno intuitivo y versatilidad de desarrollo. [34]

1.4.15.1. Laravel

Laravel es un framework para codificación y desarrollo que destaca por su sencillez, dinamismo y constante evolución. Una herramienta con sintaxis elegante y expresiva, compuesta por dos utilitarios ampliamente explotados: Full Stack Framework y Laravel API Backend. Brinda servicios de plantillas BLADE, autenticación, autorización, consola Artisan, bases de datos, Eloquent, entre otras funcionalidades. [35]

1.4.16. AWS

Amazon AWS es una plataforma web que ofrece una variedad de servicios y productos informáticos en línea, permitiendo el desarrollo, implementación, ejecución y ajuste de la escala de las aplicaciones en la nube, presenta varios servicios como sistemas de almacenamiento en la nube, programas de monitorización, seguridad e identidad, etc. Establece una arquitectura para el desarrollo de proyectos en la nube.[36]

1.4.16.1. Instancias EC2

Amazon EC2 (*Amazon Elastic Compute Cloud*) es un servicio web que pone a disposición capacidad informática segura en la nube y de tamaño modificable. Diseñado para facilitar el manejo de la informática en la nube a escala web de manera ágil y sencilla. EC2 ofrece un entorno virtual que permite utilizar interfaces de servicios web para iniciar instancias con diferentes sistemas operativos, cargarlas con su galería de aplicaciones, administrar permisos de accesibilidad y ejecutarlas mediante los sistemas disponibles a gusto del desarrollador [37].

2. METODOLOGÍA

Este trabajo trata una investigación Teórica - Aplicada, fundamenta teorías científicas aplicadas a una realidad concreta. Parte de un marco teórico que formula nuevas teorías, incrementa los conocimientos y busca su aplicación, destacando el interés del investigador hacia las consecuencias prácticas. [38].

Basado en el método científico, direccionado a la parte analítica experimental y la metodología ágil Kanban, la cual busca gestionar de manera generalizada cómo se completan las tareas. El tablero Kanban permite determinar los procesos que se están realizando, los próximos por realizar y los que ya se realizaron, actualizándose en varias etapas. [39]

En la Tabla 2.1 se presenta el tablero Kanban con las actividades para el desarrollo del trabajo.

Tabla 2.1. Tabla Kanban con las tareas establecidas para desarrollo

PENDIENTES	EN PROGRESO	REALIZADO
Configuración del código de transmisión y armar la trama de payload	Instalar y configurar el nodo Sensor	Estudio de características, manejo y configuración de implementos y servicios de Hardware y Software
Configurar el servidor		
Configurar LAMP		
Codificación base de datos		
Configuración conexión Backend		
Codificar y configurar la recuperación de datos almacenados		
Diseñar y configurar el servicio web		
Diseño y codificación del Dashboard		
Realizar pruebas del nodo sensor y aplicación web		

La figura 2.1 muestra un flujograma con las diferentes fases y tareas metodológicas

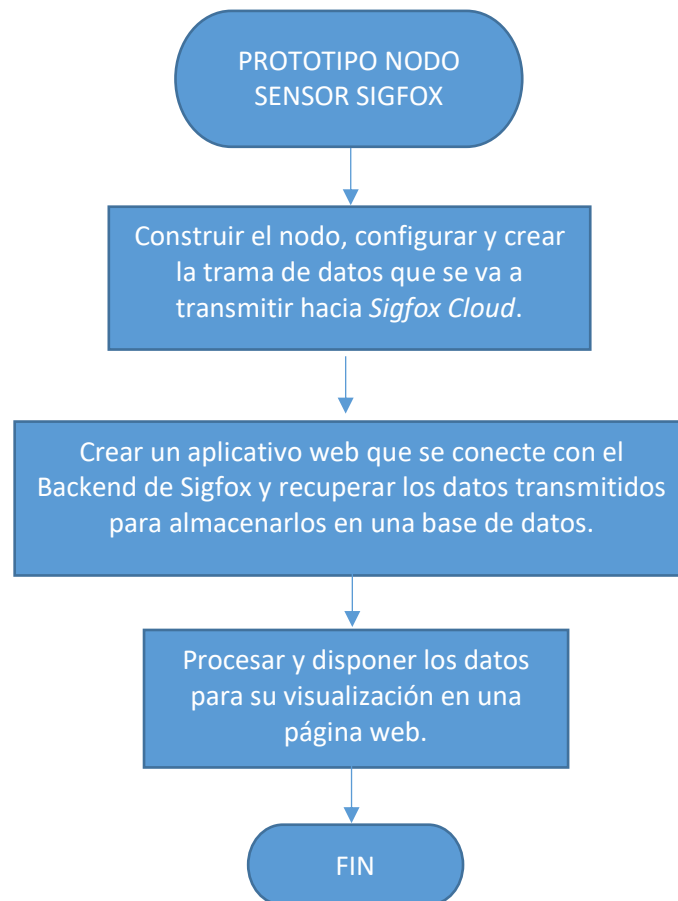


Figura 2.1. Flujograma de la metodología

2.1. DESARROLLO

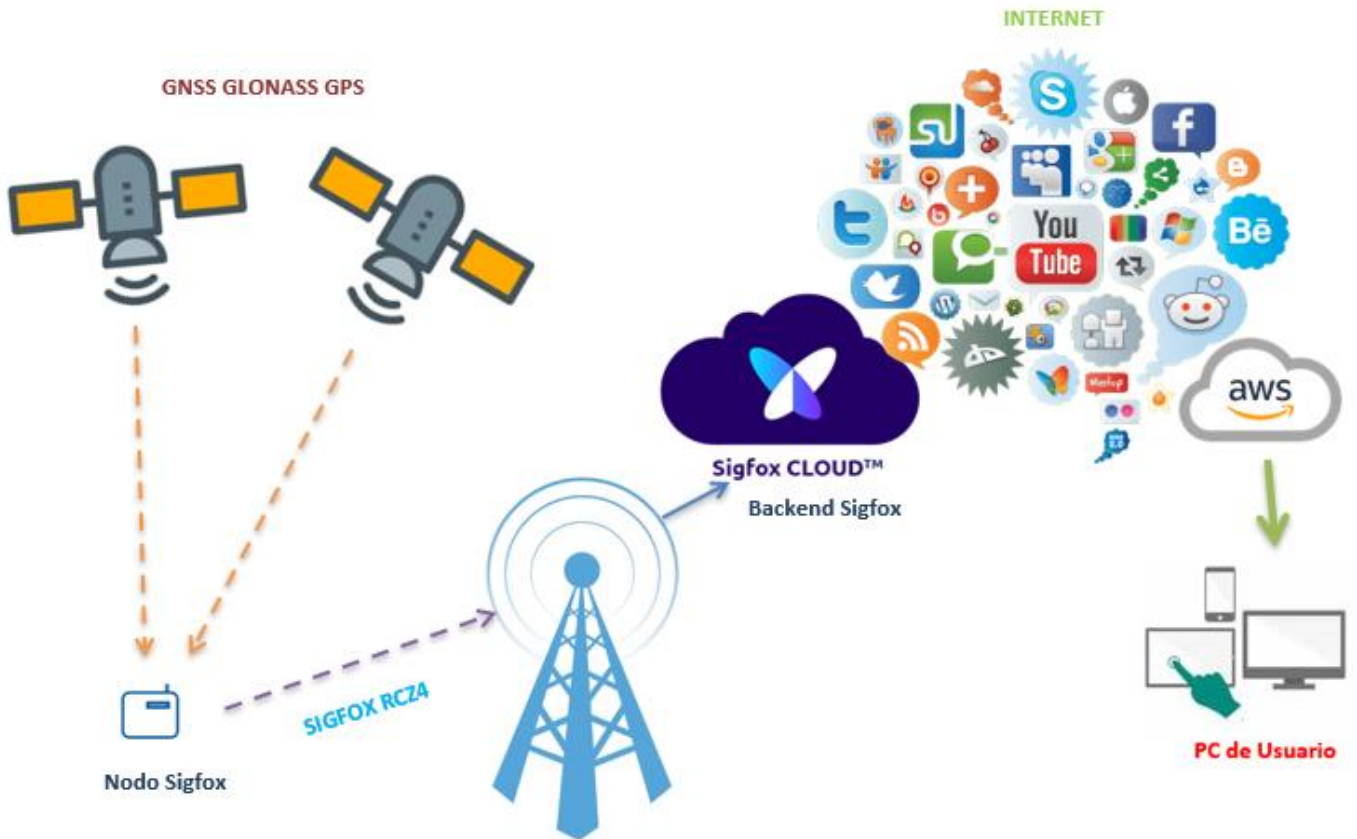


Figura 2.2. Esquema del sistema

2.2. DISEÑO DE HARDWARE

2.2.1. ELEMENTOS COMPONENTES DEL NODO

2.2.1.1. Arduino MEGA

La tarjeta Arduino MEGA 2560 es una plataforma de hardware libre que funciona con el micro-controlador ATmega2560, consta con 54 puertos de salida y entrada digitales y es capaz de utilizar 15 de sus puertos como salida PWM (Modulación de ancho de Pulso), posee 16 pines de entradas analógicas, 4 puertos seriales por hardware UART (*Universal Asynchronous Receiver / Transmitter*), un cristal con frecuencia de 16 MHz programable desde un PC a través de conexión USB.[40].

En la figura 2.3 se observa el Arduino MEGA 2560 con todos sus componentes y en la tabla 2.2 sus características técnicas operativas.

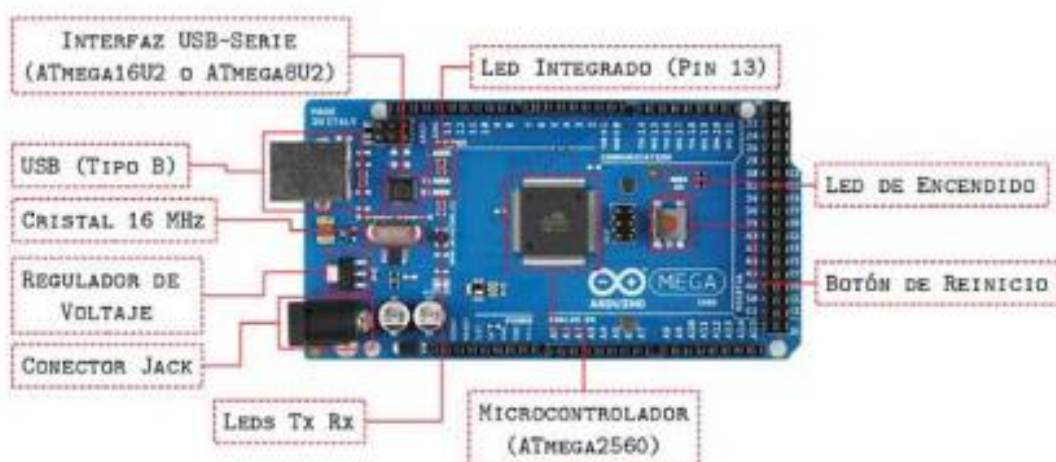


Figura 2.3. Componentes Arduino MEGA 2560 [40]

Tabla 2.2. Características técnicas Arduino MEGA 2560 [40]

Característica	Descripción
Microcontrolador	ATmega 2560
Voltaje de operación	5 V
Voltaje de entrada recomendado	7-12 V
Voltaje de entrada mínimo y máximo	6-20 V
Pines E/S digital	54 (15 como salidas PWM)
Pines de entrada analógica	16
Corriente CC por pin E/S	20 mA
Corriente CC para pin de 3.3 V	50 mA
Memoria Flash	256 KB (8 KB para bootloader)
SRAM	8 KB
EEPROM	4 KB
Frecuencia de reloj	16 MHz
Dimensiones	101.52 mm x 53.3 mm

El diagrama de pines y sus especificaciones se encuentra en el Anexo I.

2.2.1.2. Thinxtra Xkit Sigfox

El kit de desarrollo Thinxtra de Sigfox proporciona un completo conjunto de accesorios y funciones que permite configurar una aplicación o solución IoT. El modelo Xkit RCZ4 (*Radio Configuration Zone 4*), permite utilizar y configurar el plan de frecuencias que va entre 862 a 928 MHz, asignado para la zona 4 correspondiente a Australia, Nueva Zelanda y gran parte de Latinoamérica, incluido Ecuador. [41]

En la figura 2.4 se observa el Devkit con todos sus componentes desde caja.



Figura 2.4. Thinxtra Sigfox Dev Kit con todos sus elementos [41]

El Dev XKit contiene una placa Arduino UNO R3 que en el presente caso se descarta para utilizar el Arduino MEGA 2560, la placa Sigfox Thinxtra que opera con el módulo Wisol para transmisión y recepción en radiofrecuencia e interconexión con varios sensores integrados. El XKit utilizan la mayoría de los pines de entrada y salida de propósito general de las placas Arduino compatibles. [41]

En la tabla 2.3 se muestran sus características técnicas operativas.

Tabla 2.3. Características técnicas Thinxtra Devkit [41]

Característica	Descripción
Número de modelo	XKIT1-4
Categoría	Placa de desarrollo, evaluación de RF
Serie / Fabricante	Sigfox / Thinxtra Solutions Limited
Tipo	Transceiver
Configuración	RCZ4 (Sudamérica - Asia)
Frecuencia	920 MHz
Fuente de energía	Puerto USB o Batería 9V PP3 x1 (No incluida)
Protocolo Wireless	Módulo Wisol WSSF10 – Enlaces Sigfox UL / DL
Compatibilidad	Arduino, Raspberry Pi, PC, STMicro Núcleo
Dimensión largo	65.34 mm
Dimensión ancho	53.34 mm
Dimensión alto	20.7 mm
Componentes internos	Acelerómetro de 3 ejes
	Sensor digital de temperatura y presión Reed switch, Sensores de luz
Extras	Placa Clon Arduino UNO R3
	Cable USB, holder para batería Antena giratoria de 85 cm

El diagrama de pines y sus especificaciones se encuentra en el Anexo II.

2.2.1.3. Módulo GPS

Los módulos GPS son utilizados para obtener las coordenadas geográficas de un objeto; posee un rango de exactitud impresionante, aunque necesita de lugares abiertos para obtener una óptima conexión con los satélites geoestacionarios. [42]

Los resultados de la posición se envían mediante texto con codificación estándar NMEA (*National Marine Electronics Association*). Si el módulo aún no ha adquirido señal de conexión, la codificación de coordenadas presenta una serie de ceros "0". Tiene un promedio de tiempo de conexión con los satélites de 35 segundos o menos dependiendo el entorno y la altura medible. [43].

En la figura 2.5 se observa el Módulo GPS NEO 6M y en la tabla 2.4 sus características técnicas operativas.



Figura 2.5. Modulo GPS NEO 6M [43]

Tabla 2.4. Características técnicas Módulo GPS NEO 6M [43]

Característica	Descripción
Voltaje de alimentación	3-5 VDC
Default Baud rate	9600 bps
Frecuencia de refresco	5 Hz
Interfaz	Serial UART 5 V
EEPROM	Limitada solo para configuración
Frecuencia de reloj	16 MHz
Sistema de coordenadas	WGS-84
Sensibilidad de captura	-148 dBm
Sensibilidad de rastreo	-161 dBm
Máxima velocidad	515 m/s
Exactitud	1us
Dimensiones Antena Cerámica	25 mm x 25 mm
Dimensiones Módulo	25 mm x 35 mm
Peso	18 g

2.2.1.4. Fingerprinter

Los sensores de huellas dactilares permiten leer, guardar e identificar las huellas dactilares de los usuarios. El módulo R307 es un lector de huellas dactilares con interfaz UART/USB2.0 de muy bajo consumo de potencia, se utiliza para diversos propósitos de seguridad e identificación. Consta de un sensor óptico biométrico, un procesador interno de alta velocidad, sistema de coincidencia, procesador de imágenes y plantillas de huellas dactilares de alto rendimiento. [44]

En la figura 2.6 se observa el Sensor Biométrico R307 y en la tabla 2.5 sus características técnicas operativas.



Figura 2.6. Fingerprinter R307 [44]

Tabla 2.5. Características técnicas Fingerprinter R307 [44]

Característica	Descripción
Sensor Tipo	Óptico
Voltaje de alimentación	4.2 – 6 VDC
Corriente de funcionamiento	50 mA
Default Baud rate	9600 bps RS232
Resolución	500 DPI
Interfaz	Serial UART TTL
Tamaño de archivo de caracteres	256 bytes
Frecuencia de reloj	16 MHz
Velocidad de verificación	0.2 s
Velocidad de escaneo	0.3 s
Vida útil del sensor	100 millones de veces
Capacidad máxima de almacenamiento	1000 huellas
Luz de fondo	Luz Azul
Dimensiones	52 mm x 20 mm x 22 mm
Superficie de captura	15 mm x 11 mm

2.2.1.5. Conversor Serial - TTL

Dispositivo electrónico que se emplea para conectar dispositivos seriales a través de puerto USB. Brinda una adaptación estable, confiable y compatible de interfaces USB a Serial RS232 / TTL. Al conectarse con un ordenador, esta conexión será reconocida como un Puerto Virtual COM (VCP) que permite la comunicación con dispositivos externos y la utilización de sistemas implementados con estas interfaces. [45]

En la figura 2.7 se observa el Módulo Conversor Serial TTL-USB y en la tabla 2.6 sus características técnicas operativas.

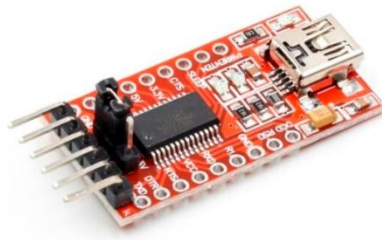


Figura 2.7. Módulo HW 417 A [45]

Tabla 2.6. Características técnicas Módulo HW 417 A [45]

Característica	Descripción
Microcontrolador	FT232RL de FTDI
Voltaje de alimentación	5 V puerto USB
Consumo de corriente	15 mA
Señales seriales	TX, RX, CTS, DTR, DSR, DCD, RI
Voltaje In/Out	5 V o 3.3 V (mediante jumper)
Conector	Mini USB V3
Dimensiones	17 mm x 33 mm

2.2.1.6. Powerbank

Arreglo de baterías de Litio que funciona como fuente externa que brinda una corriente y un voltaje de suministro constante que permite la carga de dispositivos conectados, y del mismo modo, brindar la energía para el funcionamiento continuo del nodo sensor. En la figura 2.8 se observa el Powerbank de 10000mA.



Figura 2.8. Powerbank

Desde los Datasheet, se recopila los datos de potencia y consumo de corriente de los componentes del nodo sensor, como se observa en la Tabla 2.7, se muestra este detalle y se establece el valor estándar de consumo de corriente por hora mínimo del Powerbank a 6000 mAh, que proporciona un tiempo práctico para su funcionamiento normal y autonomía de energía de 24 horas continuas.

Tabla 2.7. Consumo de corriente de los componentes y Total acumulado.

Cantidad	Descripción	Consumo de corriente mAh
1	Thinextra Dev Kit	51 mAh
1	Arduino Mega	72 – 93 mAh
1	Módulo GPS Neo 6M	45 mAh
1	Conversor Serial TTL	50 mAh
		Total (máx) = 240 mAh
		Consumo Total al día = 5760 mA

2.2.2 IMPLEMENTACIÓN FÍSICA DEL NODO

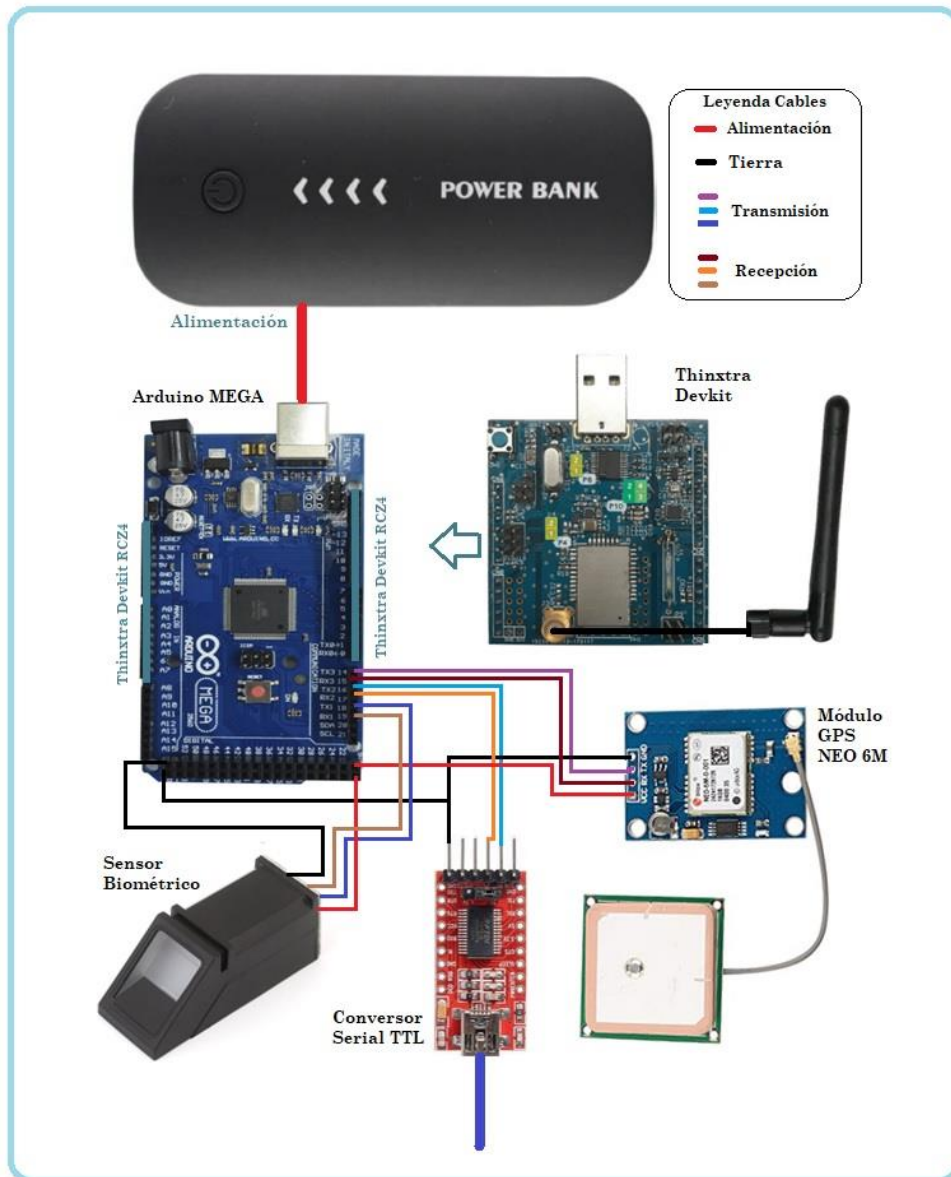


Figura 2.9. Esquema General Nodo Sensor

Mediante el Devkit Thinxtra Sigfox se realiza la transmisión de la trama de datos, al ser un dispositivo compatible con Arduino se utiliza sin problema la placa Arduino MEGA 2560, como se muestra en la figura 2.10. Funciona como placa central de procesamiento del nodo, permite anexar de manera apropiada el sensor digital biométrico, al módulo GPS y el conversor Serial – TTL.

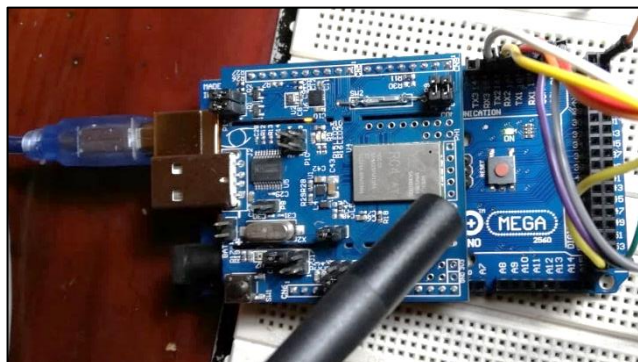


Figura 2.10. Devkit Thinxtra conectado con Arduino MEGA

El Thinxtra DevKit opera mediante el módulo transceiver WISOL/WSSFM10R4AT, el cual permite conectar la placa Arduino con internet directamente con los servidores de *Sigfox Cloud*. [32]

El primer paso es el registro del nodo Thinxtra DevKit en el portal Sigfox por medio de las credenciales de fábrica del Dev Kit. Estas credenciales, tales como el Sigfox ID y el número PAC son generadas exclusivamente para cada Devkit y vienen etiquetadas dentro y fuera de su caja contenedora.

Dicho portal permite crear una cuenta personal para el registro de placas y dispositivos Sigfox de una manera rápida y sencilla, genera una lista de dispositivos conjunto con los parámetros de información de dispositivo, tipo, nombre, estado de conexión, mensajes transmitidos, entre otros. [46].

La figura 2.11 muestra el registro del nodo en la plataforma, su ID del dispositivo, el estado de operación y demás información asociada.



Figura 2.11. Registro del Nodo en *Sigfox Cloud*, Pagina web Sigfox.

El DevKit Thinxtra cuenta con varios manuales de usuarios para configuración, tipos de conexión, tipos de alimentación, un código plantilla preliminar para transmisión, varios manuales operativos, funcionales, electrónicos, que permiten iniciar correctamente la utilización del DevKit afín a los requerimientos necesarios. En la figura 2.12, se muestra el nodo sensor prototipo Sigfox - Arduino con todos sus componentes conectados.

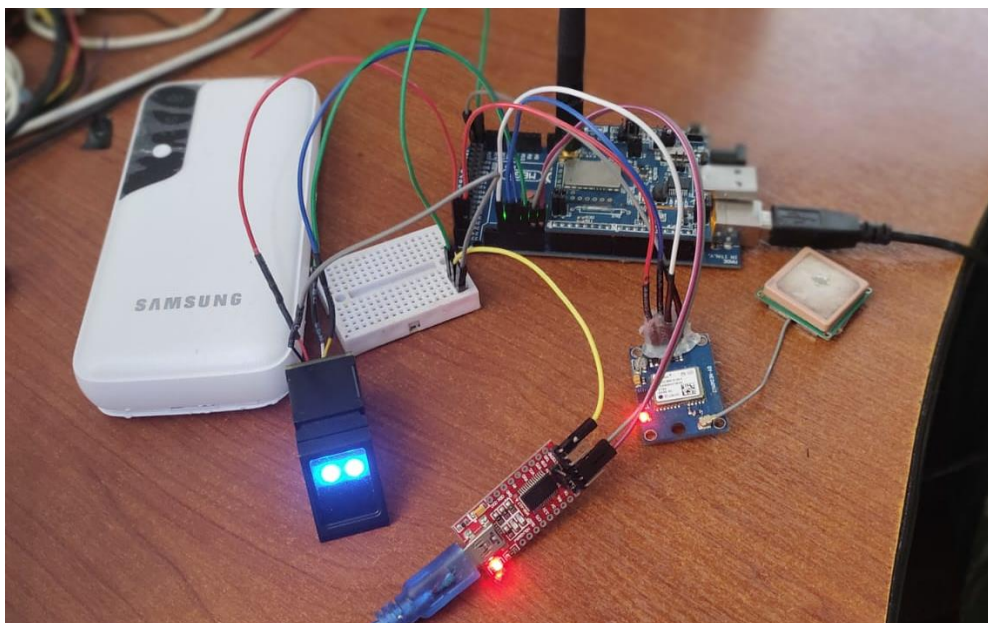


Figura 2.12. Prototipo del nodo IoT con sus componentes conectados

2.3 DISEÑO DE SOFTWARE

El Diseño de Software presenta la configuración del nodo de comunicación que permite la recolección de datos provenientes de los sensores Biométrico y el GPS, comunicación entre la placa Arduino MEGA 2560 con el Thinxtra Dev Kit para el procesamiento de datos, el levantamiento del enlace serial que permita realizar el ingreso de huellas y de usuarios en el nodo sensor, generación de la trama de transmisión que envía los datos por internet a través de la red Sigfox, el registro del nodo y la comunicación con el Backend en la plataforma Sigfox.

Se prosigue con la creación del servidor en línea mediante una instancia EC2 de AWS, la creación de la base de datos para almacenamiento de los datos recuperados del Backend Sigfox, la configuración de las plataformas de *Cloud Computing Sigfox* y *AWS*, el tratamiento, recuperación y presentación de datos por medio de una aplicación IoT, elaboración y diseño del dashboard que permita visualizar los datos transmitidos.

En la Tabla 2.8 se presenta el tablero Kanban actualizado con las actividades del trabajo.

Tabla 2.8. Actualización Tabla Kanban con el detalle de las diferentes tareas.

PENDIENTES	EN PROGRESO	REALIZADO
Diseñar y configurar el servicio web	Configuración del código de transmisión y armar la trama de payload	Estudio de características, manejo y configuración de implementos y servicios de Hardware y Software
Diseño y codificación del Dashboard	Configurar el servidor	Instalar y configurar el nodo Sensor
Realizar pruebas del nodo sensor y aplicación web	Configurar LAMP	
	Codificación base de datos	
	Configuración conexión Backend	
	Codificar y configurar la recuperación de datos almacenados	

2.3.1 CONFIGURACIÓN DE TRANSMISIÓN DEL NODO

Sigfox diseñó el Thinxtra Devkit con la capacidad de trabajar con el IDE Arduino y utilizar su gran catálogo de librerías, los diferentes dispositivos, implementos y complementos compatibles presentes en el mercado.

Dentro de la plantilla base de transmisión del Thinxtra Devkit, se codifica la comunicación de los sensores, la placa de desarrollo y el módulo Wisol del DevKit de transmisión Sigfox. se incluye las librerías y los parámetros de configuración de los sensores y las diferentes comunicaciones de los enlaces seriales UART de la placa MEGA, módulo GPS y Fingerprint, conjunto con el conversor Serial TTL para la visualización en la PC de usuario en Monitor Serie.

En el código 2.1 se incluye las librerías, definen las variables y se establecen los parámetros de configuración de los componentes del nodo sensor.

```

1. #include <Adafruit_Fingerprint.h>
2. #include <EEPROM.h>
3. #include <WISOL.h>
4. #include <Wire.h>
5. #include <avr/wdt.h>
6. #include <TinyGPS++.h>
7. #define DS3231_I2C_ADDRESS 0x68
8. TinyGPSPlus gps;
9. // INSTANCIANDO OBJETOS
10. Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial2);
11. // DECLARACION VARIABLES Y FUNCIONES
12. uint8_t numID = 1;
13. bool gravar=false;
14. uint8_t modoGravacaoID(uint8_t IDgravar);
15. Isigfox *Isigfox = new WISOL();
16. int watchdogCounter;
17. uint8_t buttonCounter;
18. uint8_t PublicModeSF;
19. uint8_t stateLED;
20. uint8_t ledCounter;
21. const uint8_t buttonPin = A1;
22. const int redLED = 6;
23. typedef union{
24.     float number;
25.     uint8_t bytes[4];
26. } FLOATUNION_t;
27. typedef union{
28.     uint16_t number;
29.     uint8_t bytes[2];
30. } UINT16_t;
31. typedef union{
32.     int16_t number;
33.     uint8_t bytes[2];
34. } INT16_t;
35. byte ingresosExitosos[6];
36. int posicionMemoria=0;
37. int ultimoId=0;
38. void setup() {
39.     Serial3.begin(9600);
40.     Serial.begin(9600);
41.     Serial1.begin(9600);
42.     finger.begin(57600);

```

Código 2.1. Librerías, variables y parámetros de configuración inicial de transmisión.

El código 2.2 permite la configuración de la EEPROM del sensor biométrico, se define un máximo de 200 huellas a registrar en el nodo sensor y la verificación de la conexión del sensor Fingerprint para adquisición de huellas dactilares.

```

1. EEPROM.get(posicionMemoria, ultimoId);
2.   if(ultimoId>200)
3.   {   ultimoId=1;}
4.   ingresosExitosos[0]=false;
5.   ingresosExitosos[1]=false;
6.   ingresosExitosos[2]=false;
7.   ingresosExitosos[3]=false;
8.   ingresosExitosos[4]=false;
9.   ingresosExitosos[5]=false;
10.  ingresosExitosos[6]=false;
11.  if(finger.verifyPassword()){
12.    Serial3.println("Sensor biometrico encontrado!");
13.  } else {
14.    Serial3.println("Sensor biometrico no encontrado! Verifique la conexion y
reinicie el sistema");
15.    while(true) {
16.      delay(1); }}

```

Código 2.2. Detección del Fingerprinter y definición de memoria para huellas dactilares.

El Código 2.3 muestra la configuración para el ingreso de la huella. Cada sección de configuración de la huella se codifica con breaks para los casos en los que exista fallo o error en la captura de la huella por el sensor biométrico.

```

1. uint8_t modoGravacaoID(uint8_t IDgravar) {
2.   int p = -1;
3.   Serial3.print("Esperando una huella para guardar usuario #");
Serial3.println(IDgravar);
4.   delay(2000);
5.   while (p != FINGERPRINT_OK) {
6.     p = finger.getImage();
7.     switch (p) {   case FINGERPRINT_OK:
8.       Serial3.println("Lectura concluida"); break;
9.     case FINGERPRINT_NOFINGER:
10.      Serial3.println(".");
11.      delay(200); break;
12.     case FINGERPRINT_PACKETRECEIVEERR:
13.      Serial3.println("Error de comunicacion"); break;
14.     case FINGERPRINT_IMAGEFAIL:
15.      Serial3.println("Error de lectura"); break;
16.     default:
17.      Serial3.println("Error desconocido"); break; } }

```

Código 2.3. Ingreso de huella dactilar y codificación de excepciones.

En el Código 2.4 se observa el proceso para los casos exitosos de lectura, pide nuevamente al usuario el ingreso de la huella para concluir el almacenado. Cada huella dactilar almacenada se asocia con un número de identificación. Se crea un modelo utilizando la ID y la huella dactilar, para establecer un valor entero de confianza de huella en cada ingreso de huella coincidente.

```

48. Serial3.println("Remueva el dedo");
49. delay(2000);
50. p = 0;
51. while (p != FINGERPRINT_NOFINGER) {
52.     p = finger.getImage();}
53. Serial3.print("ID "); Serial3.println(IDgravar);
54. p = -1;
55. Serial3.println("Coloque el mismo dedo nuevamente");
56. while (p != FINGERPRINT_OK) {
57.     p = finger.getImage();
58.     switch (p) {
59.         case FINGERPRINT_OK:
60.             Serial3.println("Lectura concluída");
61.             break;
62.         case FINGERPRINT_NOFINGER:
63.             Serial3.print(".");
64.             delay(200);
65. Serial3.print("Creando modelo para #"); Serial3.println(IDgravar);
66.     p = finger.createModel();
67.     if (p == FINGERPRINT_OK) {
68.         Serial3.println("Huella encontrada");
69. Serial3.print("ID "); Serial3.println(IDgravar);
70.     p = finger.storeModel(IDgravar);
71.     if (p == FINGERPRINT_OK) {
72.         Serial3.println("Almacenado!");

```

Código 2.4. Proceso de ingreso y almacenado de huella dactilar.

En la primera conexión del Sensor biométrico, se registra una huella con ID=0, la cual trabaja como huella administradora que permite el ingreso de nuevas huellas, las cuáles se irán registrando desde ID =1 en adelante.

En el Código 2.5 se observa la parte de actualización de cada huella almacenada en la EEPROM y el despliegue de los datos de ID de usuario y valor de confianza, luego del sensado y reconocimiento de la huella. Estos dos datos se capturan y almacenan, cada uno en un byte de información correspondiente, dentro del payload de la trama de transmisión del nodo.


```

1.  if (finger.fingerID == 0) {
2.      Serial3.print("Modo Administrador!");
3.      ultimoId++;
4.      EEPROM.update(posicionMemoria, ultimoId);
5.      modoGravacaoID(ultimoId);
6.      return 0;
7.  } else {
8.      digitalWrite(pinTrava, LOW);
9.      Serial3.print("ID encontrado #"); Serial3.print(finger.fingerID);
10.     Serial3.print(" con intervalo de confianza de ");
        Serial3.println(finger.confidence);
11.     GPS(finger.confidence, finger.fingerID);
12.     delay(500);
13.     digitalWrite(pinTrava, HIGH);
14.     return finger.fingerID; }

```

Código 2.5. Almacenamiento de huellas en memoria, validación de datos de ID y confianza de huella.

El módulo GPS recibe, decodifica y recupera los valores de longitud y latitud referentes a la posición recibida desde los 3 satélites geoestacionarios a través de la interconexión que mantienen. El Código 2.6 permite capturar los datos de geolocalización, longitud y latitud, que se almacenan cada uno en 4 bytes y posterior, se añaden al payload de la trama de información.

```

1.  while (Serial1.available() > 0){
2.      byte gpsData = Serial1.read();
3.      float ValorLati = gps.location.lat();
4.      float ValorLong = gps.location.lng();
5.      Serial3.println(ValorLati,10);
6.      Serial3.println(ValorLong,10);
7.      byte byteArray1[4];
8.      byte byteArray2[4];
9.      byte seriesOfBytes[4];
10.     float floatVariable;
11.     *((float *)byteArray1) = ValorLati;
12.     *((float *)byteArray2) = ValorLong;

```

Código 2.6. Obtención de los datos de latitud y longitud

El Código 2.7, muestra el armado del payload de la trama Sigfox, almacena los datos de geolocalización, ID de usuario, valor de confianza de huella, tipo de trama información. Este proceso se realiza después de la lectura e identificación de huella dactilar por el sensor biométrico, para posterior envío hacia la nube Sigfox.

```

1.  const uint8_t payloadSize = 12; //in bytes
2.  uint8_t buf_str[payloadSize];
3.  buf_str[0] = byteArray1[0];
4.  buf_str[1] = byteArray1[1];
5.  buf_str[2] = byteArray1[2];
6.  buf_str[3] = byteArray1[3];
7.  buf_str[4] = byteArray2[0];
8.  buf_str[5] = byteArray2[1];
9.  buf_str[6] = byteArray2[2];
10. buf_str[7] = byteArray2[3];
11. buf_str[8] = confianza;
12. buf_str[9] = idHuella;
13. buf_str[10] = 1;
14. buf_str[11] = 0;
15.  Send_Pload(buf_str, payloadSize);
16. Serial3.println("fin");
17. void Send_Pload(uint8_t *sendData, const uint8_t len){
18.     recvMsg *RecvMsg;
19.     RecvMsg = (recvMsg *)malloc(sizeof(recvMsg));
20.     Isigfox->sendPayload(sendData, len, 0, RecvMsg);
21.     for (int i = 0; i < RecvMsg->len; i++) {
22.         Serial3.print(RecvMsg->inData[i]);
23.     Serial3.println("send ok");
24.     free(RecvMsg);
25.     recvMsg *RecvMsg;
26.     RecvMsg = (recvMsg *)malloc(sizeof(recvMsg));
27.     Isigfox->sendPayload(sendData, len, 1, RecvMsg);
28.     for (int i=0; i<RecvMsg->len; i++){
29.         Serial3.print(RecvMsg->inData[i]);
30.     Serial3.println("");
31.     free(RecvMsg);}

```

Código 2.7. Armado del payload de la trama Sigfox a transmitir por la red Sigfox

El payload de la trama de información Sigfox posee un esquema de 12 bytes dispuestos como se muestra en la figura 2.13.

	TIPO	ID	CNFC	LONG	LONG	LONG	LONG	LAT	LAT	LAT	LAT
--	------	----	------	------	------	------	------	-----	-----	-----	-----

Figura 2.13. Payload de la trama Sigfox construida con la información de los sensores

LAT: Latitud

ID: Número de Identificación Huella registrada

LONG: Longitud

TIPO: Tipo de Trama Información

CNFC: Confianza de la Huella Dactilar

Luego de la recepción y alojamiento de los datos en la nube de *Sigfox Cloud*, a través de un aplicativo en un servidor web se realiza la recuperación de la información, mediante la comunicación con el Backend de Sigfox se genera una petición de todos los datos de la trama recibida para su almacenamiento en la base de datos.

2.3.2 CONFIGURACIÓN DE AMAZON WEB SERVICES

Es necesario levantar la instancia en AWS que funciona como el servidor web que recibe los datos desde el Backend de Sigfox, genera la base de datos, dispone y procesa los mismos para su presentación final.

Se debe crear una cuenta en AWS, la figura 2.14 muestra el formulario con los campos de información requeridos por la plataforma, entre ellos tenemos Nombre, Correo, Cuenta de usuario, Tarjeta para facturación, etc.

Figura 2.14. Formulario de cuenta AWS

Al ingresar se presenta la Consola de Gestión AWS, como se muestra en las figuras 2.15 y 2.16, donde se selecciona el servicio web EC2, que permite levantar la instancia requerida. En este caso se selecciona el sistema operativo Ubuntu Server 18.04 LTS (HVM), SSD Volume Type.

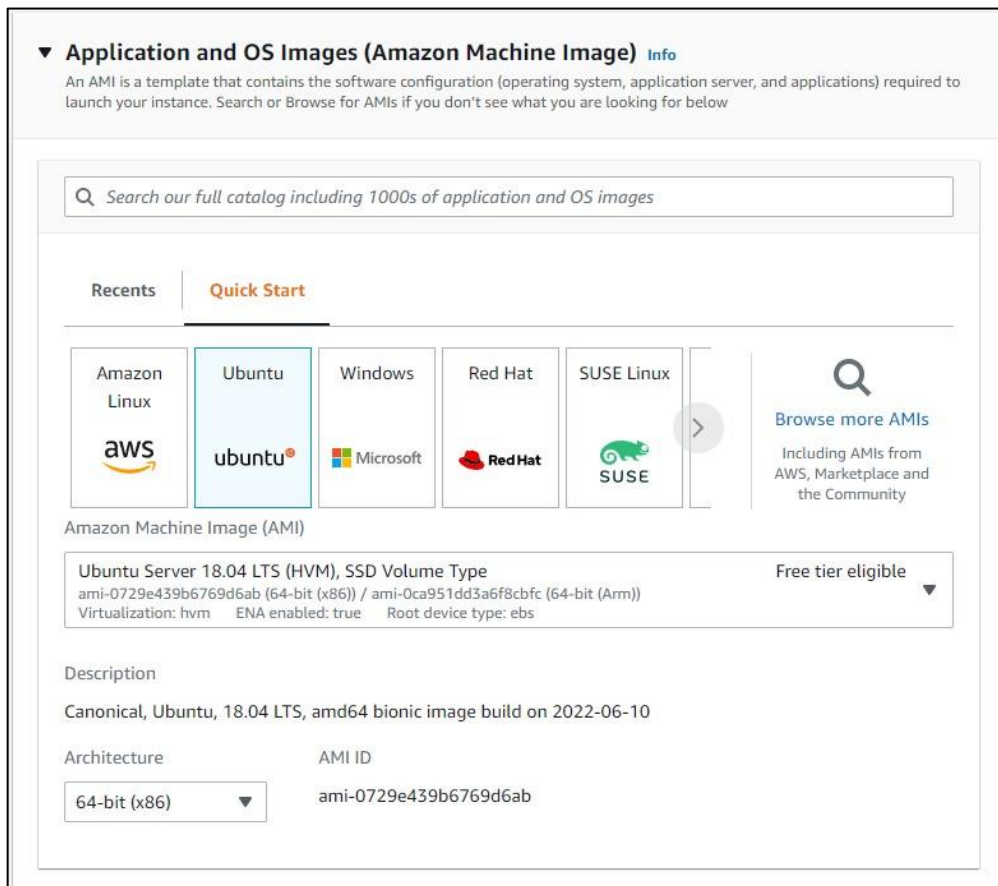


Figura 2.15. Consola de Gestión AWS, selección de sistema operativo

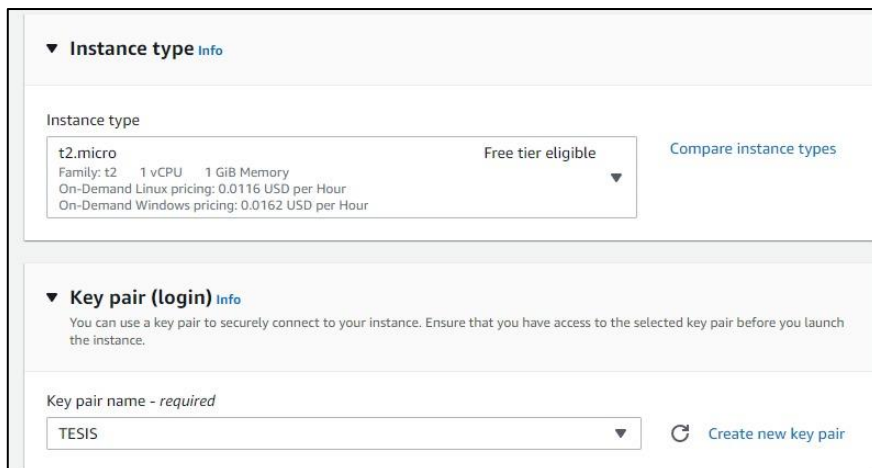


Figura 2.16. Consola de Gestión AWS, selección de tipo de instancia

Luego de configurar la instancia, AWS asigna una máquina virtual desde la cual lanza la misma, definiendo una Dirección IP elástica de dominio, servicio de IP que ofrece la Instancia AWS E2C. En la figura 2.17 se observa los detalles de la instancia levantada.

The screenshot shows the AWS Management Console interface for an EC2 instance. The breadcrumb navigation at the top reads 'EC2 > Instancias > i-03fd47abe506d92d9'. The main heading is 'Resumen de instancia de i-03fd47abe506d92d9 (Tesis) Información'. Below this, there are three columns of details:

- Column 1 (Left):**
 - ID de la instancia: i-03fd47abe506d92d9 (Tesis)
 - Dirección IPv6: -
 - Tipo de nombre de anfitrión: Nombre de IP: ip-172-31-93-207.ec2.internal
 - Responder al nombre DNS de recurso privado IPv4 (A): -
 - Dirección IP asignada automáticamente: -
 - Rol de IAM: -
- Column 2 (Middle):**
 - Dirección IPv4 pública: 52.1.9.7 | dirección abierta
 - Estado de la instancia: En ejecución
 - Nombre DNS de IP privada (solo IPv4): ip-172-31-93-207.ec2.internal
 - Tipo de instancia: t2.micro
 - ID de VPC: vpc-07dacc5e15562de57
 - ID de subred: subnet-0cc828fcd080e5369
- Column 3 (Right):**
 - Direcciones IPv4 privadas: 172.31.93.207
 - DNS de IPv4 pública: ec2-52-1-9-7.compute-1.amazonaws.com | dirección abierta
 - Direcciones IP elásticas: 52.1.9.7 [IP pública]
 - Hallazgo de AWS Compute Optimizer: Suscribirse a AWS Compute Optimizer para recibir recomendaciones. | Más información
 - Nombre del grupo de Auto Scaling: -

Figura 2.17. Portal AWS con detalles de instancia EC2 creada y en ejecución.

PAGO MEMEBRESÍA SERVICIOS AWS

La utilización y mantenimiento de una cuenta Amazon para desarrollo en la plataforma AWS (Web Services), suscripción para manejo de datos, procesamiento, almacenamiento y uso de la instancia EC2 factura un valor de **\$ 11.59** dólares americanos mensuales, como se observa en la figura 2.18.

Consumo Visa in Amazon web Services Aws.amazo	-\$9.95 > \$18.71
Impuesto isd Amazon we	-\$0.45 > \$28.66
Retencion Ivasd Amazon we	-\$1.19 > \$29.11

Figura 2.18. Desglose de pagos por el servicio de AWS

2.3.3 INSTALACIÓN DE HERRAMIENTAS LAMP

Una vez levantada la instancia, se abre el servidor a través de consola de comandos de Ubuntu para la instalación de las herramientas del paquete LAMP que contiene las configuraciones Apache, PHP. En la figura 2.19 se observa el Login y autenticación de Ubuntu desde la ventana de comandos y en la figura 2.20 el panel de control de XAMP donde se observa los servicios de MySQL y apache.

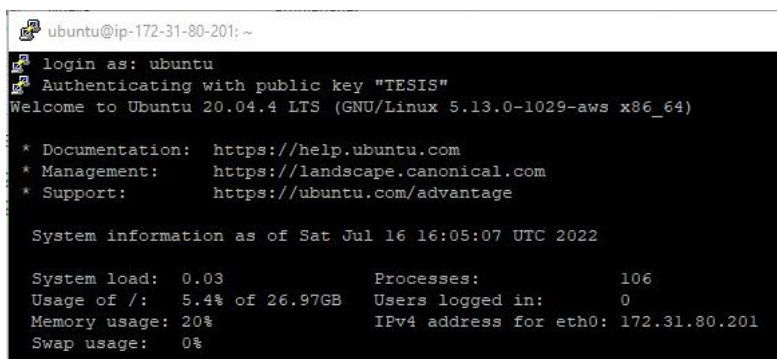


Figura 2.19. Login y autenticación de Ubuntu en CMD

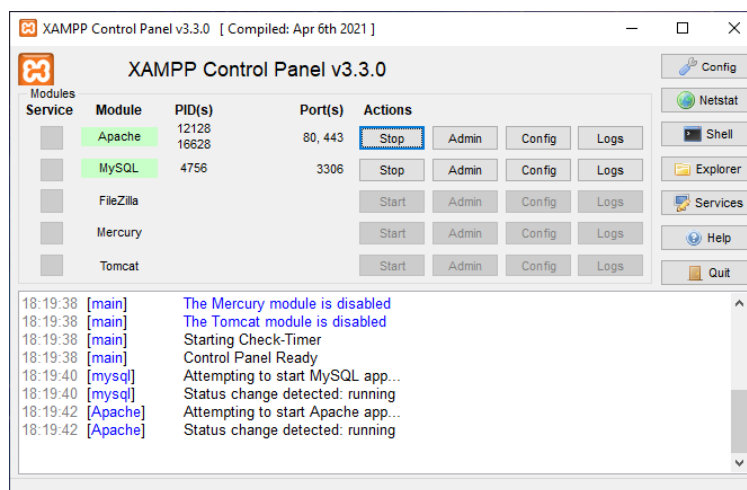


Figura 2.20. Panel de control XAMP

2.3.4 CODIFICACIÓN EN VISUAL STUDIO

Después de instalar el paquete LAMP en el servidor, se procede a instalar Composer para la configuración de Laravel como framework de desarrollo para la aplicación web. Mediante Laravel obtenemos plantillas para desarrollo en PHP, archivos de vistas con el nombre blade.php que brinda codificación con herencia de plantillas y secciones para mejor direccionamiento de rutas. [33]

2.3.4.1 Codificación de la Base de Datos

Se crea y conecta una base de datos que sirva como repositorio de almacenamiento de la información que se recupera de la nube de Sigfox. Con la ayuda de MySQL, mediante Workbench se realiza el diseño y documentación de la base de datos, se genera las tablas para ordenar y mantener una lógica acorde con la trama recibida. La base de datos, tablas, el orden y disposición de los datos, se realiza a partir de sentencias SQL.

Se configura manualmente la conexión con XAMPP y la base de datos MySQL, a través de phpMyAdmin. El código 2.21 muestra los detalles de la conexión.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=dashboard
DB_USERNAME=root
```

Figura 2.21. Código de conexión a la base de datos

A continuación, se realiza la migración que permite utilizar las tablas que Laravel genera por defecto. La implementación de Laravel en bases de datos permite trabajar con modelos predeterminados de tablas que brindan funcionalidades de consulta de datos, adición y sustracción de registros, etc. En el código 2.22 se muestra los detalles de la migración y en el código 2.23 se observa el código de creación de la tabla usuarios.

```
C:\Users\Terranova\Dropbox\Diego 1\TESIS\Desarrollo\dashboard>php artisan migrate
C:\Users\Terranova\Dropbox\Diego 1\TESIS\Desarrollo\dashboard\vendor\laravel\framework\src\Illuminate\Database\Connectors\Connector.php:70
PDO::__construct("mysql:host=127.0.0.1;port=3306;dbname=dashboard", "root", "", [])
```

Figura 2.22. Código de migración para la base de datos

```
Schema::create('usuarios', function (Blueprint $table) {
    $table->id();
    $table->integer('idUsuario');
    $table->string('Cedula');
    $table->string('Nombres');
    $table->string('Apellidos');
    $table->string('Direccion');
    $table->string('Telefono');
    $table->date('FechaNacimiento');
    $table->timestamps();
});
```

Figura 2.23. Código creación de la Tabla Usuarios

2.3.4.2 Comunicación con el Backend de Sigfox

Para capturar los datos alojados en la nube Sigfox, se configura la comunicación entre el Backend y el servidor por medio de un callback que redireccione y transmita estos datos almacenados en la nube Sigfox.

El Código 2.8 muestra el establecimiento de la conexión y la configuración del callback con el servidor y la obtención de datos.

```
1. <?php
2. header('Content-type: application/json');
3. echo "Sigfox Manager<br>";
4. if($_GET)
5. {   if(isset($_GET['id']))
6.     {   if(isset($_GET['data']) && isset($_GET['time']) && isset($_GET['seqNumber'])) )
7.       {   $varFechaHora=0;
8.           if(isset($_GET['time']))
9.           {   date_default_timezone_set('America/Guayaquil');
10.              $varFechaHora=date("Y-m-d H:i:s", $_GET['time']);
11.              echo "fecha: ".$varFechaHora;}
12.             $servername = "localhost";
13.             $username = "diego";
14.             $password = "TesisDiego123";
15.             $dbname = "dbSigfox";
16.             $conn = new mysqli($servername, $username, $password, $dbname);
17.             if ($conn->connect_error) {
18.                 die("Connection failed: " . $conn->connect_error);
19.             }else { echo "Connected successfully2";
```

Código 2.8. Configuración callback entre el servidor y el Backend

El Código 2.9 muestra la función de conversión de datos. La trama payload que parte desde el nodo sensor almacena los bytes de información en formato Hexadecimal, por lo que se debe especificar un nuevo formato de almacenamiento para posterior visualización. En este caso se realiza la conversión de cada dato a formato Decimal.


```

1. function hexfloat(string $hex): float{
2.     $aux=$hex[6];
3.     $hex[6]=$hex[0];
4.     $hex[0]=$aux;
5.     $aux=$hex[7];
6.     $hex[7]=$hex[1];
7.     $hex[1]=$aux;
8.     $aux=$hex[4];
9.     $hex[4]=$hex[2];
10.    $hex[2]=$aux;
11.    $aux=$hex[5];
12.    $hex[5]=$hex[3];
13.    $hex[3]=$aux;
14.    $dec = hexdec($hex);
15.    if ($dec === 0) {
16.        return 0;}
17.    $sup = 1 << 23;
18.    $x = ($dec & ($sup - 1)) + $sup * ($dec >> 31 | 1);
19.    $exp = ($dec >> 23 & 0xFF) - 127;
20.    $sign = ($dec & 0x80000000) ? -1 : 1;
21.    return $sign * $x * pow(2, $exp - 23);}

```

Código 2.9. Conversión Hexadecimal - Decimal de los datos de la trama

El Código 2.10 permite obtener cada uno de los datos, establece la llegada de los datos desde el Backend de Sigfox y realiza el almacenamiento en la base de datos.

```

1.         $stmt = $conn->prepare("INSERT INTO tblMensajes (numSecuencia, fechaHora,
idDispositivo, latitud, longitud, confianza, idUsuario) VALUES (?, ?, ?, ?, ?, ?, ?);");
2.         $stmt->bind_param("isiddii", $numSecuencia, $fechaHora, $idDispositivo,
$latitud, $longitud, $confianza, $idUsuario);
3.         $numSecuencia=$_GET['seqNumber'];
4.         $fechaHora=$varFechaHora;
5.         $idDispositivo=$_GET['id'];
6.         $subcadena=substr($_GET['data'],0,8);
7.         $latitud = hexfloat($subcadena);
8.         $subcadena=substr($_GET['data'],8,8);
9.         $longitud = hexfloat($subcadena);
10.        $subcadena=substr($_GET['data'],16,2);
11.        $confianza = hexdec($subcadena);
12.        $subcadena=substr($_GET['data'],18,2);
13.        $idUsuario = hexdec($subcadena);
14.        $rc = $stmt->execute();
15.        $stmt->close();
16.        $conn->close(); } } }

```

Código 2.10. Obtención y establecimiento de llegada de la información.

La figura 2.24 muestra el callback creado para comunicación desde la plataforma Sigfox.

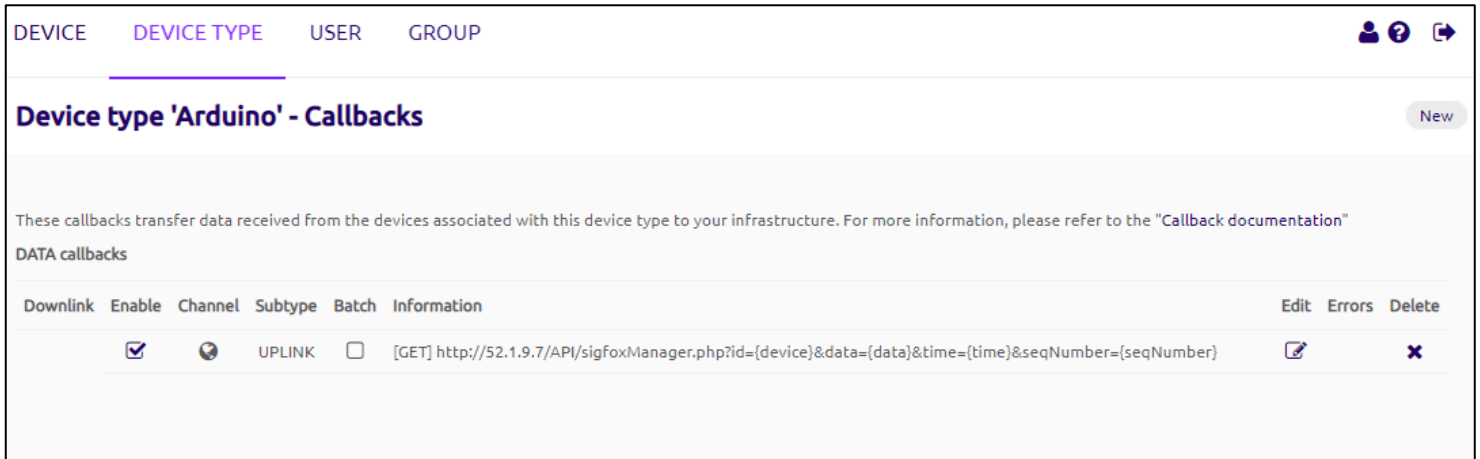


Figura 2.24. Portal *Sigfox Cloud* mostrando el registro del dispositivo y el callback.

Con la obtención y decodificación de los datos, se almacena en la tabla configurada con las cabeceras correspondientes para cada dato, como se muestra en la figura 2.25. Los datos de fecha y hora se recogen en el Backend de Sigfox de los registros de fecha y hora de recepción de la trama en la nube.

numSecuencia	fechaHora	idDispositivo	latitud	longitud	confianza	idUsuario
12	2021-04-05 13:03:20	2312	-0.122464	-78.4659	75	3
244	2021-04-09 15:27:52	418740	0	0	52	6
245	2021-04-09 15:28:06	418740	0	0	117	7
246	2021-04-09 15:28:18	418740	0	0	82	7
247	2021-04-09 15:28:28	418740	0	0	67	7
249	2021-04-09 15:34:51	418740	-0.122413	-78.466	59	6
250	2021-04-09 15:35:13	418740	-0.122431	-78.466	73	7
253	2021-04-09 16:09:41	418740	-0.121756	-78.4654	88	1
254	2021-05-21 11:21:18	418740	0	0	120	6
255	2021-05-21 11:21:35	418740	0	0	55	6
256	2021-05-21 11:21:54	418740	0	0	52	6
258	2021-06-11 10:52:13	418740	0	0	105	6
262	2021-06-11 12:46:38	418740	-0.12243	-78.466	62	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 2.25. Base datos MySQL con los datos tabulados recuperados del Backend

Se crean entidades controladoras para formar las rutas de búsqueda y llamadas a la base datos. Como se muestra en la figura 2.26, se identifican las entidades Mensajes y Usuarios que son los que interactúan en el sistema.

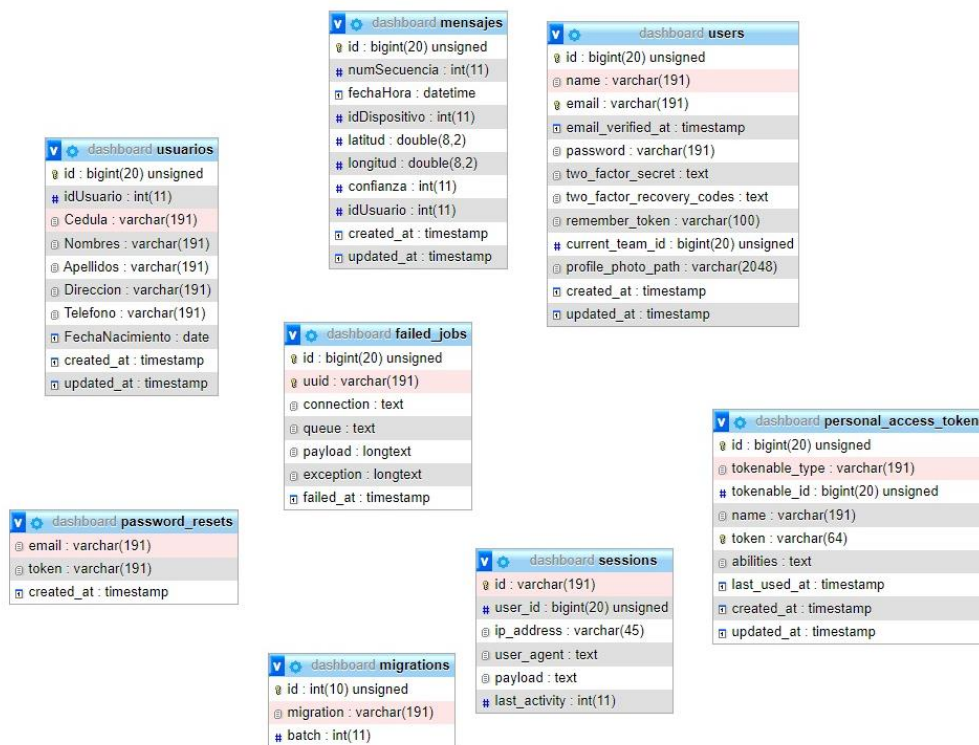


Figura 2.26. Diagrama entidad relación de la base de datos

El Código 2.11 muestra la jerarquía de rutas especificadas en el archivo web.php. Esta jerarquía proporciona acceso a los componentes del sistema, ya que las rutas establecen el inicio del servidor web.

```

1. <?php
2. use Illuminate\Support\Facades\Route;
3. use App\Http\Controllers\UsuarioController;
4. Route::get('/', function () {
5.     return redirect('mensajes');});
6. Route::get('/home', function () {
7.     return redirect('mensajes');});
8. Route::resource('mensajes', 'App\Http\Controllers\MensajeController');
9. Route::resource('usuarios', 'App\Http\Controllers\UsuarioController');
10. Route::get('usuarios/idUsuario/{id}', [UsuarioController::class,
    'showbySigfoxId']);
11. Route::middleware(['auth:sanctum', 'verified'])->get('/dashboard',
    function () {
12.     return view('dashboard');
13. })->name('dashboard');
```

Código 2.11. Jerarquía de rutas, entidades controladoras.

2.3.4.3 Codificación del Dashboard

Después del almacenamiento de los registros de la trama en la base de datos, se disponen para visualización en el dashboard mediante una página web. Los Códigos 2.12 y 2.13 son referentes al desarrollo del dashboard, que permiten la administración del sistema y la visualización de los mensajes tabulados.

```
1. @extends('adminlte::page')
2. @section('title', 'Dashboard')
3. @section('content_header')
4.     <h1>Mensajes</h1>
5. @stop
6. @section('content')
7.     <table id="table_id" class="table table-striped" style="width:100%">
8.         <thead>
9.             <tr>
10.                 <th scope="col">Numero de Secuencia</th>
11.                 <th scope="col">Fecha y Hora</th>
12.                 <th scope="col">Id Dispositivo</th>
13.                 <th scope="col">Latitud</th>
14.                 <th scope="col">Longitud</th>
15.                 <th scope="col">Confianza</th>
16.                 <th scope="col">Id Usuario</th>
17.                 <th scope="col">Ubicación</th>
18.             </tr>
19.         </thead>
20.         <tbody>
21.             @foreach ($mensajes as $msg)
22.                 <tr>
23.                     <th scope="col">{{ $msg["numSecuencia"] }}</th>
24.                     <th scope="col">{{ $msg["fechaHora"] }}</th>
25.                     <th scope="col">{{ $msg["idDispositivo"] }}</th>
26.                     <th scope="col">{{ $msg["latitud"] }}</th>
27.                     <th scope="col">{{ $msg["longitud"] }}</th>
28.                     <th scope="col">{{ $msg["confianza"] }}</th>
29.                     <th scope="col"><a
30. href="/usuarios/idUsuario/{{ $msg["idUsuario"] }}">{{ $msg["idUsuario"] }}</a></th>
31.                     <th scope="col"><a target="_blank"
32. href="http://www.google.com/maps/place/{{ $msg["latitud"] }},{{ $msg["longitud"] }}">
33. Mostrar Ubicación</a></th>
34.                 </tr>
35.             </tbody>
36.         </table>
37. @stop
38. @section('css')
39.     <link rel="stylesheet" href="/css/admin_custom.css">
```

Código 2.12. Creación de la tabla para despliegue de Mensajes.

```

40.     <link rel="stylesheet"
href="https://cdn.datatables.net/1.10.25/css/dataTables.bootstrap5.min.css">
41.     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/5.0.1/css/bootstrap.min.css">
42. @stop
43. @section('js')
44. <script src="https://code.jquery.com/jquery-3.5.1.js"></script>
45. <script
src="https://cdn.datatables.net/1.10.25/js/jquery.dataTables.min.js"></script>
46. <script
src="https://cdn.datatables.net/1.10.25/js/dataTables.bootstrap5.min.js"></script
>
47.     <script> $(document).ready( function () {
48.         $('#table_id').DataTable({
49.             "lengthMenu": [[50,100,200,-1],[50,100,200,"All"]] });
50.     } );</script>
51. @stop

```

Código 2.13. Codificación de Mensajes para consulta de datos y disposición de la tabla

Se realiza la configuración de la entidad Mensajes para que muestre la tabla de información de los datos almacenados en el orden deseado, como se observa en la figura 2.27.

Se personalizan las plantillas y las secciones de los archivos que se generaron al realizar la migración y autenticación de la base de datos.

Numero de Secuencia	Fecha y Hora	Id Dispositivo	Latitud	Longitud	Confianza	Id Usuario	Ubicación
566	2022-08-03 06:52:57	418740	-0.27071199	-78.56238556	51	2	Mostrar Ubicación
565	2022-08-02 14:21:34	418740	-0.27075949	-78.56240845	98	3	Mostrar Ubicación
564	2022-08-02 12:57:36	418740	-0.27080816	-78.56236267	74	4	Mostrar Ubicación
563	2022-08-02 12:48:37	418740	-0.27071518	-78.56240082	143	4	Mostrar Ubicación
562	2022-08-02 12:38:37	418740	-0.27076799	-78.56245422	101	4	Mostrar Ubicación
561	2022-08-02 12:28:19	418740	-0.27073231	-78.56240082	78	1	Mostrar Ubicación
560	2022-08-02 12:20:57	418740	-0.27087200	-78.56242371	84	2	Mostrar Ubicación

Figura 2.27. Dashboard desplegado en la página web – Tabla Mensajes con registros.

Se implementa un CRUD (*Create, Read, Update, Delete*), para la entidad Usuarios, estos elementos son desarrollados cada uno en los archivos formato PHP: *Create, Edit, Index* y *Show*. Este CRUD permite realizar una gestión de usuarios para leer, agregar, editar y borrar usuarios que consten con el registro de la huella digital e ID de usuario.

La entidad Usuario Controller se encarga de gestionar las peticiones para el CRUD, además de llamar a los archivos Blade.php asociados. Los Códigos 2.13 y 2.14 son referentes al desarrollo del CRUD de gestión de usuarios.

```
1. <?php
2. namespace App\Http\Controllers;
3. use Illuminate\Http\Request;
4. use App\Models\Usuario;
5. class UsuarioController extends Controller
6. {
7.     public function index()
8.     {
9.         $usuarios=Usuario::all();
10.        return view('Usuario.index')->with('usuarios',$usuarios); }
11.    public function create()
12.    {
13.        return view('Usuario.create'); }
14.    public function store(Request $request)
15.    {
16.        $request->validate([
17.            'cedula' => 'required',
18.            'nombres' => 'required',
19.            'apellidos' => 'required',
20.            'direccion' => 'required',
21.            'telefono' => 'required',
22.            'FechaNacimiento' => 'required',
23.            'idUsuario' => 'required', ]);
24.        $Usuario = new Usuario();
25.        $Usuario->Cedula = $request->get('cedula');
26.        $Usuario->Nombres = $request->get('nombres');
27.        $Usuario->Apellidos = $request->get('apellidos');
28.        $Usuario->Direccion = $request->get('direccion');
29.        $Usuario->Telefono = $request->get('telefono');
30.        $Usuario->idUsuario = $request->get('idUsuario');
31.        $Usuario->FechaNacimiento = $request->get('FechaNacimiento');
32.        $Usuario->save(); return redirect('/usuarios');
```

Código 2.13. Codificación de CRUD para Añadir nuevo usuario.

```

41. public function show($id)
42. {     $usuario = Usuario::find($id);
43.     return view('Usuario.show')->with('usuario', $usuario); }
44. public function showbySigfoxId($id)
45. {
46.     $usuario = Usuario::where('idUsuario', $id)->first();
47.     return view('Usuario.show')->with('usuario', $usuario);
48. }
49. public function edit($id)
50. {
51.     $usuario = Usuario::find($id);
52.     return view('Usuario.edit')->with('usuario', $usuario);
53. }
54. public function update(Request $request, $id)
55. {
56.     $request->validate([
57.         'cedula' => 'required',
58.         'nombres' => 'required',
59.         'apellidos' => 'required',
60.         'direccion' => 'required',
61.         'telefono' => 'required',
62.         'FechaNacimiento' => 'required',
63.         'idUsuario' => 'required',
64.     ]);
65.     $Usuario = Usuario::find($id);
66.     $Usuario->Cedula = $request->get('cedula');
67.     $Usuario->Nombres = $request->get('nombres');
68.     $Usuario->Apellidos = $request->get('apellidos');
69.     $Usuario->Direccion = $request->get('direccion');
70.     $Usuario->Telefono = $request->get('telefono');
71.     $Usuario->idUsuario = $request->get('idUsuario');
72.     $Usuario->FechaNacimiento = $request->get('FechaNacimiento');
73.     $Usuario->save();
74.     return redirect('/usuarios'); }
75. public function destroy($id)
76. {
77.     $Usuario = Usuario::find($id);
78.     $Usuario->delete();
79.     return redirect('/usuarios'); } }

```

Código 2.14. Ingreso y registro de datos de Usuario.

3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1. RESULTADOS

Este capítulo presenta las pruebas correspondientes al funcionamiento e implementación del prototipo del nodo y el sistema de control de cumplimiento de aislamiento.

En la Tabla 3.1 se presenta el tablero Kanban actualizado con las actividades del trabajo.

Tabla 3.1. Actualización Tabla Kanban con el detalle de las diferentes tareas.

PENDIENTES	EN PROGRESO	REALIZADO
	Realizar pruebas del nodo sensor y aplicación web	Estudio de características, manejo y configuración de implementos y servicios de Hardware y Software
		Instalar y configurar el nodo Sensor
		Configuración del código de transmisión y armar la trama de payload
		Configurar el servidor
		Configurar LAMP
		Codificación base de datos
		Configuración conexión Backend
		Codificar y configurar la recuperación de datos almacenados
		Diseñar y configurar el servicio web
		Diseño y codificación del Dashboard

3.1.1. PRUEBAS DE FUNCIONAMIENTO

La implementación del sistema de control de cumplimiento de cuarentena cuenta con un nodo sensor, el software de configuración del nodo Sigfox y el software de gestión de datos.

Sigfox operando con los parámetros y características de tecnología IoT, junto con el diseño específico del Nodo, permite el funcionamiento del mismo en cualquier zona geográfica que cuente con cobertura de la red Sigfox 0G. La gran mayoría de zonas del Distrito Metropolitano de Quito, cuentan con cobertura de red, como lo muestra la Figura 3.1

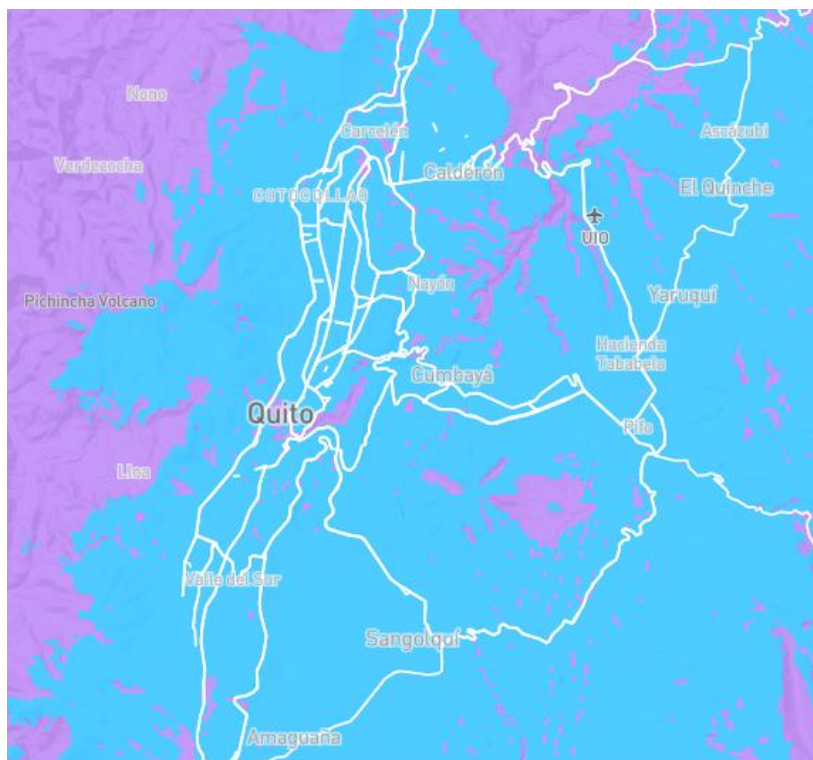


Figura 3.1. Cobertura Red 0G Sigfox en Quito

El nodo al ser energizado se mantiene operativo, sensando y capturando el ingreso de las huellas dactilares. Existen varios leds dispuestos en los componentes del nodo, que funcionan como indicadores visuales de operación, así como también el Optical Panel Light Sensor del Fingerprint para captura de huellas dactilares.

El indicador LED de la conexión del módulo GPS con los satélites geoestacionarios empieza a parpadear cuando ya recibe la información de geolocalización, esta conexión se efectiviza en un promedio de 40 segundos, dependiendo la zona geográfica y el entorno.

En la Figura 3.2 se observa el Nodo Sensor implementado y operativo.

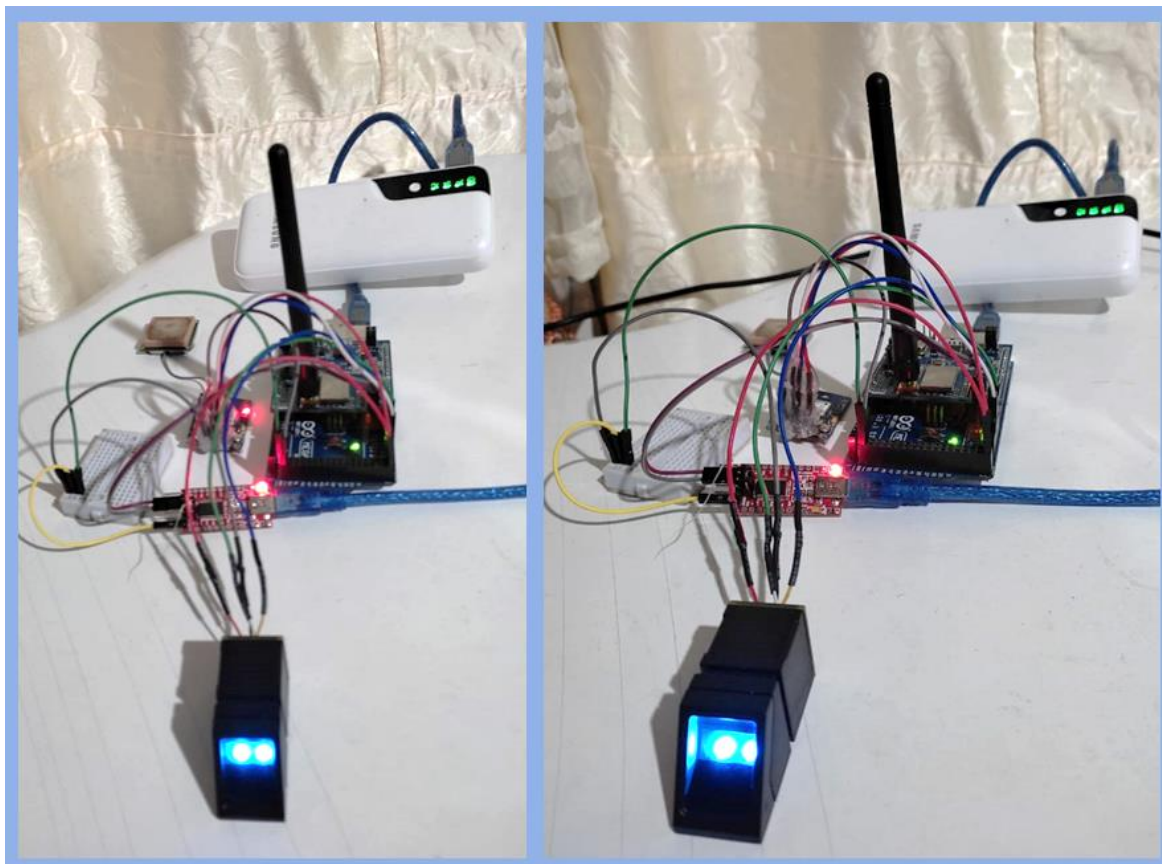


Figura 3.2. Implementación del Nodo-Sensor prototipo

De igual manera, dependiendo del entorno donde se realice la captura de huella y la transmisión de información, se obtiene un valor de calidad de enlace de transmisión. Este dato se registra en el Backend de Sigfox y se muestra como el parámetro LQI en la parte de Mensajes de Dispositivo. En este punto se puede validar el establecimiento de la conexión del nodo con el Backend, a través de los registros de Mensajes recibidos, como se muestra en la Figura 3.3.

Time	Seq Num	Data / Decoding	LQI	Callbacks	Location
2022-07-15 15:22:02	408	d9a78abef91f9dc2480301	4	↑	📍
2022-07-15 15:21:21	407	63a68abef81f9dc2430101	4	↑	📍
2022-07-15 15:19:16	406	2da88abefe1f9dc2330101	4	↑	📍
2022-07-15 15:14:23	405	000000000000000007f0101	4	↑	📍

Figura 3.3. Backend de Sigfox con los mensajes recibidos desde el nodo sensor

Para el registro de huellas en el nodo sensor, se ingresa el valor de huella ID=0, con el cual permite ingresar nuevos registros de huellas dactilares para los pacientes dispuestos a monitoreo. El proceso de registro y almacenamiento de las huellas en el sensor biométrico se realiza a través de un emulador de terminal serial en la PC, donde se observan las indicaciones necesarias para el proceso de ingreso de nueva huella y la generación de su respectivo ID de usuario. En la figura 3.4 se muestra el monitor serial mientras el nodo registra una nueva huella.

```
Serial Puerto abierto
Sensor biometrico encontrado!
Configuración de sensor dactilar completa
Prototipo Sistema Control de Cumplimiento de Cuarentena COVID 19 UIO
Iniciando...
Device ID: 00418740

Modo Administrador! Esperando una huella para guardar usuario #15
Lectura concluida
Lectura convertida
Remueva el dedo
ID 15
Coloque el mismo dedo nuevamente
...Lectura concluida
WD reset: 0
Lectura convertida
Creando modelo para #15
Huella encontrada
```

Figura 3.4. Ventana Terminal con despliegue de captura y registro de nueva huella

En la figura 3.5 se observa desde el monitor serial las notificaciones de captura, identificación, valor de confianza, visualización de datos de geolocalización (coordenadas de latitud y longitud) de la huella dactilar sensada y el indicador de transmisión hacia la nube Sigfox.

```
Serial Puerto abierto
Sensor biometrico encontrado!
Configuración de sensor dactilar completa
Prototipo Sistema Control de Cumplimiento de Cuarentena COVID 19 UIO
Iniciando...
Device ID: 00418740

...
ID encontrado #1 con intervalo de confianza de 60
GPS
-0.2707760095
-78.5623550415
-----
OK
send ok
fin
ID encontrado #2 con intervalo de confianza de 71
GPS
-0.2707760095
-78.5623550415
-----
OK
send ok
fin
```

Figura 3.5. Ventana Terminal con despliegue de captura y reconocimiento de huellas

Con la instancia EC2 corriendo, se realiza la conexión del servidor de base de datos para el ingreso a la página web mostrando el Dashboard implementado y validando los datos de cada usuario. La figura 3.6 muestra la ventana de Mensajes que despliega los datos que partieron desde el Nodo Sensor.

Numero de Secuencia	Fecha y Hora	Id Dispositivo	Latitud	Longitud	Confianza	Id Usuario	Ubicación
566	2022-08-03 06:52:57	418740	-0.27071199	-78.56238556	51	2	Mostrar Ubicación
565	2022-08-02 14:21:34	418740	-0.27075949	-78.56240845	98	3	Mostrar Ubicación
564	2022-08-02 12:57:36	418740	-0.27080816	-78.56236267	74	4	Mostrar Ubicación
563	2022-08-02 12:48:37	418740	-0.27071518	-78.56240082	143	4	Mostrar Ubicación
562	2022-08-02 12:38:37	418740	-0.27076799	-78.56245422	101	4	Mostrar Ubicación
561	2022-08-02 12:28:19	418740	-0.27073231	-78.56240082	78	1	Mostrar Ubicación
560	2022-08-02 12:20:57	418740	-0.27087200	-78.56242371	84	2	Mostrar Ubicación
559	2022-08-02 12:06:04	418740	-0.27073500	-78.56240082	168	1	Mostrar Ubicación
558	2022-08-02 12:00:21	418740	-0.27082551	-78.56214142	63	1	Mostrar Ubicación
557	2022-08-02 11:14:58	418740	-0.27074468	-78.56234741	186	3	Mostrar Ubicación
556	2022-08-02 11:11:19	418740	-0.27069116	-78.56234741	85	2	Mostrar Ubicación
555	2022-08-02 11:10:57	418740	-0.27072251	-78.56238556	209	2	Mostrar Ubicación
554	2022-08-02 11:09:12	418740	-0.27072135	-78.56236267	144	1	Mostrar Ubicación

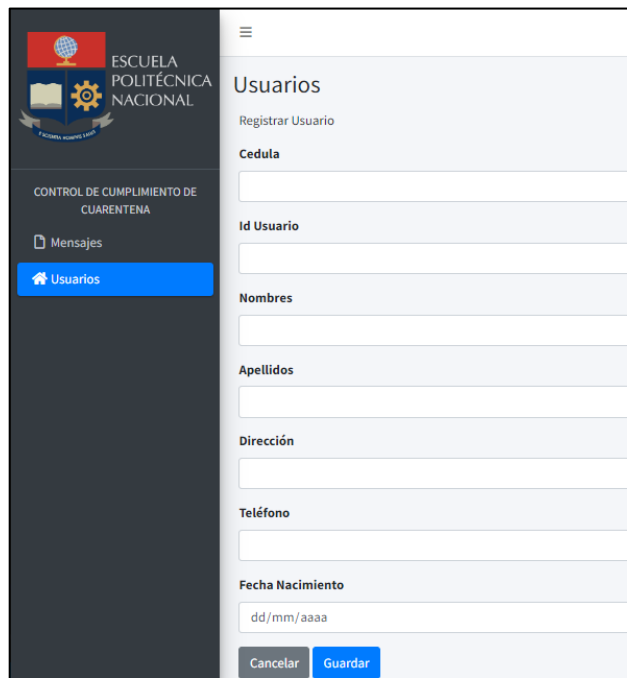
Figura 3.6. Ventana de visualización de Mensajes

La ventana de Usuarios despliega tabulados los diferentes datos de información que se añaden mediante el CRUD codificado para la gestión de usuarios. En la Figura 3.7 se muestra la Ventana Usuarios, en la parte superior, el botón celeste Crear, y en la parte derecha de la tabla se observan los botones Detalles, Editar y Borrar, que permiten ejecutar la gestión de usuarios en la página web.

Cédula	Id Usuario	Nombres	Apellidos	Dirección	Teléfono	Fecha Nacimiento	Acciones
1718192022	8	Marcel	Salamanca	San Ignacio de Loyola	0987654321	2007-02-02	Detalles Editar Borrar
1719034199	14	Franck Joseph	Velásquez Arroyo	Santa Rosa 3	0986574124	2016-09-08	Detalles Editar Borrar
1724119812	7	Julian	Loza	quito	0981717234	2022-07-08	Detalles Editar Borrar
1724987458	6	Paola	Sinchiguano	quito	+593995248514	2022-06-30	Detalles Editar Borrar

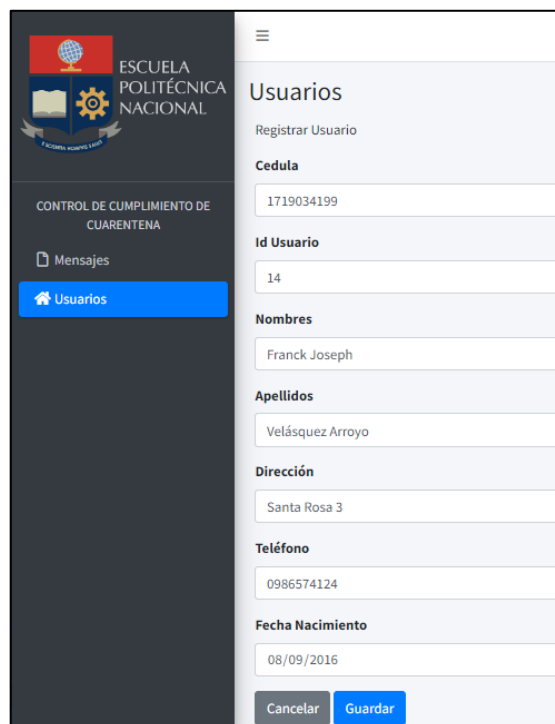
Figura 3.7. Ventana de visualización de Usuarios

Dando Clic en el botón de navegación Crear, se despliega un formulario que permite ingresar los datos de información del usuario asignado a una ID de usuario referente a la captura del nodo sensor. Como se muestra en las figuras 3.8 y 3.9, se ingresan los datos de Cédula, ID, Nombres, Apellidos, Dirección, Teléfono y Fecha de Nacimiento.



The screenshot shows a mobile application interface for 'Escuela Politécnica Nacional'. The left sidebar contains the school logo, the text 'CONTROL DE CUMPLIMIENTO DE CUARENTENA', and navigation buttons for 'Mensajes' and 'Usuarios'. The main screen displays a 'Registrar Usuario' form with the following fields: Cedula, Id Usuario, Nombres, Apellidos, Dirección, Teléfono, and Fecha Nacimiento. The 'Fecha Nacimiento' field has a placeholder 'dd/mm/aaaa'. At the bottom of the form are 'Cancelar' and 'Guardar' buttons.

Figura 3.8. Formulario para Crear Registro de Usuario Nuevo



The screenshot shows the same mobile application interface as Figure 3.8, but the 'Registrar Usuario' form is now filled with patient information. The fields contain: Cedula (1719034199), Id Usuario (14), Nombres (Franck Joseph), Apellidos (Velásquez Arroyo), Dirección (Santa Rosa 3), Teléfono (0986574124), and Fecha Nacimiento (08/09/2016). The 'Cancelar' and 'Guardar' buttons are still visible at the bottom.

Figura 3.9. Formulario lleno con los datos de información general del paciente

Con el botón Cancelar, cancela la operación de registro y regresa a la ventana Usuarios y con el botón Guardar, se guarda y registran los datos de información del usuario y se añade automáticamente a la tabla de Usuarios, como se muestra en la Figura 3.10.

Cédula	Id Usuario	Nombres	Apellidos	Dirección	Teléfono	Fecha Nacimiento	Acciones
1719034199	14	Franck Joseph	Velásquez Arroyo	Santa Rosa 3	0986574124	2016-09-08	Detalles Editar Borrar
1724119812	7	Julian	Loza	quito	0981717234	2022-07-08	Detalles Editar Borrar

Figura 3.10. Usuario añadido exitosamente

Dando Clic en el botón de navegación Detalles, se muestran los datos de información registrados del usuario correspondiente, como se observa en la Figura 3.11.

Figura 3.11. Ventana de visualización Detalles de Usuario

Con el botón de navegación Volver, regresa a la ventana Usuario.

Esta misma ventana de Detalles de Usuario ha sido configurada para que también aparezca luego de dar clic en los números de ID de usuario desde la página Mensajes, como se muestra en la Figura 3.12.

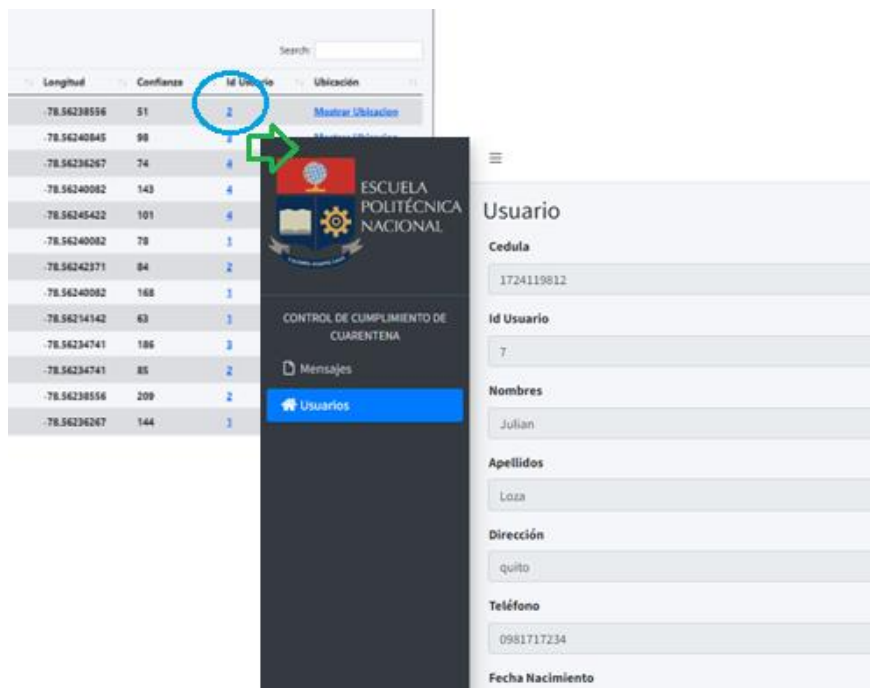


Figura 3.12. Visualización Detalles desde la ID de usuario en ventana Mensajes

Dando Clic en el botón de navegación Editar, se despliegan los datos del usuario en el formulario, donde podremos editar y/o actualizar los datos de información. Este proceso se observa en la figura 3.13.

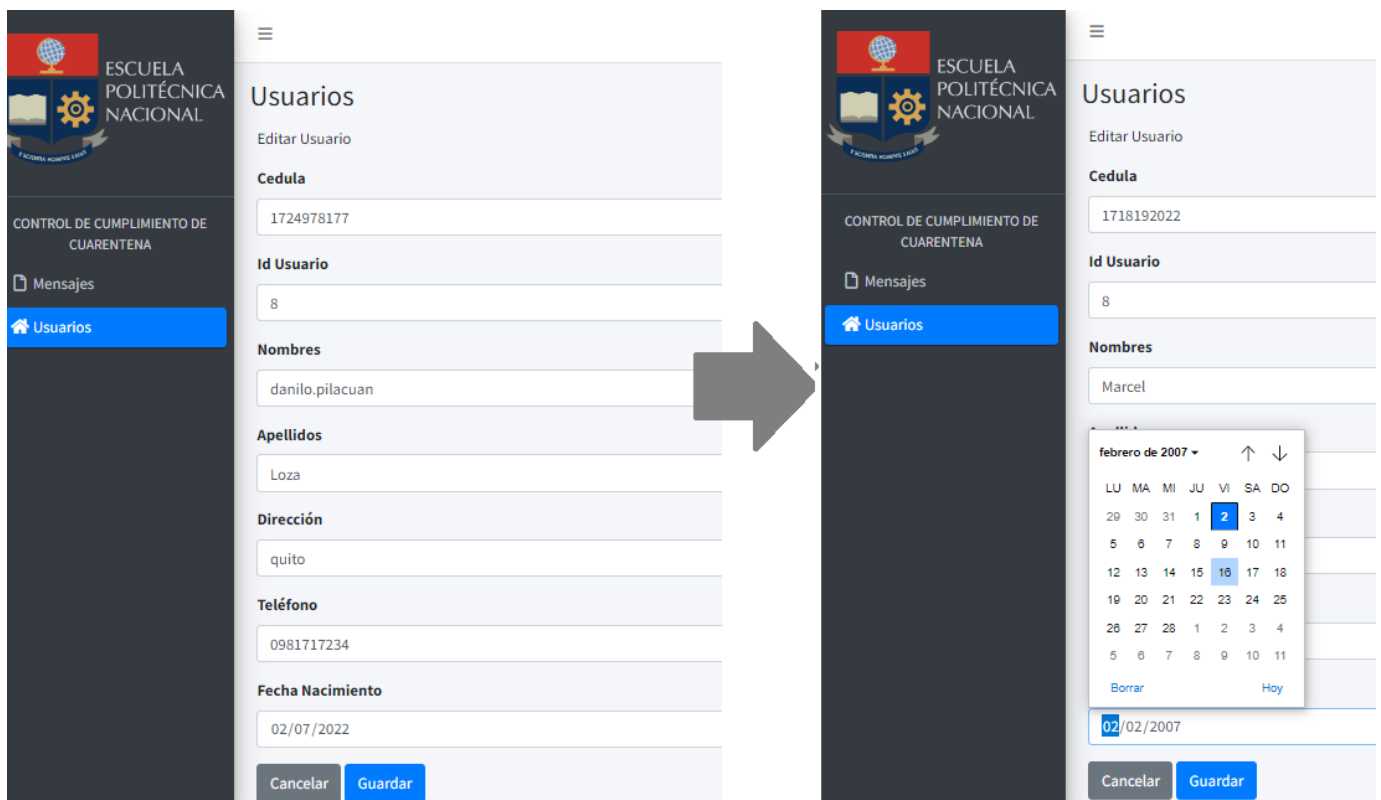


Figura 3.13. Ventana para Editar los datos de un Usuario registrado.

a) Datos anteriores, b) Registro de nuevos datos

Con el botón Cancelar, cancela la operación de edición y regresa a la Ventana Usuario y con el botón Guardar, se guardan los datos de información que fueron editados y se actualizan automáticamente en la tabla de Usuarios, como se muestra en la Figura 3.14.

Cédula	Id Usuario	Nombres	Apellidos	Dirección	Teléfono	Fecha Nacimiento	Acciones
1718192022	8	Marcel	Salamanca	San Ignacio de Loyola	0987654321	2007-02-02	Detalles Editar Borrar
1719034199	14	Franck Joseph	Velásquez Arroyo	Santa Rosa 3	0986574124	2016-09-08	Detalles Editar Borrar
1724119812	7	Julian	Loza	quito	0981717234	2022-07-08	Detalles Editar Borrar

Figura 3.14. Usuario con Datos Actualizados

Dando Clic en el botón de navegación Borrar, se puede eliminar el registro de un usuario determinado, este proceso se muestra en la figura 3.15. En la figura 3.16, se observa la actualización de la tabla de Usuarios, sin el usuario previamente borrado.

Cédula	Id Usuario	Nombres	Apellidos	Dirección	Teléfono	Fecha Nacimiento	Acciones
1718192022	8	Marcel	Salamanca	San Ignacio de Loyola	0987654321	2007-02-02	Detalles Editar Borrar
1719034199	14	Franck Joseph	Velásquez Arroyo	Santa Rosa 3	0986574124	2016-09-08	Detalles Editar Borrar
1724119812	7	Julian	Loza	quito	0981717234	2022-07-08	Detalles Editar Borrar
1724978174	5	Danilo	Pilacuan	quito	+593987171309	2022-07-05	Detalles Editar Borrar
1724987458	6	Paola	Sinchiguano	quito	+593995248514	2022-06-30	Detalles Editar Borrar

Figura 3.15. Ventana de visualización Usuarios antes de usar Botón Borrar

Cédula	Id Usuario	Nombres	Apellidos	Dirección	Teléfono	Fecha Nacimiento	Acciones
1718192022	8	Marcel	Salamanca	San Ignacio de Loyola	0987654321	2007-02-02	Detalles Editar Borrar
1719034199	14	Franck Joseph	Velásquez Arroyo	Santa Rosa 3	0986574124	2016-09-08	Detalles Editar Borrar
1724119812	7	Julian	Loza	quito	0981717234	2022-07-08	Detalles Editar Borrar
1724987458	6	Paola	Sinchiguano	quito	+593995248514	2022-06-30	Detalles Editar Borrar

Figura 3.16. Actualización Usuarios - Usuario Borrado

En la parte derecha superior de la ventana del Dashboard, se encuentra la barra de búsqueda, donde se puede realizar un filtrado práctico de coincidencia según la información que se añade en la barra. En la figura 3.17 se muestra el filtrado desde la ventana de Usuarios.

The screenshot shows a table with the following data:

Cédula	Id Usuario	Nombres	Apellidos	Dirección	Teléfono	Fecha Nacimiento	Acciones
1719034199	1	Diego René	Velásquez García	Santa Rosa 3	0987344877	1994-02-28	Detalles Editar Borrar
1719034231	3	Luis Enrique	Velásquez García	La Planicie	0983493042	1996-02-22	Detalles Editar Borrar

Search bar: Search:

Showing 1 to 2 of 2 entries (Filtered from 7 total entries)

Figura 3.17. Utilización de la barra de búsqueda en el Listado de Usuarios Registrados

Desde la Ventana de Mensajes, en la columna de Ubicación, al dar clic en Mostrar ubicación, se abre una ventana adicional, como se muestra en la figura 3.18, en la página web Google Maps se coloca el marcador en las coordenadas de latitud y longitud del Usuario correspondiente, validando el lugar donde se encuentra realizando aislamiento y el nodo registró el ingreso de huella dactilar.

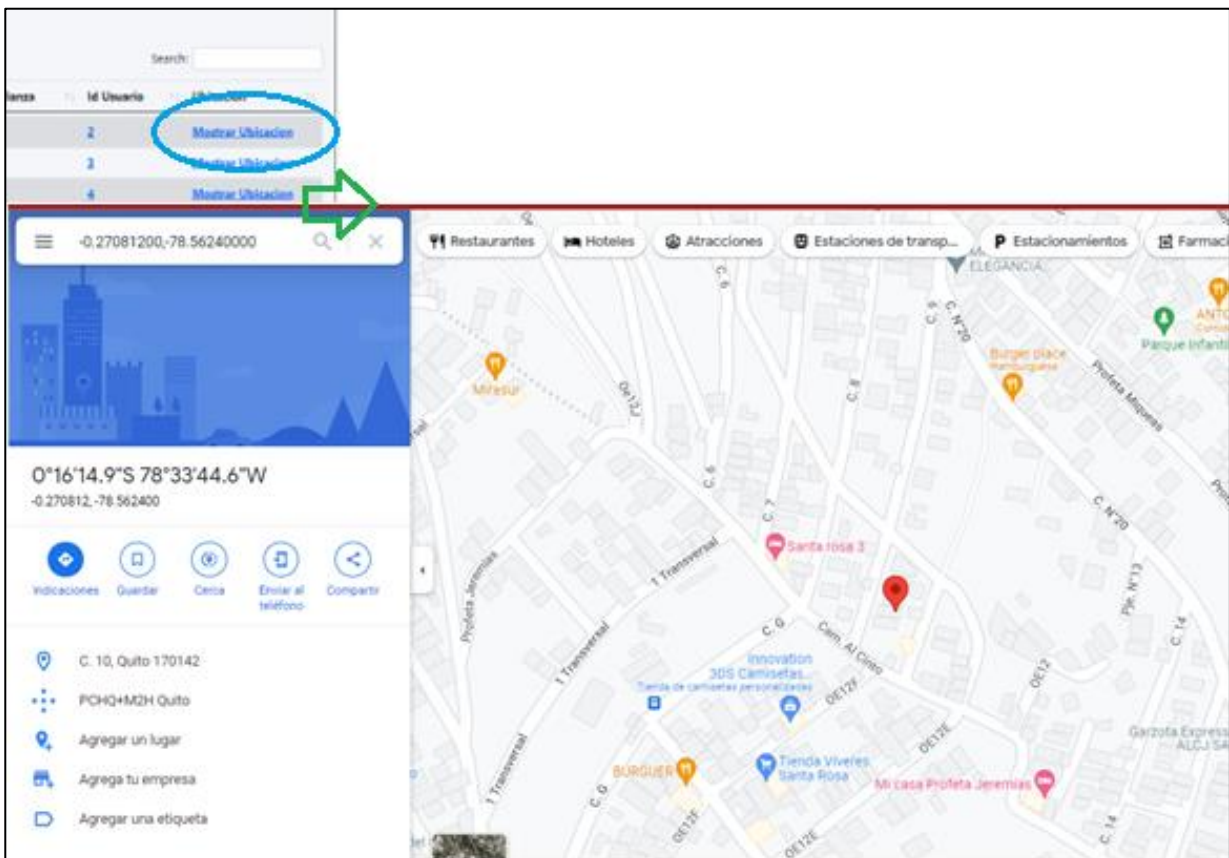


Figura 3.18 Ventana de visualización Localización en Google Maps

Una vez comprobados los resultados en capturas de pantalla, acerca del correcto recorrido de la trama de datos a través de todo el sistema desarrollado, el funcionamiento del aplicativo IoT, la gestión de los usuarios y la visualización de los mensajes con la información correspondiente de cada usuario en el Dashboard, se describen estos resultados fundamentados en funcionalidad y la experiencia de usuario.

FUNCIONALIDAD

El presente trabajo detalla el desarrollo de un sistema prototipo basado en IoT, implementando un nodo sensor transmisor, que permita controlar el cumplimiento del aislamiento debido al contagio de COVID 19 al transmitir los datos de identificación, localización, fecha y hora, y la visualización de los mismos en una página web, bajo esta premisa, el sistema desarrollado e implementado cumple con la descripción de los objetivos planteados.

El nodo transmite los datos recogidos de cada componente sensor hacia la nube Sigfox por internet, levantada una instancia que define una máquina virtual para gestión y tratamiento de los datos recibidos, mediante la comunicación con el Backend se recuperan y retransmiten los datos, se almacenan en una base de datos y se disponen para posterior visualización en un dashboard que administra los usuarios y despliega los datos. Luego de realizadas las pruebas de funcionamiento se valida y verifica el correcto funcionamiento del sistema prototipo IoT desarrollado.

EXPERIENCIA DE USUARIO

Los usuarios pacientes a cumplir aislamiento se registran en el nodo con la ayuda de un usuario admin y el nodo conectado a una PC para el registro de su huella dactilar.

El Fingerprinter cuenta con amplia memoria de almacenamiento para huellas y un sensor óptico de muy alta calidad de recepción por lo que los ingresos y registros han sido exitosos en la gran mayoría de las ocasiones.

Se debe considerar las zonas y entorno donde se realizan las capturas de huellas dactilares y la transmisión de la información ya que las comunicaciones RF en ambientes con factores poco favorables de transmisión, se podrá perder el enlace del módulo GPS, obteniendo serie de ceros "0" de geolocalización y la posible pérdida de la trama Sigfox por una baja calidad de enlace.

El funcionamiento de prototipo y la interacción de los usuarios con el aplicativo son simples, obteniendo los resultados esperados con una experiencia de usuario satisfactoria y registros favorables del sistema prototipo implementado.

3.2. CONCLUSIONES

Se recopiló información precisa sobre la utilización, configuración, programación y despliegue del sistema prototipo con el nodo sensor IoT Sigfox Arduino de manera clara, ordenada y metódica, realizando pruebas exitosas de operatividad y funcionalidad, del sistema de transmisión desarrollado.

Se implementó el nodo sensor prototipo con tecnología Sigfox para la adquisición y transmisión de los datos de identificación, geolocalización, fecha, hora y alarma.

Se desarrolló el software para la configuración y funcionamiento del nodo sensor Sigfox compatible con Arduino.

Se desarrolló el software por medio de un servidor web que permita obtener, almacenar, procesar y presentar los datos en la PC de usuario.

Se desarrolló e implementó un prototipo basado en *Internet of Things*, conjuntamente con sistemas embebidos mediante las tecnologías Sigfox, módulo de transmisión Thinxtra DevKit y Arduino, placa de desarrollo Arduino MEGA 2560, ambas comparten compatibilidad y presentan una infraestructura óptima para sistemas que permitan el monitoreo mediante el nodo sensor implementado y el aplicativo web desarrollado.

Se implementó un sistema prototipo para el control del cumplimiento de cuarentena de pacientes enfermos con COVID 19 en la ciudad de Quito.

El presente trabajo estará disponible como base de consulta para futuras investigaciones de usos, recursos y servicios disponibles que se puedan seguir desarrollando e innovando en esta nueva tecnología Sigfox, conjunto con su red 0G y el Internet de las Cosas (IoT), en varios ámbitos de la ciencia y la vida diaria como servicios de mensajería, manejo de datos, rastreo, automatización, ciudades inteligentes, sistemas domóticos, etc.

3.3. RECOMENDACIONES

El módulo Thinxtra DevKit viene conjunto con una placa Arduino UNO de procesamiento de datos, por lo que se recomienda, si en la aplicación es necesario anexar un mayor número de sensores o dispositivos compatibles, utilizar la placa de desarrollo Arduino Mega 2560, debido a que además de poseer un mayor número de entradas y brinda total compatibilidad con el módulo Thinxtra DevKit de transmisión.

Tanto el módulo Thinxtra Devkit como la placa de desarrollo para procesamiento de datos son compatibles con tecnología Arduino y su configuración principal se lo realiza con el IDE de Arduino, por lo que se recomienda conseguir, conectar y configurar sensores adicionales que sean compatibles con la tecnología Arduino.

Se recomienda realizar las mediciones desde ambientes abiertos y desde alturas medias considerables, con el fin de obtener una transmisión y recepción de datos idónea, permitiendo la conexión del módulo GPS con los satélites y el envío de la trama Sigfox, obteniendo una calidad de enlace óptima.

De ser posible, en futuras implementaciones, se puede mejorar la experiencia de usuario, referente a la interfaz gráfica del Dashboard, por el momento tiene funcionalidades sencillas. Por lo que, es recomendable tomar en cuenta la posibilidad de añadir nuevas funcionalidades para la visualización de los datos presentados, ya sea la presentación de una foto de perfil del usuario registrado, un sistema de filtrado por usuario o por número de cédula con mejor disposición de datos, desplegar el mapa de ubicación en la misma página, entre otras, todas estas diferentes funcionalidades y características acordes a estándares internacionales de calidad de software.

4. REFERENCIAS BIBLIOGRÁFICAS

[1] Observatorio Social del Ecuador, "Monitoreo del coronavirus covid-19 en Ecuador", 20 de junio de 2022, [Online] Disponible en: <https://www.covid19ecuador.org/ecuador>

[2] Observatorio Social del Ecuador, "Evolución del Coronavirus por cantones", 20 de junio de 2022 [Online] Disponible en: <https://www.covid19ecuador.org/cantones>

[3] El Español, Gobierno de España, 2019, Como se esparce el COVID 19 [Online] Disponible en: <https://espanol.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/how-covid-spreads.html>

[4] WND Group, "Qué es la tecnología @Sigfox y todo lo que podemos hacer con ella #WeNeedSigfox", 12 septiembre 2017, [En línea]. Disponible: <https://www.wndgroup.io/2017/09/12/que-es-la-tecnologia-sigfox-y-todo-lo-que-podemos-hacer-con-ella-weneedsigfox/>

[5] "Ventajas y desventajas del "internet de las cosas" (IoT)", Beetrack Blog, 2020, [Online], Disponible en: <https://www.beetrack.com/es/blog/ventajas-y-desventajas-internet-de-las-cosas-iot>

[6] "Cinco medidas para controlar Pandemia COVID 19 festividades Diciembre", Ministerio de Salud Pública del Ecuador, salud.gob.ec, diciembre 2021, [Online], Disponible en: <https://www.salud.gob.ec/cinco-medidas-controlar-pandemia-covid-19-festividades-diciembre-ecuador/>

[7] Ogonaga, Steven; Chiriboga, Silvana, "ARTÍCULO DE REVISIÓN COVID 19 EN ECUADOR: ANÁLISIS DESCRIPTIVO DE LAS PROVINCIAS Y CIUDADES MÁS AFECTADAS". Facultad de Ciencias Económicas, Universidad Central del Ecuador. Revista del Grupo de Investigación en Comunidad y Salud, Volumen 5, Depósito Legal electrónico: ME2016000090 - ISSN Electrónico: 2610797X, 2020 [Online] Disponible en: <http://erevistas.saber.ula.ve/gicos>

[8] Definición Internet de las cosas, Kevin Ashton en el RFID journal (Ashton, 2009)

[9] EOI - Escuela de Organización Industrial, "Las tecnologías IoT dentro de la industria conectada 4.0", 2020, [Online], Disponible en: <https://www.eoi.es/es/file/20417/download?token=FSUSqFlt>

[10] Brian Ray, 2017, LTE-M & 2 Other 3GPP IoT Technologies To Get Familiar With, [Online], Disponible en: <https://www.link-labs.com/lte-iot-technologies/>

- [11] Filip, B., Flaviu M., F.-I., Attila, S., Sorin, M., Doru, V., & Petru, A. (2020). LoRaWAN Based Real-Time Air Quality Monitoring System. IEEE 18th World Symposium on Applied Machine Intelligence and Informatics, 69-72.
- [12] Vilorio Núñez César; Cardona Peña Jairo; Lozano Garzón Carlos. "Análisis comparativo de tecnologías inalámbricas para una solución de servicios de telemedicina", Universidad del Norte, Barranquilla, Colombia, Ingeniería y Desarrollo, ISSN: 0122-3461, enero-junio, 2009, pp. 200-217
- [13] Modificado de Egli, P. R. (2015). Indigoo, [Online], Obtenido de http://indigoo.com/dox/itdp/12_MobileWireless/LPWAN.pdf?_ga=2.211
- [14] María Luisa Machado González, "Estudio de NB-IoT y comparativa con otras tecnologías LPWAN", Master Universitario en Ingeniería de Telecomunicaciones UOC-URL, Telemática, enero 2019
- [15] C. A. González-González, F. Arévalo-Tapias, J. Hernández-Gutiérrez, "Análisis de seguridad en redes LPWAN para dispositivos IoT". Revista Vínculos: Ciencia Tecnología y Sociedad, vol. 16, no. 2, julio-diciembre de 2019, pp. 252-261. DOI: 10.14483/2322939X.1571
- [16] T. Janssen, M. Aernouts, R. Berkvens y M. Weyn, "Outdoor Fingerprinting Localization Using Sigfox". 2020 [Online], Disponible en: https://www.researchgate.net/publication/328982852_Outdoor_Fingerprinting_Localization_Using_Sigfox
- [17] Pablo Pardal Garcés, "Redes de Área Extensa para aplicaciones de IoT: modelado de comunicaciones Sigfox", Universidad Politécnica de Valencia 14 de Julio de 2017
- [18] Antezano Gorvenia, Josué Jhonatan Sánchez Rojas, Luis Eduardo, Trabajo de Investigación "Diseño y Simulación de un Prototipo para la Localización de Incendios Forestales utilizando Tecnologías Inalámbricas basado en IoT", Para obtener el grado Académico de Bachiller en: Ingeniería de Redes y Comunicaciones Lima diciembre 2019
- [19] SIGFOX (2022), Nuestra historia, Infografía febrero 2022, (Online) Disponible en: <https://www.sigfox.com/en/sigfox-story>
- [20] K Mekki, E. Bajic, F. Chaxel, F. Meyer, "A comparative Study of LPWAN technologies for large-scale IoT deployment" France, Dec. 2017

- [21] Perry Lea. "Internet of Things for Architects" Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security, 2018. ISBN-10: 1788470591.
- [22] SIGFOX (2020), Tecnología de radio IoT Sigfox [Online], Disponible en: <https://www.sigfox.com/en/sigfox-iot-radio-technology>
- [23] Crespo Enrique, Arduino y Sigfox, Aprendiendo Arduino, 2018-03-5, [Online], Disponible en: <https://www.aprendiendoarduino.com/2018/03/05/arduino-y-sigfox/>
- [24] Ing. Pablo Martres, "Particularidades de la red Sigfox", Artículos de Ingeniería, Escuela de Desarrollo Empresarial, Universidad de la Empresa, Uruguay, [Online], Disponible en: <https://ude.edu.uy/particularidades-de-la-red-sigfox/>
- [25] Alejandro Cama, Emiro De la Hoz, Dora Cama, "Las redes de sensores inalámbricos y el Internet de las cosas" Revista INGE CDC., vol. 8, no. 1, pp. 163–172, 1 de oct. 2012.
- [26] Plácido Humberto Molina Rivillas, "MONITORIZACIÓN DE PLAZAS DE PARKING EXTERIOR MEDIANTE SENSORES WASPMOTE Y TECNOLOGÍA LPWAN SIGFOX", UNIVERSIDAD DE JAÉN Escuela Politécnica Superior de Linares, octubre 2018 [Online], Disponible en: <http://tauja.ujaen.es/handle/10953.1/8385>
- [27] Portal Backend Sigfox (2022), (Online) Disponible en: <https://backend.sigfox.com/welcome/news>, 2021
- [28] Lledó Sánchez, E. (2012). Diseño de un sistema de control domótico basado en la plataforma Arduino. <http://hdl.handle.net/10251/18228>.
- [29] A. Pedrea Caicedo, Arduino para Principiantes: 2da Ed., IT Campus Academy, 2017
- [30] Molina Ríos, J.R., Zea Ordóñez, M.P., Contento Segarra, M.J. y García Zerda, F. G "Estado del arte: Metodologías de desarrollo en aplicaciones web. 3C Tecnología: glosas de innovación aplicadas a la pyme", (2017). DOI: <<http://dx.doi.org/10.17993/3ctecno.2017.v6n3e23.54-71/>>.

- [31] C. Gómez Montoya, C. Candela Uribe, y L. Sepúlveda Rodríguez, Seguridad en la configuración del servidor web Apache, INGE CUC, vol. 9, n.º 2, pp. 31-38, dic. 2013.
- [32] Diego Chaparro, "LAMP: Linux, Apache, MySQL y PHP/Perl" (dchaparro@acm.org) 5 de abril de 2006, [Online], Disponible en: <http://viejo.dchaparro.net/doc/lamp.pdf>
- [33] desarrolloweb.com, Artículo Web, "Composer, gestor de dependencias para PHP", 24 de enero 2020, [En línea] Disponible en: <https://desarrolloweb.com/articulos/composer-gestor-dependencias-para-php.html>
- [34] MICROSOFT. docs.microsoft.com. 2017, [Online] Disponible en: <https://docs.microsoft.com/eses/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-netframework>
- [35] Quality devs, "Que es Laravel" 23 de junio de 2021. [En línea]. Disponible en: <https://www.qualitydevs.com/2021/06/23/que-es-laravel/>
- [36] Varcarcel Tejon Miguel, "Arquitectura serverless en AWS" Thesis, Universidad Politecnica de Madrid, junio 2018, [Online], Disponible en: https://oa.upm.es/52770/1/TFG_MIGUEL_VALCARCEL_TEJON.pdf
- [37] Jesús Alejandro Paz Villafaña, "Arquitectura de Amazon Elastic Compute Cloud (Amazon EC2) y la utilización del hipervisor XEN", 23 de Junio del 2020, [Online], Disponible en : https://www.researchgate.net/profile/Jesus-Paz-Villafana/publication/342588920_Arquitectura_de_Amazon_Elastic_Compute_Cloud_Amazon_EC2_y_la_utilizacion_del_hipervisor_XEN_1_Introduccion/links/5efc1dc8458515505080fd02/Arquitectura-de-Amazon-Elastic-Compute-Cloud-Amazon-EC2-y-la-utilizacion-del-hipervisor-XEN-1-Introduccion.pdf
- [38] MetInvestigacion, Alba Lucía Marín Villada, "METODOLOGÍA DE LA INVESTIGACIÓN, CLASIFICACION DE LA INVESTIGACION", 7 marzo 2008, [En línea]. Disponible: <https://metinvestigacion.wordpress.com/>
- [39] Kanbanize, "Que es Kanban: Fundamentos," 2019. [En línea]. Disponible en: <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban/>.
- [40] Blog Programa Fácil, "Arduino Mega 2560 el hermano mayor de Arduino UNO", José Guerra Carmenate, [Online], Disponible en: <https://programarfacil.com/blog/arduino-blog/arduino-mega-2560/>
- [41] Sigfox, <https://partners.sigfox.com/products/xkit-rc4>

[42] Francisco Alonso Sarría, “Sistemas de Información Geográfica”, [Online], Disponible en: <https://www.um.es/geograf/sigmur/sigpdf/temario.pdf>

[43] u-blox. (2011, 4 de enero). NEO-6 Series, Versatil u-blox 6 GPS modules. [Archivo electrónico]. de u-blox AG: <https://goo.gl/1TfwXq>.

[44] Proyectos y Robótica, “Lector de Huellas Dactilares R307, Descripción” [Online], Disponible en: <https://www.carrod.mx/products/modulo-sensor-r307-lector-de-huella-digital>

[45] TECmikro, “Conversor convertidor USB Serial TTL”, 2022. [En línea]. Disponible en: <https://tecmikro.com/conversores-convertidores/517-conversor-convertidor-usb-serial-ttl.html>

[46] Winiarski, Benjamin, “Piscina inteligente controlada mediante Alexa”, Arduino Project, febrero 2018, [Online], Disponible en: <https://create.arduino.cc/projecthub/benwiniarski/smart-pool-alexa-controlled-pool-manager-5454c1>

5. ANEXOS

ANEXO I. Diagrama de pines de la placa de desarrollo Arduino MEGA 2560

ANEXO II. Diagrama de pines del módulo Thinxtra DevKit de Sigfox y módulo Wisol TX/RX.

ANEXO III. Código Arduino de Configuración y Transmisión de Nodo

ANEXO IV. Código de Desarrollo de Servicio Web Aplicación IoT

ANEXO V. Datasheet Arduino MEGA 2560

ANEXO VI. Datasheet Módulo GPS NEO 6M Ublox

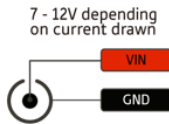
ANEXO VII. Datasheet Informativo Fingerprint Arduino

ANEXO VIII. Datasheet FTDI232 Conversor TTL Serial

ANEXO IX. Guías de desarrollo e instructivos Thinxtra Devkit

ANEXO I

Diagrama de pines de la placa de desarrollo Arduino MEGA 2560.

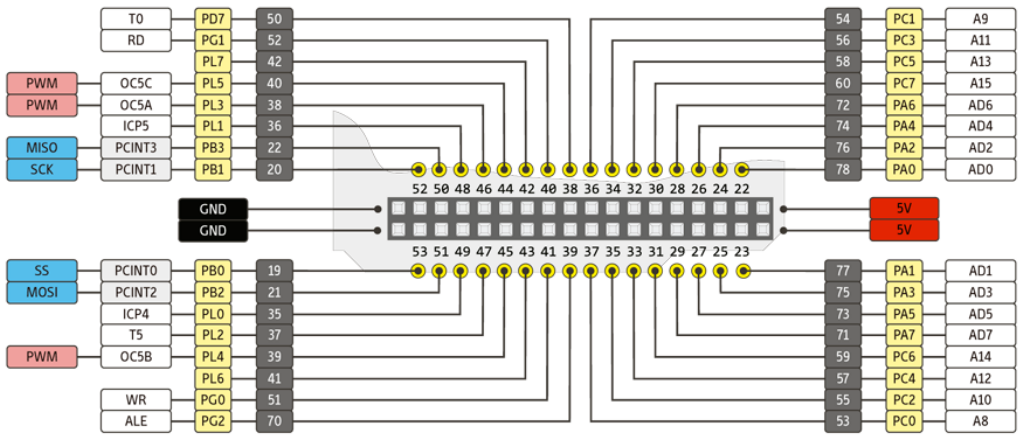
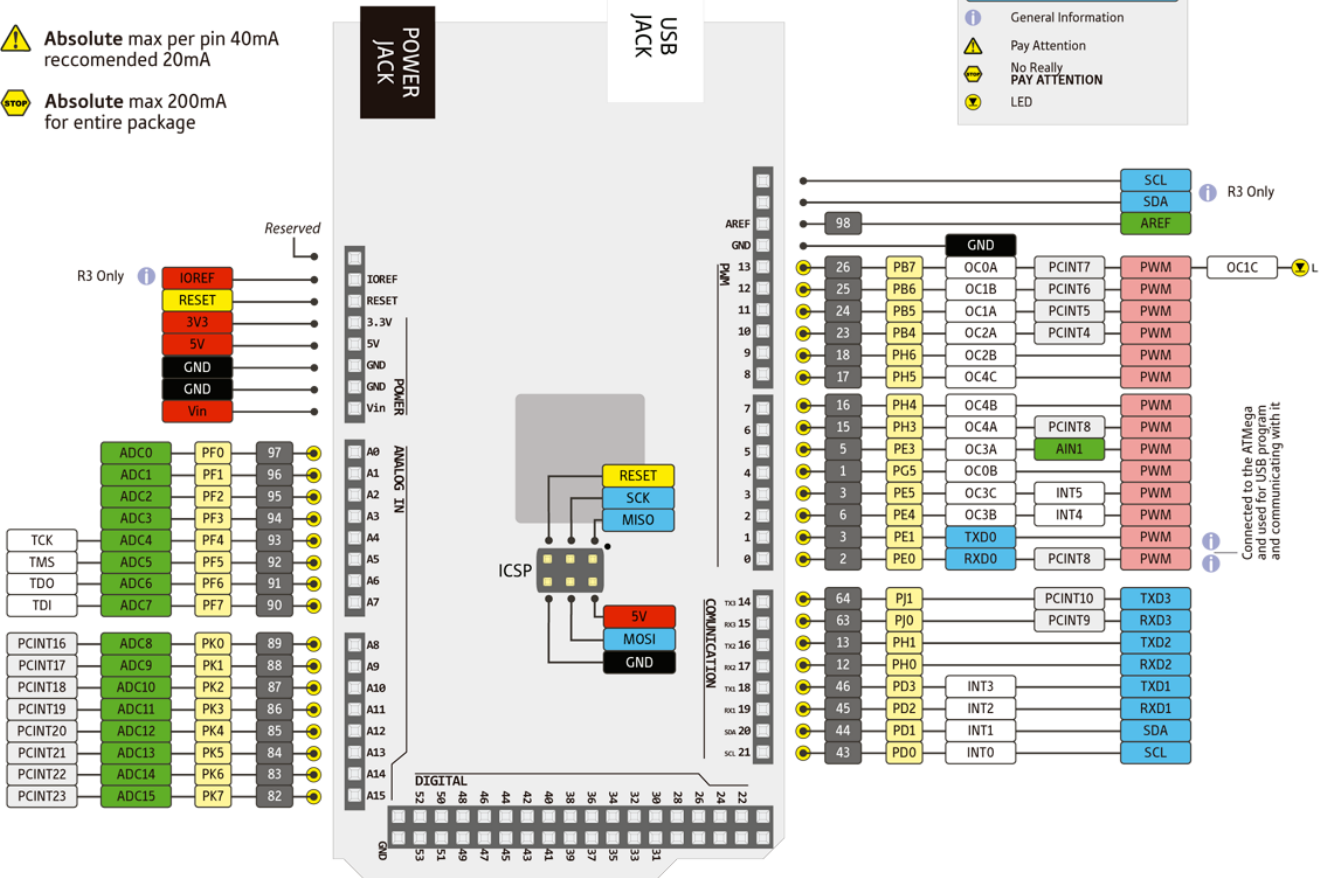


 	GND
 	POWER
 	CONTROL
 	PHYSICAL PIN
 	PORT PIN
 	ATMEGA PIN FUNC
 	DIGITAL PIN
 	ANALOG-RELATED PIN
 	PWM PIN
 	SERIAL PIN

i	General Information
!	Pay Attention
!	No Really PAY ATTENTION
!	LED

! Absolute max per pin 40mA recommended 20mA

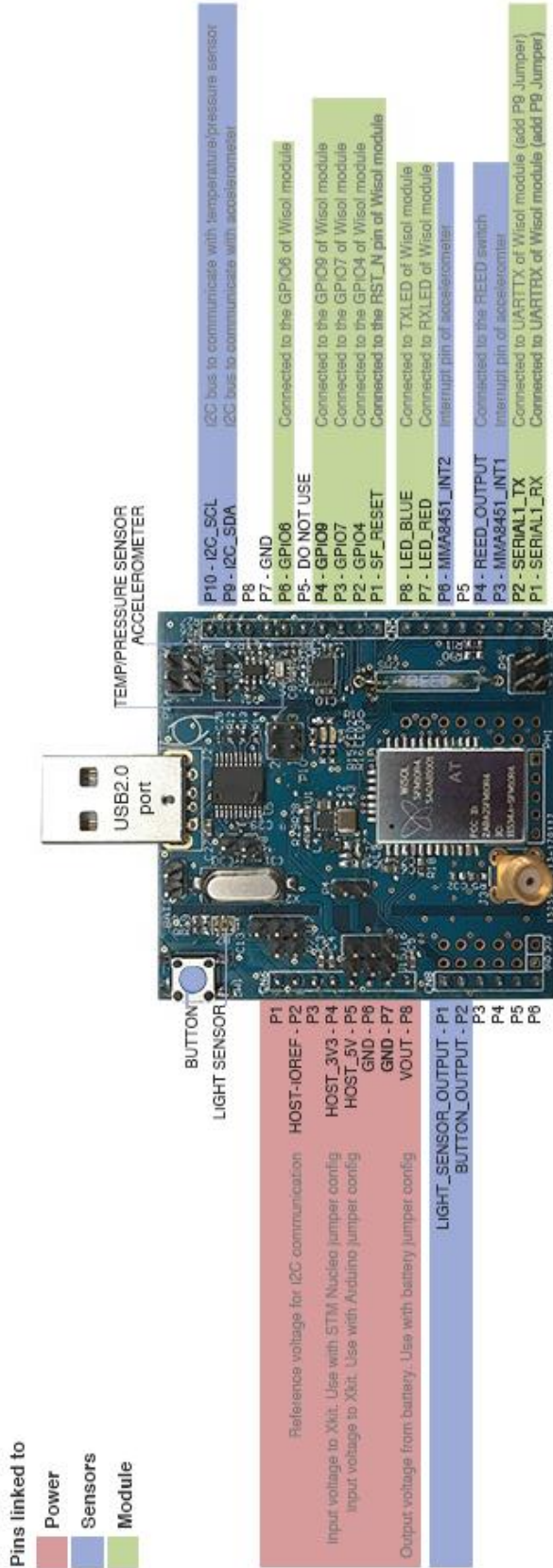
STOP Absolute max 200mA for entire package



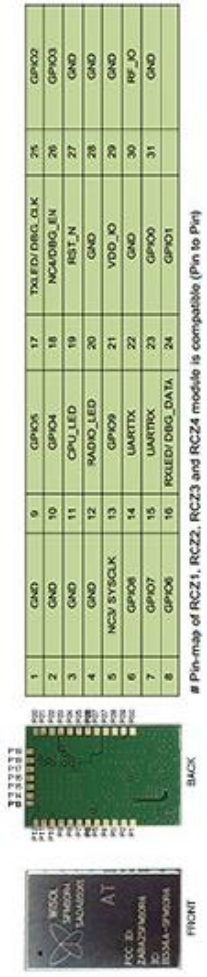
ANEXO II

Diagrama de pines del módulo Thinxtra DevKit de Sigfox y módulo Wisol TX/RX.

Thinxtra Xkit pin-out configuration



Wisol module pin-out configuration



ORDEN DE EMPASTADO